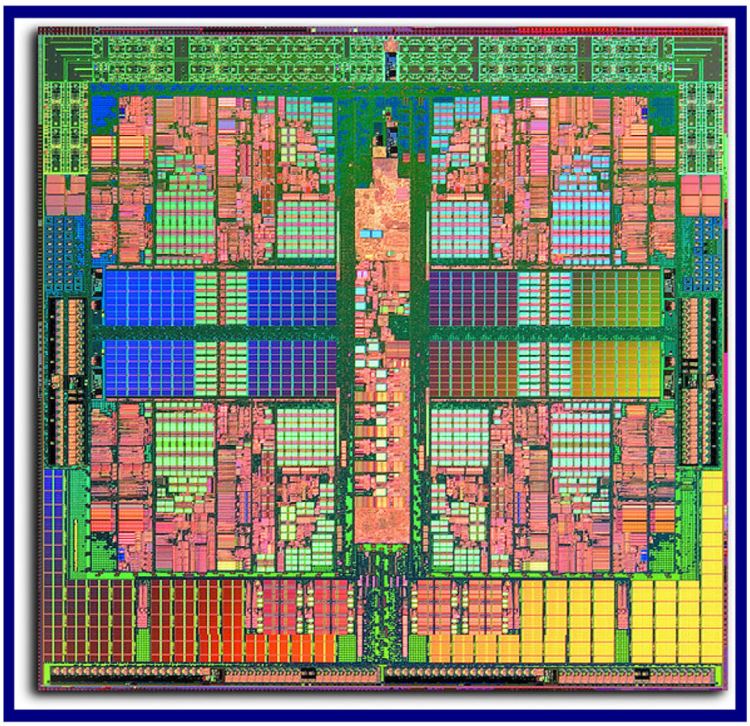
Μεταπτυχιακή Εξειδίκευση στα
Πληροφοριακά Συστήματα

Θεματική Ενότητα ΠΛΣ51

ΒΑΣΙΚΕΣ ΕΞΕΙΔΙΚΕΥΣΕΙΣ ΣΕ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

- ΣΗΜΕΙΩΣΕΙΣ ΔΙΔΑΣΚΑΛΙΑΣ -



Λάμπρος Μπισδούνης

Μέλος Σ.Ε.Π. Ελληνικού Ανοικτού Πανεπιστημίου
Καθηγητής Πανεπιστημίου Πελοποννήσου

ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ 2022

Σκοπός της θεματικής ενότητας

- Εξειδίκευση στην ανάλυση και το σχεδιασμό συνδυαστικών και ακολουθιακών ψηφιακών κυκλωμάτων, τα οποία αποτελούν τις βασικές δομικές μονάδες των ψηφιακών συστημάτων και του υλικού (hardware) των υπολογιστών.

- Εξειδίκευση στην αρχιτεκτονική και οργάνωση των υπολογιστών με έμφαση στη βασική δομή των υπολογιστών, στον προγραμματισμό μικροεπεξεργαστών, στη δομή και λειτουργία της κεντρικής μονάδας επεξεργασίας, του συστήματος μνήμης και της διασύνδεσης τους με μονάδες εισόδου-εξόδου και στη διασωλήνωση (pipelining).

- Εξειδίκευση σε βασικές έννοιες των δικτύων υπολογιστών με έμφαση σε θέματα όπως απόδοση και αρχιτεκτονική δικτύων, μοντέλα αναφοράς OSI και TCP/IP, επίπεδο ελέγχου ζεύξης δεδομένων (DLC), τοπικά δίκτυα, πρωτόκολλα διαδικτύωσης και ελέγχου μεταφοράς και εφαρμογές διαδικτύου.

Σκοπός της θεματικής ενότητας

Περιεχόμενα θεματικής ενότητας

Μέρος Α: Ψηφιακά συστήματα

Μέρος Β: Οργάνωση και αρχιτεκτονική υπολογιστών

Μέρος Γ: Δίκτυα υπολογιστών

Μέρος Α: Ψηφιακά συστήματα

- Κεφάλαιο 1: αναλογικά και ψηφιακά σήματα, μετατροπή σημάτων, ψηφιακά συστήματα, αριθμητικά συστήματα, παράσταση αριθμητικών δεδομένων με έμφαση στο δυαδικό σύστημα, εκτέλεση αριθμητικών πράξεων, αριθμοί κινητής υποδιαστολής, δυαδικοί κώδικες και κώδικες ανίχνευσης και διόρθωσης λαθών.
- Κεφάλαιο 2: άλγεβρα Boole (ορισμοί, αξιώματα, θεωρήματα, ιδιότητες), τρόποι περιγραφής λογικών παραστάσεων και συναρτήσεων, αντιστοίχιση βασικών λογικών συναρτήσεων σε λογικές πύλες και υλοποίηση λογικών συναρτήσεων με λογικές πύλες.
- Κεφάλαιο 3: ελαχιστοποίηση (απλοποίηση) λογικών συναρτήσεων και κυκλωμάτων με τη μέθοδο του χάρτη Karnaugh.
- Κεφάλαιο 4: σχεδίαση (σύνθεση) & ανάλυση συνδυαστικών κυκλωμάτων, λειτουργικότητα και σχεδίαση σύνθετων συνδυαστικών κυκλωμάτων που χρησιμοποιούνται στα ψηφιακά συστήματα (κωδικοποιητές, αποκωδικοποιητές, πολυπλέκτες, αθροιστές, αφαιρέτες, συγκριτές, πολλαπλασιαστές).
- Κεφάλαιο 5: βασικά στοιχεία μνήμης (μανταλωτές, flip-flops) των ψηφιακών κυκλωμάτων, μοντέλα περιγραφής και μελέτης σύγχρονων ακολουθιακών κυκλωμάτων, σχεδίαση (σύνθεση) και ανάλυση σύγχρονων ακολουθιακών κυκλωμάτων.
- Κεφάλαιο 6: διάφορα είδη καταχωρητών και μετρητών που αποτελούν χρήσιμα δομικά στοιχεία για την υλοποίηση ψηφιακών συστημάτων.

Κεφάλαιο 1: Παράσταση ψηφιακών δεδομένων και αριθμητικά συστήματα

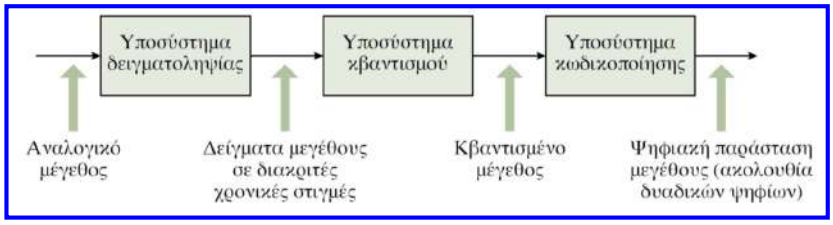
Ψηφιακά και αναλογικά σήματα

- Στις επιστήμες, την τεχνολογία, την επιχειρηματικότητα, χειριζόμαστε πολύ συχνά ποσοτικά μεγέθη, που μετρούνται, συλλέγονται, επεξεργάζονται, καταγράφονται ή αποθηκεύονται και είναι σημαντικό να παριστάνονται με αποδοτικό και ακριβή τρόπο.
- Οι βασικοί τρόποι παράστασης ποσοτικών μεγεθών είναι δύο: ο αναλογικός και ο ψηφιακός και η μορφή της πληροφορίας λέγεται αναλογικό ή ψηφιακό σήμα, αντίστοιχα.
- Παραδείγματα: επιλογές του διακόπτη που ελέγχει το «μάτι» ηλεκτρικής κουζίνας (ψηφιακό), ρεύμα που διαρρέει ηλεκτρικό κύκλωμα (αναλογικό), θερμοκρασία χώρου (αναλογικό), πάπιες σε λίμνη (ψηφιακό).
- Στην αναλογική παράσταση, ένα μέγεθος λαμβάνει συνεχείς τιμές, ενώ στην ψηφιακή παράσταση λαμβάνει διακριτές τιμές.
- Τα περισσότερα μεγέθη που μετρούνται ποσοτικά, όπως θερμοκρασία, πίεση, ταχύτητα, ένταση ηλεκτρικού ρεύματος κ.ά. εμφανίζονται στη φύση με αναλογική μορφή και επειδή τα ψηφιακά συστήματα χειρίζονται ψηφιακά σήματα, απαιτούνται κατάλληλα συστήματα μετατροπής, τα οποία επιτελούν τη μετατροπή ενός μεγέθους που παριστάνεται με αναλογικό τρόπο σε ψηφιακή μορφή.

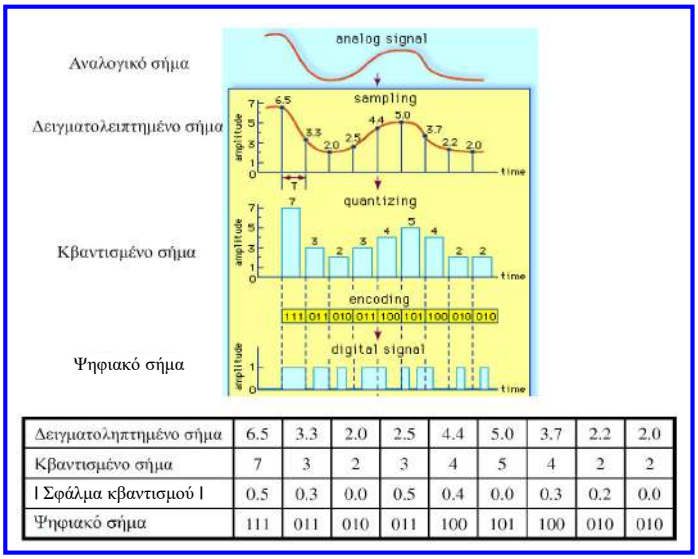
Μετατροπή αναλογικών σημάτων σε ψηφιακά

- Αναλογική παράσταση μεγέθους (αναλογικό σήμα): γραφική παράσταση ως προς το χρόνο.
- Δειγματοληψία (sampling): λήψη δειγμάτων του μεγέθους σε διακριτές χρονικές στιγμές.
- Η αναπαράσταση της τιμής κάθε δείγματος με έναν αριθμό πεπερασμένου πλήθους ψηφίων, αποτελεί την ψηφιοποίηση (digitization), αποτέλεσμα της οποίας είναι η ψηφιακή παράσταση του μεγέθους, δηλαδή μια απλή ακολουθία αριθμών που αναπαριστά το μέγεθος των διαδοχικών δειγμάτων (ψηφιακό σήμα).
- Η ψηφιοποίηση προϋποθέτει αρχικά τη διαίρεση της κλίμακας του μεγέθους σε επίπεδα (στάθμες), ώστε κάθε δείγμα του μεγέθους να αντιστοιχιστεί στο πλησιέστερο επίπεδο, διαδικασία που αναφέρεται ως κβαντισμός (quantization).
- Η διαδικασία της ψηφιοποίησης ολοκληρώνεται με την κωδικοποίηση (encoding, coding), κατά την οποία κάθε επίπεδο κβαντισμού και κατά συνέπεια κάθε δείγμα που αντιστοιχεί στο επίπεδο αυτό, κωδικοποιείται με μία απλή ακολουθία αριθμών.
- Συνήθως, ακολουθείται η δυαδική κωδικοποίηση, στην οποία χρησιμοποιούνται δυαδικά ψηφία (binary digits ή bits) που μπορούν να λάβουν δύο τιμές 0 και 1.
- Για τη δυαδική κωδικοποίηση 2^N τιμών ενός μεγέθους, απαιτούνται N δυαδικά ψηφία.
- Η παράσταση που προκύπτει από τον κβαντισμό (κβαντισμένο σήμα) και κατά συνέπεια η ψηφιακή παράσταση (ψηφιακό σήμα), δεν αποδίδουν με ακρίβεια το αναλογικό μέγεθος. Το σφάλμα που εισάγεται αναφέρεται ως σφάλμα κβαντισμού.

Μετατροπή αναλογικών σημάτων σε ψηφιακά



Θεώρημα Nyquist-Shannon: η συχνότητα δειγματοληψίας ενός αναλογικού περιοδικού σήματος πρέπει να είναι τουλάχιστον διπλάσια από τη μέγιστη συχνότητα που εμφανίζει το σήμα, για να μην εισάγεται αλλοίωση στην υπάρχουσα πληροφορία του σήματος



Δειγματοληπτημένο σήμα	6.5	3.3	2.0	2.5	4.4	5.0	3.7	2.2	2.0
Κβαντισμένο σήμα	7	3	2	3	4	5	4	2	2
| Σφάλμα κβαντισμού |	0.5	0.3	0.0	0.5	0.4	0.0	0.3	0.2	0.0
Ψηφιακό σήμα	111	011	010	011	100	101	100	010	010

Μετατροπή αναλογικών σημάτων σε ψηφιακά

- Η δειγματοληψία ενός αναλογικού σήματος $x_a(t)$ επιτυγχάνεται λαμβάνοντας δείγματα αυτού ανά Τ δευτερόλεπτα: $x = x_\alpha(nT)$.
- x είναι το σήμα διακριτού χρόνου που προκύπτει από την δειγματοληψία, T το χρονικό διάστημα μεταξύ διαδοχικών δειγμάτων (περίοδος δειγματοληψίας) και n ο αριθμός των δειγμάτων. $F = 1 / T$: ρυθμός ή συχνότητα δειγματοληψίας.
- Ο κβαντισμός είναι μη αντιστρέψιμη διαδικασία. Στον ομοιόμορφο κβαντισμό η απόσταση (διαφορά) Δ μεταξύ δύο επιπέδων κβαντισμού, δηλαδή των προκαθορισμένων τιμών στις οποίες στρογγυλοποιούνται οι τιμές του δειγματοληπτημένου σήματος, είναι σταθερή.
- Σφάλμα κβαντισμού: διαφορά μεταξύ αρχικής (x) και κβαντισμένης τιμής (x_q). Η απόλυτη τιμή του περιορίζεται (μέγιστη τιμή) στο μισό της διαφοράς μεταξύ 2 επιπέδων κβαντισμού.

$$e_q = x - x_q$$
$$-\frac{\Delta}{2} \le e_q \le \frac{\Delta}{2}$$

- Η αύξηση του πλήθους των επιπέδων κβαντισμού (L) οδηγεί στην μείωση του σφάλματος.
- Τύποι ομοιόμορφου κβαντιστή: μέσου πατήματος (mid-tread), μέσης ανύψωσης (mid-rise).

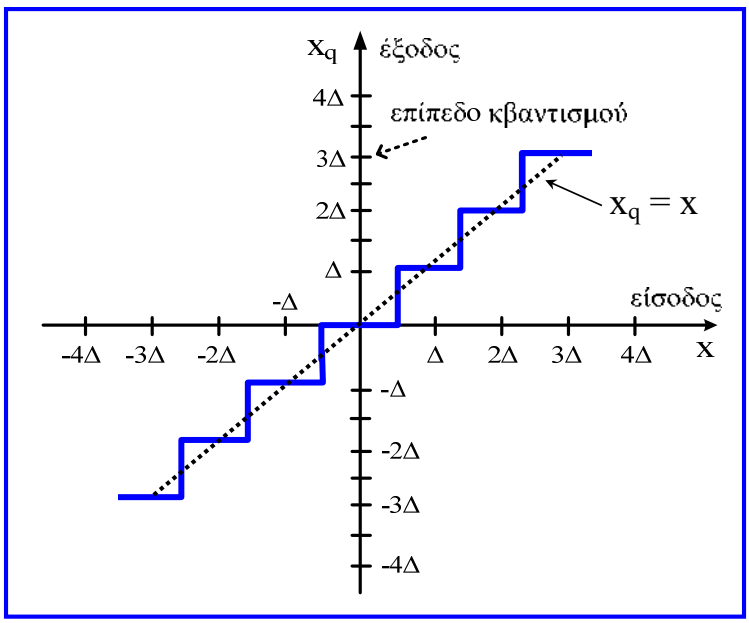
Μετατροπή αναλογικών σημάτων σε ψηφιακά

Κβαντιστής μέσου πατήματος (mid-tread): παρέχει μηδενική έξοδο, αφού η αρχή των αξόνων βρίσκεται στο μέσο οριζόντιου τμήματος της χαρακτηριστικής μεταφοράς.

$$x_q = \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor \cdot \Delta$$

$$\Delta = \frac{x_{max} - x_{min}}{L-1} = \frac{R}{L-1}$$

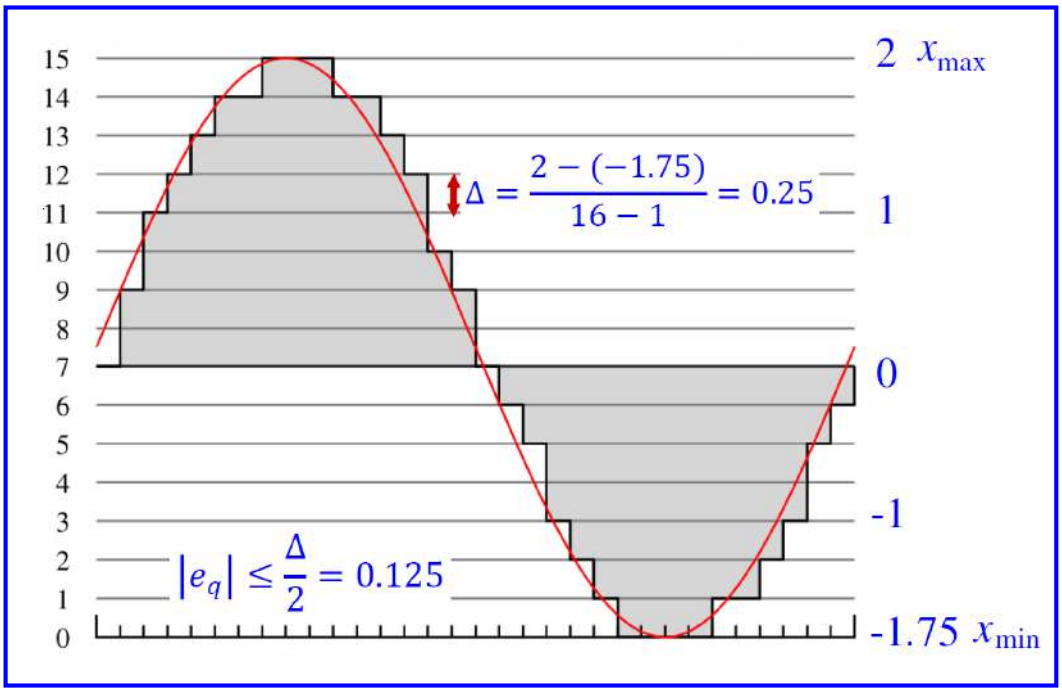
R: περιοχή κβαντιστή



Χαρακτηριστική μεταφοράς κβαντιστή μέσου πατήματος

Μετατροπή αναλογικών σημάτων σε ψηφιακά

Παράδειγμα κβαντισμού σήματος με κβαντιστή μέσου πατήματος (mid-tread)



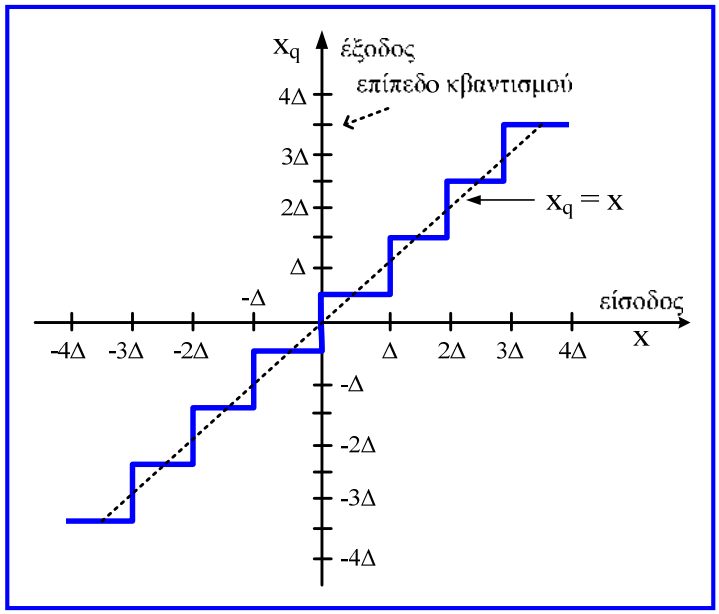
Μετατροπή αναλογικών σημάτων σε ψηφιακά

Κβαντιστής μέσης ανύψωσης (mid-rise): δεν παρέχει μηδενική έξοδο, αφού η αρχή των αξόνων βρίσκεται στο μέσο κατακόρυφου τμήματος της χαρακτηριστικής μεταφοράς.

$$x_q = \left\lfloor \frac{x}{\Delta} \right\rfloor \cdot \Delta + \frac{\Delta}{2}$$

$$\Delta = \frac{x_{max} - x_{min}}{L} = \frac{R}{L}$$

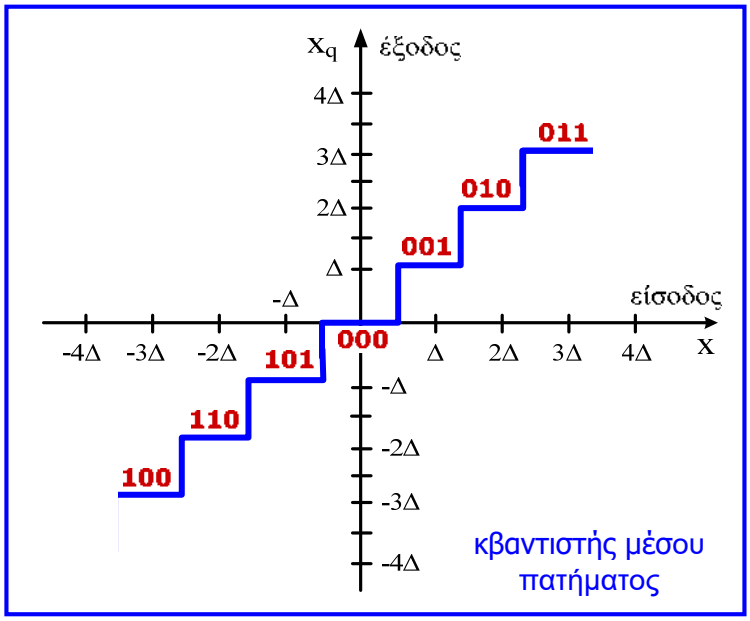
R: περιοχή κβαντιστή



Χαρακτηριστική μεταφοράς κβαντιστή μέσης ανύψωσης

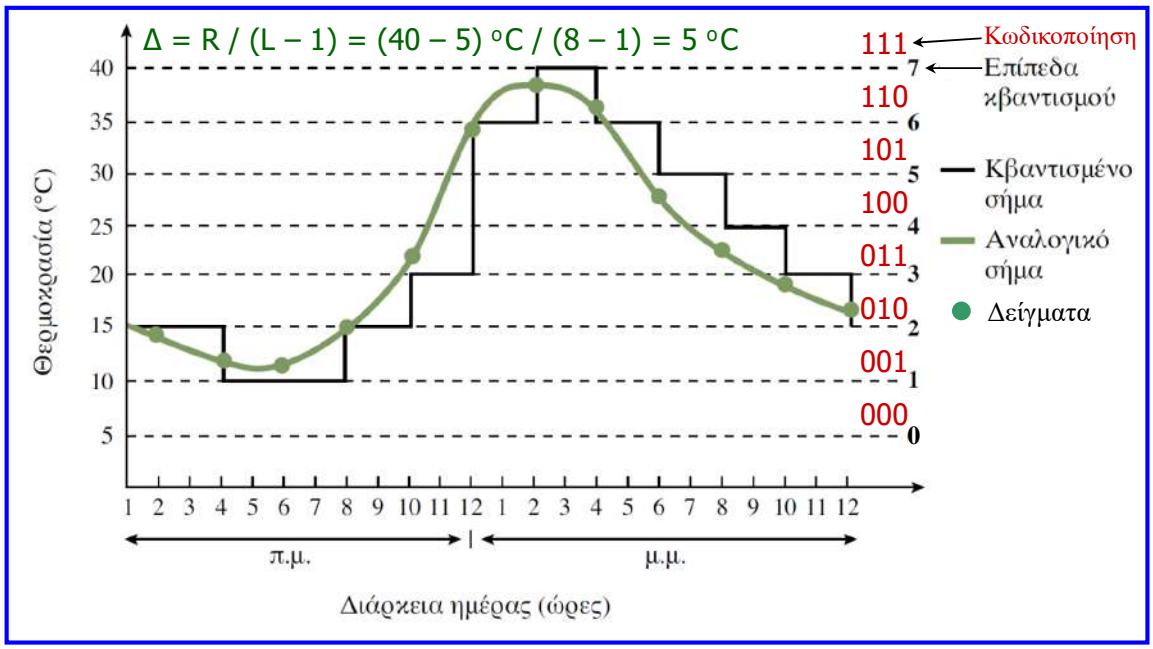
Μετατροπή αναλογικών σημάτων σε ψηφιακά

- Κωδικοποίηση: απεικόνιση κάθε επιπέδου κβαντισμού σε έναν συνδυασμό Ν δυαδικών ψηφίων (bits).
- Με Ν δυαδικά ψηφία μπορούν να σχηματιστούν 2^N διαφορετικοί συνδυασμοί, συνεπώς για L επίπεδα κβαντισμού θα πρέπει $2^N \geq L \Rightarrow N \geq \log_2 L$.



Για κωδικοποίηση με Ν δυαδικά ψηφία, ο κβαντιστής μέσου πατήματος επιτρέπει τη χρήση $2^N - 1$ διαφορετικών συνδυασμών, ενώ ο κβαντιστής μέσης ανύψωσης επιτρέπει τη χρήση 2^N διαφορετικών συνδυασμών.

Μετατροπή αναλογικών σημάτων σε ψηφιακά



Ακολουθία ψηφιακού σήματος: 010 001 001 010 011 110 111 110 101 100 011 010

Ψηφιακά συστήματα

- Η πρόοδος στον τομέα των ηλεκτρονικών και της πληροφορικής, έχει οδηγήσει στην ανάπτυξη ψηφιακών συστημάτων και υπηρεσιών που χρησιμοποιούνται σε διάφορα είδη υπολογιστών και σε διάφορες εφαρμογές (κινητή τηλεφωνία, ευρυζωνικά δίκτυα, τηλεόραση, συστήματα αναπαραγωγής ήχου, φωτογραφικές μηχανές, κάμερες, πλατφόρμες ηλεκτρονικών παιχνιδιών, ιατρική κ.ά.)
- Τα ψηφιακά συστήματα επεξεργάζονται μεγέθη ή δεδομένα που παριστάνονται με ψηφιακό τρόπο (δηλαδή λαμβάνουν διακριτές τιμές σε μία ορισμένη περιοχή τιμών).
- Η ψηφιακή επεξεργασία αφορά μετασχηματισμούς των δεδομένων που το ψηφιακό σύστημα λαμβάνει στις εισόδους του, το αποτέλεσμα των οποίων παρέχεται στις εξόδους του συστήματος.
- Στα ψηφιακά συστήματα τα διακριτά στοιχεία πληροφορίας παριστάνονται με ποσότητες (ψηφιακά σήματα) τα οποία έχουν τη μορφή διακριτών τιμών ηλεκτρικής τάσης.
- Τα ψηφιακά σήματα είναι συνήθως δυαδικά σήματα ηλεκτρικής τάσης, χρησιμοποιούν, δηλαδή, δύο διακριτές τιμές ή καταστάσεις, οι οποίες αναφέρονται και ως στάθμες.
- Η χαμηλή και η υψηλή στάθμη ενός δυαδικού σήματος αντιστοιχούν σε δύο διακριτές τιμές τάσης (π.χ. 0 και 3.3 Volt, αντίστοιχα), με τις ενδιάμεσες τιμές να είναι πρακτικά μη υπαρκτές. Οι δύο στάθμες εκφράζονται με δύο λογικές τιμές, τη λογική τιμή 0 και τη λογική τιμή 1, αντίστοιχα.

Ψηφιακά συστήματα

- Τα ψηφιακά συστήματα συνίστανται σε ψηφιακά ηλεκτρονικά κυκλώματα, στα οποία οι βασικές πράξεις μεταξύ δυαδικών σημάτων επιτελούνται από απλά δομικά στοιχεία που αναφέρονται ως λογικές πύλες (logic gates).
- Οι λογικές πύλες ανταποκρίνονται στις δύο προαναφερθείσες στάθμες τάσης και υλοποιούνται με διασυνδεδεμένα ηλεκτρονικά στοιχεία που ονομάζονται τρανζίστορ (transistor).
- Το σημαντικότερο πλεονέκτημα των στοιχείων αυτών, σε σχέση με το χειρισμό δυαδικών σημάτων, είναι ότι έχουν τη δυνατότητα να λειτουργούν ως διακόπτες, παρουσιάζοντας δύο επιτρεπτές καταστάσεις λειτουργίας αντίστοιχες με τις λογικές τιμές 0 και 1.
- Οι βασικοί λόγοι για τους οποίους χρησιμοποιούνται τα ψηφιακά συστήματα στην πλειονότητα των σύγχρονων εφαρμογών, έναντι των αναλογικών συστημάτων, είναι η υψηλή αξιοπιστία και ακρίβεια που παρέχουν κατά την επεξεργασία, αποθήκευση και μεταφορά δεδομένων, η ευελιξία και το χαμηλό κόστος κατασκευής τους.
- Το βασικό μειονέκτημα των ψηφιακών συστημάτων είναι η ανάγκη για μετατροπή της κατά φύση αναλογικής μορφής των μεγεθών σε ψηφιακή, που οδηγεί σε πρόσθετη καθυστέρηση, πολυπλοκότητα και κόστος.

Αριθμητικά συστήματα

- Το δεκαδικό σύστημα αποτελεί το πιο οικείο αριθμητικό σύστημα, αφού χρησιμοποιείται σε πολλές δραστηριότητες (μέτρηση διάφορων μεγεθών, οι εμπορικές συναλλαγές κ.ά.).
- Ωστόσο, έχουν αναπτυχθεί και άλλα αριθμητικά συστήματα, η γνώση των οποίων είναι αναγκαία για την κατανόηση και την ανάλυση της λειτουργίας των ψηφιακών συστημάτων.
- Η αναγκαιότητα αυτή προκύπτει από το γεγονός ότι τα ψηφιακά συστήματα σχεδιάζονται ώστε να λειτουργούν με δυαδικά σήματα, συνεπώς τα αριθμητικά συστήματα που χρησιμοποιούνται στο εσωτερικό τους θα πρέπει να είναι συμβατά με τον τρόπο λειτουργίας τους και την τεχνολογία με την οποία αναπτύσσονται.
- Για να είναι δυνατή η υιοθέτηση του οικείου δεκαδικού αριθμητικού συστήματος στα ψηφιακά κυκλώματα και συστήματα, θα έπρεπε να χρησιμοποιηθούν ηλεκτρονικά στοιχεία με δέκα καταστάσεις λειτουργίας, ώστε να μπορούν να παρασταθούν όλα τα αναγκαία ψηφία.
- Ωστόσο, λόγω του ότι δεν είναι διαθέσιμα τέτοιου είδους στοιχεία, αλλά έχουν αναπτυχθεί ηλεκτρονικά στοιχεία όπως τα τρανζίστορ με δύο καταστάσεις λειτουργίας, το δυαδικό αριθμητικό σύστημα έχει πρωταρχική σημασία και χρησιμότητα για τους σχεδιαστές ψηφιακών κυκλωμάτων και συστημάτων.
- Αριθμητικά συστήματα, όπως το οκταδικό και το δεκαεξαδικό, χρησιμοποιούνται, επίσης, κατά το σχεδιασμό και την ανάλυση ψηφιακών συστημάτων.

Αριθμητικά συστήματα

- Βασικό διακριτικό χαρακτηριστικό των αριθμητικών συστημάτων είναι ένας φυσικός αριθμός μεγαλύτερος του 1 που αναφέρεται ως βάση (base ή radix) του συστήματος.
- Τα αριθμητικά συστήματα λαμβάνουν το όνομά τους από τον αριθμό που αποτελεί τη βάση τους. Έτσι, η βάση του δεκαδικού, του δυαδικού, του οκταδικού και του δεκαεξαδικού συστήματος είναι ο αριθμός 10, 2, 8 και 16, αντίστοιχα.
- Η βάση ενός συστήματος καθορίζεται από το πλήθος των ψηφίων ή συμβόλων που μπορούν να χρησιμοποιηθούν σε αυτό.
- Τα ψηφία αυτά είναι φυσικοί αριθμοί μικρότεροι από τη βάση ή σύμβολα που αντιστοιχούν σε φυσικούς αριθμούς μικρότερους από τη βάση του συστήματος:
  - ✓ στο δυαδικό σύστημα χρησιμοποιούνται τα ψηφία 0 και 1,
  - ✓ στο οκταδικό τα ψηφία από 0 έως 7,
  - ✓ στο δεκαδικό σύστημα τα ψηφία 0 έως 9
  - ✓ και στο δεκαεξαδικό σύστημα τα ψηφία 0 έως 9, καθώς και τα πρώτα έξι γράμματα του λατινικού αλφαβήτου (A, B, C, D, E, F) που αντιστοιχούν κατά σειρά στους δεκαδικούς αριθμούς 10 έως 15.

Αριθμητικά συστήματα

- Οι αριθμοί των αριθμητικών συστημάτων παριστάνονται με βάση τη σχέση:

$$(a_{n-1}a_{n-2}\ldots a_2a_1a_0.a_{-1}a_{-2}\ldots a_{-m})_r = \sum_{-m}^{n-1} a_i \times r^i =$$
$$a_{n-1}\times r^{n-1} + a_{n-2}\times r^{n-2} + \ldots + a_1 \times r^1 + a_0 \times r^0 + a_{-1}\times r^{-1} + \ldots + a_{-m}\times r^{-m}$$

r βάση του αριθμητικού συστήματος

a_i ψηφία του αριθμού που λαμβάνουν τιμές μεταξύ 0 και $r - 1$

n πλήθος των ακέραιων ψηφίων του αριθμού

m πλήθος των κλασματικών ψηφίων του αριθμού,

i υποδεικνύει τη θέση ή τάξη του ψηφίου a_i

- Κάθε ψηφίο, ανάλογα με τη θέση στην οποία βρίσκεται, κατέχει διαφορετική βαρύτητα. Για το λόγο αυτόν, τα εν λόγω αριθμητικά συστήματα αναφέρονται ως θεσιακά (positional) αριθμητικά συστήματα.
- Στα συστήματα αυτά η τιμή των αριθμών υπολογίζεται με βάση την τιμή και τη θέση κάθε ψηφίου, ενώ, αντιθέτως, στα μη θεσιακά συστήματα, όπως, για παράδειγμα, το ρωμαϊκό σύστημα, τα ψηφία ή τα σύμβολα που χρησιμοποιούνται (π.χ. I, II, III, IV) έχουν την ίδια τιμή ανεξάρτητα από τη θέση που κατέχουν στον αριθμό.

Αριθμητικά συστήματα

- Το πρώτο από αριστερά ψηφίο ενός αριθμού αναφέρεται ως περισσότερο σημαντικό ψηφίο (most significant digit, MSD) ή περισσότερο σημαντικό δυαδικό ψηφίο (most significant bit, MSB), όταν πρόκειται για δυαδικό αριθμό, ενώ το τελευταίο αναφέρεται ως λιγότερο σημαντικό ψηφίο (least significant digit, LSD) ή λιγότερο σημαντικό δυαδικό ψηφίο (least significant bit, LSB), όταν πρόκειται για δυαδικό αριθμό.
- Στο δεκαδικό σύστημα, για να παρασταθούν αριθμοί μεγαλύτεροι του μεγαλύτερου επιτρεπτού ψηφίου (δηλαδή του 9), απαιτούνται περισσότερα του ενός ψηφία (δύο ψηφία για αριθμούς μεταξύ του 10 και του 99, τρία ψηφία για αριθμούς μεταξύ του 100 και του 999, κ.ο.κ).
- Γενικεύοντας, προκύπτει ότι για την παράσταση έως 10^n δεκαδικών αριθμών απαιτούνται n δεκαδικά ψηφία, για την παράσταση έως 2^n δυαδικών αριθμών απαιτούνται n δυαδικά ψηφία και για την παράσταση έως 8^n οκταδικών αριθμών και έως 16^n δεκαεξαδικών αριθμών απαιτούνται n οκταδικά και δεκαεξαδικά ψηφία, αντίστοιχα.
- Ο μεγαλύτερος αριθμός που μπορεί να παρασταθεί με n δυαδικά ψηφία είναι εκείνος που αποτελείται από n μονάδες και η αντίστοιχη δεκαδική τιμή είναι:

$$1 \times 2^{n-1} + 1 \times 2^{n-2} + \ldots + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 2^n - 1.$$

- Παρομοίως, προκύπτει ότι ο μεγαλύτερος αριθμός που μπορεί να παρασταθεί με n ψηφία σε ένα αριθμητικό σύστημα με βάση r, είναι ο $r^n - 1$.

Δυαδικοί, οκταδικοί και δεκαεξαδικοί αριθμοί

Οι πρώτοι 16 δεκαδικοί, δυαδικοί, οκταδικοί και δεκαεξαδικοί αριθμοί, έχουν ως εξής:

Δεκαδικό	Δυαδικό	Οκταδικό	Δεκαεξαδικό
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Μετατροπή αριθμού βάσης r σε δεκαδικό αριθμό

$$(a_{n-1}a_{n-2}\ldots a_2a_1a_0.a_{-1}a_{-2}\ldots a_{-m})_r = \sum_{-m}^{n-1} a_i \times r^i =$$
$$a_{n-1}\times r^{n-1} + a_{n-2}\times r^{n-2} + \ldots + a_1 \times r^1 + a_0 \times r^0 + a_{-1}\times r^{-1} + \ldots + a_{-m}\times r^{-m}$$

$(110.101)_2$

$$1\times 2^2 + 1\times 2^1 + 0\times 2^0 + 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} =$$
$$4 + 2 + 0 + 0.5 + 0 + 0.125 = (6.625)_{10}$$

$(B43.5C8)_{16}$

$$11\times 16^2 + 4\times 16^1 + 3\times 16^0 + 5\times 16^{-1} + 12\times 16^{-2} + 8\times 16^{-3} =$$
$$2816 + 64 + 3 + 0.3125 + 0.046875 + 0.001953125 = (2883.36132812)_{10}$$

Μετατροπή δεκαδικού αριθμού σε αριθμό βάσης r

- Για τη μετατροπή ενός δεκαδικού αριθμού στον ισοδύναμό του αριθμό που παριστάνεται σε σύστημα με βάση r, διαφορετική από το 10, ακολουθούμε τα παρακάτω βήματα:
  - ✓ Το ακέραιο μέρος του δεκαδικού αριθμού διαιρείται επαναληπτικά με τη βάση r, μέχρι να μηδενιστεί το πηλίκο που προκύπτει. Το υπόλοιπο της πρώτης διενεργηθείσας διαίρεσης αποτελεί το λιγότερο σημαντικό ψηφίο και το υπόλοιπο της τελευταίας διαίρεσης αποτελεί το περισσότερο σημαντικό ψηφίο της παράστασης του ακέραιου μέρους του αριθμού στο σύστημα με βάση r.
  - ✓ Το κλασματικό μέρος πολλαπλασιάζεται επαναληπτικά με τη βάση r, μέχρι να προκύψει ως γινόμενο ένας αριθμός με μηδενικό κλασματικό μέρος. Αυτό, ωστόσο, δεν είναι πάντα δυνατό, δηλαδή μπορεί η μετατροπή να είναι ατελής. Στην περίπτωση αυτή, το κλασματικό μέρος θα περιλαμβάνει τον αριθμό ψηφίων που καθορίζεται από την επιθυμητή ακρίβεια. Το ακέραιο μέρος του γινομένου του πρώτου διενεργηθέντος πολλαπλασιασμού αποτελεί το περισσότερο σημαντικό ψηφίο και το ακέραιο μέρος του τελευταίου πολλαπλασιασμού αποτελεί το λιγότερο σημαντικό ψηφίο της παράστασης του κλασματικού μέρους του αριθμού στο σύστημα με βάση r.
- Η μετατροπή αριθμών μεταξύ συστημάτων με βάση διαφορετική του 10 μπορεί να επιτευχθεί εύκολα με τη χρησιμοποίηση του δεκαδικού συστήματος ως ενδιάμεσο στάδιο.

Μετατροπή δεκαδικού αριθμού σε αριθμό βάσης r

Μετατροπή του $(39.84375)_{10}$ σε δυαδικό αριθμό:

39 / 2 = πηλίκο 19 + υπόλοιπο 1 (LSB)
19 / 2 = πηλίκο 9 + υπόλοιπο 1
9 / 2 = πηλίκο 4 + υπόλοιπο 1
4 / 2 = πηλίκο 2 + υπόλοιπο 0
2 / 2 = πηλίκο 1 + υπόλοιπο 0
1 / 2 = πηλίκο 0 + υπόλοιπο 1 (MSB)

$0.84375 \times 2 = 1.6875 = 0.6875 + 1$ (MSB)
$0.6875 \times 2 = 1.375 = 0.375 + 1$
$0.375 \times 2 = 0.75 = 0.75 + 0$
$0.75 \times 2 = 1.5 = 0.5 + 1$
$0.5 \times 2 = 1 = 0.0 + 1$ (LSB)

$(100111.11011)_2$

Μετατροπή του $(345.158)_{10}$ σε οκταδικό αριθμό:

345 / 8 = πηλίκο 43 + υπόλοιπο 1 (LSB)
43 / 8 = πηλίκο 5 + υπόλοιπο 3
5 / 8 = πηλίκο 0 + υπόλοιπο 5 (MSB)

$0.158 \times 8 = 1.264 = 0.264 + 1$ (MSB)
$0.264 \times 8 = 2.112 = 0.112 + 2$
$0.112 \times 8 = 0.896 = 0.896 + 0$
$0.896 \times 8 = 7.168 = 0.168 + 7$, κ.ο.κ.

$(531.1207\ldots)_8$

Μετατροπή δεκαδικού αριθμού σε αριθμό βάσης r

Μετατροπή του **$(7400.125)_{10}$** σε δεκαεξαδικό αριθμό:

7400 / 16 = πηλίκο 462 + υπόλοιπο 8 (LSB)
462 / 16 = πηλίκο 28 + υπόλοιπο 14 (E στο δεκαεξαδικό σύστημα)
28 / 16 = πηλίκο 1 + υπόλοιπο 12 (C στο δεκαεξαδικό σύστημα)
1 / 16 = πηλίκο 0 + υπόλοιπο 1 (MSB)

$0.125 \times 16 = 2.0 = 0.0 + 2$ (MSB)

$(1CE8.2)_{16}$

Μετατροπή του **$(1354.24)_6$** σε τετραδικό αριθμό:

$$(1354.24)_6 = 1 \times 6^3 + 3 \times 6^2 + 5 \times 6^1 + 4 \times \cdot 6^0 + 2 \times 6^{-1} + 4 \times \cdot 6^{-2} = (358.4444)_{10}$$

358 / 4 = πηλίκο 89 + υπόλοιπο 2 (LSB)
89 / 4 = πηλίκο 22 + υπόλοιπο 1
22 / 4 = πηλίκο 5 + υπόλοιπο 2
5 / 4 = πηλίκο 1 + υπόλοιπο 1
1 / 4 = πηλίκο 0 + υπόλοιπο 1 (MSB)

$0.4444 \times 4 = 1.7776 = 0.7776 + 1$ (MSB)
$0.7776 \times 4 = 3.1104 = 0.1104 + 3$
$0.1104 \times 4 = 0.4414 = 0.4414 + 0$
$0.4414 \times 4 = 1.7656 = 0.7656 + 1$ (LSB)

$(11212.1301)_4$

Μετατροπή μεταξύ δυαδικού και 8δικού ή 16δικού συστήματος

- Από τις σχέσεις $2^3 = 8$ και $2^4 = 16$, προκύπτει ότι για την παράσταση των ψηφίων του οκταδικού συστήματος ή για την παράσταση των ψηφίων του δεκαεξαδικού συστήματος απαιτούνται τρία ή τέσσερα δυαδικά ψηφία, αντίστοιχα.

- Αφού, σε κάθε οκταδικό ψηφίο αντιστοιχούν τρία δυαδικά ψηφία και σε κάθε δεκαεξαδικό ψηφίο αντιστοιχούν τέσσερα δυαδικά ψηφία, η μετατροπή ενός δυαδικού αριθμού στον ισοδύναμό του οκταδικό ή δεκαεξαδικό προϋποθέτει τον τεμαχισμό του σε ομάδες των τριών ή τεσσάρων ψηφίων, αντίστοιχα, ξεκινώντας από την υποδιαστολή και προχωρώντας προς τα αριστερά για τη μετατροπή του ακέραιου μέρους και προς τα δεξιά για τη μετατροπή του κλασματικού μέρους.

- Οι ομάδες των λιγότερο και των περισσότερο σημαντικών ψηφίων του κλασματικού και του ακέραιου μέρους, αντίστοιχα, συμπληρώνονται ανάλογα, ώστε να αριθμούν τρία ή τέσσερα ψηφία.

- Η μετατροπή ενός οκταδικού ή δεκαεξαδικού αριθμού στον ισοδύναμό του δυαδικό αριθμό συνίσταται στην αντικατάσταση κάθε οκταδικού ή δεκαεξαδικού ψηφίου με τον αντίστοιχο τριψήφιο ή τετραψήφιο δυαδικό αριθμό.

Μετατροπή μεταξύ δυαδικού και 8δικού ή 16δικού συστήματος

Μετατροπή του **$(10110001101011.1111)_2$** σε οκταδικό και δεκαεξαδικό αριθμό:

010	110	001	101	011 .	111	100
2	6	1	5	3	7	4

0010	1100	0110	1011 .	1111
2	C	6	B	F

$$(10110001101011.1111)_2 = (26153.74)_8 = (2C6B.F)_{16}$$

Μετατροπή του **$(306.D)_{16}$** σε δυαδικό αριθμό:

3	0	6	D
0011	0000	0110 .	1101

$$(306.D)_{16} = (001100000110.1101)_2$$

Μετατροπή βάσης αριθμών: σύνοψη

- Μετατροπή από αριθμητικό σύστημα βάσης r στο δεκαδικό σύστημα: Πολλαπλασιάζουμε τους συντελεστές με τις αντίστοιχες δυνάμεις του r και προσθέτουμε τα γινόμενα.
- Μετατροπή από δεκαδικό σύστημα σε σύστημα βάσης r:
  - ✓ Διακρίνουμε ακέραιο και κλασματικό μέρος.
  - ✓ Διαιρούμε το ακέραιο μέρος συνεχώς με τη βάση r και κρατάμε τα υπόλοιπα (με αντίστροφη παράθεση).
  - ✓ Πολλαπλασιάζουμε το κλασματικό μέρος συνεχώς με τη βάση r και κρατάμε τα ακέραια μέρη (με ορθή παράθεση).
- Η μετατροπή αριθμών μεταξύ συστημάτων με βάση διαφορετική του 10 μπορεί να επιτευχθεί με τη χρησιμοποίηση του δεκαδικού συστήματος ως ενδιάμεσο στάδιο.
- Μετατροπή από οκταδικό ή δεκαεξαδικό σε δυαδικό σύστημα: Αντικαθιστούμε κάθε ψηφίο με τον αντίστοιχο τριψήφιο ή τετραψήφιο δυαδικό αριθμό.
- Μετατροπή από δυαδικό σε οκταδικό ή δεκαεξαδικό σύστημα: Ομαδοποιούμε τα δυαδικά ψηφία σε τριάδες ή τετράδες και τις αντικαθιστούμε με το αντίστοιχο ψηφίο του οκταδικού ή δεκαεξαδικού συστήματος.

Προσημασμένοι δυαδικοί αριθμοί

- Οποιαδήποτε πληροφορία στα ψηφιακά συστήματα πρέπει να παριστάνεται με χρήση δυαδικών ψηφίων και μόνο. Αυτό, θα πρέπει να εφαρμοστεί όσον αφορά την παράσταση θετικών και αρνητικών δυαδικών αριθμών και κατά συνέπεια για τη μεταξύ τους διάκριση.

- Ακολουθώντας τον τρόπο παράστασης που χρησιμοποιείται στην κοινή αριθμητική, μπορούμε να παραστήσουμε τους προσημασμένους δυαδικούς αριθμούς, έτσι ώστε να αποτελούνται από ένα δυαδικό ψηφίο που υποδηλώνει το πρόσημο του αριθμού και από μία ακολουθία δυαδικών ψηφίων που συνιστά το μέγεθος ή μέτρο ή τιμή του αριθμού.

- Σε αυτό τον τρόπο παράστασης προσημασμένων αριθμών, που αναφέρεται ως παράσταση προσημασμένου μεγέθους (signed magnitude) ή προσήμου-μέτρου έχει καθοριστεί συμβατικά η παράσταση του θετικού προσήμου με το δυαδικό ψηφίο 0 και η παράσταση του αρνητικού προσήμου με το δυαδικό ψηφίο 1, καθώς και ότι το ψηφίο-πρόσημο καταλαμβάνει την περισσότερο σημαντική θέση του προσημασμένου αριθμού.

- Ωστόσο, η χρήση ενός δυαδικού ψηφίου ως προσήμου μειώνει το εύρος των θετικών τιμών που μπορούν να παρασταθούν με συγκεκριμένο πλήθος δυαδικών ψηφίων. Έτσι, με n δυαδικά ψηφία δεν μπορούν πλέον να παρασταθούν οι θετικοί αριθμοί από 0 έως $2^n - 1$, αλλά οι θετικοί αριθμοί από 0 έως $2^{n-1} - 1$.

Προσημασμένοι δυαδικοί αριθμοί

- Για την υλοποίηση πράξεων με προσημασμένους αριθμούς στα ψηφιακά συστήματα είναι καταλληλότερη η χρήση της παράστασης προσημασμένου συμπληρώματος (signed complement).

- Για τον καθορισμό του συμπληρώματος ενός αριθμού χρησιμοποιείται ως αναφορά η βάση του αριθμητικού συστήματος ή η βάση μειωμένη κατά 1.

- Το συμπλήρωμα ως προς 1 ενός δυαδικού αριθμού A ορίζεται ως $(2^n - 1) - A$, ενώ το συμπλήρωμα ως προς 2 του ίδιου αριθμού ορίζεται ως $2^n - A$.

- Γενικεύοντας, το συμπλήρωμα ως προς τη βάση r μειωμένη κατά 1 ενός αριθμού A με n ψηφία ορίζεται ως $(r^n - 1) - A$, ενώ το συμπλήρωμα ως προς τη βάση r ορίζεται ως $r^n - A$.

- Το συμπλήρωμα του συμπληρώματος ενός αριθμού επαναφέρει τον αριθμό στην αρχική του τιμή, αφού το συμπλήρωμα ως προς r του συμπληρώματος ως προς r ενός αριθμού A με n ψηφία είναι: $r^n - (r^n - A) = A$.

- Το συμπλήρωμα ως προς τη βάση προκύπτει με πρόσθεση μιας μονάδας στο συμπλήρωμα ως προς τη βάση μειωμένη κατά 1.

- Το συμπλήρωμα ως προς 9 του δεκαδικού αριθμού 23567 είναι 99999 – 23567 = 76432, ενώ το συμπλήρωμα ως προς 10 του ίδιου αριθμού είναι 76432 + 1 = 76433.

Προσημασμένοι δυαδικοί αριθμοί

- Ο αριθμός 2^n - 1 στο δυαδικό σύστημα είναι ο μεγαλύτερος που μπορεί να παρασταθεί με n δυαδικά ψηφία και προφανώς αποτελείται από n μονάδες.
- Το συμπλήρωμα ως προς 1 ενός αριθμού προκύπτει, εάν αφαιρέσουμε κάθε ψηφίο του από το 1 και αφού 1 - 0 = 1 και 1 - 1 = 0, το συμπλήρωμα ως προς 1 ενός δυαδικού αριθμού προκύπτει εάν αντικαταστήσουμε τα μηδενικά του αριθμού με μονάδες και τις μονάδες με μηδενικά.
- Ο αριθμός 2^n στο δυαδικό σύστημα μπορεί να παρασταθεί με n + 1 ψηφία και συνίσταται από μία μονάδα ακολουθούμενη από n μηδενικά.
- Το συμπλήρωμα ως προς 2 ενός αριθμού n δυαδικών ψηφίων προκύπτει, εάν αφαιρέσουμε τον αριθμό αυτόν από τον αριθμό 2^n. Η ενέργεια αυτή ισοδυναμεί με το να διατηρήσουμε τα συνεχόμενα λιγότερο σημαντικά μηδενικά και την πρώτη μονάδα του αριθμού και να αντικαταστήσουμε τις μονάδες με μηδενικά και τα μηδενικά με μονάδες στις υπόλοιπες πιο σημαντικές θέσεις του αριθμού.
- Με βάση τα προαναφερθέντα, προκύπτει ότι τα συμπληρώματα ως προς 1 και 2 του αριθμού 1101100 είναι 0010011 και 0010100, αντίστοιχα.
- Σε περίπτωση ύπαρξης υποδιαστολής, αυτή δε λαμβάνεται υπόψη κατά τον υπολογισμό των συμπληρωμάτων και διατηρεί την αρχική της θέση και στα συμπληρώματα.

Προσημασμένοι δυαδικοί αριθμοί

- Κατά την παράσταση προσημασμένου συμπληρώματος, οι θετικοί αριθμοί n δυαδικών ψηφίων παριστάνονται με μηδενικό πρόσημο και τιμή που καθορίζεται από τα n - 1 λιγότερο σημαντικά ψηφία, ενώ οι αρνητικοί αριθμοί μπορούν να παρασταθούν με το συμπλήρωμα ως προς 1 ή 2 των αντίστοιχων θετικών αριθμών.
- Από τον ορισμό των συμπληρωμάτων προκύπτει ότι αφού το περισσότερο σημαντικό ψηφίο των θετικών αριθμών είναι πάντα 0, το αντίστοιχο ψηφίο στους αρνητικούς αριθμούς είναι πάντα 1, παρέχοντας διάκριση θετικών και αρνητικών αριθμών.
- Οι θετικοί αριθμοί είναι ίδιοι και στους τρεις τρόπους παράστασης προσημασμένων αριθμών.
- Στις παραστάσεις προσημασμένου μεγέθους και προσημασμένου συμπληρώματος ως προς 1, ο αριθμός μηδέν μπορεί να παρασταθεί με δύο τρόπους.
- Ο μεγαλύτερος θετικός αριθμός που μπορεί να παρασταθεί και με τους τρεις τρόπους είναι ο 2^{n-1} - 1 (για n δυαδικά ψηφία) και ο μικρότερος αρνητικός αριθμός είναι ο αριθμός - 2^{n-1} + 1, με εξαίρεση την παράσταση προσημασμένου συμπληρώματος ως προς 2, στην οποία μπορεί να παρασταθεί και ο αρνητικός αριθμός - 2^{n-1}.

Προσημασμένοι δυαδικοί αριθμοί

Δεκαδική παράσταση	Παράσταση προσημασμένου μεγέθους	Παράσταση προσημασμένου συμπληρώματος ως προς 1	Παράσταση προσημασμένου συμπληρώματος ως προς 2
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
0	0000 ή 1000	0000 ή 1111	0000
−1	1001	1110	1111
−2	1010	1101	1110
−3	1011	1100	1101
−4	1100	1011	1100
−5	1101	1010	1011
−6	1110	1001	1010
−7	1111	1000	1001
−8	–	–	1000

Πρόσθεση και αφαίρεση δυαδικών αριθμών

- Στις αριθμητικές πράξεις με αριθμούς συστημάτων βάσης διαφορετικής από το 10, ισχύουν οι ίδιοι κανόνες με αυτούς που χρησιμοποιούνται για τους δεκαδικούς αριθμούς, αλλά θα πρέπει σε κάθε σύστημα να χρησιμοποιούνται τα επιτρεπτά ψηφία και μόνο.
- Η πρόσθεση δύο δυαδικών αριθμών εκτελείται ανά ζεύγη ψηφίων της ίδιας θέσης, ξεκινώντας από το ζεύγος των λιγότερο σημαντικών ψηφίων των αριθμών.
- Εάν προκύπτει κρατούμενο ψηφίο μετά την πρόσθεση σε κάποια θέση, τότε αυτό προστίθεται στα ψηφία της αμέσως πιο σημαντικής θέσης.
- Οι δυνατές περιπτώσεις πρόσθεσης δύο δυαδικών ψηφίων είναι οι εξής: 0 + 0 = 0, 0 + 1 = 1 + 0 = 1 και 1 + 1 = 0 με κρατούμενο ψηφίο 1 (δηλαδή 10 που αντιστοιχεί στο δεκαδικό αριθμό 2). Από την πρόσθεση τριών μονάδων προκύπτει άθροισμα 1 και κρατούμενο 1 (δηλαδή 11 που αντιστοιχεί στο δεκαδικό αριθμό 3).
- Για την αφαίρεση δυαδικών αριθμών μπορούν να ακολουθηθούν οι κανόνες που χρησιμοποιούνται στο δεκαδικό σύστημα, με τη διαφορά ότι όταν προκύπτει δανειζόμενο ψηφίο από την αφαίρεση σε μία θέση (δηλαδή όταν εκτελείται η αφαίρεση 0 − 1 = 1 και απαιτείται δανειζόμενο ψηφίο 1), θα πρέπει να προσθέσουμε 2 (αντί για 10, όπως απαιτείται στο δεκαδικό σύστημα) στο ψηφίο του μειωτέου.
- Η μέθοδος του δανειζόμενου ψηφίου δεν είναι αποδοτική, όσον αφορά την υλοποίηση της πράξης της αφαίρεσης στα ψηφιακά συστήματα, για την οποία είναι καταλληλότερη η χρήση του συμπληρώματος ως προς 2 του αφαιρετέου.

Πρόσθεση και αφαίρεση δυαδικών αριθμών

	1	1	1	1		1		← Κρατούμενα από προηγούμενη θέση
		1	0	1	1	0	1	$(45)_{10}$
+		1	1	1	1	0	1	$(61)_{10}$
	1	1	0	1	0	1	0	$(106)_{10}$

	10	10	10	10	1	10		← Κρατούμενη ποσότητα προηγούμενης θέσης
		1	0	1	1	0	1	$(45)_{10}$
		1	1	0	1	0	1	$(53)_{10}$
		0	0	1	1	0	1	$(13)_{10}$
+		0	1	0	0	0	1	$(17)_{10}$
1	0	0	0	0	0	0	0	$(128)_{10}$

Πρόσθεση και αφαίρεση δυαδικών αριθμών

- Για την εκτέλεση της αφαίρεσης δύο μη προσημασμένων αριθμών n δυαδικών ψηφίων, προσθέτουμε στον μειωτέο (M) το συμπλήρωμα ως προς 2 του αφαιρετέου (A), δηλαδή εκτελούμε την πράξη: $M + (2^n - A) = M - A + 2^n$.
- Λόγω του ότι ο αριθμός 2^n στο δυαδικό σύστημα συνίσταται από μία μονάδα ακολουθούμενη από n μηδενικά, εάν $M \geq A$, στο αποτέλεσμα της πρόσθεσης προκύπτει μονάδα στην πιο σημαντική θέση (τελικό κρατούμενο), η οποία θα πρέπει να αγνοηθεί.
- Εάν $M < A$, το αποτέλεσμα της πρόσθεσης θα είναι: $2^n - (A - M)$, δηλαδή το συμπλήρωμα ως προς 2 του $A - M$.
- Στην περίπτωση αυτή, η διαφορά $M - A$ είναι αρνητικός αριθμός και η τιμή του προκύπτει με τον υπολογισμό του συμπληρώματος ως προς 2 του αποτελέσματος της πρόσθεσης του μειωτέου με το συμπλήρωμα ως προς 2 του αφαιρετέου.
- Αφαίρεση 77 – 23 = 54 στο δυαδικό σύστημα:

$(23)_{10} = 0010111$, συμπλήρωμα ως προς 2: 1101001

	1	0	0	1	1	0	1	$(77)_{10}$
+	1	1	0	1	0	0	1	Συμπλήρωμα ως προς 2 του $(23)_{10}$
1	0	1	1	0	1	1	0	

- Λόγω του ότι ο μειωτέος είναι μεγαλύτερος του αφαιρετέου, αγνοούμε το τελικό κρατούμενο και το αποτέλεσμα της αφαίρεσης είναι: 0110110 = $(54)_{10}$.

Πρόσθεση και αφαίρεση δυαδικών αριθμών

- Αφαίρεση 23 – 77 = – 54 στο δυαδικό σύστημα:

$(77)_{10} = 1001101$, συμπλήρωμα ως προς 2: 0110011

	0	0	1	0	1	1	1	$(23)_{10}$
+	0	1	1	0	0	1	1	Συμπλήρωμα ως προς 2 του $(77)_{10}$
	1	0	0	1	0	1	0	

- Λόγω του ότι ο μειωτέος είναι μικρότερος του αφαιρετέου, η διαφορά τους είναι αρνητικός αριθμός, το μέγεθος (τιμή) του οποίου είναι το συμπλήρωμα ως προς 2 του αποτελέσματος της διενεργηθείσας πρόσθεσης: 0110110 = $(54)_{10}$.
- Κατά την πρόσθεση προσημασμένων δυαδικών αριθμών σε παράσταση προσημασμένου μεγέθους, θα πρέπει αρχικά να διενεργείται σύγκριση των προσήμων των αριθμών.
- Εάν οι αριθμοί έχουν ίδιο πρόσημο, προσθέτουμε τα μεγέθη των αριθμών και διατηρούμε το πρόσημο.
- Εάν οι αριθμοί έχουν διαφορετικό πρόσημο, τότε διενεργούμε σύγκριση του μεγέθους των αριθμών, αφαιρούμε το μικρότερο από το μεγαλύτερο μέγεθος και διατηρούμε το πρόσημο του αριθμού με το μεγαλύτερο μέγεθος.
- Η απαιτούμενη σύγκριση προσήμων και μεγεθών των αριθμών αποτελεί μειονέκτημα της χρήσης της παράστασης προσημασμένου μεγέθους.

Πρόσθεση και αφαίρεση δυαδικών αριθμών

- Η χρήση της παράστασης προσημασμένου συμπληρώματος είναι απλούστερη και καταλληλότερη για τα ψηφιακά συστήματα.
- Οι προσημασμένοι δυαδικοί αριθμοί που παριστάνονται με μορφή προσημασμένου συμπληρώματος ως προς 2, προστίθενται (συμπεριλαμβανομένου του ψηφίου που κατέχει θέση προσήμου), χωρίς προηγούμενη επεξεργασία.
- Εάν στο αποτέλεσμα της πρόσθεσης, το οποίο, επίσης, παριστάνεται με μορφή προσημασμένου συμπληρώματος ως προς 2, προκύψει τελικό κρατούμενο (δηλαδή κρατούμενο στην πιο σημαντική θέση), αγνοείται.
- Η πράξη της αφαίρεσης προσημασμένων αριθμών ανάγεται σε πρόσθεση του μειωτέου με τον αντίθετο αριθμό του αφαιρετέου, ο οποίος μπορεί να παρασταθεί με το συμπλήρωμα ως προς 2 του αφαιρετέου.

A = 0011 = $(+3)_{10}$

B = 1100 = $(-4)_{10}$

	0	0	1	1	***A***		0	0	1	1	***A***
+	1	1	0	0	***B***	+	0	1	0	0	***–B***
	1	1	1	1	$(-1)_{10}$		0	1	1	1	$(+7)_{10}$
	1	1	0	1	***–A***		1	1	0	1	***–A***
+	1	1	0	0	***+B***	+	0	1	0	0	***–B***
✗	1	0	0	1	$(-7)_{10}$	✗	0	0	0	1	$(+1)_{10}$

Υπερχείλιση

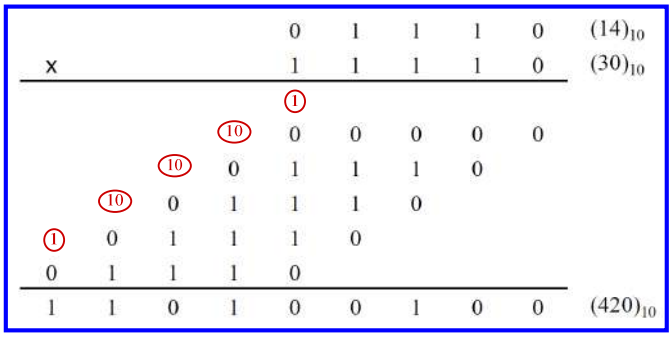
- Κατά την πρόσθεση ή αφαίρεση δύο προσημασμένων αριθμών n δυαδικών ψηφίων, υπάρχει πιθανότητα το αποτέλεσμα να απαιτεί για την ορθή παράστασή του n + 1 ψηφία.
- Στην περίπτωση αυτή συμβαίνει μετατόπιση του ψηφίου-προσήμου από την αναμενόμενη θέση και η κατάσταση αυτή αναφέρεται ως υπερχείλιση (overflow).
- Η υπερχείλιση αποτελεί πρόβλημα στα ψηφιακά συστήματα στα οποία το πλήθος των θέσεων μνήμης όπου αποθηκεύονται οι αριθμοί είναι συγκεκριμένο, με συνέπεια να μην μπορεί να αποθηκευτεί το αποτέλεσμα μιας πράξης που το πλήθος των ψηφίων του υπερβαίνει το πλήθος των διαθέσιμων θέσεων μνήμης.
- Στους ψηφιακούς επεξεργαστές, το πρόβλημα της υπερχείλισης αντιμετωπίζεται συνήθως με χρήση ειδικής θέσης μνήμης, με βάση το περιεχόμενο της οποίας υποδεικνύεται εάν υπήρξε υπερχείλιση στην πράξη που εκτελέστηκε.
- Υπερχείλιση κατά την πρόσθεση δύο προσημασμένων αριθμών συμβαίνει όταν οι αριθμοί είναι ομόσημοι και το αποτέλεσμα που προκύπτει έχει διαφορετικό πρόσημο από τους αριθμούς που προστίθενται.
- Αυτό ισοδυναμεί με το να είναι διαφορετικά τα κρατούμενα ψηφία που προκύπτουν κατά την πρόσθεση στις δύο πιο σημαντικές θέσεις

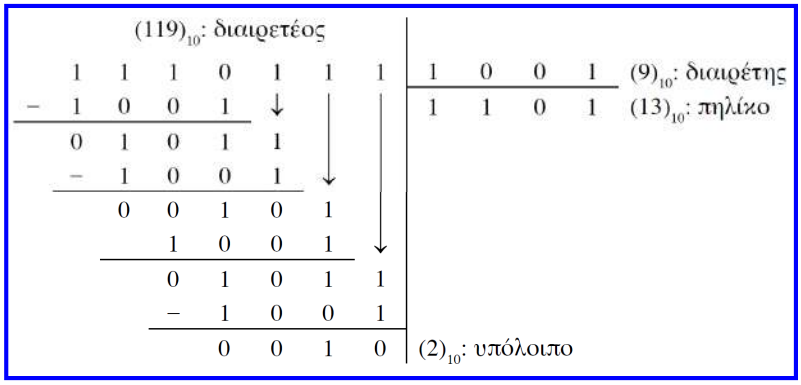
Υπερχείλιση

- Κατά την πρόσθεση των θετικών δυαδικών αριθμών 0111 = $(+7)_{10}$ και 0100 = $(+4)_{10}$ προκύπτει λανθασμένο αρνητικό αποτέλεσμα, δηλαδή 1011 = $(-5)_{10}$, λόγω της υπερχείλισης.
- Αυτό συμβαίνει διότι το ορθό αποτέλεσμα 01011 = $(+11)_{10}$ είναι μεγαλύτερο από το μεγαλύτερο δυνατό θετικό αριθμό που μπορεί να παρασταθεί με τέσσερα δυαδικά ψηφία, δηλαδή τον αριθμό $(+7)_{10}$.
- Επίσης, κατά την πρόσθεση των αρνητικών δυαδικών αριθμών 1100 = $(-4)_{10}$ και 1010 = $(-6)_{10}$ προκύπτει λανθασμένο θετικό αποτέλεσμα, δηλαδή 0110 = $(+6)_{10}$, λόγω της υπερχείλισης.
- Αυτό συμβαίνει διότι το ορθό αποτέλεσμα 10110 = $(-10)_{10}$ είναι μικρότερο από το μικρότερο δυνατό αρνητικό αριθμό που μπορεί να παρασταθεί με τέσσερα δυαδικά ψηφία, δηλαδή το $(-8)_{10}$.

Πολλαπλασιασμός και διαίρεση δυαδικών αριθμών

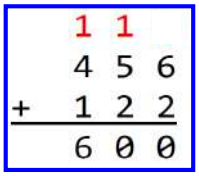
Στο δυαδικό σύστημα οι πράξεις του πολλαπλασιασμού και της διαίρεσης ανάγονται σε διαδοχικές προσθέσεις και διαδοχικές συγκρίσεις / αφαιρέσεις, αντίστοιχα.

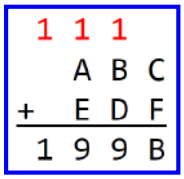




Πράξεις αριθμών στο οκταδικό & δεκαεξαδικό σύστημα

- Στο οκταδικό σύστημα εκτελούμε την πρόσθεση ανά ψηφίο και όταν προκύπτει άθροισμα S > 7, το άθροισμα ισούται με [S – 8 (βάση)] και το κρατούμενο με 1 (μία οκτάδα).
- Παρομοίως, εκτελούμε την πρόσθεση στο δεκαεξαδικό σύστημα, όπου γνωρίζουμε ότι βάση είναι το 16 και ότι για τα ψηφία άνω του 9 χρησιμοποιούνται τα λατινικά γράμματα A, B, C, D, E, F.
- Εκτελούμε δηλαδή την πρόσθεση ανά ψηφίο και όταν προκύπτει άθροισμα S > 15, το άθροισμα ισούται με (S – 16) και το κρατούμενο με 1 (μία δεκαεξάδα).





- Για την αφαίρεση, προσθέτουμε στον μειωτέο το συμπλήρωμα ως προς τη βάση (8 ή 16) του αφαιρετέου, τηρώντας αντίστοιχους κανόνες με αυτούς που αναφέρθηκαν για τους δυαδικούς αριθμούς.

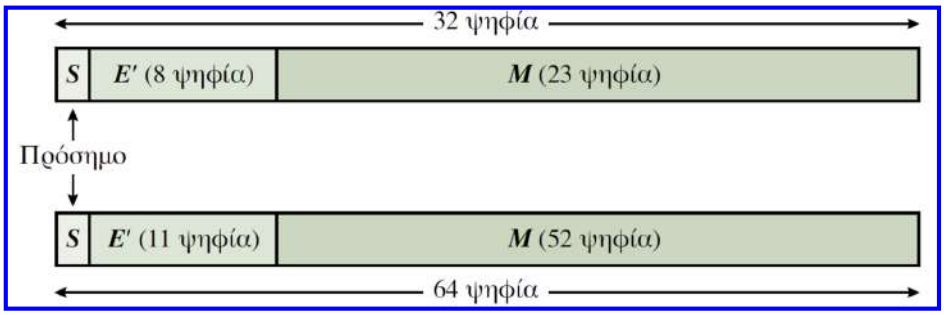
Αριθμοί κινητής υποδιαστολής

- Έως τώρα παρουσιάστηκαν τρόποι παράστασης και πράξεις για ακέραιους αριθμούς ή για αριθμούς που περιλαμβάνουν κλασματικό μέρος, στους οποίους η θέση της υποδιαστολής είναι προκαθορισμένη και αναφέρονται ως αριθμοί σταθερής υποδιαστολής (fixed point numbers).
- Το μειονέκτημα των αριθμών σταθερής υποδιαστολής είναι το σχετικά μικρό εύρος τιμών που μπορεί να παρασταθεί με αυτούς.
- Στα ψηφιακά υπολογιστικά συστήματα είναι απαραίτητος ένας πιο αποδοτικός τρόπος παράστασης πολύ μεγάλων ακέραιων και πολύ μικρών κλασματικών αριθμών.
- Για παράδειγμα, ενώ ο μεγαλύτερος ακέραιος αριθμός που μπορεί να παρασταθεί με 32 δυαδικά ψηφία είναι ο αριθμός $2^{32} - 1$, σε επιστημονικούς υπολογισμούς όπου χρησιμοποιούνται σταθερές όπως ο αριθμός Avogadro (6.022×10^{23} $mole^{-1}$), το παρεχόμενο εύρος τιμών δεν είναι επαρκές.
- Παρομοίως, το παρεχόμενο εύρος τιμών των αριθμών σταθερής υποδιαστολής δεν είναι επαρκές για πολύ μικρούς κλασματικούς αριθμούς, όπως, για παράδειγμα, η σταθερά Boltzmann (1.38×10^{-23} Joule/^{o}K) που χρησιμοποιείται στη χημεία και την επιστήμη των υλικών.
- Για να ικανοποιηθεί, λοιπόν, η ανάγκη παράστασης πολύ μεγάλων ακέραιων και πολύ μικρών κλασματικών αριθμών, στα ψηφιακά υπολογιστικά συστήματα έχει υιοθετηθεί η παράσταση αριθμών κινητής υποδιαστολής (floating point numbers).

Αριθμοί κινητής υποδιαστολής

- Στην παράσταση αυτή, η θέση της υποδιαστολής των αριθμών δεν είναι προκαθορισμένη αλλά μεταβλητή, ώστε να προσαρμόζεται στις υπολογιστικές ανάγκες.
- Οι αριθμοί κινητής υποδιαστολής περιλαμβάνουν τρία τμήματα: το πρόσημο, τον εκθέτη, που στην ουσία καθορίζει τη θέση της υποδιαστολής, και το κλασματικό μέρος.
- Το πρότυπο παράστασης αριθμών κινητής υποδιαστολής IEEE 754, χρησιμοποιεί 32 (παράσταση απλής ακρίβειας) ή 64 (παράσταση διπλής ακρίβειας) δυαδικά ψηφία.
- Η τιμή του ψηφίου-προσήμου (S) για τους θετικούς αριθμούς είναι 0, ενώ για τους αρνητικούς είναι 1.
- Στην παράσταση απλής ακρίβειας, στον εκθέτη αντιστοιχούν τα 8 που ακολουθούν μετά το πρόσημο, ενώ στο κλασματικό μέρος αντιστοιχούν τα 23 λιγότερο σημαντικά ψηφία.
- Στην παράσταση αυτή, αντί για τον προσημασμένο εκθέτη E, παριστάνεται ένας μη προσημασμένος αριθμός E′, τέτοιος ώστε $E' = E + 127$, ο οποίος λαμβάνει τιμές στο διάστημα 0 έως 255.
- Η παράσταση διπλής ακρίβειας περιλαμβάνει 64 ψηφία, 11 από τα οποία, εκτός του ψηφίου-προσήμου (S), αντιστοιχούν στον εκθέτη (E) και 52 στο κλασματικό μέρος (M).
- Ο αριθμός E′ είναι τέτοιος ώστε $E' = E + 1023$ και λαμβάνει τιμές στο διάστημα 0 έως 2047.

Αριθμοί κινητής υποδιαστολής



- Η κωδικοποίηση που χρησιμοποιείται στον εκθέτη, αναφέρεται ως παράσταση πλεονάσματος, κατά την οποία προστίθεται στην τιμή του εκθέτη ο αριθμός $2^{n-1} - 1$ (n = πλήθος ψηφίων εκθέτη). Ο αριθμός $2^{n-1} - 1$ αναφέρεται ως πόλωση.
- Η τιμή ενός αριθμού κινητής υποδιαστολής απλής ακρίβειας είναι $\pm 1.M \times 2^{E'-127}$, ενώ η αντίστοιχη τιμή αριθμού διπλής ακρίβειας είναι $\pm 1.M \times 2^{E'-1023}$. Οι ακραίες τιμές του E′ χρησιμοποιούνται για την παράσταση ειδικών τιμών, όπως 0 (E′ = M = 0), ∞ (E′ = 255, M = 0), NaN ή «Not-A-Number» (E′ = 255, M ≠ 0).
- Το ψηφίο που βρίσκεται αριστερά της υποδιαστολής είναι πάντοτε 1 και δεν αναπαρίσταται, αλλά θεωρείται ότι υπάρχει στη θέση αυτή. Για το λόγο αυτόν, οι αριθμοί που πρόκειται να παρασταθούν σύμφωνα με το πρότυπο IEEE 754 θα πρέπει να διαμορφώνονται έτσι ώστε να υπάρχει μονάδα στη θέση αριστερά της υποδιαστολής, διαδικασία που αναφέρεται ως κανονικοποίηση.

Αριθμοί κινητής υποδιαστολής

- Η κανονικοποίηση προϋποθέτει τη μετατόπιση προς τα αριστερά ή δεξιά της υποδιαστολής, ώστε να είναι 1 το πιο σημαντικό ψηφίο του αριθμού που προκύπτει αριστερά της υποδιαστολής.
- Παράλληλα, για μετατόπιση της υποδιαστολής κατά μία θέση, ο εκθέτης θα πρέπει να αυξάνεται κατά μία μονάδα, όταν πρόκειται για μετατόπιση προς τα αριστερά, ή να μειώνεται αντίστοιχα, όταν πρόκειται για μετατόπιση προς τα δεξιά.
- Για την παράσταση του δεκαδικού αριθμού 47.3125, σύμφωνα με το πρότυπο IEEE 754 απλής ακρίβειας, αρχικά μετατρέπουμε το δεκαδικό αριθμό σε δυαδικό: $(47.3125)_{10} = (101111.0101)_2$.
- Ο αριθμός αυτός γράφεται ως 101111.0101×2^0, χωρίς να αλλοιωθεί η τιμή του.
- Για να κανονικοποιήσουμε τον αριθμό, μετατοπίζουμε την υποδιαστολή 5 θέσεις προς τα αριστερά, με αποτέλεσμα την αύξηση του εκθέτη κατά 5, δηλαδή: 1.011110101×2^5.
- Η τιμή του αριθμού E′ που επέχει θέση εκθέτη, σύμφωνα με το πρότυπο IEEE 754 είναι E′ = 5 + 127 = 132, με αντίστοιχο δυαδικό αριθμό τον αριθμό 10000100.

0	1 0 0 0 0 1 0 0	0 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

- Διενεργείται προσάρτηση μηδενικών στις λιγότερο σημαντικές θέσεις του κλασματικού μέρους και στις περισσότερο σημαντικές θέσεις του εκθέτη, όταν αυτό χρειάζεται για τη συμπλήρωση του πλήθους των ψηφίων του κλασματικού μέρους και του εκθέτη.

Βασικές πράξεις αριθμών κινητής υποδιαστολής

- Κατά την πρόσθεση ή αφαίρεση δύο αριθμών κινητής υποδιαστολής, αρχικά επιλέγουμε τον αριθμό με το μικρότερο εκθέτη και μετατοπίζουμε το κλασματικό του μέρος δεξιά τόσες θέσεις, όση είναι η διαφορά των εκθετών των δύο αριθμών.
- Ως εκθέτης του αποτελέσματος λαμβάνεται ο μεγαλύτερος από τους δύο εκθέτες.
- Στη συνέχεια εκτελούμε πρόσθεση ή αφαίρεση των κλασματικών μερών (μαζί με τα ψηφία που βρίσκονται αριστερά της υποδιαστολής), και καθορίζουμε το πρόσημο του αθροίσματος ή της διαφοράς.
- Κατά τον πολλαπλασιασμό δύο αριθμών κινητής υποδιαστολής εκτελούμε αρχικά πρόσθεση των εκθετών και αφαίρεση από το άθροισμα του αριθμού 127 (1023, όταν πρόκειται για αριθμούς διπλής ακρίβειας), ώστε να καθοριστεί ο εκθέτης του γινομένου.
- Στη συνέχεια, εκτελούμε πολλαπλασιασμό των κλασματικών μερών (μαζί με τα ψηφία που βρίσκονται αριστερά της υποδιαστολής) και καθορίζουμε το πρόσημο του γινομένου.
- Κατά τη διαίρεση, εκτελούμε αφαίρεση των εκθετών και πρόσθεση στη διαφορά του αριθμού 127 (1023, όταν πρόκειται για αριθμούς διπλής ακρίβειας), ώστε να καθοριστεί ο εκθέτης του πηλίκου.
- Στη συνέχεια, διαιρούμε τα κλασματικά μέρη και καθορίζουμε το πρόσημο του πηλίκου.
- Το αποτέλεσμα των τεσσάρων πράξεων και οι αρχικοί αριθμοί που συμμετέχουν σε αυτές θα πρέπει να κανονικοποιούνται.

Βασικές πράξεις αριθμών κινητής υποδιαστολής

Παράδειγμα: Εκτέλεση πράξης X + Y με X = 53.625 και Y = – 215.25

- Δυαδική παράσταση: $X = 110101.101 = 110101.101 \times 2^0$, $Y = -11010111.01 \times 2^0$.
- Κανονικοποίηση αριθμών, δηλ. μετατόπιση υποδιαστολής προς τα αριστερά τόσες θέσεις όσες απαιτούνται σε κάθε αριθμό, έτσι ώστε να είναι 1 το πιο σημαντικό ψηφίο τους, με αντίστοιχη αύξηση των εκθετών: $X = 1.10101101 \times 2^5$ και $Y = -1.101011101 \times 2^7$.
- Υπολογισμός εκθετών σύμφωνα με το πρότυπο IEEE 754 απλής ακρίβειας: $E'_X = 5 + 127 = 132$ ή 10000100, $E'_Y = 7 + 127 = 134$ ή 10000110.

X:	0	1	0	0	0	0	1	0	0	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y:	1	1	0	0	0	0	1	1	0	1	0	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

- Ολίσθηση του αριθμού με το μικρότερο εκθέτη (X) τόσες θέσεις προς τα δεξιά, όσες χρειάζονται για να εξισωθούν οι 2 εκθέτες.
- Ο μεγαλύτερος εκθέτης είναι και ο εκθέτης του αποτελέσματος των πράξεων.
- Ο αριθμός X μετά από ολίσθηση κατά 2 θέσεις δεξιά γίνεται: 0.0110101101.

Βασικές πράξεις αριθμών κινητής υποδιαστολής

- Για να εκτελέσουμε την πράξη X + Y, υπολογίζουμε το συμπλήρωμα ως προς 2 του κλασματικού μέρους του Y (= 0.010100011) και το προσθέτουμε στο κλασματικό μέρος του μετατοπισμένου X:

	0.	0	1	1	0	1	0	1	1	0	1
+	0.	0	1	0	1	0	0	0	1	1	0
	0.	1	0	1	1	1	1	0	0	1	1

- Επειδή το κλασματικό μέρος του Y είναι μεγαλύτερο από εκείνο του X, υπολογίζουμε το συμπλήρωμα ως προς 2 του αποτελέσματος και θέτουμε αρνητικό ψηφίο-πρόσημο (1) στην παράσταση κινητής υποδιαστολής του αθροίσματος:

1	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Δυαδικοί κώδικες παράστασης δεδομένων

- Τα ψηφιακά συστήματα έχουν τη δυνατότητα να χειρίζονται, εκτός από δυαδικούς αριθμούς, διακριτά στοιχεία πληροφορίας που ανήκουν σε ένα πεπερασμένο σύνολο, αφού το καθένα από αυτά μπορεί να παρασταθεί με μία ακολουθία δυαδικών ψηφίων, ακολουθώντας τους κανόνες ενός δυαδικού κώδικα.
- Κάθε διαφορετικό στοιχείο πληροφορίας θα πρέπει να αντιστοιχίζεται σε μία μοναδική ακολουθία ή συνδυασμό δυαδικών ψηφίων.
- Για την παράσταση ενός συνόλου 2^n στοιχείων πληροφορίας, όπως, για παράδειγμα, αριθμών ή γραμμάτων, απαιτείται δυαδικός κώδικας που να χρησιμοποιεί τουλάχιστον n δυαδικά ψηφία.
- Βασικοί δυαδικοί κώδικες αφορούν την παράσταση δεκαδικών αριθμών και αλφαριθμητικών χαρακτήρων, καθώς και την ανίχνευση και διόρθωσης σφαλμάτων κατά τη μετάδοση και την επεξεργασία ψηφιακών δεδομένων.
- Για την κωδικοποίηση των 10 δεκαδικών ψηφίων απαιτούνται 4 δυαδικά ψηφία, συνεπώς από τους 16 συνδυασμούς δυαδικών ψηφίων θα παραμείνουν 6 αχρησιμοποίητοι.
- Ο κώδικας που χρησιμοποιείται συχνότερα είναι ο BCD (Binary Coded Decimal) και η κωδικοποίηση σε αυτόν βασίζεται στη χρήση των συντελεστών βαρύτητας 8, 4, 2, 1 για τα τέσσερα δυαδικά ψηφία (κατά σειρά σημαντικότητας) κάθε συνδυασμού.
- Ο BCD είναι ισοδύναμος με την παράσταση δεκαδικών ψηφίων στο δυαδικό σύστημα.

Κώδικας BCD

Δεκαδικό ψηφίο	Κωδικοποίηση BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

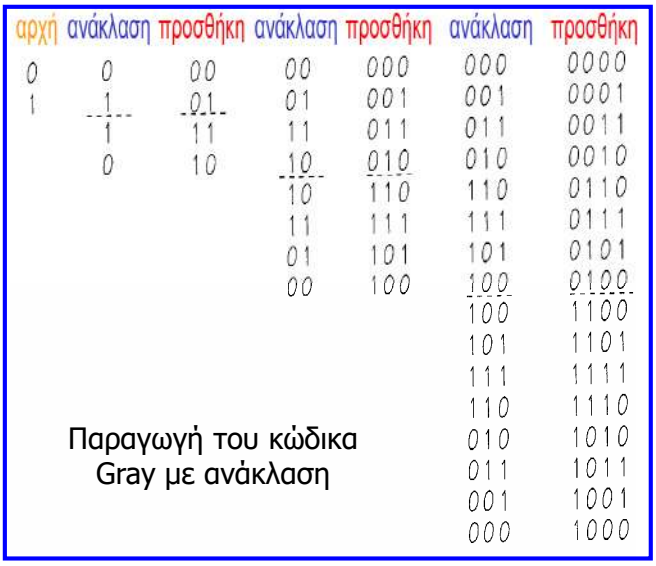
- Η παράσταση ενός δεκαδικού αριθμού στο δυαδικό σύστημα είναι διαφορετική από την παράσταση του ίδιου αριθμού σε κώδικα BCD.
- Για παράδειγμα, η παράσταση του δεκαδικού αριθμού 18 στο δυαδικό σύστημα είναι 10010 και απαιτεί 5 ψηφία, ενώ η κωδικοποίηση BCD του ίδιου αριθμού προκύπτει από τη συνένωση των κωδικοποιημένων ψηφίων 1 και 8, είναι 00011000 και απαιτεί 8 ψηφία.

Άλλοι κώδικες παράστασης δεκαδικών ψηφίων

- Με ομαδοποίηση 4 δυαδικών ψηφίων σε δέκα διαφορετικούς μεταξύ τους συνδυασμούς, μπορούν να σχηματιστούν διάφοροι δυαδικοί κώδικες των δεκαδικών ψηφίων.
- Ένας τρόπος αντιστοίχισης διαφορετικών συνδυασμών στα δεκαδικά ψηφία είναι ο καθορισμός συντελεστών βαρύτητας σε κάθε θέση των συνδυασμών.
- Έτσι, για παράδειγμα, αντί των συντελεστών βαρύτητας 8, 4, 2, 1 που χρησιμοποιούνται στον κώδικα BCD, μπορούν να χρησιμοποιηθούν συντελεστές βαρύτητας 2, 4, 2, 1 ή 8, 4, -2, -1 ή 5, 4, 2, 1 και να προκύψουν παρόμοιοι κώδικες για τα δεκαδικά ψηφία.
- Επίσης, ένας κώδικας δεκαδικών ψηφίων που χρησιμοποιούνταν σε υπολογιστές παλαιότερης γενιάς είναι ο κώδικας «υπέρβασης κατά 3» (excess 3), οι συνδυασμοί του οποίου δεν προκύπτουν με βάση συντελεστές βαρύτητας αλλά από τις αντίστοιχες δυαδικές τιμές με πρόσθεση του αριθμού 3.
- Εάν δημιουργήσετε τους συνδυασμούς του κώδικα με συντελεστές βαρύτητας 2, 4, 2, 1 και του κώδικα «υπέρβασης κατά 3», θα παρατηρήσετε ότι οι κώδικες αυτοί είναι αυτοσυμπληρωματικοί, δηλαδή το συμπλήρωμα ως προς 9 των δεκαδικών αριθμών που σχηματίζονται με βάση τους κώδικες αυτούς προκύπτει με εναλλαγή των 0 και 1.

Κώδικας Gray

- Για την παράσταση ψηφιακών δεδομένων που προέκυψαν από μετατροπή αναλογικού μεγέθους, είναι πιο ασφαλές να υιοθετείται κωδικοποίηση, στην οποία κατά τη μετάβαση μεταξύ δύο διαδοχικών αριθμών να υφίσταται αλλαγή μόνο ένα δυαδικό ψηφίο.
- Τέτοιος είναι ο κώδικας Gray, ο οποίος παρουσιάζει την ανακλαστική ιδιότητα και δεν ακολουθεί συντελεστές βαρύτητας.

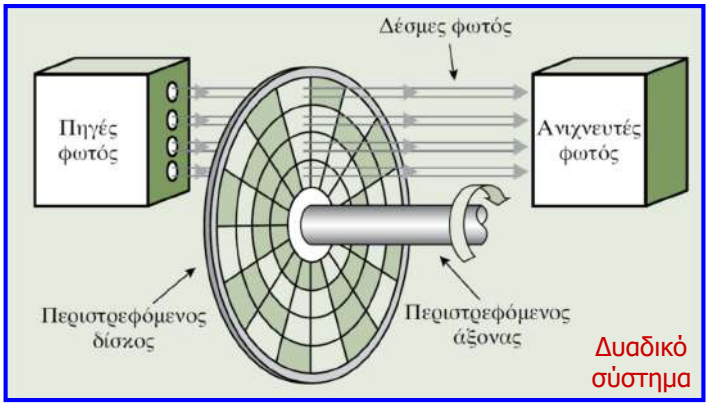


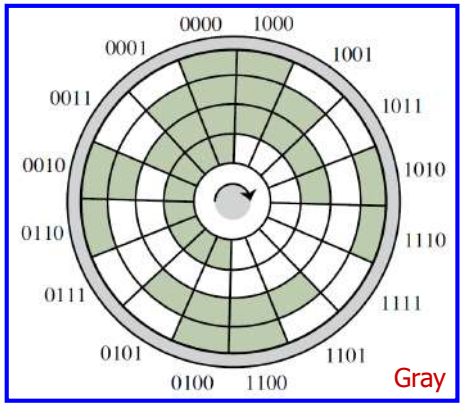
Παραγωγή του κώδικα Gray με ανάκλαση

Δεκαδικός αριθμός	Κωδικοποίηση Gray
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

Κώδικας Gray

Οπτικός κωδικοποιητής μέτρησης γωνίας περιστροφής άξονα





Τοποθέτηση παραθύρων ώστε σε κάθε τομέα να αντιστοιχεί ένας συνδυασμός δυαδικών ψηφίων (διαφανές παράθυρο = 1 και αδιαφανές παράθυρο = 0)

Οι δέσμες φωτός δεν ευθυγραμμίζονται απόλυτα και μπορούν να προκύψουν σφάλματα στα όρια των τομέων λόγω χρήσης του δυαδικού συστήματος, αφού σε πολλές περιπτώσεις γειτονικών τομέων προκαλείται ταυτόχρονη αλλαγή κατάστασης σε περισσότερα από ένα παράθυρα. Για παράδειγμα, στο όριο των τομέων που αντιστοιχούν στους αριθμούς 0001 και 0010, λόγω μη απόλυτης ευθυγράμμισης μπορεί να ανιχνευτεί εσφαλμένα 0011 ή 0000. Μεγαλύτερο σφάλμα μπορεί να προκληθεί όταν αλλάζει ταυτόχρονα η τιμή 3 ή 4 ψηφίων.

Δυαδικοί κώδικες αλφαριθμητικών χαρακτήρων

- Σε αρκετές εφαρμογές χρησιμοποιούνται εκτός από αριθμητικά δεδομένα και δεδομένα που συνίστανται από γράμματα ή σύμβολα.
- Ένας δυαδικός κώδικας αλφαριθμητικών χαρακτήρων που χρησιμοποιείται ευρύτατα, είναι ο κώδικας ASCII (American Standard Code for Information Interchange, Πρότυπος Αμερικανικός Κώδικας Ανταλλαγής Πληροφοριών).

Χρησιμοποιεί 7 ψηφία για την κωδικοποίηση 128 χαρακτήρων (94 αλφαριθμητικούς και σύμβολα) και άλλους 34 χαρακτήρες ελέγχου (μη εκτυπώσιμους) που χρησιμοποιούνται για διάφορες λειτουργίες

Παράδειγμα:

Λέξη «if» σε ASCII
1101001 1100110

$b_4\,b_3\,b_2\,b_1$	$b_7\,b_6\,b_5$ 000	$b_7\,b_6\,b_5$ 001	$b_7\,b_6\,b_5$ 010	$b_7\,b_6\,b_5$ 011	$b_7\,b_6\,b_5$ 100	$b_7\,b_6\,b_5$ 101	$b_7\,b_6\,b_5$ 110	$b_7\,b_6\,b_5$ 111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	g	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Δυαδικοί κώδικες αλφαριθμητικών χαρακτήρων

Χαρακτήρες ελέγχου κώδικα ASCII

NUL	Null	DLE	Data link escape
SOH	Start of heading	DC1	Device control 1
STX	Start of text	DC2	Device control 2
ETX	End of text	DC3	Device control 3
EOT	End of transmission	DC4	Device control 4
ENQ	Enquiry	NAK	Negative acknowledge
ACK	Acknowledge	SYN	Synchronize
BEL	Bell	ETB	End transmitted block
BS	Backspace	CAN	Cancel
HT	Horizontal tab	EM	End of medium
LF	Line feed	SUB	Substitute
VT	Vertical tab	ESC	Escape
FF	Form feed	FS	File separator
CR	Carriage return	GS	Group separator
SO	Shift out	RS	Record separator
SI	Shift in	US	Unit separator
SP	Space	DEL	Delete or rubout

- Ο Ελληνικός Οργανισμός Τυποποίησης (ΕΛΟΤ) έχει αναπτύξει τον κώδικα ΕΛΟΤ-928 (επέκταση του ASCII), ο οποίος χρησιμοποιεί 8 δυαδικά ψηφία και παρέχει ενιαία κωδικοποίηση λατινικών και ελληνικών χαρακτήρων.
- Έως σήμερα έχουν γίνει πολλές προσπάθειες ανάπτυξης κωδίκων αλφαριθμητικών χαρακτήρων, με σημαντικότερη αυτήν της κοινοπραξίας Unicode, η οποία αναπτύσσει το ομώνυμο πρότυπο, για την κωδικοποίηση των χαρακτήρων όλων των υπαρχόντων συστημάτων γραφής, χρησιμοποιώντας έως και 32 δυαδικά ψηφία ανά χαρακτήρα.

Κώδικες ανίχνευσης και διόρθωσης σφαλμάτων

- Η επίδραση του θορύβου (ανεπιθύμητων διακυμάνσεων των ψηφιακών σημάτων) στα ψηφιακά συστήματα, κατά τη μετάδοση ή την επεξεργασία δεδομένων, μπορεί να έχει αποτέλεσμα την αλλοίωση της τιμής ενός ή περισσότερων δυαδικών ψηφίων.
- Ένας τρόπος ανίχνευσης των σφαλμάτων είναι η προσθήκη ενός επιπλέον ψηφίου (ψηφίο ισοτιμίας, parity bit) σε κάθε κωδικοποιημένο χαρακτήρα ή αριθμό, έτσι ώστε το πλήθος των μονάδων που περιέχονται σε αυτόν να είναι περιττό ή άρτιο, δηλαδή να δημιουργείται κώδικας με περιττή ή άρτια ισοτιμία, αντίστοιχα.
- Για σύστημα που χρησιμοποιεί κωδικοποίηση ASCII, προσαρτούμε στην πιο σημαντική θέση κάθε κωδικοποιημένου χαρακτήρα ένα ψηφίο ισοτιμίας, ώστε να υποδηλώσουμε την ισοτιμία του.
- Για παράδειγμα, η λέξη «if» σε κώδικα ASCII συνίσταται από τους κωδικοποιημένους χαρακτήρες 1101001 και 1100110, στην πιο σημαντική θέση των οποίων προσαρτούμε ένα ψηφίο ισοτιμίας με τιμή 0, προκειμένου να υποδηλώσουμε άρτια ισοτιμία.
- Στη συνέχεια, οι κωδικοποιημένοι χαρακτήρες που συνίστανται πλέον από 8 δυαδικά ψηφία μεταδίδονται στον προορισμό τους και η ισοτιμία τους ελέγχεται από το δέκτη. Εάν η ισοτιμία των χαρακτήρων που ελήφθησαν δεν είναι άρτια, αυτό σημαίνει ότι κατά τη διάρκεια της μετάδοσης έχει αλλάξει η τιμή τουλάχιστον ενός δυαδικού ψηφίου.
- Ωστόσο, όταν έχουμε άρτιο πλήθος σφαλμάτων, δεν εξασφαλίζεται η ανίχνευσή τους, οπότε απαιτούνται διαφορετικού είδους κώδικες ανίχνευσης σφαλμάτων.

Κώδικες ανίχνευσης και διόρθωσης σφαλμάτων

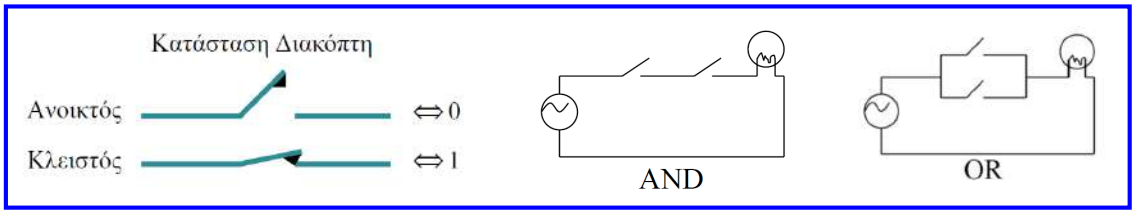
- Όλοι οι κώδικες διόρθωσης / ανίχνευσης σφαλμάτων προσθέτουν πλεονασματική πληροφορία στα δεδομένα που αποστέλλονται.
- Πληρέστερος κώδικας ανίχνευσης σφαλμάτων είναι ο κώδικας Hamming.
- Εκτός από τη δυνατότητα ανίχνευσης της ύπαρξης σφαλμάτων, έχει τη δυνατότητα προσδιορισμού της θέσης των σφαλμάτων σε έναν κωδικοποιημένο χαρακτήρα, έτσι ώστε να μπορούν να ανακτηθούν τα αρχικά δεδομένα.
- Με βάση τη μέθοδο αυτή, δημιουργούνται κωδικοποιημένοι χαρακτήρες στους οποίους τα ψηφία ισοτιμίας συνδυάζονται με επιλεγμένες ομάδες ψηφίων των αρχικών κωδικοποιημένων χαρακτήρων.
- Κατά τη λήψη κάθε χαρακτήρα, ανιχνεύονται τυχόν σφάλματα μέσω ελέγχου ισοτιμίας και σχηματίζεται ένας δυαδικός αριθμός με ψηφία ελέγχου που δηλώνει τη θέση του ψηφίου του οποίου η τιμή έχει αλλάξει, ώστε αυτό να μπορεί να διορθωθεί.
- Ο κώδικας Hamming θα μελετηθεί αναλυτικά στο τρίτο μέρος της ενότητας που αφορά τα δίκτυα υπολογιστών.

Άλγεβρα Boole

- Τα ψηφιακά κυκλώματα υλοποιούνται με τρανζίστορ, τα οποία παρουσιάζουν δύο επιτρεπτές καταστάσεις λειτουργίας, αντίστοιχες με τις λογικές τιμές 0 και 1.
- Για τη σχεδίαση και την ανάλυση των ψηφιακών συστημάτων είναι απαραίτητη η χρήση ενός αλγεβρικού συστήματος, οι μεταβλητές του οποίου λαμβάνουν μόνο αυτές τις δύο τιμές.
- Οι βάσεις για το σύστημα αυτό τέθηκαν το 1854 από τον George Boole, που ανέπτυξε ένα αλγεβρικό σύστημα για τη συστηματική αντιμετώπιση της λογικής.
- Τα αξιώματα της άλγεβρας αυτής διατυπώθηκαν το 1904 από τον Edward Huntington και το 1938, ο Claude Elwood Shannon εισήγαγε τη δίτιμη άλγεβρα Boole, για να εκφράσει τις ιδιότητες ηλεκτρικών κυκλωμάτων διακοπτών με δύο καταστάσεις, η οποία μας επιτρέπει να εκφράσουμε και τη σχέση μεταξύ των εισόδων και των εξόδων ενός ψηφιακού κυκλώματος.
- Η δίτιμη άλγεβρα Boole πραγματεύεται δυαδικές μεταβλητές και λογικές πράξεις οι οποίες υλοποιούνται με ηλεκτρονικά κυκλώματα που ονομάζονται λογικές πύλες, τα κύρια στοιχεία των οποίων είναι τα τρανζίστορ.

Άλγεβρα Boole

- Η δίτιμη άλγεβρα Boole ορίζεται από ένα σύνολο που περιλαμβάνει τα στοιχεία 0 και 1, καθώς και τους δυαδικούς τελεστές •, + και ′, που αντιστοιχούν κατά σειρά στις λογικές πράξεις AND (λογικό γινόμενο), OR (λογικό άθροισμα) και NOT (λογική άρνηση).
- Το αποτέλεσμα του λογικού γινομένου μεταξύ δύο στοιχείων είναι 0, όταν τουλάχιστον ένα από τα δύο στοιχεία ισούται με 0, και 1, όταν και τα δύο στοιχεία ισούνται με 1.
- Το αποτέλεσμα του λογικού αθροίσματος μεταξύ δύο στοιχείων είναι 0, όταν και τα δύο στοιχεία ισούνται με 0, και 1, όταν τουλάχιστον ένα στοιχείο ισούται με 1.
- Ο τελεστής αντιστροφής είναι μοναδιαίος (εφαρμόζεται σε ένα μόνο στοιχείο) και η λογική πράξη στην οποία αντιστοιχεί έχει αποτέλεσμα 0, όταν το στοιχείο ισούται με 1, και 1, όταν το στοιχείο ισούται με 0.
- Η δίτιμη άλγεβρα Boole αναφέρεται και ως άλγεβρα των διακοπτών, αφού υπάρχει αντιστοιχία βασικών λογικών πράξεων με απλά κυκλώματα διακοπτών.



Αξιώματα άλγεβρας Boole

- **Α1.** Κλειστότητα ως προς τους τρεις τελεστές, αφού σύμφωνα με τους ορισμούς των τριών λογικών πράξεων το αποτέλεσμά τους δεν μπορεί να είναι διαφορετικό από 0 ή 1, δηλαδή, εάν $x, y \in A = \{0, 1\}$, τότε $x \cdot y \in A$, $x + y \in A$ και $x', y' \in A$.
- **Α2.** Αντιμεταθετικότητα: η διάταξη των στοιχείων κατά την εκτέλεση των λογικών πράξεων AND και OR δεν επηρεάζει το αποτέλεσμα, δηλαδή, εάν $x, y \in A$, τότε $x \cdot y = y \cdot x$ και $x + y = y + x$.
- **Α3.** Επιμεριστικότητα: το λογικό γινόμενο επιμερίζεται στο λογικό άθροισμα και το λογικό άθροισμα επιμερίζεται στο λογικό γινόμενο, δηλαδή, εάν $x, y, z \in A$, τότε $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$ και $x + y \cdot z = (x + y) \cdot (x + z)$.
- **Α4.** Ουδέτερα στοιχεία: για τις λογικές πράξεις AND και OR τα ουδέτερα στοιχεία είναι 1 και 0, αντίστοιχα, δηλαδή, εάν $x \in A$, τότε $x \cdot 1 = x$ και $x + 0 = x$.
- **Α5.** Συμπλήρωμα: από τον ορισμό της λογικής αντιστροφής προκύπτει ότι, εάν $x \in A$, τότε $x \cdot x' = 0$ και $x + x' = 1$, όπου το στοιχείο x' αναφέρεται ως συμπλήρωμα του στοιχείου ή της μεταβλητής x. Το συμπλήρωμα συμβολίζεται και ως $\overline{x}$.
- Σε κάθε ζεύγος αξιωμάτων το ένα τμήμα μπορεί να προκύψει από το άλλο, με αμοιβαία εναλλαγή των λογικών πράξεων AND, OR και των στοιχείων 0 και 1 (αρχή δυϊσμού). Συνεπώς, κάθε αλγεβρική έκφραση η οποία μπορεί να παραχθεί από τα αξιώματα εξακολουθεί να ισχύει, εάν εναλλάξουμε τους τελεστές και τα ουδέτερα στοιχεία.

Θεωρήματα άλγεβρας Boole

- **Θ1.** Οι λογικές πράξεις μιας μεταβλητής με τον εαυτό της έχουν αποτέλεσμα τη μεταβλητή αυτή, δηλαδή $x \cdot x = x$, $x + x = x$.
- **Θ2.** Οι λογικές πράξεις μιας μεταβλητής με το αντίστοιχο ουδέτερο στοιχείο έχουν αποτέλεσμα το ουδέτερο στοιχείο, δηλαδή $x \cdot 0 = 0$, $x + 1 = 1$.
- **Θ3.** Θεώρημα διπλής άρνησης: το συμπλήρωμα του συμπληρώματος μιας μεταβλητής ισούται με τη μεταβλητή αυτή, δηλαδή $(x')' = x$.
- **Θ4.** Προσεταιριστική ιδιότητα λογικού αθροίσματος και λογικού γινομένου: $(x + y) + z = x + (y + z)$, $(x \cdot y) \cdot z = x \cdot (y \cdot z)$.
- **Θ5.** Θεώρημα De Morgan: το συμπλήρωμα του αποτελέσματος μιας λογικής πράξης AND ή OR ανάμεσα σε δύο μεταβλητές λαμβάνεται, εάν συμπληρώσουμε κάθε μεταβλητή και εναλλάξουμε τους τελεστές, δηλαδή $(x + y)' = x' \cdot y'$, $(x \cdot y)' = x' + y'$.
- **Θ6.** Θεωρήματα απορρόφησης: κατά τη λογική πρόσθεση μιας μεταβλητής με το λογικό γινόμενο της μεταβλητής αυτής με άλλη μεταβλητή, το λογικό γινόμενο απορροφάται, δηλαδή $x + x \cdot y = x$. Αντίστοιχα ισχύει και η απορρόφηση του λογικού αθροίσματος, δηλαδή $x \cdot (x + y) = x$. Ειδική περίπτωση: $x + x' \cdot y = x + y$, $x \cdot (x' + y) = x \cdot y$.
- **Θ7.** Θεώρημα ομοφωνίας: αποτελεί ειδική περίπτωση του θεωρήματος απορρόφησης, που εκφράζεται αλγεβρικά με τις σχέσεις $x \cdot y + x' \cdot z + y \cdot z = x \cdot y + x' \cdot z$ και $(x + y) \cdot (x' + z) \cdot (y + z) = (x + y) \cdot (x' + z)$.

Θεωρήματα άλγεβρας Boole

- Τα θεωρήματα που αναφέρθηκαν δεν είναι μοναδικά, ωστόσο είναι τα πιο βασικά και η εξοικείωση με αυτά είναι σημαντική για το χειρισμό αλγεβρικών εκφράσεων.

- Για την απόδειξη των θεωρημάτων χρησιμοποιούνται:

  ✓ τα αξιώματα ή/και τα αποδεδειγμένα θεωρήματα και ξεκινώντας από το ένα μέλος της εξίσωσης ενός θεωρήματος καταλήγουμε στο άλλο,

  ✓ ή και από τα δύο μέλη καταλήγουμε στο ίδιο αποτέλεσμα.

  ✓ Εναλλακτική μέθοδος, είναι η αναγωγή ενός θεωρήματος σε αξίωμα.

Προτεραιότητα τελεστών

- Όπως σε κάθε αλγεβρικό σύστημα, έτσι και στην άλγεβρα Boole είναι καθορισμένη μια προτεραιότητα τελεστών, ώστε να επιτυγχάνεται ορθός χειρισμός των αλγεβρικών εκφράσεων.

- Πρώτη προτεραιότητα, αποτελεί ο υπολογισμός των εκφράσεων που βρίσκονται εντός παρενθέσεων.

- Ακολουθεί ο υπολογισμός των συμπληρωμάτων (τελεστής ′), στη συνέχεια ο υπολογισμός των λογικών γινομένων (τελεστής ·) και στο τέλος διενεργείται ο υπολογισμός των λογικών αθροισμάτων (τελεστής +).

- Εάν, για παράδειγμα, κατά τον υπολογισμό του συμπληρώματος της έκφρασης x + (y · z) εφαρμόσουμε μια γενίκευση του θεωρήματος De Morgan, συμπληρώνοντας τις μεταβλητές και εναλλάσσοντας τους τελεστές, χωρίς, όμως, να τηρήσουμε την ορθή προτεραιότητά τους, θα καταλήξουμε στην έκφραση x′ · y′ + z′, που είναι εσφαλμένη.

- Ακολουθώντας την προαναφερθείσα προτεραιότητα τελεστών, καταλήγουμε στην ορθή έκφραση που ακολουθεί:

$$[x + (y \cdot z)]' = x' \cdot (y \cdot z)' = x' \cdot (y' + z') = x' \cdot y' + x' \cdot z'$$

Απλοποίηση αλγεβρικών εκφράσεων

- Οι αλγεβρικοί μετασχηματισμοί που βασίζονται στα αξιώματα και θεωρήματα της άλγεβρας Boole, εκτελούνται για την απλοποίηση αλγεβρικών εκφράσεων, έτσι ώστε αυτές να μπορούν να υλοποιηθούν με μικρότερο αριθμό λογικών πυλών, καθεμία από τις οποίες υλοποιεί μία λογική πράξη.

- Η χρήση αλγεβρικών μετασχηματισμών μειονεκτεί επειδή δεν οδηγεί πάντοτε εύκολα στην απλούστερη δυνατή (ελαχιστοποιημένη) έκφραση και οι αλγεβρικοί μετασχηματισμοί δεν μπορούν να συστήσουν συστηματική μεθοδολογία απλοποίησης αλγεβρικών εκφράσεων.

- Το αντικείμενο της ελαχιστοποίησης αλγεβρικών εκφράσεων είναι πολύ σημαντικό για τη σχεδίαση και την ανάλυση ψηφιακών κυκλωμάτων.

- Συστηματικός τρόπος ελαχιστοποίησης αλγεβρικών εκφράσεων είναι η μέθοδος του χάρτη Karnaugh, ενώ για σύνθετες αλγεβρικές εκφράσεις μεγάλου πλήθους μεταβλητών, χρησιμοποιούνται εργαλεία απλοποίησης με τη βοήθεια ηλεκτρονικού υπολογιστή.

- Στη συνέχεια των παρουσιάσεων της ενότητας, για λόγους απλότητας, όπως συμβαίνει και στη διατύπωση του πολλαπλασιασμού της κοινής άλγεβρας, ο τελεστής · θα παραλείπεται εντός των αλγεβρικών εκφράσεων, αλλά θα υπονοείται.

Απλοποίηση αλγεβρικών εκφράσεων

- Για την απλοποίηση της αλγεβρικής έκφρασης x'y + xz + yz + yzw', εκτελούμε τους ακόλουθους μετασχηματισμούς:

$$x'y + xz + yz + yzw' = x'y + xz + yz(1 + w') = x'y + xz + yz$$
$$= x'y + xz + yz(x + x') = x'y + xz + xyz + x'yz = x'y(1 + z) + xz(1 + y)$$
$$= x'y + xz$$

- Αρχικά, χρησιμοποιήσαμε το αξίωμα της επιμεριστικότητας στα δύο τελευταία λογικά γινόμενα της έκφρασης και στη συνέχεια το αξίωμα για τα ουδέτερα στοιχεία (1 + w' = 1).

- Αξιοποιώντας το αξίωμα για το συμπλήρωμα ενός στοιχείου (x + x' = 1), έγινε προσπάθεια δημιουργίας λογικών γινομένων, τέτοιων ώστε στη συνέχεια να είναι δυνατή η εφαρμογή του αξιώματος της επιμεριστικότητας που θα οδηγήσει σε μείωση των λογικών γινομένων της έκφρασης.

Απλοποίηση αλγεβρικών εκφράσεων

- Για την απλοποίηση της αλγεβρικής έκφρασης xy + xy′ + x′y, εκτελούμε τους ακόλουθους μετασχηματισμούς:

$$xy + xy' + x'y = xy + xy' + x'y + xy = x(y + y') + (x' + x)y = x + y$$

- Κατά τον πρώτο μετασχηματισμό προσθέσαμε το λογικό γινόμενο xy (το οποίο περιλαμβάνεται στην αρχική έκφραση), αφού με βάση το πρώτο θεώρημα της άλγεβρας Boole η ενέργεια αυτή δεν αλλοιώνει την αρχική έκφραση.
- Στόχος της ενέργειας αυτής είναι η δημιουργία ενός συνδυασμού λογικών γινομένων, ο οποίος θα μας δίνει τη δυνατότητα εφαρμογής του αξιώματος της επιμεριστικότητας κατά τον επόμενο μετασχηματισμό, ώστε τελικά να λάβουμε μια απλοποιημένη μορφή της αρχικής έκφρασης.
- Οι επιλογές που ακολουθήθηκαν στις παραπάνω απλοποιήσεις εκφράσεων, η χρήση, δηλαδή, του αξιώματος για το συμπλήρωμα ενός στοιχείου και η πρόσθεση λογικών γινομένων που περιλαμβάνονται στην αρχική έκφραση, αποτελούν χρήσιμες πρακτικές απλοποίησης, ωστόσο δεν μπορεί να υποστηρίξει κανείς ότι συνιστούν μεθοδολογία απλοποίησης.

Απλοποίηση αλγεβρικών εκφράσεων

- Κατά την απόδειξη της σχέσης x′y′ + y′z + xz + xy + yz′ = x′y′ + xz + yz′, μπορείτε να αξιοποιήσετε το αξίωμα για το συμπλήρωμα ενός στοιχείου (x + x′ = 1), έτσι ώστε να δημιουργηθούν λογικά γινόμενα κατάλληλα για την εφαρμογή, στη συνέχεια, του αξιώματος της επιμεριστικότητας που οδηγεί σε μείωση των λογικών γινομένων της έκφρασης:

$$\begin{aligned} &x'y' + y'z + xz + xy + yz' = x'y' + y'z(x + x') + xz + xy + yz' \\ &= x'y' + xy'z + x'y'z + xz + xy + yz' = x'y'(1 + z) + xz(y' + 1) + xy + yz' \\ &= x'y'1 + xz1 + xy(z + z') + yz' = x'y' + xz + xyz + xyz' + yz' \\ &= x'y' + xz(1 + y) + yz'(1 + x) = x'y' + xz1 + yz'1 = x'y' + xz + yz' \end{aligned}$$

- Για την απόδειξη του πρώτου τμήματος του θεωρήματος ομοφωνίας, θα πρέπει και πάλι να χρησιμοποιήσετε το αξίωμα x + x′ = 1, για να δημιουργήσετε λογικά γινόμενα τέτοια, ώστε να είναι δυνατή, στη συνέχεια, η εφαρμογή του αξιώματος της επιμεριστικότητας που θα σας οδηγήσει σε μείωση των λογικών γινομένων της έκφρασης:

$$\begin{aligned} &xy + x'z + yz = xy + x'z + yz(x + x') = xy + x'z + yzx + yzx' \\ &= xy(1 + z) + x'z(1 + y) = xy1 + x'z1 = xy + x'z \end{aligned}$$

Λογικές συναρτήσεις

- Οι λογικές συναρτήσεις περιγράφονται συνήθως από αλγεβρικές εκφράσεις που περιλαμβάνουν δυαδικές μεταβλητές, τις σταθερές τιμές 0 και 1, τους τελεστές των τριών λογικών πράξεων, παρενθέσεις και αγκύλες.
- Κάθε λογική συνάρτηση λαμβάνει τιμή 0 ή 1, ανάλογα με τις λογικές τιμές των δυαδικών μεταβλητών που συμμετέχουν σε αυτήν.
- Για παράδειγμα, θεωρήστε τη συνάρτηση F(x,y,z) = x'y + xz, που περιλαμβάνει τρεις δυαδικές μεταβλητές και λαμβάνει λογική τιμή 1, όταν η μεταβλητή x και η μεταβλητή y έχουν τιμή 0 και 1, αντίστοιχα, ή όταν οι μεταβλητές x και z έχουν τιμή 1, ενώ λαμβάνει λογική τιμή 0 για τους υπόλοιπους δυνατούς συνδυασμούς των μεταβλητών που συμμετέχουν σε αυτήν.
- Εκτός, από την περιγραφή μιας λογικής συνάρτησης μέσω αλγεβρικής έκφρασης, υπάρχουν πρόσθετες επιλογές περιγραφής, όπως η δημιουργία ενός πίνακα που αναφέρεται ως πίνακας αλήθειας και περιλαμβάνει όλους τους δυνατούς συνδυασμούς τιμών των μεταβλητών που συμμετέχουν στη συνάρτηση, καθώς και την τιμή της συνάρτησης αυτής για κάθε συνδυασμό.
- Το πλήθος των δυνατών συνδυασμών n μεταβλητών που συμμετέχουν σε μία συνάρτηση είναι 2^n και προκύπτουν εύκολα, εάν γράψουμε κατά σειρά τους δυαδικούς αριθμούς από 0 έως 2^{n-1} και αντιστοιχίσουμε κάθε δυαδικό ψηφίο σε μία από τις μεταβλητές.

Λογικές συναρτήσεις

- Ο πίνακας αλήθειας της συνάρτησης F(x,y,z) = x'y + xz, περιλαμβάνει 8 (2^3) γραμμές και 4 στήλες (μία στήλη για κάθε μεταβλητή και μία στήλη για τη συνάρτηση).
- Είναι προφανές ότι μια λογική συνάρτηση μπορεί να περιγραφεί με έναν και μοναδικό πίνακα αλήθειας, ενώ με βάση όσα είδαμε προηγουμένως, η αλγεβρική έκφραση μιας συνάρτησης μπορεί να λάβει περισσότερες από μία μορφές.
- Στην περίπτωση λοιπόν που δύο ή περισσότερες αλγεβρικές εκφράσεις παράγουν τον ίδιο πίνακα αλήθειας, οι εκφράσεις αυτές είναι ισοδύναμες και αντιστοιχούν στην ίδια λογική συνάρτηση.
- Το συμπέρασμα αυτό μπορεί να χρησιμοποιηθεί για την απόδειξη αλγεβρικών σχέσεων, όπως τα θεωρήματα της άλγεβρας Boole.
- Προφανές μειονέκτημα της περιγραφής μιας λογικής συνάρτησης μέσω πίνακα αλήθειας είναι ότι το μέγεθος του πίνακα αυξάνεται εκθετικά με το πλήθος των δυαδικών μεταβλητών.
- Για παράδειγμα, στην περίπτωση 16 μεταβλητών, απαιτείται πίνακας αλήθειας με περισσότερες από 65.5 χιλιάδες (2^{16}) γραμμές.

x	*y*	*z*	*F*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Λογικές συναρτήσεις

- Απόδειξη ισοδυναμίας αλγεβρικών εκφράσεων με πίνακες αλήθειας, π.χ. το πρώτο τμήμα του θεωρήματος De Morgan, δηλαδή η σχέση (x + y)′ = x′y′,

x	y	x'	y'	$x+y$	$(x+y)'$	$x'y'$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

- Οι δύο τελευταίες στήλες του πίνακα είναι ταυτόσημες, γεγονός που σημαίνει ότι οι δύο εκφράσεις είναι ισοδύναμες.
- Για να εξαγάγουμε την αλγεβρική έκφραση μιας συνάρτησης από τον πίνακα αλήθειας, θα πρέπει να εντοπίσουμε τους συνδυασμούς τιμών των μεταβλητών για τους οποίους η συνάρτηση λαμβάνει τιμή 1.
- Στη συνέχεια εκφράζουμε κάθε συνδυασμό ως λογικό γινόμενο των μεταβλητών με τιμή 1 και των συμπληρωμάτων των μεταβλητών με τιμή 0 και αθροίζουμε τα λογικά γινόμενα.
- Ακολουθώντας τη διαδικασία αυτή για τον πίνακα αλήθειας της συνάρτησης F(x,y,z) = x′y + xz, προκύπτει η συνάρτηση F(x,y,z) = x′yz′ + x′yz + xy′z + xyz, η οποία είναι ισοδύναμη με την αρχική συνάρτηση, δηλαδή την F(x,y,z) = x′y + xz.

Συμπληρωματικές λογικές συναρτήσεις

- Μια λογική συνάρτηση F′ ονομάζεται συμπληρωματική συνάρτηση μιας συνάρτησης F, όταν λαμβάνει λογική τιμή 1 ή 0, για τους συνδυασμούς τιμών των μεταβλητών για τους οποίους η συνάρτηση F λαμβάνει τιμή 0 ή 1, αντίστοιχα.
- Ο τιμές στον πίνακα αλήθειας της συμπληρωματικής συνάρτησης της F(x,y,z) = x′y + xz, προκύπτουν από εκείνες της συνάρτησης F(x,y,z) για κάθε συνδυασμό τιμών των μεταβλητών.

x	y	z	F	F'
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

$$F'(x,y,z) = x'y'z' + x'y'z + xy'z' + xyz'$$

- Για να εξαγάγουμε την αλγεβρική έκφραση της συνάρτησης F′ με τη βοήθεια του πίνακα αλήθειας, ενεργούμε με τρόπο όμοιο με εκείνον που ακολουθήσαμε για την εξαγωγή της έκφρασης της F.

Γενικευμένο θεώρημα De Morgan

- Η συμπληρωματική συνάρτηση F′ προκύπτει από τη συνάρτηση F, ως εξής:

$$F'(x, y, \ldots, w, +, \cdot, 0, 1) = F(x', y', \ldots, w', \cdot, +, 1, 0)$$

- Αυτό σημαίνει ότι κατά την παραγωγή της F′, στη θέση κάθε μεταβλητής τίθεται το συμπλήρωμά της, οι τελεστές + και · εναλλάσσονται και οι σταθερές τιμές 0 και 1, επίσης εναλλάσσονται.
- Αυτό ισοδυναμεί με την εφαρμογή αρχικά της αρχής του δυϊσμού και στη συνέχεια με την αντικατάσταση κάθε μεταβλητής με το συμπλήρωμά της.
- Επισημαίνεται ότι, κατά την παραγωγή της συμπληρωματικής συνάρτησης, θα πρέπει να τηρείται η προτεραιότητα των τελεστών.
- Για την εξαγωγή της συμπληρωματικής συνάρτησης F′ της συνάρτησης: F(x,y,z,w) = x′(y + z)′(y + wz′)(x + z), εφαρμόζουμε το γενικευμένο θεώρημα De Morgan, ως εξής:

$$F'(x,y,z,w) = [x'(y+z)'(y+wz')(x+z)]' = [x'(y+z)']' + (y+wz')' + (x+z)'$$
$$= x + (y+z) + y'(w'+z) + x'z' = x + y + z + y'w' + y'z + x'z'$$

Γενικευμένο θεώρημα De Morgan

- Για τον υπολογισμό της συμπληρωματικής συνάρτησης της

F(A,B,C,D) = (AC′ + A′B′)[(B′ + C′)D′ + D]

και την απλοποίησή της, εφαρμόζουμε το γενικευμένο θεώρημα De Morgan και στη συνέχεια εκτελούμε αλγεβρικούς μετασχηματισμούς:

$$F'(A,B,C,D) = \{(AC' + A'B')[(B'+C')D' + D]\}'$$
$$= [(AC' + A'B')]' + [(B'+C')D' + D]' = (AC')'(A'B')' + [(B'+C')D']'D'$$
$$= (A' + C)(A + B) + [(B'+C')' + D]D' = (A' + C)(A + B) + (BC + D)D'$$
$$= A'A + A'B + AC + BC + BCD' + DD' = A'B + AC + BC + BCD'$$
$$= A'B + AC + BC(1 + D') = A'B + AC + BC = A'B + AC$$

- Στην τελευταία ισότητα εφαρμόστηκε το θεώρημα ομοφωνίας.

Ελαχιστόροι και μεγιστόροι λογικών συναρτήσεων

- Ένα λογικό γινόμενο, στο οποίο συμμετέχουν όλες οι μεταβλητές μιας συνάρτησης (στην κανονική μορφή τους ή στη μορφή συμπληρώματος) μία φορά, αναφέρεται ως ελαχιστόρος (minterm) ενώ ένα λογικό άθροισμα μεταβλητών με τα ίδια χαρακτηριστικά αναφέρεται ως μεγιστόρος (maxterm).
- Για παράδειγμα, όταν πρόκειται για τρεις μεταβλητές x, y και z, το λογικό γινόμενο xy'z και το λογικό άθροισμα x' + y + z' είναι ένας ελαχιστόρος και ένας μεγιστόρος, αντίστοιχα.
- Κατά τη δημιουργία του πίνακα αλήθειας μιας λογικής συνάρτησης με n μεταβλητές, γράφουμε κατά σειρά τους δυαδικούς αριθμούς από 0 έως 2^{n-1} και αντιστοιχίζουμε κάθε δυαδικό ψηφίο σε μία από τις μεταβλητές.
- Έτσι, για 3 μεταβλητές, όταν x = y = z = 0, σχηματίζεται ο μικρότερος δυαδικός αριθμός (0) και όταν x = y = z = 1 ο μεγαλύτερος δυνατός δυαδικός αριθμός (7).
- Υπάρχει λοιπόν αντιστοιχία μεταξύ των 8 δυνατών συνδυασμών τιμών των 3 μεταβλητών και των 8 ελαχιστόρων.
- Όταν η τιμή μιας μεταβλητής είναι 0, τότε στον αντίστοιχο ελαχιστόρο συμμετέχει το συμπλήρωμα της μεταβλητής αυτής, ενώ όταν η τιμή της μεταβλητής είναι 1, τότε στον αντίστοιχο ελαχιστόρο συμμετέχει η μεταβλητή αυτή με την κανονική μορφή της.

Ελαχιστόροι και μεγιστόροι λογικών συναρτήσεων

- Οι ελαχιστόροι ονομάζονται ως m_0 έως m_7 (ο δείκτης συμπίπτει με το δεκαδικό αριθμό που αντιστοιχεί σε κάθε ελαχιστόρο).
- Οι μεγιστόροι (M_0 έως M_7) είναι τα συμπληρώματα των αντίστοιχων ελαχιστόρων και προκύπτουν εύκολα από τους ελαχιστόρους με εφαρμογή του θεωρήματος De Morgan.
- Για n μεταβλητές, σχηματίζονται 2^n ελαχιστόροι και 2^n μεγιστόροι.

x	y	z	Ελαχιστόροι		Μεγιστόροι	
0	0	0	$x'y'z'$	(m_0)	$x+y+z$	(M_0)
0	0	1	$x'y'z$	(m_1)	$x+y+z'$	(M_1)
0	1	0	$x'y\,z'$	(m_2)	$x+y'+z$	(M_2)
0	1	1	$x'y\,z$	(m_3)	$x+y'+z'$	(M_3)
1	0	0	$x\,y'z'$	(m_4)	$x'+y+z$	(M_4)
1	0	1	$x\,y'z$	(m_5)	$x'+y+z'$	(M_5)
1	1	0	$x\,y\,z'$	(m_6)	$x'+y'+z$	(M_6)
1	1	1	$x\,y\,z$	(m_7)	$x'+y'+z'$	(M_7)

Κανονικές και πρότυπες μορφές λογικής συνάρτησης

- Η συνάρτηση F(x,y,z) = x'yz' + x'yz + xy'z + xyz, μπορεί να γραφεί και ως άθροισμα ελαχιστόρων:

$$F = m_2 + m_3 + m_5 + m_7$$

- Η F' μπορεί να εκφραστεί σε μορφή αθροίσματος ελαχιστόρων, εάν προσθέσουμε τους ελαχιστόρους που αντιστοιχούν στους συνδυασμούς τιμών των μεταβλητών, για τους οποίους η συνάρτηση λαμβάνει τιμή 0, που είναι οι ελαχιστόροι που λείπουν από την F:

$$F'(x,y,z) = x'y'z' + x'y'z + xy'z' + xyz' = m_0 + m_1 + m_4 + m_6$$

- Η συμπληρωματική συνάρτηση της F' είναι η F (σύμφωνα με το θεώρημα διπλής άρνησης) και προκύπτει εύκολα από την F' με εφαρμογή του θεωρήματος De Morgan:

$$F = (m_0 + m_1 + m_4 + m_6)' = m'_0 m'_1 m'_4 m'_6 = M_0 M_1 M_4 M_6$$
$$= (x + y + z)(x + y + z')(x' + y + z)(x' + y' + z)$$

- Με αυτό τον τρόπο, εκφράσαμε τη συνάρτηση σε μορφή γινομένου μεγιστόρων. Στη μορφή αυτή συμμετέχουν οι μεγιστόροι που αντιστοιχούν στους συνδυασμούς τιμών των μεταβλητών, για τους οποίους η συνάρτηση λαμβάνει λογική τιμή 0.
- Οι εκφράσεις μιας λογικής συνάρτησης σε μορφή αθροίσματος ελαχιστόρων και γινομένου μεγιστόρων είναι μοναδικές και αναφέρονται ως κανονικές μορφές (canonical forms).

Κανονικές και πρότυπες μορφές λογικής συνάρτησης

- Για να μετατρέψουμε την έκφραση μιας λογικής συνάρτησης από τη μια κανονική μορφή στην άλλη, εναλλάσσουμε τα σύμβολα m, M και τους τελεστές · και +, και ως δείκτες στα σύμβολα θέτουμε τους δείκτες των ελαχιστόρων ή των μεγιστόρων που λείπουν από την αρχική κανονική μορφή.
- Για να προκύψουν οι δείκτες αυτοί, θα πρέπει να λάβουμε υπόψη ότι ο αριθμός των ελαχιστόρων ή μεγιστόρων για συναρτήσεις n μεταβλητών ανέρχεται σε 2^n.
- Το άθροισμα ελαχιστόρων και το γινόμενο μεγιστόρων μιας λογικής συνάρτησης αναφέρονται και ως Σ( ), Π( ), αντίστοιχα, θέτοντας εντός των παρενθέσεων τους δείκτες των ελαχιστόρων ή των μεγιστόρων, αντίστοιχα.
- Οι κανονικές μορφές μιας λογικής συνάρτησης προκύπτουν απευθείας από τον πίνακα αλήθειας και λόγω του ότι στους ελαχιστόρους και στους μεγιστόρους συμμετέχουν όλες οι μεταβλητές της συνάρτησης, οι μορφές αυτές δεν είναι συνήθως οι απλούστερες δυνατές που μπορούν να οδηγήσουν σε υλοποίηση με μικρό πλήθος λογικών πυλών.
- Έτσι, οι λογικές συναρτήσεις που υλοποιούνται σε ψηφιακά κυκλώματα εκφράζονται συχνά σε μορφή αθροίσματος γινομένων ή σε μορφή γινομένου αθροισμάτων, χωρίς όμως στα γινόμενα και τα αθροίσματα αυτά να συμμετέχουν όλες οι μεταβλητές.
- Οι μορφές αυτές αναφέρονται και ως πρότυπες μορφές (standard forms).

Κανονικές και πρότυπες μορφές λογικής συνάρτησης

- Για να μετατρέψουμε μια πρότυπη μορφή αθροίσματος γινομένων σε κανονική, ελέγχουμε κάθε γινόμενο, ώστε να διαπιστώσουμε ποιες μεταβλητές δε συμμετέχουν σε αυτό.
- Για κάθε μεταβλητή που δε συμμετέχει σε κάποιο από τα γινόμενα (έστω x), πολλαπλασιάζουμε το γινόμενο με τον όρο (x + x′).
- Με βάση το αξίωμα της άλγεβρας Boole για το συμπλήρωμα μιας μεταβλητής, ο όρος αυτός ισούται με 1 και η σχετική πράξη δεν επηρεάζει τη λογική τιμή του γινομένου.
- Στη συνέχεια εφαρμόζουμε το αξίωμα της επιμεριστικότητας σε κάθε ελλιπές γινόμενο και λαμβάνουμε την επιθυμητή κανονική μορφή αθροίσματος ελαχιστόρων.
- Στην περίπτωση πρότυπης μορφής γινομένου αθροισμάτων, για κάθε μεταβλητή που δε συμμετέχει σε κάποιο άθροισμα (έστω x), προσθέτουμε στο άθροισμα τον όρο xx′, αφού η πράξη αυτή δεν επηρεάζει τη λογική τιμή του αθροίσματος (λόγω του ότι xx′ = 0).
- Κατόπιν, εφαρμόζουμε το αξίωμα της επιμεριστικότητας σε κάθε ελλιπές άθροισμα και λαμβάνουμε την επιθυμητή κανονική μορφή γινομένου μεγιστόρων.
- Η μετατροπή μεταξύ πρότυπων μορφών γίνεται εύκολα με χρήση του αξιώματος της επιμεριστικότητας.

Κανονικές και πρότυπες μορφές λογικής συνάρτησης

- Μετατροπή της πρότυπης μορφής F(x,y,z) = xy + x′z + yz σε κανονική μορφή αθροίσματος ελαχιστόρων:

$$F(x,y,z) = xy + x'z + yz = xy(z+z') + x'z(y+y') + yz(x+x')$$
$$= xyz + xyz' + x'yz + x'y'z + xyz + x'yz = m_7 + m_6 + m_3 + m_1 = \Sigma(1,3,6,7)$$

- Οι όροι m_3 και m_7 προκύπτουν και δεύτερη φορά στο τελικό άθροισμα, αλλά απαλείφονται λόγω του πρώτου θεωρήματος της άλγεβρας Boole.
- Η κανονική μορφή γινομένου μεγιστόρων προκύπτει απευθείας με χρήση του θεωρήματος De Morgan, ως εξής: F(x,y,z) = Σ(1, 3, 6, 7) = Π(0, 2, 4, 5).
- Μετατροπή της πρότυπης μορφής F(x,y,z) = (x + y + z)(x′ + y)(y + z)(x+z)(x′ + y + z) σε κανονική μορφή γινομένου μεγιστόρων:

$$F(x,y,z) = (x+y+z)(x'+y)(y+z)(x+z)(x'+y+z)$$
$$= (x+y+z)(x'+y+zz')(y+z+xx')(x+z+yy')(x'+y+z)$$
$$(x+y+z)(x'+y+z)(x'+y+z')(y+z+x)(y+z+x')(x+z+y)(x+z+y')\,(x'+y+z)$$
$$= M_0M_4M_5M_2 = \Pi(0,2,4,5)$$

Λογικές πύλες

- Από τις 16 συναρτήσεις δύο μεταβλητών, 2 ισούνται με σταθερές λογικές τιμές (F_0 και F_{15}), 4 αφορούν πράξεις μιας μεταβλητής (οι F_3 και F_5 που ισούνται με τη μία από τις δύο μεταβλητές και οι F_{10} και F_{12} που ισούνται με το συμπλήρωμα μιας από τις δύο μεταβλητές) και οι υπόλοιπες 10 αφορούν πράξεις δύο μεταβλητών.

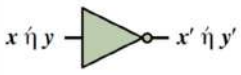
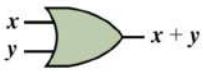
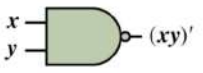
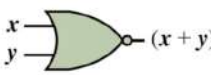
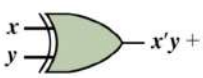
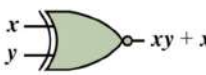
Πίνακες αλήθειας συναρτήσεων δύο μεταβλητών

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- Οι λογικές πράξεις μεταξύ δυαδικών σημάτων στα ψηφιακά κυκλώματα επιτελούνται από απλά δομικά στοιχεία που αναφέρονται ως λογικές πύλες (logic gates).

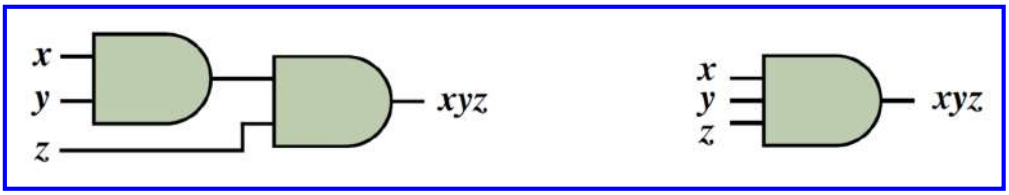
Λογικές πύλες

- Οι λογικές πύλες ανταποκρίνονται στη χαμηλή και την υψηλή στάθμη των δυαδικών σημάτων, οι οποίες εκφράζονται με τις λογικές τιμές 0 και 1, αντίστοιχα.
- Όταν στις εισόδους μιας λογικής πύλης εφαρμόζονται δυαδικά σήματα, τότε στην έξοδό της παράγεται δυαδικό σήμα, η τιμή του οποίου προκύπτει από τη λογική πράξη που επιτελεί η εν λόγω πύλη μεταξύ των τρεχουσών τιμών των σημάτων εισόδου.

Λογική έκφραση	Ονομασία συνάρτησης	Ονομασία λογικής πύλης	Σύμβολο λογικής πύλης
$F_{10} = y'$ και $F_{12} = x'$	Συμπλήρωμα ή λογική άρνηση	Αντιστροφέας ή NOT	
$F_1 = xy$	Λογικό γινόμενο	AND	
$F_7 = x + y$	Λογικό άθροισμα	OR	
$F_{14} = (xy)'$	Συμπλήρωμα λογικού γινομένου	NAND	
$F_8 = (x + y)'$	Συμπλήρωμα λογικού αθροίσματος	NOR	
$F_6 = xy' + x'y = x \oplus y$	Αποκλειστικό OR (exclusive OR)	XOR	
$F_9 = xy + x'y' = (x \oplus y)' = x \odot y$	Ισοδυναμία	XNOR	

Λογικές πύλες

- Ο αντιστροφέας ή πύλη NOT είναι μια λογική πύλη η οποία αντιστρέφει την είσοδο, παράγει, δηλαδή, στην έξοδό της το συμπλήρωμα της εισόδου. Ο κύκλος στην έξοδο της πύλης χρησιμοποιείται για να δηλώσει την αντιστροφή ή συμπλήρωση που επιτελείται.
- Η πύλη AND παράγει στην έξοδό της λογική τιμή 1, όταν όλες οι είσοδοι λαμβάνουν λογική τιμή 1, διαφορετικά παράγει στην έξοδό της λογική τιμή 0.
- Για τη λογική πράξη AND ισχύουν η αντιμεταθετική και η προσεταιριστική ιδιότητα. Αυτό σημαίνει ότι η πύλη AND μπορεί να επεκταθεί σε τρεις ή περισσότερες εισόδους.



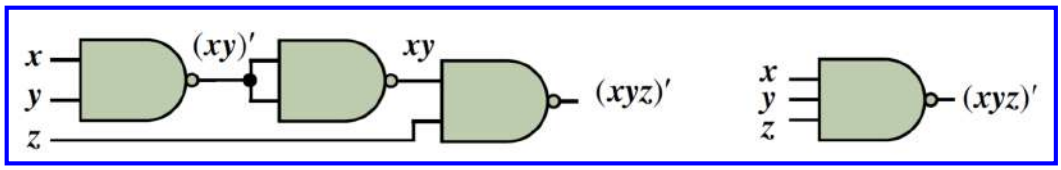
- Η πύλη OR παράγει στην έξοδό της λογική τιμή 1, όταν έστω μία από τις δύο εισόδους της λαμβάνει λογική τιμή 1, ενώ όταν όλες οι είσοδοι λαμβάνουν τιμή 0, τότε παράγει στην έξοδό της λογική τιμή 0.
- Η πύλη OR μπορεί, επίσης, να επεκταθεί σε τρεις ή περισσότερες εισόδους.

Λογικές πύλες

- Η πύλη NAND παράγει ως έξοδο το συμπλήρωμα της εξόδου της πύλης AND, με αποτέλεσμα η έξοδός της να ισούται με 1, όταν τουλάχιστον μία από τις δύο εισόδους της λαμβάνει λογική τιμή 0, διαφορετικά η έξοδος ισούται με 0.
- Στην πράξη του συμπληρώματος λογικού γινομένου που επιτελείται από την πύλη NAND, ενώ ισχύει η αντιμεταθετική ιδιότητα, δεν ισχύει η προσεταιριστική.
- Για παράδειγμα, οι αλγεβρικές εκφράσεις (xyz)’ και [(xy)’z]’ δεν είναι ισοδύναμες, με συνέπεια να μην είναι δυνατή η υλοποίηση της λογικής έκφρασης (xyz)’ με διαδοχική σύνδεση δύο πυλών NAND δύο εισόδων.
- Για να δημιουργήσουμε πύλες NAND με τρεις ή περισσότερες εισόδους, αρκεί να ορίσουμε την πράξη που επιτελούν ως (xyzw…)’.
- Η πύλη NOR παράγει ως έξοδο το συμπλήρωμα της εξόδου της πύλης OR, με αποτέλεσμα η έξοδός της να ισούται με 0, όταν τουλάχιστον μία από τις δύο εισόδους της λαμβάνει λογική τιμή 1, ενώ όταν και οι δύο είσοδοι λαμβάνουν τιμή 0, η έξοδος της πύλης ισούται με 1.
- Όπως συμβαίνει και στην περίπτωση της πράξης του συμπληρώματος λογικού γινομένου, και εδώ δεν ισχύει η προσεταιριστική ιδιότητα και για να δημιουργήσουμε πύλες NOR με τρεις ή περισσότερες εισόδους, αρκεί να ορίσουμε την πράξη που επιτελούν ως (x + y + z + w + …)’.

Λογικές πύλες

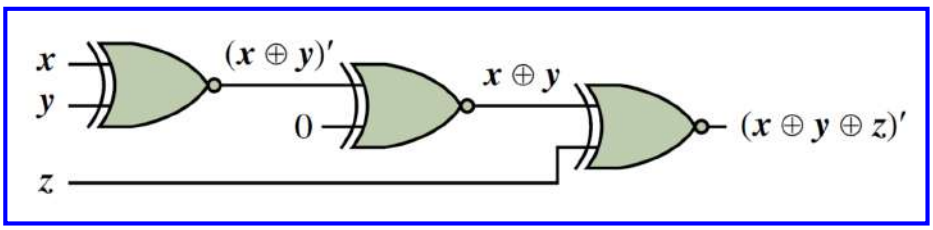
- Για την υλοποίηση μιας πύλης NAND τριών εισόδων, απαιτούνται τρεις πύλες NAND δύο εισόδων.
- Όταν θέτουμε στις δύο εισόδους μιας πύλης NAND την ίδια μεταβλητή εισόδου, τότε η πύλη λειτουργεί ως αντιστροφέας, αφού (AA)′ = A′.



- Η πύλη XOR δύο εισόδων παράγει στην έξοδό της λογική τιμή 0, όταν οι είσοδοι λαμβάνουν την ίδια τιμή, και λογική τιμή 1, όταν οι λογικές τιμές των εισόδων είναι διαφορετικές.
- Για την πράξη αυτή ισχύει η αντιμεταθετική και η προσεταιριστική ιδιότητα, επομένως μπορεί να επεκταθεί σε τρεις ή περισσότερες εισόδους.
- Για περισσότερες από δύο εισόδους, η έξοδος μιας πύλης XOR λαμβάνει λογική τιμή 1, όταν περιττός αριθμός εισόδων λαμβάνει τιμή 1, και λογική τιμή 0, όταν άρτιος αριθμός εισόδων λαμβάνει τιμή 1

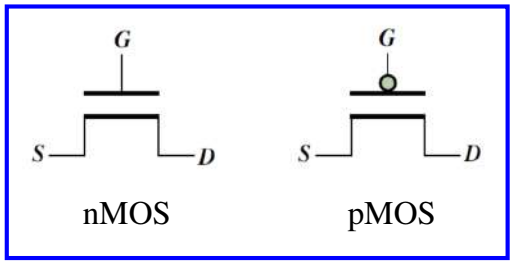
Λογικές πύλες

- Η πύλη XNOR δύο εισόδων παράγει ως έξοδο το συμπλήρωμα της εξόδου της πύλης XOR, αφού οι συναρτήσεις που επιτελούνται από τις πύλες XOR και XNOR είναι συμπληρωματικές μεταξύ τους.
- Για περισσότερες από δύο εισόδους, η έξοδος μιας πύλης XNOR λαμβάνει λογική τιμή 1, όταν άρτιος αριθμός εισόδων λαμβάνει τιμή 1, και λογική τιμή 0, όταν περιττός αριθμός εισόδων λαμβάνει τιμή 1.
- Εάν εξετάσουμε την πράξη που επιτελείται από την πύλη XNOR, προκύπτει ότι δεν είναι δυνατή η υλοποίηση της λογικής έκφρασης $(x \oplus y \oplus z)'$ με διαδοχική σύνδεση δύο πυλών XNOR δύο εισόδων.
- Για την υλοποίηση μιας πύλης XNOR τριών εισόδων, απαιτούνται τρεις πύλες XNOR δύο εισόδων.



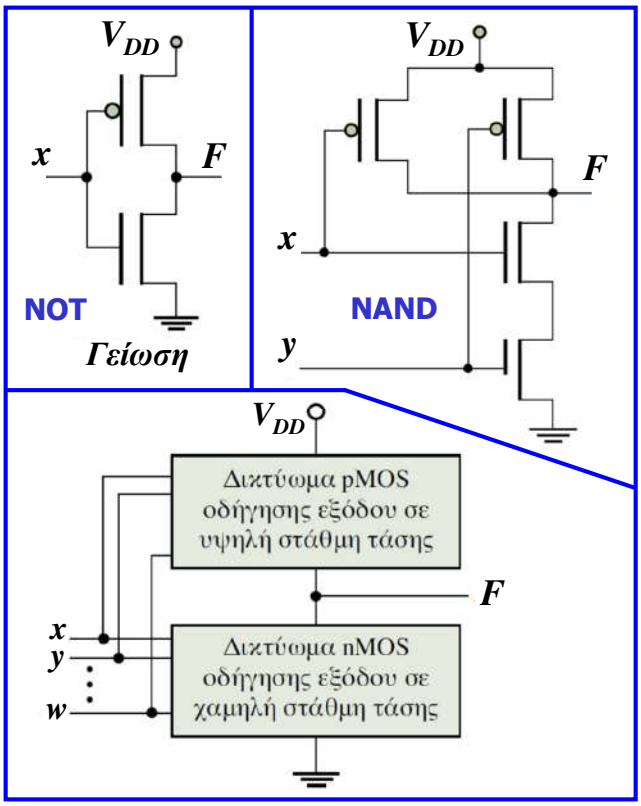
Υλοποίηση λογικών πυλών

- Για την υλοποίηση λογικών πυλών χρησιμοποιούνται τρανζίστορ (transistor), τα οποία είναι ηλεκτρονικά στοιχεία που λειτουργούν ως διακόπτες, παρουσιάζοντας δύο επιτρεπτές καταστάσεις λειτουργίας αντίστοιχες με τις λογικές τιμές 0 και 1.
- Η κυριότερη τεχνολογία υλοποίησης βασίζεται στη χρήση τρανζίστορ επίδρασης πεδίου μετάλλου-οξειδίου-ημιαγωγού (metal-oxide-semiconductor field-effect transistors, MOSFETs), τα οποία διακρίνονται σε τρανζίστορ διαύλου αρνητικού φορτίου (nMOS) και τρανζίστορ διαύλου θετικού φορτίου (pMOS), ανάλογα με την πολικότητα των φορέων φορτίου που συμμετέχουν στη λειτουργία τους.
- Η συμπληρωματική τεχνολογία μετάλλου-οξειδίου-ημιαγωγού (complementary metal-oxide-semiconductor, CMOS) βασίζεται στη συνδυασμένη χρήση των 2 τύπων MOSFETs.
- Σε σχέση με άλλες τεχνολογίες υλοποίησης, η τεχνολογία CMOS παρουσιάζει πλεονεκτήματα, όπως η δυνατότητα ενσωμάτωσης μεγάλου αριθμού πυλών σε ένα κύκλωμα και η χαμηλή κατανάλωση ενέργειας.
- Το τρανζίστορ nMOS συμπεριφέρεται ως κλειστός διακόπτης, όταν στον ακροδέκτη της πύλης του (G) εφαρμόζεται υψηλή στάθμη τάσης (λογική τιμή 1), ενώ όταν εφαρμόζεται χαμηλή στάθμη τάσης (λογική τιμή 0) συμπεριφέρεται ως ανοιχτός.
- Το τρανζίστορ pMOS λειτουργεί αντίθετα.



Υλοποίηση λογικών πυλών

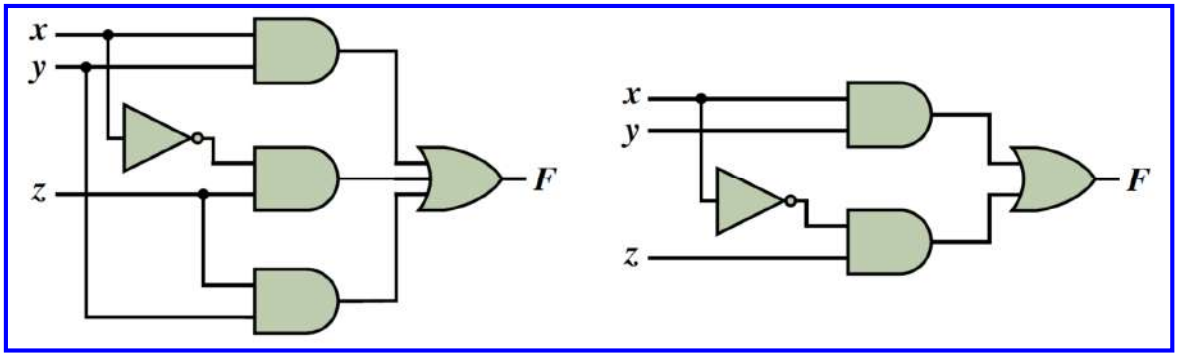
- Ο αντιστροφέας υλοποιείται εύκολα με συνδυασμένη χρήση ενός τρανζίστορ nMOS και ενός τρανζίστορ pMOS.
- Η βάση για την υλοποίηση άλλων λογικών πυλών είναι η δομή που περιλαμβάνει δικτύωμα από τρανζίστορ nMOS για την οδήγηση της εξόδου του κυκλώματος σε χαμηλή στάθμη τάσης και δικτύωμα από τρανζίστορ pMOS για την οδήγηση της εξόδου σε υψηλή στάθμη τάσης.
- Τα δύο δικτυώματα δομούνται έτσι ώστε να δημιουργείται διαδρομή κλειστών διακοπτών στο δικτύωμα από τρανζίστορ nMOS για τους συνδυασμούς τιμών των εισόδων, για τους οποίους η λογική συνάρτηση της πύλης λαμβάνει τιμή 0, ή στο δικτύωμα από τρανζίστορ pMOS για τους συνδυασμούς τιμών των εισόδων, για τους οποίους η συνάρτηση λαμβάνει τιμή 1.



Λογικά κυκλώματα με πύλες NOT, AND και OR

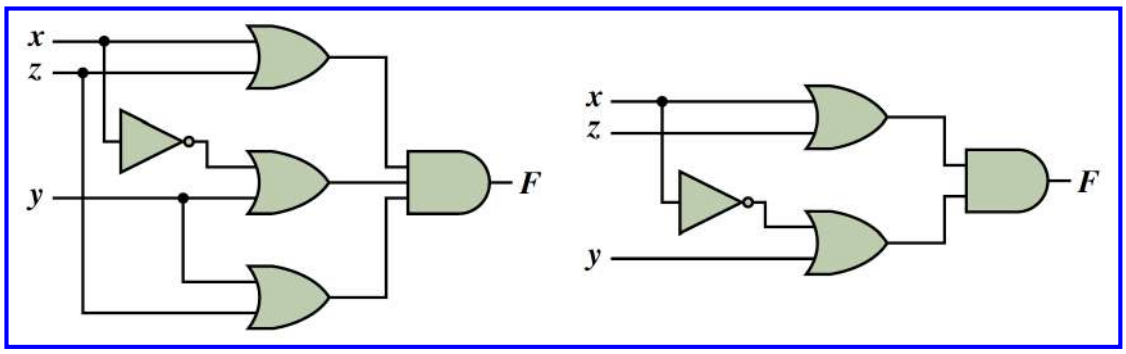
- Ένα λογικό κύκλωμα αποτελείται από γραμμές διασύνδεσης λογικών πυλών που αντιστοιχούν στις διαδρομές των δυαδικών σημάτων και από λογικές πύλες που επιτελούν την επεξεργασία μεταξύ των σημάτων, η οποία δηλώνεται στη λογική συνάρτηση που επιτελείται από κάθε πύλη.
- Τα τρία λογικά γινόμενα που περιλαμβάνονται στη συνάρτηση F(x,y,z) = xy + x'z + yz, μπορούν να υλοποιηθούν με ισάριθμες πύλες AND.
- Λόγω του ότι στα γινόμενα συμμετέχουν δύο μεταβλητές, θα πρέπει να χρησιμοποιήσουμε πύλες AND με δύο εισόδους.
- Η συμπληρωματική μορφή της μεταβλητής x που εμφανίζεται στο δεύτερο κατά σειρά γινόμενο θα πρέπει να παραχθεί με χρήση ενός αντιστροφέα, ο οποίος θα προηγείται της πύλης AND που υλοποιεί το δεύτερο κατά σειρά λογικό γινόμενο.
- Το λογικό άθροισμα των τριών γινομένων μπορεί να υλοποιηθεί με χρήση μιας πύλης OR τριών εισόδων, η οποία θα λαμβάνει ως εισόδους τις εξόδους των πυλών AND.
- Το λογικό κύκλωμα που υλοποιεί την F περιλαμβάνει δύο επίπεδα πυλών, εάν δε συνυπολογίσουμε τον αντιστροφέα που απαιτείται για την παραγωγή της συμπληρωματικής μορφής της εισόδου x.

Λογικά κυκλώματα με πύλες NOT, AND και OR



- Εάν στη συνάρτηση εφαρμόσουμε το θεώρημα ομοφωνίας, προκύπτει η ισοδύναμη συνάρτηση F(x,y,z) = xy + x'z, η οποία μπορεί να υλοποιηθεί με μία πύλη AND λιγότερη, αφού τα λογικά γινόμενα που συμμετέχουν στη συνάρτηση μειώνονται κατά ένα.

Λογικά κυκλώματα με πύλες NOT, AND και OR



- Εφαρμόζοντας το αξίωμα της επιμεριστικότητας F(x,y,z) = xy + x′z = (xy + x′)(xy + z) = (x + x′)(y + x′)(x + z)(y + z) = (x′ + y)(x + z)(y + z), καταλήγουμε σε ισοδύναμη συνάρτηση, η οποία έχει μετασχηματιστεί σε μορφή γινομένου λογικών αθροισμάτων.
- Τα τρία λογικά αθροίσματα που περιλαμβάνονται σε αυτήν μπορούν να υλοποιηθούν με ισάριθμες πύλες OR δύο εισόδων και το λογικό γινόμενο των τριών αθροισμάτων μπορεί να υλοποιηθεί με χρήση μιας πύλης AND τριών εισόδων, η οποία θα λαμβάνει ως εισόδους τις εξόδους των πυλών OR.
- Εάν στη συνάρτηση που υλοποιήθηκε εφαρμόσουμε το θεώρημα ομοφωνίας, προκύπτει η επίσης ισοδύναμη συνάρτηση F(x,y,z) = (x + z)(x′ + y), η οποία υλοποιείται με μία πύλη OR λιγότερη, αφού τα λογικά αθροίσματα μειώνονται κατά ένα.

Λογικά κυκλώματα με πύλες NOT, AND και OR

- Συμπεραίνουμε ότι μια δεδομένη λογική συνάρτηση μπορεί να υλοποιηθεί με λογικά κυκλώματα διαφορετικής δομής που αποτελούνται από:
  - ✓ αντιστροφείς για την παραγωγή των συμπληρωματικών μορφών των μεταβλητών εισόδου (όπου αυτή είναι απαραίτητη) και
  - ✓ δύο επίπεδα πυλών σε διάταξη AND-OR ή OR-AND, ανάλογα με το αν η συνάρτηση είναι σε μορφή αθροίσματος γινομένων ή γινομένου αθροισμάτων, αντίστοιχα.
- Κάποια από τα λογικά κυκλώματα που υλοποιούν μια συνάρτηση είναι απλούστερα από άλλα και ένας σημαντικός στόχος των σχεδιαστών είναι η μείωση του κόστους υλοποίησης, μια ένδειξη του οποίου υπολογίζεται ως το άθροισμα του πλήθους των λογικών πυλών και του πλήθους των εισόδων των πυλών του κυκλώματος.

Λογικά κυκλώματα με πύλες NAND και NOR

- Εάν εφαρμόσουμε το θεώρημα διπλής άρνησης και στη συνέχεια το θεώρημα De Morgan στη λογική συνάρτηση μορφής αθροίσματος γινομένων F(x,y,z) = xy + x′z + yz, λαμβάνουμε την ισοδύναμη λογική συνάρτηση:

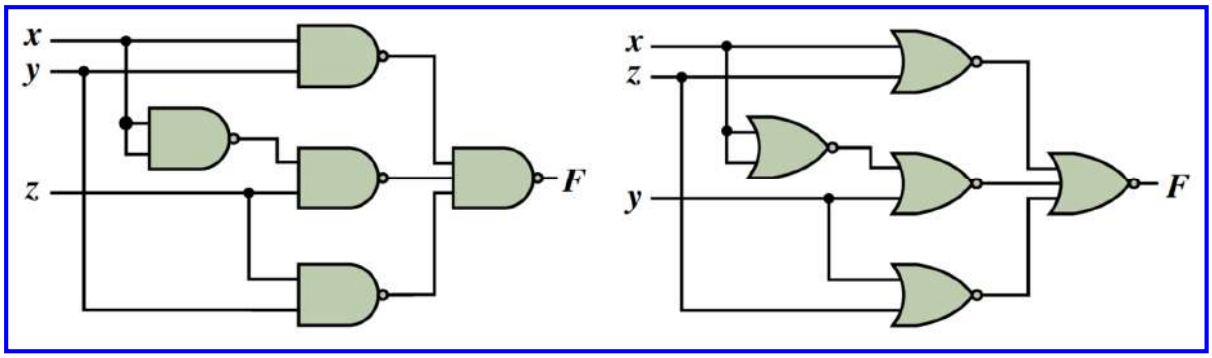
$$F(x,y,z) = [(xy + x'z + yz)']' = [(xy)'(x'z)'(yz)']'$$

- Τα συμπληρώματα των λογικών γινομένων μπορούν να υλοποιηθούν με 3 πύλες NAND δύο εισόδων, ενώ το συμπλήρωμα του συνολικού γινομένου και συνεπώς η συνολική συνάρτηση μπορεί να υλοποιηθεί, εάν θέσουμε τις εξόδους των πυλών αυτών ως εισόδους σε μία πύλη NAND τριών εισόδων.
- Για την παραγωγή της συμπληρωματικής μορφής της μεταβλητής x που απαιτείται, μπορεί να χρησιμοποιηθεί μία πύλη NAND δύο εισόδων, στις εισόδους της οποίας θα πρέπει να τεθεί η μεταβλητή εισόδου x.
- Εάν εφαρμόσουμε όμοια διαδικασία στη συνάρτηση μορφής γινομένου αθροισμάτων, F(x,y,z) = (x + z)(x′ + y)(y + z), λαμβάνουμε την ισοδύναμη λογική συνάρτηση:

$$F(x,y,z) = \{[(x + z)(x' + y)(y + z)]'\}' = [(x + z)' + (x' + y)' + (y + z)']'.$$

- Τα συμπληρώματα των λογικών αθροισμάτων υλοποιούνται με 3 πύλες NOR δύο εισόδων, ενώ το συμπλήρωμα του συνολικού αθροίσματος υλοποιείται εάν θέσουμε τις εξόδους των πυλών αυτών ως εισόδους σε μία πύλη NOR τριών εισόδων.
- Η αντιστροφή της μεταβλητής x υλοποιείται με μία επιπλέον πύλη NOR δύο εισόδων.

Λογικά κυκλώματα με πύλες NAND και NOR



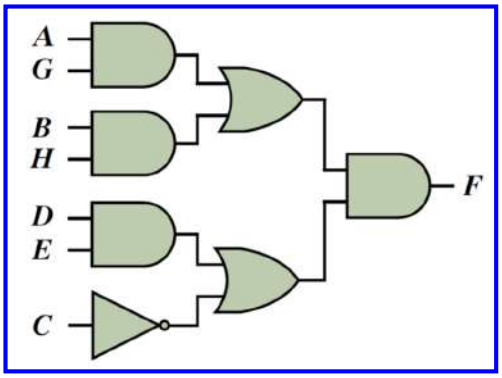
- Συμπεραίνουμε ότι:
  - ✓ οποιοδήποτε λογικό κύκλωμα διάταξης AND-OR μπορεί να μετατραπεί σε λογικό κύκλωμα διάταξης NAND-NAND με την ίδια τοπολογία,
  - ✓ οποιοδήποτε λογικό κύκλωμα διάταξης OR-AND μπορεί να μετατραπεί σε λογικό κύκλωμα διάταξης NOR-NOR ίδιας τοπολογίας.

Λογικά κυκλώματα πολλών επιπέδων

- Οι υλοποιήσεις λογικών συναρτήσεων με δύο επίπεδα πυλών είναι ικανοποιητικές στις περιπτώσεις λογικών συναρτήσεων με λίγες μεταβλητές.
- Στην περίπτωση συναρτήσεων πολλών μεταβλητών, μπορεί να χρησιμοποιηθεί η μέθοδος της παραγοντοποίησης (δηλαδή το αξίωμα της επιμεριστικότητας), ώστε να επιτευχθεί υλοποίηση σε περισσότερα επίπεδα πυλών, αλλά με πύλες περιορισμένου πλήθους εισόδων.
- Οι πύλες αυτές παρουσιάζουν καλύτερα χαρακτηριστικά σε σχέση με τις πύλες πολλών εισόδων, όπως μικρότερη καθυστέρηση απόκρισης και μεγαλύτερη ανεκτικότητα στην παρουσία θορύβου.
- Η συνάρτηση F(A, B, C, D, E, G, H) = AC'G + ADEG + BC'H + BDEH, απαιτεί για την υλοποίησή της έναν αντιστροφέα για την παραγωγή της συμπληρωματικής μορφής της μεταβλητής εισόδου C, 4 πύλες AND (2 με τρεις εισόδους και 2 με τέσσερις εισόδους) για την παραγωγή των λογικών γινομένων και μία πύλη OR τεσσάρων εισόδων για την παραγωγή του συνολικού λογικού αθροίσματος.
- Χρησιμοποιώντας τη μέθοδο της παραγοντοποίησης, εφαρμόζοντας, δηλαδή, το αξίωμα επιμεριστικότητας, προκύπτει η ισοδύναμη συνάρτηση:

$$F = AC'G + ADEG + BC'H + BDEH$$
$$= AG(C' + DE) + BH(C' + DE) = (AG + BH)(C' + DE)$$

Λογικά κυκλώματα πολλών επιπέδων



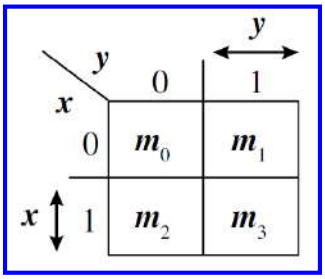
- Η συνάρτηση F = (AG + BH)(C' + DE) που προέκυψε, σύμφωνα και με την προτεραιότητα τελεστών της άλγεβρας Boole, απαιτεί για την υλοποίησή της, τρία επίπεδα πυλών.
- Το 1ο αφορά την παραγωγή των τριών λογικών γινομένων δύο μεταβλητών που περιλαμβάνονται στη συνάρτηση και την παραγωγή της συμπληρωματικής μορφής της μεταβλητής εισόδου C, το 2ο αφορά την παραγωγή των λογικών αθροισμάτων και το 3ο την παραγωγή του συνολικού λογικού γινομένου.
- Καταλήξαμε, λοιπόν, σε μια υλοποίηση η οποία περιλαμβάνει 7 λογικές πύλες, δηλαδή μία παραπάνω από το πλήθος των πυλών που απαιτεί η υλοποίηση δύο επιπέδων. Ωστόσο, στο λογικό κύκλωμα, καμία λογική πύλη δε διαθέτει περισσότερες από δύο εισόδους, γεγονός που αποτελεί πλεονέκτημα.

Κεφάλαιο 3: Ελαχιστοποίηση λογικών συναρτήσεων

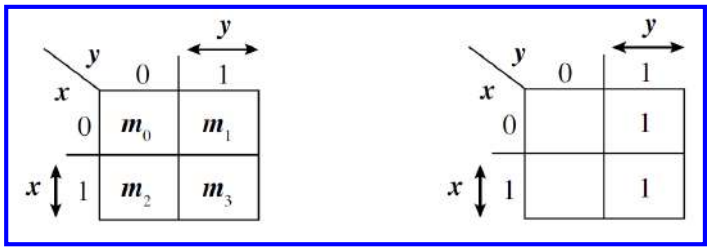
- Οι αλγεβρικοί μετασχηματισμοί που βασίζονται στα αξιώματα και τα θεωρήματα της άλγεβρας Boole μπορούν να οδηγήσουν σε απλοποιημένες μορφές λογικών συναρτήσεων, έτσι ώστε να μπορεί να επιτευχθεί η υλοποίησή τους με μικρότερο αριθμό λογικών πυλών.
- Ωστόσο, η χρήση αλγεβρικών μετασχηματισμών δεν είναι εύκολο να οδηγεί πάντα στην απλούστερη δυνατή μορφή μιας λογικής συνάρτησης και δε συνιστά συστηματική μεθοδολογία απλοποίησης.
- Στα ολοκληρωμένα κυκλώματα πολύ μεγάλης κλίμακας ολοκλήρωσης (very large scale integrated circuits, VLSICs) και στα σύγχρονα ολοκληρωμένα κυκλώματα ειδικού σκοπού (application specific integrated circuits, ASICs) απαιτούνται χαρακτηριστικά όπως υψηλή ταχύτητα, χαμηλή κατανάλωση ενέργειας, μικρή επιφάνεια για την ανάπτυξή τους και χαμηλό κόστος.
- Για να επιτευχθούν τα χαρακτηριστικά αυτά, επιβάλλεται η συστηματική προσπάθεια για μείωση του πλήθους των λογικών πυλών που υλοποιούν τα υποκυκλώματα, τα οποία συνθέτουν τα ολοκληρωμένα κυκλώματα.
- Για το λόγο αυτόν έχουν αναπτυχθεί συστηματικοί τρόποι ελαχιστοποίησης (απλοποίησης) των λογικών συναρτήσεων.
- Αρκετοί από αυτούς μπορούν να υλοποιηθούν σε εργαλεία σχεδιασμού με υπολογιστή (computer-aided design tools, CAD), ώστε να επιτευχθεί η αυτοματοποίησή τους.

Περιγραφή λογικών συναρτήσεων με χάρτες Karnaugh

- Ο γραφικός τρόπος ελαχιστοποίησης, που επινοήθηκε το 1952 από τον Edward Veitch, βελτιώθηκε το 1953 από τον Maurice Karnaugh και αναφέρεται ως μέθοδος του χάρτη Karnaugh (Karnaugh map), αν και δεν είναι κατάλληλος για υλοποίηση σε εργαλεία CAD, βοηθάει στην κατανόηση της βασικής πρακτικής ελαχιστοποίησης λογικών συναρτήσεων.
- Έχουν επίσης αναπτυχθεί αλγοριθμικές μέθοδοι κατάλληλες για υλοποίηση σε αυτοματοποιημένα εργαλεία σχεδιασμού, όπως η μέθοδος που αναπτύχθηκε το 1952 αρχικά από τον Willard Van Orman Quine, τελειοποιήθηκε το 1956 από τον Edward McCluskey και είναι γνωστή ως μέθοδος των Quine και McCluskey.
- Ο χάρτης Karnaugh μιας λογικής συνάρτησης αποτελεί γραφική απεικόνιση του πίνακα αλήθειας της συνάρτησης. Η γραφική αυτή απεικόνιση σχηματίζεται από τετράγωνα, καθένα από τα οποία αντιστοιχεί σε έναν ελαχιστόρο της συνάρτησης.
- Για την περιγραφή μιας συνάρτησης 2 μεταβλητών απαιτείται χάρτης Karnaugh με 4 τετράγωνα, δηλαδή όσοι είναι και οι ελαχιστόροι που μπορούν να συμμετέχουν στη συνάρτηση.



Περιγραφή λογικών συναρτήσεων με χάρτες Karnaugh

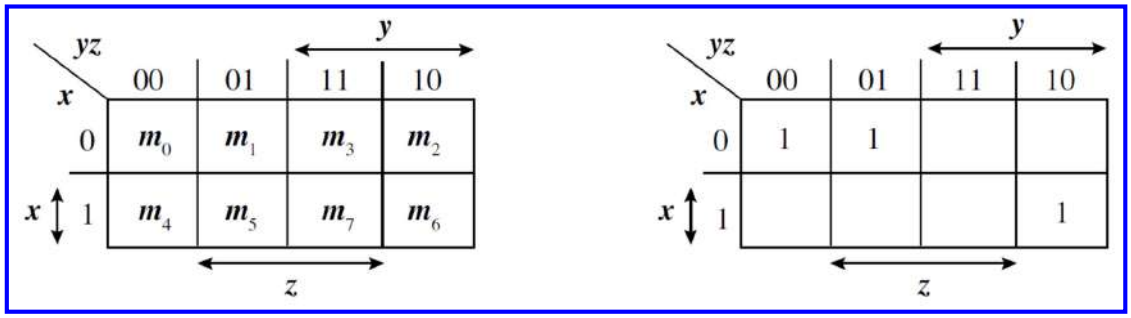


F(x,y) = x′y + xy

- Η πρώτη γραμμή του χάρτη αντιστοιχεί στη συμπληρωματική μορφή της μεταβλητής x, ενώ η δεύτερη γραμμή αντιστοιχεί στην κανονική μορφή της ίδιας μεταβλητής.
- Παρομοίως, οι στήλες του χάρτη αντιστοιχούν, κατά σειρά, στη συμπληρωματική και την κανονική μορφή της μεταβλητής y.
- Οι περιοχές του χάρτη στις οποίες κάθε μεταβλητή λαμβάνει λογική τιμή 1, υποδεικνύονται με βέλη συνδυαζόμενα με το όνομα της μεταβλητής.
- Οι λογικές τιμές 0 και 1 που σημειώνονται στις γραμμές και τις στήλες του χάρτη προσδιορίζουν τις τιμές των δύο μεταβλητών, για τις οποίες ο ελαχιστόρος που αντιστοιχεί σε κάθε τετράγωνο λαμβάνει λογική τιμή 1.
- Οποιαδήποτε λογική συνάρτηση δύο μεταβλητών μπορεί να περιγραφεί με την κατάλληλη τοποθέτηση μονάδων στα τετράγωνα του χάρτη που αντιστοιχούν στους ελαχιστόρους που συμμετέχουν στη συνάρτηση.

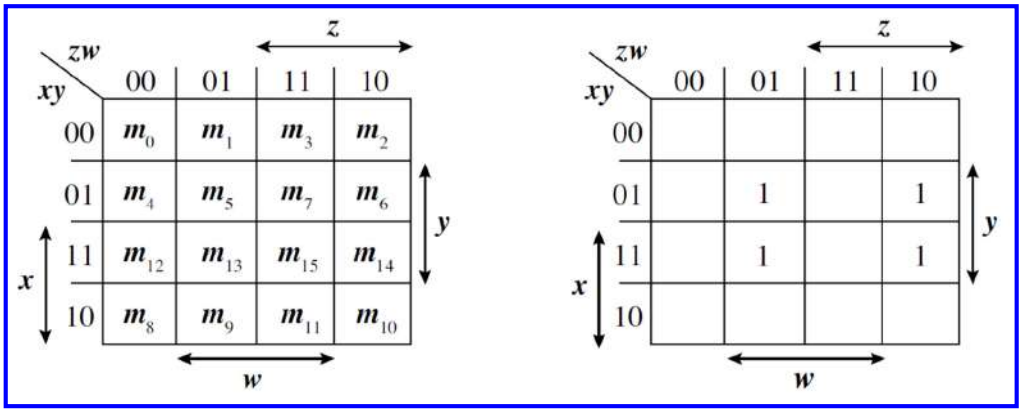
Περιγραφή λογικών συναρτήσεων με χάρτες Karnaugh

- Για την περιγραφή μιας λογικής συνάρτησης 3 μεταβλητών θα πρέπει να χρησιμοποιηθεί χάρτης Karnaugh με 8 τετράγωνα, αφού τρεις μεταβλητές συνιστούν 8 ελαχιστόρους.
- Οι λογικές τιμές των μεταβλητών y και z, στις οποίες αντιστοιχούν οι στήλες του χάρτη, δε διατάσσονται με βάση την κανονική δυαδική ακολουθία αλλά με βάση την ακολουθία του κώδικα Gray.
- Οι λογικές τιμές που σημειώνονται στις γραμμές και τις στήλες του χάρτη προσδιορίζουν τις τιμές των μεταβλητών x και y, z, αντίστοιχα, για τις οποίες ο ελαχιστόρος που αντιστοιχεί σε κάθε τετράγωνο λαμβάνει λογική τιμή 1.
- F(x,y,z) = x′y′z′ + x′y′z + xyz′ = m_0 + m_1 + m_6:



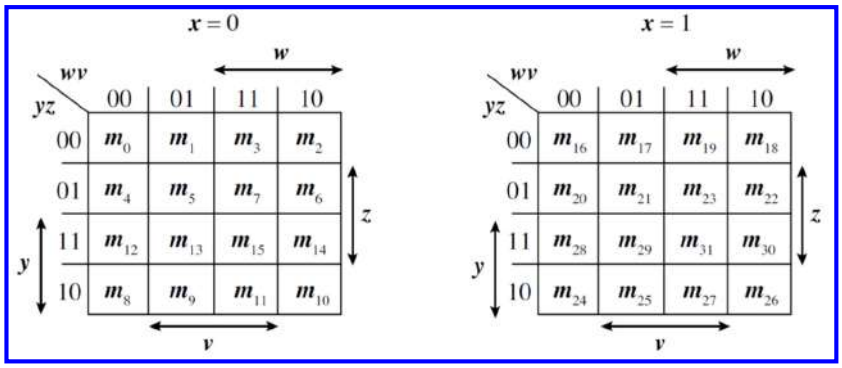
Περιγραφή λογικών συναρτήσεων με χάρτες Karnaugh

- Για την περιγραφή μιας λογικής συνάρτησης 4 μεταβλητών χρησιμοποιείται χάρτης με 16 τετράγωνα.
- Οι λογικές τιμές των ζευγών μεταβλητών, στις οποίες αντιστοιχούν οι γραμμές και οι στήλες του χάρτη, διατάσσονται με βάση την ακολουθία του κώδικα Gray.
- Οι λογικές τιμές που σημειώνονται στις γραμμές και τις στήλες του χάρτη προσδιορίζουν τις τιμές των ζευγών μεταβλητών (x, y) και (z, w), αντίστοιχα, για τις οποίες ο ελαχιστόρος που αντιστοιχεί σε κάθε τετράγωνο λαμβάνει λογική τιμή 1.
- $F(x,y,z,w) = \Sigma(5, 6, 13, 14)$



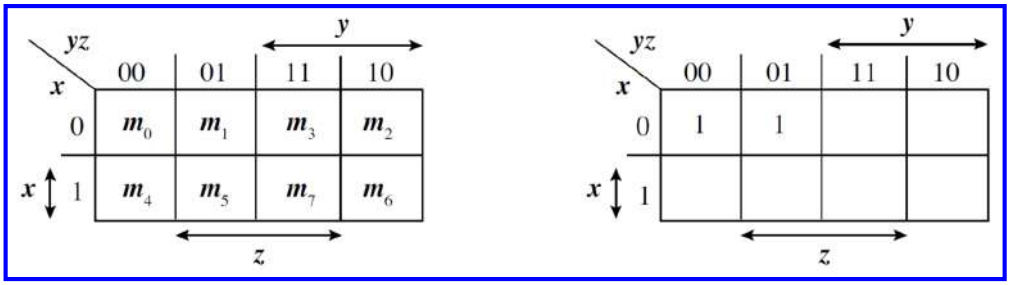
Περιγραφή λογικών συναρτήσεων με χάρτες Karnaugh

- Για την περιγραφή λογικών συναρτήσεων 5 ή 6 μεταβλητών θα μπορούσαν να χρησιμοποιηθούν χάρτες Karnaugh με 32 ή 64 τετράγωνα, αντίστοιχα.
- Ωστόσο, οι χάρτες αυτοί δεν είναι εύχρηστοι για την εφαρμογή της μεθοδολογίας ελαχιστοποίησης λογικών συναρτήσεων.
- Για την περιγραφή συναρτήσεων 5 μεταβλητών μπορούν να χρησιμοποιηθούν δύο χάρτες 4 μεταβλητών. Για τον 1ο χάρτη, σε μία από τις 5 μεταβλητές τίθεται η λογική τιμή 0, ενώ για το 2ο χάρτη τίθεται στην ίδια μεταβλητή η λογική τιμή 1.
- Με τον τρόπο αυτόν αναπτύσσουμε δύο χάρτες για τις υπόλοιπες 4 μεταβλητές.
- Τα τετράγωνα του 1ου χάρτη αντιστοιχούν στους ελαχιστόρους με τη μεταβλητή x' και τα τετράγωνα του 2ου χάρτη αντιστοιχούν στους ελαχιστόρους με την x.



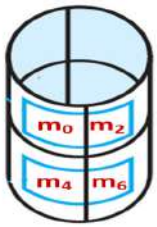
Ελαχιστοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

- Οι ελαχιστόροι που αντιστοιχούν σε 2 γειτονικά τετράγωνα ενός χάρτη (δηλαδή 2 τετράγωνα με κοινή πλευρά) διαφέρουν μόνο κατά μία μεταβλητή, η οποία συμμετέχει με τη συμπληρωματική της μορφή στον ελαχιστόρο που αντιστοιχεί στο ένα τετράγωνο και με την κανονική της μορφή στον ελαχιστόρο που αντιστοιχεί στο γειτονικό του τετράγωνο.
- Στον χάρτη 3 μεταβλητών, οι ελαχιστόροι m_0 και m_1 που αντιστοιχούν σε τετράγωνα με κοινή πλευρά διαφέρουν στο ότι η μεταβλητή z συμμετέχει με τη συμπληρωματική της μορφή στον m_0 και με την κανονική της μορφή στον m_1.
- Το λογικό άθροισμα των όρων αυτών είναι: $m_0 + m_1 = x'y'z' + x'y'z = x'y'(z' + z) = x'y'$. Αυτό σημαίνει ότι κατά τη λογική άθροιση ελαχιστόρων που αντιστοιχούν σε γειτονικά τετράγωνα, απαλείφεται η μεταβλητή κατά την οποία αυτοί διαφέρουν.



Ελαχιστοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

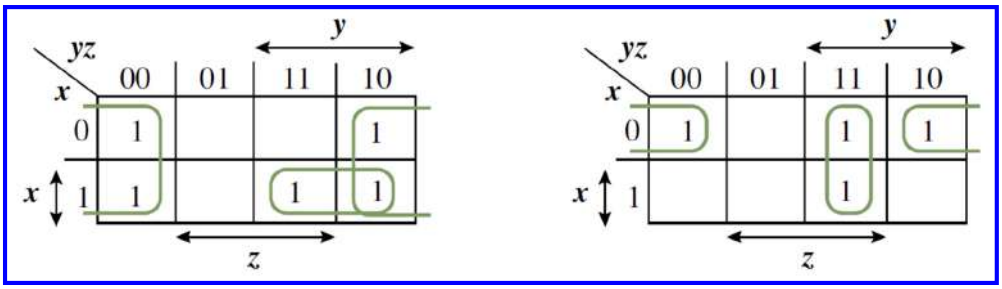
- Στο χάρτη Karnaugh 3 μεταβλητών εκτός των γειτονικών τετραγώνων συναντώνται και άλλα τετράγωνα των οποίων οι αντίστοιχοι ελαχιστόροι διαφέρουν κατά μία μεταβλητή.
- Τέτοια ζεύγη τετραγώνων είναι εκείνα που αντιστοιχούν στους ελαχιστόρους m_0, m_2 και στους m_4, m_6.
- Τα τετράγωνα καθενός από τα ζεύγη αυτά λαμβάνονται ως γειτονικά, εάν θεωρήσουμε ότι η αριστερή και η δεξιά πλευρά του χάρτη εφάπτονται, σχηματίζοντας κύλινδρο.



- Τα λογικά γινόμενα που αντιστοιχούν σε γειτονικά ζεύγη τετραγώνων διαφέρουν κατά μία μόνο μεταβλητή, συνεπώς, το άθροισμα των ελαχιστόρων που αντιστοιχούν σε μία τέτοια τετράδα τετραγώνων έχει αποτέλεσμα έναν όρο μιας μόνο μεταβλητής.
- Στην περίπτωση που περιέχονται μονάδες και στα 8 τετράγωνα του χάρτη, η λογική συνάρτηση ισούται με 1, αφού συμμετέχουν σε αυτήν όλοι οι ελαχιστόροι.
- Ελαχιστοποίηση συνάρτησης 3 μεταβλητών μορφής αθροίσματος γινομένων: (α) σχηματίζουμε το χάρτη, (β) επιλέγουμε στο χάρτη ζεύγη γειτονικών τετραγώνων που περιέχουν μονάδες και, εάν υπάρχουν γειτονικά ζεύγη που περιέχουν μονάδες, επιλέγουμε την τετράδα τετραγώνων που αυτά σχηματίζουν, (γ) εξάγουμε την ελαχιστοποιημένη συνάρτηση σε μορφή αθροίσματος γινομένων, λαμβάνοντας υπόψη ότι κάθε ζεύγος γειτονικών τετραγώνων οδηγεί στην απαλοιφή μιας μεταβλητής και ότι κάθε τετράδα που αποτελείται από γειτονικά ζεύγη οδηγεί σε όρο μιας μεταβλητής.

Ελαχιστοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

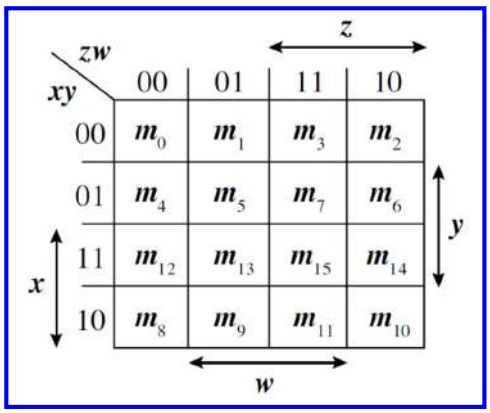
- Το πλήθος των τετραγώνων μιας ομάδας θα πρέπει να ισούται με δύναμη του 2 και κάθε ομάδα είναι επιθυμητό να περιλαμβάνει τον μέγιστο δυνατό αριθμό τετραγώνων που περιέχουν μονάδα.
- Βασικός στόχος της διαδικασίας ελαχιστοποίησης είναι η συμπερίληψη όλων των τετραγώνων που περιέχουν μονάδες σε μία τουλάχιστον ομάδα.
- Ένα τετράγωνο που περιέχει μονάδα μπορεί να ανήκει σε περισσότερες από μία ομάδες.
- Για την εξαγωγή της ελαχιστοποιημένης συνάρτησης, θα πρέπει να επιλέγονται ομάδες τετραγώνων που προκαλούν τις λιγότερες δυνατές επικαλύψεις των μονάδων του χάρτη που περιγράφει τη συνάρτηση.
- Ελαχιστοποίηση των συναρτήσεων $F_1(x,y,z) = x'y'z' + xy'z' + xyz + x'yz' + xyz'$ και $F_2(x,y,z) = x'y'z' + xyz + x'yz + x'yz'$:

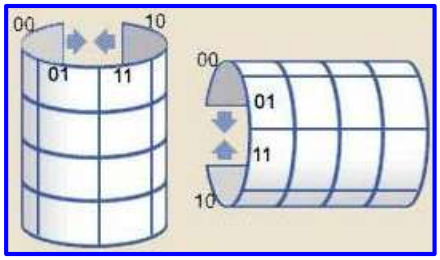


$F_1(x,y,z) = z' + xy$ $F_2(x,y,z) = yz + x'z'$

Ελαχιστοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

- Στον χάρτη Karnaugh 4 μεταβλητών, το άθροισμα των ελαχιστόρων που αντιστοιχούν σε δύο γειτονικά τετράγωνα έχει ως αποτέλεσμα λογικό γινόμενο 3 μεταβλητών.
- Τα λογικά γινόμενα που αντιστοιχούν σε γειτονικά ζεύγη τετραγώνων (τετράδα τετραγώνων) διαφέρουν κατά 2 μόνο μεταβλητές, με αποτέλεσμα το λογικό άθροισμα των γινομένων αυτών να απλοποιείται σε ένα γινόμενο 2 μεταβλητών.
- Το άθροισμα των ελαχιστόρων που αντιστοιχούν σε μία οκτάδα τετραγώνων που σχηματίζεται από 2 τετράδες με τα παραπάνω χαρακτηριστικά, απλοποιείται σε έναν όρο μιας μεταβλητής.
- Οι ακραίες στήλες περιλαμβάνουν γειτονικά μεταξύ τους τετράγωνα, αφού οι ελαχιστόροι που αντιστοιχούν σε αυτά διαφέρουν κατά μία μόνο μεταβλητή. Το ίδιο συμβαίνει και με τις ακραίες γραμμές του χάρτη.
- Τα τετράγωνα που βρίσκονται στις 4 γωνίες του χάρτη σχηματίζουν μία τετράδα τετραγώνων που αντιστοιχεί σε λογικό γινόμενο δύο μεταβλητών.





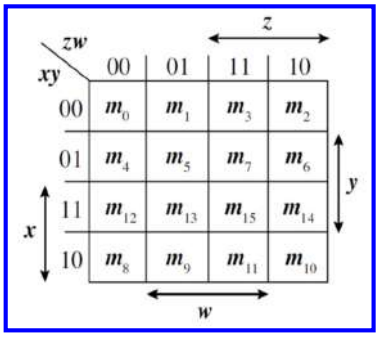
Ελαχιστοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

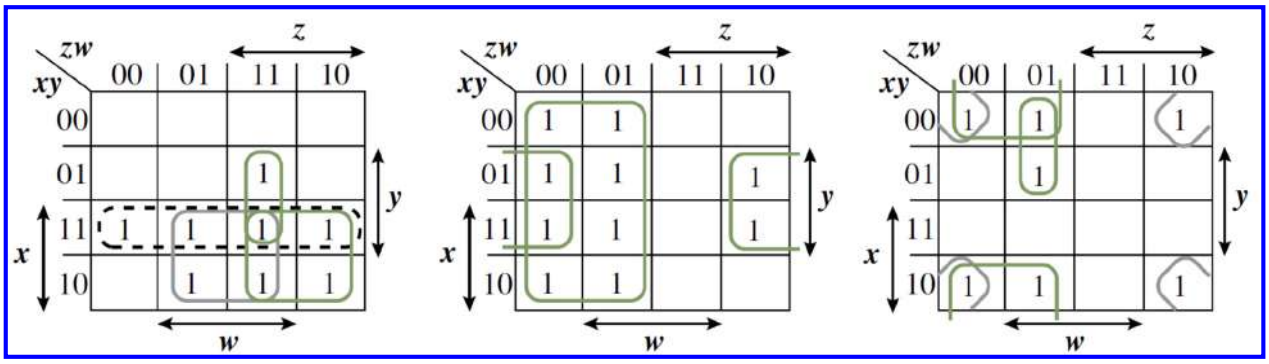
Ελαχιστοποίηση των συναρτήσεων:

$F_3(x,y,z,w) = \Sigma(7, 9, 10, 11, 12, 13, 14, 15) = xy + xw + xz + yzw$

$F_4(x,y,z,w) = \Sigma(0, 1, 4, 5, 6, 8, 9, 12, 13, 14) = z' + yw'$

$F_5(x,y,z,w) = \Sigma(0, 1, 2, 5, 8, 9, 10) = y'z' + y'w' + x'z'w$



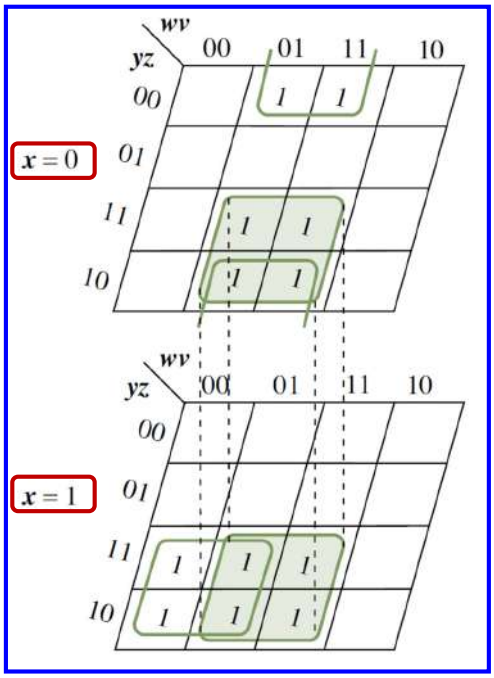


Ελαχιστοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

- Για την περιγραφή συναρτήσεων 5 μεταβλητών χρησιμοποιούνται 2 χάρτες 4 μεταβλητών.
- Για τον πρώτο χάρτη, σε μία από τις πέντε μεταβλητές τίθεται η λογική τιμή 0, ενώ για το δεύτερο χάρτη τίθεται στην ίδια μεταβλητή η λογική τιμή 1.
- Για την ελαχιστοποίηση λογικών συναρτήσεων 5 μεταβλητών ακολουθούνται οι ίδιες αρχές με εκείνες που αναφέρθηκαν για τις συναρτήσεις 4 μεταβλητών.
- Το νέο στοιχείο είναι ότι κάθε τετράγωνο του πρώτου χάρτη θεωρείται γειτονικό του αντίστοιχου τετραγώνου του δεύτερου χάρτη, αφού τα τετράγωνα αυτά διαφέρουν μόνο κατά τη μεταβλητή που λαμβάνει λογική τιμή 0 στον πρώτο χάρτη και λογική τιμή 1 στον δεύτερο χάρτη.

$F(x,y,z,w,v) = \Sigma(1, 3, 9, 11, 13, 15, 24, 25, 27, 28, 29, 31) =$

$x'z'v + yv + xyw'$



Βασικές οδηγίες για τη μέθοδο χάρτη Karnaugh

- Κάθε τετράγωνο του χάρτη Karnaugh που περιέχει μονάδα αντιστοιχεί σε έναν ελαχιστόρο της λογικής συνάρτησης.
- Για κάθε τετράγωνο ενός χάρτη Karnaugh n μεταβλητών υπάρχουν n γειτονικά τετράγωνα και κάθε ζεύγος τετραγώνων αντιστοιχεί σε ελαχιστόρους που διαφέρουν κατά μία μόνο μεταβλητή.
- Κατά την ελαχιστοποίηση μιας λογικής συνάρτησης, ο αριθμός των τετραγώνων που ομαδοποιείται είναι πάντα δύναμη του 2.
- Η δημιουργία ζεύγους γειτονικών τετραγώνων που περιέχουν μονάδες οδηγεί στην απαλοιφή μιας μεταβλητής, η δημιουργία τετράδας τετραγώνων οδηγεί στην απαλοιφή δύο μεταβλητών και γενικότερα η ομαδοποίηση 2^n τετραγώνων οδηγεί στην απαλοιφή n μεταβλητών.
- Επιδιώκεται η δημιουργία ομάδων με το μεγαλύτερο δυνατό πλήθος τετραγώνων που περιέχουν μονάδες, έτσι ώστε η μορφή της συνάρτησης που θα προκύψει να περιλαμβάνει λογικά γινόμενα με όσο το δυνατόν λιγότερες μεταβλητές.
- Επιδιώκεται η δημιουργία του μικρότερου δυνατού πλήθους ομάδων τετραγώνων, έτσι ώστε η μορφή της συνάρτησης που θα προκύψει να περιλαμβάνει όσο το δυνατόν λιγότερα λογικά γινόμενα.

Βασικές οδηγίες για τη μέθοδο χάρτη Karnaugh

- Κάθε τετράγωνο που περιέχει μονάδα και αντιστοιχεί σε έναν ελαχιστόρο της συνάρτησης θα πρέπει να περιλαμβάνεται σε μία τουλάχιστον ομάδα.
- Κάθε τετράγωνο που περιέχει μονάδα μπορεί να περιλαμβάνεται σε περισσότερες από μία ομάδες, εάν αυτό εξυπηρετεί τους στόχους του μικρότερου δυνατού πλήθους λογικών γινομένων και του μικρότερου δυνατού πλήθους μεταβλητών ανά λογικό γινόμενο.
- Ωστόσο, θα πρέπει να επιλέγονται ομάδες τετραγώνων με τις λιγότερες δυνατές επικαλύψεις.
- Η ομαδοποίηση των τετραγώνων του χάρτη που περιέχουν μονάδες είναι προτιμότερο να ξεκινά με τα «μοναχικά» τετράγωνα (δηλαδή εκείνα που παρουσιάζουν περιορισμένη γειτνίαση με άλλα τετράγωνα που περιέχουν μονάδες).
- Στη συνέχεια, η ομαδοποίηση επεκτείνεται στις περιοχές του χάρτη όπου υπάρχει συγκέντρωση τετραγώνων που περιέχουν μονάδες.

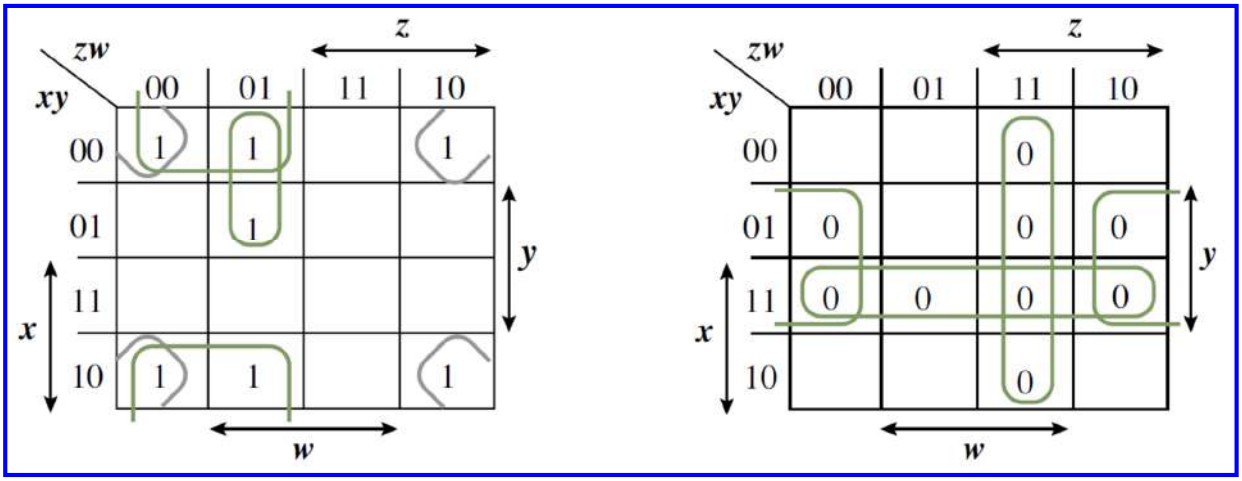
Υλοποίηση ελαχιστοποιημένων συναρτήσεων

- Η ελαχιστοποιημένη μορφή μιας συνάρτησης που προκύπτει από κατάλληλη ομαδοποίηση των τετραγώνων του χάρτη Karnaugh που περιέχουν μονάδες, εξάγεται σε μορφή αθροίσματος γινομένων.
- Έτσι, μπορούν εύκολα να προκύψουν λογικά κυκλώματα αποτελούμενα από δύο επίπεδα πυλών σε διάταξη AND-OR ή μόνο από πύλες NAND, που υλοποιούν την ελαχιστοποιημένη μορφή της συνάρτησης.
- Τα τετράγωνα του χάρτη που περιγράφει μια λογική συνάρτηση, τα οποία δεν περιέχουν μονάδες, αντιστοιχούν στους ελαχιστόρους που δεν περιλαμβάνονται στη συνάρτηση.
- Το άθροισμα των ελαχιστόρων αυτών συνιστά τη συμπληρωματική συνάρτηση.
- Εάν, λοιπόν, τοποθετήσουμε μηδενικά στα άδεια τετράγωνα του χάρτη και τα ομαδοποιήσουμε με βάση τη διαδικασία που παρουσιάστηκε προηγουμένως, μπορούμε να καταλήξουμε στην ελαχιστοποιημένη μορφή της συμπληρωματικής συνάρτησης, σε μορφή αθροίσματος γινομένων.
- Στη συνέχεια, εάν εφαρμόσουμε σε αυτήν το θεώρημα De Morgan, μπορούμε να εξαγάγουμε την ελαχιστοποιημένη αρχική συνάρτηση σε μορφή γινομένου αθροισμάτων, ώστε να προκύψουν λογικά κυκλώματα με δύο επίπεδα πυλών, σε διάταξη OR-AND ή μόνο με πύλες NOR.

Υλοποίηση ελαχιστοποιημένων συναρτήσεων

$$F(x,y,z,w) = \Sigma(0, 1, 2, 5, 8, 9, 10) = y'z' + y'w' + x'z'w$$

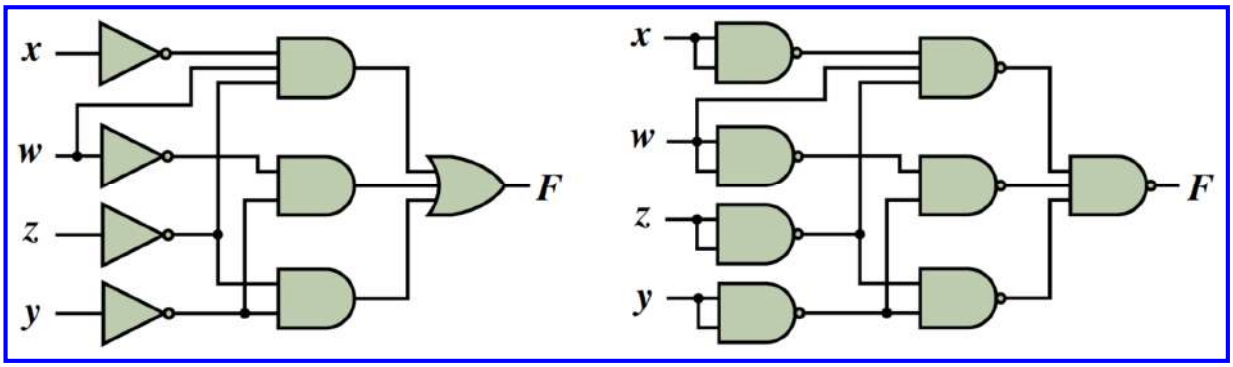
$$F'(x,y,z,w) = xy + zw + yw'$$



Υλοποίηση ελαχιστοποιημένων συναρτήσεων

$F(x,y,z,w) = y'z' + y'w' + x'z'w$

Θεώρημα διπλής άρνησης & θεώρημα De Morgan $\Rightarrow F(x,y,z,w) = [(y'z')'(y'w')'(x'z'w)']'$



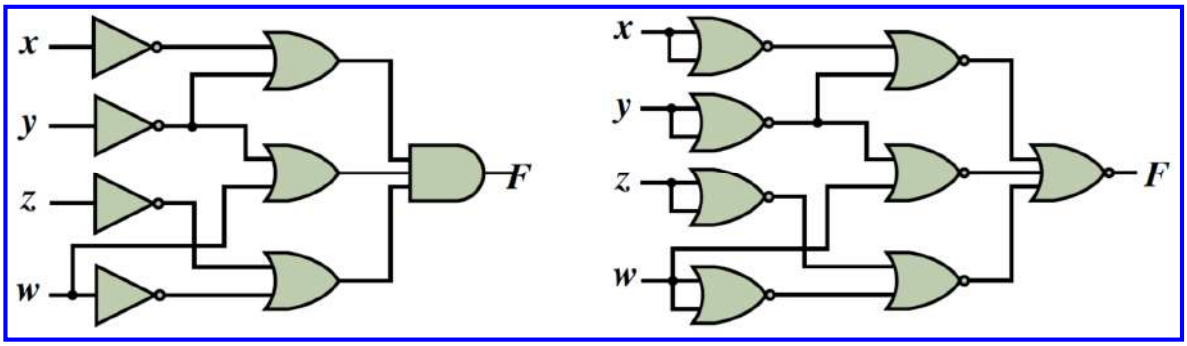
AND-OR NAND-NAND

Υλοποίηση ελαχιστοποιημένων συναρτήσεων

$F(x,y,z,w) = (xy + zw + yw')'$

Θεώρημα De Morgan $\Rightarrow F(x,y,z,w) = (x' + y')(z' + w')(y' + w)$

Διπλή άρνηση & De Morgan $\Rightarrow F(x,y,z,w) = [(x' + y')' + (z' + w')' + (y' + w)']'$



OR-AND NOR-NOR

Μερικώς καθορισμένες συναρτήσεις

- Σε αρκετές περιπτώσεις λογικών κυκλωμάτων υπάρχουν συνδυασμοί εισόδων οι οποίοι δεν μπορούν να συμβούν ή δεν είναι επιτρεπτοί.
- Για παράδειγμα, υποθέτουμε ότι τρεις μεταβλητές x, y και z αντιστοιχούν στους ισάριθμους λαμπτήρες (κόκκινο, πορτοκαλί, πράσινο) ενός φαναριού κυκλοφορίας και ότι η καθεμία από αυτές λαμβάνει λογική τιμή 0 ή 1, όταν ο αντίστοιχος λαμπτήρας είναι σβηστός ή ανοιχτός, αντίστοιχα.
- Οι επιτρεπτοί συνδυασμοί τιμών που μπορούν να λάβουν οι μεταβλητές αυτές είναι εκείνοι στους οποίους μόνο μία μεταβλητή έχει λογική τιμή 1 και οι υπόλοιπες δύο έχουν λογική τιμή 0, αφού για την ορθή λειτουργία του φαναριού, μόνο ένας λαμπτήρας μπορεί να είναι ανοιχτός.
- Οι συνδυασμοί, δηλαδή, τιμών των μεταβλητών (x,y,z) = 000, 011, 101, 110 111 δεν επιτρέπονται ή δε χρησιμοποιούνται και αναφέρονται ως αδιάφορες λογικές συνθήκες (don't care logic conditions).
- Ένα λογικό κύκλωμα με εισόδους τις μεταβλητές x, y και z μπορεί να σχεδιαστεί αγνοώντας τις καταστάσεις αυτές.
- Οι ελαχιστόροι που αντιστοιχούν στις αδιάφορες λογικές συνθήκες, αναφέρονται ως αδιάφοροι όροι (don't care terms) και μια λογική συνάρτηση που περιλαμβάνει αδιάφορους όρους αναφέρεται ως μερικώς καθορισμένη (incompletely specified).

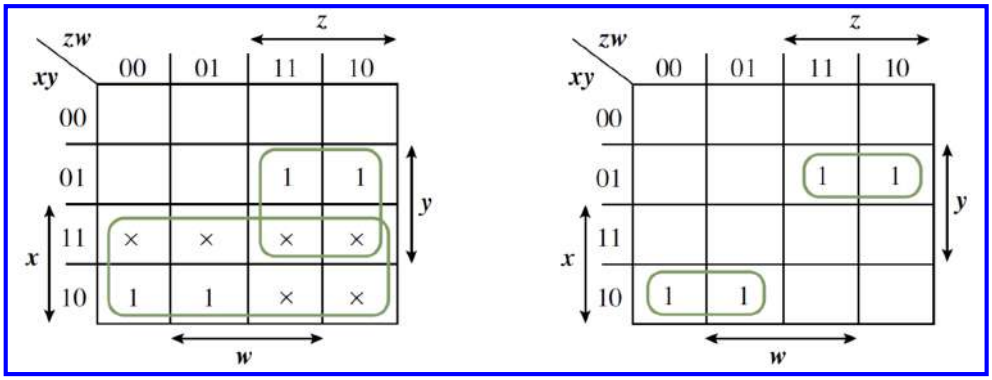
Μερικώς καθορισμένες συναρτήσεις

- Οι αδιάφοροι όροι μπορούν να χρησιμοποιηθούν, ώστε να προκύψουν λογικά κυκλώματα με μικρότερο αριθμό πυλών.
- Η λογική τιμή των συναρτήσεων που αντιστοιχεί στους αδιάφορους όρους μπορεί να θεωρηθεί ότι είναι 0 ή 1, ανάλογα με το ποια από τις δύο λογικές τιμές είναι κατάλληλη για ευρύτερη απλοποίηση της λογικής συνάρτησης.
- Στον πίνακα αλήθειας ή στο χάρτη Karnaugh που περιγράφουν μια λογική συνάρτηση, για να διακρίνουμε έναν αδιάφορο όρο από τις τιμές της συνάρτησης που αντιστοιχούν σε επιτρεπτούς συνδυασμούς εισόδων, χρησιμοποιούμε το σύμβολο ×.
- Κατά την ομαδοποίηση γειτονικών τετραγώνων του χάρτη Karnaugh μιας λογικής συνάρτησης η οποία γίνεται με στόχο την ελαχιστοποίησή της, μπορούμε να επιλέξουμε να χρησιμοποιήσουμε για κάθε αδιάφορο όρο (×) λογική τιμή 0 ή 1, ανάλογα με το ποια από τις δύο τιμές του μπορεί να μας οδηγήσει στην απλούστερη μορφή της συνάρτησης.
- Στην έκφραση μιας λογικής συνάρτησης που περιλαμβάνει αδιάφορους όρους, θα πρέπει να περιλαμβάνονται και οι αδιάφοροι όροι.
- Για παράδειγμα, η συνάρτηση F(x,y,z) = Σ(1, 2, 5) + d(0, 3, 4) ή F(x,y,z) = Σ(1, 2, 5) + ×(0, 3, 4), περιλαμβάνει τους ελαχιστόρους m_1, m_2, m_5 και ως αδιάφορους όρους τα γινόμενα x'y'z', x'yz, xy'z'.

Μερικώς καθορισμένες συναρτήσεις

- Παράδειγμα: Σχεδίαση λογικού κυκλώματος που λαμβάνει στις εισόδους του x, y, z, w, τέσσερα δυαδικά ψηφία που αντιστοιχούν σε δυαδικά κωδικοποιημένους δεκαδικούς αριθμούς (δηλαδή στα δεκαδικά ψηφία 0 έως 9 κωδικοποιημένα σύμφωνα με τον κώδικα BCD) και στην έξοδό του F παράγεται λογική τιμή 0, όταν οι είσοδοι αποτελούν κωδικοποίηση δεκαδικού αριθμού μικρότερου ή ίσου του 5, και λογική τιμή 1, όταν οι είσοδοι αποτελούν κωδικοποίηση δεκαδικού αριθμού μεγαλύτερου του 5.
- Στον κώδικα BCD δε χρησιμοποιούνται 6 από τους 16 δυνατούς συνδυασμούς τιμών 4 δυαδικών ψηφίων.
- Αυτό έχει αποτέλεσμα κάθε λογική συνάρτηση με μεταβλητές που αντιστοιχούν σε δυαδικά κωδικοποιημένα δεκαδικά ψηφία να είναι μερικώς ορισμένη και να περιλαμβάνει 6 αδιάφορους όρους, που αντιστοιχούν στους δυαδικούς συνδυασμούς (x,y,z,w) = 1010, 1011, 1100, 1101, 1110 και 1111, δηλαδή στους συνδυασμούς με ισοδύναμο δεκαδικό αριθμό μεγαλύτερο του 9.
- Στο χάρτη Karnaugh της συνάρτησης θα πρέπει να σημειωθούν μονάδες στα τετράγωνα που αντιστοιχούν στους ελαχιστόρους που αφορούν δυαδικούς συνδυασμούς με ισοδύναμο δεκαδικό αριθμό από 6 έως και 9.
- Επίσης, θα πρέπει να σημειωθούν με το σύμβολο × οι αδιάφοροι όροι που αφορούν τους δυαδικούς συνδυασμούς που δε χρησιμοποιούνται στον κώδικα BCD.

Μερικώς καθορισμένες συναρτήσεις



- Κατά την ελαχιστοποίηση της συνάρτησης, επιλέγουμε να χρησιμοποιήσουμε για όλους τους αδιάφορους όρους τη λογική τιμή 1, αφού με την επιλογή αυτή δημιουργούνται μία οκτάδα και μία τετράδα γειτονικών τετραγώνων που οδηγούν σε συνάρτηση [F(x,y,z,w) = x + yz] που συνίσταται από το άθροισμα ενός όρου μιας μεταβλητής (x) και ενός γινομένου δύο μεταβλητών (yz).
- Η απλοποίηση χωρίς τη χρησιμοποίηση αδιάφορων όρων καταλήγει στη συνάρτηση F(x,y,z,w) = xy'z' + x'yz.
- Η 1η μορφή υλοποιείται με 1 πύλη AND και 1 πύλη OR δύο εισόδων, ενώ η 2η μορφή απαιτεί πιο σύνθετο λογικό κύκλωμα (3 αντιστροφείς, 2 πύλες AND 3 εισόδων, 1 πύλη OR 2 εισόδων).

Πρώτοι και βασικοί πρώτοι συνεπαγωγοί συνάρτησης

- Όταν ομαδοποιούμε γειτονικά τετράγωνα με μονάδες στο χάρτη Karnaugh μιας συνάρτησης, θα πρέπει να εξασφαλίσουμε ότι στις ομάδες που σχηματίζουμε περιλαμβάνεται ο μέγιστος δυνατός αριθμός τετραγώνων και ότι τα τετράγωνα που αντιστοιχούν σε όλους τους ελαχιστόρους που συμμετέχουν σε μια συνάρτηση ανήκουν σε μία τουλάχιστον ομάδα.
- Επίσης, ένα τετράγωνο που περιέχει μονάδα μπορεί να ανήκει σε περισσότερες από μία ομάδες, για την εξαγωγή, ωστόσο, της ελαχιστοποιημένης συνάρτησης θα πρέπει να επιλέγονται ομάδες τετραγώνων που προκαλούν τις λιγότερες δυνατές επικαλύψεις των μονάδων του χάρτη.
- Τα λογικά γινόμενα που συμμετέχουν σε μια λογική συνάρτηση αναφέρονται και ως συνεπαγωγοί (implicants) της συνάρτησης, λόγω του ότι όταν αυτά λαμβάνουν λογική τιμή 1, τότε και η συνάρτηση λαμβάνει λογική τιμή 1, δηλαδή η λογική συνάρτηση συνεπάγεται από αυτά τα λογικά γινόμενα.
- Ως πρώτος συνεπαγωγός (prime implicant) αναφέρεται το λογικό γινόμενο το οποίο, εάν απαλειφθεί από αυτό μία μεταβλητή, παύει να είναι συνεπαγωγός της λογικής συνάρτησης.
- Συνδυάζοντας τον ορισμό αυτόν με τη μέθοδο του χάρτη Karnaugh, προκύπτει ότι πρώτοι συνεπαγωγοί είναι τα λογικά γινόμενα που προκύπτουν από την ομαδοποίηση του μέγιστου δυνατού αριθμού τετραγώνων του χάρτη που περιέχουν μονάδες.

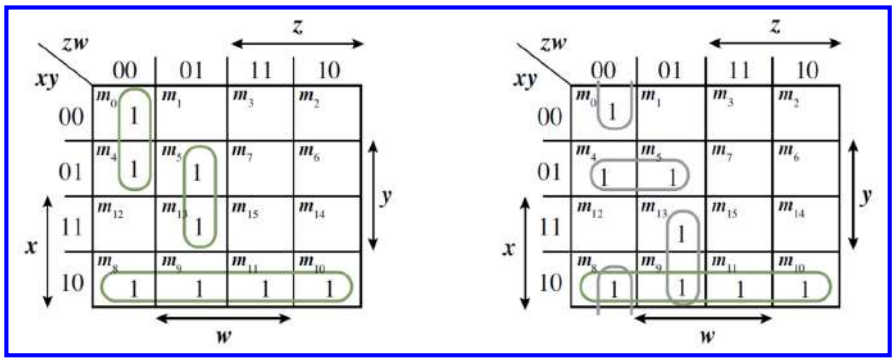
Πρώτοι και βασικοί πρώτοι συνεπαγωγοί συνάρτησης

- Οι συνεπαγωγοί αυτοί εκφράζουν ή καλύπτουν τους ελαχιστόρους της λογικής συνάρτησης που αντιστοιχούν στα ομαδοποιημένα γειτονικά τετράγωνα.
- Κατά την ελαχιστοποίηση μιας λογικής συνάρτησης με τη μέθοδο του χάρτη, θα πρέπει, λοιπόν, να εντοπίζονται οι πρώτοι συνεπαγωγοί της συνάρτησης και από αυτούς να επιλέγονται, σε πρώτη φάση, εκείνοι που εκφράζουν κάποιον ή κάποιους ελαχιστόρους, οι οποίοι δεν εκφράζονται από άλλον πρώτο συνεπαγωγό.
- Οι πρώτοι συνεπαγωγοί, οι οποίοι εκφράζουν έναν τουλάχιστον ελαχιστόρο μιας συνάρτησης που δεν εκφράζεται από άλλον πρώτο συνεπαγωγό, αναφέρονται ως βασικοί πρώτοι συνεπαγωγοί (essential prime implicants) της συνάρτησης.
- Μετά την επιλογή των βασικών πρώτων συνεπαγωγών, θα πρέπει να επιλέγονται οι πρώτοι συνεπαγωγοί που εκφράζουν ή καλύπτουν τους υπόλοιπους ελαχιστόρους, ώστε να προκύπτει η απλούστερη δυνατή (ελαχιστοποιημένη) μορφή της συνάρτησης.
- Οι ομάδες τετραγώνων του χάρτη Karnaugh που περιέχουν μονάδες οι οποίες αντιστοιχούν στους πρώτους συνεπαγωγούς που επιλέγονται, θα πρέπει να προκαλούν τις λιγότερες δυνατές επικαλύψεις.
- Σε αρκετές περιπτώσεις, ενδέχεται να υπάρχουν περισσότερες από μία επιλογές πρώτων συνεπαγωγών, οι οποίες οδηγούν σε ελαχιστοποιημένη έκφραση μιας λογικής συνάρτησης.

Πρώτοι και βασικοί πρώτοι συνεπαγωγοί συνάρτησης

F(x,y,z,w) =

Σ(0, 4, 5, 8, 9, 10, 11, 13)

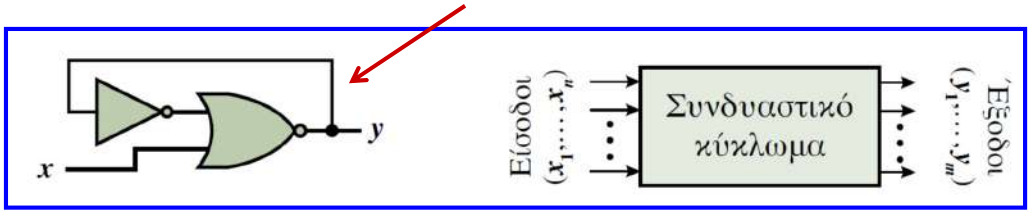


- 6 πρώτοι συνεπαγωγοί: xy′, x′z′w′, yz′w από την τετράδα (m_8, m_9, m_{10}, m_{11}) και από τα ζεύγη (m_0, m_4), (m_5, m_{13}), y′z′w′, x′yz′, xz′w από τα ζεύγη (m_0, m_8), (m_4, m_5) και (m_9, m_{13}).
- Από τους 6, μόνο ο xy′ είναι βασικός πρώτος συνεπαγωγός, αφού είναι ο μοναδικός που καλύπτει τους ελαχιστόρους m_{10} και m_{11}.
- Θα πρέπει να επιλέξουμε από τους 5 πρώτους συνεπαγωγούς, εκείνους που εκφράζουν όλους τους ελαχιστόρους που δεν καλύπτονται από το βασικό πρώτο συνεπαγωγό.
- Η 1η επιλογή δεν προκαλεί επικάλυψη ομάδων τετραγώνων και οδηγεί στη μορφή της συνάρτησης F(x,y,z,w) = xy′ + x′z′w′ + yz′w (απλούστερη δυνατή ή ελαχιστοποιημένη).
- Η 2η επιλογή προκαλεί επικαλύψεις ομάδων τετραγώνων και οδηγεί στη μορφή της συνάρτησης F(x,y,z,w) = xy′ + y′z′w′ + x′yz′ + xz′w.

Κεφάλαιο 4: Συνδυαστικά κυκλώματα

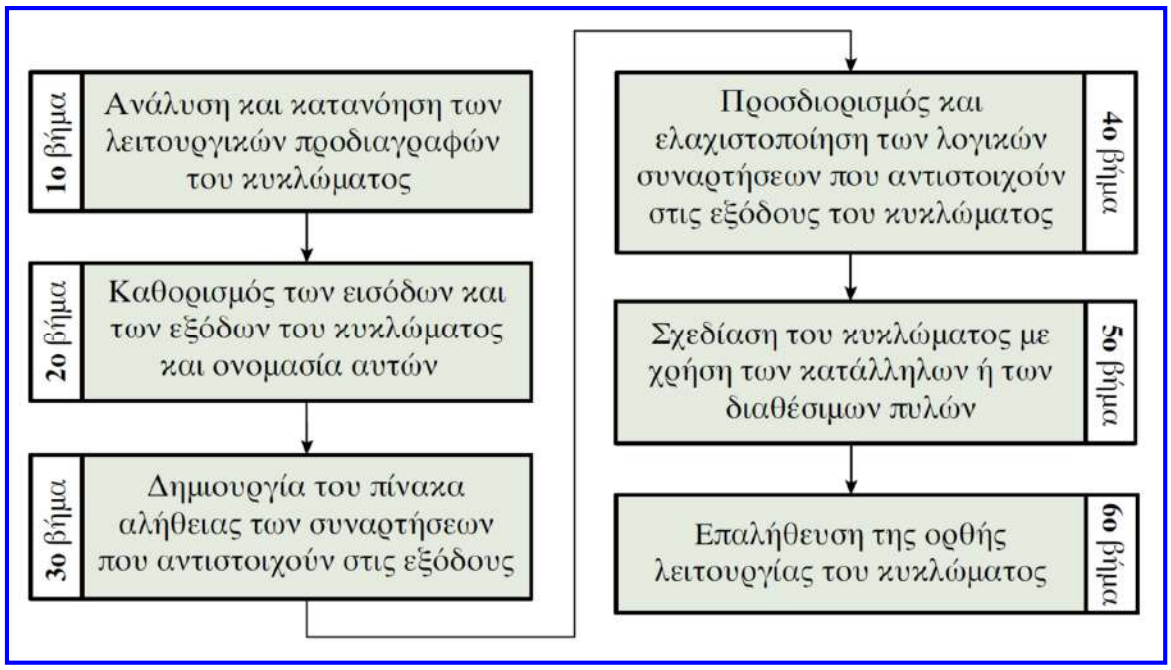
Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

- Τα λογικά κυκλώματα που έχουμε μελετήσει, αναφέρονται ως συνδυαστικά κυκλώματα (combinational circuits). Ο χαρακτηρισμός αυτός αφορά λογικά κυκλώματα των οποίων η λογική τιμή της εξόδου ή των εξόδων, κάθε χρονική στιγμή, εξαρτάται μόνο από τη λογική τιμή των εισόδων που εφαρμόζεται σε αυτά την ίδια χρονική στιγμή.
- Οι λογικές πύλες που συνθέτουν ένα συνδυαστικό κύκλωμα διασυνδέονται με τέτοιο τρόπο, ώστε να μη δημιουργείται ανατροφοδότηση.



- Η λειτουργία ενός συνδυαστικού κυκλώματος μπορεί να περιγραφεί με μοναδικό τρόπο από έναν πίνακα αλήθειας με 2^n γραμμές (που αντιστοιχούν στους δυνατούς συνδυασμούς λογικών τιμών των μεταβλητών εισόδου) και n + m στήλες (που αντιστοιχούν στο πλήθος των μεταβλητών εισόδου και εξόδου).
- Επίσης, η λειτουργία του μπορεί να περιγραφεί από m λογικές συναρτήσεις, μία για κάθε μεταβλητή εξόδου.

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων



Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

- Μια πινακοθήκη αποτελείται από 3 αίθουσες, οι οποίες για λόγους ασφαλείας είναι εξοπλισμένες με 3 ανιχνευτές κίνησης, αντίστοιχα. Κάθε νύχτα, στην πινακοθήκη υπάρχει ένας φύλακας που κινείται συνεχώς από αίθουσα σε αίθουσα.
- Θα συνθέσουμε συνδυαστικό κύκλωμα που θα δέχεται ως εισόδους τις εξόδους των 3 ανιχνευτών και θα ενεργοποιεί το συναγερμό, στις περιπτώσεις που ανιχνεύεται κίνηση σε περισσότερες από μία αίθουσες (δηλαδή όταν υπάρχει εισβολέας).
- Το συνδυαστικό κύκλωμα θα ενεργοποιεί επίσης μία φωτεινή ένδειξη στις εγκαταστάσεις της εταιρείας φύλαξης, όταν υπάρχει υποψία εισβολής στην τρίτη αίθουσα (στην οποία εκτίθεται πίνακας μεγάλης αξίας), ώστε να σπεύσει στην πινακοθήκη ενισχυτικό προσωπικό ασφαλείας.
- Ανάλυση προδιαγραφών: οι είσοδοι του κυκλώματος θα πρέπει να είναι 3 (x, y, z), με καθεμία από αυτές να αντιστοιχεί στην έξοδο ενός από τους αισθητήρες κίνησης.
- Θεωρούμε ότι όταν ένας από τους αισθητήρες ανιχνεύει κίνηση, η αντίστοιχη είσοδος του κυκλώματος λαμβάνει λογική τιμή 1, διαφορετικά λαμβάνει τιμή 0. Οι έξοδοι του κυκλώματος θα πρέπει να είναι δύο: η Α θα ενεργοποιεί το συναγερμό της πινακοθήκης και η Β θα ενεργοποιεί τη φωτεινή ένδειξη στο υποκατάστημα της εταιρείας φύλαξης.
- Θεωρούμε, επίσης, ότι οι έξοδοι Α και Β λαμβάνουν λογική τιμή 1, όταν ικανοποιούνται οι συνθήκες που ενεργοποιούν το συναγερμό και τη φωτεινή ένδειξη, αντίστοιχα, διαφορετικά λαμβάνουν λογική τιμή 0.

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Πίνακας αλήθειας των συναρτήσεων $A(x,y,z)$ *και* $B(x,y,z)$

x	y	z	A	B	*Παρατηρήσεις*
0	0	0	×	×	Αδιάφορη συνθήκη, αφού ο φύλακας κινείται συνεχώς από αίθουσα σε αίθουσα
0	0	1	0	0	
0	1	0	0	0	
0	1	1	1	1	Εισβολέας στη 2η ή στην 3η αίθουσα
1	0	0	0	0	
1	0	1	1	1	Εισβολέας στην 1η ή στην 3η αίθουσα
1	1	0	1	0	Εισβολέας στην 1η ή στη 2η αίθουσα
1	1	1	1	1	Εισβολείς σε δύο αίθουσες

Ο συνδυασμός τιμών των εισόδων που αντιστοιχεί σε ανυπαρξία κίνησης σε όλες τις αίθουσες (x = y = z = 0) συνιστά αδιάφορη λογική συνθήκη, αφού δεν μπορεί να συμβεί, λόγω του ότι, με βάση τις προδιαγραφές, ο φύλακας κινείται συνεχώς από αίθουσα σε αίθουσα.

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

- Εξαγωγή αλγεβρικών εκφράσεων των συναρτήσεων A(x,y,z) και B(x,y,z): εντοπίζουμε στον πίνακα αλήθειας τους συνδυασμούς τιμών των μεταβλητών για τους οποίους κάθε συνάρτηση λαμβάνει τιμή 1, εκφράζουμε κάθε συνδυασμό ως το λογικό γινόμενο των εισόδων με τιμή 1 και των συμπληρωμάτων των εισόδων με τιμή 0 και αθροίζουμε τα λογικά γινόμενα που προκύπτουν, ώστε να εκφράσουμε τις λογικές συναρτήσεις σε κανονική μορφή αθροίσματος ελαχίστων όρων.

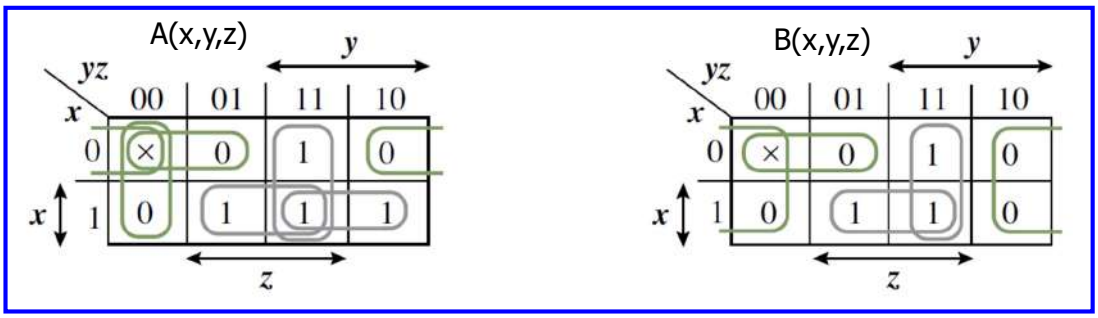
$$A(x,y,z) = x'yz + xy'z + xyz' + xyz$$
$$B(x,y,z) = x'yz + xy'z + xyz$$

- Αξιοποιώντας τους συνδυασμούς τιμών των μεταβλητών για τους οποίους κάθε συνάρτηση λαμβάνει τιμή 0 και το θεώρημα De Morgan, μπορούμε εύκολα να εκφράσουμε τις συναρτήσεις αυτές σε κανονική μορφή γινομένου μεγίστων όρων:

$$A(x,y,z) = (x'y'z + x'yz' + xy'z')' = (x + y + z')(x + y' + z)(x'+ y + z)$$
$$B(x,y,z) = (x'y'z + x'yz' + xy'z' + xyz')' = (x + y + z')(x + y' + z)(x'+ y + z)(x'+ y' + z)$$

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Ελαχιστοποίηση αλγεβρικών εκφράσεων των συναρτήσεων



$$A(x,y,z) = xy + xz + yz$$

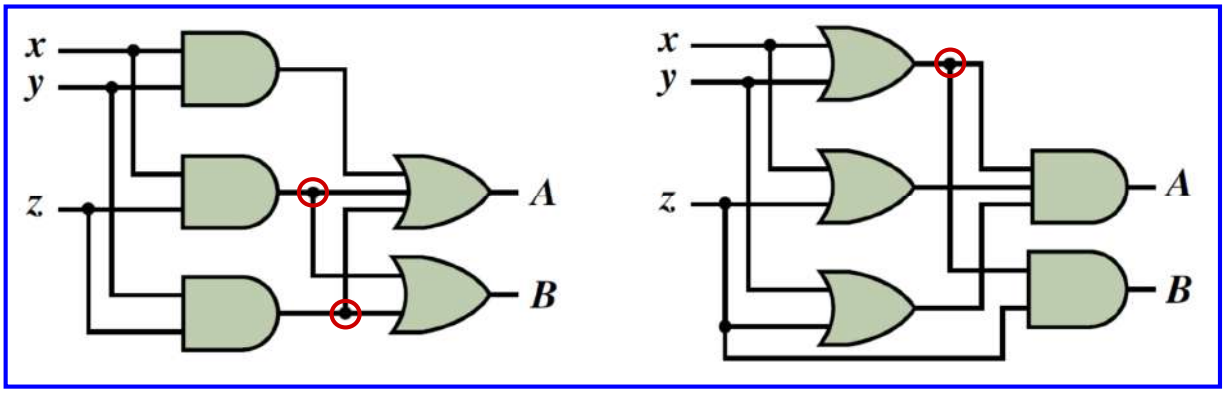
$$B(x,y,z) = xz + yz$$

$$A(x,y,z) = (x'y' + x'z' + y'z')' = (x + y)(x + z)(y + z)$$

$$B(x,y,z) = (x'y' + z')' = (x + y)z$$

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Υλοποίηση κυκλώματος με δύο επίπεδα πυλών διάταξης AND-OR και OR-AND



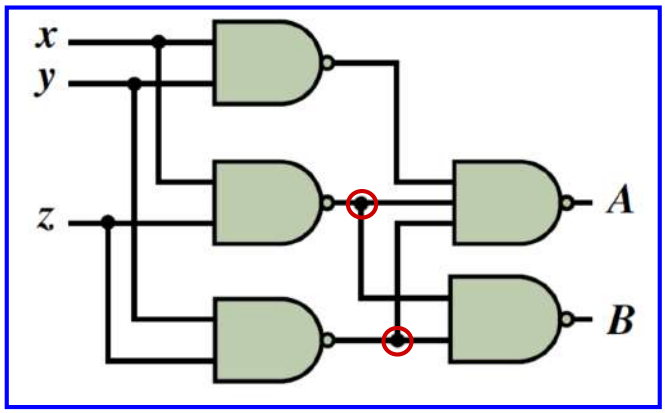
Κατά τη σύνθεση ενός συνδυαστικού κυκλώματος με πολλαπλές εξόδους, ενδέχεται οι λογικές συναρτήσεις δύο ή περισσότερων εξόδων να περιλαμβάνουν ορισμένους κοινούς όρους (λογικά γινόμενα ή αθροίσματα). Στην περίπτωση αυτή, οι κοινοί όροι υλοποιούνται μία φορά και επαναχρησιμοποιούνται για την παραγωγή των εξόδων, στις αλγεβρικές εκφράσεις των οποίων συμμετέχουν.

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Εφαρμογή του θεωρήματος διπλής άρνησης και στη συνέχεια του θεωρήματος De Morgan στη μορφή αθροίσματος γινομένων:

$$A(x,y,z) = [(xy)'(xz)'(yz)']'$$
$$B(x,y,z) = [(xz)'(yz)']'$$

Υλοποίηση κυκλώματος με πύλες NAND:

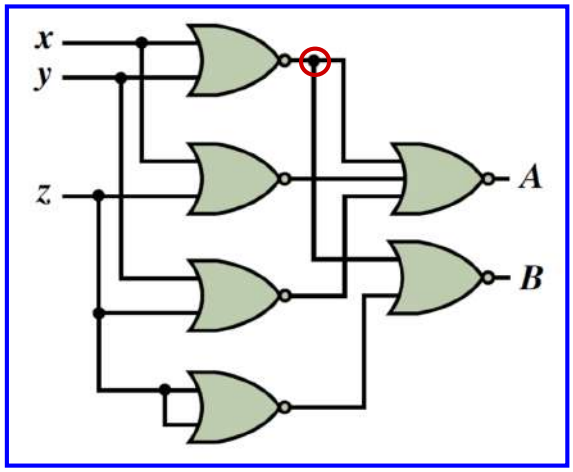


Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Εφαρμογή του θεωρήματος διπλής άρνησης και στη συνέχεια του θεωρήματος De Morgan στη μορφή γινομένου αθροισμάτων:

$$A(x,y,z) = [(x+y)' + (x+z)' + (y+z)']'$$
$$B(x,y,z) = [(x+y)' + z']'$$

Υλοποίηση κυκλώματος με πύλες NOR:

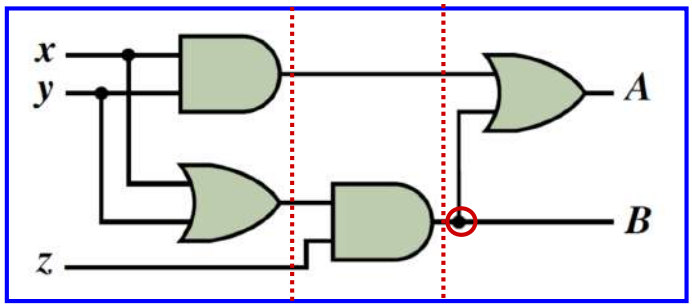


Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

- Εκτός από τις υλοποιήσεις δύο επιπέδων, με κατάλληλο χειρισμό των αλγεβρικών εκφράσεων των εξόδων ενός κυκλώματος μπορούν να προκύψουν υλοποιήσεις περισσότερων επιπέδων με μικρότερο κόστος.
- Εάν στην ελαχιστοποιημένη μορφή αθροίσματος γινομένων των συναρτήσεων εξόδου χρησιμοποιήσουμε τη μέθοδο της παραγοντοποίησης (αξίωμα επιμεριστικότητας), μπορούμε να εξάγουμε ισοδύναμες λογικές συναρτήσεις:

$$A(x,y,z) = (x + y)z + xy \quad \text{και} \quad B(x,y,z) = (x + y)z$$

οι οποίες υλοποιούνται από ένα κύκλωμα τριών επιπέδων πυλών που αποτελείται από μόνο 4 πύλες με δύο εισόδους η καθεμία. Ωστόσο, λόγω του επιπλέον επιπέδου πυλών, παρουσιάζει μεγαλύτερη καθυστέρηση απόκρισης από τα υπόλοιπα.



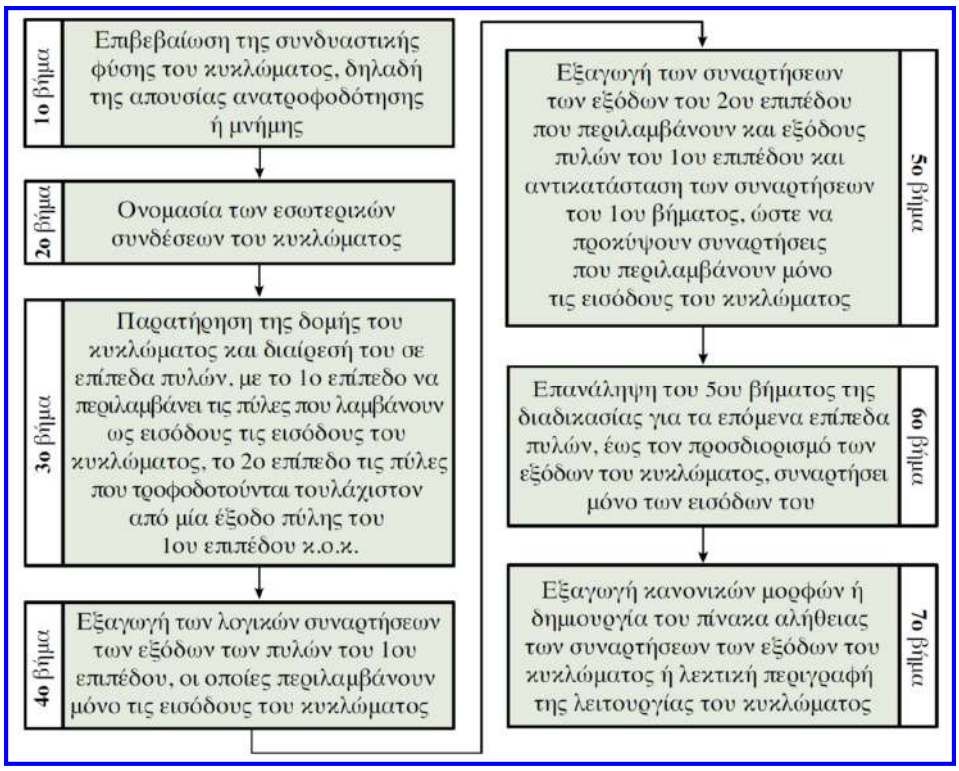
Ανάλυση συνδυαστικών κυκλωμάτων

- Όλα τα κυκλώματα που παρουσιάστηκαν στο παράδειγμα της πινακοθήκης είναι λειτουργικά ισοδύναμα, αφού έχουν το ίδιο πλήθος εισόδων και εξόδων και αντιστοιχούν στον ίδιο πίνακα αλήθειας ή στις ίδιες κανονικές αλγεβρικές μορφές.

- Προκειμένου, να διαπιστωθεί η λειτουργική ισοδυναμία δύο ή περισσότερων συνδυαστικών κυκλωμάτων, θα πρέπει αυτά να αναλυθούν, ώστε να διαπιστωθεί εάν αντιστοιχούν στον ίδιο πίνακα αλήθειας ή στις ίδιες κανονικές αλγεβρικές μορφές

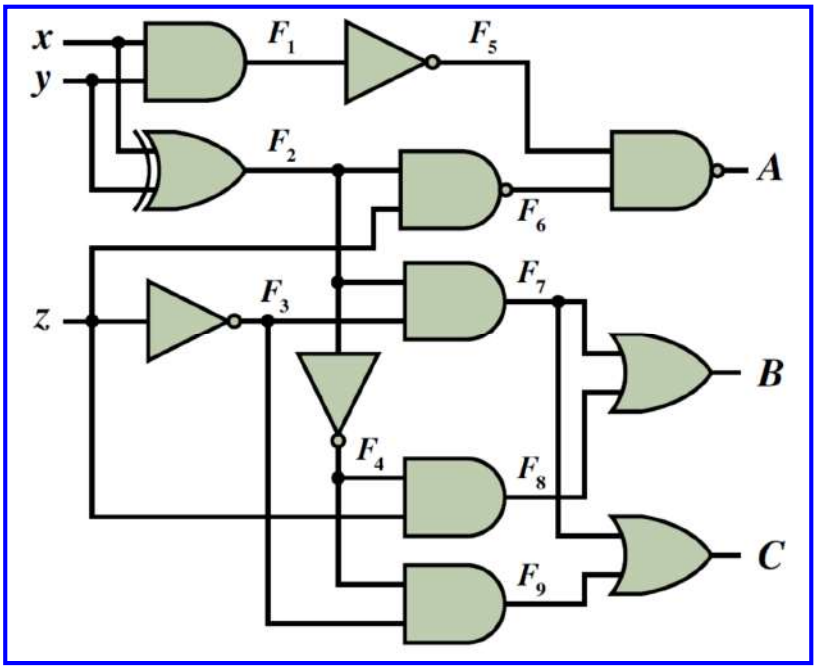
- Ανάλυση ενός συνδυαστικού κυκλώματος είναι λοιπόν ο προσδιορισμός της λειτουργίας ή των λειτουργιών που εκτελεί, ο οποίος συνίσταται στην εξαγωγή του συνόλου των λογικών συναρτήσεων ή του πίνακα αλήθειας ή ακόμη και μιας λεκτικής περιγραφής της λειτουργίας ή των λειτουργιών που εκτελούνται.

Ανάλυση συνδυαστικών κυκλωμάτων



Ανάλυση συνδυαστικών κυκλωμάτων

Ανάλυση συνδυαστικού κυκλώματος που λαμβάνει και παράγει τριψήφιους μη προσημασμένους αριθμούς



Ανάλυση συνδυαστικών κυκλωμάτων

Επίπεδα πυλών	Λογικές συναρτήσεις
1ο επίπεδο	$F_1 = xy, F_2 = x \oplus y, F_3 = z'$
2ο επίπεδο	$F_4 = F_2' = (x \oplus y)', F_5 = F_1' = (xy)',$ $F_6 = (zF_2)' = z' + F_2' = z' + (x \oplus y)', F_7 = F_2F_3 = (x \oplus y)z'$
3ο επίπεδο	$F_8 = zF_4 = z(x \oplus y)', F_9 = F_3F_4 = z'(x \oplus y)',$ $A = (F_5F_6)' = F_5' + F_6' = xy + [z' + (x \oplus y)']' = xy + z(x \oplus y)$
4ο επίπεδο	$B = F_7 + F_8 = (x \oplus y)z' + z(x \oplus y)' = x \oplus y \oplus z,$ $C = F_7 + F_9 = (x \oplus y)z' + z'(x \oplus y)'$

Κανονικές μορφές αθροίσματος ελαχίστων όρων των συναρτήσεων εξόδου:

$$A = xy + z(x \oplus y) = xy(z + z') + z(x'y + xy') = xyz + xyz' + x'yz + xy'z$$

$$B = x \oplus y \oplus z = (x'y + xy')z' + (xy + x'y')z = x'yz' + xy'z' + xyz + x'y'z$$

$$C = (x \oplus y)z' + z'(x \oplus y)' = (x'y + xy')z' + z'(xy + x'y') = x'yz' + xy'z' + xyz' + x'y'z'$$

Ανάλυση συνδυαστικών κυκλωμάτων

Από τις κανονικές μορφές αθροίσματος ελαχιστόρων των συναρτήσεων εξόδου του κυκλώματος, μπορούμε να δημιουργήσουμε άμεσα τον πίνακα αλήθειας που τις περιγράφει:

x	y	z	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

Η λειτουργία του κυκλώματος δεν είναι εύκολο να περιγραφεί λεκτικά από τις αλγεβρικές εκφράσεις των συναρτήσεων εξόδου του κυκλώματος. Ωστόσο, αφού πρόκειται για κύκλωμα που λαμβάνει και παράγει τριψήφιους μη προσημασμένους αριθμούς, από τον πίνακα αλήθειας διαπιστώνουμε ότι όταν ο δυαδικός αριθμός εισόδου ισούται με 0, 1, 2 ή 3 (σε δεκαδική μορφή), ο δυαδικός αριθμός εξόδου είναι κατά ένα μεγαλύτερος, ενώ όταν ο αριθμός εισόδου ισούται με 4, 5, 6 ή 7, ο αριθμός εξόδου είναι κατά ένα μικρότερος.

Ανάλυση συνδυαστικών κυκλωμάτων

Ο ίδιος πίνακας αλήθειας μπορεί να προκύψει χωρίς την εξαγωγή των λογικών συναρτήσεων των εξόδων. Η διαδικασία συνίσταται στη σταδιακή εξαγωγή των στηλών του πίνακα αλήθειας που αντιστοιχούν στις εσωτερικές συνδέσεις του κυκλώματος, ώστε από αυτές να προκύψουν οι στήλες του πίνακα που αντιστοιχούν στις εξόδους του κυκλώματος, χρησιμοποιώντας μόνο τους ορισμούς των βασικών λογικών πράξεων:

x	y	z	$F_1 = xy$	$F_2 = x \oplus y$	$F_3 = z'$	$F_4 = F'_2$	$F_5 = F'_1$	$F_6 = (zF_2)'$	$F_7 = F_2F_3$	$F_8 = zF_4$	$F_9 = F_3F_4$	$A = (F_5F_6)'$	$B = F_7 + F_8$	$C = F_7 + F_9$
0	0	0	0	0	1	1	1	1	0	0	1	0	0	1
0	0	1	0	0	0	1	1	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1	1	0	0	0	1	1
0	1	1	0	1	0	0	1	0	0	0	0	1	0	0
1	0	0	0	1	1	0	1	1	1	0	0	0	1	1
1	0	1	0	1	0	0	1	0	0	0	0	1	0	0
1	1	0	1	0	1	1	0	1	0	0	1	1	0	1
1	1	1	1	0	0	1	0	1	0	1	0	1	1	0

Σύνθετα συνδυαστικά κυκλώματα

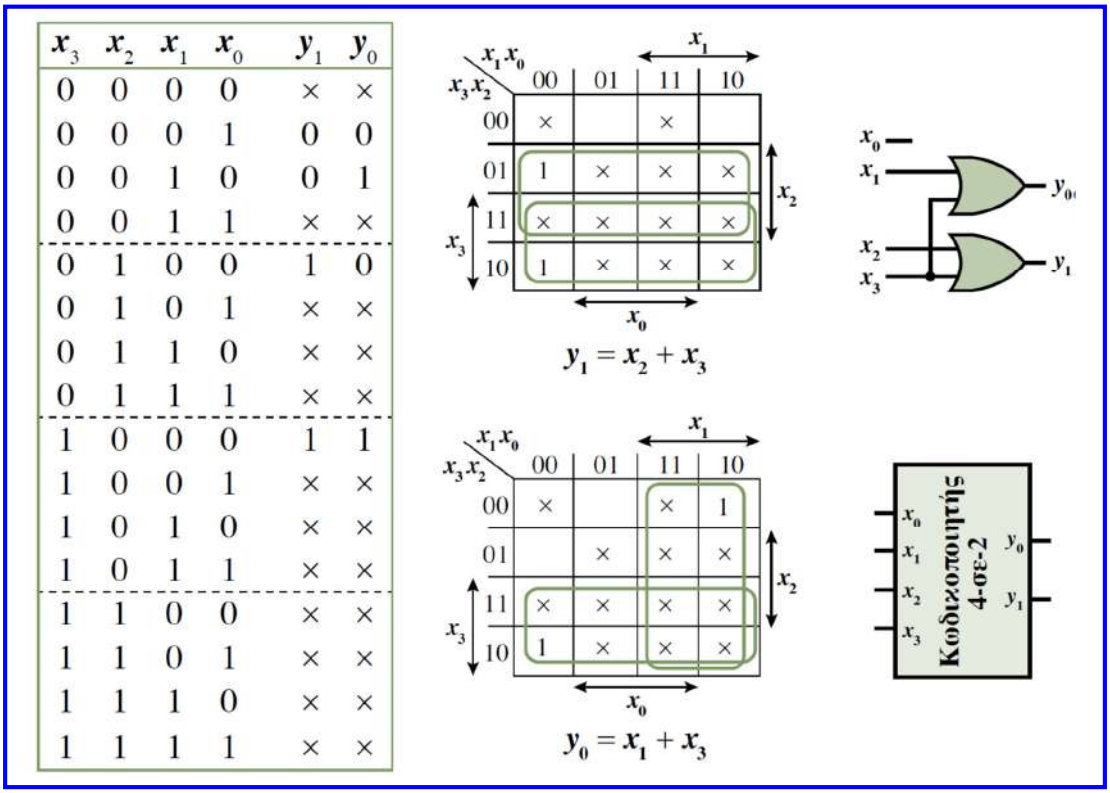
- Τα ψηφιακά συστήματα σχεδιάζονται και υλοποιούνται κυρίως για να μετασχηματίζουν δεδομένα.
- Κάθε μετασχηματισμός μπορεί να παρασταθεί από μία ή περισσότερες λογικές συναρτήσεις.
- Ορισμένες κατηγορίες μετασχηματισμών εμφανίζονται αρκετά συχνά σε πρακτικά προβλήματα, όπως π.χ. η κωδικοποίηση και η αποκωδικοποίηση δεδομένων, η επιλογή και μεταφορά δεδομένων, οι αριθμητικές πράξεις κ.ά.
- Είναι λοιπόν χρήσιμο να υπάρχουν τα κυκλώματα αυτά προσχεδιασμένα και τυποποιημένα, έτσι ώστε να είναι εύκολη η χρησιμοποίησή τους στα ψηφιακά συστήματα.

Κωδικοποιητές

- Τα ψηφιακά συστήματα έχουν τη δυνατότητα να χειρίζονται διακριτά στοιχεία πληροφορίας που ανήκουν σε ένα πεπερασμένο σύνολο, αφού το καθένα από αυτά μπορεί να παρασταθεί με μία ακολουθία δυαδικών ψηφίων, ακολουθώντας τους κανόνες ενός δυαδικού κώδικα.
- Κάθε διαφορετικό στοιχείο πληροφορίας αντιστοιχίζεται σε μία μοναδική ακολουθία ή συνδυασμό δυαδικών ψηφίων.
- Για την παράσταση ενός συνόλου 2^n στοιχείων πληροφορίας, όπως, για παράδειγμα, αριθμών ή γραμμάτων, απαιτείται δυαδικός κώδικας που να χρησιμοποιεί τουλάχιστον n δυαδικά ψηφία.
- Οι κωδικοποιητές (encoders) είναι συνδυαστικά κυκλώματα τα οποία παράγουν στην έξοδό τους ψηφιακά δεδομένα σε πιο συμπαγή μορφή από εκείνη των δεδομένων που λαμβάνουν στην είσοδό τους.
- Ένας δυαδικός κωδικοποιητής, λαμβάνει στην είσοδό του έως 2^n δυαδικά ψηφία και παράγει στην έξοδό του n δυαδικά ψηφία.
- Λόγω του ότι το πλήθος των ψηφίων εξόδου δεν επαρκεί για την παράσταση του μεγαλύτερου πλήθους των δεδομένων εισόδου, θεωρούμε ότι ένα μόνο από τα ψηφία εισόδου μπορεί να έχει λογική τιμή 1, έτσι ώστε το κύκλωμα να παράγει στην έξοδό του ένα δυαδικό αριθμό n ψηφίων, που αντιστοιχεί στο ψηφίο της εισόδου με λογική τιμή 1.

Απλός κωδικοποιητής 4-σε-2

x_3	x_2	x_1	x_0	y_1	y_0
0	0	0	0	×	×
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	×	×
0	1	0	0	1	0
0	1	0	1	×	×
0	1	1	0	×	×
0	1	1	1	×	×
1	0	0	0	1	1
1	0	0	1	×	×
1	0	1	0	×	×
1	0	1	1	×	×
1	1	0	0	×	×
1	1	0	1	×	×
1	1	1	0	×	×
1	1	1	1	×	×

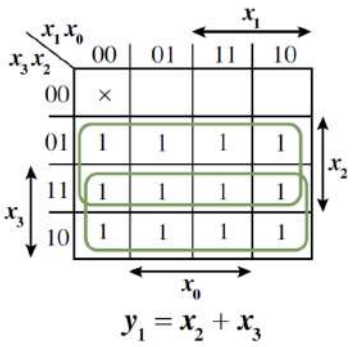


Κωδικοποιητής προτεραιότητας

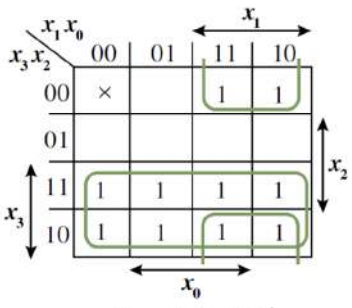
- Στη λειτουργία του απλού κωδικοποιητή εντοπίζονται δύο προβλήματα.
- Το πρώτο αφορά το γεγονός ότι οι έξοδοι του κυκλώματος μηδενίζονται, όταν όλες οι είσοδοι έχουν λογική τιμή 0, με αποτέλεσμα αυτός ο συνδυασμός τιμών των εισόδων να μην μπορεί να διακριθεί από την περίπτωση όπου μόνο η είσοδος x_0 έχει λογική τιμή 1.
- Το δεύτερο πρόβλημα οφείλεται στο ότι δεν προβλέπονται οι περιπτώσεις όπου περισσότερες από μία εισόδους έχουν τιμή 1.
- Το πρώτο πρόβλημα αντιμετωπίζεται με την προσθήκη μιας επιπλέον εξόδου εγκυρότητας (V) που λαμβάνει τιμή 0 όταν όλες οι είσοδοι είναι 0 και τιμή 1 σε όλες τις υπόλοιπες περιπτώσεις.
- Οι υπόλοιπες δύο έξοδοι δεν ορίζονται όταν μηδενίζεται η V, συνεπώς ο συνδυασμός μηδενικών εισόδων αποτελεί αδιάφορη λογική συνθήκη για τις εξόδους αυτές.
- Το δεύτερο πρόβλημα λύνεται εάν ο κωδικοποιητής τροποποιηθεί ώστε να υποστηρίζει προκαθορισμένη προτεραιότητα εισόδων και τότε αναφέρεται ως κωδικοποιητής προτεραιότητας (priority encoder).
- Έτσι όταν η τιμή περισσότερων από μία εισόδων είναι 1, η έξοδος του κυκλώματος καθορίζεται από την είσοδο με τη μεγαλύτερη προτεραιότητα.
- Για παράδειγμα, μπορεί να καθοριστεί ως είσοδος με τη μεγαλύτερη προτεραιότητα η είσοδος x_3 και να ακολουθούν σε σειρά προτεραιότητας οι είσοδοι x_2, x_1 και x_0.

Κωδικοποιητής προτεραιότητας

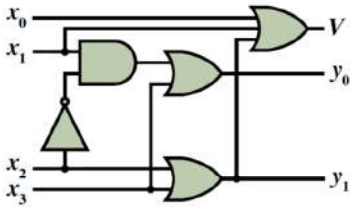
x_3	x_2	x_1	x_0	y_1	y_0	V
0	0	0	0	×	×	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	1	0	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1



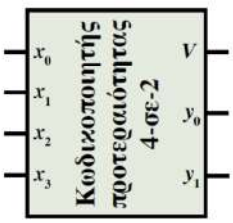
$$y_1 = x_2 + x_3$$



$$y_0 = x_3 + x_1x_2'$$



$$V' = x_3'x_2'x_1'x_0' \Rightarrow$$
$$V = x_3 + x_2 + x_1 + x_0$$



Κωδικοποιητής προτεραιότητας

- Αφού η τιμή της εξόδου του κωδικοποιητή προτεραιότητας καθορίζεται από την είσοδο με τη μεγαλύτερη προτεραιότητα, μπορούμε να καταστρώσουμε τον πίνακα αλήθειας σε μια συμπτυγμένη μορφή, που περιλαμβάνει αδιάφορες τιμές εισόδων και αποτελείται από 5 γραμμές αντί για 16 γραμμές.

x_3	x_2	x_1	x_0	y_1	y_0	V
0	0	0	0	×	×	**0**
0	0	0	**1**	0	0	1
0	0	**1**	×	0	**1**	1
0	**1**	×	×	**1**	0	1
1	×	×	×	**1**	**1**	1

- Οι ελαχιστοποιημένες συναρτήσεις των εξόδων του κωδικοποιητή προτεραιότητας μπορούν να εξαχθούν απευθείας από την συμπτυγμένη μορφή του πίνακα αλήθειας:

$$y_1 = x'_3x_2 + x_3 = (x_3 + x_2)(x_3 + x'_3) = x_3 + x_2$$
$$y_0 = x'_3x'_2x_1 + x_3 = (x_3 + x'_2x_1)(x_3 + x'_3) = x_3 + x_1x'_2$$
$$V' = x'_3x'_2x'_1x'_2 \Rightarrow V = x_3 + x_2 + x_1 + x_0$$

Αποκωδικοποιητές

- Οι αποκωδικοποιητές (decoders) είναι συνδυαστικά κυκλώματα τα οποία εκτελούν λειτουργία αντίστροφη από εκείνη των κωδικοποιητών.

- Λαμβάνουν στην είσοδό τους n δυαδικά ψηφία και παράγουν στην έξοδό τους έως 2^n δυαδικά ψηφία.

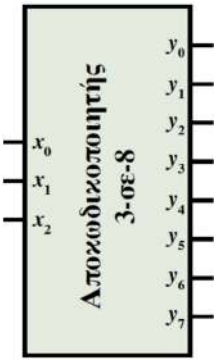
- Για κάθε συνδυασμό λογικών τιμών εισόδου, μόνο το ψηφίο εξόδου που αντιστοιχεί σε αυτόν λαμβάνει λογική τιμή 1.

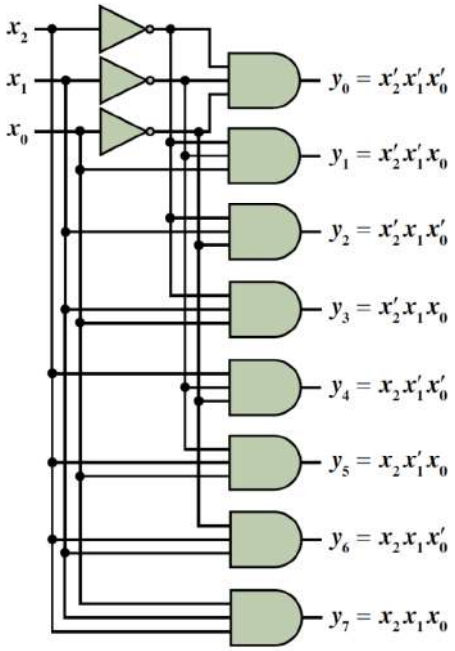
- Ουσιαστικά, ένας αποκωδικοποιητής n-σε-2^n αποτελεί γεννήτρια ελαχιστόρων, αφού κάθε έξοδος αντιστοιχεί σε έναν ελαχιστόρο n μεταβλητών εισόδου.

- Το λογικό κύκλωμα του αποκωδικοποιητή σχεδιάζεται εύκολα με βάση τον πίνακα αλήθειας, χρησιμοποιώντας αντιστροφείς για την παραγωγή των συμπληρωματικών μορφών των εισόδων και λογικές πύλες AND για την παραγωγή του συνόλου των ελαχιστόρων.

Αποκωδικοποιητής 3-σε-8

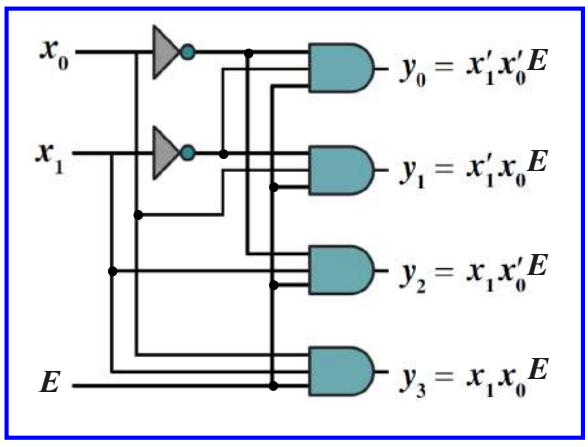
x_2	x_1	x_0	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

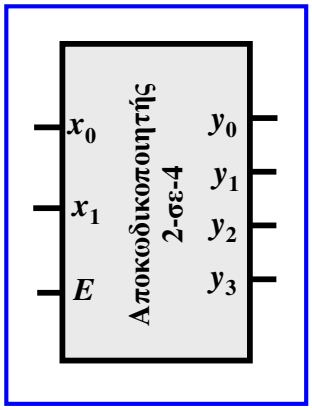




Αποκωδικοποιητής 2-σε-4 με είσοδο ενεργοποίησης

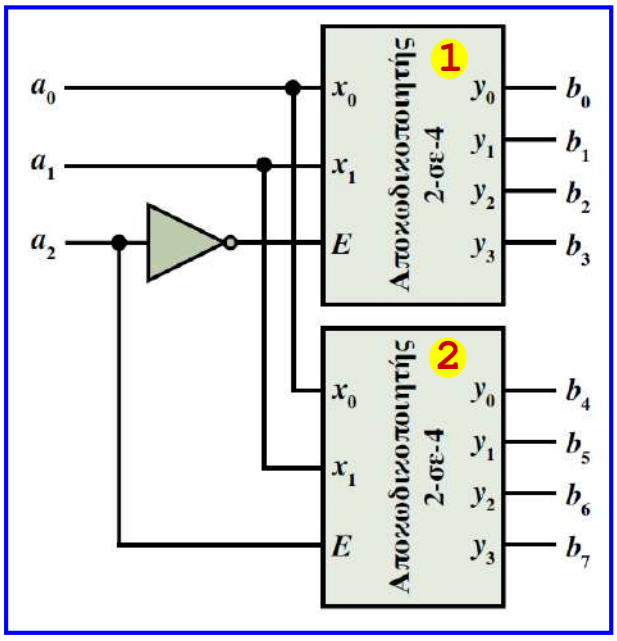
- Εκτός από τα ψηφία εισόδου που πρόκειται να αποκωδικοποιηθούν, το κύκλωμα ενός αποκωδικοποιητή μπορεί να περιλαμβάνει μία ακόμη είσοδο ενεργοποίησης (enable input), ανάλογα με την τιμή της οποίας να επιτρέπεται ή όχι η αποκωδικοποίηση.
- Η προσθήκη αυτή προϋποθέτει τη χρήση πυλών AND με μία επιπλέον είσοδο.
- Όταν E = 0 οι έξοδοι του κυκλώματος μηδενίζονται, ενώ όταν E = 1 οι είσοδοί του αποκωδικοποιούνται με βάση τον πίνακα αλήθειας που περιγράφει τη λειτουργία του.





Σύνθεση πολύπλοκων αποκωδικοποιητών

- Η είσοδος Ε παρέχει τη δυνατότητα συνδυασμού αποκωδικοποιητών με στόχο τη σύνθεση αποκωδικοποιητών με περισσότερες εισόδους.
- Στη σχεδίαση αποκωδικοποιητή 3-σε-8 με 2 αποκωδικοποιητές 2-σε-4:
  - ✓ όταν $a_2 = 0$, ενεργοποιείται ο πρώτος αποκωδικοποιητής 2-σε-4 και παράγει στις εξόδους b_0 έως b_3 τους ελαχιστόρους m_0 έως m_3, αντίστοιχα,
  - ✓ όταν $a_2 = 1$, ενεργοποιείται ο δεύτερος αποκωδικοποιητής 2-σε-4 και παράγει στις εξόδους b_4 έως b_7 τους ελαχιστόρους m_4 έως m_7, αντίστοιχα.

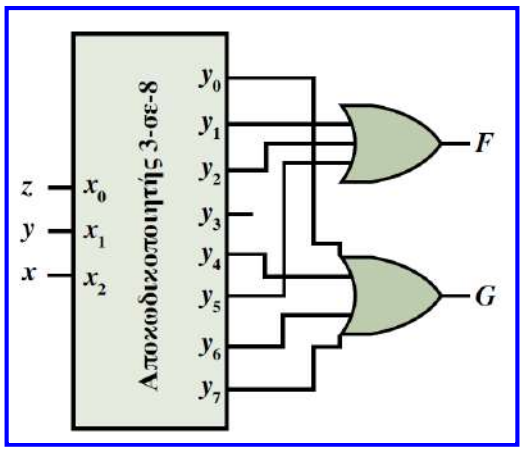


Υλοποίηση λογικών συναρτήσεων με αποκωδικοποιητή

- Αφού ένας αποκωδικοποιητής n-σε-2^n αποτελεί και γεννήτρια ελαχιστόρων, προκύπτει ότι συνδυάζοντάς τον με μία λογική πύλη OR, η οποία παράγει το λογικό άθροισμα κατάλληλων εξόδων του, μπορούμε να υλοποιήσουμε οποιαδήποτε λογική συνάρτηση μορφής αθροίσματος ελαχιστόρων.
- Για να γίνει αυτό θα πρέπει το πλήθος των εισόδων του αποκωδικοποιητή να ισούται με το πλήθος των μεταβλητών της συνάρτησης και το πλήθος των εισόδων της πύλης OR να ισούται με το πλήθος των ελαχιστόρων που συμμετέχουν στη συνάρτηση.

$$F(x,y,z) = \Sigma(1, 2, 5)$$

$$\begin{aligned} G(x,y,z) &= xy + y'z' \\ &= xy(z + z') + y'z'(x + x') \\ &= xyz + xyz' + xy'z' + x'y'z' \\ &= \Sigma(7, 6, 4, 0) \end{aligned}$$

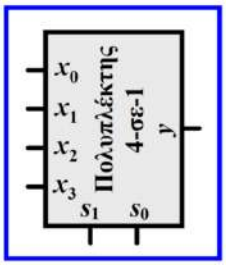


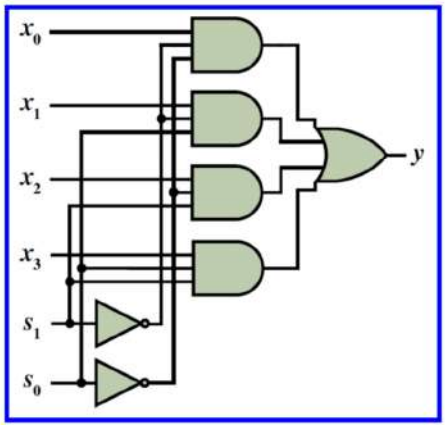
Πολυπλέκτες

- Οι πολυπλέκτες (multiplexers) είναι συνδυαστικά κυκλώματα τα οποία λαμβάνουν στην είσοδό τους 2^n δυαδικά ψηφία και μεταφέρουν στην έξοδό τους την τιμή ενός από τα ψηφία εισόδου.
- Για την επιλογή του ψηφίου εισόδου που θα μεταφερθεί στην έξοδο, οι πολυπλέκτες περιλαμβάνουν n πρόσθετες εισόδους, οι οποίες αναφέρονται ως είσοδοι επιλογής (select inputs), έτσι ώστε να διακρίνονται από τις 2^n εισόδους δεδομένων (data inputs).
- Πολυπλέκτης 4-σε-1: 4 (2^2) είσοδοι δεδομένων, 2 είσοδοι επιλογής και 1 έξοδος.
- Για καθέναν από τους 4 συνδυασμούς τιμών των εισόδων επιλογής, μεταφέρεται στην έξοδο του πολυπλέκτη 1 από τις 4 εισόδους δεδομένων.

$$y = s_1' s_0' x_0 + s_1' s_0 x_1 + s_1 s_0' x_2 + s_1 s_0 x_3$$

s_1	s_0	y
0	0	x_0
0	1	x_1
1	0	x_2
1	1	x_3





Υλοποίηση λογικών συναρτήσεων με πολυπλέκτες

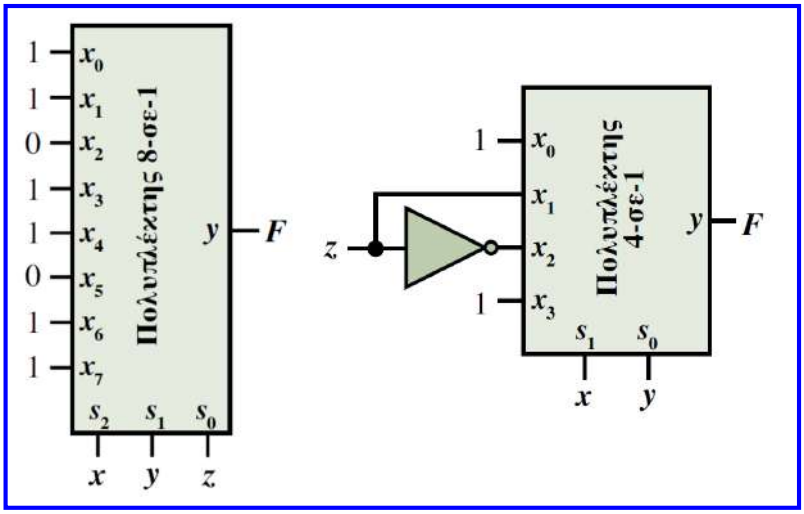
- Με όμοιο τρόπο, μπορούμε να συνθέσουμε πολυπλέκτες με μικρότερο (2-σε-1) ή μεγαλύτερο (8-σε-1, 16-σε-1 κ.ο.κ.) αριθμό εισόδων δεδομένων, τηρώντας πάντα την αναλογία 2^n / n για τις εισόδους δεδομένων και επιλογής, αντίστοιχα.
- Συχνά στους πολυπλέκτες γίνεται προσθήκη μιας εισόδου ενεργοποίησης (E) και όταν αυτή λαμβάνει τιμή 0 η έξοδος του πολυπλέκτη είναι 0, ενώ όταν λαμβάνει τιμή 1 ο πολυπλέκτης λειτουργεί κανονικά.
- Αυτό προϋποθέτει τη χρήση πυλών AND με μία επιπλέον είσοδο.
- Μπορούμε να παράγουμε στην έξοδο ενός πολυπλέκτη οποιοδήποτε άθροισμα ελαχιστόρων των μεταβλητών που αντιστοιχούν στις εισόδους επιλογής, θέτοντας λογική τιμή 1 στις εισόδους δεδομένων που αντιστοιχούν στους ελαχιστόρους που συμμετέχουν στο επιθυμητό άθροισμα και λογική τιμή 0 στους ελαχιστόρους που δε συμμετέχουν.
- Με τον τρόπο αυτόν, χρησιμοποιώντας έναν πολυπλέκτη 2^n-σε-1 μπορούμε να υλοποιήσουμε οποιαδήποτε λογική συνάρτηση n μεταβλητών, εάν τροφοδοτήσουμε με τις μεταβλητές αυτές τις εισόδους επιλογής του πολυπλέκτη και θέσουμε στις εισόδους δεδομένων τις κατάλληλες λογικές τιμές.
- Προκύπτει, ότι είναι πιο αποδοτικό να υλοποιήσουμε οποιαδήποτε λογική συνάρτηση n μεταβλητών, με έναν μικρότερο πολυπλέκτη 2^{n-1}-σε-1, τροφοδοτώντας τις (n – 1) εισόδους επιλογής του με (n – 1) μεταβλητές της συνάρτησης και χρησιμοποιώντας τη μεταβλητή της συνάρτησης που απομένει για την τροφοδότηση εισόδου ή εισόδων δεδομένων.

Υλοποίηση λογικών συναρτήσεων με πολυπλέκτες

- Για να επιτευχθεί η υλοποίηση λογικών συναρτήσεων, ενδέχεται να απαιτηθεί η χρήση ενός αντιστροφέα, έτσι ώστε να λαμβάνεται η συμπληρωματική μορφή της μεταβλητής που συμμετέχει στην τροφοδότηση των εισόδων δεδομένων του πολυπλέκτη.

$$F(x,y,z) = x'y' + xz' + yz$$

x	y	z	F	
0	0	0	1	$F = 1$
0	0	1	1	
0	1	0	0	$F = z$
0	1	1	1	
1	0	0	1	$F = z'$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	



Θεώρημα ανάπτυξης συναρτήσεων (Shannon)

- Κάθε λογική συνάρτηση μπορεί να αναπτυχθεί ως προς μία από τις μεταβλητές που συμμετέχουν σε αυτήν, ως εξής:

$$F(x, y\ldots, w) = xF(1, y,\ldots, w) + x'F(0, y,\ldots, w)$$

- Οι συναρτήσεις F(1, y,..., w) και F(0, y,..., w) προκύπτουν από τη συνάρτηση F(x, y..., w) για x = 1 και x = 0, αντίστοιχα.
- Οι συναρτήσεις αυτές, ως συναρτήσεις πλέον των μεταβλητών y, ..., w, μπορούν με όμοιο τρόπο να αναπτυχθούν ως προς μία από τις μεταβλητές αυτές, κ.ο.κ.

Ανάπτυξη συνάρτησης με 3 μεταβλητές:

$$F(x,y,z) = xF(1,y,z) + x'F(0,y,z)$$
$$= x[yF(1,1,z) + y'F(1,0,z)] + x'[yF(0,1,z) + y'F(0,0,z)]$$

$$\begin{aligned} F(x,y,z) &= \Sigma(0,1,3,4,6,7) \\ &= x'y'z + x'y'z' + xyz' + xy'z' + xyz + x'yz \\ &= x(yz' + y'z' + yz) + x'(y'z + y'z' + yz) \\ &= x[y(z' + z) + y'z'] + x'[yz + y'(z + z')] \\ &= x(y\mathbf{1} + y'\mathbf{z'}) + x'(\mathbf{y}z + y'\mathbf{1}) \end{aligned}$$

$$F(1,1,z) = 1$$
$$F(1,0,z) = z'$$
$$F(0,1,z) = z$$
$$F(0,0,z) = 1$$

Υλοποίηση συναρτήσεων με πολυπλέκτες 2-σε-1

- Από την ανάπτυξη μιας λογικής συνάρτησης σύμφωνα με το θεώρημα Shannon, προκύπτει μια μορφή της συνάρτησης που είναι άμεσα υλοποιήσιμη με πολυπλέκτες 2-σε-1:

$$F(x, y, z) = x[yF(1, 1, z) + y'F(1, 0, z)] + x'[yF(0, 1, z) + y'F(0, 0, z)]$$

- Οι συναρτήσεις στις αγκύλες υλοποιούνται με έναν πολυπλέκτη 2-σε-1 η καθεμία.
- Η είσοδος επιλογής των δύο πολυπλεκτών τροφοδοτείται με τη μεταβλητή y.
- Οι είσοδοι δεδομένων τους τροφοδοτούνται με τις συναρτήσεις $F(1, 1, z)$, $F(1, 0, z)$, $F(0, 1, z)$, $F(0, 0, z)$, οι οποίες ισούνται με 0 ή 1 ή z ή z'.
- Οι έξοδοι των δύο πολυπλεκτών τροφοδοτούν τις εισόδους δεδομένων ενός τρίτου πολυπλέκτη 2-σε-1 με είσοδο επιλογής τη μεταβλητή x.

Υλοποίηση συναρτήσεων με πολυπλέκτες 2-σε-1

$$F(x,y,z) = \Sigma(0,1,3,4,6,7)$$

$$F(x, y, z) = x[yF(1, 1, z) + y'F(1, 0, z)] + x'[yF(0, 1, z) + y'F(0, 0, z)]$$

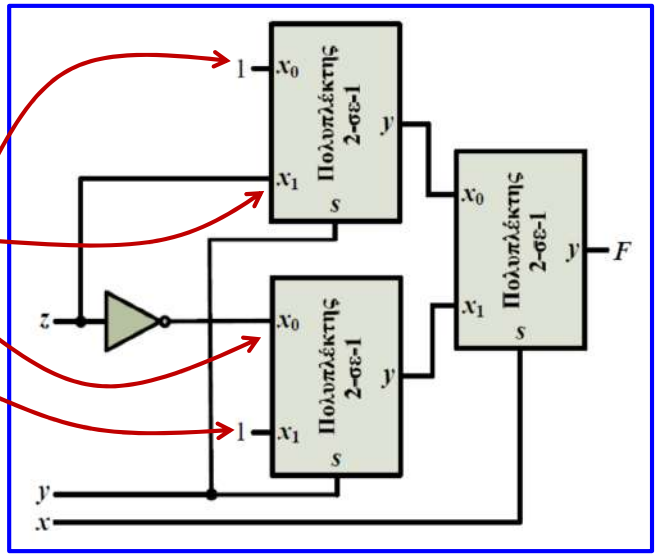
x	y	z	F	
0	0	0	1	$F = 1$
0	0	1	1	
0	1	0	0	$F = z$
0	1	1	1	
1	0	0	1	$F = z'$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	

$F(0, 0, z) = 1$

$F(0, 1, z) = z$

$F(1, 0, z) = z'$

$F(1, 1, z) = 1$



$$F(x,y,z) = x(y\mathbf{1} + y'\mathbf{z'}) + x'(y\mathbf{z} + y'\mathbf{1})$$

Υλοποίηση συναρτήσεων με πολυπλέκτες 2 σε 1

- Με την προαναφερόμενη μεθοδολογία, μπορεί να υλοποιηθεί οποιαδήποτε λογική συνάρτηση n μεταβλητών, χρησιμοποιώντας n – 1 επίπεδα πολυπλεκτών 2-σε-1.

- Στο πρώτο επίπεδο συμμετέχουν έως 2^{n-2} πολυπλέκτες 2-σε-1, με το πλήθος τους να υποδιπλασιάζεται σε κάθε επόμενο επίπεδο, έως το τελευταίο επίπεδο, το οποίο περιλαμβάνει έναν πολυπλέκτη 2-σε-1.

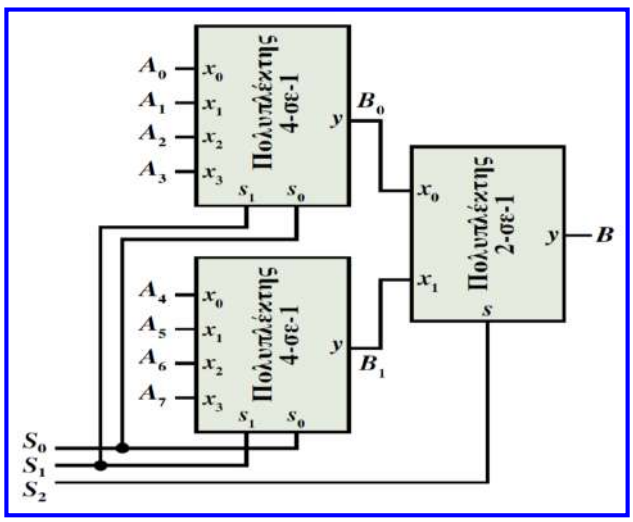
- Στην περίπτωση όπου από το ανάπτυγμα μιας λογικής συνάρτησης προκύπτει ότι οι είσοδοι δεδομένων ενός πολυπλέκτη 2-σε-1 τροφοδοτούνται με την ίδια λογική τιμή, μεταβλητή ή συμπληρωματική μορφή μεταβλητής, ο αντίστοιχος πολυπλέκτης του πρώτου επιπέδου απαλείφεται, οδηγώντας σε απλούστερο κύκλωμα υλοποίησης.

Σύνθεση πολύπλοκων πολυπλεκτών

Είναι δυνατό να συνθέσουμε πολυπλέκτες πολλών εισόδων με μικρότερους πολυπλέκτες.

Έξοδος πολυπλέκτη 8-σε-1

$$B = S'_2 S'_1 S'_0 A_0 + S'_2 S'_1 S_0 A_1 + S'_2 S_1 S'_0 A_2 + S'_2 S_1 S_0 A_3 + S_2 S'_1 S'_0 A_4 + S_2 S'_1 S_0 A_5 + S_2 S_1 S'_0 A_6 + S_2 S_1 S_0 A_7$$
$$= S'_2 (S'_1 S'_0 A_0 + S'_1 S_0 A_1 + S_1 S'_0 A_2 + S_1 S_0 A_3) + S_2 (S'_1 S'_0 A_4 + S'_1 S_0 A_5 + S_1 S'_0 A_6 + S_1 S_0 A_7)$$

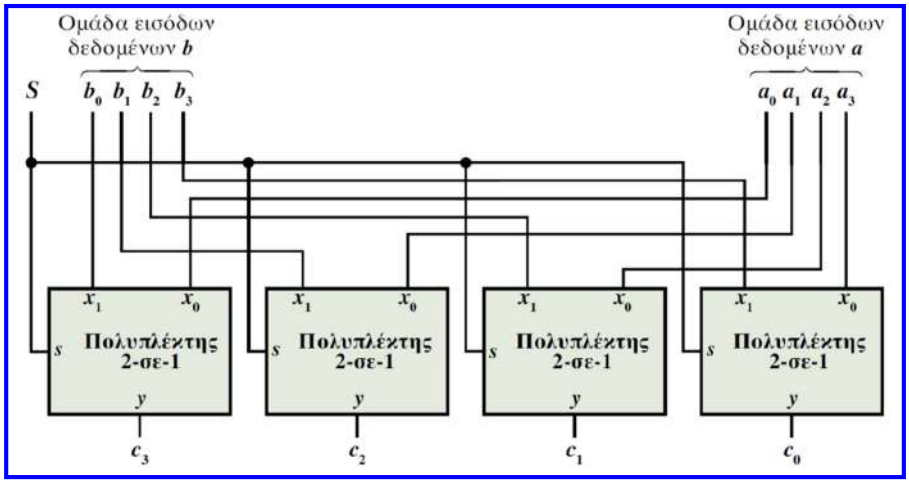


Πολυπλέκτες επιλογής πολλαπλών εισόδων

- Σε αρκετές εφαρμογές, απαιτούνται κυκλώματα για την επιλογή ομάδας δεδομένων.
- Στα κυκλώματα αυτά, που αναφέρονται ως πολυπλέκτες επιλογής πολλαπλών εισόδων (multiple-input selection multiplexers) ή απλούστερα ως πολλαπλοί πολυπλέκτες, συνδυάζονται πολυπλέκτες που χρησιμοποιούν κοινές εισόδους επιλογής.
- Για τη σύνθεσή τους απαιτείται ο συνδυασμός τόσων πολυπλεκτών όσες είναι οι είσοδοι δεδομένων κάθε ομάδας.
- Το πλήθος των εισόδων δεδομένων κάθε πολυπλέκτη ταυτίζεται με το πλήθος των ομάδων δεδομένων.

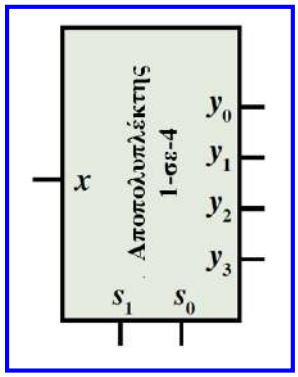
Παράδειγμα πολλαπλού πολυπλέκτη:

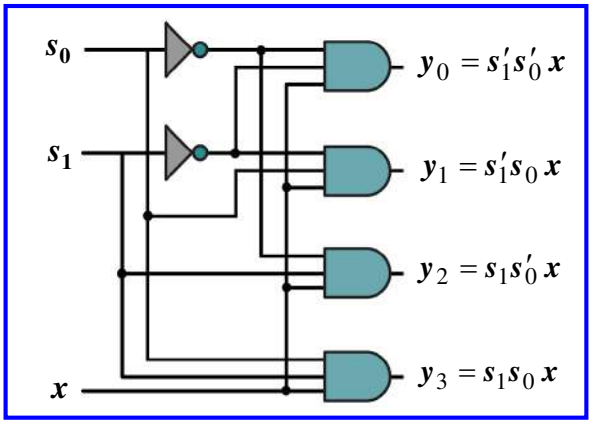
Τετραπλός πολυπλέκτης 2-σε-1



Αποπολυπλέκτες

- Οι αποπολυπλέκτες (demultiplexers) επιτελούν την αντίστροφη λειτουργία από εκείνη των πολυπλεκτών.
- Διαθέτουν μία είσοδο δεδομένων και 2^n εξόδους δεδομένων, σε μία από τις οποίες μεταφέρεται η είσοδος δεδομένων, ανάλογα με το συνδυασμό τιμών των n εισόδων επιλογής.
- Ο αποκωδικοποιητής με είσοδο ενεργοποίησης που παρουσιάστηκε, γίνεται αποπολυπλέκτης, εάν οι είσοδοι x_1, x_0 είναι είσοδοι επιλογής και η είσοδος Ε είναι είσοδος δεδομένων.





Αριθμητικά συνδυαστικά κυκλώματα

- Στα ψηφιακά συστήματα χρησιμοποιούνται ευρέως συνδυαστικά κυκλώματα που επιτελούν βασικές αριθμητικές λειτουργίες, όπως πρόσθεση, αφαίρεση, πολλαπλασιασμό, σύγκριση δυαδικών αριθμών κ.ά.

- Η πρόσθεση δύο δυαδικών ψηφίων υλοποιείται από ένα συνδυαστικό κύκλωμα δύο εισόδων, δηλαδή των ψηφίων που προστίθενται, και δύο εξόδων, δηλαδή του αθροίσματος και του κρατουμένου.

- Ωστόσο, κατά την πρόσθεση δύο αριθμών με περισσότερα ψηφία, για την υλοποίηση της πρόσθεσης των ψηφίων μιας θέσης απαιτείται συνδυαστικό κύκλωμα με τρεις εισόδους, έτσι ώστε να συμμετέχει στην πράξη και το κρατούμενο της πρόσθεσης των ψηφίων της προηγούμενης θέσης.

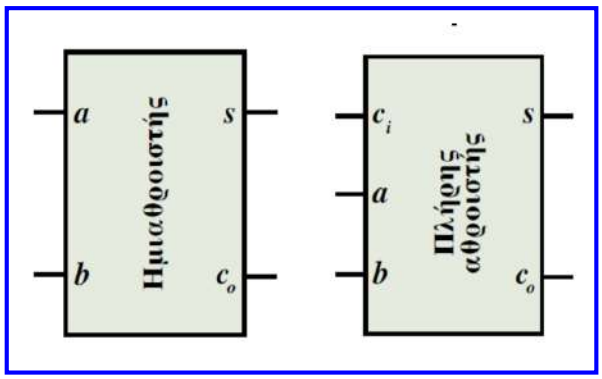
- Το πρώτο κύκλωμα αναφέρεται ως ημιαθροιστής (half adder, HA), ενώ το δεύτερο ως πλήρης αθροιστής (full adder, FA).

Ημιαθροιστής και πλήρης αθροιστής

a	b	s	c_o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$s = a \oplus b$$
$$c_o = ab$$

a	b	c_i	s	c_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

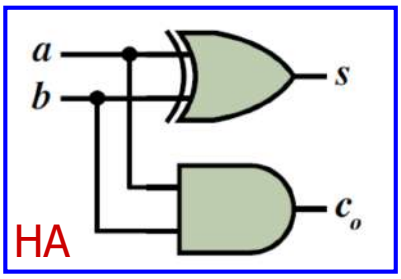


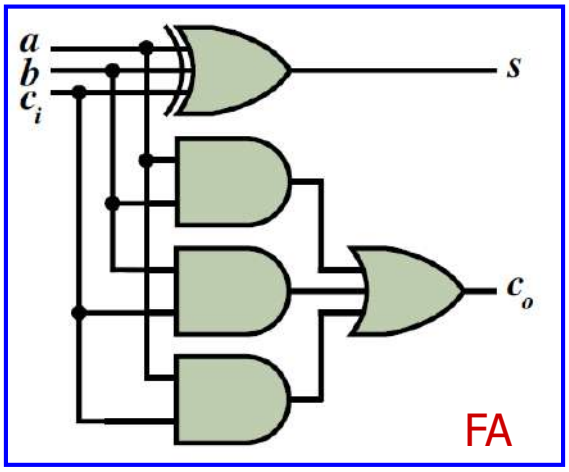
$$s = a'b'c_i + a'bc_i' + ab'c_i' + abc_i = (ab' + a'b)c_i' + (ab + a'b')c_i = a \oplus b \oplus c_i$$

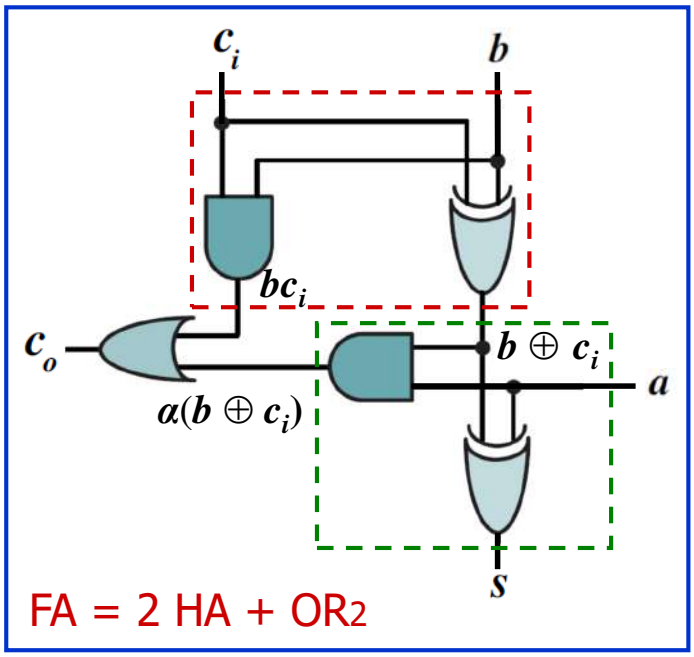
$$c_o = a'bc_i + ab'c_i + abc_i' + abc_i = a'bc_i + ab'c_i + abc_i' + abc_i + abc_i + abc_i$$
$$= ab(c_i + c_i') + bc_i(a + a') + ac_i(b + b') = ab + bc_i + ac_i$$

$$c_o = a'bc_i + ab'c_i + abc_i' + abc_i = a(bc_i' + b'c_i) + (a + a')bc_i = a(b \oplus c_i) + bc_i$$

Ημιαθροιστής και πλήρης αθροιστής





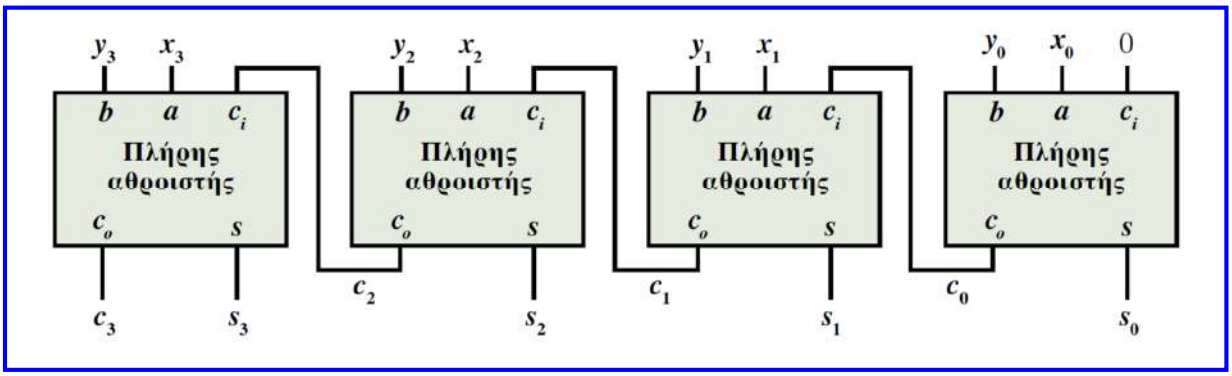


Παράλληλος αθροιστής

- Χρησιμοποιώντας ως δομικό στοιχείο τον πλήρη αθροιστή, μπορούμε να συνθέσουμε συνδυαστικά κυκλώματα παράλληλων αθροιστών για αριθμούς με περισσότερα ψηφία.
- Το αποτέλεσμα της πρόσθεσης δύο μη προσημασμένων δυαδικών αριθμών n ψηφίων αποτελείται από n + 1 ψηφία, συμπεριλαμβανομένου του τελικού κρατουμένου, συνεπώς το συνδυαστικό κύκλωμα του αθροιστή θα πρέπει να διαθέτει n + 1 εξόδους.
- Η πρόσθεση 2 δυαδικών αριθμών εκτελείται ανά ζεύγη ψηφίων της ίδιας θέσης, ξεκινώντας από το ζεύγος των λιγότερο σημαντικών ψηφίων των αριθμών και αν προκύπτει κρατούμενο ψηφίο μετά την πρόσθεση σε κάποια θέση, τότε αυτό προστίθεται στα ψηφία τής αμέσως πιο σημαντικής θέσης.
- Έτσι, ο παράλληλος αθροιστής δύο δυαδικών αριθμών n ψηφίων προκύπτει εύκολα, εάν συνδέσουμε n πλήρεις αθροιστές.
- Κάθε πλήρης αθροιστής δέχεται ένα ζεύγος ψηφίων των αριθμών εισόδου και παράγει ένα ψηφίο αθροίσματος και ένα ψηφίο κρατουμένου, το οποίο διαδίδεται στον πλήρη αθροιστή της επόμενης (πιο σημαντικής) θέσης.
- Το τελικό κρατούμενο της πρόσθεσης των δύο αριθμών λαμβάνεται στην έξοδο κρατουμένου του πλήρους αθροιστή της πιο σημαντικής θέσης.
- Οι παράλληλοι αθροιστές αυτής της δομής αναφέρονται ως αθροιστές διάδοσης κρατουμένου ή κυματικοί αθροιστές (ripple carry adders).

Παράλληλος αθροιστής

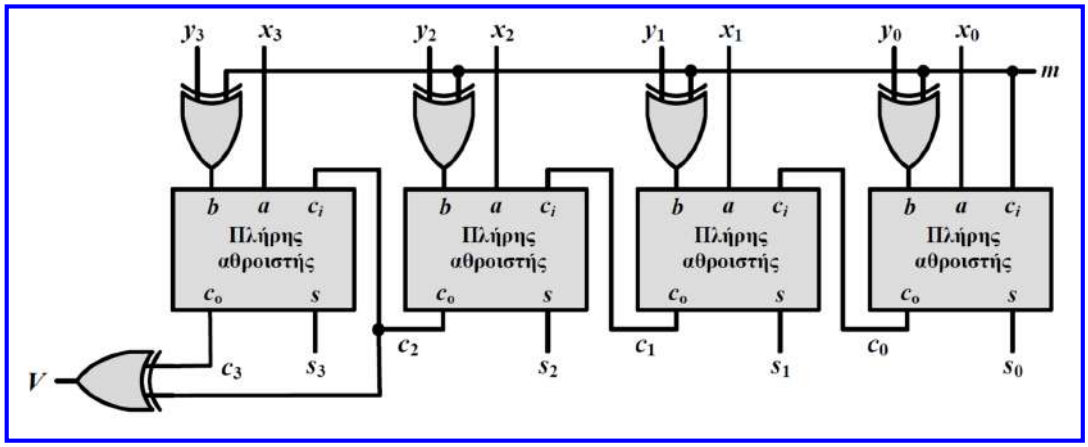
Παράλληλη πρόσθεση δύο τετραψήφιων δυαδικών αριθμών



Παράλληλος αθροιστής / αφαιρέτης

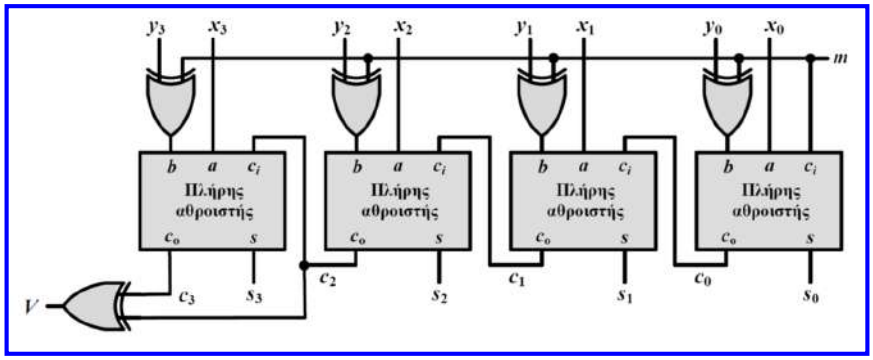
- Η αφαίρεση δυαδικών αριθμών ανάγεται σε πρόσθεση του μειωτέου με τον αντίθετο αριθμό του αφαιρετέου, ο οποίος μπορεί να παρασταθεί με το συμπλήρωμα ως προς 2 του αφαιρετέου.
- Το συμπλήρωμα ως προς 2 ενός δυαδικού αριθμού προκύπτει με πρόσθεση μιας μονάδας στο συμπλήρωμά του ως προς 1 και το συμπλήρωμα ως προς 1 υπολογίζεται με εναλλαγή των μονάδων και των μηδενικών του αριθμού.
- Επομένως, για να αφαιρέσουμε τον αριθμό Y από τον αριθμό X, αρκεί να εκτελέσουμε την πρόσθεση X + Y′ + 1, όπου Y′ είναι το συμπλήρωμα ως προς 1 του αριθμού Y, το οποίο προκύπτει εάν τροφοδοτήσουμε τα ψηφία του Y σε ισάριθμους αντιστροφείς.
- Η πύλη XOR δύο εισόδων παράγει στην έξοδό της τιμή 0, όταν οι είσοδοι λαμβάνουν την ίδια τιμή, και τιμή 1, όταν οι είσοδοι λαμβάνουν διαφορετικές τιμές.
- Επομένως, όταν στη μία είσοδο μιας πύλης XOR δύο εισόδων τεθεί λογική τιμή 1, τότε η έξοδός της ισούται με το συμπλήρωμα της άλλης εισόδου.
- Επομένως, οι αντιστροφείς που απαιτούνται για τον υπολογισμό του Y′ μπορούν να υποκατασταθούν από ισάριθμες πύλες XOR δύο εισόδων με λογική τιμή 1 στη μία είσοδό τους και τα ψηφία του αριθμού Y στην άλλη.
- Τα παραπάνω χρησιμοποιούνται για τη σύνθεση ενός παράλληλου αθροιστή / αφαιρέτη.

Παράλληλος αθροιστής / αφαιρέτης



- Όταν m = 1, οι πλήρεις αθροιστές τροφοδοτούνται με το συμπλήρωμα των ψηφίων του αριθμού Υ (δηλαδή με τον Υ′) και η είσοδος κρατουμένου του πλήρους αθροιστή της λιγότερο σημαντικής θέσης (c_o) τροφοδοτείται με τιμή 1, γεγονός που εξασφαλίζει την πρόσθεση της μονάδας που απαιτείται για τον υπολογισμό του συμπληρώματος ως προς 2 του Υ και κατά συνέπεια για την εκτέλεση της αφαίρεσης Χ – Υ.
- Όταν m = 0, οι πλήρεις αθροιστές τροφοδοτούνται με τα ψηφία του αριθμού Υ και η λειτουργία του κυκλώματος είναι όμοια με εκείνη του παράλληλου αθροιστή.

Παράλληλος αθροιστής / αφαιρέτης



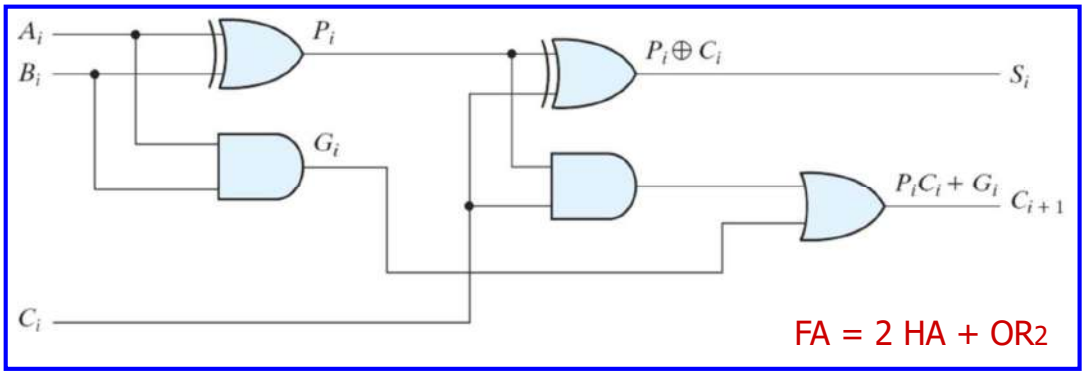
a	b	c_i	s	c_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- Η έξοδος V υποδεικνύει την υπερχείλιση κατά την πρόσθεση ή αφαίρεση, δηλ. τη μετατόπιση του ψηφίου-προσήμου του αποτελέσματος από την αναμενόμενη (πιο σημαντική) θέση.
- Υπερχείλιση κατά την πρόσθεση δύο προσημασμένων αριθμών συμβαίνει όταν οι αριθμοί είναι ομόσημοι και το αποτέλεσμα που προκύπτει έχει διαφορετικό πρόσημο από αυτούς.
- Αφού η αφαίρεση ανάγεται σε πρόσθεση του μειωτέου με τον αντίθετο του αφαιρετέου, υπερχείλιση κατά την αφαίρεση δύο προσημασμένων αριθμών (Χ – Υ) συμβαίνει όταν ο Χ είναι θετικός, ο Υ είναι αρνητικός και το αποτέλεσμα αρνητικό, καθώς και όταν ο Χ είναι αρνητικός, ο Υ είναι θετικός και το αποτέλεσμα θετικό.
- Η υπερχείλιση ανιχνεύεται μέσω της διαπίστωσης ότι τα κρατούμενα ψηφία που προκύπτουν κατά την πράξη στις δύο πιο σημαντικές θέσεις λαμβάνουν διαφορετική τιμή (V = 1).

Τεχνική πρόβλεψης κρατουμένου

- Οι αθροιστές διάδοσης κρατουμένου μειονεκτούν όσον αφορά την ταχύτητα υπολογισμού του αποτελέσματος, αφού απαιτούν αρκετό χρόνο για τη διάδοση του κρατουμένου έως την έξοδο κρατουμένου του πλήρους αθροιστή της πιο σημαντικής θέσης.
- Το κύκλωμα ενός πλήρους αθροιστή στο οποίο διαδίδεται το κρατούμενο εισάγει καθυστέρηση 2 λογικών πυλών.
- Έτσι, η καθυστέρηση διάδοσης κρατουμένου ενός παράλληλου αθροιστή n ψηφίων ισούται με την καθυστέρηση 2 × n λογικών πυλών.
- Με στόχο την επιτάχυνση των αθροιστών, έχουν αναπτυχθεί διάφορες τεχνικές σύνθεσής τους και μία από αυτές είναι η πρόβλεψη κρατουμένου (carry look-ahead).
- Η τεχνική αυτή στηρίζεται στο γεγονός ότι είναι δυνατό να υπολογίσουμε ένα κρατούμενο χωρίς να αναμένουμε την άφιξη του κρατουμένου της αμέσως λιγότερο σημαντικής θέσης.
- Η επιτάχυνση της πρόσθεσης συνοδεύεται ωστόσο από αύξηση της πολυπλοκότητας και συνεπώς του κόστους του κυκλώματος.

Τεχνική πρόβλεψης κρατουμένου



P_i: συνάρτηση διάδοσης κρατουμένου G_i: συνάρτηση γέννησης κρατουμένου	$P_i = A_i \oplus B_i$ $G_i = A_iB_i$	$S_i = P_i \oplus C_i$ $C_{i+1} = G_i + P_iC_i$

C_0 = κρατούμενο εισόδου

$$C_1 = G_0 + P_0C_0$$

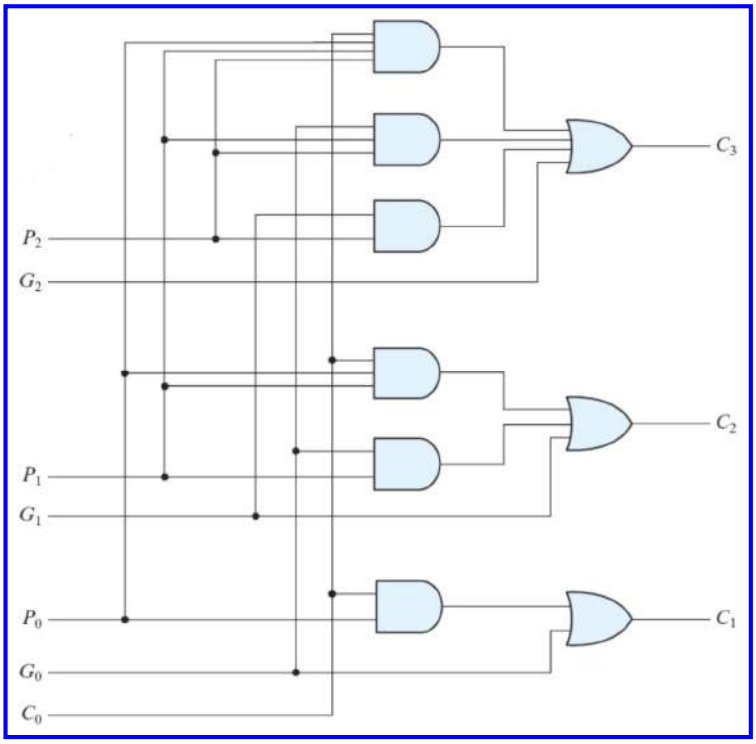
$$C_2 = G_1 + P_1C_1 = G_1 + P_1(G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0$$

$$C_3 = G_2 + P_2C_2 = \ldots = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$$

Γεννήτρια πρόβλεψης κρατουμένου

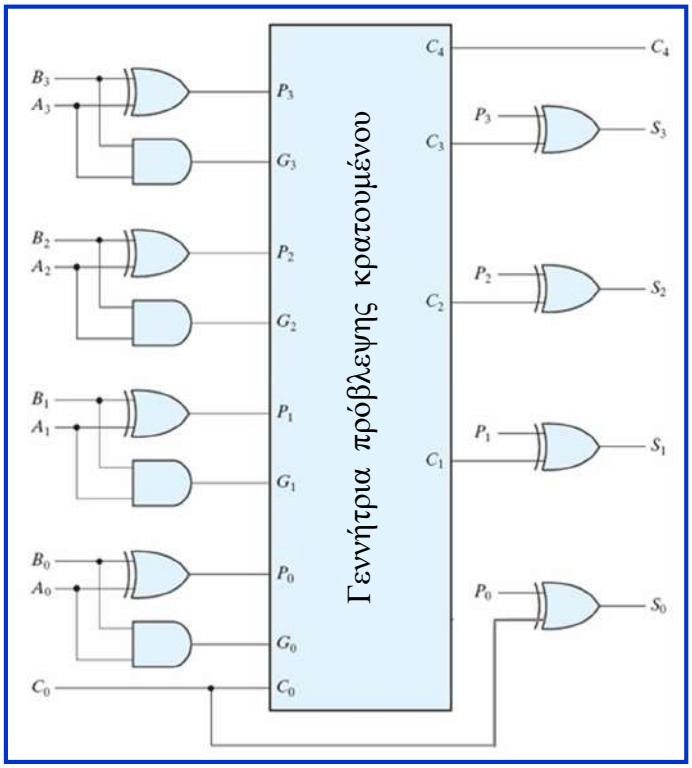
$$C_1 = G_0 + P_0C_0$$
$$C_2 = G_1 + P_1G_0 + P_1P_0C_0$$
$$C_3 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$$

Κάθε κρατούμενο υλοποιείται με ένα επίπεδο πυλών AND ακολουθούμενο από μία πύλη OR



Αθροιστής πρόβλεψης κρατουμένου

$$P_i = A_i \oplus B_i$$
$$G_i = A_iB_i$$



$$S_i = P_i \oplus C_i$$

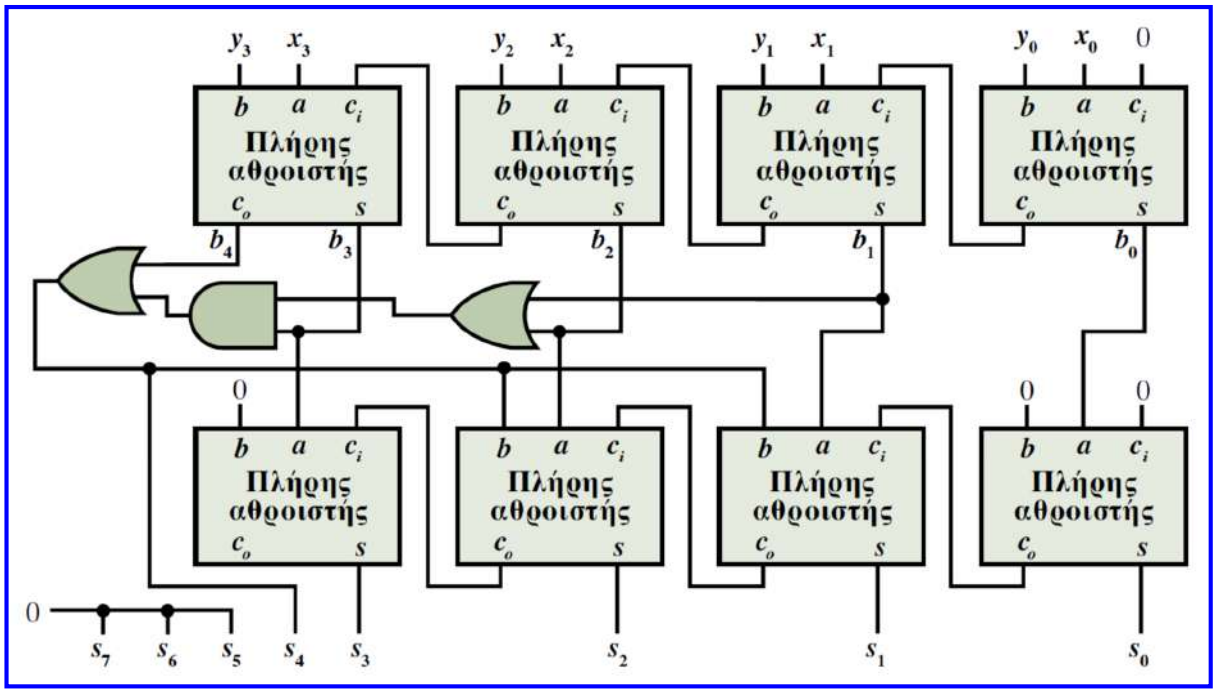
Αθροιστής BCD

- Ο κώδικας BCD χρησιμοποιείται στην κωδικοποίηση των 10 δεκαδικών ψηφίων και είναι ισοδύναμος με τη δυαδική παράσταση των ψηφίων αυτών.
- Κατά την πρόσθεση 2 δεκαδικών ψηφίων κωδικοποιημένων κατά BCD, όταν το δυαδικό άθροισμα είναι ίσο ή μικρότερο του 1001 (δηλαδή του 9), τότε ταυτίζεται με το άθροισμα κατά BCD.
- Όταν το δυαδικό άθροισμα είναι μεγαλύτερο του 1001 (και προφανώς δεν περιλαμβάνεται στον κώδικα BCD), για να ληφθεί το σωστό άθροισμα κατά BCD, θα πρέπει να προσθέσουμε στο δυαδικό άθροισμα τον αριθμό 0110 (δηλαδή το 6).
- Αυτό θα έχει αποτέλεσμα τη δημιουργία κρατουμένου δυαδικού ψηφίου.
- Για πρόσθεση 2 δεκαδικών ψηφίων κωδικοποιημένων κατά BCD, χρησιμοποιούνται δύο αθροιστές διάδοσης κρατουμένου με 4 δυαδικά ψηφία.
- Με τον πρώτο από αυτούς εξάγεται το δυαδικό άθροισμα των ψηφίων, το οποίο, όταν είναι μικρότερο του 1001, αποτελεί το επιθυμητό αποτέλεσμα, αλλά όταν είναι μεγαλύτερο του 1001, απαιτείται διόρθωση μέσω πρόσθεσης με τον αριθμό 0110, κατά την οποία προκύπτει τελικό κρατούμενο.
- Η πρόσθεση αυτή μπορεί υλοποιείται από τον δεύτερο αθροιστή διάδοσης κρατουμένου.

Αθροιστής BCD

- Διόρθωση του δυαδικού αθροίσματος απαιτείται όταν αυτό ισούται με τους συνδυασμούς που δεν περιλαμβάνονται στον κώδικα BCD (1010, 1011, 1100, 1101, 1110, 1111), καθώς και όταν το δυαδικό άθροισμα συνοδεύεται από τη δημιουργία τελικού κρατουμένου (b_4), δηλαδή είναι μεγαλύτερο ή ίσο του 16.
- Επομένως, διόρθωση του δυαδικού αθροίσματος $b_3b_2b_1b_0$, απαιτείται όταν $b_3 = 1$ και επιπλέον $b_2 = 1$ ή $b_1 = 1$, καθώς και όταν $b_4 = 1$.
- Η προϋπόθεση αυτή περιγράφεται από τη λογική έκφραση $b_3(b_2 + b_1) + b_4$ και, όταν συμβαίνει, το άθροισμα κατά BCD αποτελείται από 2 δεκαδικά ψηφία με πιο σημαντικό το 1 (0001 σε κώδικα BCD).
- Η διορθωτική πρόσθεση του δυαδικού αθροίσματος με τον αριθμό 0110 (6) γίνεται μόνο όταν $b_3(b_2 + b_1) + b_4 = 1$ και η λογική τιμή της έκφρασης αυτής χρησιμοποιείται ως το πιο σημαντικό ψηφίο του αθροίσματος κατά BCD

Αθροιστής BCD



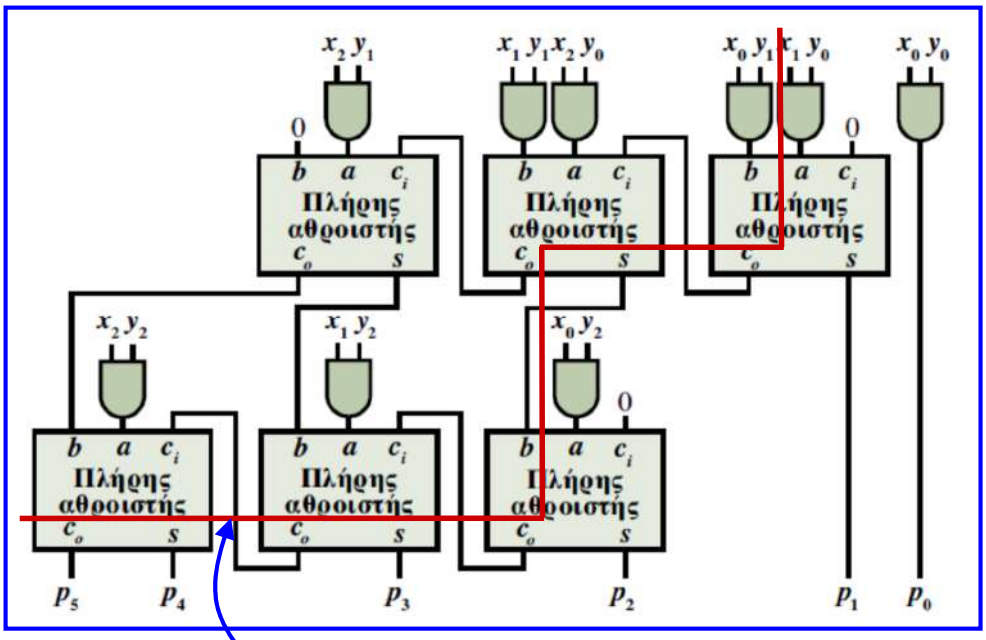
Πολλαπλασιαστής τριψήφιων δυαδικών αριθμών

- Για τη σχεδίαση πολλαπλασιαστή τριψήφιων δυαδικών αριθμών απαιτούνται 9 πύλες AND 2 εισόδων, ώστε να προκύψουν τα ισάριθμα μερικά γινόμενα.
- Το λιγότερο σημαντικό ψηφίο του γινομένου λαμβάνεται στην έξοδο της πύλης AND με εισόδους x_0 και y_0.
- Ο υπολογισμός των υπόλοιπων ψηφίων του γινομένου, προκύπτει με πρόσθεση των μερικών γινομένων κάθε στήλης, αφού ληφθεί υπόψη κάθε φορά το κρατούμενο που προκύπτει από την πρόσθεση στην προηγούμενη στήλη.

$$
\begin{array}{cccccc}
 & & x_2 & x_1 & x_0 & \\
\times & & y_2 & y_1 & y_0 & \\
\hline
 & & x_2y_0 & x_1y_0 & x_0y_0 & \\
 & x_2y_1 & x_1y_1 & x_0y_1 & & \\
+ & x_2y_2 & x_1y_2 & x_0y_2 & & \\
\hline
p_5 & p_4 & p_3 & p_2 & p_1 & p_0
\end{array}
$$

- Με βάση τη διάταξη των στηλών των μερικών γινομένων, καθώς και τον αριθμό των προσθετέων που προκύπτουν κατά την εκτέλεση της πράξης, προκύπτει για να υλοποιηθεί ο πολλαπλασιαστής τριψήφιων δυαδικών αριθμών με αθροιστές διάδοσης κρατουμένου, απαιτούνται 2 αθροιστές διάδοσης κρατουμένου για δυαδικούς αριθμούς με 3 ψηφία.

Πολλαπλασιαστής τριψήφιων δυαδικών αριθμών



Διαδρομή μέγιστης καθυστέρησης:
1 AND + 2 FA + 3 FA = 1 AND + (2 AND + 1 XOR) + 3 x 2 AND = 10 πύλες

Συγκριτής προσημασμένων δυαδικών αριθμών

- Η σύγκριση δύο προσημασμένων αριθμών, οι οποίοι παριστάνονται με μορφή συμπληρώματος ως προς 2, γίνεται μέσω ελέγχου του αποτελέσματος της αφαίρεσής τους, η οποία εκτελείται με το προαναφερόμενο κύκλωμα παράλληλου αφαιρέτη.

- Το μηδενικό αποτέλεσμα της αφαίρεσης ανιχνεύεται μέσω μιας εξόδου z που λαμβάνει τιμή 1, όταν και οι 4 έξοδοι του αφαιρέτη λαμβάνουν τιμή 0 (υλοποίηση με μία πύλη NOR 4 εισόδων).

- Το αρνητικό αποτέλεσμα της αφαίρεσης ανιχνεύεται μέσω μιας εξόδου n, η οποία λαμβάνει τιμή 1, όταν ο αριθμός που συνιστούν οι έξοδοι του αφαιρέτη είναι αρνητικός (η έξοδος n ταυτίζεται με την πιο σημαντική έξοδο του αφαιρέτη, αφού το πρόσημο των αρνητικών αριθμών σε παράσταση συμπληρώματος ως προς 2 έχει λογική τιμή 1).

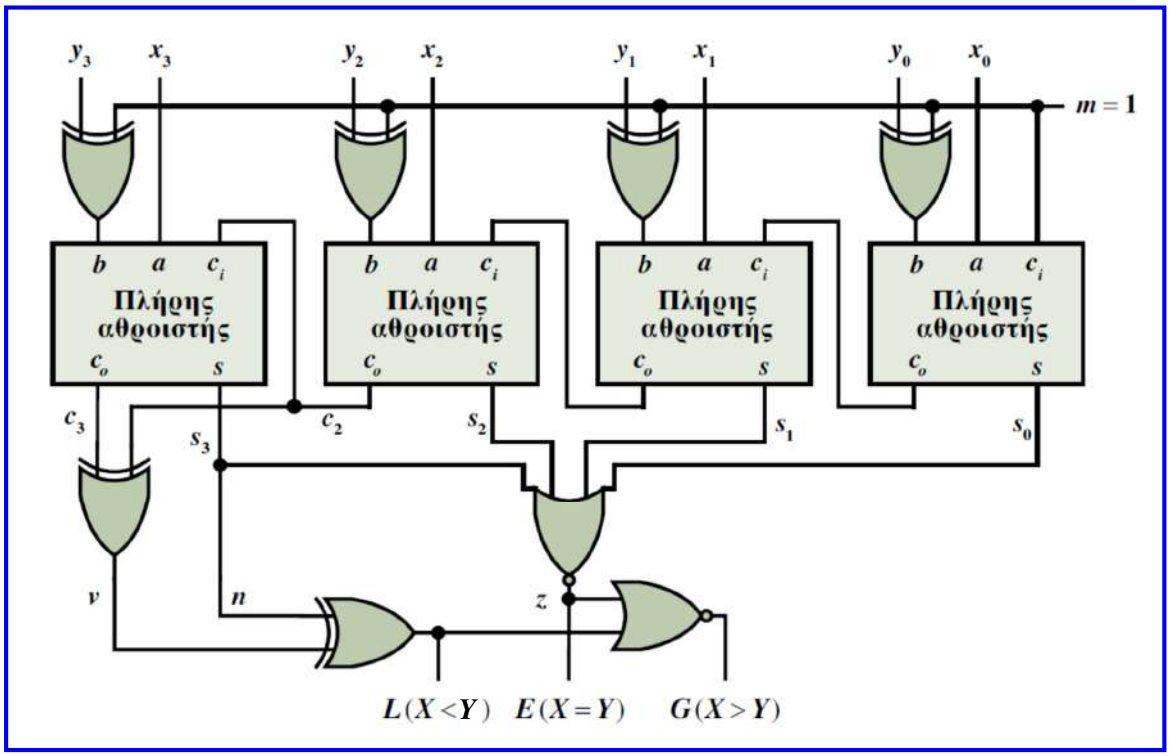
- Η υπερχείλιση (δηλαδή η μετατόπιση του ψηφίου-προσήμου του αποτελέσματος της αφαίρεσης από την αναμενόμενη πιο σημαντική θέση) ανιχνεύεται όταν η έξοδος v = 1.

- Όπως προαναφέρθηκε, υπερχείλιση κατά την αφαίρεση δύο προσημασμένων αριθμών (X - Y) συμβαίνει όταν ο αριθμός X είναι θετικός, ο αριθμός Y είναι αρνητικός και το αποτέλεσμα της πράξης είναι αρνητικό, καθώς και όταν ο αριθμός X είναι αρνητικός, ο αριθμός Y είναι θετικός και το αποτέλεσμα της πράξης είναι θετικό.

Συγκριτής προσημασμένων δυαδικών αριθμών

- Για να είναι ίσοι δύο προσημασμένοι αριθμοί X και Y, θα πρέπει το αποτέλεσμα της αφαίρεσης X - Y να είναι 0, δηλαδή θα πρέπει z = 1, συνεπώς η ζητούμενη έξοδος E (equal) ταυτίζεται με την έξοδο z.
- Όταν οι δύο αριθμοί είναι ομόσημοι (που σημαίνει ότι δεν συμβαίνει υπερχείλιση κατά την αφαίρεση X – Y, δηλαδή v = 0) και το αποτέλεσμα της αφαίρεσης είναι αρνητικό (δηλαδή n = 1), τότε X < Y.
- Επίσης, η ανισότητα X < Y ισχύει και όταν ο αριθμός X είναι αρνητικός και ο αριθμός Y είναι θετικός. Στην περίπτωση αυτή το αποτέλεσμα της αφαίρεσης είναι αρνητικό (n = 1), εάν δε συμβεί υπερχείλιση (v = 0), ή θετικό (n = 0), εάν συμβεί υπερχείλιση (v = 1).
- Επομένως, η ανισότητα X < Y ισχύει, όταν οι έξοδοι n και v λαμβάνουν διαφορετική λογική τιμή, με αποτέλεσμα η έξοδος L (less) να παράγεται εάν τροφοδοτηθεί μια πύλη XOR δύο εισόδων με τις εξόδους του κυκλώματος n και v.
- Όταν X > Y, μια πρόσθετη έξοδος G (great) θα πρέπει να λαμβάνει τιμή 1.
- Στην περίπτωση αυτή, όμως, οι έξοδοι E και L λαμβάνουν τιμή 0, συνεπώς G = (E + L)', με την έξοδο G να υλοποιείται ευκολά με μία πύλη NOR δύο εισόδων.

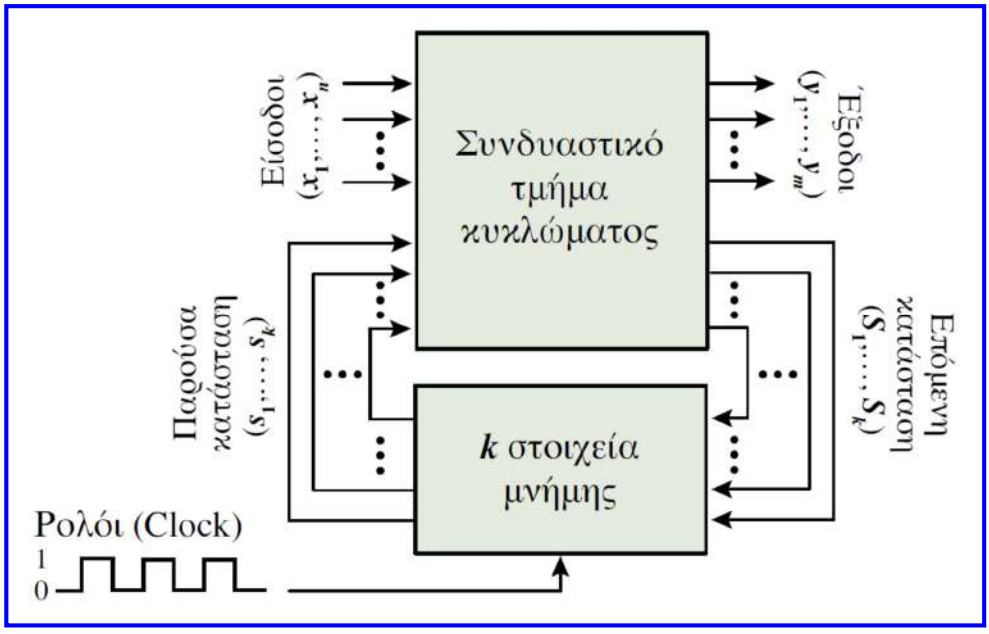
Συγκριτής προσημασμένων δυαδικών αριθμών



Κεφάλαιο 5: Σύγχρονα ακολουθιακά κυκλώματα

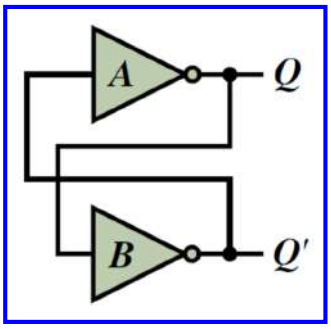
- Στα συνδυαστικά κυκλώματα, η λογική τιμή της εξόδου ή των εξόδων τους, κάθε χρονική στιγμή, εξαρτάται μόνο από τη λογική τιμή των εισόδων που εφαρμόζεται σε αυτά την ίδια χρονική στιγμή.
- Τα ακολουθιακά κυκλώματα (sequential circuits), εκτός από ένα συνδυαστικό τμήμα, περιλαμβάνουν στοιχεία (ή κυκλώματα) μνήμης που αναφέρονται ως μανταλωτές (latches) και φλιπ-φλοπ (flip-flops).
- Κάθε στοιχείο μνήμης μπορεί να αποθηκεύσει πληροφορία ενός δυαδικού ψηφίου.
- Έτσι, σε ένα ακολουθιακό κύκλωμα συμμετέχουν τόσα στοιχεία μνήμης, όσα και τα δυαδικά ψηφία των οποίων απαιτείται η αποθήκευση.
- Η πληροφορία που είναι αποθηκευμένη στα στοιχεία μνήμης ενός ακολουθιακού κυκλώματος αποτελεί την κατάσταση (state) του κυκλώματος.
- Η παρούσα ή τρέχουσα κατάσταση (current state) του κυκλώματος ανατροφοδοτείται στην είσοδο του συνδυαστικού τμήματός του.
- Οι τιμές των εισόδων και η παρούσα κατάσταση ενός ακολουθιακού κυκλώματος καθορίζουν τις τιμές των εξόδων του και την επόμενη κατάστασή (next state) του.
- Η κατάσταση ενός σύγχρονου ακολουθιακού κυκλώματος μπορεί να αλλάξει μόνο σε διακριτές χρονικές στιγμές, οι οποίες καθορίζονται από μία περιοδική σειρά παλμών που συνιστά ένα σήμα, το οποίο αναφέρεται ως ρολόι (clock).

Βασική δομή σύγχρονου ακολουθιακού κυκλώματος



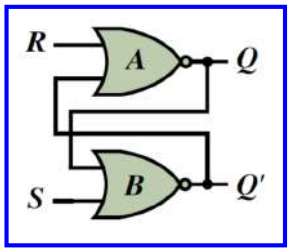
Κυκλώματα μνήμης ακολουθιακών κυκλωμάτων

- Οι μανταλωτές είναι κυκλώματα με ανατροφοδότηση, τα οποία «παρακολουθούν» τις εισόδους τους και ανάλογα με την τιμή τους διατηρούν ή μεταβάλλουν τις τιμές των εξόδων τους σε οποιαδήποτε στιγμή (ή σε οποιαδήποτε στιγμή του διαστήματος στο οποίο μια είσοδος ενεργοποίησης ή ελέγχου έχει τιμή 1).
- Η μεταβολή των εξόδων των ακμοπυροδοτούμενων φλιπ-φλοπ συμβαίνει μόνο σε διακριτές χρονικές στιγμές, οι οποίες καθορίζονται από το σήμα ρολογιού που εφαρμόζεται σ' αυτά. Χρησιμοποιούνται ως στοιχεία μνήμης στα σύγχρονα ακολουθιακά κυκλώματα.
- Στο διπλανό απλό κύκλωμα με ανατροφοδότηση, αν υποθέσουμε ότι Q = 0, τότε η έξοδος του αντιστροφέα B θα λάβει τιμή 1 (δηλαδή, Q'), αφού η έξοδος Q του αντιστροφέα A συνδέεται στην είσοδο του αντιστροφέα B.
- Η έξοδος του αντιστροφέα B συνδέεται στην είσοδο του αντιστροφέα A, με αποτέλεσμα Q = 0, γεγονός που είναι σύμφωνο με την υπόθεσή μας.
- Παρομοίως, αν υποθέσουμε ότι Q = 1, τότε η έξοδος του αντιστροφέα B λαμβάνει τιμή 0 (δηλ. ξανά Q').



- Οι έξοδοι του κυκλώματος είναι πάντα συμπληρωματικές μεταξύ τους και όταν συμβεί μία από τις δύο καταστάσεις (δηλαδή, Q = 1 ή Q = 0), αυτή παραμένει, με αποτέλεσμα η πληροφορία ενός ψηφίου να κλειδώνεται (μανταλώνεται) στο κύκλωμα.

Μανταλωτής SR

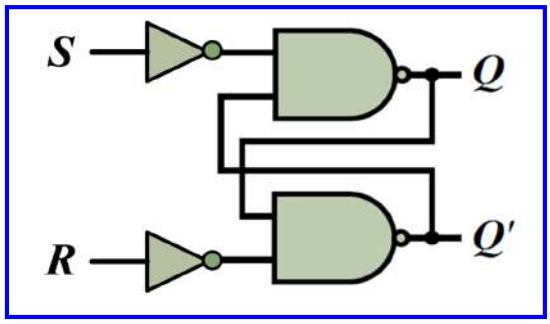


S	R	Q_{t+1}	Q'_{t+1}	Παρατηρήσεις
0	0	Q_t	Q'_t	Αμετάβλητη κατάσταση
0	1	0	1	Κατάσταση μηδενισμού
1	0	1	0	Κατάσταση θέσης
1	1	0	0	Απροσδιόριστη κατάσταση

- Το προηγούμενο κύκλωμα δεν μπορεί να αποθηκεύσει την επιθυμητή δυαδική πληροφορία.
- Η προσθήκη της δυνατότητας αυτής μπορεί να γίνει μέσω της αντικατάστασης των αντιστροφέων με λογικές πύλες NOR.
- Όταν S = R = 1, τότε οι έξοδοι των πυλών NOR λαμβάνουν τιμή 0, με αποτέλεσμα να μην είναι συμπληρωματικές μεταξύ τους.
- Αυτές οι τιμές εισόδου συνιστούν μια απροσδιόριστη κατάσταση, λόγω του ότι όταν οι δύο είσοδοι λάβουν τιμή 0, η επόμενη κατάσταση του μανταλωτή δεν μπορεί να προβλεφθεί.
- Εάν, για παράδειγμα, οι δύο είσοδοι επιστρέψουν ταυτόχρονα στη λογική τιμή 0, τότε η τιμή των εξόδων και των δύο πυλών θα αλλάξει από 0 σε 1 και θα ανατροφοδοτηθεί στις εισόδους, με αποτέλεσμα οι έξοδοι να επιστρέψουν σε τιμή 0.
- Η εναλλαγή λογικών τιμών στις εξόδους του μανταλωτή θα συνεχιστεί. Αυτή η μη επιθυμητή κατάσταση αναφέρεται ως ταλάντωση (oscillation) του μανταλωτή.

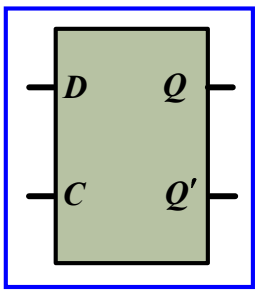
Μανταλωτής SR

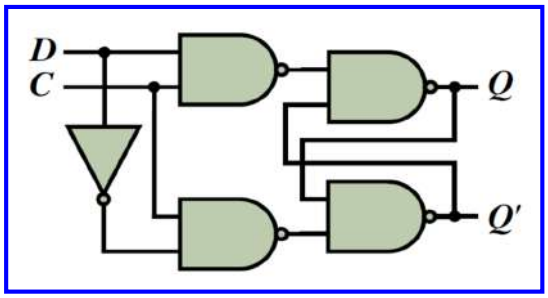
- Ο μανταλωτής SR μπορεί επίσης να υλοποιηθεί με δύο πύλες NAND, τροφοδοτούμενες με τις συμπληρωματικές μορφές των εισόδων.
- Όταν S = R = 1, οι έξοδοι των πυλών NAND λαμβάνουν τιμή 0 και δεν είναι συμπληρωματικές μεταξύ τους (απροσδιόριστη κατάσταση).



Μανταλωτής D

- Μια λύση, για την μη επιθυμητή (απροσδιόριστη) κατάσταση είναι ο μανταλωτής D, ο οποίος περιλαμβάνει μία μόνο είσοδο δεδομένων (D), μια είσοδο ελέγχου (C), έναν αντιστροφέα, έναν μανταλωτή SR και δύο πύλες NAND 2 εισόδων.
- Όταν C = 0, η κατάσταση του μανταλωτή παραμένει αμετάβλητη, ενώ όταν C = 1 η έξοδος του μανταλωτή ακολουθεί την τιμή της εισόδου δεδομένων.

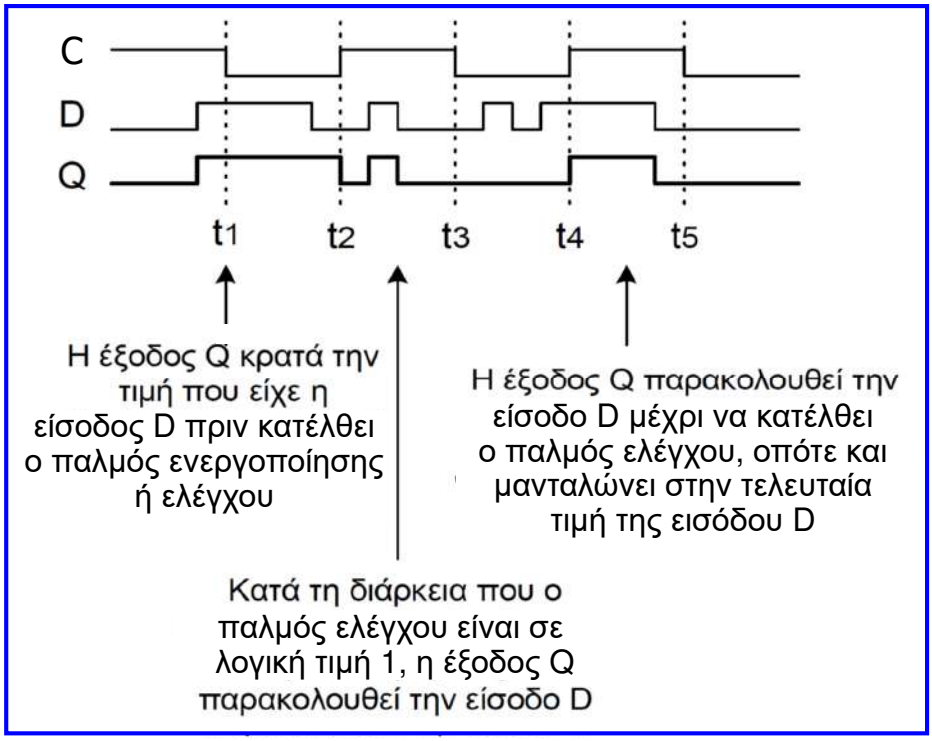




C	D	Q_{t+1}	Q'_{t+1}	Παρατηρήσεις
0	0	Q_t	Q'_t	Αμετάβλητη κατάσταση
0	1	Q_t	Q'_t	Αμετάβλητη κατάσταση
1	0	0	1	Κατάσταση μηδενισμού
1	1	1	0	Κατάσταση θέσης

$$Q_{t+1} = CD + C'Q_t$$

Μανταλωτής D



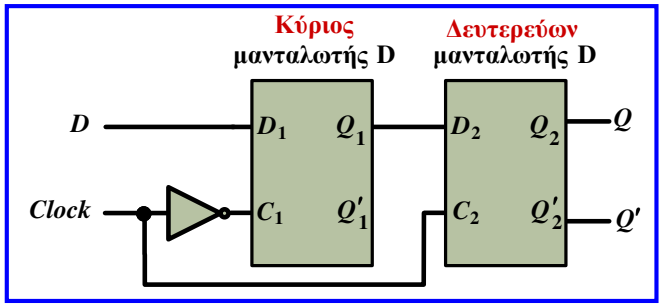
Ακμοπυροδοτούμενα φλιπ-φλοπ

- Εάν στην είσοδο ελέγχου (C) του μανταλωτή D εφαρμοστεί ένα σήμα ρολογιού (δηλαδή μια περιοδική σειρά παλμών με δύο στάθμες που αντιστοιχούν στις λογικές τιμές 0 και 1), η είσοδος δεδομένων του μανταλωτή μπορεί να μεταφερθεί στην έξοδό του οποιαδήποτε στιγμή του χρονικού διαστήματος κατά το οποίο το σήμα ρολογιού έχει λογική τιμή 1.
- Ωστόσο, στα σύγχρονα ακολουθιακά κυκλώματα (ΣΑΚ) είναι επιθυμητό η κατάσταση των στοιχείων μνήμης να αλλάζει μόνο σε διακριτές χρονικές στιγμές.
- Για το λόγο αυτόν έχουν αναπτυχθεί φλιπ-φλοπ που επιτρέπουν αλλαγή κατάστασης μόνο κατά την αλλαγή της λογικής τιμής του σήματος ρολογιού (δηλ. στις ακμές των παλμών του), τα οποία αναφέρονται ως ακμοπυροδοτούμενα φλιπ-φλοπ (edge-triggered flip-flops).
- Ένα ακμοπυροδοτούμενο φλιπ-φλοπ μπορεί να αλλάξει κατάσταση κατά την ανερχόμενη ή την κατερχόμενη ακμή του σήματος ρολογιού.
- Η βασική διαφορά των φλιπ-φλοπ με τους μανταλωτές με είσοδο ελέγχου έγκειται στο ότι μεταβολή της τιμής των εξόδων των φλιπ-φλοπ συμβαίνει μόνο σε διακριτές χρονικές στιγμές, οι οποίες καθορίζονται από ένα σήμα ρολογιού.
- Τα ακμοπυροδοτούμενα φλιπ-φλοπ χρησιμοποιούνται ως στοιχεία μνήμης στα ΣΑΚ.
- Υπάρχουν διάφορα φλιπ-φλοπ (D, JK, T), τα οποία διαφέρουν στον τρόπο με τον οποίο οι είσοδοι δεδομένων τους επηρεάζουν τη δυαδική πληροφορία που αποθηκεύεται σε αυτά, δηλαδή την κατάστασή τους.

Φλιπ-φλοπ D

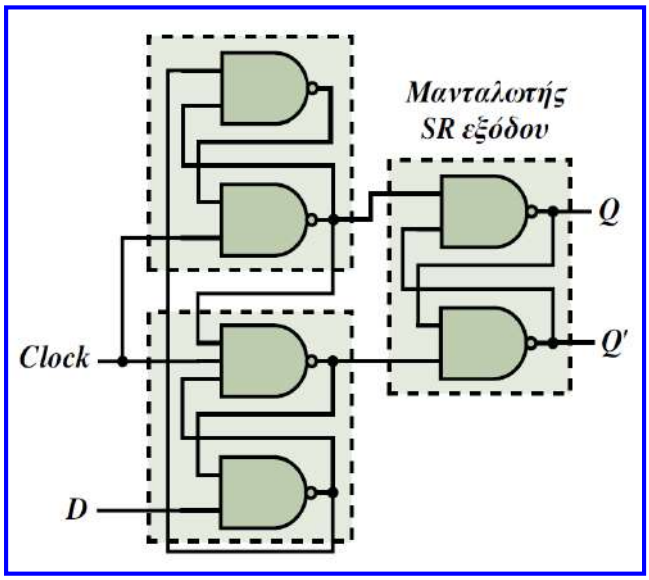
- Ένα ακμοπυροδοτούμενο στην ανερχόμενη ακμή φλιπ-φλοπ D, συνδυάζει 2 μανταλωτές D και αναφέρεται ως φλιπ-φλοπ κύριου-δευτερεύοντος (master-slave flip-flop).
- Η είσοδος D μεταφέρεται στην έξοδο του κύριου μανταλωτή, όταν Clock = 0.
- Στο διάστημα κατά το οποίο Clock = 0, η έξοδος του κύριου και, κατά συνέπεια, η είσοδος του δευτερεύοντος μανταλωτή δεν μπορεί να μεταφερθεί στην έξοδο Q.
- Όταν Clock 0 → 1, η έξοδος του κύριου μανταλωτή μεταφέρεται στην έξοδο Q.
- Στο διάστημα κατά το οποίο Clock = 1, η έξοδος του κύριου μανταλωτή δεν αλλάζει.
- Έτσι, η κατάσταση του φλιπ-φλοπ μπορεί να αλλάξει μόνο στην ανερχόμενη ακμή του Clock.
- Για φλιπ-φλοπ D με αλλαγή κατάστασης στην κατερχόμενη ακμή του Clock τροφοδοτούμε τον κύριο μανταλωτή με την κανονική μορφή του Clock και τον δευτερεύοντα με Clock'.

Ακμοπυροδοτούμενο φλιπ-φλοπ D στην ανερχόμενη ακμή του ρολογιού



Φλιπ-φλοπ D

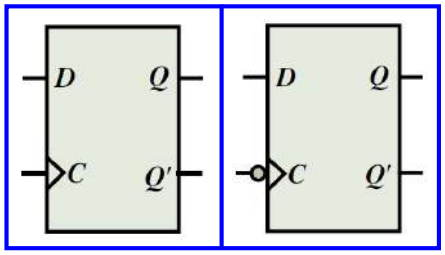
- Μια εναλλακτική υλοποίηση ακμοπυροδοτούμενου στην ανερχόμενη ακμή φλιπ-φλοπ D, που αποτελείται από μικρότερο αριθμό πυλών (μικρότερο κόστος), χρησιμοποιεί τρεις μανταλωτές SR.
- Clock = 0 ⇒ S = R = 1 ⇒ αμετάβλητη έξοδος του μανταλωτή εξόδου.
- Clock: 0 → 1, D = 0 ⇒ R: 1 → 0 ⇒ κατάσταση μηδενισμού του μανταλωτή εξόδου Q = 0 (ακόμη κι αν αλλάξει η τιμή του D, όσο Clock = 1, η είσοδος R του μανταλωτή εξόδου παραμένει 0 και δεν επηρεάζεται η έξοδός του).
- Clock: 0 → 1, D = 1 ⇒ S: 1 → 0 ⇒ κατάσταση θέσης του μανταλωτή εξόδου Q = 1 (ακόμη κι αν αλλάξει η τιμή του D, όσο Clock = 1, η είσοδος S του μανταλωτή εξόδου παραμένει 0 και δεν επηρεάζεται η έξοδός του).
- Clock: 1 → 0 ⇒ S = R = 1 ⇒ αμετάβλητη έξοδος του μανταλωτή εξόδου.



Φλιπ-φλοπ D

- Για την περιγραφή της λειτουργίας των ακμοπυροδοτούμενων φλιπ-φλοπ χρησιμοποιούνται ο πίνακας λειτουργίας (operation table) ή χαρακτηριστικός πίνακας (characteristic table) και η χαρακτηριστική εξίσωση (characteristic equation).
- Κατά τη σύνθεση σύγχρονων ακολουθιακών κυκλωμάτων είναι επιθυμητό με βάση την παρούσα και την επόμενη κατάσταση ενός φλιπ-φλοπ να προσδιορίσουμε τις τιμές της εισόδου ή των εισόδων του που προκαλούν τη σχετική μετάβαση.
- Αυτό επιτυγχάνεται με τον πίνακα διέγερσης (excitation table) των φλιπ-φλοπ.

Σύμβολα του φλιπ-φλοπ D



$$Q_{t+1} = D$$

Χαρακτηριστική εξίσωση

D	Q_t	Q_{t+1}
0	0	0
0	1	0
1	0	1
1	1	1

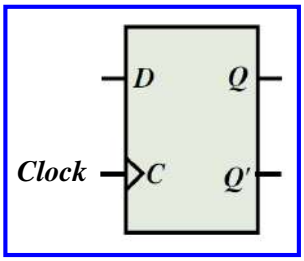
D	Q_{t+1}
0	0
1	1

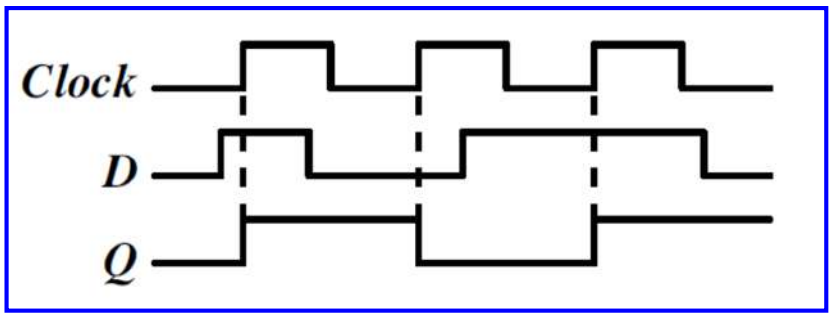
Πίνακες λειτουργίας ή χαρακτηριστικοί πίνακες

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Πίνακας διέγερσης

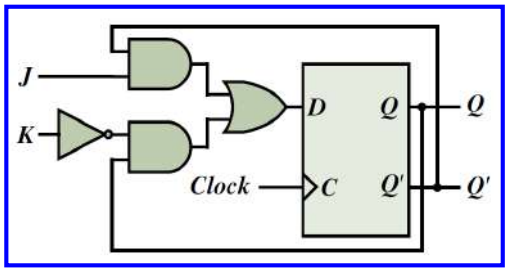
Φλιπ-φλοπ D





Φλιπ-φλοπ JK

- Με το φλιπ-φλοπ D είναι δυνατή η μεταφορά των τιμών της εισόδου D στην έξοδό του, κατά την ακμή του σήματος ρολογιού, ενώ με το JK flip-flop είναι δυνατή η οδήγηση της εξόδου σε τιμή 0 ή 1, καθώς και η λήψη της συμπληρωματικής της τιμής.
- Για να επιτευχθούν οι τρεις αυτές λειτουργίες, το JK flip-flop περιλαμβάνει δύο εισόδους δεδομένων (J, K) και μπορεί να υλοποιηθεί με ένα φλιπ-φλοπ D και 4 λογικές πύλες.



Q_t	Q_{t+1}	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

Πίνακας διέγερσης

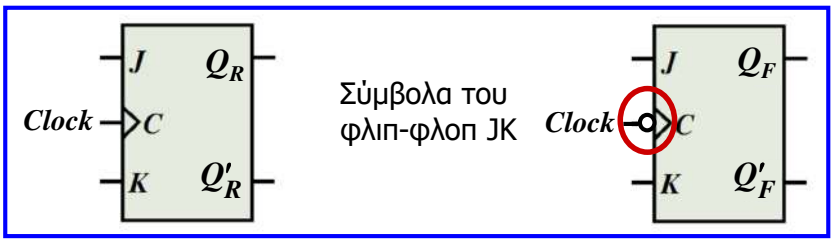
J	K	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t

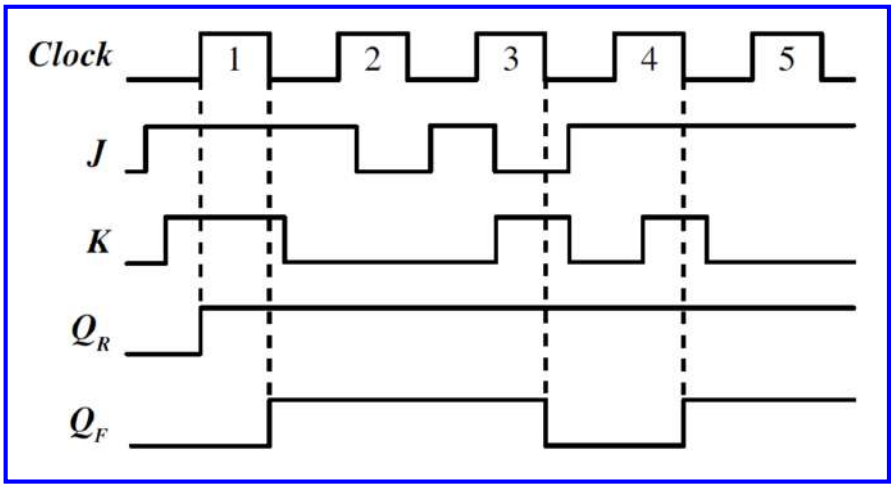
Πίνακες λειτουργίας

Χαρακτηριστική εξίσωση $Q_{t+1} = JQ'_t + K'Q_t$

Φλιπ-φλοπ JK



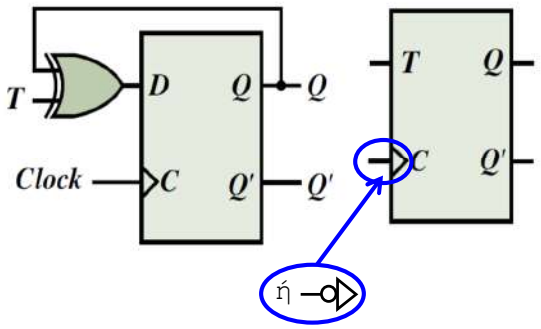
J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t



Φλιπ-φλοπ Τ

Το φλιπ-φλοπ Τ περιλαμβάνει μία είσοδο δεδομένων (Τ) και υλοποιείται εύκολα με συνδυασμό ενός φλιπ-φλοπ D και μιας πύλης XOR δύο εισόδων

Υλοποίηση



Σύμβολο

Πίνακες λειτουργίας

T	Q_t	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

T	Q_{t+1}
0	Q_t
1	Q'_t

Πίνακας διέγερσης

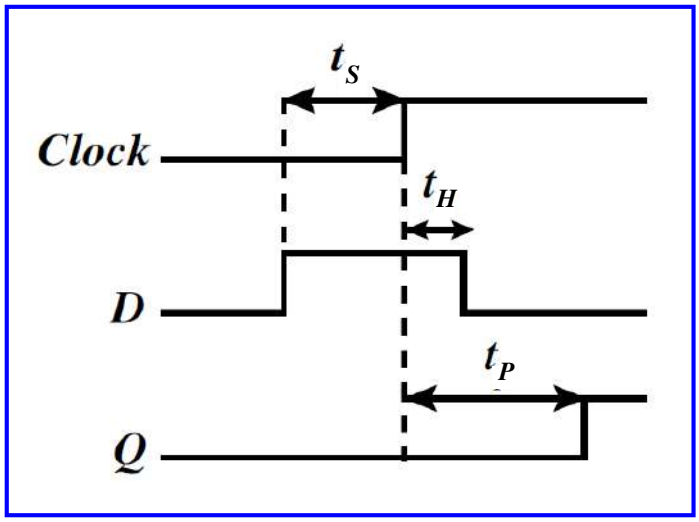
Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Χαρακτηριστική εξίσωση $Q_{t+1} = T \oplus Q_t$

Χρονικοί περιορισμοί στα φλιπ-φλοπ

- Η αλλαγή κατάστασης ενός ακμοπυροδοτούμενου φλιπ-φλοπ γίνεται ακριβώς τη χρονική στιγμή έλευσης της ακμής του σήματος ρολογιού, στην οποία πυροδοτείται το φλιπ-φλοπ.
- Η νέα κατάσταση σταθεροποιείται μετά την πάροδο χρονικού διαστήματος που αναφέρεται ως καθυστέρηση διάδοσης (propagation delay, t_P) και οφείλεται στην καθυστέρηση απόκρισης των λογικών πυλών που παρεμβάλλονται μεταξύ της εισόδου του σήματος ρολογιού και της εξόδου.
- Στα ακμοπυροδοτούμενα φλιπ-φλοπ, θα πρέπει να λαμβάνονται υπόψη δύο χρονικοί περιορισμοί που σχετίζονται με την ανταπόκριση τους στις αλλαγές τιμής της (των) εισόδου(ων) δεδομένων και του σήματος ρολογιού.
- Οι περιορισμοί αυτοί αφορούν το χρόνο προετοιμασίας (setup time, t_S) και το χρόνο παραμονής (hold time, t_H).
- Χρόνος προετοιμασίας σε ένα φλιπ-φλοπ D είναι το ελάχιστο χρονικό διάστημα πριν από την έλευση της ακμής του σήματος ρολογιού, στην οποία πυροδοτείται το φλιπ-φλοπ, κατά το οποίο η είσοδος δεδομένων θα πρέπει να διατηρείται σταθερή στην επιθυμητή λογική τιμή, ώστε αυτή να μεταφερθεί στην έξοδο του φλιπ-φλοπ.
- Χρόνος παραμονής είναι το ελάχιστο χρονικό διάστημα μετά την έλευση της ακμής του σήματος ρολογιού στην οποία πυροδοτείται το φλιπ-φλοπ, κατά το οποίο η είσοδος δεδομένων θα πρέπει να παραμείνει σταθερή στην επιθυμητή λογική τιμή.

Χρονικοί περιορισμοί στα φλιπ-φλοπ



Ασύγχρονες είσοδοι στα φλιπ-φλοπ

- Η είσοδος δεδομένων (D) του φλιπ-φλοπ αποτελεί σύγχρονη είσοδο (synchronous input), αφού η επίδρασή της στη δυαδική πληροφορία που αποθηκεύεται στο φλιπ-φλοπ είναι συγχρονισμένη με τις ακμές του σήματος ρολογιού στις οποίες πυροδοτείται το στοιχείο.

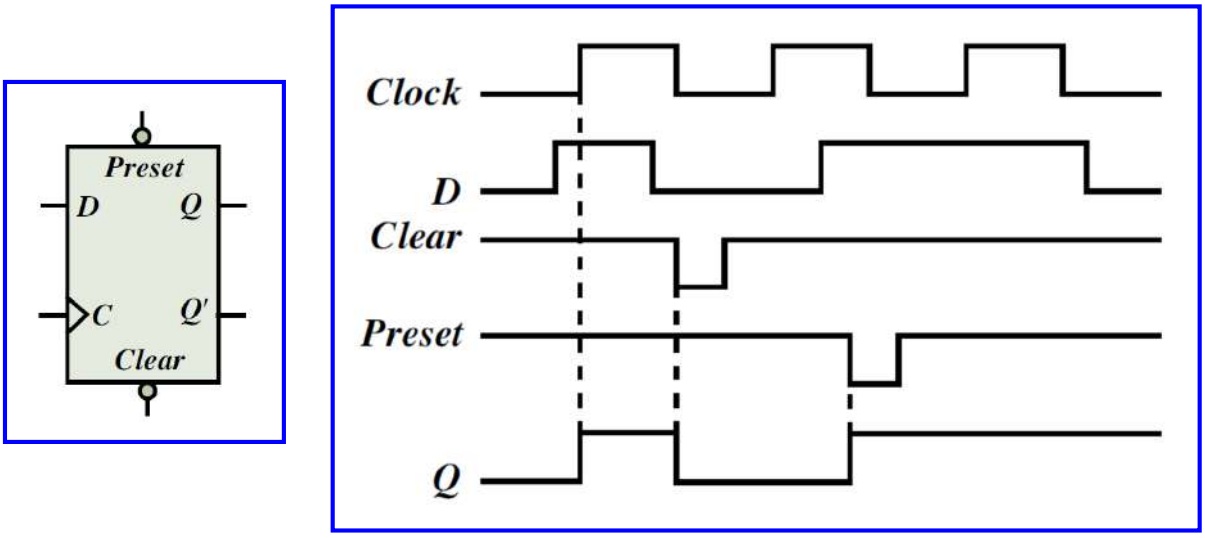
- Ωστόσο, στα φλιπ-φλοπ είναι δυνατή η προσθήκη ασύγχρονων εισόδων (asynchronous inputs), των οποίων η επίδραση στη λειτουργία των στοιχείων είναι άμεση και ανεξάρτητη από το σήμα ρολογιού.

- Οι είσοδοι αυτές χρησιμοποιούνται για να οδηγήσουν τα φλιπ-φλοπ σε κατάσταση θέσης ή σε κατάσταση μηδενισμού, σε οποιαδήποτε χρονική στιγμή και ανεξάρτητα από την τιμή των υπόλοιπων εισόδων.

- Για το λόγο αυτόν αναφέρονται ως είσοδοι πρόθεσης (preset) και καθαρισμού (clear).

- Οι είσοδοι πρόθεσης και καθαρισμού ενεργοποιούνται όταν λαμβάνουν λογική τιμή 0, και η περίπτωση ταυτόχρονης ενεργοποίησής τους θα πρέπει να αποφεύγεται, αφού οδηγεί το φλιπ-φλοπ σε απροσδιόριστη κατάσταση.

Ασύγχρονες είσοδοι στα φλιπ-φλοπ



Σύγχρονα ακολουθιακά κυκλώματα (ΣΑΚ)

- Η λειτουργία ενός συνδυαστικού κυκλώματος με n εισόδους (x_1 έως x_n) και m εξόδους (y_1 έως y_m) περιγράφεται πλήρως από ένα σύνολο m εκφράσεων:

$$y_i = F_i(x_1, x_2, \ldots, x_n) \quad i \in [1, m]$$

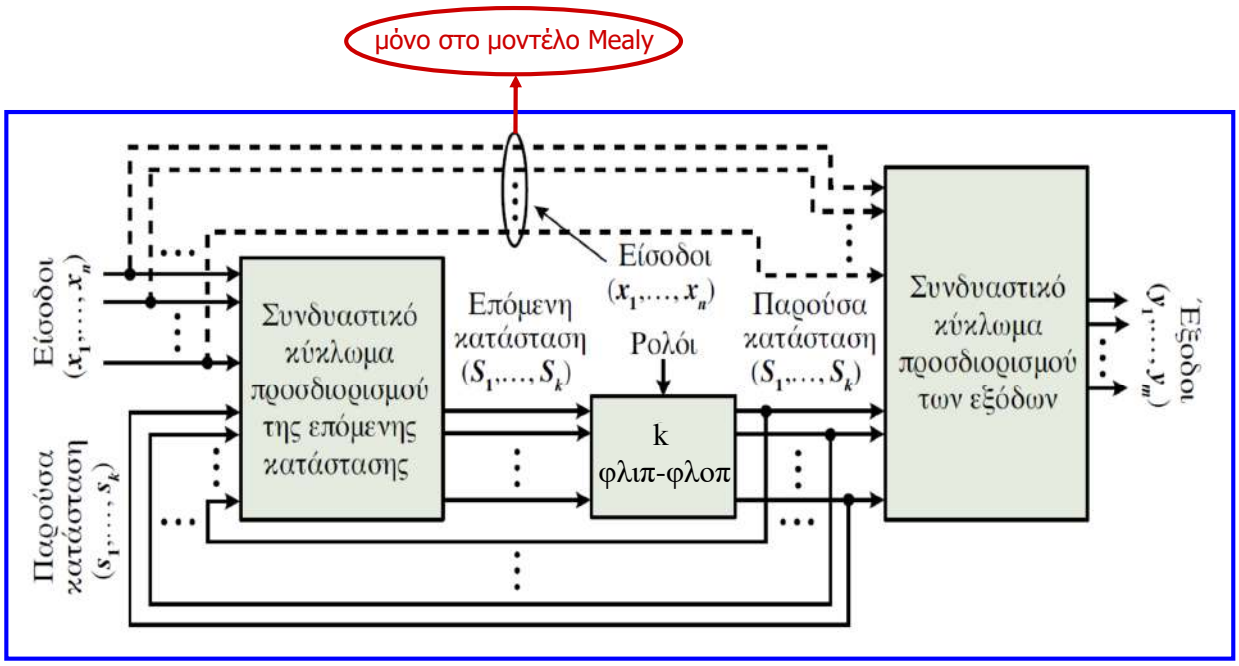
- Γενικά, η λειτουργία ενός σύγχρονου ακολουθιακού κυκλώματος (ΣΑΚ) με n εισόδους (x_1 έως x_n), m εξόδους (y_1 έως y_m) και k στοιχεία μνήμης (φλιπ-φλοπ), στα οποία αντιστοιχούν οι μεταβλητές κατάστασης, με τις μεταβλητές s_1 έως s_k και S_1 έως S_k να εκφράζουν την παρούσα και την επόμενη κατάσταση του, περιγράφεται ως εξής:

$$y_i = G_i(x_1, x_2, \ldots, x_n, s_1, s_2, \ldots, s_k) \qquad i \in [1, m]$$

$$S_i = H_i(x_1, x_2, \ldots, x_n, s_1, s_2, \ldots, s_k) \qquad i \in [1, k]$$

- Όταν οι έξοδοι ενός ΣΑΚ περιγράφονται από λογικές συναρτήσεις των μεταβλητών που εκφράζουν την παρούσα κατάσταση του και των μεταβλητών εισόδου, τότε το ΣΑΚ περιγράφεται με το μοντέλο Mealy.
- Στην περίπτωση όπου οι έξοδοι ενός ΣΑΚ περιγράφονται από λογικές συναρτήσεις μόνο των μεταβλητών που εκφράζουν την παρούσα κατάσταση του, τότε το ΣΑΚ περιγράφεται με το μοντέλο Moore.

Σύγχρονα ακολουθιακά κυκλώματα (ΣΑΚ)



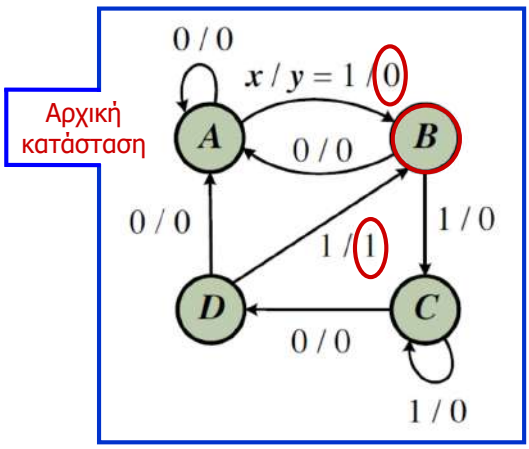
Περιγραφή λειτουργίας ΣΑΚ

- Ένας «φιλικός» και παραστατικός τρόπος περιγραφής των λειτουργικών σχέσεων ανάμεσα στις μεταβλητές που συμμετέχουν σε ένα ΣΑΚ, είναι το διάγραμμα καταστάσεων (state diagram).
- Οι διαφορετικές καταστάσεις του κυκλώματος παριστάνονται με κύκλους και οι μεταβάσεις από μια κατάσταση στην επόμενη παριστάνονται με βέλη από τον έναν κύκλο στον άλλο. Κάθε βέλος χαρακτηρίζεται από ένα συνδυασμό τιμών των εισόδων και τις αντίστοιχες λογικές τιμές των εξόδων του κυκλώματος.
- Η σχέση που συνδέει το πλήθος των καταστάσεων (s) ενός ΣΑΚ με το πλήθος των μεταβλητών (k) που απαιτούνται για την έκφρασή τους, έχει ως εξής: $2^{k-1} < s \leq 2^k$.
- Το πλήθος των μεταβλητών κατάστασης συμπίπτει με τον απαιτούμενο αριθμό των στοιχείων μνήμης (φλιπ-φλοπ) του ακολουθιακού κυκλώματος.
- Έτσι, για να εκφράσουμε 2 καταστάσεις απαιτείται 1 μεταβλητή και το ΣΑΚ περιλαμβάνει 1 φλιπ-φλοπ, από 3 έως 4 καταστάσεις απαιτούνται 2 μεταβλητές και 2 φλιπ-φλοπ, από 5 έως 8 καταστάσεις απαιτούνται 3 μεταβλητές και 3 φλιπ-φλοπ, ενώ για να εκφράσουμε από 9 έως 16 καταστάσεις απαιτούνται 4 μεταβλητές και 4 φλιπ-φλοπ, κ.ο.κ.
- Η έκφραση των καταστάσεων ενός ΣΑΚ με διαφορετικούς συνδυασμούς τιμών των μεταβλητών κατάστασης αναφέρεται ως κωδικοποίηση καταστάσεων (state encoding).

Περιγραφή λειτουργίας ΣΑΚ

Διάγραμμα καταστάσεων ΣΑΚ με 4 καταστάσεις (A, B, C, D), μία είσοδο (x) και μία έξοδο (y)

- Το ΣΑΚ λαμβάνει στην είσοδό x μία ακολουθία δυαδικών ψηφίων (σειριακά, ένα ψηφίο ανά παλμό σήματος ρολογιού) και η έξοδος y γίνεται 1, όταν στα ψηφία εισόδου ανιχνεύεται η ακολουθία 1101 και επικαλυπτόμενες εκδοχές της.
- Πρόκειται για ανιχνευτή ακολουθίας (sequence recognizer) που ενεργοποιεί την έξοδό του, όταν λάβει στην είσοδο του συγκεκριμένη ακολουθία.
- Η έξοδος y, κατά τη μετάβαση στις καταστάσεις A, C και D λαμβάνει πάντα τιμή 0.
- Ωστόσο, κατά τη μετάβαση στην κατάσταση B η έξοδος y λαμβάνει τιμή 0 ή 1.
- Επομένως, η έξοδος δεν είναι συνάρτηση μόνο της παρούσας κατάστασης, και η εν λόγω περιγραφή ακολουθεί το μοντέλο Mealy.

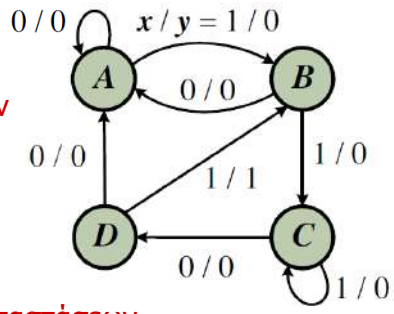


Περιγραφή λειτουργίας ΣΑΚ

- Ένας άλλος τρόπος παράστασης της λειτουργικής συμπεριφοράς ενός ΣΑΚ, ο οποίος παρέχει τις ίδιες ακριβώς πληροφορίες με το διάγραμμα καταστάσεων, είναι ο πίνακας καταστάσεων (state table).
- Ο πίνακας αυτός περιλαμβάνει στήλες με όλους τους δυνατούς συνδυασμούς τιμών των εισόδων και των καταστάσεων του κυκλώματος.
- Με βάση τη λειτουργική συμπεριφορά του ΣΑΚ, από το περιεχόμενο των στηλών αυτών προκύπτουν αντίστοιχες στήλες για την επόμενη κατάσταση και τις εξόδους του κυκλώματος.
- Ο πίνακας καταστάσεων ενός κυκλώματος με n εισόδους και s καταστάσεις περιλαμβάνει $2^n \times s$ γραμμές, δηλαδή για κάθε συνδυασμό τιμών εισόδου απαιτείται η προσθήκη τόσων γραμμών όσες και οι καταστάσεις του κυκλώματος.
- Όταν οι καταστάσεις του κυκλώματος κωδικοποιηθούν ή ανατεθούν σε αυτές διαφορετικοί συνδυασμοί τιμών των απαιτούμενων k μεταβλητών κατάστασης, τότε προκύπτει ο κωδικοποιημένος πίνακας καταστάσεων (encoded state table), ο οποίος περιλαμβάνει έως 2^{n+k} γραμμές.

Περιγραφή λειτουργίας ΣΑΚ

Διάγραμμα καταστάσεων



Πίνακας καταστάσεων

Είσοδος (x)	Παρούσα κατάσταση	Επόμενη κατάσταση	Έξοδος (y)
0	A	A	0
0	B	A	0
0	C	D	0
0	D	A	0
1	A	B	0
1	B	C	0
1	C	C	0
1	D	B	1

Κωδικοποιημένος πίνακας καταστάσεων

x	Παρούσα κατάσταση Q_1	Παρούσα κατάσταση Q_0	Επόμενη κατάσταση Q_1	Επόμενη κατάσταση Q_0	y
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	1	1

Οι καταστάσεις A, B, C, D κωδικοποιήθηκαν με τους συνδυασμούς τιμών των μεταβλητών κατάστασης Q_1Q_0 = 00, 01, 10, 11, αντίστοιχα

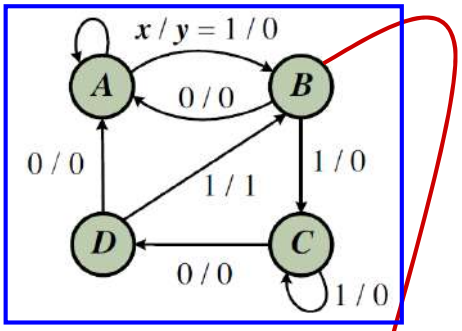
Περιγραφή λειτουργίας ΣΑΚ

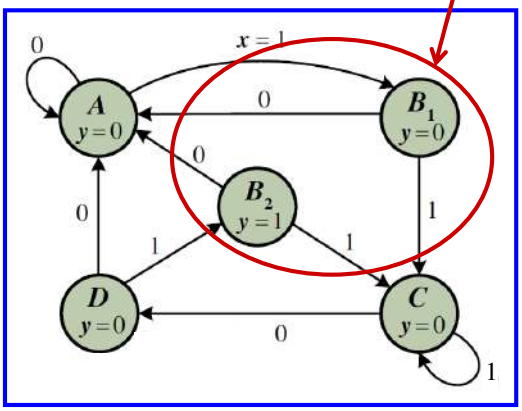
- Λόγω του ότι στις μεταβάσεις προς μία κατάσταση μοντέλου Mealy μπορούν να αντιστοιχούν διαφορετικές τιμές εξόδου, η περιγραφή μοντέλου Moore απαιτεί συνήθως περισσότερες καταστάσεις από εκείνη του μοντέλου Mealy με ισοδύναμη συμπεριφορά.
- Μια κατάσταση μοντέλου Mealy που αντιστοιχεί σε διαφορετικές τιμές εξόδου, αντιστοιχεί σε περισσότερες από μία καταστάσεις του μοντέλου Moore, λόγω του ότι κάθε κατάσταση μοντέλου Moore συσχετίζεται με μία μόνο τιμή εξόδου.
- Επομένως, τα ΣΑΚ που ακολουθούν το μοντέλο Mealy, συνήθως οδηγούν σε οικονομικότερη υλοποίηση, αφού τα κυκλώματα αυτά περιλαμβάνουν μικρότερο αριθμό καταστάσεων.
- Στην περιγραφή μοντέλου Moore, η τιμή της εξόδου δεν αναφέρεται στα βέλη που υποδεικνύουν τις μεταβάσεις αλλά μέσα στους κύκλους που υποδεικνύουν τις καταστάσεις.
- Στον αντίστοιχο πίνακα καταστάσεων οι τιμές της στήλης εξόδου αφορούν την παρούσα και όχι την επόμενη κατάσταση.

Περιγραφή λειτουργίας ΣΑΚ

- Αφού η κατάσταση Β του διαγράμματος Mealy συσχετίζεται με 2 διαφορετικές τιμές εξόδου, για την περιγραφή ενός κυκλώματος με ισοδύναμη λειτουργία, σύμφωνα με το μοντέλο Moore, θα πρέπει η κατάσταση Β να αντικατασταθεί με 2 επιμέρους καταστάσεις (B_1 και B_2), μία για κάθε τιμή εξόδου με την οποία συσχετίζεται.
- Οι επόμενες καταστάσεις των B_1 και B_2 είναι η κατάσταση Α για x = 0 και η C για x = 1.
- Προκύπτει ότι ενώ για την υλοποίηση του κυκλώματος που ακολουθεί το μοντέλο Mealy απαιτούνται 2 flip-flops (αφού το κύκλωμα διατρέχει 4 καταστάσεις), για την υλοποίηση του κυκλώματος που ακολουθεί το μοντέλο Moore απαιτούνται 3 flip-flops.
- Ωστόσο, το μοντέλο Moore πλεονεκτεί στο ότι οι αλλαγές των τιμών εξόδου είναι συγχρονισμένες με το σήμα του ρολογιού, αφού συμβαίνουν μόνο όταν αλλάζει η παρούσα κατάσταση των κυκλωμάτων.

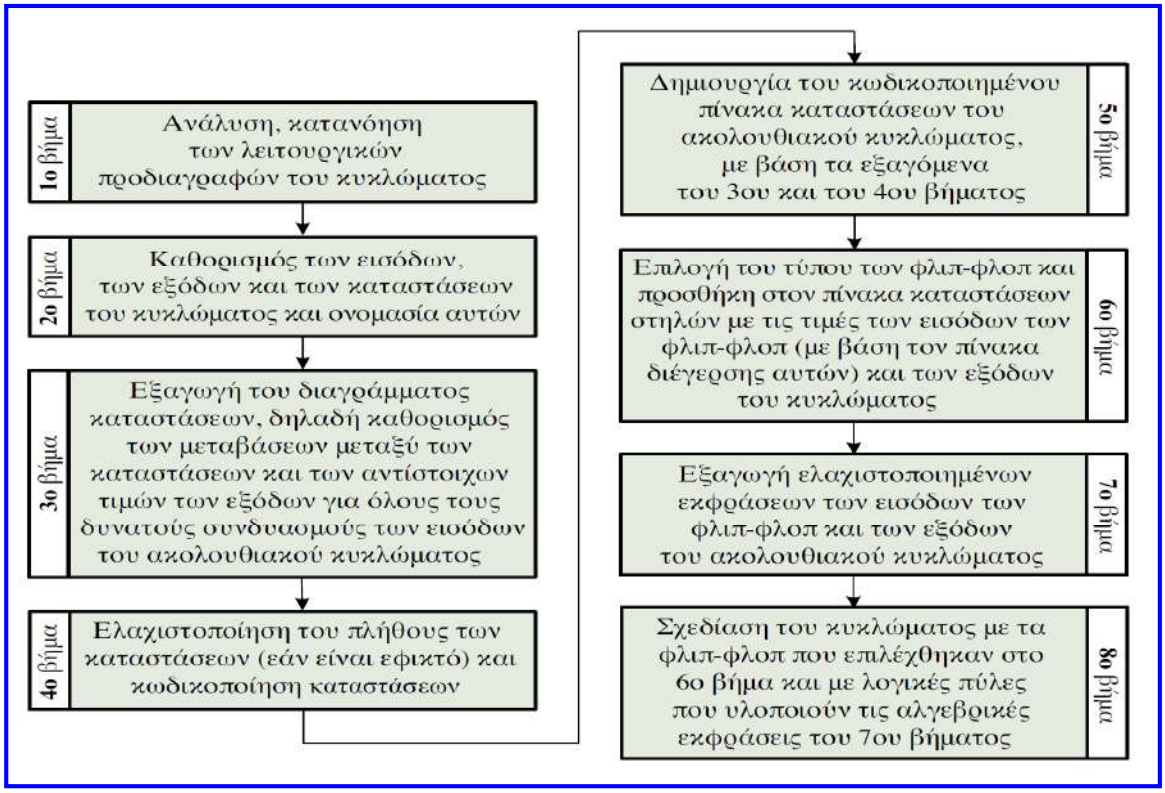
Περιγραφή μοντέλου Mealy





Περιγραφή μοντέλου Moore

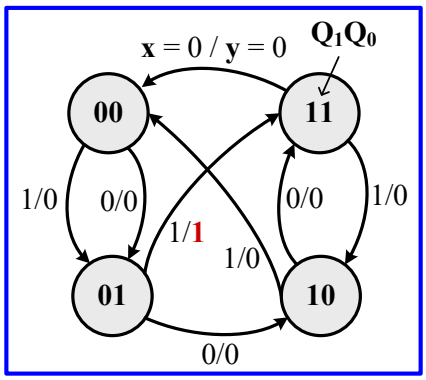
Σύνθεση (σχεδίαση) ΣΑΚ



Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 1ο

- Επιθυμούμε να συνθέσουμε ΣΑΚ με φλιπ-φλοπ D, πυροδοτούμενα στην ανερχόμενη ακμή ενός σήματος ρολογιού, το οποίο να διατρέχει επαναλαμβανόμενα τις καταστάσεις (τιμές) 0, 1, 2, 3 (κανονική δυαδική αρίθμηση) ή 0, 1, 3, 2 (αρίθμηση σύμφωνα με τον κώδικα Gray), ανάλογα με τον αν η είσοδος του κυκλώματος x έχει τιμή 0 ή 1, αντίστοιχα. Το κύκλωμα πρέπει να διαθέτει μία έξοδο y, η οποία να ενεργοποιείται (τιμή 1) όταν το κύκλωμα μεταβαίνει στην κατάσταση 3 για την αρίθμηση σύμφωνα με τον κώδικα Gray.
- Για την περιγραφή του ΣΑΚ επιλέγουμε το μοντέλο Mealy, λόγω του ότι η έξοδος του y εξαρτάται από την παρούσα κατάσταση και την είσοδο x του κυκλώματος.
- Για την δυαδική παράσταση των 4 καταστάσεων απαιτούνται 2 δυαδικά ψηφία, συνεπώς για την υλοποίηση του ΣΑΚ θα χρειαστούν 2 φλιπ-φλοπ.

Διάγραμμα καταστάσεων (μοντέλο Mealy):



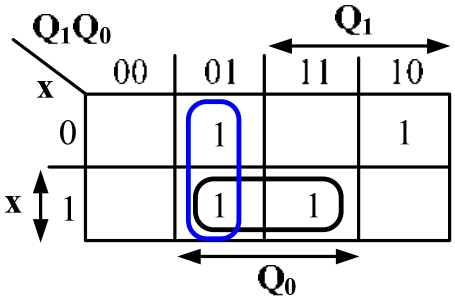
Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 1ο

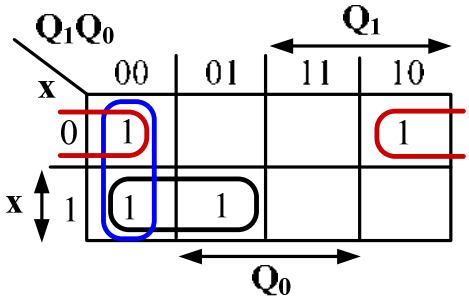
Κωδικοποιημένος πίνακας καταστάσεων [για το φλιπ-φλοπ D ισχύει ότι D = Q(t+1)]:

Είσοδος	Παρούσα κατάσταση		Επόμενη κατάσταση		Είσοδοι των φλιπ-φλοπ		Έξοδος
x	Q_1	Q_0	Q_1	Q_0	D_1	D_0	y
0	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0
0	1	0	1	1	1	1	0
0	1	1	0	0	0	0	0
1	0	0	0	1	0	1	0
1	0	1	1	1	1	1	1
1	1	0	0	0	0	0	0
1	1	1	1	0	1	0	0

Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 1ο

Ελαχιστοποιημένες εκφράσεις των εισόδων των φλιπ-φλοπ:





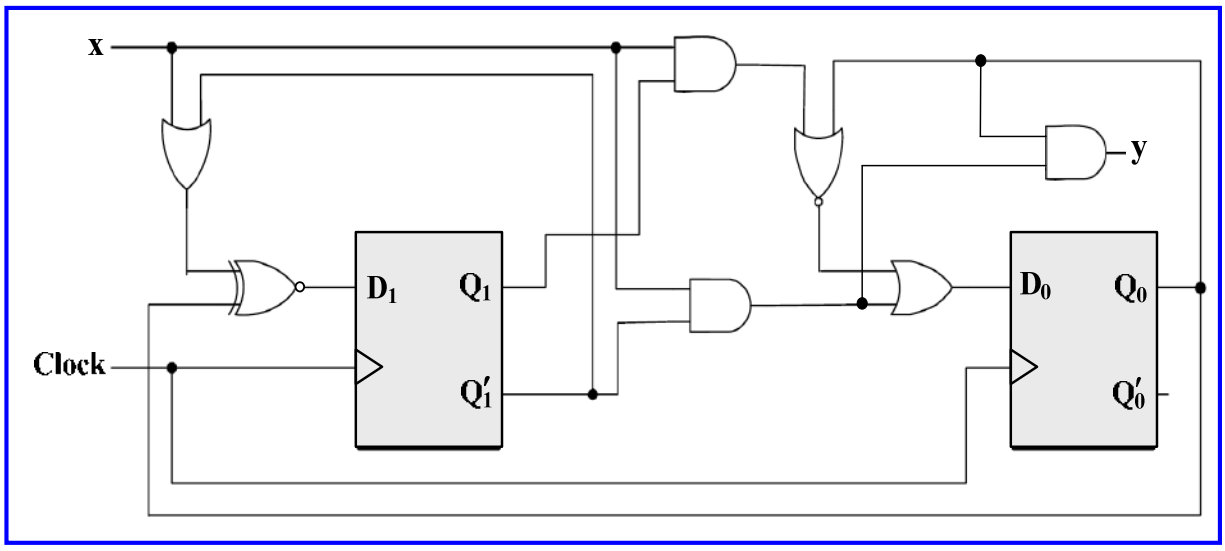
$$D_1 = Q'_1Q_0 + xQ_0 + x'Q_1Q'_0$$
$$= Q_0(Q'_1 + x) + Q'_0(Q'_1 + x)'$$
$$= [Q_0 \oplus (Q'_1 + x)]'$$

$$D_0 = Q'_1Q'_0 + x'Q'_0 + xQ'_1$$
$$= Q'_0(Q'_1 + x') + Q'_1x$$
$$= Q'_0(Q_1x)' + Q'_1x$$
$$= (Q_0 + Q_1x)' + Q'_1x$$

Λογική έκφραση της εξόδου: $y = xQ'_1Q_0$

Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 1ο

Λογικό διάγραμμα του ΣΑΚ με φλιπ-φλοπ D και λογικές πύλες:

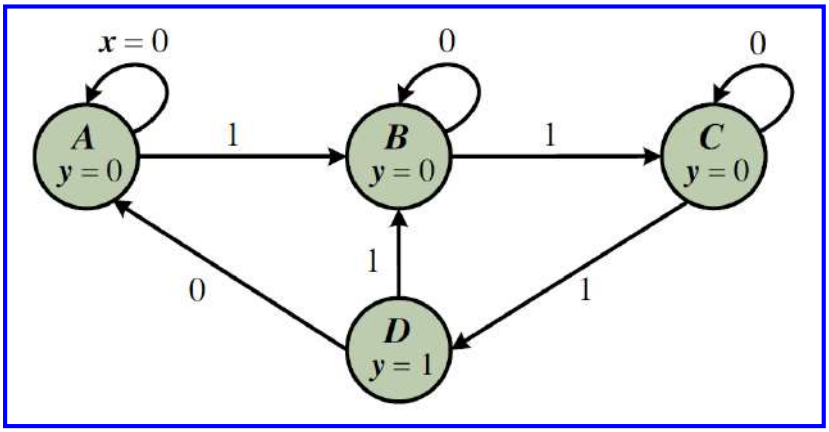


Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 2ο

- Επιθυμούμε να συνθέσουμε ΣΑΚ για τον έλεγχο λειτουργίας της μηχανής ενός ξενοδοχείου που παρέχει πορτοκαλάδα σε ποτήρι, κατά τη διάρκεια του πρωινού.
- Όταν ο αισθητήρας στάθμης της μηχανής ανιχνεύσει 3 φορές στη διάρκεια του γεύματος διαφορετική στάθμη πορτοκαλάδας από την προκαθορισμένη, θα πρέπει το ΣΑΚ να ενεργοποιήσει την εκτέλεση μιας σύντομης αυτόματης διαδικασίας ρύθμισης της στάθμης.
- Η διαδικασία αυτή θα πρέπει να ενεργοποιείται ξανά υπό την ίδια προϋπόθεση, δηλαδή ύστερα από 3 νέες ανιχνεύσεις διαφορετικής στάθμης, κατά τη διάρκεια του πρωινού.
- Για την περιγραφή της λειτουργίας σύμφωνα με το μοντέλο Moore, στο οποίο κάθε κατάσταση σχετίζεται με μία τιμή εξόδου, συμπεραίνουμε ότι οι καταστάσεις του κυκλώματος είναι 4: η 1η (A) αφορά τη μη ανίχνευση διαφορετικής στάθμης, η 2η (B) και η 3η (C) αφορούν μία και δύο ανιχνεύσεις διαφορετικής στάθμης, αντίστοιχα, στη διάρκεια του πρωινού και η 4η (D) αφορά 3 ανιχνεύσεις διαφορετικής στάθμης στη διάρκεια του πρωινού και την εκτέλεση ρύθμισης της στάθμης.
- Το κύκλωμα διαθέτει μία είσοδο x που λαμβάνει τιμή 1 κάθε φορά που ανιχνεύεται διαφορετική στάθμη από την προκαθορισμένη, και μία έξοδο y που λαμβάνει τιμή 1 όταν προκύπτουν 3 ανιχνεύσεις διαφορετικής στάθμης κατά τη διάρκεια του πρωινού, ώστε να λάβει χώρα η διαδικασία ρύθμισης.

Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 2ο

Διάγραμμα καταστάσεων (μοντέλο Moore)



Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 2ο

- Αφού το κύκλωμα διατρέχει 4 καταστάσεις, απαιτούνται 2 φλιπ-φλοπ.
- Στην περίπτωση περιγραφής του ΣΑΚ σύμφωνα με το μοντέλο Mealy, οι καταστάσεις του θα ήταν 3, αφού θα μπορούσαμε να συσχετίσουμε την κατάσταση C με δύο τιμές εξόδου και να αποφύγουμε τη χρήση της κατάστασης D.
- Ωστόσο, και σε αυτή την περίπτωση θα χρειαζόμασταν 2 φλιπ-φλοπ για την υλοποίηση του κυκλώματος.

Κατάσταση	Q_1	Q_0
A	0	0
B	0	1
C	1	0
D	1	1

Είσοδοι	Παρούσα κατάσταση		Επόμενη κατάσταση		Έξοδος
x	Q_1	Q_0	Q_1	Q_0	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	0	1	1

μοντέλο Moore

Κωδικοποίηση καταστάσεων και πίνακας καταστάσεων

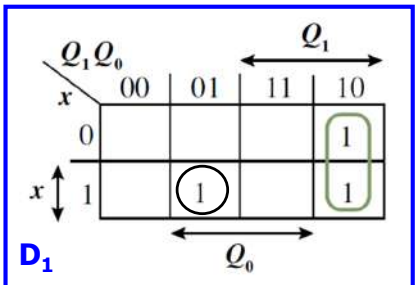
Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 2ο

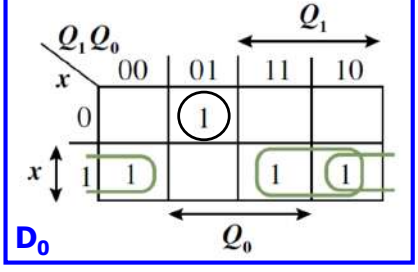
$Q_{t+1} = D$

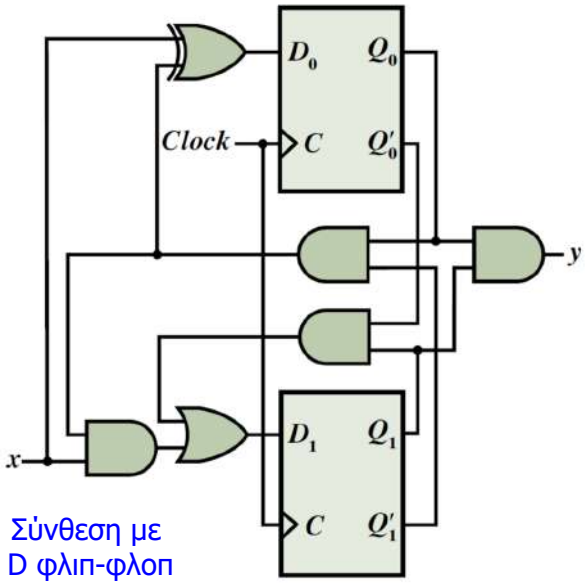
$$D_1 = xQ_1'Q_0 + Q_1Q_0'$$

$$D_0 = x'Q_1'Q_0 + xQ_1 + xQ_0' = x'Q_1'Q_0 + x(Q_1 + Q_0') = x'Q_1'Q_0 + x(Q_1'Q_0)' = x \oplus (Q_1'Q_0)$$

$$y = x'Q_1Q_0 + xQ_1Q_0 = (x' + x)Q_1Q_0 = Q_1Q_0$$







Σύνθεση με D φλιπ-φλοπ

Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 2ο

- Για τη σχεδίαση του ΣΑΚ με T ή JK flip-flops, θα πρέπει να εντάξουμε στον πίνακα καταστάσεων στήλες που αντιστοιχούν στις εισόδους των φλιπ-φλοπ.
- Οι τιμές των στηλών αυτών προκύπτουν από τις στήλες της παρούσας και της επόμενης κατάστασης, με χρήση των πινάκων διέγερσης των φλιπ-φλοπ T και JK.

Είσοδος	Παρούσα κατάσταση		Επόμενη κατάσταση		Είσοδοι των φλιπ-φλοπ						Έξοδος
x	Q_1	Q_0	Q_1	Q_0	T_1	T_0	J_1	K_1	J_0	K_0	y
0	0	0	0	0	0	0	0	×	0	×	0
0	0	1	0	1	0	0	0	×	×	0	0
0	1	0	1	0	0	0	×	0	0	×	0
0	1	1	0	0	1	1	×	1	×	1	1
1	0	0	0	1	0	1	0	×	1	×	0
1	0	1	1	0	1	1	1	×	×	1	0
1	1	0	1	1	0	1	×	0	1	×	0
1	1	1	0	1	1	0	×	1	×	0	1

Πίνακες διέγερσης των φλιπ-φλοπ

Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

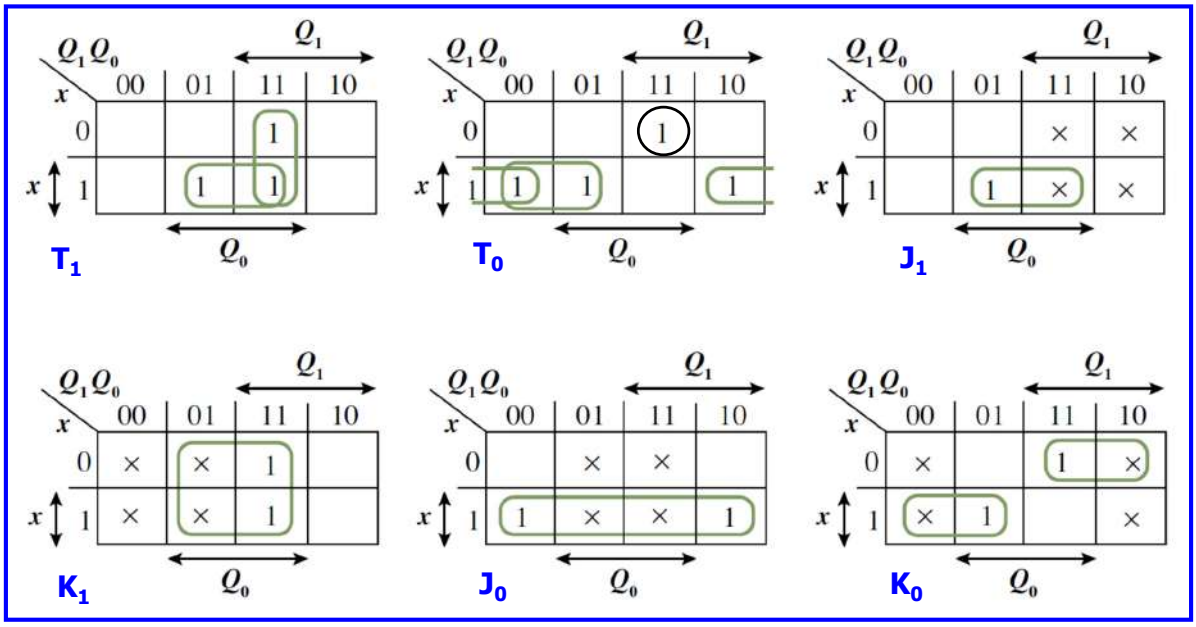
Q_t	Q_{t+1}	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 2ο

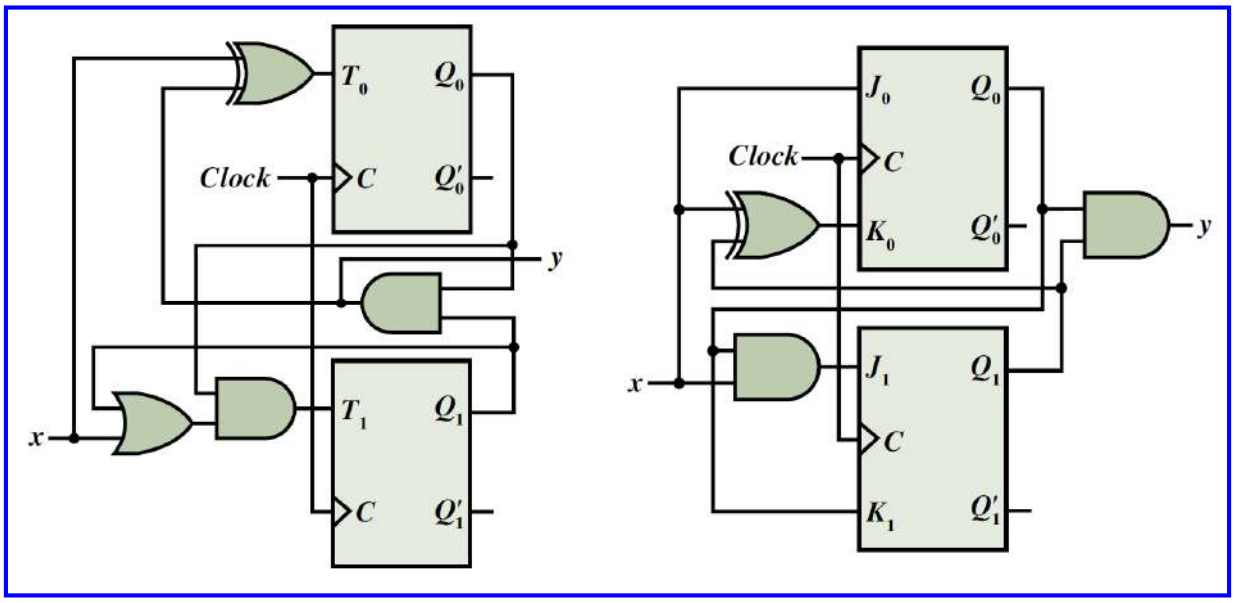
$$T_1 = xQ_0 + Q_1Q_0 = (x + Q_1)Q_0$$

$$T_0 = x'Q_1Q_0 + xQ_1' + xQ_0' = x'Q_1Q_0 + x(Q_1' + Q_0') = x'Q_1Q_0 + x(Q_1Q_0)' = x \oplus (Q_1Q_0)$$

$$J_1 = xQ_0, \quad K_1 = Q_0, \quad J_0 = x, \quad K_0 = xQ_1' + x'Q_1 = x \oplus Q_1$$



Σύνθεση (σχεδίαση) ΣΑΚ – Παράδειγμα 2ο



Η σύνθεση ΣΑΚ με χρήση φλιπ-φλοπ JK οδηγεί συνήθως σε απλούστερο συνδυαστικό τμήμα κυκλώματος, λόγω των αδιάφορων όρων που περιλαμβάνονται στους χάρτες Karnaugh των εισόδων των φλιπ-φλοπ

Μερικώς καθορισμένα ΣΑΚ

- Στο ΣΑΚ που υλοποιήσαμε για κάθε συνδυασμό κατάστασης και τιμής εισόδου καθορίζεται ένα ζεύγος επόμενης κατάστασης και τιμής εξόδου (πλήρως καθορισμένο κύκλωμα, completely specified circuit).

- Ωστόσο, υπάρχουν περιπτώσεις ΣΑΚ όπου, λόγω της φύσης των προδιαγραφών τους, κάποιες τιμές εισόδων δεν μπορούν να συμβούν, με αποτέλεσμα να υπάρχουν στοιχεία του πίνακα καταστάσεων που δεν καθορίζονται (μερικώς καθορισμένα κυκλώματα, incompletely specified circuits).

- Τα στοιχεία του πίνακα καταστάσεων των μερικώς καθορισμένων κυκλωμάτων (επόμενες καταστάσεις και τιμές εξόδων) που δεν καθορίζονται λόγω των μη επιτρεπτών τιμών εισόδων, αντιμετωπίζονται ως αδιάφοροι όροι.

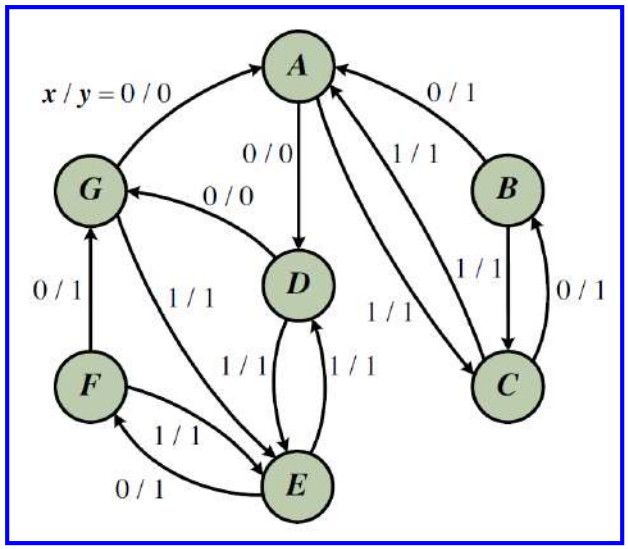
- Αυτό έχει αποτέλεσμα την εξαγωγή απλούστερων εκφράσεων προσδιορισμού των εισόδων των φλιπ-φλοπ και των εξόδων του κυκλώματος.

Ελαχιστοποίηση πλήθους καταστάσεων ΣΑΚ

- Η ελαχιστοποίηση του πλήθους των καταστάσεων ενός ΣΑΚ μπορεί να οδηγήσει σε μείωση του πλήθους των φλιπ-φλοπ που συμμετέχουν στο κύκλωμα, καθώς και σε μείωση της πολυπλοκότητας του συνδυαστικού τμήματος του κυκλώματος.
- Για να είναι εφικτή η μείωση των αρχικών καταστάσεων που διατρέχει ένα ΣΑΚ, θα πρέπει ορισμένες από αυτές να είναι μεταξύ τους ισοδύναμες.
- Δύο ή περισσότερες καταστάσεις είναι ισοδύναμες, όταν για κάθε δυνατό συνδυασμό τιμών των εισόδων παράγονται από αυτές οι ίδιες τιμές εξόδων και προκαλείται μετάβαση του κυκλώματος στην ίδια κατάσταση ή σε ισοδύναμες μεταξύ τους καταστάσεις.
- Μια συστηματική μέθοδος ελαχιστοποίησης του πλήθους των καταστάσεων ενός ΣΑΚ βασίζεται σε διαδοχικούς διαμερισμούς (successive partitions) του συνόλου των καταστάσεων του κυκλώματος.
- 1ος διαμερισμός καταστάσεων σε υποσύνολα, ώστε οι καταστάσεις κάθε υποσυνόλου να παράγουν τις ίδιες τιμές εξόδων για κάθε συνδυασμό τιμών των εισόδων.
- 2ος διαμερισμός καταστάσεων σε υποσύνολα, ώστε οι καταστάσεις κάθε υποσυνόλου να προκαλούν μετάβαση σε καταστάσεις που ανήκουν στο ίδιο υποσύνολο του 1ου διαμερισμού, για κάθε συνδυασμό τιμών των εισόδων.

Ελαχιστοποίηση πλήθους καταστάσεων ΣΑΚ

- Παρομοίως, ελέγχονται τα υποσύνολα καταστάσεων του 2ου διαμερισμού και η διαδικασία ολοκληρώνεται, όταν προκύψουν πανομοιότυπα υποσύνολα σε 2 διαδοχικούς διαμερισμούς.
- Τα υποσύνολα που τελικά προκύπτουν περιλαμβάνουν ισοδύναμες μεταξύ τους καταστάσεις, οι οποίες μπορούν να αντικατασταθούν από μία.



Παρούσα κατάσταση	Επόμενη κατάσταση / y	
	$x = 0$	$x = 1$
A	D / 0	C / 1
B	A / 1	C / 1
C	B / 1	A / 1
D	G / 0	E / 1
E	F / 1	D / 1
F	G / 1	E / 1
G	A / 0	E / 1

Ελαχιστοποίηση πλήθους καταστάσεων ΣΑΚ

1ος διαμερισμός: **{A, D, G}**, **{B, C, E, F}**

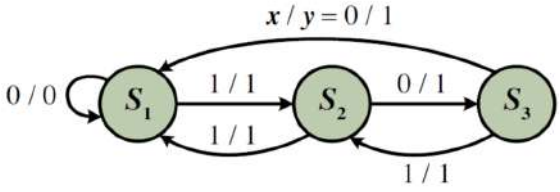
2ος διαμερισμός: **{A, D, G}**, **{C, E}**, **{B, F}**

3ος διαμερισμός: **{A, D, G}**, **{C, E}**, **{B, F}** ← υποσύνολα ισοδύναμων καταστάσεων

Παρούσα κατάσταση	Επόμενη κατάσταση / y	
	$x=0$	$x=1$
A	D / 0	C / 1
B	A / 1	C / 1
C	B / 1	A / 1
D	G / 0	E / 1
E	F / 1	D / 1
F	G / 1	E / 1
G	A / 0	E / 1

Παρούσα κατάσταση	Επόμενη κατάσταση / y	
	$x=0$	$x=1$
$\{A, D, G\} \rightarrow S_1$	S_1 / 0	S_2 / 1
$\{C, E\} \rightarrow S_2$	S_3 / 1	S_1 / 1
$\{B, F\} \rightarrow S_3$	S_1 / 1	S_2 / 1

Ελαχιστοποιημένος πίνακας καταστάσεων & ελαχιστοποιημένο διάγραμμα καταστάσεων



Κωδικοποίηση καταστάσεων ΣΑΚ

- Κωδικοποίηση καταστάσεων ενός ΣΑΚ είναι η αντιστοίχιση των καταστάσεών του με διαφορετικούς συνδυασμούς τιμών των μεταβλητών κατάστασης.
- Η κωδικοποίηση των καταστάσεων επηρεάζει την πολυπλοκότητα του συνδυαστικού κυκλώματος προσδιορισμού των εισόδων των φλιπ-φλοπ.
- Μια επιλογή κωδικοποίησης καταστάσεων, η οποία επιτρέπει καλύτερη ομαδοποίηση των ελαχιστόρων που συνιστούν τις λογικές εκφράσεις των εισόδων των φλιπ-φλοπ, οδηγεί σε απλούστερο κύκλωμα και επομένως σε υλοποίηση μικρότερου κόστους.
- Η αύξηση του πλήθους των καταστάσεων ενός κυκλώματος, οδηγεί σε ραγδαία αύξηση του πλήθους των επιλογών κωδικοποίησής τους.
- Για κύκλωμα με n καταστάσεις απαιτούνται $k = [\log_2 n]$ μεταβλητές κατάστασης και το πλήθος των συνδυασμών τιμών των μεταβλητών κατάστασης που είναι διαθέσιμοι για την κωδικοποίηση των n καταστάσεων είναι 2^k συνδυασμοί.
- Αποδεικνύεται ότι σε κυκλώματα 4, 5 και 8 καταστάσεων, το πλήθος των επιλογών κωδικοποίησης είναι 24, 6720 και 40320, αντίστοιχα.
- Ωστόσο, αποδεικνύεται επίσης ότι όλες αυτές οι επιλογές δεν είναι μοναδικές, όσον αφορά τις εκφράσεις προσδιορισμού των εισόδων των φλιπ-φλοπ στις οποίες οδηγούν.

Κωδικοποίηση καταστάσεων ΣΑΚ

- Αυτό συμβαίνει διότι οι επιλογές κωδικοποίησης που προκύπτουν από άλλες με αντιμετάθεση των μεταβλητών κατάστασης οδηγούν σε εκφράσεις προσδιορισμού των εισόδων των φλιπ-φλοπ με όμοια πολυπλοκότητα.
- Το ίδιο συμβαίνει και με τις επιλογές κωδικοποίησης που προκύπτουν από άλλες, με αντιστροφή μιας τουλάχιστον μεταβλητής κατάστασης, αφού οι έξοδοι των φλιπ-φλοπ είναι διαθέσιμες στην κανονική και στη συμπληρωματική τους μορφή.
- Συνεπώς, το πλήθος των μοναδικών επιλογών κωδικοποίησης είναι αισθητά μικρότερο από εκείνο των διαθέσιμων επιλογών κωδικοποίησης (για 4 καταστάσεις υπάρχουν 3 μοναδικές επιλογές, για 5 καταστάσεις 140 και για 8 καταστάσεις 840).
- Η επιλογή της βέλτιστης κωδικοποίησης βασίζεται στην τήρηση 2 κανόνων.
- 1ος κανόνας: οι καταστάσεις με ίδια επόμενη κατάσταση για έναν τουλάχιστον συνδυασμό τιμών των εισόδων, κωδικοποιούνται με συνδυασμούς των μεταβλητών κατάστασης που διαφέρουν μεταξύ τους στην τιμή μιας μόνο μεταβλητής.
- 2ος κανόνας: οι καταστάσεις που αποτελούν επόμενη κατάσταση της ίδιας κατάστασης για διαδοχικούς συνδυασμούς τιμών των εισόδων, κωδικοποιούνται με συνδυασμούς των μεταβλητών κατάστασης που διαφέρουν μεταξύ τους στην τιμή μιας μόνο μεταβλητής.
- Εάν υπάρχει σύγκρουση κατά την τήρηση των δύο κανόνων, τότε έχει προτεραιότητα η τήρηση του πρώτου κανόνα.

Κωδικοποίηση καταστάσεων ΣΑΚ

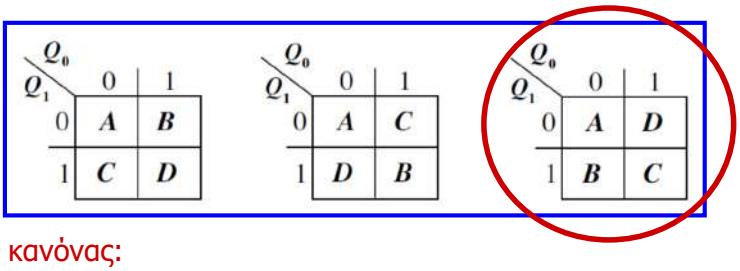
- Για τον εντοπισμό των 3 επιλογών μοναδικής κωδικοποίησης 4 καταστάσεων, από το σύνολο των 24 επιλογών, ξεκινάμε από μία επιλογή κωδικοποίησης, επιλέγουμε μια ακόμη κωδικοποίηση, η οποία δεν μπορεί να παραχθεί από την 1η με αντιμετάθεση των μεταβλητών κατάστασης ή με αντιστροφή μίας τουλάχιστον μεταβλητής κατάστασης και τέλος επιλέγουμε μια 3η κωδικοποίηση, η οποία δεν μπορεί να παραχθεί από τις 2 πρώτες με αντιμετάθεση των μεταβλητών κατάστασης ή με αντιστροφή 1 ή 2 μεταβλητών.
- Οι υπόλοιπες 21 κωδικοποιήσεις καταστάσεων μπορούν να παραχθούν (ανά 7) από τις 3 μοναδικές κωδικοποιήσεις, με αντιμετάθεση των μεταβλητών κατάστασης ή με αντιστροφή 1 ή 2 μεταβλητών κατάστασης.

Μοναδικές επιλογές κωδικοποίησης 4 καταστάσεων →

Κατάσταση	Q_1	Q_0	Q_1	Q_0	Q_1	Q_0
A	0	0	0	0	0	0
B	0	1	1	1	1	0
C	1	0	0	1	1	1
D	1	1	1	0	0	1

Καταστάσεις	Αντιμετάθεση μεταβλητών στην αρχική κωδικοποίηση		Αντιστροφή μίας ή δύο μεταβλητών στην αρχική κωδικοποίηση						Αντιστροφή μίας ή δύο μεταβλητών στην κωδικοποίηση που προέκυψε με αντιμετάθεση των μεταβλητών					
	Q_1	Q_0	Q_1	Q_0	Q_1	Q_0	Q_1	Q_0	Q_1	Q_0	Q_1	Q_0	Q_1	Q_0
A	0	0	1	0	0	1	1	1	1	0	0	1	1	1
B	1	0	1	1	0	0	1	0	0	0	1	1	0	1
C	0	1	0	0	1	1	0	1	1	1	0	0	1	0
D	1	1	0	1	1	0	0	0	0	1	1	0	0	0

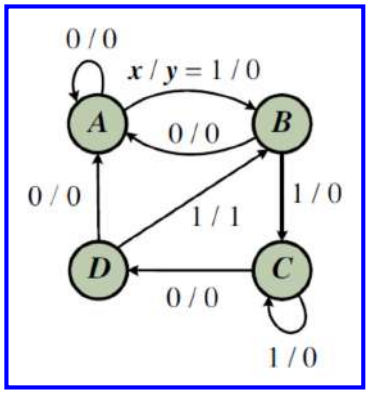
Κωδικοποίηση καταστάσεων ΣΑΚ



Q_1 \ Q_0	0	1
0	A	B
1	C	D

Q_1 \ Q_0	0	1
0	A	C
1	D	B

Q_1 \ Q_0	0	1
0	A	D
1	B	C



1ος κανόνας:

επόμενη κατάσταση των A, B για x = 0 είναι η ίδια κατάσταση A
επόμενη κατάσταση των A, D για x = 0 είναι η ίδια κατάσταση A
επόμενη κατάσταση των A, D για x = 1 είναι η ίδια κατάσταση B
επόμενη κατάσταση των B, D για x = 0 είναι η ίδια κατάσταση A
επόμενη κατάσταση των B, C για x = 1 είναι η ίδια κατάσταση C

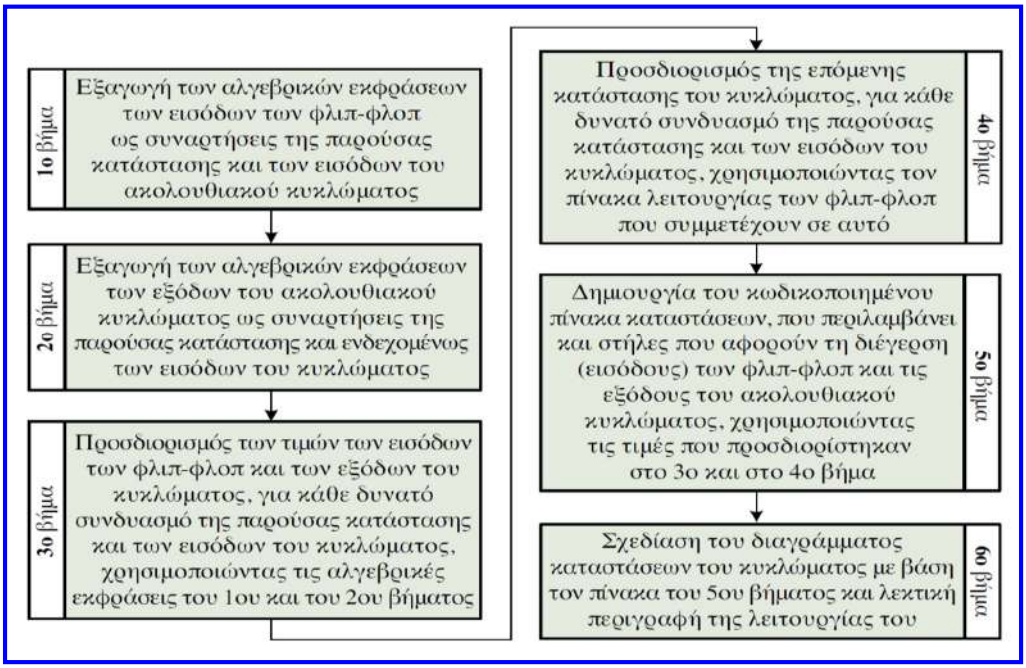
2ος κανόνας:

οι καταστάσεις A, B είναι επόμενες της ίδιας κατάστασης D για x = 0, 1, αντίστοιχα
οι καταστάσεις A, C είναι επόμενες της ίδιας κατάστασης B για x = 0, 1, αντίστοιχα
οι καταστάσεις C, D είναι επόμενες της ίδιας κατάστασης C για x = 1, 0, αντίστοιχα

Η 3η επιλογή κωδικοποιεί με γειτονικούς συνδυασμούς τα ζεύγη (A, B), (A, D), (B, C), (C, D) από τα 6 ζεύγη που ικανοποιούν τουλάχιστον έναν από τους κανόνες. Οι άλλες 2 επιλογές κωδικοποιούν με γειτονικούς συνδυασμούς επίσης 4 από τα 6 ζεύγη, χωρίς όμως να ανήκουν σε αυτά και τα 2 ζεύγη (A, B), (A, D) που εμφανίζονται 2 φορές ⇒ 3η = βέλτιστη επιλογή.

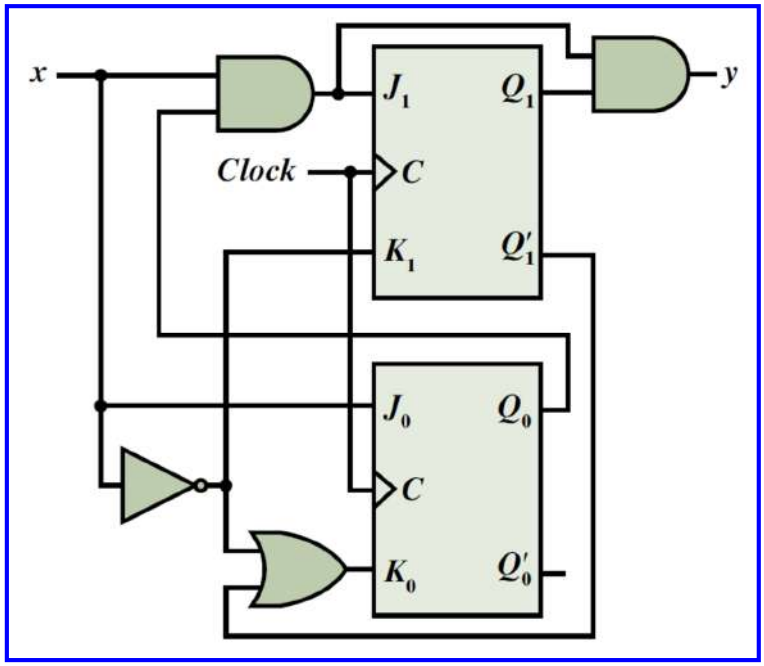
Ανάλυση ΣΑΚ

Ανάλυση ενός σύγχρονου ακολουθιακού κυκλώματος είναι ο προσδιορισμός της λειτουργίας ή των λειτουργιών που εκτελεί, ο οποίος συνίσταται στην εξαγωγή του πίνακα ή του διαγράμματος καταστάσεων ή ακόμη και μιας λεκτικής περιγραφής



Ανάλυση ΣΑΚ - Παράδειγμα

Στην είσοδό του x το διπλανό ΣΑΚ λαμβάνει μία ακολουθία δυαδικών ψηφίων σε συγχρονισμό με το σήμα ρολογιού



$$J_1 = xQ_0, \qquad K_1 = x', \qquad J_0 = x, \qquad K_0 = x' + Q'_1, \qquad y = xQ_1Q_0$$

Ανάλυση ΣΑΚ - Παράδειγμα

Υπολογίζουμε τις τιμές των εισόδων των φλιπ-φλοπ με βάση τις εκφράσεις τους για κάθε συνδυασμό τιμών εισόδου και παρούσας κατάστασης και στη συνέχεια, με βάση τον πίνακα λειτουργίας του φλιπ-φλοπ JK, προσδιορίζουμε τις τιμές των μεταβλητών της επόμενης κατάστασης του κυκλώματος, έτσι ώστε να συμπληρώσουμε τις αντίστοιχες στήλες του πίνακα καταστάσεων:

Είσοδος	Παρούσα κατάσταση		Επόμενη κατάσταση		Είσοδοι των φλιπ-φλοπ				Έξοδος
x	Q_1	Q_0	Q_1	Q_0	J_1	K_1	J_0	K_0	y
0	0	0	0	0	0	1	0	1	0
0	0	1	0	0	0	1	0	1	0
0	1	0	0	0	0	1	0	1	0
0	1	1	0	0	0	1	0	1	0
1	0	0	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1	1	0
1	1	0	1	1	0	0	1	0	0
1	1	1	1	1	1	0	1	0	1

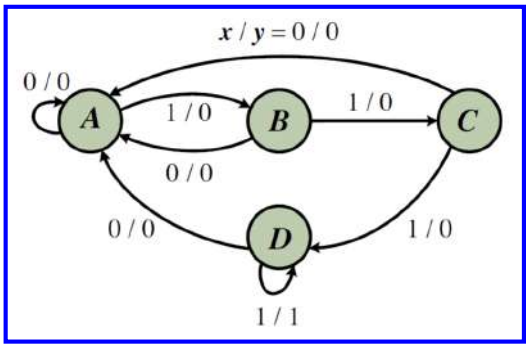
Πίνακας λειτουργίας φλιπ-φλοπ JK

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t

Ανάλυση ΣΑΚ - Παράδειγμα

- Κωδικοποίηση καταστάσεων: Q_1Q_0 = 00 (A), 01 (B), 10 (C), 11 (D).

Διάγραμμα καταστάσεων



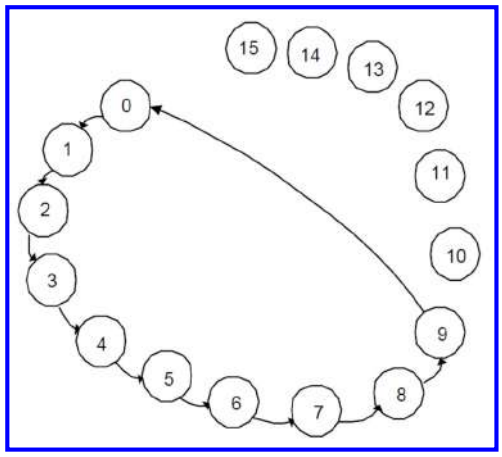
- Διαπιστώνουμε ότι το κύκλωμα, ξεκινώντας από την κατάσταση A, διατρέχει κατά σειρά τις καταστάσεις B, C και D, όταν η ακολουθία των δυαδικών ψηφίων που λαμβάνεται στην είσοδο x αποτελείται από 3 μονάδες.
- Στις περιπτώσεις που υπάρχει παρεμβολή ψηφίου εισόδου με τιμή 0, το κύκλωμα επιστρέφει στην κατάσταση A.
- Η έξοδος y λαμβάνει τιμή 1, μόνο όταν το κύκλωμα βρίσκεται στην κατάσταση D και ληφθεί μία ακόμη μονάδα στην είσοδο. Τότε το κύκλωμα παραμένει στην κατάσταση D.
- Συνεπώς, το κύκλωμα ενεργοποιεί την έξοδό του, όταν λαμβάνει στην είσοδο την ακολουθία 1111 και επικαλυπτόμενες εκδοχές αυτής (ανιχνευτής ακολουθίας 1111).

ΣΑΚ με αχρησιμοποίητες καταστάσεις

- Η σχέση που συνδέει το πλήθος των καταστάσεων (s) ενός ΣΑΚ με το πλήθος των μεταβλητών κατάστασης (k) είναι $2^{k-1} < s \leq 2^k$.
- Όταν $s < 2^k$, τότε οι μεταβλητές κατάστασης εκφράζουν περισσότερες καταστάσεις από εκείνες που καθορίζονται με βάση τις προδιαγραφές του κυκλώματος, με αποτέλεσμα να υπάρχουν $(2^k - s)$ αχρησιμοποίητες καταστάσεις (unused states).
- Για παράδειγμα, για ένα κύκλωμα 5 καταστάσεων χρησιμοποιούνται 3 φλιπ-φλοπ, δηλαδή υπάρχουν 3 μεταβλητές κατάστασης που εκφράζουν 8 διαφορετικές καταστάσεις, με αποτέλεσμα 3 καταστάσεις να μη χρησιμοποιούνται.
- Μία τεχνική είναι να αντιμετωπίσουμε ως αδιάφορους όρους τις επόμενες καταστάσεις και πιθανώς τις τιμές των εξόδων που αντιστοιχούν στις αχρησιμοποίητες καταστάσεις.
- Στην περίπτωση αυτή, όμως, μετά τη σύνθεση θα πρέπει το κύκλωμα να αναλυθεί, για να διαπιστωθεί εάν όταν βρεθεί σε μια από τις αχρησιμοποίητες καταστάσεις (λόγω αστοχίας υλικού ή απρόσμενης εισόδου), επιστρέφει σε κάποια από τις έγκυρες καταστάσεις.
- Τότε πρόκειται για κύκλωμα με αυτόματη διόρθωση ή αυτοδιορθούμενο κύκλωμα (self-correcting circuit).
- Εναλλακτικά, μπορούμε, κατά τη σύνθεση του κυκλώματος, να καθορίσουμε μετάβαση από τις αχρησιμοποίητες καταστάσεις σε κάποια ή κάποιες από τις έγκυρες καταστάσεις.
- Ο πρώτος τρόπος πλεονεκτεί, αφού οδηγεί σε απλούστερο συνδυαστικό τμήμα.

ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 1ο

- Επιθυμούμε να συνθέσουμε ΣΑΚ με JK flip-flops πυροδοτούμενα στην κατερχόμενη ακμή ενός σήματος ρολογιού, το οποίο να διατρέχει επαναλαμβανόμενα τις καταστάσεις (τιμές) από 0 έως 9 σε δυαδική παράσταση.
- Για την δυαδική παράσταση των 10 καταστάσεων απαιτούνται 4 δυαδικά ψηφία, συνεπώς για την υλοποίηση του ΣΑΚ απαιτούνται 4 flip-flops.
- Οι καταστάσεις 10, 11, 12, 13, 14 και 15 δεν χρησιμοποιούνται και κατά τη σύνθεση του ΣΑΚ μπορούμε να αντιμετωπίσουμε ως αδιάφορους όρους τις επόμενες καταστάσεις που αντιστοιχούν στις αχρησιμοποίητες αυτές καταστάσεις.
- Διάγραμμα καταστάσεων:



ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 1ο

Παρούσα κατάσταση				Επόμενη κατάσταση				Είσοδοι φλιπ-φλοπ			
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	J_3K_3	J_2K_2	J_1K_1	J_0K_0
0	0	0	0	0	0	0	1	0X	0X	0X	1X
0	0	0	1	0	0	1	0	0X	0X	1X	X1
0	0	1	0	0	0	1	1	0X	0X	X0	1X
0	0	1	1	0	1	0	0	0X	1X	X1	X1
0	1	0	0	0	1	0	1	0X	X0	0X	1X
0	1	0	1	0	1	1	0	0X	X0	1X	X1
0	1	1	0	0	1	1	1	0X	X0	X0	1X
0	1	1	1	1	0	0	0	1X	X1	X1	X1
1	0	0	0	1	0	0	1	X0	0X	0X	1X
1	0	0	1	0	0	0	0	X1	0X	0X	X1
1	0	1	0	X	X	X	X	XX	XX	XX	XX
1	0	1	1	X	X	X	X	XX	XX	XX	XX
1	1	0	0	X	X	X	X	XX	XX	XX	XX
1	1	0	1	X	X	X	X	XX	XX	XX	XX
1	1	1	0	X	X	X	X	XX	XX	XX	XX
1	1	1	1	X	X	X	X	XX	XX	XX	XX

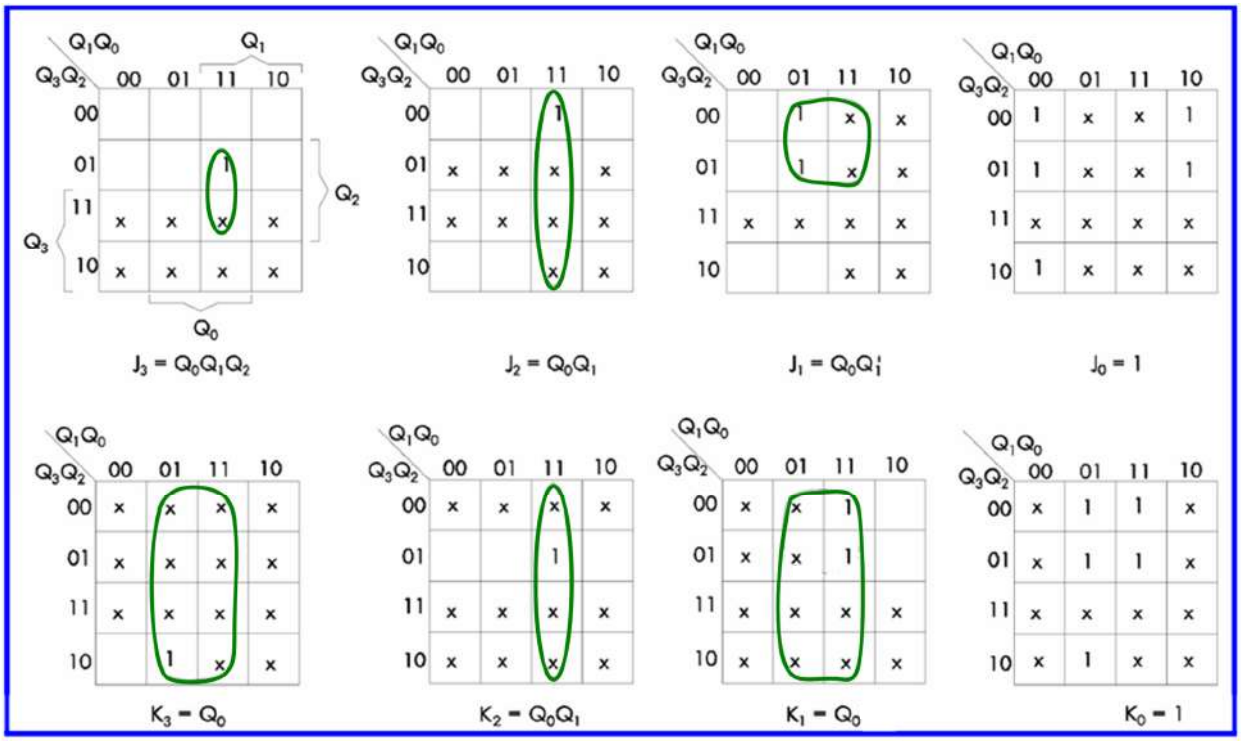
Πίνακας καταστάσεων

Πίνακας διέγερσης φλιπ-φλοπ JK

Q_t	Q_{t+1}	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

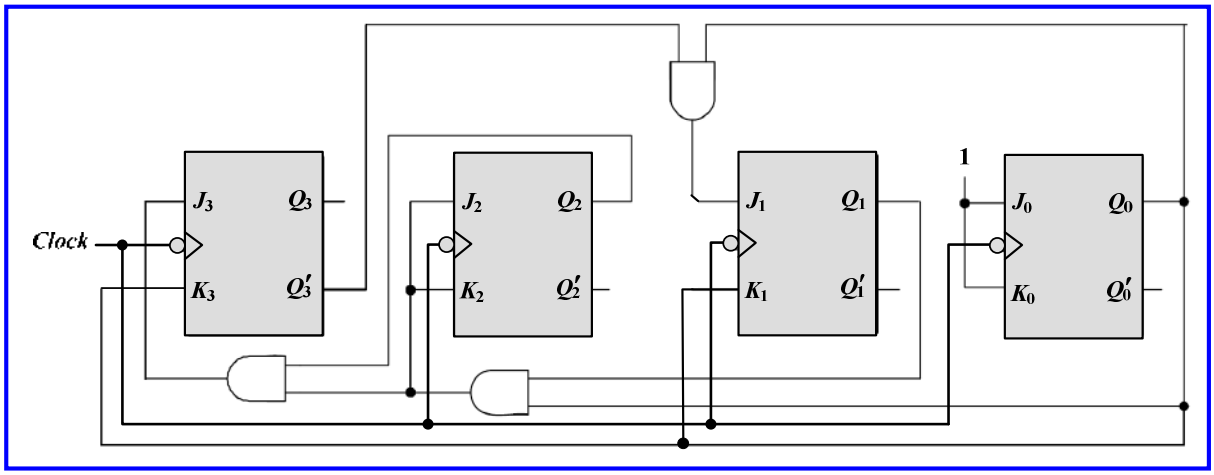
ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 1ο

Προσδιορισμός των συναρτήσεων εισόδων των φλιπ-φλοπ



ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 1ο

$J_3 = Q_2Q_1Q_0 \quad J_2 = K_2 = Q_1Q_0 \quad J_1 = Q'_3Q_0 \quad J_0 = K_0 = 1 \quad K_1 = K_3 = Q_0$



ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 1ο

Ανάλυση του ΣΑΚ που σχεδιάστηκε, για να διαπιστωθεί εάν όταν βρεθεί σε μια από τις αχρησιμοποίητες καταστάσεις επιστρέφει σε κάποια από τις έγκυρες καταστάσεις ή εάν εγκλωβίζεται σε κάποια αχρησιμοποίητη κατάσταση.

$J_3 = Q_2Q_1Q_0 \quad J_2 = K_2 = Q_1Q_0 \quad J_1 = Q'_3Q_0 \quad J_0 = K_0 = 1 \quad K_1 = K_3 = Q_0$

Παρούσα κατάσταση				Είσοδοι φλιπ-φλοπ				Επόμενη κατάσταση			
Q_3	Q_2	Q_1	Q_0	J_3 K_3	J_2 K_2	J_1 K_1	J_0 K_0	Q_3	Q_2	Q_1	Q_0
1	0	1	0	0 0	0 0	0 0	1 1	1	0	1	1
1	0	1	1	0 1	1 1	0 1	1 1	0	1	0	0
1	1	0	0	0 0	0 0	0 0	1 1	1	1	0	1
1	1	0	1	0 1	0 0	0 1	1 1	0	1	0	0
1	1	1	0	0 0	0 0	0 0	1 1	1	1	1	1
1	1	1	1	1 1	1 1	0 1	1 1	0	0	0	0

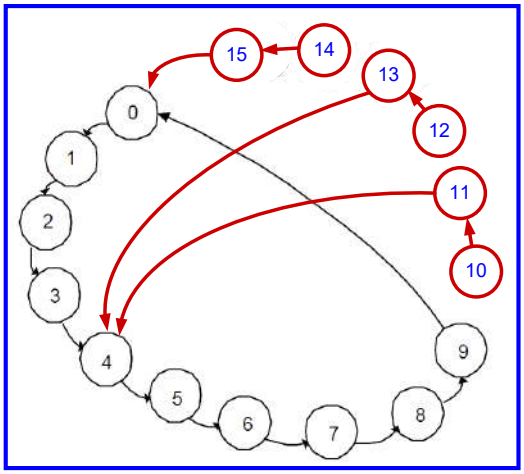
Αχρησιμοποίητες καταστάσεις

Πίνακας λειτουργίας φλιπ-φλοπ JK

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t

ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 1ο

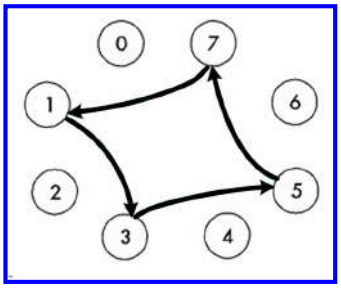
Διάγραμμα καταστάσεων που περιλαμβάνει και τις αχρησιμοποίητες καταστάσεις

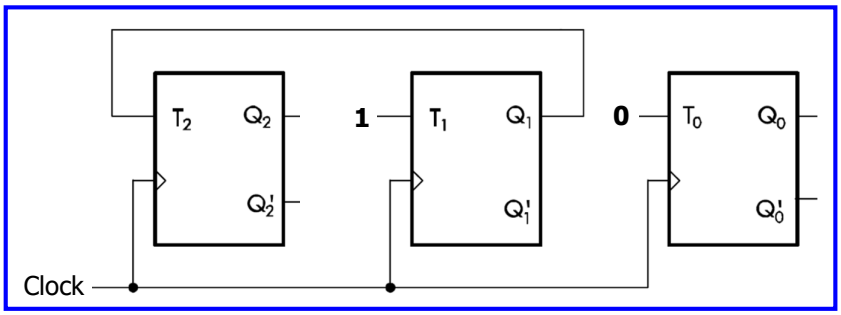


- Από την ανάλυση του κυκλώματος διαπιστώνουμε ότι εάν το ΣΑΚ βρεθεί σε μία από τις αχρησιμοποίητες καταστάσεις, το αργότερο μετά από 2 παλμούς ρολογιού, επιστρέφει σε μία από τις έγκυρες καταστάσεις (κύκλωμα με αυτόματη διόρθωση).
- Στην περίπτωση που το κύκλωμα εγκλωβιζόταν σε μία ή περισσότερες καταστάσεις, τότε θα έπρεπε να επανασχεδιάσουμε το διάγραμμα καταστάσεων έτσι ώστε το ΣΑΚ να μεταβαίνει από τις αχρησιμοποίητες καταστάσεις που εγκλωβίζεται σε έγκυρη κατάσταση και στη συνέχεια να επαναλάβουμε την διαδικασία σύνθεσης του ΣΑΚ.

ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 2ο

- Έστω ότι έχουμε συνθέσει ΣΑΚ με 3 φλιπ-φλοπ Τ, το οποίο διατρέχει επαναλαμβανόμενα τις καταστάσεις 1, 3, 5 και 7 σε δυαδική παράσταση, με την μεθοδολογία που παρουσιάστηκε προηγουμένως, δηλαδή αντιμετωπίζοντας ως αδιάφορους όρους τις επόμενες καταστάσεις που αντιστοιχούν στις καταστάσεις 0, 2, 4, και 6 που δεν χρησιμοποιούνται.





ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 2ο

- Μετά τη σύνθεση του ΣΑΚ θα πρέπει να το αναλύουμε (σε ότι αφορά τις μη έγκυρες / αχρησιμοποίητες καταστάσεις) για να διαπιστώσουμε εάν αυτό διαθέτει δυνατότητα αυτόματης διόρθωσης ή εάν εγκλωβίζεται σε μία ή περισσότερες μη έγκυρες καταστάσεις.

$T_2=Q_1 \quad T_1=1 \quad T_0=0$

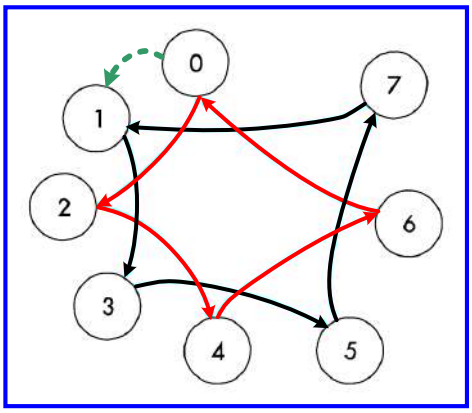
Πίνακας λειτουργίας φλιπ-φλοπ Τ

T	Q_{t+1}
0	Q_t
1	Q_t'

Παρούσα κατάσταση			Είσοδοι φλιπ-φλοπ			Επόμενη κατάσταση		
Q_2	Q_1	Q_0	T_2	T_1	T_0	Q_2	Q_1	Q_0
0	0	0	0	1	0	0	1	0
0	0	1	0	1	0	0	1	1
0	1	0	1	1	0	1	0	0
0	1	1	1	1	0	1	0	1
1	0	0	0	1	0	1	1	0
1	0	1	0	1	0	1	1	1
1	1	0	1	1	0	0	0	0
1	1	1	1	1	0	0	0	1

ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 2ο

- Από το διάγραμμα καταστάσεων διαπιστώνουμε ότι εάν το ΣΑΚ βρεθεί σε μία από τις μη έγκυρες καταστάσεις, τότε εγκλωβίζεται στον βρόχο μη έγκυρων καταστάσεων 0, 2, 4, 6, 0, ...
- Μια λύση στο πρόβλημα αυτό είναι να παρέμβουμε στο διάγραμμα καταστάσεων, αναγκάζοντας το κύκλωμα να μεταβεί από την κατάσταση 0 στην κατάσταση 1, έτσι ώστε να «σπάσει» ο βρόχος μη έγκυρων καταστάσεων και το ΣΑΚ να επιστρέφει σε έναν ή περισσότερους παλμούς ρολογιού στην κανονική ακολουθία καταστάσεων.

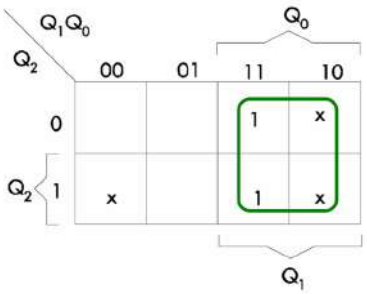


Πίνακας διέγερσης φλιπ-φλοπ T

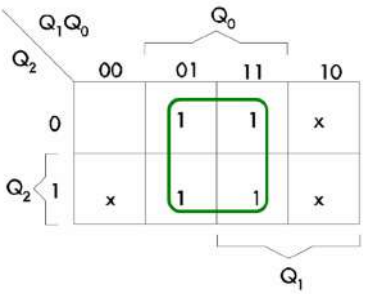
Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Παρούσα κατάσταση			Επόμενη κατάσταση			Είσοδοι φλιπ-φλοπ		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	X	X	X	X	X	X
0	1	1	1	0	1	1	1	0
1	0	0	X	X	X	X	X	X
1	0	1	1	1	1	0	1	0
1	1	0	X	X	X	X	X	X
1	1	1	0	0	1	1	1	0

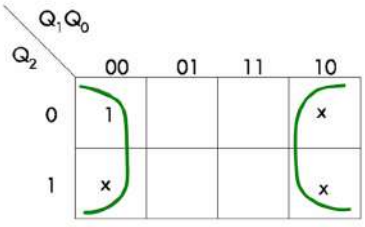
ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 2ο



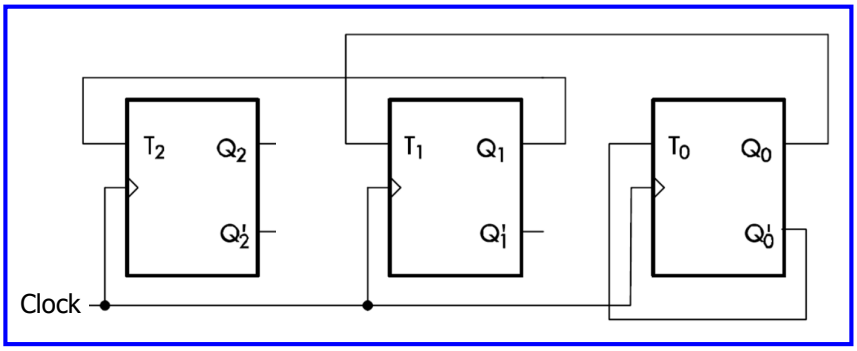
$T_2 = Q_1$



$T_1 = Q_0$



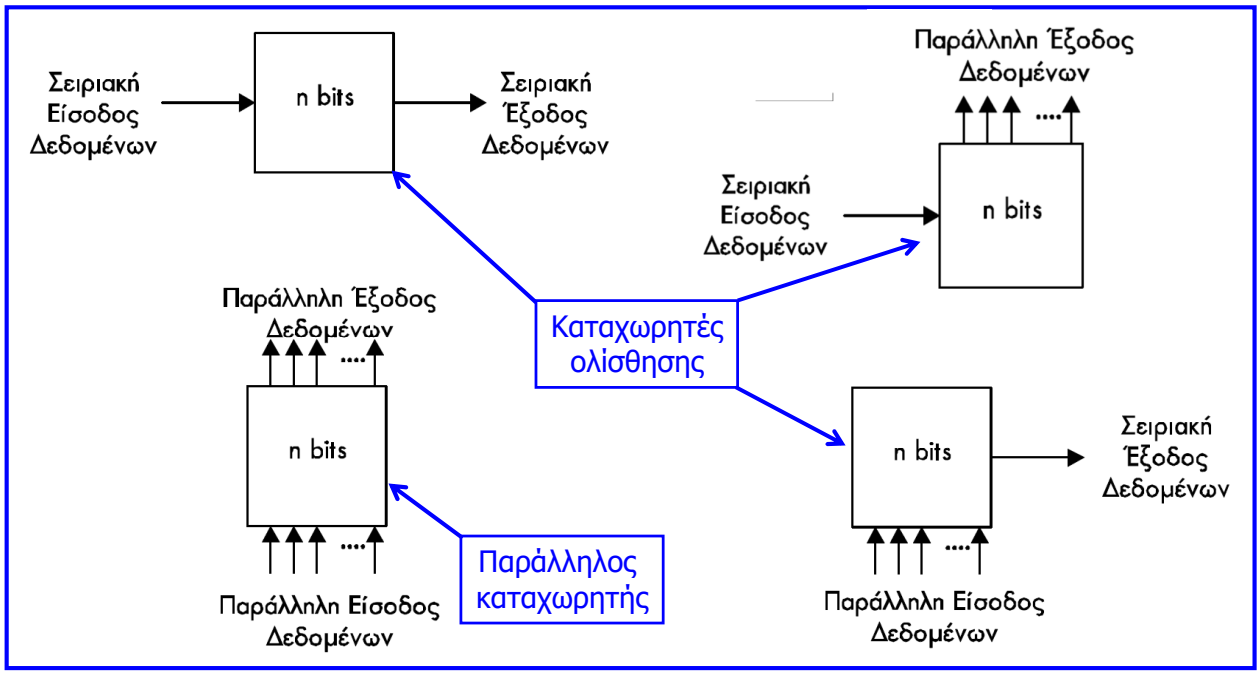
$T_0 = Q'_0$



Καταχωρητές

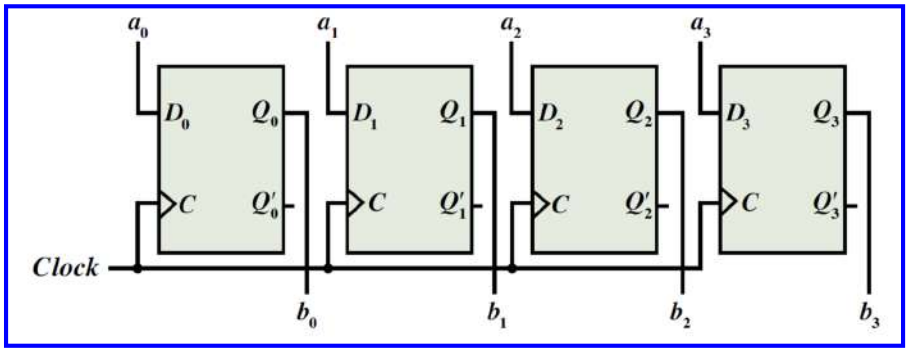
- Οι καταχωρητές (registers) είναι σύγχρονα ακολουθιακά κυκλώματα τα οποία χρησιμοποιούνται στα ψηφιακά συστήματα για την προσωρινή αποθήκευση δυαδικών δεδομένων, ώστε να διευκολυνθεί η επεξεργασία τους.
- Περιλαμβάνουν φλιπ-φλοπ που συνήθως συνοδεύεται από ένα συνδυαστικό τμήμα.
- Με δεδομένο ότι ένα φλιπ-φλοπ μπορεί να αποθηκεύσει πληροφορία ενός δυαδικού ψηφίου, προκύπτει εύκολα ότι ένας καταχωρητής θα πρέπει να περιλαμβάνει τόσα φλιπ-φλοπ, όσα είναι τα δυαδικά ψηφία που πρόκειται να αποθηκευτούν σε αυτόν.
- Η φόρτωση (loading) ή είσοδος των δεδομένων σε έναν καταχωρητή γίνεται παράλληλα (ταυτόχρονη φόρτωση όλων των δυαδικών ψηφίων σε έναν παλμό σήματος ρολογιού) ή σειριακά (δηλ. φόρτωση ενός δυαδικού ψηφίου σε κάθε παλμό του σήματος ρολογιού).
- Παρομοίως γίνεται η ανάγνωση (reading) ή έξοδος των δυαδικών δεδομένων.
- Οι καταχωρητές με παράλληλη φόρτωση (είσοδο) και ανάγνωση (έξοδο) των δυαδικών δεδομένων αναφέρονται ως παράλληλοι καταχωρητές (parallel registers)
- Οι καταχωρητές με σειριακή φόρτωση ή/και ανάγνωση των δεδομένων, αναφέρονται ως καταχωρητές ολίσθησης (shift registers), αφού η δυαδική πληροφορία ολισθαίνει κατά μήκος της αλυσίδας των φλιπ-φλοπ των καταχωρητών αυτών.

Καταχωρητές



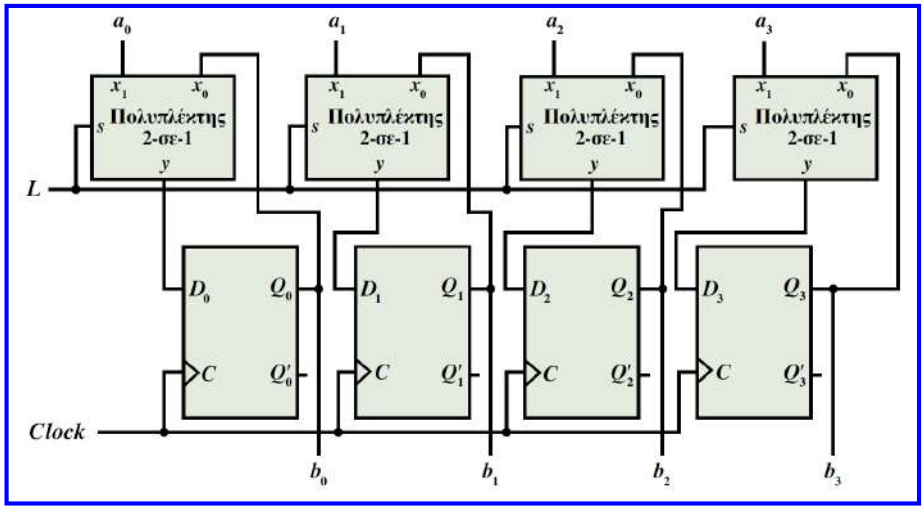
Απλός παράλληλος καταχωρητής

- Ο παράλληλος καταχωρητής του σχήματος έχει τη δυνατότητα αποθήκευσης 4 δυαδικών ψηφίων.
- Οι τιμές των ψηφίων εισόδου (a_0, a_1, a_2, a_3) που φορτώνονται (εισάγονται) παράλληλα στον καταχωρητή, μεταφέρονται ταυτόχρονα στις αντίστοιχες εξόδους (b_0, b_1, b_2, b_3), κατά την ανερχόμενη ακμή του σήματος ρολογιού, οπότε και γίνεται παράλληλη ανάγνωσή (έξοδός) τους.
- Η αποθήκευση περισσότερων δυαδικών ψηφίων επιτυγχάνεται εύκολα με αντίστοιχη αύξηση του πλήθους των φλιπ-φλοπ.
- Η προσθήκη δυνατότητας μηδενισμού της πληροφορίας του καταχωρητή μπορεί να επιτευχθεί με τη χρησιμοποίηση φλιπ-φλοπ με ασύγχρονη είσοδο καθαρισμού (clear).



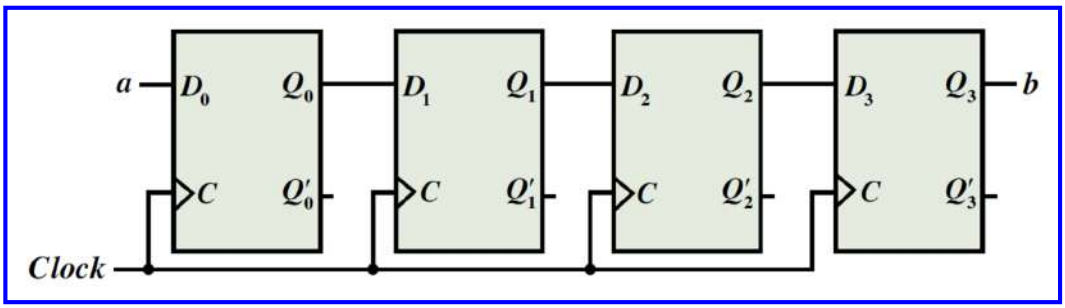
Παράλληλος καταχωρητής με είσοδο φόρτωσης

- Σε πολλές εφαρμογές είναι επιθυμητό να μην αλλάζουν για ορισμένο χρονικό διάστημα τα δεδομένα που είναι αποθηκευμένα σε έναν καταχωρητή.
- Αυτό επιτυγχάνεται με την προσθήκη πολυπλεκτών 2-σε-1, ισάριθμων με τα φλιπ-φλοπ, η είσοδος επιλογής των οποίων τροφοδοτείται με μία επιπλέον είσοδο L του καταχωρητή, που αναφέρεται ως είσοδος ελέγχου παράλληλης εισόδου ή είσοδος φόρτωσης.
- Όταν L = 0, η είσοδος δεδομένων κάθε φλιπ-φλοπ τροφοδοτείται με την τιμή της παρούσας κατάστασής του (τα δεδομένα του καταχωρητή παραμένουν αναλλοίωτα).
- Όταν L = 1, στις εισόδους δεδομένων των φλιπ-φλοπ τροφοδοτούνται τα δυαδικά ψηφία εισόδου του καταχωρητή (νέα δεδομένα στη ανερχόμενη ακμή του σήματος ρολογιού).



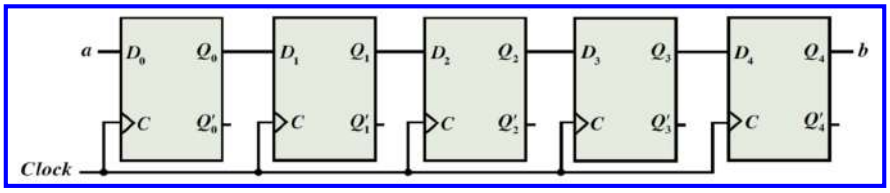
Καταχωρητής ολίσθησης με σειριακή είσοδο & έξοδο

- Ο καταχωρητής ολίσθησης με σειριακή είσοδο και έξοδο είναι μία αλυσίδα από φλιπ-φλοπ, τα οποία τροφοδοτούνται με κοινό σήμα ρολογιού.
- Με την έλευση της ανερχόμενης ακμής κάθε παλμού του σήματος ρολογιού, φορτώνεται ένα δυαδικό ψηφίο στην είσοδο a του καταχωρητή και τα περιεχόμενα του καταχωρητή μετατοπίζονται ή ολισθαίνουν μία θέση προς τα δεξιά.
- Για καταχωρητή με n φλιπ-φλοπ, ύστερα από n παλμούς ρολογιού, το δυαδικό ψηφίο που εισήλθε πρώτο θα είναι αποθηκευμένο στο τελευταίο φλιπ-φλοπ της αλυσίδας, ενώ το δυαδικό ψηφίο που εισήλθε τελευταίο θα είναι αποθηκευμένο στο πρώτο φλιπ-φλοπ.

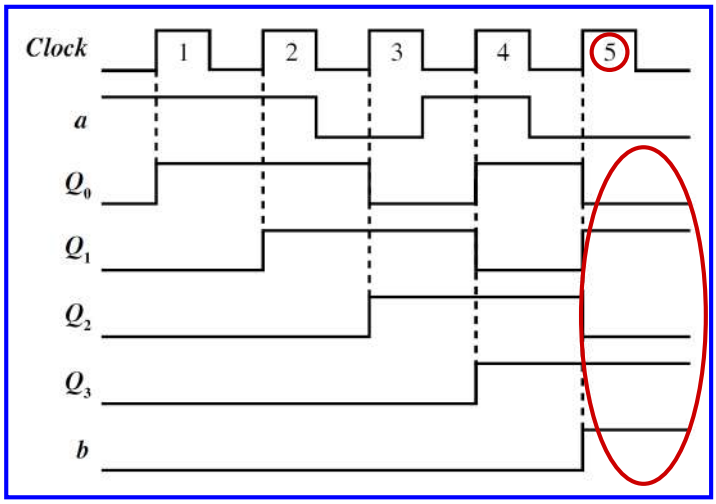


Καταχωρητής ολίσθησης με σειριακή είσοδο & έξοδο

Παράδειγμα: έστω ότι στην είσοδο a του ακόλουθου καταχωρητή ολίσθησης, εισάγεται σειριακά η ακολουθία δυαδικών ψηφίων 11010:

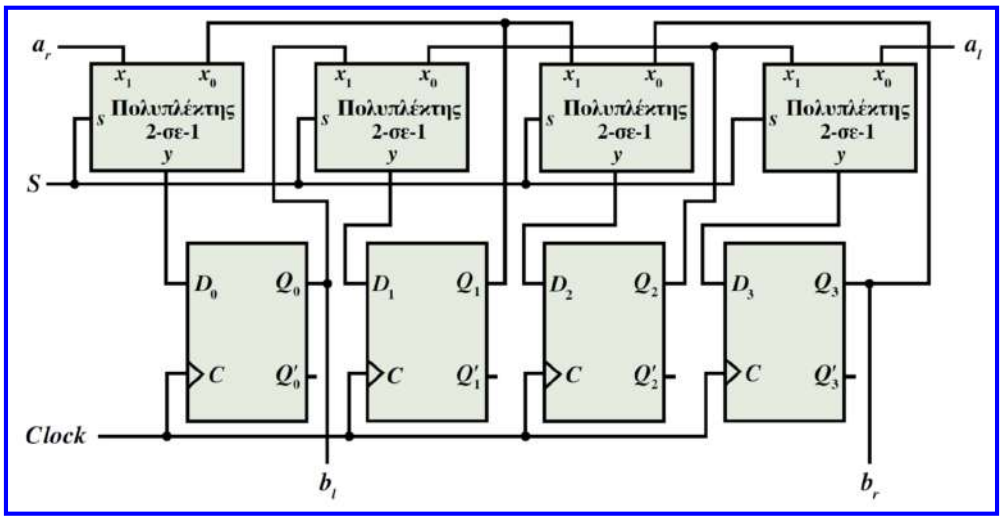


Μετά από 5 παλμούς του σήματος ρολογιού, το περιεχόμενο του καταχωρητή ολίσθησης (δηλ. οι τιμές των εξόδων των φλιπ-φλοπ) είναι η ακολουθία $bQ_3Q_2Q_1Q_0$ = 11010, η οποία συμπίπτει με την εισερχόμενη ακολουθία



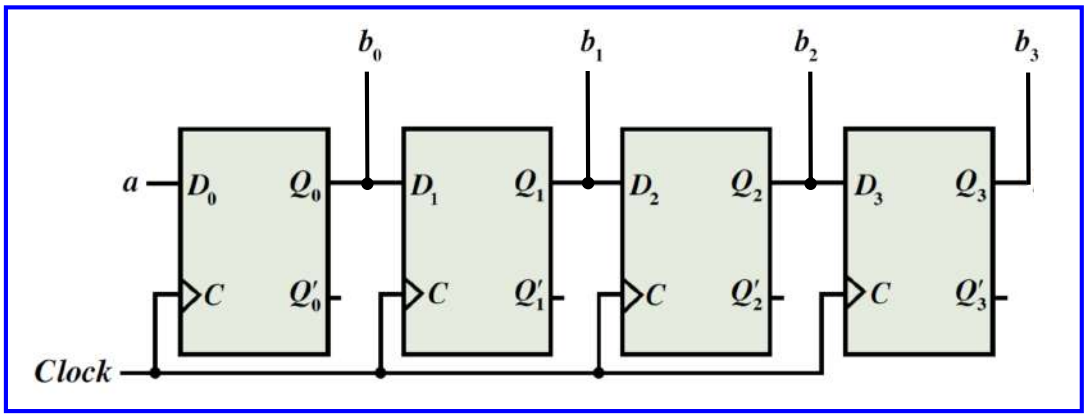
Αμφίδρομος καταχωρητής ολίσθησης

- Ο καταχωρητής ολίσθησης που είδαμε μετατοπίζει τα δυαδικά ψηφία εισόδου προς τα δεξιά.
- Μπορούμε να συνθέσουμε αμφίδρομο καταχωρητή ολίσθησης (bidirectional shift register) με κατάλληλη προσθήκη πολυπλεκτών 2-σε-1, ισάριθμων με τα φλιπ-φλοπ.
- Η είσοδος S ελέγχει τη φορά της ολίσθησης.
- Όταν S = 1, στην είσοδο κάθε φλιπ-φλοπ συνδέεται η έξοδος του προηγούμενου και εκτελείται ολίσθηση προς τα δεξιά, με σειριακή είσοδο a_r και σειριακή έξοδο b_r.
- Όταν S = 0, στην είσοδο κάθε φλιπ-φλοπ συνδέεται η έξοδος του επόμενου και εκτελείται ολίσθηση προς τα αριστερά, με σειριακή είσοδο a_l και σειριακή έξοδο b_l.



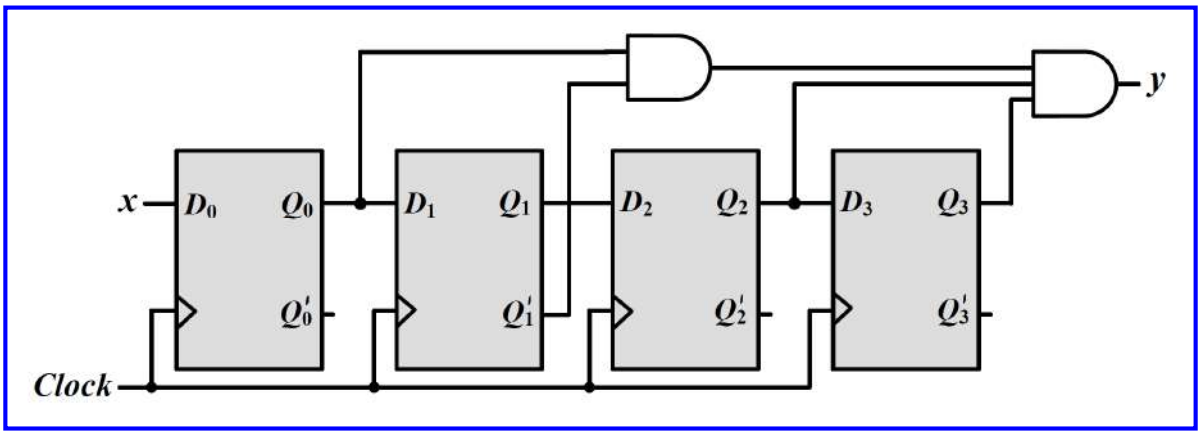
Καταχωρητής με σειριακή είσοδο & παράλληλη έξοδο

- Ο καταχωρητής ολίσθησης με σειριακή είσοδο και παράλληλη έξοδο είναι όμοιος με τον καταχωρητή ολίσθησης με σειριακή είσοδο και σειριακή έξοδο, με μόνη διαφορά ότι είναι διαθέσιμες οι έξοδοι των flip-flops που συμμετέχουν στο καταχωρητή.



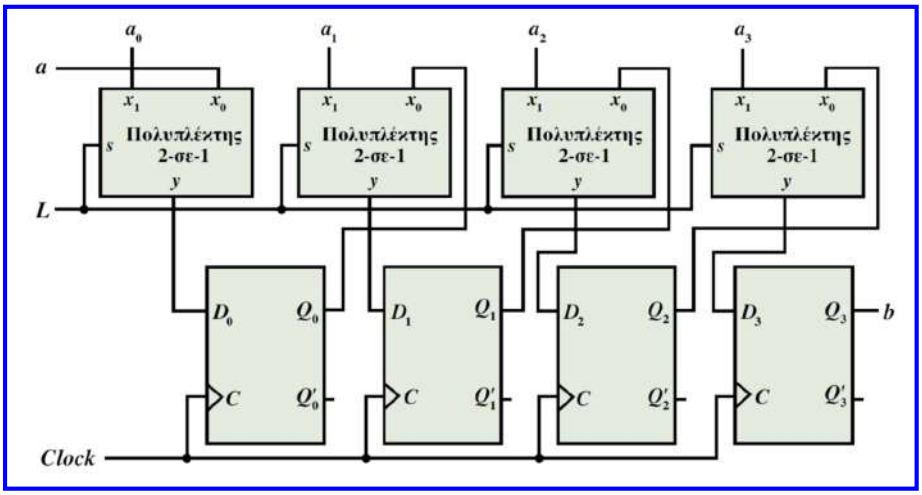
Καταχωρητής με σειριακή είσοδο & παράλληλη έξοδο

Παράδειγμα: Χρησιμοποιώντας καταχωρητή 4 ψηφίων με σειριακή είσοδο και παράλληλη έξοδο και πύλες AND 2 και 3 εισόδων, να σχεδιαστεί σύγχρονο ακολουθιακό κύκλωμα που ανιχνεύει την ακολουθία 1101 όταν αυτή λαμβάνεται στη σειριακή του είσοδο.



Καταχωρητής με παράλληλη είσοδο & σειριακή έξοδο

- Η σύνθεση ενός καταχωρητή ολίσθησης με δυνατότητα παράλληλης φόρτωσης (εισόδου), επιτυγχάνεται με χρήση πολυπλεκτών 2-σε-1.
- Η είσοδος L ελέγχει την παράλληλη είσοδο (φόρτωση) δεδομένων.
- Όταν L = 0, τότε η λειτουργία του κυκλώματος είναι όμοια με εκείνη του καταχωρητή ολίσθησης με σειριακή είσοδο (a) και σειριακή έξοδο (b).
- Όταν L = 1, τότε ενεργοποιείται η παράλληλη φόρτωση των δεδομένων, μέσω των γραμμών εισόδου a_0, a_1, a_2 και a_3.

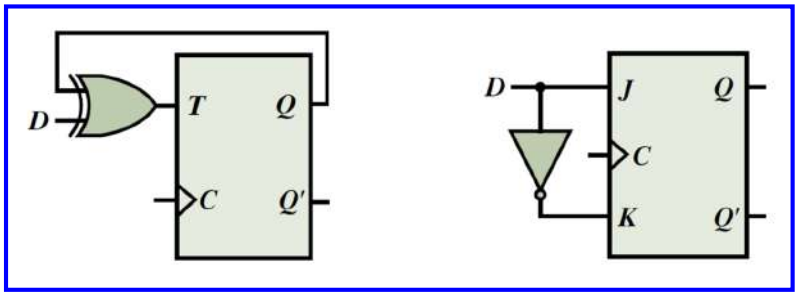


Υλοποίηση καταχωρητών με τα υπόλοιπα φλιπ-φλοπ

- Για τη σύνθεση των διάφορων παραλλαγών καταχωρητών μπορούν να χρησιμοποιηθούν φλιπ-φλοπ JK ή T αντί για D φλιπ-φλοπ με βάση απλές μετατροπές μεταξύ των φλιπ-φλοπ.
- Για να μετατρέψουμε ένα T ή JK σε φλιπ-φλοπ D, στον πίνακα λειτουργίας του D flip-flop προσθέτουμε επιπλέον στήλες, στις οποίες καταγράφονται οι τιμές των T, J και K για κάθε ζεύγος τιμών παρούσας και επόμενης κατάστασης. Οι τιμές αυτές προκύπτουν άμεσα από τους πίνακες διέγερσης των T ή JK σε φλιπ-φλοπ D.

D	Q_t	Q_{t+1}	T	J	K
0	0	0	0	0	×
0	1	0	1	×	1
1	0	1	1	1	×
1	1	1	0	×	0

$\Rightarrow$ $T = D \oplus Q$
$J = D$ και $K = D'$

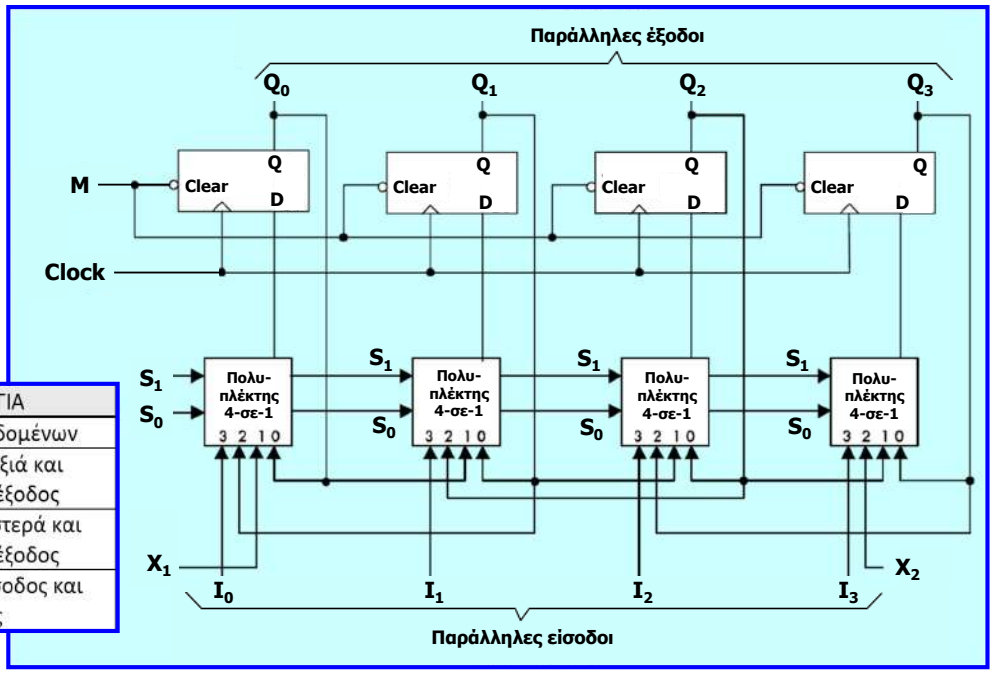


Υλοποίηση σύνθετων καταχωρητών

Οι διάφορες παραλλαγές καταχωρητών μπορούν να συνδυαστούν σε πιο σύνθετα κυκλώματα, με χρήση πολυπλεκτών περισσότερων εισόδων ή ακόμη και με την προσθήκη πιο σύνθετων συνδυαστικών κυκλωμάτων

Αμφίδρομος καταχωρητής παράλληλης εισόδου και εξόδου με δυνατότητα σειριακής ολίσθησης προς τα δεξιά και προς τα αριστερά

S_1	S_0	ΛΕΙΤΟΥΡΓΙΑ
0	0	Διατήρηση δεδομένων
0	1	Ολίσθηση δεξιά και παράλληλη έξοδος
1	0	Ολίσθηση αριστερά και παράλληλη έξοδος
1	1	Παράλληλη είσοδος και έξοδος



Μετρητές

- Οι μετρητές (counters) είναι ακολουθιακά κυκλώματα που διατρέχουν μια καθορισμένη ακολουθία καταστάσεων, καθεμία από τις οποίες αντιστοιχεί σε ένα φυσικό αριθμό.
- Ένας μετρητής που περιλαμβάνει n φλιπ-φλοπ μπορεί να διατρέξει μέχρι 2^n διαφορετικές καταστάσεις, δηλαδή έχει τη δυνατότητα να μετράει μια καθορισμένη ακολουθία που μπορεί να περιλαμβάνει μέχρι 2^n αριθμούς.
- Οι μετρητές διακρίνονται σε δύο βασικές κατηγορίες:
  - ✓ τους σύγχρονους μετρητές (synchronous counters), όπου όλα τα φλιπ-φλοπ και συνεπώς η αλλαγή κατάστασης ελέγχεται από ένα κοινό σήμα ρολογιού, και
  - ✓ τους ασύγχρονους μετρητές (asynchronous counters) ή μετρητές διάδοσης (ripple counters), όπου δεν υφίσταται έλεγχος από κοινό σήμα ρολογιού και η έξοδος κάθε φλιπ-φλοπ μπορεί να χρησιμοποιηθεί για την πυροδότηση άλλων φλιπ-φλοπ.
- Οι δυαδικοί μετρητές (binary counters) είναι σύγχρονοι μετρητές οι οποίοι, όταν αποτελούνται από n φλιπ-φλοπ, ακολουθούν τη φυσική αρίθμηση, μετρώντας ανοδικά ή καθοδικά μεταξύ των αριθμών 0 και $2^n - 1$.
- Οι σύγχρονοι μετρητές επαναλαμβανόμενης ακολουθίας αριθμών, που αναφέρονται και ως μετρητές υπολοίπου διαίρεσης ως προς N (modulo-N counters), διατρέχουν ακολουθία μέτρησης N αριθμών, χωρίς να ακολουθείται απαραιτήτως η φυσική αρίθμηση και το πλήθος N των αριθμών δεν αποτελεί δύναμη του 2.

Σύγχρονοι δυαδικοί μετρητές

- Οι σύγχρονοι μετρητές (synchronous counters) είναι μετρητές οι οποίοι με την έλευση των ακμών του σήματος ρολογιού που πυροδοτούν τα φλιπ-φλοπ τους, διατρέχουν μια καθορισμένη ακολουθία καταστάσεων, καθεμία από τις οποίες αντιστοιχεί σε ένα φυσικό αριθμό.
- Ένας σύγχρονος μετρητής που περιλαμβάνει n φλιπ-φλοπ, μπορεί να διατρέξει μέχρι 2^n διαφορετικές καταστάσεις, δηλαδή έχει τη δυνατότητα να μετράει μία καθορισμένη ακολουθία που μπορεί να περιλαμβάνει μέχρι $N = 2^n$ αριθμούς.
- Οι σύγχρονοι δυαδικοί μετρητές (synchronous binary counters) αποτελούνται από n φλιπ-φλοπ και ακολουθούν τη φυσική αρίθμηση, μετρώντας ανοδικά ή καθοδικά μεταξύ των αριθμών 0 και $2^n - 1$.
- Η αλλαγή της κατάστασης των σύγχρονων μετρητών, δηλαδή η μετάβαση από έναν αριθμό της ακολουθίας μέτρησης στον επόμενο, λαμβάνει χώρα με την έλευση της ακμής του κοινού σήματος ρολογιού (ανερχόμενης ή κατερχόμενης), κατά την οποία πυροδοτούνται τα φλιπ-φλοπ που τους συνθέτουν.
- Η ακολουθία μέτρησης των σύγχρονων δυαδικών μετρητών είναι επαναλαμβανόμενη.

Σύνθεση σύγχρονου δυαδικού μετρητή

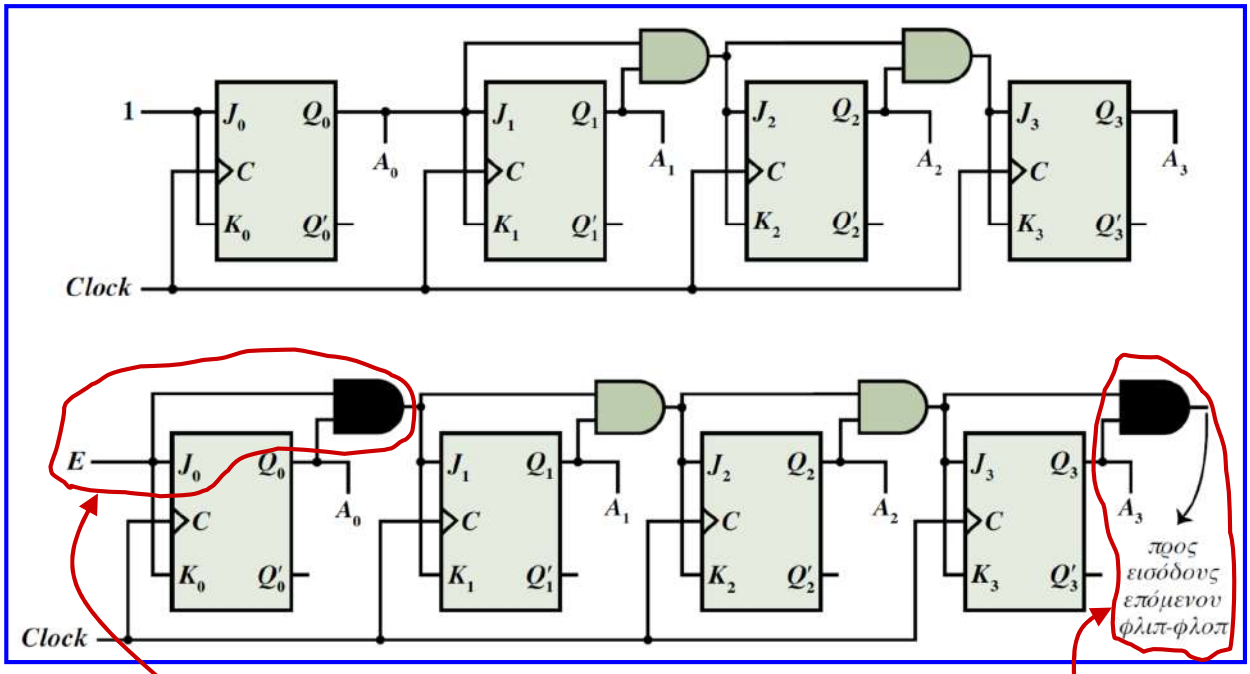
- Σύγχρονος δυαδικός μετρητής με 4 ψηφία: σύνθεση με ισάριθμα φλιπ-φλοπ.
- Αφού, n = 4, ο μετρητής διατρέχει 2^4 καταστάσεις, μετρώντας από τον αριθμό 0 έως τον αριθμό $2^4 - 1 = 15$.
- Για τη σύνθεση του κυκλώματος μπορούμε να χρησιμοποιήσουμε τη διαδικασία σύνθεσης ΣΑΚ.
- Στην περίπτωση αυτή, αφού σχεδιάσουμε το διάγραμμα με τις 16 καταστάσεις που αντιστοιχούν, κατά σειρά, στους δυαδικούς αριθμούς από 0000 έως 1111, θα πρέπει να δημιουργήσουμε τον κωδικοποιημένο πίνακα καταστάσεων και στη συνέχεια να υλοποιήσουμε το κύκλωμα με φλιπ-φλοπ της επιλογής μας.
- Τα βήματα της ελαχιστοποίησης και της κωδικοποίησης των καταστάσεων δε βρίσκουν εφαρμογή στην περίπτωση των μετρητών, αφού οι καταστάσεις αντιστοιχούν σε δυαδικούς αριθμούς με πλήθος ψηφίων ίσο με το πλήθος των μεταβλητών κατάστασης.
- Ωστόσο, τα κυκλώματα των σύγχρονων δυαδικών μετρητών παρουσιάζουν κανονικότητα, με αποτέλεσμα η σύνθεσή τους να μην απαιτεί την εφαρμογή της διαδικασίας σύνθεσης ΣΑΚ.

Σύνθεση σύγχρονου δυαδικού μετρητή

A_3	A_2	A_1	A_0	Δεκαδικός αριθμός
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

- Η μεταβλητή A_0 συμπληρώνεται κατά τη μετάβαση από μία κατάσταση στην επόμενη
- Οι υπόλοιπες μεταβλητές συμπληρώνονται όταν όλες οι λιγότερο σημαντικές μεταβλητές έχουν τιμή 1.
- Έστω, ότι επιθυμούμε υλοποίηση με φλιπ-φλοπ JK πυροδοτούμενα στην ανερχόμενη ακμή του σήματος ρολογιού.
- Στο φλιπ-φλοπ με έξοδο τη μεταβλητή A_0, πρέπει να θέσουμε $J_0 = K_0 = 1$, ώστε να συμπληρώνεται η έξοδός του με την έλευση κάθε ανερχόμενης ακμής του σήματος ρολογιού.
- Η έξοδος αυτή θα τροφοδοτήσει απευθείας τις εισόδους του φλιπ-φλοπ με έξοδο την A_1, με αποτέλεσμα, όταν $A_0 = J_1 = K_1 = 1$, να λαμβάνει χώρα η επιθυμητή συμπλήρωση.
- Η συμπλήρωση των εξόδων των υπόλοιπων φλιπ-φλοπ συμβαίνει όταν έχουν τιμή 1 οι έξοδοι των φλιπ-φλοπ που αντιστοιχούν στις λιγότερο σημαντικές μεταβλητές.

Σύνθεση σύγχρονου δυαδικού μετρητή

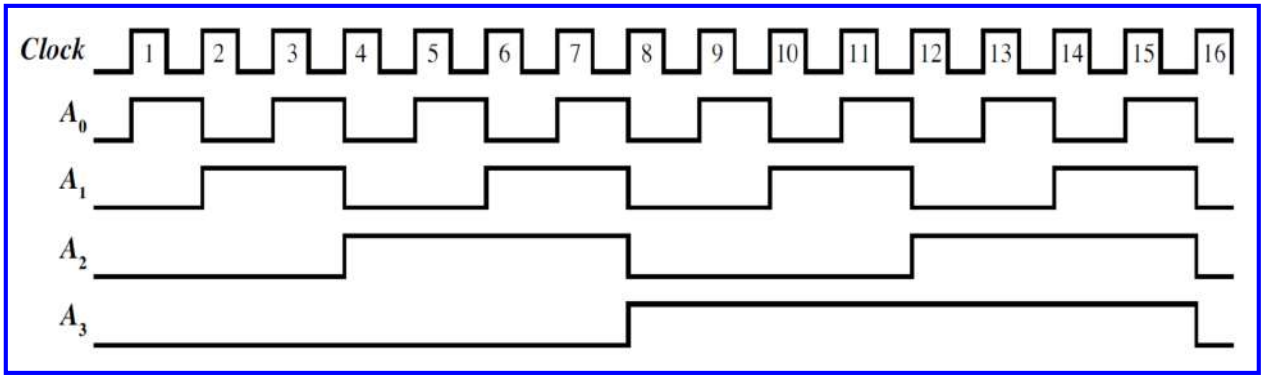


Δυνατότητα διακοπής/επανεκκίνησης μέτρησης

Δυνατότητα επέκτασης σε περισσότερα ψηφία

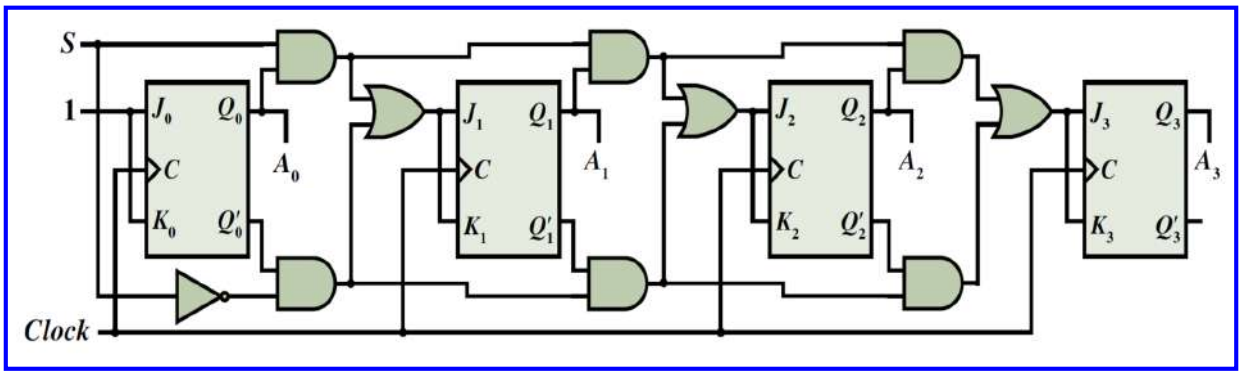
Σύνθεση σύγχρονου δυαδικού μετρητή

Κυματομορφές εξόδου ανοδικού σύγχρονου δυαδικού μετρητή 4 ψηφίων



Αμφίδρομος σύγχρονος δυαδικός μετρητής

- Όπως προκύπτει από την αντίστροφη ακολουθία μέτρησης, αντίστοιχη κανονικότητα με τον ανοδικό μετρητή, παρουσιάζει και ο καθοδικός μετρητής.
- Ξεκινώντας από τον τελευταίο αριθμό της ακολουθίας, διαπιστώνουμε ότι η μεταβλητή κατάστασης A_0 συμπληρώνεται κατά τη μετάβαση από μία κατάσταση στην επόμενη, ενώ οι υπόλοιπες μεταβλητές κατάστασης συμπληρώνονται όταν όλες οι λιγότερο σημαντικές μεταβλητές έχουν τιμή 0.
- Ο αμφίδρομος σύγχρονος δυαδικός μετρητής εκτελεί ανοδική ή καθοδική μέτρηση, όταν η είσοδος ελέγχου της κατεύθυνσης της μέτρησης (S) λαμβάνει τιμή 1 ή 0, αντίστοιχα.



Σύγχρονοι μετρητές επαναλαμβανόμενης ακολουθίας

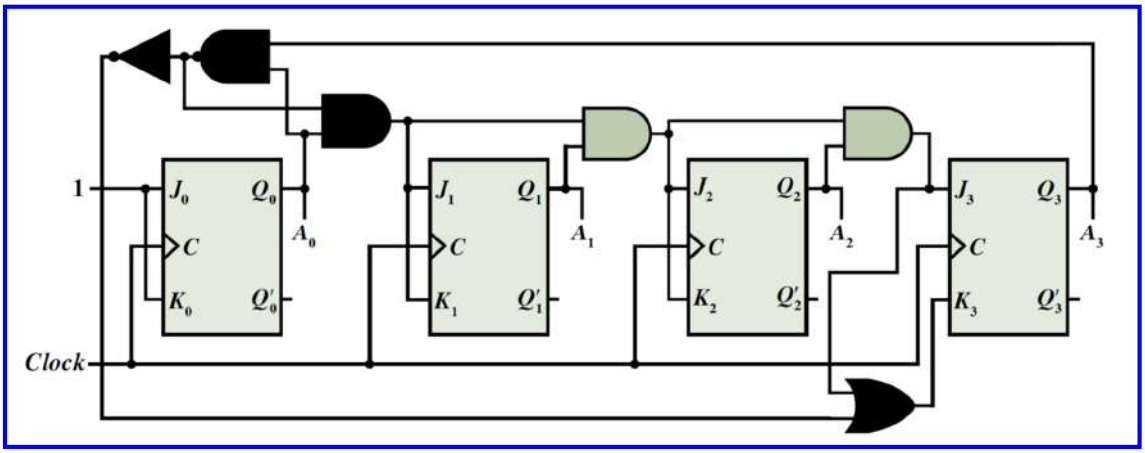
- Οι σύγχρονοι δυαδικοί μετρητές που παρουσιάστηκαν, έχουν τη δυνατότητα μέτρησης μεταξύ των αριθμών 0 και $2^n - 1$.
- Ωστόσο, μπορούμε να συνθέσουμε μετρητές που να παράγουν μια επαναλαμβανόμενη ακολουθία N αριθμών, που δεν ακολουθούν απαραιτήτως τη φυσική αρίθμηση και το πλήθος των αριθμών της ακολουθίας μέτρησης δεν αποτελεί δύναμη του 2.
- Αυτοί αναφέρονται ως μετρητές υπολοίπου διαίρεσης ως προς N (modulo-N counters).
- Για τη σύνθεσή τους χρησιμοποιούμε τη διαδικασία σύνθεσης ΣΑΚ (σχεδιασμός διαγράμματος καταστάσεων, κατάστρωση κωδικοποιημένου πίνακα καταστάσεων, υλοποίηση).
- Στους μετρητές αυτούς, το πλήθος N των αριθμών της ακολουθίας μέτρησης είναι τέτοιο, ώστε $2^{k-1} < N < 2^k$, όπου k είναι το πλήθος των φλιπ-φλοπ που πρέπει να χρησιμοποιηθούν για τη σύνθεσή τους.
- Αυτό έχει αποτέλεσμα την ύπαρξη $2^k - N$ αχρησιμοποίητων καταστάσεων (ή αριθμών), οι οποίες αντιμετωπίζονται συνήθως ως αδιάφοροι όροι.
- Μετά τη σύνθεση, αναλύεται η λειτουργία του μετρητή, ώστε να διαπιστωθεί ότι, εάν βρεθεί στις αχρησιμοποίητες καταστάσεις, δεν εγκλωβίζεται σε αυτές, αλλά επιστρέφει στην επιθυμητή ακολουθία μέτρησης, ύστερα από έναν ή περισσότερους παλμούς του σήματος ρολογιού (μετρητής με αυτόματη διόρθωση, self-correcting counter).

Σύγχρονοι μετρητές επαναλαμβανόμενης ακολουθίας

- Εναλλακτικά, όταν ακολουθείται η φυσική αρίθμηση, αλλά το πλήθος των αριθμών της ακολουθίας μέτρησης δεν αποτελεί δύναμη του 2, ακολουθούμε την παρακάτω διαδικασία.
- Για σύνθεση μετρητή με N καταστάσεις, συνθέτουμε ένα σύγχρονο δυαδικό μετρητή που αποτελείται από k φλιπ-φλοπ (με $2^{k-1} < N < 2^k$) και με 1 πύλη NAND αποκωδικοποιούμε την κατάσταση N – 1 (θέτουμε στις εισόδους της NAND τις εξόδους με τιμή 1).
- Εντοπίζουμε τα φλιπ-φλοπ, με έξοδο 0, όταν ο μετρητής βρίσκεται στην κατάσταση N – 1 και έξοδο 1 όταν ο μετρητής βρίσκεται στην κατάσταση N.
- Κατόπιν, διευρύνουμε κατά 1 τις εισόδους της πύλης AND που τροφοδοτεί την είσοδο J των φλιπ-φλοπ που εντοπίσαμε και τροφοδοτούμε τη νέα είσοδο με την έξοδο της NAND.
- Όταν πρόκειται για φλιπ-φλοπ, των οποίων η είσοδος J δεν τροφοδοτείται μέσω πύλης AND, προσθέτουμε μια νέα πύλη AND.
- Στη συνέχεια, εντοπίζουμε τα φλιπ-φλοπ, των οποίων η έξοδος έχει τιμή 1, όταν ο μετρητής βρίσκεται στις καταστάσεις N – 1 και N.
- Τροφοδοτούμε τη συμπληρωματική μορφή της εξόδου της πύλης NAND στην είσοδο K των φλιπ-φλοπ που εντοπίσαμε, μέσω μιας πύλης OR.
- Με την 1η παρέμβαση εμποδίζουμε τη μετάβαση από την κατάσταση N – 1 στην N, ενώ με τη 2η προκαλούμε καθαρισμό των φλιπ-φλοπ, έτσι ώστε να γίνει μετάβαση από την κατάσταση N – 1 στη 0.

Σύγχρονοι μετρητές επαναλαμβανόμενης ακολουθίας

- Σύνθεση σύγχρονου μετρητή BCD: N – 1 = 1001 (9) και N = 1010 (10).
- Για να εμποδίσουμε τη μετάβαση από την N – 1 = 1001 στην N = 1010 μέσω μιας πύλης AND 2 εισόδων, τροφοδοτούμε με την έξοδο της πύλης NAND την είσοδο J_1 του φλιπ-φλοπ, του οποίου η έξοδος A_1 είναι 0 στην κατάσταση 1001 και 1 στην κατάσταση 1010.
- Για να προκαλέσουμε μετάβαση από την N = 1001 στην κατάσταση 0000, με μία πύλη OR, τροφοδοτούμε τη συμπληρωματική μορφή της εξόδου της πύλης NAND στην είσοδο K_3 του φλιπ-φλοπ, του οποίου η έξοδος A_3 είναι 1 στις καταστάσεις 1001 και 1010.



Σύγχρονοι μετρητές επαναλαμβανόμενης ακολουθίας

- Μια 2η εναλλακτική μέθοδος σύνθεσης μετρητών οι οποίοι ακολουθούν τη φυσική αρίθμηση, αλλά το πλήθος των αριθμών της ακολουθίας δεν αποτελεί δύναμη του 2, βασίζεται στην αξιοποίηση της ασύγχρονης εισόδου καθαρισμού των φλιπ-φλοπ.

- Αφού συνθέσουμε το σύγχρονο δυαδικό μετρητή με k φλιπ-φλοπ (με $2^{k-1} < N < 2^k$), αποκωδικοποιούμε την κατάσταση N χρησιμοποιώντας μία πύλη NAND και συνδέουμε την έξοδο της πύλης αυτής στις εισόδους καθαρισμού των φλιπ-φλοπ.

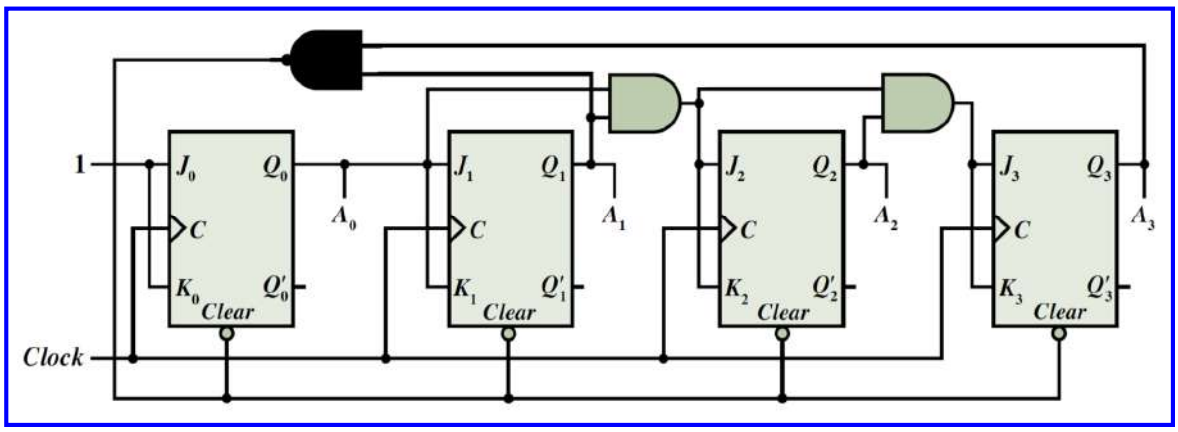
- Κατά τη μετάβαση του μετρητή από την κατάσταση N – 1 στην N, η έξοδος της πύλης λαμβάνει τιμή 0, οι έξοδοι του μετρητή μηδενίζονται και επαναλαμβάνεται η επιθυμητή ακολουθία μέτρησης.

- Η κατάσταση N, η οποία δεν περιλαμβάνεται στην επιθυμητή ακολουθία μέτρησης, εμφανίζεται για πολύ μικρό χρονικό διάστημα με τη μορφή στενών παλμών (αιχμών) στις κυματομορφές ορισμένων εξόδων του μετρητή.

- Οι ανεπιθύμητοι αυτοί παλμοί ενδέχεται να δημιουργήσουν προβλήματα στη λειτουργία ενός ψηφιακού συστήματος, ιδιαίτερα στην περίπτωση που η συχνότητα του σήματος ρολογιού είναι μεγάλη.

Σύγχρονοι μετρητές επαναλαμβανόμενης ακολουθίας

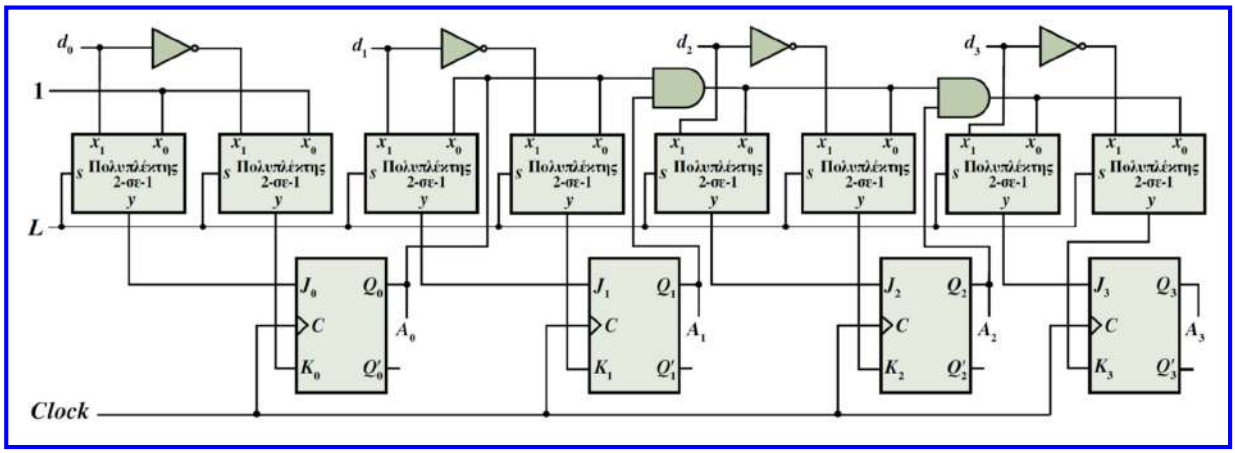
- Σύνθεση σύγχρονου μετρητή BCD: N – 1 = 1001 (9) και N = 1010 (10).
- Αποκωδικοποιούμε την κατάσταση N = 1010 του σύγχρονου δυαδικού μετρητή 4 ψηφίων, με μία πύλη NAND 2 εισόδων, και συνδέουμε την έξοδο της πύλης αυτής στις εισόδους καθαρισμού των φλιπ-φλοπ του μετρητή.
- Η μη έγκυρη κατάσταση N = 1010 υφίσταται για μικρό χρονικό διάστημα, με τη λήξη του οποίου ο μετρητής μεταβαίνει στην κατάσταση 0000.



Σύγχρονοι μετρητές με παράλληλη φόρτωση

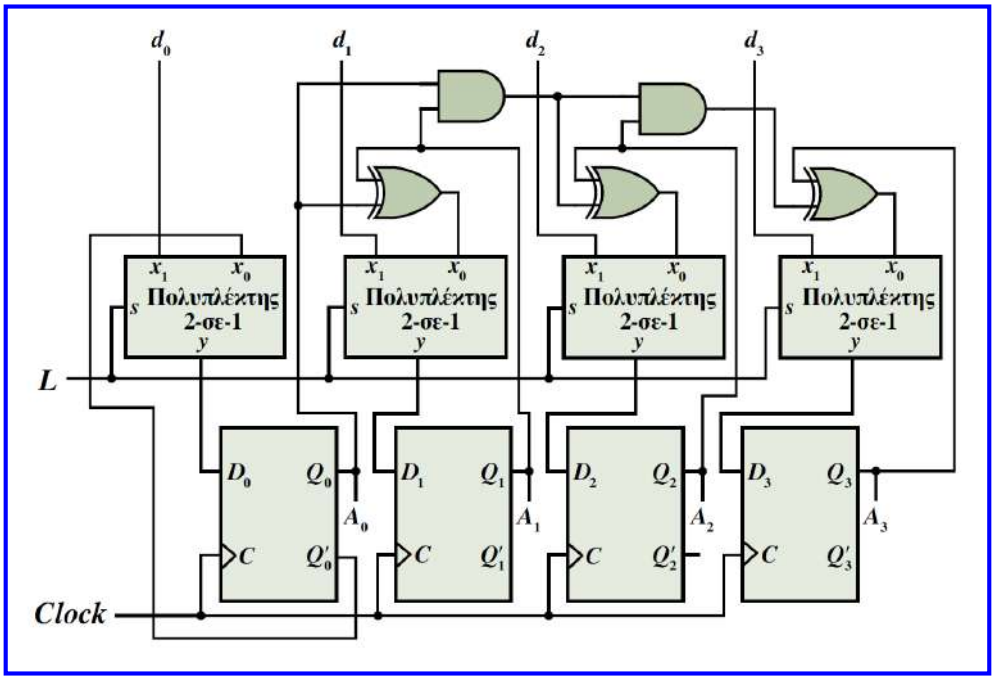
- Οι μετρητές με δυνατότητα παράλληλης φόρτωσης, διαθέτουν είσοδο ελέγχου παράλληλης φόρτωσης (L), με την ενεργοποίηση της οποίας αναστέλλεται η μέτρηση και μεταφέρεται ο επιθυμητός αριθμός εισόδου στις εξόδους των φλιπ-φλοπ του μετρητή.
- Όταν L = 0, η λειτουργία του μετρητή είναι όμοια με του σύγχρονου δυαδικού μετρητή και όταν L = 1, το κύκλωμα λειτουργεί ως καταχωρητής με παράλληλη φόρτωση και κατά την ανερχόμενη ακμή του ρολογιού μεταφέρονται οι είσοδοι φόρτωσης δεδομένων στις εξόδους του μετρητή.

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t



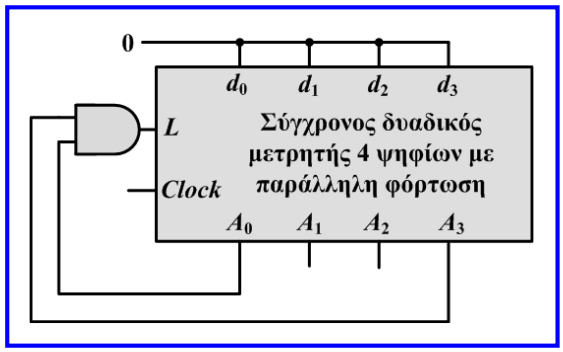
Σύγχρονοι μετρητές με παράλληλη φόρτωση

Υλοποίηση σύγχρονου μετρητή με δυνατότητα παράλληλης φόρτωσης με φλιπ-φλοπ D



Σύγχρονοι μετρητές με παράλληλη φόρτωση

- Με έναν σύγχρονο δυαδικό μετρητή με παράλληλη φόρτωση, μπορούμε να συνθέσουμε σύγχρονους μετρητές με ακολουθία μέτρησης μεταξύ δύο επιθυμητών αριθμών, η οποία ακολουθεί τη φυσική αρίθμηση.
- Για παράδειγμα, η σύνθεση ενός σύγχρονου μετρητή BCD επιτυγχάνεται, εάν αρχικά μηδενίσουμε τις εισόδους φόρτωσης δεδομένων του μετρητή με παράλληλη φόρτωση.
- Κατόπιν, αποκωδικοποιούμε με μία πύλη AND την τελευταία κατάσταση της ακολουθίας μέτρησης (1001 = 9) και τροφοδοτούμε την είσοδο ελέγχου παράλληλης φόρτωσης (L) με την έξοδο της πύλης αυτής.
- Έτσι, όταν ο μετρητής μεταβεί στην κατάσταση 1001, η έξοδος της AND λαμβάνει τιμή 1 και οι έξοδοι του μετρητή μηδενίζονται, ώστε να επαναληφθεί η ακολουθία μέτρησης.



Ασύγχρονοι μετρητές

- Στους ασύγχρονους μετρητές (asynchronous counters), η αλλαγή των καταστάσεων δεν ελέγχεται από ένα κοινό σήμα ρολογιού που εφαρμόζεται στα φλιπ-φλοπ.
- Το χαρακτηριστικό αυτό παρέχει τη δυνατότητα απλούστευσης των κυκλωμάτων υλοποίησής τους, σε σχέση με τους σύγχρονους μετρητές.
- Όπως προαναφέρθηκε, στους σύγχρονους μετρητές, τα φλιπ-φλοπ πυροδοτούνται στις ανερχόμενες ακμές των παλμών ενός κοινού σήματος ρολογιού.
- Ωστόσο, στους ασύγχρονους μετρητές (που αποτελούνται από σειριακά συνδεδεμένα φλιπ-φλοπ), κάθε φλιπ-φλοπ πυροδοτείται κατά τη μετάβαση που συμβαίνει στην έξοδο ή στη συμπληρωματική έξοδο προηγούμενων στη σειρά φλιπ-φλοπ.
- Μολονότι τα κυκλώματα υλοποίησης των ασύγχρονων μετρητών είναι πιο απλά από εκείνα των σύγχρονων μετρητών, παρουσιάζουν πιο αργή απόκριση, αφού κάθε φλιπ-φλοπ που περιλαμβάνουν πυροδοτείται από έξοδο του προηγούμενου.
- Ο παλμός του σήματος ρολογιού που εισέρχεται στο πρώτο φλιπ-φλοπ διαδίδεται στην αλυσίδα των φλιπ-φλοπ του ασύγχρονου μετρητή, μέχρι να φτάσει στην έξοδο του τελευταίου φλιπ-φλοπ.
- Για το λόγο αυτό, οι ασύγχρονοι μετρητές αναφέρονται και ως μετρητές διάδοσης (ripple counters).

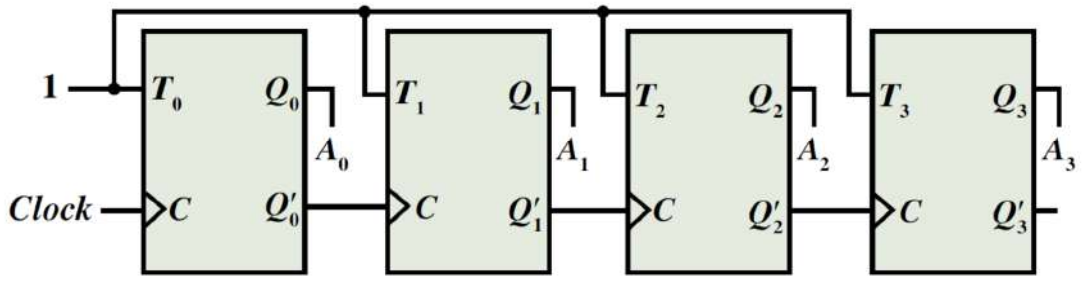
Ασύγχρονος δυαδικός μετρητής

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

- Η μεταβλητή A_0 συμπληρώνεται κατά τη μετάβαση από μία κατάσταση στην επόμενη.
- Για τις υπόλοιπες μεταβλητές, όταν η μεταβλητή Ai (i = 1, 2) μεταβαίνει από την τιμή 1 στην τιμή 0, τότε συμπληρώνεται η μεταβλητή Ai+1.
- Συνεπώς, ο ασύγχρονος δυαδικός μετρητής πρέπει να αποτελείται από σειριακά συνδεδεμένα φλιπ-φλοπ, το πρώτο από τα οποία συμπληρώνει την έξοδό του στην ανερχόμενη ακμή του σήματος ρολογιού, ενώ τα υπόλοιπα συμπληρώνουν την έξοδό τους κατά τη μετάβαση της εξόδου του προηγούμενου φλιπ-φλοπ από τιμή 1 σε 0.
- Για να επιτευχθεί η απαιτούμενη συμπλήρωση, οι είσοδοι των φλιπ-φλοπ T ή JK πρέπει να τροφοδοτούνται με τιμή 1, ενώ οι είσοδοι των φλιπ-φλοπ D πρέπει να τροφοδοτείται με τη συμπληρωματική έξοδό τους.
- Για να διασφαλιστεί η συμπλήρωση της εξόδου κάθε φλιπ-φλοπ κατά τη μετάβαση της εξόδου του προηγούμενου από τιμή 1 σε 0, αρκεί η είσοδος ρολογιού κάθε φλιπ-φλοπ να τροφοδοτείται από τη συμπληρωματική έξοδο του προηγούμενου.

Ασύγχρονος δυαδικός μετρητής

Ανοδικός ασύγχρονος δυαδικός μετρητής 4 ψηφίων με φλιπ-φλοπ T



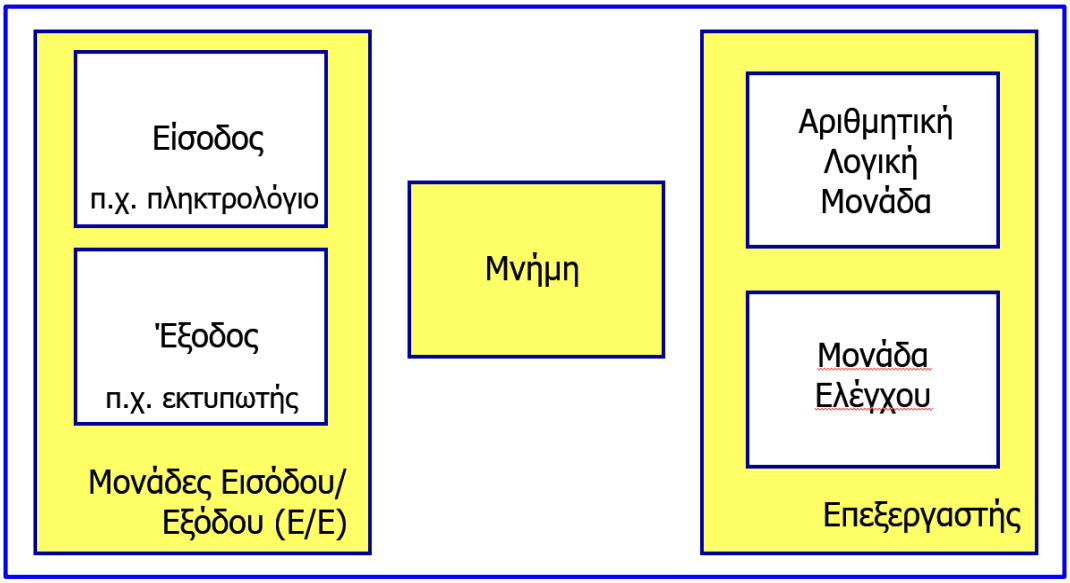
- Λόγω της κανονικότητάς του, ένας ασύγχρονος δυαδικός μετρητής μπορεί να επεκταθεί με αύξηση του αριθμού των σειριακά συνδεδεμένων φλιπ-φλοπ.
- Για τη σύνθεση ενός σύγχρονου δυαδικού μετρητή 5 ψηφίων (που μετρά από 0 έως 31), αρκεί στον μετρητή 4 ψηφίων, να συνδέσουμε με όμοιο τρόπο ένα ακόμη φλιπ-φλοπ.
- Η σύνθεση καθοδικού ασύγχρονου δυαδικού μετρητή προκύπτει με 3 τρόπους: (α) με λήψη των καταστάσεων της ακολουθίας μέτρησης από τις συμπληρωματικές εξόδους των φλιπ-φλοπ ή (β) με χρησιμοποίηση φλιπ-φλοπ που πυροδοτούνται στην κατερχόμενη ακμή των παλμών που δέχονται στην είσοδο ρολογιού ή (γ) με τροφοδοσία της εισόδου ρολογιού κάθε φλιπ-φλοπ με την κανονική έξοδο του προηγούμενου.

Μέρος Β: Οργάνωση και αρχιτεκτονική υπολογιστών

- Κεφάλαιο 1: βασικές μονάδες του υπολογιστή, αρχές λειτουργίας και απόδοση του υπολογιστή, ιστορική αναδρομή.
- Κεφάλαιο 2: οργάνωση και λειτουργία της μνήμης, τρόποι διευθυνσιοδότησης, εντολές και εκτέλεση προγραμμάτων συμβολικής γλώσσας (assembly).
- Κεφάλαιο 3: Μελέτη και προγραμματισμός του μικροεπεξεργαστή Intel 8085.
- Κεφάλαιο 4: λειτουργίες εισόδου/εξόδου (Ε/Ε) ελεγχόμενες από πρόγραμμα, διακοπές (interrupts), DMA, σύγχρονοι και ασύγχρονοι δίαυλοι, κυκλώματα διασύνδεσης, λειτουργίες Ε/Ε στον μικροεπεξεργαστή Intel 8085.
- Κεφάλαιο 5: αρχές λειτουργίας μνήμης, οργάνωση μνήμης, τύποι μνημών, συστήματα μνήμης, λανθάνουσα μνήμη, απόδοση συστήματος μνήμης, διαφύλλωση κύριας μνήμης, ιδεατή μνήμη, σκληρός δίσκος.
- Κεφάλαιο 6: οργάνωση και λειτουργίες κεντρικής μονάδας επεξεργασίας υπολογιστή, προκατασκευασμένος και μικροπρογραμματιζόμενος έλεγχος.
- Κεφάλαιο 7: διασωλήνωση για παράλληλη εκτέλεση εντολών, απόδοση διασωλήνωσης, δομικοί κίνδυνοι, εξαρτήσεις και κίνδυνοι δεδομένων, κίνδυνοι εντολών, απόδοση διασωλήνωσης, υπερβαθμωτοί επεξεργαστές.

Κεφάλαιο 1: Βασική δομή των υπολογιστών

Βασικές λειτουργικές μονάδες του υπολογιστή



Μονάδες μνήμης

- Ρόλος: αποθήκευση προγραμμάτων και δεδομένων.
- Βασικές κατηγορίες: κύρια και δευτερεύουσες αποθηκευτικές μονάδες.
- Κύρια αποθηκευτική μονάδα:
  - ✓ Διαθέτει μεγάλο αριθμό αποθηκευτικών κυττάρων, 1 bit κάθε κύτταρο (cell).
  - ✓ Τα προγράμματα πρέπει να είναι αποθηκευμένα σε αυτή όταν εκτελούνται.
  - ✓ Οργάνωση περιεχομένων σε λέξεις (τυπικά: 8-64 bits), σε καθεμία από τις οποίες αντιστοιχεί μια διεύθυνση.
  - ✓ Προσπέλαση λέξης με καθορισμό της διεύθυνσης της σε μια εντολή που εκκινεί τη διαδικασία αποθήκευσης (εγγραφής) ή ανάκτησης (ανάγνωσης).
  - ✓ Μνήμη τυχαίας προσπέλασης (RAM): κάθε λέξη της μπορεί να προσπελασθεί σε μικρό και σταθερό χρόνο (μερικά ns έως 100 ns), αφού καθοριστεί η διεύθυνσή της.
  - ✓ Ιεραρχική υλοποίηση: λανθάνουσα μνήμη (cache) ισχυρά συζευγμένη με τον επεξεργαστή για μεγαλύτερη ταχύτητα προσπέλασης και κύρια μνήμη μεγαλύτερου μεγέθους και πιο αργή.
- Δευτερεύουσες αποθηκευτικές μονάδες: για μεγάλη ποσότητα δεδομένων και προγραμμάτων μη συχνής προσπέλασης (μαγνητικοί δίσκοι, CD-ROM κ.ά.)

Αριθμητική λογική μονάδα (ΑΛΜ) & μονάδα ελέγχου

- Αριθμητική και λογική μονάδα:
  - ✓ Ρόλος: εκτέλεση των λογικών και αριθμητικών λειτουργιών των εντολών.
  - ✓ Τα ορίσματα (operands, δεδομένα στα οποία εκτελούνται οι λειτουργίες) μεταφέρονται από τη μνήμη στον επεξεργαστή, καταχωρούνται σε αποθηκευτικά στοιχεία υψηλής ταχύτητας (καταχωρητές) και εκτελείται η πράξη στην ΑΛΜ.
  - ✓ Η ΑΛΜ (και η μονάδα ελέγχου) είναι πιο γρήγορες από άλλες συσκευές ενός υπολογιστικού συστήματος, ώστε να είναι ικανός ο επεξεργαστής να εξυπηρετεί έναν αριθμό από περιφερειακές μονάδες, όπως πληκτρολόγιο, οθόνη, δίσκους, αισθητήρες.
- Μονάδα ελέγχου:
  - ✓ Ρόλος: συντονισμός των υπολοίπων μονάδων του υπολογιστή για την εκτέλεση των διαφόρων λειτουργιών που περιλαμβάνονται στις εντολές.
  - ✓ Παράγει σήματα χρονισμού που ελέγχουν τις μεταφορές δεδομένων μεταξύ περιφερειακών μονάδων και επεξεργαστή, μνήμης και επεξεργαστή.
  - ✓ Στην πράξη τα κυκλώματα ελέγχου είναι κατανεμημένα σε διάφορα μέρη του υπολογιστή και τα σήματα που παράγουν μεταφέρονται μέσω γραμμών ελέγχου στις υπόλοιπες μονάδες, για να συγχρονίσουν τις διάφορες ενέργειες.

Βασικές λειτουργίες του επεξεργαστή

- Κάθε πρόγραμμα αποτελείται από μια σειρά εντολών και αποθηκεύεται στη μνήμη.
- Οι εντολές προσκομίζονται (fetch) από τη μνήμη στον επεξεργαστή, ο οποίος εκτελεί τις λειτουργίες που ορίζονται σε αυτές.
- Τα δεδομένα που χρειάζονται ως ορίσματα (operands) των πράξεων είναι αποθηκευμένα στη μνήμη ή σε καταχωρητές (Ri) του επεξεργαστή.

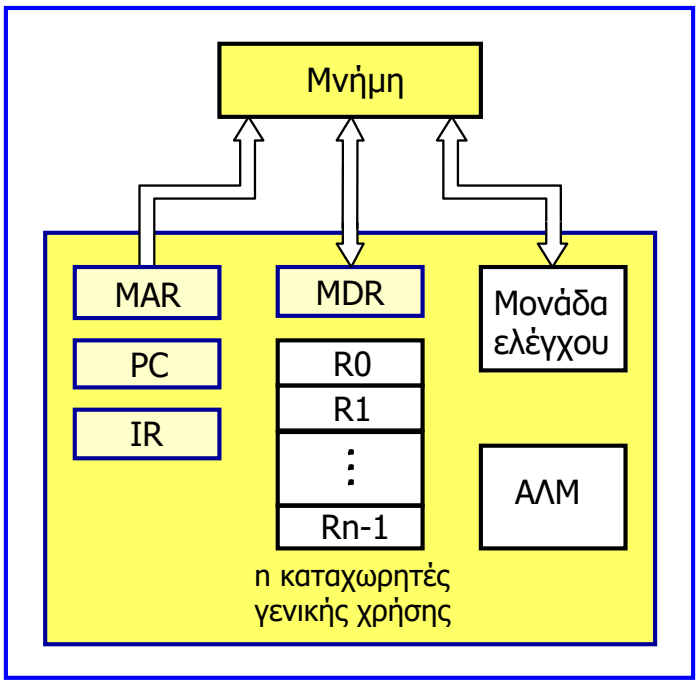
LOAD LOC, R1: Μεταφορά περιεχομένου θέσης μνήμης LOC στον R1

ADD R1,R0: Πρόσθεση περιεχομένων R0, R1 και τοποθέτηση αποτελέσματος στον R0.

- Οι μεταφορές μεταξύ μνήμης και επεξεργαστή ξεκινούν με την αποστολή από τον επεξεργαστή, της διεύθυνσης της θέσης μνήμης που πρόκειται να προσπελαστεί, στη μνήμη και την παραγωγή των κατάλληλων σημάτων ελέγχου.
- Εκτός των καταχωρητών (γενικής χρήσης) που χρησιμοποιούνται για την τοποθέτηση ορισμάτων ή αποτελεσμάτων, ο επεξεργαστής διαθέτει και έναν αριθμό καταχωρητών ειδικής χρήσης που εξυπηρετούν τη λειτουργία του.

Βασικές λειτουργίες του επεξεργαστή

- IR (καταχωρητής εντολών): περιέχει την εντολή που εκτελείται. Η έξοδός του είναι διαθέσιμη στη μονάδα ελέγχου, έτσι ώστε να συγχρονίσει τα μέρη που εμπλέκονται στην εκτέλεση.
- PC (μετρητής προγράμματος): περιέχει τη διεύθυνση της θέσης μνήμης της επόμενης εντολής.
- MAR (καταχωρητής διεύθυνσης μνήμης): περιέχει τη διεύθυνση της θέσης μνήμης που πρόκειται να προσπελαστεί.
- MDR (καταχωρητής περιεχομένου μνήμης): περιέχει τα δεδομένα που πρόκειται να αποθηκευτούν (εγγραφούν) στη θέση μνήμης ή εκείνα που ανακτώνται (διαβάζονται) από τη θέση μνήμης.



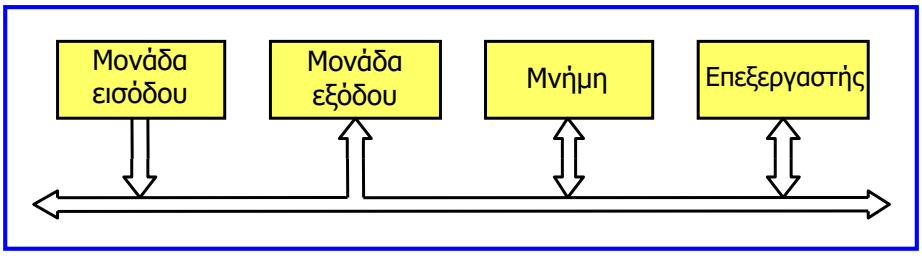
Εκτέλεση προγράμματος

- Έναρξη εκτέλεσης προγράμματος όταν ο PC περιέχει τη διεύθυνση της πρώτης εντολής.
- Το περιεχόμενο του PC μεταφέρεται στον MAR και ένα σήμα ελέγχου ανάγνωσης αποστέλλεται στη μνήμη, ώστε να ακολουθήσει η ανάγνωση της πρώτης εντολής.
- Η πρώτη εντολή προσκομίζεται από τη μνήμη, τοποθετείται στον MDR και στη συνέχεια στον IR.
- Η εντολή είναι έτοιμη να αποκωδικοποιηθεί και να εκτελεστεί.
- Εάν η εντολή εμπεριέχει λειτουργία που πρόκειται να εκτελεστεί στην ΑΛΜ, τότε πρέπει να γίνει παραλαβή των ορισμάτων που μπορεί να βρίσκονται στη μνήμη ή σε καταχωρητές γενικής χρήσης του επεξεργαστή.
- Τα ορίσματα που βρίσκονται στη μνήμη ανακτώνται και μεταφέρονται στην ΑΛΜ (με χρήση των καταχωρητών MAR, MDR).
- Η πράξη εκτελείται και αν το αποτέλεσμα πρόκειται να αποθηκευτεί στη μνήμη, αποθηκεύεται αρχικά στον MDR.
- Η διεύθυνση μνήμης της θέσης στην οποία πρόκειται να αποθηκευτεί το αποτέλεσμα, αποθηκεύεται στον MAR, ώστε να ακολουθήσει η εγγραφή.
- Κατά την εκτέλεση της εντολής, αυξάνεται το περιεχόμενο του PC, ώστε αυτός να περιέχει τη διεύθυνση της επόμενης εντολής.

Εκτέλεση προγράμματος

- Η εκτέλεση ενός προγράμματος μπορεί να διακοπεί, εάν κάποια μονάδα Ε/Ε απαιτήσει επείγουσα εξυπηρέτηση.
- Η μονάδα Ε/Ε συνήθως διαβιβάζει στον επεξεργαστή ένα σήμα διακοπής (interrupt), αιτούμενη εξυπηρέτηση από αυτόν.
- Ο επεξεργαστής παρέχει την απαιτούμενη εξυπηρέτηση εκτελώντας μια κατάλληλη ρουτίνα εξυπηρέτησης διακοπών (interrupt service routine, ISR) για τη μονάδα αυτή.
- Η κατάσταση (state) του επεξεργαστή (περιεχόμενα PC και καταχωρητών γενικής χρήσης, πληροφορίες ελέγχου) πρέπει να αποθηκευτεί στη μνήμη πριν την εξυπηρέτηση της διακοπής.
- Όταν η ISR εκτελεστεί, η κατάσταση του επεξεργαστή αποκαθίσταται, έτσι ώστε να συνεχιστεί η εκτέλεση του προγράμματος που διακόπηκε.

Δίαυλοι (buses)



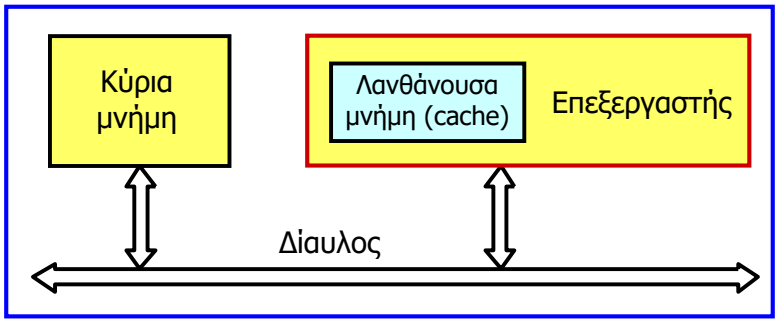
- Δίαυλος: σύνολο των γραμμών μεταφοράς δεδομένων (ενός bit ανά γραμμή) που λειτουργεί ως συνδετικό μονοπάτι για τις μονάδες του υπολογιστή.
- Εκτός από τη μεταφορά δεδομένων, οι δίαυλοι χρησιμεύουν για διευθυνσιοδότηση και έλεγχο.
- Ο απλός δίαυλος (που συνδέει όλες τις μονάδες) μπορεί να χρησιμοποιηθεί σε κάθε χρονική στιγμή από δύο μόνο μονάδες, αλλά διακρίνεται για το χαμηλό του κόστος και την ευχέρεια που προσφέρει στη σύνδεση περιφερειακών μονάδων.
- Καλύτερη απόδοση παρέχουν οι πολλαπλοί δίαυλοι αλλά με μεγαλύτερο κόστος.
- Στα μονάδες εισόδου-εξόδου ενσωματώνονται απομονωτές καταχωρητές (buffer registers), έτσι ώστε να μην περιορίζεται η ταχύτητα μεταφοράς από αργές μονάδες (π.χ. εκτυπωτής).

Λογισμικό συστήματος και λειτουργικό σύστημα

- Το λογισμικό συστήματος είναι μια συλλογή προγραμμάτων για:
  - ✓ Διαχείριση αποθήκευσης, ανάκτησης αρχείων από δευτερεύουσες μονάδες αποθήκευσης.
  - ✓ Έλεγχος μονάδων Ε/Ε (λήψη πληροφορίας εισόδου και παραγωγή αποτελεσμάτων).
  - ✓ Μετατροπή προγραμμάτων σε εντολές μηχανής (μεταγλωττιστής, compiler).
  - ✓ Επικοινωνία του υπολογιστή (πρωτόκολλα επικοινωνίας).
  - ✓ Δημιουργία των προγραμμάτων του χρήστη (γλώσσες προγραμματισμού, βιβλιοθήκες υποπρογραμμάτων).
  - ✓ Σύνδεση (linking) και εκτέλεση εντολών των προγραμμάτων του χρήστη με ήδη υπάρχουσες υπορουτίνες (π.χ. αριθμητικοί υπολογισμοί).
  - ✓ Ασφάλεια του υπολογιστή (προγράμματα antivirus).
- Απαραίτητο συστατικό του λογισμικού συστήματος αποτελεί το λειτουργικό σύστημα (ΛΣ, operating system, OS): συλλογή από ρουτίνες για έλεγχο ανάθεσης υπολογιστικών πόρων στα προγράμματα εφαρμογών (ανάθεση μνήμης και χώρου αποθήκευσης, μετακίνηση δεδομένων μεταξύ μνήμης και δίσκων, συντονισμός λειτουργιών εισόδου/εξόδου).
- Το ΛΣ διευθύνει την ταυτόχρονη εκτέλεση προγραμμάτων με στόχο την καλύτερη δυνατή χρήση των πόρων (πολυπρογραμματισμός ή πολυεπεξεργασία).

Απόδοση

- Η απόδοση (ταχύτητα εκτέλεσης προγράμματος) ενός υπολογιστή επηρεάζεται κυρίως από τη σχεδίαση υλικού, το μεταγλωττιστή και το σύνολο εντολών που διαθέτει.
- Χρόνος παρέλευσης (elapsed time): συνολικός χρόνος εκτέλεσης προγράμματος που επηρεάζεται από το σύνολο των μονάδων που συμμετέχουν και αποτελεί μέτρο απόδοσης ολόκληρου του υπολογιστικού συστήματος.
- Χρόνος επεξεργαστή (processor time): περιλαμβάνει τα χρονικά διαστήματα εκτέλεσης του προγράμματος, στα οποία ο επεξεργαστής είναι ενεργός.
- Όσον αφορά τη σχεδίαση, μια επιλογή για γρηγορότερη εκτέλεση ενός προγράμματος είναι η ελαχιστοποίηση της μετακίνησης εντολών και δεδομένων μεταξύ κύριας μνήμης και επεξεργαστή (χρήση λανθάνουσας μνήμης που παρέχει μεγαλύτερη ταχύτητα διακίνησης).



Απόδοση

- Ο επεξεργαστής ελέγχεται από σήμα ρολογιού, κάθε εντολή γλώσσας μηχανής διαιρείται σε μια ακολουθία βημάτων και κάθε βήμα εκτελείται σε έναν κύκλο ρολογιού.
- Διάρκεια κύκλου σήματος ρολογιού T, συχνότητα σήματος ρολογιού F = 1 / T (στο 2ο τόμο διδακτικού υλικού αναφέρεται ως R).
- N: αριθμός εντολών γλώσσας μηχανής ενός προγράμματος.
- CPI (cycles per instruction): μέσος αριθμός βημάτων (κύκλων ρολογιού) που απαιτείται για την εκτέλεση μιας εντολής ενός προγράμματος (στο 2ο τόμο αναφέρεται ως S).
- Χρόνος εκτέλεσης προγράμματος που περιλαμβάνει N εντολές:

$$t = (N \times CPI) / F$$ (βασική εξίσωση απόδοσης A: A = 1 / t)

- Μέσος αριθμός κύκλων ρολογιού ανά εντολή (CPI) προγράμματος:

$$CPI = \frac{CC}{N} = \frac{\sum_{i=1}^{n} (CPI_i \times N_i)}{N}$$

όπου CC (cycles count) είναι το πλήθος των κύκλων που απαιτείται για την εκτέλεση του προγράμματος, CPI_i είναι το πλήθος των κύκλων που απαιτείται για την εκτέλεση μιας εντολής τύπου i, N_i είναι ο αριθμός των εντολών τύπου i που συμμετέχουν στο πρόγραμμα.

Απόδοση

- Προφανώς, η διάρκεια κύκλου T (και η συχνότητα ρολογιού F = 1 / T) επηρεάζουν την απόδοση.
- Σημερινοί επεξεργαστές: T (ns), F (GHz).
- Ο αριθμός εντολών γλώσσας μηχανής ενός προγράμματος (N) μπορεί να μειωθεί με μεταγλώττιση του προγράμματος σε όσο το δυνατόν λιγότερες εντολές γλώσσας μηχανής.
- Ο μέσος αριθμός βημάτων ή κύκλων ρολογιού (CPI) εκτέλεσης μιας εντολής, μπορεί να μειωθεί με επικάλυψη της εκτέλεσης των εντολών (διασωλήνωση, pipelining).
- Μείωση των N και CPI ή αύξηση της συχνότητας ρολογιού (μέσω βελτίωσης τεχνολογίας κατασκευής κυκλωμάτων ή μείωσης του ποσού επεξεργασίας ανά βήμα) οδηγεί σε μείωση του χρόνου επεξεργασίας και επομένως σε αύξηση της απόδοσης.
- Οι παράμετροι N, CPI και F δεν είναι ανεξάρτητες, αφού αλλάζοντας μία από αυτές μπορεί να επηρεαστεί η τιμή κάποιας άλλης.
- Ενδεχόμενη αλλαγή στη σχεδίαση ενός επεξεργαστή μπορεί να οδηγήσει σε αυξημένη απόδοση, μόνο όταν το συνολικό αποτέλεσμα οδηγεί σε μείωση του χρόνου t.
- Ποιος επεξεργαστής από τους E1 και E2 έχει μεγαλύτερη απόδοση, εάν F_{E1} = 1.5 GHz και F_{E2} = 1.7 GHz; Δεν μπορούμε να απαντήσουμε με σιγουριά, εάν δεν έχουμε πληροφορία για τον CPI.

Απόδοση, νόμος Amdahl

- t: χρόνος εκτέλεσης ενός προγράμματος
- t_S: χρόνος εκτέλεσης του προγράμματος, όταν ένα μέρος του με συμμετοχή (αναλογία) p στον χρόνο εκτέλεσης του προγράμματος βελτιώνεται, με αποτέλεσμα τη μείωση του χρόνου εκτέλεσής του ή την αύξηση της απόδοσης, κατά έναν παράγοντα s.
- Η μείωση του χρόνου εκτέλεσης του προγράμματος μπορεί να οφείλεται στο σχεδιασμό του, στους βελτιωμένους πόρους που χρησιμοποιεί για την εκτέλεσή του, σε προηγμένο μεταγλωττιστή κ.ά.
- Ο συνολικός χρόνος εκτέλεσης του προγράμματος πριν τη βελτίωση, μπορεί να γραφτεί ως:

$$t = (1 - p) \times t + p \times t$$

- Ο χρόνος εκτέλεσης του προγράμματος μετά την βελτίωση είναι (θεωρητικό άνω όριο):

$$t_S = (1 - p) \times t + p \times (t / s)$$

- Μείωση του συνολικού χρόνου εκτέλεσης του προγράμματος ή αύξηση της απόδοσης (speedup), λόγω της βελτίωσης:

$$\text{speedup} = t / t_S = t / [(1 - p) \times t + (p / s) \times t] = 1 / [(1 - p) + p / s]$$

Απόδοση, νόμος Amdahl: παράδειγμα

Ένα πρόγραμμα αποτελείται από δύο μέρη Α και Β, που συμμετέχουν στο χρόνο εκτέλεσής του κατά 75% και 25%, αντίστοιχα. Εάν το μέρος Β του προγράμματος σχεδιαστεί έτσι ώστε να εκτελείται 5 φορές γρηγορότερα, ποια είναι η αύξηση της απόδοσης, όσον αφορά την εκτέλεση του συνολικού προγράμματος;

Χρόνος εκτέλεσης του προγράμματος μετά τη βελτίωση:

$t_S = (1 - p) \times t + p \times (t / s) = (1 - 0.25) \times t + 0.25 \times (t / 5) = 0.8 \times t,$

δηλαδή, ο νέος χρόνος εκτέλεσης θα είναι το 80% του αρχικού χρόνου εκτέλεσης.

Αύξηση απόδοσης προγράμματος μετά τη βελτίωση:

$speedup = t / t_S = t / (0.8 \times t) = 1.25$ ή

$speedup = 1 / [(1 - p) + p / s] = 1 / [(1 - 0.25) + 0.25 / 5] = 1.25,$

δηλαδή, ο χρόνος εκτέλεσης μειώνεται κατά έναν παράγοντα 1.25 ή η απόδοση αυξάνεται κατά έναν παράγοντα 1.25.

Διασωλήνωση και υπερβαθμωτή λειτουργία

- Διασωλήνωση (pipelining) είναι η τεχνική που χρησιμοποιείται για βελτίωση της απόδοσης ενός επεξεργαστή μέσω επικάλυψης της εκτέλεσης διαδοχικών εντολών.
- Στην ιδανική περίπτωση όπου όλες οι εντολές ενός προγράμματος μπορούν να επικαλυφθούν στο μέγιστο δυνατό βαθμό, η εκτέλεση μπορεί να εξελίσσεται με ρυθμό μιας εντολής ανά κύκλο ρολογιού (δηλαδή, η ενεργός τιμή του CPI γίνεται 1), ενώ η ανεξάρτητη εκτέλεση κάθε εντολής θα χρειαζόταν CPI κύκλους ρολογιού.
- Υπερβαθμωτή εκτέλεση (superscalar execution): Μεγαλύτερη αύξηση της απόδοσης επιτυγχάνεται με χρησιμοποίηση πολλαπλών μονάδων επεξεργασίας (δημιουργία παράλληλων μονοπατιών επεξεργασίας για την ταυτόχρονη εκτέλεση των εντολών).
- Με την παράλληλη εκτέλεση των εντολών θα πρέπει να διατηρείται η λογική ορθότητα του προγράμματος.
- Εάν σε κάθε παράλληλη μονάδα εφαρμόζεται διασωλήνωση, η ενεργός τιμή του CPI μπορεί να γίνει μικρότερη του 1, κατά τη διάρκεια εκτέλεσης ενός προγράμματος.

Σύνολο εντολών του επεξεργαστή

- Οι απλές εντολές απαιτούν μικρό αριθμό βημάτων για να εκτελεστούν, αλλά ένας επεξεργαστής με απλές εντολές χρειάζεται μεγάλο αριθμό εντολών για να εκτελέσει ένα πρόγραμμα (μεγάλο N, μικρό CPI).

$$t = (N \times CPI) / F$$

- Οι σύνθετες εντολές εμπεριέχουν μεγαλύτερο αριθμό βημάτων, αλλά ένας επεξεργαστής με εντολές που επιτελούν πιο σύνθετες πράξεις χρειάζεται λιγότερες εντολές για να εκτελέσει ένα πρόγραμμα (μικρό N, μεγάλο CPI).
- Οι σύνθετες εντολές σε συνδυασμό με διασωλήνωση θα μπορούσαν να οδηγήσουν στην υψηλότερη δυνατή απόδοση.
- Ωστόσο, η υλοποίηση αποδοτικής διασωλήνωσης σε επεξεργαστές με σύνθετες εντολές είναι πιο δύσκολη και αυτό είναι συχνά καθοριστικός παράγοντας.
- RISC (Reduced Instruction Set Computers): υπολογιστές μειωμένου συνόλου εντολών.
- CISC (Complex Instruction Set Computers): υπολογιστές πολύπλοκου συνόλου εντολών.
- Αρκετοί από τους σημερινούς επεξεργαστές χρησιμοποιούν στοιχεία και από τις δύο μεθοδολογίες σχεδιασμού.

Σύνολο εντολών του επεξεργαστή

- Αρχιτεκτονική Pentium (CISC):
  - ✓ Πλήθος εντολών: 235
  - ✓ Μορφή εντολών: μεταβλητή (8 έως 88 bits)
  - ✓ Πλήθος τρόπων διευθυνσιοδότησης: 11
  - ✓ Πλήθος καταχωρητών γενικού σκοπού: 8
- Αρχιτεκτονική MIPS I (RISC):
  - ✓ Πλήθος εντολών: 70
  - ✓ Μορφή εντολών: σταθερή (32 bits)
  - ✓ Πλήθος τρόπων διευθυνσιοδότησης: 1
  - ✓ Πλήθος καταχωρητών γενικού σκοπού: 64

Μεταγλωττιστής

- Μεταφράζει ένα πρόγραμμα που είναι γραμμένο σε γλώσσα προγραμματισμού υψηλού επιπέδου σε μια ακολουθία εντολών μηχανής.
- Ένας μεταγλωττιστής βελτιστοποίησης (optimizing compiler) χρησιμοποιεί τα χαρακτηριστικά του επεξεργαστή για να ελαττώσει το γινόμενο N × CPI, δηλαδή το συνολικό πλήθος κύκλων ρολογιού που απαιτείται για την εκτέλεση του προγράμματος.
- Συνεπώς, ο μεταγλωττιστής πρέπει να είναι στενά συνδεδεμένος με την αρχιτεκτονική του επεξεργαστή.
- Ο αριθμός των κύκλων εξαρτάται από την επιλογή των εντολών μηχανής, αλλά και από τη σειρά με την οποία αυτές εκτελούνται, συνεπώς ο μεταγλωττιστής επιλέγει και αναδιατάσσει τις εντολές για να επιτύχει καλύτερη απόδοση.
- Οι αναδιατάξεις εντολών δεν πρέπει να επηρεάζουν τη λογική ορθότητα του προγράμματος.

Μέτρηση απόδοσης

- Με στόχο την αντικειμενικότερη μέτρηση απόδοσης και για να είναι δυνατές οι συγκρίσεις μεταξύ επεξεργαστών, χρησιμοποιούνται μετροπρογράμματα (benchmark programs).
- Για μέτρηση της απόδοσης επεξεργαστών χρησιμοποιείται μια προσυμφωνημένη συλλογή μετροπρογραμμάτων (εφαρμογών) που επιλέγονται από τον οργανισμό SPEC (system performance evaluation corporation).
- Το πρόγραμμα μεταγλωττίζεται και εκτελείται στον υπό δοκιμή επεξεργαστή και σε έναν επεξεργαστή αναφοράς. Η πιο πρόσφατη (2017) συλλογή εφαρμογών περιλαμβάνει 43 εφαρμογές και ο επεξεργαστής αναφοράς είναι ο UltraSPARC IV 2.1GHz της Sun.
- Λόγος SPEC για κάθε μετροπρόγραμμα:

$$\text{Λόγος SPEC (i)} = \frac{\text{Χρόνος εκτέλεσης μετροπρογράμματος i στον επεξεργαστή αναφοράς}}{\text{Χρόνος εκτέλεσης μετροπρογράμματος i στον επεξεργαστή υπό δοκιμή}}$$

- Συνολικός λόγος SPEC (για το σύνολο n μετροπρογραμμάτων):

$$\text{Λόγος SPEC} = \left(\prod_{i=1}^{n} \text{SPEC(i)} \right)^{\frac{1}{n}}$$

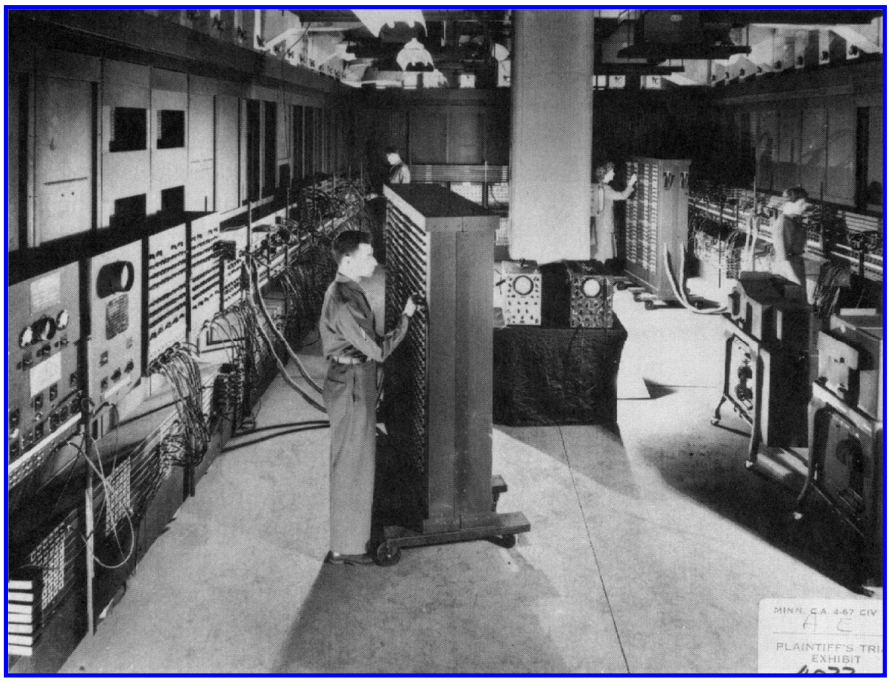
Πολυεπεξεργαστές και πολυ-υπολογιστές

- Πολυεπεξεργαστικά συστήματα: συστήματα υπολογιστών με πολλαπλές επεξεργαστικές μονάδες.
- Εκτελούν παράλληλα διαφορετικές εφαρμογές ή υποπρογράμματα μιας μεγάλης προγραμματιστικής διαδικασίας.
- Πολυεπεξεργαστικά συστήματα διαμοιρασμένης μνήμης (shared-memory): όλοι οι επεξεργαστές έχουν πρόσβαση στο σύνολο της μνήμης του συστήματος.
- Αυξημένη απόδοση, αλλά και αυξημένο κόστος και πολυπλοκότητα, λόγω των πολλαπλών επεξεργαστικών μονάδων, του μεγέθους της μνήμης, αλλά και των σύνθετων διασυνδέσεων.
- Εναλλακτικός τρόπος για υψηλή υπολογιστική ισχύ είναι η χρήση διασυνδεδεμένης ομάδας αυτόνομων υπολογιστών που διαθέτουν ο καθένας τη δική του μονάδα μνήμης.
- Ανταλλάσσουν δεδομένα με ανταλλαγή μηνυμάτων (message-passing multicomputers) σε αντίθεση με την τεχνική της διαμοιρασμένης μνήμης των πολυεπεξεργαστών.

Ιστορική αναδρομή

- 1η γενιά (1945-55): αρχιτεκτονική von Neumann, συμβολική γλώσσα, μετάφραση σε γλώσσα μηχανής για εκτέλεση, μνήμες υδραργύρου αλλά και μαγνητικές μνήμες, τεχνολογία λυχνιών για υλοποίηση λογικών πράξεων, αριθμητικές πράξεις σε χιλιοστά του δευτερολέπτου.
- 2η γενιά (1955-65): τεχνολογία τρανζίστορ, μαγνητικές μνήμες, γλώσσες προγραμματισμού υψηλού επιπέδου (Fortran), πρώτοι μεταγλωττιστές.
- 3η γενιά (1965-75): ολοκληρωμένα κυκλώματα (πολλαπλά τρανζίστορ σε ένα δισκίο πυριτίου), μεγαλύτερες ταχύτητες, μικρότερο κόστος, ημιαγωγικές μνήμες ολοκληρωμένων κυκλωμάτων, εμφάνιση μικροπρογραμματισμού, διασωλήνωσης, παράλληλης επεξεργασίας, λειτουργικά συστήματα, λανθάνουσες και ιδεατές μνήμες.
- 4η γενιά (1975-σήμερα): τεχνολογία VLSI (ολοκλήρωση πάρα πολύ υψηλής κλίμακας) για ακόμη υψηλότερη απόδοση, πλήρεις επεξεργαστές σε ένα ολοκληρωμένο κύκλωμα (μικροεπεξεργαστές), εξέλιξη διασωλήνωσης και παράλληλης επεξεργασίας (πολυπύρηνοι επεξεργαστές), λανθάνουσα και ιδεατή μνήμη, φορητοί υπολογιστές, ενσωματωμένα συστήματα, δίκτυα υπολογιστών, διαδίκτυο.
- Μετά την 4η γενιά: τεχνητή νοημοσύνη, κατανεμημένα υπολογιστικά συστήματα, ισχυρές παράλληλες υπολογιστικές μηχανές.

Ο πρώτος ηλεκτρονικός υπολογιστής (1946)



- ENIAC (Electronic Numerical Integrator And Computer)
- 20000 λυχνίες
- 7200 κρυσταλοδιόδους
- 1500 διακόπτες
- 70000 αντιστάσεις
- 10000 πυκνωτές
- 128 γραμμές καθυστέρησης υδραργύρου (χωρητικότητα 1024 λέξεων)
- 200 KW
- 167 m^2

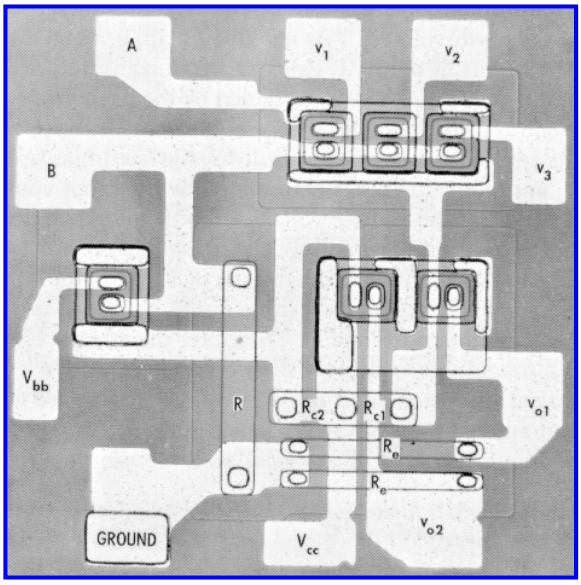
Χρησιμοποιήθηκε για τη σχεδίαση της ατομικής βόμβας...

Ο πρώτος εμπορικός ηλεκτρονικός υπολογιστής (1951)

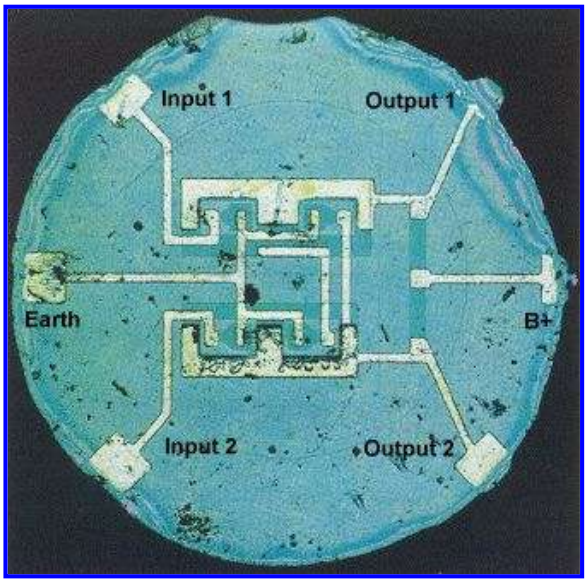
- UNIVAC I (UNIVersal Automatic Computer I)
- Ο πρώτος εμπορικός υπολογιστής γενικού σκοπού.
- Υλοποιήθηκαν 48 τέτοια συστήματα με κόστος 1 εκατ. $ το καθένα.
- Χρησιμοποιήθηκε για την πρόβλεψη του αποτελέσματος των Αμερικανικών εκλογών του 1952.

Τα πρώτα ολοκληρωμένα κυκλώματα (1961)

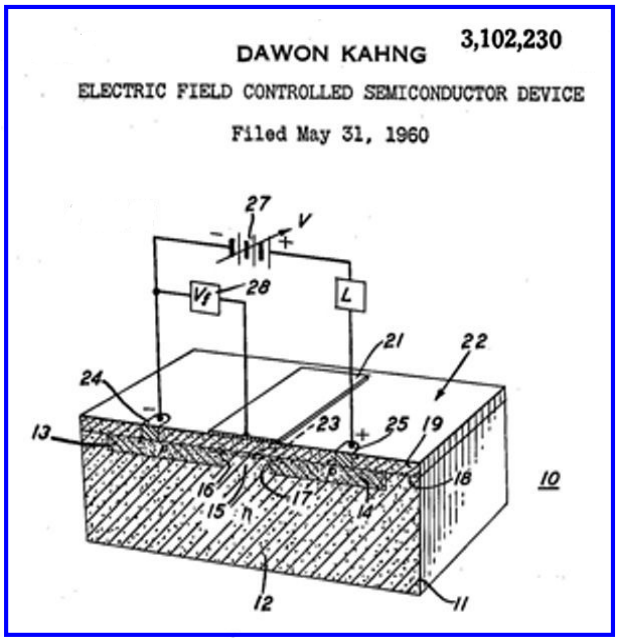
Πύλη διπολικής λογικής 3 εισόδων (Motorola)



Μανταλωτής RS (Fairchild) με 4 τρανζίστορ και 5 αντιστάσεις



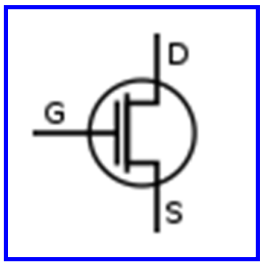
Η εφεύρεση της κατασκευής του MOSFET (1960)

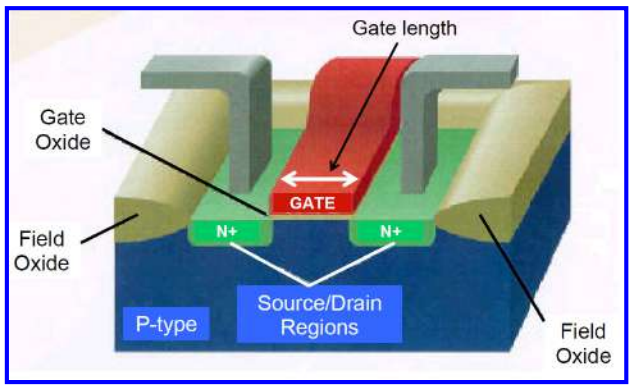

DAWON KAHNG 3,102,230

ELECTRIC FIELD CONTROLLED SEMICONDUCTOR DEVICE

Filed May 31, 1960

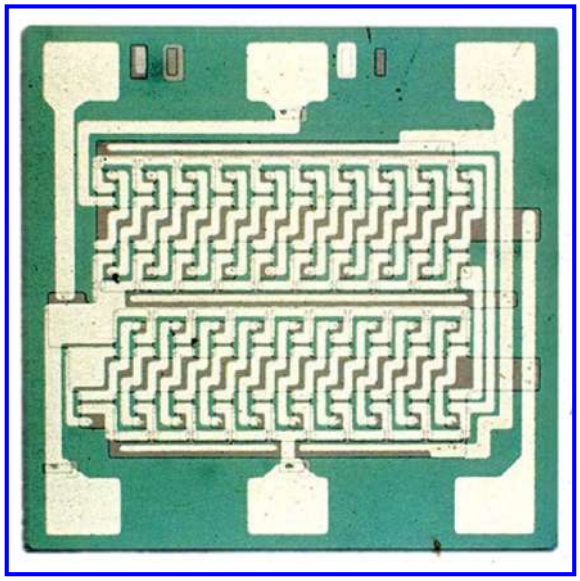
Bell Labs



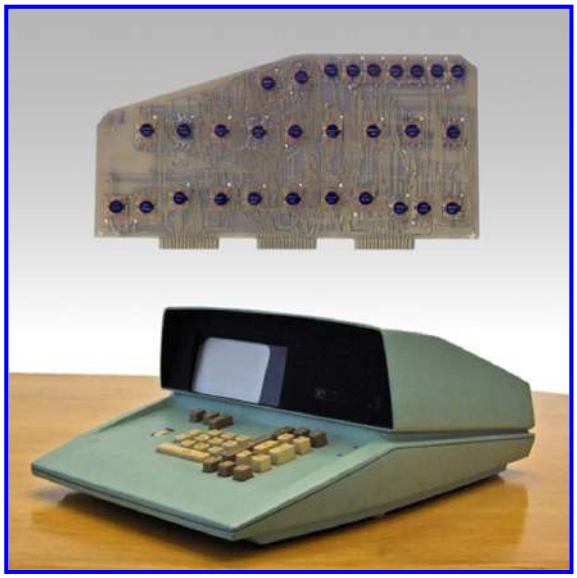


Τα πρώτα εμπορικά Ο.Κ. με MOSFETs (1964, 1965)

Το πρώτο εμπορικό ολοκληρωμένο κύκλωμα με MOSFETs: καταχωρητής ολίσθησης των 20-bit με 120 τρανζίστορ PMOS (General Microelectronics)



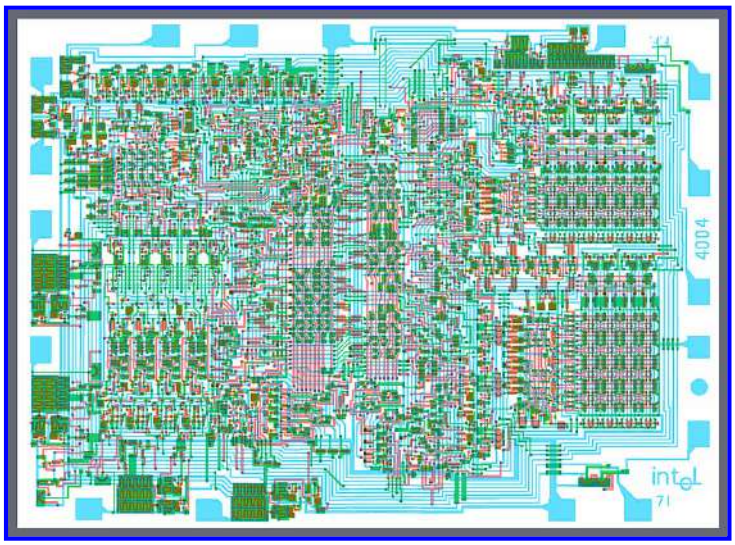
Η πρώτη ηλεκτρονική αριθμομηχανή με MOSFETs (Victor Comptometer) με 20 ολοκληρωμένα κυκλώματα και σειριακή μνήμη με 6 καταχωρητές ολίσθησης των 100-bit (General Microelectronics)



Μικροεπεξεργαστές

- Το αποτέλεσμα της εμφάνισης της τεχνολογίας ολοκληρωμένων κυκλωμάτων ήταν η ενσωμάτωση σε ένα μόνο ολοκληρωμένο κύκλωμα όλης της κεντρικής μονάδας επεξεργασίας ενός υπολογιστικού συστήματος, δηλαδή οι μικροεπεξεργαστές.
- Οι μικροεπεξεργαστές υποστηρίζονται από διάφορα είδη ολοκληρωμένων κυκλωμάτων, όπως κυκλώματα μνήμης (εκτός των μνημών που ενσωματώνονται στους μικροεπεξεργαστές), κυκλώματα διασύνδεσης με τις υπόλοιπες μονάδες του υπολογιστικού συστήματος, κυκλώματα ελέγχου, χρονισμού κ.ά.
- Η ανάπτυξη της τεχνολογίας των ολοκληρωμένων κυκλωμάτων τις τελευταίες δεκαετίες έδωσε τη δυνατότητα ενσωμάτωσης σε ένα ολοκληρωμένο κύκλωμα όλο και πιο πολύπλοκων κυκλωμάτων.
- Ξεκινώντας από τους μικροεπεξεργαστές της δεκαετίας του 1970 που περιλάμβαναν μερικές χιλιάδες τρανζίστορς έχουμε φτάσει σήμερα σε μικροεπεξεργαστές που περιλαμβάνουν μερικά δισεκατομμύρια τρανζίστορς.
- Αυτό έχει ως αποτέλεσμα την ταχύτατη ανάπτυξη των μικροεπεξεργαστών και την μεγάλη αύξηση της χρήσης τους τόσο σε πολύπλοκες υπολογιστικές μηχανές, όσο και σε συστήματα καθημερινής χρήσης (προσωπικούς και φορητούς υπολογιστές, ταμπλέτες, έξυπνα κινητά τηλέφωνα, ηλεκτρονικές συσκευές διασκέδασης κ.ά.).

Μικροεπεξεργαστής Intel 4004 (1971)



Τεχνολογία κατασκευής: 10 μm, NMOS

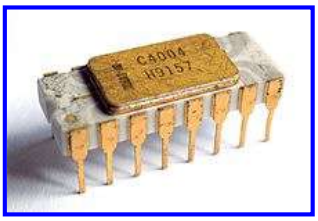
Επιφάνεια: 12 mm^2

Πλήθος τρανζίστορς: 2300

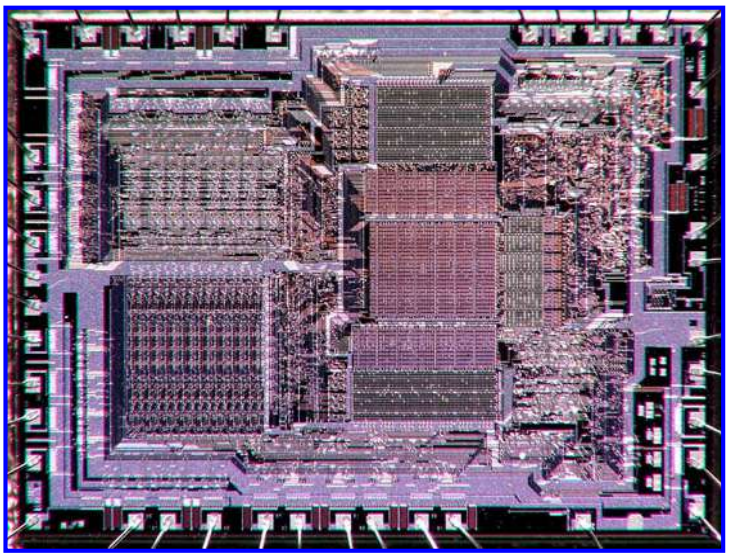
Συχνότητα λειτουργίας: 108 KHz

Η τεχνολογία κατασκευής αναφέρεται στο ελάχιστο μήκος καναλιού (L) ενός τρανζίστορ

Ένας από τους πρώτους εμπορικούς μικροεπεξεργαστές



Μικροεπεξεργαστής Intel 8085 (1976)



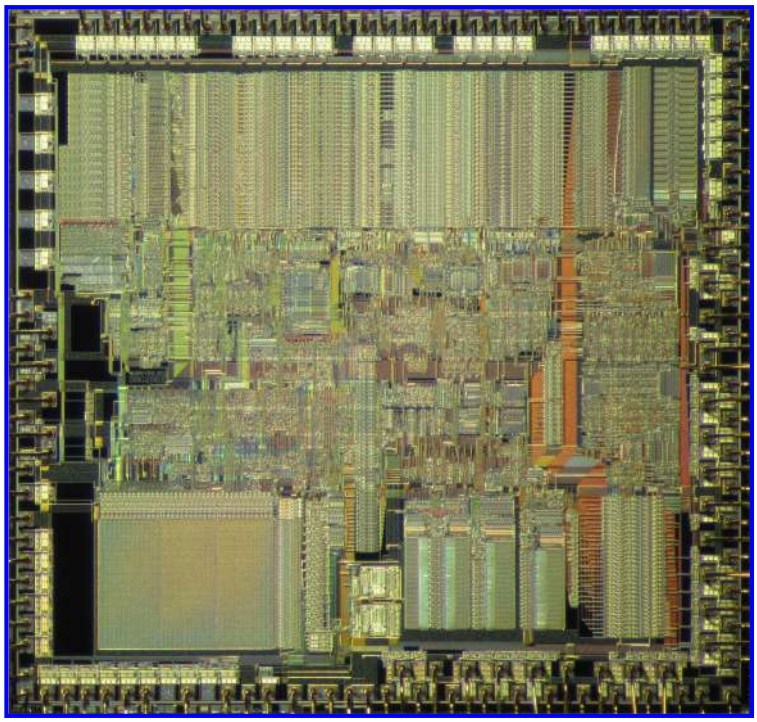
Τεχνολογία κατασκευής: 3 μm, NMOS

Επιφάνεια: 20 mm^2

Πλήθος τρανζίστορς: 6500

Συχνότητα λειτουργίας: 3 MHz

Μικροεπεξεργαστής Intel 80386 (1985)



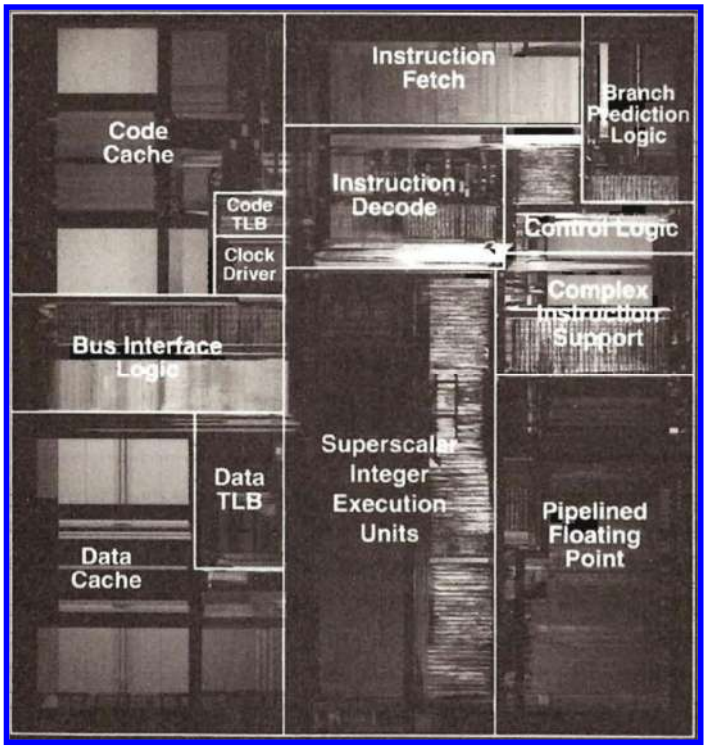
Τεχνολογία κατασκευής: 1.5 μm CMOS

Επιφάνεια: 104 mm^2

Πλήθος τρανζίστορς: 275000

Συχνότητα λειτουργίας: 16 MHz

Μικροεπεξεργαστής Intel Pentium (1993)



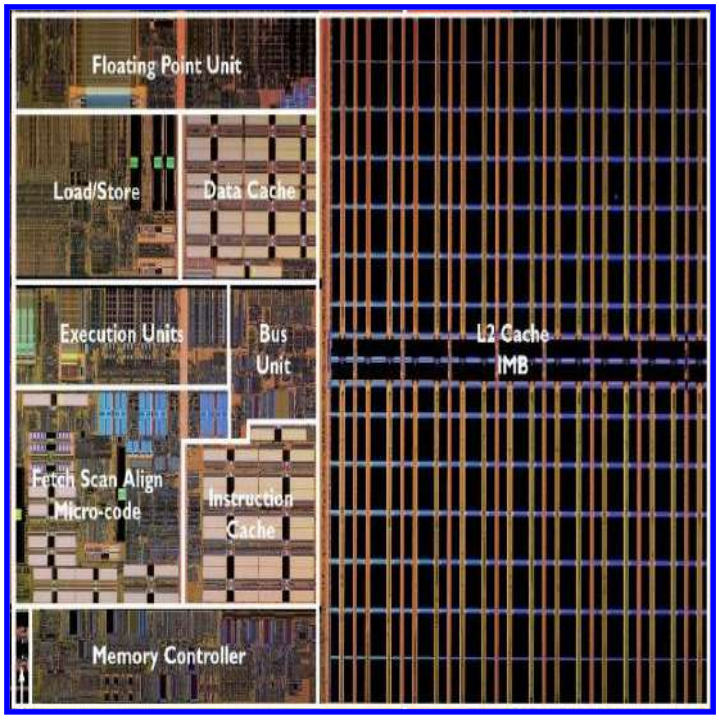
Τεχνολογία κατασκευής: 0.8 μm

Επιφάνεια: 294 mm^2

Πλήθος τρανζίστορς: 3.1 εκατομμύρια

Συχνότητα λειτουργίας: 66 MHz

Μικροεπεξεργαστής AMD Athlon 64 (2004)



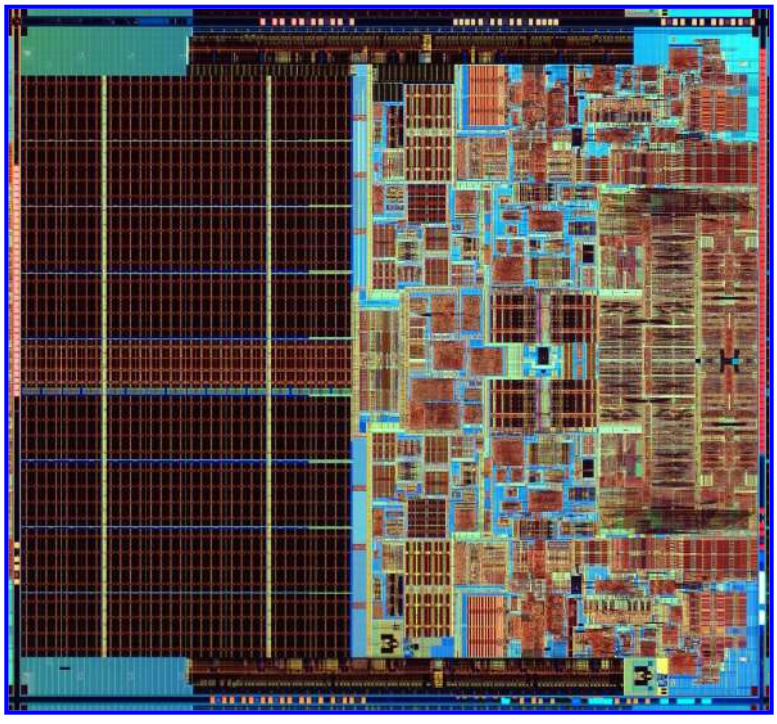
Τεχνολογία κατασκευής: 0.13 μm

Επιφάνεια: 193 mm^2

Πλήθος τρανζίστορς: 106 εκατομμύρια

Συχνότητα λειτουργίας: 1.6 GHz

Διπύρηνος μικροεπεξεργαστής Intel Core 2 Duo (2007)



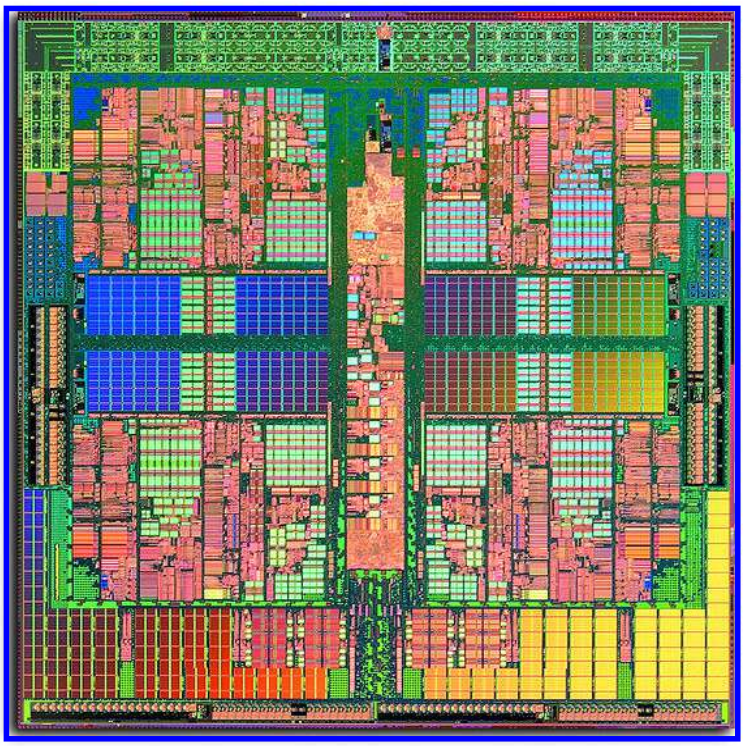
Τεχνολογία κατασκευής: 65 nm

Επιφάνεια: 111 mm^2

Πλήθος τρανζίστορς: 169 εκατομμύρια

Συχνότητα λειτουργίας: 1.8 GHz

Τετραπύρηνος μικροεπεξεργαστής AMD Opteron (2008)



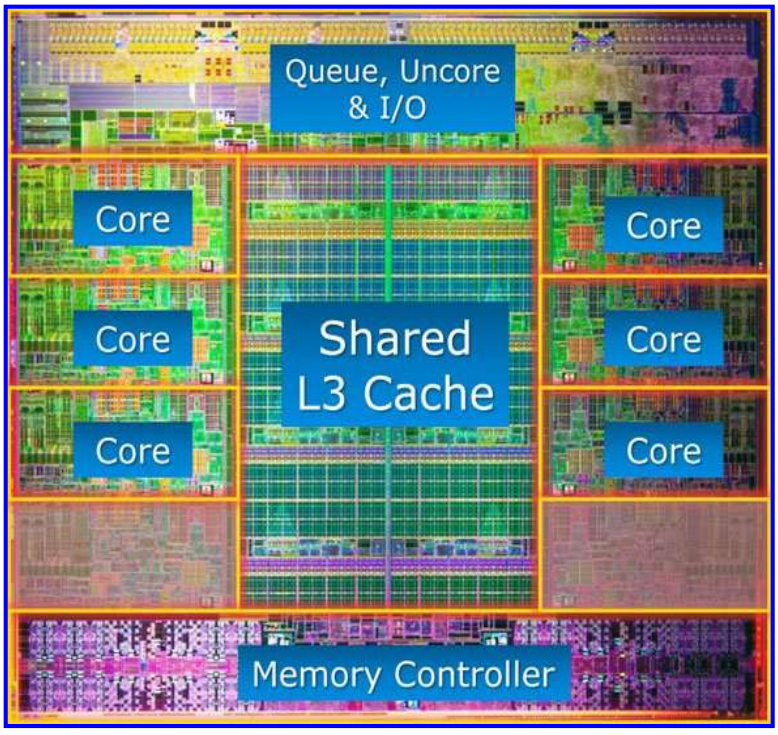
Τεχνολογία κατασκευής: 65 nm

Επιφάνεια: 283 mm^2

Πλήθος τρανζίστορς:
463 εκατομμύρια

Συχνότητα λειτουργίας: 2 GHz

Εξαπύρηνος μικροεπεξεργαστής Intel Core i7 (2011)



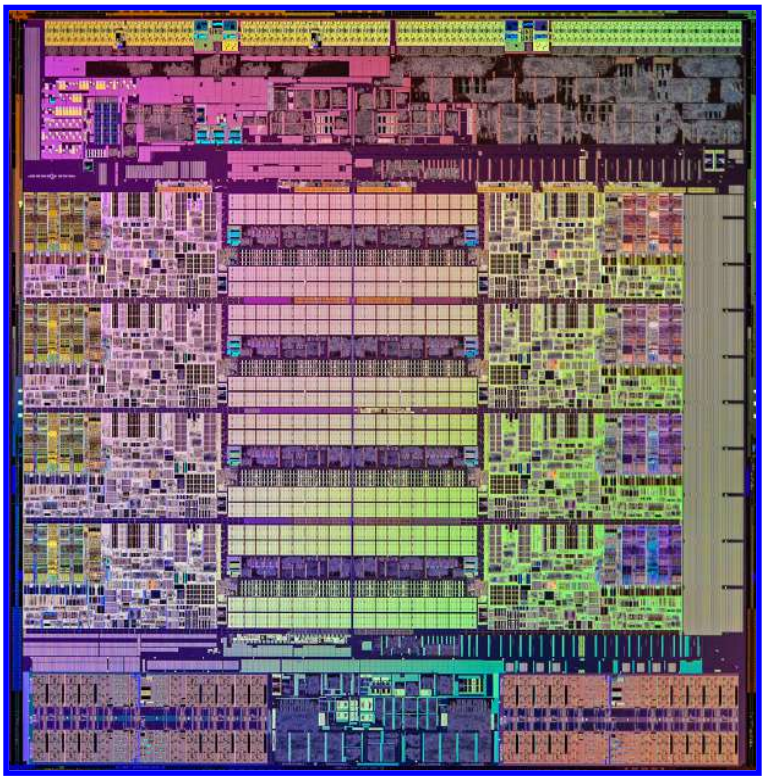
Τεχνολογία κατασκευής: 32 nm

Επιφάνεια: 240 mm^2

Πλήθος τρανζίστορς:
1.17 δισεκατομμύρια

Συχνότητα λειτουργίας: 3.3 GHz

Οκταπύρηνος μικροεπεξεργαστής Intel Core i7 (2014)



Τεχνολογία κατασκευής: 22 nm

Επιφάνεια: 355 mm

Πλήθος τρανζίστορς:
2.6 δισεκατομμύρια

Συχνότητα λειτουργίας: 3.6 GHz

Η εξέλιξη των μικροεπεξεργαστών συνεχίζεται...

- Την διετία 2017-2018, η Intel ανέπτυξε τη σειρά μικροεπεξεργαστών Core i9 σε τεχνολογία κατασκευής 14 nm, που περιλαμβάνουν από 8 έως 14 πυρήνες με περισσότερα από 5 δισεκατομμύρια τρανζίστορς.

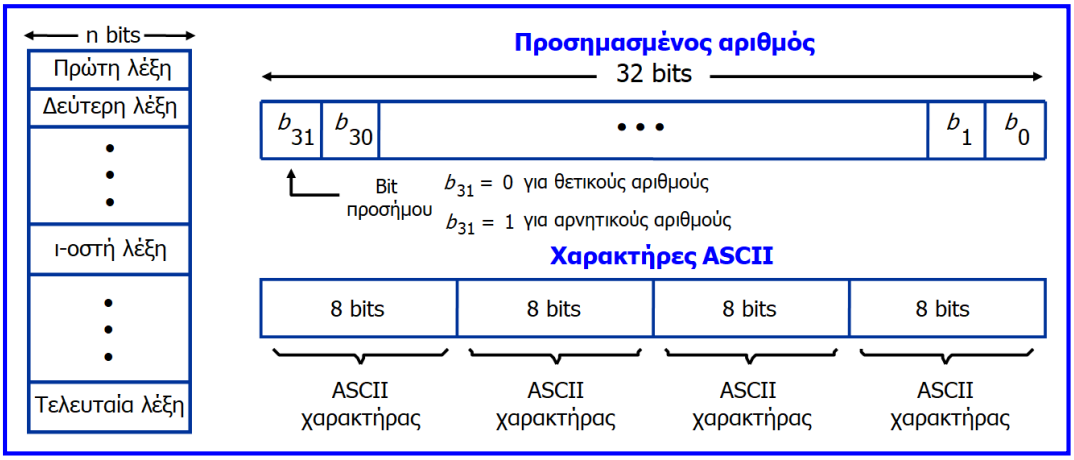
- Το 2020 η Intel ανέπτυξε την 11η γενιά μικροεπεξεργαστών της σε τεχνολογία κατασκευής 10 nm, υποστηρίζοντας ότι η τεχνολογία αυτή παρέχει 25% καλύτερη επίδοση και 45% χαμηλότερη κατανάλωση ενέργειας, καθώς επίσης και ότι με την τεχνολογία αυτή μπορούν να αναπτυχθούν 100 εκατ. τρανζίστορ σε επιφάνεια πυριτίου ενός τετραγωνικού χιλιοστού.

- Την διετία 2017-2018, η AMD ανέπτυξε τη σειρά μικροεπεξεργαστών Ryzen σε τεχνολογία 14 nm, που περιλαμβάνουν έως 16 πυρήνες, ενώ πρόσφατα ανέπτυξε την 3η γενιά μικροεπεξεργαστών Ryzen σε τεχνολογία 7 nm που περιλαμβάνει έως 32 πυρήνες, βελτιώνοντας τις επιδόσεις των μικροεπεξεργαστών της.

- Επίσης, τα τελευταία χρόνια, η AMD ανέπτυξε τη σειρά μικροεπεξεργαστών Epic 7000 (multi-chip modules) που περιλαμβάνει 4 οκταπύρηνους επεξεργαστές με περισσότερα από 19 δισεκατομμύρια τρανζίστορς.

Οργάνωση μνήμης

- Η μνήμη αποτελείται από πολλαπλά κελιά αποθήκευσης (storage cells) καθένα από τα οποία αποθηκεύει ένα ψηφίο (bit) πληροφορίας (0 ή 1).
- Η μνήμη οργανώνεται σε λέξεις (words) με μήκος n ψηφίων (word length).
- Οι σύγχρονοι υπολογιστές έχουν συνήθως μήκη λέξεων 16 - 64 bits (8 bits = 1 byte) και οι λέξεις μπορεί να παριστάνουν αριθμούς ή χαρακτήρες.

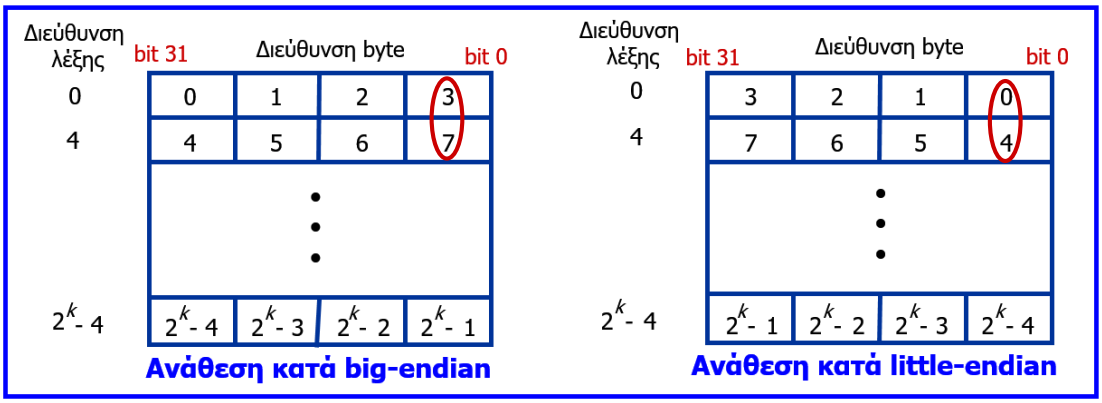


Οργάνωση μνήμης

- Για τον προσδιορισμό των θέσεων μνήμης για αποθήκευση ή ανάκτηση μιας ποσότητας πληροφορίας (λέξης ή byte) απαιτείται η χρήση διευθύνσεων.
- Για τον καθορισμό 2^k διευθύνσεων διαδοχικών θέσεων (χώρος διευθύνσεων, address space) μιας μνήμης χρησιμοποιούμε αριθμούς από 0 έως $2^k - 1$ (π.χ. διευθύνσεις των 32 bits παράγουν χώρο διευθύνσεων 2^{32} ή 4G θέσεων, 1G = 2^{30}).
- Υπάρχουν τρεις βασικές ποσότητες πληροφορίας: ψηφίο (bit), byte, λέξη (word). Το byte έχει σταθερό μήκος (8 bits), ενώ το μήκος λέξης ποικίλει στους επεξεργαστές (συνήθως από 16 έως 64 bits).
- Η ανάθεση ξεχωριστών διευθύνσεων σε κάθε θέση bit της μνήμης δεν είναι πρακτική, συνεπώς η πιο πρακτική ανάθεση προκύπτει από την ανάθεση διαδοχικών διευθύνσεων σε διαδοχικές θέσεις bytes της μνήμης.
- Έτσι, στους περισσότερους σύγχρονους επεξεργαστές, η ανάθεση ξεχωριστών διευθύνσεων γίνεται σε διαδοχικές θέσεις bytes στη μνήμη και όχι σε διαδοχικές λέξεις (μνήμη διευθυνσιοδοτούμενη κατά byte).
- Οι θέσεις των bytes έχουν διευθύνσεις 0, 1, 2 κ.ο.κ. και εάν για παράδειγμα το μήκος λέξης που χρησιμοποιεί ο επεξεργαστής είναι 32 bits, οι διαδοχικές λέξεις βρίσκονται στις διευθύνσεις 0, 4, 8 κ.ο.κ. με κάθε λέξη να αποτελείται από 4 bytes.

Οργάνωση μνήμης

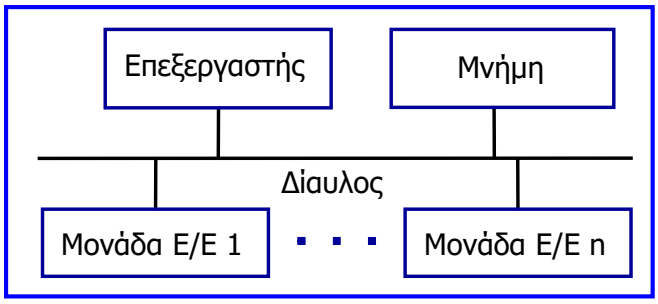
- Υπάρχουν δύο τρόποι ανάθεσης (διάταξης) λέξεων στη μνήμη: στην ανάθεση κατά big-endian το πιο σημαντικό byte της λέξης αποθηκεύεται στη λιγότερο σημαντική θέση byte της μνήμης, ενώ στην ανάθεση κατά little-endian το λιγότερο σημαντικό byte της λέξης αποθηκεύεται στη λιγότερο σημαντική θέση byte της μνήμης.
- Δεν υπάρχει σημαντική αιτία επιλογής ενός από τους δύο τρόπους και η επιλογή αποτελεί περισσότερο ζήτημα παράδοσης (Intel, ARM: προεπιλεγμένη ανάθεση little-endian, Motorola: big-endian).
- Και στους δύο τρόπους, οι διευθύνσεις των bytes 0, 4, 8 κ.ο.κ. θεωρούνται και χρησιμοποιούνται ως οι διευθύνσεις των διαδοχικών λέξεων 32 bits στη μνήμη.



Λειτουργίες που σχετίζονται με τη μνήμη

- Οι εντολές και τα δεδομένα (ορίσματα και αποτελέσματα) αποθηκεύονται στη μνήμη.
- Για να εκτελεστεί μια εντολή, τα κυκλώματα ελέγχου πρέπει να προκαλέσουν προσκόμιση της λέξης που περιέχει την εντολή αυτή από τη μνήμη. Ορίσματα και αποτελέσματα πρέπει επίσης να μεταφερθούν μεταξύ επεξεργαστή και μνήμης.
- Δύο βασικές λειτουργίες εμπλέκουν τη μνήμη: φόρτωση ή ανάγνωση ή προσκόμιση (load ή read ή fetch) και αποθήκευση ή εγγραφή (store ή write), με τις αντίστοιχες εντολές να αναφέρονται ως LOAD, STORE ή MOVE.
- Ανάγνωση: μεταφορά αντιγράφου περιεχομένου μιας θέσης μνήμης στον επεξεργαστή, με τα περιεχόμενα της μνήμης να παραμένουν αναλλοίωτα. Ο επεξεργαστής παρέχει τη διεύθυνση της επιθυμητής θέσης στη μνήμη, η μνήμη διαβάζει τα δεδομένα της θέσης αυτής και τα μεταφέρει στον επεξεργαστή.
- Εγγραφή: μεταφορά πληροφορίας από τον επεξεργαστή σε μια θέση μνήμης, διαγράφοντας το προηγούμενο περιεχόμενο της θέσης αυτής. Ο επεξεργαστής παρέχει τη διεύθυνση της επιθυμητής θέσης στη μνήμη μαζί με τα δεδομένα που πρόκειται να αποθηκευτούν σε αυτή.
- Με μία λειτουργία ανάγνωσης ή εγγραφής μπορεί να μεταφερθεί 1 λέξη ή 1 byte.

Προσπέλαση μονάδων εισόδου/εξόδου



- Για την προσπέλαση περιφερειακών μονάδων (μονάδες εισόδου/εξόδου ή Ε/Ε), δηλαδή για ανταλλαγή δεδομένων με μονάδες Ε/Ε, χρησιμοποιείται συνήθως η τεχνική που αναφέρεται ως Ε/Ε απεικονιζόμενη στη μνήμη (memory-mapped I/O), στην οποία ορισμένες διευθύνσεις μνήμης χρησιμοποιούνται για αναφορά σε καταχωρητές μονάδων Ε/Ε, ώστε να μην απαιτούνται ειδικές εντολές για την προσπέλασή τους και να αρκούν εντολές όπως οι εντολές Move, Load, Store κ.ά.
- Ωστόσο, κάποιοι επεξεργαστές διαθέτουν και ειδικές εντολές εισόδου και εξόδου (π.χ. IN, OUT) για να εκτελούν μεταφορές Ε/Ε και ξεχωριστό χώρο διευθύνσεων για τις μονάδες Ε/Ε (I/O-mapped I/O ή peripheral-mapped I/O).
- Κατάλληλα σήματα ελέγχου στο δίαυλο, υποδεικνύουν ότι μία αιτούμενη μεταφορά αποτελεί λειτουργία Ε/Ε και τότε η μονάδα μνήμης αγνοεί το σχετικό αίτημα.

Βασικοί τύποι εντολών

- Κάθε υπολογιστής πρέπει να διαθέτει εντολές ικανές να εκτελέσουν 4 τύπους λειτουργιών: μεταφορά δεδομένων μεταξύ μνήμης και καταχωρητών του επεξεργαστή, αριθμητικές και λογικές πράξεις σε δεδομένα, έλεγχο ακολουθίας εκτέλεσης εντολών (εντολές διακλάδωσης), μεταφορά δεδομένων από και προς μονάδες εισόδου/εξόδου (Ε/Ε).
- Συμβολισμοί:
  - ✓ LOC: διεύθυνση θέσης μνήμης
  - ✓ [LOC]: περιεχόμενο θέσης μνήμης με διεύθυνση LOC.
  - ✓ Ri: καταχωρητής του επεξεργαστή, [Ri]: περιεχόμενο καταχωρητή του επεξεργαστή
- Συμβολισμοί για μεταφορά ή εκτέλεση πράξης (π.χ. πρόσθεσης) και μεταφορά δεδομένων:
  - ✓ R1 ← [LOC]
  - ✓ R3 ← [R1] + [R2]
- Συμβολική γλώσσα Assembly:
  - ✓ MOVE LOC, R1
  - ✓ ADD R1, R2, R3

Στη γλώσσα Assembly συχνότερα αναφέρεται πρώτος στη σειρά ο προορισμός και στη συνέχεια η πηγή (οι πηγές) των δεδομένων.

Ταξινόμηση υπολογιστών βάσει του συνόλου εντολών

- Οι αρχιτεκτονικές υπολογιστών σε επίπεδο γλώσσας μηχανής ταξινομούνται σε:
  - ✓ αρχιτεκτονικές συσσωρευτή (accumulator architectures),
  - ✓ αρχιτεκτονικές καταχωρητών γενικού σκοπού (general purpose register architectures).
  - ✓ αρχιτεκτονικές μηχανισμού στοίβας (stack architectures),
- Οι αρχιτεκτονικές καταχωρητών γενικού σκοπού διακρίνονται σε:
  - ✓ αρχιτεκτονικές καταχωρητή-μνήμης (register-memory architectures),
  - ✓ και αρχιτεκτονικές καταχωρητή-καταχωρητή (register-register architectures).

Αρχιτεκτονική συσσωρευτή

- Οι αρχιτεκτονικές συσσωρευτή βασίζονται στη χρήση ενός καταχωρητή που αναφέρεται ως συσσωρευτής (accumulator) και συνήθως οι εντολές τους περιλαμβάνουν μία διεύθυνση μνήμης όπου βρίσκεται το ένα όρισμα, ενώ υπονοείται ότι το άλλο όρισμα βρίσκεται στον συσσωρευτή. Το αποτέλεσμα κάθε πράξης αποθηκεύεται στο συσσωρευτή.
- Για την αντιγραφή του περιεχομένου μιας θέσης κύριας μνήμης στον συσσωρευτή χρησιμοποιείται η εντολή LOAD, ενώ για την αντιγραφή του περιεχομένου του συσσωρευτή σε μία θέση κύριας μνήμης χρησιμοποιείται η εντολή STORE.

Παράδειγμα: Η εκτέλεση της πράξης D = B • C – A απαιτεί 4 εντολές μίας διεύθυνσης

LOAD B	Μεταφορά (αντιγραφή) στον συσσωρευτή του περιεχομένου της θέσης μνήμης B
MUL C	Πολλαπλασιασμός του περιεχομένου του συσσωρευτή με το περιεχόμενο της θέσης μνήμης C και αποθήκευση του γινομένου στον συσσωρευτή
SUB A	Αφαίρεση του περιεχομένου της θέσης μνήμης A από το γινόμενο που είναι αποθηκευμένο στον συσσωρευτή και αποθήκευση της διαφοράς στον συσσωρευτή
STORE D	Μεταφορά (αντιγραφή) του τελικού αποτελέσματος στη θέση μνήμης D.

Αρχιτεκτονική καταχωρητή-μνήμης

- Στις εντολές πράξεων των αρχιτεκτονικών καταχωρητή-μνήμης το ένα όρισμα είναι αποθηκευμένο σε μία διεύθυνση κύριας μνήμης, ενώ το άλλο όρισμα βρίσκεται σε έναν καταχωρητή γενικού σκοπού.
- Για την αντιγραφή του περιεχομένου μιας θέσης κύριας μνήμης σε έναν καταχωρητή γενικού σκοπού χρησιμοποιείται η εντολή LOAD, ενώ για την αντιγραφή του περιεχομένου καταχωρητή γενικού σκοπού σε μία θέση κύριας μνήμης χρησιμοποιείται η εντολή STORE.

Παράδειγμα: Η εκτέλεση της πράξης D = B • C – A απαιτεί 4 εντολές

LOAD R1, B	Μεταφορά στον καταχωρητή R1 του περιεχομένου της θέσης μνήμης B
MUL R1, C	Πολλαπλασιασμός του περιεχομένου του καταχωρητή R1 με το περιεχόμενο της θέσης μνήμης C και αποθήκευση του γινομένου στον καταχωρητή R1
SUB R1, A	Αφαίρεση του περιεχομένου της θέσης μνήμης A από το γινόμενο που είναι αποθηκευμένο στον καταχωρητή R1 και αποθήκευση της διαφοράς στον καταχωρητή R1
STORE D, R1	Μεταφορά του τελικού αποτελέσματος στη θέση μνήμης D

Αρχιτεκτονική καταχωρητή-καταχωρητή

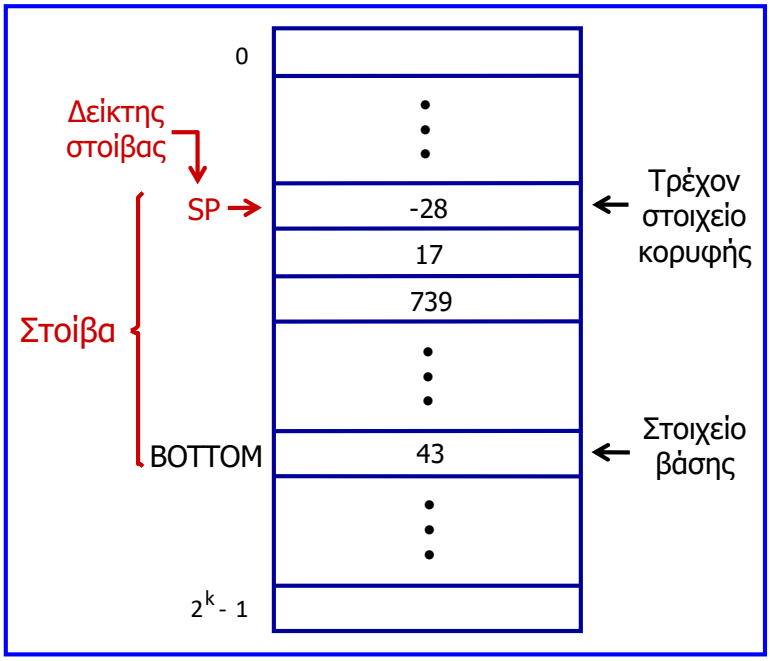
- Στις εντολές πράξεων των αρχιτεκτονικών καταχωρητή-καταχωρητή δηλώνονται 3 ή 2 καταχωρητές γενικού σκοπού, από τους οποίους οι δύο περιέχουν τα ορίσματα και ο τρίτος (ή ο δεύτερος) χρησιμοποιείται για την αποθήκευση του αποτελέσματος.
- Για την αντιγραφή των περιεχομένων της κύριας μνήμης σε καταχωρητές γενικού σκοπού χρησιμοποιείται η εντολή LOAD, ενώ για την αντιγραφή των περιεχομένων των καταχωρητών γενικού σκοπού στην κύρια μνήμη χρησιμοποιείται η εντολή STORE.
- Η εκτέλεση πράξεων σε ορίσματα που είναι αποθηκευμένα σε καταχωρητές γενικού σκοπού είναι πιο γρήγορη από την εκτέλεση πράξεων με χρήση της μνήμης.

Παράδειγμα: Η εκτέλεση της πράξης D = B • C – A απαιτεί 6 εντολές

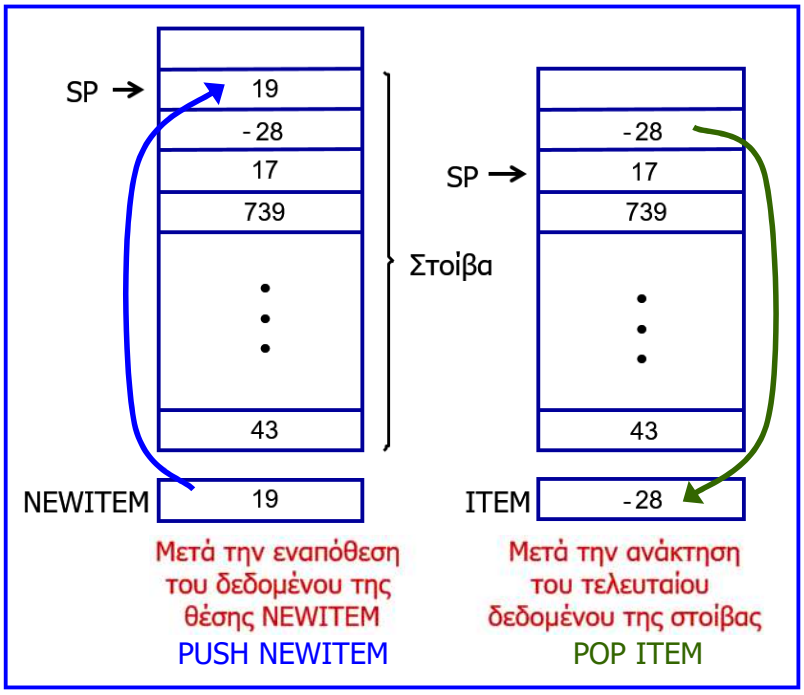
LOAD R1, A	Μεταφορά στον καταχωρητή R1 του περιεχομένου της θέσης μνήμης A
LOAD R2, B	Μεταφορά στον καταχωρητή R2 του περιεχομένου της θέσης μνήμης B
LOAD R3, C	Μεταφορά στον καταχωρητή R3 του περιεχομένου της θέσης μνήμης C
MUL R4, R2, R3 (ή MUL R2, R3)	Πολλαπλασιασμός του περιεχομένου των καταχωρητών R2 και R3 και αποθήκευση του γινομένου στον καταχωρητή R4 (ή στον R2)
SUB R4, R4, R1 (ή SUB R2, R1)	Αφαίρεση του περιεχομένου του καταχωρητή R1 από το περιεχόμενο του καταχωρητή R4 (ή R2) & αποθήκευση της διαφοράς στον R4 (ή R2)
STORE D, R4 (ή R2)	Μεταφορά του αποτελέσματος από τον R4 (ή R2) στη θέση μνήμης D

Οργάνωση δεδομένων σε στοίβα

- Στοίβα (stack) είναι μία δομή δεδομένων στη μνήμη του υπολογιστή που χρησιμοποιείται για την υποστήριξη των πράξεων στις αρχιτεκτονικές μηχανισμού στοίβας και για την επικοινωνία μεταξύ του κυρίου προγράμματος και μιας υπορουτίνας.
- Ένας καταχωρητής του επεξεργαστή χρησιμοποιείται ως δείκτης στοίβας (stack pointer) που περιέχει τη διεύθυνση κορυφής της στοίβας σε κάθε χρονική στιγμή.
- Εναπόθεση (PUSH) νέου δεδομένου στη στοίβα.
- Ανάκτηση (POP) του τελευταίου δεδομένου από τη στοίβα.



Οργάνωση δεδομένων σε στοίβα



Αρχιτεκτονική μηχανισμού στοίβας

Στις αρχιτεκτονικές μηχανισμού στοίβας:

- Τα δεδομένα που απαιτούνται για την εκτέλεση μιας πράξης μεταφέρονται αρχικά από την κύρια μνήμη στις κορυφαίες θέσεις της στοίβας, με χρήση της εντολής PUSH.
- Από τη στοίβα μεταφέρονται στην αριθμητική λογική μονάδα για την εκτέλεση της επιθυμητής πράξης.
- Το αποτέλεσμα αποθηκεύεται στην κορυφή της στοίβας.
- Από τη στοίβα μεταφέρεται στην κύρια μνήμη με χρήση της εντολής POP.

Αρχιτεκτονική μηχανισμού στοίβας

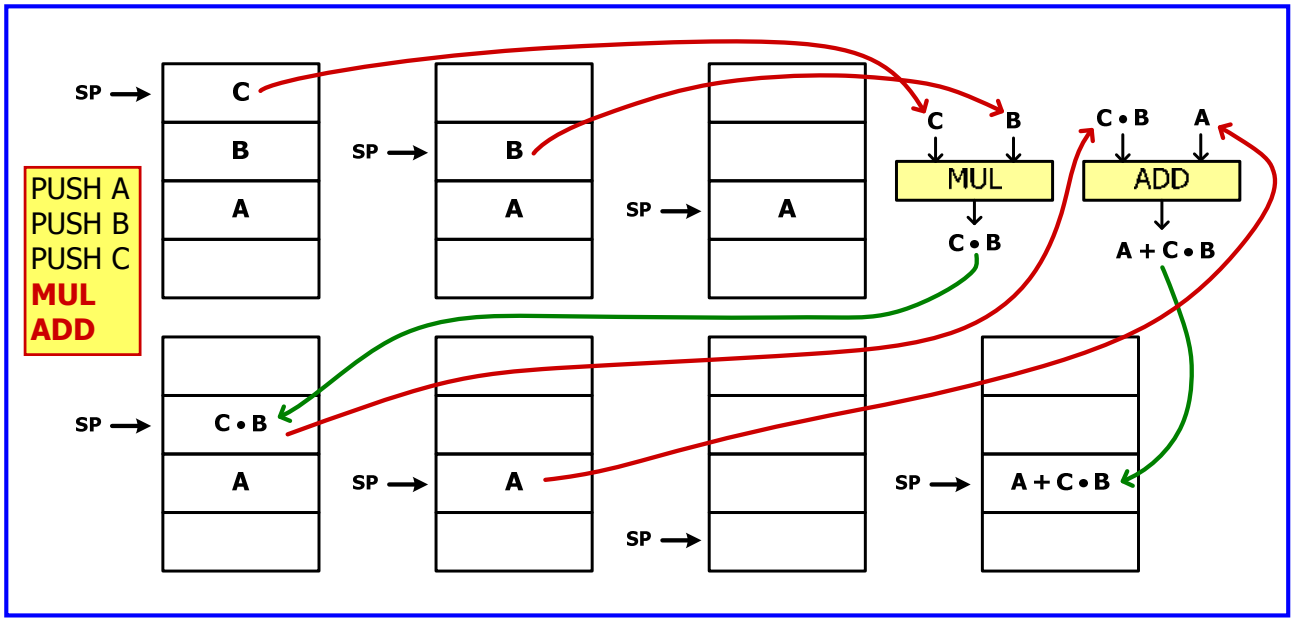
- Η δημιουργία συμβολικού κώδικα (Assembly) για αρχιτεκτονικές μηχανισμού στοίβας, διευκολύνεται εάν γράψουμε την προς υπολογισμό παράσταση σε σημειογραφία επιθέματος (postfix notation).
- Σε αυτή δεν χρησιμοποιούνται παρενθέσεις, οι τελεστές ακολουθούν τα ορίσματα (operands) και εάν υπάρχουν πολλαπλοί τελεστές, καθένας από αυτούς σημειώνεται μετά το δεύτερο όρισμά του.

Παράδειγμα: D = A + B • C, Postfix notation → D = A + (B C) • = A B C • +

PUSH A	Εναπόθεση του περιεχομένου της θέσης μνήμης A στη στοίβα
PUSH B	Εναπόθεση του περιεχομένου της θέσης μνήμης B στη στοίβα
PUSH C	Εναπόθεση του περιεχομένου της θέσης μνήμης C στη στοίβα
MUL	Πολλαπλασιασμός των περιεχομένων των B και C και εναπόθεση του γινομένου στη στοίβα
ADD	Πρόσθεση του περιεχομένων της A και του γινομένου B • C και εναπόθεση του αθροίσματος στη στοίβα
POP D	Μεταφορά του τελικού αποτελέσματος στη θέση D της κύριας μνήμης

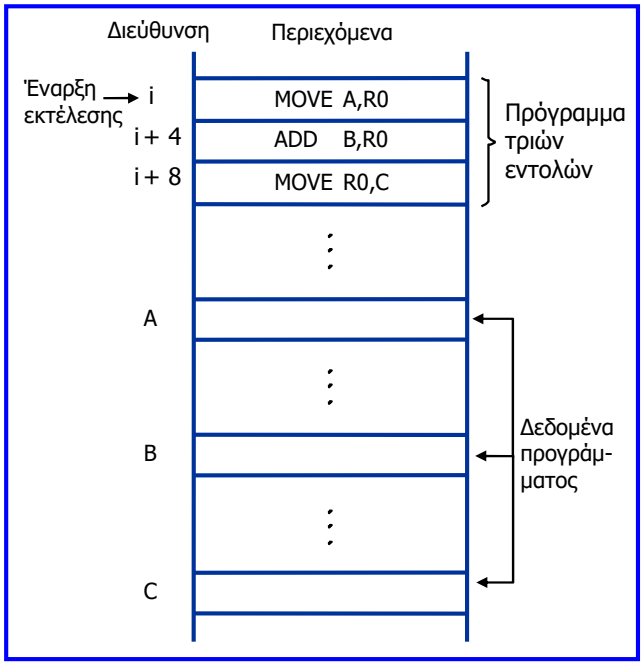
Αρχιτεκτονική μηχανισμού στοίβας

Στην αρχιτεκτονική στοίβας, οι πράξεις δύο ορισμάτων εκτελούνται μεταξύ των περιεχομένων των δύο κορυφαίων θέσεων της στοίβας, σε τρία στάδια και το αποτέλεσμα αποθηκεύεται στην κορυφή της στοίβας.



Ακολουθιακή εκτέλεση εντολών

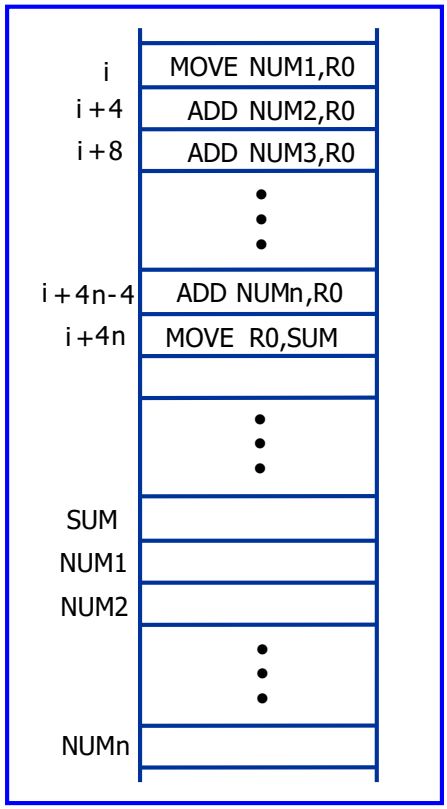
- Έστω, μνήμη οργανωμένη σε bytes και μήκος λέξης 32 bits.
- Αρχικά στον PC τοποθετείται η διεύθυνση της 1ης εντολής (i) και κατά τη διάρκεια εκτέλεσης κάθε εντολής η τιμή του PC αυξάνεται κατά 4, ώστε να δείχνει την επόμενη εντολή.
- 2 φάσεις εκτέλεσης εντολής:
- 1η φάση: προσκόμιση εντολής από τη μνήμη (instruction fetch) με βάση τη διεύθυνση που βρίσκεται στον PC και τοποθέτησή της στον IR.
- 2η φάση: εξέταση της εντολής του IR για καθορισμό της λειτουργίας της (αποκωδικοποίηση), ανάκτηση ορισμάτων από τη μνήμη ή από καταχωρητές, εκτέλεση λειτουργίας και αποθήκευση του αποτελέσματος στη θέση προορισμού.



Σε πολλούς επεξεργαστές η φάση εκτέλεσης διαιρείται σε 2 ή 3 επιμέρους φάσεις

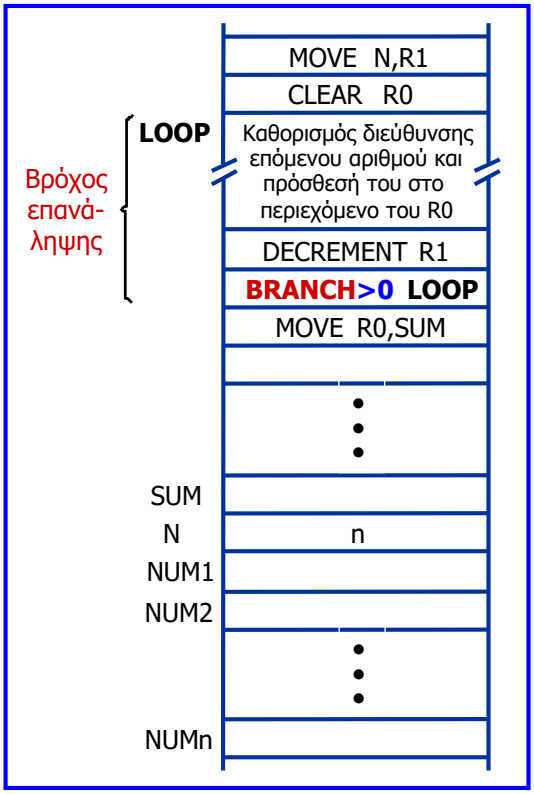
Ακολουθιακή εκτέλεση εντολών

- Υπολογισμός αθροίσματος n αριθμών.
- Διευθύνσεις μνήμης που περιέχονται οι αριθμοί αυτοί: NUM1, NUM2, …, NUMn.
- Χρησιμοποιείται εντολή ADD για να προσθέτει κάθε αριθμό στο περιεχόμενο του καταχωρητή R0.
- Όταν προστεθούν όλοι οι αριθμοί, τοποθετείται το τελικό αποτέλεσμα στη θέση μνήμης SUM.
- Μπορούμε να αποφύγουμε την επανάληψη των εντολών πρόσθεσης με τοποθέτηση μιας μόνο εντολής πρόσθεσης σε βρόχο επανάληψης (loop).



Εντολές διακλάδωσης

- Ο βρόχος επανάληψης ξεκινά από μία θέση μνήμης LOOP και καταλήγει στην εντολή BRANCH>0.
- Ο καταχωρητής R1 χρησιμοποιείται ως μετρητής για να καθορίσει τον αριθμό επαναλήψεων του βρόχου, και αρχικά φορτώνεται σε αυτόν το πλήθος αριθμών n.
- Η εκτέλεση του βρόχου επαναλαμβάνεται όσο η εντολή μείωσης κατά 1 (DECREMENT) δίνει αποτέλεσμα μεγαλύτερο από το 0.
- Η εντολή διακλάδωσης (branch instruction) φορτώνει στον PC διεύθυνση διαφορετική από εκείνη της εντολής που ακολουθεί.
- Η εντολή διακλάδωσης υπό συνθήκη (conditional branch) προκαλεί διακλάδωση εάν ικανοποιείται μια συνθήκη που σχετίζεται με το αποτέλεσμα της αμέσως προηγούμενης εντολής, διαφορετικά η ακολουθιακή εκτέλεση του προγράμματος συνεχίζεται κανονικά.



Σημαίες συνθηκών ή κατάστασης (flags)

- Ο επεξεργαστής παρακολουθεί τα αποτελέσματα των πράξεων που προηγούνται των εντολών διακλάδωσης υπό συνθήκη, καταγράφοντας την απαιτούμενη πληροφορία σε ψηφία (bits) που αναφέρονται ως σημαίες συνθηκών ή κατάστασης (conditional, status flags) και αποθηκεύονται στον καταχωρητή κατάστασης (status register) του επεξεργαστή.

N (αρνητικό) ή S (πρόσημο)	Γίνεται 1 όταν το αποτέλεσμα μιας εντολής είναι αρνητικό, διαφορετικά τίθεται στο 0.
Z (μηδέν)	Γίνεται 1 όταν το αποτέλεσμα μιας εντολής είναι 0, διαφορετικά τίθεται στο 0.
C (κρατούμενο)	Γίνεται 1 όταν μια εντολή παράγει κρατούμενο εξόδου, διαφορετικά τίθεται στο 0.
V (υπερχείλιση)	Γίνεται 1 όταν μια εντολή προκαλεί υπερχείλιση, διαφορετικά τίθεται στο 0.
P (ισοτιμία)	Γίνεται 1 όταν το πλήθος των μονάδων του αποτελέσματος μιας εντολής είναι άρτιο, διαφορετικά τίθεται στο 0.

- Για παράδειγμα, κατά την εκτέλεση της εντολής BRANCH>0 ελέγχονται οι σημαίες N (ή S), Z (οι τιμές των οποίων διαμορφώνονται από την εντολή DECREMENT R1) και η διακλάδωση λαμβάνει χώρα όταν οι σημαίες N (ή S) και Z είναι 0.
- Οι συνθήκες εντολών διακλάδωσης είναι λογικές εκφράσεις με τις σημαίες κατάστασης.
- Σε κάποιους επεξεργαστές οι σημαίες επηρεάζονται αυτόματα από όλες τις αριθμητικές ή λογικές εντολές, ενώ σε άλλους διατίθενται δύο εκδοχές εντολών (μία που επηρεάζει και μία που δεν επηρεάζει τις σημαίες κατάστασης).

Πρόσθετες κατηγορίες εντολών

Εκτός από τις εντολές μεταφοράς δεδομένων, τις εντολές των βασικών αριθμητικών πράξεων (πρόσθεση, αφαίρεση, σύγκριση), τις εντολές διαχείρισης στοίβας και τις εντολές ελέγχου ροής προγράμματος (ή διακλάδωσης), χρησιμοποιούνται και οι παρακάτω πρόσθετοι τύποι εντολών:

- Λογικές εντολές.
- Εντολές μετατόπισης.
- Εντολές περιστροφής.

Λογικές εντολές

- Οι λογικές εντολές εφαρμόζουν τις βασικές λογικές λειτουργίες (NOT, AND, OR, XOR) σε όλα τα ψηφία (bit) μίας λέξης ή ενός byte.
- Για να λάβουμε την αρνητική έκδοση ενός αριθμού στο σύστημα συμπληρώματος ως προς 2, προσθέτουμε 1 στο συμπλήρωμα ως προς 1 του θετικού αριθμού:

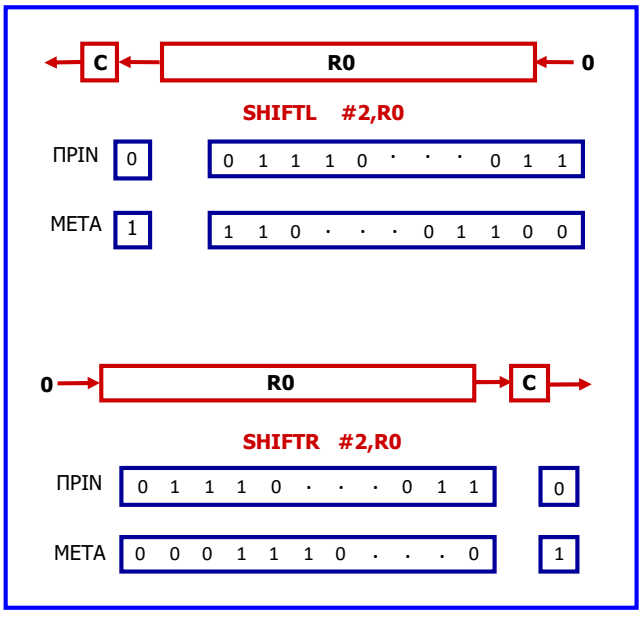
  NOT ή CMA R0

  ADD #1, R0

- Πολλοί επεξεργαστές περιλαμβάνουν την εντολή NEGATE που επιτυγχάνει το ίδιο αποτέλεσμα, χωρίς να χρησιμοποιηθεί η ακολουθία δύο εντολών.
- Η εντολή AND χρησιμοποιείται συνήθως σε πρακτικά προγραμματιστικά εγχειρήματα όταν επιθυμούμε να μηδενίσουμε κάποια από τα ψηφία (bits) ενός ορίσματος.

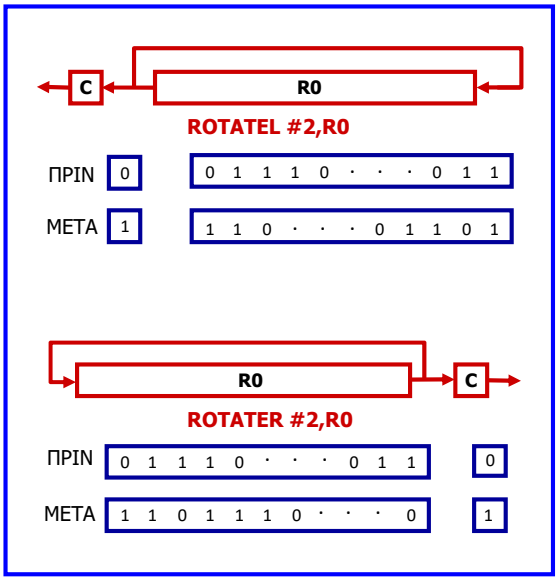
Εντολές μετατόπισης

- Μετατόπιση προς τα αριστερά (SHIFTL) και προς τα δεξιά (SHIFTR).
- Οι θέσεις που αδειάζουν πληρώνονται με μηδενικά και τα ψηφία που μετατοπίζονται χάνονται εκτός του τελευταίου που κρατείται στη σημαία κρατουμένου (C flag).
- Στην παράσταση αριθμών στο σύστημα συμπληρώματος ως προς 2, η μετατόπιση μίας θέσης αριστερά ισοδυναμεί με πολλαπλασιασμό επί 2, ενώ η μετατόπιση μίας θέσης δεξιά ισοδυναμεί με διαίρεση δια 2.



Εντολές περιστροφής

- Οι εντολές αυτές λειτουργούν παρόμοια με εκείνες της μετατόπισης, αλλά τα ψηφία που μετατοπίζονται εκτός του ενός άκρου του ορίσματος μετακινούνται στο άλλο άκρο.
- Η σημαία κρατουμένου κρατάει το τελευταίο ψηφίο που έχει μετακινηθεί εκτός του άκρου.
- Οι εντολές μετατόπισης και περιστροφής επιτελούν την εκτέλεση αριθμητικών πράξεων όπως ο πολλαπλασιασμός και η διαίρεση.



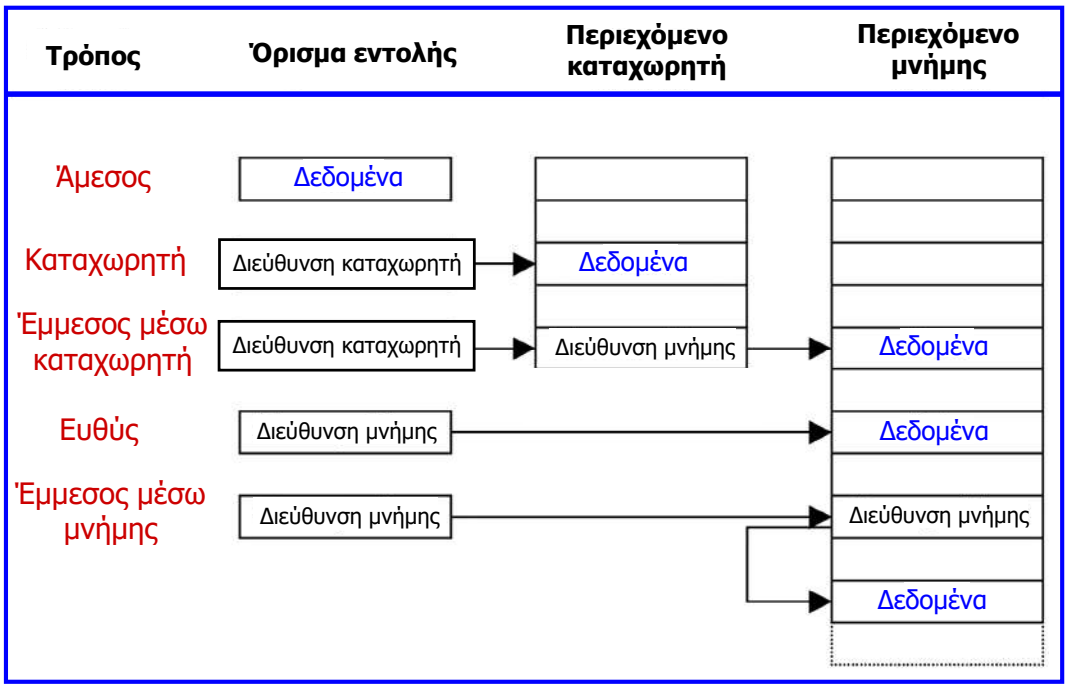
Τρόποι διευθυνσιοδότησης

Οι τρόποι με τους οποίους καθορίζεται η θέση στην οποία βρίσκεται το όρισμα (operand) μιας εντολής, αναφέρονται ως τρόποι διευθυνσιοδότησης (addressing modes)

	Τρόπος	Σύνταξη assembly	Λειτουργία
ΒΑΣΙΚΟΙ	Άμεσος (immediate)	#Value	Όρισμα = Value
	Καταχωρητή	Ri	EA = Ri
	Ευθύς (direct)	LOC	EA = LOC
	Έμμεσοι: μέσω καταχωρητή (indirect) μέσω μνήμης	(Ri) (LOC)	EA = [Ri] EA = [LOC]
	Με δείκτη (indexed)	X(Ri)	EA = [Ri] + X
	Βάσης (base) με δείκτη	(Ri, Rj)	EA = [Ri] + [Rj]
	Βάσης με δείκτη και μετατόπιση (offset)	X(Ri, Rj)	EA = [Ri] + [Rj] + X
	Σχετικός (relative)	X(PC)	EA = [PC] + X
	Αυτοαυξανόμενος (autoincrement)	(Ri)+	EA = [Ri]; Αύξηση του Ri
	Αυτομειούμενος (autodecrement)	-(Ri)	Μείωση του Ri; EA = [Ri]

EA: ενεργός διεύθυνση (effective address) ορίσματος εντολής
Value: προσημασμένος αριθμός
X: μετατόπιση (offset)

Βασικοί τρόποι διευθυνσιοδότησης



Παραδείγματα:

MOVE #200, R0

MOVE R1, R2

ADD (R1), R0

MOVE LOC, R1

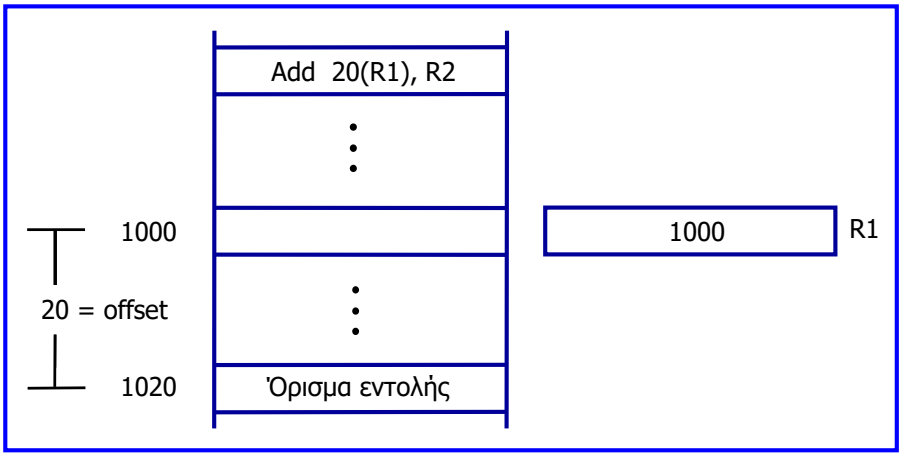
ADD (LOC), R0

Ο καταχωρητής ή η θέση μνήμης που περιέχει τη διεύθυνση, στην οποία είναι αποθηκευμένο το όρισμα μιας εντολής αναφέρεται ως δείκτης (index).

Τρόποι διευθυνσιοδότησης με δείκτη

- Διευθυνσιοδότηση με δείκτη: η ενεργός διεύθυνση του ορίσματος παράγεται με την πρόσθεση μιας σταθερής τιμής στο περιεχόμενο ενός καταχωρητή. Το περιεχόμενο του καταχωρητή δείκτη δεν αλλάζει κατά τη διάρκεια παραγωγής της ενεργής διεύθυνσης

X(Ri) ⇒ EA = [Ri] + X, X: σταθερή τιμή που περιέχεται στην εντολή (μετατόπιση, offset)



Ο τρόπος αυτός καθιστά δυνατή την προσπέλαση ενός ορίσματος που η διεύθυνσή του είναι ορισμένη σε σχέση με ένα σημείο αναφοράς μέσα στη δομή δεδομένων που ανήκει το όρισμα

Τρόποι διευθυνσιοδότησης με δείκτη

- Διευθυνσιοδότηση βάσης με δείκτη: παραλλαγή του προηγούμενου τρόπου, όπου η ενεργός διεύθυνση του ορίσματος παράγεται με την πρόσθεση μιας σταθερής τιμής που αποτελεί περιεχόμενο ενός καταχωρητή στο περιεχόμενο ενός άλλου καταχωρητή.

$$(Ri, Rj) \Rightarrow EA = [Ri] + [Rj]$$

- Ο καταχωρητής Rj περιέχει τη μετατόπιση (offset) και συνήθως ονομάζεται καταχωρητής βάσης (base register).

- Αυτός ο τρόπος διευθυνσιοδότησης δείκτη παρέχει μεγαλύτερη ευελιξία στην προσπέλαση ορισμάτων, λόγω του ότι και οι δυο συνιστώσες της ενεργής διεύθυνσης μπορούν να μεταβληθούν.

- Διευθυνσιοδότηση βάσης με δείκτη και μετατόπιση: μια ακόμη παρόμοια εκδοχή, όπου η ενεργός διεύθυνση του ορίσματος είναι το άθροισμα της σταθεράς X και των περιεχομένων δύο καταχωρητών.

$$X(Ri, Rj) \Rightarrow EA = [Ri] + [Rj] + X$$

Σχετική διευθυνσιοδότηση

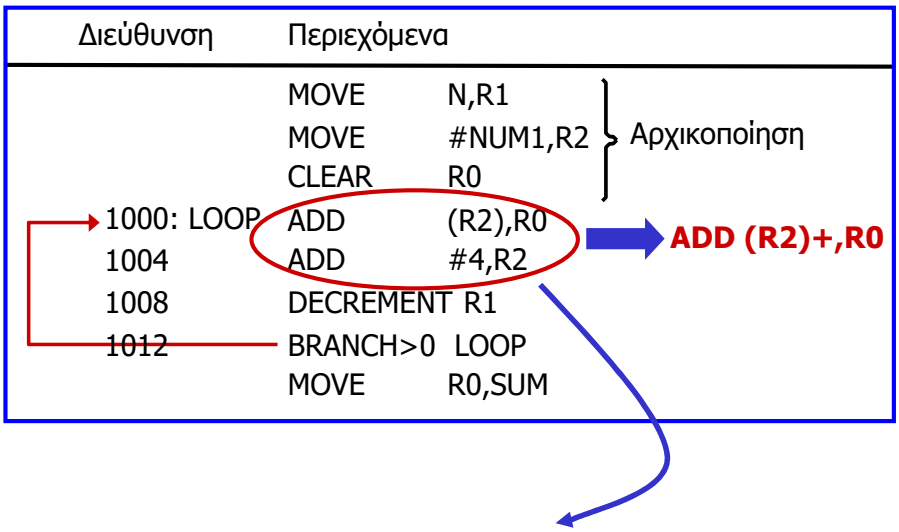
- Διευθυνσιοδότηση με δείκτη, όπου αντί για καταχωρητή γενικού σκοπού χρησιμοποιείται ο μετρητής προγράμματος (PC): $X(PC) \Rightarrow EA = [PC] + X$.
- Η πιο συχνή του χρήση συνίσταται στον καθορισμό της διεύθυνσης προορισμού μιας εντολής διακλάδωσης (στο παράδειγμα που ακολουθεί $X = -16$).
- Η συμβολική γλώσσα Assembly επιτρέπει τις εντολές διακλάδωσης με ετικέτες (labels) στη θέση του προορισμού διακλάδωσης. Ο συμβολομεταφραστής (assembler) επεξεργάζεται μια τέτοια εντολή, υπολογίζει πρώτα την απαιτούμενη απόσταση με βάση την τιμή του PC και στη συνέχεια παράγει τις κατάλληλες εντολές μηχανής.

Διεύθυνση	Περιεχόμενα		
	MOVE	N,R1	Αρχικοποίηση
	MOVE	#NUM1,R2	
	CLEAR	R0	
1000: LOOP	ADD	(R2),R0	
1004	ADD	#4,R2	
1008	DECREMENT	R1	
1012	BRANCH>0	LOOP	
	MOVE	R0,SUM	

Αυτοαυξανόμενος και αυτομειούμενος τρόπος

- Αυτοαυξανόμενος τρόπος: η ενεργός διεύθυνση του ορίσματος είναι το περιεχόμενο ενός καταχωρητή, ο οποίος καθορίζεται στην εντολή. Μετά την προσπέλαση του ορίσματος, το περιεχόμενο του συγκεκριμένου καταχωρητή αυξάνεται αυτόματα, έτσι ώστε να δείχνει στο επόμενο στοιχείο μιας λίστας δεδομένων (Ri)+.
- Αυτομειούμενος τρόπος: το περιεχόμενο του καταχωρητή, ο οποίος καθορίζεται στην εντολή, πρώτα μειώνεται αυτόματα και στη συνέχεια χρησιμοποιείται ως ενεργός διεύθυνση για την προσπέλαση του ορίσματος –(Ri).
- Οι δύο αυτοί τρόποι χρησιμοποιούνται για την προσπέλαση στοιχείων δεδομένων που είναι αποθηκευμένα σε διαδοχικές θέσεις στη μνήμη.
- Για να προσπελάσουμε διαδοχικές λέξεις σε μία μνήμη οργανωμένη σε bytes (διευθυνσιοδοτούμενη κατά byte) με μήκος λέξης 32 bits, το βήμα αύξησης ή μείωσης του περιεχομένου του καταχωρητή πρέπει να είναι ίσο με 4.
- Οι επεξεργαστές που υποστηρίζουν τους τρόπους αυτούς, αυξάνουν ή μειώνουν αυτόματα το περιεχόμενο του καταχωρητή κατά ένα βήμα ίσο με τον αριθμό των bytes του ορίσματος στο οποίο διενεργείται προσπέλαση.

Αυτοαυξανόμενος και αυτομειούμενος τρόπος



Συνδυασμός δύο πράξεων σε μία, με αποτέλεσμα τη μείωση του αριθμού των εντολών

Συμβολική γλώσσα Assembly

- Οι εντολές μηχανής αναπαριστάνονται από συνδυασμούς 0 και 1 και προφανώς είναι δύσχρηστες για τη δημιουργία προγραμμάτων.
- Έτσι χρησιμοποιούμε συμβολικά ονόματα εντολών (MOVE, ADD κ.ά.) που ονομάζονται μνημονικές λέξεις (mnemonics) και συμβολικά ονόματα για τους καταχωρητές (Ri) και τις θέσεις μνήμης (LOC).
- Το σύνολο αυτών των ονομάτων μαζί με τους κανόνες χρήσης τους αναφέρονται ως συμβολική γλώσσα Assembly.
- Τα προγράμματα συμβολικής γλώσσας μεταφράζονται αυτόματα σε ακολουθίες εντολών μηχανής, μέσω ενός προγράμματος που ονομάζεται συμβολομεταφραστής (assembler).
- Το πρόγραμμα του χρήστη αναφέρεται ως πηγαίο (source) πρόγραμμα και το πρόγραμμα που προκύπτει από την εφαρμογή του συμβολομεταφραστή και περιέχει εντολές γλώσσας μηχανής αναφέρεται ως αντικειμενικό (object) πρόγραμμα.

Συμβολική γλώσσα Assembly

- Η μνημονική λέξη μιας εντολής (π.χ. MOVE) αναπαριστά τον κωδικό λειτουργίας (op-code ή operation code) της πράξης που επιτελείται από την εντολή.
- Στη συνέχεια μιας εντολής (π.χ. R0,SUM) αναγράφεται η πληροφορία που καθορίζει τα ορίσματα (operands) της εντολής.
- Το όρισμα προορισμού (SUM) διαχωρίζεται από το όρισμα αφετηρίας (R0) με κόμμα, χωρίς ενδιάμεσα κενά.
- Όπως προαναφέρθηκε, η συμβολική γλώσσα καταδεικνύει με συγκεκριμένους συμβολισμούς τον τρόπο διευθυνσιοδότησης που χρησιμοποιείται σε κάθε εντολή.
- Το σύμβολο της δίεσης δηλώνει άμεσο όρισμα δεκαδικού αριθμού (ADD #5,R3), ενώ για δυαδικό αριθμό συνήθως θέτουμε #% και για δεκαεξαδικό #$. Σε κάποιους συμβολομεταφραστές το άμεσο όρισμα σε δεκαεξαδική μορφή αναγνωρίζεται όταν ακολουθείται από το γράμμα H (Hex).
- Σε κάποιες εκδόσεις της συμβολικής γλώσσας το άμεσο όρισμα καθορίζεται στον κωδικό λειτουργίας τη εντολής (ADDI 5,R3).

Συμβολομετάφραση και εκτέλεση προγραμμάτων

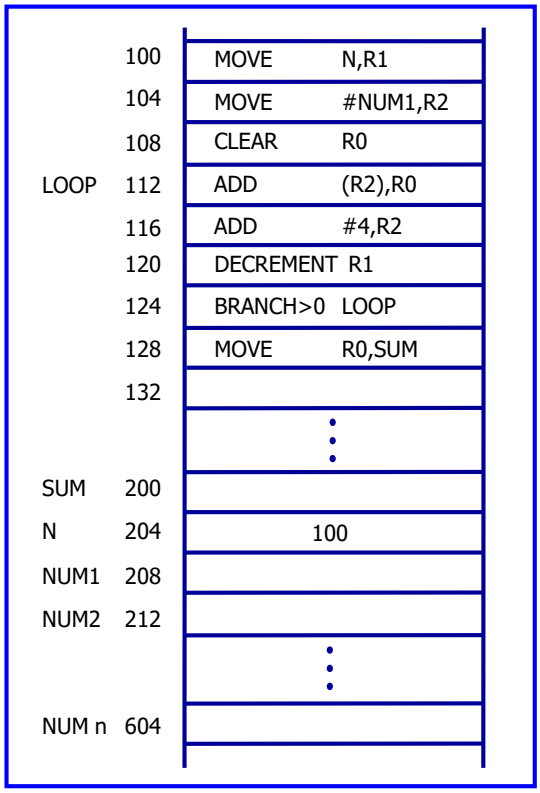
- Οι περισσότερες εκδόσεις γλώσσας Assembly απαιτούν σε ένα πηγαίο πρόγραμμα οι εντολές να έχουν την εξής μορφή:

Label Ετικέτα	Operation Κωδικός λειτουργίας	Operand(s) Όρισμα(τα)	Comments Σχόλια

- Η ετικέτα εισάγεται προαιρετικά και αποτελεί το όνομα που σχετίζεται με τη διεύθυνση μνήμης, όπου πρόκειται να φορτωθεί η εντολή γλώσσας μηχανής που παράγεται από την εντολή συμβολικής γλώσσας.
- Το πεδίο σχολίων αγνοείται από τον συμβολομεταφραστή και χρησιμοποιείται για καλύτερη κατανόηση του πηγαίου προγράμματος.
- Για να παραχθεί ένα αντικειμενικό πρόγραμμα, ο συμβολομεταφραστής θα πρέπει να γνωρίζει:
  - ✓ Πως να ερμηνεύει τα ονόματα και τις ετικέτες.
  - ✓ Που να τοποθετήσει τις εντολές και τα δεδομένα στη μνήμη.
- Για να παράσχει την πληροφορία αυτή, το πηγαίο πρόγραμμα περιλαμβάνει οδηγίες προς τον συμβολομεταφραστή (assembler directives).

Συμβολομετάφραση και εκτέλεση προγραμμάτων

Διευθέτηση μνήμης για το πρόγραμμα υπολογισμού αθροίσματος 100 αριθμών που περιέχονται στις διευθύνσεις μνήμης NUM1, NUM2, … , NUMn



Συμβολομετάφραση και εκτέλεση προγραμμάτων

- EQU: τιμή διεύθυνσης SUM (200).
- ORIGIN (ή ORG) (1): θέση αποθήκευσης δεδομένων (204).
- DATAWORD (ή DW): αριθμός καταχωρήσεων δεδομένων (100) που καταχωρείται στη διεύθυνση 204.
- RESERVE: μνήμη μεγέθους 400 bytes για δεδομένα και η ετικέτα NUM1 αντιστοιχεί στην διεύθυνση 208.
- ORIGIN (ή ORG) (2): θέση αποθήκευσης εντολών (100).
- RETURN: τερματισμός προγράμματος, επιστροφή ελέγχου στο λειτουργικό σύστημα.
- END: τελευταία εντολή πηγαίου κώδικα που περιλαμβάνει την πληροφορία (ετικέτα START) της διεύθυνσης μνήμης έναρξης εκτέλεσης του προγράμματος.

	Ετικέτες διευθύνσεων μνήμης	Λειτουργία	Πληροφορίες δεδομένων και διευθυνσιοδότησης
Οδηγίες στον Συμβολομεταφραστή (assembler directives)	SUM	EQU	200
		ORIGIN	204
	N	DATAWORD	100
	NUM1	RESERVE	400
		ORIGIN	100
Κυρίως πρόγραμμα που παράγει εντολές μηχανής	START	MOVE	N,R1
		MOVE	#NUM1,R2
		CLR	R0
	LOOP	ADD	(R2),R0
		ADD	#4,R2
		DEC	R1
		BGTZ	LOOP
		MOVE	R0,SUM
Οδηγίες στον συμβολομεταφραστή		RETURN	
		END	START

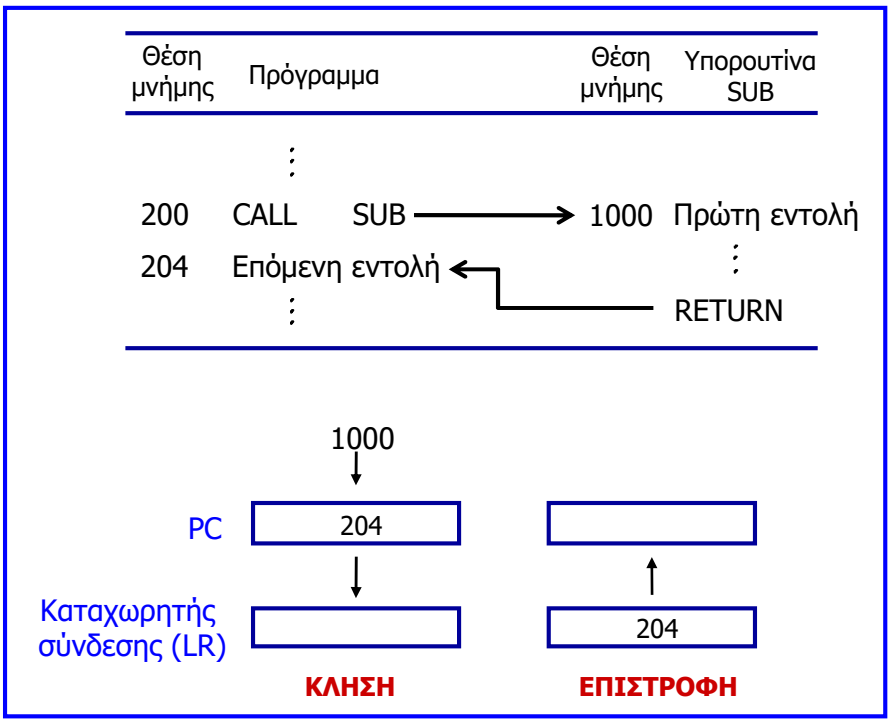
Συμβολομετάφραση και εκτέλεση προγραμμάτων

- Ο συμβολομεταφραστής αντικαθιστά όλα τα σύμβολα που δηλώνουν πράξεις και τρόπους διευθυνσιοδότησης με τους αντίστοιχους δυαδικούς κωδικούς και όλα τα ονόματα και τις ετικέτες με τις πραγματικές τους τιμές, ώστε το πρόγραμμα να είναι εκτελέσιμο από τον επεξεργαστή.
- Συχνά, ο συμβολομεταφραστής δεν αντικαθιστά άμεσα ένα όνομα το οποίο παριστάνει μια διεύθυνση, με την πραγματική τιμή, αλλά πρέπει να διενεργήσει σχετικούς υπολογισμούς (π.χ. υπολογισμός μετατόπισης εντολής διακλάδωσης).
- Ο συμβολομεταφραστής διατηρεί πίνακα συμβόλων με τα ονόματα και τις αντίστοιχες τιμές τους, ώστε όταν επανεμφανίζονται να τα αντικαθιστά με τις κατάλληλες τιμές.
- Για ονόματα που εμφανίζονται στο πρόγραμμα ως ορίσματα πριν δοθούν τιμές σε αυτά (π.χ. σε εντολή διακλάδωσης προς τα εμπρός), ο συμβολομεταφραστής διατρέχει το ίδιο πρόγραμμα δεύτερη φορά (συμβολομεταφραστής δύο-περασμάτων, two-pass assembler).
- Το αντικειμενικό πρόγραμμα φορτώνεται στη μνήμη μέσω βοηθητικού προγράμματος (φορτωτής, loader), στο οποίο παρέχονται πληροφορίες (μέγεθος και θέση μνήμης προγράμματος) από τον συμβολομεταφραστή.
- Ο συμβολομεταφραστής εντοπίζει συντακτικά λάθη, ενώ μέσω του αποσφαλματωτή (debugger) που περιλαμβάνεται στο λογισμικό συστήματος, ο χρήστης εξετάζει τα περιεχόμενα των καταχωρητών και θέσεων μνήμης και εντοπίζει τα λογικά λάθη.

Υπορουτίνες: κλήση και επιστροφή

- Σε ένα πρόγραμμα είναι συχνά αναγκαίο να εκτελεστεί μία συγκεκριμένη διεργασία πολλές φορές, με διαφορετικά δεδομένα (υπορουτίνα, subroutine).
- Ο προγραμματιστής από το να επεκτείνει επιτόπου (inline expansion) το πρόγραμμα κάθε φορά που απαιτείται η διεργασία αυτή, για εξοικονόμηση χώρου, τοποθετεί στη μνήμη μόνο ένα αντίγραφο των εντολών της υπορουτίνας και έτσι το πρόγραμμα που απαιτεί τη χρήση της, εκτελεί διακλάδωση με προορισμό τη διεύθυνση στην οποία αρχίζει η υπορουτίνα. Αυτό βέβαια αυξάνει το χρόνο εκτέλεσης του προγράμματος.
- Η εντολή που επιτελεί τη λειτουργία διακλάδωσης ονομάζεται εντολή κλήσης της υπορουτίνας (CALL).
- Μετά την εκτέλεση της υπορουτίνας, το πρόγραμμα συνεχίζει από το σημείο κλήσης της υπορουτίνας, εκτελώντας μία εντολή επιστροφής (RETURN ή RET).
- Η θέση στην οποία το πρόγραμμα θα πρέπει να αναλάβει ξανά τον έλεγχο της εκτέλεσης, καταδεικνύεται από το περιεχόμενο του PC, επομένως το περιεχόμενό του θα πρέπει να διατηρηθεί, ώστε να καταστεί δυνατή η σωστή επιστροφή στο πρόγραμμα.
- Ο καταχωρητής σύνδεσης (link register) είναι αφιερωμένος στην αποθήκευση της διεύθυνσης επιστροφής.

Υπορουτίνες: κλήση και επιστροφή



Υπορουτίνες: εμφώλευση (subroutine nesting)

- Η εμφώλευση υπορουτινών αποτελεί συνηθισμένη πρακτική προγραμματισμού κατά την οποία μία υπορουτίνα καλεί κάποια άλλη.
- Η διεύθυνση επιστροφής της επόμενης κλήσης αποθηκεύεται στον καταχωρητή σύνδεσης καταστρέφοντας το προηγούμενο περιεχόμενό του.
- Για να διασωθούν τα περιεχόμενα του καταχωρητή σύνδεσης πριν γίνει κλήση της επόμενης υπορουτίνας, οι διευθύνσεις επιστροφής οι οποίες σχετίζονται με τις κλήσεις των υπορουτινών θα πρέπει να εναποτίθενται σε μια δομή στοίβας.
- Η εντολή κλήσης μιας υπορουτίνας εναποθέτει το περιεχόμενο του PC σε μία ειδική δομή στοίβας που καλείται στοίβα επεξεργαστή (processor stack) και φορτώνει τη διεύθυνση της υπορουτίνας στον PC.
- Η εντολή επιστροφής ανακτά τη διεύθυνση επιστροφής από τη στοίβα του επεξεργαστή και την τοποθετεί στον PC.
- Ειδική περίπτωση αποτελεί η αναδρομική κλήση μιας υπορουτίνας, δηλαδή η περίπτωση όπου μία υπορουτίνα καλεί τον εαυτό της, οπότε επαναλαμβάνεται μέχρι μία συνθήκη να διακόψει την αναδρομική κλήση.

Υπορουτίνες: πέρασμα παραμέτρων

- Πέρασμα παραμέτρων: ανταλλαγή πληροφοριών μεταξύ καλούντος προγράμματος και κληθείσας υπορουτίνας, δηλαδή, εφοδιασμός υπορουτίνας με τα απαιτούμενα ορίσματα ή τις διευθύνσεις τους και επιστροφή των αποτελεσμάτων των υπολογισμών της υπορουτίνας στο πρόγραμμα.
- Το πέρασμα παραμέτρων διενεργείται άμεσα και αποδοτικά μέσω καταχωρητών.
- Στην υπορουτίνα πρόσθεσης λίστας αριθμών, το πέρασμα του μεγέθους της λίστας (N) και της διεύθυνσης του πρώτου αριθμού (NUM1) γίνεται μέσω των R1, R2.
- Η υπορουτίνα υπολογίζει το άθροισμα και το τοποθετεί στον R0.

Πρόγραμμα

```
        MOVE      N,R1
        MOVE      #NUM1,R2
        CALL      LISTADD
        MOVE      R0,SUM
```

Υπορουτίνα

```
LISTADD CLEAR     R0
LOOP    ADD       (R2)+,R0
        DECREMENT R1
        BRANCH>0  LOOP
        RETURN
```

Αντί για το πέρασμα των τιμών όλων των αριθμών της λίστας, το πρόγραμμα περνά στην υπορουτίνα μόνο τη διεύθυνση NUM1 του πρώτου αριθμού (πέρασμα κατά αναφορά), και το πλήθος n των αριθμών (πέρασμα κατά τιμή) που είναι αποθηκευμένο στη διεύθυνση μνήμης N.

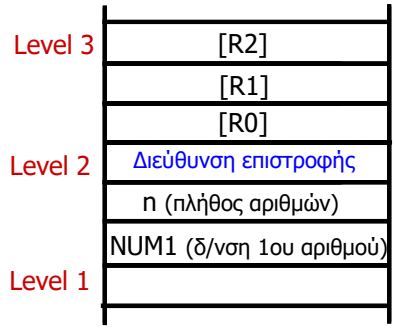
Υπορουτίνες: πέρασμα παραμέτρων

Εάν οι παράμετροι είναι πολλές και οι διαθέσιμοι καταχωρητές δεν είναι αρκετοί, χρησιμοποιούμε μια δομή στοίβας.

Έστω ότι αρχικά ο SP «δείχνει» στο Level 1

```
        PUSH        #NUM1
        PUSH        N           Εναπόθεση παραμέτρων
        CALL        LISTADD     Κλήση υπορουτίνας (SP: Level 2)
        MOVE        4(SP),SUM   Αποθήκευση αποτελέσματος

LISTADD PUSHMULTIPLE R0-R2      Εναπόθεση R0-R2 (SP: Level 3)
        MOVE        16(SP),R1   Αρχικοποίηση μετρητή στο n
        MOVE        20(SP),R2   Τοποθέτηση της NUM1 στον R2
        CLEAR       R0          Αρχικοποίηση αθροίσματος
LOOP    ADD         (R2)+,R0    Πρόσθεση αριθμού λίστας
        DECREMENT   R1
        BRANCH>0    LOOP
        MOVE        R0,20(SP)   Τοποθέτηση αποτελέσματος
                                στη στοίβα (στη θέση της NUM1)
        POPMULTIPLE R0-R2       Ανάκτηση περιεχομένων καταχω-
        RETURN                  ρητών και επιστροφή στο πρό-
                                γραμμα (SP: Level 2)
```

Δομή στοίβας



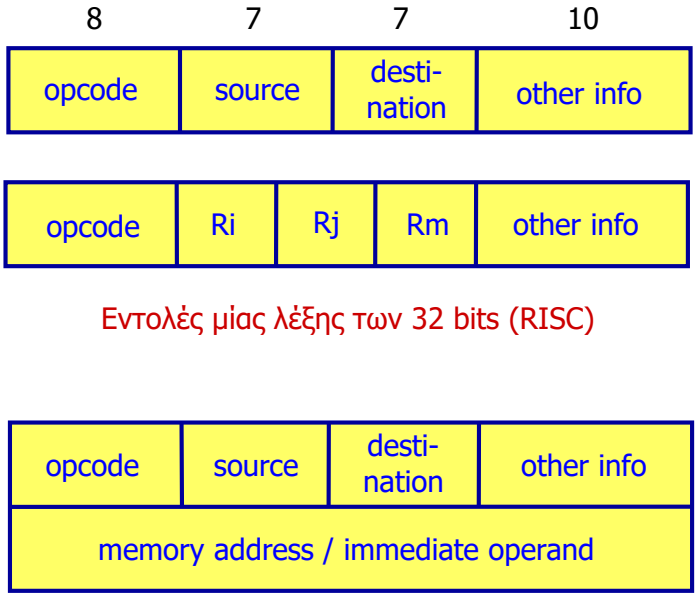
Η εντολή CALL εναποθέτει στη στοίβα τη διεύθυνση επιστροφής.

Το πρόγραμμα περνά στην υπορουτίνα μόνο τη διεύθυνση NUM1 του πρώτου αριθμού και το πλήθος n των αριθμών που είναι αποθηκευμένο στη διεύθυνση N.

Κωδικοποίηση εντολών

- Η μορφή με την οποία παρουσιάστηκαν οι εντολές είναι ενδεικτική (typical ή generic Assembly) των μορφών της γλώσσας Assembly διάφορων επεξεργαστών.
- Για να εκτελεστεί μία εντολή γλώσσας Assembly (που χρησιμοποιεί συμβολικό ακρωνύμιο) από τον επεξεργαστή, θα πρέπει να κωδικοποιηθεί σε ένα δυαδικό πρότυπο (binary pattern) που αναφέρεται ως εντολή μηχανής (machine instruction).
- Το μήκος κάθε εντολής (π.χ. 32 bits) θα πρέπει να είναι αρκετό για την αναπαράσταση της αναγκαίας πληροφορίας.
- Ο τύπος λειτουργίας που εκτελεί μία εντολή καθορίζεται με ένα δυαδικό πρότυπο (π.χ. 8 bits) που αναφέρεται ως κωδικός λειτουργίας (operation code, opcode).
- Οι καταχωρητές που συμμετέχουν στην εντολή θα πρέπει επίσης να καθοριστούν (π.χ. εάν χρησιμοποιούνται 16 καταχωρητές σε έναν επεξεργαστή, τότε 4 bits είναι αναγκαία για τον καθορισμό κάθε καταχωρητή μέσα σε μία εντολή).
- Επίσης, ένα μέρος της εντολής μηχανής διατίθεται για τον καθορισμό του τρόπου διευθυνσιοδότησης που ακολουθείται για κάθε όρισμα και των τιμών του δείκτη (index) ή της μετατόπισης (offset), όταν χρησιμοποιούνται.
- Τέλος, μέρος της εντολής μπορεί να διατίθεται για περιγραφή άμεσων ορισμάτων ή διευθύνσεων μνήμης.

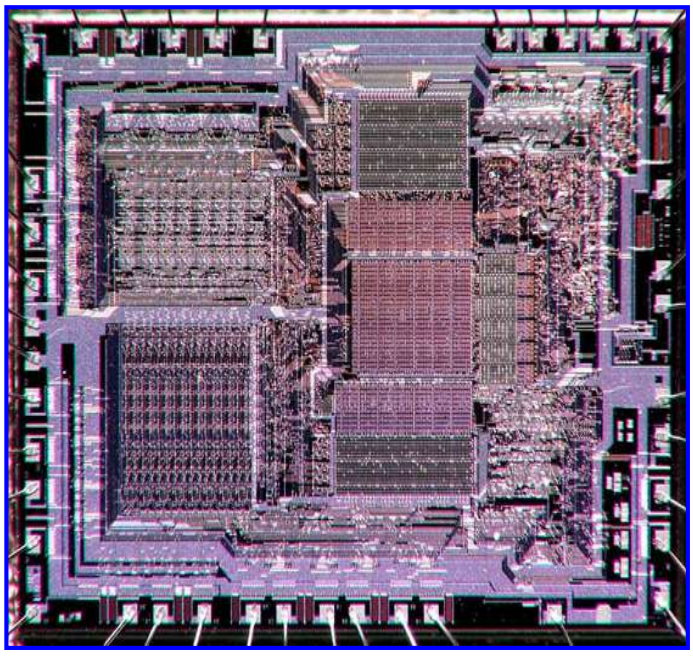
Κωδικοποίηση εντολών



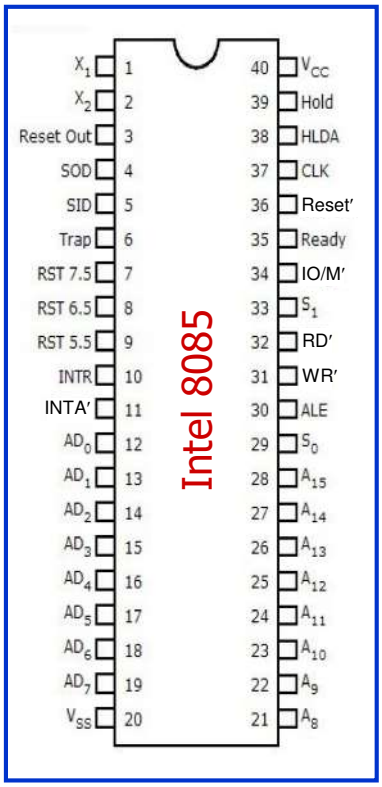
Εντολές μίας λέξης των 32 bits (RISC)

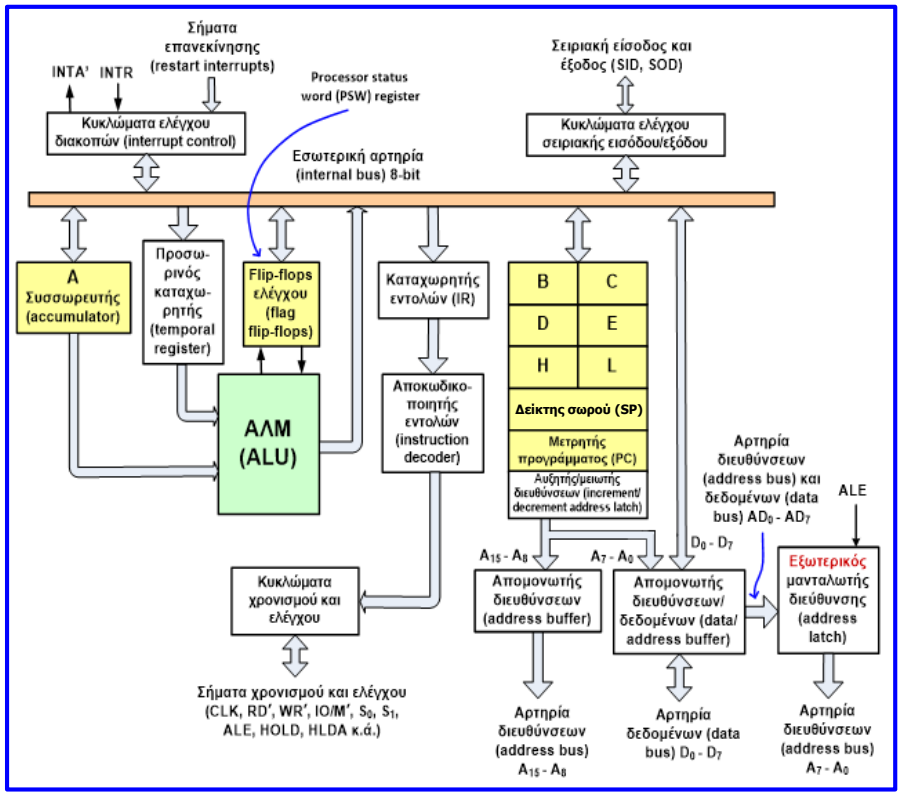
Εντολή δύο λέξεων (CISC)

Κεφάλαιο 3: Μικροεπεξεργαστής Intel 8085



Ακροδέκτες και αρχιτεκτονική του Intel 8085





Καταχωρητές του Intel 8085

- Ο μικροεπεξεργαστής Intel 8085 διαθέτει 12 βασικούς καταχωρητές των 8 bits από τους οποίους 4 λειτουργούν μόνο ως ζεύγη και δημιουργούν 2 καταχωρητές των 16 bits:

A	Συσσωρευτής (accumulator)	8 bits
PC	Μετρητής προγράμματος	16 bits
B-C, D-E, H-L	Γενικής χρήσης, δείκτες (H-L)	(8 bits x 6) ή (16 bits x 3)
SP	Δείκτης στοίβας (stack pointer)	16 bits
PSW	Καταχωρητής κατάστασης	8 bits (5 σημαίες ελέγχου)

- Σημαίες κατάστασης ή ελέγχου (flags):

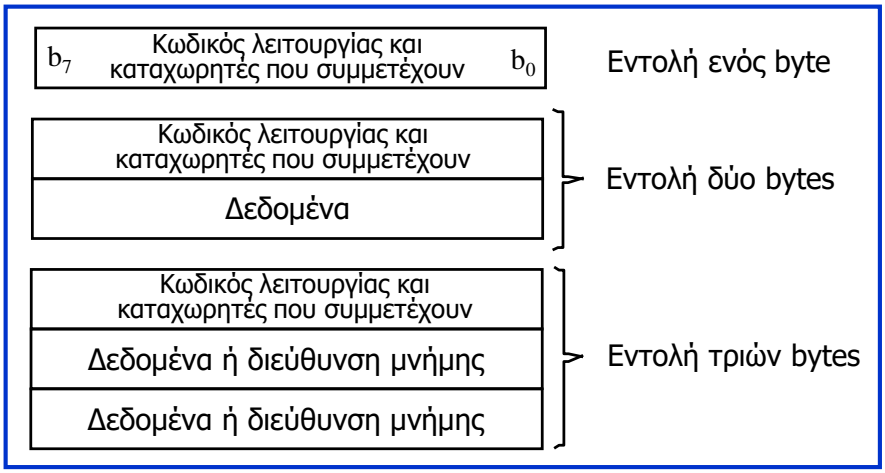
b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
S	Z		AC		P		CY

PSW: processor state word

  - ✓ μηδέν (zero, Z), 1 όταν το αποτέλεσμα μιας εντολής είναι 0, αλλιώς 0,
  - ✓ πρόσημο (sign, S), 1 όταν το MSB του αποτελέσματος μιας εντολής είναι 1, αλλιώς 0,
  - ✓ ισοτιμία (parity, P), 1 όταν το αποτέλεσμα μιας εντολής έχει άρτια ισοτιμία, αλλιώς 0,
  - ✓ κρατούμενο (carry, CY), 1 όταν το αποτέλεσμα μιας εντολής προκαλεί κρατούμενο (πρόσθεση) ή δανεικό ψηφίο (αφαίρεση) ή για αποτέλεσμα σύγκρισης <, αλλιώς 0,
  - ✓ βοηθητικό κρατούμενο (auxiliary carry, AC), 1 όταν μια εντολή προκαλεί κρατούμενο μεταξύ του τρίτου και του τέταρτου ψηφιού του αποτελέσματος, αλλιώς 0.

Μορφή εντολών του Intel 8085

- Η μνήμη του μικροεπεξεργαστή 8085 είναι οργανωμένη κατά byte και κάθε byte έχει μία μοναδική διεύθυνση των 16 bits, επομένως ο διευθυνσιοδοτούμενος χώρος μνήμης είναι 64 KB (2^{16}). Χρησιμοποιείται ανάθεση κατά little-endian.
- Ο 8085 χειρίζεται δεδομένα των 8 bits και οι εντολές του απαρτίζονται από 1, 2 ή 3 bytes, τα οποία αποθηκεύονται σε διαδοχικές θέσεις μνήμης.
- Στην περίπτωση εντολής με περισσότερα από 1 byte, η διεύθυνση του πρώτου byte χρησιμοποιείται ως διεύθυνση της εντολής.



Κατηγορίες εντολών του Intel 8085

- Λογικές εντολές: εκτελούν λογικές πράξεις (AND, OR, XOR, NOT) και περιστροφή δεδομένων που είναι τοποθετημένα στον συσσωρευτή.
- Αριθμητικές εντολές: προσθέτουν, αφαιρούν, αυξάνουν ή μειώνουν κατά ένα και συγκρίνουν δεδομένα που είναι τοποθετημένα σε καταχωρητές ή θέσεις μνήμης.
- Οι αριθμητικές και λογικές εντολές επηρεάζουν τις σημαίες ελέγχου (κατάστασης), εκτός εάν ορίζεται διαφορετικά.
- Εντολές μεταφοράς δεδομένων: μεταφέρουν δεδομένα μεταξύ των καταχωρητών ή μεταξύ θέσεων μνήμης και καταχωρητών.
- Εντολές διακλάδωσης (άλματος): εκτελούν άλματα με ή χωρίς συνθήκη (J) και διαδικασίες σχετικές με την εκτέλεση υπορουτινών, δηλαδή κλήση (CALL) και επιστροφή (RET).
- Εντολές διαχείρισης σωρού: εκτελούν λειτουργίες σχετικές με τη διαχείριση της στοίβας, όπως εναπόθεση (PUSH) στη στοίβα και ανάκτηση (POP) από τη στοίβα.
- Τέλος, είναι διαθέσιμες εντολές για ανάγνωση δεδομένων από πόρτα εισόδου (IN) και εγγραφή δεδομένων σε πόρτα εξόδου (OUT), εντολές που σχετίζονται με διακοπές (EI, enable interrupts, DI, disable interrupts) και θέση ή επαναφορά σημαίας ελέγχου (STC, set carry, CMC, complement carry), αδρανοποίηση (HLT) και επανεκκίνηση (RST) του μικροεπεξεργαστή ή εισαγωγή καθυστέρησης (NOP, no operation).

Λογικές εντολές του Intel 8085

addr: διεύθυνση 16 bits, byte: δεδομένα 8 bits, dble: δεδομένα 16 bits, byte 2, byte 3: το 2ο και το 3ο byte της εντολής, r, r1, r2: ένας από τους καταχωρητές A, B, C, D, E, H, L, rp: ένα από τα ζεύγη καταχωρητών (B = BC, D = DE, H = HL)

ANA r	Εκτελείται πράξη λογικού AND μεταξύ του περιεχομένου του καταχωρητή r και του περιεχομένου του A και το αποτέλεσμα τοποθετείται στον A. Σημαίες ελέγχου: Z, S, P, C, AC.
ORA r	Εκτελείται πράξη λογικού OR μεταξύ του περιεχομένου του καταχωρητή r και του περιεχομένου του A και το αποτέλεσμα τοποθετείται στον A. Σημαίες ελέγχου: Z, S, P, C, AC.
ANI byte	Εκτελείται πράξη λογικού AND μεταξύ του byte 2 της εντολής και του περιεχομένου του A και το αποτέλεσμα τοποθετείται στον A. Σημαίες ελέγχου: Z, S, P, C, AC.
ORI byte	Εκτελείται πράξη λογικού OR μεταξύ του byte 2 της εντολής και του περιεχομένου του A και το αποτέλεσμα τοποθετείται στον A. Σημαίες ελέγχου: Z, S, P, C, AC.
XRA r	Εκτελείται πράξη λογικού XOR μεταξύ του περιεχομένου του καταχωρητή r και του περιεχομένου του A και το αποτέλεσμα τοποθετείται στον A. Σημαίες ελέγχου: Z, S, P, C, AC.
CMA	Το περιεχόμενο του A συμπληρώνεται (τα 0 γίνονται 1 και τα 1 γίνονται 0).
CMP M	Το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση περιέχεται στους καταχωρητές H και L αφαιρείται από τον συσσωρευτή A. Σημαίες ελέγχου: Z, S, P, C, AC.
CPI byte	Το περιεχόμενο του byte 2 της εντολής συγκρίνεται με το περιεχόμενο του A. Σημαίες ελέγχου: Z, S, P, C, AC.

Εντολές περιστροφής του Intel 8085

RLC	Το περιεχόμενο του A ολισθαίνει αριστερά κατά μία θέση. Το λιγότερο σημαντικό bit τίθεται στην τιμή που ολισθαίνει έξω από τη θέση του πιο σημαντικού bit. Σημαίες ελέγχου: C.
RRC	Το περιεχόμενο του A ολισθαίνει δεξιά κατά μία θέση. Το περισσότερο σημαντικό bit τίθεται στην τιμή που ολισθαίνει έξω από τη θέση του λιγότερου σημαντικού bit. Σημαίες ελέγχου: C.
RAL	Το περιεχόμενο του A ολισθαίνει αριστερά κατά μία θέση. Το λιγότερο σημαντικό bit τίθεται στην τιμή της σημαίας Carry και η σημαία Carry τίθεται στην τιμή που ολισθαίνει έξω από τη θέση του περισσότερου σημαντικού bit. Σημαίες ελέγχου: C.
RAR	Το περιεχόμενο του A ολισθαίνει δεξιά κατά μία θέση. Το περισσότερο σημαντικό bit τίθεται στην τιμή της σημαίας Carry και η σημαία Carry τίθεται στην τιμή που ολισθαίνει έξω από τη θέση του λιγότερο σημαντικού bit. Σημαίες ελέγχου: C.

Αριθμητικές εντολές του Intel 8085

ADD r	Το περιεχόμενο του καταχωρητή r προστίθεται στο περιεχόμενο του συσσωρευτή A και το αποτέλεσμα τοποθετείται στο συσσωρευτή A. Σημαίες ελέγχου: Z, S, P, C, AC.
ADC r	Το περιεχόμενο του καταχωρητή r προστίθεται, με χρήση κρατουμένου, στο περιεχόμενο του συσσωρευτή A και το αποτέλεσμα τοποθετείται στο συσσωρευτή A. Σημαίες ελέγχου: Z, S, P, C, AC.
ADD M	Το περιεχόμενο της θέσης μνήμης, της οποίας η διεύθυνση περιέχεται σους καταχωρητές H και L, προστίθεται στο περιεχόμενο του συσσωρευτή A και το αποτέλεσμα τοποθετείται στο συσσωρευτή A. Σημαίες ελέγχου: Z, S, P, C, AC.
ADI byte	Το περιεχόμενο του byte 2 της εντολής προστίθεται στο περιεχόμενο του συσσωρευτή A και το αποτέλεσμα τοποθετείται στο συσσωρευτή A. Σημαίες ελέγχου: Z, S, P, C, AC.
SUB r	Το περιεχόμενο του καταχωρητή r αφαιρείται από το περιεχόμενο του συσσωρευτή A και το αποτέλεσμα τοποθετείται στο συσσωρευτή A. Σημαίες ελέγχου: Z, S, P, C, AC.
SUI byte	Το περιεχόμενο του byte 2 της εντολής αφαιρείται από το περιεχόμενο του συσσωρευτή A και το αποτέλεσμα τοποθετείται στο συσσωρευτή A. Σημαίες ελέγχου: Z, S, P, C, AC.
INR r	Το περιεχόμενο του καταχωρητή r αυξάνεται κατά ένα. Σημαίες ελέγχου: Z, S, P, AC.
DCR r	Το περιεχόμενο του καταχωρητή r μειώνεται κατά ένα. Σημαίες ελέγχου: Z, S, P, AC.
INX rp	Το περιεχόμενο του ζεύγους καταχωρητών rp αυξάνεται κατά ένα.
DCX rp	Το περιεχόμενο του ζεύγους καταχωρητών rp μειώνεται κατά ένα.
DAD rp	Το περιεχόμενο του ζεύγους καταχωρητών rp προστίθεται στο περιεχόμενο του ζεύγους καταχωρητών H-L και το αποτέλεσμα τοποθετείται στο ζεύγος H-L. Σημαίες ελέγχου: C.

Εντολή κωδικοποίησης BCD του Intel 8085

DAA	Η εντολή αυτή χρησιμοποιείται έτσι ώστε ο οκταψήφιος δυαδικός αριθμός (π.χ. το άθροισμα δύο οκταψήφιων δυαδικών αριθμών) που περιέχεται στον καταχωρητή A, να προσαρμοστεί σε έναν έγκυρο αριθμό δυο ψηφίων κωδικοποιημένων σύμφωνα με τον κώδικα BCD. Εάν η τιμή των τεσσάρων λιγότερο σημαντικών ψηφίων του οκταψήφιου δυαδικού αριθμού που είναι αποθηκευμένος στον A είναι μεγαλύτερη του 9 ή η σημαία ελέγχου AC έχει τιμή 1, τότε προστίθεται στο περιεχόμενο του A ο αριθμός 6. Εάν η τιμή των τεσσάρων περισσότερο σημαντικών ψηφίων του οκταψήφιου δυαδικού αριθμού που είναι αποθηκευμένος στον A είναι μεγαλύτερη του 9 ή η σημαία ελέγχου C έχει τιμή 1, τότε προστίθεται ο αριθμός 6 στα τέσσερα περισσότερο σημαντικά ψηφία του περιεχομένου του A. Εάν δεν συμβαίνει κάποιο από τα παραπάνω ενδεχόμενα, τότε η τιμή των τεσσάρων λιγότερο ή των τεσσάρων περισσότερο σημαντικών ψηφίων δεν αλλάζει, αφού αποτελούν έγκυρα ψηφία κωδικοποιημένα σύμφωνα με τον κώδικα BCD.

Εντολές μεταφοράς δεδομένων του Intel 8085

MOV r1,r2	Το περιεχόμενο του καταχωρητή r2 μετακινείται στο καταχωρητή r1.
MOV r,M	Το περιεχόμενο της θέσης μνήμης, της οποίας η διεύθυνση περιέχεται σους καταχωρητές H και L, μετακινείται στο καταχωρητή r..
MOV M,r	Το περιεχόμενο του καταχωρητή r μετακινείται στη θέσης μνήμης, της οποίας η διεύθυνση περιέχεται σους καταχωρητές H και L.
MVI r,byte	Το περιεχόμενο του byte 2 της εντολής μετακινείται στον καταχωρητή r.
LXI rp,dble	Το byte 3 της εντολής μετακινείται στο καταχωρητή υψηλής τάξης (rh) του ζεύγους καταχωρητών rp. Το byte 2 της εντολής μετακινείται στο καταχωρητή χαμηλής τάξης (rl) του ζεύγους καταχωρητών rp.
LDA addr	Το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση καθορίζεται στα bytes 2 και 3 της εντολής μετακινείται στο καταχωρητή A.
LDAX rp	Το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση υπάρχει στο ζεύγος καταχωρητών rp μετακινείται στο καταχωρητή A (ισχύει μόνο για τα ζεύγη BC, DE).
STA addr	Το περιεχόμενο του καταχωρητή A μετακινείται στη θέση μνήμης της οποίας η διεύθυνση καθορίζεται στα bytes 2 και 3 της εντολής.
STAX rp	Το περιεχόμενο του καταχωρητή A μετακινείται στη θέση μνήμης της οποίας η διεύθυνση καθορίζεται από το ζεύγος των καταχωρητών rp (ισχύει μόνο για τα ζεύγη BC, DE).
LHLD addr	Το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση καθορίζεται στα bytes 2 και 3 της εντολής, μετακινείται στον καταχωρητή L και το περιεχόμενο της επόμενης θέσης μνήμης μετακινείται στον καταχωρητή H.
SHLD addr	Το περιεχόμενο του καταχωρητή H αποθηκεύεται στη θέση της μνήμης της οποίας η διεύθυνση καθορίζεται στα bytes 2 και 3 της εντολής και το περιεχόμενο του καταχωρητή L αποθηκεύεται στην επόμενη θέση μνήμης.
XCHG	Τα περιεχόμενα των καταχωρητών H και L ανταλλάσσονται με τα περιεχόμενα των καταχωρητών D και E, αντίστοιχα.

Εντολές διακλάδωσης του Intel 8085

JMP addr	Ο έλεγχος μεταφέρεται στην εντολή με διεύθυνση τα bytes 2 και 3 της εντολής.
JZ addr	Εάν η σημαία zero είναι ενεργοποιημένη (Z = 1), ο έλεγχος μεταφέρεται στην εντολή με διεύθυνση τα bytes 2 και 3 της εντολής.
JNZ addr	Εάν η σημαία zero είναι απενεργοποιημένη (Z = 0), ο έλεγχος μεταφέρεται στην εντολή με διεύθυνση τα bytes 2 και 3 της εντολής.
JC addr	Εάν η σημαία carry είναι ενεργοποιημένη (C = 1), ο έλεγχος μεταφέρεται στην εντολή με διεύθυνση τα bytes 2 και 3 της εντολής.
JNC addr	Εάν η σημαία carry είναι απενεργοποιημένη (C = 0), ο έλεγχος μεταφέρεται στην εντολή με διεύθυνση τα bytes 2 και 3 της εντολής.
JM addr	Εάν η σημαία sign είναι ενεργοποιημένη (S = 1), ο έλεγχος μεταφέρεται στην εντολή με διεύθυνση τα bytes 2 και 3 της εντολής.
JP addr	Εάν η σημαία sign είναι απενεργοποιημένη (S = 0), ο έλεγχος μεταφέρεται στην εντολή με διεύθυνση τα bytes 2 και 3 της εντολής.
CALL addr	Τα 8 bits υψηλής τάξης της διεύθυνσης της επόμενης εντολής μετακινούνται στη θέση μνήμης με διεύθυνση κατά 1 μικρότερη από το περιεχόμενο του SP. Τα 8 bits χαμηλής τάξης της διεύθυνσης της επόμενης εντολής μετακινούνται στη θέση μνήμης με διεύθυνση κατά 2 μικρότερη από το περιεχόμενο του SP. Το περιεχόμενο του SP μειώνεται κατά 2. Ο έλεγχος μεταφέρεται στην εντολή με διεύθυνση τα bytes 2 και 3 της εντολής.
RET	Το περιεχόμενο της θέσης μνήμης με διεύθυνση που καθορίζεται στον SP, μετακινείται στα 8 bits χαμηλής τάξης του PC. Το περιεχόμενο της θέσης μνήμης με διεύθυνση κατά 1 μεγαλύτερη από το περιεχόμενο του SP, μετακινείται στα 8 bits υψηλής τάξης του PC. Το περιεχόμενο του SP αυξάνεται κατά 2.

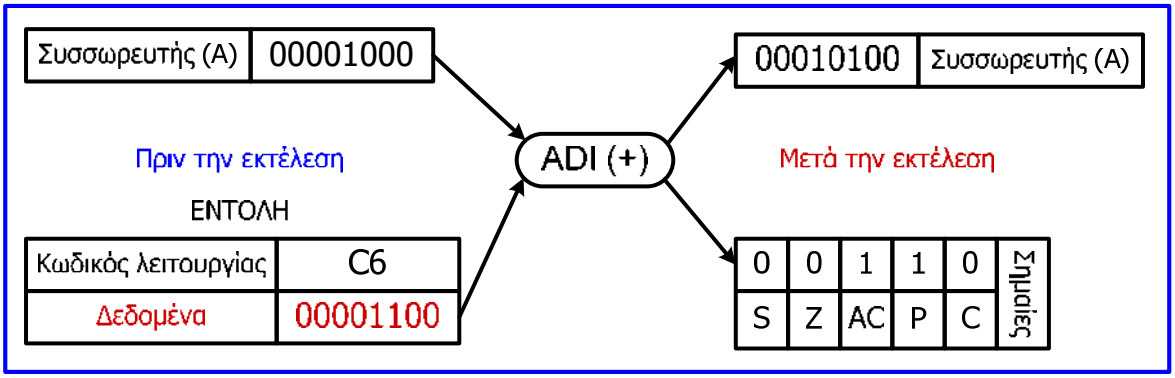
Εντολές διαχείρισης στοίβας του Intel 8085

PUSH rp	Το περιεχόμενο του καταχωρητή υψηλής τάξης του ζεύγους καταχωρητών rp, μετακινείται στη θέση μνήμης της οποίας η διεύθυνση είναι κατά 1 μικρότερη από το περιεχόμενο του SP. Το περιεχόμενο του καταχωρητή χαμηλής τάξης του ζεύγους καταχωρητών rp, μετακινείται στη θέση μνήμης της οποίας η διεύθυνση είναι κατά 2 μικρότερη από το περιεχόμενο του SP. Το περιεχόμενο του SP μειώνεται κατά 2.
PUSH PSW	Το περιεχόμενο του A μετακινείται στη θέση μνήμης της οποίας η διεύθυνση είναι κατά 1 μικρότερη από το περιεχόμενο του SP. Οι σημαίες ελέγχου σχηματίζουν μια λέξη η οποία μετακινείται στη θέση μνήμης της οποίας η διεύθυνση είναι κατά 2 μικρότερη από το περιεχόμενο του SP. Το περιεχόμενο του SP μειώνεται κατά 2.
POP rp	Το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση καθορίζεται στον SP, μετακινείται στο χαμηλής τάξης καταχωρητή του ζεύγους rp. Το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση είναι κατά 1 μεγαλύτερη από το περιεχόμενο του SP, μετακινείται στο καταχωρητή υψηλής τάξης του ζεύγους rp. Το περιεχόμενο του SP αυξάνεται κατά 2.
POP PSW	Το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση καθορίζεται στον SP, χρησιμοποιείται για την αποκατάσταση των σημαιών ελέγχου. Το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση είναι κατά 1 μεγαλύτερη από το περιεχόμενο του SP, μετακινείται στον A. Το περιεχόμενο του SP αυξάνεται κατά 2.
XTHL	Το περιεχόμενο του L ανταλλάσσεται με το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση καθορίζεται στον SP και το περιεχόμενο του H ανταλλάσσεται με το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση είναι κατά 1 μεγαλύτερη από το περιεχόμενο του SP.
SPHL	Το περιεχόμενο των καταχωρητών H και L μετακινείται στον καταχωρητή SP.

Πλήρες σύνολο εντολών του Intel 8085

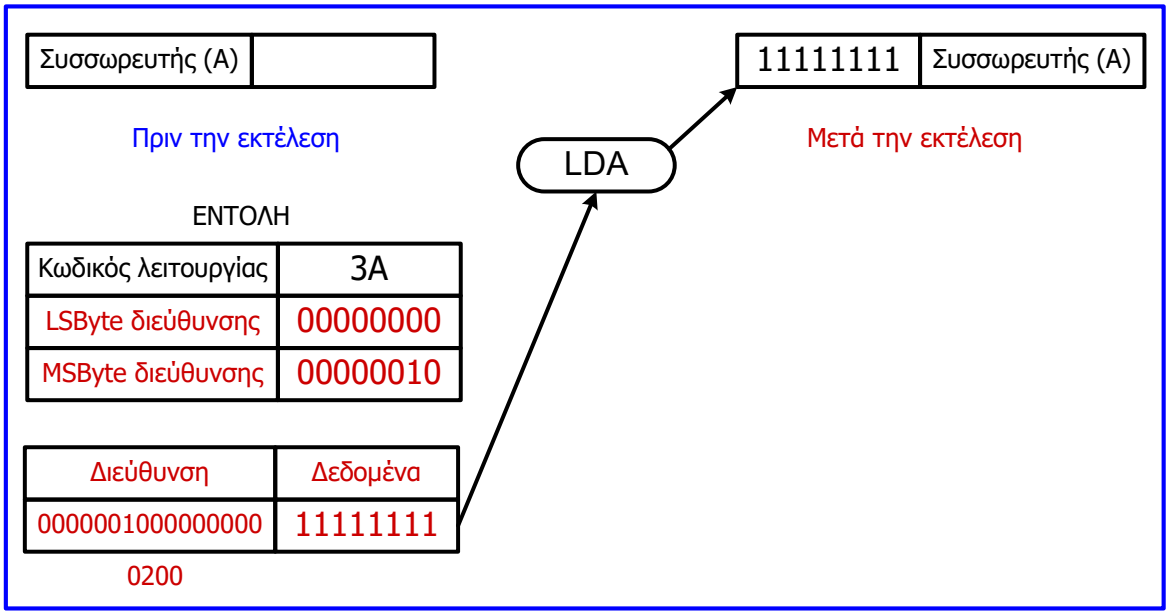
Τρόποι διευθυνσιοδότησης στον Intel 8085

- Στον επεξεργαστή Intel 8085 χρησιμοποιούνται 4 τρόποι διευθυνσιοδότησης: άμεσος (immediate), ευθύς (direct), έμμεσος (indirect) μέσω καταχωρητή, καταχωρητή.
- Άμεσος (immediate): τα δεδομένα (8 ή 16 bits) εμπεριέχονται στην ίδια την εντολή. Στην περίπτωση δεδομένων 16 bits, το λιγότερο σημαντικό byte προηγείται.



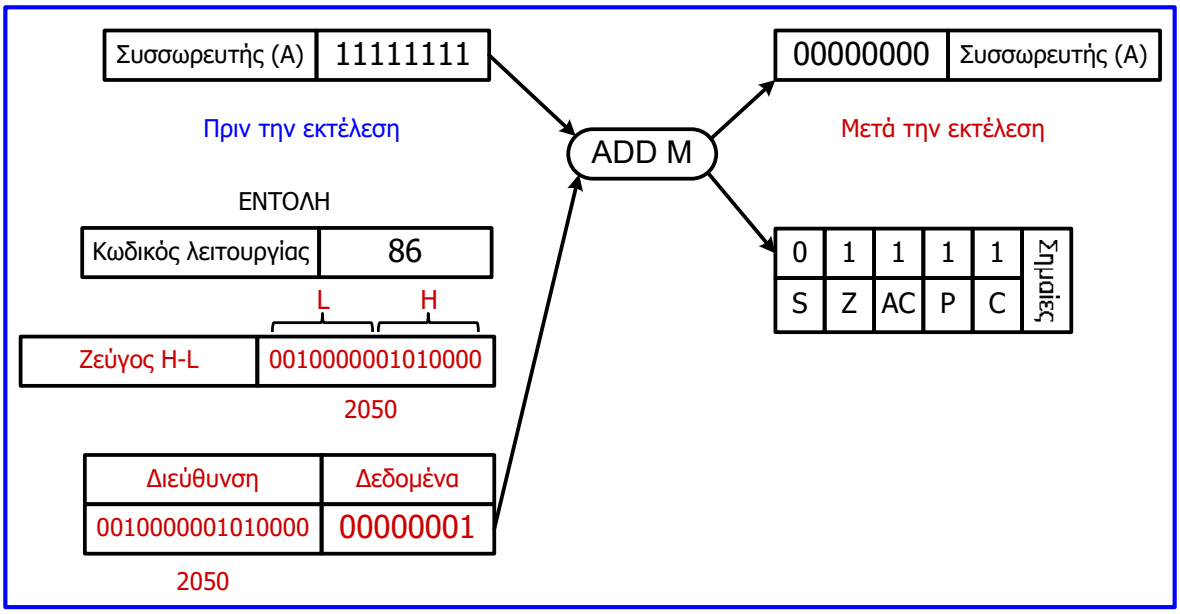
Τρόποι διευθυνσιοδότησης στον Intel 8085

- Ευθύς (direct): Το 2ο και 3ο byte της εντολής περιέχουν τη διεύθυνση μνήμης των δεδομένων που πρόκειται να ανακτηθούν από τη μνήμη ή να αποθηκευτούν σε αυτή.



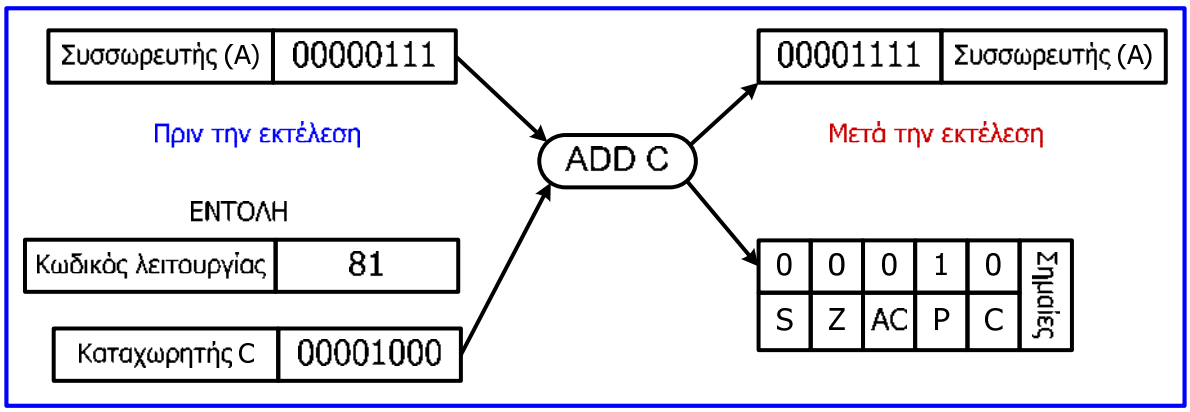
Τρόποι διευθυνσιοδότησης στον Intel 8085

- Έμμεσος (indirect) μέσω καταχωρητή: η εντολή καθορίζει ένα ζεύγος καταχωρητών που περιέχει τη διεύθυνση μνήμης όπου είναι αποθηκευμένα τα δεδομένα. Τα 8 λιγότερο σημαντικά ψηφία της διεύθυνσης είναι τοποθετημένα στον 1ο καταχωρητή του ζεύγους, ενώ τα 8 περισσότερο σημαντικά ψηφία είναι τοποθετημένα στο 2ο καταχωρητή.



Τρόποι διευθυνσιοδότησης στον Intel 8085

- Καταχωρητή (register): η εντολή καθορίζει τον καταχωρητή ή το ζεύγος καταχωρητών όπου είναι τοποθετημένα τα δεδομένα.



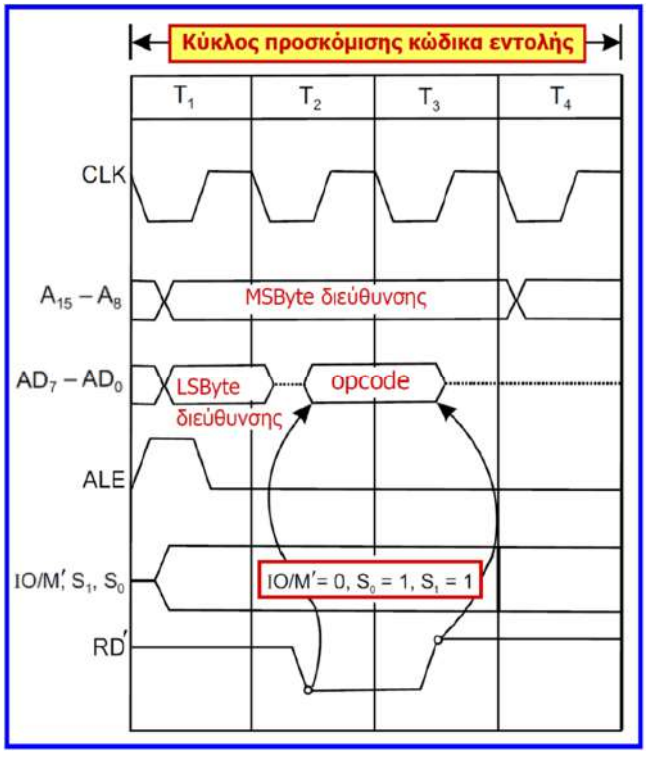
- Οι εντολές διακλάδωσης και η εντολή κλήσης υπορουτίνας εμπεριέχουν (στο 2ο και 3ο byte) τη διεύθυνση της επόμενης εντολής που θα εκτελεστεί (ευθύς τρόπος διευθυνσιοδότησης).
- Η εντολή επιστροφής από υπορουτίνα υποδεικνύει ένα ζεύγος καταχωρητών (SP) που περιλαμβάνει τη διεύθυνση της επόμενης εντολής (έμμεσος τρόπος διευθυνσιοδότησης μέσω καταχωρητή).

Χρονισμός των εντολών του Intel 8085

- Ο χρόνος που απαιτείται για την ολοκλήρωση μιας εντολής εξαρτάται από τον τύπο της εντολής, δηλαδή από τις επιμέρους λειτουργίες που περιλαμβάνει. Ο χρόνος αυτός, στον μικροεπεξεργαστή Intel 8085, αποτελείται από κύκλους μηχανής (machine cycles).
- Βασικοί κύκλοι μηχανής: προσκόμισης κώδικα εντολής (opcode fetch), ανάγνωσης από τη μνήμη (memory read), εγγραφής στη μνήμη (memory write), ανάγνωσης ή εγγραφής μονάδας εισόδου/εξόδου (I/O read, write).
- Ο κύκλος προσκόμισης κώδικα εντολής διαρκεί 4 κύκλους ρολογιού (ή καταστάσεις, states) και οι κύκλοι ανάγνωσης & εγγραφής (μνήμης ή μονάδων Ε/Ε) διαρκούν 3 κύκλους ρολογιού.
- Η εκτέλεση των εντολών με μέγεθος ενός byte (π.χ. ADD B, MOV C,B, RRC) ολοκληρώνεται στη διάρκεια του 4ου κύκλου ρολογιού του κύκλου μηχανής προσκόμισης κώδικα εντολής.
- Για την εκτέλεση εντολών που λειτουργούν με δεδομένα 2 bytes (π.χ. INX B, DCX D), ο κύκλος προσκόμισης κώδικα εντολής επεκτείνεται κατά 2 κύκλους ρολογιού.
- Σύνθετες εντολές, στις οποίες για παράδειγμα υπάρχει ανάγκη ανάγνωσης δεδομένων από τη μνήμη ή εγγραφής δεδομένων στη μνήμη, απαιτούν περισσότερους κύκλους μηχανής.
- Μερικές εντολές, όπως οι εντολές διακλάδωσης υπό συνθήκη παρουσιάζουν δύο πιθανούς χρόνους ολοκλήρωσης, ανάλογα με το αν ακολουθείται η διακλάδωση ή όχι.
- Για παράδειγμα, η εντολή JNZ απαιτεί 2 κύκλους μηχανής (opcode fetch + memory read = 7 κύκλοι ρολογιού) όταν δεν ακολουθείται η διακλάδωση και 3 κύκλους μηχανής (opcode fetch + 2 × memory read = 10 κύκλοι ρολογιού) όταν ακολουθείται η διακλάδωση.

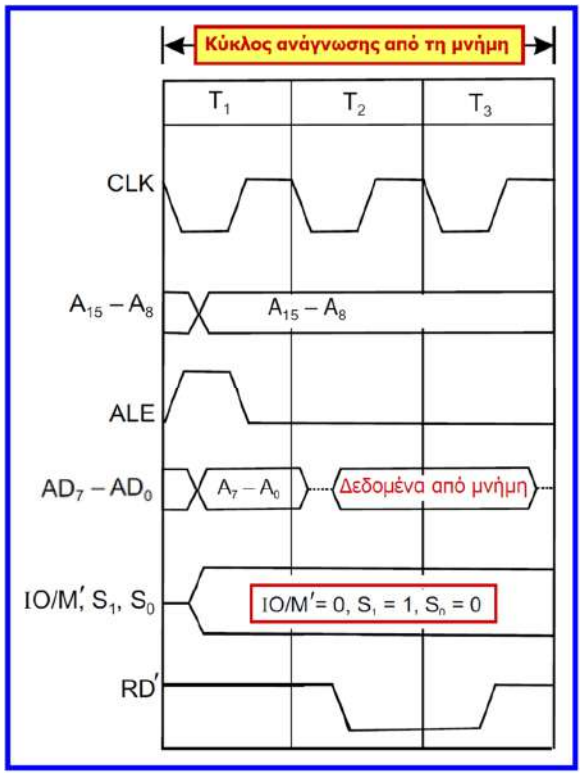
Βασικοί κύκλοι μηχανής του Intel 8085

- Σήματα (έξοδοι) που καθορίζουν τις λειτουργίες των κύκλων μηχανής: ALE, IO/M′, S_0, S_1, RD′, WR′.
- Οι γραμμές διεύθυνσης $A_0 – A_7$ πολυπλέκονται με τις γραμμές δεδομένων $D_0 – D_7$ στους ακροδέκτες $AD_0 – AD_7$.
- Το LSByte της διεύθυνσης ($A_0 – A_7$) λαμβάνεται στους ακροδέκτες $AD_0 – AD_7$ στη διάρκεια του κύκλου T_1 κάθε κύκλου μηχανής και μαζί με το MSByte ($A_8 – A_{15}$) γίνονται διαθέσιμα στις αντίστοιχες εξόδους του επεξεργαστή στη διάρκεια του κύκλου T_2 και παραμένουν διαθέσιμα έως το τέλος του κύκλου μηχανής, ώστε να είναι δυνατή η πρόσβαση του επεξεργαστή σε θέσεις μνήμης ή θύρες (ports) των μονάδων Ε/Ε.
- Το σήμα ALE λαμβάνει τιμή 1 στην αρχή του κύκλου T_1 κάθε κύκλου μηχανής και επιστρέφει σε τιμή 0 πριν το τέλος του T_1.



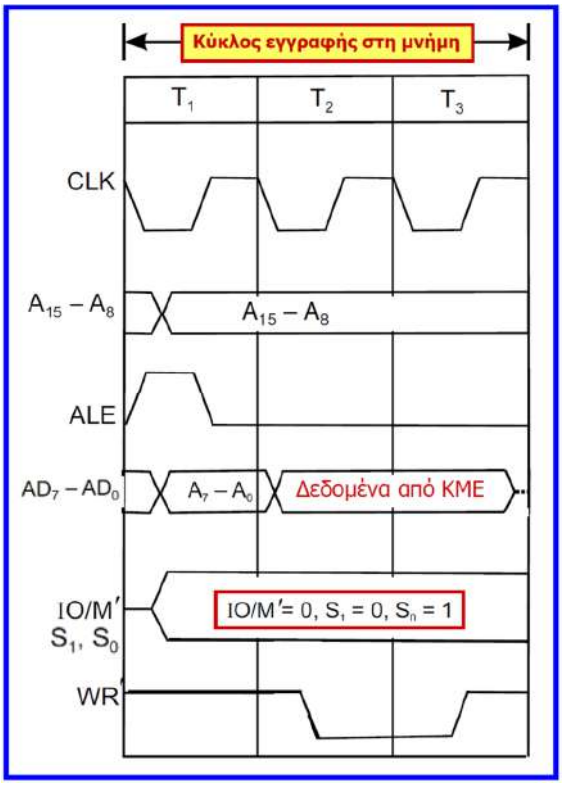
Βασικοί κύκλοι μηχανής του Intel 8085

- Κατά τη μετάβαση του από το 1 στο 0, «κλειδώνεται» (αποθηκεύεται) το LSByte της διεύθυνσης σε έναν εξωτερικό μανταλωτή (latch), έτσι ώστε να παραμείνει διαθέσιμο έως το τέλος του κύκλου μηχανής.
- Κατά το διάστημα στο οποίο το σήμα ALE έχει τιμή 0, από τους ακροδέκτες AD_0 – AD_7 είναι δυνατή η ανάγνωση ή η εγγραφή δεδομένων.
- Τα σήματα IO/M', S_0, S_1 υποδεικνύουν τον τύπο (status) του κύκλου μηχανής.
- Όταν IO/M' = 0 υποδεικνύεται λειτουργία προσπέλασης μνήμης, ενώ όταν IO/M' = 1, υποδεικνύεται λειτουργία Ε/Ε.
- S_0=S_1=1: κύκλος προσκόμισης κώδικα εντολής
  S_0=0, S_1=1: κύκλος ανάγνωσης
  S_0=1, S_1=0: κύκλος εγγραφής
- Οι τιμές των 3 σημάτων τίθενται στην αρχή κάθε κύκλου μηχανής, «κλειδώνονται» στην κατερχόμενη ακμή του σήματος ALE και παραμένουν σταθερές για όλο τον κύκλο.



Βασικοί κύκλοι μηχανής του Intel 8085

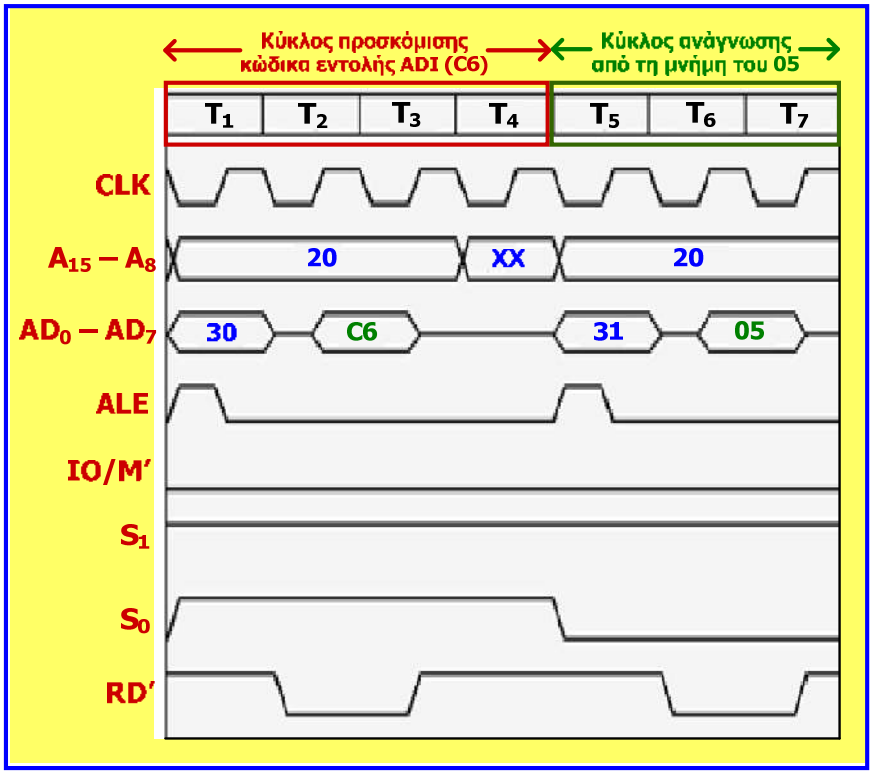
- Στη διάρκεια του κύκλου ρολογιού T_2 ενός κύκλου προσκόμισης κώδικα εντολής ή ενός κύκλου ανάγνωσης, ενεργοποιείται (γίνεται 0) από τον μικροεπεξεργαστή το σήμα RD' και εκτελείται λειτουργία ανάγνωσης κώδικα εντολής ή δεδομένων από τη μνήμη (ή από καταχωρητή μονάδας Ε/Ε).
- Στη διάρκεια του κύκλου T_2 ενός κύκλου εγγραφής, ενεργοποιείται το σήμα WR' και εκτελείται λειτουργία εγγραφής δεδομένων στη μνήμη ή σε καταχωρητή μονάδας Ε/Ε.
- Η μεταφορά, δηλαδή η προσκόμιση κώδικα εντολής ή η ανάγνωση δεδομένων από τη μνήμη ή η εγγραφή δεδομένων στη μνήμη, λαμβάνει χώρα στη διάρκεια των κύκλων T_2 και T_3 των αντίστοιχων κύκλων μηχανής.
- Τα σήματα RD', WR' απενεργοποιούνται κατά τη διάρκεια του κύκλου T_3 του αντίστοιχου κύκλου μηχανής, υποδεικνύοντας τον τερματισμό της μεταφοράς.



Βασικοί κύκλοι μηχανής του Intel 8085: παράδειγμα

- Εντολή: 2030 ADI 05 (πρόσθεση της τιμής 5 στο περιεχόμενο του συσσωρευτή)
- Για να ολοκληρωθεί η παραπάνω εντολή, χρειάζονται 2 κύκλοι μηχανής: κύκλος προσκόμισης κώδικα εντολής (ADI = C6) που διαρκεί 4 κύκλους ρολογιού και κύκλος ανάγνωσης ορίσματος (05) από τη μνήμη που διαρκεί 3 κύκλους ρολογιού.
- Μεταφορά της διεύθυνσης 2030 από τον μετρητή προγράμματος (PC) στους ακροδέκτες $AD_0 - AD_7$ και $A_8 - A_{15}$.
- Ανάγνωση από τη διεύθυνση 2030 της μνήμης του κώδικα εντολής και μεταφορά του στον καταχωρητή εντολών (IR), που τροφοδοτεί τον αποκωδικοποιητή εντολών.
- Ο αποκωδικοποιητής εντολών τροφοδοτεί τα κυκλώματα χρονισμού και ελέγχου με την πληροφορία που είναι απαραίτητη για την παραγωγή των κατάλληλων σημάτων, ώστε να εκτελεστεί η λειτουργία ανάγνωσης του ορίσματος 05 από την επόμενη διεύθυνση της μνήμης.
- Μεταφορά της επόμενης διεύθυνσης (2031) στους ακροδέκτες $AD_0 - AD_7$, $A_8 - A_{15}$, ανάγνωση από τη μνήμη του ορίσματος 05 και μεταφορά του στον προσωρινό καταχωρητή της αριθμητικής λογικής μονάδας, ώστε να προστεθεί με το περιεχόμενο του συσσωρευτή και το αποτέλεσμα να αποθηκευτεί στον συσσωρευτή.
- Η πρόσθεση και η αποθήκευση του αποτελέσματος στο συσσωρευτή, γίνονται κατά τον τρίτο κύκλο ρολογιού του κύκλου ανάγνωσης από τη μνήμη.

Βασικοί κύκλοι μηχανής του Intel 8085: παράδειγμα



Παραδείγματα προγραμματισμού του Intel 8085

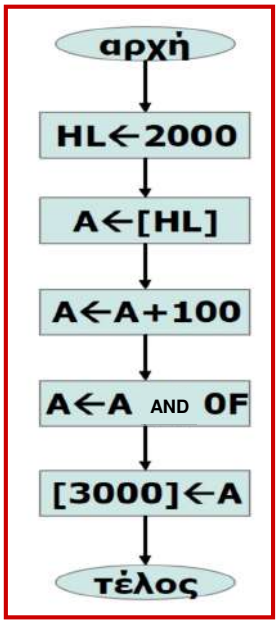
Τα παραδείγματα προγραμματισμού του επεξεργαστή Intel 8085 που ακολουθούν είναι διαθέσιμα σε βιντεοδιάλεξη διάρκειας 30 λεπτών, η οποία είναι προσβάσιμη στο σύνδεσμο:

https://www.youtube.com/watch?v=r4Wk6L8QUEU&list=PLSn1VWfV79Ejc--C8uFypGdMaJtF0kh-L&index=3

Παράδειγμα 1ο

Σύνταξη απλού προγράμματος που προσθέτει το δεκαδικό αριθμό 100, στον αριθμό 8 δυαδικών ψηφίων που είναι αποθηκευμένος στη διεύθυνση μνήμης 2000, μηδενίζει τα 4 πιο σημαντικά ψηφία του αποτελέσματος και αποθηκεύει το τελικό αποτέλεσμα στη διεύθυνση μνήμης 3000.

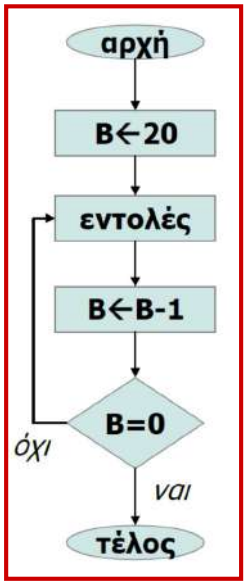
Παράδειγμα 1ο



LXI H,2000	Φόρτωσε στο ζεύγος καταχωρητών HL τη διεύθυνση 2000
MOV A,M	Μετέφερε στον συσσωρευτή A το περιεχόμενο της διεύθυνσης 2000 που είναι αποθηκευμένη στο ζεύγος HL
ADI 64	Πρόσθεσε τον δεκαδικό αριθμό 100 στο περιεχόμενο του A ($100_{10} = 01100100_2 = 64_{16}$)
ANI 0F	Μηδένισε με λογική πράξη AND, τα 4 πιο σημαντικά ψηφία του A
STA 3000	Αποθήκευσε το περιεχόμενο του A στη διεύθυνση 3000
HLT	Αδρανοποίησε τον μικροεπεξεργαστή

Παράδειγμα 2ο

Δημιουργία βρόχου που επαναλαμβάνει 20 φορές μια ακολουθία εντολών

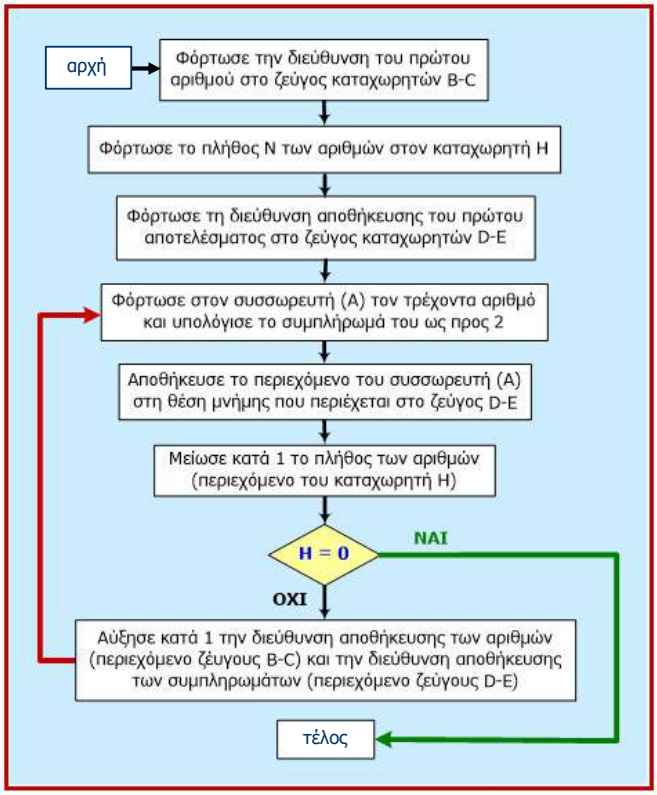



```
        MOV B,14        ο καταχωρητής B θα είναι ο μετρητής μας
Loop:   … … … …         (20₁₀ = 00010100₂ = 14₁₆)
        Ακολουθία εντολών
        … … … …
        DCR B           μείωση του μετρητή κατά 1
        JNZ Loop        εάν ο μετρητής μας δεν μηδενίστηκε, επιστροφή στην διεύθυνση Loop
        HLT             διαφορετικά τέλος και αδρανοποίηση του μικροεπεξεργαστή
```


Παράδειγμα 3ο

Σύνταξη προγράμματος υπολογισμού του συμπληρώματος ως προς 2, N αριθμών με 8 δυαδικά ψηφία ο καθένας, όταν το πλήθος των αριθμών είναι αποθηκευμένο στην θέση μνήμης 02C4 και η διεύθυνση του πρώτου αριθμού είναι αποθηκευμένη στη διεύθυνση 0300 (το λιγότερο σημαντικό byte) και στη διεύθυνση 0301 (το περισσότερο σημαντικό byte), ενώ οι υπόλοιποι αριθμοί ακολουθούν σε διαδοχικές μεγαλύτερες διευθύνσεις. Τα αποτελέσματα θα πρέπει να αποθηκεύονται στην περιοχή μνήμης που αρχίζει από τη διεύθυνση 0500 και συνεχίζεται σε διαδοχικές μεγαλύτερες διευθύνσεις.

Παράδειγμα 3ο: διάγραμμα ροής



Παράδειγμα 3ο: συμβολικό πρόγραμμα

	Διεύθυνση	Κώδικας	Σχόλια
	0000	LDA 0300	Φόρτωσε στον A το LSByte της διεύθυνσης του πρώτου αριθμού
	0003	MOV C,A	Μετάφερε το περιεχόμενο του A στον C
	0004	LDA 0301	Φόρτωσε στον A το MSByte της διεύθυνσης του πρώτου αριθμού
	0007	MOV B,A	Μετάφερε το περιεχόμενο του A στον B
	0008	LDA 02C4	Φόρτωσε στον A το πλήθος N των αριθμών
	000B	MOV H,A	Μετάφερε το περιεχόμενο του A στον H
	000C	LXI D, 0500	Φόρτωσε στο ζεύγος D-E τη διεύθυνση του πρώτου αποτελέσματος
Βρόχος προγράμματος	000F	LDAX B	Φόρτωσε από τη μνήμη στον A τον τρέχοντα αριθμό
	0010	XRI FF	Υπολόγισε το συμπλήρωμα κάθε bit του A (A'1 + 1'A = A') και αποθήκευσέ το στον A (εναλλακτικά με την εντολή CMA)
	0012	INR A	Αύξησε κατά 1 το περιεχόμενο του A (συμπλήρωμα ως προς 2)
	0013	STAX D	Αποθήκευσε το περιεχόμενο του A στη θέση μνήμης του τρέχοντος αποτελέσματος (η διεύθυνση της οποίας βρίσκεται στο ζεύγος D-E)
	0014	DCR H	Μείωσε κατά 1 το περιεχόμενο του H (πλήθος αριθμών)
	0015	JZ 001D	Εάν το αποτέλεσμα της τελευταίας πράξης είναι 0, τέλος
	0018	INX B	Αύξησε κατά 1 τη διεύθυνση ανάγνωσης των αριθμών (ζεύγος B-C)
	0019	INX D	Αύξησε κατά 1 τη διεύθυνση μνήμης που αποθηκεύονται τα αποτελέσματα (ζεύγος D-E)
	001A	JMP 000F	Επανέλαβε τη διαδικασία για τον επόμενο αριθμό
	001D	HLT	Τέλος. Τα συμπληρώματα των N αριθμών είναι αποθηκευμένα στη μνήμη από την διεύθυνση 0500 και πάνω

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	XX
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	XX
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	XX
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	XX
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	XX
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	XX
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	XX
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	XX
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

$A = 65_{16} = 01100101$
$A \text{ XOR } FF_{16} = A'$
$= 1001\ 1010 = 9A_{16}$

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	XX
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	XX
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0
9B	04 01	05 00	01 XX	0 0 0 0

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0
9B	04 01	05 00	01 XX	0 0 0 0
9B	04 01	05 01	01 XX	0 0 0 0

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0
9B	04 01	05 00	01 XX	0 0 0 0
9B	04 01	05 01	01 XX	0 0 0 0

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0
9B	04 01	05 00	01 XX	0 0 0 0
9B	04 01	05 01	01 XX	0 0 0 0
D2	04 01	05 01	01 XX	0 0 0 0

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	XX

$A = D2_{16} = 11010010$
$A \text{ XOR } FF_{16} = A'$
$= 0010\ 1101 = 2D_{16}$

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0
9B	04 01	05 00	01 XX	0 0 0 0
9B	04 01	05 01	01 XX	0 0 0 0
D2	04 01	05 01	01 XX	0 0 0 0
2D	04 01	05 01	01 XX	0 0 0 1

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0
9B	04 01	05 00	01 XX	0 0 0 0
9B	04 01	05 01	01 XX	0 0 0 0
D2	04 01	05 01	01 XX	0 0 0 0
2D	04 01	05 01	01 XX	0 0 0 1
2E	04 01	05 01	01 XX	0 0 0 1

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	2E

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0
9B	04 01	05 00	01 XX	0 0 0 0
9B	04 01	05 01	01 XX	0 0 0 0
D2	04 01	05 01	01 XX	0 0 0 0
2D	04 01	05 01	01 XX	0 0 0 1
2E	04 01	05 01	01 XX	0 0 0 1
2E	04 01	05 01	01 XX	0 0 0 1

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	2E

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0
9B	04 01	05 00	01 XX	0 0 0 0
9B	04 01	05 01	01 XX	0 0 0 0
D2	04 01	05 01	01 XX	0 0 0 0
2D	04 01	05 01	01 XX	0 0 0 1
2E	04 01	05 01	01 XX	0 0 0 1
2E	04 01	05 01	01 XX	0 0 0 1
2E	04 01	05 01	00 XX	1 0 0 1

Παράδειγμα 3ο: εκτέλεση

```
0000  LDA 0300
0003  MOV C,A
0004  LDA 0301
0007  MOV B,A
0008  LDA 02C4
000B  MOV H,A
000C  LXI D, 0500
000F  LDAX B
0010  XRI FF
0012  INR A
0013  STAX D
0014  DCR H
0015  JZ 001D
0018  INX B
0019  INX D
001A  JMP 000F
001D  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2
0500	9B
0501	2E

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
00	XX XX	XX XX	XX XX	X X X X
00	XX 00	XX XX	XX XX	X X X X
04	XX 00	XX XX	XX XX	X X X X
04	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	XX XX	X X X X
02	04 00	XX XX	02 XX	X X X X
02	04 00	05 00	02 XX	X X X X
65	04 00	05 00	02 XX	X X X X
9A	04 00	05 00	02 XX	0 0 1 1
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	02 XX	0 0 1 0
9B	04 00	05 00	01 XX	0 0 0 0
9B	04 01	05 00	01 XX	0 0 0 0
9B	04 01	05 01	01 XX	0 0 0 0
D2	04 01	05 01	01 XX	0 0 0 0
2D	04 01	05 01	01 XX	0 0 0 1
2E	04 01	05 01	01 XX	0 0 0 1
2E	04 01	05 01	01 XX	0 0 0 1
2E	04 01	05 01	00 XX	1 0 0 1

Παράδειγμα 3ο: οδηγίες στον συμβολομεταφραστή

- Εκτός της σύνταξης των εντολών του συμβολικού προγράμματος, χρησιμοποιούνται κατάλληλες οδηγίες προς τον συμβολομεταφραστή (assembler directives), που αποτελούν ψευδοεντολές με σκοπό την αποθήκευση των δεδομένων ή του κώδικα στη μνήμη.
- Με την οδηγία #ORG (origin) δηλώνεται η αρχική θέση μνήμης από την οποία και μετά θα αποθηκευτούν δεδομένα ή ο κώδικας που ακολουθεί.
- Με την οδηγία #DB (define byte) δηλώνονται τα δεδομένα που θα αποθηκευτούν στη μνήμη.
- Εάν οι εντολές του συμβολικού προγράμματος ξεκινούν από τη διεύθυνση 0000, τότε μετά το τέλος τους τίθενται οι οδηγίες του συμβολομεταφραστή που αφορούν τα δεδομένα.
- Εάν ο κώδικας ξεκινά μετά τα αποθηκευμένα δεδομένα (για παράδειγμα από την διεύθυνση 0600), τότε οι δηλώσεις για τα δεδομένα προηγούνται και στη συνέχεια τίθενται οι οδηγίες #ORG 0600, #BEGIN 0600 και ακολουθεί ο κώδικας.

Μνήμη	
Διεύθυνση	Περιε-χόμενο
02C4	02
0300	00
0301	04
0400	65
0401	D2

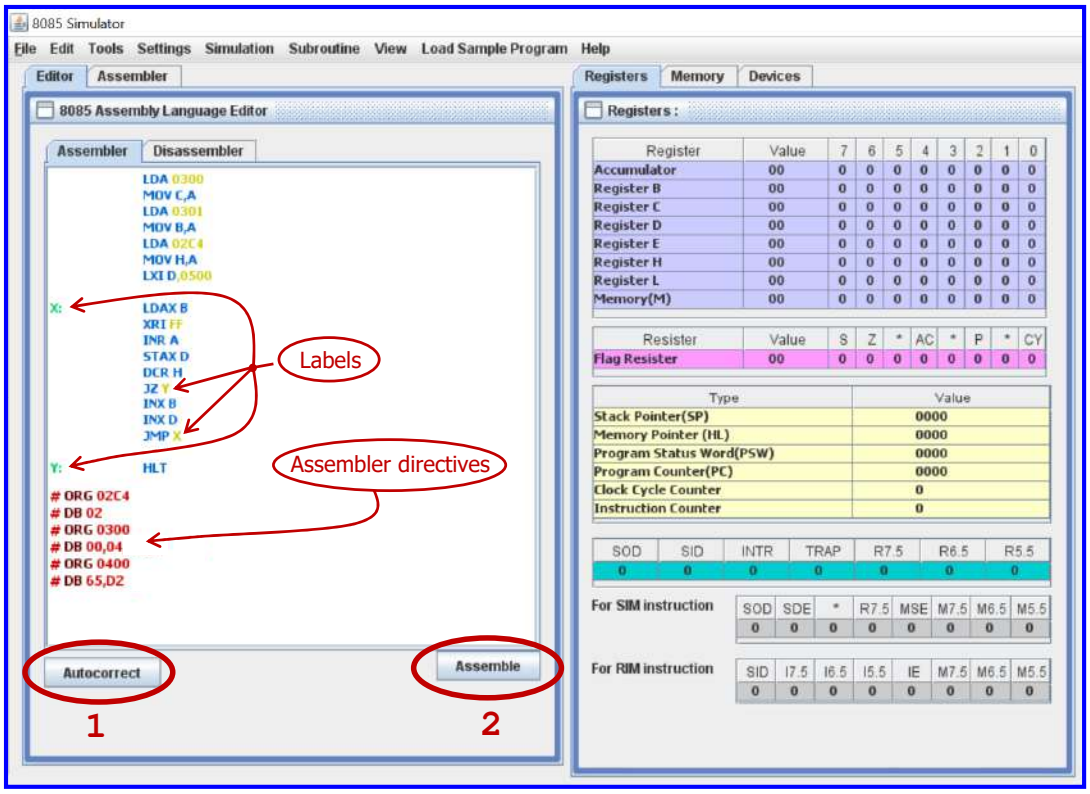
```
# ORG 02C4
# DB 02
# ORG 0300
# DB 00,04
# ORG 0400
# DB 65,D2
```


Παράδειγμα 3ο: προσομοίωση

- Έχουν αναπτυχθεί αρκετοί προσομοιωτές λειτουργίας του μικροεπεξεργαστή Intel 8085.
- Ενδεικτικά παραδείγματα προσομοιωτών:
  - ✓ Jubin's 8085 simulator (jar format) by Jubin Mitra, https://8085simulator.github.io
  - ✓ 8sj 8085 simulator (jar format) by John Sinu, Kurian Jinsmon, Devassy Deepu, https://sourceforge.net/projects/j8085sim
  - ✓ Sim8085 (online simulator) by Debjit Biswas, https://www.sim8085.com
  - ✓ GNU 8085 simulator by Sridhar Ratnakumar, https://gnusim8085.github.io
- Για το παράδειγμα μας θα χρησιμοποιήσουμε τον πρώτο από τους προαναφερόμενους προσομοιωτές που είναι διαθέσιμος σε java virtual machine executable format (.jar).
- Ο προσομοιωτής αυτός παρέχει:
  - ✓ συντάκτη συμβολικού κώδικα (editor) και συμβολομεταφραστή (assembler),
  - ✓ παρακολούθηση περιεχομένων καταχωρητών και θέσεων μνήμης,
  - ✓ υποστήριξη οδηγιών συμβολομεταφραστή και εκτέλεση κώδικα ανά εντολή,
  - ✓ στατιστικά στοιχεία προγραμμάτων (αριθμός εντολών, χρόνος εκτέλεσης κ.ά.).
  - ✓ παρουσίαση χρονοδιαγραμμάτων των κύκλων μηχανής κάθε εντολής.

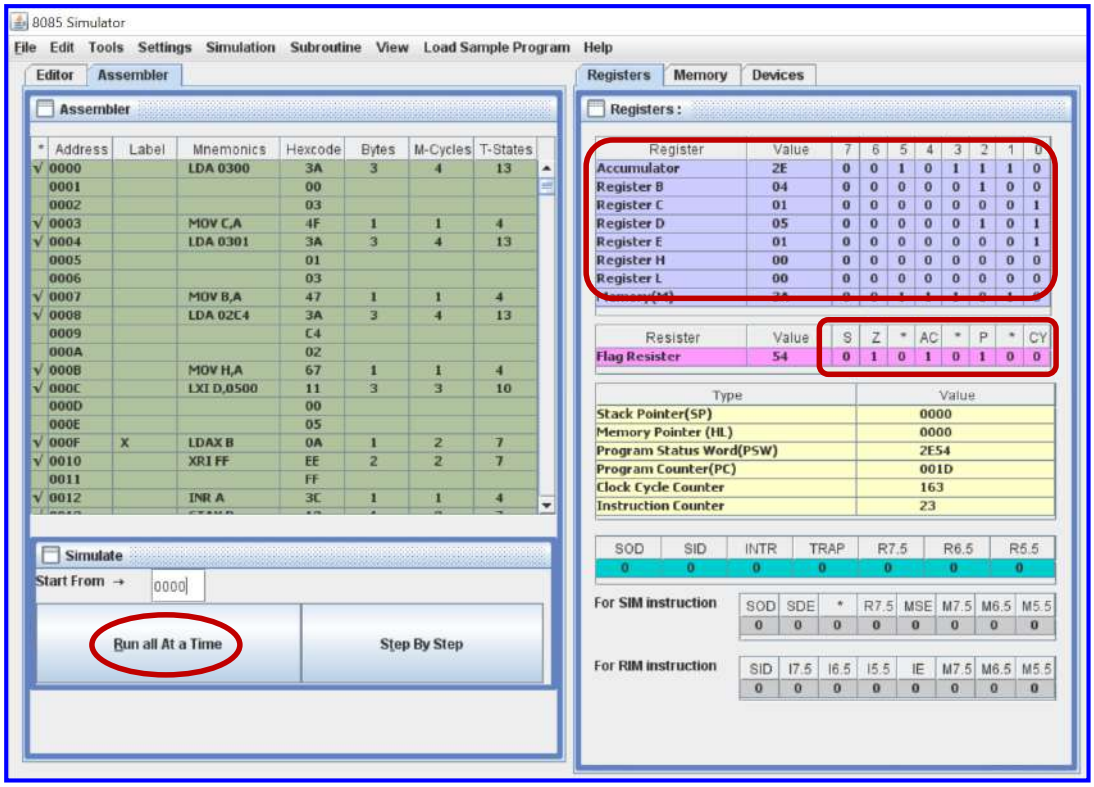
Παράδειγμα 3ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra



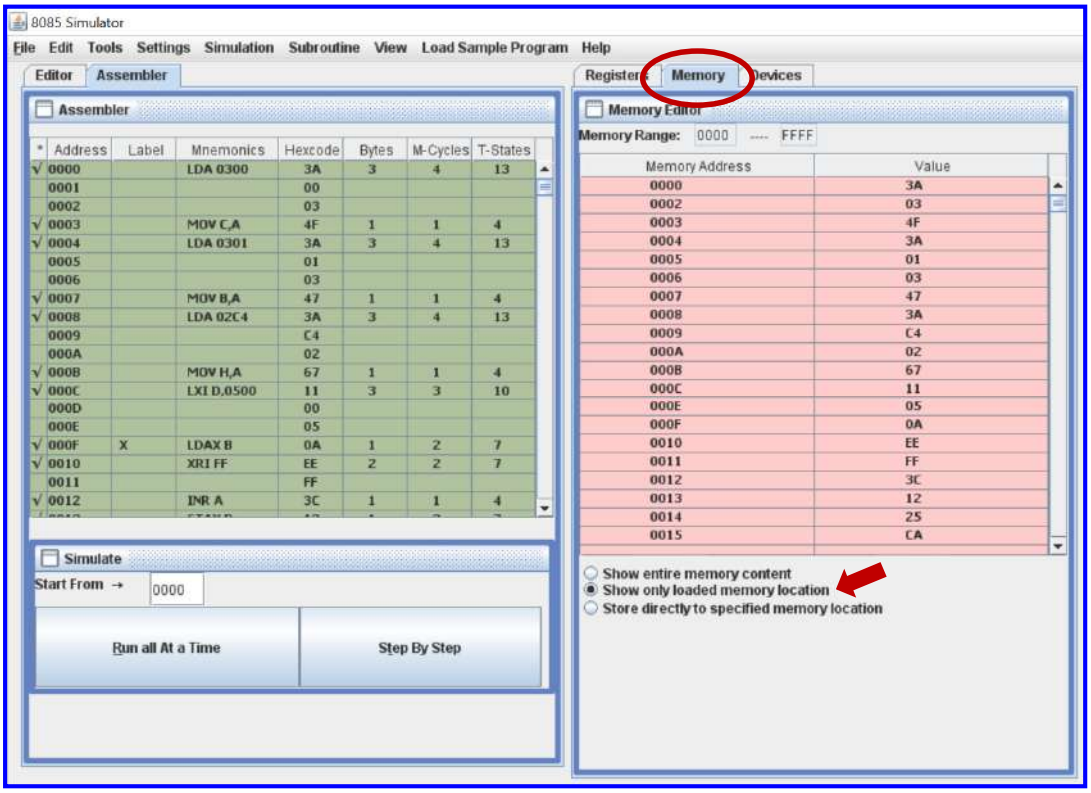
Παράδειγμα 3ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra



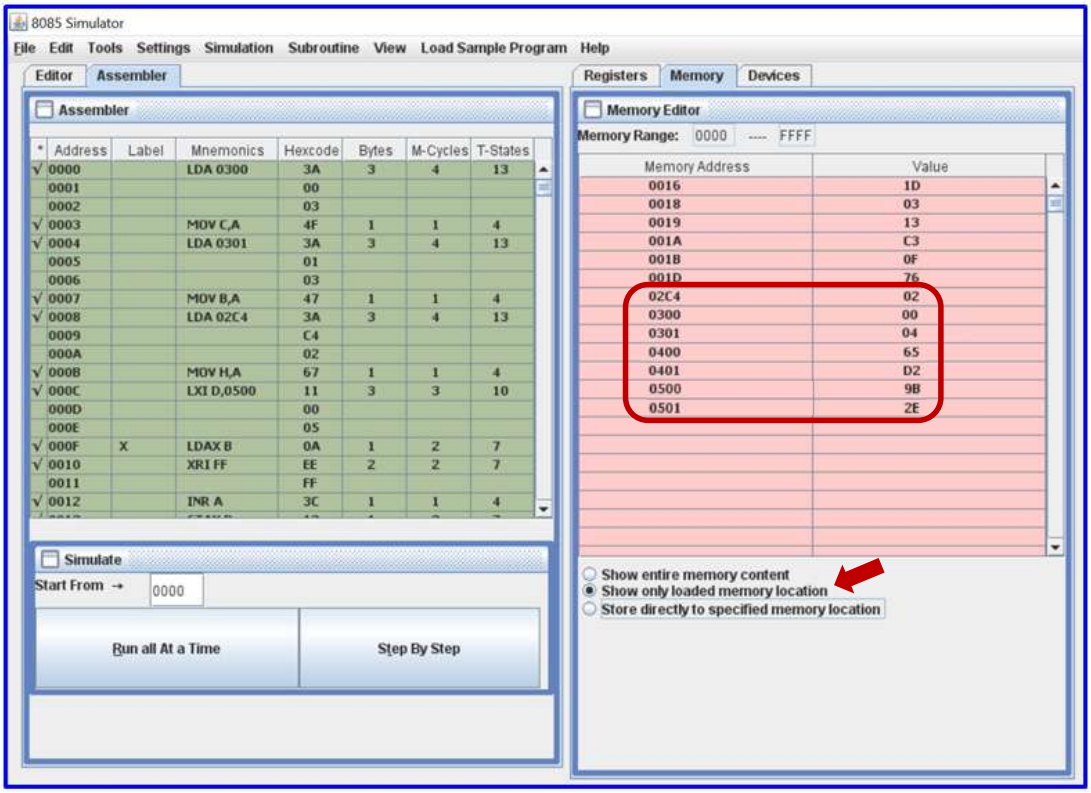
Παράδειγμα 3ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra



Παράδειγμα 3ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra



Παράδειγμα 4ο

Αρκετές φορές απαιτείται από τη φύση του προβλήματος για το οποίο γράφεται ένα πρόγραμμα, να εισαχθεί ο μικροεπεξεργαστής σε μία αλληλουχία εκτέλεσης εντολών, με σκοπό να απασχοληθεί (καθυστερήσει) τόσο χρόνο όσο απαιτείται από τις συνθήκες του προβλήματος. Τότε, απαιτείται η δημιουργία μιας υπορουτίνας καθυστέρησης.

Έστω ότι διαθέτουμε την έκδοση του μικροεπεξεργαστή Intel 8085 με συχνότητα λειτουργίας 3 MHz (δηλαδή, με περίοδο ή κύκλο σήματος ρολογιού T = 1/3 μs) και ζητείται να συντάξουμε υπορουτίνα που να τον καθυστερεί 508 μs.

Παράδειγμα 4ο: συμβολικό πρόγραμμα

- Η αντιμετώπιση του προβλήματος, ανάγεται στη δημιουργία ενός βρόχου (loop), που να ελαττώνει την τιμή ενός καταχωρητή, στον οποίο έχει τεθεί αρχική τιμή που δείχνει πόσες φορές θα πρέπει να εκτελεστεί ο βρόχος.

	MVI A,Count	Φορτώνεται στον συσσωρευτή A η αρχική τιμή Count (άγνωστη προς το παρόν)
LOOP	DCR A	Μείωση της τιμής του A κατά 1
	JNZ LOOP	Επανάληψη του βρόχου εάν η τιμή του A δεν είναι 0, διαφορετικά συνέχεια στην επόμενη εντολή
	RET	Τερματισμός υπορουτίνας καθυστέρησης και επιστροφή στο κυρίως πρόγραμμα.

- Στη συνέχεια, πρέπει να υπολογίσουμε την τιμή Count, έτσι ώστε να επιτυγχάνεται η καθυστέρηση των 508 μs.

Παράδειγμα 4ο: συμβολικό πρόγραμμα

- Στο εγχειρίδιο (datasheet) του επεξεργαστή εντοπίζουμε τους χρόνους εκτέλεσης των εντολών του προγράμματος σε κύκλους ρολογιού ή καταστάσεις (states):

	MVI A,Count	Χρόνος εκτέλεσης: 7·T
LOOP	DCR A	Χρόνος εκτέλεσης: 4·T
	JNZ LOOP	Χρόνος εκτέλεσης: 7·T όταν δεν επαναλαμβάνεται ο βρόχος και 10·T όταν επαναλαμβάνεται
	RET	Χρόνος εκτέλεσης: 10·T

- Η μικρότερη καθυστέρηση προκύπτει όταν αρχικά A = 1 και η μεγαλύτερη όταν αρχικά A = 0, αφού για αρχική τιμή A = 0 με την πρώτη εκτέλεση της εντολής DCR προκύπτει $A - 1 = FF_{16} = 255_{10}$, με αποτέλεσμα το μέγιστο δυνατό πλήθος εκτελέσεων του βρόχου.

$$\text{Ελάχιστη καθυστέρηση} = (7 + 4 + 7 + 10)\cdot T = 28\cdot T = 9.33\ \mu s.$$

$$\text{Μέγιστη καθυστέρηση} = (7 + 4 + 7 + 10)\cdot T + 255\cdot(4 + 10)\cdot T =$$

$$= \text{Ελάχιστη καθυστέρηση} + 255\cdot(4 + 10)\cdot T = 3598\cdot T = 1199.33\ \mu s.$$

- Συνεπώς, για να πετύχουμε την επιθυμητή καθυστέρηση των 508 μs ή 1524·T, πρέπει:

$$1524\cdot T = \text{Ελάχιστη καθυστέρηση} + X\cdot(4 + 10)\cdot T \Rightarrow$$

$$1524\cdot T = 28\cdot T + 14\cdot T\cdot X \Rightarrow X = 106.86.$$

Παράδειγμα 4ο: συμβολικό πρόγραμμα

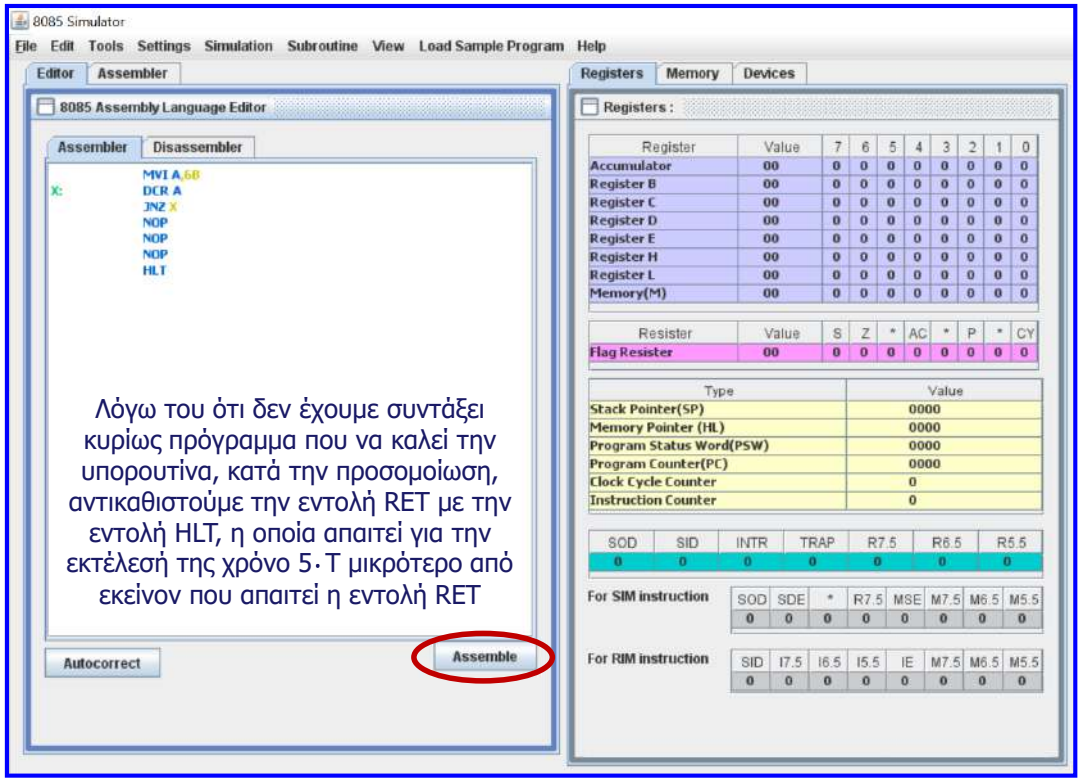
- Επιλέγουμε X = 106, δηλαδή, Count = X + 1 = 107 = $6B_{16}$ (αρχική τιμή που φορτώνεται στον καταχωρητή A).
- Η τιμή του Count που υπολογίστηκε, οδηγεί σε συνολική καθυστέρηση:

  Ελάχιστη καθυστέρηση + 106·(4 + 10)·T = 28·T + 1484·T = 1512·T = 504 μs.
- Λόγω του ότι η ζητούμενη καθυστέρηση είναι 508 μs, επιλέγουμε να καλύψουμε τον υπολειπόμενο χρόνο των 4 μs ή 12·T με την προσθήκη, πριν την εντολή RET, τριών εντολών NOP (no operation) που η καθεμία απαιτεί για την εκτέλεσή της χρόνο 4·T.

	MVI A,6B	Φορτώνεται στον A η αρχική τιμή $6B_{16} = 107_{10}$
LOOP	DCR A	Μείωση της τιμής του A κατά 1
	JNZ LOOP	Επανάληψη του βρόχου εάν η τιμή του A δεν είναι 0, διαφορετικά συνέχεια στην επόμενη εντολή
	NOP	Καμία λειτουργία
	NOP	Καμία λειτουργία
	NOP	Καμία λειτουργία
	RET	Τερματισμός υπορουτίνας καθυστέρησης και επιστροφή στο κυρίως πρόγραμμα.

Παράδειγμα 4ο: προσομοίωση

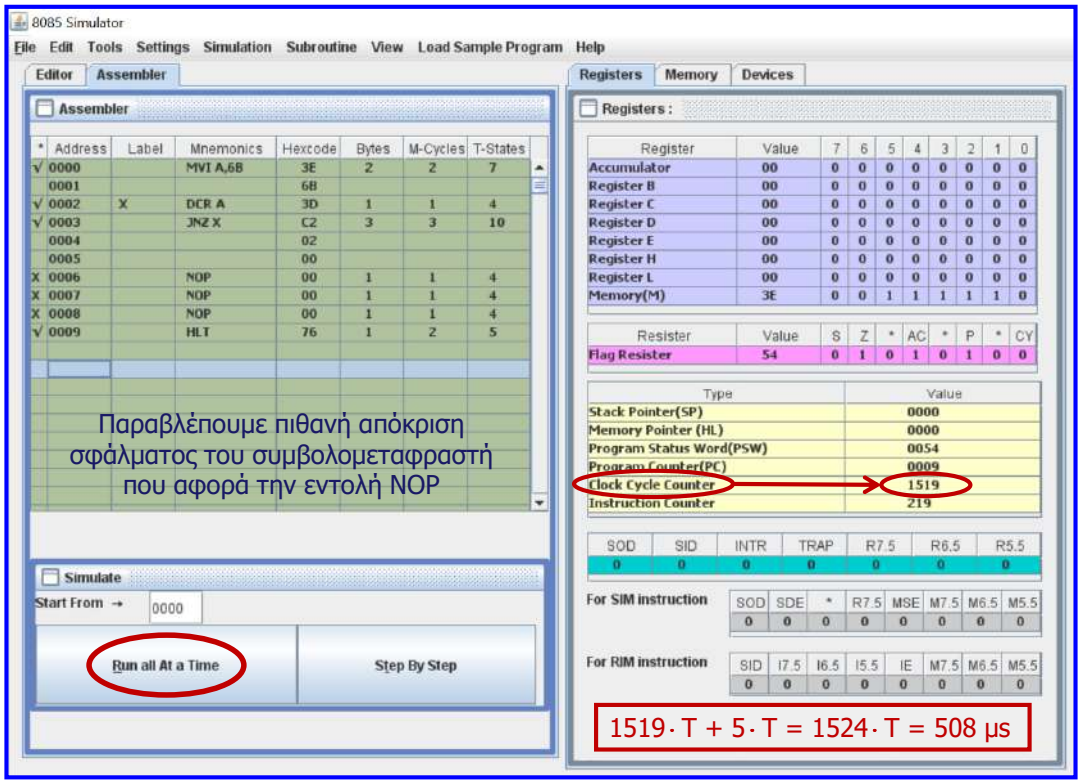
Jubin's 8085 Simulator
© Jubin Mitra



Λόγω του ότι δεν έχουμε συντάξει κυρίως πρόγραμμα που να καλεί την υπορουτίνα, κατά την προσομοίωση, αντικαθιστούμε την εντολή RET με την εντολή HLT, η οποία απαιτεί για την εκτέλεσή της χρόνο 5·T μικρότερο από εκείνον που απαιτεί η εντολή RET

Παράδειγμα 4ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra



Παραβλέπουμε πιθανή απόκριση σφάλματος του συμβολομεταφραστή που αφορά την εντολή NOP

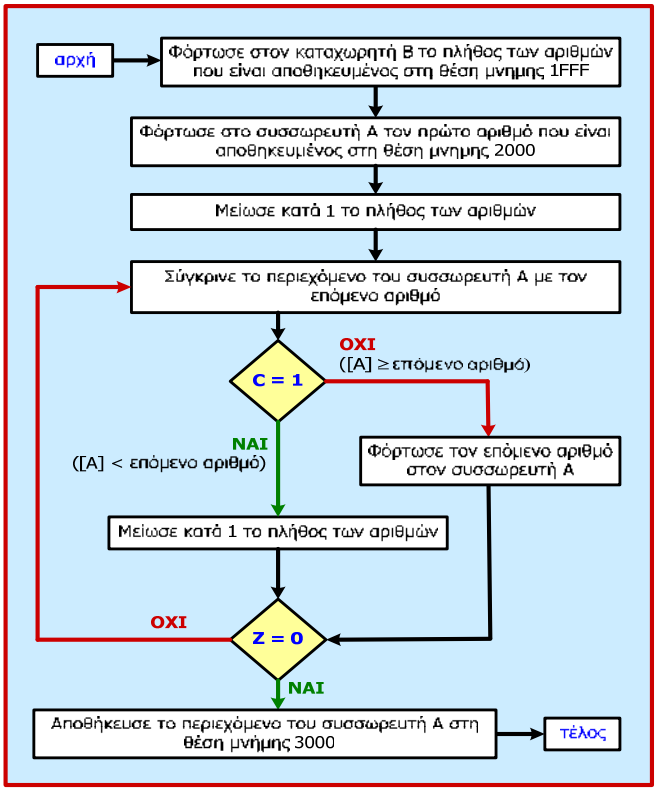
1519·T + 5·T = 1524·T = 508 μs

Παράδειγμα 5ο

Σύνταξη προγράμματος υπολογισμού του ελάχιστου μεταξύ Ν μη προσημασμένων αριθμών, με 8 δυαδικά ψηφία ο καθένας, όταν το πλήθος των αριθμών είναι αποθηκευμένο στην θέση μνήμης 1FFF και οι αριθμοί είναι αποθηκευμένοι στις θέσεις μνήμης από την διεύθυνση 2000 και μετά. Ο ελάχιστος αριθμός θα πρέπει να αποθηκεύεται στη θέση μνήμης με διεύθυνση 3000.

Παράδειγμα 5ο: διάγραμμα ροής

Διενεργούμε σύγκριση των αριθμών ανά ζεύγη με αποθήκευση κάθε φορά του μικρότερου αριθμού στον συσσωρευτή

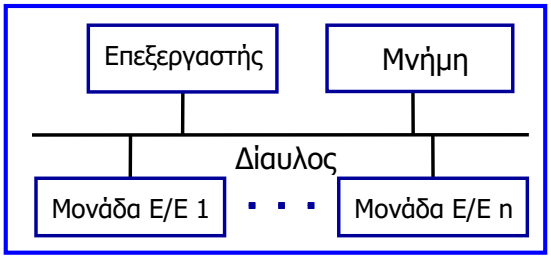


Παράδειγμα 5ο: συμβολικό πρόγραμμα

Διεύθυνση	Κώδικας	Σχόλια
0000	LXI H,1FFF	Φόρτωσε στο ζεύγος H-L τη διεύθυνση μνήμης του πλήθους N
0003	MOV B,M	Φόρτωσε στον B το πλήθος N των αριθμών
0004	INX H	Αύξησε κατά 1 το περιεχόμενο του ζεύγους H-L, ώστε να ισούται με τη διεύθυνση μνήμης 2000 που είναι αποθηκευμένος ο 1ος αριθμός
0005	MOV A,M	Φόρτωσε στον A τον πρώτο αριθμό
0006	DCR B	Μείωσε κατά 1 το περιεχόμενο του B (πλήθος αριθμών)
0007	INX H	Αύξησε κατά 1 το περιεχόμενο του ζεύγους H-L, ώστε να ισούται με τη διεύθυνση μνήμης που είναι αποθηκευμένος ο επόμενος αριθμός
0008	CMP M	Σύγκρινε το περιεχόμενο του A με τον επόμενο αριθμό
0009	JC 000D	Αν προκύψει κρατούμενο (A < επόμενο αριθμό) εξέτασε τον αριθμό που έπεται στη λίστα
000C	MOV A,M	Αλλιώς (A ≥ επόμενο αριθμό), φόρτωσε στον A τον επόμενο αριθμό
000D	DCR B	Μείωσε κατά 1 το περιεχόμενο του B (πλήθος αριθμών)
000E	JNZ 0007	Εάν δεν ολοκληρώθηκε η επεξεργασία όλων των αριθμών, εξέτασε τον αριθμό που έπεται στη λίστα
0011	STA 3000	Αλλιώς, αποθήκευσε το περιεχόμενο του A στη θέση μνήμης 3000
0014	HLT	Τέλος. Ο ελάχιστος αριθμός βρίσκεται στη θέση μνήμης 3000

Κεφάλαιο 4: Λειτουργίες εισόδου/εξόδου (Ε/Ε)

Προσπέλαση μονάδων εισόδου/εξόδου

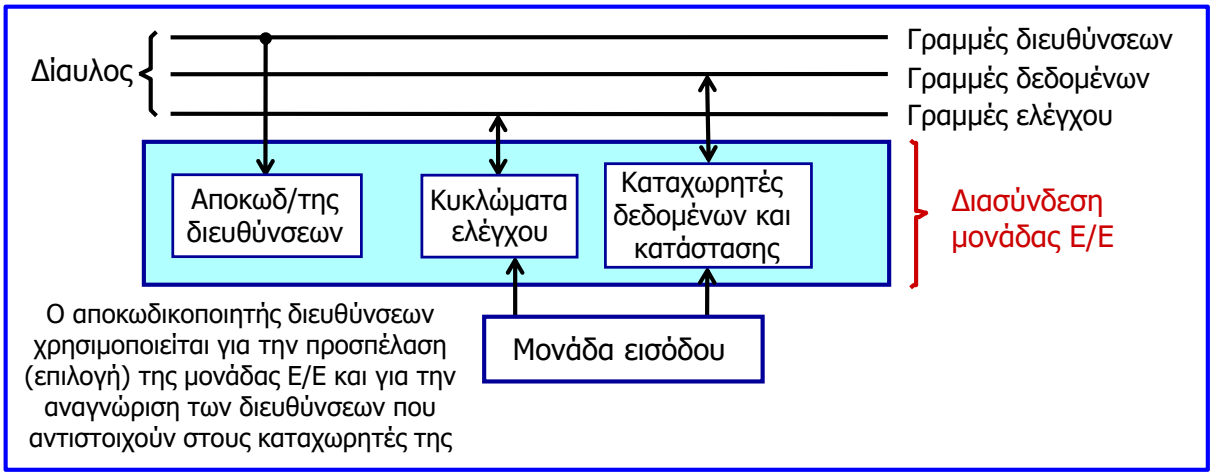


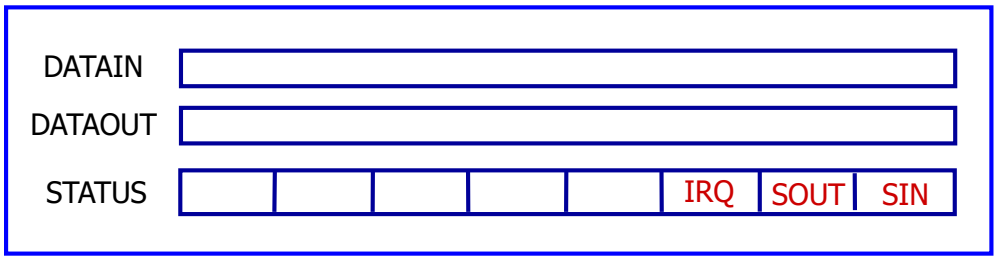
- Για την προσπέλαση περιφερειακών μονάδων (μονάδες εισόδου/εξόδου ή Ε/Ε), δηλαδή για ανταλλαγή δεδομένων με μονάδες Ε/Ε, χρησιμοποιείται συνήθως η τεχνική που αναφέρεται ως Ε/Ε απεικονιζόμενη στη μνήμη (memory-mapped I/O), στην οποία ορισμένες διευθύνσεις μνήμης χρησιμοποιούνται για αναφορά σε καταχωρητές μονάδων Ε/Ε, ώστε να μην απαιτούνται ειδικές εντολές για την προσπέλασή τους και να αρκούν εντολές όπως οι εντολές Move, Load, Store κ.ά.
- Ωστόσο, κάποιοι επεξεργαστές διαθέτουν και ειδικές εντολές εισόδου και εξόδου (π.χ. IN, OUT) για να εκτελούν μεταφορές Ε/Ε και ξεχωριστό χώρο διευθύνσεων για τις μονάδες Ε/Ε (I/O-mapped I/O ή peripheral-mapped I/O).
- Κατάλληλα σήματα ελέγχου στο δίαυλο, υποδεικνύουν ότι μία αιτούμενη μεταφορά αποτελεί λειτουργία Ε/Ε και τότε η μονάδα μνήμης αγνοεί το σχετικό αίτημα.

Τεχνικές υλοποίησης λειτουργιών Ε/Ε

- Οι βασικές μέθοδοι ή τεχνικές που χρησιμοποιούνται στα υπολογιστικά συστήματα για την υλοποίηση λειτουργιών Ε/Ε είναι:
  - ✓ Λειτουργίες Ε/Ε ελεγχόμενες από πρόγραμμα με επερώτηση (program-controlled I/O with polling).
  - ✓ Λειτουργίες Ε/Ε με χρήση διακοπών (interrupt-driven I/O).
  - ✓ Λειτουργίες Ε/Ε με απευθείας πρόσβαση στη μνήμη (direct memory access, DMA).
- Όταν χρησιμοποιείται λειτουργία Ε/Ε ελεγχόμενη από πρόγραμμα με επερώτηση ο επεξεργαστής διαβάζει (ελέγχει) περιοδικά μέσω βρόχου επανάληψης τον καταχωρητή κατάστασης (status register) της διασύνδεσης της μονάδας Ε/Ε (δηλαδή, ο επεξεργαστής εκτελεί περιοδική δειγματοληψία στη μονάδα ή στις μονάδες Ε/Ε).
- Ο επεξεργαστής διενεργεί μεταφορά δεδομένων από ή προς τους καταχωρητές δεδομένων της διασύνδεσης της μονάδας Ε/Ε, όταν η τιμή της αντίστοιχης σημαίας κατάστασης (SIN ή SOUT) τίθεται από την μονάδα Ε/Ε σε τιμή 1, δηλώνοντας ότι έχει ολοκληρώσει την ενέργεια που της ανατέθηκε και είναι έτοιμη να διαθέσει ή να λάβει δεδομένα.
- Μετά τη μεταφορά των δεδομένων, ο επεξεργαστής ή η διασύνδεση της μονάδας Ε/Ε μηδενίζει την αντίστοιχη σημαία του καταχωρητή κατάστασης της διασύνδεσης.

Λειτουργίες Ε/Ε ελεγχόμενες από πρόγραμμα

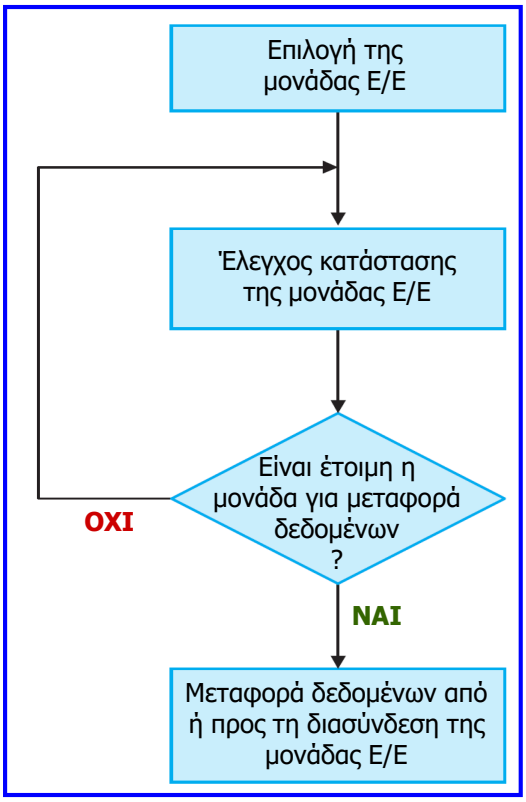




Καταχωρητές διασύνδεσης μονάδας εισόδου-εξόδου

Λειτουργίες Ε/Ε ελεγχόμενες από πρόγραμμα

- Η συχνότητα με την οποία γίνεται ανάγνωση του καταχωρητή κατάστασης της διασύνδεσης της μονάδας Ε/Ε, δεν πρέπει να είναι μικρότερη από τη μέγιστη συχνότητα με την οποία μπορεί να αλλάξει η κατάσταση της μονάδας Ε/Ε.
- Για παράδειγμα, στην περίπτωση όπου η μονάδα εισόδου είναι το πληκτρολόγιο, ο επεξεργαστής πρέπει να διαβάζει τον καταχωρητή κατάστασης της διασύνδεσης του πληκτρολογίου με συχνότητα ίση ή μεγαλύτερη από τη μέγιστη δυνατή συχνότητα πληκτρολόγησης (δηλαδή, τη συχνότητα πληκτρολόγησης που δεν μπορεί να ξεπεράσει ο άνθρωπος).
- Η λειτουργία Ε/Ε ελεγχόμενη από πρόγραμμα με επερώτηση είναι απλή διαδικασία, ωστόσο όταν η ταχύτητα της μονάδας Ε/Ε είναι υψηλή ή/και το πλήθος των μονάδων Ε/Ε είναι μεγάλο, τότε ο χρόνος που δαπανά ο επεξεργαστής για τον έλεγχο της κατάστασης των μονάδων Ε/Ε είναι απαγορευτικά μεγάλος.



Παράδειγμα

- Θεωρήστε υπολογιστή με επεξεργαστή συχνότητας λειτουργίας 1 GHz, ο οποίος, εκτός των άλλων, χρησιμοποιείται για τον έλεγχο λειτουργίας 1000 μονάδων Ε/Ε μέσω λειτουργιών Ε/Ε ελεγχόμενων από πρόγραμμα με επερώτηση. Ο καταχωρητής κατάστασης κάθε μονάδας θα πρέπει να ελέγχεται 10 φορές το δευτερόλεπτο και για τον έλεγχο και τη μεταφορά της απαιτούμενης πληροφορίας ελέγχου για κάθε μονάδα απαιτούνται 1000 κύκλοι ρολογιού. Ποιο είναι το ποσοστό του χρόνου του επεξεργαστή που δαπανάται για τον έλεγχο της κατάστασης των μονάδων;

- Αφού η συχνότητα λειτουργίας του επεξεργαστή είναι 1 GHz, η διάρκεια ενός κύκλου ρολογιού (CPU clock cycle) είναι 1 ns.
- Κάθε δευτερόλεπτο (s) ο επεξεργαστής απασχολείται για τον έλεγχο των μονάδων για χρόνο: $10 \times 1000 \times 1000 \times 1 \times 10^{-9}$ s = 0.01 s.
- Συνεπώς, το ποσοστό του χρόνου του επεξεργαστή που δαπανάται για τον έλεγχο της κατάστασης των μονάδων είναι: (0.01 / 1) x 100 = 1%.
- Για τη μεταφορά δεδομένων μεταξύ του επεξεργαστή και των μονάδων Ε/Ε που χρειάζονται εξυπηρέτηση, απαιτείται πρόσθετος χρόνος.

Διακοπές (interrupts)

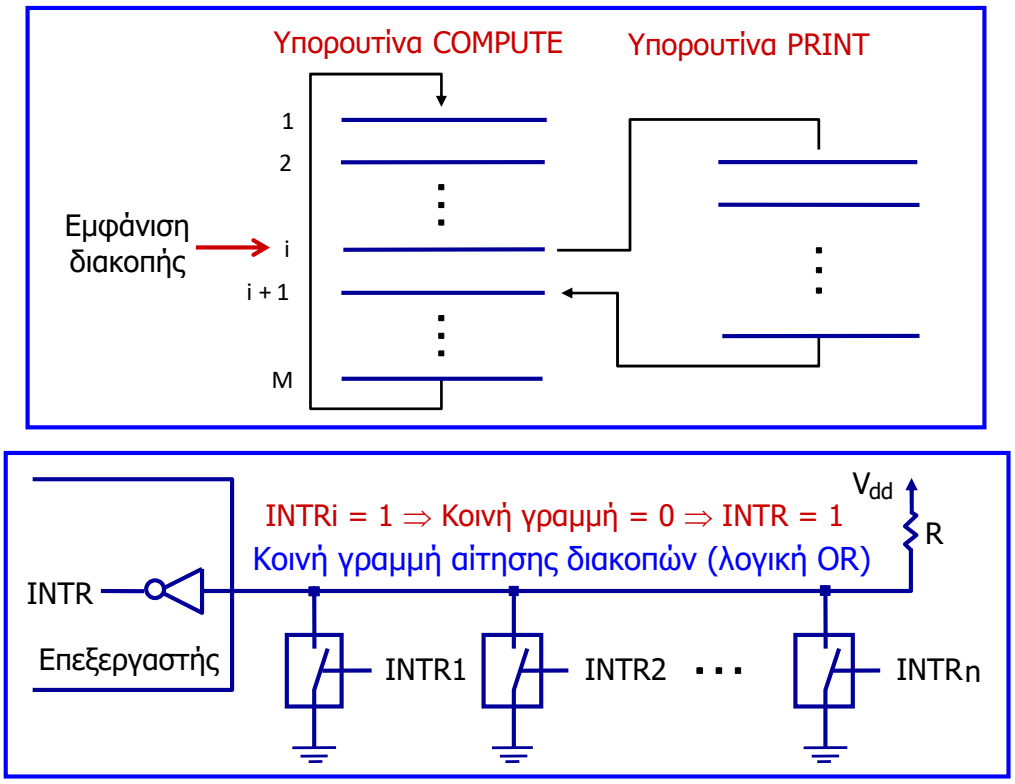
- Βασικό μειονέκτημα της προηγούμενης μεθόδου είναι ότι κατά τη διαδικασία της επερώτησης ο επεξεργαστής δεν εκτελεί άλλες χρήσιμες λειτουργίες.
- Στην περίπτωση του μηχανισμού των διακοπών, ο συγχρονισμός επιτυγχάνεται με τη μονάδα Ε/Ε να στέλνει ειδικό σήμα διακοπής (interrupt) προς τον επεξεργαστή διαμέσου του διαύλου όταν είναι έτοιμη για μεταφορά δεδομένων και τον επεξεργαστή να το αναγνωρίζει με σήμα επιβεβαίωσης διακοπής (interrupt-acknowledge).
- Για το σκοπό αυτό διατίθεται μία γραμμή ελέγχου (αίτησης διακοπής, interrupt request) και η υπορουτίνα που εκτελείται σε απόκριση μιας αίτησης διακοπής ονομάζεται ρουτίνα εξυπηρέτησης διακοπών (interrupt service routine, ISR).
- Οι περίοδοι αναμονής αξιοποιούνται από τον επεξεργαστή για άλλες λειτουργίες.
- Η αποθήκευση από τον επεξεργαστή των καταχωρητών που χρησιμοποιούνται πριν την εκτέλεση της ISR εισάγει καθυστέρηση στην εκτέλεση του προγράμματος, που μαζί με την καθυστέρηση μεταξύ της λήψης αίτησης διακοπής και της έναρξης εξυπηρέτησής της, συνιστούν την καθυστέρηση διακοπών (interrupt latency).
- Σε μερικές περιπτώσεις αποθηκεύεται μόνο το περιεχόμενο του PC και του SR, ενώ το περιεχόμενο των υπόλοιπων καταχωρητών αποθηκεύεται στην αρχή της ISR.
- Σε κάποιους επεξεργαστές είναι διαθέσιμα αντίγραφα καταχωρητών που χρησιμοποιούνται από ρουτίνες εξυπηρέτησης διακοπών (π.χ. ARM).

Παράδειγμα

- Θεωρήστε υπολογιστή με επεξεργαστή συχνότητας λειτουργίας 1 GHz, που εκτός των άλλων, χρησιμοποιείται για τον έλεγχο λειτουργίας 1000 μονάδων Ε/Ε, οι οποίες έχουν τη δυνατότητα αποστολής σημάτων διακοπής, ώστε να εξυπηρετηθούν από τον επεξεργαστή. Το κόστος διακοπής λειτουργίας του επεξεργαστή για την εξυπηρέτηση της αιτούμενης μονάδας μαζί με το κόστος εξυπηρέτησης της μονάδας (μεταφορά δεδομένων) είναι 1300 κύκλοι ρολογιού. Κατά μέσο όρο οι μονάδες Ε/Ε στέλνουν συνολικά 500 σήματα διακοπής το λεπτό. Ποιο είναι το ποσοστό του χρόνου του επεξεργαστή που δαπανάται για την εξυπηρέτηση των μονάδων;

- Αφού η συχνότητα λειτουργίας του επεξεργαστή είναι 1 GHz, η διάρκεια ενός κύκλου ρολογιού είναι 1 ns.
- Αφού κατά μέσο όρο όλες οι μονάδες στέλνουν συνολικά 500 σήματα διακοπής το λεπτό και το (χρονικό) κόστος κάθε διακοπής είναι 1300 κύκλοι ρολογιού, τότε στα 60 s του επεξεργαστή τα 500 x 1300 x 1 x 10^{-9} s = 0.00065 s, δαπανώνται για την εξυπηρέτηση των περιφερειακών μονάδων.
- Συνεπώς, το ποσοστό του χρόνου του επεξεργαστή που δαπανάται για την εξυπηρέτηση των περιφερειακών μονάδων (0.00065 / 60) x 100 = 0.001%.

Εξυπηρέτηση διακοπών (interrupts service)



Εξυπηρέτηση διακοπών (interrupts service)

- Απενεργοποίηση των διακοπών (π.χ. όταν εκτελείται ISR) μπορεί να γίνει με 3 τρόπους:
  - ✓ Χρησιμοποίηση εντολών όπως interrupt-enable και interrupt-disable.
  - ✓ Μέσω του μηδενισμού μιας σημαίας ενεργοποίησης διακοπών του καταχωρητή κατάστασης.
  - ✓ Μέσω ειδικού σήματος διακοπών για το οποίο το κύκλωμα χειρισμού διακοπών αντιδρά στην ανερχόμενη ακμή του και τότε ο επεξεργαστής λαμβάνει μία μόνο αίτηση διακοπής ανεξάρτητα από το χρόνο που η γραμμή είναι ενεργοποιημένη.
- Τυπικό σενάριο χειρισμού μίας διακοπής:
  - ✓ Μία μονάδα Ε/Ε αιτείται μία διακοπή.
  - ✓ Ο επεξεργαστής εκτελεί την τρέχουσα εντολή και διακόπτει το πρόγραμμα.
  - ✓ Οι διακοπές απενεργοποιούνται με τη σχετική αλλαγή στον καταχωρητή κατάστασης.
  - ✓ Η μονάδα Ε/Ε πληροφορείται ότι αναγνωρίστηκε η αίτησή της και απενεργοποιεί το σήμα αίτησης διακοπών.
  - ✓ Εκτελείται η ISR, οι διακοπές ενεργοποιούνται και συνεχίζεται η εκτέλεση του προγράμματος που είχε διακοπεί.

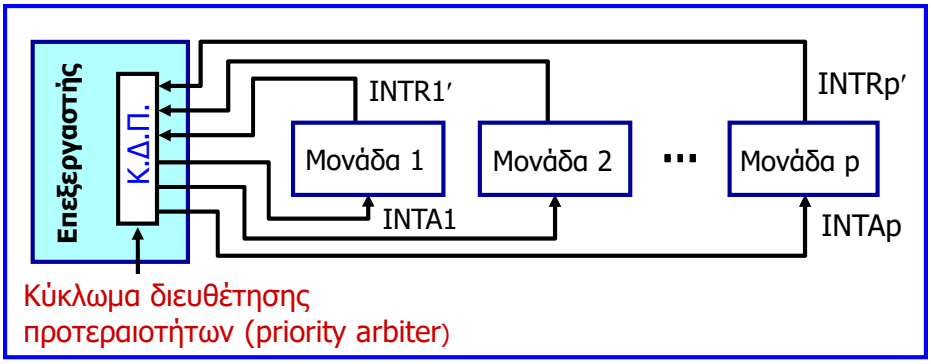
Χειρισμός διακοπών από πολλαπλές μονάδες Ε/Ε

- Στην περίπτωση πολλαπλών μονάδων Ε/Ε με κοινή γραμμή αίτησης διακοπής, όταν μία μονάδα εγείρει αίτηση διακοπής ενεργοποιεί ένα ψηφίο του καταχωρητή κατάστασης (IRQ).

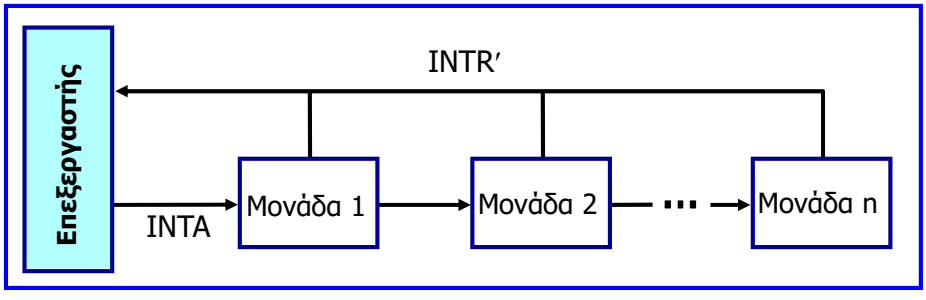
- Ο επεξεργαστής εκτελεί επερώτηση (polling) σε όλες τις μονάδες και η πρώτη που θα εντοπιστεί με ενεργοποιημένο το ψηφίο IRQ είναι αυτή που θα πρέπει να εξυπηρετηθεί (μειονέκτημα: χρόνος ανίχνευσης των IRQ).
- Εναλλακτικά χρησιμοποιούνται οι διανυσματικές διακοπές (vectored interrupts).
- Κάθε μονάδα που αιτείται διακοπή, στέλνει μέσω του διαύλου έναν ειδικό κωδικό (διάνυσμα διακοπής, interrupt vector) που αντιστοιχεί στη διεύθυνση έναρξης της ISR για την εν λόγω μονάδα, την οποία διαβάζει ο επεξεργαστής και την καταχωρεί στον PC.
- Το μήκος του κωδικού αυτού είναι συνήθως 4-8 bits και αντιστοιχεί σε μία καταχώρηση στην περιοχή μνήμης όπου βρίσκονται αποθηκευμένες οι διευθύνσεις των ISR (interrupt address table).
- Ο επεξεργαστής εκτελεί την τρέχουσα εντολή και όταν είναι έτοιμος να δεχθεί το διάνυσμα διακοπής ενεργοποιεί μια γραμμή επιβεβαίωσης λήψης διακοπών (interrupt-acknowledge line, INTA).

Ταυτόχρονες αιτήσεις διακοπών

Όταν εγείρονται ταυτόχρονες αιτήσεις διακοπών από μονάδες Ε/Ε, ο επεξεργαστής πρέπει να αποφασίσει τη σειρά εξυπηρέτησής τους.



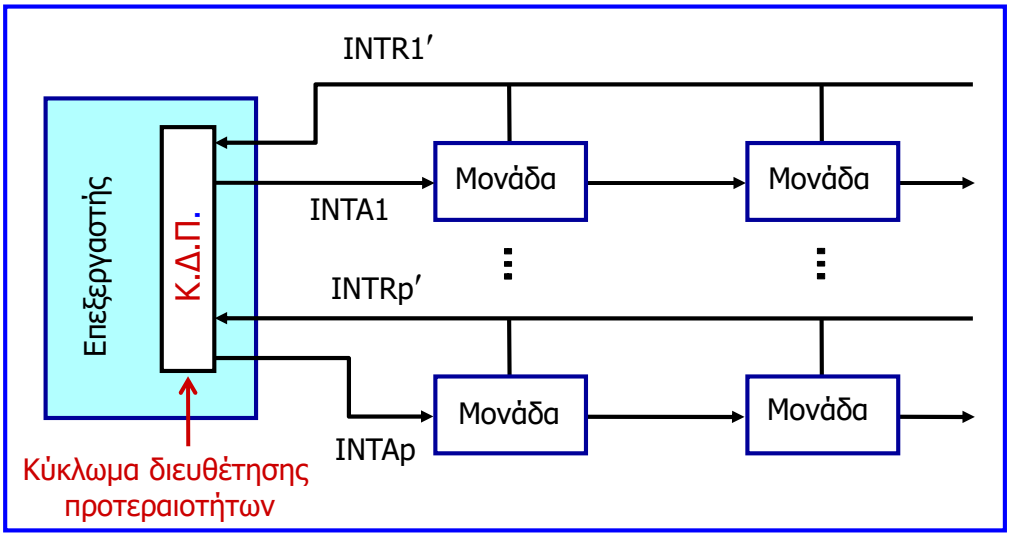
Διάταξη μονάδων με ξεχωριστές γραμμές αίτησης και επιβεβαίωσης λήψης διακοπών, όπου ο επεξεργαστής αποδέχεται την αίτηση με τη μεγαλύτερη προτεραιότητα.



Διάταξη μονάδων με κοινή γραμμή αίτησης διακοπών και σήμα επιβεβαίωσης διακοπών που διατρέχει όλες τις μονάδες που είναι συνδεδεμένες σε διάταξη αλυσίδας (daisy chain)

Ταυτόχρονες αιτήσεις διακοπών

- Συνδυασμός των προηγούμενων διατάξεων με οργάνωση των μονάδων σε ομάδες από τις οποίες καθεμία έχει διαφορετικό επίπεδο προτεραιότητας.
- Σε κάθε ομάδα οι μονάδες συνδέονται σε διάταξη αλυσίδας.
- Η διευθέτηση ομάδων προτεραιοτήτων είναι η οργάνωση που χρησιμοποιείται σήμερα σε πολλά υπολογιστικά συστήματα

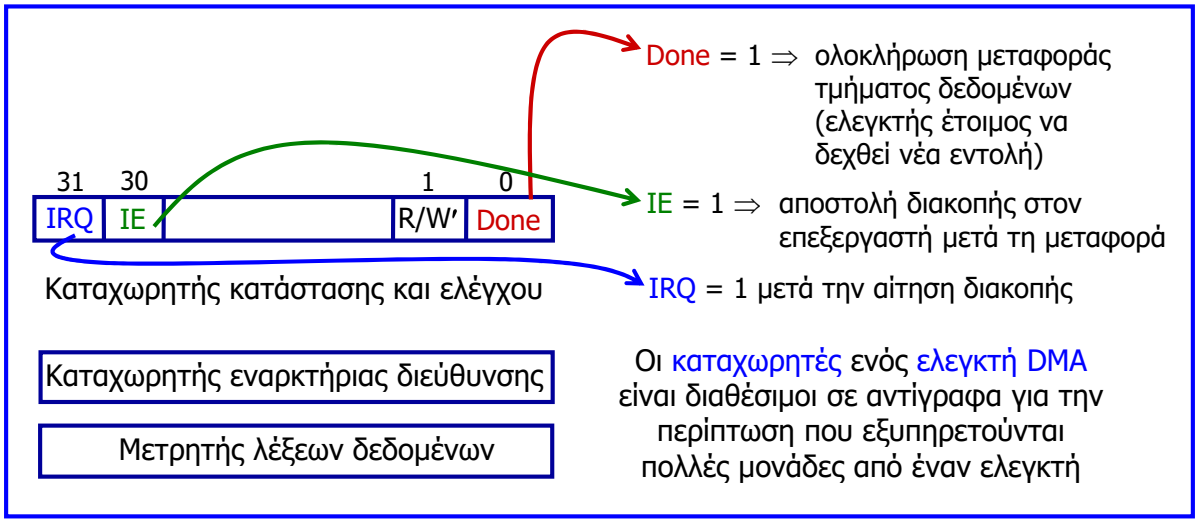


Απευθείας πρόσβαση μνήμης (DMA)

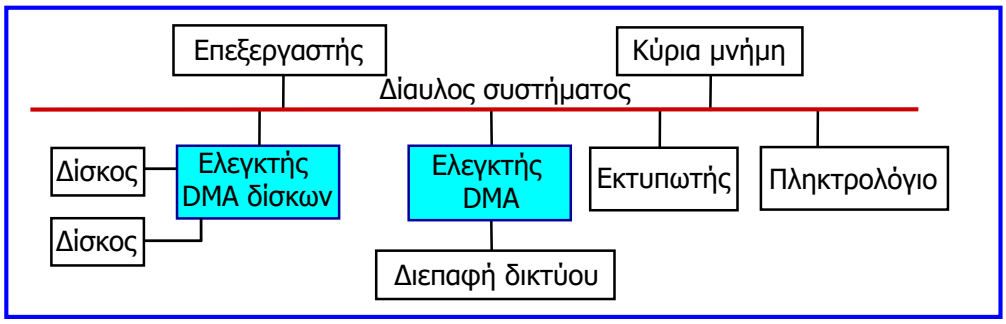
- Τα δεδομένα των μονάδων Ε/Ε θα πρέπει να αποθηκευτούν στη μνήμη, έτσι ώστε να έχει πρόσβαση σε αυτά ο επεξεργαστής.
- Αυτό γίνεται μέσω αίτησης διακοπής από τις μονάδες και εκτέλεση ISR, η οποία θα μεταφέρει τα δεδομένα στη μνήμη και στη συνέχεια ο επεξεργαστής θα συνεχίσει την εκτέλεση του προγράμματος.
- Ωστόσο, η μεταφορά του επεξεργαστή στην ISR προσθέτει καθυστέρηση αφού θα πρέπει να αποθηκευτεί η κατάσταση (καταχωρητές) του επεξεργαστή μέσω εκτέλεσης μιας σειράς εντολών και να αποκατασταθεί με την επιστροφή στην εκτέλεση του προγράμματος.
- Εναλλακτικά, χρησιμοποιείται η τεχνική που λέγεται απευθείας πρόσβαση στη μνήμη (direct memory access, DMA), μέσω της οποίας ένα τμήμα δεδομένων μεταφέρεται μεταξύ μιας μονάδας Ε/Ε και της μνήμης, χωρίς τη συνεχή παρεμβολή του επεξεργαστή, ο οποίος απελευθερώνεται για να χρησιμοποιηθεί στην εκτέλεση άλλου προγράμματος.
- Οι μεταφορές DMA επιτελούνται από ειδικό κύκλωμα (ελεγκτής DMA) που αποτελεί μέρος της διασύνδεσης των μονάδων Ε/Ε και για κάθε λέξη δεδομένων που μεταφέρεται παρέχει τη διεύθυνση μνήμης (αυξάνοντάς την κατάλληλα) και όλα τα σήματα διαύλου που ελέγχουν τη μεταφορά δεδομένων.

Απευθείας πρόσβαση μνήμης (DMA)

Πριν την έναρξη της μεταφοράς DMA, ο επεξεργαστής παρέχει την εναρκτήρια διεύθυνση, τον αριθμό των λέξεων και την κατεύθυνση μεταφοράς (μέσω σήματος R/W'), ενώ κατά τη λήξη ο ελεγκτής DMA στέλνει ένα σήμα διακοπής, ώστε να ενημερώσει για την ολοκλήρωση της μεταφοράς και ο επεξεργαστής να επιστρέψει στην εκτέλεση του προγράμματος το οποίο είχε αιτηθεί τη μεταφορά.

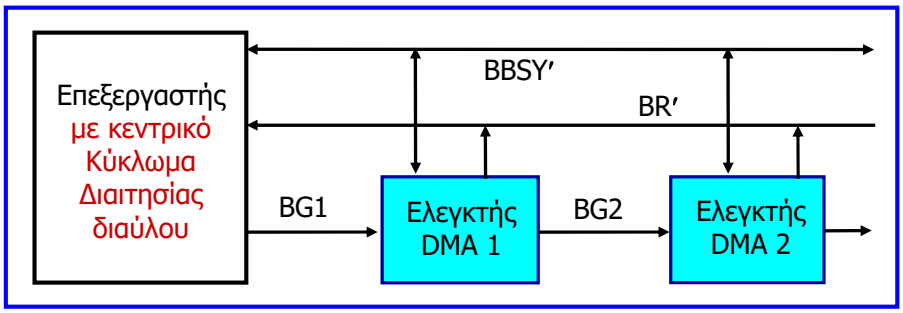


Απευθείας πρόσβαση μνήμης (DMA)



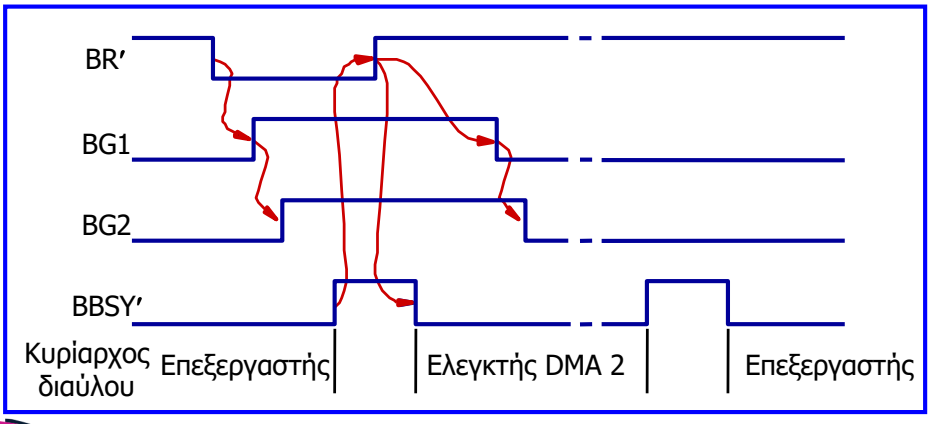
- Οι προσπελάσεις μνήμης του επεξεργαστή και του ελεγκτή DMA πολυπλέκονται, δηλαδή έχουν και οι δύο πρόσβαση στο δίαυλο του συστήματος.
- Οι αιτήσεις χρήσης διαύλου από ελεγκτές DMA έχουν συνήθως ανώτερη προτεραιότητα από εκείνες του επεξεργαστή και μέγιστη προτεραιότητα παρέχεται σε μονάδες Ε/Ε υψηλής ταχύτητας (π.χ. δίσκοι, διεπαφές δικτύου υψηλής ταχύτητας).
- Εναλλακτικά, μπορεί να ανατεθεί στον ελεγκτή DMA αποκλειστική πρόσβαση στο δίαυλο και κατά συνέπεια στην κύρια μνήμη, ώστε να μεταφέρει ένα τμήμα δεδομένων χωρίς παρεμβολή (block ή burst mode).
- Για ταχύτερο χειρισμό δεδομένων, συνήθως οι ελεγκτές DMA περιλαμβάνουν απομονωτή αποθήκευσης δεδομένων (data storage buffer), στον οποίο τοποθετούνται τα δεδομένα από την κύρια μνήμη και στη συνέχεια διατίθενται στη μονάδα Ε/Ε.

Επικεντρωμένη διαιτησία διαύλου



Διαιτησία διαύλου: (arbitration) τρόπος με τον οποίο επιλέγεται η μονάδα που πρόκειται να γίνει κυρίαρχος του διαύλου (bus master), δηλαδή η μονάδα που θα έχει τον έλεγχο του διαύλου για να εκτελεί μεταφορές δεδομένων.

Επικεντρωμένη διαιτησία διαύλου με διάταξη αλυσίδας (daisy-chain)

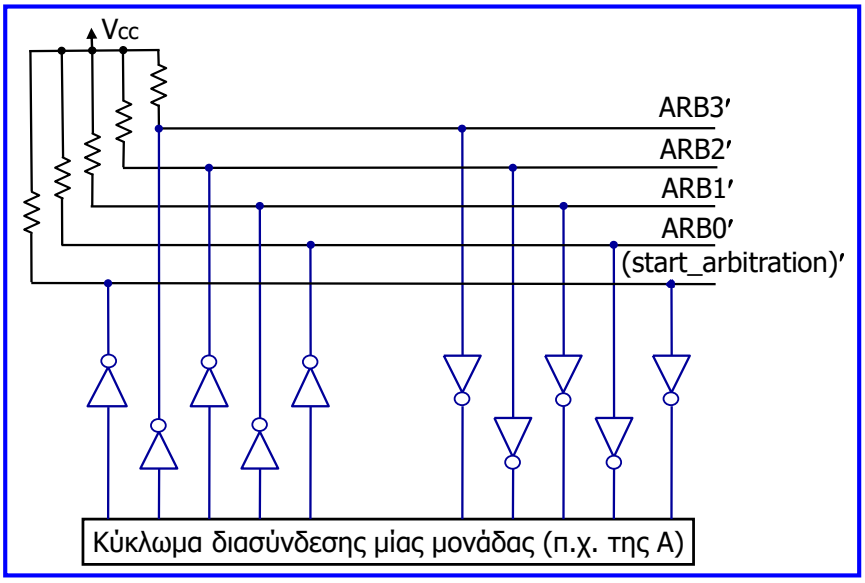


BGi: bus-grant signal (απόδοση διαύλου)

BR′: bus-request signal (αίτηση διαύλου)

BBSY′: bus-busy (απασχόληση διαύλου)

Κατανεμημένη διαιτησία διαύλου



Κατανεμημένη διαιτησία διαύλου: συμμετέχουν όλες οι μονάδες χωρίς να χρησιμοποιείται κεντρικό κύκλωμα διαιτησίας.

Παράδειγμα:

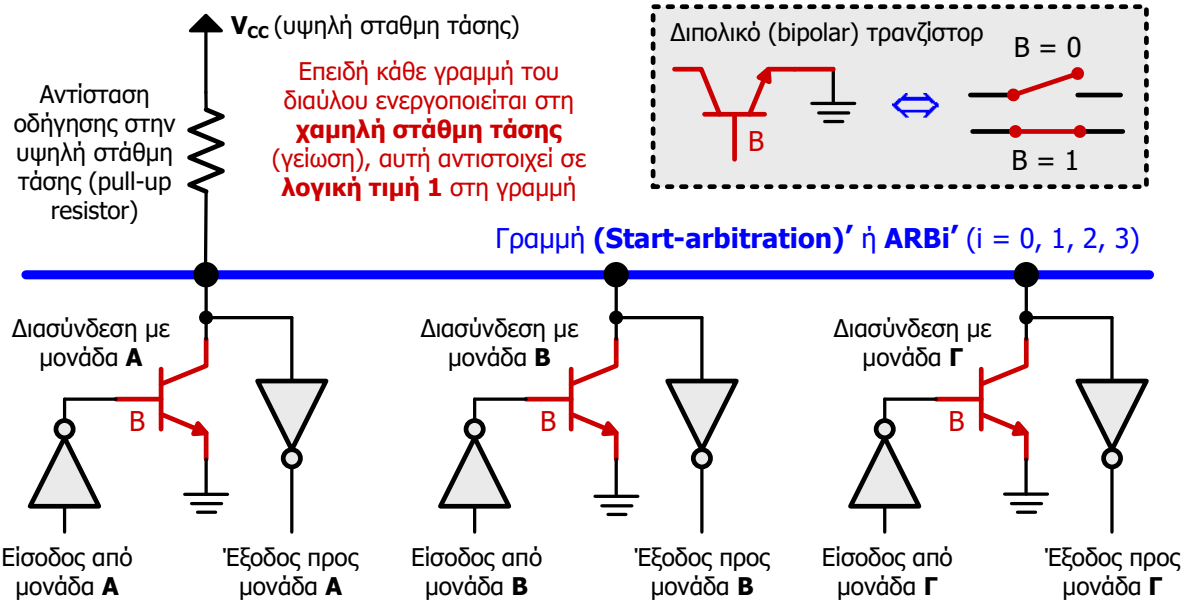
ID_A = 0101 (5)
ID_B = 0110 (6)

Αρχικά: ID_A OR ID_B = 0111

Η μονάδα Α λαμβάνει στη γραμμή ARB1′ λογική τιμή 1 # 0 και απενεργοποιεί τη λιγότερο σημαντική ARB0′ (θέτει τιμή 0 στην είσοδο του αντίστοιχου αντιστροφέα) και το αποτέλεσμα της νέας πράξης OR που λαμβάνει στη συνέχεια είναι: 0110 = ID_B.

- Οι μονάδες που αιτούνται χρήση του διαύλου ενεργοποιούν τη γραμμή εκκίνησης. Κάθε μονάδα διαθέτει 4 bits ταυτοποίησής της (ID) με βάση τα οποία ενεργοποιεί τις γραμμές ARB′.
- Το αποτέλεσμα του κυκλώματος διαιτησίας που προκύπτει από λογικό OR και γίνεται αντιληπτό από όλες τις μονάδες, είναι οι τιμές των 4 γραμμών που αναπαριστούν την αίτηση με το μεγαλύτερο ID.

Κατανεμημένη διαιτησία διαύλου

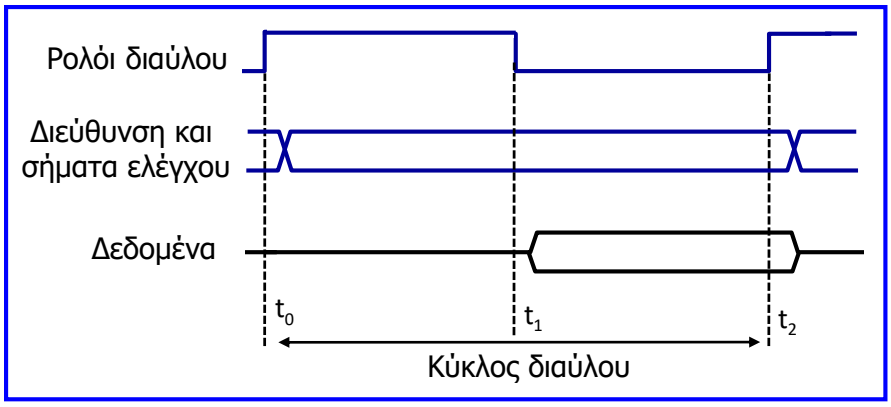


Η διάταξη (κύκλωμα ανοικτού συλλέκτη, open collector ή ανοικτού απαγωγού / υποδοχής, open drain όταν περιλαμβάνει MOSFETs) υλοποιεί λογικό άθροισμα (καλωδιωμενη λογική OR, wired-OR)

Σύγχρονοι δίαυλοι

- Δίαυλος είναι οι γραμμές (δεδομένων, διευθύνσεων και ελέγχου) που απαιτούνται για τη μεταφορά δεδομένων (και την υποστήριξη διακοπών και διαιτησίας) μεταξύ επεξεργαστή, κύριας μνήμης και μονάδων Ε/Ε.
- Πρωτόκολλο διαύλου είναι το σύνολο των κανόνων οι οποίοι καθορίζουν τη συμπεριφορά των διαφόρων μονάδων, οι οποίες συνδέονται στο δίαυλο.
- Τα σήματα ελέγχου καθορίζουν πότε μπορεί να εκτελεστεί μία εντολή ανάγνωσης ή εγγραφής (R/W') και τους χρόνους στους οποίους ο επεξεργαστής και οι μονάδες Ε/Ε μπορούν να τοποθετούν δεδομένα στο δίαυλο.
- Οι μονάδες ενός υπολογιστικού συστήματος διακρίνονται σε εκείνες που έχουν ρόλο εκκινητή ή κυρίαρχου (initiator ή master), οι οποίες μπορούν να προκαλέσουν μεταφορά δεδομένων εκδίδοντας εντολές ανάγνωσης ή εγγραφής στο δίαυλο και σε εκείνες με ρόλο προορισμού ή σκλάβου (slave).
- Στους σύγχρονους διαύλους, όλες οι μονάδες αντλούν πληροφορίες χρονισμού από κοινό σήμα ρολογιού.
- Κύκλος διαύλου (bus cycle) είναι το χρονικό διάστημα στο οποίο μπορεί να ολοκληρωθεί μια μεταφορά δεδομένων.

Σύγχρονοι δίαυλοι

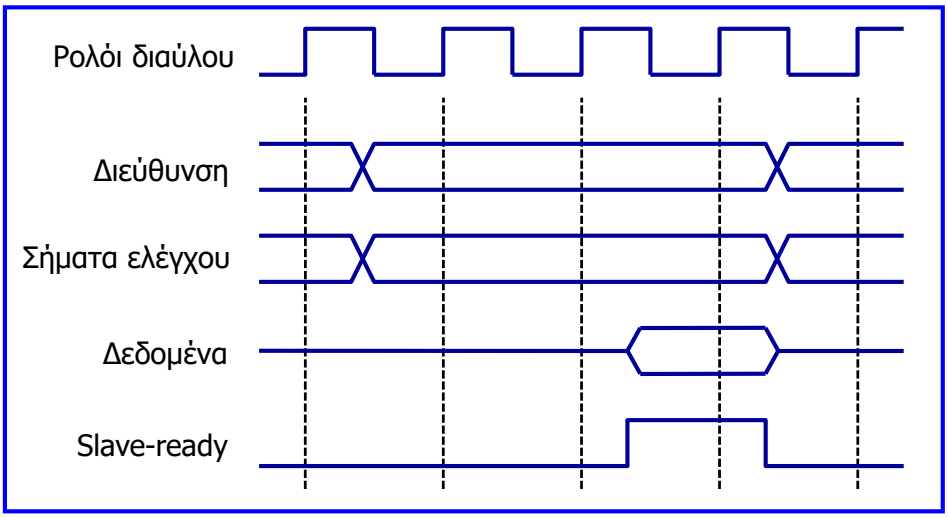


Χρονισμός μεταφοράς δεδομένων εισόδου (ανάγνωση) σε σύγχρονο δίαυλο.

Παρόμοια είναι η διαδικασία μεταφοράς δεδομένων εξόδου (εγγραφή).

- t_0: η κυρίαρχη μονάδα (π.χ. επεξεργαστής) εναποθέτει διεύθυνση και σήματα ελέγχου στο δίαυλο. Στο διάγραμμα συμπεριλαμβάνονται οι καθυστερήσεις διάδοσης σημάτων.
- Το εύρος του παλμού ($t_1 - t_0$) πρέπει να είναι μεγαλύτερο από το μέγιστο χρόνο μετάδοσης μεταξύ δύο μονάδων του διαύλου και να αρκεί για την αποκωδικοποίηση της διεύθυνσης και των σημάτων ελέγχου από κάθε μονάδα εισόδου με ρόλο σκλάβου, ώστε αυτή να ανταποκριθεί την χρονική στιγμή t_1, κατά την οποία εναποθέτει τα δεδομένα εισόδου στις γραμμές δεδομένων.
- Το διάστημα ($t_2 - t_1$) πρέπει να είναι μεγαλύτερο από το μέγιστο χρόνο μετάδοσης στο δίαυλο συν το χρόνο αρχικοποίησης του καταχωρητή εισόδου της κυρίαρχης μονάδας.

Σύγχρονοι δίαυλοι: μεταφορές πολλαπλών κύκλων



- Χρησιμοποιείται ρολόι υψηλότερης συχνότητας, ώστε η ταχύτητα του διαύλου να μην προσαρμόζεται σε αυτήν της αργότερης μονάδας.
- Επίσης, χρησιμοποιείται ένα σήμα ελέγχου (Slave-ready) που αποτελεί την επιβεβαίωση της μονάδας εισόδου για την αποστολή έγκυρων δεδομένων, ώστε να εξασφαλίζεται μεγαλύτερη αξιοπιστία στη μεταφορά δεδομένων.

Διασυνδέσεις Ε/Ε με σύγχρονο δίαυλο

- Οι διασυνδέσεις (ή διεπαφές) Ε/Ε αποτελούνται από τα κυκλώματα που απαιτούνται για τη σύνδεση μιας μονάδας Ε/Ε στο δίαυλο ενός υπολογιστικού συστήματος.

- Τα κυκλώματα αυτά υλοποιούν:

  ✓ την αποκωδικοποίηση της διεύθυνσης ή των διευθύνσεων που αντιστοιχούν στη μονάδα Ε/Ε, ώστε αυτή να επιλέγεται από τον επεξεργαστή για μεταφορά δεδομένων,

  ✓ το χειρισμό των σημάτων ελέγχου του διαύλου (π.χ. R/W', ρολόι),

  ✓ και τη μεταφορά δεδομένων από ή προς τη μονάδα ή τις μονάδες Ε/Ε.

Παράδειγμα

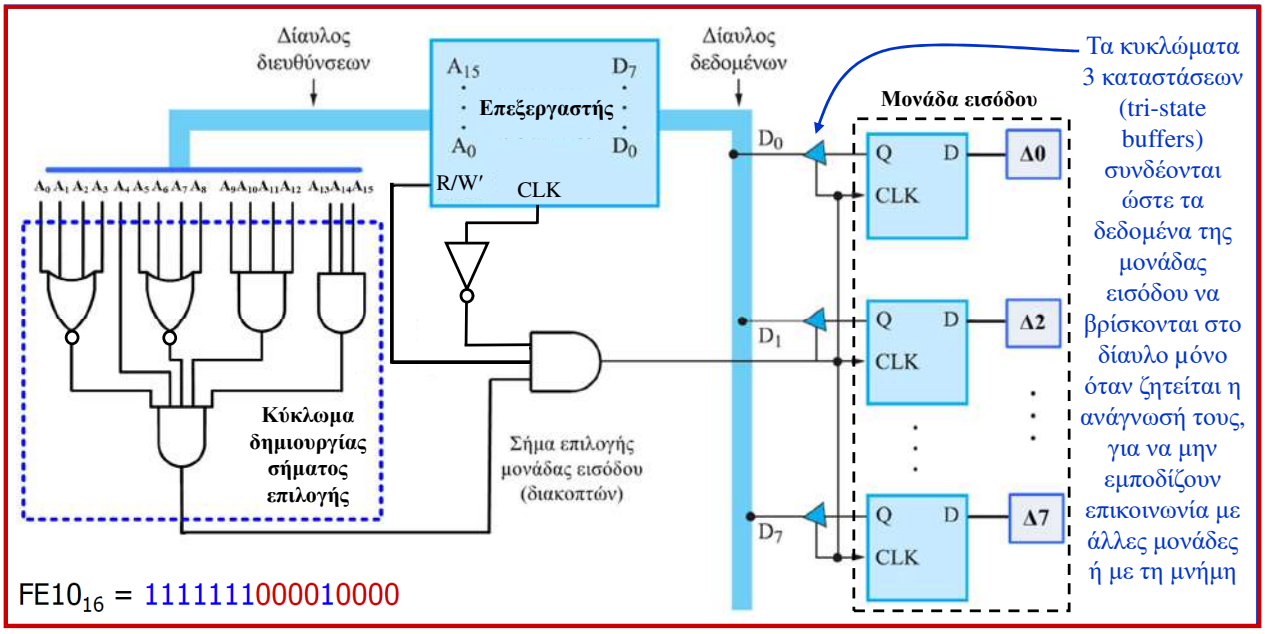
- Ένα υπολογιστικό σύστημα με πεδίο διευθύνσεων μνήμης (memory address space) 2^{16} λέξεων των 8 bit, χρησιμοποιεί τεχνική Ε/Ε απεικονιζόμενη στη μνήμη (memory-mapped I/O) για προσπέλαση μονάδων Ε/Ε. Ζητείται ο σχεδιασμός κατάλληλου κυκλώματος διασύνδεσης για την ανάγνωση της κατάστασης (0 ή 1) οκτώ διακοπτών, με τη μορφή ενός byte στη διεύθυνση FE10h. Το σύστημα διαθέτει σύγχρονο δίαυλο ενός κύκλου ρολογιού.

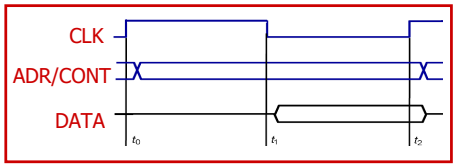
- Κάθε διεύθυνση αποτελείται από 16 bit και η διεύθυνση FE10h στο δυαδικό σύστημα έχει ως εξής: 1111111000010000.

- Θα πρέπει λοιπόν να σχεδιαστεί κατάλληλο συνδυαστικό κύκλωμα που να ενεργοποιεί την ανάγνωση της κατάστασης των οκτώ διακοπτών, όταν ζητηθεί πρόσβαση στην διεύθυνση αυτή.

- Όταν μέσω του κυκλώματος αποκωδικοποίησης διεύθυνσης ενεργοποιηθεί η ανάγνωση των δεδομένων της μονάδας εισόδου (διακόπτες), τότε η πληροφορία εισόδου (κατάσταση διακοπτών) θα πρέπει να οδηγηθεί στις γραμμές του διαύλου δεδομένων του επεξεργαστή.

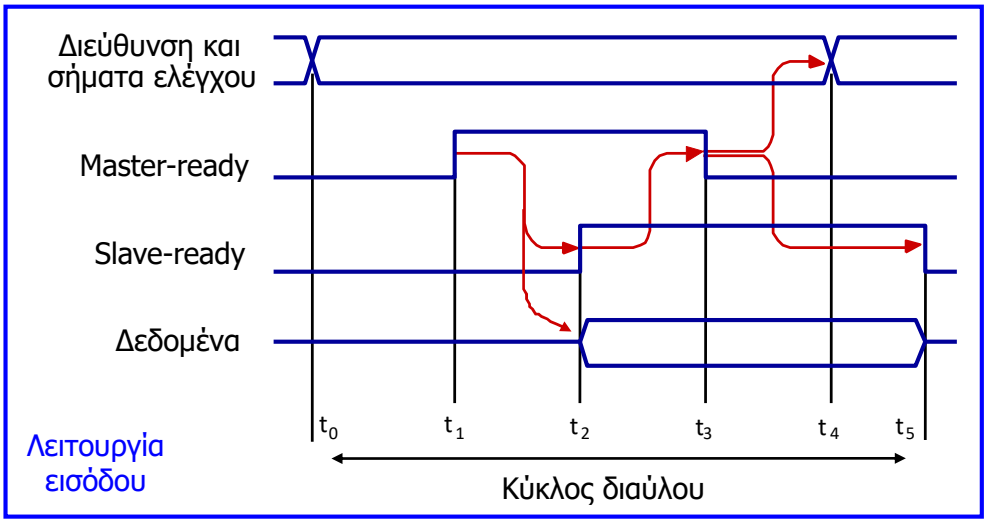
Παράδειγμα





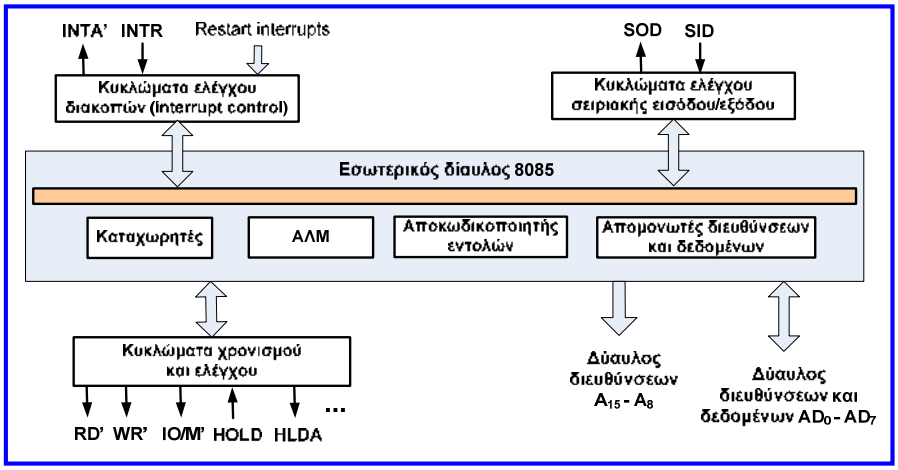
Ασύγχρονοι δίαυλοι

- Εναλλακτικός σχεδιασμός διαύλων που βασίζεται στη χρήση χειραψίας (handshake) μεταξύ κυρίαρχης μονάδας και μονάδας-σκλάβου.
- Το κοινό σήμα ρολογιού αντικαθίσταται από δύο γραμμές ελέγχου (master-ready και slave-ready), από τις οποίες η 1η υποδεικνύει ότι η κυρίαρχη μονάδα είναι έτοιμη για μία συναλλαγή δεδομένων και η 2η αποτελεί απόκριση από τη πλευρά της μονάδας-σκλάβου.



Λειτουργίες εισόδου/εξόδου στον Intel 8085

- Ο Intel 8085 διαθέτει παράλληλη και σειριακή επικοινωνία με μονάδες εισόδου/εξόδου.
- Η παράλληλη ανταλλαγή δεδομένων γίνεται μέσω του διαύλου δεδομένων, ενώ η σειριακή μέσω των γραμμών SID & SOD.



- Ο έλεγχος διακίνησης δεδομένων μεταξύ του μικροεπεξεργαστή και της μνήμης ή μονάδων εισόδου/εξόδου, επιτυγχάνεται μέσω τριών σημάτων εξόδου: IO/M', RD', WR'.
- Το σήμα IO/M' καθορίζει εάν η προσπέλαση αφορά τη μνήμη ή μονάδα εισόδου/εξόδου, δηλαδή IO/M' = 0 δηλώνει επικοινωνία του Intel 8085 με τη μνήμη, ενώ IO/M' = 1 δηλώνει επικοινωνία του επεξεργαστή με μονάδα εισόδου/εξόδου.
- Τα άλλα δύο σήματα ελέγχουν αντίστοιχα την εγγραφή και την ανάγνωση δεδομένων (RD' = 0 → ανάγνωση, WR' = 0 → εγγραφή).

Λειτουργίες εισόδου/εξόδου στον Intel 8085

- Στον επεξεργαστή 8085 η παράλληλη ανταλλαγή δεδομένων μέσω του διαύλου δεδομένων μπορεί να γίνει χρησιμοποιώντας την τεχνική Ε/Ε απεικονιζόμενη στη μνήμη (memory-mapped I/O), στην οποία ένα μέρος των διευθύνσεων μνήμης χρησιμοποιούνται για αναφορά σε καταχωρητές μονάδων Ε/Ε, ώστε να μην απαιτούνται ειδικές εντολές για την προσπέλασή τους (αρκούν οι εντολές που μεταφέρουν δεδομένα από και προς τη μνήμη).
- Ωστόσο, η ανταλλαγή δεδομένων μέσω του διαύλου δεδομένων, μπορεί να γίνει και με χρήση των ειδικών εντολών IN & OUT, η εκτέλεση των οποίων ενεργοποιεί τα κατάλληλα για την ανταλλαγή σήματα ελέγχου (τεχνική I/O-mapped IO ή peripheral-mapped I/O).
- Κατά τη χρήση της τεχνικής memory-mapped I/O το σήμα IO/M' τίθεται σε τιμή 0, ενώ κατά τη χρήση της τεχνικής peripheral-mapped I/O το σήμα IO/M' τίθεται σε τιμή 1.
- Με την εντολή IN τα δεδομένα που τοποθετούνται από μια θύρα εισόδου στο δίαυλο δεδομένων, μετακινούνται στον συσσωρευτή Α, ενώ με την εντολή OUT το περιεχόμενο του συσσωρευτή Α τοποθετείται στο δίαυλο δεδομένων για να μεταφερθεί σε θύρα εξόδου.
- Η σειριακή ανταλλαγή δεδομένων ενεργοποιείται με τις ειδικές εντολές SIM και RIM.
- Κατά την εκτέλεση της εντολής SIM, το MSB του συσσωρευτή Α μεταφέρεται στη γραμμή SOD, ενώ κατά την εκτέλεση της εντολής RIM η τιμή της γραμμής SID μεταφέρεται στην πιο σημαντική θέση του συσσωρευτή Α. Επισημαίνεται ότι οι εντολές SIM (set interrupt mask), RIM (read interrupt mask) χρησιμοποιούνται και για τη διαχείριση σημάτων διακοπής.

Λειτουργίες εισόδου/εξόδου στον Intel 8085

- Ο μηχανισμός διακοπής με τον οποίο ο επεξεργαστής Intel 8085 ανταλλάσσει δεδομένα μέσω του διαύλου δεδομένων με μονάδες εισόδου/εξόδου, ενώ εκτελεί κάποιο πρόγραμμα, ενεργοποιείται μέσω εξωτερικού σήματος διακοπής που δέχεται ο μικροεπεξεργαστής στην είσοδό του INTR από μία μονάδα εισόδου/εξόδου.

- Η τιμή της εισόδου INTR ελέγχεται από τον επεξεργαστή στον επόμενο κύκλο ρολογιού από την ολοκλήρωση μιας εντολής και εάν είναι 1, παρεμποδίζεται η αύξηση του μετρητή προγράμματος και ενεργοποιείται το σήμα εξόδου INTA' (INTA' = 0), το οποίο στέλνεται στη μονάδα εισόδου/εξόδου που ενεργοποίησε την είσοδο INTR.

- Μόλις η μονάδα εισόδου/εξόδου λάβει τον παλμό INTA', απενεργοποιεί την αίτηση διακοπής (INTR) και ο επεξεργαστής οδηγείται στην ρουτίνα εξυπηρέτησης διακοπής (ISR).

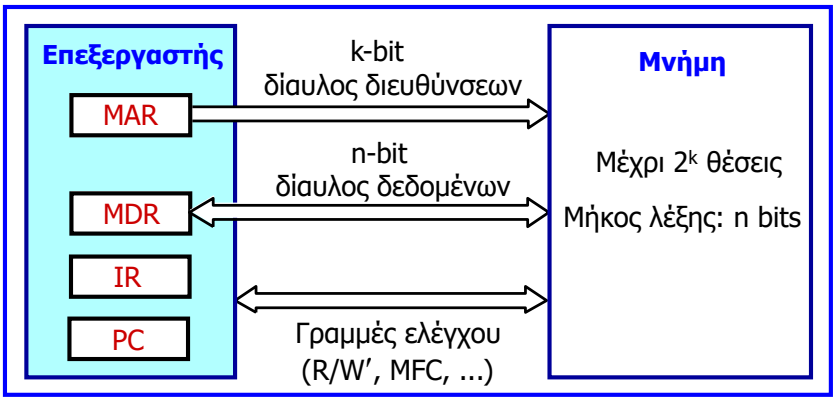
- Ο μηχανισμός DMA ενεργοποιείται όταν μία μονάδα εισόδου/εξόδου ενεργοποιεί την είσοδο HOLD του επεξεργαστή, οπότε αυτός οδηγεί την έξοδο HLDA σε τιμή 1, δίνοντας τον έλεγχο των διαύλων δεδομένων και διευθύνσεων στη μονάδα εισόδου/εξόδου, η οποία πλέον είναι σε θέση να διενεργήσει μεταφορά δεδομένων από και προς τη μνήμη.

Κεφάλαιο 5: Σύστημα μνήμης του υπολογιστή

Βασικές έννοιες

- Το μέγιστο μέγεθος μνήμης (δηλαδή, ο μέγιστος αριθμός θέσεων μνήμης) ενός υπολογιστή προσδιορίζεται από το πλήθος των ψηφίων (bits) των διευθύνσεων που μπορεί να παράγει (16 bits: μνήμη 2^{16} = 64K θέσεις, 32 bits: μνήμη 2^{32} = 4G θέσεις).
- Οι περισσότεροι υπολογιστές διαθέτουν μνήμη διευθυνσιοδοτούμενη κατά byte.
- Μία μονάδα μνήμης λέγεται μνήμη τυχαίας προσπέλασης (RAM) εάν οποιαδήποτε θέση της μπορεί να προσπελαστεί για μία λειτουργία ανάγνωσης ή εγγραφής, μέσα σε προκαθορισμένο χρονικό διάστημα, που είναι ανεξάρτητο από τη διεύθυνση της θέσης αυτής (σε αντίθεση με τις σειριακές μονάδες αποθήκευσης: μαγνητικές ταινίες, δίσκοι).
- Βασική τεχνολογία κατασκευής: ημιαγωγικά ολοκληρωμένα κυκλώματα.
- Ο επεξεργαστής είναι πιο γρήγορος από τη μνήμη, συνεπώς για να μειωθεί ο χρόνος προσπέλασης μνήμης χρησιμοποιείται η λανθάνουσα (cache) μνήμη που είναι ταχύτερη, τοποθετείται μεταξύ της κύριας μνήμης και του επεξεργαστή και κρατά τα «ενεργά» τμήματα ενός προγράμματος και τα αντίστοιχα δεδομένα.
- Η ιδεατή (εικονική, virtual) μνήμη χρησιμοποιείται για την αύξηση του φαινόμενου μεγέθους της φυσικής μνήμης. Τα δεδομένα διευθυνσιοδοτούνται σε μεγαλύτερο ιδεατό (εικονικό) χώρο διευθύνσεων (λογικές διευθύνσεις), μέρος του οποίου απεικονίζεται σε συσκευές δευτερεύουσας αποθήκευσης (π.χ. μαγνητικός σκληρός δίσκος).

Σύνδεση μνήμης και επεξεργαστή



MAR: καταχωρητής διευθύνσεων μνήμης

MDR: καταχωρητής δεδομένων μνήμης

IR: καταχωρητής εντολής

PC: μετρητής προγράμματος

R: read (ανάγνωση)

W': write (εγγραφή)

MFC: memory function completed (ολοκλήρωση λειτουργίας μνήμης)

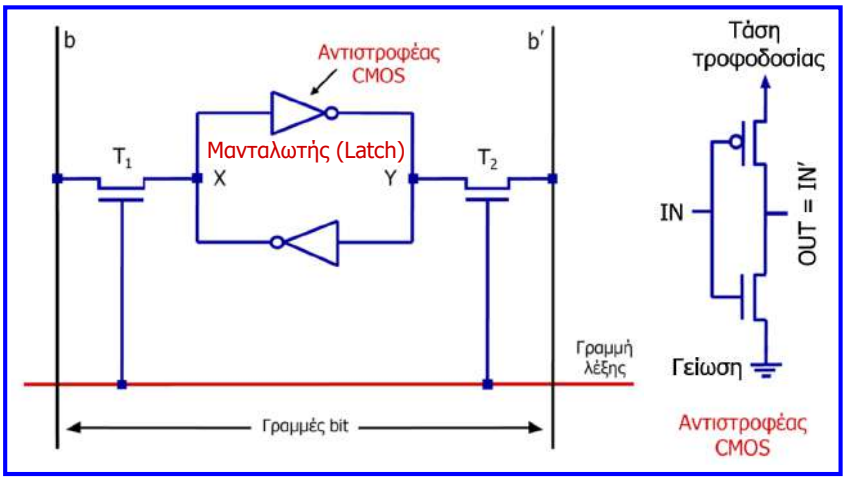
- Η μεταφορά εντολών και δεδομένων μεταξύ μνήμης και επεξεργαστή γίνεται με χρήση των MAR και MDR. Οι εντολές (με βάση τη διεύθυνση του PC) από τη μνήμη τοποθετούνται στον MDR και έπειτα στον IR.
- Χρόνος προσπέλασης μνήμης: μεταξύ αρχικοποίησης λειτουργίας (π.χ. ανάγνωσης) και ενεργοποίησης του MFC.
- Χρόνος κύκλου μνήμης: χρόνος μεταξύ αρχικοποίησης δύο λειτουργιών (λίγο μεγαλύτερος).
- Εκτός από λειτουργίες απλής λέξης εκτελούνται και λειτουργίες μεταφοράς τμήματος (block transfer operations) που αφορούν δεδομένα σε διαδοχικές διευθύνσεις μνήμης.

Μνήμες RAM

- Οι μνήμες οργανώνονται σε διάταξη πίνακα (array), κάθε κυψελίδα (cell) του οποίου αποθηκεύει ένα δυαδικό ψηφίο (bit).
- Οι κυψελίδες κάθε στήλης της διάταξης συνδέονται με κυκλώματα αίσθησης / εγγραφής.
- Κατά τη λειτουργία της ανάγνωσης, τα κυκλώματα αυτά ανιχνεύουν τα δεδομένα που έχουν επιλεγεί ανάλογα με τη διεύθυνση ανάγνωσης και τα μεταδίδουν στις γραμμές δεδομένων.
- Κατά τη λειτουργία της εγγραφής, τα κυκλώματα αυτά λαμβάνουν τα δεδομένα από τις γραμμές δεδομένων και τα αποθηκεύουν στις κυψελίδες που υποδεικνύονται από τη διεύθυνση εγγραφής.
- Οργάνωση μνήμης: A x B, A = αριθμός λέξεων, B = αριθμός ψηφίων (bits) ανά λέξη.
  - ✓ Μνήμη 128 bits: οργάνωση 16 x 8, απαιτούνται 4 γραμμές (bits) διεύθυνσης και 8 γραμμές δεδομένων.
  - ✓ Μνήμη 1K bits (1024 bits): 128 x 8, 7 γραμμές διεύθυνσης, 8 γραμμές δεδομένων.
  - ✓ Μνήμη 1K bits: 1024 x 1, 10 γραμμές διεύθυνσης, 1 γραμμή δεδομένων.
  - ✓ Μνήμη 4M bits: 512K x 8, 19 γραμμές διεύθυνσης, 8 γραμμές δεδομένων.
  - ✓ Μνήμη 256M bits: 64M x 4, 26 γραμμές διεύθυνσης, 4 γραμμές δεδομένων.
  - ✓ Απαιτούνται 2 γραμμές ελέγχου (R/W', CS / chip select ή CE / chip enable) και 2 γραμμές τροφοδοσίας και γείωσης.

Κυψελίδα στατικής μνήμης RAM (SRAM)

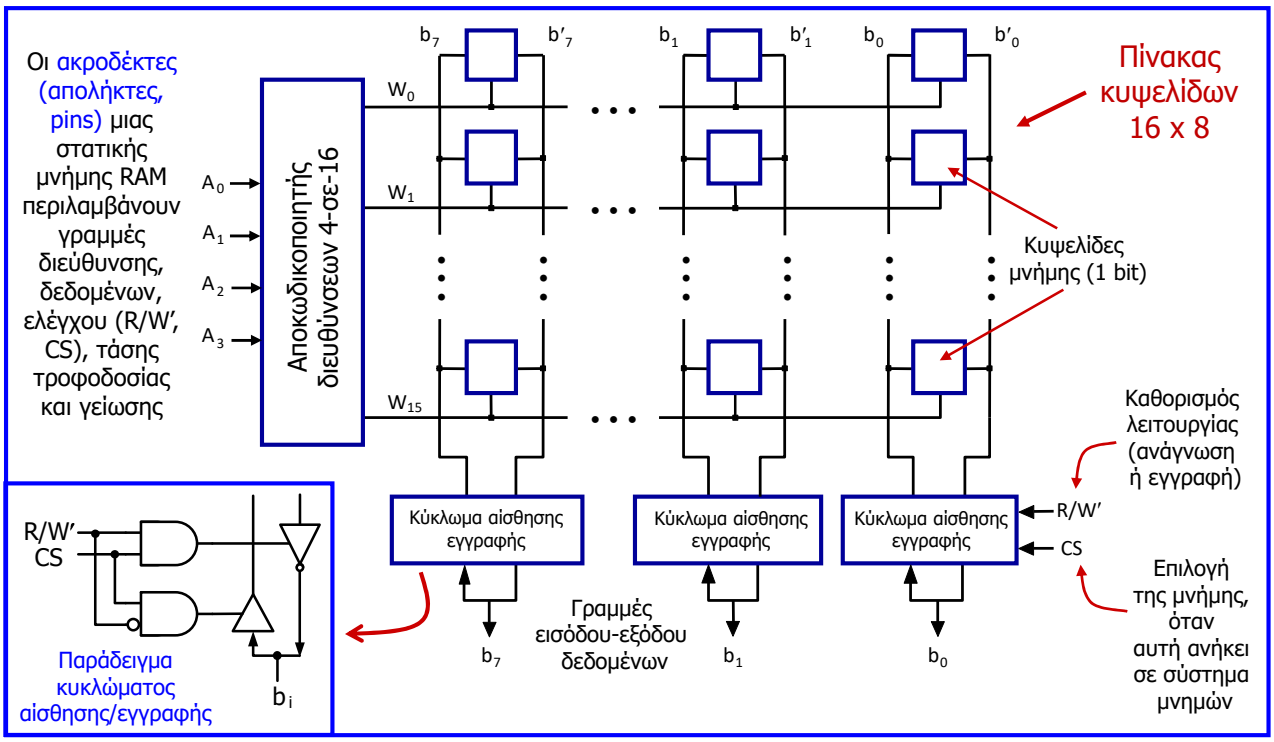
- Στατικές μνήμες: αποτελούνται από κυκλώματα (κυψελίδες) ικανά να διατηρούν την κατάστασή τους μόνο όσο ρευματοδοτούνται.
- Πολύ γρήγορες αφού μπορούν να προσπελαστούν σε μερικά ns.
- Ανάγνωση: Αν η κυψελίδα έχει αποθηκευμένη τη λογική κατάσταση 1 (X=1, Y=0) τότε με ενεργοποίηση της γραμμής λέξης οι γραμμές bit γίνονται b=1, b'=0 και λαμβάνονται στην έξοδο της μνήμης (αντίθετα για κατάσταση 0).



- Όσο η γραμμή λέξης είναι απενεργοποιημένη (0), τα τρανζίστορ είναι «ανοιχτοί διακόπτες» και η κυψελίδα διατηρεί τη λογική της κατάσταση.
- Εγγραφή: Για εγγραφή της λογικής κατάστασης 1 οι γραμμές bit τίθενται στις τιμές b=1, b'=0 (αντίθετα για κατάσταση 0) και στη συνέχεια ενεργοποιείται η γραμμή λέξης.

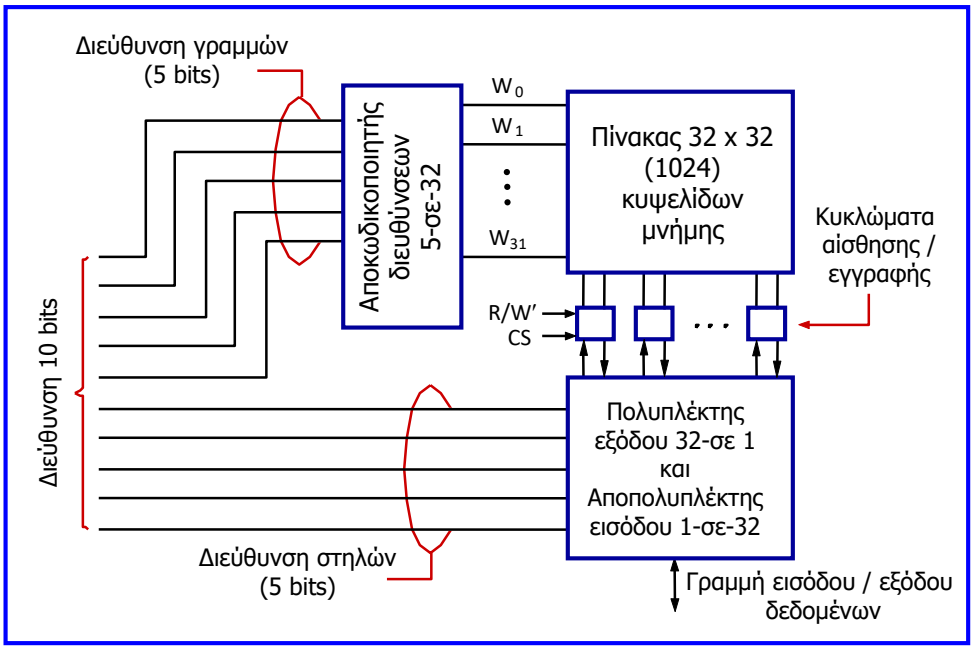
Εσωτερική οργάνωση στατικών μνημών RAM (SRAM)

Κύκλωμα μνήμης SRAM 128 bits με οργάνωση 16 x 8



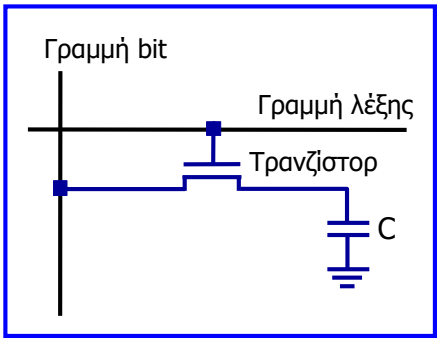
Εσωτερική οργάνωση στατικών μνημών RAM (SRAM)

Κύκλωμα μνήμης 1024 bits με οργάνωση 1K × 1, όπου οι κυψελίδες είναι τοποθετημένες σε πίνακα 32 × 32, αλλά μέσω ενός πολυπλέκτη 32-σε-1 ή ενός αποπολυπλέκτη 1-σε-32 εξάγεται ή εισάγεται στη μνήμη 1 ψηφίο (bit) κάθε φορά



Κυψελίδα δυναμικής μνήμης RAM (DRAM)

- Οι SRAM είναι γρήγορες, αλλά τα πολλά τρανζίστορ τους, δημιουργούν κόστος και υψηλή κατανάλωση ενέργειας.
- Αντιθέτως, οι κυψελίδες των μνημών DRAM μπορούν να περιλαμβάνουν μόνο ένα τρανζίστορ και έναν πυκνωτή.
- Δεν μπορούν να διατηρήσουν μόνιμα την κατάστασή τους, αφού απαιτείται ανανέωση της περιεχόμενης πληροφορίας (μειονέκτημα) και για το λόγο αυτό αναφέρονται ως δυναμικές μνήμες.
- Η πληροφορία αποθηκεύεται ως φορτίο στον πυκνωτή.

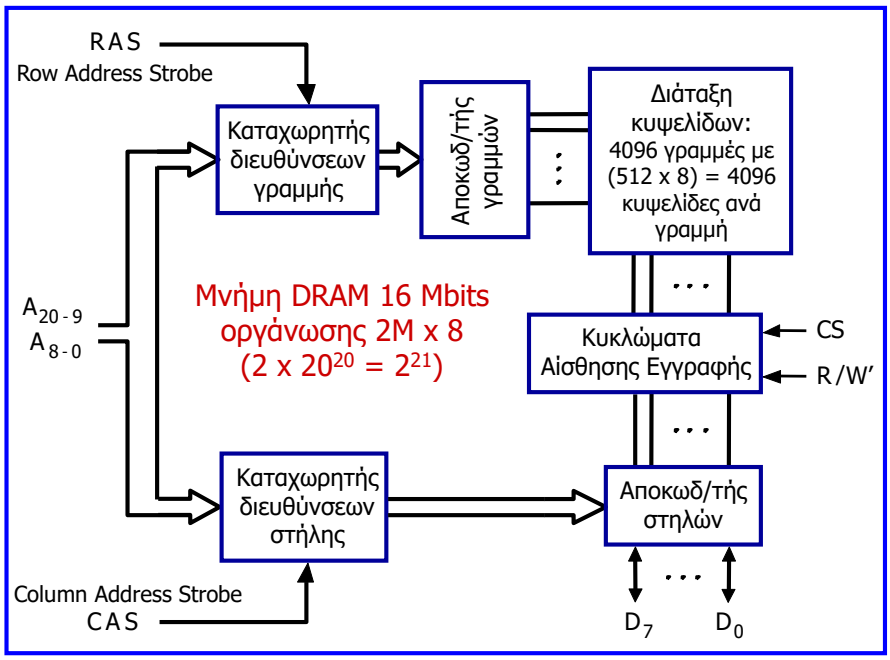


- Εγγραφή: τίθεται στη γραμμή bit κατάλληλη τιμή τάσης (λογική τιμή 1) και έπειτα γίνεται το ίδιο στη γραμμή λέξης με αποτέλεσμα τη φόρτιση του πυκνωτή με προκαθορισμένο φορτίο (λογική κατάσταση 1 στην κυψελίδα). Όταν το τρανζίστορ αποκοπεί (γραμμή λέξης = λογικό 0), ο πυκνωτής εκφορτίζεται (λόγω των απωλειών του και του ρεύματος διαρροής του τρανζίστορ), συνεπώς απαιτείται συχνή ανανέωση της αποθηκευμένης πληροφορίας (συνήθως σε διαστήματα < 64 ms).
- Ανάγνωση: ενεργοποιείται η γραμμή λέξης και αν μέσω κυκλώματος αίσθησης ανιχνευτεί ότι το φορτίο του πυκνωτή ξεπερνά ένα κατώφλι, η γραμμή bit οδηγείται από τον ενισχυτή σε πλήρη τάση (ανάγνωση 1). Η τάση της γραμμής bit επαναφορτίζει τον πυκνωτή (ανανέωση λογικής κατάστασης). Εάν ανιχνευτεί φορτίο μικρότερο του κατωφλιού, η γραμμή bit γειώνεται (ανάγνωση 0). Οι κυψελίδες μιας γραμμής διαβάζονται/ανανεώνονται ταυτόχρονα.

Δυναμικές μνήμες RAM (DRAM)

- Τα σήματα RAS, CAS καθορίζουν το χρονισμό της μνήμης και παράγονται από εξειδικευμένο κύκλωμα ελέγχου (ασύγχρονες μνήμες DRAM). Το σήμα RAS προηγείται.

- Η μνήμη διαθέτει 4096 γραμμές, άρα απαιτούνται 12 bits διευθύνσεων για την επιλογή μίας γραμμής.
- Οι 4096 κυψελίδες κάθε γραμμής διαιρούνται σε 512 ομάδες των 8, άρα κάθε γραμμή αποθηκεύει 512 bytes.
- Επομένως, απαιτούνται 9 bits για καθορισμό ενός byte στην επιλεγμένη γραμμή.



Δυναμικές μνήμες RAM (DRAM)

- Η τοποθέτηση καταχωρητών στις εξόδους των κυκλωμάτων αίσθησης κάθε στήλης επιτρέπει την προσπέλαση και άλλων bytes της ίδιας γραμμής, χωρίς να χρειαστεί επανεπιλογή της.
- Με διαδοχικούς παλμούς CAS και εφαρμογή διαδοχικών διευθύνσεων στηλών, επιτυγχάνεται γρήγορη μεταφορά τμήματος δεδομένων (λειτουργία γρήγορης σελίδας, fast page mode).
- Η λειτουργία των σύγχρονων δυναμικών μνημών (SDRAM) είναι παρόμοια, αλλά συγχρονίζεται από σήμα ρολογιού (αποτελούν εξέλιξη των ασύγχρονων DRAM).
- Στη λειτουργία ριπής (burst operation) που είναι αντίστοιχη με τη λειτουργία γρήγορης σελίδας, δεν είναι απαραίτητη η παροχή παλμού ελέγχου CAS, αφού με χρήση του σήματος ρολογιού και ενός μετρητή στηλών, τοποθετούνται νέα δεδομένα στις γραμμές δεδομένων σε κάθε παλμό ρολογιού.
- Καθυστέρηση μεταφοράς μνήμης (memory latency) είναι ο χρόνος που απαιτείται για τη μεταφορά ενός byte ή λέξης δεδομένων.
- Εύρος ζώνης μνήμης (memory bandwidth) είναι ο αριθμός bits ή bytes που μπορούν να μεταφερθούν ανά sec και εξαρτάται από την ταχύτητα προσπέλασης στα αποθηκευμένα δεδομένα και από τον αριθμό των bits που μπορούν να μεταφερθούν παράλληλα.
- Σύγχρονες DRAM διπλού ρυθμού: μεταφορά δεδομένων και στις δύο ακμές ρολογιού.

Σχεδιασμός μεγάλων μνημών με μικρότερες μνήμες

1. Υπολογισμός του πλήθους N των κυκλωμάτων μικρής μνήμης που απαιτούνται:
   N = (χωρητικότητα μεγάλης μνήμης) / (χωρητικότητα μικρής μνήμης).
2. Διάταξη των N μικρών μνημών σε πίνακα P x Q, όπου P = N / Q και
   Q = (πλήθος bits ανά λέξη μεγάλης μνήμης) / (πλήθος bits ανά λέξη μικρής μνήμης).

Παράδειγμα:

Σχεδιασμός κυκλώματος μνήμης με οργάνωση 2M x 32, με χρήση μικρότερων κυκλωμάτων μνήμης με οργάνωση 512K x 8.

N = (2M x 32) / (512K x 8) = 16, Διάταξη P x Q με Q = 32 / 8 = 4 και P = N / Q = 16 / 4 = 4.

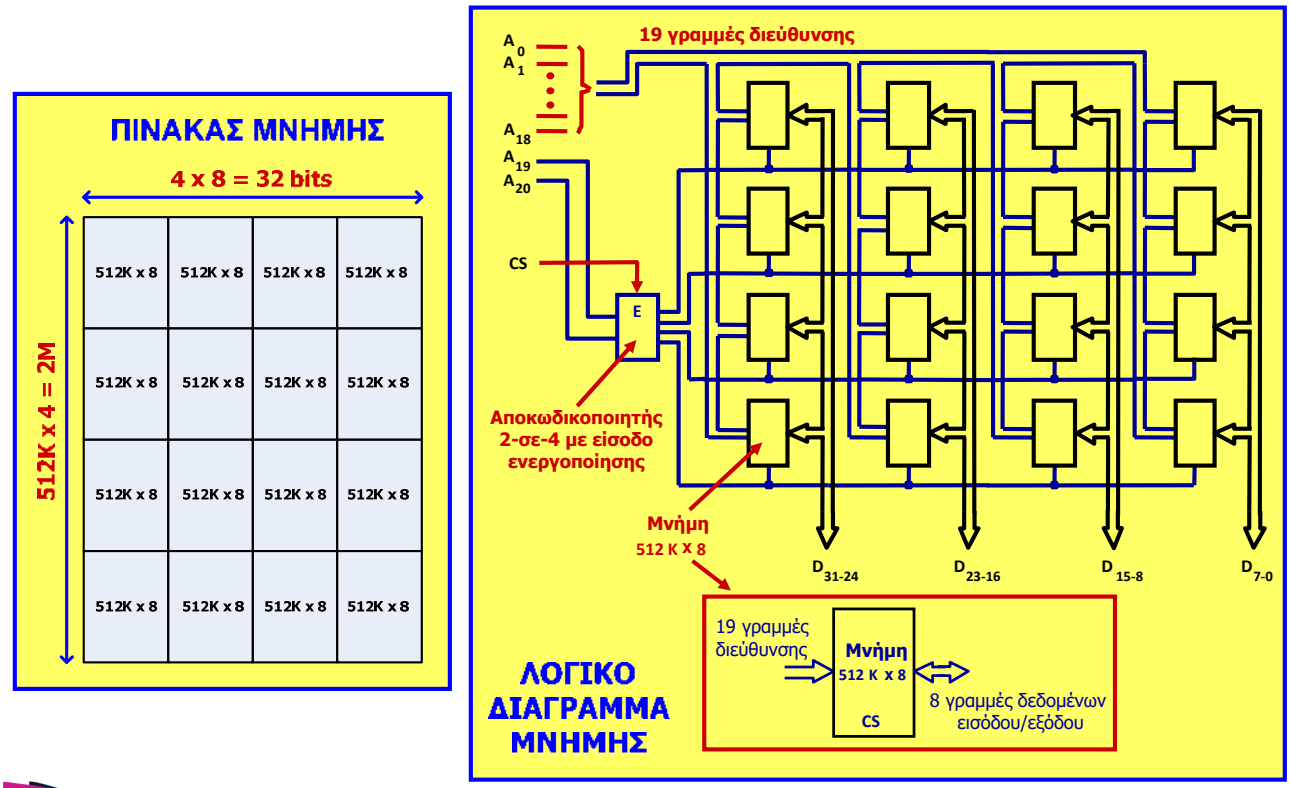
Η μεγάλη μνήμη διαθέτει 21 γραμμές διευθύνσεων, αφού 2M = 2^1 x 2^{20} = 2^{21}.

Η μικρή μνήμη διαθέτει 19 γραμμές διευθύνσεων, αφού 512K = 2^9 x 2^{10} = 2^{19}, συνεπώς τα 19 λιγότερο σημαντικά bits της διεύθυνσης της μεγάλης μνήμης που σχεδιάζουμε θα πρέπει να συνδεθούν στις 19 γραμμές διεύθυνσης κάθε μικρής μνήμης.

Με τις 2 γραμμές διεύθυνσης που απομένουν επιλέγεται μέσω ενός αποκωδικοποιητή 2-σε-4 μία από τις 4 γραμμές του πίνακα. Επομένως, κάθε έξοδος του αποκωδικοποιητή συνδέεται στις εισόδους επιλογής (CS, chip select) των μικρών μνημών μιας γραμμής.

Οι γραμμές R/W′ όλων των μικρών μνημών συνδέονται στην αντίστοιχη γραμμή της μεγάλης μνήμης (για λόγους απλότητας, δεν παρουσιάζονται στο διάγραμμα της μεγάλης μνήμης).

Σχεδιασμός μεγάλων μνημών με μικρότερες μνήμες



Συστήματα μνημών

- Ένα σύστημα μνήμης αποτελείται από επιμέρους ολοκληρωμένα κυκλώματα μνημών που διαθέτουν μία γραμμή ελέγχου εισόδου (επιλογή chip, chip select, CS).
- Μόνο το επιλεγμένο κύκλωμα μνήμης (για το οποίο CS = 1) τοποθετεί δεδομένα στη γραμμή δεδομένων, ενώ οι υπόλοιπες έξοδοι είναι αποκομμένες.
- Τα συστήματα δυναμικής μνήμης περιλαμβάνουν πολλαπλά ολοκληρωμένα κυκλώματα μνήμης SDRAM σε κάρτα, που τοποθετείται κάθετα στη μητρική κάρτα του Η/Υ και αναφέρονται ως μνήμες SIMM ή DIMM (single ή dual in-line memory modules) με 32 ή 64 γραμμές δεδομένων, αντίστοιχα.

Παράδειγμα: ένα τυπικό σύστημα μνήμης DIMM 2 Gbytes έχει χωρητικότητα $2 \times 2^{30} = 2^{31}$ bytes και οργάνωση 2^{28} x 64 bits = 256M x 64 bits. Η κάρτα του έχει 8 ολοκληρωμένα κυκλώματα SDRAM (synchronous DRAM) με οργάνωση 256M x 8 bits, καθένα από τα οποία έχει χωρητικότητα 2^{8} x 2^{20} = 2^{28} bytes.

- Οι SRAM (μεγαλύτερη ταχύτητα, αλλά υψηλότερο κόστος ανά bit) επιλέγονται ως λανθάνουσα (cache) μνήμη και οι SDRAM ως κύρια μνήμη των υπολογιστών.

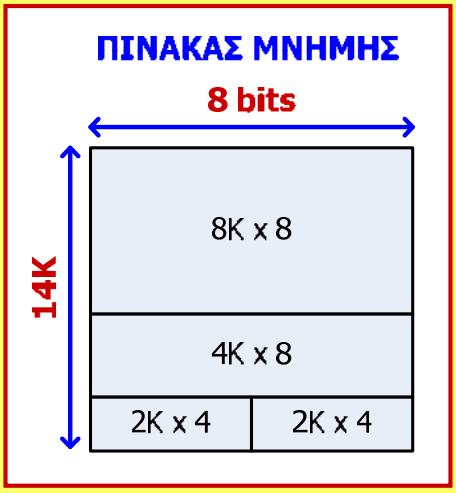
Παράδειγμα σχεδιασμού συστήματος μνήμης

- Ένα υπολογιστικό σύστημα διαθέτει δίαυλο διευθύνσεων 16 bits και δίαυλο δεδομένων 8 bits. Εάν είναι διαθέσιμα 4 ολοκληρωμένα κυκλώματα μνήμης RAM με οργάνωση 8K x 8, 4K x 8, 2K x 4 και 2K x 4, αντίστοιχα, καθώς και λογικές πύλες, αποκωδικοποιητές, να σχεδιαστεί το λογικό διάγραμμα του μνήμης του συστήματος, έτσι ώστε να καλύπτονται οι διευθύνσεις μνήμης από τη διεύθυνση 0000h έως τη διεύθυνση που ορίζεται από τη συνολική χωρητικότητα των διαθέσιμων ολοκληρωμένων κυκλωμάτων μνήμης.
- Αρχικά, καταστρώνουμε τον πίνακα και το χάρτη της μνήμης του συστήματος με βάση το μέγεθος των διαθέσιμων ολοκληρωμένων κυκλωμάτων μνήμης.

Κύκλωμα μνήμης	Μέγεθος (λέξεις)	Ψηφία διεύθυνσης	Πρώτη και τελευταία διεύθυνση (δυαδικό)	Πρώτη και τελευταία διεύθυνση (16δικό)
Μνήμη 1	8K	13 ($8 \cdot 2^{10} = 2^{13}$)	0000000000000000 – 0001111111111111	0000 – 1FFF
Μνήμη 2	4K	12 ($4 \cdot 2^{10} = 2^{12}$)	0010000000000000 – 0010111111111111	2000 – 2FFF
Μνήμη 3 Μνήμη 4	2K	11 ($2 \cdot 2^{10} = 2^{11}$)	0011000000000000 – 0011011111111111	3000 – 37FF

Για τον υπολογισμό της τελευταίας διεύθυνσης, προσθέτουμε στην πρώτη τον αριθμό $2^{n} - 1$, δηλαδή τον αριθμό με τα n λιγότερο σημαντικά ψηφία 1 και τα υπόλοιπα ψηφία 0 (n = πλήθος ψηφίων διεύθυνσης)

Παράδειγμα σχεδιασμού συστήματος μνήμης



- Το σήμα επιλογής κάθε ολοκληρωμένου κυκλώματος μνήμης, προκύπτει από τα δυαδικά ψηφία των διευθύνσεων που έχουν την ίδια λογική τιμή στην πρώτη και στην τελευταία διεύθυνση.
- Οι λογικές συναρτήσεις των σημάτων επιλογής προκύπτουν άμεσα από τη λογική τιμή των προαναφερόμενων δυαδικών ψηφίων:

$$CS_1 = A'_{15} \cdot A'_{14} \cdot A'_{13}$$

$$CS_2 = A'_{15} \cdot A'_{14} \cdot A_{13} \cdot A'_{12}$$

$$CS_{3,4} = A'_{15} \cdot A'_{14} \cdot A_{13} \cdot A_{12} \cdot A'_{11}$$

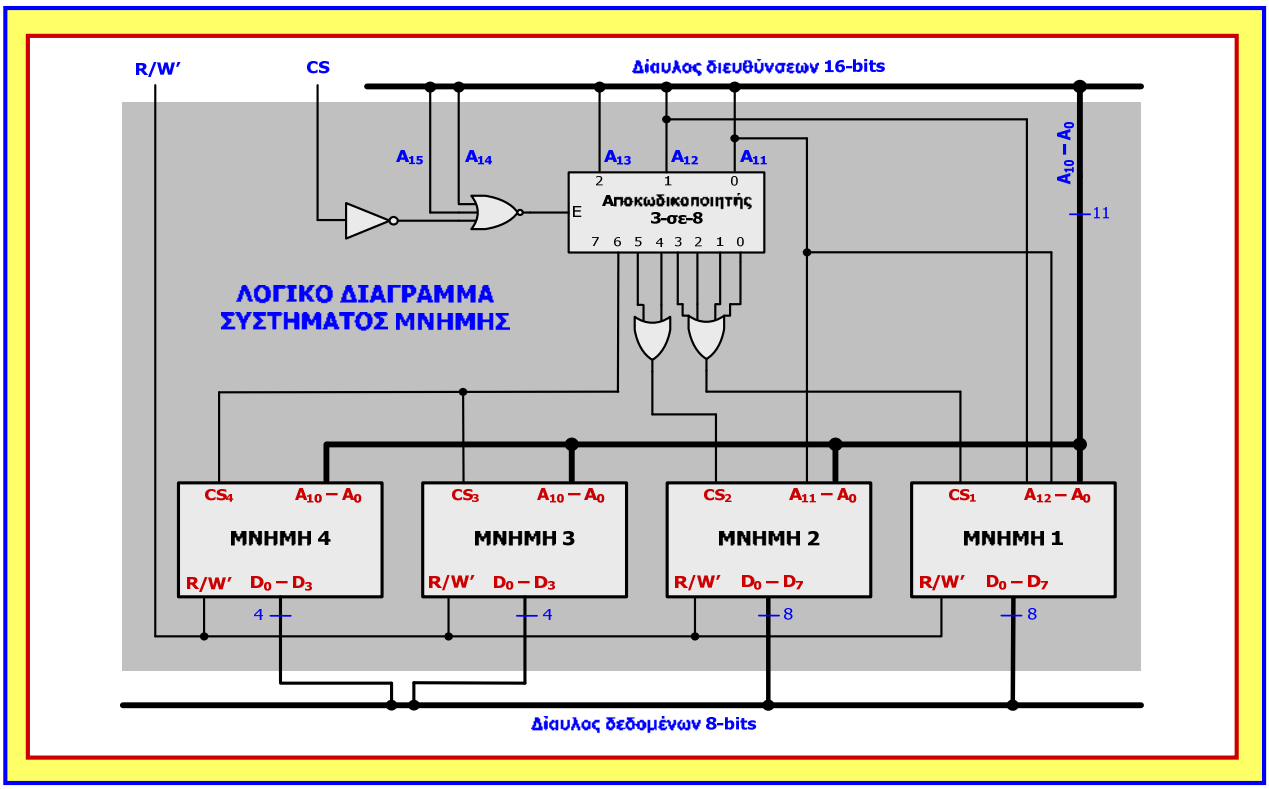
όπου A_{15}, A_{14}, A_{13}, A_{12} και A_{11} είναι τα πέντε πιο σημαντικά ψηφία της διεύθυνσης.

- Οι λογικές συναρτήσεις των σημάτων επιλογής των ολοκληρωμένων κυκλωμάτων μνήμης, μπορούν να υλοποιηθούν εύκολα με 5 αντιστροφείς και ένα δικτύωμα από πύλες AND με 2 και 3 εισόδους.
- Εναλλακτικά, οι ίδιες συναρτήσεις μπορούν να υλοποιηθούν συνδυάζοντας έναν αποκωδικοποιητή 3-σε-8 με είσοδο ενεργοποίησης (E) και λογικές πύλες.

Παράδειγμα σχεδιασμού συστήματος μνήμης

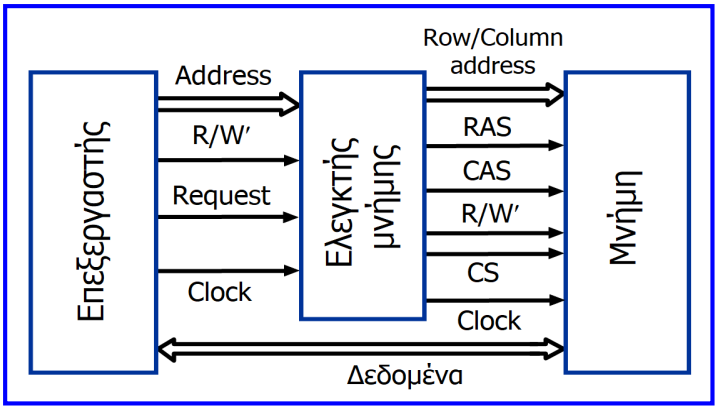
- Τα σήματα επιλογής όλων των μνημών ενεργοποιούνται όταν $A_{15} = A_{14} = 0$, συνεπώς μπορεί η έξοδος μιας πύλης NOR με εισόδους τα ψηφία διεύθυνσης A_{15} και A_{14} να χρησιμοποιηθεί ως είσοδος ενεργοποίησης του αποκωδικοποιητή, με τη βοήθεια του οποίου θα υλοποιήσουμε τα σήματα επιλογής των ολοκληρωμένων κυκλωμάτων μνήμης.
- Ως είσοδος της ίδιας πύλης NOR, χρησιμοποιείται και η συμπληρωματική μορφή του σήματος CS της συνολικής μνήμης, έτσι ώστε όταν CS = 1 να ενεργοποιείται ο αποκωδικοποιητής, ενώ όταν CS = 0 να απενεργοποιείται.
- Το σήμα CS_1 ενεργοποιείται όταν $A_{13}\,A_{12}\,A_{11}$ = 000 ή 001 ή 010 ή 011, επομένως μπορεί να παραχθεί από μια πύλη OR με εισόδους τις εξόδους 0, 1, 2, 3 του αποκωδικοποιητή.
- Το σήμα CS_2 ενεργοποιείται όταν $A_{13}\,A_{12}\,A_{11}$ = 100 ή 101, επομένως μπορεί να παραχθεί από μια πύλη OR με εισόδους τις εξόδους 4 και 5 του αποκωδικοποιητή.
- Το σήματα $CS_{3,4}$ ενεργοποιούνται όταν $A_{13}\,A_{12}\,A_{11}$ = 110, επομένως μπορούν να τροφοδοτηθούν από την έξοδο 6 του αποκωδικοποιητή.
- Στις γραμμές διεύθυνσης του πρώτου κυκλώματος μνήμης συνδέονται τα ψηφία διεύθυνσης A_0 έως A_{12}, στις γραμμές διεύθυνσης του δεύτερου κυκλώματος μνήμης συνδέονται τα ψηφία A_0 έως A_{11}, ενώ στις γραμμές διεύθυνσης του τρίτου και του τέταρτου κυκλώματος μνήμης συνδέονται τα ψηφία A_0 έως A_{10}.
- Το σήμα ανάγνωσης/εγγραφής R/W' της συνολικής μνήμης συνδέεται στην αντίστοιχη είσοδο κάθε κυκλώματος μνήμης.

Παράδειγμα σχεδιασμού συστήματος μνήμης



Ελεγκτής μνήμης

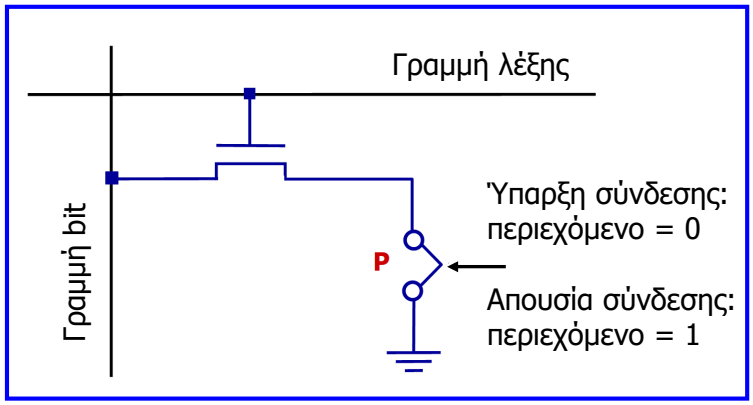
- Ένας τυπικός επεξεργαστής εκδίδει τα ψηφία μιας διεύθυνσης όλα μαζί ταυτόχρονα.
- Η απαιτούμενη πολύπλεξη των ψηφίων διεύθυνσης (ώστε με τα ψηφία ανώτερης τάξης να επιλέγεται μία γραμμή και με τα ψηφία κατώτερης τάξης μία στήλη), καθώς και η παραγωγή των κατάλληλων σημάτων ελέγχου της μνήμης, παρέχονται από τον ελεγκτή μνήμης (memory controller).
- Επίσης, ο ελεγκτής υποστηρίζει και τη διαδικασία ανανέωσης στις μνήμες DRAM που δεν έχουν ενσωματωμένη τη δυνατότητα αυτή.



Μνήμες μόνο ανάγνωσης (ROM)

- Οι μνήμες RAM χάνουν την αποθηκευμένη πληροφορία όταν διακοπεί η τροφοδοσία τους με ρεύμα και για το λόγο αυτό αναφέρονται ως προσωρινές (volatile) μνήμες.
- Υπάρχουν εφαρμογές που απαιτούν τη χρήση μη προσωρινών (non-volatile) μνημών (π.χ. προγράμματα συστήματος, firmware μονάδων υλικού), οι οποίες απαιτούν ειδική διαδικασία εγγραφής (κατά την κατασκευή τους) και αναφέρονται ως μνήμες μόνο ανάγνωσης (read-only memories, ROM).

Κυψελίδα μνήμης ROM

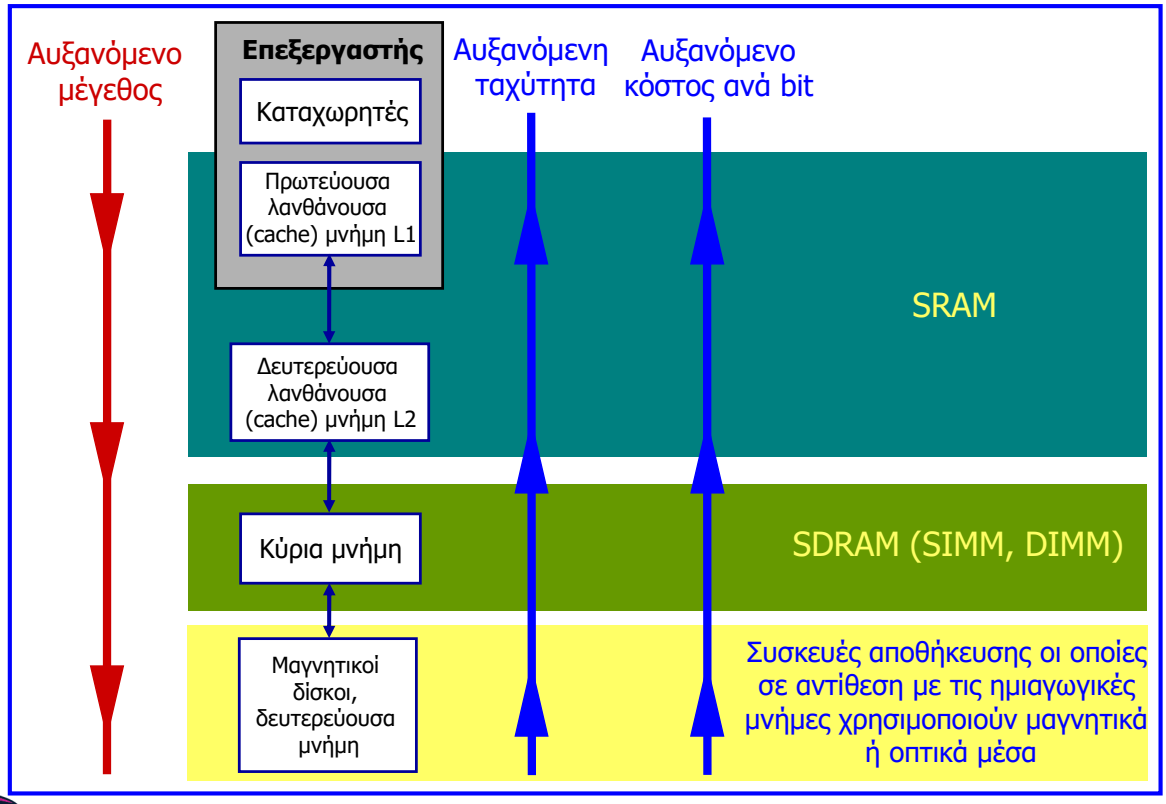


Για την ανάγνωση μιας κυψελίδας, η γραμμή bit τίθεται σε λογική τιμή 1 (τάση τροφοδοσίας) και ακολούθως ενεργοποιείται (επίσης με λογική τιμή 1) η γραμμή λέξης. Το τρανζίστορ είναι σε κατάσταση αγωγής (ON) και η τάση στη γραμμή bit μηδενίζεται (ανάγνωση τιμής 0) όταν υπάρχει σύνδεση με τη γείωση ή παραμένει σε λογική τιμή 1 (ανάγνωση τιμής 1) όταν δεν υπάρχει σύνδεση με τη γείωση.

Μνήμες PROM, EPROM, EEPROM και Flash

- Προγραμματιζόμενη ROM (programmable ROM, PROM): δυνατότητα προγραμματισμού από το χρήστη μέσω ασφάλειας στο σημείο P (μη αντιστρεπτή διαδικασία).
- Απαλείψιμη, επαναπρογραμματιζόμενη ROM (erasable programmable ROM, EPROM): χρησιμοποίηση τρανζίστορ στο σημείο P, το οποίο προγραμματίζεται με εισαγωγή φορτίου και απαλείφεται η τιμή του με έκθεση του κυκλώματος μνήμης σε υπεριώδες φως.
- Ηλεκτρικά απαλείψιμη επαναπρογραμματιζόμενη ROM (electrically erasable programmable ROM, EEPROM): διαγράφονται, διαβάζονται και προγραμματίζονται (εγγράφονται) με εφαρμογή διαφορετικών σταθμών τάσης στις κυψελίδες τους.
- Μνήμη Flash: ταχύτερη από την EEPROM, αφού επιτρέπει εγγραφή τμημάτων (blocks) κυψελίδων, διαθέτει μεγαλύτερη πυκνότητα που οδηγεί σε χαμηλότερο κόστος και καταναλώνει λιγότερη ενέργεια, γεγονός που την καθιστά ελκυστική για φορητές συσκευές.

Ιεραρχία συστήματος μνήμης



Λανθάνουσα μνήμη: βασικές έννοιες

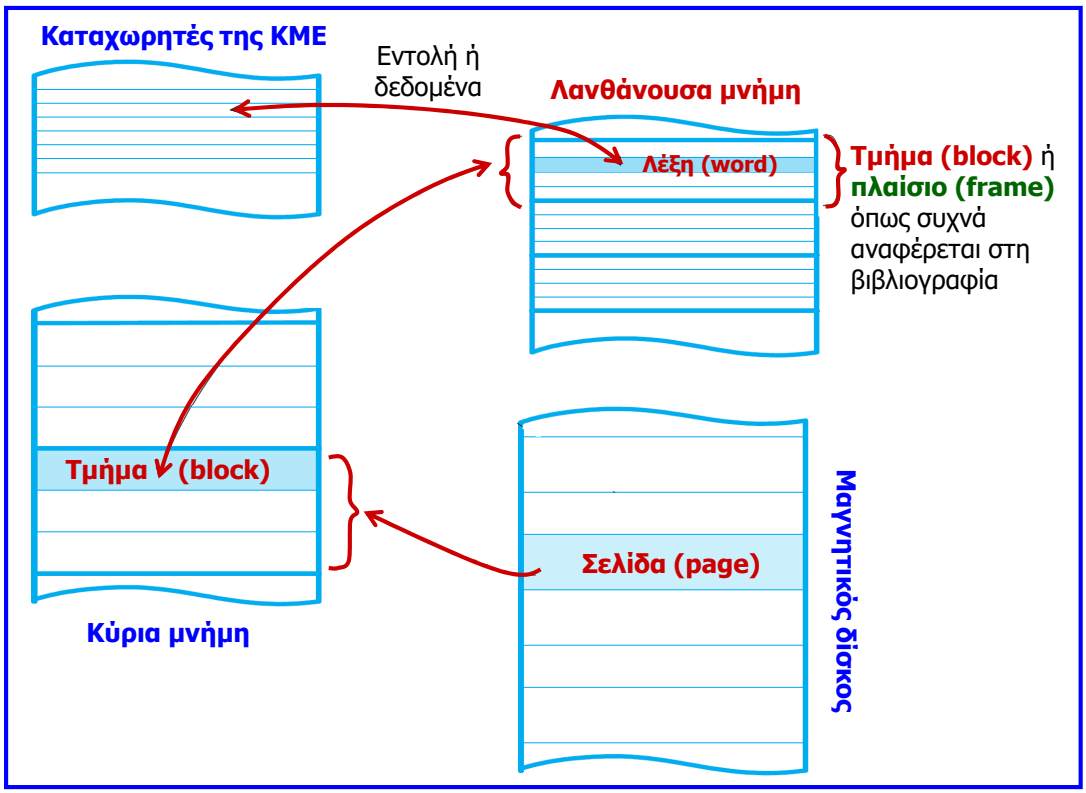
- Η ταχύτητα της κύριας μνήμης είναι χαμηλή σε σχέση με την ταχύτητα των σύγχρονων επεξεργαστών, συνεπώς για καλύτερη απόδοση ο επεξεργαστής δεν πρέπει να αναλώνει πολύ χρόνο αναμένοντας την προσπέλαση δεδομένων και εντολών από την κύρια μνήμη. Η κατάσταση γίνεται ακόμη χειρότερη όταν πρέπει να μεταφερθεί πληροφορία στην κύρια μνήμη από τη δευτερεύουσα.
- Το μέσο που χρησιμοποιείται για τη βελτίωση της απόδοσης του επεξεργαστή, είναι η λανθάνουσα (cache) μνήμη (στη βιβλιογραφία αναφέρεται και ως κρυφή μνήμη).
- Η λανθάνουσα μνήμη είναι σχετικά μικρής χωρητικότητας μνήμη που χρησιμοποιείται για την αποθήκευση πληροφορίας που αναμένεται να χρησιμοποιηθεί άμεσα ή με μεγάλη συχνότητα στο μέλλον, με αποτέλεσμα η κύρια μνήμη «παρουσιάζεται» στον επεξεργαστή ως πιο γρήγορη από όσο είναι στην πραγματικότητα.
- Η αύξηση της απόδοσης με χρήση της λανθάνουσας μνήμης βασίζεται στη χωρική και χρονική τοπικότητα αναφορών (locality of references) που χαρακτηρίζει την πλειονότητα των προγραμμάτων, αφού το μεγαλύτερο ποσοστό χρόνου εκτέλεσης προγραμμάτων αφορά ρουτίνες (που εκτελούνται συχνά) ή βρόχους όπου πολλές εντολές εκτελούνται επαναληπτικά, δηλαδή συνήθως οι εντολές ή τα δεδομένα που χρησιμοποιήθηκαν πρόσφατα ή βρίσκονται κοντά στην πληροφορία που χρησιμοποιείται, είναι πολύ πιθανό να χρησιμοποιηθούν στο άμεσο μέλλον.
- Εάν λοιπόν τα «ενεργά» τμήματα ενός προγράμματος τοποθετηθούν σε μία γρήγορη λανθάνουσα μνήμη, τότε ο συνολικός χρόνος εκτέλεσης μειώνεται σημαντικά.

Λανθάνουσα μνήμη: βασικές έννοιες

- Για τη συνέχεια, θεωρούμε ότι στο ιεραρχικό σύστημα μνήμης υπάρχει μόνο ένα επίπεδο λανθάνουσας μνήμης μεταξύ του επεξεργαστή (ΚΜΕ) και της κύριας μνήμης:

- Όταν γίνονται αιτήματα ανάγνωσης, το κύκλωμα ελέγχου της λανθάνουσας μνήμης φροντίζει να προσαχθούν από την κύρια μνήμη στη λανθάνουσα μνήμη, τμήματα (blocks) εντολών ή δεδομένων, τα οποία εμπεριέχουν τις καθοριζόμενες στα αιτήματα διευθύνσεις.
- Κατόπιν, όταν το πρόγραμμα αναφέρεται σε κάποια από τις διευθύνσεις των τμημάτων αυτών, τα επιθυμητά περιεχόμενα διαβάζονται απευθείας από τη λανθάνουσα μνήμη.
- Η αντιστοίχιση μεταξύ των τμημάτων της κύριας μνήμης και αυτών που βρίσκονται στη λανθάνουσα μνήμη, καθορίζεται από έναν τρόπο απεικόνισης (mapping function).
- Το πλήθος τμημάτων κύριας μνήμης υπερβαίνει σημαντικά το πλήθος τμημάτων λανθάνουσας μνήμης, αφού το μέγεθος της λανθάνουσας μνήμης είναι αρκετά μικρότερο.
- Όταν η λανθάνουσα μνήμη είναι πλήρης και γίνεται αναφορά σε μια λέξη, η οποία δε βρίσκεται στη λανθάνουσα μνήμη, τότε το κύκλωμα ελέγχου αποφασίζει ποιο τμήμα θα αντικατασταθεί με το νέο που περιέχει την αναφερόμενη λέξη.

Λανθάνουσα μνήμη: βασικές έννοιες



Λανθάνουσα μνήμη: ανάγνωση και εγγραφή

- Όταν πρόκειται για λειτουργία ανάγνωσης και το τμήμα που περιλαμβάνει την επιθυμητή από την ΚΜΕ λέξη, βρίσκεται στη λανθάνουσα μνήμη, η επιθυμητή λέξη προωθείται στην ΚΜΕ και τότε έχουμε επιτυχία (hit) ανάγνωσης.

- Όταν πρόκειται για λειτουργία εγγραφής και το τμήμα στο οποίο αντιστοιχεί η λέξη, που πρόκειται να εγγραφεί από την ΚΜΕ στη μνήμη, βρίσκεται στη λανθάνουσα μνήμη, έχουμε επιτυχία εγγραφής και το σύστημα συνεχίζει με έναν από τους δυο ακόλουθους τρόπους:

  ✓ Πρωτόκολλο write-through: η θέση στη λανθάνουσα μνήμη και η αντίστοιχη θέση στην κύρια μνήμη ενημερώνονται ταυτόχρονα, δηλαδή η λέξη γράφεται ταυτόχρονα στο τμήμα της λανθάνουσας μνήμης και στο αντίστοιχο τμήμα της κύριας μνήμης.

  ✓ Πρωτόκολλο write-back: ενημερώνεται η θέση (δηλαδή γράφεται η λέξη) στη λανθάνουσα μνήμη και σηματοδοτείται ως ενημερωμένη μέσω της ενεργοποίησης ενός ψηφίου τροποποίησης (dirty ή modified bit). Η θέση της λέξης αυτής στην κύρια μνήμη ενημερώνεται μόνο όταν το τμήμα της λανθάνουσας μνήμης που περιλαμβάνει τη λέξη αυτή πρόκειται να απομακρυνθεί από τη λανθάνουσα μνήμη, έτσι ώστε να δημιουργηθεί χώρος για νέο τμήμα κύριας μνήμης. Στην περίπτωση αυτή γίνεται ενημέρωση (εγγραφή) στην κύρια μνήμη όλου του τμήματος της λανθάνουσας μνήμης που περιλαμβάνει τη λέξη.

Λανθάνουσα μνήμη: ανάγνωση και εγγραφή

- Όταν πρόκειται για λειτουργία ανάγνωσης και το τμήμα που περιλαμβάνει την επιθυμητή λέξη δεν βρίσκεται στη λανθάνουσα μνήμη, έχουμε αστοχία (miss) ανάγνωσης.

- Τότε, το τμήμα στο οποίο αντιστοιχεί η επιθυμητή λέξη, αντιγράφεται από την κύρια μνήμη στη λανθάνουσα μνήμη.

- Όταν το τμήμα φορτωθεί στη λανθάνουσα μνήμη, η επιθυμητή λέξη προωθείται στην ΚΜΕ.

- Εναλλακτικά, η επιθυμητή λέξη μπορεί να προωθηθεί στην ΚΜΕ ταυτόχρονα με την ανάγνωσή της από την κύρια μνήμη (προσέγγιση load-through).

- Όταν πρόκειται για λειτουργία εγγραφής και το τμήμα στο οποίο αντιστοιχεί η λέξη που πρόκειται να εγγραφεί από την ΚΜΕ στη μνήμη δεν βρίσκεται στη λανθάνουσα μνήμη, έχουμε αστοχία εγγραφής και το σύστημα συνεχίζει με έναν από τους δύο ακόλουθους τρόπους:

  ✓ Πρωτόκολλο write-allocate: το τμήμα στο οποίο αντιστοιχεί η επιθυμητή λέξη αντιγράφεται από την κύρια μνήμη στη λανθάνουσα μνήμη και η ενημέρωση της κύριας μνήμης γίνεται με έναν από τους δύο τρόπους που προαναφέρθηκαν για την περίπτωση της επιτυχίας εγγραφής (πρωτόκολλα write-through και write-back).

  ✓ Πρωτόκολλο no-write-allocate: η επιθυμητή λέξη εγγράφεται κατευθείαν μόνο στο τμήμα της κύριας μνήμης που αντιστοιχεί.

Μέσος χρόνος προσπέλασης συστήματος μνήμης

- Μέσος χρόνος λειτουργίας ανάγνωσης, όπως τον αντιλαμβάνεται η ΚΜΕ:

$$t_{ave} = h \times t_h + (1 - h) \times t_m$$

✓ h: ποσοστό ή ρυθμός ευστοχίας (hit rate), δηλαδή ο λόγος του αριθμού επιτυχιών προς το σύνολο των προσπελάσεων ανάγνωσης.

✓ t_h: χρόνος προσπέλασης / μεταφοράς πληροφορίας από τη λανθάνουσα μνήμη στην ΚΜΕ.

✓ t_m: χρόνος προσπέλασης / μεταφοράς της πληροφορίας από την κύρια μνήμη στην ΚΜΕ (περιλαμβάνει το χρόνο προσπέλασης / μεταφοράς ενός τμήματος λέξεων από την κύρια μνήμη στη λανθάνουσα, ο οποίος αναφέρεται ως ποινή αστοχίας, miss penalty, M και το χρόνο προώθησης της επιθυμητής λέξης από τη λανθάνουσα μνήμη στην ΚΜΕ).

- Σε περίπτωση επιτυχίας ο χρόνος ανάγνωσης είναι t_h (time for a hit), ενώ σε περίπτωση αστοχίας ο χρόνος ανάγνωσης είναι t_m (time for a miss).
- Σε αρκετά συστήματα, η αναφορά στην κύρια μνήμη γίνεται παράλληλα με την προσπέλαση της λανθάνουσας μνήμης, έτσι ώστε αν ο έλεγχος της λανθάνουσας μνήμης είναι ανεπιτυχής (αστοχία), να έχει ήδη αρχίσει ο κύκλος προσπέλασης της κύριας μνήμης.
- Για τον υπολογισμό του μέσου χρόνου προσπέλασης του συστήματος μνήμης σε μία λειτουργία εγγραφής, θα πρέπει να λαμβάνεται υπόψη ο τρόπος ενημέρωσης της κύριας μνήμης που ακολουθείται από το σύστημα μνήμης.

Παράδειγμα μέσου χρόνου προσπέλασης μνήμης

Λανθάνουσα μνήμη με τμήματα 8 λέξεων, t_h = 1 κύκλος ρολογιού και h = 80%. Απαιτούνται: 1 κύκλος για μεταφορά της εναρκτήριας διεύθυνσης από την ΚΜΕ στην κύρια μνήμη, 8 κύκλοι για προσπέλαση της 1ης λέξης, 4 κύκλοι ανά λέξη για την προσπέλαση των επόμενων λέξεων του τμήματος. Η μεταφορά μιας λέξης από την κύρια στη λανθάνουσα μνήμη απαιτεί 1 κύκλο, όσο και η μεταφορά από τη λανθάνουσα μνήμη στον επεξεργαστή. Η αναφορά στην κύρια μνήμη γίνεται παράλληλα με την προσπέλαση της λανθάνουσας.

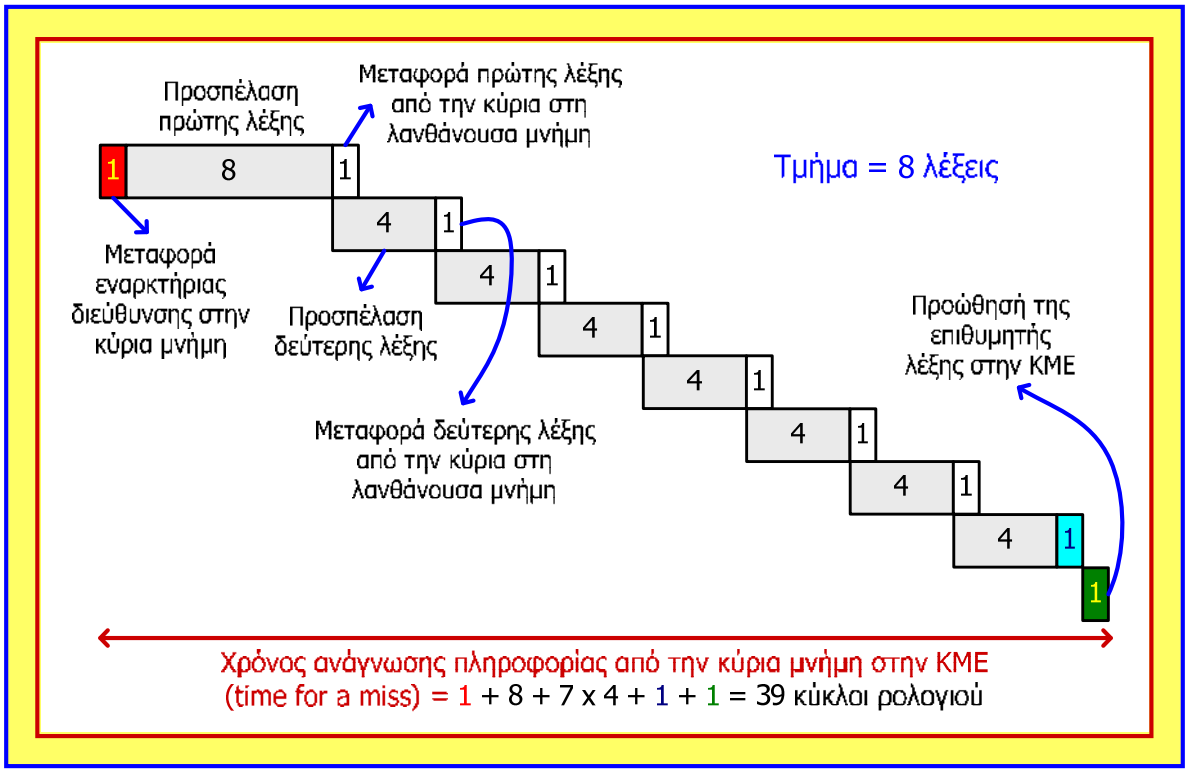
- Χρόνος ανάγνωσης λέξης από τη λανθάνουσα μνήμη: t_h = 1 κύκλος ρολογιού.
- Χρόνος ανάγνωσης τμήματος από κύρια μνήμη: t_m = 1 + 8 + 7 x 4 + 1 + 1 = 39 κύκλοι ρολογιού (η μεταφορά από την κύρια μνήμη στη λανθάνουσα μνήμη γίνεται ανά τμήμα).
- Μέσος χρόνος ανάγνωσης συστήματος μνήμης:

$$t_{ave\text{-}read} = h \times t_h + (1 - h) \times t_m = 0.8 \times 1 + 0.2 \times 39 = 8.6 \text{ κύκλοι ρολογιού}$$

Δίνεται επίσης, ότι στην περίπτωση επιτυχίας εγγραφής ακολουθείται το πρωτόκολλο write-through, ενώ στην περίπτωση αστοχίας εγγραφής το πρωτόκολλο no-write-allocate.

- Χρόνος εγγραφής μιας λέξης στην κύρια μνήμη: M_1 = 1 + 8 = 9 κύκλοι ρολογιού.
- Αφού με το συνδυασμό των 2 πρωτοκόλλων η επιθυμητή λέξη γράφεται στη λανθάνουσα και στην κύρια μνήμη και η αναφορά στην κύρια μνήμη γίνεται παράλληλα με την εγγραφή στη λανθάνουσα, ο μέσος χρόνος εγγραφής είναι $t_{ave\text{-}write}$ = 9 κύκλοι ρολογιού.

Παράδειγμα μέσου χρόνου προσπέλασης μνήμης

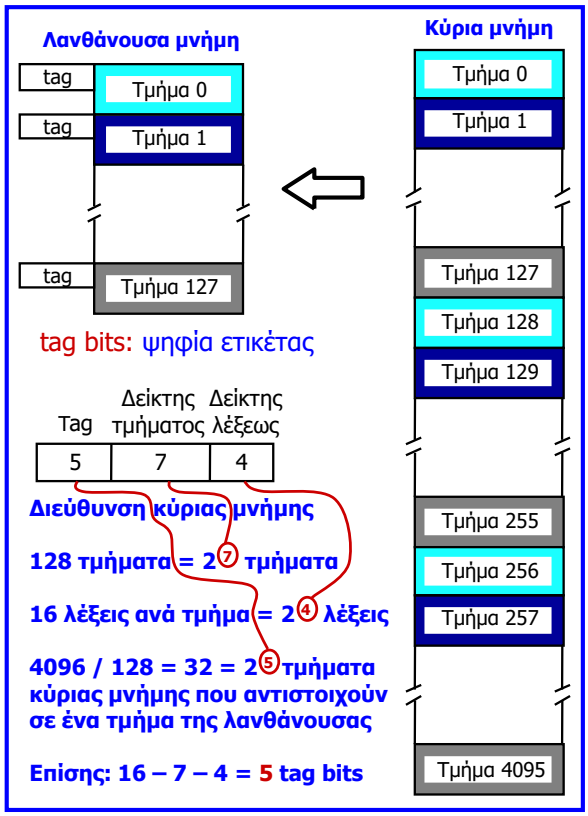


Τρόποι απεικόνισης (οργάνωση) λανθάνουσας μνήμης

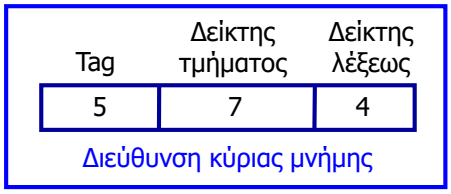
- Ο τρόπος απεικόνισης ή αντιστοίχισης (mapping function) των τμημάτων της κύριας μνήμης σε τμήματα (ή πλαίσια) της λανθάνουσας μνήμης, καθορίζει την οργάνωση της λανθάνουσας μνήμης:
  - ✓ άμεση απεικόνιση (direct mapping),
  - ✓ συσχετιστική απεικόνιση (associative mapping) ή απεικόνιση πλήρους συσχέτισης και
  - ✓ απεικόνιση συσχέτισης συνόλου (set-associative mapping).
- Το πλήθος λέξεων ανά τμήμα λανθάνουσας μνήμης και το πλήθος τμημάτων της λανθάνουσας μνήμης και της κύριας μνήμης είναι δυνάμεις του 2.
- Το τμήμα (πλαίσιο) λανθάνουσας μνήμης και το τμήμα κύριας μνήμης έχουν το ίδιο πλήθος λέξεων.
- Ωστόσο, ένα τμήμα λανθάνουσας μνήμης εκτός από τις λέξεις δεδομένων ή εντολών (που περιέχονται στο αντίστοιχο τμήμα κύριας μνήμης) περιλαμβάνει και έναν αριθμό δυαδικών ψηφίων που αναφέρονται ως ψηφία ετικέτας (tag bits).
- Τα ψηφία ετικέτας κάθε τμήματος είναι ένα μέρος της διεύθυνσης της κύριας μνήμης που υποδεικνύει ποιο τμήμα κύριας μνήμης βρίσκεται κάθε στιγμή αποθηκευμένο στη συγκεκριμένη θέση της λανθάνουσας μνήμης.

Λανθάνουσα μνήμη: άμεση απεικόνιση

- Έστω λανθάνουσα μνήμη 128 τμημάτων με 16 λέξεις το καθένα (δηλαδή μνήμη 2048 λέξεων) και κύρια μνήμη με διευθύνσεις 16 bit και χωρητικότητα 64K λέξεις (4K τμήματα 16 λέξεων).
- Άμεση απεικόνιση (direct mapping): το τμήμα j της κύριας μνήμης απεικονίζεται στο τμήμα [j mod 128] της λανθάνουσας μνήμης.
- Δηλαδή τα τμήματα 0, 128, 256... της κύριας μνήμης αποθηκεύονται στο τμήμα 0 της λανθάνουσας μνήμης, τα 1, 129, 257... στο τμήμα 1 κ.ο.κ.
- Σε κάθε τμήμα της λανθάνουσας μνήμης μπορεί κάθε χρονική στιγμή να βρίσκεται μόνο ένα τμήμα της κύριας μνήμης από το σύνολο των τμημάτων που το υπόλοιπο της διαίρεσης των δεικτών τους διά του πλήθους των τμημάτων της λανθάνουσας μνήμης, δίνει τον αριθμό του τμήματος.



Λανθάνουσα μνήμη: άμεση απεικόνιση

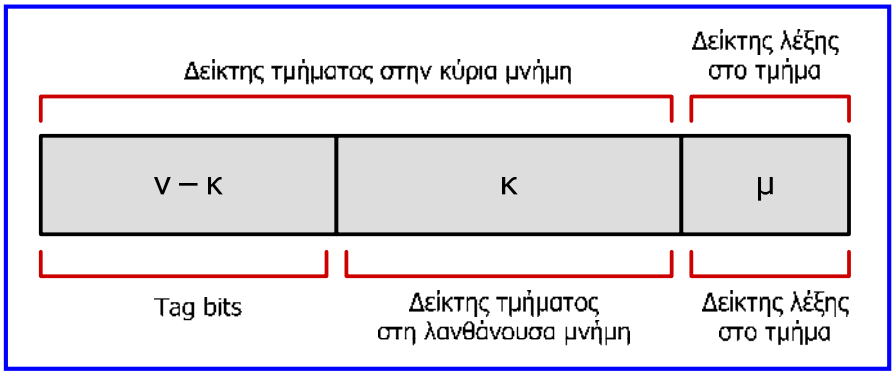


- Τα 4 λιγότερο σημαντικά ψηφία της διεύθυνσης επιλέγουν μία από τις 16 λέξεις ενός τμήματος.
- Τα 7 επόμενα ψηφία της διεύθυνσης δείχνουν μία συγκεκριμένη διεύθυνση τμήματος στην λανθάνουσα μνήμη.
- Τα 5 περισσότερο σημαντικά ψηφία της διεύθυνσης καθορίζουν ποιο από τα 32 τμήματα ($4096 / 128 = 32 = 2^5$), τα οποία αντιστοιχούν στη συγκεκριμένη θέση τμήματος στη λανθάνουσα μνήμη, απαιτείται από το πρόγραμμα την τρέχουσα χρονική στιγμή.
- Κατά την εκτέλεση προγράμματος, τα 5 περισσότερο σημαντικά ψηφία της διεύθυνσης κύριας μνήμης συγκρίνονται με τα ψηφία ετικέτας (tag bits) που έχουν αποθηκευτεί στη λανθάνουσα μνήμη, ώστε να δείχνουν ποιο από τα 32 τμήματα που απεικονίζονται στη συγκεκριμένη θέση τμήματος, βρίσκεται την τρέχουσα χρονική στιγμή στη λανθάνουσα μνήμη.
- Εάν συμπίπτουν, τότε η επιθυμητή λέξη βρίσκεται μέσα σε αυτό το τμήμα της λανθάνουσας μνήμης, διαφορετικά θα πρέπει να γίνει αναφορά στην κύρια μνήμη.

Λανθάνουσα μνήμη: άμεση απεικόνιση

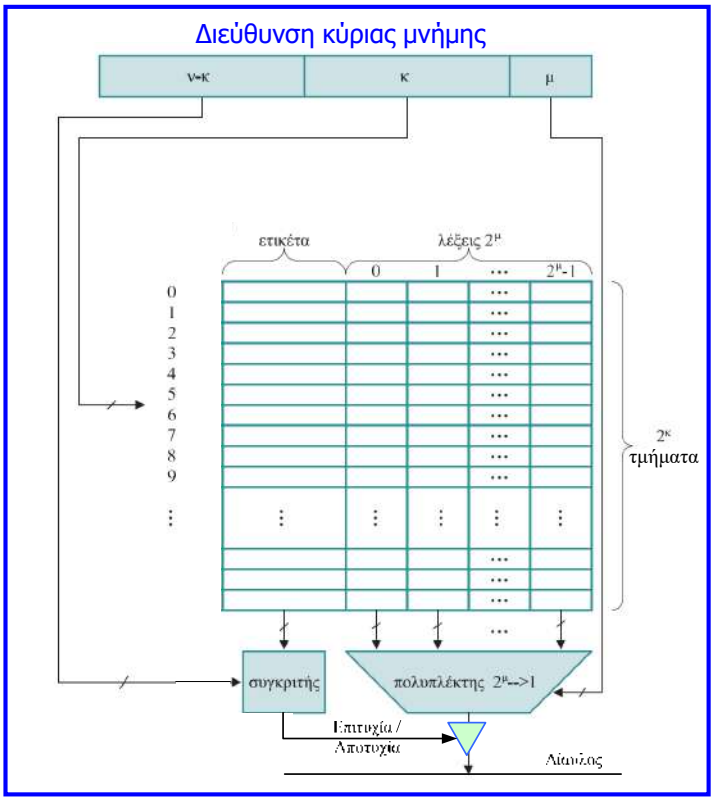
- Πλήθος τμημάτων κύριας μνήμης: 2^{v}
- Πλήθος λέξεων ανά τμήμα: 2^{μ}
- Μέγεθος κύριας μνήμης: $2^{v+\mu}$ λέξεις
- Πλήθος τμημάτων λανθάνουσας μνήμης: 2^{κ}
- Μέγεθος λανθάνουσας μνήμης: $2^{\kappa+\mu}$ λέξεις

Τρόπος χρήσης της διεύθυνσης που παράγει η ΚΜΕ για προσπέλαση λανθάνουσας μνήμης με άμεση απεικόνιση

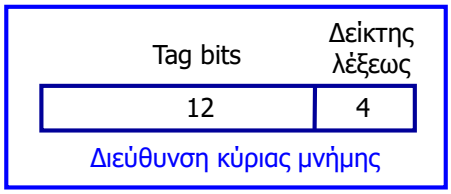


Λανθάνουσα μνήμη: άμεση απεικόνιση

- Εάν από το συγκριτή προκύψει ότι τα v-κ ψηφία της διεύθυνσης κύριας μνήμης συμπίπτουν με τα ψηφία ετικέτας του τμήματος που καθορίζεται από τα κ ψηφία, τότε η ζητούμενη λέξη περιλαμβάνεται στα δεδομένα των εισόδων του πολυπλέκτη.
- Η ζητούμενη λέξη επιλέγεται με βάση τα μ ψηφία της διεύθυνσης (δείκτης λέξης) και μεταφέρεται από την έξοδο του πολυπλέκτη στον επεξεργαστή μέσω διαύλου.
- Εάν δε συμπίπτουν, η έξοδος του πολυπλέκτη αγνοείται και με βάση τη διεύθυνση της κύριας μνήμης τα ζητούμενα δεδομένα προσκομίζονται από την κύρια μνήμη στη λανθάνουσα μνήμη.



Λανθάνουσα μνήμη: συσχετιστική απεικόνιση

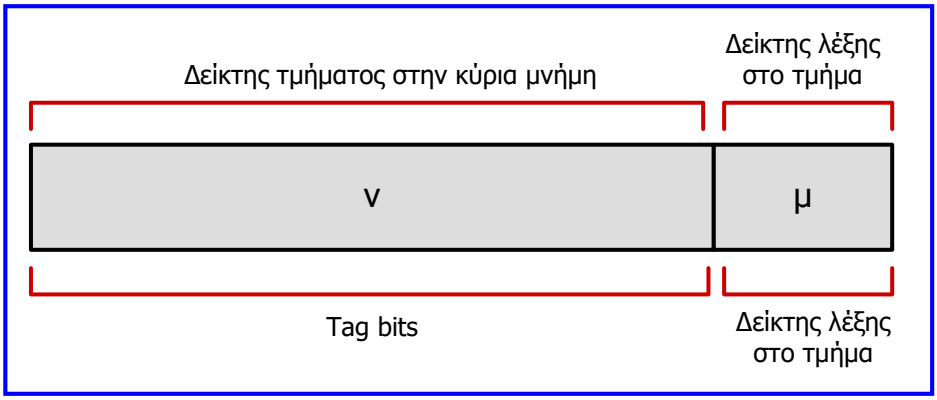


- Συσχετιστική απεικόνιση (associative mapping): πιο ευέλικτη μέθοδος κατά την οποία ένα τμήμα της κύριας μνήμης μπορεί να τοποθετηθεί σε οποιαδήποτε θέση τμήματος της λανθάνουσας μνήμης.
- Τα 12 ψηφία ετικέτας της διεύθυνσης συγκρίνονται με τα ψηφία ετικέτας όλων των τμημάτων της λανθάνουσας μνήμης για να διαπιστωθεί εάν το επιθυμητό τμήμα περιλαμβάνεται στη λανθάνουσα μνήμη.
- Με την τεχνική αυτή, ένα νέο τμήμα το οποίο θα πρέπει να εισαχθεί στη λανθάνουσα μνήμη θα πρέπει να αντικαταστήσει ένα ήδη υπάρχον τμήμα μόνο στη περίπτωση όπου η λανθάνουσα μνήμη είναι εντελώς γεμάτη.
- Η επιβάρυνση που εισάγει η τεχνική αυτή έγκειται στο γεγονός ότι θα πρέπει να γίνει σύγκριση με το σύνολο των $2^7 = 128$ συνδυασμών ψηφίων ετικέτας (tag bits).

Λανθάνουσα μνήμη: συσχετιστική απεικόνιση

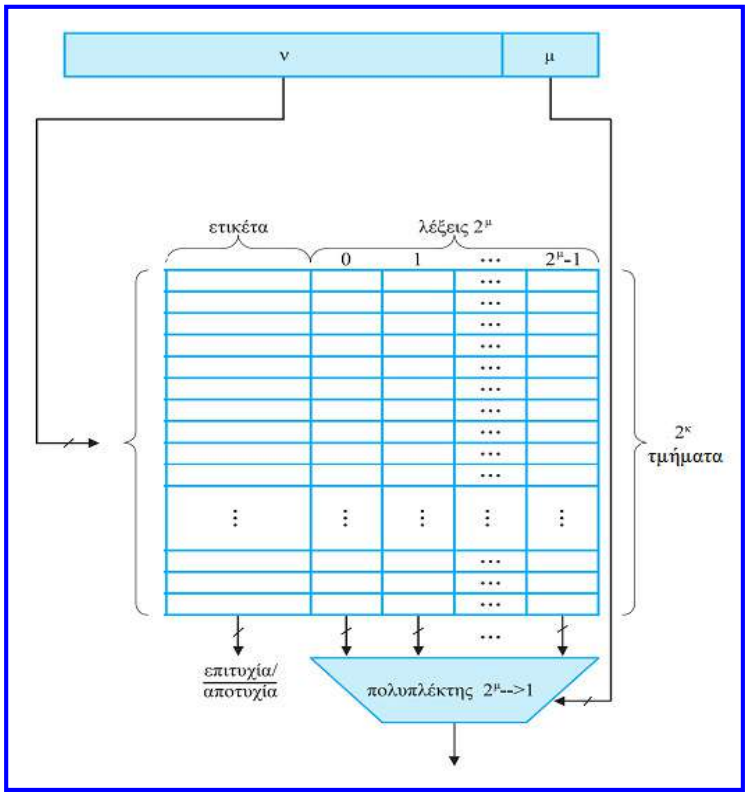
- Πλήθος τμημάτων κύριας μνήμης: 2^v
- Πλήθος λέξεων τμήματος: 2^μ
- Μέγεθος κύριας μνήμης: $2^{v+\mu}$ λέξεις
- Πλήθος τμημάτων λανθάνουσας μνήμης: 2^κ
- Μέγεθος λανθάνουσας μνήμης: $2^{\kappa+\mu}$ λέξεις

Τρόπος χρήσης της διεύθυνσης που παράγει η ΚΜΕ για προσπέλαση λανθάνουσας μνήμης με συσχετιστική απεικόνιση



Λανθάνουσα μνήμη: συσχετιστική απεικόνιση

- Το πεδίο της ετικέτας της διεύθυνσης που παράγει η ΚΜΕ συγκρίνεται ταυτόχρονα με όλες τις ετικέτες που είναι αποθηκευμένες στη λανθάνουσα μνήμη.
- Εάν το πεδίο της ετικέτας είναι ίδιο με κάποια από τις αποθηκευμένες ετικέτες, τότε συμβαίνει επιτυχία.
- Η πληροφορία του τμήματος που συνδέεται με αυτή την ετικέτα εμφανίζεται στις εισόδους του πολυπλέκτη και με βάση το δείκτη της λέξης προκύπτει στην έξοδο του πολυπλέκτη η επιθυμητή λέξη.



Λανθάνουσα μνήμη: συσχετιστική απεικόνιση

Παράδειγμα:

Κύρια μνήμη: 64K λέξεις

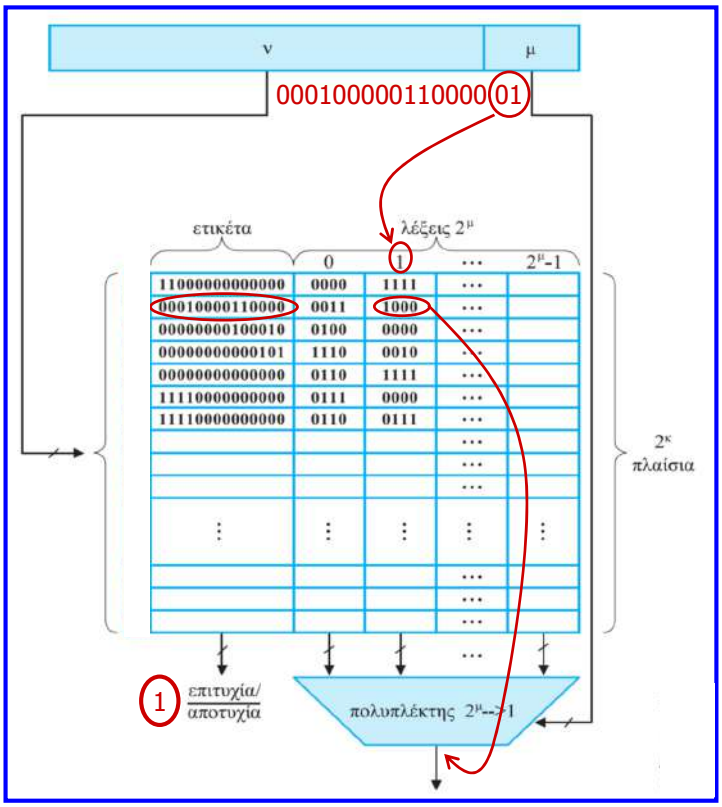
Λανθάνουσα μνήμη: 64 λέξεις

Τμήμα: 4 λέξεις

Επομένως, η λανθάνουσα μνήμη αποτελείται από 16 τμήματα και η κύρια μνήμη αποτελείται από 16K τμήματα.

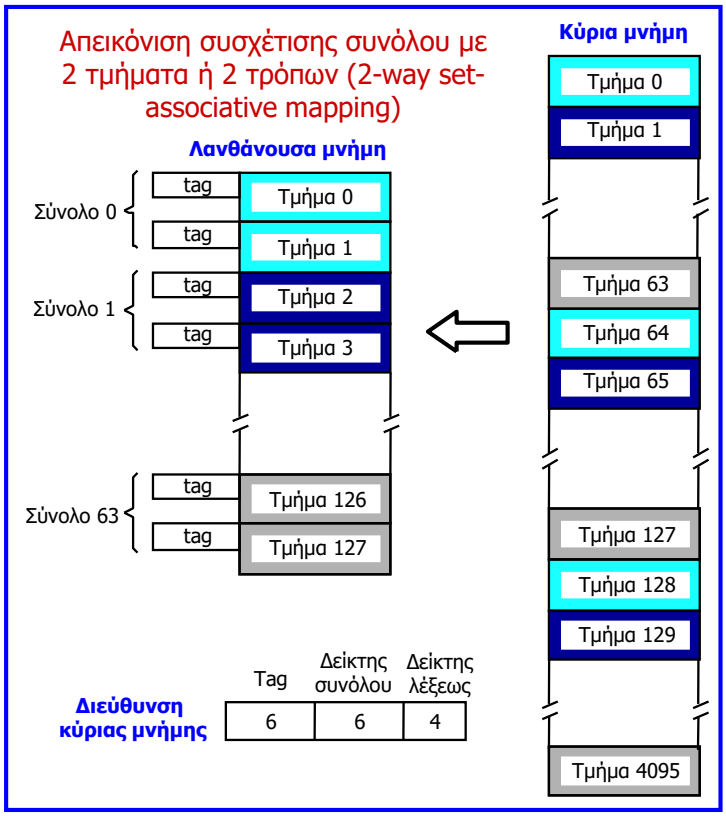
Αφού $4 = 2^2 \Rightarrow \mu = 2$ και

αφού $16K = 2^4 \cdot 2^{10} = 2^{14} \Rightarrow \nu = 14$



Λανθάνουσα μνήμη: απεικόνιση συσχέτισης συνόλου

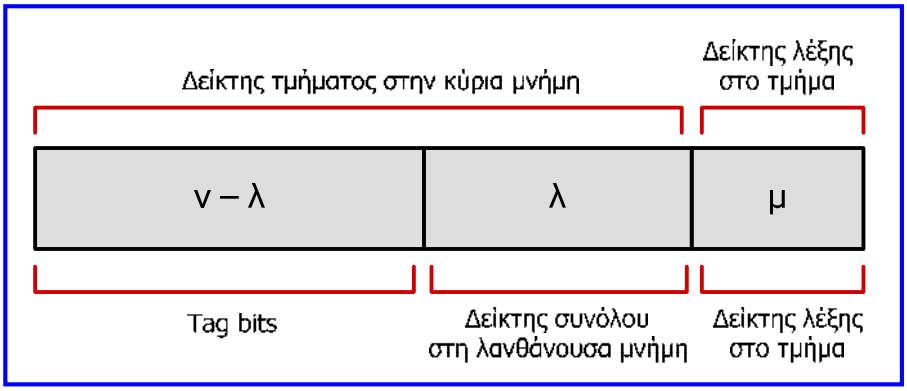
- Απεικόνιση συσχέτισης συνόλου (set-associative mapping): συνδυασμός των δύο προηγούμενων τρόπων.
- Τα 128 τμήματα της λανθάνουσας μνήμης ομαδοποιούνται σε 64 σύνολα και ένα τμήμα κύριας μνήμης μπορεί να αποθηκευτεί σε οποιοδήποτε τμήμα ενός συγκεκριμένου συνόλου.
- Το τμήμα j της κύριας μνήμης απεικονίζεται στο σύνολο [j mod 64] της λανθάνουσας μνήμης.
- Έτσι, τα τμήματα κύριας μνήμης 0, 64, 128, ..., 4032 απεικονίζονται στο σύνολο 0 της λανθάνουσας μνήμης και μπορούν να καταλαμβάνουν οποιαδήποτε από τις 2 θέσεις τμήματος στο σύνολο αυτό.



Λανθάνουσα μνήμη: απεικόνιση συσχέτισης συνόλου

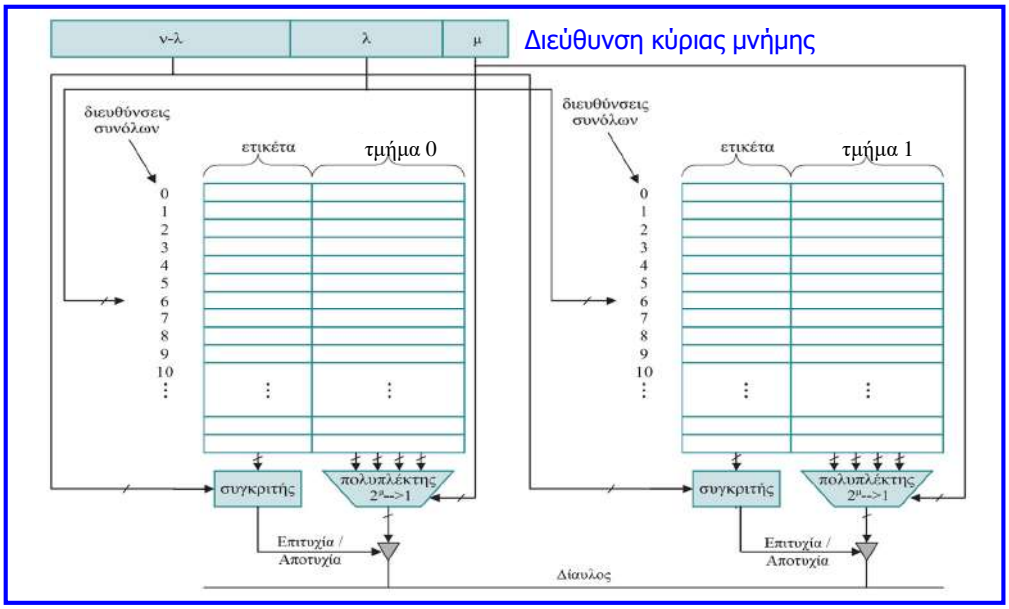
- Πλήθος τμημάτων κύριας μνήμης: 2^{v}
- Πλήθος λέξεων ανά τμήμα: 2^{μ}
- Μέγεθος κύριας μνήμης: $2^{v+\mu}$ λέξεις
- Πλήθος συνόλων λανθάνουσας μνήμης: 2^{λ}
- Πλήθος τμημάτων ανά σύνολο: 2^{ρ}
- Μέγεθος λανθάνουσας μνήμης: $2^{\lambda+\rho+\mu}$ λέξεις

Τρόπος χρήσης της διεύθυνσης που παράγει η ΚΜΕ για προσπέλαση λανθάνουσας μνήμης με απεικόνιση συσχέτισης συνόλου



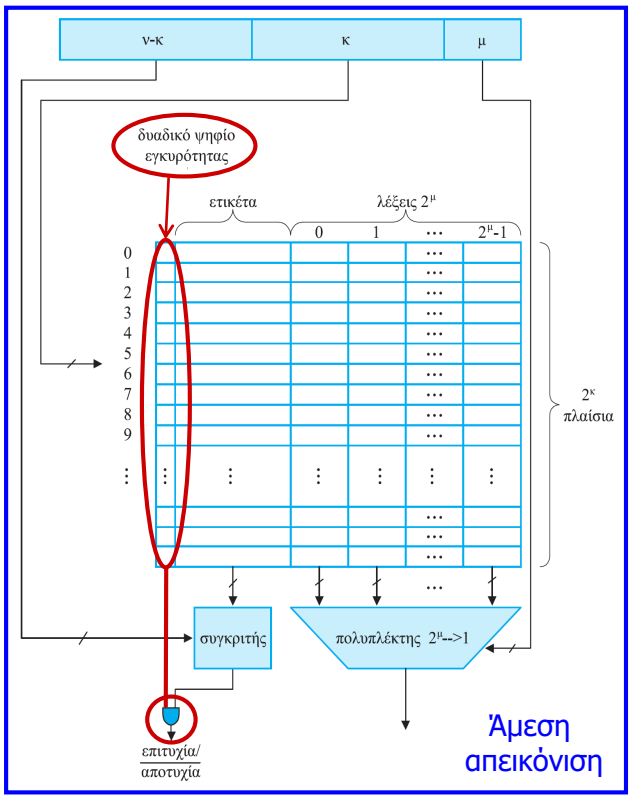
Λανθάνουσα μνήμη: απεικόνιση συσχέτισης συνόλου

- Τα ψηφία του πεδίου ετικέτας της διεύθυνσης συγκρίνονται με τις δύο ετικέτες καθεμία από τις οποίες είναι αποθηκευμένη σε μία από τις δύο μνήμες που συνιστούν τη λανθάνουσα μνήμη (στη θέση που υποδεικνύει ο δείκτης του συνόλου της διεύθυνσης).
- Εάν μία από τις συγκρίσεις έχει θετικό αποτέλεσμα, τότε συμβαίνει επιτυχία.



Λανθάνουσα μνήμη: ψηφία εγκυρότητας

- Κατά την 1η προσπέλαση ενός τμήματος, το πεδίο ετικέτας της διεύθυνσης που παράγει ο επεξεργαστής, μπορεί να συμπίπτει με την ετικέτα που είναι αποθηκευμένη στη λανθάνουσα μνήμη.
- Ωστόσο κατά την έναρξη εκτέλεσης ενός προγράμματος, η λανθάνουσα μνήμη περιέχει τυχαία (μη έγκυρη) πληροφορία, με αποτέλεσμα αυτή να ληφθεί ως επιθυμητή (έγκυρη).
- Για το λόγο αυτό, στη λανθάνουσα μνήμη χρησιμοποιούνται ψηφία εγκυρότητας, τα οποία κατά την έναρξη εκτέλεσης ενός προγράμματος γίνονται 0 και όταν προσκομίζεται ένα τμήμα κύριας μνήμης στη θέση τμήματος της λανθάνουσας, μνήμης, τότε στο αντίστοιχο ψηφίο εγκυρότητας γίνεται 1.
- Για να συμβεί επιτυχία θα πρέπει και το ψηφίο εγκυρότητας να είναι 1.



Λανθάνουσα μνήμη: σύγκριση τρόπων απεικόνισης

- Τα πλεονεκτήματα της άμεσης απεικόνισης είναι ο μικρός χρόνος προσπέλασης και το χαμηλό κόστος υλοποίησης.
- Αντιθέτως, τα μειονεκτήματα της συσχετιστικής απεικόνισης είναι ο μεγάλος χρόνος προσπέλασης και το υψηλό κόστος υλοποίησης.
- Το γεγονός ότι στην άμεση απεικόνιση, δύο ή περισσότερα τμήματα της κύριας μνήμης που αντιστοιχούν στο ίδιο τμήμα της λανθάνουσας μνήμης δεν μπορούν να βρίσκονται ταυτόχρονα στη λανθάνουσα μνήμη, έχει σαν αποτέλεσμα υψηλότερο ρυθμό αστοχιών.
- Αντιθέτως, στην συσχετιστική απεικόνιση, επειδή η λανθάνουσα μνήμη μπορεί να περιέχει οποιοδήποτε συνδυασμό από τμήματα κύριας μνήμης, ο ρυθμός αστοχιών είναι μικρότερος.
- Η απεικόνιση συσχέτισης συνόλου με τ τμήματα ανά σύνολο, για τ = 1 ισοδυναμεί με άμεση απεικόνιση, ενώ όταν το πλήθος τμημάτων ανά σύνολο είναι ίσο με το πλήθος των τμημάτων της λανθάνουσας μνήμης, τότε ισοδυναμεί με συσχετιστική απεικόνιση.
- Επομένως, η απεικόνιση συσχέτισης συνόλου αποτελεί μια λύση εξισορρόπησης των μειονεκτημάτων και των πλεονεκτημάτων των δύο άλλων τρόπων απεικόνισης που χρησιμοποιούνται στη λανθάνουσα μνήμη.

Λανθάνουσα μνήμη: αλγόριθμοι αντικατάστασης

- Όταν η ΚΜΕ ζητήσει πληροφορία από τη λανθάνουσα μνήμη και αυτή δεν είναι διαθέσιμη, τότε θα πρέπει το τμήμα της κύριας μνήμης που περιέχει τη ζητούμενη πληροφορία να προσκομιστεί από την κύρια μνήμη.
- Εάν η λανθάνουσα μνήμη είναι γεμάτη, τότε το τμήμα που θα προσκομιστεί θα πρέπει να αντικαταστήσει κάποιο από τα αποθηκευμένα στη λανθάνουσα μνήμη.
- Οι αλγόριθμοι αντικατάστασης τμημάτων λανθάνουσας μνήμης για την προσκόμιση τμημάτων κύριας μνήμης καθορίζουν το τμήμα που θα αντικατασταθεί.
- Στη λανθάνουσα μνήμη με άμεση απεικόνιση, η θέση κάθε τμήματος είναι προκαθορισμένη, επομένως δεν απαιτείται αλγόριθμος αντικατάστασης.
- Στη λανθάνουσα μνήμη με συσχετιστική απεικόνιση ή με απεικόνιση συσχέτισης συνόλου, αφού ένα τμήμα της κύριας μνήμης μπορεί να τοποθετηθεί σε περισσότερα από ένα τμήματα της λανθάνουσας μνήμης, θα πρέπει να επιλεγεί για αντικατάσταση ένα από αυτά τα τμήματα με βάση κάποιον αλγόριθμο αντικατάστασης.
- Στις περιπτώσεις αυτές, είναι συνήθως ζητούμενο ο ελεγκτής της λανθάνουσας μνήμης να κρατήσει τμήματα τα οποία είναι πιθανό να είναι χρήσιμα από το πρόγραμμα στο κοντινό μέλλον.

Λανθάνουσα μνήμη: αλγόριθμοι αντικατάστασης

- Οι πιο κοινοί αλγόριθμοι είναι οι παρακάτω και αντικαθιστούν:
  - ✓ τυχαίο τμήμα,
  - ✓ το μη χρησιμοποιηθέν πρόσφατα τμήμα (least-recently used, LRU): αντικαθίσταται το τμήμα του οποίου δεν έχει γίνει χρήση (ανάγνωση ή εγγραφή) πρόσφατα, έτσι ώστε να ελαττωθεί η πιθανότητα απομάκρυνσης πληροφορίας που θα χρειαστεί στο άμεσο μέλλον (σπανιότερα χρησιμοποιείται ο αλγόριθμος MRU, most-recently used),
  - ✓ το τμήμα που προσκομίστηκε πρώτο στη λανθάνουσα μνήμη (first-in / first-out, FIFO),
  - ✓ το μη χρησιμοποιηθέν συχνά τμήμα (least-frequently-used, LFU).
- Κατά την εφαρμογή του LRU (που είναι δημοφιλής αλγόριθμος), ο ελεγκτής λανθάνουσας μνήμης διαθέτει έναν μετρητή για κάθε τμήμα. Για παράδειγμα, στην απεικόνιση συσχέτισης συνόλου με 4 τμήματα ανά σύνολο, γίνεται χρήση μετρητών των 2 bits.
- Όταν συμβαίνει επιτυχία, ο μετρητής του αντίστοιχου τμήματος μηδενίζεται, ενώ οι μετρητές με τιμές μικρότερες από αυτόν αυξάνονται κατά 1 και οι υπόλοιποι δεν αλλάζουν.
- Όταν συμβεί αστοχία και το σύνολο δεν είναι πλήρες, ο μετρητής του νέου τμήματος μηδενίζεται και οι υπόλοιποι αυξάνονται κατά 1.
- Όταν συμβεί αστοχία και το σύνολο είναι πλήρες, το τμήμα με τιμή 3 απομακρύνεται, το νέο τμήμα λαμβάνει τη θέση του, ο μετρητής του μηδενίζεται, ενώ οι υπόλοιποι 3 μετρητές αυξάνονται κατά 1.

Λανθάνουσα μνήμη: παραδείγματα τρόπων απεικόνισης

Παράδειγμα 1:

- Θεωρούμε ότι ο επεξεργαστής ενός υπολογιστή παράγει διευθύνσεις με 8 δυαδικά ψηφία, κάθε διεύθυνση δείχνει σε μια θέση μνήμης και κάθε θέση μνήμης περιέχει 32 bits (μήκος λέξης δεδομένων).
- Η κεντρική μονάδα επεξεργασίας δεδομένων παράγει την παρακάτω ακολουθία διευθύνσεων:

  16h, 8Ch, 14h, 03h, 12h, 1Bh, 90h, 8Ah.
- Για λανθάνουσα μνήμη με άμεση απεικόνιση και χωρητικότητα 64 λέξεων με 8 λέξεις ανά τμήμα, θα υπολογίσουμε το ποσοστό ή ρυθμό ευστοχίας (hit ratio) και το συνολικό πλήθος των bytes που διαβάζονται από την κύρια μνήμη.
- Θεωρούμε επίσης, ότι αρχικά η λανθάνουσα μνήμη είναι κενή ή ότι δεν περιέχει έγκυρα δεδομένα.

Λανθάνουσα μνήμη: παραδείγματα τρόπων απεικόνισης

- Η λανθάνουσα μνήμη χρησιμοποιεί άμεση απεικόνιση και έχει χωρητικότητα 64 λέξεων με 8 λέξεις ανά τμήμα, επομένως περιλαμβάνει 8 τμήματα με 8 λέξεις το καθένα.
- Έτσι, από τη διεύθυνση των 8 bits, χρησιμοποιούνται 3 bits για τον καθορισμό της λέξης μέσα στο τμήμα, 3 bits για τον καθορισμό του τμήματος και 8 – 3 – 3 = 2 bits για ετικέτα.

Τμήμα κύριας μνήμης:
Μ (διεύθυνση πρώτης λέξης – διεύθυνση τελευταίας λέξης)

Διεύθυνση μνήμης	Ετικέτα – Τμήμα – Λέξη	HIT ή MISS	Τμήμα κύριας μνήμης → Τμήμα λανθάνουσας μνήμης
16	00-010-110	MISS	M(10-17) → T2
8C	10-001-100	MISS	M(88-8F) → T1
14	00-010-100	**HIT**	T2
03	00-000-011	MISS	M(0-7) → T0
12	00-010-010	**HIT**	T2
1B	00-011-011	MISS	M(18-1F) → T3
90	10-010-000	MISS	M(90-97) → T2, **αντικατάσταση**
8A	10-001-010	**HIT**	T1

Ποσοστό ή ρυθμός ευστοχίας (hit rate): 3/8 ή 37.5%

Λανθάνουσα μνήμη: παραδείγματα τρόπων απεικόνισης

- Αφού το ποσοστό ευστοχίας για τη λανθάνουσα μνήμη με άμεση απεικόνιση είναι 3/8, απαιτείται οι 5 από τις 8 προσβάσεις να εξυπηρετηθούν από την κύρια μνήμη.
- Με δεδομένο ότι στις περιπτώσεις αστοχίας πρέπει από την κύρια μνήμη να μεταφερθεί ένα τμήμα, το οποίο αποτελείται από 8 λέξεις, απαιτείται η μεταφορά από την κύρια μνήμη 5 x 8 = 40 λέξεων. Αφού η κάθε λέξη είναι 32 bits, δηλαδή 4 bytes, θα πρέπει να μεταφερθούν από την κύρια μνήμη συνολικά 160 bytes.

Λανθάνουσα μνήμη: παραδείγματα τρόπων απεικόνισης

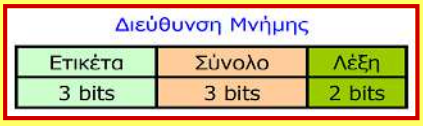
Παράδειγμα 2:

- Θεωρούμε ότι ο επεξεργαστής ενός υπολογιστή παράγει διευθύνσεις με 8 δυαδικά ψηφία, κάθε διεύθυνση δείχνει σε μια θέση μνήμης και κάθε θέση μνήμης περιέχει 32 bits (μήκος λέξης δεδομένων).
- Η κεντρική μονάδα επεξεργασίας δεδομένων παράγει την παρακάτω ακολουθία διευθύνσεων:

  16h, 8Ch, 14h, 03h, 37h, 75h, 8Eh, 0Ch, 2Fh.
- Για λανθάνουσα μνήμη με απεικόνιση συσχέτισης συνόλου δύο (2) τρόπων και χωρητικότητα 64 λέξεων και 4 λέξεις ανά τμήμα, θα υπολογίσουμε το ποσοστό ή ρυθμό ευστοχίας (hit ratio) και το συνολικό πλήθος των bytes που διαβάζονται από την κύρια μνήμη.
- Θεωρούμε επίσης, ότι αρχικά η λανθάνουσα μνήμη είναι κενή ή δεν περιέχει έγκυρα δεδομένα και ότι χρησιμοποιεί τον αλγόριθμο αντικατάστασης τμημάτων LRU.

Λανθάνουσα μνήμη: παραδείγματα τρόπων απεικόνισης

- Η λανθάνουσα μνήμη χρησιμοποιεί απεικόνιση συσχέτισης συνόλου με 2 τμήματα ανά σύνολο και έχει χωρητικότητα 64 λέξεων με 4 λέξεις ανά τμήμα, επομένως περιλαμβάνει 16 / 2 = 8 σύνολα με 2 τμήματα το καθένα.
- Έτσι, από τη διεύθυνση των 8 bits, χρησιμοποιούνται 2 bits για τον καθορισμό της λέξης μέσα στο τμήμα, 3 bits για τον καθορισμό του συνόλου και 8 – 2 – 3 = 3 bits για ετικέτα.



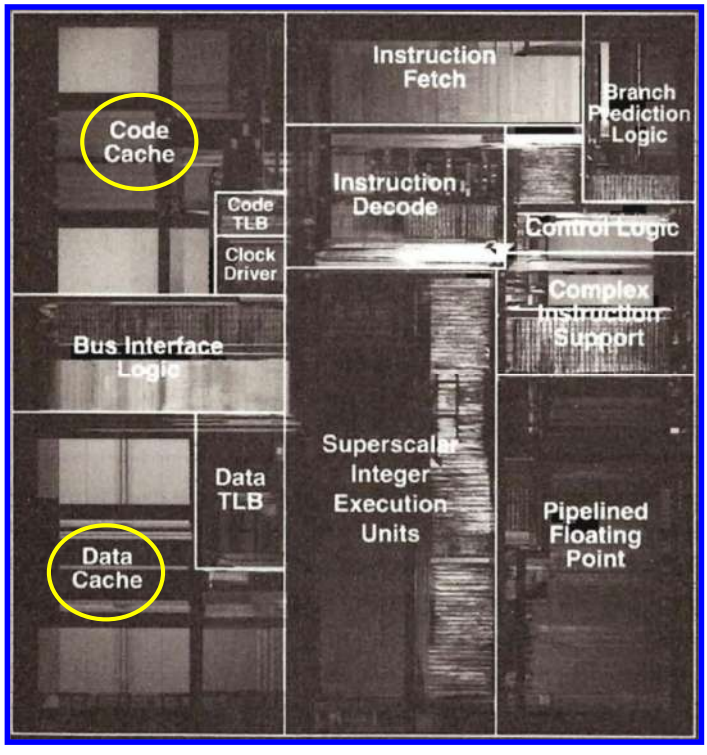
Διεύθυνση μνήμης	Ετικέτα – Σύνολο – Λέξη	HIT ή MISS	Τμήμα κύριας μνήμης → Σύνολο και Τμήμα λανθάνουσας μνήμης
16	000-101-10	MISS	M(14-17) → (Σ5, T0)
8C	100-011-00	MISS	M(8C-8F) → (Σ3, T0)
14	000-101-00	**HIT**	(Σ5, T0)
03	000-000-11	MISS	M(0-3) → (Σ0, T0)
37	001-101-11	MISS	M(34-37) → (Σ5, T1)
75	011-101-01	MISS	M(74-77) → (Σ5, T0), **αντικατάσταση**
8E	100-011-10	**HIT**	(Σ3, T0)
0C	000-011-00	MISS	M(0C-0F) → (Σ3, T1)
2F	001-011-11	MISS	M(2C-2F) → (Σ3, T0), **αντικατάσταση**

Ποσοστό ή ρυθμός ευστοχίας (hit rate): 2/9 ή 22.2%

Λανθάνουσα μνήμη: παραδείγματα τρόπων απεικόνισης

- Αφού το ποσοστό ευστοχίας για τη λανθάνουσα μνήμη με απεικόνιση συσχέτισης συνόλου δύο τρόπων είναι 2/9, απαιτείται οι 7 από τις 9 προσβάσεις να εξυπηρετηθούν από την κύρια μνήμη.
- Με δεδομένο ότι στις περιπτώσεις αστοχίας πρέπει από την κύρια μνήμη να μεταφερθεί ένα τμήμα, το οποίο αποτελείται από 4 λέξεις, απαιτείται η μεταφορά από την κύρια μνήμη 7 x 4 = 28 λέξεων. Αφού η κάθε λέξη είναι 32 bits, δηλαδή 4 bytes, θα πρέπει να μεταφερθούν από την κύρια μνήμη συνολικά 112 bytes.

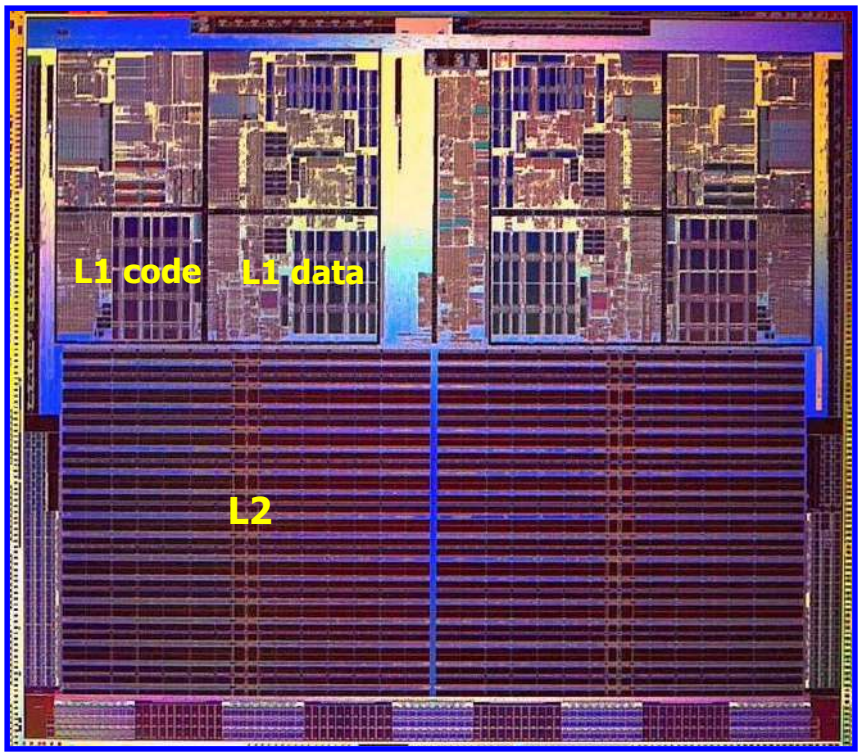
Λανθάνουσα μνήμη του Intel Pentium (1993)



8 Kbytes L1 δεδομένων και 8 Kbytes L1 εντολών με απεικόνιση συσχέτισης συνόλου με 2 τμήματα ανά σύνολο και αλγόριθμο αντικατάστασης τμημάτων LRU.

Χρησιμοποιεί δύο εναλλακτικές τακτικές ενημέρωσης της μνήμης ανώτερου επιπέδου: write-through και write-back

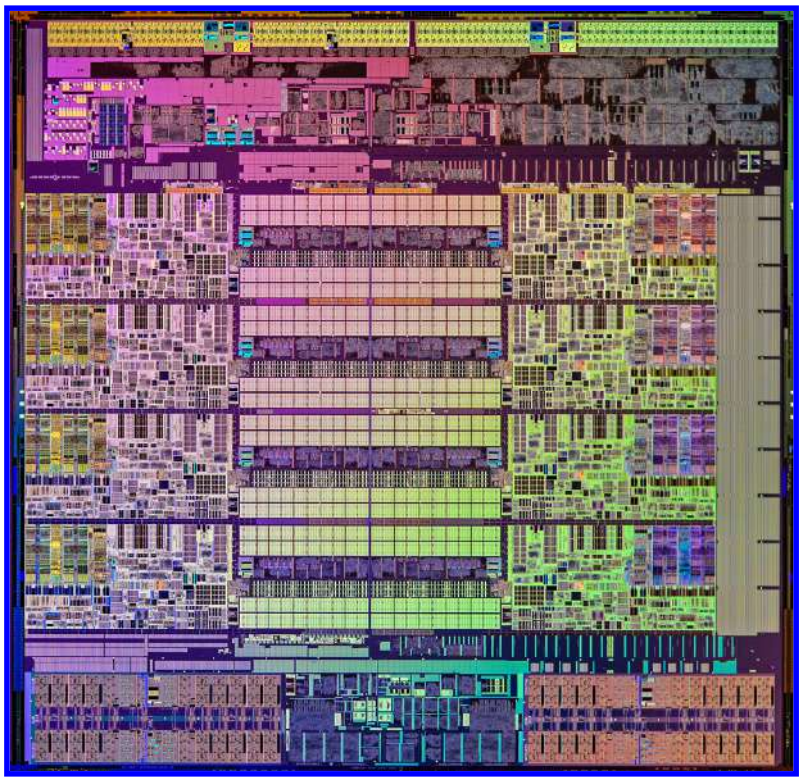
Λανθάνουσα μνήμη διπύρηνου AMD Athlon 64 (2005)



64 Kbytes L1 δεδομένων και 64 Kbytes L1 εντολών ανά πυρήνα με απεικόνιση συσχέτισης συνόλου με 2 τμήματα ανά σύνολο.

512 Kbytes L2 ανά πυρήνα με απεικόνιση συσχέτισης συνόλου με 16 τμήματα ανά σύνολο.

Λανθάνουσα μνήμη οκταπύρηνου Intel Core i7 (2014)



32 Kbytes L1 δεδομένων και 32 Kbytes L1 εντολών, ανά πυρήνα με απεικόνιση συσχέτισης συνόλου με 8 τμήματα ανά σύνολο.

256 Kbytes L2 ανά πυρήνα με απεικόνιση συσχέτισης συνόλου με 8 τμήματα ανά σύνολο.

20 Mbytes L3, κοινή για τους 8 πυρήνες με απεικόνιση συσχέτισης συνόλου με 20 τμήματα ανά σύνολο.

Απόδοση συστήματος μνήμης

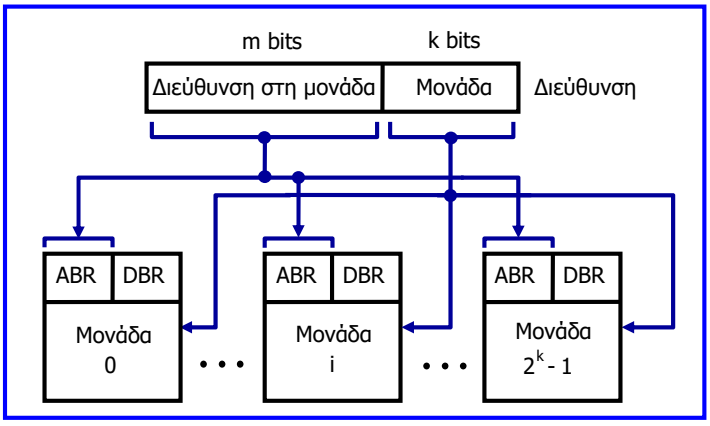
- Η απόδοση ενός υπολογιστικού συστήματος εξαρτάται από την ταχύτητα προσαγωγής των εντολών μηχανής στον επεξεργαστή για εκτέλεση και την ταχύτητα με την οποία στη συνέχεια μπορούν οι εντολές αυτές να εκτελεστούν.
- Η ταχύτητα εκτέλεσης εντολών εξαρτάται και από την ταχύτητα μεταφοράς των δεδομένων που συμμετέχουν σε αυτές.
- Η ιεραρχία μνήμης που χρησιμοποιείται σήμερα στα υπολογιστικά συστήματα αποτελεί αποτέλεσμα της αναζήτησης χαμηλότερου κόστους, μικρότερου χρόνου προσπέλασης και μεγαλύτερης χωρητικότητας.
- Για αύξηση της ταχύτητας των αργών αποθηκευτικών μέσων της ιεραρχίας, εφαρμόζεται παραλληλοποίηση με χρήση οργάνωσης διαφύλλωσης (interleaved organization).
- Σημαντικό δείκτη αποδοτικότητας ενός συστήματος μνήμης αποτελεί ο ρυθμός ευστοχίας κατά την προσπέλαση πληροφορίας στα διάφορα επίπεδα ιεραρχίας.
- Η βέλτιστη θέση της λανθάνουσας μνήμης (όσον αφορά την ταχύτητα) είναι στο εσωτερικό του ολοκληρωμένου κυκλώματος του επεξεργαστή.
- Για βελτίωση της απόδοσης του συστήματος μνήμης, χρησιμοποιούνται διάφορες τεχνικές όπως είναι η προανάκτηση δεδομένων (prefetching) από την κύρια στη λανθάνουσα μνήμη και η χρήση λανθάνουσας μνήμης που μπορεί να εξυπηρετήσει πολλαπλές ανακτήσεις από την κύρια μνήμη (lockup-free cache).

Απόδοση συστήματος μνήμης

- Όπως προαναφέρθηκε, ο αριθμός επιτυχιών κατά την προσπέλαση της λανθάνουσας μνήμης προς το συνολικό αριθμό προσπαθειών προσπέλασης ονομάζεται ρυθμός ευστοχίας (hit rate, h). Αντίστοιχα, ορίζεται ο ρυθμός αστοχίας (miss rate, 1 – h).
- Ο χρόνος που απαιτείται για να προσπελαστεί και να μεταφερθεί η επιθυμητή πληροφορία από την κύρια μνήμη στη λανθάνουσα, ονομάζεται ποινή αστοχίας (miss penalty, M) και κατά το χρόνο αυτό η ΚΜΕ τελεί σε στασιμότητα, αφού οι απαιτούμενες εντολές (ή δεδομένα) δεν είναι στη διάθεσή της για τη συνέχιση της εκτέλεσης του προγράμματος.
- Για ένα επίπεδο λανθάνουσας μνήμης, ο μέσος χρόνος προσπέλασης του συστήματος μνήμης είναι: $\mathbf{h} \times \mathbf{C} + \mathbf{(1 - h)} \times \mathbf{(M + C)} = \mathbf{C} + \mathbf{(1 - h)} \times \mathbf{M}$, όπου C ο χρόνος προσπέλασης / μεταφοράς πληροφορίας από τη λανθάνουσα μνήμη στον επεξεργαστή.
- Για 2 επίπεδα λανθάνουσας μνήμης (L1, L2), ο μέσος χρόνος προσπέλασης του συστήματος μνήμης είναι: $\mathbf{h_1} \times \mathbf{C_1} + \mathbf{(1 - h_1)} \times \mathbf{h_2} \times \mathbf{C_2} + \mathbf{(1 - h_1)} \times \mathbf{(1 - h_2)} \times \mathbf{(M + C_2)}$, όπου C_1 ο χρόνος προσπέλασης / μεταφοράς πληροφορίας από την L1 στον επεξεργαστή, C_2 ο χρόνος προσπέλασης / μεταφοράς πληροφορίας από την L2 στον επεξεργαστή, h_1, h_2 ο ρυθμός ευστοχίας της L1 και L2, αντίστοιχα και M ο χρόνος προσπέλασης / μεταφοράς πληροφορίας από την κύρια μνήμη στην L2.
- Επειδή, το γινόμενο $(1 - h_1) \times (1 - h_2)$ είναι αρκετά μικρό (0.01 για h_1, $h_2 = 0.9$), η ποινή αστοχίας M καθίσταται λιγότερο κρίσιμη για την απόδοση του συστήματος.

Διαφύλλωση (interleaving) κύριας μνήμης

- Η διαφύλλωση μνήμης είναι τεχνική που ελαττώνει την ποινή αστοχίας.
- Η κύρια μνήμη αποτελείται από ξεχωριστές μονάδες (modules).
- Κάθε μονάδα διαθέτει ABR (address buffer register) και DBR (data buffer register).
- Παρέχεται η εκτέλεση ταυτόχρονων λειτουργιών πρόσβασης σε περισσότερες από μία μονάδες.



- Τα k λιγότερο σημαντικά ψηφία της διεύθυνσης επιλέγουν μία μονάδα και τα m περισσότερο σημαντικά ψηφία της διεύθυνσης επιλέγουν μία θέση στη μονάδα αυτή.
- Με τον τρόπο αυτό, διαδοχικές διευθύνσεις εντοπίζονται σε διαδοχικές μονάδες μνήμης, με αποτέλεσμα όταν απαιτείται πρόσβαση σε διαδοχικές θέσεις μνήμης να απασχολούνται ταυτόχρονα περισσότερες από μία μονάδες μνήμης, οδηγώντας σε ταχύτερη πρόσβαση σε ένα τμήμα δεδομένων.
- Για την υλοποίηση της δομής αυτής απαιτούνται 2^k μονάδες μνήμης, ώστε να καλυφθεί το σύνολο του χώρου διευθύνσεων της μνήμης.

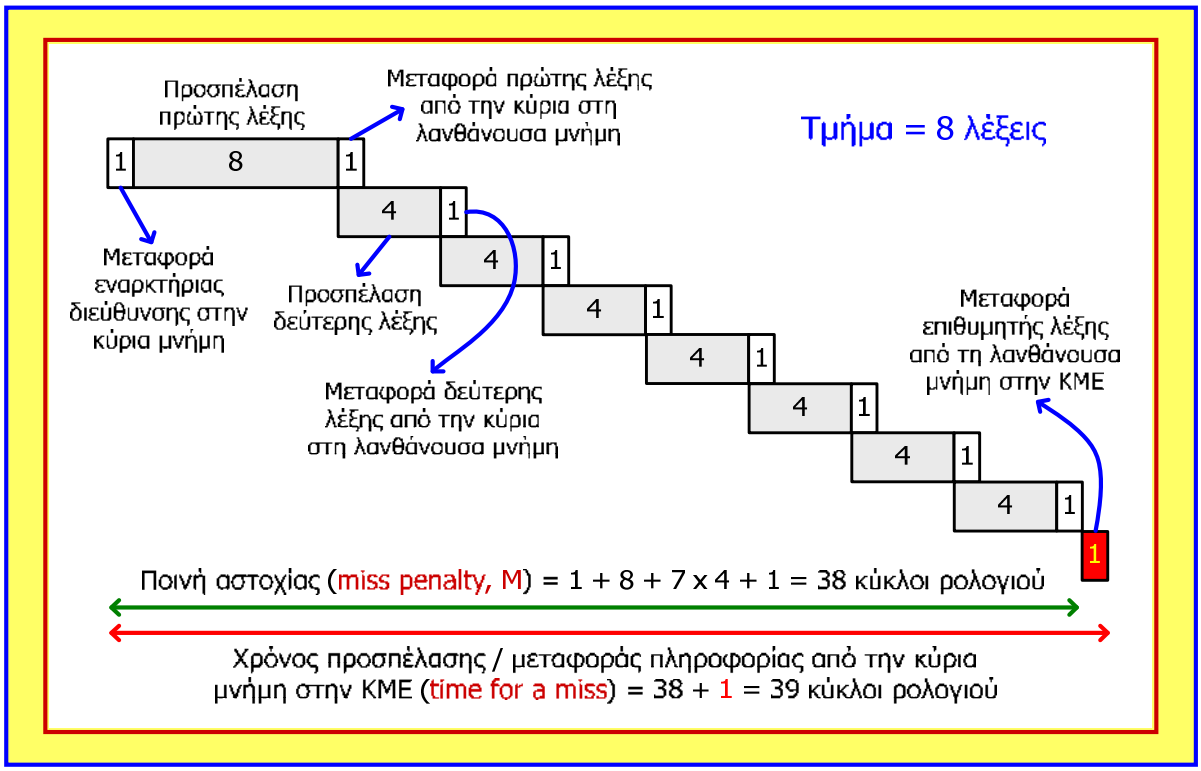
Απόδοση συστήματος μνήμης: παράδειγμα

- Σε επεξεργαστή, που χρησιμοποιεί λανθάνουσες μνήμες εντολών και δεδομένων με 8 λέξεις ανά τμήμα, όταν συμβαίνει αποτυχία ανάγνωσης, το τμήμα που περιλαμβάνει την επιθυμητή λέξη θα πρέπει να μεταφερθεί από την κύρια μνήμη στη λανθάνουσα μνήμη.
- Το σύστημα μνήμης διαθέτει τα εξής χαρακτηριστικά: η μεταφορά της εναρκτήριας διεύθυνσης από τον επεξεργαστή προς την κύρια μνήμη απαιτεί 1 κύκλο ρολογιού, η προσπέλαση της πρώτης λέξης ενός τμήματός της απαιτεί 8 κύκλους ρολογιού, αλλά η προσπέλαση καθεμιάς από τις επόμενες λέξεις του ιδίου τμήματος απαιτεί 4 κύκλους ρολογιού. Η μεταφορά μιας λέξης από την κύρια στη λανθάνουσα μνήμη απαιτεί 1 κύκλο ρολογιού, όσο και η μεταφορά από τη λανθάνουσα μνήμη στον επεξεργαστή (C).
- Χρόνος προσπέλασης και μεταφοράς τμήματος από την κύρια σε μία λανθάνουσα μνήμη (ποινή αστοχίας): 1 κύκλος για τη μεταφορά της διεύθυνσης στην κύρια μνήμη, 8 κύκλοι για προσπέλαση της πρώτης λέξης, 1 κύκλος για μεταφορά της λέξης αυτής στη λανθάνουσα μνήμη, στον ίδιο κύκλο ξεκινά η προσπέλαση της δεύτερης λέξης η οποία ολοκληρώνεται σε επιπλέον 3 κύκλους και κάθε λέξη μετά την πρώτη απαιτεί 1 κύκλο για τη μεταφορά της στη λανθάνουσα μνήμη,

$$M = 1 + 8 + 7 \times 4 + 1 = 38 \text{ κύκλοι ρολογιού}$$

- Έστω ότι 30% των εντολών ενός τυπικού προγράμματος απαιτούν πρόσβαση στη μνήμη, δηλαδή θα γίνονται 13 προσβάσεις στη μνήμη για κάθε 10 εντολές προγράμματος που εκτελούνται.

Απόδοση συστήματος μνήμης: παράδειγμα



Απόδοση συστήματος μνήμης: παράδειγμα

- Έστω ρυθμός ευστοχίας λανθάνουσας μνήμης εντολών $h_1 = 0.85$ και δεδομένων $h_2 = 0.9$. Ο μέσος χρόνος προσπέλασης του συστήματος μνήμης από την ΚΜΕ είναι:

$$T = (10/13) \times T_{IN} + (3/13) \times T_D = 0.77 \times [h \times C + (1 - h) \times (M + C)] + 0.23 \times [h \times C + (1 - h) \times (M + C)] = 0.77 \times (0.85 \times 1 + 0.15 \times 39) + 0.23 \times (0.9 \times 1 + 0.1 \times 39) = 6.26 \text{ κύκλοι ρολογιού}$$

- Θεωρούμε τώρα ότι η κύρια μνήμη διαθέτει δομή διαφύλλωσης με 4 μονάδες. Στην περίπτωση αυτή, όταν η εναρκτήρια διεύθυνση φθάνει στη μνήμη, όλες οι μονάδες ξεκινούν την προσπέλαση των δεδομένων χρησιμοποιώντας τα περισσότερο σημαντικά ψηφία της διεύθυνσης και μετά από 8 κύκλους υπάρχει μία λέξη δεδομένων στον DBR κάθε μονάδας.
- Στους επόμενους 4 κύκλους οι 4 λέξεις μεταφέρονται (μία σε κάθε κύκλο) στη λανθάνουσα μνήμη και στη διάρκεια των ίδιων κύκλων υπάρχει η επόμενη λέξη δεδομένων στον DBR κάθε μονάδας.
- Τέλος, απαιτούνται 4 κύκλοι ρολογιού για τη μεταφορά της δεύτερης σειράς λέξεων στη λανθάνουσα μνήμη.
- Συνεπώς για τη μεταφορά των 8 λέξεων ενός τμήματος από την κύρια μνήμη στη λανθάνουσα, απαιτούνται τώρα: M = 1 + 8 + 4 + 4 = 17 κύκλοι ρολογιού (δηλαδή, λιγότεροι από τους μισούς των 38 κύκλων που απαιτούνταν στην ενιαία κύρια μνήμη).

Απόδοση συστήματος μνήμης: παράδειγμα

- Ο μέσος χρόνος προσπέλασης του συστήματος μνήμης με 4 μονάδες διαφύλλωσης, γίνεται:

$$T = (10/13) \times T_{IN} + (3/13) \times T_D = 0.77 \times [h \times C + (1 - h) \times (M + C)] + 0.23 \times [h \times C + (1 - h) \times (M + C)] = 0.77 \times (0.85 \times 1 + 0.15 \times 18) + 0.23 \times (0.9 \times 1 + 0.1 \times 18) = 3.35 \text{ κύκλοι ρολογιού}$$

(περίπου υποδιπλασιάζεται σε σχέση με τη λύση της απλής μνήμης).

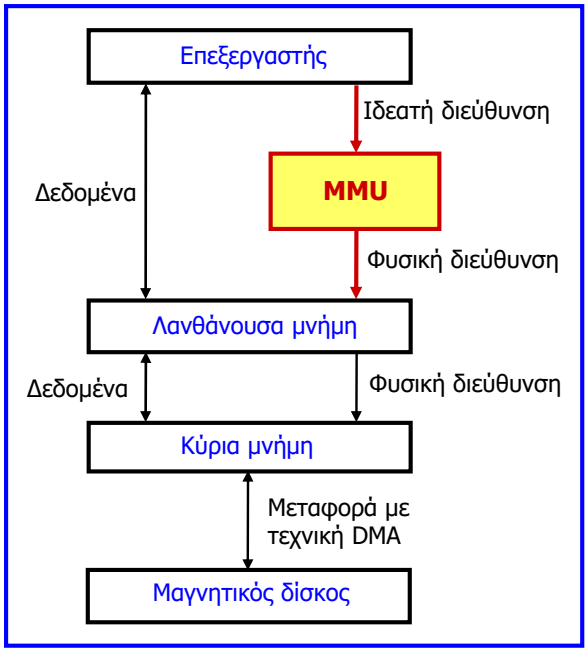
- Από ένα πλήθος μονάδων διαφύλλωσης και πάνω η λύση της δομής διαφύλλωσης δεν αυξάνει το όφελος, όσον αφορά την μείωση της ποινής αστοχίας.
- Επίσης, όσο περισσότερες είναι οι μονάδες διαφύλλωσης, τόσο πιο πολύπλοκη και φυσικά ακριβότερη είναι η μνήμη, αφού απαιτούνται πρόσθετα κυκλώματα διασύνδεσης και ελέγχου των μονάδων.

Ιδεατή μνήμη: βασικές έννοιες

- Στα περισσότερα σύγχρονα υπολογιστικά συστήματα, η φυσική κύρια μνήμη δεν είναι τόσο μεγάλη, όσο ο χώρος των διευθύνσεων μνήμης που εκδίδονται από τον επεξεργαστή.
- Για παράδειγμα, διευθύνσεις των 48 bits δημιουργούν θεωρητικά χώρο διευθύνσεων 256 Tbytes (2^{48} bytes), ενώ η φυσική κύρια μνήμη (SDRAM) ενός τυπικού Η/Υ είναι πολύ μικρότερη (π.χ. 8 Gbytes).
- Όταν ένα πρόγραμμα δεν χωρά ολόκληρο στη κύρια μνήμη, τα μέρη αυτού που δεν εκτελούνται την τρέχουσα χρονική στιγμή, αποθηκεύονται σε συσκευές δευτερεύουσας αποθήκευσης (π.χ. στο μαγνητικό δίσκο) και προσάγονται στη φυσική κύρια μνήμη όταν χρειαστεί, με τη βοήθεια του λειτουργικού συστήματος και χρήση της τεχνικής DMA.
- Οι τεχνικές οι οποίες επιτελούν τη μετακίνηση προγραμμάτων και τμημάτων δεδομένων από και προς τη φυσική κύρια μνήμη, όταν αυτά απαιτούνται για εκτέλεση, αναφέρονται ως τεχνικές ιδεατής ή εικονικής (virtual) μνήμης.
- Οι διευθύνσεις που παράγει ο επεξεργαστής για εντολές ή δεδομένα αναφέρονται ως ιδεατές (εικονικές, virtual) ή λογικές (logical) διευθύνσεις.
- Οι διευθύνσεις αυτές μεταφράζονται σε φυσικές (physical) διευθύνσεις με τη βοήθεια της μονάδας διαχείρισης μνήμης (memory management unit, MMU).

Ιδεατή μνήμη: βασικές έννοιες

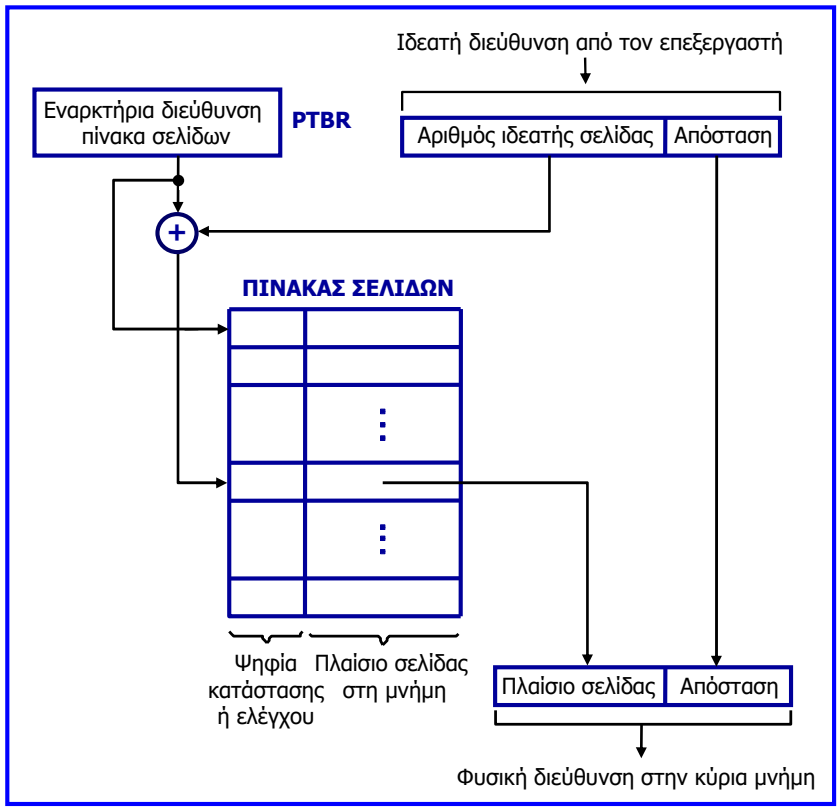
- Όταν μία ιδεατή διεύθυνση αναφέρεται σε τμήμα δεδομένων ή εντολών το οποίο την τρέχουσα χρονική στιγμή βρίσκεται στη φυσική κύρια μνήμη, τότε έχουμε άμεση προσπέλαση.
- Όταν δεν συμβαίνει αυτό, τα περιεχόμενα της αναφερόμενης διεύθυνσης θα πρέπει να προσαχθούν σε μία κατάλληλη θέση της φυσικής κύριας μνήμης για να χρησιμοποιηθούν.
- Η MMU «αναγκάζει» το λειτουργικό σύστημα να προσαγάγει τα δεδομένα από το δίσκο στη φυσική κύρια μνήμη.
- Η μεταφορά των δεδομένων μεταξύ δίσκου και κύριας μνήμης γίνεται με χρήση της τεχνικής DMA (direct memory access).



Ιδεατή μνήμη: μετάφραση διευθύνσεων

- Τα προγράμματα και τα δεδομένα αποτελούνται από σελίδες (pages) καθορισμένου μεγέθους (1K έως 16K bytes) που περιέχουν λέξεις διαδοχικών θέσεων στην κύρια μνήμη.
- Οι σελίδες είναι η βασική οντότητα πληροφορίας που μετακινείται μεταξύ κύριας μνήμης και δίσκου και πρέπει να έχουν σχετικά μεγάλο μέγεθος επειδή ο χρόνος προσπέλασης του δίσκου είναι αρκετά μεγαλύτερος (~10 ms) από τον αντίστοιχο της κύριας μνήμης (~100 ns).
- Ωστόσο, εάν το μέγεθος της σελίδας είναι πολύ μεγάλο, είναι πιθανό, σημαντικό ποσοστό της σελίδας να μη χρησιμοποιηθεί και να καταλάβει πολύτιμο χώρο στην κύρια μνήμη.
- Κάθε ιδεατή διεύθυνση που παράγεται από τον επεξεργαστή περιλαμβάνει έναν αριθμό ιδεατής σελίδας (virtual page number), που ακολουθείται από μία απόσταση (offset) που καθορίζει τη θέση μιας λέξης εντός της σελίδας.
- Η εναρκτήρια διεύθυνση κάθε σελίδας στην κύρια μνήμη και η τρέχουσα κατάσταση της (ψηφία κατάστασης ή ελέγχου) αποθηκεύονται στον πίνακα σελίδων (page table). Η περιοχή κύριας μνήμης που μπορεί να περιέχει μία σελίδα λέγεται πλαίσιο σελίδας (page frame).
- Η εναρκτήρια διεύθυνση του πίνακα σελίδων κρατείται σε έναν καταχωρητή βάσης πίνακα σελίδων (page table base register, PTBR).
- Με πρόσθεση του αριθμού ιδεατής σελίδας στο περιεχόμενο του PTBR, προκύπτει η διεύθυνση της αντίστοιχης καταχώρησης στον πίνακα σελίδων. Το περιεχόμενο της καταχώρησης αυτής παρέχει την εναρκτήρια διεύθυνση της σελίδας, εάν η σελίδα αυτή βρίσκεται στην κύρια μνήμη την τρέχουσα χρονική στιγμή.

Ιδεατή μνήμη: μετάφραση διευθύνσεων



Ιδεατή μνήμη: μετάφραση διευθύνσεων

- Τα ψηφία κατάστασης ή ελέγχου περιγράφουν την κατάσταση μιας σελίδας, όσο αυτή βρίσκεται στην κύρια μνήμη.
- Το ψηφίο εγκυρότητας (valid bit), υποδεικνύει αν η σελίδα έχει πράγματι φορτωθεί στην κύρια μνήμη και επιτρέπει στο λειτουργικό σύστημα (θέτοντάς του τιμή 0) να καταστήσει άκυρη τη σελίδα, χωρίς να την αφαιρέσει από τη μνήμη.
- Το ψηφίο τροποποίησης (dirty bit) υποδεικνύει εάν η σελίδα έχει τροποποιηθεί κατά την παραμονής της στην κύρια μνήμη, ώστε να είναι γνωστό εάν θα πρέπει να επαναποθηκευτεί στο δίσκο πριν την απομάκρυνση της από την κύρια μνήμη, ώστε να αφήσει ελεύθερο χώρο για μια άλλη σελίδα.
- Επειδή μια πλήρης υλοποίηση του αλγορίθμου αντικατάστασης LRU προσθέτει αρκετή πολυπλοκότητα, μια απλούστερη τεχνική αντικατάστασης σελίδας (σε περίπτωση που θα πρέπει να προσαχθεί από το δίσκο μία νέα σελίδα και η κύρια μνήμη είναι γεμάτη) βασίζεται σε ένα ψηφίο χρησιμοποίησης ή αναφοράς (used, reference bit), το οποίο τίθεται σε τιμή 1, όταν η αντίστοιχη σελίδα προσπελαύνεται (αναφέρεται).
- Το λειτουργικό σύστημα ανά τακτά χρονικά διαστήματα μηδενίζει το ψηφίο χρησιμοποίησης ή αναφοράς σε όλες τις καταχωρήσεις του πίνακα σελίδων, παρέχοντας έναν απλό τρόπο καθορισμού των σελίδων που δεν έχουν χρησιμοποιηθεί πρόσφατα.
- Άλλα ψηφία κατάστασης ή ελέγχου περιλαμβάνουν πληροφορίες σχετικές με δικαιώματα πρόσβασης (π.χ. read, write bits: ψηφία δικαιώματος ανάγνωσης ή εγγραφής).

Ιδεατή μνήμη: παράδειγμα μετάφρασης διευθύνσεων

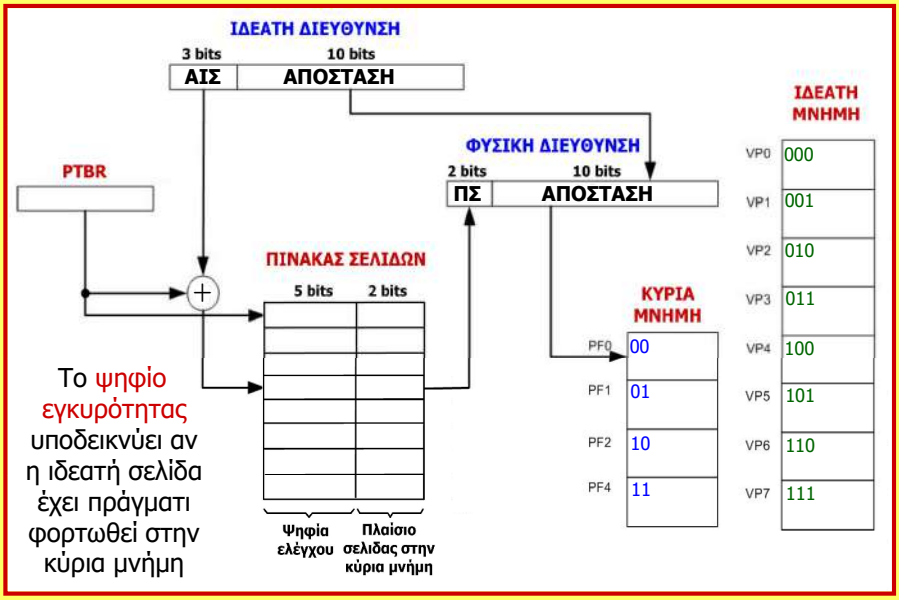
Ιδεατή μνήμη: 8K λέξεις
Κύρια μνήμη: 4K λέξεις
Σελίδα: 1K λέξεις
Ψηφία ελέγχου: 5

Ιδεατή μνήμη: 8 σελίδες
Κύρια μνήμη: 4 σελίδες

$8K = 2^3 \cdot 2^{10} = 2^{13} \Rightarrow$
13 bits ιδεατή διεύθυνση

$4K = 2^2 \cdot 2^{10} = 2^{12} \Rightarrow$
12 bits φυσική διεύθυνση

$1K = 2^{10} \Rightarrow$ 10 bits για τον καθορισμό της θέσης μιας λέξης εντός της σελίδας (απόσταση)



Η επιλογή της καταχώρησης στον πίνακα σελίδων γίνεται με τη διεύθυνση που προκύπτει από την πρόσθεση του περιεχομένου του PTBR και του πεδίου ΑΙΣ της ιδεατής διεύθυνσης

Ιδεατή μνήμη: παράδειγμα μετάφρασης διευθύνσεων

Ιδεατή διεύθυνση = 4000_{10} = 011 1110100000, ΑΙΣ = 011 = 3_{10}, Απόσταση = 1110100000 = $= 928_{10}$

Εναρκτήρια φυσική διεύθυνση πίνακα σελίδων (PTBR) = 2000_{10} = 011111010000

Διεύθυνση καταχώρησης πίνακα σελίδων = PTBR + ΑΙΣ = $2000_{10} + 3_{10} = 2003_{10} =$
$= 011111010011$

Φυσική διεύθυνση = M[PTBR + ΑΙΣ] $\times 2^n$ + απόσταση, n = πλήθος ψηφίων απόστασης = 10

$= M[2003_{10}] \times 2^{10} + 928_{10} =$

$= 110000000000 + 1110100000 =$

$= 11\ 1110100000$

Πλαίσιο σελίδας = 11 = 2_{10}

Απόσταση = 1110100000 = 928_{10}

	Ψηφία ελέγχου	Πλαίσιο σελίδας
2000_{10}		01
		11
		00
2003_{10}	1XXXX	11
		10
		01
		00
		10

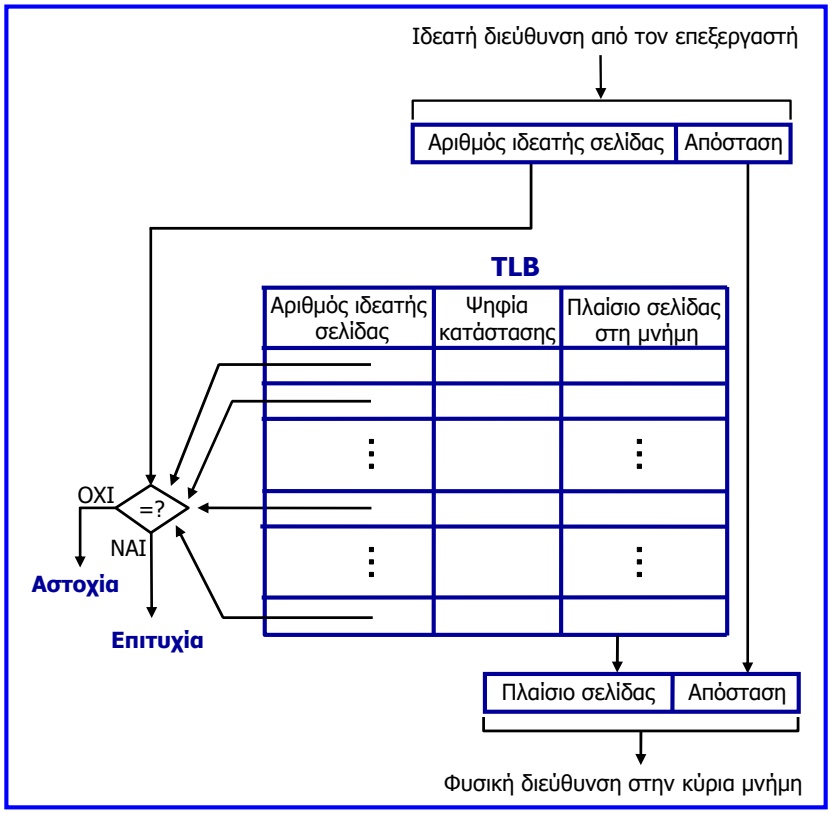
ΠΙΝΑΚΑΣ ΣΕΛΙΔΩΝ

ψηφίο εγκυρότητας

Ιδεατή μνήμη: TLB

- Κάθε φορά που ένα πρόγραμμα διενεργεί αναφορά στη μνήμη, απαιτούνται δύο προσπελάσεις μνήμης: προσπέλαση για τη λήψη της φυσικής διεύθυνσης από τον πίνακα σελίδων που βρίσκεται στη μνήμη και προσπέλαση για τη μεταφορά των δεδομένων.
- Για να μειωθεί ο αριθμός προσπελάσεων της MMU στον πίνακα σελίδων, χρησιμοποιείται μια λανθάνουσα μνήμη, η οποία περιέχει ένα μέρος του πίνακα σελίδων με τις πιο πρόσφατα προσπελασθείσες σελίδες.
- Η μνήμη αυτή αναφέρεται ως TLB (translation lookaside buffer, παράπλευρος απομονωτής μετάφρασης), αν και περιγράφεται ακριβέστερα με τους όρους λανθάνουσα μνήμη μετάφρασης ή λανθάνουσα μνήμη για τον πίνακα σελίδων.
- Ο TLB υλοποιείται μαζί με την MMU στο ολοκληρωμένο κύκλωμα του επεξεργαστή.
- Εκτός από καταχωρήσεις του πίνακα σελίδων, δηλαδή για κάθε περιλαβανόμενη ιδεατή σελίδα, τα ψηφία κατάστασης ή ελέγχου και το πλαίσιο σελίδας στη μνήμη, ο TLB περιλαμβάνει και τους αντίστοιχους αριθμούς των ιδεατών σελίδων ως ετικέτα (tag).
- Όταν γίνεται αναφορά σε μια ιδεατή διεύθυνση, η MMU αναζητά το πλαίσιο σελίδας της αναφερόμενης σελίδας στον TLB και αν αυτό υπάρχει στον TLB (επιτυχία, hit), τότε παράγεται άμεσα η φυσική διεύθυνση από την MMU.
- Εάν υπάρξει αστοχία (miss) στον TLB, τότε το απαιτούμενο πλαίσιο σελίδας αναζητείται στον πίνακα σελίδων και κατόπιν ενημερώνεται ο TLB και παράγεται η φυσική διεύθυνση.

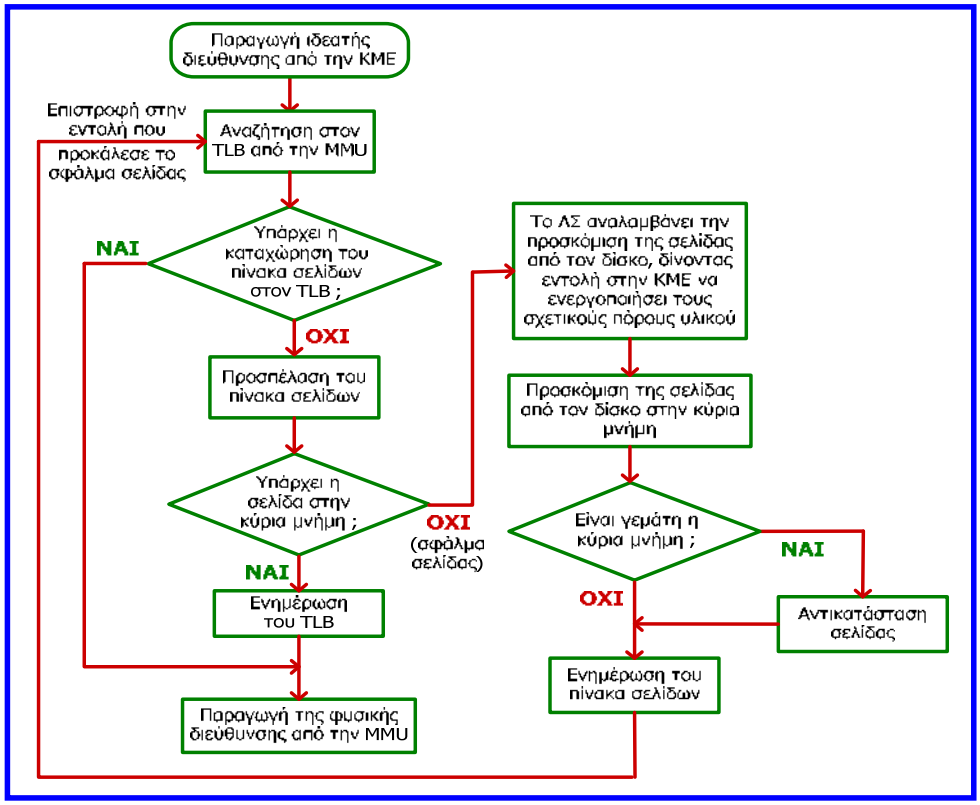
Ιδεατή μνήμη: TLB



Ιδεατή μνήμη: σφάλμα και αντικατάσταση σελίδας

- Εάν η ζητούμενη σελίδα δεν βρίσκεται στην κύρια μνήμη, τότε έχει συμβεί σφάλμα σελίδας (page fault) και το λειτουργικό σύστημα (ΛΣ) αναλαμβάνει να προσκομίσει τη ζητούμενη σελίδα από το δίσκο, να ενημερώσει τον πίνακα σελίδων και να επαναλάβει την εντολή που δημιούργησε το σφάλμα.
- Για να προσκομιστεί η ζητούμενη σελίδα από το δίσκο, απαιτείται η γνώση της θέσης της σε αυτόν και συνήθως η πληροφορία αυτή κρατείται σε μία ειδική δομή δεδομένων.
- Όταν ο χώρος που καταλαμβάνεται από μία σελίδα στην κύρια μνήμη πρέπει να ελευθερωθεί για να καταληφθεί από άλλη σελίδα που θα μεταφερθεί από το δίσκο και το περιεχόμενο της υπό αντικατάσταση σελίδας έχει αλλάξει (π.χ. λόγω εγγραφής από την ΚΜΕ), θα πρέπει το περιεχόμενό της να αποθηκευτεί στο δίσκο.
- Επομένως, το ΛΣ πρέπει να γνωρίζει σε ποια θέση του δίσκου θα αποθηκευτούν τα περιεχόμενα της σελίδας, δηλαδή πρέπει στην ειδική δομή δεδομένων να διατηρείται η θέση των σελίδων στο δίσκο, ακόμη και για τις σελίδες που βρίσκονται στην κύρια μνήμη.
- Το ΛΣ παρέχει έναν πίνακα σελίδων για κάθε πρόγραμμα και αλλάζοντας τα περιεχόμενα του PTBR, μπορεί να μεταβαίνει μεταξύ των πινάκων σελίδων διαφορετικών προγραμμάτων.
- Όταν ένα πρόγραμμα που εκτελείται παράγει μία ιδεατή διεύθυνση, το ΛΣ αρχικά εξετάζει τα ψηφία ελέγχου του πίνακα σελίδων για να διαπιστώσει εάν βρίσκεται η ζητούμενη σελίδα στην κύρια μνήμη (εγκυρότητα), καθώς και τα δικαιώματα πρόσβασης σε αυτή.

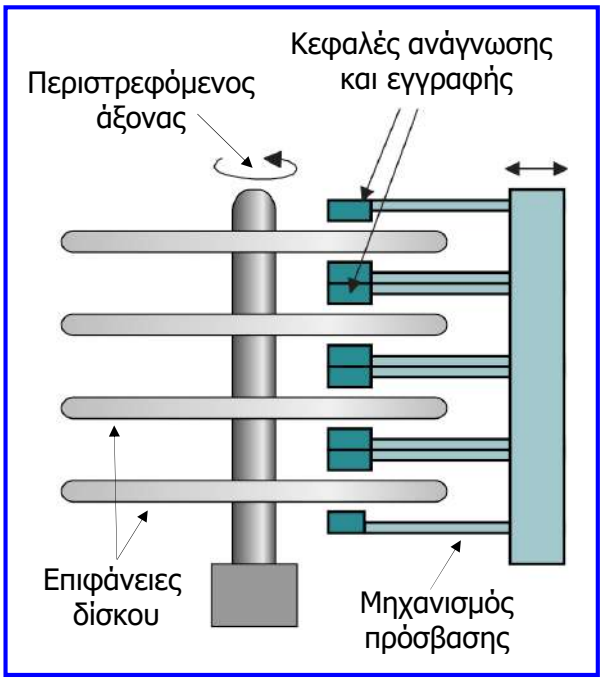
Ιδεατή μνήμη: διάγραμμα μετάφρασης διευθύνσεων



Μαγνητικός σκληρός δίσκος

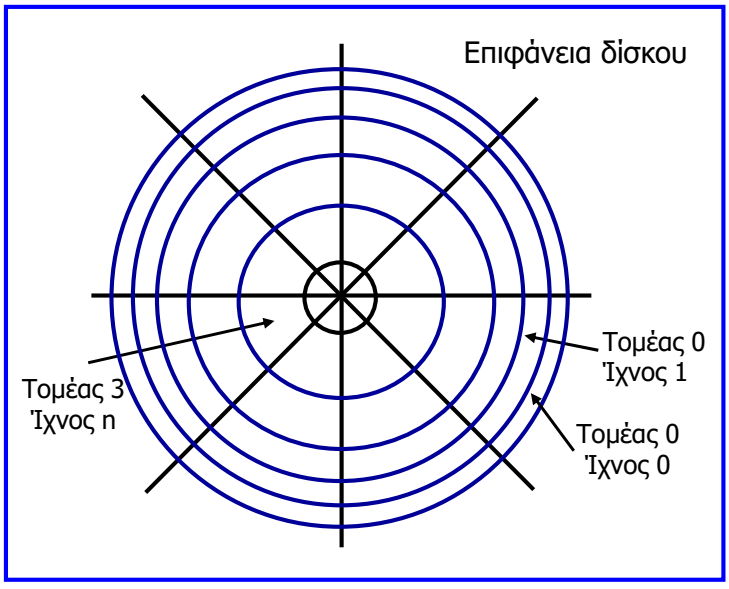
Το σύστημα του σκληρού δίσκου αποτελείται από 3 υποσυστήματα:

- Επιφάνειες των δίσκων στις οποίες εναποτίθεται μαγνητικό υλικό.
- Ηλεκτρομηχανικός εξοπλισμός (οδηγός δίσκου, disk drive) που περιστρέφει τους δίσκους και μετακινεί ακτινικά τις κεφαλές ανάγνωσης και εγγραφής
- Ηλεκτρονικό κύκλωμα (ελεγκτής δίσκου, disk controller) που ελέγχει τη λειτουργία του συστήματος και συνήθως είναι ενσωματωμένο στην ίδια συσκευασία.



Μαγνητικός σκληρός δίσκος

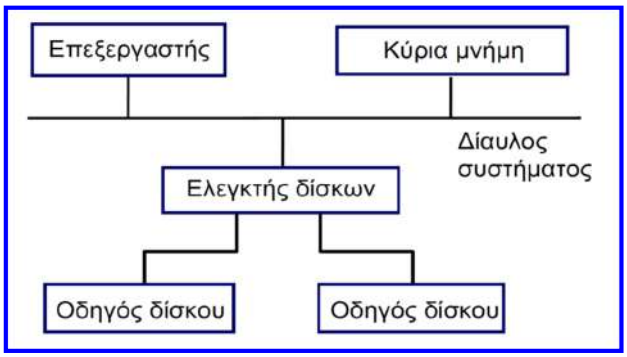
- Κάθε επιφάνεια διαιρείται σε ομόκεντρα ίχνη (tracks) και κάθε ίχνος διαμερίζεται σε τομείς (sectors).
- Η προσπέλαση των δεδομένων γίνεται με καθορισμό του αριθμού επιφανείας, ίχνους και τομέα.
- Τα δεδομένα αποθηκεύονται σειριακά σε κάθε ίχνος και κάθε τομέας έχει συνήθως χωρητικότητα 512 bytes.
- Τα δεδομένα συνοδεύονται από επικεφαλίδα τομέα (sector header) για την ανεύρεση του επιθυμητού τομέα στο επιλεγμένο ίχνος και καταλήγουν με κώδικα διόρθωσης σφαλμάτων.



Χαρακτηριστικά μαγνητικού σκληρού δίσκου

- Η χρονική καθυστέρηση μεταξύ της παραλαβής μίας διεύθυνσης (επιφάνεια, ίχνος, τομέας) και της έναρξης μεταφοράς δεδομένων αναφέρεται ως χρόνος προσπέλασης δίσκου (disk access time) και περιλαμβάνει δύο συνιστώσες:
  - ✓ Χρόνος ανίχνευσης (seek time) που απαιτείται για τη μετακίνηση της κεφαλής ανάγνωσης/εγγραφής στο επιθυμητό ίχνος (μέσος χρόνος ανίχνευσης ~ 6 ms).
  - ✓ Καθυστέρηση περιστροφής (rotational delay) ή καθυστέρηση μεταφοράς (latency) που απαιτείται για το πέρασμα της αρχής του επιθυμητού τομέα κάτω από την κεφαλή ανάγνωσης / εγγραφής. Μέση καθυστέρηση περιστροφής = χρόνος μισής περιστροφής, δηλαδή για ταχύτητα 10,000 rpm (περιστροφές ανά λεπτό), είναι 0.5 x 60 / 10,000 = 3 ms.
- Χαρακτηριστικά ενός τυπικού δίσκου διαμέτρου 2.5 ιντσών: 28 επιφάνειες, 31,330 ίχνη ανά επιφάνεια, 765 τομείς ανά ίχνος, 512 bytes ανά τομέα.
- Συνολική χωρητικότητα τυπικού δίσκου: 28 x 31,330 x 765 x 512
  = 343,597,363,200 bytes ($\div 2^{30}$) = 320 GB.
- Ο ρυθμός μεταφοράς δεδομένων ενός ίχνους προς τον απομονωτή δεδομένων του ελεγκτή δίσκου, είναι ο λόγος του μεγέθους ενός ίχνους προς τον χρόνο μιας πλήρους περιστροφής (765 τομείς x 512 bytes ÷ 6 ms = 65,280,000 bytes/s ($\div 10^6$) = 65.28 Mbytes/s).

Ελεγκτής μαγνητικού σκληρού δίσκου



- Ο ελεγκτής δίσκου χρησιμοποιεί τεχνική DMA για μεταφορά των δεδομένων μεταξύ δίσκου και κύριας μνήμης.
- Βασικές λειτουργίες του ελεγκτή δίσκου:
  - ✓ Ανίχνευση (seek): δίνει εντολή στον οδηγό του δίσκου για την μετακίνηση της κεφαλής ανάγνωσης/εγγραφής στην επιθυμητή θέση.
  - ✓ Σειριακή ανάγνωση δεδομένων από το δίσκο και αποθήκευση στον απομονωτή δεδομένων του, για μεταφορά στην κύρια μνήμη.
  - ✓ Εγγραφή δεδομένων στο δίσκο.
  - ✓ Έλεγχος σφαλμάτων (με χρήση κώδικα διόρθωσης σφαλμάτων).

Βασικές έννοιες

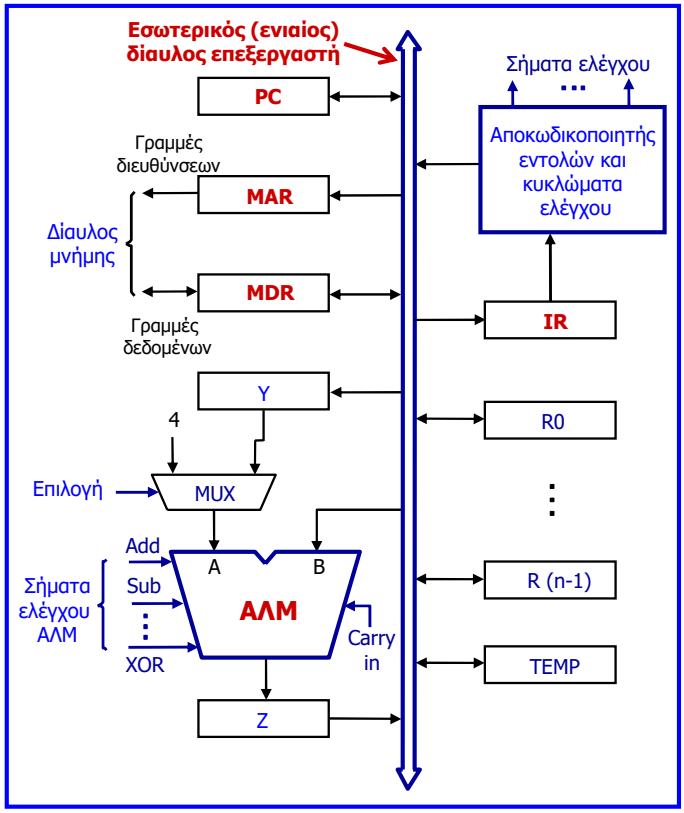
- Η μονάδα επεξεργασίας ενός υπολογιστικού συστήματος εκτελεί εντολές μηχανής και συγχρονίζει τις δραστηριότητες των υπόλοιπων μονάδων.
- Πολλά σύγχρονα υπολογιστικά συστήματα διαθέτουν πολλαπλές μονάδες επεξεργασίας.
- Για την εκτέλεση ενός προγράμματος, ο επεξεργαστής ανακτά (προσκομίζει) μία εντολή κάθε φορά και εκτελεί τις λειτουργίες που καθορίζονται από αυτή.
- Ο επεξεργαστής παρακολουθεί τη διεύθυνση της θέσης μνήμης, η οποία εμπεριέχει την επόμενη εντολή προς ανάκτηση, χρησιμοποιώντας τον μετρητή προγράμματος (PC).
- Το περιεχόμενο του PC ενημερώνεται με την ανάκτηση μίας εντολής, ώστε να δείχνει την επόμενη εντολή. Μία εντολή διακλάδωσης μπορεί να φορτώσει στον PC διαφορετική τιμή.
- Για να εκτελέσει μία εντολή ο επεξεργαστής επιτελεί τα εξής βασικά βήματα:
  - ✓ Μεταφέρει στον καταχωρητή εντολής (IR) το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση βρίσκεται στον PC.
  - ✓ Αυξάνει το περιεχόμενο του PC κατά 4 (υποθέτουμε μνήμη διευθυνσιοδοτούμενη κατά byte και εντολές των 32 bits).

  } Φάση ανάκτησης (ή προσκόμισης)
  - ✓ Εκτελεί τις λειτουργίες που καθορίζονται από την εντολή που βρίσκεται στον IR.

  } Φάση εκτέλεσης

Οργάνωση ΚΜΕ με έναν εσωτερικό δίαυλο

- Καταχωρητές, ΑΛΜ & εσωτερικός δίαυλος: μονοπάτι δεδομένων (datapath).
- Σύνολο n καταχωρητών Ri γενικής χρήσης, ωστόσο μερικοί από αυτούς μπορούν να έχουν ειδική χρήση (π.χ. SP, LR).
- ΑΛΜ (ALU) για την εκτέλεση αριθμητικών και λογικών πράξεων.
- Καταχωρητές MAR και MDR για την επικοινωνία του εσωτερικού διαύλου με το δίαυλο της μνήμης.
- Κυκλώματα αποκωδικοποίησης εντολών και ελέγχου που λαμβάνουν είσοδο από τον IR και παράγουν σήματα ελέγχου, ώστε να υλοποιηθούν οι λειτουργίες που καθορίζονται από την εντολή που είναι φορτωμένη στον IR.
- Καταχωρητές Y, Z, TEMP για προσωρινή αποθήκευση αποτελεσμάτων (μη ορατοί στον προγραμματιστή).



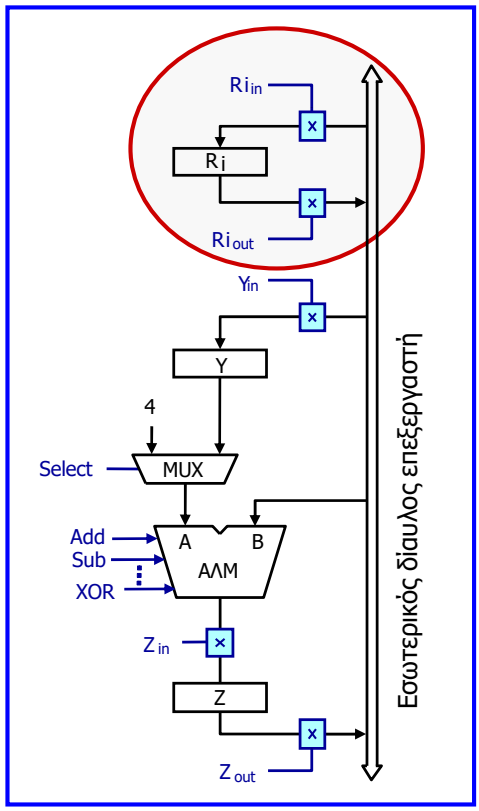
Εκτέλεση εντολών στην ΚΜΕ

Εκτός μερικών εξαιρέσεων, η εκτέλεση μίας εντολής περιλαμβάνει μία ή περισσότερες από τις ακόλουθες ενέργειες:

- Μεταφορά μίας λέξης δεδομένων από έναν καταχωρητή του επεξεργαστή σε κάποιον άλλο.
- Εκτέλεση μίας αριθμητικής ή λογικής πράξης και αποθήκευση του αποτελέσματος σε έναν καταχωρητή του επεξεργαστή.
- Ανάκτηση (ανάγνωση) των περιεχομένων μίας συγκεκριμένης θέσης μνήμης και φόρτωση τους σε έναν καταχωρητή του επεξεργαστή.
- Αποθήκευση (εγγραφή) μίας λέξης δεδομένων από έναν καταχωρητή του επεξεργαστή στη μνήμη.

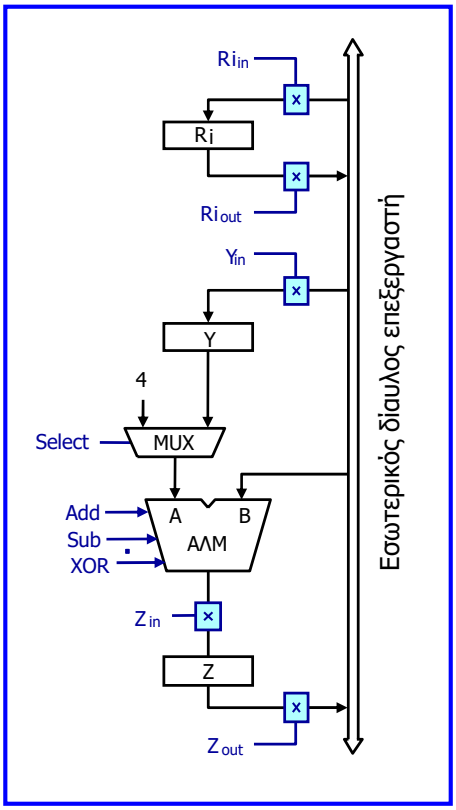
Μεταφορές δεδομένων μεταξύ καταχωρητών

- Η είσοδος και η έξοδος ενός καταχωρητή συνδέεται στον εσωτερικό δίαυλο μέσω κατάλληλων διακοπτών, οι οποίοι ελέγχονται μέσω δύο σημάτων ελέγχου (R_{out}, R_{in}) που καθορίζουν την κατεύθυνση ροής των δεδομένων μεταξύ καταχωρητή και διαύλου.
- Η μεταφορά δεδομένων μεταξύ καταχωρητών γίνεται διαμέσου του εσωτερικού διαύλου.
- Όλες οι λειτουργίες και οι μεταφορές δεδομένων εντός του επεξεργαστή λαμβάνουν χώρα σε χρονικές περιόδους που καθορίζονται από το ρολόι του επεξεργαστή (processor clock).
- Οι καταχωρητές της ΚΜΕ αποτελούνται από ακμοπυροδοτούμενα φλιπ-φλοπ.
- Το σήμα R_{out} του καταχωρητή αφετηρίας και το σήμα R_{in} του καταχωρητή προορισμού ενεργοποιούνται από τη μονάδα ελέγχου στην ανερχόμενη ακμή ρολογιού και στην επόμενη ενεργή ακμή ρολογιού διενεργείται η μεταφορά δεδομένων, οπότε τα σήματα ελέγχου απενεργοποιούνται.

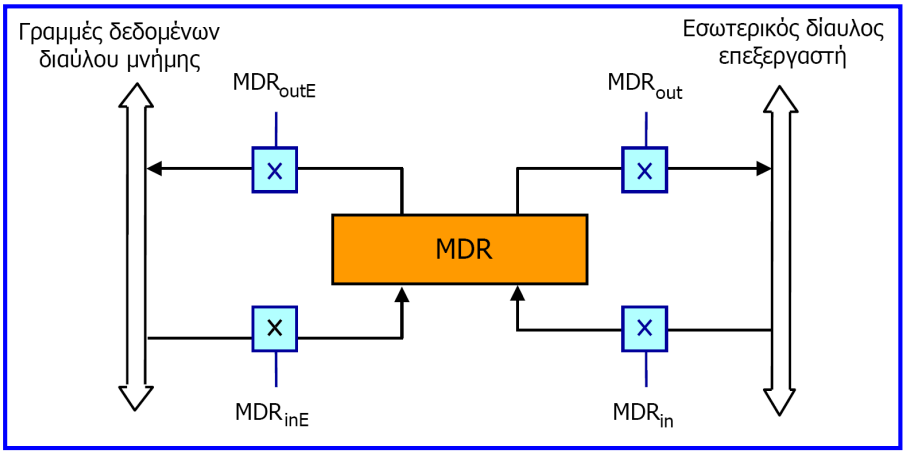


Εκτέλεση αριθμητικών ή λογικών πράξεων

- Η ΑΛΜ είναι συνδυαστικό κύκλωμα που δε διαθέτει εσωτερικές μονάδες αποθήκευσης και εκτελεί αριθμητικές και λογικές πράξεις μεταξύ δύο ορισμάτων A και B που εφαρμόζονται στην είσοδό της.
- Το αποτέλεσμα που παράγεται αποθηκεύεται προσωρινά στον καταχωρητή Z.
- Παράδειγμα: Ακολουθία λειτουργιών (βημάτων ελέγχου) για την πρόσθεση των περιεχομένων των R1 και R2 και την αποθήκευση του αποτελέσματος στον R3:
  1. $R1_{out}$, Y_{in}: ενεργά στη διάρκεια ενός κύκλου ρολογιού που αντιστοιχεί στο 1ο βήμα.
  2. $R2_{out}$, SelectY, Add, Z_{in}: ενεργά κατά το 2ο βήμα.
  3. Z_{out}, $R3_{in}$: ενεργά κατά το 3ο βήμα.
- Η μεταφορά του περιεχομένου του Z στον καταχωρητή προορισμού R3 δεν μπορεί να λάβει χώρα κατά τη διάρκεια του βήματος 2, λόγω του ότι μόνο μία έξοδος καταχωρητή μπορεί να βρίσκεται συνδεδεμένη στο δίαυλο κατά τη διάρκεια ενός κύκλου ρολογιού.



Λειτουργίες ανάγνωσης και εγγραφής μνήμης

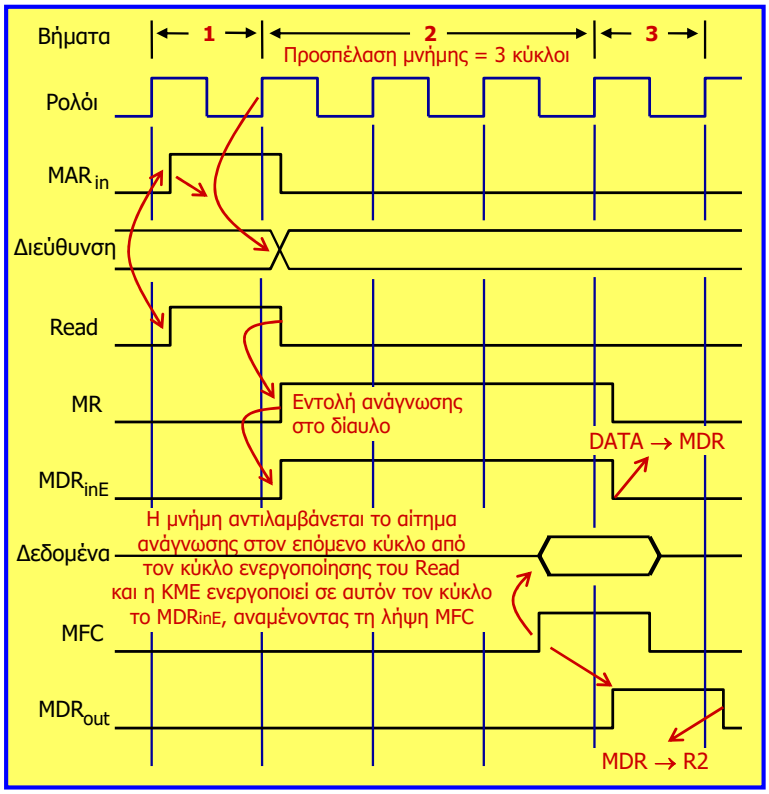


- Η επικοινωνία του επεξεργαστή με τη μνήμη γίνεται μέσω των καταχωρητών MAR και MDR και των κατάλληλων σημάτων ελέγχου από τη μονάδα ελέγχου.
- Ο επεξεργαστής μεταφέρει την επιθυμητή διεύθυνση στον MAR, η έξοδος του οποίου συνδέεται στις γραμμές διευθύνσεων του διαύλου μνήμης.
- Ο MDR αποθηκεύει τα δεδομένα που διαβάζονται από τη μνήμη ή αποθηκεύονται στη μνήμη και διαθέτει 4 σήματα ελέγχου, που ελέγχουν τις συνδέσεις του στον εσωτερικό δίαυλο και στις γραμμές δεδομένων του δίαυλου μνήμης.

Λειτουργίες ανάγνωσης και εγγραφής μνήμης

- Βήματα (λειτουργίες) ανάγνωσης μνήμης [MOV (R1), R2]:
  1. $R1_{out}$, MAR_{in}, Read
  2. MDR_{inE}, WMFC
  3. $R2_{in}$, MDR_{out}

  Παράδειγμα
- Το WMFC είναι σήμα που αναγκάζει τη μονάδα ελέγχου σε αναμονή για άφιξη του MFC (memory function completed).
- Τα δεδομένα φορτώνονται στον MDR στο τέλος του παλμού ρολογιού κατά τον οποίο λαμβάνεται το MFC από τη μνήμη.
- Βήματα εγγραφής μνήμης [MOV R2, (R1)]: 1. $R1_{out}$, MAR_{in}
  2. $R2_{out}$, MDR_{in}, Write
  3. MDR_{outE}, WMFC
- Τα σήματα MDR_{inE} και MDR_{outE} παραλείπονται, αφού συσχετίζονται με τα Read και Write.



Εκτέλεση πλήρους εντολής

- Οι λειτουργίες (βήματα ελέγχου) που απαιτούνται για την εκτέλεση πλήρους εντολής είναι:

i. Ανάκτηση εντολής: βήματα 1 έως 3.

ii. Ανάκτηση του 1ου ορίσματος που βρίσκεται στη θέση μνήμης που υποδεικνύεται στον R3: βήμα 4.

iii. Εκτέλεση πρόσθεσης: στο βήμα 5 το περιεχόμενο του R1 μεταφέρεται στον Y και στο 6 εκτελείται η πρόσθεση.

iv. Μεταφορά του αποτελέσματος στον R1 στο βήμα 7

Βήμα	Λειτουργίες	ADD (R3), R1
1	PC_{out} , MAR_{in}, Read, Select4, Add, Z_{in}	
2	Z_{out} , PC_{in}, Y_{in}, WMFC	Το WMFC ενεργοποιείται 1 κύκλο μετά το Read (αλλά πάντα πριν το MDR_{out}), αφού η μνήμη αντιλαμβάνεται το αίτημα ανάγνωσης στον επόμενο κύκλο. Έτσι, στους κύκλους 2 και 5 μπορούν να γίνουν πρόσθετες ενέργειες
3	MDR_{out} , IR_{in}	
4	$R3_{out}$, MAR_{in}, Read	
5	$R1_{out}$, Y_{in}, WMFC	
6	MDR_{out} , SelectY, Add, Z_{in}	
7	Z_{out} , $R1_{in}$, End	

- Κατά τη διάρκεια του βήματος 2, ο PC προσαυξημένος κατά 4, μεταφέρεται στον καταχωρητή Y, ενέργεια που είναι περιττή για την εκτέλεση της εντολής Add.
- Σε εντολές διακλάδωσης, όμως, η ενημερωμένη τιμή του PC απαιτείται για τον υπολογισμό της διεύθυνσης προορισμού διακλάδωσης, συνεπώς για να επιταχυνθεί η εκτέλεση πιθανής εντολής διακλάδωσης, η τιμή αυτή αντιγράφεται στον καταχωρητή Y.
- Το σήμα τερματισμού (End) από τη μονάδα ελέγχου προκαλεί την έναρξη ενός νέου κύκλου ανάκτησης εντολής (με επιστροφή στο βήμα 1).

Εκτέλεση εντολής διακλάδωσης

- Μετά την ανάκτηση μίας εντολής διακλάδωσης χωρίς συνθήκη, δηλαδή μετά τα βήματα 1, 2 και 3, η μετατόπιση (offset) εξάγεται από τον IR μέσω της μονάδας αποκωδικοποίησης και προστίθεται με την ενημερωμένη τιμή του PC που ήδη από το βήμα 2 βρίσκεται στον καταχωρητή Y.

Βήμα	Λειτουργίες	BRANCH offset
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}	
2	Z_{out}, PC_{in}, Y_{in}, WMFC	
3	MDR_{out}, IR_{in}	
4	Offset-field-of-IR_{out}, SelectY, Add, Z_{in}	
5	Z_{out}, PC_{in}, End	

- Η πρόσθεση γίνεται στο βήμα 4 και το αποτέλεσμά της (διεύθυνση προορισμού διακλάδωσης), φορτώνεται στον PC στο βήμα 5.
- Η μετατόπιση, η οποία χρησιμοποιείται σε μία εντολή διακλάδωσης, αποτελεί τη διαφορά μεταξύ της διεύθυνσης προορισμού διακλάδωσης και της τιμής του PC (επόμενη εντολή).
- Στην περίπτωση εντολής διακλάδωσης υπό συνθήκη (π.χ. BRANCH<0), αρχικά ελέγχουμε την κατάσταση της κατάλληλης σημαίας κωδικού συνθήκης (conditional code) πριν υπολογίσουμε και φορτώσουμε τη νέα τιμή στον PC:

  Βήμα 4: If N=0 End (εάν N=0, ανάκτηση επόμενης εντολής με επιστροφή στο βήμα 1).

  Βήμα 5: Offset-filed-of-IR_{out}, SelectY, Add, Z_{in} (εάν N=1, μεταφορά του offset στην ΑΛΜ, πρόσθεσή του στον Y που περιέχει την τιμή του PC και αποθήκευση αποτελέσματος στον Z).

  Βήμα 6: Z_{out}, PC_{in}, End (μεταφορά της τιμής του Z στον PC & ανάκτηση της κατάλληλης εντολής).
- Εναλλακτικός τρόπος για τα βήματα 4 & 5: Offset-field-of-IRout, SelectY, Add, Z_{in}, if N=0 End.

Εκτέλεση εντολής κλήσης υπορουτίνας

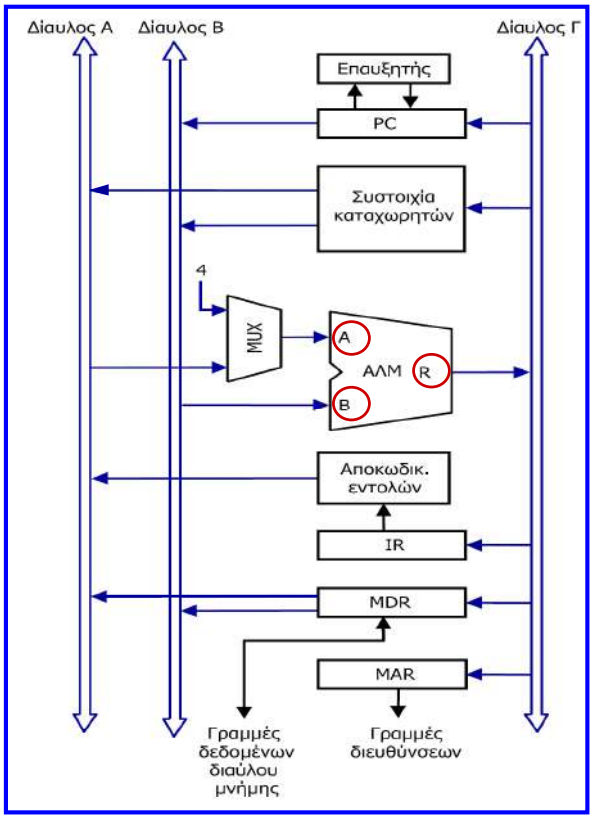
- Η εντολή κλήσης υπορουτίνας αποτελείται από 2 λέξεις, από τις οποίες η πρώτη περιλαμβάνει τον κωδικό λειτουργίας (opcode) και η δεύτερη περιλαμβάνει τη διεύθυνση της υπορουτίνας (address).
- Κατά την εκτέλεση της εντολής, ο έλεγχος μεταφέρεται στη διεύθυνση Address [PC ← address], η διεύθυνση της επόμενης εντολής (επιστροφής) μεταφέρεται στη θέση μνήμης της οποίας η διεύθυνση είναι αποθηκευμένη στον SP [Mem(SP) ← PC], και το περιεχόμενο του SP μειώνεται κατά 4 [SP ← SP−4].

Βήμα	Λειτουργίες	CALL address
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}	
2	Z_{out}, PC_{in}, Y_{in}, WMFC	
3	MDR_{out}, IR_{in}	
4	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}, WMFC	
5	MDR_{out}, PC_{in}	
6	Z_{out}, MDR_{in}	
7	SP_{out}, MAR_{in}, Write, Select4, Sub, Z_{in}	
8	Z_{out}, SP_{in}, WMFC, End	

- Μετά την εκτέλεση του βήματος 4, ο καταχωρητής Z περιέχει τη διεύθυνση επιστροφής, ενώ μετά την εκτέλεση του βήματος 5 ο PC περιέχει τη διεύθυνση της υπορουτίνας (address).
- Μετά την εκτέλεση του βήματος 6, η διεύθυνση επιστροφής αποθηκεύεται στον MDR και μετά την εκτέλεση των βημάτων 7 και 8 η διεύθυνση της επόμενης εντολής (περιεχόμενο PC) μεταφέρεται στη θέση μνήμης με διεύθυνση που βρίσκεται στον SP και SP ← SP−4.

Οργάνωση ΚΜΕ πολλαπλών διαύλων

- Μειονέκτημα της οργάνωσης ενός διαύλου αποτελεί το γεγονός ότι μόνο ένα όρισμα δεδομένων μπορεί να μεταφερθεί διαμέσου του διαύλου σε έναν κύκλο ρολογιού.
- Η οργάνωση πολλαπλών διαύλων μειώνει το πλήθος βημάτων που απαιτούνται για την ολοκλήρωση μίας εντολής και συνεπώς το πλήθος των κύκλων ρολογιού.
- Οι καταχωρητές γενικής χρήσης υλοποιούνται σε συστοιχία (register file) με 3 θύρες που επιτρέπουν ταυτόχρονη προσπέλαση σε δύο καταχωρητές και φόρτωση τρίτου στον ίδιο κύκλο (δεν χρειάζονται οι καταχωρητές Y, Z, TEMP).
- Ο PC αυξάνεται κατά 4 χωρίς τη χρήση της ΑΛΜ, λόγω της προσθήκης ειδικού κυκλώματος επαυξητή.



Εκτέλεση εντολών σε ΚΜΕ πολλαπλών διαύλων

Ανάκτηση εντολής

Βήμα	Λειτουργίες	ADD R4, R5, R6
1	PC_{out}, R=B, MAR_{in}, Read, IncPC	
2	WMFC	Εκτέλεση σε ένα μόνο βήμα
3	MDR_{outB}, R=B, IR_{in}	
4	$R4_{outA}$, $R5_{outB}$, SelectA, Add, $R6_{in}$, End	

- Στο βήμα 1, το περιεχόμενο του PC διοχετεύεται μέσω του διαύλου Β στην ΑΛΜ, περνά αναλλοίωτο από την ΑΛΜ χρησιμοποιώντας το σήμα ελέγχου R=B και φορτώνεται στον MAR μέσω του διαύλου Γ, ώστε να ξεκινήσει μία διαδικασία ανάγνωσης εντολής από τη μνήμη (σήμα ελέγχου Read). Στο ίδιο βήμα ο PC αυξάνεται κατά 4 (μέσω του επαυξητή).
- Η επαυξημένη τιμή του PC φορτώνεται στον MAR στο τέλος του 1ου κύκλου ρολογιού και δεν επηρεάζει το αρχικό περιεχόμενό του.
- Στο βήμα 2, ο επεξεργαστής αναμένει το σήμα MFC, φορτώνει την εντολή που παρέλαβε στον MDR και κατόπιν στο βήμα 3 τη μεταφέρει στον IR, ακολουθώντας τη διαδρομή «δίαυλος Β – ΑΛΜ – δίαυλος Γ».
- Η φάση εκτέλεσης απαιτεί μόνο το βήμα 4: μεταφορά περιεχομένου του R4 μέσω διαύλου Α στην ΑΛΜ, μεταφορά περιεχομένου του R5 μέσω διαύλου Β στην ΑΛΜ, πρόσθεσή τους και αποθήκευση του αποτελέσματος στον R6.

Εκτέλεση εντολών σε ΚΜΕ πολλαπλών διαύλων

Βήμα	Λειτουργίες	ADD (R2), R3
1	PC_{out}, R=B, MAR_{in}, Read, IncPC	Ανάκτηση εντολής.
2	WMFC	
3	MDR_{outB}, R=B, IR_{in}	
4	$R2_{outB}$, R=B, MAR_{in}, Read	Μεταφορά του περιεχομένου του R2 μέσω του διαύλου Β και της ΑΛΜ στον MAR.
5	WMFC	Περίοδος αδράνειας για αναμονή δεδομένων.
6	MDR_{outB}, $R3_{outA}$, SelectA, Add, $R3_{in}$, End	Μεταφορά δεδομένων μέσω διαύλου Β στην ΑΛΜ, πρόσθεση με το περιεχόμενο του R3 που μεταφέρεται μέσω του διαύλου Α στην ΑΛΜ και αποθήκευση αποτελέσματος στον καταχωρητή R3 μέσω του διαύλου Γ.

Σημειώνεται ξανά ότι το σήμα τερματισμού (End) από τη μονάδα ελέγχου προκαλεί την έναρξη ενός νέου κύκλου ανάκτησης εντολής.

Εκτέλεση εντολών σε ΚΜΕ πολλαπλών διαύλων

Βήμα	Λειτουργίες	LOAD Address, R2 (εντολή 2 λέξεων)
1	PC_{out}, R=B, MAR_{in}, Read, IncPC	Ανάκτηση 1ης λέξης εντολής.
2	WMFC	
3	MDR_{outB}, R=B, IR_{in}	
4	PC_{out}, R=B, MAR_{in}, Read, IncPC	Ανάκτηση 2ης λέξης εντολής (διεύθυνση Address).
5	WMFC	
6	MDR_{outB}, R=B, MAR_{in}, Read	Μεταφορά της διεύθυνσης μέσω του διαύλου Β και της ΑΛΜ στον MAR, ώστε να εκτελεστεί λειτουργία ανάγνωσης των δεδομένων από την μνήμη.
7	WMFC	
8	MDR_{outB}, R=B, $R2_{in}$, End	Μεταφορά των δεδομένων μέσω του διαύλου Β, της ΑΛΜ και του διαύλου Γ στον R2.

Προσεγγίσεις σχεδιασμού της μονάδας ελέγχου

- Για να εκτελεστούν οι διάφορες εντολές στον επεξεργαστή, πρέπει να παραχθούν τα κατάλληλα σήματα ελέγχου, έτσι ώστε να τηρηθεί η σωστή ακολουθία των λειτουργιών (βημάτων) που περιλαμβάνει κάθε εντολή.

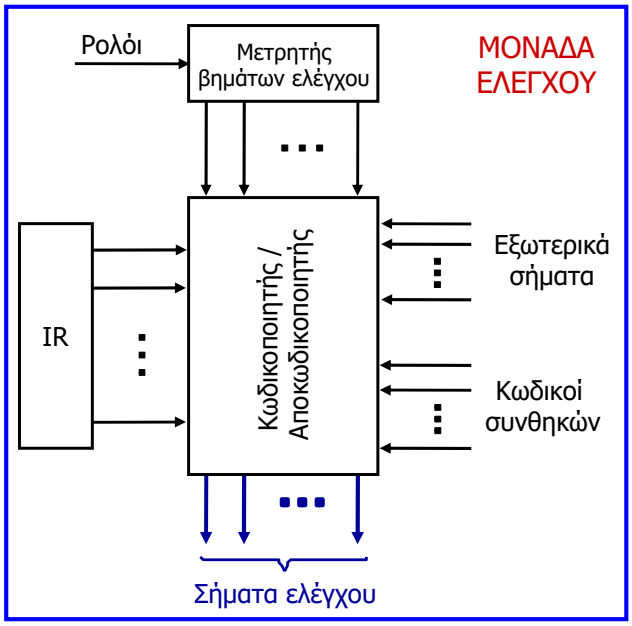
- Οι βασικές προσεγγίσεις που ακολουθούνται για την παραγωγή των κατάλληλων σημάτων ελέγχου, δηλαδή για το σχεδιασμό της μονάδας ελέγχου του επεξεργαστή είναι δύο:

  ✓ προκατασκευασμένος έλεγχος (hardwired control) και

  ✓ μικροπρογραμματιζόμενος έλεγχος (microprogrammed control).

Προκατασκευασμένος έλεγχος

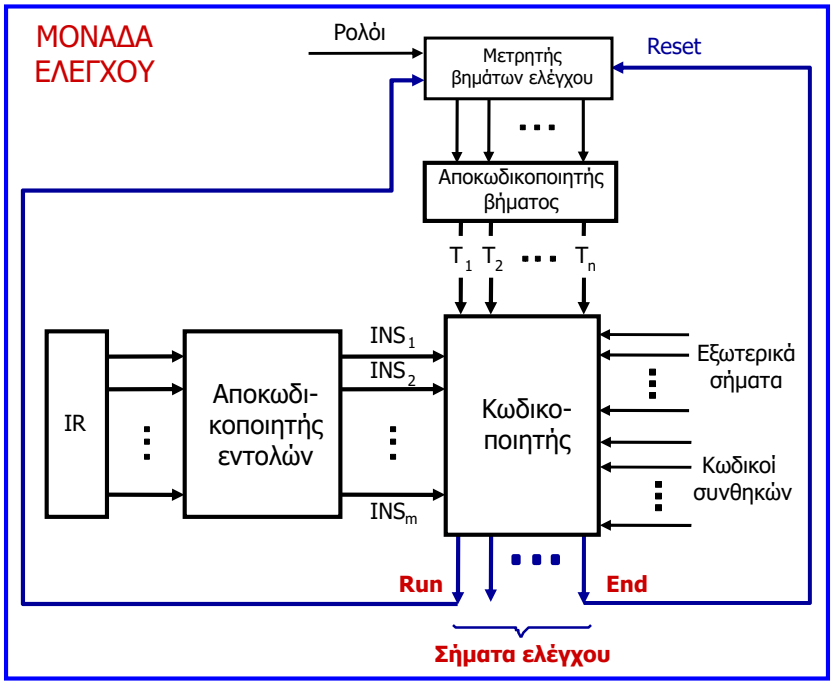
- Όταν χρησιμοποιείται η τεχνική του προκατασκευασμένου ελέγχου για την παραγωγή των σημάτων που ελέγχουν τη σωστή ακολουθία ενεργειών κατά την εκτέλεση των εντολών, χρησιμοποιείται μετρητής ο οποίος παρακολουθεί την ακολουθία των βημάτων εκτέλεσης εντολών (control step counter).
- Στην ουσία η μονάδα ελέγχου είναι ένα ακολουθιακό σύστημα που αλλάζει κατάσταση ανάλογα με τα παρακάτω:
  ✓ Περιεχόμενο μετρητή βημάτων.
  ✓ Περιεχόμενο IR.
  ✓ Περιεχόμενο σημαιών κατάστασης (condition flags).
  ✓ Εξωτερικά σήματα (π.χ. MFC).



- Σε κάθε κατάσταση, καθορίζεται η τιμή των σημάτων ελέγχου που ρυθμίζουν τη λειτουργία των μονάδων του επεξεργαστή στις οποίες συνδέονται.

Προκατασκευασμένος έλεγχος

- Ο αποκωδικοποιητής βήματος (step decoder) παρέχει μία ξεχωριστή γραμμή για κάθε βήμα ελέγχου.
- Ο αποκωδικοποιητής εντολών (instruction decoder) παρέχει μία ξεχωριστή γραμμή για κάθε εντολή (μία γραμμή είναι κάθε φορά ενεργή).
- Ο κωδικοποιητής με βάση τις εισόδους του, παράγει με συνδυασμό λογικών πυλών (hardwired) τα επιμέρους σήματα ελέγχου Y_{in}, PC_{out}, Add κ.ά.
- Όταν Run=1, ο μετρητής βήματος αυξάνεται κατά 1 στο τέλος κάθε κύκλου ρολογιού, ενώ όταν Run=0 ο μετρητής σταματά (για αναμονή απόκρισης μνήμης).



- Όταν End=1, ενεργοποιείται η ανάκτηση νέας εντολής, ενώ όταν End=0 επιτρέπεται η συνέχιση της εκτέλεσης της τρέχουσας εντολής.

Μικροπρογραμματιζόμενος έλεγχος

- Η μονάδα ελέγχου που χρησιμοποιεί προκατασκευασμένο έλεγχο, λειτουργεί σε υψηλές ταχύτητες, αλλά διαθέτει περιορισμένη ευελιξία και η πολυπλοκότητα του συνόλου εντολών που μπορεί να χειριστεί είναι περιορισμένη.
- Ο μικροπρογραμματιζόμενος έλεγχος είναι μία εναλλακτική τεχνική, κατά την οποία τα σήματα ελέγχου παράγονται από πρόγραμμα γλώσσας μηχανής.
- Η λέξη ελέγχου (control word, CW) συνιστάται από ψηφία που αναπαριστούν τα διάφορα σήματα ελέγχου. Για κάθε βήμα ελέγχου στην ακολουθία μίας εντολής υπάρχει ένας μοναδικός συνδυασμός από 1 και 0 στην CW.
- Μια ακολουθία από CW που αντιστοιχεί στην ακολουθία ελέγχου μιας εντολής, λέγεται μικρορουτίνα (microroutine) της εντολής και οι CW της μικρορουτίνας λέγονται μικροεντολές (microinstructions).

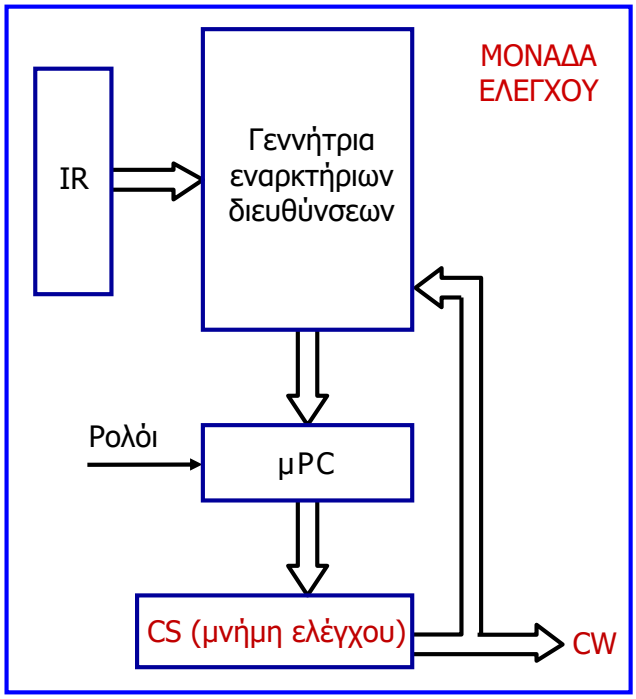
Βήμα	Ενέργειες	ADD (R3), R1
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}	
2	Z_{out}, PC_{in}, Y_{in}, WMFC	
3	MDR_{out}, IR_{in}	
4	$R3_{out}$, MAR_{in}, Read	
5	$R1_{out}$, Y_{in}, WMFC	
6	MDR_{out}, SelectY, Add, Z_{in}	
7	Z_{out}, $R1_{in}$, End	

CW ή μικροεντολή

Μικρο-εντολή	PC_{in}	PC_{out}	MAR_{in}	Read	MDR_{out}	IR_{in}	Y_{in}	Select	Add	Z_{in}	Z_{out}	$R1_{out}$	$R1_{in}$	$R3_{out}$	WMFC	End
1	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0
2	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
3	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0
6	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1

Μικροπρογραμματιζόμενος έλεγχος

- Οι μικροροutίνες των εντολών του συνόλου εντολών ενός επεξεργαστή αποθηκεύονται στο χώρο αποθήκευσης ελέγχου (μνήμη ελέγχου, control store, CS).
- Η μονάδα ελέγχου παράγει τα σήματα ελέγχου για κάθε εντολή, διαβάζοντας σειριακά τις CW της αντίστοιχης μικροροutίνας από την CS.
- Ο μετρητής μικροπρογράμματος (microprogram counter, μPC) χρησιμοποιείται για την σειριακή ανάγνωση των CW από την CS
- Κάθε φορά που μία νέα εντολή φορτώνεται στον IR, η έξοδος της γεννήτριας εναρκτήριων διευθύνσεων (starting address generator) φορτώνεται στον μPC, ο οποίος στη συνέχεια αυξάνεται αυτόματα από το ρολόι, ενεργοποιώντας τη σειριακή ανάγνωση των μικροεντολών της εντολής από την CS.



Τρόποι μορφοποίησης μικροεντολών

- Άμεσος τρόπος μορφοποίησης μικροεντολών είναι η ανάθεση μιας θέσης ψηφίου σε κάθε σήμα ελέγχου, αλλά αυτό οδηγεί σε μικροεντολές μεγάλου μήκους. Πρόκειται για σχήμα οριζόντιας οργάνωσης (horizontal organization) όπου μία μικροεντολή ελέγχει πολλούς υπολογιστικούς πόρους, οδηγώντας σε αύξηση της ταχύτητας λειτουργίας.

F1 (4 bits)	F2 (3 bits)	F3 (3 bits)	F4 (4 bits)	F5 (2 bits)
0000: No transfer 0001: PC_{out} 0010: MDR_{out} 0011: Z_{out} 0100: $R0_{out}$ 0101: $R1_{out}$ 0110: $R2_{out}$ 0111: $R3_{out}$ 1010: $TEMP_{out}$ 1011: $Offset_{out}$	000: No transfer 001: PC_{in} 010: IR_{in} 011: Z_{in} 100: $R0_{in}$ 101: $R1_{in}$ 110: $R2_{in}$ 111: $R3_{in}$	000: No transfer 001: MAR_{in} 010: MDR_{in} 011: $TEMP_{in}$ 100: Y_{in}	0000: Add 0001: Sub ⋮ 1111: XOR 16 λειτουργίες ΑΛΜ	00: No action 01: Read 10: Write

F6 (1 bit)	F7 (1 bit)	F8 (1 bit)
0: SelectY 1: Select4	0: No action 1: WMFC	0: Continue 1: End

Μερική μορφοποίηση για κωδικοποίηση πεδίων μικροεντολών

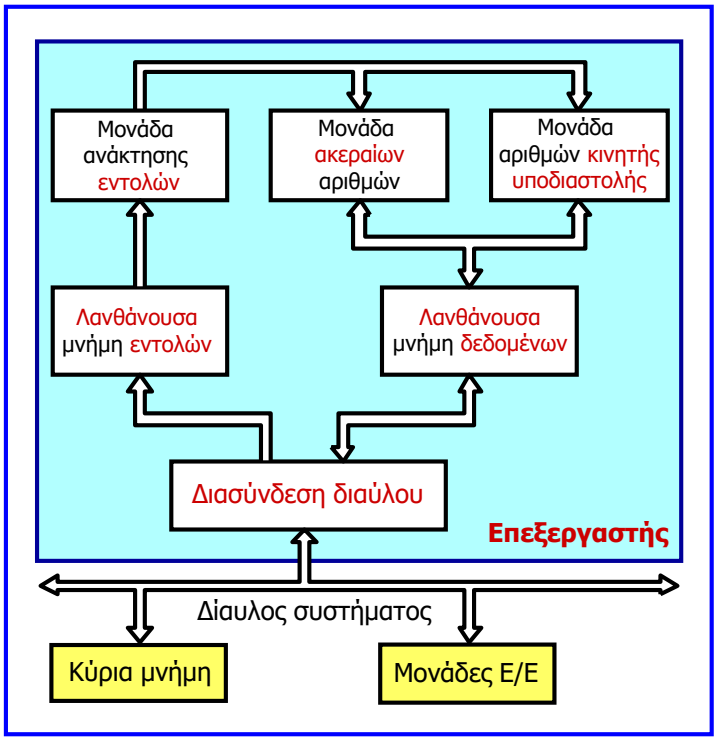
- Δεύτερη λύση είναι ο προσδιορισμός συνδυασμών σημάτων ελέγχου που έχουν νόημα και η αντιστοίχιση κάθε συνδυασμού σε μοναδικό κωδικό που αναπαριστά την αντίστοιχη μικροεντολή. Πρόκειται για σχήμα καθετοποιημένης οργάνωσης (vertical organization) που απαιτεί περισσότερες από μία μικροεντολές για την επίτευξη των επιθυμητών λειτουργιών ελέγχου, με αποτέλεσμα μειωμένη ταχύτητα λειτουργίας.
- Ενδιάμεση λύση είναι η μερική μορφοποίηση (partial format) μικροεντολών, όπου κάθε ομάδα σημάτων καταλαμβάνει ένα πεδίο της μικροεντολής και σε κάθε ομάδα περιλαμβάνονται σήματα που αντιστοιχούν σε αμοιβαία αποκλειόμενες λειτουργίες ή σήματα για τα οποία δεν υπάρχει ανάγκη ταυτόχρονης ύπαρξης.

Σύγκριση προσεγγίσεων σχεδιασμού μονάδας ελέγχου

- Η μονάδα ελέγχου που χρησιμοποιεί προκατασκευασμένο (hardwired) έλεγχο:
  - ☑ Λειτουργεί σε υψηλές ταχύτητες.
  - ☒ Απαιτεί κυκλώματα σύνθετης λογικής που αυξάνουν το κόστος.
  - ☒ Διαθέτει περιορισμένη ευελιξία (δυσκολία προσθήκης νέας εντολής).
  - ☒ Μπορεί να χειριστεί σύνολα εντολών περιορισμένης πολυπλοκότητας.
- Η μονάδα ελέγχου που χρησιμοποιεί μικροπρογραμματιζόμενο (microprogrammed) έλεγχο:
  - ☒ Λειτουργεί σε σχετικά χαμηλότερες ταχύτητες.
  - ☑ Είναι απλούστερη, αφού τα σήματα ελέγχου παράγονται από πρόγραμμα γλώσσας μηχανής και όχι από σύνθετα κυκλώματα.
  - ☑ Είναι πιο αξιόπιστη (μικρότερη πιθανότητα σφαλμάτων).
  - ☑ Προσφέρει αυξημένη ευελιξία (εύκολη προσθήκη νέων εντολών).
- Προκατασκευασμένος έλεγχος ακολουθείται συνήθως σε επεξεργαστές RISC, ενώ μικροπρογραμματιζόμενος έλεγχος ακολουθείται συνήθως σε επεξεργαστές CISC.

Βασική δομή πλήρους επεξεργαστή

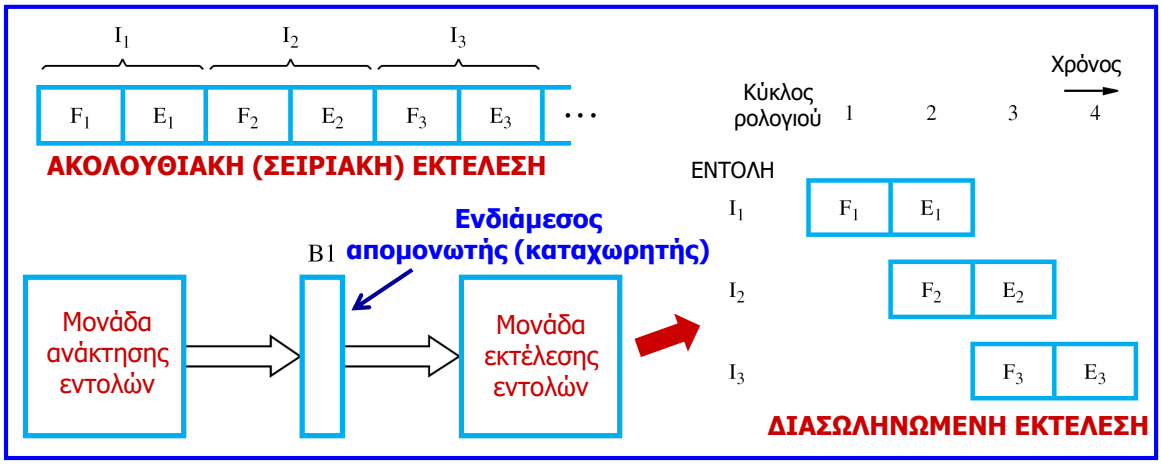
- Σε έναν πλήρη επεξεργαστή, η μονάδα ανάκτησης εντολών ανακτά εντολές από την λανθάνουσα μνήμη εντολών ή από την κύρια μνήμη.
- Ο πλήρης επεξεργαστής διαθέτει ξεχωριστές μονάδες επεξεργασίας για το χειρισμό ακεραίων δεδομένων και δεδομένων κινητής υποδιαστολής, οι οποίες ανακτούν δεδομένα από τη λανθάνουσα ή την κύρια μνήμη.
- Η μονάδα ελέγχου κατανέμεται συνήθως μεταξύ της μονάδας ανάκτησης εντολών και των μονάδων επεξεργασίας.
- Ο επεξεργαστής επικοινωνεί με το υπόλοιπο σύστημα μέσω της διασύνδεσης διαύλου.



Βασικές έννοιες

- Ένας τρόπος αύξησης της ταχύτητας των επεξεργαστών είναι η αναδιάταξη του υλικού ώστε περισσότερες από μία λειτουργίες να μπορούν να εκτελούνται ταυτόχρονα.
- Η διασωλήνωση (pipelining) είναι ιδιαίτερα αποδοτικός τρόπος οργάνωσης παράλληλων δραστηριοτήτων σε ένα υπολογιστικό σύστημα.
- Ένας επεξεργαστής που δεν χρησιμοποιεί διασωλήνωση εκτελεί ένα πρόγραμμα ανακτώντας και εκτελώντας τις εντολές του (καθεμία από τις οποίες αποτελείται από τα βήματα ανάκτησης και εκτέλεσης) τη μία μετά την άλλη (ακολουθιακή εκτέλεση).
- Βασική ιδέα της διασωλήνωσης είναι η τοποθέτηση ενός ενδιάμεσου απομονωτή (intermediate buffer), στην ουσία ενός καταχωρητή, μεταξύ των μονάδων ανάκτησης και εκτέλεσης εντολών (fetch, execution units), ώστε να είναι δυνατή η εκτέλεση μιας εντολής από την μονάδα εκτέλεσης, ενώ η μονάδα ανάκτησης ήδη προχωρά στην ανάκτηση της επόμενης εντολής.
- Ο επεξεργαστής ελέγχεται από ρολόι, η περίοδος του οποίου θα πρέπει να επιτρέπει στα βήματα ανάκτησης και εκτέλεσης να ολοκληρωθούν σε έναν κύκλο ρολογιού.

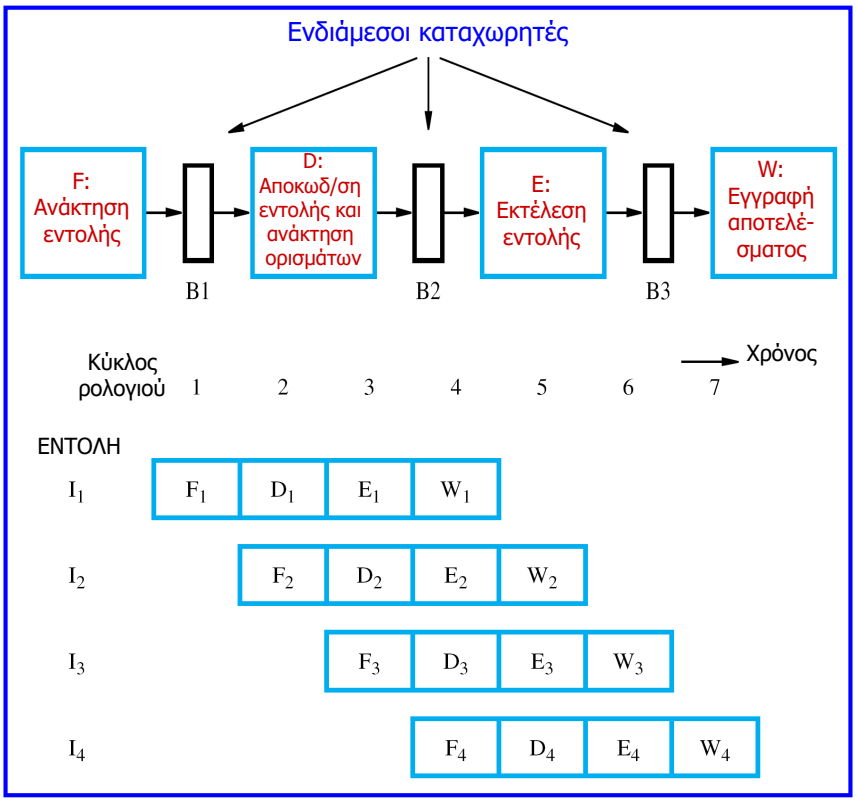
Βασικές έννοιες



- Η παραπάνω δομή των μονάδων ανάκτησης και εκτέλεσης αποτελεί μία σωλήνωση δύο βαθμίδων (two-stage pipeline), κατά την οποία κάθε βαθμίδα επιτελεί ένα βήμα επεξεργασίας της εντολής.
- Ο ενδιάμεσος καταχωρητής απαιτείται για την κατακράτηση της πληροφορίας, η οποία διοχετεύεται από τη μία βαθμίδα στην επόμενη.
- Νέα πληροφορία φορτώνεται στον απομονωτή στο τέλος κάθε κύκλου ρολογιού.

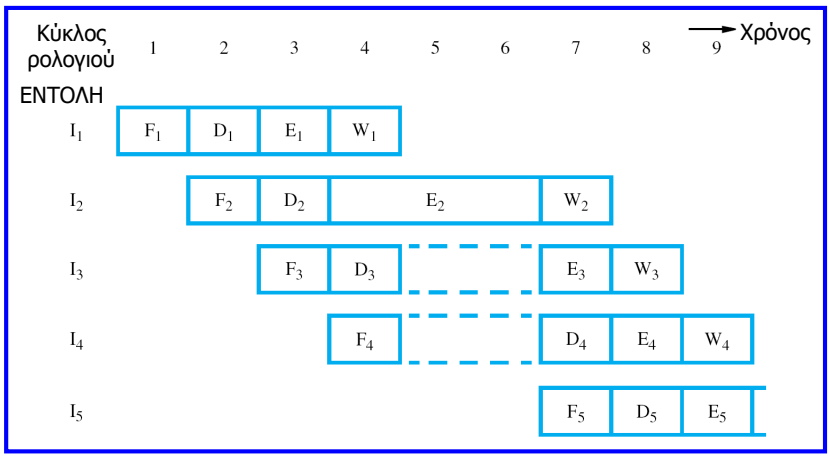
Βασικές έννοιες

- Η επεξεργασία μίας εντολής μπορεί να επιμεριστεί σε περισσότερα βήματα ώστε να προκύψει σωλήνωση περισσότερων βαθμίδων, π.χ. σωλήνωση 4 βαθμίδων (4-stage pipeline).
- Εάν οι μονάδες απαιτούν διαφορετικό χρόνο εκτέλεσης, η περίοδος ρολογιού θα πρέπει να επιτρέπει στη βραδύτερη (δηλαδή στη μνήμη από την οποία ανακτώνται οι εντολές) να ολοκληρώσει τη λειτουργία της.
- Η χρήση λανθανουσών μνημών εντολών και δεδομένων επιλύει το πρόβλημα της αργής προσπέλασης μνήμης.



Απόδοση διασωλήνωσης και κίνδυνοι δεδομένων

- Σε ιδανικές συνθήκες (δηλαδή μη διακοπτόμενη διασωλήνωση και ολοκλήρωση κάθε φάσης σε έναν κύκλο), ο επεξεργαστής με διασωλήνωση 4 βαθμίδων ολοκληρώνει μία εντολή σε κάθε κύκλο ρολογιού (τετραπλάσιος ρυθμός από εκείνον της ακολουθιακής επεξεργασίας).

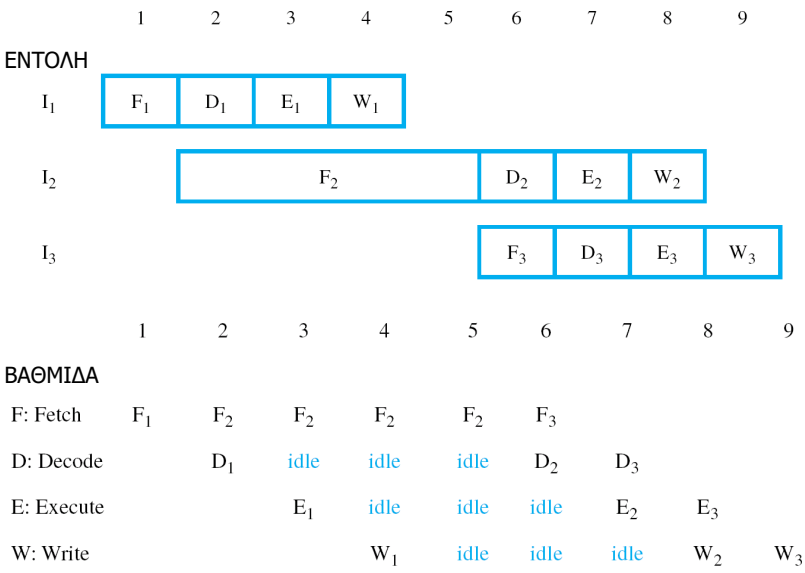


- Υπάρχουν όμως καταστάσεις που λέγονται κίνδυνοι (hazards) και αναστέλλουν (stall) τη διασωληνωμένη λειτουργία.
- Για παράδειγμα, εάν η 2η εντολή αφορά πράξη διαίρεσης, απαιτεί περισσότερους από έναν κύκλο για την εκτέλεσή της, με αποτέλεσμα η μονάδα εγγραφής να παραμένει αδρανής για δύο κύκλους (αναστολή διασωληνωμένης λειτουργίας) και το περιεχόμενο του απομονωτή (B2) που προηγείται της μονάδας εκτέλεσης να πρέπει να παραμείνει ανέπαφο.
- Η κατάσταση λέγεται κίνδυνος δεδομένων (data hazard), όπως και κάθε κατάσταση όπου ορίσματα αφετηρίας ή προορισμού εντολών δεν είναι διαθέσιμα στον κατάλληλο χρόνο.

Απόδοση διασωλήνωσης και κίνδυνοι εντολών

- Η διασωληνωμένη λειτουργία μπορεί να ανασταλεί λόγω καθυστέρησης διαθεσιμότητας μιας εντολής (αστοχία λανθάνουσας μνήμης και ανάκτηση από την κύρια μνήμη). Οι καταστάσεις αυτού του είδους, αναφέρονται ως κίνδυνοι εντολών (instruction hazards) ή κίνδυνοι ελέγχου (control hazards).
- Οι περίοδοι αδράνειας (stalls) μονάδων αναφέρονται και ως φυσαλίδες (bubbles) σωλήνωσης.

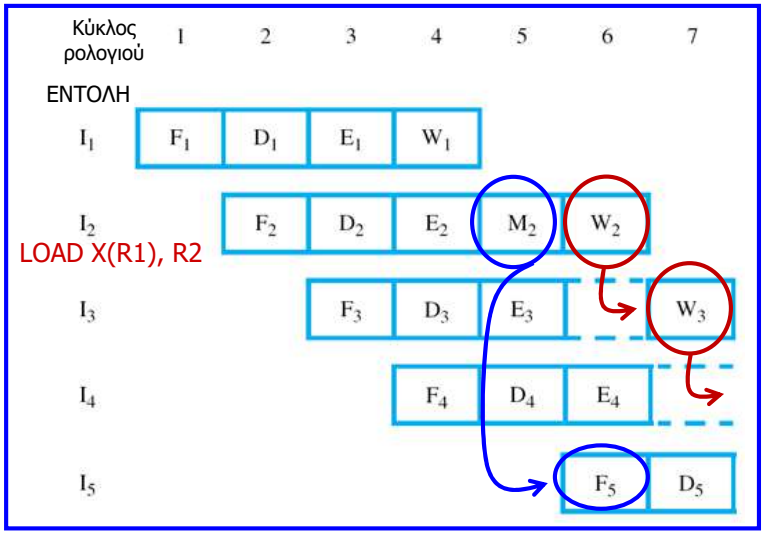
Φάσεις εκτέλεσης εντολής σε διαδοχικούς κύκλους ρολογιού



Λειτουργίες σε κάθε βαθμίδα σε διαδοχικούς κύκλους ρολογιού

Απόδοση διασωλήνωσης και δομικοί κίνδυνοι

- Ένας τρίτος τύπος κινδύνου είναι ο δομικός κίνδυνος (structural hazard), που συμβαίνει όταν δύο εντολές απαιτούν ταυτόχρονη χρήση του ίδιου υπολογιστικού πόρου υλικού.
- Συναντάται συνήθως όταν μία εντολή απαιτεί προσπέλαση της μνήμης για εκτέλεση ή εγγραφή, ενώ μία δεύτερη βρίσκεται στη φάση ανάκτησης.
- Λύση αποτελεί η χρησιμοποίηση ξεχωριστών λανθανουσών μνημών εντολών και δεδομένων.
- Οι εντολές 2, 3 απαιτούν ταυτόχρονα πρόσβαση στη συστοιχία καταχωρητών, αφού η εντολή 2 (LOAD) υπολογίζει τη διεύθυνση μνήμης στον κύκλο 4 και την θέτει στον MAR, ανακτά το όρισμα στον κύκλο 5 (δηλ. απαιτεί 2 κύκλους εκτέλεσης (4, 5) και το φορτώνει στον R2 στον κύκλο 6, όταν και η εντολή 3 απαιτεί πρόσβαση στη συστοιχία για εγγραφή αποτελέσματος.
- Εάν η συστοιχία καταχωρητών διαθέτει 2 πόρτες εισόδου, δε θα συμβεί αναστολή.

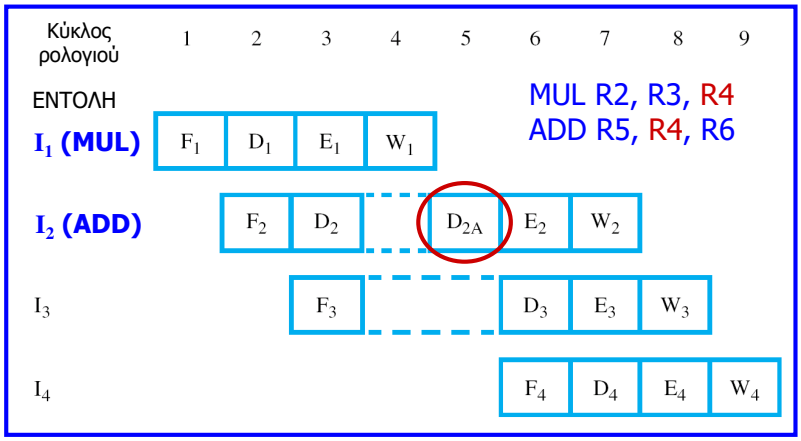


Απόδοση διασωλήνωσης: σύνοψη

- Η διασωλήνωση δεν έχει ως αποτέλεσμα τη γρηγορότερη εκτέλεση των επιμέρους εντολών.
- Αντί αυτού αυξάνεται η παροχή (throughput) εντολών, δηλαδή ο ρυθμός ολοκλήρωσης της εκτέλεσης των εντολών.
- Σε οποιαδήποτε χρονική στιγμή, εάν οποιαδήποτε από τις μονάδες (βαθμίδες) διασωλήνωσης αδυνατεί να ολοκληρώσει τη λειτουργία που της ανατέθηκε σε έναν κύκλο ρολογιού, τότε η διασωληνωμένη λειτουργία αναστέλλεται, με αποτέλεσμα αναπόφευκτη μείωση της αναμενόμενης απόδοσης.
- Επομένως, το επίπεδο απόδοσης ολοκλήρωσης μίας εντολής σε κάθε κύκλο ρολογιού αποτελεί το άνω όριο παροχής εντολών (ρυθμού ολοκλήρωσής τους) που μπορεί να επιτευχθεί σε έναν επεξεργαστή με διασωληνωμένη λειτουργία.
- Γενικά, στόχος κατά τη σχεδίαση των επεξεργαστών είναι ο εντοπισμός όλων των πιθανών κινδύνων, οι οποίοι ενδέχεται να προκαλέσουν αναστολή της διασωληνωμένης λειτουργίας, καθώς και η ανεύρεση τεχνικών ελαχιστοποίησης των επιπτώσεών τους.

Κίνδυνοι δεδομένων

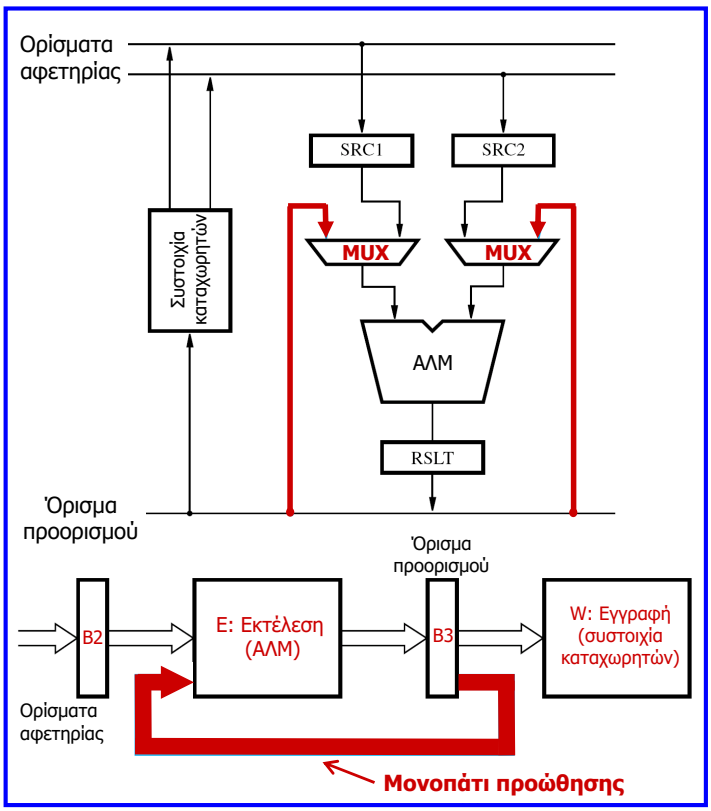
- Όταν δύο εντολές είναι αλληλεξαρτώμενες (σε σχέση με τα δεδομένα τους), θα πρέπει να εκτελεστούν με τη σωστή σειρά.
- Στο διπλανό σχήμα, το αποτέλεσμα της MUL τοποθετείται στον R4, που είναι ένα από τα ορίσματα αφετηρίας της ADD.
- Η φάση D της ADD δε μπορεί να ολοκληρωθεί πριν τη λήξη της φάσης W της MUL και καθυστερεί έως τον 5ο κύκλο.
- Η 3η εντολή ανακτάται κανονικά, αλλά δε μπορεί να αποκωδικοποιηθεί πριν από τη 2η εντολή που ανακτήθηκε νωρίτερα, με αποτέλεσμα την αναστολή 2 κύκλων.



- Ο κίνδυνος δεδομένων εγείρεται διότι η ανάγνωση ορισμάτων γίνεται στη φάση D και μία εντολή αναμένει την εγγραφή δεδομένων (ενός ορίσματος) στη συστοιχία των καταχωρητών από προηγούμενη.
- Το πρόβλημα αντιμετωπίζεται με την τροποποίηση του μονοπατιού δεδομένων (datapath) του επεξεργαστή και την εισαγωγή του μηχανισμού προώθησης ορίσματος (operand forwarding).

Κίνδυνοι δεδομένων και προώθηση ορίσματος

- Ο μηχανισμός προώθησης ορίσματος περιλαμβάνει δύο γραμμές σύνδεσης και δύο πολυπλέκτες.
- Οι πολυπλέκτες επιτρέπουν την επιλογή του ορίσματος προορισμού αντί του περιεχομένου των καταχωρητών SRC1, SRC2.
- Οι καταχωρητές SRC1, SRC2 αντιστοιχούν στον καταχωρητή B2 και ο καταχωρητής RSLT αντιστοιχεί στον καταχωρητή B3.
- Εναλλακτική αντιμετώπιση (χωρίς αύξησης πολυπλοκότητας υλικού) είναι η εισαγωγή από τον μεταγλωττιστή, καθυστέρησης 2 κύκλων με την προσθήκη 2 εντολών αδράνειας (NOP) μεταξύ των εντολών MUL και ADD.



Παράδειγμα

- Έστω το πρόγραμμα:

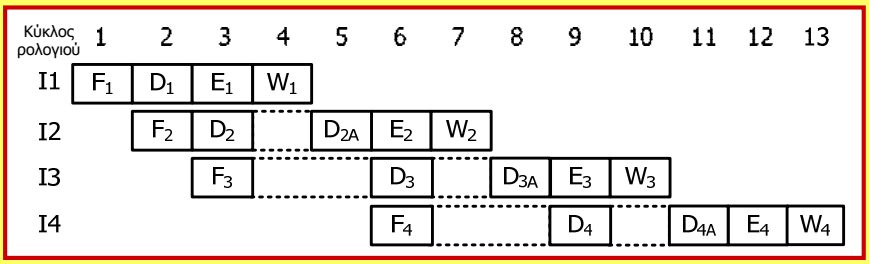
```
I1:  SUB  R2, R3, R1;    R2 = R3 – R1
I2:  AND  R6, R5, R2;    R6 = R5 ∧ R2
I3:  OR   R4, R6, #03;   R4 = R6 ∨ 03
I4:  ADD R6, R4, #05;    R6 = R4 + 05
```

  που εκτελείται σε επεξεργαστή με διασωλήνωση 4 βαθμίδων, δηλαδή: ανάκτηση (fetch), αποκωδικοποίηση (decode), εκτέλεση (execute) και εγγραφή (write).

- Να δοθεί το διάγραμμα της ακολουθίας γεγονότων στη διασωλήνωση κατά την εκτέλεση του παραπάνω προγράμματος.
- Να δοθεί το διάγραμμα της ακολουθίας γεγονότων στη διασωλήνωση κατά την εκτέλεση του προγράμματος, μετά την προσθήκη μονοπατιού προώθησης ορίσματος (operand forwarding) προς μια από τις εισόδους της βαθμίδας εκτέλεσης.
- Θα υπάρξει βελτίωση στην απόδοση της σωλήνωσης, λόγω της προσθήκης του μονοπατιού προώθησης ορίσματος; Αν ναι, ποιο το όφελος σε κύκλους ρολογιού;

Παράδειγμα

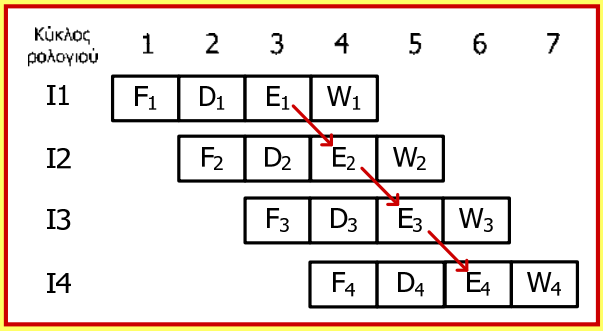
```
I1:  SUB  R2, R3, R1
I2:  AND R6, R5, R2
I3:  OR   R4, R6, #03
I4:  ADD R6, R4, #05
```



- Στον 3ο κύκλο δεν μπορεί να ανακτηθεί το όρισμα της I2 που βρίσκεται στον R2, καθώς δεν έχει ολοκληρωθεί η εγγραφή του από την I1 της οποίας είναι ο καταχωρητής προορισμού.
- Στον 4ο κύκλο γίνεται εγγραφή αποτελέσματος στον R2 από την I1, ενώ η I2 αναμένει στη φάση αποκωδικοποίησης περιμένοντας την I1 να ολοκληρώσει την εγγραφή και εμποδίζοντας την αποκωδικοποίηση της I3 και την προσαγωγή της I4.
- Στον 6ο κύκλο, ξεκινά η εκτέλεση της I2, η αποκωδικοποίηση της I3 και η προσαγωγή της I4. Επειδή, ο καταχωρητής R6 περιέχει πηγαίο όρισμα της I3 αλλά είναι και καταχωρητής προορισμού της I2, η I3 αναμένει στο στάδιο αποκωδικοποίησης μέχρι να ολοκληρωθεί η εγγραφή του αποτελέσματος της I2 στον R6, η οποία γίνεται στον 7ο κύκλο.
- Η εκτέλεση της I3 και η αποκωδικοποίηση της I4 ξεκινούν στον 9ο κύκλο και λόγω της εξάρτησης δεδομένων μεταξύ I3 και I4, η I4 ολοκληρώνει την αποκωδικοποίηση στον 11ο κύκλο. Συνολικός χρόνος εκτέλεσης προγράμματος: 13 κύκλοι ρολογιού.

Παράδειγμα

```
I1:  SUB  R2, R3, R1
I2:  AND  R6, R5, R2
I3:  OR   R4, R6, #03
I4:  ADD R6, R4, #05
```



- Προσθέτοντας το μονοπάτι προώθησης ορίσματος προς μία από τις εισόδους της ΑΛΜ, το αποτέλεσμα της ΑΛΜ ταυτόχρονα με την εγγραφή του στη συστοιχία καταχωρητών μπορεί να χρησιμοποιηθεί ως όρισμα πηγής για την εκτέλεση επόμενης εντολής.
- Έτσι, στον 4ο κύκλο ρολογιού μπορεί να ξεκινήσει η εκτέλεση της I2, αφού το περιεχόμενο του R2 είναι διαθέσιμο νωρίτερα μέσω του μονοπατιού προώθησης, χωρίς να απαιτείται η προσπέλαση της συστοιχίας καταχωρητών. Στον ίδιο κύκλο ρολογιού, ξεκινά η αποκωδικοποίηση της I3 και η προσαγωγή της I4.
- Με τον ίδιο τρόπο αντιμετωπίζεται η εξάρτηση δεδομένων μεταξύ των εντολών I2, I3 και I4. Επιτυγχάνεται έτσι μείωση του χρόνου εκτέλεσης του προγράμματος κατά 2 κύκλους ρολογιού ανά εξάρτηση (συνολική μείωση 6 κύκλων). Συνολικός χρόνος εκτέλεσης προγράμματος: 7 κύκλοι ρολογιού, δηλαδή όφελος 6 κύκλων ρολογιού.

Εξαρτήσεις δεδομένων μεταξύ εντολών

- Κατά τη διασωληνωμένη λειτουργία είναι πιθανό να προκύψουν κίνδυνοι δεδομένων, λόγω διάφορων εξαρτήσεων δεδομένων που αποτελούν ορίσματα των εντολών ενός προγράμματος.
- Θεωρώντας δύο εντολές σε ένα πρόγραμμα (ακολουθία εντολών), με την 1η να προηγείται της 2ης, οι πιθανές εξαρτήσεις δεδομένων μεταξύ τους είναι:
  - ✓ RAW (read after write, ανάγνωση μετά από εγγραφή): η 2η εντολή προσπαθεί να διαβάσει ένα όρισμα πηγής (source operand) πριν αυτό γραφτεί από την 1η εντολή, με πιθανό αποτέλεσμα η 2η εντολή να διαβάσει λανθασμένη τιμή του ορίσματος και να εκτελεστεί σε αυτή.
  - ✓ WAR (write after read, εγγραφή μετά από ανάγνωση): η 2η εντολή προσπαθεί να γράψει ένα όρισμα προορισμού (destination operand) πριν αυτό διαβαστεί από την 1η εντολή, με πιθανό αποτέλεσμα η 1η εντολή να διαβάσει λανθασμένη τιμή του ορίσματος και να εκτελεστεί σε αυτή.
  - ✓ WAW (write after write, εγγραφή μετά από εγγραφή): η 2η εντολή προσπαθεί να γράψει ένα όρισμα πριν αυτό γραφτεί από την 1η εντολή, με πιθανό αποτέλεσμα η εγγραφή του ορίσματος να γίνει με λανθασμένη σειρά, αφήνοντας ως τελική τιμή του αυτή που έχει εγγραφεί από την 1η εντολή, αντί αυτής που θα έπρεπε να εγγραφεί από τη 2η εντολή.

Παράδειγμα

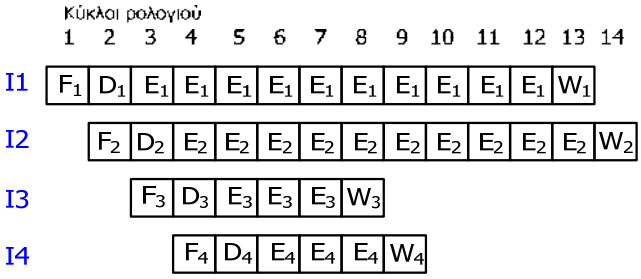
- Έστω το πρόγραμμα:

```
I1:  DIVF  R2, R4, R0;     R2 = R4 / R0
I2:  DIVF  R6, R8, R10;    R6 = R8 / R10
I3:  ADDF R0, R6, R0;      R0 = R6 + R0
I4:  SUBF R6, R4, R10;     R6 = R4 – R10
```

  που εκτελείται στη μονάδα αριθμών κινητής υποδιαστολής ενός επεξεργαστή με διασωλήνωση 4 βαθμίδων, δηλαδή, ανάκτηση (fetch), αποκωδικοποίηση (decode), εκτέλεση (execute) και εγγραφή (write), η μονάδα εκτέλεσης του οποίου διαθέτει κύκλωμα που διενεργεί διαίρεση αριθμών κινητής υποδιαστολής και απαιτεί 10 κύκλους ρολογιού, καθώς και κύκλωμα που διενεργεί πρόσθεση / αφαίρεση αριθμών κινητής υποδιαστολής και απαιτεί 3 κύκλους ρολογιού.
- Να εντοπιστούν οι εξαρτήσεις δεδομένων μεταξύ των εντολών του προγράμματος.
- Να προσδιοριστούν οι δομικοί κίνδυνοι και οι κίνδυνοι δεδομένων της διασωληνωμένης λειτουργίας της μονάδας που υφίστανται κατά την εκτέλεση του προγράμματος.
- Να δοθεί το διάγραμμα της ακολουθίας γεγονότων στη διασωλήνωση κατά την εκτέλεση του προγράμματος, μετά την αντιμετώπιση των υπαρκτών κινδύνων χρησιμοποιώντας αναστολή της διασωληνωμένης λειτουργίας και να υπολογιστεί το πλήθος των κύκλων ρολογιού που απαιτούνται για την ολοκλήρωση της εκτέλεσης του προγράμματος.

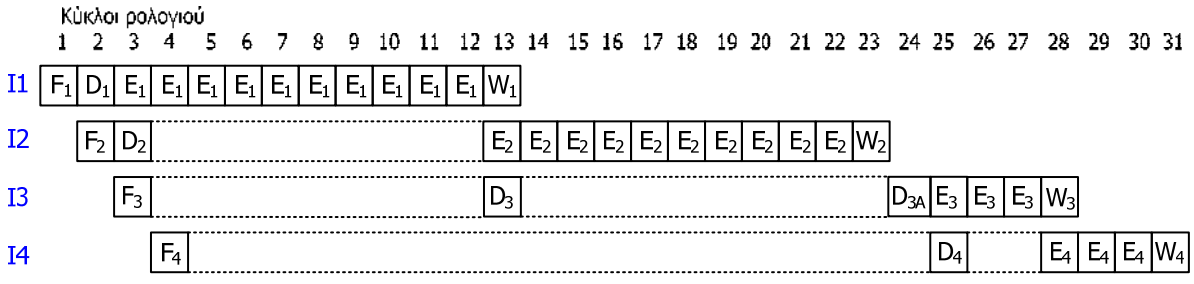
Παράδειγμα

```
I1:  DIVF  R2, R4, R0;     R2 = R4 / R0
I2:  DIVF  R6, R8, R10;    R6 = R8 / R10
I3:  ADDF R0, R6, R0;      R0 = R6 + R0
I4:  SUBF R6, R4, R10;     R6 = R4 – R10
```



- Εξαρτήσεις δεδομένων: RAW μεταξύ I2 και I3 λόγω του ορίσματος R6, WAR μεταξύ I1 και I3 λόγω του ορίσματος R0, WAR μεταξύ I3 και I4 λόγω του ορίσματος R6, WAW μεταξύ I2 και I4 λόγω του ορίσματος R6.
- Παρατηρούμε ότι υφίσταται δομικός κίνδυνος μεταξύ των εντολών I1 και I2, αφού η μονάδα διαθέτει μόνο ένα κύκλωμα διαίρεσης. Επομένως, θα πρέπει να ανασταλεί η εκτέλεσης της εντολής I2 για 9 κύκλους ρολογιού. Έτσι, δημιουργείται και δομικός κίνδυνος μεταξύ των εντολών I2 και I3, που αφορά το στάδιο αποκωδικοποίησης.
- Η εξάρτηση δεδομένων RAW μεταξύ των εντολών I2 και I3 δημιουργεί κίνδυνο δεδομένων, αφού ο καταχωρητής R6 είναι πηγαίο όρισμα της I3 αλλά είναι και όρισμα προορισμού της I2. Επομένως, η I3 θα πρέπει να αναμείνει στο στάδιο αποκωδικοποίησης (D_3) μέχρι να ολοκληρωθεί η εγγραφή (W_2) του αποτελέσματος της εντολής I2 στον R6.

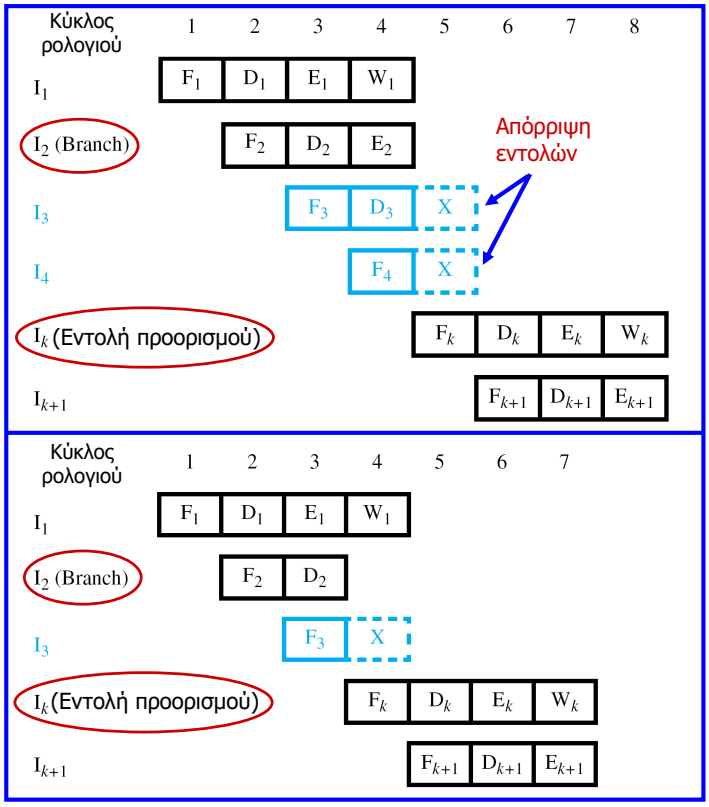
Παράδειγμα



- Επίσης, υφίσταται δομικός κίνδυνος μεταξύ των εντολών I3 και I4, που αφορά το στάδιο αποκωδικοποίησης, αλλά και το στάδιο εκτέλεσης (αφού είναι διαθέσιμο μόνο ένα κύκλωμα πρόσθεσης / αφαίρεσης). Επομένως, θα πρέπει να ανασταλλεί για κατάλληλο πλήθος κύκλων ρολογιού η αποκωδικοποίηση και η εκτέλεση της εντολής I4.
- Μετά τις παραπάνω ενέργειες, παρατηρούμε ότι δε δημιουργούνται πρόσθετοι δομικοί κίνδυνοι ή κίνδυνοι λόγω των υπόλοιπων εξαρτήσεων δεδομένων μεταξύ των εντολών.
- Για την ολοκλήρωση της εκτέλεσης του προγράμματος απαιτούνται 31 κύκλοι ρολογιού.

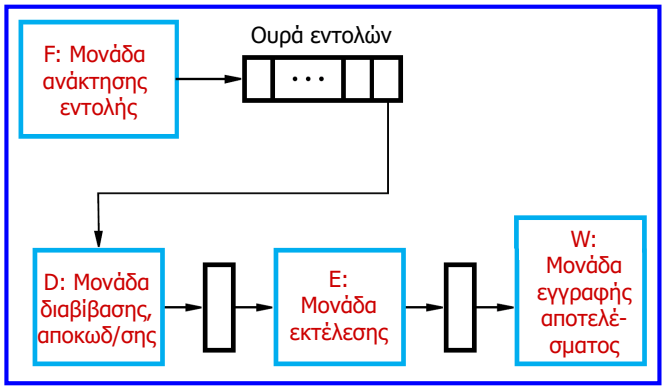
Κίνδυνοι εντολών

- Εκτός από τον κίνδυνο εντολής λόγω αστοχίας της λανθάνουσας μνήμης, μια εντολή διακλάδωσης μπορεί επίσης να προκαλέσει αναστολή της διασωλήνωσης.
- Η διεύθυνση προορισμού της διακλάδωσης υπολογίζεται στη φάση E_2, αλλά μέχρι τότε έχει προχωρήσει η επεξεργασία της 3ης και 4ης εντολής, οι οποίες θα πρέπει να απορριφθούν και να ανακτηθεί η εντολή προορισμού (I_k), με αποτέλεσμα αναστολή 2 κύκλων.
- Ο χαμένος χρόνος που είναι αποτέλεσμα μίας εντολής διακλάδωσης αναφέρεται ως ποινή διακλάδωσης (branch penalty).
- Για μείωση της ποινής, απαιτείται ο πρότερος υπολογισμός της διεύθυνσης προορισμού μέσω ειδικού υλικού στη φάση D_2.



Κίνδυνοι εντολών και μονάδα διαβίβασης

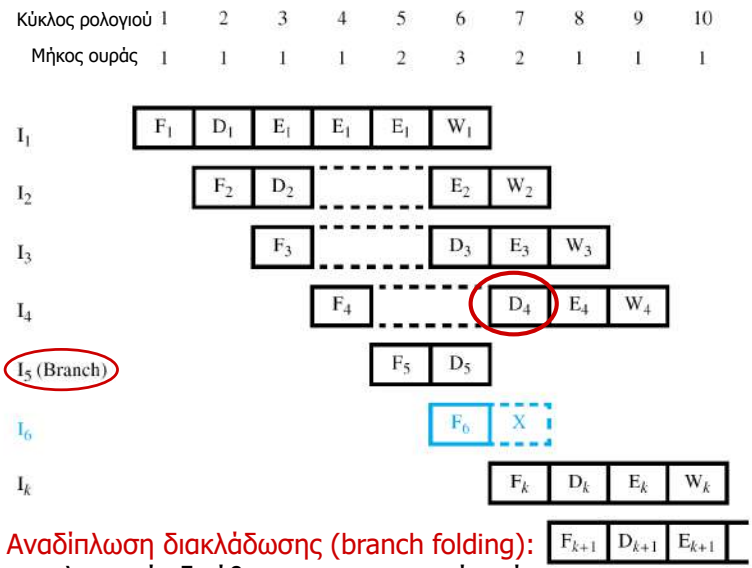
- Για να ελαττωθεί η επίπτωση των ανεπιθύμητων διακοπών, χρησιμοποιούνται εξειδικευμένες μονάδες ανάκτησης εντολών που προανακτούν εντολές και τις τοποθετούν σε ουρά.
- Οι εντολές παραλαμβάνονται από τη μονάδα διαβίβασης (dispatch unit) που επιτελεί και την αποκωδικοποίηση και αποστέλλονται στη μονάδα εκτέλεσης.



- Η μονάδα ανάκτησης θα πρέπει να έχει δυνατότητες επεξεργασίας ώστε να αποκωδικοποιεί εντολές διακλάδωσης χωρίς συνθήκη και να υπολογίζει τη διεύθυνση προορισμού.
- Όταν η διασωλήνωση αναστέλλεται (π.χ. λόγω κινδύνου δεδομένων) η μονάδα διαβίβασης δε μπορεί να εκδώσει εντολές από την ουρά, αλλά η μονάδα ανάκτησης θα πρέπει να εξακολουθήσει να φορτώνει την ουρά με νέες εντολές.
- Αντιστρόφως, εάν υπάρχει καθυστέρηση στην ανάκτηση εντολών (π.χ. λόγω διακλάδωσης ή αστοχίας λανθάνουσας μνήμης), η μονάδα διαβίβασης θα πρέπει να εξακολουθήσει να λαμβάνει εντολές από την ουρά.

Κίνδυνοι εντολών και αναδίπλωση διακλάδωσης

- Κάθε λειτουργία ανάκτησης προσθέτει μία εντολή στην ουρά και κάθε λειτουργία διαβίβασης μειώνει το μήκος ουράς κατά 1.
- Το μήκος ουράς παραμένει το ίδιο για τους 4 πρώτους κύκλους.
- Εάν η πρώτη εντολή εισάγει αναστολή 2 κύκλων, η μονάδα ανάκτησης συνεχίζει να ανακτά εντολές και το μήκος ουράς αυξάνεται σε 3 στον 6ο κύκλο.
- Με την ανάκτηση της εντολής διακλάδωσης, η μονάδα ανάκτησης υπολογίζει τη διεύθυνση προορισμού κατά τη διάρκεια επεξεργασίας στη σωλήνωση των I_1, I_2, I_3.

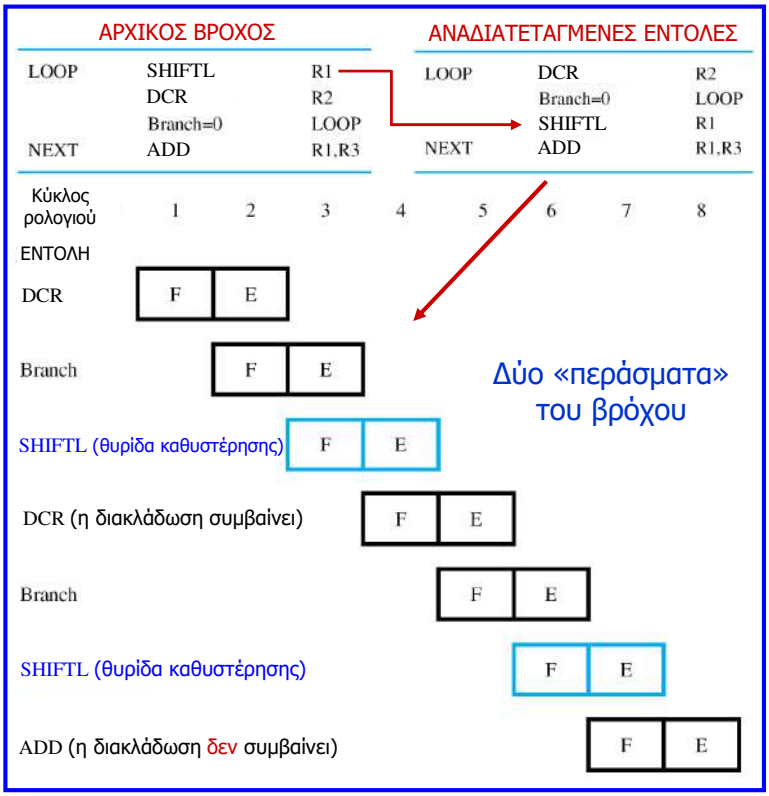


Αναδίπλωση διακλάδωσης (branch folding): υπολογισμός διεύθυνσης προορισμού από τη μονάδα ανάκτησης, παράλληλα με την εκτέλεση άλλων εντολών

- Η εντολή προορισμού (I_k) ανακτάται στον 7ο κύκλο, αλλά αντί να ανασταλεί η λειτουργία της διασωλήνωσης λόγω απόρριψης της I_6, διαβιβάζεται από την ουρά η I_4 προς τη φάση αποκωδικοποίησης και μειώνεται το μήκος της ουράς μετά την απόρριψη της I_6.

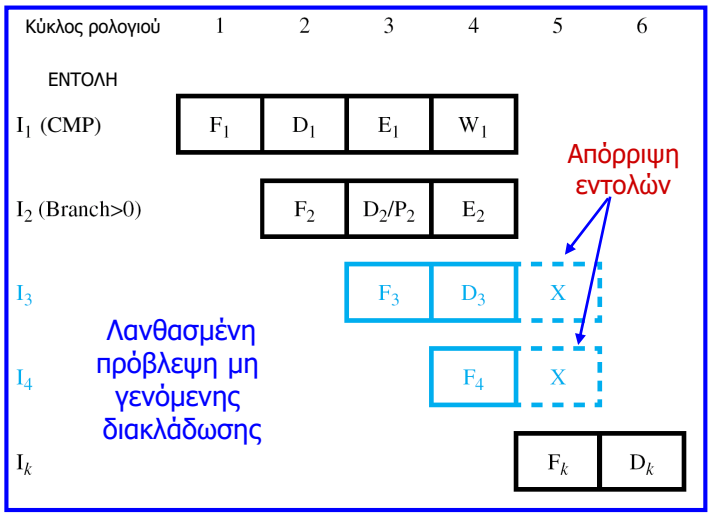
Κίνδυνοι εντολών και διακλαδώσεις υπό συνθήκη

- Οι εντολές διακλάδωσης υπό συνθήκη (20% των εντολών τυπικών προγραμμάτων) εισάγουν πρόσθετο κίνδυνο λόγω της εξάρτησης τους από το αποτέλεσμα προηγούμενης εντολής.
- Η θέση που ακολουθεί μετά από μία διακλάδωση, λέγεται θυρίδα καθυστέρησης διακλάδωσης (branch delay slot).
- Ενδέχεται να υπάρχουν πάνω από μία θυρίδες, ανάλογα με το χρόνο που απαιτεί ο υπολογισμός της διεύθυνσης προορισμού.
- Η τεχνική καθυστερημένης διακλάδωσης (delayed branching) αναδιατάσσει τις εντολές ώστε να τοποθετηθούν στις θυρίδες χρήσιμες εντολές (ή NOPs) που εκτελούνται είτε συμβεί η διακλάδωση είτε όχι.



Στατική πρόβλεψη διακλάδωσης

- Μία άλλη τεχνική για τη μείωση της ποινής διακλάδωσης των εντολών διακλάδωσης με συνθήκη είναι η προσπάθεια πρόβλεψης του εάν θα συμβεί ή όχι μία διακλάδωση.
- Η απλούστερη μορφή πρόβλεψης είναι να υποθέσουμε ότι πάντα η διακλάδωση δεν συμβαίνει ή ότι πάντα συμβαίνει (στατική πρόβλεψη διακλάδωσης, static branch prediction).
- Η εκτέλεση των εντολών που έπονται χρονικά θα πρέπει να είναι υποθετική και εάν η απόφαση διακλάδωσης είναι διαφορετική, τότε οι εντολές αυτές και τα σχετικά δεδομένα θα πρέπει να απορριφθούν και να ανακτηθούν οι ορθές προς εκτέλεση εντολές.
- Η πρόβλεψη ανατίθεται συνήθως στο μεταγλωττιστή και η μέση ελάττωση του χρόνου ελέγχου είναι 50%.

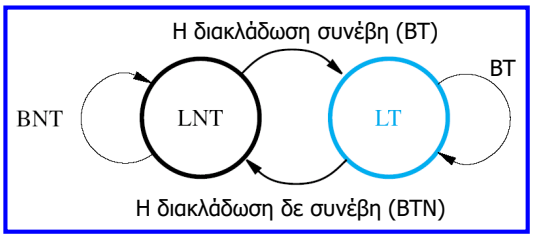


- Καλύτερη απόδοση προκύπτει όταν από τη συμπεριφορά του προγράμματος, μερικές διακλαδώσεις υποτεθεί ότι πάντα συμβαίνουν και άλλες ότι πάντα δε συμβαίνουν.

Παράδειγμα: οι εντολές διακλάδωσης στο τέλος βρόχου προκαλούν διακλάδωση πάντα εκτός της τελευταίας επανάληψης, ενώ στην αρχή του βρόχου προκαλούν διακλάδωση μόνο στην τελευταία επανάληψη.

Δυναμική πρόβλεψη διακλάδωσης

Αλγόριθμος πρόβλεψης δύο κατα-στάσεων



LT: η διακλάδωση μάλλον θα συμβεί

LNT: η διακλάδωση μάλλον δε θα συμβεί

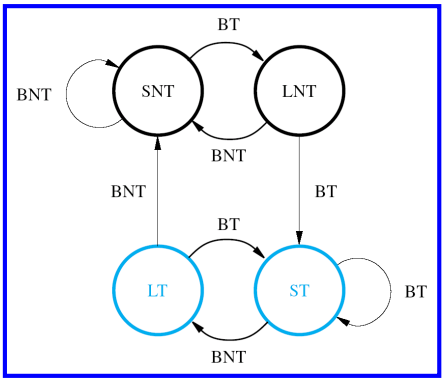
(B = BRANCH, L = LIKELY, N = NOT, T= TAKEN)

- Μία άλλη προσέγγιση κατά την οποία η πρόβλεψη μπορεί να αλλάξει, ανάλογα με τις αποφάσεις διακλάδωσης κάθε φορά που εκτελέστηκε η εντολή στο παρελθόν, είναι η δυναμική πρόβλεψη διακλάδωσης (dynamic branch prediction).
- Στον πιο απλό αλγόριθμο δυναμικής πρόβλεψης, χρησιμοποιείται το αποτέλεσμα της τελευταίας εκτέλεσης της εντολής και ο επεξεργαστής υποθέτει ότι υπάρχει μεγάλη πιθανότητα να συμβεί το ίδιο κατά την επόμενη εκτέλεση.
- Ο αλγόριθμος (δύο καταστάσεων) απαιτεί ένα bit ιστορικής πληροφορίας για κάθε εντολή διακλάδωσης και είναι αρκετά αποδοτικός στους βρόχους επανάληψης.

Παράδειγμα: Όταν το πρόγραμμα εισέρχεται σε βρόχο επανάληψης με εντολή διακλάδωσης στο τέλος του και η αρχική κατάσταση είναι η LNT, ο αλγόριθμος θα δίνει πάντα το ίδιο αποτέλεσμα μέχρι να το τελευταίο πέρασμα όπου η πρόβλεψη θα αποδειχθεί λανθασμένη και η μηχανή καταστάσεων θα αλλάξει κατάσταση με αποτέλεσμα και την επόμενη φορά που θα εισέλθει το πρόγραμμα στο βρόχο να προκύψει πάλι λανθασμένη πρόβλεψη.

Δυναμική πρόβλεψη διακλάδωσης

Αλγόριθμος πρόβλεψης τεσσάρων καταστάσεων



LT: η διακλάδωση μάλλον θα συμβεί

LNT: η διακλάδωση μάλλον δε θα συμβεί

ST: ισχυρή πιθανότητα να συμβεί

SNT: ισχυρή πιθανότητα να μη συμβεί

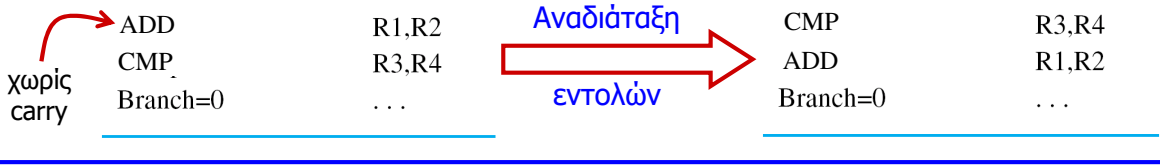
(B = BRANCH, L = LIKELY,
S = STRONGLY LIKELY, N = NOT, T= TAKEN)

- Καλύτερη απόδοση επιτυγχάνεται με τον αλγόριθμο τεσσάρων καταστάσεων που χρησιμοποιεί περισσότερη ιστορική πληροφορία εκτέλεσης για κάθε εντολή διακλάδωσης και απαιτεί δύο bits ιστορικής πληροφορίας για κάθε εντολή διακλάδωσης.

Παράδειγμα: Όταν το πρόγραμμα εισέρχεται σε βρόχο επανάληψης με εντολή διακλάδωσης στο τέλος του και η αρχική κατάσταση είναι η LNT, η κατάσταση μετά το 1ο πέρασμα (λανθασμένη πρόβλεψη) θα γίνει ST και σε όλα τα επόμενα περάσματα η πρόβλεψη θα είναι σωστή εκτός του τελευταίου, οπότε η κατάσταση θα αλλάξει σε LT, με αποτέλεσμα όταν το πρόγραμμα εισέλθει ξανά στο βρόχο η πρόβλεψη να είναι ορθή.

- Η αρχική κατάσταση του αλγορίθμου μπορεί να καθοριστεί με στατική πρόβλεψη.

Επίδραση συνόλου εντολών στη διασωλήνωση



- Αρκετές εντολές ενός συνόλου εντολών μεταβάλλουν την τιμή των κωδικών συνθήκης (condition code flags) και το αποτέλεσμα της εκτέλεσης αρκετών εντολών καθορίζεται από την τιμή των κωδικών συνθήκης.

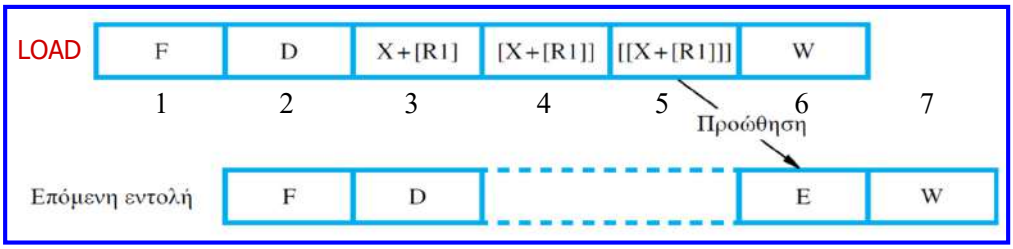
- Ένας μεταγλωττιστής βελτιστοποίησης για διασωληνωμένο επεξεργαστή έχει τη δυνατότητα να αναδιατάσσει τις εντολές ώστε να αποφύγει την αναστολή της διασωληνωμένης λειτουργίας, εξασφαλίζοντας ότι η αναδιάταξη δεν προκαλεί αλλαγές στα αποτελέσματα των υπολογισμών.

- Η εξάρτηση των εντολών από τις τιμές των κωδικών συνθήκης, ελαττώνει την ευελιξία του μεταγλωττιστή κατά την αναδιάταξη των εντολών.

- Γενικά, θα πρέπει να τηρούνται δύο κανόνες: οι κώδικες συνθήκης να επηρεάζονται από όσο το δυνατόν λιγότερες εντολές και ο μεταγλωττιστής να μπορεί να προσδιορίζει ποιες εντολές ενός προγράμματος επηρεάζουν τους κώδικες συνθήκης.

Επίδραση συνόλου εντολών στη διασωλήνωση

- Κατά την επιλογή σύνθετων τρόπων διευθυνσιοδότησης σε έναν επεξεργαστή με διασωλήνωση, θα πρέπει να εξετάζεται η έκταση στην οποία οι τρόποι αυτοί προκαλούν αναστολή της διασωλήνωσης.
- Για παράδειγμα κατά την εκτέλεση της εντολής LOAD (X(R1)), R2, μετά τον υπολογισμό της διεύθυνσης του ορίσματος στον κύκλο 3, ο επεξεργαστής θα πρέπει να προσπελάσει τη μνήμη δύο φορές (για ανάγνωση του περιεχομένου της θέσης X+[R1] στον κύκλο 4 και για ανάγνωση του περιεχομένου της θέσης [X+[R1]] στον κύκλο 5).
- Εάν ο καταχωρητής R2 αποτελεί όρισμα της επόμενης εντολής, προκαλείται αναστολή 2 κύκλων, ακόμη και με τη δυνατότητα προώθησης ορίσματος.

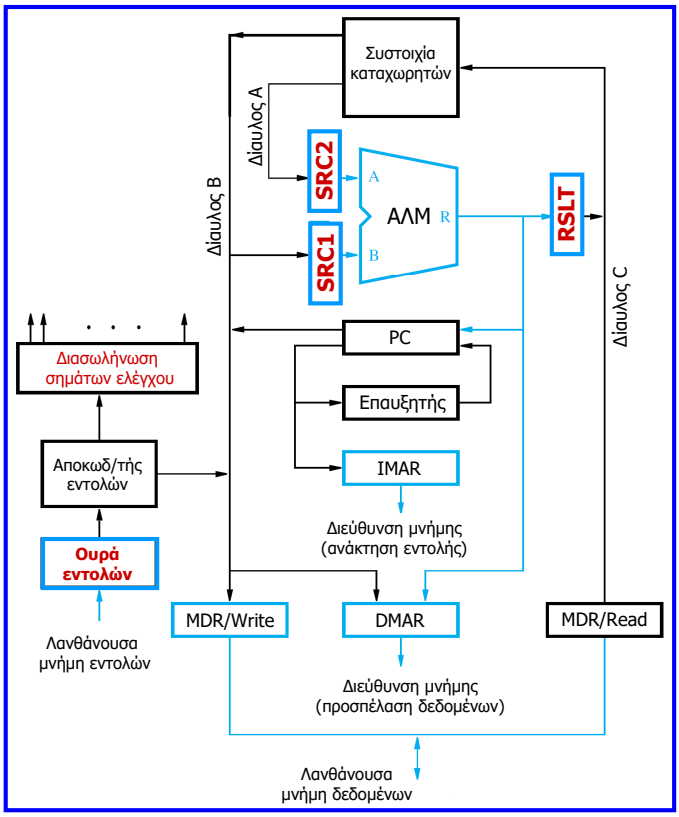


- Επομένως, οι σύνθετοι τρόποι διευθυνσιοδότησης πρέπει να ανάγονται σε απλούς με χρήση περισσότερων εντολών. Για παράδειγμα η ακολουθία εντολών ADD #X,R1,R2, LOAD (R2),R2, LOAD (R2),R2, απαιτεί ίδιο αριθμό κύκλων με την εντολή LOAD (X(R1)), R2.

Δομή επεξεργαστή για διασωληνωμένη λειτουργία

Για να καταστήσουμε κατάλληλη τη δομή επεξεργαστή τριών διαύλων για αποδοτική διασωληνωμένη λειτουργία, θα πρέπει να γίνουν οι εξής αλλαγές:

1. Διαφορετικές λανθάνουσες μνήμες εντολών και δεδομένων (δύο εκδόσεις του καταχωρητή MAR: IMAR, DMAR).
2. Σύνδεση του PC στον IMAR.
3. Η διεύθυνση στον DMAR λαμβάνεται από τη συστοιχία καταχωρητών ή από την ΑΛΜ για υποστήριξη σύνθετων τρόπων διευθυνσιοδότησης.
4. Δύο εκδόσεις MDR (εγγραφής και ανάγνωσης).
5. Προσθήκη καταχωρητών στις εισόδους και την έξοδο της ΑΛΜ και μηχανισμός προώθησης ορίσματος.
6. Αντικατάσταση του IR με ουρά εντολών και διασωλήνωση εξόδου (σήματα ελέγχου) αποκωδικοποιητή.



Απόδοση διασωλήνωσης

- Χρόνος εκτέλεσης προγράμματος: T = (N x S) / F (βασική εξίσωση απόδοσης, κεφάλαιο 1).
- N: αριθμός εντολών προγράμματος, S ή CPI: μέσος αριθμός κύκλων ρολογιού για την ολοκλήρωση μιας εντολής, F: συχνότητα ρολογιού (1 / περίοδος ρολογιού).
- Χρήσιμος δείκτης απόδοσης: παροχή εντολών (instruction throughput) ή πλήθος εντολών που εκτελούνται ανά sec (P).
- Για σειριακή εκτέλεση εντολών χωρίς επικάλυψη: $P_s = F / S$
- Για ιδανική διασωληνωμένη λειτουργία (S = 1): $P_p = F$
- Το μοναδικό πραγματικό μέτρο απόδοσης είναι ο συνολικός χρόνος εκτέλεσης προγράμματος, αφού η μεγαλύτερη παροχή εντολών δεν οδηγεί πάντα σε μεγαλύτερη απόδοση εάν ο συνολικός αριθμός εντολών του διασωληνωμένου επεξεργαστή για την υλοποίηση ενός προγράμματος είναι αρκετά μεγαλύτερος.
- Μία διασωλήνωση n βαθμίδων, θεωρητικά πολλαπλασιάζει την παροχή εντολών κατά n φορές (αυξάνεται η F αφού μειώνεται η διάρκεια του κύκλου ρολογιού).
- Εγείρεται όμως το ερώτημα σχετικά με την πρακτική και εφικτή τιμή του πλήθους n των βαθμίδων διασωλήνωσης, εάν λάβουμε υπόψη ότι η απόδοση της διασωλήνωσης επηρεάζεται από παράγοντες όπως η ποινή αστοχίας λανθάνουσας μνήμης ή η ποινή διακλάδωσης.

Επίδραση κινδύνων εντολών στην απόδοση

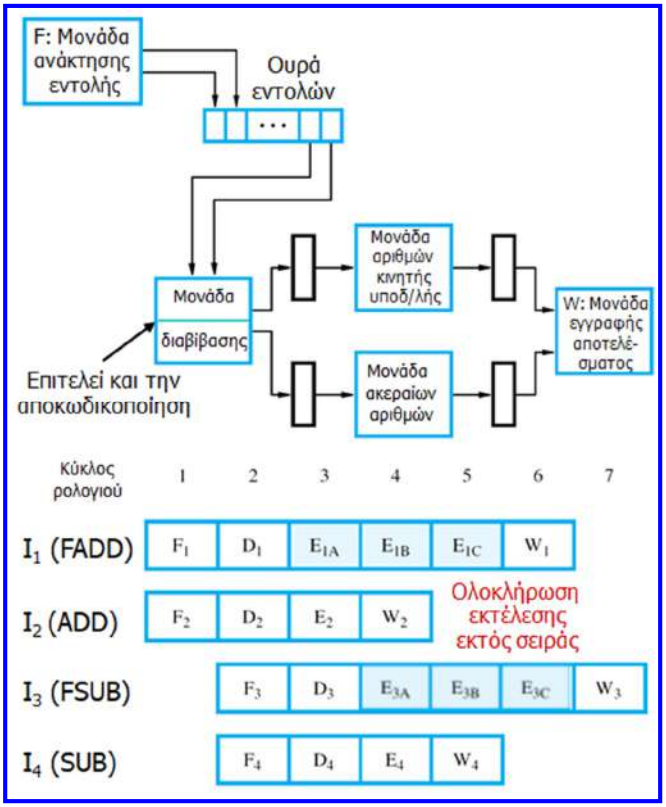
- Μία ποινή αστοχίας επιφέρεται στο σύνολό της στην περίπτωση κατά την οποία καθυστερεί η εκτέλεση της εντολής που ακολουθεί αμέσως μετά την εντολή που αναφέρεται στη μνήμη, ενώ ο επεξεργαστής αναμένει για την ολοκλήρωση της προσπέλασης μνήμης.
- Στα περισσότερα επεξεργαστικά συστήματα χρησιμοποιείται μεταγλωττιστής βελτιστοποίησης, ο οποίος προσπαθεί να αυξήσει την απόσταση μεταξύ δύο εξαρτημένων εντολών, τοποθετώντας άλλες εντολές μεταξύ τους.
- Αυτό δεν είναι πάντα δυνατό, αφού η τοποθέτηση αυτή των εντολών δεν θα πρέπει να επηρεάζει τη λογική ορθότητα του προγράμματος.
- Επίσης, σε επεξεργαστές που χρησιμοποιούν ουρά εντολών, η ποινή αστοχίας της λανθάνουσας μνήμης κατά τη διάρκεια ανάκτησης εντολών, ενδέχεται να έχει αρκετά μειωμένη επίδραση, καθώς οι επεξεργαστές αυτοί είναι σε θέση να διαβιβάσουν άλλες εντολές από την ουρά που διαθέτουν.

Πλήθος βαθμίδων και απόδοση διασωλήνωσης

- Όσο αυξάνεται το πλήθος βαθμίδων της διασωλήνωσης, αυξάνεται ταυτόχρονα και η πιθανότητα αναστολής της διασωλήνωσης λόγω του ότι περισσότερες εντολές βρίσκονται σε ταυτόχρονη επεξεργασία.
- Επομένως, εξαρτήσεις εντολών που βρίσκονται μακριά η μία από την άλλη μπορεί να προκαλέσουν αναστολή της διασωλήνωσης, καθώς επίσης και οι ποινές διακλάδωσης μπορεί να γίνουν σημαντικότερες.
- Για τους λόγους αυτούς, το κέρδος από την αύξηση του πλήθους των βαθμίδων διασωλήνωσης αρχίζει να ελαττώνεται, μέχρι που να μη δικαιολογεί πλέον το κόστος υλοποίησής του.
- Σε πολλούς επεξεργαστές ο χρόνος κύκλου ρολογιού επιλέγεται ώστε μία βασική λειτουργία της ΑΛΜ (π.χ. πρόσθεση) να ολοκληρώνεται σε έναν κύκλο ρολογιού (με τις πιο σύνθετες λειτουργίες να διαιρούνται σε αντίστοιχα τμήματα).
- Έτσι πολλοί διασωληνωμένοι επεξεργαστές χρησιμοποιούν 4 έως 12 βαθμίδες διασωλήνωσης, αλλά υπάρχουν και κάποιοι που διαθέτουν διασωλήνωση έως και 20 βαθμίδων με πολύ μικρό κύκλο ρολογιού, αλλά αυξημένη πολυπλοκότητα.
- Τέλος, το κέρδος απόδοσης που εισάγει η διασωλήνωση εξαρτάται από 3 βασικούς παράγοντες: το σύνολο εντολών του επεξεργαστή, τη σχεδίαση του υλικού που υποστηρίζει τη διασωληνωμένη λειτουργία και τη σχεδίαση του μεταγλωττιστή.

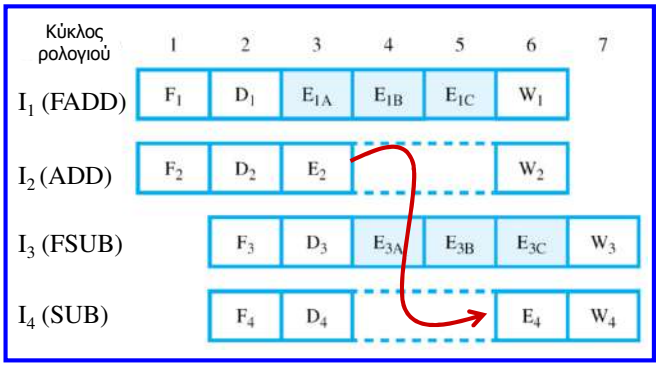
Υπερβαθμωτοί επεξεργαστές

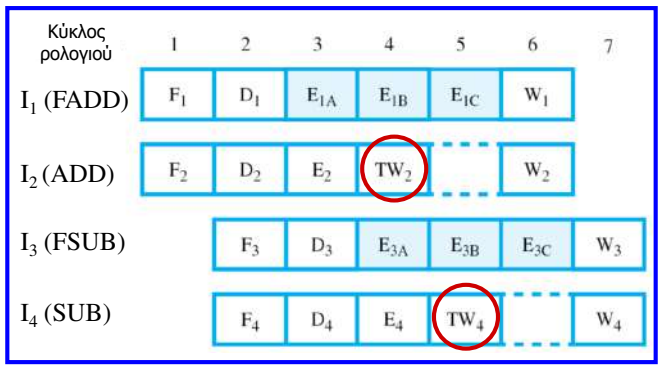
- Οι υπερβαθμωτοί επεξεργαστές (superscalar processors) περιλαμβάνουν πολλαπλές μονάδες επεξεργασίας και χειρίζονται παράλληλα πολλαπλές εντολές.
- Χρησιμοποιούν μηχανισμό πολλαπλής έκδοσης εντολών (multiple-issue) και μπορούν να πετύχουν παροχή εκτέλεσης εντολών (throughput) μεγαλύτερη του 1.
- Ο επεξεργαστής του διπλανού σχήματος ανακτά και αποκωδικοποιεί δύο εντολές (μία ακεραίων αριθμών και μία αριθμών κινητής υποδιαστολής) σε έναν κύκλο.
- Εκτελεί πράξεις ακεραίων αριθμών σε 1 κύκλο και αριθμών κινητής υποδιαστολής σε 3 (διασωλήνωση 3 βαθμίδων).
- Ο μεταγλωττιστής επιλέγει τις εντολές και τη διάταξή τους ώστε να αποφύγει δομικούς κινδύνους και να αξιοποιεί το διαθέσιμο υλικό στο μέγιστο βαθμό.



Υπερβαθμωτοί επεξεργαστές

- Αν και οι εντολές εκδίδονται με την ίδια σειρά που συναντώνται στο υπό εκτέλεση πρόγραμμα, είναι πιθανό η εκτέλεσή τους να ολοκληρωθεί εκτός σειράς.
- Αυτό μπορεί να αποφευχθεί με καθυστέρηση εγγραφής του αποτελέσματος της εντολής που ολοκληρώνεται εκτός σειράς.
- Μπορεί λοιπόν να εισαχθεί καθυστέρηση, τέτοια ώστε η φάση W_2 να διεξαχθεί στον 6ο κύκλο, με αποτέλεσμα όμως η μονάδα ακέραιων αριθμών να μη μπορεί να δεχθεί την I_4 πριν τον 6ο κύκλο.
- Εναλλακτικά, τα αποτελέσματα των μονάδων επεξεργασίας εγγράφονται σε προσωρινούς καταχωρητές (φάση TW), που μπορούν να χρησιμοποιηθούν από τις επόμενες εντολές, και αργότερα στους τελικούς καταχωρητές προορισμού, έτσι ώστε να μην ανακόπτεται η εκτέλεση επόμενων εντολών.





Μέρος Γ: Δίκτυα υπολογιστών

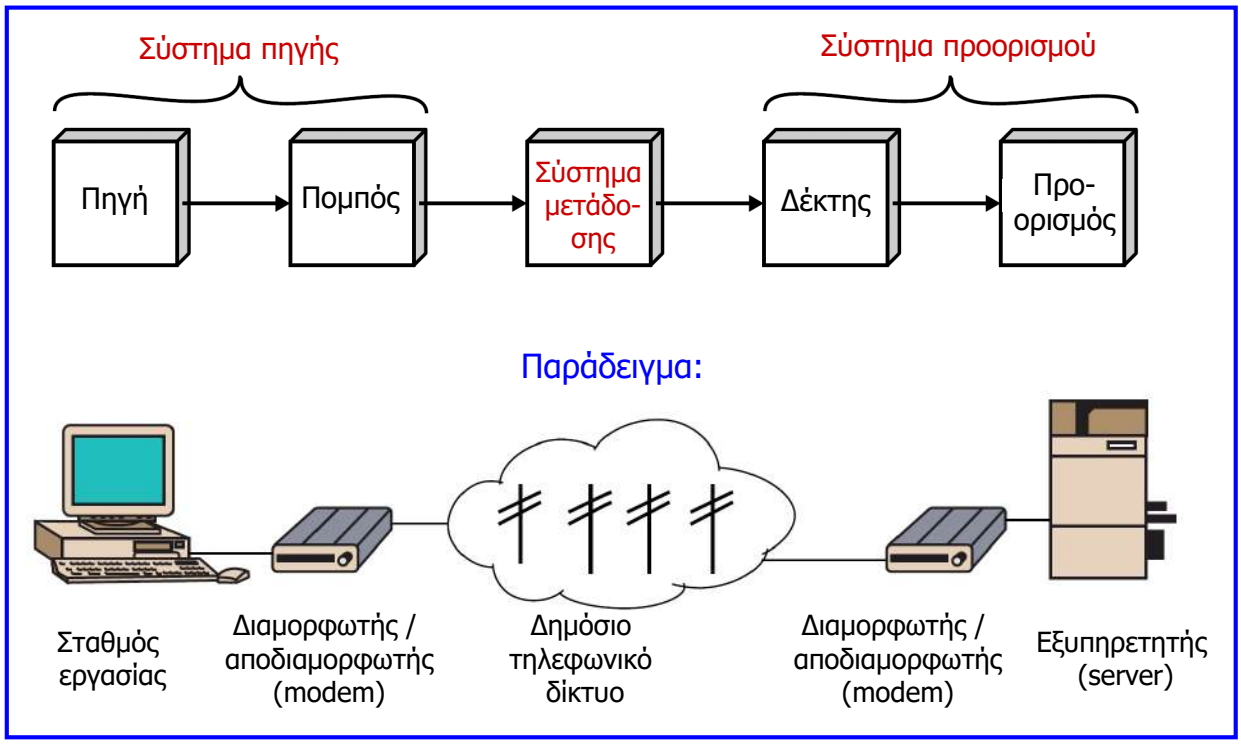
- Κεφάλαιο 1: βασικές έννοιες και εφαρμογές δικτύων, ταξινόμηση δικτύων, δίκτυα μεταγωγής, αξιοπιστία και απόδοση δικτύου, βασικές επικοινωνιακές διεργασίες.
- Κεφάλαιο 2: ιεραρχική οργάνωση δικτύων και πρωτόκολλα, μοντέλα αναφοράς OSI και TCP/IP, λειτουργίες επιπέδων των μοντέλων αναφοράς.
- Κεφάλαιο 3: πρωτόκολλο ελέγχου ζεύξης δεδομένων (DLC), έλεγχος ροής, έλεγχος σφαλμάτων, τεχνικές αυτόματης αίτησης επανάληψης (ARQ) και απόδοσή τους, κώδικες διόρθωσης και ανίχνευσης σφαλμάτων.
- Κεφάλαιο 4: τοπικά δίκτυα, πρωτόκολλα τοπικών δικτύων, τεχνικές πολλαπλής πρόσβασης και απόδοσή τους, δίκτυο Ethernet.
- Κεφάλαιο 5: πρωτόκολλα διαδικτύωσης, πρωτόκολλο IPv4, κερματισμός και επανασυναρμολόγηση πακέτων, έλεγχος ροής και σφαλμάτων, δρομολόγηση, αλγόριθμος Dijkstra, διευθυνσιοδότηση, πρωτόκολλο ICMP, πρωτόκολλο IPv6.
- Κεφάλαιο 6: πρωτοκόλλα μεταφοράς, διευθυνσιοδότηση, πολυπλεξία, έλεγχος ροής, έλεγχος σφαλμάτων και αναμετάδοση, εγκαθίδρυση και τερματισμός σύνδεσης, πρωτόκολλο TCP, έλεγχος συμφόρησης δικτύου, πρωτόκολλο UPD.
- Κεφάλαιο 7: εφαρμογές διαδικτύου, παγκόσμιος ιστός, πρωτόκολλα DNS και HTTP, απομακρυσμένη μεταφορά αρχείων (FTP), σύνδεση σε απομακρυσμένο εξυπηρετητή (TELNET, SSH), ηλεκτρονικό ταχυδρομείο, πρωτόκολλα SMTP, POP, IMAP και MIME.

Γενικό μοντέλο επικοινωνίας

Ένα επικοινωνιακό σύστημα έχει ως βασικό σκοπό την ανταλλαγή δεδομένων μεταξύ δύο πλευρών και περιλαμβάνει τα παρακάτω βασικά στοιχεία:

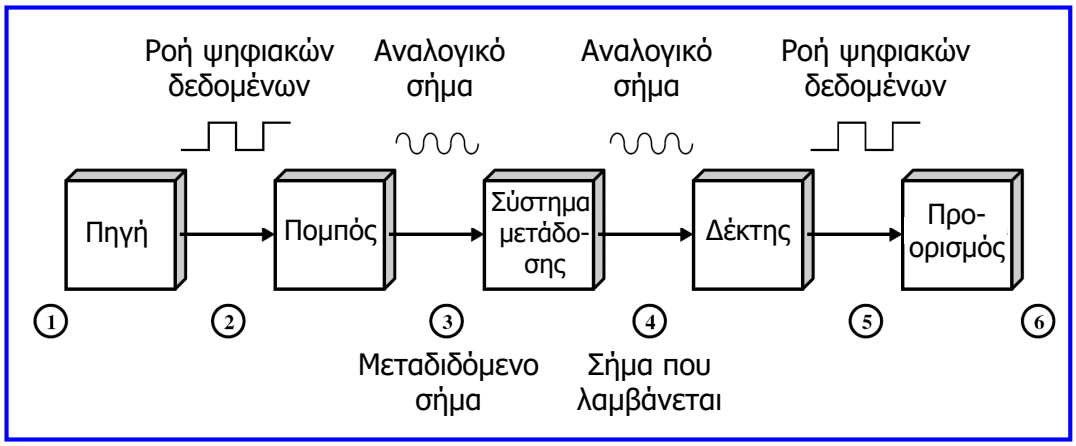
- Πηγή (source): συσκευή που παράγει τα δεδομένα που πρόκειται να μεταδοθούν.
- Πομπός (transmitter): διαμορφώνει και κωδικοποιεί την πληροφορία με τέτοιο τρόπο ώστε να παράγει ηλεκτρομαγνητικά σήματα τα οποία θα μπορούν να μεταδοθούν μέσω κάποιου συστήματος μετάδοσης.
- Σύστημα μετάδοσης (transmission system): μπορεί να είναι μία απλή γραμμή μετάδοσης ή ένα πολύπλοκο δίκτυο που συνδέει την πηγή με τον προορισμό.
- Δέκτης (receiver): δέχεται το σήμα από το σύστημα μετάδοσης και το μετατρέπει σε τέτοια μορφή ώστε να μπορεί να είναι κατανοητό από τη συσκευή προορισμού.
- Προορισμός (destination): συσκευή που λαμβάνει τα δεδομένα από το δέκτη.

Γενικό διάγραμμα επικοινωνίας



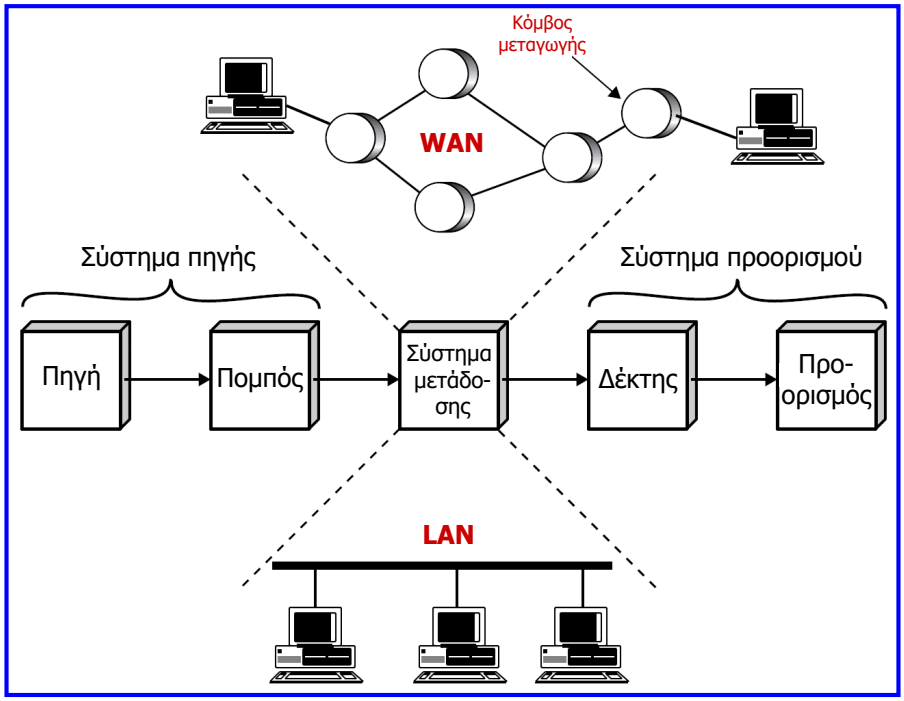
Επικοινωνία δεδομένων

- Το σήμα που λαμβάνεται (received signal) μπορεί να διαφέρει από το μεταδιδόμενο (transmitted signal) και ο δέκτης θα προσπαθήσει να υπολογίσει το αυθεντικό βασισμένος στο λαμβανόμενο σήμα και στη γνώση που έχει για το μέσο μετάδοσης.
- Το σύστημα προορισμού προσπαθεί να προσδιορίσει τυχόν σφάλματα (για να ζητήσει επαναμετάδοση), έτσι ώστε η τελική πληροφορία (6) να είναι ακριβές αντίγραφο της αυθεντικής πληροφορίας εισόδου (1).



Δικτυακή επικοινωνία δεδομένων

- Δεν είναι συνήθως πρακτικό δύο συσκευές να είναι άμεσα συνδεδεμένες (point-to-point), διότι οι συσκευές μπορεί να είναι πολύ απομακρυσμένες η μία από την άλλη ή ο αριθμός των συσκευών που πρέπει να συνδεθούν να είναι αρκετά μεγάλος, με αποτέλεσμα ανέφικτα μεγάλο αριθμό συνδέσεων μεταξύ τους.
- Λύση αποτελεί η σύνδεση κάθε συσκευής σε ένα επικοινωνιακό δίκτυο δεδομένων.



- Δύο κύριες κατηγορίες επικοινωνιακών δικτύων: δίκτυα ευρείας περιοχής (WAN, wide area networks) και τοπικά δίκτυα (LAN, local area networks).

Σύγχρονες εφαρμογές δικτύων

- Επιχειρησιακά δίκτυα:
  - ✓ Τηλεφωνία, πολυμέσα.
  - ✓ Πρόσβαση στο διαδίκτυο.
  - ✓ Πρόσβαση σε επιχειρησιακά δεδομένα.
  - ✓ Υποστήριξη επιχειρηματικών διεργασιών (business processes).
  - ✓ Δικτύωση εργαζομένων, χρηστών (corporate intranet, VPNs, corporate data centers).
  - ✓ Δικτύωση με συνεργάτες, προμηθευτές (partner networks, supply chain networks).
  - ✓ Δίκτυα υπολογιστικού νέφους (cloud networks), η πιο πρόσφατη εξέλιξη.
- Οικιακά Δίκτυα:
  - ✓ Οικιακός αυτοματισμός (εξειδικευμένα δίκτυα που διασυνδέουν έξυπνες οικιακές συσκευές, συστήματα ασφαλείας).
  - ✓ Πρόσβαση σε πληροφορίες (κοινή πρόσβαση στο διαδίκτυο, κοινή χρήση περιφερειακών συσκευών, κοινή χρήση μέσων αποθήκευσης για πρόσβαση σε εφαρμογές και δεδομένα).
  - ✓ Σύγχρονες εφαρμογές επικοινωνίας πραγματικού χρόνου (δικτυακά παιχνίδια πολλαπλών παικτών, δικτυακά συστήματα ψηφιακού ήχου και βίντεο).

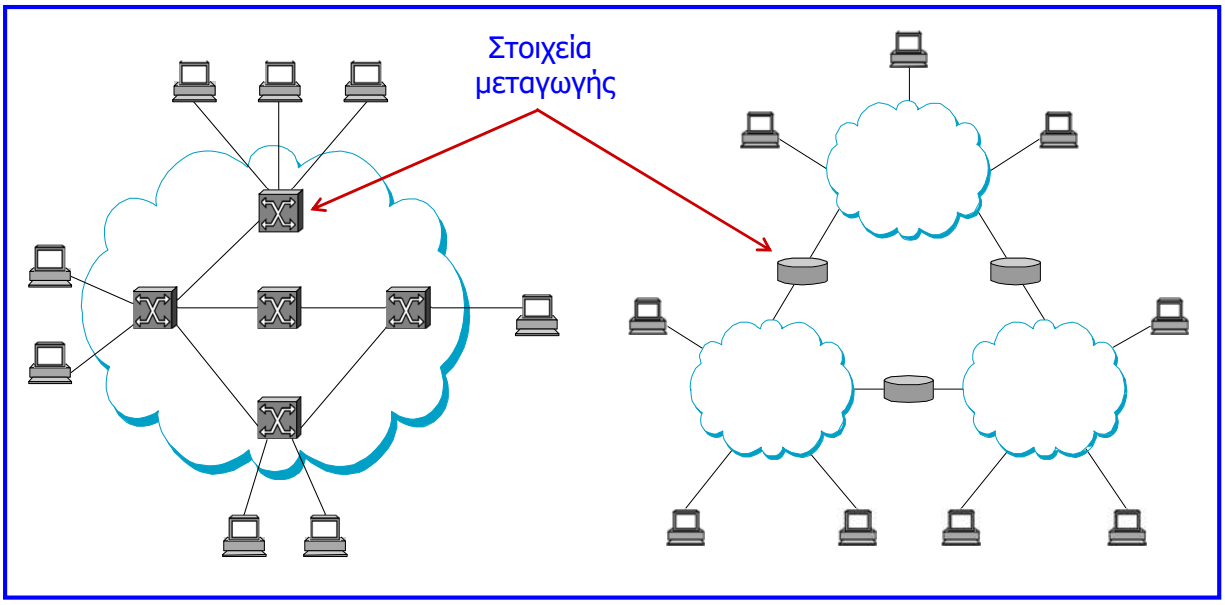
Βασικοί όροι δικτύων

- Δικτύωση (networking, data networking, data communication): διασύνδεση υπολογιστών ή γενικότερα συσκευών σε ένα δίκτυο και αποστολή πληροφοριών (δεδομένων) μέσα από αυτό.
- Δίκτυο επικοινωνιών (network, data network, communication network): εγκατάσταση που παρέχει υπηρεσία μεταφοράς πληροφοριών (δεδομένων) μεταξύ των συνδεδεμένων υπολογιστών (συσκευών).
- Κυκλοφορία (traffic): πληροφορίες (δεδομένα) που μεταδίδονται μέσω ενός δικτύου.
- Εύρος ζώνης (digital bandwidth): πλήθος πληροφοριών (δεδομένων) που μπορεί να μεταφερθεί μέσα σε δεδομένο χρόνο ενός δευτερολέπτου (bps, kbps, Mbps).
- Στοιχεία μεταγωγής (switching elements) ή κόμβοι μεταγωγής (switch nodes) ή ενδιάμεσα συστήματα (intermediate systems) ή κέντρα μεταγωγής δεδομένων (data switching exchanges): ειδικά υπολογιστικά συστήματα για τη σύνδεση δύο ή περισσότερων γραμμών μετάδοσης (επιλέγουν την εξερχόμενη γραμμή που θα μεταδώσουν τα εισερχόμενα δεδομένα).
- Γραμμές μετάδοσης (transmission lines) ή κυκλώματα (circuits) ή κανάλια (channels) ή ζεύξεις (trunks) ή συνδέσεις (links): στοιχεία του δικτύου που μεταφέρουν τα δεδομένα.
- Διεπαφή δικτύου (network interface card, NIC) ή προσαρμογέας δικτύου: στοιχείο υλικού μέσω του οποίου μια συσκευή (π.χ. υπολογιστής) συνδέεται στους κόμβους μεταγωγής.

Βασικοί όροι δικτύων

- Κεντρικός υπολογιστής (host): συσκευή του δικτύου, στην οποία εκτελούνται οι εφαρμογές των χρηστών.
- Τερματικό σύστημα (end system): συσκευές δικτύου που χρησιμοποιούν οι χρήστες.
- Πακέτο (packet): βασική μονάδα πληροφορίας που μεταφέρεται σε ένα δίκτυο. Τα δεδομένα διαιρούνται σε μικρές ενότητες δεδομένων (πακέτα), οι οποίες στέλνονται μεμονωμένα (δίκτυα μεταγωγής πακέτων, packet switching networks).
- Πρωτόκολλο (protocol): προκαθορισμένος τρόπος ή σύνολο κανόνων με βάση τους οποίους οι συσκευές (υπολογιστές) ενός δικτύου επικοινωνούν μεταξύ τους, προκειμένου να συντονιστούν και να ανταλλάξουν πληροφορίες (δεδομένα).
- Διαδίκτυο (internet): παγκόσμιο σύστημα διασυνδεδεμένων δικτύων επικοινωνιών (WAN, LAN κ.ά.) μέσω στοιχείων μεταγωγής.
- Ενδοδίκτυο (intranet): ιδιωτικό δίκτυο υπολογιστών (εταιρίας ή οργανισμού) που παρέχει στους χρήστες του τις βασικές εφαρμογές διαδικτύου (ηλεκτρονικό ταχυδρομείο, πρόσβαση στον παγκόσμιο ιστό www, FTP κ.ά.). Αποτελεί μικρή, ιδιωτική έκδοση του διαδικτύου που χρησιμοποιείται αποκλειστικά από έναν οργανισμό και λειτουργεί για εσωτερικούς σκοπούς με σύνδεση στο διαδίκτυο ή απομονωμένα (ανεξάρτητα).

Δομές δικτύων



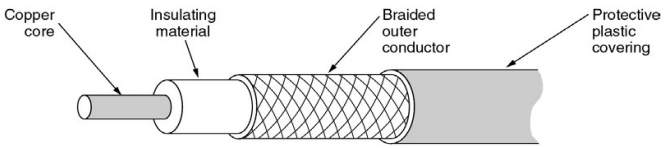
Ταξινόμηση δικτύων ως προς το μέσο μετάδοσης

- Ενσύρματα ή καλωδιακά: δίκτυα οπτικών ινών, χάλκινα καλωδιακά δίκτυα.
- Ασύρματα: επικοινωνία μέσω ηλεκτρομαγνητικών κυμάτων, επίγεια δίκτυα (δίκτυα ραδιοεπικοινωνιών, δίκτυα μικροκυματικών ζεύξεων), δορυφορικά δίκτυα.

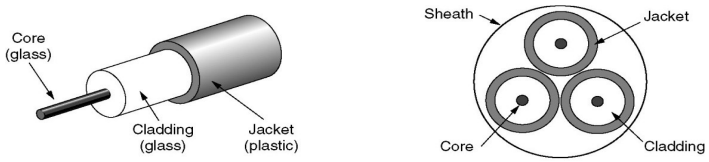
Επίγειες ασύρματες επικοινωνίες (WiFi, WiMAX, 4G…)

Δορυφορικές επικοινωνίες

Συνεστραμμένο ζεύγος καλωδίων (twisted pair)

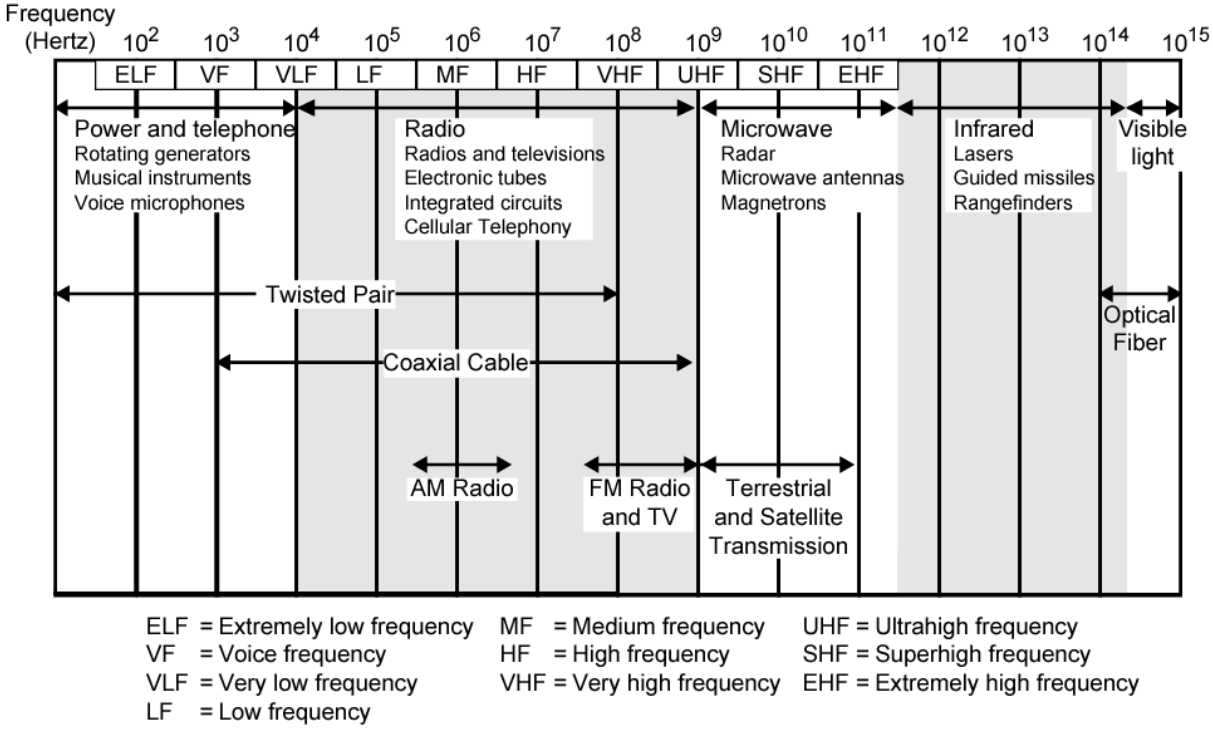


Ομοαξονικό καλώδιο (co-axial)



Οπτικές ίνες

Ηλεκτρομαγνητικό φάσμα

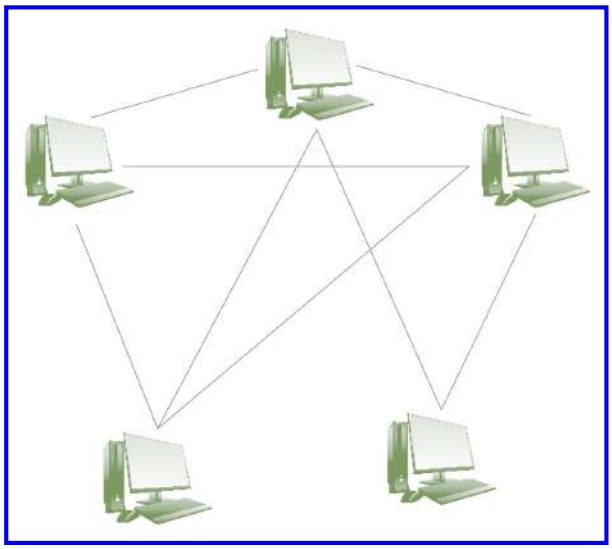


ELF = Extremely low frequency
VF = Voice frequency
VLF = Very low frequency
LF = Low frequency
MF = Medium frequency
HF = High frequency
VHF = Very high frequency
UHF = Ultrahigh frequency
SHF = Superhigh frequency
EHF = Extremely high frequency

Ταξινόμηση δικτύων ως προς το είδος σύνδεσης

Σύνδεση σημείου προς σημείο (point-to-point connection): συνδέει δύο μόνο κόμβους κάθε φορά και πραγματοποιείται με απευθείας σύνδεση των κόμβων με κάποια γραμμή επικοινωνίας.

- Η σύνδεση μπορεί να γίνεται και τμηματικά, όταν δύο κόμβοι δεν επικοινωνούν απευθείας μεταξύ τους αλλά μέσω άλλων κόμβων.
- Ενδέχεται, η επικοινωνία μεταξύ δύο κόμβων να αλλάζει διαδρομές για διάφορους λόγους και γι' αυτές τις περιπτώσεις, έχουν αναπτυχθεί τεχνικές που καθορίζουν τη δρομολόγηση των δεδομένων από τον αποστολέα στον παραλήπτη.
- Το διαδίκτυο, τα δίκτυα δεδομένων ευρείας περιοχής και το τηλεφωνικό δίκτυο είναι δίκτυα με συνδέσεις σημείου προς σημείο.



Ταξινόμηση δικτύων ως προς το είδος σύνδεσης

Σύνδεση ευρείας εκπομπής (broadcasting): συνδέει δύο ή περισσότερους κόμβους ταυτόχρονα.

- Τα δίκτυα αυτής της κατηγορίας διαθέτουν ένα μόνο κανάλι επικοινωνίας, το οποίο μοιράζονται όλοι οι κόμβοι του δικτύου (κοινό μέσο μετάδοσης).
- Κάθε μήνυμα που αποστέλλεται, παραλαμβάνεται από όλους τους χρήστες του δικτύου (σύνδεση σημείου με πολλαπλά σημεία, point-to-multipoint connection).
- Όταν στέλνεται ένα μήνυμα από έναν κόμβο σε κάποιον άλλο, είναι εφοδιασμένο με τη διεύθυνση του παραλήπτη και λαμβάνεται από όλους τους κόμβους του δικτύου.
- Όταν ένας κόμβος δεχτεί ένα μήνυμα, ελέγχει τη διεύθυνση παραλήπτη και αν είναι έγκυρη, τότε καταχωρεί το μήνυμα, διαφορετικά το αγνοεί.
- Για τον καθορισμό της χρήσης του μέσου μετάδοσης από τους κόμβους, έχουν αναπτυχθεί αρκετές τεχνικές.

- Δίκτυα Η/Υ, δίκτυα ραδιοφώνου και τηλεόρασης με δέκτες χωρίς δυνατότητα εκπομπής.

Ταξινόμηση ως προς τη γεωγραφική κάλυψη

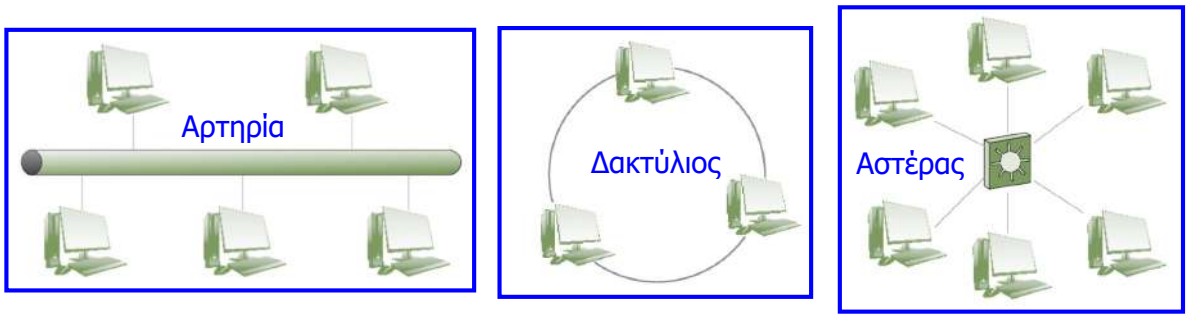
- Τοπικά δίκτυα (local area networks, LAN): δίκτυα που η γεωγραφική τους κάλυψη είναι περιορισμένη και δεν εκτείνονται πέραν των 100 m.
  - ✓ Υψηλός ρυθμός μετάδοσης (από 1 Mbps έως μερικά Gbps).
  - ✓ Τα φυσικά μέσα μετάδοσης των τοπικών δικτύων είναι συνήθως ιδιόκτητα.
  - ✓ Τοπικό δίκτυο μπορεί να είναι το δίκτυο κάποιων δωματίων, ενός κτηρίου ή γειτονικών κτιρίων, είναι συνήθως δίκτυο ευρείας εκπομπής που συγκροτείται ανάμεσα σε προσωπικούς υπολογιστές, χωρίς βέβαια να αποκλείεται ο συνδυασμός μεσαίων και μεγάλων υπολογιστών.
  - ✓ Φυσικό μέσο μετάδοσης: χάλκινο καλώδιο (ομοαξονικό, συνεστραμμένο ζεύγος καλωδίων), οπτική ίνα, ασύρματη ζεύξη.
- Οι νεότερες εξελίξεις έχουν οδηγήσει και σε δίκτυα μικρότερης έκτασης (συνήθως για δικτύωση αισθητήρων):
  - ✓ Personal area networks (PAN),
  - ✓ Body area networks (BAN).

Ταξινόμηση ως προς τη γεωγραφική κάλυψη

- Μητροπολιτικά δίκτυα (metropolitan area networks, MAN): δίκτυα με γεωγραφική εμβέλεια μεταξύ των τοπικών και των δικτύων ευρείας περιοχής.
  - ✓ Καλύπτουν την ανάγκη για δημιουργία δικτύωσης σε μεγάλες γεωγραφικές αποστάσεις, διατηρώντας χαρακτηριστικά των τοπικών δικτύων, κυρίως όσον αφορά το ρυθμό μετάδοσης δεδομένων.
  - ✓ Ρυθμός μετάδοσης από 56 Kbps έως 100 Mbps.
  - ✓ Μήκος εγκατεστημένης καλωδίωσης έως 200 km
  - ✓ Τα δίκτυα αυτής της κατηγορίας μπορεί να είναι ιδιόκτητα ή δημόσια.
- Δίκτυα ευρείας περιοχής (wide area networks, WAN): δίκτυα γεωγραφικής εμβέλειας μεγαλύτερης των 200 km και γενικότερα όταν οι εμπλεκόμενοι υπολογιστές βρίσκονται σε πολύ μεγάλες αποστάσεις μεταξύ τους.
  - ✓ Πλήθος κόμβων (υπολογιστών) από 10000 έως αρκετά εκατομμύρια.
  - ✓ Υψηλοί ρυθμοί μετάδοσης (> 1 Gbps) με διάφορα μέσα μετάδοσης.
  - ✓ Τα δίκτυα αυτά είναι συνήθως διεθνή (παρέχονται από δημόσιους φορείς), όπως τηλεφωνικά δίκτυα, διαδίκτυο, δημόσια δίκτυα μεταφοράς δεδομένων.
  - ✓ Στα δίκτυα ευρείας περιοχής μπορούν να συνυπάρχουν μικρότερα δίκτυα ευρείας περιοχής, μητροπολιτικά δίκτυα, τοπικά δίκτυα και αυτόνομοι υπολογιστές.

Ταξινόμηση ως προς την τοπολογία

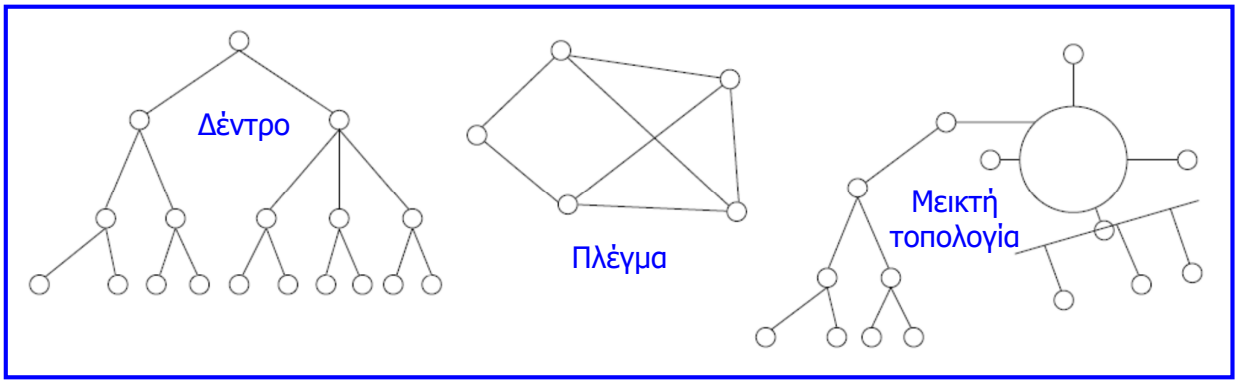
- Τοπολογία δικτύου είναι η φυσική διάταξη των καλωδίων που συνδέουν τους κόμβους του δικτύου.
- Τοπολογία διαύλου ή αρτηρίας (bus): οι κόμβοι του δικτύου συνδέονται με καλώδιο του οποίου τα άκρα είναι ανοικτά, χρησιμοποιείται μόνο σε τοπικά δίκτυα, με βασικό μειονέκτημα την μη λειτουργία του σε περίπτωση διακοπής του μέσου (καλωδίου).



- Τοπολογία δακτυλίου (ring): οι κόμβοι του δικτύου συνδέονται με ένα μόνο καλώδιο του οποίου τα άκρα είναι ενωμένα μεταξύ τους.
- Τοπολογία αστέρα (star): ένας κατανεμητής ή ένας κεντρικός υπολογιστής συνδέεται με μια μόνιμη γραμμή απευθείας με τους άλλους κόμβους (υπολογιστές) του δικτύου και η σύνδεση δύο κόμβων γίνεται μόνο μέσω του κατανεμητή.

Ταξινόμηση ως προς την τοπολογία

- Τοπολογία δέντρου (tree): παράγωγο της τοπολογίας αστέρα με σχήμα δέντρου στο οποίο η κύρια ευθύνη της ρίζας είναι να μοιράζεται ιεραρχικά τους κόμβους των κλάδων του δέντρου.
- Τοπολογία πλέγματος (mesh): τοπολογία χωρίς καθορισμένη μορφή (σχήμα).
- Τοπολογία μεικτή (mixed): είναι η συνένωση (συνύπαρξη) διάφορων από τις τοπολογίες που προαναφέρθηκαν.



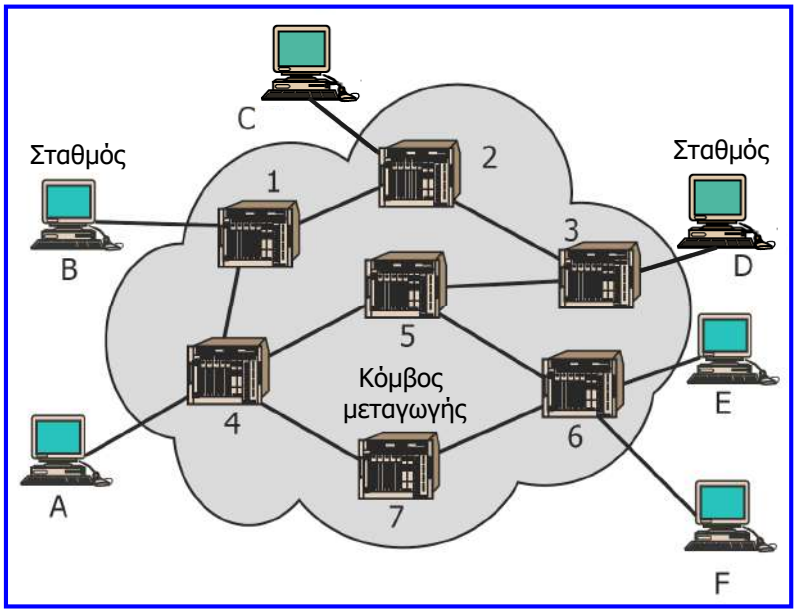
Ταξινόμηση ως προς την τεχνολογία μετάδοσης

Βασικές τεχνολογίες μετάδοσης (τεχνικές διασύνδεσης) για περισσότερους από δύο κόμβους:

- μεταγωγή κυκλώματος (circuit switching),
- μεταγωγή πακέτου (packet switching) ή τεχνική αποθήκευσης και προώθησης (store and forward).
- ασύγχρονος τρόπος μεταφοράς (asynchronous transfer mode, ATM).

Δίκτυα μεταγωγής

Στα δίκτυα μεταγωγής (συνήθως ευρείας περιοχής) χρησιμοποιούνται δύο τεχνολογίες μετάδοσης: μεταγωγή κυκλώματος (circuit switching) και μεταγωγή πακέτου (packet switching), οι οποίες διαφέρουν στον τρόπο με τον οποίο οι κόμβοι μεταφέρουν πληροφορία από την πηγή στον προορισμό.



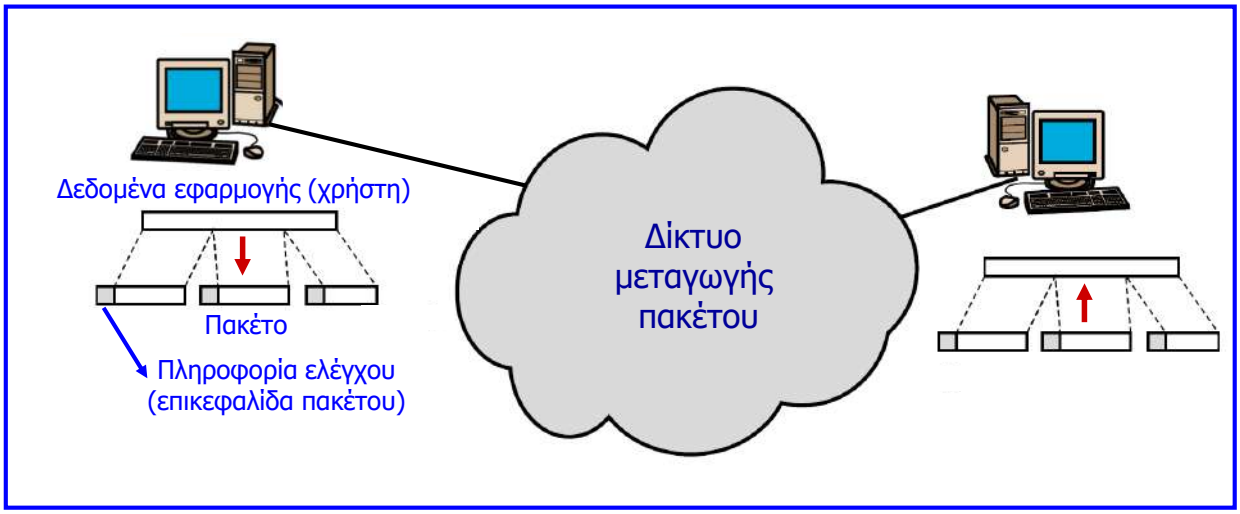
Μεταγωγή κυκλώματος

- Για την επικοινωνία μεταξύ δύο σταθμών σε ένα δίκτυο μεταγωγής κυκλώματος δημιουργείται ένα αποκλειστικό μονοπάτι ανάμεσα στους δύο σταθμούς, το οποίο είναι μία ακολουθία από ζεύξεις ανάμεσα σε κόμβους μεταγωγής του δικτύου.
- Η επικοινωνία μέσω μεταγωγής κυκλώματος περιλαμβάνει τρεις φάσεις:
  - ✓ Αποκατάσταση κυκλώματος: πριν μεταδοθεί οποιοδήποτε σήμα, πρέπει να αποκατασταθεί ένα κύκλωμα από σταθμό σε σταθμό. Ένας σταθμός αιτείται σε έναν κόμβο σύνδεση με έναν άλλο σταθμό και στη συνέχεια οι κόμβοι επιλέγουν το μονοπάτι (κύκλωμα που θα ακολουθηθεί).
  - ✓ Μεταφορά δεδομένων: μετάδοση και διάδοση της πληροφορίας μεταξύ δύο σταθμών (η σύνδεση γενικά είναι αμφίδρομη).
  - ✓ Αποσύνδεση κυκλώματος: έπειτα από κάποια περίοδο μεταφοράς δεδομένων, η σύνδεση τερματίζεται συνήθως με πρωτοβουλία ενός από τους δύο σταθμούς και αφού σταλούν κατάλληλα σήματα στους εμπλεκόμενους κόμβους ώστε να ελευθερώσουν τους πόρους των οποίων τη χρήση κατείχαν.
- Κατά την αποκατάσταση πρέπει να δεσμευτεί χωρητικότητα καναλιού ανάμεσα σε κάθε ζεύγος κόμβων του μονοπατιού και κάθε κόμβος πρέπει να έχει διαθέσιμη εσωτερική χωρητικότητα μεταγωγής για να χειριστεί την αιτούμενη σύνδεση.
- Οι κόμβοι μεταγωγής θα πρέπει να διαθέτουν «νοημοσύνη», ώστε να επινοήσουν την κατάλληλη διαδρομή μέσω του δικτύου.

Μεταγωγή πακέτου

- Στη μεταγωγή κυκλώματος που σχεδιάστηκε για κίνηση φωνής (τηλεφωνία), οι πόροι του δικτύου αφιερώνονται σε μία κλήση και το κύκλωμα που προκύπτει έχει υψηλό βαθμό χρήσης, αφού τον περισσότερο χρόνο της σύνδεσης υπάρχει συνομιλία.
- Για κίνηση δεδομένων όμως, συνήθως τον περισσότερο χρόνο το κύκλωμα παραμένει αδρανές με αποτέλεσμα η μεταγωγή κυκλώματος να μην είναι αποδοτική.
- Η τεχνική της μεταγωγής πακέτου σχεδιάστηκε για να παρέχει πιο αποδοτική λειτουργία από τη μεταγωγή κυκλώματος για κίνηση δεδομένων.
- Τα δεδομένα μεταδίδονται σε μικρά πακέτα με τυπικό μέγεθος (π.χ. 1000 οκτάδες δυαδικών ψηφίων, bytes), με τα μεγαλύτερα μηνύματα να τεμαχίζονται σε πακέτα και κάθε πακέτο να περιέχει ένα τμήμα των δεδομένων του χρήστη και πληροφορία ελέγχου, η οποία περιλαμβάνει οπωσδήποτε τη διεύθυνση του σταθμού προορισμού.
- Σε κάθε κόμβο, τα πακέτα λαμβάνονται, αποθηκεύονται προσωρινά και κατόπιν στέλνονται στον επόμενο κόμβο (τεχνική αποθήκευσης και προώθησης, store and forward).

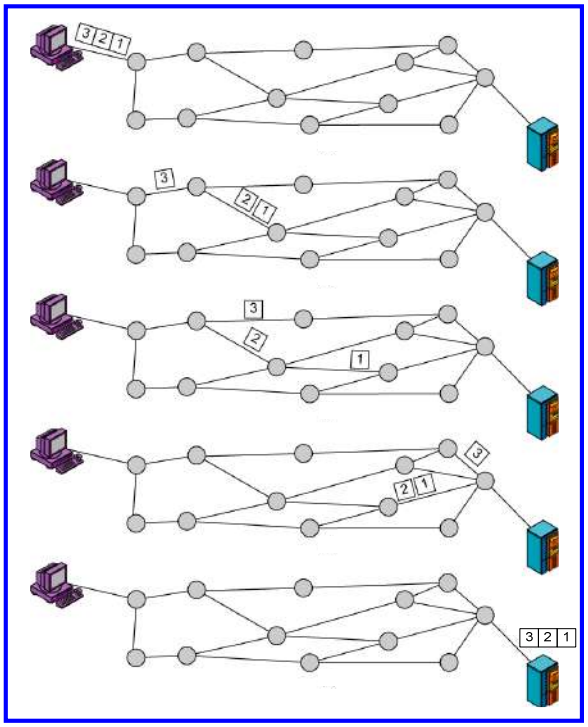
Μεταγωγή πακέτου



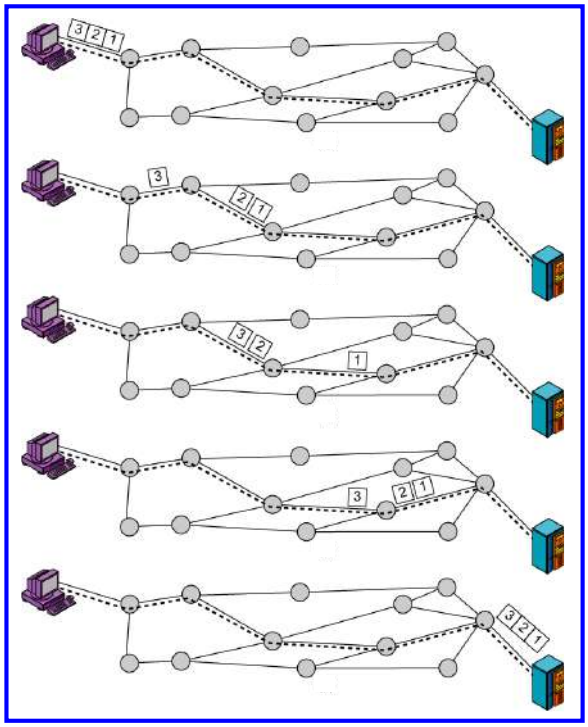
Μεταγωγή πακέτου

- Για το χειρισμό των πακέτων χρησιμοποιούνται δύο προσεγγίσεις:
- Στην προσέγγιση του αυτόνομου πακέτου (datagram):
  - ✓ Κάθε πακέτο αντιμετωπίζεται ξεχωριστά και λόγω του ότι τα πακέτα δεν ακολουθούν την ίδια διαδρομή στο δίκτυο, είναι πιθανό να παραδοθούν στον προορισμό (τη διεύθυνση του οποίου περιέχουν τα πακέτα) με διαφορετική σειρά από εκείνη με την οποία στάλθηκαν.
  - ✓ Επίσης, είναι πιθανό κάποιο ή κάποια πακέτα να καταστραφούν στο δίκτυο (π.χ. σε κόμβο με βλάβη μπορεί να χαθούν πακέτα που υπάρχουν στην ουρά του).
  - ✓ Ο δέκτης αναδιατάσσει τα πακέτα ή ανιχνεύει την απώλεια πακέτου και το ανακτά.
- Στην προσέγγιση νοητού κυκλώματος (virtual circuit):
  - ✓ Πριν τη μετάδοση ενός πακέτου αποκαθίσταται μία προσχεδιασμένη διαδρομή.
  - ✓ Αρχικά στέλνονται ειδικά πακέτα αίτησης κλήσης (call request) και αποδοχής κλήσης (call accept) για να ζητηθεί και να εγκατασταθεί η σύνδεση.
  - ✓ Κάθε πακέτο περιέχει ταυτότητα νοητού κυκλώματος αντί για διεύθυνση προορισμού.
  - ✓ Δεν απαιτούνται αποφάσεις δρομολόγησης για κάθε πακέτο και μία σύνδεση τερματίζεται με ειδικό πακέτο αίτησης τερματισμού (clear request).

Μεταγωγή πακέτου



Προσέγγιση αυτόνομου πακέτου (datagram)



Προσέγγιση νοητού κυκλώματος (virtual circuit)

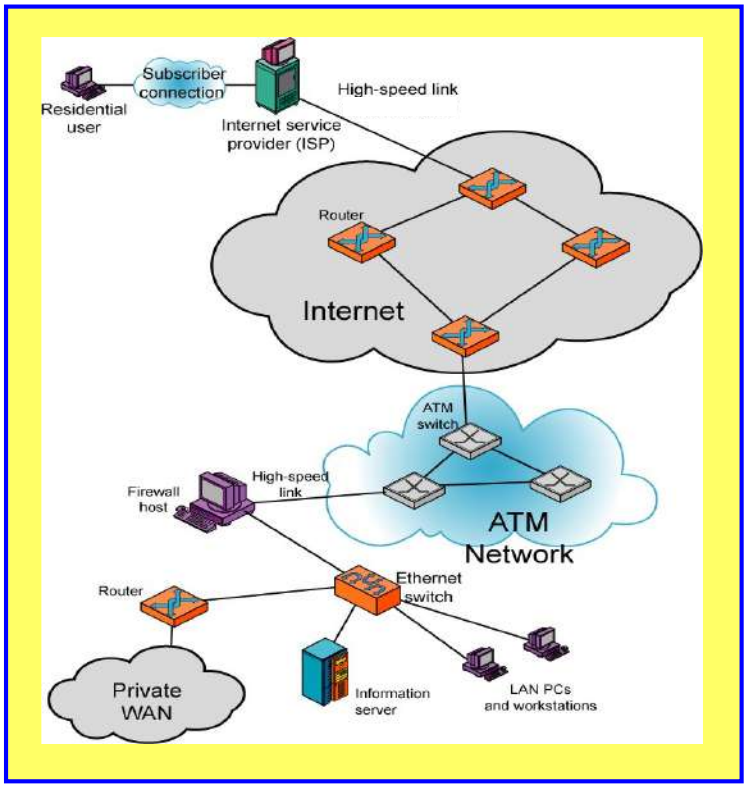
Μεταγωγή πακέτου

Διαφορές προσεγγίσεων χειρισμού πακέτων	
Προσέγγιση αυτόνομου πακέτου	Προσέγγιση νοητού κυκλώματος
Δεν υφίσταται φάση αποκατάστασης σύνδεσης, γεγονός που κάνει αποδοτικότερη την προσέγγιση για μικρό αριθμό πακέτων.	Υφίσταται φάση αποκατάστασης σύνδεσης.
Σε περίπτωση συμφόρησης σε ένα μέρος του δικτύου, τα εισερχόμενα πακέτα μπορούν να ακολουθήσουν εναλλακτική διαδρομή, γεγονός που κάνει την προσέγγιση πιο ευέλικτη.	Μετά την εγκατάσταση σύνδεσης, όλα τα πακέτα ακολουθούν την ίδια διαδρομή.
Σε περίπτωση βλάβης ενός κόμβου, τα πακέτα που ακολουθούν μπορούν να επιλέξουν εναλλακτική διαδρομή.	Μικρότερη αξιοπιστία, αφού σε περίπτωση βλάβης ενός κόμβου, τα νοητά κυκλώματα που περνούν από τον κόμβο αυτό χάνονται.
Απαιτούνται αποφάσεις δρομολόγησης (επιπλέον επεξεργασία) με αποτέλεσμα υποβάθμιση της απόδοσης.	Τα πακέτα διατρέχουν το δίκτυο πιο γρήγορα, αφού δεν απαιτούνται αποφάσεις δρομολόγησης.
Είναι πιθανό κάποια πακέτα να παραδοθούν με διαφορετική σειρά από εκείνη που στάλθηκαν	Τα πακέτα φθάνουν στον προορισμό με την ίδια σειρά που στάλθηκαν.

Ασύγχρονος τρόπος μεταφοράς (ATM)

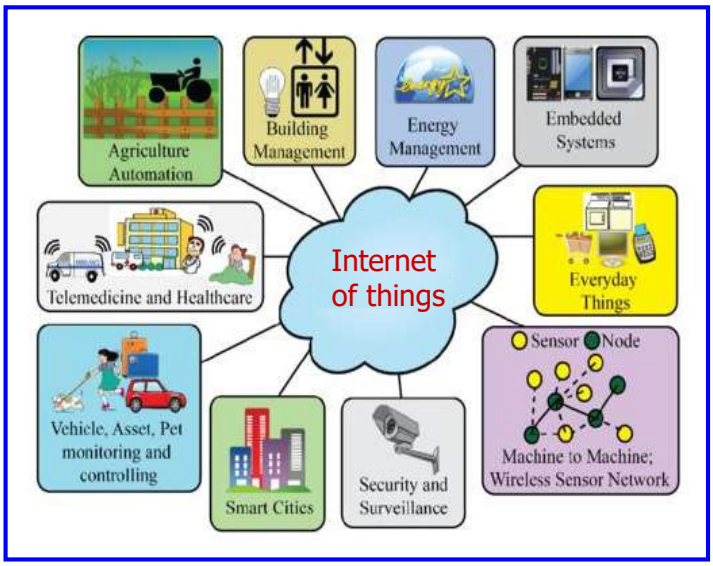
- Η τεχνική ATM είναι μια τεχνική πολυπλεξίας και μεταγωγής πληροφορίας που συνδυάζει τις τεχνικές της μεταγωγής κυκλώματος και της μεταγωγής πακέτου.
- Στην τεχνική ATM, η πληροφορία τεμαχίζεται σε μικρά πακέτα των 53 bytes που αναφέρονται ως κελιά (cells), τα οποία πολυπλέκονται ασύγχρονα στο χρόνο και μεταδίδονται μέσα από συνδέσεις νοητού καναλιού (virtual channel connections, VCCs).
- Οι VCCs είναι νοητές συνδέσεις που αποκαθίστανται μεταξύ τερματικών σταθμών.
- Χρησιμοποιούνται επίσης συνδέσεις νοητού μονοπατιού (virtual path connections, VPCs), οι οποίες είναι μια δέσμη από VCCs, με αποτέλεσμα τα κελιά που αντιστοιχούν σε όλες τις VCCs ενός VPC να μετάγονται μαζί.
- Τα VPCs οδηγούν σε μείωση της πολυπλοκότητας διαχείρισης δικτύου, αφού ομαδοποιούν τις συνδέσεις, ώστε να μοιράζονται κοινά μονοπάτια του δικτύου, με αποτέλεσμα οι ενέργειες διαχείρισης του δικτύου να εφαρμόζονται σε μικρό πλήθος ομάδων συνδέσεων, παρά σε μεγάλο πλήθος ανεξάρτητων συνδέσεων.
- Η τεχνική ATM μπορεί να χρησιμοποιηθεί για δίκτυα διαφόρων μεγεθών (LAN, WAN) και μέσων μετάδοσης (οπτική ίνα, ομοαξονικό).
- Τα ευρυζωνικά ψηφιακά δίκτυα ενοποιημένων υπηρεσιών (broadband integrated services digital network, B-ISDN) βασίζονται στην ATM.

Παράδειγμα σύνθετης δικτύωσης



Διαδίκτυο των αντικειμένων (internet-of-things, IoT)

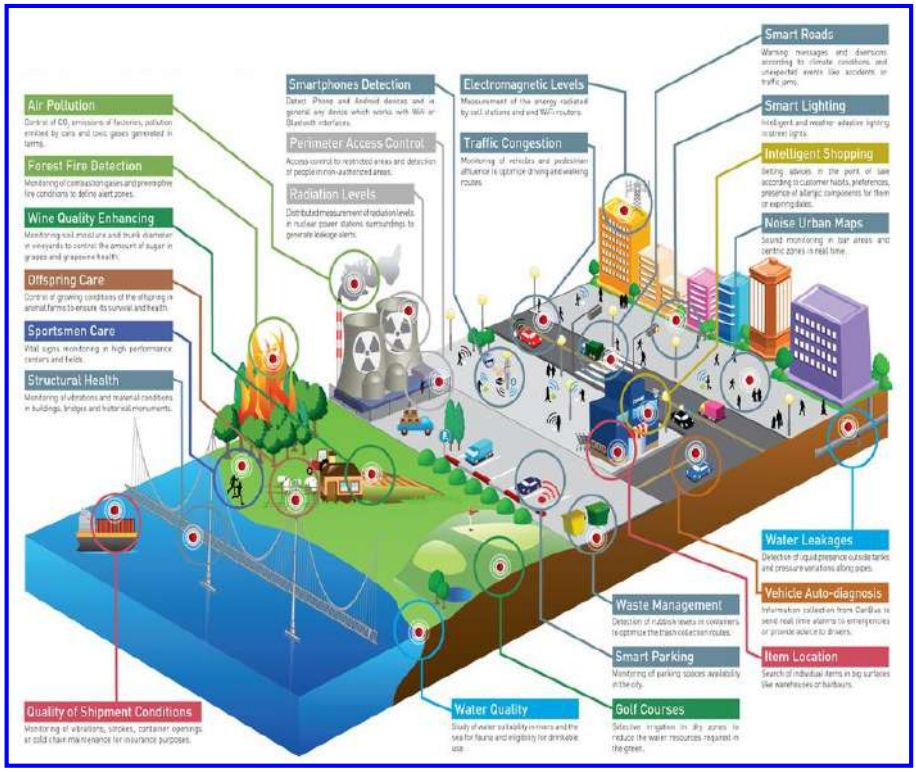
- Το IoT είναι ένα ταχέως αναπτυσσόμενο δίκτυο καθημερινών αντικειμένων, το οποίο μπορεί να μοιράζεται πληροφορίες και να ολοκληρώνει εργασίες με αυτοματοποιημένο τρόπο.
- Το IoT περιλαμβάνει:
  - ✓ τα αντικείμενα, όπως αισθητήρες (sensors) και σύνθετες συσκευές επεξεργασίας και παράστασης πληροφορίας (ενσωματωμένα συστήματα, όπως τα έξυπνα τηλέφωνα),
  - ✓ τα επικοινωνιακά δίκτυα που συνδέουν τα αντικείμενα,
  - ✓ τα υπολογιστικά συστήματα και τις εφαρμογές που χρησιμοποιούν τα δεδομένα που ρέουν προς και από τα αντικείμενα.



Δίκτυα αισθητήρων (sensor networks)

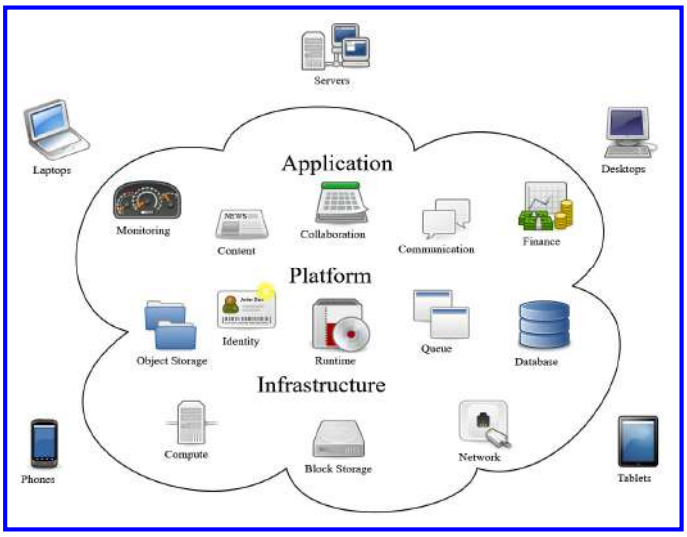
Συνυφασμένα με το IoT είναι τα δίκτυα αισθητήρων που εξυπηρετούν ένα μεγάλο μέρος των αντίστοιχων εφαρμογών.

Δίκτυο αισθητήρων είναι ένα διασυνδεδεμένο σύνολο αισθητήρων προς παρακολούθηση γεγονότων ή συνθηκών σε κατανεμημένες (σε μικρότερη ή μεγαλύτερη κλίμακα) γεωγραφικά περιοχές.



IoT και υπολογιστικό νέφος (cloud)

- Οι σύγχρονες εφαρμογές του IoT χαρακτηρίζονται από διασύνδεση ευρείας κλίμακας και κατανεμημένο μοντέλο αποθήκευσης και επεξεργασίας δεδομένων.
- Η χρησιμότητα των εφαρμογών του IoT βασίζεται στη συγκέντρωση και επεξεργασία της πληροφορίας που μπορεί να συλλέγεται από διάφορες πηγές. Η επεξεργασία αυτή μπορεί να γίνεται κατανεμημένα με εφαρμογές που αναφέρονται ως εφαρμογές υπολογιστικού νέφους (cloud computing).
- Το υπολογιστικό νέφος παρέχει υπολογιστικούς πόρους (όπως διάφορες εφαρμογές, βάσεις δεδομένων, υπηρεσίες αρχείων κ.ά.) μέσω επικοινωνιακού δικτύου.



- Πρόκειται για μία καθολική τεχνολογική υποδομή (global technological infrastructure), στην οποία ο χρήστης ενός υπολογιστή έχει πρόσβαση και χρησιμοποιεί εφαρμογές και δεδομένα τα οποία είναι εγκατεστημένα εκτός του προσωπικού του υπολογιστικού συστήματος.

Αξιοπιστία δικτύου

- Ένα πλήρως αξιόπιστο δίκτυο μεταφέρει την πληροφορία από το ένα άκρο στο άλλο χωρίς σφάλματα.
- Ωστόσο, στην πράξη αυτό δε συμβαίνει, παρά μόνο όταν το δίκτυο είναι εξοπλισμένο με μηχανισμούς εντοπισμού και αντιμετώπισης των σφαλμάτων μετάδοσης.
- Αιτίες σφαλμάτων:
  - ✓ τα σφάλματα μετάδοσης (αντιστροφή της τιμής ενός δυαδικού ψηφίου) οφείλονται στα φυσικά μέσα μεταφοράς, όπως χάλκινα καλώδια, οπτική ίνα, ασύρματη σύζευξη και η αιτία εμφάνισής τους είναι η ηλεκτρομαγνητική παρεμβολή, ο εξωτερικός θόρυβος και ο θόρυβος από τα κυκλώματα αποστολέα και παραλήπτη,
  - ✓ η απόρριψη πακέτων δεδομένων στους κόμβους, αφού συνήθως τα πακέτα δεδομένων αποθηκεύονται προσωρινά στη μνήμη των κόμβων και επειδή αυτές έχουν συγκεκριμένη χωρητικότητα, εάν ένα πακέτο στην άφιξή του βρει τη μνήμη γεμάτη, τότε απορρίπτεται από τον κόμβο,
  - ✓ οι βλάβες ή δυσλειτουργίες του τηλεπικοινωνιακού εξοπλισμού και η εσφαλμένη διαμόρφωση και παραμετροποίηση του δικτυακού λογισμικού, π.χ. μια λανθασμένη ενημέρωση των πινάκων δρομολόγησης στους κόμβους, μπορεί να αλλάξει την κατεύθυνση των πακέτων και ο κόμβος-παραλήπτης μπορεί να μη λάβει τα δεδομένα που του έχουν αποσταλεί ή να λάβει δεδομένα που δεν προορίζονται γι' αυτόν.

Αξιοπιστία δικτύου

- Ένα δίκτυο πρέπει να έχει τη δυνατότητα να αναγνωρίζει τα εσφαλμένα πακέτα, πριν τα παραδώσει στον προορισμό τους.
- Αυτός είναι ο λόγος που το δίκτυο προσθέτει σε κάθε πακέτο δεδομένων μια πλεονάζουσα πληροφορία πριν από τη μεταφορά.
- Ο χρησιμοποιούμενος μηχανισμός μπορεί να εντοπίζει αλλοιώσεις στις τιμές των δυαδικών ψηφίων (ανίχνευση σφαλμάτων).
- Η διόρθωση σφαλμάτων γίνεται απλά με την αντιστροφή του 0 σε 1 ή του 1 σε 0 στις περιπτώσεις εσφαλμένης τιμής ψηφίων.
- Η επαναμετάδοση των εσφαλμένων ή απολεσθέντων πακέτων είναι μια εναλλακτική τεχνική αντιμετώπισης (ελέγχου ή διόρθωσης) των σφαλμάτων μετάδοσης.
- Σε αυτή την περίπτωση, ο παραλήπτης ζητά από τον αποστολέα την επανεκπομπή του εσφαλμένου πακέτου.
- Ως δείκτης αξιόπιστης μετάδοσης συχνά χρησιμοποιείται ο μέσος χρόνος μεταξύ δύο διαδοχικών βλαβών (mean time between failures, MTBF).
- Σε δίκτυα Η/Υ, χρησιμοποιείται ο δείκτης αξιόπιστης μετάδοσης:

$$\text{Δείκτης αξιόπιστης μετάδοσης} = 1 - \frac{\text{Εσφαλμένα δεδομένα}}{\text{Σύνολο ληφθέντων δεδομένων}} \quad (\text{max} = 1)$$

Απόδοση δικτύου – ρυθμός διέλευσης

- Για τη μέτρηση της απόδοσης του δικτύου συνήθως χρησιμοποιούνται:
  - ✓ ο ρυθμός διέλευσης (throughput) και
  - ✓ η καθυστέρηση μετάδοσης (transmission delay).
- Ο ρυθμός διέλευσης (throughput) εκφράζει το πλήθος των δυαδικών ψηφίων που μπορεί να μεταδοθεί αξιόπιστα μέσα από το δίκτυο σε ένα συγκεκριμένο χρονικό διάστημα.
- Αναφέρεται και ως ρυθμός μεταφοράς (transfer rate) ή εξυπηρέτησης, εύρος ζώνης (digital bandwidth), χωρητικότητα ή διαμετακομιστική ικανότητα (capacity).
- Οι δύο τελευταίοι όροι τυπικά αναφέρονται στη δυνατότητα μεταφοράς ενός μέσου μετάδοσης, αλλά συχνά γενικεύονται και για την περίπτωση ενός δικτύου. Ως χωρητικότητα συχνά αναφέρεται ο μέγιστος ρυθμός διέλευσης.
- Για παράδειγμα, δίκτυο με ρυθμό διέλευσης 10 Mbps, μπορεί να μεταφέρει $10 \cdot 10^6$ δυαδικά ψηφία σε ένα δευτερόλεπτο. Στη βιβλιογραφία μπορεί να αναφέρεται και ως $10 \cdot 2^{20}$, αλλά οι δυνάμεις του 2 χρησιμοποιούνται συνήθως για ποσότητες δεδομένων και όχι για ρυθμό μεταφοράς δεδομένων.
- Ο ρυθμός διέλευσης ενός δικτύου υπολογιστών εξαρτάται από τους ρυθμούς μετάδοσης των κόμβων, από τις λειτουργίες ελέγχου και διαχείρισης της κυκλοφορίας των κόμβων και από το ρυθμό εμφάνισης σφαλμάτων κατά τη μεταφορά δεδομένων.

Απόδοση δικτύου – καθυστέρηση μετάδοσης

- Η καθυστέρηση μετάδοσης εκφράζει το χρονικό διάστημα που απαιτείται για τη μεταφορά ενός δυαδικού ψηφίου από το ένα άκρο του δικτύου στο άλλο:

> καθυστέρηση μετάδοσης = χρόνος μετάδοσης (εκπομπής) στο δίκτυο
> + χρόνος διάδοσης στο μέσο
> + χρόνος αναμονής στους κόμβους

- Ο χρόνος διάδοσης στο μέσο είναι αυτός που απαιτείται για να διαδοθεί ένα δυαδικό ψηφίο διαμέσου των φυσικών μέσων μιας διαδρομής και υπολογίζεται ως το πηλίκο της απόστασης των δύο άκρων διά την ταχύτητα διάδοσης στο μέσο.

- Ο χρόνος μετάδοσης (εκπομπής) στο δίκτυο ισούται με το αντίστροφο του ρυθμού διέλευσης ενός δικτύου, δηλαδή εάν ο ρυθμός διέλευσης είναι 8 Mbps, ο χρόνος μετάδοσης (εκπομπής) ενός δυαδικού ψηφίου είναι $1\,\text{bit} / 8\,\text{Mbps} = 0.125\ \mu\text{s}$.

- Ο χρόνος αναμονής στους κόμβους είναι αυτός που αναμένει ένα πακέτο στην προσωρινή μνήμη κάθε κόμβου μέχρι να εξυπηρετηθεί και μπορεί να περιλαμβάνει πιθανή επεξεργασία του πακέτου δεδομένων στον κόμβο.

Παράδειγμα καθυστέρησης μετάδοσης

- Υπολογίζουμε την καθυστέρηση μετάδοσης που αφορά ζεύξη με ρυθμό διέλευσης 1 Mbps, μέσω της οποίας επικοινωνούν δύο κόμβοι που απέχουν μεταξύ τους 220 km και το φυσικό μέσο μετάδοσης είναι οπτική ίνα.

- Αρχικά παρατηρούμε ότι ο σύνδεσμος διατίθεται αποκλειστικά για την εξυπηρέτηση των δύο κόμβων, επομένως τα πακέτα δεδομένων κάθε κόμβου δεν επιβαρύνονται με χρόνο αναμονής.
- Η καθυστέρηση μετάδοσης έχει ως εξής:

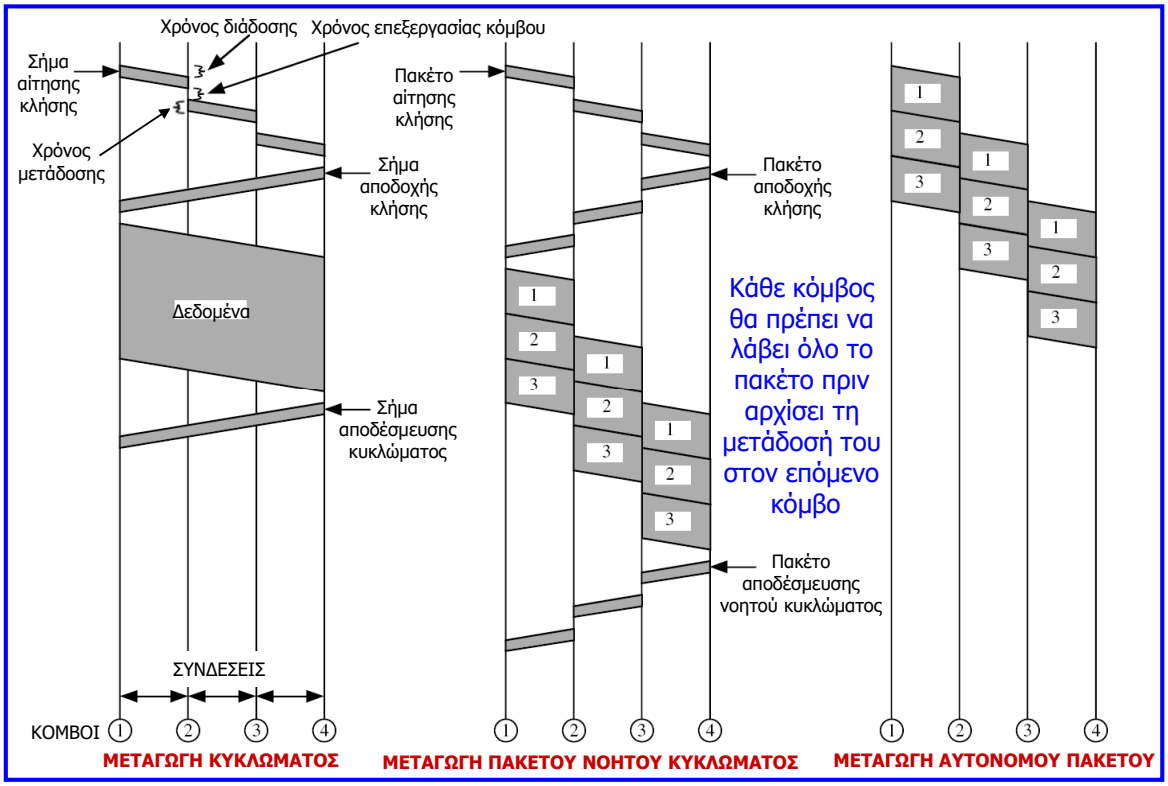
> καθυστέρηση μετάδοσης = χρόνος μετάδοσης στο δίκτυο
> + χρόνος διάδοσης στο μέσο
> + χρόνος αναμονής στους κόμβους
>
> $= (1\ \text{bit} / 10^6\ \text{bps}) + (220\ \text{km} / 300000\ \text{km/s}) + 0$
> $= 0.001\ \text{ms} + 0.73\ \text{ms} = 0.731\ \text{ms}$

- Στο εν λόγω παράδειγμα, διαπιστώνουμε ότι ο χρόνος διάδοσης στο μέσο είναι περίπου 3 τάξεις μεγέθους μεγαλύτερος από το χρόνο μετάδοσης στο δίκτυο.

Απόδοση τεχνικών μεταγωγής

- Η καθυστέρηση που καθορίζει την απόδοση των δικτύων μεταγωγής, περιλαμβάνει:
  - ✓ Χρόνος διάδοσης: χρόνος που απαιτείται για να διαδοθεί το σήμα από έναν κόμβο στον επόμενο (συνήθως είναι μικρός και αντιστρόφως ανάλογος της ταχύτητας ηλεκτρομαγνητικών σημάτων διαμέσου ενός καλωδίου, δηλαδή $2 \cdot 10^8$ m/s).
  - ✓ Χρόνος μετάδοσης: χρόνος εκπομπής τμήματος δεδομένων από τον πομπό.
  - ✓ Χρόνος επεξεργασίας (αναμονής) κόμβου: χρόνος που απαιτείται για να εκτελέσει ο κόμβος την κατάλληλη επεξεργασία καθώς μεταγάγει τα δεδομένα.
- Στη μεταγωγή κυκλώματος εισάγεται καθυστέρηση (διάδοσης & κόμβων) για αποκατάσταση σύνδεσης κατά τη διάρκεια αίτησης κλήσης, ενώ κατά την επιστροφή (σήμα αποδοχής κλήσης) δεν εισάγεται καθυστέρηση κόμβων αφού η σύνδεση έχει ήδη αποκατασταθεί.
- Όταν αποκατασταθεί η σύνδεση, το μήνυμα μεταδίδεται ως ένα μπλοκ χωρίς διακριτές καθυστερήσεις στους κόμβους.
- Η μεταγωγή πακέτου νοητού κυκλώματος έχει ομοιότητες με τη μεταγωγή κυκλώματος, αλλά εισάγεται καθυστέρηση λόγω αναμονής στους ενδιάμεσους κόμβους και στην αποδοχή κλήσης, διότι κάθε πακέτο περιμένει στην ουρά τη σειρά του για μετάδοση από ένα κόμβο. Όταν αποκαθίσταται το νοητό κύκλωμα, το μήνυμα μεταδίδεται σε πακέτα (πιο αργά).
- Η μεταγωγή αυτόνομου πακέτου δεν απαιτεί αποκατάσταση σύνδεσης και για σύντομα μηνύματα είναι πιο γρήγορη από τις προηγούμενες, αλλά με αυξημένες καθυστερήσεις κόμβων, λόγω ξεχωριστής δρομολόγησης κάθε πακέτου.

Απόδοση τεχνικών μεταγωγής



Βασικές επικοινωνιακές διεργασίες

- Βαθμός χρήσης (utilization) συστήματος μετάδοσης.
- Διεπαφή (interfacing) με το σύστημα μετάδοσης.
- Παραγωγή σήματος (signal generation) για την επικοινωνία.
- Συγχρονισμός (synchronization) πομπού και δέκτη.
- Διαχείριση ανταλλαγής (exchange management) πληροφορίας.
- Ανίχνευση & διόρθωση σφαλμάτων (error detection & correction) στο μεταδιδόμενο σήμα.
- Έλεγχος ροής (flow control) πληροφορίας.
- Διευθυνσιοδότηση και δρομολόγηση (addressing and routing) όταν το σύστημα μετάδοσης χρησιμοποιείται από περισσότερες από δύο συσκευές.
- Ανάκτηση (recovery) πληροφορίας.
- Μορφοποίηση των μηνυμάτων (message formatting) που ανταλλάσσονται.
- Ασφάλεια (security) επικοινωνίας.
- Διαχείριση δικτύου (network management).

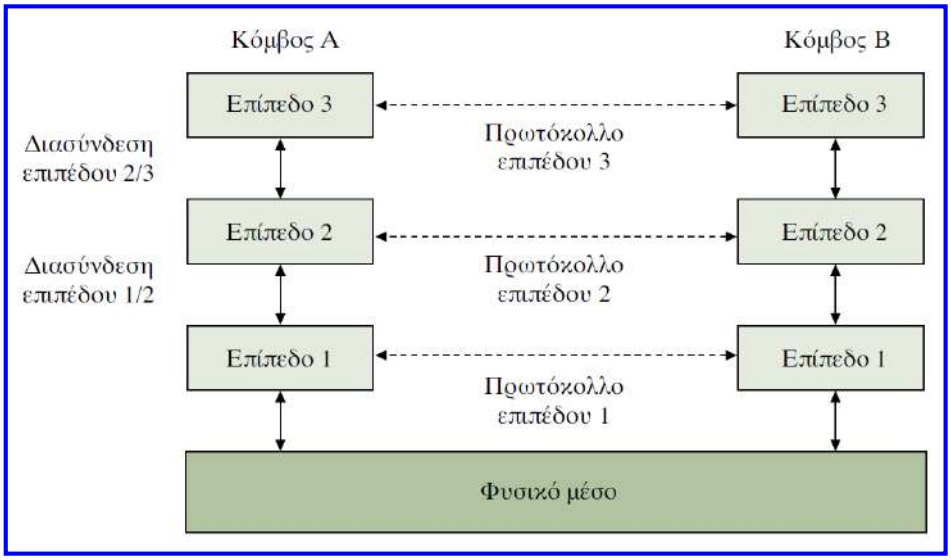
Κεφάλαιο 2: Αρχιτεκτονική δικτύων

Ιεραρχική οργάνωση δικτύων

- Τα δίκτυα οργανώνονται σε στοίβες από επίπεδα ή στρώματα (levels, layers) με βασικό στόχο τη μείωση της πολυπλοκότητάς τους.
- Τα χαρακτηριστικά των δικτύων (πλήθος, ονομασία και λειτουργίες των επιπέδων) μπορεί να διαφοροποιούνται από δίκτυο σε δίκτυο.
- Κάθε επίπεδο προσφέρει συγκεκριμένες υπηρεσίες στα ανώτερα επίπεδα χωρίς να τα απασχολεί με λεπτομέρειες σχετικές με το πώς υλοποιούνται οι παρεχόμενες υπηρεσίες.
- Έτσι, κατά την επικοινωνία δύο κόμβων, το επίπεδο n του ενός κόμβου επικοινωνεί με το επίπεδο n του άλλου κόμβου και η επικοινωνία πραγματοποιείται μέσω ενός συνόλου κανόνων και συνθηκών που αναφέρεται ως πρωτόκολλο του επιπέδου n (layer n protocol).
- Η επικοινωνία των ομότιμων επιπέδων δύο κόμβων είναι νοητή (δεν γίνεται απευθείας μεταφορά δεδομένων από το επίπεδο n ενός κόμβου στο επίπεδο n ενός άλλου).
- Υλοποιείται μια από πάνω προς τα κάτω προσέγγιση, δηλαδή τα δεδομένα και οι πληροφορίες ελέγχου μεταφέρονται από το ανώτερο στο κατώτερο επίπεδο, μέχρι αυτά να φτάσουν στο κατώτατο επίπεδο και η φυσική επικοινωνία (μεταφορά) μεταξύ δύο κόμβων γίνεται μέσω του φυσικού μέσου (physical medium), το οποίο βρίσκεται κάτω από το κατώτατο επίπεδο.

Ιεραρχική οργάνωση δικτύων

- Ο τρόπος με τον οποίο επικοινωνούν δύο διαδοχικά επίπεδα του ίδιου κόμβου ονομάζεται διεπαφή ή διασύνδεση (interface) και μέσω της διεπαφής καθορίζονται οι λειτουργίες και υπηρεσίες που προσφέρει ένα επίπεδο στο επίπεδο που βρίσκεται πάνω από αυτό.
- Αρχιτεκτονική δικτύου (network architecture) είναι το σύνολο των επιπέδων και πρωτοκόλλων που συγκροτούν ένα δίκτυο.



Βασικές λειτουργίες πρωτοκόλλων

- Ενθυλάκωση (encapsulation): προσθήκη πληροφορίας ελέγχου (διεύθυνση αποστολέα-παραλήπτη, κώδικας ανίχνευσης σφαλμάτων, άλλες πληροφορίες ελέγχου) στα δεδομένα.
- Κερματισμός (segmentation) δεδομένων από τα χαμηλότερα επίπεδα του κόμβου-πηγή και επανασυναρμολόγηση (reassembly) τους κατά τη λήψη.
- Έλεγχος σύνδεσης (connection control) που περιλαμβάνει τρεις φάσεις: αποκατάσταση σύνδεσης, μεταφορά δεδομένων, τερματισμός σύνδεσης.
- Διατεταγμένη παράδοση (ordered delivery) δεδομένων.
- Έλεγχος ροής (flow control) που εκτελείται από τον κόμβο-προορισμό (λήψης) για να περιορίσει την ποσότητα ή το ρυθμό δεδομένων που στέλνονται από τον κόμβο-πηγή.
- Έλεγχος σφαλμάτων (error control): ανίχνευση σφαλμάτων και επαναμετάδοση δεδομένων.
- Διευθυνσιοδότηση (addressing): κάθε κόμβος ή ενδιάμεσο σύστημα διαθέτει μοναδική διεύθυνση για χρήση στη δρομολόγηση (routing) των μονάδων δεδομένων.
- Πολυπλεξία (multiplexing) για την υποστήριξη πολλαπλών συνδέσεων στο ίδιο σύστημα: πολλαπλές συνδέσεις (χρήστες) υψηλότερων επιπέδων μοιράζονται μία σύνδεση χαμηλότερου επιπέδου (ανοδική πολυπλεξία) ή η κίνηση της υψηλότερης σύνδεσης διαιρείται ανάμεσα στις πολλαπλές χαμηλότερες συνδέσεις (καθοδική πολυπλεξία).
- Υπηρεσίες μετάδοσης (transmission services): προτεραιότητα, ποιότητα υπηρεσίας (π.χ. δεδομένα που απαιτούν όριο καθυστέρησης), ασφάλεια (π.χ. απαγόρευση πρόσβασης).

Πρότυπα πρωτόκολλα επικοινωνίας

- Τα πρότυπα πρωτόκολλα επικοινωνίας αναπτύσσονται για να εξασφαλίσουν τη διαλειτουργικότητα (interoperability) του τηλεπικοινωνιακού εξοπλισμού.
- Ως βάση για την ανάπτυξη προτύπων χρησιμοποιούνται οι αρχιτεκτονικές πρωτοκόλλων TCP/IP και OSI (μοντέλα αναφοράς), που αναλύονται στη συνέχεια.
- Πλεονεκτήματα προτύπων: Εξασφαλίζουν ευρεία αγορά για τον τηλεπικοινωνιακό εξοπλισμό και το αντίστοιχο λογισμικό και επιτρέπουν σε προϊόντα διαφορετικών κατασκευαστών να επικοινωνούν μεταξύ τους, παρέχοντας στον αγοραστή μεγαλύτερη ευελιξία στην επιλογή και τη χρήση του εξοπλισμού.
- Μειονεκτήματα προτύπων: Δεν ακολουθούν πάντα την εξέλιξη της τεχνολογίας αφού μέχρι να αναπτυχθεί ένα πρότυπο εμφανίζονται πιο αποτελεσματικές τεχνικές και επίσης μπορεί να υπάρξουν περισσότερα από ένα πρότυπα για τον ίδιο σκοπό (από διαφορετικούς οργανισμούς προτυποποίησης).
- Οργανισμοί προτυποποίησης: Internet Society, ISO (international organization for standardization), ITU (international telecommunication union), ETSI (European telecommunications standards institute), IEEE-SA (IEEE standards association) κ.ά.

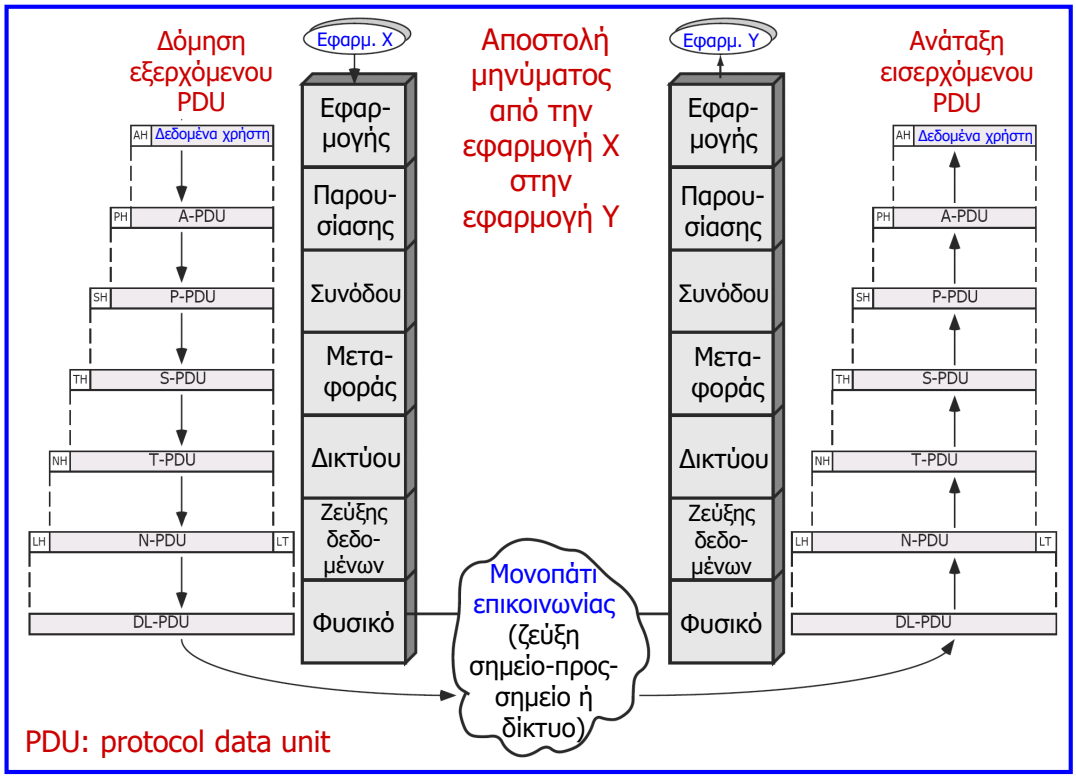
Μοντέλο αναφοράς OSI

- Το μοντέλο αναφοράς διασύνδεσης ανοικτών συστημάτων (OSI, open systems interconnection) είναι το μοντέλο που προέκυψε από επιτροπή του ISO και δημοσιεύτηκε το 1984.
- Οι λειτουργίες επικοινωνιών χωρίζονται σε ένα ιεραρχικό σύνολο από επίπεδα, όπου κάθε επίπεδο εκτελεί ένα υποσύνολο λειτουργιών που απαιτούνται για την επικοινωνία.
- Κάθε επίπεδο βασίζεται στο αμέσως χαμηλότερο για να εκτελέσει πιο στοιχειώδεις λειτουργίες που σημαίνει ότι κάθε επίπεδο παρέχει υπηρεσίες στο αμέσως υψηλότερο.
- Τα επίπεδα καθορίζονται έτσι ώστε αλλαγές σε ένα επίπεδο να μην απαιτούν αλλαγές και στα άλλα επίπεδα.
- Εκτός από το φυσικό επίπεδο (που είναι το κατώτατο και εξασφαλίζει τη φυσική διεπαφή μεταξύ των συστημάτων), δεν υπάρχει άμεση επικοινωνία ανάμεσα στα ομότιμα επίπεδα διαφορετικών συστημάτων, που σημαίνει ότι πάνω από το φυσικό επίπεδο κάθε οντότητα πρωτοκόλλου διαβιβάζει τα δεδομένα στο αμέσως χαμηλότερο επίπεδο, για να φτάσουν στη συνέχεια στην ομότιμη οντότητα πρωτοκόλλου ενός άλλου συστήματος.
- Στο φυσικό επίπεδο η σύνδεση μπορεί να είναι άμεση ή μέσω δικτύου μεταγωγής (έμμεση).

Μοντέλο αναφοράς OSI

Ένα επίπεδο μπορεί να κερματίσει (τεμαχίσει) τη μονάδα δεδομένων που λαμβάνει, ώστε να εξυπηρετούνται οι ανάγκες του.

Συναρμολόγηση των μονάδων δεδομένων γίνεται από το ομότιμο επίπεδο του συστήματος λήψης.



Μοντέλο αναφοράς OSI

- Σε κάθε επίπεδο της αρχιτεκτονικής OSI διακρίνονται τρία σημαντικά στοιχεία:
  - ✓ Προδιαγραφή πρωτοκόλλου (protocol specification): δύο οντότητες του ιδίου επιπέδου σε διαφορετικά συστήματα συνεργάζονται και αλληλεπιδρούν μέσω ενός πρωτοκόλλου που θα πρέπει να είναι επακριβώς ορισμένο όσον αφορά τη μορφή μονάδων δεδομένων, τη σημασιολογία όλων των πεδίων και την επιτρεπτή ακολουθία των μονάδων δεδομένων.
  - ✓ Καθορισμός υπηρεσιών (service definition): εκτός από το πρωτόκολλο απαιτείται και η λειτουργική περιγραφή που καθορίζει τις υπηρεσίες που παρέχονται από κάθε επίπεδο στο αμέσως υψηλότερο του.
  - ✓ Διευθυνσιοδότηση (addressing): κάθε επίπεδο παρέχει υπηρεσίες στις οντότητες του αμέσως υψηλότερου και οι οντότητες στις οποίες παρέχονται οι υπηρεσίες δηλώνονται (προσδιορίζονται) με σημεία πρόσβασης υπηρεσίας (service access points, SAP).
- Οι υπηρεσίες ανάμεσα σε γειτονικά επίπεδα εκφράζονται με βασικά μηνύματα και παραμέτρους.
- Ένα βασικό μήνυμα καθορίζει τη λειτουργία που πρέπει να εκτελεστεί και τις παραμέτρους που θα χρησιμοποιηθούν για την αποστολή δεδομένων και πληροφορίας ελέγχου (π.χ. δεδομένα, διεύθυνση προορισμού κ.ά.).

Μοντέλο αναφοράς OSI

- Φυσικό επίπεδο (physical layer): καλύπτει τη φυσική διεπαφή ανάμεσα στις συσκευές και τους κανόνες με τους οποίους τα ψηφία (bits) μεταδίδονται στο φυσικό μέσο.
- Επίπεδο (ελέγχου) ζεύξης δεδομένων (DLC, data link control layer): εξασφαλίζει την αξιοπιστία της φυσικής ζεύξης (με έλεγχο ή ρύθμιση ροής) και παρέχει υπηρεσίες που αφορούν την ενεργοποίηση, διόρθωση (υπηρεσία ανίχνευσης και ελέγχου σφαλμάτων) και απενεργοποίηση της ζεύξης.
- Επίπεδο δικτύου (network layer): δεν χρησιμοποιείται σε απευθείας ζεύξεις, καθορίζει τον τρόπο δρομολόγησης των μονάδων δεδομένων (σε συνεργασία με το δίκτυο) από την αφετηρία τους έως τον προορισμό τους και αιτείται συγκεκριμένες υπηρεσίες δικτύου (π.χ. προτεραιότητα), απαλλάσσοντας τα υψηλότερα επίπεδα από το να γνωρίζουν οτιδήποτε σχετικά με τη μετάδοση δεδομένων και τις χρησιμοποιούμενες τεχνολογίες μεταγωγής.
- Επίπεδο μεταφοράς (transport layer): αφορά μόνο τους τερματικούς σταθμούς και υλοποιεί το κανάλι επικοινωνίας μεταξύ τους (νοητό κύκλωμα ή αυτόνομα πακέτα δεδομένων), αναλαμβάνει την πολύπλεξη μηνυμάτων, διαθέτει μηχανισμούς επαλήθευσης ορθής παράδοσης των μηνυμάτων (απαλλαγμένα από σφάλματα, στη σωστή σειρά και χωρίς απώλειες ή πολλαπλά αντίγραφα) και ελέγχου ροής δεδομένων μεταξύ των τερματικών σταθμών (που είναι ανεξάρτητοι από τους αντίστοιχους μηχανισμούς του επιπέδου ζεύξης δεδομένων, οι οποίοι αφορούν ζεύξη δύο σημείων) και παρέχει τις υπηρεσίες που ζητούνται από το επίπεδο συνόδου (π.χ. μέγιστη καθυστέρηση, προτεραιότητα, ασφάλεια).

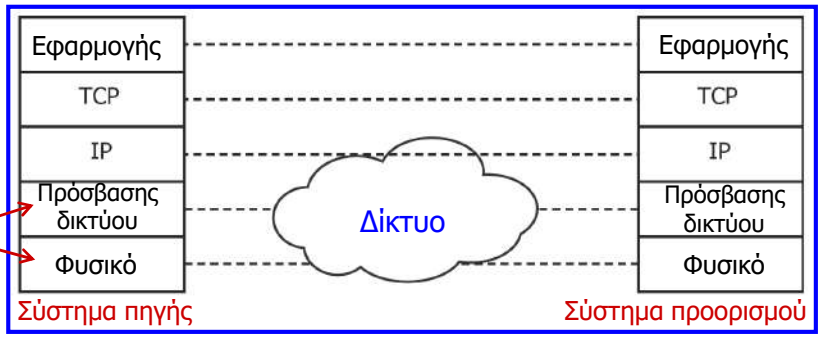
Μοντέλο αναφοράς OSI

- Επίπεδο συνόδου (session layer): παρέχει μηχανισμό για τον έλεγχο του διαλόγου ανάμεσα σε εφαρμογές που περιλαμβάνει τον τρόπο διαλόγου που μπορεί να είναι δύο κατευθύνσεων ταυτόχρονα (full-duplex) ή δύο κατευθύνσεων εναλλάξ (half-duplex), το συγχρονισμό των δεδομένων μεταξύ των δύο πλευρών και την ανάκτηση (recovery) δεδομένων με υποστήριξη της επαναμετάδοσής τους σε περίπτωση βλάβης.
- Επίπεδο παρουσίασης (presentation layer): καθορίζει τη μορφή (κωδικοποίηση) των δεδομένων που πρόκειται να ανταλλαχθούν ανάμεσα στις εφαρμογές και παρέχει στις εφαρμογές υπηρεσίες μετασχηματισμού δεδομένων (π.χ. συμπίεση ή κρυπτογράφηση δεδομένων).
- Επίπεδο εφαρμογής (application layer): παρέχει το μέσο ώστε οι εφαρμογές να έχουν πρόσβαση στο περιβάλλον της αρχιτεκτονικής OSI, δηλαδή περιλαμβάνει λειτουργίες και μηχανισμούς για την υποστήριξη εφαρμογών των χρηστών, καθώς και εφαρμογές γενικού σκοπού (μεταφορά αρχείων, ηλεκτρονικό ταχυδρομείο, πρόσβαση τερματικού σε απομακρυσμένους υπολογιστές κ.ά.).

Μοντέλο αναφοράς TCP/IP

Λεπτομέρειες για τα πρωτόκολλα IP και TCP στα κεφάλαια 5 και 6.

Συνιστούν το επίπεδο πρόσβασης στο φυσικό μέσο



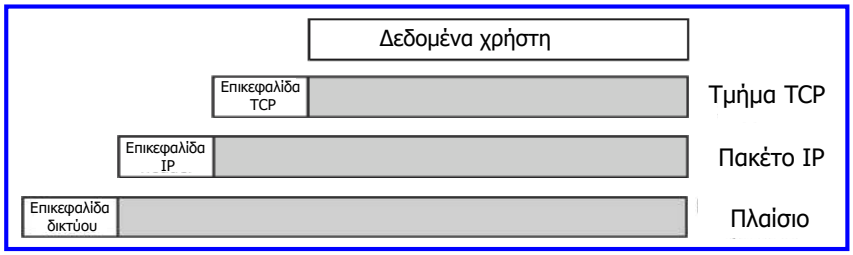
- Το διαδίκτυο (internet) έχει ως βάση το μοντέλο TCP/IP και η δραματική εξάπλωσή του σφράγισε τη νίκη του TCP/IP έναντι του OSI. Αναπτύχθηκε από την DAPRA (US Defense Advanced Research Projects Agency) και δημοσιεύτηκε στο τέλος του 1981.
- Επίπεδο εφαρμογής: παρέχει τους μηχανισμούς που απαιτούνται για την επικοινωνία των εφαρμογών χρηστών (π.χ. SMTP, FTP, HTTP).
- Επίπεδο μεταφοράς (TCP, transmission control protocol): παρέχει υπηρεσίες μεταφοράς δεδομένων από άκρο σε άκρο, «κρύβοντας» τις λεπτομέρειες του δικτύου ή δικτύων που χρησιμοποιούνται και περιλαμβάνει μηχανισμούς αξιοπιστίας της μεταφοράς.
- Επίπεδο διαδικτύου (IP, internet protocol): διενεργεί τη δρομολόγηση μονάδων δεδομένων από το σύστημα πηγής στο σύστημα προορισμού μέσω ενός ή περισσοτέρων δικτύων που συνδέονται μεταξύ τους με δρομολογητές.

Μοντέλο αναφοράς TCP/IP

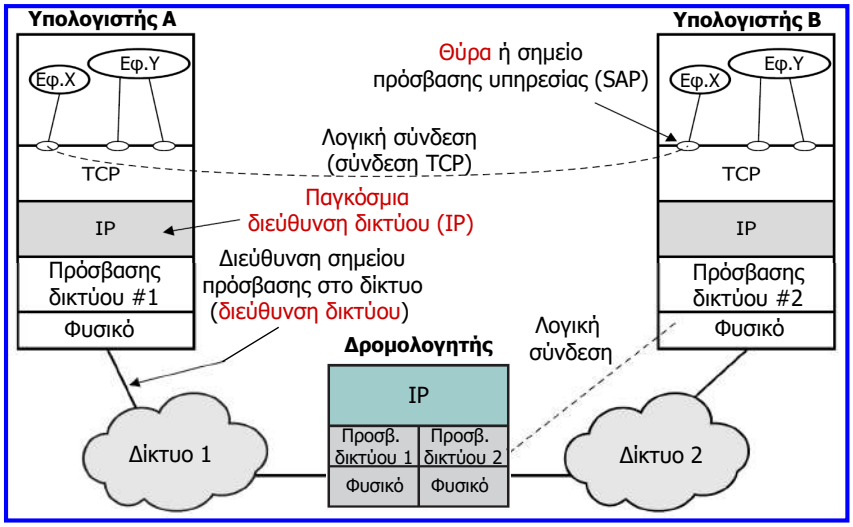
- Επίπεδο πρόσβασης δικτύου (network access layer): διενεργεί τη λογική διεπαφή ανάμεσα σε δύο κόμβους ενός δικτύου.
- Φυσικό επίπεδο (physical layer): καλύπτει τη φυσική διεπαφή μεταξύ μιας συσκευής μετάδοσης δεδομένων και ενός μέσου μετάδοσης ή δικτύου και σχετίζεται με τα χαρακτηριστικά του μέσου μετάδοσης, το ρυθμό μετάδοσης σήματος και την κωδικοποίηση σήματος.
- Τα δύο κατώτερα επίπεδα δεν καθορίζονται επακριβώς στο μοντέλο αναφοράς TCP/IP.
- Το επίπεδο IP υλοποιείται σε όλα τα τερματικά συστήματα και στους δρομολογητές, αφού αναλαμβάνει τη μετακίνηση / δρομολόγηση δεδομένων διαμέσου δρομολογητών, ενώ το TCP υλοποιείται μόνο στα τερματικά συστήματα.
- Στο μοντέλο αναφοράς TCP/IP καθορίζονται δύο επίπεδα διευθυνσιοδότησης:
  - ✓ Παγκόσμια διεύθυνση διαδικτύου (IP) για κάθε σύστημα, που επιτρέπει στα δεδομένα να παραδίδονται στο σωστό σύστημα.
  - ✓ Διεύθυνση που αναφέρεται ως θύρα (port), αφορά κάθε εφαρμογή ενός συστήματος (μοναδική εσωτερικά στο σύστημα) και επιτρέπει στο πρωτόκολλο TCP να παραδίδει δεδομένα στην κατάλληλη εφαρμογή.

Μοντέλο αναφοράς TCP/IP

Μονάδες δεδομένων στην αρχιτεκτονική TCP/IP



Επίπεδα διευθυνσιοδότησης στην αρχιτεκτονική TCP/IP

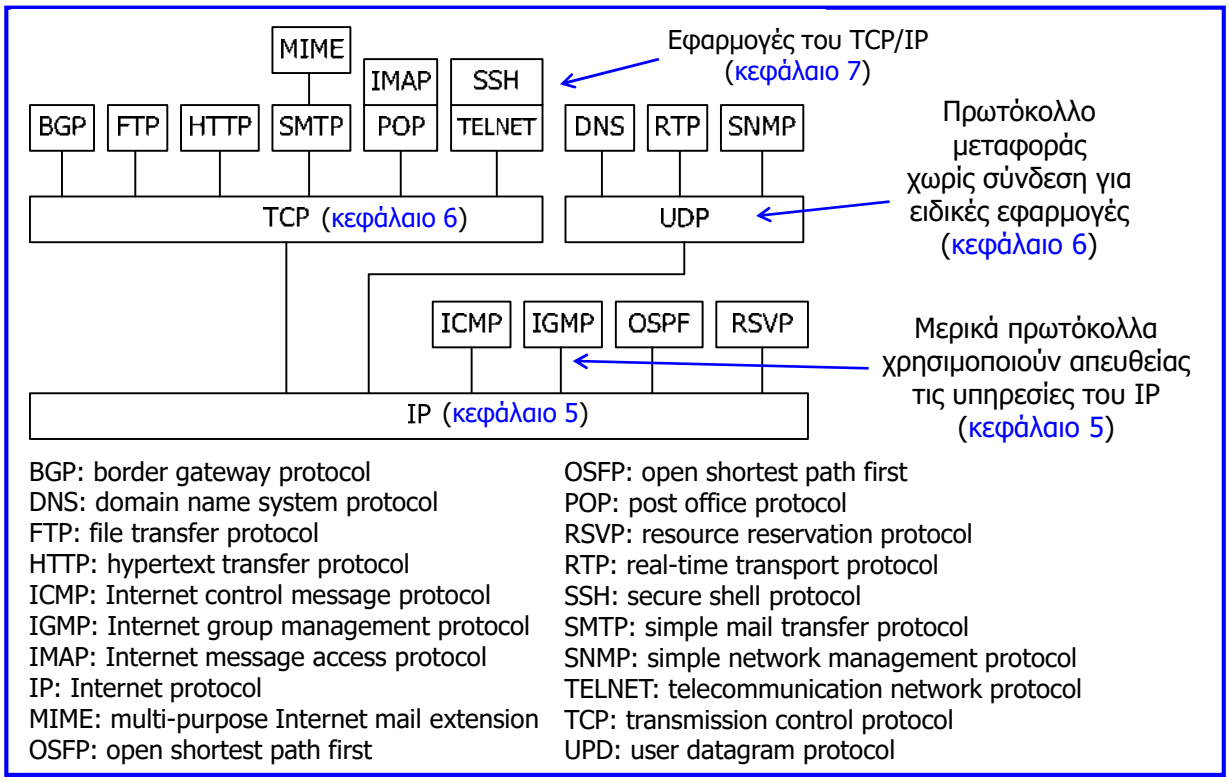


Οι δρομολογητές χρησιμοποιούν μόνο τα τρία πρώτα επίπεδα

Μοντέλο αναφοράς TCP/IP

- Για τον έλεγχο λειτουργίας των επιπέδων, εκτός από τα δεδομένα του χρήστη πρέπει να μεταδίδονται και πληροφορίες ελέγχου.
- Το TCP όταν λαμβάνει μια μονάδα δεδομένων από το στρώμα εφαρμογής, το τεμαχίζει σε μικρότερα κομμάτια για ευκολότερο χειρισμό και προσθέτει πληροφορία ελέγχου (επικεφαλίδα TCP, TCP header), δημιουργώντας τμήματα TCP (TCP segments).
- Η επικεφαλίδα TCP περιλαμβάνει τη θύρα προορισμού, τον αύξοντα αριθμό (αριθμό ακολουθίας) του τμήματος και το άθροισμα ελέγχου (checksum) που είναι συνάρτηση του υπόλοιπου περιεχομένου του τμήματος.
- Στη συνέχεια, το TCP παραδίδει κάθε τμήμα στο IP, το οποίο προσθέτει μία επικεφαλίδα (IP header) που περιλαμβάνει τη διεύθυνση προορισμού (IP), δημιουργώντας ένα αυτόνομο πακέτο IP (IP packet).
- Τέλος, το πακέτο IP παραδίδεται στο επίπεδο πρόσβασης δικτύου, το οποίο προσθέτει επικεφαλίδα (network header) δημιουργώντας ένα πακέτο δικτύου ή πλαίσιο (network-level packet, frame).
- Η επικεφαλίδα δικτύου περιλαμβάνει τη διεύθυνση προορισμού δικτύου (δηλαδή, τη διεύθυνση της διεπαφής δικτύου του κόμβου προορισμού), καθώς και αιτήσεις υπηρεσιών (π.χ. προτεραιότητα).

Μοντέλο αναφοράς TCP/IP



BGP: border gateway protocol
DNS: domain name system protocol
FTP: file transfer protocol
HTTP: hypertext transfer protocol
ICMP: Internet control message protocol
IGMP: Internet group management protocol
IMAP: Internet message access protocol
IP: Internet protocol
MIME: multi-purpose Internet mail extension
OSFP: open shortest path first
OSFP: open shortest path first
POP: post office protocol
RSVP: resource reservation protocol
RTP: real-time transport protocol
SSH: secure shell protocol
SMTP: simple mail transfer protocol
SNMP: simple network management protocol
TELNET: telecommunication network protocol
TCP: transmission control protocol
UPD: user datagram protocol

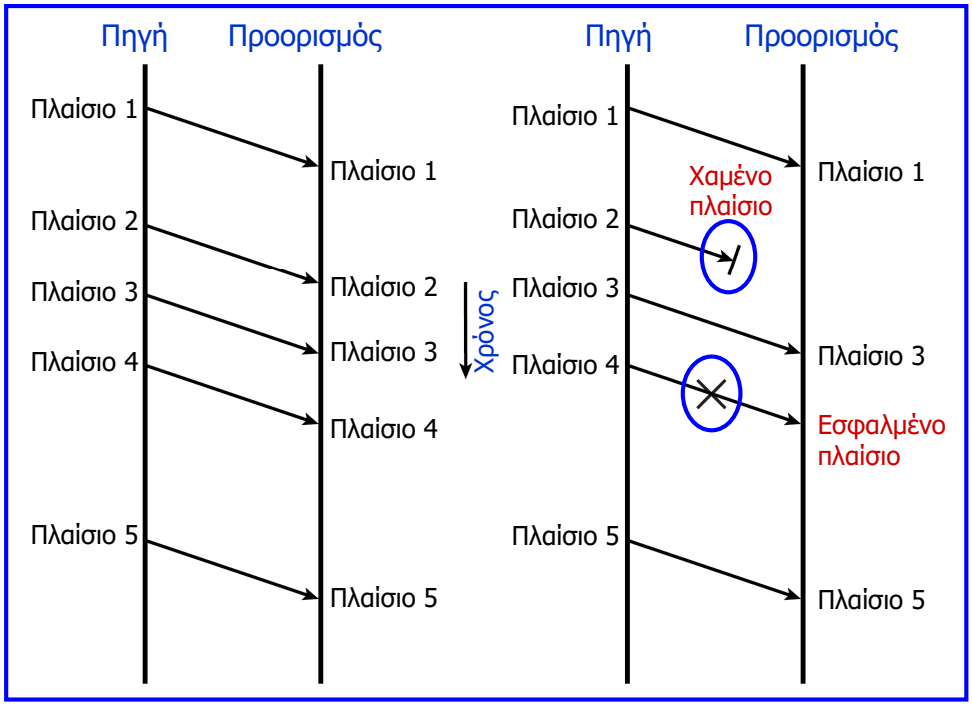
Αντιστοίχιση μοντέλων αναφοράς OSI και TCP/IP

Κεφάλαιο 3: Πρωτόκολλα ελέγχου ζεύξης δεδομένων

Βασικές έννοιες

- Λόγω της πιθανότητας σφαλμάτων μετάδοσης και της πιθανής ανάγκης του δέκτη να ρυθμίσει το ρυθμό άφιξης των δεδομένων, σε κάθε συσκευή επικοινωνίας υλοποιείται ένα πρωτόκολλο (επίπεδο) ελέγχου ζεύξης δεδομένων (DLC, data link control).
- Ο έλεγχος ή ρύθμιση ροής (flow control) επιτρέπει στο δέκτη να ρυθμίσει τη ροή δεδομένων που λαμβάνει από τον πομπό, έτσι ώστε η προσωρινή μνήμη του (buffer) να μην ξεπεράσει (overflow) τη χωρητικότητά της.
- Η ανίχνευση σφαλμάτων (error detection) διενεργείται με τον υπολογισμό ενός κώδικα ανίχνευσης σφάλματος ο οποίος αποτελεί συνάρτηση των ψηφίων (bits) που μεταδίδονται. Ο δέκτης υπολογίζει τον κώδικα βασισμένος στα εισερχόμενα ψηφία και συγκρίνει το αποτέλεσμα με τον εισερχόμενο κώδικα, ώστε να ανιχνεύσει την ύπαρξη σφαλμάτων.
- Μια προσέγγιση για έλεγχο ή διόρθωση σφαλμάτων (error control) βασίζεται στην αναμετάδοση των χαμένων ή εσφαλμένων μονάδων δεδομένων (πλαισίων), των οποίων δεν έχει επιβεβαιωθεί η λήψη ή για τα οποία η πλευρά του δέκτη ζητά αναμετάδοση.
- Απαιτήσεις για αποτελεσματική επικοινωνία δεδομένων: συγχρονισμός πλαισίων (αρχή και τέλος κάθε πλαισίου πρέπει να είναι αναγνωρίσιμα), έλεγχος ροής, έλεγχος σφαλμάτων, διευθυνσιοδότηση, έλεγχος και δεδομένα στην ίδια ζεύξη (ο δέκτης θα πρέπει να τα διακρίνει), διαχείριση ζεύξεων (έναρξη, συντήρηση και λήξη ανταλλαγής δεδομένων απαιτούν συνεργασία και συντονισμό μεταξύ των σταθμών).

Μοντέλο μετάδοσης πλαισίων



Έλεγχος ροής (flow control)

- Έλεγχος ροής είναι μία τεχνική διαβεβαίωσης ότι ο πομπός δεν κατακλύζει με δεδομένα τον δέκτη.
- Ο δέκτης δεσμεύει μία προσωρινή μνήμη (buffer) δεδομένων με κάποιο μέγιστο μέγεθος, αφού όταν λαμβάνει τα δεδομένα πρέπει να τα επεξεργαστεί πριν τα προωθήσει στα υψηλότερα επίπεδα.
- Χωρίς έλεγχο ροής, η μνήμη του δέκτη μπορεί να γεμίσει και να υπερχειλίσει, ενώ επεξεργάζεται παλιά δεδομένα.
- Τα δεδομένα στέλνονται σε ακολουθίες πλαισίων και κάθε πλαίσιο (frame) περιέχει δεδομένα καθώς και πληροφορίες ελέγχου.
- Ο χρόνος που χρειάζεται ένας σταθμός για να εκπέμψει (μεταδώσει) όλα τα ψηφία (bits) ενός πλαισίου στο μέσο αναφέρεται ως χρόνος μετάδοσης και είναι ανάλογος του μήκους (μεγέθους) πλαισίου, ενώ ο χρόνος διάδοσης είναι αυτός που χρειάζεται ένα ψηφίο για να καλύψει τη ζεύξη μεταξύ του πομπού και του δέκτη.

Χρόνοι διάδοσης, μετάδοσης και μήκος ζεύξης

- Θεωρούμε αρχικά μετάδοση χωρίς σφάλματα και απώλειες, καθώς επίσης και ότι τα πλαίσια λαμβάνονται με τη σειρά που στέλνονται.
- Χρόνος μετάδοσης ενός ψηφίου (bit): ο χρόνος που απαιτείται για τη εκπομπή ενός ψηφίου (bit) στο μέσο μετάδοσης (εάν R ο ρυθμός μετάδοσης σε bps, τότε ισούται με 1/R)
- Ο χρόνος μετάδοσης ενός πλαισίου (t_{frame}) είναι ανάλογος του μήκους πλαισίου, δηλαδή εάν το πλαίσιο περιέχει L bits τότε: $t_{frame} = L / R$.
- Χρόνος διάδοσης (t_{prop}): εάν η απόσταση πομπού-δέκτη είναι d και η ταχύτητα διάδοσης είναι V, τότε $t_{prop} = d / V$.
- Λόγος χρόνου διάδοσης προς χρόνο μετάδοσης (a):

$$a = t_{prop} / t_{frame} = (d / V) / (L / R) = (R \cdot d) / (V \cdot L)$$

- Μήκος ζεύξης είναι ο αριθμός των παρόντων bit (B) στο μέσο όταν μια ροή από bits έχει καταλάβει πλήρως το μέσο. Υπολογίζεται με εξίσωση των χρόνων διάδοσης και μετάδοσης:

$$B / R = d / V \Rightarrow B = R \cdot (d / V) \Rightarrow B = a \cdot L$$

- Διαπιστώνουμε ότι, ο λόγος a εκφράζει επίσης τη σχέση του μήκους του μέσου (μήκος ζεύξης) με το μήκος (μέγεθος) πλαισίου L .
- Παράδειγμα: R = 100 Mbps, V = $2 \cdot 10^8$ m/s, d = 100 m $\Rightarrow$ Μήκος ζεύξης: B = 50 bits

Έλεγχος ροής παύσης και αναμονής (stop and wait)

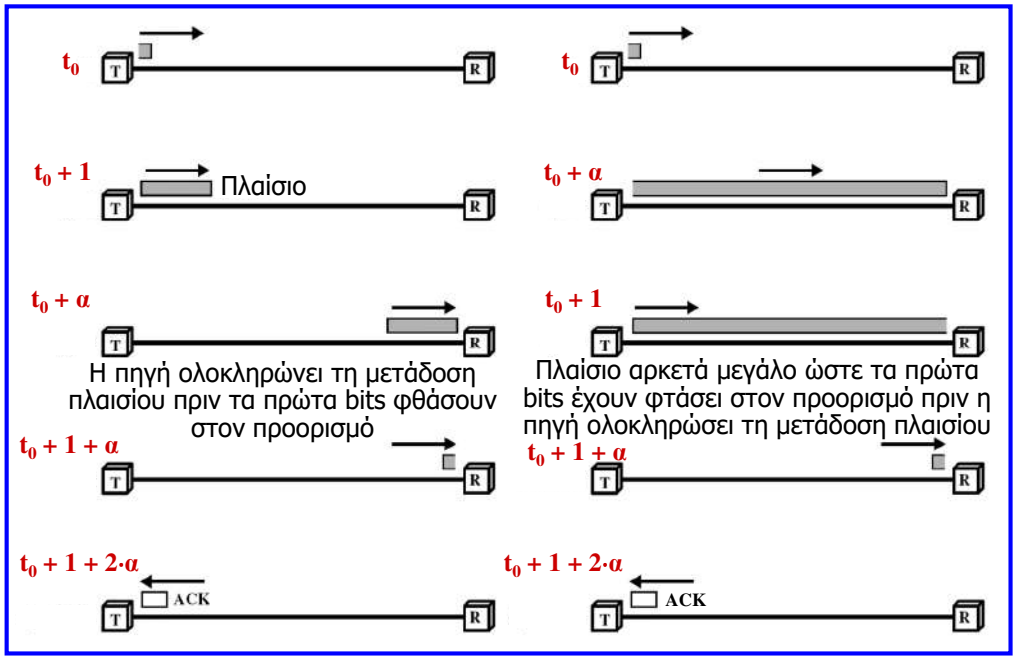
- Η απλούστερη τεχνική ελέγχου ροής είναι ο έλεγχος ροής παύσης και αναμονής (stop and wait). Στη τεχνική αυτή, η πηγή μεταδίδει ένα πλαίσιο δεδομένων που λαμβάνεται από τον προορισμό, ο οποίος στέλνει επιβεβαίωση λήψης (acknowledgment, ACK).
- Η πηγή αναμένει την επιβεβαίωση λήψης πριν μεταδώσει το επόμενο πλαίσιο, συνεπώς ο προορισμός μπορεί να σταματήσει τη ροή δεδομένων με παρακράτηση της επιβεβαίωσης λήψης.
- Η τεχνική αυτή λειτουργεί αποδοτικά για μετάδοση μικρού αριθμού πλαισίων μεγάλου μεγέθους.
- Ωστόσο, συχνά η πηγή τεμαχίζει μία μεγάλη ενότητα δεδομένων σε πολλά μικρά πλαίσια διότι το μέγεθος της ενδιάμεσης μνήμης δέκτη είναι περιορισμένο, η ανίχνευση και ο έλεγχος σφαλμάτων γίνονται γρηγορότερα και η πρόσβαση σε διαμοιραζόμενο μέσο κατανέμεται δικαιότερα στους σταθμούς.
- Το μειονέκτημα της τεχνικής παύσης και αναμονής είναι ότι στην ουσία μόνο ένα πλαίσιο βρίσκεται σε κατάσταση διέλευσης.
- Όταν το μήκος ζεύξης (B = α · L) είναι μεγαλύτερο από το μήκος πλαισίου L (δηλαδή, α > 1) προκύπτει σοβαρή αναποτελεσματικότητα όσον αφορά τη χρήση της ζεύξης, ενώ ακόμη και όταν α < 1, η ζεύξη χρησιμοποιείται ανεπαρκώς.

Βαθμός χρήσης ζεύξης παύσης και αναμονής

Έστω ότι,

χρόνος μετάδοσης: $t_{frame} = 1$

χρόνος διάδοσης: $\alpha = t_{prop} / t_{frame}$



α > 1
Υψηλός ρυθμός μετάδοσης δεδομένων και/ή μεγάλη απόσταση μεταξύ σταθμών
(Μήκος ζεύξης > Μήκος πλαισίου)

α < 1
Πλαίσιο μεγάλου μεγέθους
(Μήκος ζεύξης < Μήκος πλαισίου)

Βαθμός χρήσης ζεύξης παύσης και αναμονής

- Χρόνος αποστολής ενός πλαισίου και λήψης μιας επιβεβαίωσης:

$$T_F = t_{frame} + t_{prop} + t_{proc} + t_{ack} + t_{prop} + t_{proc}$$

- Εάν θεωρήσουμε ότι ο χρόνος επεξεργασίας (t_{proc}) σε κάθε σταθμό ώστε αυτός να ανταποκριθεί σε ένα εισερχόμενο γεγονός είναι αμελητέος και ότι το πλαίσιο ACK είναι πολύ μικρό, τότε ο αντίστοιχος χρόνος για το σύνολο των δεδομένων (n πλαίσια), που αναφέρεται ως συνολικός χρόνος αποστολής n πλαισίων, είναι:

$$T = n \cdot (2 \cdot t_{prop} + t_{frame})$$

- Από το παραπάνω χρονικό διάστημα, μόνο ο χρόνος $n \cdot t_{frame}$ αφορά τη μετάδοση δεδομένων ενώ ο υπόλοιπος χρόνος αποτελεί επιβάρυνση, συνεπώς ο βαθμός χρήσης ζεύξης (αποδοτικότητα ζεύξης) δίνεται ως εξής:

$$U = (n \cdot t_{frame}) / [n \cdot (2 \cdot t_{prop} + t_{frame})] \Rightarrow U = t_{frame} / (2 \cdot t_{prop} + t_{frame})$$

- Χρησιμοποιώντας τον λόγο α: $U = 1 / (1 + 2 \cdot a)$

- Ο λόγος α εκφράζεται και ως: $a = (d / V) / (L / R) = (R \cdot d) / (V \cdot L)$, συνεπώς:

$$U = (V \cdot L) / (V \cdot L + 2 \cdot R \cdot d)$$

Παράδειγμα

- Μία δορυφορική ζεύξη που συνδέει δύο σταθμούς (αποστολέα και δέκτη) χρησιμοποιεί τεχνική ελέγχου ροής παύσης και αναμονής και διαθέτει τα παρακάτω χαρακτηριστικά: ρυθμό μετάδοσης δεδομένων 64 kbps, μήκος πλαισίου δεδομένων 2048 bytes, χρόνο διάδοσης 180 ms, μήκος πλαισίου επιβεβαίωσης 100 bytes, χρόνο επεξεργασίας σταθμών 10 ms. Υπολογίζουμε το χρόνο αποστολής ενός αρχείου 40960 bytes (μαζί με τη λήψη επιβεβαιώσεων) και το βαθμό χρήσης της ζεύξης.

- Χρόνος μετάδοσης πλαισίου: $t_{frame} = L / R = (2048 \cdot 8) / 64000 = 0.256$ sec = 256 ms

- Χρόνος μετάδοσης επιβεβαίωσης: $t_{ack} = (100 \cdot 8) / 64000 = 12.5$ ms

- Χρόνος μετάδοσης ενός πλαισίου και λήψης επιβεβαίωσης:

$$T_F = t_{frame} + t_{prop} + t_{proc} + t_{ack} + t_{prop} + t_{proc} = 648.5 \text{ ms}$$

- Συνολικός χρόνος αποστολής για αρχείο 40960 bytes ή 40960 / 2048 = 20 πλαίσια:

$$T = n \cdot (t_{prop} + t_{frame} + t_{proc} + t_{prop} + t_{ack} + t_{proc}) = 20 \cdot 648.5 \text{ ms} = 12.97 \text{ s}$$

- Βαθμός χρήσης ζεύξης:

Ο ρυθμός διέλευσης δεδομένων που μεταδίδονται από τον αποστολέα είναι: $U \cdot R = 0.395 \cdot 64$ kbps = 25.3 kbps

$$U = (n \cdot t_{frame}) / [n \cdot (t_{prop} + t_{frame} + t_{proc} + t_{prop} + t_{ack} + t_{proc})] \Rightarrow U = 0.3947 \sim 39.5\ \%$$

Έλεγχος ροής ολισθαίνοντος παραθύρου

- Ο βαθμός χρήσης ζεύξης (αποδοτικότητα) μπορεί να βελτιωθεί εάν επιτρέπεται σε πολλά πλαίσια να βρίσκονται συγχρόνως σε κατάσταση διέλευσης, γεγονός που επιτυγχάνεται με την τεχνική ελέγχου ροής ολισθαίνοντος παραθύρου (sliding window).
- Όταν ο δέκτης διαθέτει χώρο ενδιάμεσης μνήμης για W πλαίσια, ο πομπός μπορεί να στείλει μέχρι W πλαίσια χωρίς να αναμένει για επιμέρους επιβεβαιώσεις.
- Κάθε πλαίσιο περιλαμβάνει ετικέτα με έναν αριθμό ακολουθίας.
- Ο δέκτης λοιπόν λαμβάνει ένα πλήθος πλαισίων (έως W), παρακρατά τις επιμέρους επιβεβαιώσεις λήψης και επιστρέφει μία επιβεβαίωση λήψης που περιλαμβάνει τον αριθμό ακολουθίας του επόμενου αναμενόμενου πλαισίου.
- Με αυτή την επιβεβαίωση λήψης των πλαισίων που στάλθηκαν, ο δέκτης δηλώνει συγχρόνως ότι είναι έτοιμος να λάβει τα επόμενα πλαίσια (έως W), ξεκινώντας από τον αριθμό που περιέχει η ετικέτα της επιβεβαίωσης λήψης.
- Ο πομπός διατηρεί κατάλογο αριθμών ακολουθίας των πλαισίων που μπορεί να στείλει και ο δέκτης διατηρεί αντίστοιχο κατάλογο για τα πλαίσια που είναι προετοιμασμένος να λάβει. Κάθε κατάλογος θεωρείται ως ένα παράθυρο πλαισίων.
- Κάθε φορά που στέλνεται ένα πλαίσιο το παράθυρο πομπού μικραίνει και κάθε φορά που παραλαμβάνεται μία επιβεβαίωση λήψης το παράθυρο μεγαλώνει (ολισθαίνον παράθυρο).
- Για ετικέτα αριθμού ακολουθίας k bits το εύρος αριθμών ακολουθίας είναι από 0 έως $2^k - 1$ και το μέγιστο μέγεθος παραθύρου (W) που μπορεί να χρησιμοποιηθεί είναι $2^k - 1$ πλαίσια.

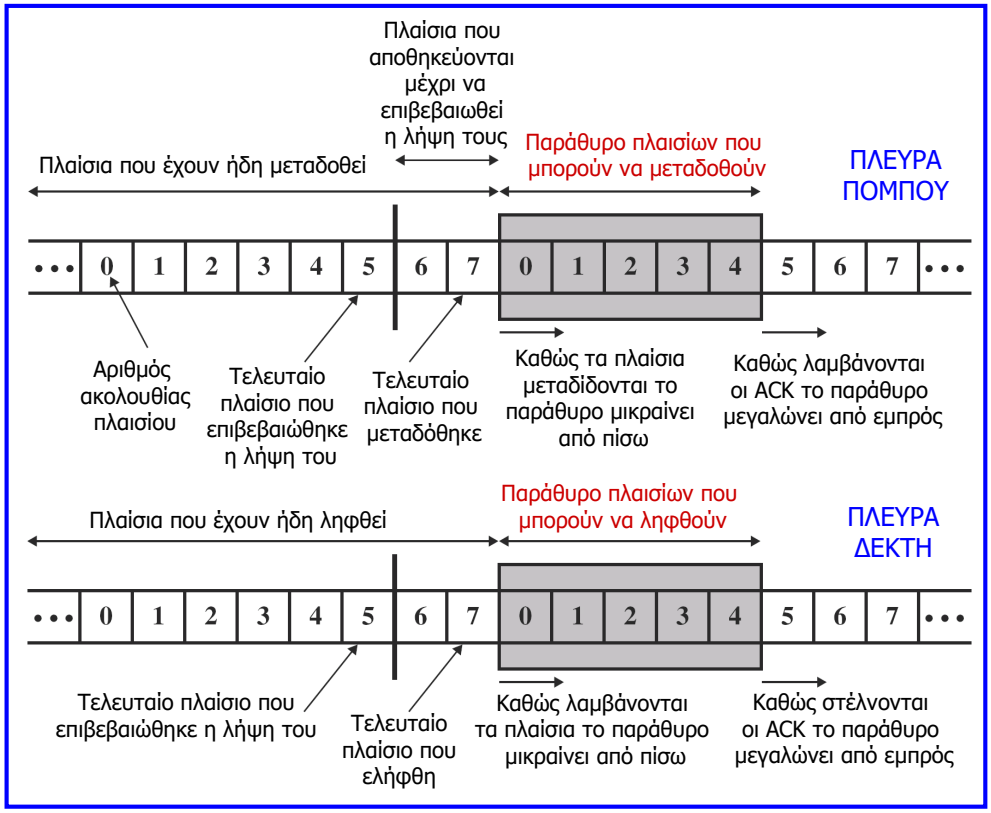
Έλεγχος ροής ολισθαίνοντος παραθύρου

Στη διπλανή απεικόνιση:

Χρησιμοποιείται μέγιστο μέγεθος παραθύρου επτά πλαισίων (W=7)

Επίσης, πεδίο αριθμού ακολουθίας 3 bits, συνεπώς γίνεται αρίθμηση πλαισίων από 0 έως 7 και επαναχρησιμοποίηση των ίδιων αριθμών για τα επόμενα πλαίσια.

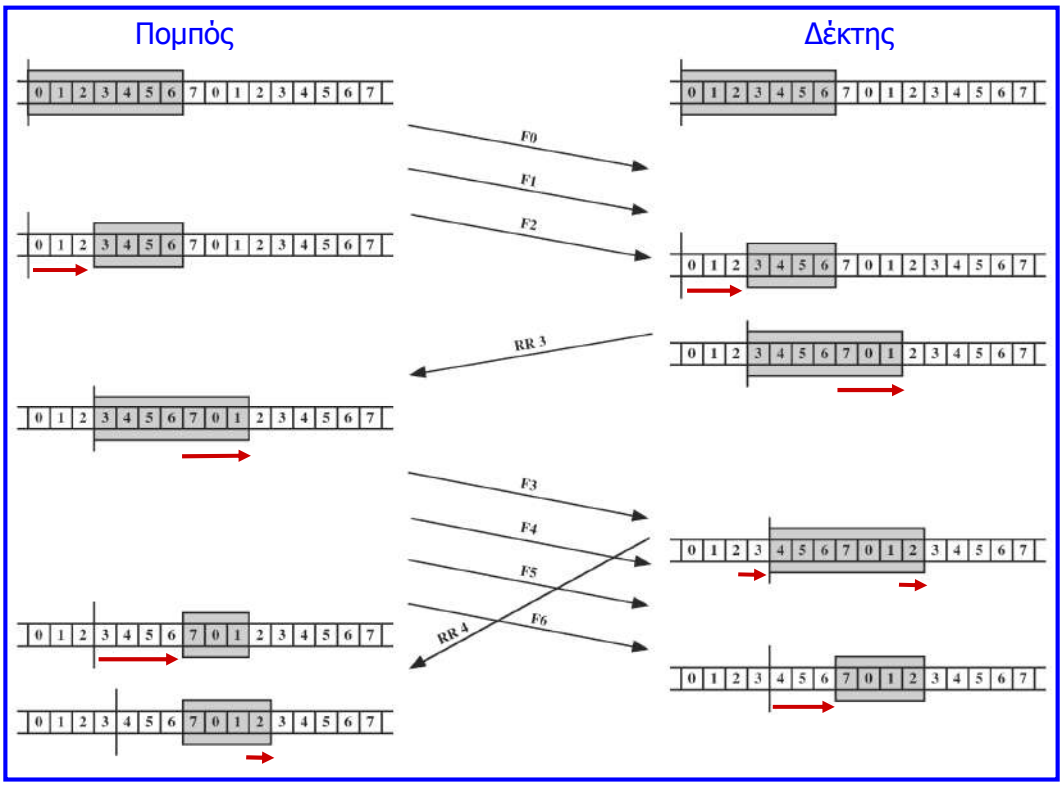
Στην παρούσα φάση, ο πομπός μπορεί να μεταδώσει 5 πλαίσια, αρχίζοντας από το πλαίσιο 0.



Έλεγχος ροής ολισθαίνοντος παραθύρου

Αριθμός ακολουθίας 3 bit και μέγιστο μέγεθος παραθύρου W=7 πλαίσια

Επιβεβαί-ωση λήψης = πλαίσιο RR: ready to receive (έτοιμος για λήψη)



Βαθμός χρήσης ζεύξης ολισθαίνοντος παραθύρου

- Θεωρούμε ξανά ότι ο χρόνος επεξεργασίας σε κάθε σταθμό, ώστε αυτός να ανταποκριθεί σε ένα εισερχόμενο γεγονός, είναι αμελητέος, ότι το πλαίσιο επιβεβαίωσης λήψης είναι πολύ μικρό και ότι ο δέκτης επιβεβαιώνει κάθε πλαίσιο που λαμβάνει.
- Στη ζεύξη ολισθαίνοντος παραθύρου, η αρχή του πρώτου πλαισίου φθάνει στο δέκτη σε χρόνο t_{prop}, απορροφάται εξ' ολοκλήρου σε χρόνο $t_{prop} + t_{frame}$ και η επιβεβαίωση του πρώτου πλαισίου φθάνει στον αποστολέα σε χρόνο $T_F = 2 \cdot t_{prop} + t_{frame}$.
- Υπάρχουν δύο πιθανά σενάρια που αφορούν το μέγιστο μέγεθος παραθύρου W.
- 1ο σενάριο: το πλαίσιο ACK φθάνει στον αποστολέα πριν αυτός εξαντλήσει το παράθυρό του, δηλ. $W \cdot t_{frame} \geq T_F$ (ή $W \geq 2 \cdot a + 1$). Συνεπώς, ο αποστολέας μπορεί να μεταδίδει συνεχώς χωρίς παύση και επομένως προκύπτει βέλτιστος βαθμός χρήσης ζεύξης: $U = 1$.
- 2ο σενάριο: ο αποστολέας εξαντλεί το παράθυρό του μέσα στο χρόνο T_F, δηλαδή $W \cdot t_{frame} < T_F$ (ή $W < 2 \cdot a + 1$), και δεν μπορεί να στείλει πρόσθετα πλαίσια πριν το τέλος του χρόνου αυτού.
- Συνεπώς, αφού σε χρόνο T_F μεταδίδονται από τον αποστολέα W πλαίσια, ο βαθμός χρήσης ζεύξης είναι:

$$U = (W \cdot t_{frame}) / T_F = (W \cdot t_{frame}) / (2 \cdot t_{prop} + t_{frame}) = W / (2 \cdot a + 1) \text{ για } W < 2 \cdot a + 1$$

- Συνοψίζοντας: $U = 1$ για $W \geq 2 \cdot a + 1$ και $U = W / (2 \cdot a + 1)$ για $W < 2 \cdot a + 1$.

Χρόνος αποστολής σε ζεύξη ολισθαίνοντος παραθύρου

- Στο 1ο σενάριο ($W \geq 2a + 1$), όπου ο αποστολέας μπορεί να μεταδίδει πλαίσια συνεχώς χωρίς παύση, ο χρόνος από την έναρξη μετάδοσης ενός συνόλου δεδομένων (n πλαίσια) έως και τη λήψη των αντίστοιχων πλαισίων επιβεβαίωσης (συνολικός χρόνος αποστολής n πλαισίων), είναι:

$$T = n \cdot t_{frame} + 2 \cdot t_{prop}$$

- Στο 2ο σενάριο ($W < 2a + 1$), όπου ο αποστολέας εξαντλεί το παράθυρό του μέσα στο χρόνο T_F και δεν μπορεί να στείλει πρόσθετα πλαίσια πριν το τέλος του χρόνου αυτού, ο χρόνος από την έναρξη μετάδοσης συνόλου δεδομένων (n πλαίσια) έως και τη λήψη των αντίστοιχων πλαισίων επιβεβαίωσης (συνολικός χρόνος αποστολής n πλαισίων), είναι:

$$T = \left\lceil \frac{n}{W} \right\rceil \cdot T_F = \left\lceil \frac{n}{W} \right\rceil (2 \cdot t_{prop} + t_{frame})$$

Η σχέση είναι προσεγγιστική και η ακρίβειά της είναι υψηλή όταν n >> W.

όπου $\lceil x \rceil$ είναι ο μικρότερος ακέραιος με τιμή μεγαλύτερη ή ίση του x.

- Η περίπτωση όπου W = 1 αντιστοιχεί στην περίπτωση ελέγχου ροής παύσης και αναμονής.
- Στην περίπτωση όπου οι χρόνοι t_{proc} και t_{ack} δε θεωρούνται αμελητέοι, οι παραπάνω σχέσεις θα πρέπει να διαφοροποιηθούν κατάλληλα.

Βελτιώσεις πρωτοκόλλου ολισθαίνοντος παραθύρου

- Πολλά πρωτόκολλα επιτρέπουν στο δέκτη να διακόψει τη ροή των πλαισίων από τον πομπό με την αποστολή ενός πλαισίου RNR (receive not ready, μη έτοιμος για λήψη), το οποίο επιβεβαιώνει τη λήψη των προηγούμενων πλαισίων, αλλά απαγορεύει την αποστολή μελλοντικών πλαισίων.
- Σε κάποιο επόμενο σημείο, ο δέκτης πρέπει να στείλει μία κανονική επιβεβαίωση λήψης για να ανοίξει πάλι το παράθυρο.
- Στην περίπτωση αμφίδρομης επικοινωνίας, κάθε σταθμός πρέπει να διατηρεί δύο παράθυρα (μετάδοσης και λήψης) και να στέλνει δεδομένα και επιβεβαιώσεις λήψης.
- Για την υποστήριξη της αμφίδρομης επικοινωνίας, συνήθως χρησιμοποιείται η τεχνική εμβόλιμης επιβεβαίωσης λήψης (piggy-backing):
  - ✓ Κάθε πλαίσιο δεδομένων περιλαμβάνει ένα πεδίο με τον αριθμό ακολουθίας του και ένα πεδίο που κρατά τον αριθμό ακολουθίας για επιβεβαίωση λήψης.
  - ✓ Όταν ένας σταθμός διαθέτει δεδομένα προς αποστολή και μία επιβεβαίωση λήψης, τα στέλνει και τα δύο μαζί εξοικονομώντας χωρητικότητα επικοινωνίας, όταν δεν έχει δεδομένα αλλά μόνο επιβεβαίωση λήψης στέλνει ένα πλαίσιο RR (ή RNR) και όταν έχει μόνο δεδομένα θα πρέπει να επαναλάβει τον τελευταίο αριθμό ακολουθίας επιβεβαίωσης λήψης που έστειλε.

Πιθανότητες σφαλμάτων ψηφίου και πλαισίου

- Τα σφάλματα κατά τη μετάδοση έχουν ως συνέπεια την αλλαγή ενός ή περισσότερων δυαδικών ψηφίων (bits) μέσα σε ένα πλαίσιο κατά την αποστολή του μεταξύ 2 σταθμών.
- Εάν P_b είναι η πιθανότητα να υπάρχει ένα εσφαλμένο δυαδικό ψηφίο (bit) σε ένα πλαίσιο και L ο αριθμός των δυαδικών ψηφίων (μέγεθος) του πλαισίου, τότε:
  - ✓ $P_1 = (1 - P_b)^L$ είναι πιθανότητα ένα πλαίσιο να φθάσει στο δέκτη χωρίς εσφαλμένα bits.
  - ✓ $P_2 = 1 - P_1 = 1 - (1 - P_b)^L$ είναι η πιθανότητα ένα πλαίσιο να φθάσει στο δέκτη με ένα ή περισσότερα εσφαλμένα bits.
- Η πιθανότητα, ένα πλαίσιο να φθάσει στο δέκτη χωρίς εσφαλμένα bits, μειώνεται όταν αυξάνεται η πιθανότητα να υπάρχει ένα εσφαλμένο bit.
- Η πιθανότητα ένα πλαίσιο να φθάσει στο δέκτη χωρίς εσφαλμένα bits, μειώνεται όταν αυξάνεται το μέγεθος του πλαισίου (L), αφού το μεγάλο μέγεθος πλαισίου συνεπάγεται περισσότερα bits στο ίδιο πλαίσιο και φυσικά υψηλότερη πιθανότητα ένα από αυτά να μεταδοθεί εσφαλμένα.
- Όταν δύο τερματικοί σταθμοί συνδέονται μέσω ενδιάμεσων κόμβων με n άλματα (π.χ. μέσω δικτύου μεταγωγής πακέτου), τότε: $P_2 = 1 - (1 - P_b)^{n \cdot L}$.

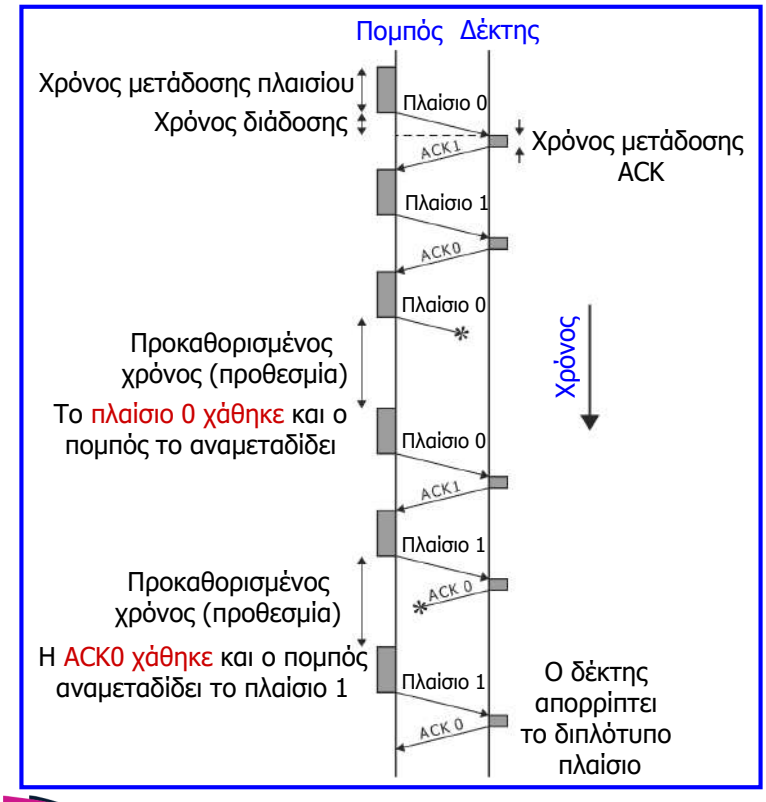
Έλεγχος σφαλμάτων

- Ο έλεγχος σφαλμάτων (error control) αναφέρεται σε μηχανισμούς ανίχνευσης και διόρθωσης σφαλμάτων που συμβαίνουν κατά τη μετάδοση πλαισίων. Μπορούν να συμβούν δύο τύποι σφαλμάτων:
  - ✓ Χαμένο πλαίσιο: πλαίσιο που αποτυγχάνει να φτάσει στο δέκτη, λόγω του ότι ένας καταιγισμός θορύβου κατέστρεψε το πλαίσιο μέχρι το σημείο που ο δέκτης να μην αναγνωρίζει ότι ένα πλαίσιο έχει μεταδοθεί.
  - ✓ Κατεστραμμένο (εσφαλμένο) πλαίσιο: αναγνωρίσιμο πλαίσιο με τις τιμές μερικών από τα bits να έχουν αντιστραφεί κατά τη διάρκεια της μετάδοσης (εσφαλμένα bits).
- Οι τεχνικές ελέγχου σφαλμάτων βασίζονται στα ακόλουθα χαρακτηριστικά:
  - ✓ ανίχνευση σφαλμάτων,
  - ✓ θετική επιβεβαίωση λήψης που επιστρέφεται από το δέκτη για τα επιτυχώς (χωρίς σφάλματα) λαμβανόμενα πλαίσια,
  - ✓ αναμετάδοση από τον πομπό ενός πλαισίου που δεν έχει επιβεβαιωθεί η λήψη του μετά από προκαθορισμένο χρονικό διάστημα (εκπνοή προθεσμίας, timeout),
  - ✓ αρνητική επιβεβαίωση λήψης από το δέκτη για τα εσφαλμένα πλαίσια και αναμετάδοση των εσφαλμένων πλαισίων από τον πομπό.
- Οι τεχνικές ελέγχου σφαλμάτων αναφέρονται ως μηχανισμοί αυτόματης αίτησης επανάληψης (ARQ, automatic repeat request).

ARQ παύσης και αναμονής

- Ο μηχανισμός ARQ παύσης και αναμονής βασίζεται στην ομώνυμη τεχνική ελέγχου ροής.
- Ο πομπός μεταδίδει ένα πλαίσιο και πρέπει να αναμείνει για επιβεβαίωση λήψης ώστε να στείλει το επόμενο πλαίσιο.
- Σε περιπτώσεις κατεστραμμένου ή χαμένου πλαισίου, που στάλθηκε από τον πομπό, ο δέκτης στην πρώτη περίπτωση το ανιχνεύει και το απορρίπτει, ενώ στη δεύτερη δεν το αναγνωρίζει.
- Ο πομπός, που διαθέτει χρονόμετρο, αναμένει επιβεβαίωση λήψης και αν δεν τη λάβει μέχρι τη λήξη προκαθορισμένου χρόνου (προθεσμία), τότε επαναμεταδίδει το πλαίσιο, αφού θα πρέπει να διατηρεί αντίγραφο του πλαισίου που μεταδόθηκε μέχρι να παραλάβει την αντίστοιχη επιβεβαίωση λήψης.
- Στην περίπτωση κατεστραμμένης ή χαμένης επιβεβαίωσης λήψης, ο πομπός δεν αναγνωρίζει την επιβεβαίωση λήψης και μόλις λήξει ο προκαθορισμένος χρόνος στέλνει ξανά το ίδιο πλαίσιο.
- Για να αποφευχθούν τα διπλότυπα πλαίσια στο δέκτη, τα πλαίσια και οι επιβεβαιώσεις αριθμούνται εναλλάξ με 0 ή 1 (ACK0 επιβεβαιώνει την λήψη πλαισίου 1 και ACK1 επιβεβαιώνει τη λήψη πλαισίου 0).
- Ο δέκτης που λαμβάνει δύο πλαίσια σε σειρά με την ίδια αρίθμηση (π.χ. 1), απορρίπτει το δεύτερο πλαίσιο και επιστρέφει επιβεβαίωση λήψης ACK0.
- Πλεονέκτημα: απλότητα Μειονέκτημα: χαμηλή απόδοση

ARQ παύσης και αναμονής



Ο βαθμός χρήσης ζεύξης (U) με σφάλματα, μετά την προσθήκη του μηχανισμού ARQ παύσης και αναμονής, προκύπτει με διαίρεση του U χωρίς σφάλματα με τον αναμενόμενο αριθμό μεταδόσεων ενός πλαισίου (N_r):

$$U = t_{frame} / [N_r \cdot (2 \cdot t_{prop} + t_{frame})]$$

$$\Rightarrow U = 1 / [N_r \cdot (1 + 2 \cdot a)] =$$

$$(1 - P) / (1 + 2 \cdot a)$$

$$N_r = 1 / (1 - P)$$

P: πιθανότητα ενός εσφαλμένου πλαισίου

$$a = t_{prop} / t_{frame}$$

Χρόνος αποστολής n πλαισίων:

$$T = N_r \cdot n \cdot (2 \cdot t_{prop} + t_{frame})$$

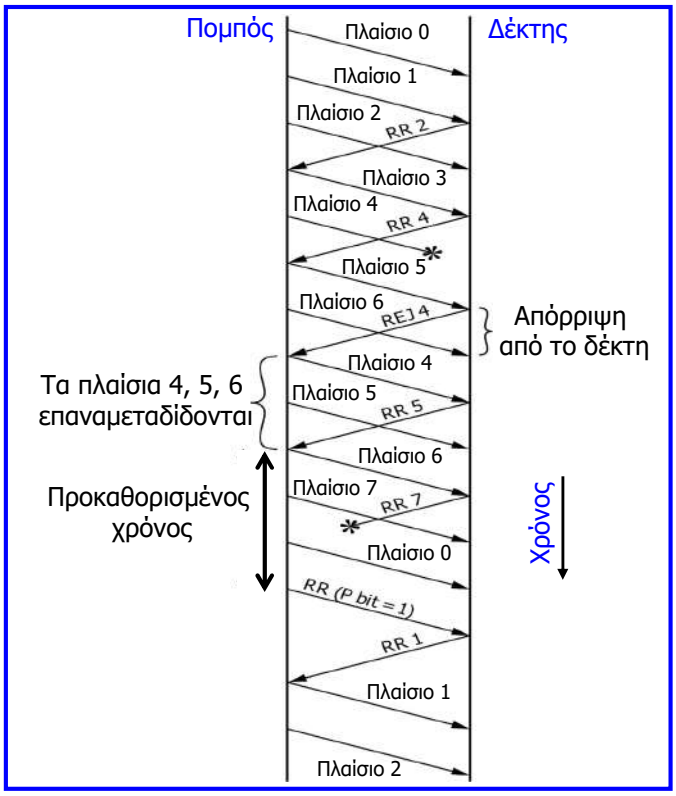
ARQ οπισθοδρόμησης κατά N

- Ο μηχανισμός ARQ οπισθοδρόμησης κατά N (go back N), βασίζεται στην τεχνική ελέγχου ροής ολισθαίνοντος παραθύρου.
- Όταν δε συμβαίνει κανένα σφάλμα, ο δέκτης επιβεβαιώνει τη λήψη των πλαισίων (με πλαίσιο RR) σύμφωνα με την τεχνική ελέγχου ροής ολισθαίνοντος παραθύρου.
- Εάν ο δέκτης ανιχνεύσει σφάλμα σε ένα πλαίσιο, στέλνει ένα πλαίσιο απόρριψης (reject, REJ) για το πλαίσιο αυτό, απορρίπτοντας το, καθώς και όλα τα μελλοντικά λαμβανόμενα πλαίσια μέχρι να ληφθεί σωστά το εσφαλμένο πλαίσιο.
- Όταν ο πομπός λάβει ένα πλαίσιο REJ, πρέπει να αναμεταδώσει το εσφαλμένο πλαίσιο και όλα τα επόμενα πλαίσια που μεταδόθηκαν στο μεσοδιάστημα.
- Εάν ένα μεταδιδόμενο πλαίσιο χαθεί, ο δέκτης δεν εκτελεί καμία ενέργεια και ο πομπός στέλνει το επόμενο πλαίσιο.
- Ο δέκτης λαμβάνει το επόμενο πλαίσιο εκτός σειράς και στέλνει απόρριψη (REJ) για το προηγούμενο, οπότε ο πομπός αναμεταδίδει το εσφαλμένο πλαίσιο και όλα τα επόμενα.
- Εάν ένα μεταδιδόμενο πλαίσιο χαθεί και ο πομπός δε στείλει σύντομα πρόσθετα πλαίσια, ο δέκτης δεν επιστρέφει τίποτα, αλλά όταν λήξει ο προκαθορισμένος χρόνος του πομπού αυτός μεταδίδει ένα πλαίσιο RR με ένα συγκεκριμένο bit (P) ίσο με 1 ως μία εντολή της οποίας η λήψη πρέπει να επιβεβαιωθεί με την αποστολή από το δέκτη ενός RR που να δηλώνει το επόμενο πλαίσιο που αναμένει (δηλαδή το χαμένο πλαίσιο).
- Όταν ο πομπός λάβει το πλαίσιο RR από το δέκτη, επαναμεταδίδει το χαμένο πλαίσιο.

ARQ οπισθοδρόμησης κατά N

- Μπορεί να συμβεί η περίπτωση όπου ο δέκτης λαμβάνει κανονικά ένα πλαίσιο, αλλά χάνεται η επιβεβαίωση λήψης (RR) που περιλαμβάνει τον αριθμό ακολουθίας του επόμενου αναμενόμενου πλαισίου.
- Επειδή οι επιβεβαιώσεις λήψης είναι συσσωρευτικές (δηλαδή η RR6 επιβεβαιώνει τη λήψη των πλαισίων μέχρι και το 5), η επόμενη επιβεβαίωση λήψης μπορεί να φτάσει στον πομπό πριν λήξει ο προκαθορισμένος χρόνος για το πλαίσιο του οποίου η επιβεβαίωση λήψης χάθηκε.
- Εάν όμως λήξει ο προκαθορισμένος χρόνος του πομπού πριν φτάσει η επόμενη επιβεβαίωση λήψης, ο πομπός μεταδίδει ένα πλαίσιο RR με ένα συγκεκριμένο bit (P) ίσο με 1 ως μία εντολή της οποίας η λήψη πρέπει να επιβεβαιωθεί με την αποστολή από το δέκτη ενός πλαισίου RR που να δηλώνει το επόμενο πλαίσιο που αναμένει.
- Εάν ο δέκτης αποτύχει να αποκριθεί στην εντολή RR, μετά από προκαθορισμένο χρόνο ο πομπός θα προσπαθήσει πάλι, αλλά μετά από κάποιο μέγιστο αριθμό αποτυχημένων προσπαθειών κινεί μια διαδικασία επαναφοράς αρχικών συνθηκών (reset).
- Στην περίπτωση που χαθεί ένα πλαίσιο απόρριψης (REJ), ακολουθείται η διαδικασία χαμένου πλαισίου, δηλαδή όταν λήξει ο προκαθορισμένος χρόνος του πομπού, αυτός μεταδίδει ένα πλαίσιο RR με ένα συγκεκριμένο bit (P) ίσο με 1 ως μία εντολή της οποίας η λήψη πρέπει να επιβεβαιωθεί με την αποστολή από το δέκτη ενός πλαισίου RR που να δηλώνει το επόμενο πλαίσιο που αναμένει.

ARQ οπισθοδρόμησης κατά N



ARQ οπισθοδρόμησης κατά N

Ο βαθμός χρήσης ζεύξης (U) στον μηχανισμό ARQ οπισθοδρόμησης κατά N, προκύπτει με διαίρεση του U της τεχνικής ελέγχου ροής ολισθαίνοντος παραθύρου χωρίς σφάλματα με τον αναμενόμενο αριθμό μεταδόσεων ενός πλαισίου (N_r):

$$U = 1 / N_r, \quad W \geq 2 \cdot a + 1$$

$$U = W / [N_r \cdot (1 + 2 \cdot a)], \quad W < 2 \cdot a + 1$$

W: μέγιστο μέγεθος παραθύρου, $a = t_{prop} / t_{frame}$

$$N_r = (1 - P + K \cdot P) / (1 - P)$$

P: πιθανότητα ενός εσφαλμένου πλαισίου

Προκύπτει, ότι ένα εσφαλμένο πλαίσιο, αντί της μετάδοσης ενός πλαισίου, προκαλεί μετάδοση $K = 2 \cdot a + 1$ πλαισίων όταν $W \geq 2 \cdot a + 1$ και $K = W$ πλαισίων όταν $W < 2 \cdot a + 1$:

$$U = (1 - P) / (1 + 2 \cdot a \cdot P) \quad \text{για} \quad W \geq 2 \cdot a + 1$$

$$U = [W \cdot (1 - P)] / [(2 \cdot a + 1) \cdot (1 - P + W \cdot P)] \quad \text{για} \quad W < 2 \cdot a + 1$$

Χρόνος αποστολής n πλαισίων:

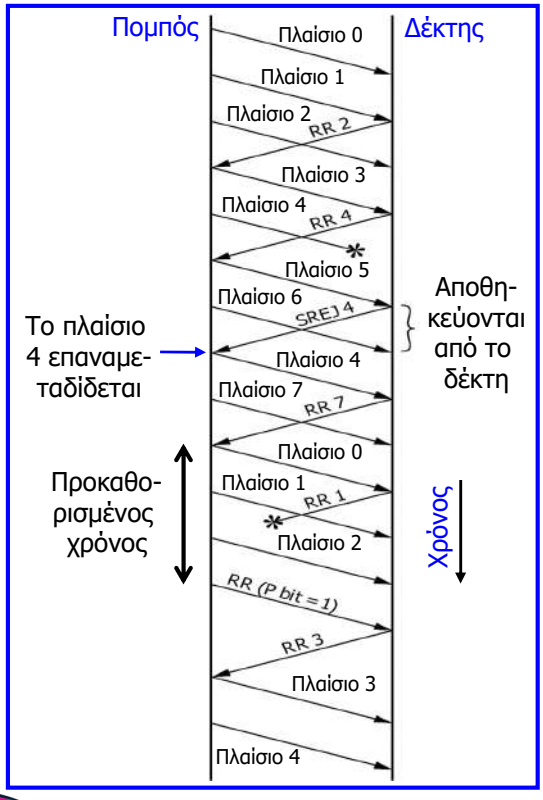
$$T = N_r \cdot n \cdot t_{frame} + 2 \cdot t_{prop}, \quad W \geq 2 \cdot a + 1$$

$$T = N_r \cdot \left\lceil \frac{n}{W} \right\rceil \cdot (2 \cdot t_{prop} + t_{frame}), \quad W < 2 \cdot a + 1$$

ARQ επιλεκτικής απόρριψης

- Στο μηχανισμό ARQ επιλεκτικής απόρριψης (selective reject), τα μόνα πλαίσια που επαναμεταδίδονται είναι εκείνα που έχουν απορριφθεί [ο δέκτης γνωστοποιεί την απόρριψη τους με μία αρνητική επιβεβαίωση λήψης SREJ (selective reject), όταν ανιχνεύσει σφάλμα ή λάβει το επόμενο πλαίσιο εκτός σειράς] ή εκείνα για τα οποία έχει λήξει ο προκαθορισμένος χρόνος του πομπού.
- Μετά την απόρριψη, ο δέκτης συνεχίζει να αποδέχεται τα πλαίσια που ακολουθούν και τα αποθηκεύει σε μία ενδιάμεση μνήμη λήψης μέχρι να ληφθεί το εσφαλμένο ή χαμένο (επαναμεταδιδόμενο) πλαίσιο.
- Ο δέκτης θα πρέπει να τοποθετήσει τα πλαίσια που ακολουθούν σε κατάλληλη κατάταξη, για την παράδοσή τους στο αμέσως υψηλότερο επίπεδο.
- Η τεχνική αυτή ελαχιστοποιεί τον αριθμό των επαναμεταδόσεων, αλλά ο δέκτης θα πρέπει να διαθέτει σχετικά μεγάλη ενδιάμεση μνήμη για να αποθηκεύει τα πλαίσια μέχρι τη λήψη του επαναμεταδιδόμενου πλαισίου, καθώς και ειδικό μηχανισμό για την επανατοποθέτηση των πλαισίων στη σωστή σειρά.
- Επίσης, απαιτείται πολύπλοκη λογική στον πομπό, αφού θα πρέπει να έχει τη δυνατότητα να αποστέλλει κάποια πλαίσια εκτός σειράς.
- Το μέγιστο μέγεθος παραθύρου στην τεχνική αυτή δεν πρέπει να είναι μεγαλύτερο από το μισό του εύρους των αριθμών ακολουθίας, δηλαδή για εύρος αριθμών ακολουθίας 2^k (από 0 έως $2^k - 1$), το μέγιστο μέγεθος παραθύρου περιορίζεται σε $2^k / 2 = 2^{k-1}$.

ARQ επιλεκτικής απόρριψης



Ο βαθμός χρήσης ζεύξης (U) στο μηχανισμό ARQ επιλεκτικής απόρριψης, προκύπτει με διαίρεση του U της τεχνικής ελέγχου ροής ολισθαίνοντος παραθύρου χωρίς σφάλματα, με τον αναμενόμενο αριθμό μεταδόσεων ενός πλαισίου (N_r):

$$U = 1 / N_r \Rightarrow U = 1 - P \quad \text{για } W \geq 2 \cdot a + 1$$

$$U = W / [N_r \cdot (2 \cdot a + 1)] \Rightarrow U = [W \cdot (1 - P)] / (2 \cdot a + 1) \quad \text{για } W < 2 \cdot a + 1$$

W: μέγιστο μέγεθος παραθύρου, $a = t_{prop} / t_{frame}$

$$N_r = 1 / (1 - P)$$

P: πιθανότητα ενός εσφαλμένου πλαισίου

Χρόνος αποστολής n πλαισίων:

$$T = N_r \cdot n \cdot t_{frame} + 2 \cdot t_{prop}, \quad W \geq 2 \cdot a + 1$$

$$T = N_r \cdot \left\lceil \frac{n}{W} \right\rceil \cdot (2 \cdot t_{prop} + t_{frame}) + (N_r - 1) \cdot n \cdot t_{frame}, \quad W < 2 \cdot a + 1$$

Παράδειγμα

Σε ένα κανάλι με χρόνο διάδοσης 45 ms γίνεται μετάδοση πλαισίων σταθερού μεγέθους 128 bytes με ρυθμό μετάδοσης δεδομένων 56 kbps, χρησιμοποιώντας το μηχανισμό ARQ οπισθοδρόμησης κατά N με μέγεθος παραθύρου W = 7. Το μέγεθος των πλαισίων επιβεβαίωσης και ο χρόνος επεξεργασίας στους σταθμούς, θεωρούνται αμελητέες ποσότητες.

A. Υπολογίζουμε το βαθμό χρήσης της ζεύξης όταν η πιθανότητα να υποστεί σφάλμα κάποιο bit δεδομένων είναι $P_b = 2 \cdot 10^{-4}$. Υποθέστε ότι τα σφάλματα είναι ανεξάρτητα και ομοιόμορφα κατανεμημένα.

B. Υπολογίζουμε το συνολικό χρόνο T που απαιτείται για την αποστολή ενός αρχείου 374 KB (ο οποίος συμπεριλαμβάνει και τη λήψη επιβεβαιώσεων), στην περίπτωση ύπαρξης των σφαλμάτων του A.

Παράδειγμα

A. Η πιθανότητα εσφαλμένου bit είναι: $P_b = 2 \cdot 10^{-4}$ και η πιθανότητα να μην υποστεί σφάλμα κάποιο bit είναι: $1 - P_b = 0.9998$.

- Επειδή κάθε πλαίσιο περιέχει 1024 bits ($128 \cdot 8$), η πιθανότητα να μην υποστεί σφάλμα κάποιο πλαίσιο είναι: $(1 - P_b)^L = (1 - P_b)^{1024} = 0.815$.
- Η πιθανότητα να υποστεί σφάλμα ένα πλαίσιο είναι: $P = 1 - 0.815 = 0.185$
- Στον μηχανισμό ARQ οπισθοδρόμησης κατά N κάθε σφάλμα σε πλαίσιο δεδομένων προκαλεί αναμετάδοση K πλαισίων.
- Για $W \geq (2 \cdot a + 1)$, $K = 2 \cdot a + 1$ και για $W < (2 \cdot a + 1)$, $K = W$, όπου $a = t_{prop} / t_{frame}$.
- t_{prop} = χρόνος διάδοσης = 45 ms.
- $t_{frame} = \frac{L}{R} = \frac{8 \cdot 128}{56 \cdot 10^3}\,s = 18.3$ ms, $a = t_{prop} / t_{frame} = 2.46$.
- Επειδή $2 \cdot a + 1 = 5.92 < 7 = W$, σημαίνει ότι κάθε σφάλμα προκαλεί αναμετάδοση $K = 2 \cdot a + 1$ πλαισίων.
- Επομένως, ο βαθμός χρήσης ζεύξης δίνεται ως εξής:

$U = (1 - P) / (1 + 2 \cdot a \cdot P) = (1 - 0.185) / (1 + 2 \cdot 2.46 \cdot 0.185) \Rightarrow U = 0.426$ ή 42.6 %.

Παράδειγμα

B. Για το πρωτόκολλο ARQ οπισθοδρόμησης κατά N, ο αναμενόμενος αριθμός μεταδόσεων ενός πλαισίου N_r δίνεται από την παρακάτω σχέση και επειδή $2 \cdot a + 1 = 5.92 < 7 = W$, σημαίνει ότι $K = 2 \cdot a + 1 = 5.92$:

$$N_r = \frac{1 - P + K \cdot P}{1 - P} \Rightarrow N_r = \frac{1 - P + (2 \cdot a + 1) \cdot P}{1 - P} \Rightarrow N_r = \frac{1 - 0.185 + (2 \cdot 5.92 + 1) \cdot 0.185}{1 - 0.185} = 3.915$$

Το μέγεθος του αρχείου των 374 KB είναι $M = 374 \cdot 1024 = 382976$ bytes. Με μήκος πλαισίου $L_{frame} = 128$ bytes, για τη μετάδοση του αρχείου θα απαιτηθούν:

$$n = \lceil M / L_{frame} \rceil = \lceil 382976 / 128 \rceil = \lceil 2992 \rceil = 2992 \text{ πλαίσια.}$$

Ο συνολικός χρόνος αποστολής T που ζητείται υπολογίζεται ως εξής:

$$T = N_r \cdot n \cdot t_{frame} + 2 \cdot t_{prop} \Rightarrow T = (3.915 \cdot 2992 \cdot 18.3 + 2 \cdot 45) \text{ ms} \Rightarrow T = 214.45 \text{ s.}$$

Στρατηγικές αντιμετώπισης σφαλμάτων

- Έχουν αναπτυχθεί δύο βασικές στρατηγικές αντιμετώπισης σφαλμάτων, οι οποίες προσθέτουν πλεονασματική πληροφορία στα δεδομένα που αποστέλλονται.
- Η πρώτη που χρησιμοποιεί κώδικες διόρθωσης σφαλμάτων, βασίζεται στην προσθήκη πλεονασματικής πληροφορίας στα δεδομένα, έτσι ώστε να μπορέσει ο δέκτης να συμπεράνει ποια είναι τα δεδομένα που διαβιβάστηκαν από τον πομπό.
- Η χρήση των κωδίκων διόρθωσης σφαλμάτων αναφέρεται συχνά ως εμπρόσθια διόρθωση σφαλμάτων (forward error correction, FEC).
- Η δεύτερη στρατηγική που χρησιμοποιεί κώδικες ανίχνευσης σφαλμάτων, βασίζεται στην προσθήκη στα δεδομένα μόνο εκείνης της πληροφορίας που επιτρέπει στο δέκτη να συμπεράνει εάν έχει συμβεί σφάλμα, αλλά όχι ποιο σφάλμα, και να ζητήσει αναμετάδοση.
- Στα κανάλια μετάδοσης που είναι ιδιαίτερα αξιόπιστα, όπως οι οπτικές ίνες, είναι οικονομικότερη η χρήση κώδικα ανίχνευσης σφαλμάτων και η αναμετάδοση των ελαττωματικών δεδομένων (δεύτερη στρατηγική).
- Ωστόσο, στα ασύρματα κανάλια όπου προκύπτουν πολλά σφάλματα, είναι καλύτερα να γίνεται προσθήκη πληροφορίας στα δεδομένα, τέτοια ώστε ο δέκτης να είναι σε θέση να καταλάβει ποια είναι τα δεδομένα που διαβιβάστηκαν (πρώτη στρατηγική).
- Οι κώδικες διόρθωσης και ανίχνευσης σφαλμάτων δεν μπορούν να αντιμετωπίσουν όλα τα πιθανά σφάλματα.

Κώδικες διόρθωσης σφαλμάτων

- Στους κώδικες διόρθωσης σφαλμάτων, που αναφέρονται ως δομικοί κώδικες (block codes), τα πλαίσια που αποστέλλονται (κωδικολέξεις, codewords) αποτελούνται από m δυαδικά ψηφία δεδομένων και r πλεονασματικά ψηφία, τα οποία υπολογίζονται ως συνάρτηση των δυαδικών ψηφίων δεδομένων με τα οποία συνδέονται.
- Το συνολικό μήκος των κωδικολέξεων είναι n = m + r δυαδικά ψηφία και οι κώδικες αναφέρονται ως κώδικες (n, m).
- Έτσι, έχουμε ένα σύνολο από λέξεις δεδομένων (datawords) με m δυαδικά ψηφία και ένα σύνολο από κωδικολέξεις με n = m + r δυαδικά ψηφία.
- Με n δυαδικά ψηφία σχηματίζονται 2^n κωδικολέξεις, ενώ με m δυαδικά ψηφία σχηματίζονται 2^m λέξεις δεδομένων και αφού n > m, το πλήθος των πιθανών κωδικολέξεων είναι μεγαλύτερο από εκείνο των πιθανών λέξεων δεδομένων.
- Κάθε λέξη δεδομένων κωδικοποιείται πάντα από μία κωδικολέξη, συνεπώς ένα πλήθος $2^n - 2^m$ κωδικολέξεων δε χρησιμοποιούνται και αναφέρονται ως μη έγκυρες κωδικολέξεις.
- Παράδειγμα: υποθέτουμε κώδικα διόρθωσης σφαλμάτων με m = 2, n = 5 και τις λέξεις δεδομένων και κωδικολέξεις του διπλανού πίνακα. Για τη διόρθωση των σφαλμάτων, ο δέκτης πρέπει να καταλάβει ποια είναι κάθε φορά η κωδικολέξη που έχει σταλεί.

Λέξεις δεδομένων	Κωδικολέξεις
00	00000
01	01011
10	10101
11	11110

Κώδικες διόρθωσης σφαλμάτων

- Υποθέτουμε ότι η λέξη δεδομένων που πρόκειται να σταλεί είναι η 01 και ο πομπός χρησιμοποιώντας το σχετικό πίνακα (ή στη γενική περίπτωση χρησιμοποιώντας κάποιον αλγόριθμο) σχηματίζει την αντίστοιχη κωδικολέξη 01011.
- Η κωδικολέξη αποστέλλεται στον δέκτη, αλλά κατά τη μετάδοσή της συμβαίνει σφάλμα στο δεύτερο από δεξιά ψηφίο, με αποτέλεσμα ο δέκτης να λάβει την κωδικολέξη 01001, η οποία είναι μη έγκυρη.
- Ο δέκτης που γνωρίζει τις έγκυρες κωδικολέξεις, λαμβάνοντας μία μη έγκυρη, έχει ήδη ανιχνεύσει ότι έχει συμβεί σφάλμα και υποθέτοντας ότι έχει συμβεί απλό σφάλμα (δηλαδή έχει αλλάξει η τιμή μόνο ενός ψηφιού), θα πρέπει να καταλάβει ποια κωδικολέξη έχει σταλεί και κατά συνέπεια να εντοπίσει την αντίστοιχη λέξη δεδομένων.
- Συγκρίνει την εσφαλμένη κωδικολέξη που έλαβε με όλες τις έγκυρες κωδικολέξεις και διαπιστώνει ότι η έγκυρη κωδικολέξη που εστάλη είναι η 01011, αφού μόνο αυτή διαφέρει σε ένα μόνο ψηφίο συγκρινόμενη με τη ληφθείσα εσφαλμένη κωδικολέξη.
- Τέλος, διορθώνει το σφάλμα αντικαθιστώντας την εσφαλμένη κωδικολέξη 01001 με την 01011 και από τον πίνακα διαπιστώνει ότι η αντίστοιχη λέξη δεδομένων είναι η 01.
- Οι περισσότεροι κώδικες διόρθωσης σφαλμάτων βασίζονται στην **απόσταση Hamming D** μεταξύ δύο λέξεων, η οποία είναι το πλήθος των δυαδικών ψηφίων των οποίων η τιμή πρέπει να αλλάξει στη μία λέξη, ώστε οι δύο λέξεις να ταυτιστούν. Για παράδειγμα, οι λέξεις 00000 και 01011 έχουν απόσταση Hamming D = 3.

Κώδικες διόρθωσης σφαλμάτων

- Το σημαντικό στην ανάπτυξη ενός κώδικα διόρθωσης σφαλμάτων είναι η επιλογή των έγκυρων κωδικολέξεων, ώστε να είναι δυνατή η διόρθωση σφαλμάτων.
- Κάθε κώδικας διόρθωσης σφαλμάτων χαρακτηρίζεται από 3 παραμέτρους: το μέγεθος των κωδικολέξεων (n), το μέγεθος των λέξεων δεδομένων (m) και την ελάχιστη απόσταση Hamming D_{min} μεταξύ των κωδικολέξεων που περιλαμβάνει. Στον κώδικα (n, m) = (5, 2) του προηγούμενου παραδείγματος προκύπτει εύκολα ότι $D_{min} = 3$.
- Όταν μια κωδικολέξη τροποποιείται κατά τη μετάδοση, η απόσταση Hamming μεταξύ της κωδικολέξης που στάλθηκε και της κωδικολέξης που ελήφθη ισούται με τον αριθμό των ψηφίων στα οποία έχει συμβεί σφάλμα.
- Παρατηρούμε ότι, ο δέκτης του παραδείγματος, υποθέτοντας ότι έχει συμβεί απλό σφάλμα (δηλαδή έχει αλλάξει η τιμή μόνο ενός ψηφιού), επέλεξε ως έγκυρη κωδικολέξη εκείνη με την μικρότερη απόσταση Hamming από την εσφαλμένη λέξη που έλαβε.
- Συμπεραίνουμε, ότι με τις επιλεγμένες έγκυρες κωδικολέξεις που έχουν $D_{min} = 3$, εξασφαλίζεται η διόρθωση απλού σφάλματος και γενικεύοντας προκύπτει ότι η διόρθωση N σφαλμάτων προϋποθέτει ελάχιστη απόσταση Hamming $D_{min} = 2 \cdot N + 1$.
- Παρατηρούμε επίσης, ότι όταν σταλεί οποιαδήποτε έγκυρη κωδικολέξη και συμβούν 2 σφάλματα, προκύπτουν πάντα μη έγκυρες κωδικολέξεις, δηλαδή ο κώδικας του παραδείγματος με $D_{min} = 3$ ανιχνεύει 2 σφάλματα και γενικεύοντας προκύπτει ότι η ανίχνευση N σφαλμάτων προϋποθέτει ελάχιστη απόσταση Hamming $D_{min} = N + 1$.

Κώδικας Hamming

- Ο κώδικας Hamming ανήκει στην κατηγορία των δομικών κωδικών (block codes) διόρθωσης σφάλματος, έχοντας τη δυνατότητα προσδιορισμού της θέσης του σφάλματος στη μεταδιδόμενη κωδικολέξη, έτσι ώστε να μπορούν να ανακτηθούν τα ορθά δεδομένα.
- Στον κώδικα Hamming, δημιουργούνται κωδικολέξεις στις οποίες ορισμένα ψηφία ελέγχου (ισοτιμίας) συνδυάζονται με επιλεγμένες ομάδες ψηφίων δεδομένων.
- Κατά τη λήψη κάθε κωδικολέξης, ανιχνεύεται τυχόν σφάλμα μέσω ελέγχου ισοτιμίας και σχηματίζεται ένας δυαδικός αριθμός (σύνδρομο σφάλματος, error syndrome), του οποίου η δεκαδική τιμή δηλώνει τη θέση του ψηφίου του οποίου η τιμή έχει αλλάξει (σφάλμα) κατά τη μετάδοση, ώστε αυτό να μπορεί να διορθωθεί με απλή συμπλήρωση.
- Στον κώδικα Hamming τα ψηφία μιας λέξης είναι αριθμημένα διαδοχικά, ξεκινώντας με το ψηφίο 1 στο αριστερό άκρο.
- Οι θέσεις που αποτελούν δυνάμεις του 2 (1, 2, 4, 8, 16 ...) είναι για ψηφία ελέγχου (ισοτιμίας) και οι υπόλοιπες (3, 5, 6, 7, 9 ....) «γεμίζουν» με τα ψηφία δεδομένων.
- Τα ψηφία του συνδρόμου σφάλματος ορίζουν την ισοτιμία μιας ομάδας ψηφίων, συμπεριλαμβανομένων των ψηφίων ελέγχου (ισοτιμίας), ώστε η ισοτιμία της ομάδας ψηφίων να είναι άρτια ή περιττή.

Κώδικας Hamming

- Ένα ψηφίο δεδομένων συμμετέχει σε αρκετούς υπολογισμούς για την εύρεση ισοτιμίας και για τον υπολογισμό των ψηφίων του συνδρόμου σφάλματος.
- Για να βρεθεί σε ποιο ψηφίο ελέγχου (ισοτιμίας) ή σε πιο ψηφίο του συνδρόμου σφάλματος συμβάλλει το ψηφίο δεδομένων της θέσης k, γράφουμε το k ως άθροισμα δυνάμεων του 2.
- Για παράδειγμα: 9 = 1 + 8, δηλαδή το 9ο ψηφίο δεδομένων συμμετέχει στον υπολογισμό ισοτιμίας που αφορά τα ψηφία ελέγχου (ισοτιμίας) P_1 και P_8 και στον υπολογισμό των ψηφίων του συνδρόμου σφάλματος C_1 και C_8.
- Στον υπολογισμό των ισοτιμιών δεν συμμετέχουν τα ψηφία ελέγχου, ενώ στον υπολογισμό των ψηφίων του συνδρόμου σφάλματος συμμετέχουν και τα ψηφία ελέγχου.
- Με αυτόν τον τρόπο, προκύπτει για παράδειγμα ότι στην περίπτωση 8 ψηφίων δεδομένων και 4 ψηφίων ελέγχου, δηλαδή για κώδικα (12, 8), στον υπολογισμό του ψηφίου ελέγχου P_1 συμμετέχουν τα ψηφία 3, 5, 7, 9, 11, στον υπολογισμό του P_2 συμμετέχουν τα ψηφία 3, 6, 7, 10, 11, στον υπολογισμό του P_4 συμμετέχουν τα ψηφία 5, 6, 7, 12 και στον υπολογισμό του P_8 συμμετέχουν τα ψηφία 9, 10, 11, 12.
- Προκύπτει ότι, στον υπολογισμό των ψηφίων ελέγχου P_1, P_2, P_4 και P_8, ξεκινώντας από το αντίστοιχο ψηφίο ελέγχου, συμμετέχουν ανά 1, ανά 2, ανά 4 και ανά 8 τα ψηφία δεδομένων, αντίστοιχα.
- Ίδιες είναι και οι ομάδες υπολογισμού των ψηφίων του συνδρόμου σφάλματος (C_1, C_2, C_4 και C_8, αντίστοιχα), με τη διαφορά ότι σε αυτές συμμετέχουν και τα ψηφία ελέγχου.

Κώδικας Hamming – Παράδειγμα

Θέση	1	2	3	4	5	6	7	8	9	10	11	12
Ψηφίο	**P_1**	**P_2**	**1**	**P_4**	**0**	**0**	**1**	**P_8**	**1**	**0**	**1**	**0**
Ψηφία δεδομένων για τον υπολογισμό του **P_1**	✓		✓		✓		✓		✓		✓	
Ψηφία δεδομένων για τον υπολογισμό του **P_2**		✓	✓			✓	✓			✓	✓	
Ψηφία δεδομένων για τον υπολογισμό του **P_4**				✓	✓	✓	✓					✓
Ψηφία δεδομένων για τον υπολογισμό του **P_8**								✓	✓	✓	✓	✓

Αφού συγκροτήσαμε τις κατάλληλες ομάδες ψηφίων δεδομένων, υπολογίζουμε τα ψηφία ελέγχου για άρτια ισοτιμία:

P_1 = {1,0,1,1,1} = **0**
P_2 = {1,0,1,0,1} = **1**
P_4 = {0,0,1,0} = **1**
P_8 = {1,0,1,0} = **0**

→

Λέξη που αποστέλλεται:
0 1 1 **1** 0 0 1 **0** 1 0 1 0

Κώδικας Hamming – Παράδειγμα

Θέση	1	2	3	4	5	6	7	8	9	10	11	12
Ψηφίο	P_1	P_2	1	P_4	0	0	1	P_8	1	0	1	0
Ψηφία λέξης για τον υπολογισμό του C_1	✓		✓		✓		✓		✓		✓	
Ψηφία λέξης για τον υπολογισμό του C_2		✓	✓			✓	✓			✓	✓	
Ψηφία λέξης για τον υπολογισμό του C_4				✓	✓	✓	✓					✓
Ψηφία λέξης για τον υπολογισμό του C_8								✓	✓	✓	✓	✓

A. Έστω ότι λαμβάνεται η λέξη:
0 1 1 1 0 **0** 1 0 1 0 1 0

B. Έστω ότι λαμβάνεται η λέξη:
0 1 1 1 0 **1** 1 0 1 0 1 0

Υπολογίζουμε τα ψηφία του συνδρόμου σφάλματος ($C_8C_4C_2C_1$) για άρτια ισοτιμία:

$C_1 = \{0,1,0,1,1,1\} = \mathbf{0}$
$C_2 = \{1,1,0,1,0,1\} = \mathbf{0}$
$C_4 = \{1,0,0,1,0\} = \mathbf{0}$
$C_8 = \{0,1,0,1,0\} = \mathbf{0}$

$C_8C_4C_2C_1 = 0000 \Rightarrow$ σωστή λήψη

$C_1 = \{0,1,0,1,1,1\} = \mathbf{0}$
$C_2 = \{1,1,1,1,0,1\} = \mathbf{1}$
$C_4 = \{1,0,1,1,0\} = \mathbf{1}$
$C_8 = \{0,1,0,1,0\} = \mathbf{0}$

$\rightarrow 2 + 4 = 6$

$C_8C_4C_2C_1 = 0110 = 6_{10} \Rightarrow$ σφάλμα στη θέση 6

Κώδικας Hamming

- Ο τρόπος ανάπτυξής του κώδικα Hamming, παρέχει έναν κώδικα με ελάχιστη απόσταση Hamming μεταξύ των έγκυρων κωδικολέξεών του ίση με 3, που σημαίνει ότι εκτός της διόρθωσης απλού σφάλματος, ο κώδικας επιτυγχάνει ανίχνευση 2 σφαλμάτων.
- Στον κώδικα Hamming, το πλήθος (r) των ψηφίων ελέγχου (ισοτιμίας) που απαιτούνται για τη διόρθωση απλού σφάλματος, πρέπει να ικανοποιεί την ακόλουθη σχέση:

$$r + m + 1 \leq 2^r$$

- Από την παραπάνω σχέση προκύπτει ο πίνακας που ακολουθεί:

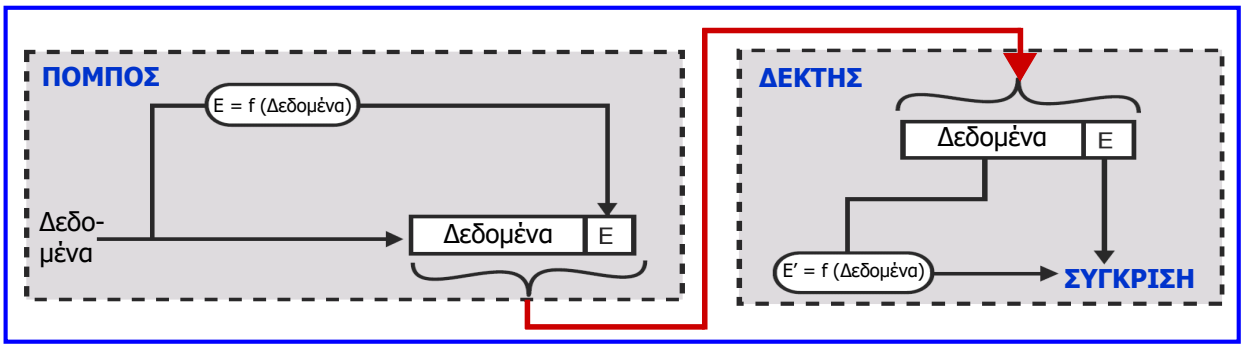
Πλήθος ψηφίων ελέγχου (r)	Πλήθος ψηφίων δεδομένων (m)
3	2 – 4
4	5 – 11
5	12 – 26
6	27 – 57

Παράδειγμα:
$4 + 11 + 1 = 2^4$

- Για την διόρθωση πολλαπλών σφαλμάτων, χρησιμοποιούνται πιο πολύπλοκοι κώδικες και περισσότερα ψηφία ελέγχου.

Κώδικες ανίχνευσης σφαλμάτων

- Η λειτουργία των κωδίκων ανίχνευσης σφαλμάτων, βασίζεται στην προσθήκη από τον πομπό σε κάθε μεταδιδόμενη ενότητα δεδομένων, επιπλέον δυαδικών ψηφίων που αποτελούν τον κώδικα ανίχνευσης σφάλματος (E).
- Η προσθήκη αυτή επιτρέπει στο δέκτη να συμπεράνει εάν έχει συμβεί σφάλμα, αλλά όχι ποιο σφάλμα, και να ζητήσει αναμετάδοση.
- Βασικοί κώδικες ανίχνευσης σφαλμάτων:
  - ✓ ισοτιμία (parity),
  - ✓ άθροισμα ελέγχου (checksum) και
  - ✓ κυκλικός έλεγχος πλεονασμού (CRC: cyclic redundancy check).



Ισοτιμία

- Η απλούστερη μέθοδος ανίχνευσης σφαλμάτων είναι ο έλεγχος ισοτιμίας (parity check), κατά την οποία γίνεται προσθήκη ενός δυαδικού ψηφίου ισοτιμίας στο τέλος μίας ενότητας δεδομένων, του οποίου η τιμή είναι τέτοια ώστε η μεταδιδόμενη ενότητα να περιέχει άρτιο (άρτια ισοτιμία) ή περιττό αριθμό μονάδων (περιττή ισοτιμία).
- Έτσι, όταν αποστέλλεται η ενότητα δεδομένων 1011010 με άρτια ισοτιμία, γίνεται προσθήκη ενός μηδενικού στο τέλος της ενότητας ώστε αυτή να γίνει 10110100, ενώ όταν αποστέλλεται με περιττή ισοτιμία, γίνεται προσθήκη μιας μονάδας και η ενότητα δεδομένων 1011010 γίνεται 10110101.
- Ωστόσο, εάν για παράδειγμα στην περίπτωση άρτιας ισοτιμίας αντιστραφεί άρτιο πλήθος από ψηφία δεδομένων λόγω σφαλμάτων, δημιουργείται μη ανιχνεύσιμο σφάλμα.
- Ο έλεγχος ισοτιμίας μπορεί να ανιχνεύσει αξιόπιστα μόνο απλό σφάλμα σε μια ενότητα δεδομένων.
- Η μέθοδος δεν είναι ασφαλής, ιδιαίτερα για υψηλούς ρυθμούς μετάδοσης, όπου είναι πολύ πιθανή η αλλοίωση της τιμής περισσοτέρων του ενός ψηφίων δεδομένων.

Άθροισμα ελέγχου

- Ο όρος άθροισμα ελέγχου χρησιμοποιείται για να δηλώσει μια ομάδα από δυαδικά ψηφία ελέγχου που συνδέονται με ένα μήνυμα, ανεξαρτήτως του τρόπου υπολογισμού τους.
- Μια ομάδα από ψηφία ισοτιμίας είναι ένα παράδειγμα αθροίσματος ελέγχου.
- Ωστόσο, υπάρχουν και άλλα ισχυρότερα αθροίσματα ελέγχου που βασίζονται σε ένα άθροισμα των ψηφίων μιας ενότητας δεδομένων.
- Συνήθως, το άθροισμα ελέγχου προσαρτάται στα δεδομένα, ως το συμπλήρωμα ως προς 2 της συνάρτησης του αθροίσματος.
- Τα σφάλματα μπορούν να ανιχνευτούν στο δέκτη με πρόσθεση ολόκληρης της ληφθείσας λέξης, δηλαδή τόσο των ψηφίων δεδομένων όσο και των ψηφίων του αθροίσματος ελέγχου και αν το αποτέλεσμα είναι 0, σημαίνει ότι δεν έχει εντοπιστεί σφάλμα.
- Παράδειγμα:
  - ✓ Το άθροισμα ελέγχου της επικεφαλίδας ενός πακέτου δεδομένων του διαδικτύου (IP) υπολογίζεται ως το συμπλήρωμα ως προς 1 του αθροίσματος των 10 λέξεων με 16 bits της επικεφαλίδας (θεωρώντας αρχικά 0 τα 16 bits του αθροίσματος ελέγχου).
  - ✓ Το άθροισμα ελέγχου προσαρτάται (ως η $6^η$ λέξη με 16 bits) στην επικεφαλίδα και εάν στο δέκτη το συμπλήρωμα ως προς 1 του αθροίσματος όλων των λέξεων με 16 bits της επικεφαλίδας είναι 0, σημαίνει ότι δεν έχει εντοπιστεί σφάλμα.

Κυκλικός έλεγχος πλεονασμού

- Ασφαλής και αποτελεσματική τεχνική ανίχνευσης σφαλμάτων είναι ο κυκλικός έλεγχος πλεονασμού (cyclic redundancy check, CRC).
- Για μία ενότητα δεδομένων από k δυαδικά ψηφία, ο πομπός παράγει μία ακολουθία από n δυαδικά ψηφία που αναφέρεται ως ακολουθία ελέγχου πλαισίου (frame check sequence, FCS), ώστε το πλαίσιο που προκύπτει και αποτελείται από (k + n) δυαδικά ψηφία να διαιρείται ακριβώς με κάποιο προκαθορισμένο δυαδικό αριθμό.
- Ο δέκτης διαιρεί το εισερχόμενο πλαίσιο με τον προκαθορισμένο αριθμό και αν δεν υπάρχει υπόλοιπο υποθέτει ότι δεν υπήρξε κανένα σφάλμα.
- Ο προσδιορισμός της ακολουθίας FCS προκύπτει με χρήση αριθμητικής modulo-2, όπου οι πράξεις εκτελούνται ψηφίο προς ψηφίο χωρίς να λαμβάνονται υπόψη κρατούμενα και δανεικά ψηφία (η πρόσθεση και η αφαίρεση γίνονται ταυτόσημες με την πράξη XOR).
- Έστω: T = πλαίσιο (k + n) ψηφίων που θα μεταδοθεί, M = ενότητα δεδομένων k ψηφίων, F = FCS n ψηφίων και G = προκαθορισμένος διαιρέτης (n + 1) ψηφίων.
- $T = 2^n \cdot M + F$ (ο πολ/σμός της ενότητας M με 2^n, την μετατοπίζει αριστερά κατά n ψηφία).
- Διαιρούμε το $2^n \cdot M$ με το G & προκύπτει πηλίκο Q & υπόλοιπο R: $(2^n \cdot M) / G = Q + (R / G)$.
- Χρησιμοποιούμε το υπόλοιπο R ως ακολουθία FCS, συνεπώς: $T = 2^n \cdot M + R$.
- Ελέγχουμε εάν το R ικανοποιεί τον όρο ότι η πράξη (T / G) έχει υπόλοιπο 0:

$$T / G = (2^n \cdot M + R) / G = Q + (R / G) + (R / G) = Q$$

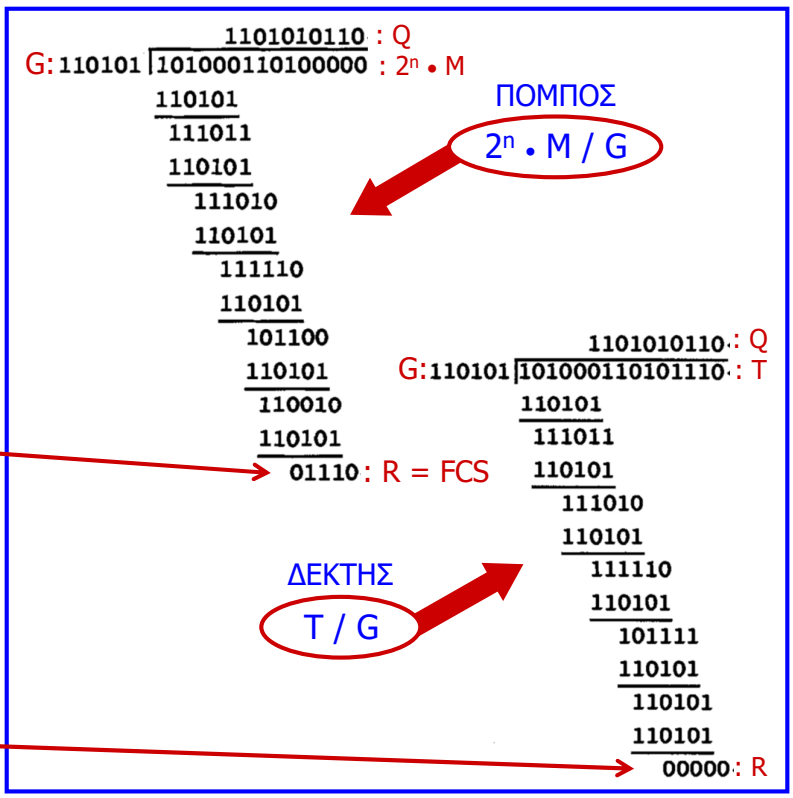
Κυκλικός έλεγχος πλεονασμού

- Η προηγούμενη ισότητα προκύπτει διότι η πρόσθεση αριθμού με τον εαυτό του με χρήση αριθμητικής modulo-2 (δηλαδή η πράξη XOR), δίνει αποτέλεσμα 0.
- Παρατηρούμε λοιπόν ότι το T διαιρείται ακριβώς με το G, συνεπώς το R μπορεί να χρησιμοποιηθεί ως ακολουθία ελέγχου πλαισίου (FCS).
- Ο διαιρέτης G, συνήθως εκφράζεται ως πολυώνυμο G(x) μίας πλασματικής μεταβλητής x, με συντελεστές που αντιστοιχούν στα δυαδικά ψηφία του διαιρέτη G. Το πολυώνυμο G(x) αναφέρεται ως γεννήτρια πολυωνύμων. Για παράδειγμα: η γεννήτρια πολυωνύμων G(x) = $x^5 + x^4 + x^2 + 1$, αντιστοιχεί στον διαιρέτη G = 110101.
- Οι υπόλοιπες ενότητες ψηφίων της διαδικασίας εκφράζονται παρομοίως ως πολυώνυμα.
- Για να προσδιορίσουμε λοιπόν την FCS αρκεί να εκτελέσουμε την πράξη: $2^n \bullet M / G$, με αποτέλεσμα εάν μετά τη λήψη ο δέκτης διαιρέσει το T με το G και δεν προκύψει υπόλοιπο, τότε δεν έχουν υπάρξει σφάλματα κατά τη μετάδοση.
- Ένα σφάλμα οδηγεί στην αντιστροφή ενός ψηφίου, συνεπώς αυτό ισοδυναμεί με την πράξη XOR αυτού του ψηφίου με το 1 (αφού $0 \oplus 1 = 1$ και $1 \oplus 1 = 0$).
- Επομένως, τα σφάλματα σε ένα πλαίσιο από (k + n) ψηφία μπορούν να παρασταθούν μέσω ενός πεδίου E από (k + n) ψηφία με μονάδες στη θέση κάθε σφάλματος και το λαμβανόμενο εσφαλμένο πλαίσιο μπορεί να εκφραστεί ως: $Tr = T \oplus E$.
- Ο δέκτης θα αποτύχει να ανιχνεύσει ένα σφάλμα όταν το Tr διαιρείται ακριβώς με το G και αποδεικνύεται ότι αυτό ισοδυναμεί με το E να διαιρείται ακριβώς με το G.

Κυκλικός έλεγχος πλεονασμού

Παράδειγμα:

- M = 1010001101 (10 bits), G = 110101 (6 bits), FCS = R (5 bits). Το G επιλέγεται ένα bit μακρύτερο από την ακολουθία FCS και τα LSB, MSB του G είναι πάντα μονάδες.
- $2^5 \bullet M$ = 101000110100000.
- Η πράξη ($2^n \bullet M / G$) σε αριθμητική modulo-2 δίνει υπόλοιπο R = 01110 και η πρόσθεση του R στο $2^5 \bullet M$ μας δίνει T = 101000110101110 το οποίο και μεταδίδεται.
- Εάν δεν υπάρξουν σφάλματα, ο δέκτης λαμβάνει το T άθικτο και γι' αυτό κατά την πράξη (T / G) σε αριθμητική modulo-2, θα προκύψει μηδενικό υπόλοιπο.



Κυκλικός έλεγχος πλεονασμού

- Σφάλματα που αντιστοιχούν σε πολυώνυμο E(x) που διαιρείται ακριβώς με το πολυώνυμο G(x), είναι μη ανιχνεύσιμα.
- Αποδεικνύεται ότι οι ακόλουθοι τύποι σφαλμάτων είναι πάντα ανιχνεύσιμοι:
  - ✓ τα απλά σφάλματα (σφάλματα ενός ψηφίου) εφόσον στο πολυώνυμο G(x) μήκους k ψηφίων, περιλαμβάνονται οι όροι x^k και x^0, δηλαδή τα LSB, MSB του διαιρέτη G είναι 1.
  - ✓ τα σφάλματα δύο ψηφίων εφόσον το G(x) περιέχει τουλάχιστον 3 όρους,
  - ✓ οποιοσδήποτε περιττός αριθμός σφαλμάτων εφόσον το G(x) περιέχει τον όρο (x + 1),
  - ✓ οποιαδήποτε συνεχόμενα (burst) σφάλματα, εφόσον το μήκος ακολουθίας σφαλμάτων είναι μικρότερο ή ίσο του μήκους της ακολουθίας FCS, αλλά και οι περισσότερες περιπτώσεις μεγαλύτερων ακολουθιών.
- Οι παρακάτω γεννήτριες πολυωνύμων G(x) χρησιμοποιούνται ευρέως (διεθνή πρότυπα):
  - ✓ CRC-12 = $x^{12} + x^{11} + x^3 + x^2 + x + 1$
  - ✓ CRC-16 = $x^{16} + x^{15} + x^2 + 1$
  - ✓ CRC-CCITT = $x^{16} + x^{12} + x^5 + 1$
  - ✓ CRC-32 = $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Κεφάλαιο 4: Πρωτόκολλα τοπικών δικτύων

Βασικές έννοιες τοπικών δικτύων

- Ένα τοπικό δίκτυο (local area network, LAN) αποτελείται από ένα διαμοιραζόμενο μέσο μετάδοσης καθώς και από υλικό και λογισμικό που υλοποιεί τη διεπαφή των σταθμών (υπολογιστές ή άλλες συσκευές) με το μέσο και καθορίζει τη σειρά πρόσβασης σε αυτό.
- Η μετάδοση από οποιονδήποτε σταθμό μπορεί να ληφθεί από όλους τους άλλους σταθμούς που είναι συνδεδεμένοι στο ίδιο τοπικό δίκτυο.
- Τα συνήθη τοπικά δίκτυα παρέχουν ρυθμούς μετάδοσης δεδομένων 1 – 100 Mbps, αλλά έχουν αναπτυχθεί και τοπικά δίκτυα υψηλών ταχυτήτων (έως μερικά Gbps).
- Μέσα μετάδοσης: συνεστραμμένο ζεύγος καλωδίων (twisted pair cable), ομοαξονικό καλώδιο (coaxial cable), οπτική ίνα (optical fiber), ασύρματη ζεύξη (wireless link).
- Τοπολογίες: οι σταθμοί τοπικών δικτύων συνδέονται σε τοπολογίες δακτυλίου (ring), διαύλου ή αρτηρίας (bus), δένδρου (tree) και αστέρα (star).
- Για την επικοινωνία σταθμών σε LAN έχουν καθοριστεί πρότυπα (όπως η σειρά IEEE 802), που καλύπτουν ένα εύρος ρυθμών μετάδοσης δεδομένων, τοπολογιών & μέσων μετάδοσης.
- Τα τοπικά δίκτυα ενός οργανισμού διασυνδέονται μέσω γεφυρών (bridges).
- Στα τοπικά δίκτυα ο έλεγχος ζεύξης δεδομένων διακρίνεται στον έλεγχο πρόσβασης μέσου (medium access control, MAC) και στον έλεγχο λογικής σύνδεσης (logical link control, LLC).

Πρωτόκολλα τοπικών δικτύων

- Ο τρόπος που έχουν οριστεί τα ανώτερα επίπεδα (από το επίπεδο δικτύου και πάνω) του μοντέλου OSI είναι κατάλληλος για τοπικά δίκτυα και το ίδιο ισχύει για το μοντέλο TCP/IP (από το επίπεδο IP και πάνω).
- Το φυσικό επίπεδο περιλαμβάνει λειτουργίες όπως: κωδικοποίηση και αποκωδικοποίηση σήματος, δημιουργία και αφαίρεση εισαγωγής ή προοιμίου (preamble) σήματος (που χρησιμοποιείται για συγχρονισμό στο δέκτη), μετάδοση και λήψη πληροφορίας.
- Επίσης, περιλαμβάνει τις προδιαγραφές του μέσου μετάδοσης και της τοπολογίας που χρησιμοποιούνται στο τοπικό δίκτυο.
- Το επίπεδο MAC περιλαμβάνει λειτουργίες όπως: δημιουργία πλαισίων δεδομένων προς μετάδοση με προσάρτηση πεδίων διευθύνσεων και ανίχνευσης σφαλμάτων, αναγνώριση διεύθυνσης και ανίχνευση σφαλμάτων κατά τη λήψη και απομάκρυνση των αντίστοιχων πεδίων, έλεγχος πρόσβασης των σταθμών στο μέσο μετάδοσης του τοπικού δικτύου (αλγόριθμοι πολλαπλής πρόσβασης).
- Το επίπεδο LLC περιλαμβάνει λειτουργίες όπως: διεπαφή με τα ανώτερα επίπεδα, έλεγχος ροής και έλεγχος (διόρθωση) σφαλμάτων.
- Για το ίδιο επίπεδο LLC, είναι διαθέσιμες διάφορες επιλογές ελέγχου πρόσβασης μέσου (MAC).

Στατική εκχώρηση καναλιού

- Βασικό ζήτημα στα τοπικά δίκτυα αποτελεί η ταυτόχρονη πρόσβαση πολλών χρηστών σε ένα κανάλι και κατά συνέπεια ο τρόπος εκχώρησης του καναλιού στους χρήστες.
- Το κανάλι μπορεί να είναι τμήμα του ασύρματου φάσματος σε μια γεωγραφική περιοχή ή καλώδιο ή οπτική ίνα, όπου μπορεί να συνδέονται πολλοί κόμβοι.
- Ένας στατικός τρόπος εκχώρησης καναλιού, είναι ο τεμαχισμός της χωρητικότητάς του, μέσω πολυπλεξίας με διαίρεση συχνότητας (frequency division multiplexing, FDM).
- Στην FDM σε κάθε χρήστη (από σύνολο N χρηστών) διατίθεται μια συγκεκριμένη ζώνη συχνοτήτων.
- Η τεχνική αυτή βρίσκει εφαρμογή για μικρό και σταθερό αριθμό χρηστών, όπως οι ραδιοφωνικοί σταθμοί που ο καθένας κατέχει ένα τμήμα της ζώνης FM και το χρησιμοποιεί για να μεταδώσει το σήμα του.
- Όταν, όμως, ο αριθμός των αποστολέων είναι μεγάλος και μεταβαλλόμενος ή η κίνηση είναι υψηλή, η εκχώρηση αυτή παρουσιάζει προβλήματα:
  - ✓ όταν επικοινωνούν < N χρήστες ένα μέρος του φάσματος δε χρησιμοποιείται.
  - ✓ όταν επιθυμούν να επικοινωνήσουν > N χρήστες, κάποιοι από αυτούς δεν μπορούν να το κάνουν, έστω κι αν κάποιοι από τους N χρήστες είναι αδρανείς.
- Η στατική εκχώρηση δεν ενδείκνυται για δίκτυα Η/Υ, όπου η κίνηση δεδομένων είναι υψηλή.

Στατική εκχώρηση καναλιού

- Ό,τι ισχύει για την FDM ισχύει και για άλλους τρόπους στατικής διαίρεσης (εκχώρησης) καναλιού, όπως η πολυπλεξία με διαίρεση χρόνου (time division multiplexing, TDM), κατά την οποία σε κάθε χρήστη διατίθεται κυκλικά μια χρονοθυρίδα.
- Ωστόσο, εάν κάποιοι χρήστες δεν χρησιμοποιήσουν τις χορηγηθείσες χρονοθυρίδες, θα υπάρξει μη χρήση του καναλιού για τα αντίστοιχα χρονικά διαστήματα.
- Η χαμηλή αποδοτικότητα των στατικών σχημάτων εκχώρησης καναλιού, προκύπτει από υπολογισμούς που βασίζονται στη θεωρία ουρών.
- Για να υπολογίζουμε τη μέση χρονική καθυστέρηση T, προκειμένου να σταλεί ένα πλαίσιο σε κανάλι χωρητικότητας C bps, υποθέτουμε ότι τα πλαίσια φθάνουν με ένα μέσο ρυθμό άφιξης λ πλαίσια/sec και ότι τα πλαίσια ποικίλλουν σε μήκος με μέσο μήκος 1/μ bits.
- Ο ρυθμός εξυπηρέτησης του καναλιού είναι $\mu \cdot C$ πλαίσια/sec και η μέση χρονική καθυστέρηση (T) είναι:

$$T = \frac{1}{\mu \cdot C - \lambda}$$

- Η παραπάνω σχέση προκύπτει από ένα σύστημα ουράς M/M/1: χρόνοι μεταξύ αφίξεων με εκθετική κατανομή, χρόνοι εξυπηρέτησης με εκθετική κατανομή, ένας εξυπηρετητής.

Στατική εκχώρηση καναλιού

- Αν η χωρητικότητα καναλιού είναι C = 100 Mbps, το μέσο μήκος του πλαισίου είναι 1/μ = 10000 bits και ο ρυθμός άφιξης πλαισίων είναι λ = 5000 πλαίσια/s, τότε T = 200 μs.
- Εάν αγνοούσαμε την καθυστέρηση της ουράς και λαμβάναμε υπόψη μόνο το χρόνο αποστολής ενός πλαισίου των 10000 bits σε κανάλι των 100 Mbps, θα καταλήγαμε στην εσφαλμένη καθυστέρηση των 100 μs.
- Έστω ότι διαιρούμε το κανάλι επικοινωνίας σε N ανεξάρτητα κανάλια, το καθένα με χωρητικότητα (C / N) bps και υποθέτουμε ότι τα πλαίσια φθάνουν σε κάθε κανάλι με μέσο ρυθμό άφιξης (λ / N) πλαίσια/sec και ότι έχουν μέσο μήκος 1 / μ bits.
- Ο ρυθμός εξυπηρέτησης κάθε καναλιού είναι $\mu \cdot (C/N)$ πλαίσια/sec και η μέση χρονική καθυστέρηση (T), γίνεται:

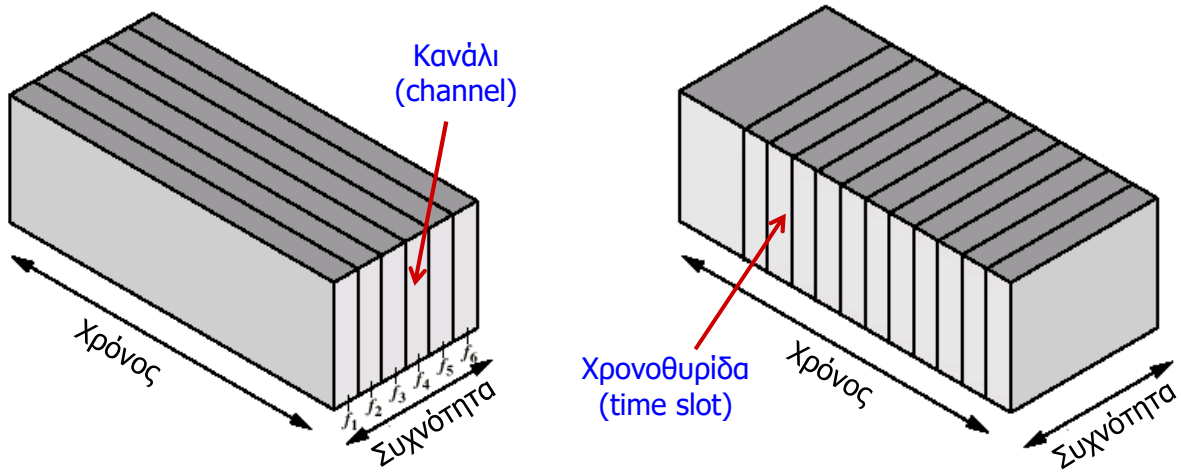
$$T = \frac{1}{\mu \cdot \left(\frac{C}{N}\right) - \left(\frac{\lambda}{N}\right)} = \frac{N}{\mu \cdot C - \lambda}$$

- Προκύπτει ότι, η μέση καθυστέρηση για το διαιρεμένο κανάλι είναι N φορές χειρότερη από την περίπτωση όπου όλα τα πλαίσια είναι τοποθετημένα σε μια ενιαία ουρά.
- Αν λοιπόν αντικαταστήσουμε το κανάλι των 100 Mbps με 10 δίκτυα των 10 Mbps και αποδώσουμε ένα σε κάθε χρήστη, η μέση καθυστέρηση αυξάνεται από 200 μs σε 2 ms.

FDM και TDM

FDM: κάθε σήμα αφού υποστεί κατάλληλη επεξεργασία (διαμόρφωση) μετατοπίζεται σε διαφορετική συχνότητα ή ζώνη συχνοτήτων (κανάλι) και η μετάδοση των σημάτων γίνεται στο ίδιο μέσο ταυτόχρονα, στα διαφορετικά κανάλια.

TDM: τα σήματα (δεδομένα) πολυπλέκονται χρονικά και μεταδίδονται στο ίδιο μέσο, δηλαδή ο χρόνος διαιρείται σε χρονοθυρίδες (time slots) και η μετάδοση των σημάτων γίνεται κυκλικά.



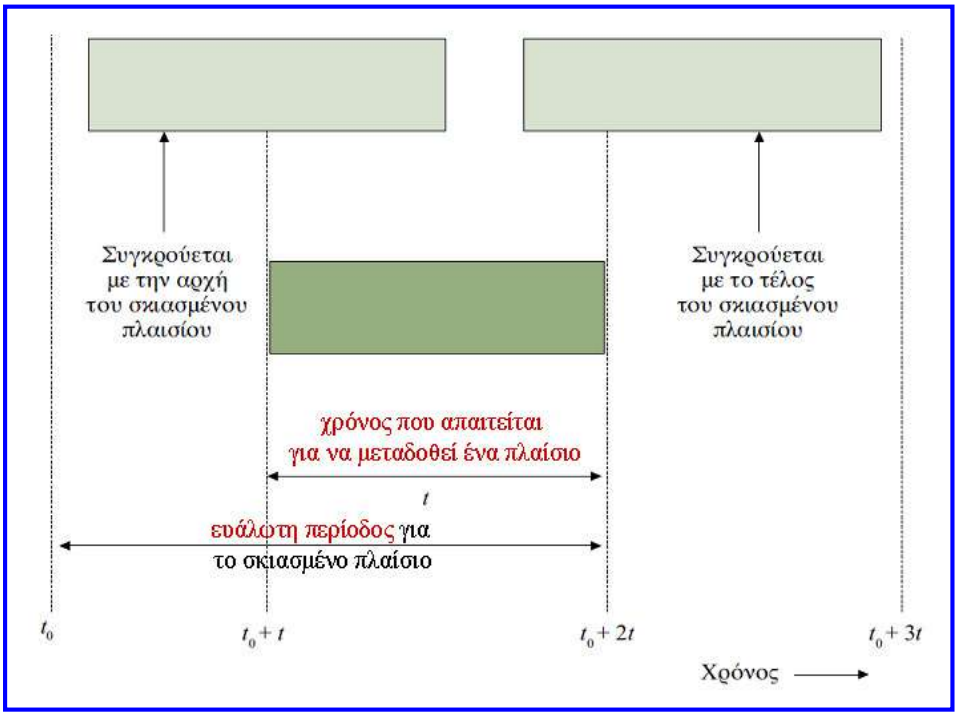
Αλγόριθμοι πολλαπλής πρόσβασης

- Έχουν αναπτυχθεί αρκετοί αλγόριθμοι για τη δυναμική εκχώρηση καναλιού πολλαπλής πρόσβασης (ή τεχνικές ελέγχου πολλαπλής πρόσβασης).
- Η τεχνική ALOHA που αναπτύχθηκε για δίκτυα εκπομπής πακέτων, ανήκει στην κατηγορία μεθόδων τυχαίας πρόσβασης (οι σταθμοί προσπελαύνουν το μέσο τυχαία) ή μεθόδων ανταγωνισμού (ανταγωνίζονται για απόκτηση χρόνου στο μέσο).
- Όταν ένας σταθμός έχει να μεταδώσει ένα πλαίσιο το πράττει και κατόπιν αναμένει για προκαθορισμένη διάρκεια χρόνου λίγο μεγαλύτερη από τη μέγιστη καθυστέρηση διάδοσης με επιστροφή του δικτύου.
- Εάν υπάρξει επιβεβαίωση λήψης στο χρόνο αυτό, ο σταθμός δεν εκτελεί άλλη ενέργεια, ενώ εάν δεν υπάρξει επαναμεταδίδει το πλαίσιο και μετά από έναν προκαθορισμένο αριθμό επαναμεταδόσεων, εγκαταλείπει την προσπάθεια.
- Ο δέκτης αποφασίζει για την ορθότητα ενός εισερχόμενου πλαισίου, εξετάζοντας το πεδίο ακολουθίας ελέγχου πλαισίου (FCS), καθώς επίσης και τη διεύθυνση προορισμού της επικεφαλίδας και εάν το πλαίσιο είναι έγκυρο και η εξεταζόμενη διεύθυνση ταιριάζει, ο δέκτης στέλνει αμέσως επιβεβαίωση λήψης.
- Το πλαίσιο μπορεί να είναι άκυρο και να αγνοηθεί από το δέκτη λόγω θορύβου καναλιού ή λόγω σύγκρουσης με άλλο πλαίσιο (χρονική επικάλυψη πλαισίων, έστω και μερική).
- Οι πολλές συγκρούσεις σε μεγάλη κίνηση, οδηγούν σε μειωμένη αξιοποίηση καναλιού.

Τεχνική ALOHA

Όταν το σκιασμένο πλαίσιο ξεκινά να μεταδίδεται σε χρόνο t_0, εάν άλλος χρήστης έχει δημιουργήσει πλαίσιο μεταξύ t_0 και $t_0 + t$ ή μεταξύ $t_0 + t$ και $t_0 + 2 \cdot t$, τότε το σκιασμένο πλαίσιο θα συγκρουστεί.

Επομένως, η ευάλωτη περίοδος ενός πλαισίου είναι $2 \cdot t$, t: χρόνος μετάδοσης πλαισίου.



Τεχνική ALOHA

- Χρόνος μετάδοσης πλαισίου (t) είναι ο χρόνος που απαιτείται για τη μετάδοση ενός πλαισίου σταθερού μήκους (μήκος του πλαισίου διά το ρυθμό μετάδοσης δεδομένων).
- Υποθέτουμε ότι η δημιουργία νέων πλαισίων από τους σταθμούς γίνεται με μια κατανομή Poisson με μέση τιμή N πλαίσια ανά χρόνο μετάδοσης πλαισίου.
- Για να εξασφαλιστεί ότι το N δε μειώνεται καθώς οι χρήστες μπλοκάρονται, υποθέτουμε άπειρο πλήθος χρηστών.
- Όταν N > 1, τότε παράγονται από το κανάλι πλαίσια με υψηλότερο ρυθμό από αυτόν που μπορεί να χειριστεί, και σχεδόν κάθε πλαίσιο θα υποστεί μια σύγκρουση.
- Οι σταθμοί μεταδίδουν εκτός από τα νέα πλαίσια και τα πλαίσια που είχαν υποστεί συγκρούσεις και υποθέτουμε ότι η δημιουργία παλιών και νέων πλαισίων γίνεται με μια κατανομή Poisson, με μέση τιμή G πλαίσια ανά χρόνο μετάδοσης πλαισίου ($G \geq N$).
- Όταν έχουμε χαμηλό φορτίο στο μέσο υπάρχουν λίγες συγκρούσεις ($G \approx N$), ενώ όταν έχουμε υψηλό φορτίο, υπάρχουν πολλές συγκρούσεις ($G > N$).
- Ο ρυθμός (επιτυχούς) διέλευσης S ανά χρόνο μετάδοσης πλαισίου ή ρυθμοαπόδοση ισούται με το προσφερόμενο φορτίο G επί την πιθανότητα P_0 να έχουμε μια επιτυχή μετάδοση (το πλαίσιο δεν υφίσταται καμιά σύγκρουση): $S = G \cdot P_0$.
- Ένα πλαίσιο δεν θα υποστεί καμία σύγκρουση, εάν δεν υπάρχουν άλλα πλαίσια προς αποστολή εντός της ευάλωτης περιόδου του πλαισίου αυτού.

Τεχνική ALOHA

- Η πιθανότητα δημιουργίας k πλαισίων κατά τη διάρκεια μετάδοσης ενός πλαισίου, στον οποίο αναμένεται η δημιουργία G πλαισίων (δηλαδή G = μέσο πλήθος προσπαθειών μετάδοσης πλαισίων ανά χρόνο μετάδοσης ενός πλαισίου), δίνεται από την κατανομή Poisson:

$$P_r[k] = \frac{G^k \cdot e^{-G}}{k!}$$

- Η πιθανότητα να μην δημιουργηθούν πλαίσια (k = 0) είναι e^{-G}.
- Σε χρονικό διάστημα που ισοδυναμεί με τη μετάδοση 2 πλαισίων (ευάλωτη περίοδος πλαισίου), ο μέσος αριθμός πλαισίων που δημιουργούνται είναι $2 \cdot G$.
- Η πιθανότητα να μην δημιουργηθούν πλαίσια (επιτυχής μετάδοση) κατά το χρονικό διάστημα που ισοδυναμεί με τη μετάδοση 2 πλαισίων, είναι: $P_0 = e^{-2 \cdot G}$.
- Αφού ο ρυθμός διέλευσης S ισούται με το προσφερόμενο φορτίο G επί την πιθανότητα P_0 να έχουμε μια επιτυχή μετάδοση, λαμβάνουμε: $S = G \cdot e^{-2 \cdot G}$.
- Ο μέγιστος ρυθμός διέλευσης υπολογίζεται ως εξής:

$$\frac{dS}{dG} = 0 \Rightarrow \frac{d(G \cdot e^{-2 \cdot G})}{dG} = 0 \Rightarrow -2 \cdot G \cdot e^{-2 \cdot G} + e^{-2 \cdot G} = 0 \Rightarrow G = 0.5$$

$$\frac{d^2S}{dG^2}\Big|_{G=0.5} < 0, \quad S_{max} = S\Big|_{G=0.5} = \frac{1}{2 \cdot e} \Rightarrow S_{max} = 0.18$$

Στην τεχνική ALOHA η μέγιστη δυνατή αξιοποίηση του καναλιού είναι 18%

Τεχνική ALOHA με χρονοθυρίδες

- Για να βελτιωθεί η απόδοσή της, η τεχνική ALOHA τροποποιήθηκε στην ALOHA με χρονοθυρίδες (slotted ALOHA), κατά την οποία ο χρόνος οργανώνεται σε θυρίδες (slots) που το μέγεθός τους ισούται με το χρόνο μετάδοσης ενός πλαισίου.
- Για το συγχρονισμό των σταθμών απαιτείται ένα κεντρικό ρολόι και μία μετάδοση επιτρέπεται να ξεκινήσει μόνο στο όριο μιας θυρίδας, συνεπώς όταν υπάρχει επικάλυψη πλαισίων, αυτή θα είναι ολική.
- Αυτό μειώνει στο μισό την ευάλωτη περίοδο ενός πλαισίου.
- Με βάση όσα προαναφέρθηκαν, η πιθανότητα να μην υπάρχει κυκλοφορία κατά τη διάρκεια της χρονοθυρίδας ενός πλαισίου ισούται με e^{-G}.
- Συνεπώς, στην περίπτωση της τεχνικής ALOHA με χρονοθυρίδες, ο ρυθμός διέλευσης είναι: $S = G \cdot e^{-G}$.
- Ο μέγιστος ρυθμός διέλευσης υπολογίζεται ως εξής:

$$\frac{dS}{dG} = 0 \Rightarrow \frac{d(G \cdot e^{-G})}{dG} = 0 \Rightarrow -G \cdot e^{-G} + e^{-G} = 0 \Rightarrow G = 1$$

$$\frac{d^2S}{dG^2}\Big|_{G=1} < 0, \quad S_{max} = S\Big|_{G=1} = \frac{1}{e} \Rightarrow S_{max} = 0.37$$

Στην τεχνική slotted ALOHA η μέγιστη δυνατή αξιοποίηση του καναλιού είναι 37%

Τεχνική ALOHA με χρονοθυρίδες

- Η πιθανότητα λοιπόν να μην έχουμε σύγκρουση στην τεχνική slotted ALOHA ισούται με e^{-G} και η πιθανότητα να έχουμε μία σύγκρουση ισούται με $(1 - e^{-G})$.
- Επομένως, η πιθανότητα μιας μετάδοσης που απαιτεί k προσπάθειες, δηλαδή k – 1 συγκρούσεις που ακολουθούνται από μια επιτυχή μετάδοση, ισούται με:

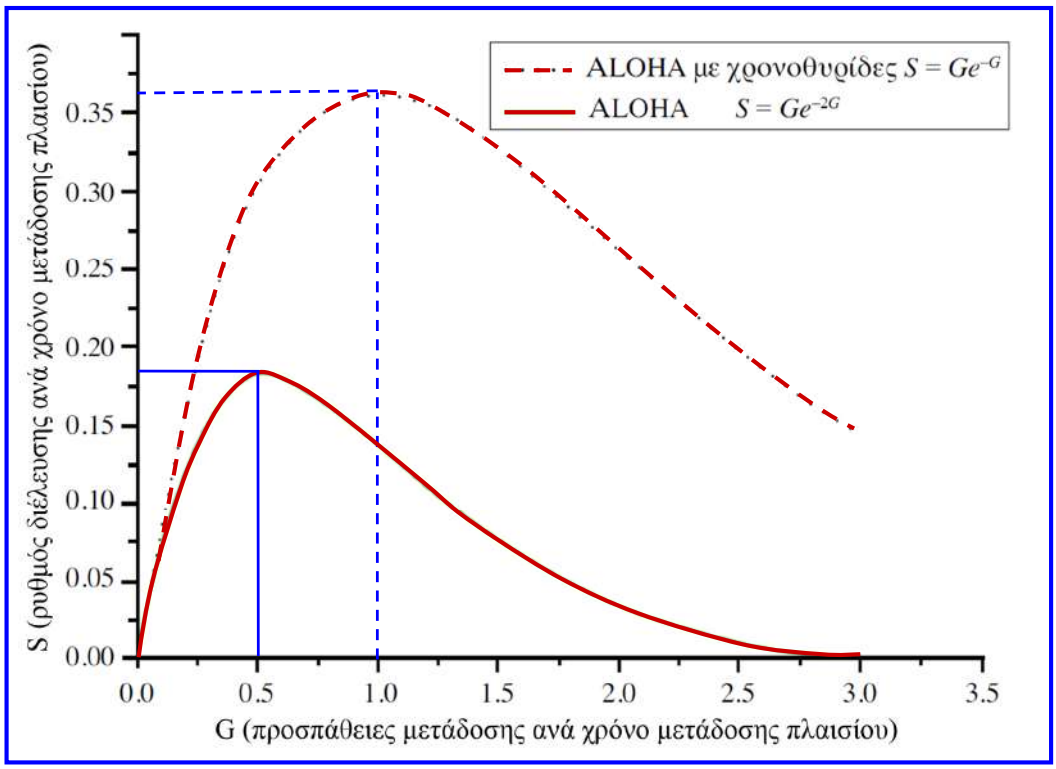
$$P_k = e^{-G} \cdot (1 - e^{-G})^{k-1}$$

- Από τη θεωρία πιθανοτήτων προκύπτει ότι ο αναμενόμενος (μέσος) αριθμός μεταδόσεων (E) ανά πλαίσιο που μεταδίδεται έχει ως εξής:

$$E = \sum_{k=1}^{\infty} k \cdot P_k = \sum_{k=1}^{\infty} k \cdot e^{-G} \cdot (1 - e^{-G})^{k-1} = e^{-G}$$

- Επειδή ο μέσος αριθμός μεταδόσεων εξαρτάται εκθετικά από το G, συμπεραίνουμε ότι μικρή αύξηση του φορτίου του καναλιού, μπορεί να μειώσει δραστικά την απόδοσή του.

Απόδοση τεχνικών ALOHA



Τεχνική CSMA

- Εάν οι σταθμοί διαθέτουν δυνατότητα ανίχνευσης μέσου (αίσθησης φορέα), δεν θα μεταδώσουν εάν υπάρχει σε εξέλιξη άλλη μετάδοση (με δεδομένο ότι ο χρόνος διάδοσης πλαισίου είναι αρκετά μικρότερος από το χρόνο μετάδοσης).

- Έχει αναπτυχθεί η τεχνική CSMA (carrier sense multiple access, αίσθηση φορέα πολλαπλής πρόσβασης), κατά την οποία όταν ένας σταθμός επιθυμεί να μεταδώσει διενεργεί αρχικά έλεγχο του μέσου για να διαπιστώσει εάν είναι σε εξέλιξη άλλη μετάδοση (αίσθηση φορέα) και αν το μέσο είναι αδρανές, μεταδίδει, διαφορετικά θα πρέπει να περιμένει.

- Αν δύο ή περισσότεροι σταθμοί αρχίσουν περίπου ταυτόχρονα να μεταδίδουν, θα συμβεί σύγκρουση και τα δεδομένα των δύο μεταδόσεων δε θα ληφθούν επιτυχώς.

- Κάθε σταθμός πρέπει να περιμένει εύλογο χρονικό διάστημα μετά τη μετάδοση για επιβεβαίωση λήψης (μέγιστη καθυστέρηση διάδοσης με επιστροφή συν το χρόνο που απαιτείται για να αγωνιστεί ο δέκτης για το μέσο, ώστε να επιβεβαιώσει τη λήψη).

- Εάν δεν υπάρξει επιβεβαίωση λήψης, ο σταθμός υποθέτει ότι έχει προκύψει σύγκρουση και αναμεταδίδει το σχετικό πλαίσιο.

Τεχνική CSMA

- Ο μέγιστος βαθμός χρήσης (απόδοση) εξαρτάται από το χρόνο διάδοσης (δηλαδή το μήκος μέσου) και από το χρόνο μετάδοσης (δηλαδή το μέγεθος πλαισίων). Μεγαλύτερα πλαίσια και μικρότεροι χρόνοι διάδοσης οδηγούν σε μεγαλύτερη απόδοση (βαθμό χρήσης):

$$U = 1 / (1 + a), \quad a = t_{prop} / t_{frame}$$

- Ο αλγόριθμος που χρησιμοποιείται για να καθορίσει την ενέργεια ενός σταθμού όταν το μέσο ανιχνευτεί απασχολημένο ονομάζεται επίμονο-1 (1-persistent) και επιβάλει σε έναν σταθμό να διενεργεί συνεχώς έλεγχο στο κανάλι μέχρι αυτό να καταστεί αδρανές και τότε ο σταθμός θα πρέπει να μεταδώσει αμέσως.

- Εάν ωστόσο δύο ή περισσότεροι σταθμοί αναμένουν να μεταδώσουν, τότε θα συμβεί σίγουρα σύγκρουση και η τάξη στο κανάλι αποκαθίσταται μετά από τη σύγκρουση.

- Η τεχνική CSMA αν και πιο αποδοτική από την τεχνική ALOHA είναι ανεπαρκής.

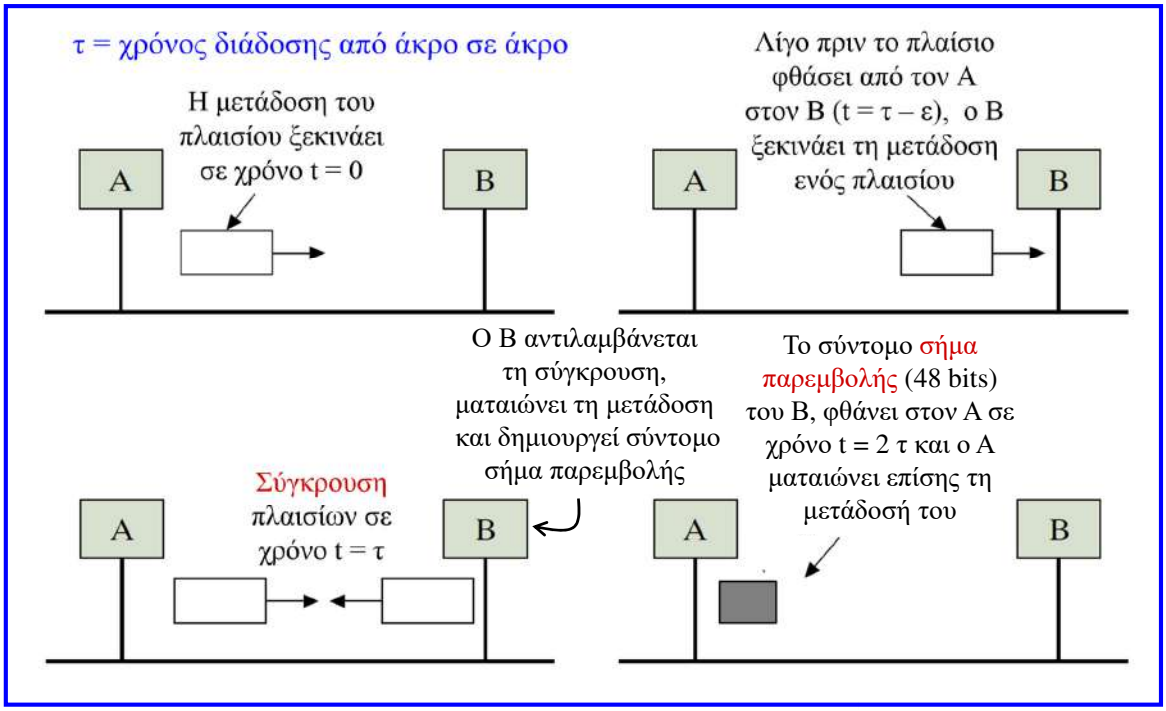
- Όταν δύο πλαίσια συγκρούονται, το μέσο παραμένει αχρησιμοποίητο για τη διάρκεια της μετάδοσης των δύο κατεστραμμένων πλαισίων.

Τεχνική CSMA/CD

Η σπατάλη αυτή του μέσου, περιορίζεται εάν ένας σταθμός εξακολουθεί να διενεργεί έλεγχο στο μέσο ενώ μεταδίδει και αυτό οδηγεί στα εξής βήματα της τεχνικής CSMA/CD (CSMA with collision detection, CSMA με ανίχνευση σύγκρουσης), που χρησιμοποιείται στο πρότυπο για τοπικά δίκτυα IEEE 802.3:

1. Εάν το μέσο είναι αδρανές μετέδωσε αμέσως, αλλιώς εκτέλεσε το βήμα 2.
2. Εάν το μέσο είναι απασχολημένο, συνέχισε να ελέγχεις το μέσο μέχρι να γίνει αδρανές και μετά μετέδωσε αμέσως.
3. Εάν μία σύγκρουση ανιχνευθεί κατά τη διάρκεια της μετάδοσης μετέδωσε ένα σύντομο σήμα παρεμβολής (ή θορύβου) για να εξασφαλίσεις ότι όλοι οι σταθμοί γνωρίζουν ότι έχει συμβεί σύγκρουση και μετά διέκοψε τη μετάδοση.
4. Μετά τη μετάδοση του σήματος παρεμβολής ανέμενε για ένα τυχαίο χρονικό διάστημα και επανέλαβε την εκτέλεση του βήματος 1.

Τεχνική CSMA/CD



Τεχνική CSMA/CD

- Ο χρόνος ανίχνευσης μιας σύγκρουσης δεν υπερβαίνει το διπλάσιο της καθυστέρησης διάδοσης από άκρο σε άκρο ($2 \cdot \tau$ = RTT: round trip time).
- Το διάστημα αυτό αναφέρεται ως χρονοθυρίδα (slot) και ο χρόνος στο μέσο διάδοσης οργανώνεται σε χρονοθυρίδες.
- Τα πλαίσια πρέπει να είναι αρκετά μεγάλα (με ελάχιστο χρόνο μετάδοσης: $P_{min} = 2 \cdot \tau$), έτσι ώστε να ανιχνεύεται η σύγκρουση πριν το τέλος της μετάδοσης.
- Ο χρόνος αναμονής του βήματος 4 στη τεχνική CSMA/CD καθορίζεται από μια τεχνική που αναφέρεται ως δυαδική εκθετική υποχώρηση (binary exponential backoff).
- Με βάση την τεχνική αυτή, μετά από την i-οστή σύγκρουση ενός πλαισίου επιλέγεται ένας τυχαίος αριθμός χρονοθυρίδων από 0 έως $2^i - 1$ για τις οποίες ο σταθμός αναμένει πριν επαναμεταδώσει το πλαίσιο, με ομοιόμορφα κατανεμημένη πιθανότητα επιλογής.
- Έτσι, μετά την 1η σύγκρουση ενός πλαισίου ο σταθμός αναμένει 0 (δηλαδή δεν αναμένει, με πιθανότητα 1/2) ή 1 χρονοθυρίδα (με πιθανότητα 1/2) πριν επαναμεταδώσει, μετά τη 2η σύγκρουση αναμένει 0 ή 1 ή 2 ή 3 χρονοθυρίδες (με πιθανότητα 1/4 σε κάθε περίπτωση), κ.ο.κ.
- Όσο αυξάνεται το πλήθος των προσπαθειών αναμετάδοσης, αυξάνεται η πιθανότητα οι σταθμοί να αποσύρονται με μεγαλύτερο χρόνο αναμονής, μειώνοντας έτσι την πιθανότητα σύγκρουσης.

Τεχνική CSMA/CD

- Επιλέγεται ένα ανώτατο όριο πλήθους συγκρούσεων (i = 10), οπότε από την δέκατη σύγκρουση και μετά, το διάστημα στο οποίο επιλέγεται τυχαία ο αριθμός χρονοθυρίδων θυρίδων αναμονής σταθεροποιείται και η μέγιστη δυνατή αναμονή είναι 1023 ($= 2^{10} - 1$) χρονοθυρίδες.
- Τέλος, μετά από 16 ανεπιτυχείς προσπάθειες μετάδοσης από τον ίδιο σταθμό, ο σταθμός εγκαταλείπει, απορρίπτει το πλαίσιο και δηλώνει σφάλμα στα ανώτερα επίπεδα.
- Αν το τυχαίο διάστημα αναμονής για όλες τις συγκρούσεις ήταν 1023 χρονοθυρίδες, η πιθανότητα να συγκρουστούν δύο σταθμοί για δεύτερη φορά θα ήταν αμελητέα, αλλά ο μέσος όρος αναμονής μετά από μια σύγκρουση θα ήταν εκατοντάδες χρονοθυρίδες, δημιουργώντας έτσι σημαντική καθυστέρηση.
- Επίσης, αν κάθε σταθμός ανέμενε πάντα για 0 ή 1 χρονοθυρίδες, τότε αν 100 σταθμοί προσπαθούσαν να μεταδώσουν ταυτόχρονα, τα πλαίσια θα συγκρούονταν ξανά και ξανά μέχρι 99 από αυτούς να επιλέξουν 1 χρονοθυρίδα και ένας σταθμός να επιλέξει 0.
- Με την εκθετική αύξηση του τυχαίου διαστήματος αναμονής, καθώς προκύπτουν όλο και περισσότερες συνεχόμενες συγκρούσεις, ο αλγόριθμος εξασφαλίζει μια σχετικά μικρή καθυστέρηση όταν μόνο λίγοι σταθμοί συγκρούονται, αλλά επίσης διασφαλίζει ότι οι συγκρούσεις επιλύονται σε εύλογο χρονικό διάστημα όταν συγκρούονται πολλοί σταθμοί.

Απόδοση τεχνικής CSMA/CD

- Θεωρούμε k σταθμούς που είναι πάντα έτοιμοι να μεταδώσουν και επειδή μία ακριβής ανάλυση του αλγορίθμου δυαδικής εκθετικής υποχώρησης είναι περίπλοκη, υποθέτουμε σταθερή πιθανότητα μετάδοσης p σε κάθε χρονοθυρίδα.
- Εάν κάθε σταθμός μεταδίδει με πιθανότητα p κατά τη διάρκεια μιας θυρίδας ανταγωνισμού, η πιθανότητα A κάποιος σταθμός να «καταλάβει» το μέσο στο ίδιο χρονικό διάστημα είναι: $A = p\cdot(1-p)^{k-1} + \ldots + p\cdot(1-p)^{k-1}$ (k όροι στο άθροισμα) $= k\cdot p\cdot(1-p)^{k-1}$.
- Η πιθανότητα A μεγιστοποιείται και τείνει στην τιμή $1/e$ όταν $p = 1/k$ και $k \to \infty$.
- Η πιθανότητα το διάστημα ανταγωνισμού να έχει j θυρίδες είναι $A\cdot(1-A)^{j-1}$, συνεπώς ο μέσος αριθμός θυρίδων ανταγωνισμού δίνεται από τη σχέση:

$$\sum_{j=0}^{\infty} j\cdot A\cdot(1-A)^{j-1} = \frac{1}{A}$$

- Επειδή κάθε θυρίδα έχει διάρκεια $2\cdot\tau$, το μέσο διάστημα ανταγωνισμού είναι $2\cdot\tau\cdot(1/A)$.
- Για βέλτιστη πιθανότητα $p = 1/k$, ο μέσος αριθμός θυρίδων ανταγωνισμού φτάνει έως e, συνεπώς το μέσο διάστημα ανταγωνισμού φθάνει έως $2\cdot\tau\cdot e \approx 5.4\cdot\tau$.
- Απόδοση:

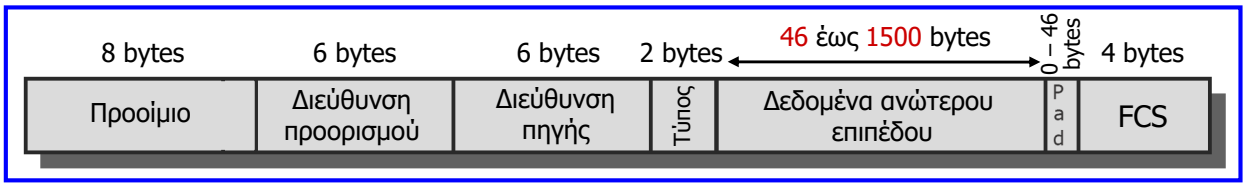
$$U = \frac{P}{P + \frac{2\cdot\tau}{A}},\quad U_{max} = \frac{P}{P + 2\cdot\tau\cdot e \text{ ή } 5.4\cdot\tau} = \frac{1}{1+2\cdot e\cdot\frac{\tau}{P}} = \frac{1}{1+2\cdot e\cdot a} = \frac{1}{1+2\cdot e\cdot\frac{L/c}{F/B}} = \frac{1}{1+\frac{2\cdot e\cdot B\cdot L}{c\cdot F}}$$

P: χρόνος μετάδοσης πλαισίου, τ: χρόνος διάδοσης στο μέσο, $a = \tau / P$, L: μήκος καλωδίου, c: ταχύτητα διάδοσης σήματος, F: μήκος πλαισίου, B: χωρητικότητα ή ρυθμός μετάδοσης δικτύου. Ρυθμός επιτυχούς μετάδοσης στο δίκτυο (ρυθμοαπόδοση δικτύου) $= B\cdot U$

Ethernet

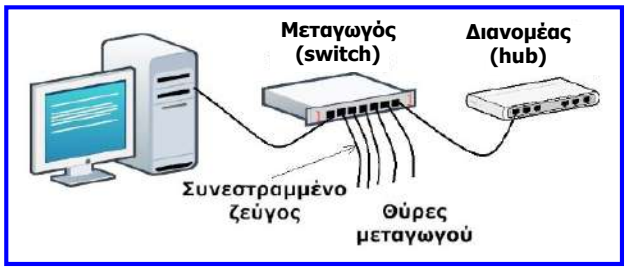
- Η τεχνική CSMA/CD χρησιμοποιείται στο πρότυπο IEEE 802.3 και στο Ethernet που είναι το πιο διαδεδομένο είδος τοπικού δικτύου και έχει πολλές ομοιότητες με το IEEE 802.3.
- Το Ethernet ως πρότυπο τοπικού δικτύου που αναπτύχθηκε από τις εταιρίες DEC, Intel και Xerox (Ethernet II, 1982), καθορίζει εκτός από το φυσικό επίπεδο και το επίπεδο ελέγχου ζεύξης δεδομένων (DLC), χωρίς να το διακρίνει στα υποεπίπεδα MAC (medium access control, IEEE 802.3) και LLC (logical link control, IEEE 802.2).
- Έχουν αναπτυχθεί δύο είδη Ethernet:
  - ✓ το κλασικό Ethernet, που αντιμετωπίζει το ζήτημα της πολλαπλής πρόσβασης στο μέσο, όπως προαναφέρθηκε,
  - ✓ και το Ethernet με μεταγωγή (switched Ethernet), στο οποίο χρησιμοποιούνται ειδικές συσκευές που αναφέρονται ως μεταγωγείς (switches) για να συνδέσουν διαφορετικούς υπολογιστές.
- Το κλασικό Ethernet αποτελεί την αρχική μορφή του τοπικού δικτύου που λειτουργούσε με ρυθμούς δεδομένων 3 – 10 Mbps.
- Το Ethernet με μεταγωγή είναι αυτό στο οποίο μετατράπηκε το κλασικό Ethernet και χρησιμοποιείται σήμερα, λειτουργώντας με ρυθμούς δεδομένων 100 Mbps, 1000 Mbps και 10000 Mbps, με τις αντίστοιχες μορφές του να αναφέρονται ως Fast Ethernet, Gigabit Ethernet και 10 Gigabit Ethernet.

Πλαίσιο Ethernet



- Προοίμιο (preamble): ακολουθία 7 bytes με εναλλασσόμενες μονάδες και μηδενικά που χρησιμοποιούνται από το προορισμό για συγχρονισμό και 1 byte με συγκεκριμένη ακολουθία ψηφίων (2 τελευταία ψηφία μονάδες) που δηλώνει ότι ακολουθεί η έναρξη του πλαισίου, έτσι ώστε να το αντιληφθεί ο προορισμός.
- Διεύθυνση προορισμού (destination address): καθορίζει το σταθμό (μοναδική διεύθυνση) ή τους σταθμούς (διεύθυνση ομάδας σταθμών) που κατευθύνεται το πλαίσιο.
- Διεύθυνση πηγής (source address): καθορίζει τον σταθμό που μεταδίδει το πλαίσιο.
- Τύπος: τύπος δεδομένων ανώτερου επιπέδου (π.χ. 0800_{16} για πακέτο IP version 4).
- Δεδομένα ανώτερου επιπέδου: μέγιστο «ωφέλιμο» φορτίο πλαισίου 1500 bytes.
- Συμπλήρωμα (pad): bytes που προστίθενται, έτσι ώστε το πλαίσιο να θεωρείται έγκυρο πλαίσιο Ethernet (χρησιμοποιείται όταν τα δεδομένα είναι λιγότερα από 46 bytes).
- Ακολουθία ελέγχου πλαισίου (FCS): κυκλικός έλεγχος πλεονασμού (CRC) 32 bits (4 bytes) που εφαρμόζεται σε όλα τα πεδία του πλαισίου εκτός από το προοίμιο.

Ethernet με μεταγωγή



Στο διανομέα (hub), οι σταθμοί αντιμετωπίζουν πιθανές συγκρούσεις και χρησιμοποιούν τον αλγόριθμο CSMA/CD για να διενεργήσουν τις μεταδόσεις τους.

Στο μεταγωγό (switch), δε συμβαίνουν συγκρούσεις μεταξύ των θυρών (ports).

- Στο Ethernet με μεταγωγή, συνδέοντας ή αποσυνδέοντας ένα καλώδιο σε μια θύρα μεταγωγού μπορούμε εύκολα να προσθαφαιρέσουμε ένα σταθμό στο τοπικό δίκτυο.
- Όταν μια θύρα μεταγωγού λάβει ένα πλαίσιο Ethernet από ένα σταθμό, ο μεταγωγός ελέγχει τις διευθύνσεις του πλαισίου για να διαπιστώσει ποια είναι η θύρα για την οποία το πλαίσιο προορίζεται.
- Ο μεταγωγός διαβιβάζει το πλαίσιο στη θύρα προορισμού, μέσα από το υψηλής ταχύτητας κύκλωμά του.
- Η θύρα προορισμού μεταδίδει το πλαίσιο, ώστε αυτό να φτάσει στο σταθμό προορισμού, ενώ οι υπόλοιπες θύρες δε γνωρίζουν την ύπαρξη του πλαισίου αυτού.
- Στην περίπτωση ταυτόχρονης αποστολής πλαισίου από σταθμό και θύρα (με καλώδιο μονής κατεύθυνσης), ο σταθμός και η θύρα πρέπει να διεκδικήσουν ποιος θα μεταδώσει χρησιμοποιώντας την τεχνική CSMA/CD.

Βασικές έννοιες διαδικτύωσης

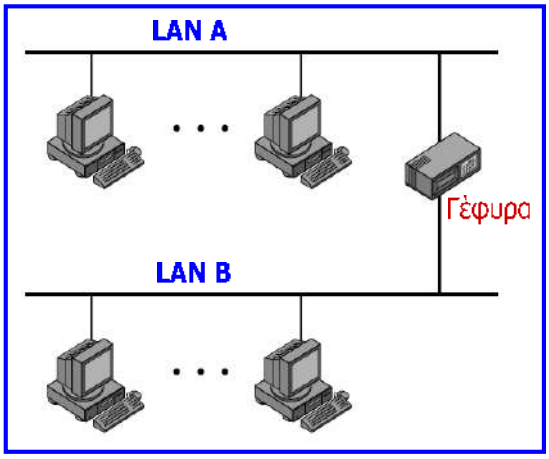
- Διαδικτύωση: διαδικασία σύνδεσης δύο ή περισσότερων δικτύων, ώστε να παρέχεται η δυνατότητα επικοινωνίας σε τερματικά συστήματα που ανήκουν σε διαφορετικά δίκτυα, τα οποία μπορεί να έχουν διαφορετική αρχιτεκτονική, να λειτουργούν με διαφορετικά πρωτόκολλα και να έχουν σχεδιαστεί για κάλυψη διαφορετικών αναγκών.
- Ενδιάμεσο σύστημα (intermediate system): χρησιμοποιείται για τη σύνδεση δύο δικτύων και επιτρέπει την επικοινωνία τερματικών συστημάτων που είναι συνδεδεμένα στο ένα δίκτυο με τερματικά συστήματα του άλλου δικτύου.
- Βασικά ενδιάμεσα συστήματα είναι οι δρομολογητές (routers), οι γέφυρες (bridges) και οι μεταγωγείς (switches).
- Δρομολογητής: ενδιάμεσο σύστημα που συνδέει δύο δίκτυα (π.χ. τοπικά δίκτυα, LANs) ανεξάρτητα από το αν είναι όμοια ή όχι, χρησιμοποιώντας το ίδιο πρωτόκολλο διαδικτύου (επίπεδο 3 ή δικτύου στο μοντέλο OSI, επίπεδο IP στο μοντέλο TCP/IP) με τα τερματικά συστήματα των δικτύων που συνδέει.
- Ο δρομολογητής εκτελεί λειτουργίες αναμετάδοσης και δρομολόγησης πακέτων δεδομένων (χρησιμοποιώντας κατάλληλο αλγόριθμο), έτσι ώστε να είναι δυνατή η ανταλλαγή δεδομένων μεταξύ τερματικών συστημάτων που είναι συνδεδεμένα σε διαφορετικά δίκτυα.

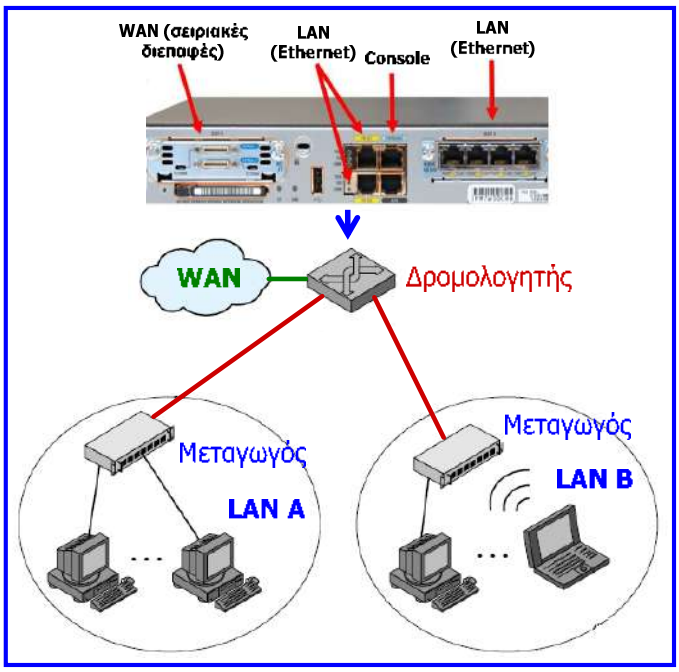
Βασικές έννοιες διαδικτύωσης

- Γέφυρα: ενδιάμεσο σύστημα που συνδέει δύο τοπικά δίκτυα (LANs) που χρησιμοποιούν όμοια ή παρόμοια πρωτόκολλα τοπικής δικτύωσης, συλλέγει και προωθεί τα πλαίσια από το ένα δίκτυο στο άλλο, χωρίς να τα τροποποιεί ή να κάνει οποιαδήποτε προσθήκη.
- Μεταγωγός (switch): συνδέει τερματικά συστήματα και τοπικά δίκτυα. Δέχεται ένα πλαίσιο σε μια θύρα εισόδου του από ένα τερματικό σύστημα, το αποθηκεύει για μικρό χρονικό διάστημα και στη συνέχεια το προωθεί στην κατάλληλη θύρα εξόδου του, ώστε να παραδοθεί στον σύστημα που απευθύνεται (store and forward switch).
- Εναλλακτικά, λόγω του ότι η διεύθυνση προορισμού είναι διαθέσιμη στην αρχή του πλαισίου, ο μεταγωγός ξεκινά τη προώθηση του εισερχόμενου πλαισίου στην κατάλληλη θύρα εξόδου, ώστε να παραδοθεί στο σύστημα που απευθύνεται, όταν έχει γίνει η αναγνώριση της διεύθυνσης προορισμού (cut-through switch).
- Στην αρχιτεκτονική των πρωτοκόλλων ενός τοπικού δικτύου, οι διευθύνσεις των σταθμών προσδιορίζονται στο επίπεδο ελέγχου πρόσβασης μέσου (MAC), δηλαδή στο επίπεδο 2 και γι' αυτό οι γέφυρες και συνήθως οι μεταγωγείς λειτουργούν σε αυτό το επίπεδο.
- Ενώ οι γέφυρες μπορούν να αναλύσουν και να προωθήσουν ένα πλαίσιο κάθε φορά, οι μεταγωγείς διαθέτουν πολλαπλές παράλληλες διαδρομές για πλαίσια δεδομένων, παρέχοντας πιο αποδοτική σύνδεση.

Βασικές έννοιες διαδικτύωσης

Οι δρομολογητές διαθέτουν διάφορες διεπαφές (physical interfaces), όπως διεπαφές για σύνδεση τοπικών δικτύων τύπου Ethernet (Ethernet, Fast Ethernet, Gigabit Ethernet), για σύνδεση τοπικών δικτύων οπτικής ίνας (FDDI, fiber distribution data interface) και σειριακές διεπαφές (serial interfaces), οι οποίες συνήθως χρησιμοποιούνται για σύνδεση δικτύων ευρείας περιοχής (wide area networks, WANs).

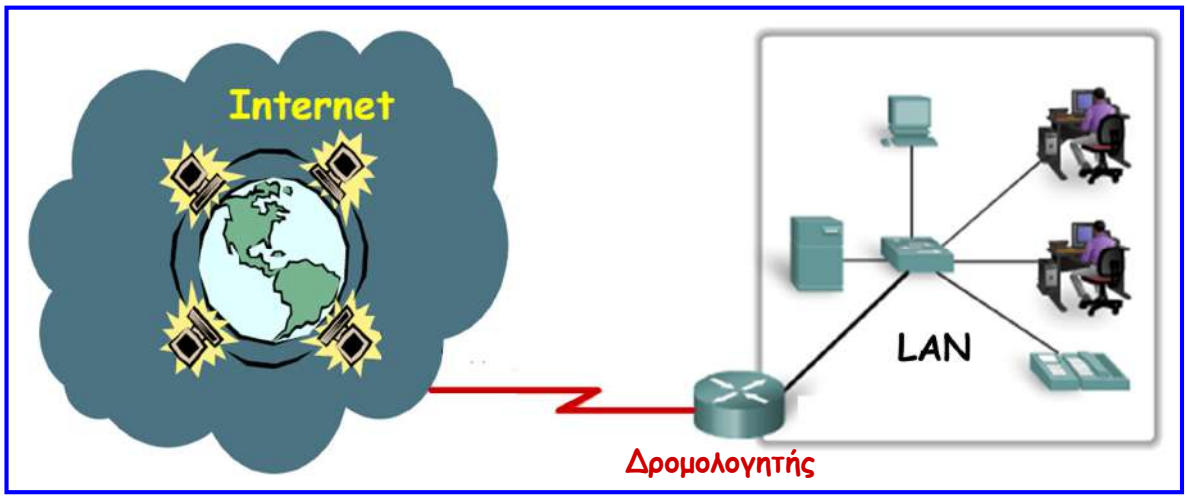




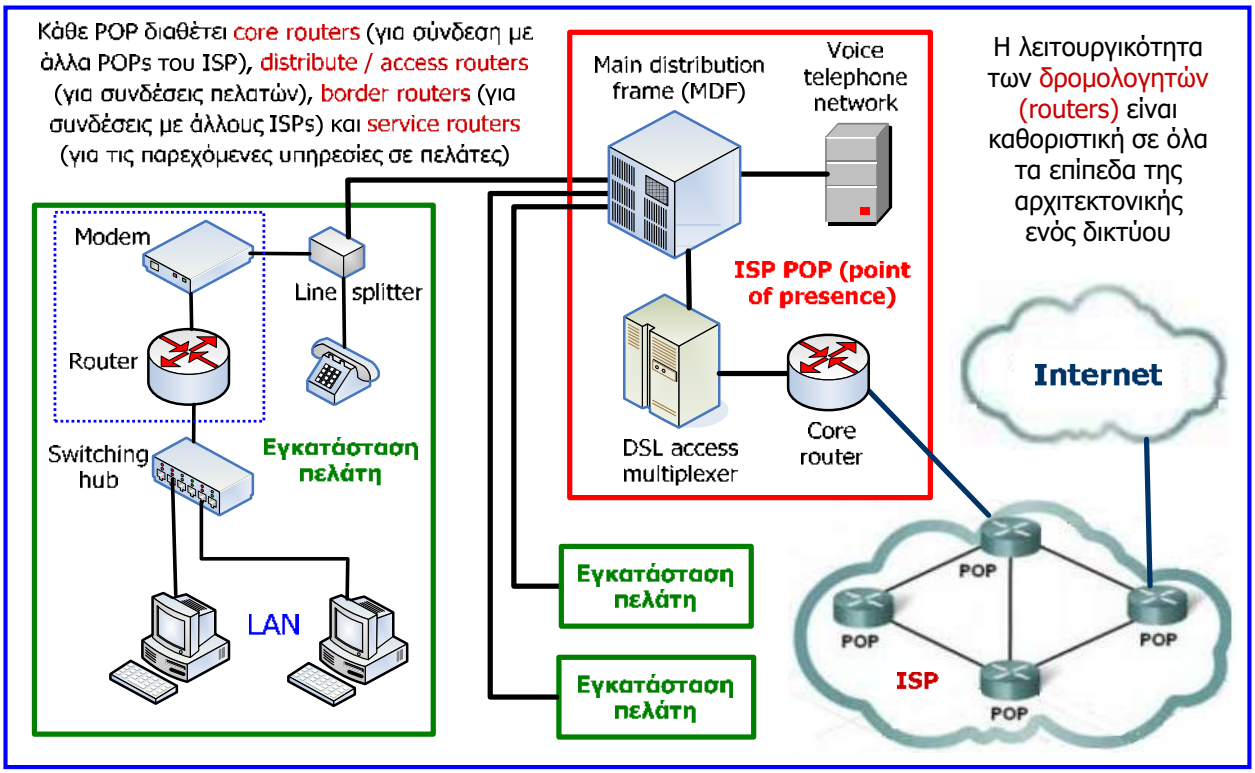
Βασικές έννοιες διαδικτύωσης

- Το διαδίκτυο (Internet) αποτελείται από πολλαπλά ξεχωριστά δίκτυα, τα οποία διασυνδέονται μεταξύ τους με ενδιάμεσα συστήματα.
- Καθένα από τα ξεχωριστά δίκτυα (συστατικό δίκτυο) υποστηρίζει την επικοινωνία μεταξύ των τερματικών συστημάτων που είναι συνδεδεμένα σε αυτό.
- Ενδοδίκτυο (intranet): διαδίκτυο που χρησιμοποιείται από έναν οργανισμό (δίκτυο με κοινή δικτυακή διαχείριση, common administrative domain), παρέχει βασικές εφαρμογές διαδικτύου (π.χ. παγκόσμιο ιστό) και λειτουργεί εσωτερικά για τους σκοπούς του οργανισμού. Μπορεί να είναι απομονωμένο ή να συνδέεται στο διαδίκτυο.
- Υποδίκτυο είναι ένα δίκτυο που ανήκει στο διαδίκτυο. Ο χρήστης «βλέπει» το διαδίκτυο ως ένα μόνο δίκτυο, ωστόσο είναι συνδεδεμένος και χρησιμοποιεί ένα μόνο υποδίκτυο.
- Τα δεδομένα αποστέλλονται ως αυτόνομα πακέτα (packets) από ένα σύστημα πηγής σε ένα σύστημα προορισμού (τερματικά συστήματα), κατά μήκος μιας διαδρομής του διαδικτύου.
- Τα αυτόνομα πακέτα ενθυλακώνονται σε πλαίσια σύμφωνα με το πρωτόκολλο του κατώτερου επιπέδου, διαμορφώνονται κατάλληλα και τελικά μεταδίδονται ως ψηφιακά σήματα διαμέσου του φυσικού καναλιού (μέσου).
- Το πιο συνηθισμένο πρωτόκολλο διαδικτύωσης είναι το πρωτόκολλο διαδικτύου (IP), που επισυνάπτει μία επικεφαλίδα με πληροφορίες ελέγχου στα τμήματα δεδομένων ανωτέρου επιπέδου (TCP), σχηματίζοντας ένα αυτόνομο πακέτο IP (IP packet).

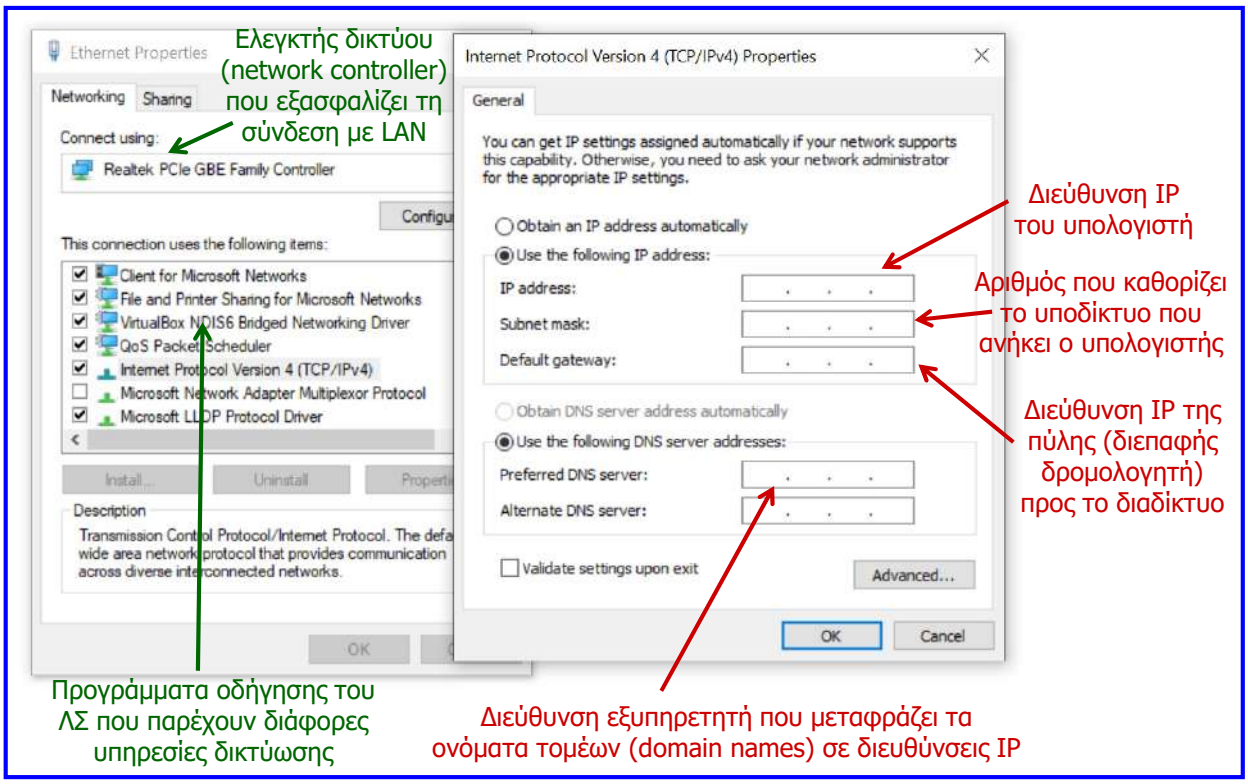
Σύνδεση στο διαδίκτυο μέσω LAN



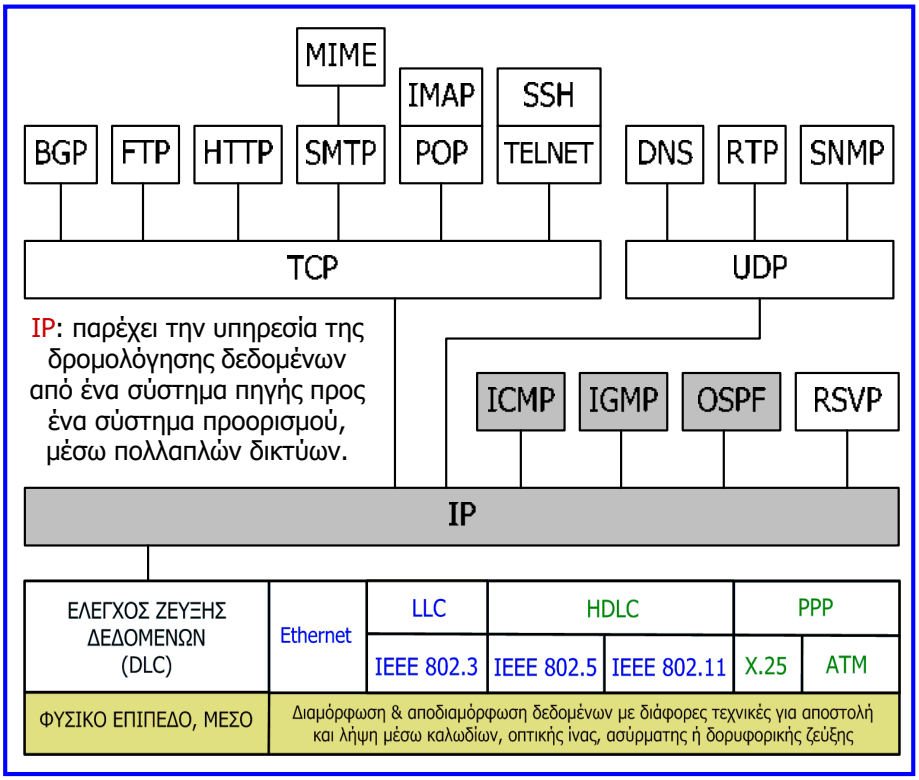
Σύνδεση στο διαδίκτυο μέσω DSL



Ρύθμιση παραμέτρων επικοινωνίας του χρήστη



Επίπεδα και πρωτόκολλα διαδικτύωσης



Πρωτόκολλο μηνυμάτων ελέγχου διαδικτύου (ICMP): παρέχει υπηρεσία μεταφοράς μηνυμάτων ελέγχου μεταξύ δρομολογητών και τερματικών συστημάτων.

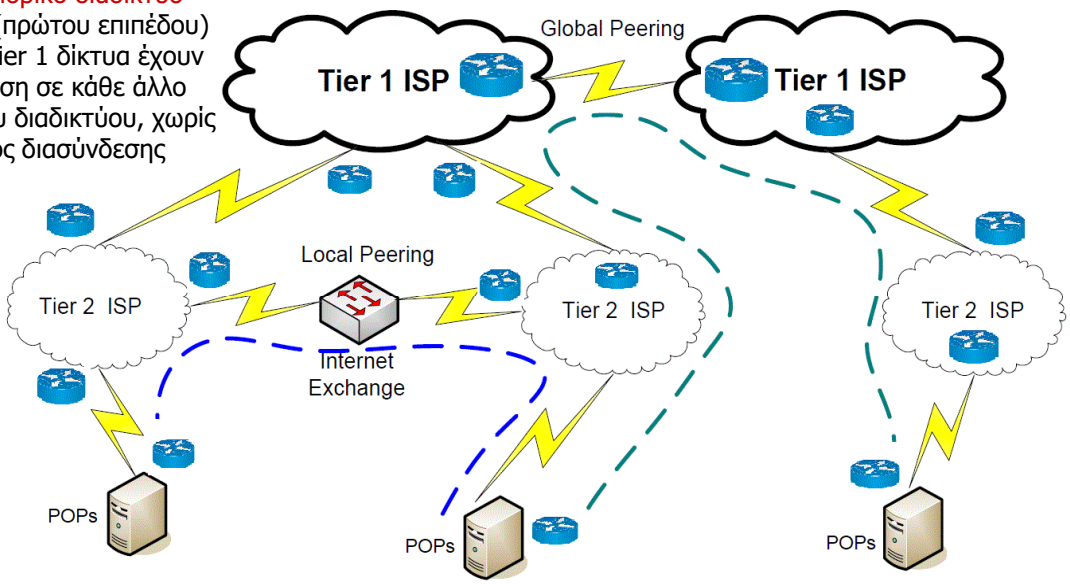
Πρωτόκολλο διαχείρισης ομάδων διαδικτύου (IGMP): υποστηρίζει την πολλαπλή αποστολή.

Τα ICMP και IGMP στην ουσία ανήκουν στο ίδιο επίπεδο με το IP, αλλά λειτουργούν ως χρήστες του, αφού τα μηνύματά τους μεταφέρονται ως πακέτα IP.

OSPF: πρωτόκολλο δρομολόγησης πακέτων IP, που χρησιμοποιεί το IP για την επιλογή μονοπατιών με το μικρότερο κόστος μεταφοράς.

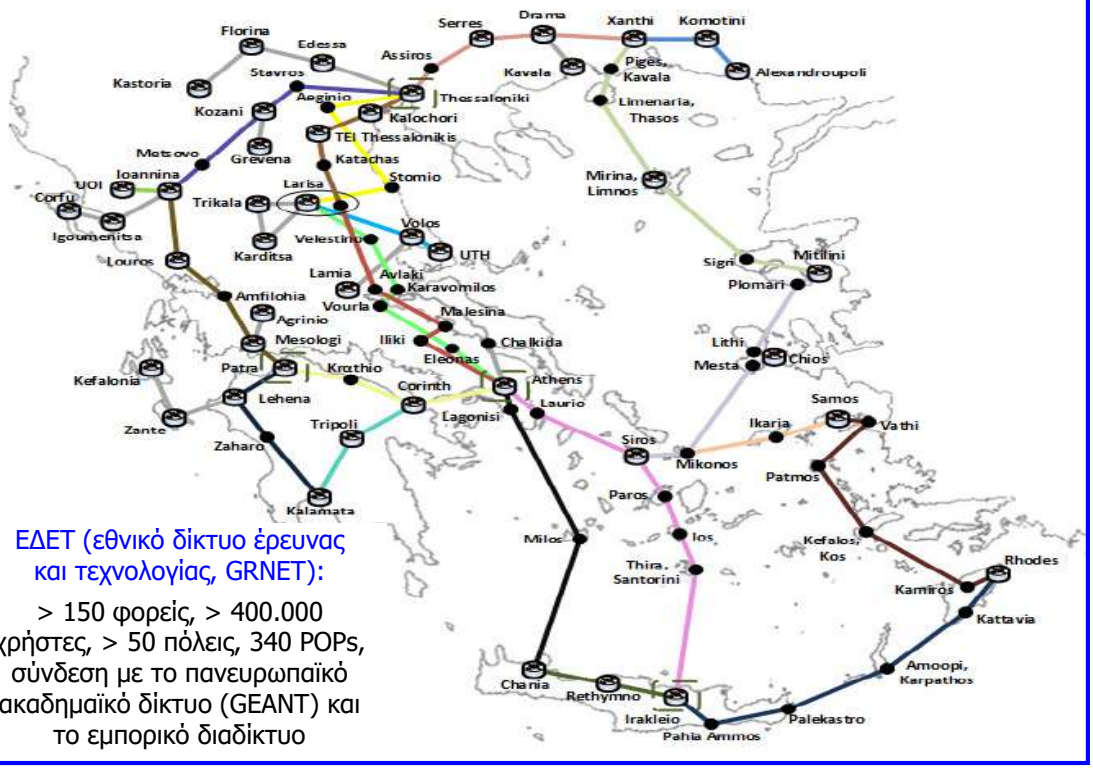
Αρχιτεκτονική του διαδικτύου

Στο εμπορικό διαδίκτυο οι Tier 1 (πρώτου επιπέδου) ISPs ή Trier 1 δίκτυα έχουν πρόσβαση σε κάθε άλλο δίκτυο του διαδικτύου, χωρίς κόστος διασύνδεσης



5,1 δις χρήστες του διαδικτύου (~ 64,5% πληθυσμού), 56% των χρηστών στην Ασία, 14,5% στην Ευρώπη, 0,16% των χρηστών στην Ελλάδα, > 1.100.000 γνωστά δίκτυα, > 105.000 αυτόνομα συστήματα, 16 Tier 1 ISPs με πρόσβαση στο σύνολο των δικτύων, 2 regional Tier 1 ISPs στην Ελλάδα και οι λοιποί Tier 2.

Το Ελληνικό δημόσιο διαδίκτυο



ΕΔΕΤ (εθνικό δίκτυο έρευνας και τεχνολογίας, GRNET):

> 150 φορείς, > 400.000 χρήστες, > 50 πόλεις, 340 POPs, σύνδεση με το πανευρωπαϊκό ακαδημαϊκό δίκτυο (GEANT) και το εμπορικό διαδίκτυο

Προϋποθέσεις διαδικτύωσης

- Η διαδικτύωση προϋποθέτει τουλάχιστον μία ζεύξη μεταξύ των δικτύων, δηλαδή μία φυσική σύνδεση και μία σύνδεση ελέγχου ζεύξης δεδομένων.
- Εξασφάλιση δρομολόγησης και παράδοσης δεδομένων μεταξύ εφαρμογών σε τερματικά συστήματα διαφορετικών δικτύων.
- Παροχή υπηρεσίας παρακολούθησης της χρήσης των δικτύων και των δρομολογητών και διατήρηση πληροφοριών κατάστασης.
- Διευθέτηση των διαφορών μεταξύ των δικτύων που συμμετέχουν στη διαδικτύωση ώστε κατά την εγκατάσταση και λειτουργία της να μην απαιτούνται τροποποιήσεις στην αρχιτεκτονική και λειτουργία των δικτύων.
- Πιθανές διαφορές των δικτύων αφορούν: σχήμα διευθυνσιοδότησης, μέγιστο μέγεθος πακέτου (μπορεί να απαιτείται κερματισμός), μηχανισμός πρόσβασης δικτύου (π.χ. Ethernet, ATM), χρόνος αναμονής έως την αναμετάδοση πακέτων (απαιτείται διαχείριση χρόνου αναμονής), ανάκτηση σφαλμάτων (η διαδικτύωση δεν παρεμβαίνει στους μηχανισμούς ανάκτησης σφαλμάτων), τεχνικές ανίχνευσης και ελέγχου συμφόρησης, έλεγχος πρόσβασης (πιστοποίηση) χρηστών, λειτουργία με ή χωρίς σύνδεση (νοητά κυκλώματα ή αυτόνομα πακέτα).
- Κάποιες από τις παραπάνω προϋποθέσεις καλύπτονται από το IP και μερικές απαιτούν συμπληρωματικές λειτουργίες που παρέχονται από άλλα επίπεδα.

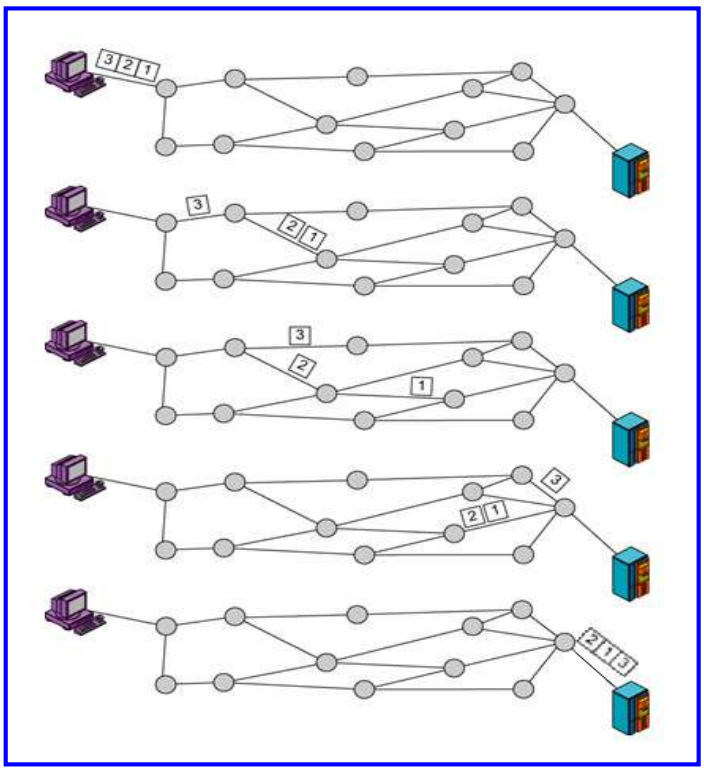
Λειτουργία διαδικτύωσης με σύνδεση

- Βασικό χαρακτηριστικό της διαδικτύωσης είναι εάν η λειτουργία της βασίζεται σε προσέγγιση με σύνδεση (connection-oriented) ή χωρίς σύνδεση (connectionless).
- Στη λειτουργία με σύνδεση κάθε δίκτυο παρέχει υπηρεσία με σύνδεση, δηλαδή μπορεί να αποκατασταθεί μία λογική σύνδεση (νοητό ή εικονικό κύκλωμα, virtual circuit) μεταξύ δύο οποιωνδήποτε τερματικών συστημάτων του ίδιου δικτύου.
- Για τη σύνδεση δύο ή περισσότερων δικτύων χρησιμοποιούνται ενδιάμεσα συστήματα, καθένα από τα οποία εμφανίζεται ως τερματικό σύστημα στο δίκτυο που συνδέεται.
- Η λογική σύνδεση μεταξύ δύο τερματικών συστημάτων διαφορετικών δικτύων αποτελείται από μία ακολουθία λογικών συνδέσεων κατά μήκος των δικτύων, οι οποίες συνδέονται μέσω των ενδιάμεσων συστημάτων.
- Η προσέγγιση αυτή απαιτεί εμπλουτισμό της υπηρεσίας κάποιων τοπικών δικτύων τα οποία χρησιμοποιούν μοντέλο αυτόνομου πακέτου για μετάδοση.
- Ένα ενδιάμεσο σύστημα σε λειτουργία με σύνδεση εκτελεί: αναμετάδοση (relaying) μονάδων δεδομένων από το ένα δίκτυο στο άλλο και δρομολόγηση (routing), δηλαδή λήψη απόφασης σχετικά με το επόμενο άλμα στην ακολουθία λογικών συνδέσεων.

Λειτουργία διαδικτύωσης χωρίς σύνδεση

- Η λειτουργία χωρίς σύνδεση αντιστοιχεί στο μηχανισμό αυτόνομων πακέτων (datagrams).
- Κάθε αυτόνομο πακέτο αντιμετωπίζεται ανεξάρτητα και διενεργείται η δρομολόγησή του από το τερματικό σύστημα πηγής στο τερματικό σύστημα προορισμού διαμέσου μιας σειράς δρομολογητών και δικτύων.
- Σε κάθε δρομολογητή λαμβάνεται μία απόφαση δρομολόγησης, για κάθε αυτόνομο πακέτο, που αφορά το επόμενο άλμα, επομένως οι μονάδες δεδομένων μπορούν να κινούνται σε διαφορετικές διαδρομές μεταξύ συστημάτων πηγής και προορισμού.
- Τα τερματικά συστήματα και οι δρομολογητές μοιράζονται ένα κοινό πρωτόκολλο επιπέδου δικτύου, που αναφέρεται ως πρωτόκολλο διαδικτύου (internet protocol, IP).
- Κάτω από το πρωτόκολλο (επίπεδο) διαδικτύου, απαιτείται άλλο πρωτόκολλο (επίπεδο) για πρόσβαση σε συγκεκριμένο δίκτυο.
- Η προσέγγιση είναι πιο ευέλικτη και ανθεκτική από τη λειτουργία με σύνδεση, αφού υποστηρίζει δίκτυα χωρίς σύνδεση και μπορεί να αντιμετωπίζει τοπικές αποτυχίες ή καταστάσεις συμφόρησης παραδίδοντας πακέτα μέσω εναλλακτικών διαδρομών.
- Επίσης, δεν υφίσταται η πρόσθετη επιβάρυνση αποκατάστασης λογικής σύνδεσης.
- Είναι όμως μη αξιόπιστη, αφού δεν εγγυάται την παράδοση όλων των πακέτων ή την παράδοση με την κατάλληλη σειρά (λόγω διαφορετικών διαδρομών). Η αξιοπιστία είναι ευθύνη του ανώτερου επιπέδου (TCP, transmission control protocol).

Λειτουργία διαδικτύωσης χωρίς σύνδεση



Πρωτόκολλο διαδικτύου (IP)

- Το IP, αποτελεί μέρος του μοντέλου TCP/IP και είναι το ευρύτερα χρησιμοποιούμενο πρωτόκολλο διαδικτύωσης.
- Το IP προδιαγράφει:
  - ✓ τη διεπαφή με το ανώτερο επίπεδο (TCP) που περιλαμβάνει τις υπηρεσίες ή λειτουργίες που παρέχει το IP στο ανώτερο επίπεδο, και
  - ✓ την κύρια δομή του πρωτοκόλλου, μαζί με τους μηχανισμούς του που απαιτούνται για τη μεταφορά των δεδομένων διαμέσου πολλαπλών δικτύων.
- Σήμερα χρησιμοποιείται κυρίως η 4η έκδοση του (IPv4), η οποία σταδιακά αντικαθίσταται από την 6η έκδοση (IPv6).
- Οι λειτουργίες που παρέχονται από ένα πρωτόκολλο στη διεπαφή του με το ανώτερο στρώμα αναφέρονται ως πρωτογενείς λειτουργίες (primitives).
- Το IP παρέχει δύο πρωτογενείς λειτουργίες: αποστολή (send) και παράδοση (deliver).
- Με την πρώτη λειτουργία το ανώτερο επίπεδο αιτείται τη μετάδοση δεδομένων, ενώ με την δεύτερη το IP ενημερώνει το ανώτερο επίπεδο (δηλαδή το χρήστη του) για την άφιξη των δεδομένων.
- Κάθε πρωτογενής λειτουργία σχετίζεται με παραμέτρους που αφορούν τα δεδομένα, καθώς και πληροφορίες ελέγχου (π.χ. διευθύνσεις συστημάτων πηγής και προορισμού).

Πρωτόκολλο διαδικτύου IPv4: επικεφαλίδα πακέτου

Η προσθήκη πληροφοριών ελέγχου στα δεδομένα αναφέρεται ως ενθυλάκωση (encapsulation). Οι πληροφορίες ελέγχου προστίθενται στην επικεφαλίδα ενός πακέτου, αλλά υπάρχουν και περιπτώσεις πρωτοκόλλων που προσθέτουν πληροφορία (όπως τον κώδικα ανίχνευσης σφαλμάτων) στην ουρά ενός πακέτου.

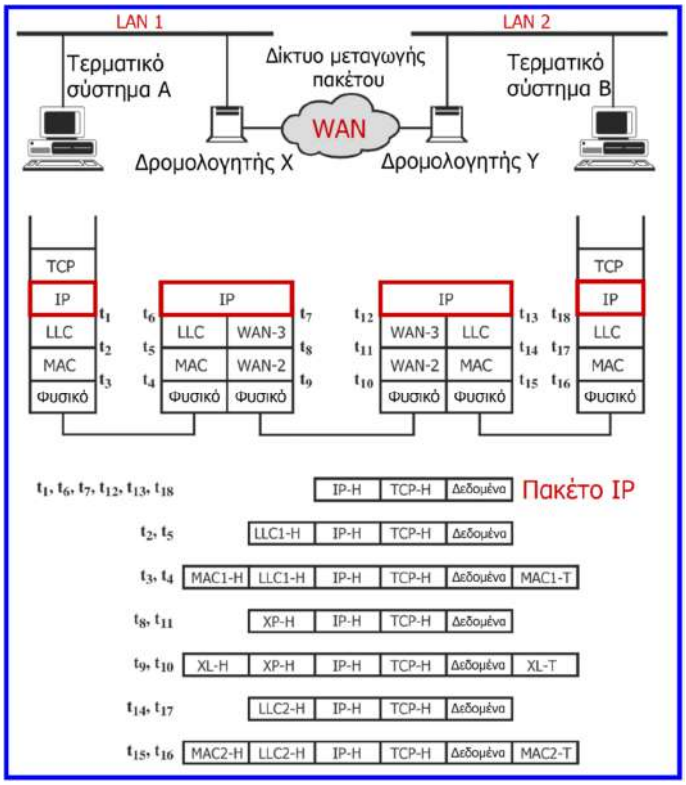
Πρωτόκολλο διαδικτύου IPv4: επικεφαλίδα πακέτου

- Έκδοση (4 bits): για την υποστήριξη της εξέλιξης του πρωτοκόλλου (0100 για το IPv4).
- Μήκος επικεφαλίδας (4 bits): σε λέξεις των 32 bits, με ελάχιστη τιμή 5 (δηλαδή 20 bytes).
- Διαφοροποιημένες υπηρεσίες (DS, 6 bits) που υποδεικνύουν υπηρεσίες (ειδική μεταχείριση πακέτων) όπως προτεραιότητα, χαμηλή καθυστέρηση κ.ά. Ρητή ειδοποίηση συμφόρησης (ECN, 2 bits). Τα 8 ψηφία έχουν τιμή 0 όταν δεν ορίζονται DS και δεν υποστηρίζεται ECN.
- Συνολικό μήκος (16 bits): συνολικό μήκος του πακέτου σε bytes.
- Ταυτότητα (16 bits): μοναδικός αριθμός για τις διευθύνσεις πηγής και προορισμού και το πρωτόκολλο αποδέκτη, για το χρόνο κατά τον οποίο το πακέτο παραμένει στο διαδίκτυο.
- Σημαίες (3 bits): 1ο bit (48ο) = 0 (δε χρησιμοποιείται), 2ο bit (49ο) = Μη κερματισμού, DF (όταν είναι 1 απαγορεύει τον κερματισμό δεδομένων), 3ο bit (50ο) = Επιπλέον, More (χρήση κατά την επανασυναρμολόγηση των επιμέρους πακέτων που προέκυψαν από κερματισμό).
- Θέση επιμέρους πακέτου (13 bits): δηλώνει τη θέση του επιμέρους πακέτου δεδομένων (που προέκυψε από κερματισμό) στο αρχικό πακέτο IP (σε πολλαπλάσια των 64 bits).
- Διάρκεια ζωής (TTL, time-to-live, 8 bits): δηλώνει το χρόνο παραμονής ενός πακέτου στο διαδίκτυο. Η μέτρηση του υλοποιείται μέσω ενός μετρητή αλμάτων, ο οποίος μειώνεται κατά 1 κάθε φορά που ένα πακέτο διέρχεται από έναν δρομολογητή.
- Πρωτόκολλο (8 bits): ανώτερο επίπεδο που θα μεταφέρει τα δεδομένα στο τερματικό σύστημα προορισμού (λαμβάνει τιμή 06_{16} για το TCP, 11_{16} για το UDP, 01_{16} για το ICMP).

Πρωτόκολλο διαδικτύου IPv4: επικεφαλίδα πακέτου

- Άθροισμα ελέγχου επικεφαλίδας (16 bits): κώδικας ανίχνευσης σφαλμάτων μόνο για την επικεφαλίδα. Επειδή κάποια πεδία (π.χ. TTL) αλλάζουν στη διάρκεια της μεταφοράς, το άθροισμα επαληθεύεται και ξαναϋπολογίζεται σε κάθε δρομολογητή / παραλήπτη.
- Το πεδίο αποτελεί το συμπλήρωμα ως προς 1 του αθροίσματος όλων των λέξεων με 16 bits της επικεφαλίδας. Κατά τον υπολογισμό, το πεδίο αθροίσματος ελέγχου θεωρείται 0.
- Το άθροισμα ελέγχου προσαρτάται (ως η 6η λέξη με 16 bits) στην επικεφαλίδα και εάν στον παραλήπτη το συμπλήρωμα ως προς 1 του αθροίσματος όλων των λέξεων με 16 bits της επικεφαλίδας προκύψει 0, σημαίνει ότι δεν εντοπίστηκε σφάλμα στην επικεφαλίδα.
- Διευθύνσεις πηγής και προορισμού (από 32 bits): καθορίζουν τα τερματικά συστήματα και το δίκτυο στο οποίο αυτά ανήκουν.
- Επιλογές (μεταβλητού μήκους έως 40 bytes): περιπτώσεις χρήσης του είναι η καταγραφή διαδρομής πακέτου και η δρομολόγηση πηγής στην οποία οι επόμενοι κόμβοι καθορίζονται από το σύστημα πηγής. Στο πεδίο μπορεί να γίνει προσθήκη έως και 9 διευθύνσεων IP.
- Συμπλήρωμα δεδομένων (μεταβλητού μήκους): εξασφαλίζει ότι το μήκος της επικεφαλίδας είναι πολλαπλάσιο των 32 bits.
- Μετά την επικεφαλίδα (20 bytes, αφού τα τελευταία 2 πεδία χρησιμοποιούνται σπάνια), ακολουθούν τα δεδομένα του ανώτερου επιπέδου με μήκος ακέραιο πολλαπλάσιο του 1 byte. Το μέγιστο μήκος ενός πακέτου IP (επικεφαλίδα + δεδομένα) είναι 65535 bytes.

Λειτουργία του πρωτοκόλλου IP



TCP-H: επικεφαλίδα τμήματος TCP

IP-H: επικεφαλίδα πακέτου IP που καθορίζει την παγκόσμια διεύθυνση που περιλαμβάνει ταυτότητα ή αναγνωριστή δικτύου και τερματικού συστήματος

LLCi-H: επικεφαλίδα πλαισίου LLC (ελέγχου λογικής σύνδεσης)

MACi-H: επικεφαλίδα πλαισίου MAC (ελέγχου πρόσβασης μέσου)

MACi-T: ουρά πλαισίου MAC

XP-H: επικεφαλίδα πλαισίου WAN-3

XL-H: επικεφαλίδα πλαισίου WAN-2

XL-T: ουρά πλαισίου WAN-2

- Στους δρομολογητές οι μονάδες δεδομένων μπορεί να κερματιστούν για προσαρμογή τους σε εξερχόμενο δίκτυο.
- Μπορούν να παρεμβληθούν όσοι δρομολογητές χρειάζονται για να παραδοθεί μια μονάδα δεδομένων στο τερματικό σύστημα προορισμού.

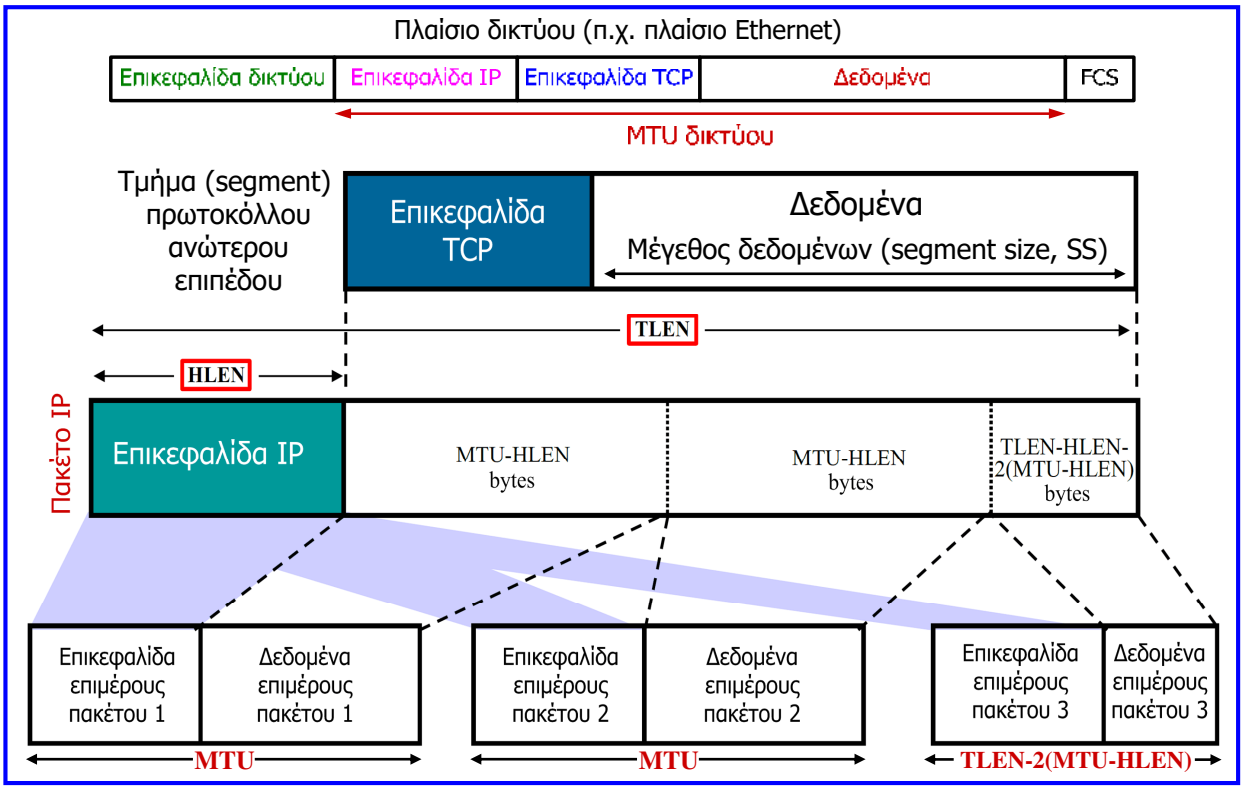
Διάρκεια ζωής αυτόνομου πακέτου

- Εάν χρησιμοποιείται δυναμική δρομολόγηση πακέτων, υπάρχει το ενδεχόμενο κάποιο πακέτο να περιφέρεται συνεχώς στο δίκτυο, επειδή το τερματικό σύστημα προορισμού δε μπορεί να βρεθεί, καταναλώνοντας πόρους και επηρεάζοντας την ορθή λειτουργία του ανώτερου επιπέδου TCP.
- Έτσι, στην επικεφαλίδα κάθε πακέτου καταγράφεται ένα πεδίο με συγκεκριμένη διάρκεια ζωής (TTL: time to live), με τη λήξη της οποίας το πακέτο απορρίπτεται.
- Πρακτικά, η μέτρηση διάρκειας ζωής (το μέγιστο διάστημα για το οποίο το πακέτο επιτρέπεται να βρίσκεται στο δίκτυο) υλοποιείται μέσω ενός μετρητή αλμάτων, ο οποίος μειώνεται κάθε φορά που ένα πακέτο διέρχεται από ένα δρομολογητή. Για το λόγο αυτό στη νέα έκδοση IPv6, το πεδίο αυτό αναφέρεται ως hop limit (όριο αλμάτων).
- Το πεδίο TTL της επικεφαλίδας ενός πακέτου IP, ορίζει λοιπόν το μέγιστο αριθμό αλμάτων μέχρι τον οποίο το πακέτο επιτρέπεται να παραμένει στο διαδίκτυο.
- Η τιμή του καθορίζεται από το τερματικό σύστημα πηγής, κάθε δρομολογητής ελαττώνει την τιμή αυτή κατά ένα, έτσι ώστε να αποτελεί το επιτρεπτό πλήθος αλμάτων που απομένουν στο πακέτο και εάν ο μετρητής λάβει τιμή 0, το πακέτο απορρίπτεται.
- Θεωρητικά, το πεδίο TTL περιέχει το χρονικό διάστημα σε sec μέσα στο οποίο θα πρέπει το πακέτο να έχει παραδοθεί στο τερματικό σύστημα προορισμού. Ωστόσο, αυτό θα απαιτούσε έναν παγκόσμιο μηχανισμό χρονομέτρησης.

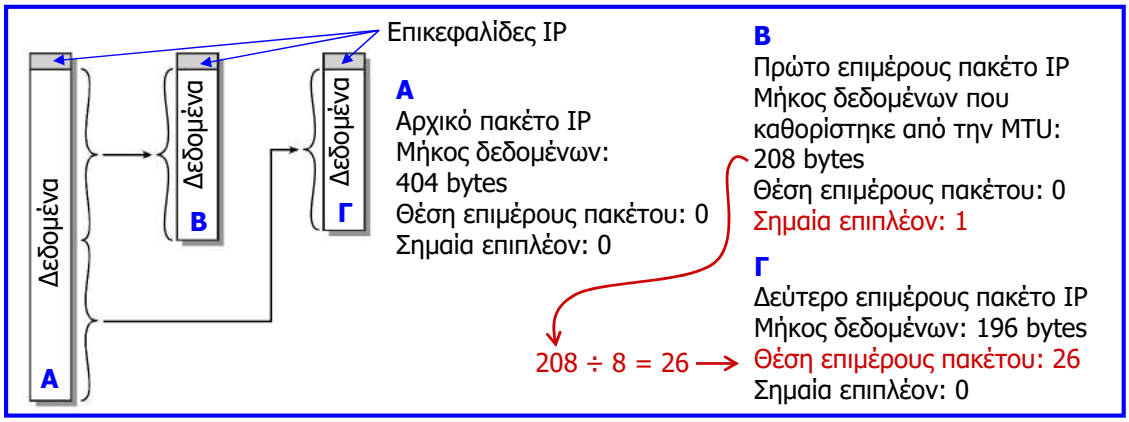
Κερματισμός και επανασυναρμολόγηση

- Ο κερματισμός (fragmentation) στο πρωτόκολλο διαδικτύου (IP) επιτρέπει τη διαίρεση μεγάλων πακέτων IP σε μικρότερα επιμέρους πακέτα (fragments), ώστε αυτά να μπορούν να μεταδοθούν σε ένα δίκτυο με μέγιστη μονάδα μετάδοσης (maximum transmission unit, MTU) μικρότερου μεγέθους από τα αρχικά πακέτα IP.
- Για τον κερματισμό χρησιμοποιούνται οι ακόλουθες πληροφορίες:
  - ✓ μέγιστη μονάδα μετάδοσης (MTU), δηλαδή το μέγιστο μέγεθος πακέτου που μπορεί να μεταδοθεί από το δίκτυο,
  - ✓ μήκος των δεδομένων και της επικεφαλίδας του αρχικού πακέτου σε bytes,
  - ✓ θέση (offset) κάθε επιμέρους πακέτου δεδομένων στο αρχικό πακέτο (σε πολλαπλάσια των 64 bits ή των 8 bytes), με την τιμή 0 να αφορά το πρώτο επιμέρους πακέτο.
- Σε κάθε επιμέρους πακέτο διενεργείται η προσθήκη αντίστοιχης επικεφαλίδας (ίδιου μήκους) με εκείνη του αρχικού πακέτου.
- Στη σημαία επιπλέον των επιμέρους πακέτων τίθεται τιμή 1, με εξαίρεση το τελευταίο επιμέρους πακέτο, στη σημαία επιπλέον του οποίου τίθεται τιμή 0.
- Επισημαίνεται η ύπαρξη της σημαίας μη κερματισμού, η οποία όταν είναι ενεργοποιημένη, δηλαδή όταν έχει τιμή 1, απαγορεύει τον κερματισμό των δεδομένων.
- Τα πακέτα IP που κερματίζονται, επανασυναρμολογούνται μόνο στο τερματικό σύστημα προορισμού (όχι στους ενδιάμεσους δρομολογητές).

Κερματισμός και επανασυναρμολόγηση



Κερματισμός και επανασυναρμολόγηση



- Για την επανασυναρμολόγηση ή ανασύνθεση (reassembly) χρησιμοποιούνται πληροφορίες της επικεφαλίδας IP όπως:
  - ✓ ταυτότητα πακέτου, δηλαδή ο μοναδικός αριθμός για τη διεύθυνση συστήματος πηγής, τη διεύθυνση συστήματος προορισμού και το πρωτόκολλο αποδέκτη (π.χ. TCP),
  - ✓ μήκος δεδομένων: μήκος του πεδίου δεδομένων σε bytes,
  - ✓ θέση (offset): θέση επιμέρους πακέτου δεδομένων στο αρχικό πακέτο (σε πολλαπλάσια των 64 bits ή των 8 bytes),
  - ✓ σημαία επιπλέον (more flag): όταν είναι 1, το επιμέρους πακέτο δεν είναι το τελευταίο.

Κερματισμός και επανασυναρμολόγηση

- Το τερματικό σύστημα προορισμού διαπιστώνει τη λήψη των επιμέρους πακέτων (που αποθηκεύει σε ενδιάμεση μνήμη) χρησιμοποιώντας την ταυτότητα κάθε επιμέρους πακέτου.
- Επανασυναρμολογεί το αρχικό πακέτο συνθέτοντας τα επιμέρους πακέτα ίδιας ταυτότητας με χρήση της θέσης και της σημαίας επιπλέον κάθε επιμέρους πακέτου.
- Όταν έχει λάβει το επιμέρους πακέτο, του οποίου η σημαία επιπλέον έχει τιμή 0, μπορεί να υπολογίσει το μήκος δεδομένων του αρχικού πακέτου, πολλαπλασιάζοντας με 8 τη θέση του τελευταίου πακέτου και προσθέτοντας στο αποτέλεσμα το μήκος δεδομένων αυτού.
- Η επανασυναρμολόγηση είναι ανεπιτυχής, εάν κάποιο ή κάποια επιμέρους πακέτα χαθούν.
- Απαιτούνται μέθοδοι ανίχνευσης αποτυχίας της επανασυναρμολόγησης:

1. Χρήση διάρκειας ζωής κάθε επιμέρους πακέτου (TTL), η οποία εάν εκπνεύσει πριν την ολοκλήρωση της επανασυναρμολόγησης, τα ληφθέντα επιμέρους πακέτα απορρίπτονται.

   Η μέθοδος περιλαμβάνεται στις αρχικές προδιαγραφές του IP, αλλά επειδή ο χρόνος TTL είναι πρακτικά πλήθος αλμάτων και όχι απόλυτος χρόνος, εάν είναι αρκετά μικρός θα απορριφθούν πολλά επιμέρους πακέτα χωρίς ουσιαστική αιτία.

2. Ανάθεση διάρκειας ζωής επανασυναρμολόγησης στο 1ο επιμέρους πακέτο (σε τοπικό ρολόι, default ≈ 30 s) που θα φτάσει στο τερματικό σύστημα προορισμού, η οποία μειώνεται όσο αποθηκεύονται τα επιμέρους πακέτα και εάν εκπνεύσει πριν την ολοκλήρωση της επανασυναρμολόγησης, τα ληφθέντα πακέτα απορρίπτονται (χρήση σε πολλά συστήματα).

Έλεγχος ροής και σφαλμάτων

- Το IP δεν εγγυάται την επιτυχή παράδοση κάθε αυτόνομου πακέτου ή ότι τα πακέτα που θα φτάσουν θα παραδοθούν με τη σωστή σειρά, δηλαδή παρέχει μη αξιόπιστη υπηρεσία.
- Ελέγχεται μόνο η ορθότητα της επικεφαλίδας των πακέτων IP, έτσι ώστε να εξασφαλίζεται η λειτουργικότητά του πρωτοκόλλου.
- Η ανάρρωση από σφάλματα είναι ευθύνη άλλων επιπέδων (π.χ. του TCP).
- Σε περίπτωση ανίχνευσης σφαλμάτων (εκπνοή TTL, μη επαλήθευση αθροίσματος ελέγχου επικεφαλίδας), το αυτόνομο πακέτο απορρίπτεται και ο δρομολογητής μπορεί να ενημερώσει με κατάλληλα μηνύματα (μέσω του πρωτοκόλλου μηνυμάτων ελέγχου διαδικτύου, ICMP) το τερματικό σύστημα πηγής.
- Το τερματικό σύστημα πηγής αξιοποιεί την πληροφορία για να ενημερώσει το ανώτερο επίπεδο (TCP).
- Ο έλεγχος ροής διαδικτύου επιτρέπει στους δρομολογητές και στους σταθμούς λήψης να περιορίσουν το ρυθμό με τον οποίο λαμβάνουν δεδομένα.
- Ο μηχανισμός ελέγχου ροής είναι περιορισμένος και περιλαμβάνει την αποστολή μηνυμάτων ελέγχου ροής σε άλλους δρομολογητές και τερματικά συστήματα, ζητώντας μειωμένη ροή δεδομένων.
- Χρήση της προσέγγισης αυτής γίνεται μέσω του ICMP (πρωτόκολλο μηνυμάτων ελέγχου διαδικτύου).

Δρομολόγηση

- Για τη λειτουργία της δρομολόγησης, οι τερματικές συσκευές και οι δρομολογητές διατηρούν πίνακα δρομολόγησης που καταγράφει για κάθε πιθανό δίκτυο προορισμού, τον επόμενο δρομολογητή, στον οποίο θα πρέπει να σταλεί ένα πακέτο IP.
- Ο πίνακας δρομολόγησης μπορεί να είναι:
  - ✓ στατικός (περιλαμβάνοντας όμως και εναλλακτικές διαδρομές στην περίπτωση που κάποιος δρομολογητής δεν είναι διαθέσιμος) ή
  - ✓ δυναμικός (δηλαδή ανανεώσιμος και πιο ευέλικτος σε συνθήκες συμφόρησης ή τμημάτων του δικτύου που δε λειτουργούν).
- Για να ληφθούν αποφάσεις δυναμικής δρομολόγησης, οι δρομολογητές ανταλλάσσουν πληροφορίες (κατάσταση διαδικτύου, ποια δίκτυα μπορούν να προσεγγισθούν και από ποιους δρομολογητές, καθυστέρηση διάδοσης των διάφορων μονοπατιών κ.ά.) με τη χρήση ειδικών πρωτοκόλλων δρομολόγησης.
- Με βάση τις πληροφορίες αυτές (πληροφορίες δρομολόγησης) και χρησιμοποιώντας έναν αλγόριθμο δρομολόγησης, κάθε δρομολογητής λαμβάνει μια απόφαση δρομολόγησης.
- Κάθε δρομολογητής μπορεί να προσαρτά τη διεύθυνσή του μέσα σε ένα πακέτο IP (στο πεδίο των επιλογών της επικεφαλίδας, στο οποίο μπορούν να καταγραφούν έως 9 διευθύνσεις), ώστε να δημιουργείται καταγραφή της διαδρομής του, η οποία μπορεί να χρησιμοποιηθεί για σκοπούς δοκιμών και έρευνας αξιοπιστίας των διαφόρων διαδρομών.

Πρωτόκολλα δρομολόγησης

- Τα πρωτόκολλα δρομολόγησης διακρίνονται σε πρωτόκολλα εσωτερικής δρομολόγησης (interior gateway protocols, IGP) και πρωτόκολλα εξωτερικής δρομολόγησης (exterior gateways protocols, EGP).
- Αυτόνομο σύστημα (autonomous system, AS) είναι μια ομάδα δρομολογητών και δικτύων ενός οργανισμού που χρησιμοποιούν το ίδιο πρωτόκολλο δρομολόγησης και στο οποίο υπάρχει μια διαδρομή μεταξύ οποιουδήποτε ζεύγους κόμβων.
- Τα πρωτόκολλα IGP εκτελούν δρομολόγηση εντός ενός αυτόνομου συστήματος (δίκτυα με κοινή δικτυακή διαχείριση, common administratrive domains).
- Παραδείγματα πρωτοκόλλων IGP: RIP (routing information protocol), OSPF (open shortest path first).
- Τα πρωτόκολλα EGP χρησιμοποιούνται για ανταλλαγή πληροφοριών δρομολόγησης μεταξύ αυτόνομων συστημάτων (μεταξύ δικτύων που δεν μοιράζονται κοινή δικτυακή διαχείριση).
- Το πιο χαρακτηριστικό πρωτόκολλο EGP είναι το BGP (border gateway protocol).

Δρομολόγηση κατάστασης συνδέσεων

- Μια δημοφιλής στρατηγική δυναμικής δρομολόγησης είναι η δρομολόγηση κατάστασης συνδέσεων (link-state routing).
- Κάθε δρομολογητής καθορίζει το κόστος μεταφοράς για κάθε σύνδεση σε κάθε δίκτυο, στο οποίο είναι συνδεδεμένος.
- Κατόπιν, κάθε δρομολογητής ανταλλάσσει αυτό το σύνολο του κόστους συνδέσεων με όλους τους άλλους δρομολογητές (όχι μόνο στους γειτονικούς).
- Κάθε δρομολογητής ελέγχει συνεχώς την κατάσταση των συνδέσεων και όταν συμβεί μια σημαντική αλλαγή (π.χ. σημαντική αλλαγή κόστους, δημιουργία νέας σύνδεσης, μη διαθεσιμότητα μιας σύνδεσης), ο δρομολογητής διαβιβάζει το νέο σύνολο κόστους ζεύξεων σε όλους τους δρομολογητές.
- Επειδή κάθε δρομολογητής λαμβάνει πληροφορία για τις συνδέσεις από όλους τους δρομολογητές, κάθε δρομολογητής μπορεί να γνωρίζει την τοπολογία του συνολικού δικτύου και μετά να υπολογίζει το πιο σύντομο μονοπάτι προς τον τελικό προορισμό των δεδομένων.
- Έτσι, προοδευτικά ο δρομολογητής καταστρώνει έναν πίνακα δρομολόγησης με μια λίστα των επόμενων διαθέσιμων βημάτων προς κάποιον προορισμό.
- Επειδή ο δρομολογητής έχει μια εικόνα του συνολικού δικτύου, μπορεί να χρησιμοποιήσει έναν αλγόριθμο καθορισμού του πιο σύντομου μονοπατιού.

Πρωτόκολλο δρομολόγησης OSPF

- Ένα πρωτόκολλο δυναμικής δρομολόγησης που ανήκει στην κατηγορία των πρωτοκόλλων IGP και αποτελεί αντιπροσωπευτικό παράδειγμα δρομολόγησης κατάστασης συνδέσεων είναι το πρωτόκολλο OSPF (open shortest path first).
- Το πρωτόκολλο OSPF χρησιμοποιείται στα δίκτυα που βασίζονται στο μοντέλο TCP/IP.
- Το OSPF (πρωτόκολλο επιλογής του πιο σύντομου μονοπατιού) προσδιορίζει ένα μονοπάτι στο δίκτυο που έχει το μικρότερο κόστος μεταφοράς.
- Ο χρήστης μπορεί να ορίσει τη μετρική του κόστους μεταφοράς, έτσι ώστε αυτή να εκφράζει την καθυστέρηση διάδοσης, το ρυθμό δεδομένων ή και άλλες παραμέτρους όπως το τρέχον φορτίο σύνδεσης (μήκος ουράς πακέτων).
- Το κόστος μεταφοράς μιας σύνδεσης μπορεί να διαφέρει στις δύο κατευθύνσεις της (όπως για παράδειγμα όταν το κόστος σύνδεσης είναι το μήκος ουράς των πακέτων που αναμένουν να μεταδοθούν από κάθε κόμβο της σύνδεσης).
- Με βάση το πρωτόκολλο OSPF κάθε δρομολογητής για κάθε πακέτο εκτελεί έναν αλγόριθμο για να αποφασίσει πού θα στείλει το πακέτο αυτό.
- Ένας αποδοτικός αλγόριθμος δρομολόγησης είναι ο αλγόριθμος του Dijkstra, με βάση τον οποίο ένας δρομολογητής υπολογίζει το μονοπάτι με το μικρότερο κόστος προς όλα τα δίκτυα προορισμού.

Αλγόριθμος του Dijkstra

- Σκοπός του αλγορίθμου είναι ο προσδιορισμός των συντομότερων μονοπατιών από ένα κόμβο-πηγή προς όλους τους άλλους κόμβους, κατατάσσοντας τα μονοπάτια κατά αύξουσα σειρά ως προς το κόστος τους.
- Είσοδος στον αλγόριθμο είναι η τοπολογία του δικτύου (οι κόμβοι και οι μεταξύ τους συνδέσεις), καθώς και τα κόστη μεταφοράς των συνδέσεων.
- Ο αλγόριθμος εκτελείται σε στάδια: στο στάδιο κ έχουν καθοριστεί τα μονοπάτια ελάχιστου κόστους (δηλαδή τα συντομότερα) από τον κόμβο-πηγή προς κ κόμβους. Οι κ κόμβοι ενσωματώνονται σε ένα σύνολο Π.
- Οι μεταβλητές του αλγορίθμου είναι οι ακόλουθες:

  N: σύνολο κόμβων δικτύου.

  π: κόμβος-πηγή.

  Π: σύνολο κόμβων που έχουν υποστεί επεξεργασία (εξεταστεί) από τον αλγόριθμο μέχρι το τρέχον στάδιο.

  w(i, j): κόστος σύνδεσης από τον κόμβο i στον κόμβο j. w(i, i) = 0, w(i, j) = ∞ όταν οι δύο κόμβοι δε συνδέονται απευθείας και w(i, j) > 0 όταν οι δύο κόμβοι συνδέονται απευθείας.

  L(v): κόστος ελάχιστου μονοπατιού από τον κόμβο-πηγή π έως τον κόμβο v, που μέχρι το τρέχον στάδιο είναι γνωστό στον αλγόριθμο.

Αλγόριθμος του Dijkstra

- Αρχικοποίηση ώστε το σύνολο Π να περιλαμβάνει μόνο τον κόμβο-πηγή π και να προσδιοριστεί το κόστος μονοπατιού των γειτονικών κόμβων του π:

  Π = {π};
  for all κόμβους v **in N do**
  if (v γειτονικός του π)
  then L(v) = w(π, v) **else** L(v) = ∞;

- Ενσωμάτωση στο σύνολο Π του πλησιέστερου στον π κόμβου x, αλλά και της σύνδεσης που τυχόν υπάρχει μεταξύ του κόμβου αυτού και ενός κόμβου που ανήκει στο Π. Επίσης, για όλους τους v γειτονικούς κόμβους του x που δεν ανήκουν στο Π προσδιόρισε το συντομότερο μονοπάτι τους από τον π (που είναι π → v = π → x → v):

  repeat
  for all κόμβους v **in N do** {
  Βρες κόμβο x **in N – Π** ώστε L(x) να είναι ελάχιστο
  Π = **Π** + {x}, L(x)
  for all v γειτονικούς κόμβους του x **και** v **in N – Π do**
  L(v) = **min** [L(v), L(x) + w(x, v)] };
  until Π = N;

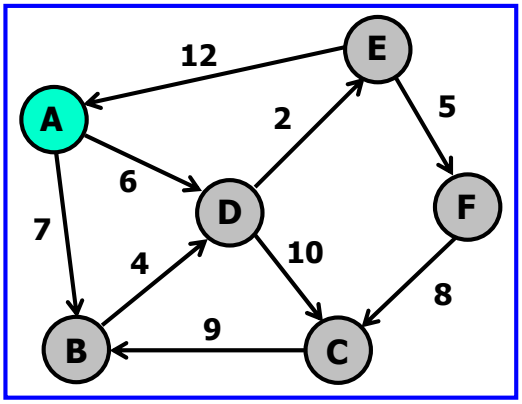
- Η παραπάνω διαδικασία επαναλαμβάνεται για όλους τους κόμβους του δικτύου, δηλαδή τερματίζεται όταν όλοι οι κόμβοι ενσωματωθούν στο σύνολο Π.

Παράδειγμα: αλγόριθμος του Dijkstra

N = 6 (σύνολο κόμβων), π = A (πηγή)

1η Επανάληψη: Αρχικοποίηση Π = {A}

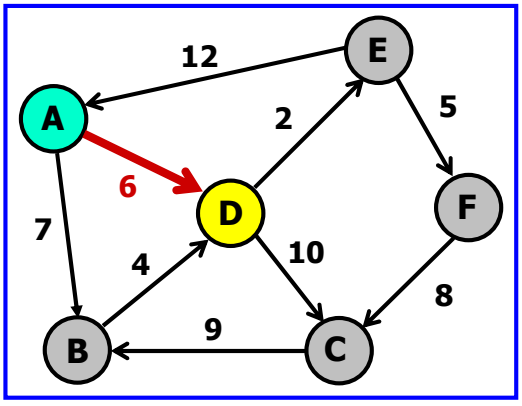
L(A) = 0
L(B) = 7
L(C) = ∞
L(D) = 6 (A-D) ⇒ ΕΛΑΧΙΣΤΗ
L(E) = ∞
L(F) = ∞



2η Επανάληψη Π = {A, D}

Ο πλησιέστερος κόμβος στον A εκτός Π, όπως προκύπτει από την αρχικοποίηση, είναι ο D και η σύνδεση με κόμβο που ανήκει στο Π είναι η AD.

L(A) = 0
L(B) = 7 (A-B) ⇒ ΕΛΑΧΙΣΤΗ
L(C) = 16
L(D) = 6 (A-D)
L(E) = 8
L(F) = ∞

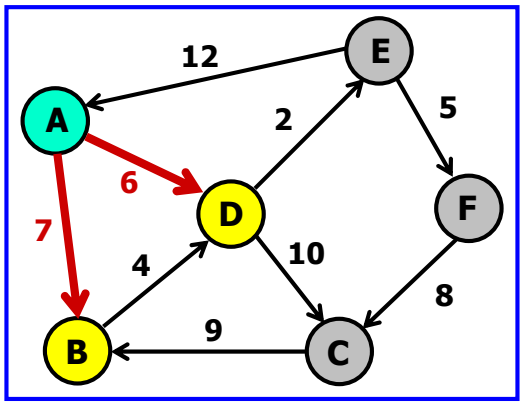


Παράδειγμα: αλγόριθμος του Dijkstra

3η Επανάληψη Π = {A, D, B}

Ο πλησιέστερος κόμβος στον Α εκτός Π, όπως προκύπτει από το 1ο βήμα, είναι ο Β και η σύνδεση που υπάρχει με κόμβο του Π είναι η ΑΒ.

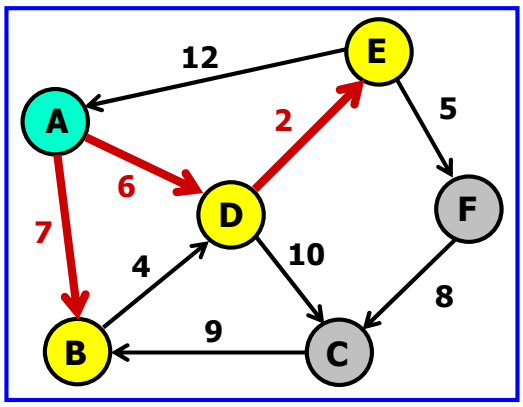
L(A) = 0
L(B) = 7 (A-B)
L(C) = 16
L(D) = 6 (A-D)
L(E) = 8 (D-E) ⇒ ΕΛΑΧΙΣΤΗ
L(F) = ∞



4η Επανάληψη Π = {A, D, B, E}

Ο πλησιέστερος κόμβος στον Α εκτός Π, όπως προκύπτει από το 2ο βήμα, είναι ο Ε και η σύνδεση που υπάρχει με κόμβο του Π είναι η DE.

L(A) = 0
L(B) = 7 (A-B)
L(C) = 16
L(D) = 6 (A-D)
L(E) = 8 (D-E)
L(F) = 13 (E-F) ⇒ ΕΛΑΧΙΣΤΗ

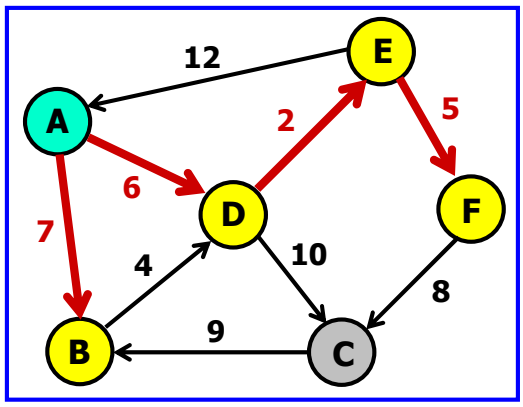


Παράδειγμα: αλγόριθμος του Dijkstra

5η Επανάληψη Π = {A, D, B, E, F}

Ο πλησιέστερος κόμβος στον Α εκτός Π, όπως προκύπτει από το 3ο βήμα, είναι ο F και η σύνδεση που υπάρχει με κόμβο που του Π είναι η EF.

L(A) = 0
L(B) = 7 (A-B)
L(C) = 16 (D-C) ⇒ ΕΛΑΧΙΣΤΗ
L(D) = 6 (A-D)
L(E) = 8 (D-E)
L(F) = 13 (E-F)



6η Επανάληψη Π = {A, D, B, E, F, C}

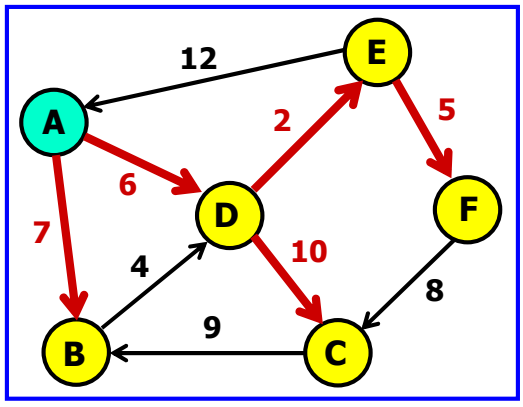
Ο πλησιέστερος κόμβος στον Α εκτός Π, όπως προκύπτει από το 4ο βήμα, είναι ο C και η σύνδεση που υπάρχει με κόμβο του Π είναι η DC.

L(A) = 0
L(B) = 7 (A-B)
L(C) = 16 (D-C)
L(D) = 6 (A-D)
L(E) = 8 (D-E)
L(F) = 13 (E-F)

⇒

L(A) = 0
L(B) = 7 (A-B)
L(C) = 16 (A-D-C)
L(D) = 6 (A-D)
L(E) = 8 (A-D-E)
L(F) = 13 (A-D-E-F)

Τέλος



Παράδειγμα: αλγόριθμος του Dijkstra

Συγκεντρωτικός πίνακας αποτελεσμάτων αλγορίθμου Dijkstra

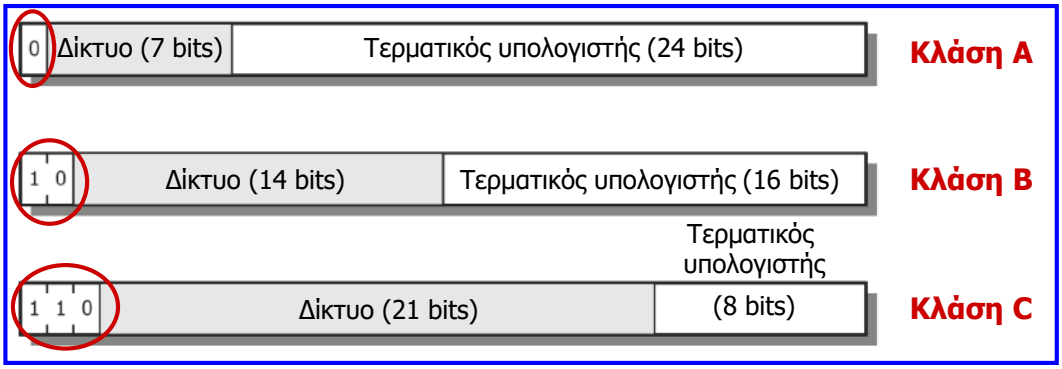
Επανά-ληψη	Σύνολο Π	L(B) / Μονοπάτι	L(C) / Μονοπάτι	L(D) / Μονοπάτι	L(E) / Μονοπάτι	L(F) / Μονοπάτι
1	Π = {A}	7 / A-B	∞ / –	6 / A-D	∞ / –	∞ / –
2	Π = {A, D}	7 / A-B	16 / A-D-C	6 / A-D	8 / A-D-E	∞ / –
3	Π = {A, D, B}	7 / A-B	16 / A-D-C	6 / A-D	8 / A-D-E	∞ / –
4	Π = {A, D, B, E}	7 / A-B	16 / A-D-C	6 / A-D	8 / A-D-E	13 / A-D-E-F
5	Π = {A, D, B, E, F}	7 / A-B	16 / A-D-C	6 / A-D	8 / A-D-E	13 / A-D-E-F
6	Π = {A, D, B, E, F, C}	7 / A-B	16 / A-D-C	6 / A-D	8 / A-D-E	13 / A-D-E-F

Διευθυνσιοδότηση στο πρωτόκολλο IPv4

- Σε κάθε τερματικό και ενδιάμεσο σύστημα του διαδικτύου ανατίθεται μια μοναδική διεύθυνση που αποτελείται από μια ακολουθία δυαδικών ψηφίων (32 bits για το IPv4).
- Η διεύθυνση IP (επιπέδου δικτύου) χρησιμοποιείται για τη δρομολόγηση ενός πακέτου μέσω ενός ή περισσότερων δικτύων προς το τελικό σύστημα προορισμού.
- Σκοπός των διευθύνσεων είναι η μοναδική αναγνώριση συστημάτων που συνδέονται σε ένα δίκτυο και γι' αυτό οι διευθύνσεις διαδικτύου είναι μοναδικές σε παγκόσμιο επίπεδο.
- Οι παγκόσμιες διευθύνσεις που έχουν ανατεθεί στα συστήματα αναγνωρίζονται από τα πρωτόκολλα και με τον τρόπο αυτό ένα σύστημα που συνδέεται σε ένα δίκτυο μπορεί να στείλει δεδομένα σε οποιοδήποτε άλλο σύστημα που συνδέεται σε κάποιο άλλο δίκτυο.
- Η ανάθεση διευθύνσεων IP σε παγκόσμιο επίπεδο γίνεται από φορείς διοίκησης του διαδικτύου (όπως η IANA, Internet assigned numbers authority), αποκεντρωμένα από δημόσιους φορείς και ISPs και τοπικά από τον διαχειριστή δικτύου, από τους χρήστες των συστημάτων ή αυτοματοποιημένα μέσω του DHCP (dynamic host configuration protocol).
- Στα δίκτυα δεδομένων, απαιτείται ένα ακόμη επίπεδο διευθυνσιοδότησης που αφορά το κατώτερο επίπεδο (π.χ. διεύθυνση MAC για δίκτυα Ethernet).
- Κάθε δίκτυο διατηρεί μία μοναδική διεύθυνση για κάθε σύστημα που συνδέεται σε αυτό, ώστε να είναι δυνατή η αποστολή πλαισίων δικτύου από το σύστημα του αποστολέα και η παράδοσή τους στο σύστημα του παραλήπτη.

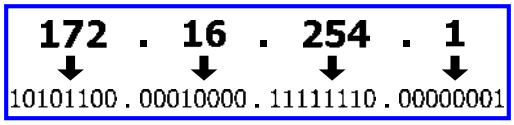
Διευθυνσιοδότηση στο πρωτόκολλο IPv4

- Οι διευθύνσεις πηγής και προορισμού στην επικεφαλίδα IP περιέχουν μία παγκόσμια διεύθυνση διαδικτύου (32 bits) που γενικά αποτελείται από αναγνωριστές (ID) δικτύου και τερματικού υπολογιστή (host).
- Η κωδικοποίηση των διευθύνσεων διενεργείται με χρήση τριών βασικών κλάσεων, ώστε να επιτρέπει διαφορετικούς συνδυασμούς πλήθους και μεγέθους δικτύων σε ένα διαδίκτυο.
- Κλάση Α (λίγα δίκτυα με πολλούς τερματικούς υπολογιστές ανά δίκτυο), κλάση Β (μέσο πλήθος δικτύων με μέσο πλήθος τερματικών υπολογιστών ανά δίκτυο), κλάση C (πολλά δίκτυα με μικρό πλήθος τερματικών υπολογιστών ανά δίκτυο).
- Σε συγκεκριμένο περιβάλλον (π.χ. εταιρικό διαδίκτυο) είναι καλύτερα να χρησιμοποιούνται διευθύνσεις της ίδιας κλάσης, αλλά είναι εφικτή και η ανάμειξη κλάσεων στο ίδιο διαδίκτυο.



Διευθυνσιοδότηση στο πρωτόκολλο IPv4

- Οι διευθύνσεις IP των 32 bits παριστάνονται με 4 δεκαδικούς αριθμούς (και τελείες μεταξύ τους), καθένας από τους οποίους αναπαριστά ένα byte:



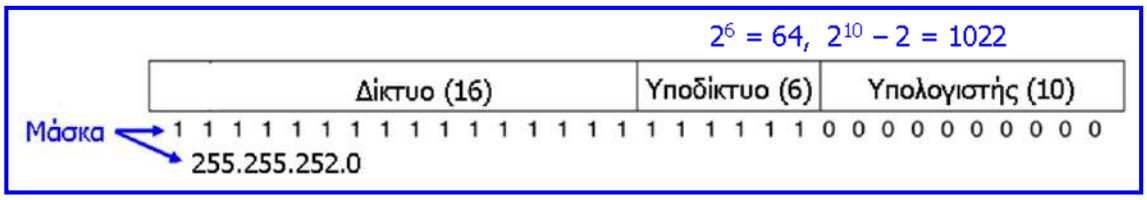
- Αυτός ο τρόπος παράστασης των διευθύνσεων IP αναφέρεται ως: διάστικτη δεκαδική σημειογραφία (dotted decimal notation).
- Στην κλάση Α, οι διευθύνσεις ξεκινούν με 0, οι διευθύνσεις δικτύων με πρώτο byte ίσο με 0 και 127 δεν αποδίδονται και έτσι υπάρχουν 126 δυνατοί αριθμοί δικτύων κλάσης Α με πρώτο δεκαδικό αριθμό διεύθυνσης από 1 έως 126.
- Στην κλάση Β, οι διευθύνσεις ξεκινούν με 10, το εύρος του πρώτου δεκαδικού αριθμού διεύθυνσης είναι από 128 έως 191 και το δεύτερο byte είναι επίσης μέρος του αναγνωριστή δικτύου, ώστε να είναι διαθέσιμοι 2^{14} = 16384 αναγνωριστές δικτύου κλάσης Β.
- Στην κλάση C, οι διευθύνσεις ξεκινούν με 110, το εύρος του πρώτου δεκαδικού αριθμού διεύθυνσης είναι από 192 έως 223 και τα δύο επόμενα bytes είναι επίσης μέρος του αναγνωριστή δικτύου, ώστε να είναι διαθέσιμοι 2^{21} = 2097152 αναγνωριστές δικτύου κλάσης C.

Διευθυνσιοδότηση χωρίς κλάσεις και υποδίκτυα

- Η διευθυνσιοδότηση χωρίς κλάσεις (classless interdomain routing, CIDR) επιτρέπει τον διαχωρισμό μεταξύ αναγνωριστών δικτύου & υπολογιστή σε οποιοδήποτε bit της διεύθυνσης.
- Το δίκτυο παριστάνεται με μια διεύθυνση IP ακολουθούμενη από μια κάθετη (/) και έναν δεκαδικό αριθμό που ισούται με το πλήθος των bits που αποτελούν τον αναγνωριστή δικτύου, π.χ. 184.13.152.0/22: αναγνωριστής δικτύου 22 bits, αναγνωριστής τερματικών υπολογιστών 10 bits, δηλαδή δίκτυο που διαθέτει έως 2^{10} = 1024 διαφορετικές διευθύνσεις.
- Η διευθυνσιοδότηση χωρίς κλάσεις επιτρέπει την δημιουργία υποδικτύων (subnets), έτσι ώστε η πολυπλοκότητα των διασυνδεδεμένων τοπικών δικτύων εντός μίας τοποθεσίας ή ενός οργανισμού να απομονώνεται από το ευρύτερο διαδίκτυο, αποφεύγοντας έτσι μεγάλη ανάπτυξη σε πλήθος δικτύων και αύξηση της πολυπλοκότητας δρομολόγησης.
- Μία προσέγγιση αποτελεί η ανάθεση ενός μοναδικού αριθμού δικτύου σε όλα τα τοπικά δίκτυα μιας τοποθεσίας ή ενός οργανισμού.
- Για το υπόλοιπο διαδίκτυο, υπάρχει ένα μοναδικό δίκτυο σε αυτή την τοποθεσία και έτσι απλοποιείται η διευθυνσιοδότηση και η δρομολόγηση.
- Σε κάθε τοπικό δίκτυο της τοποθεσίας ή του οργανισμού ανατίθεται ένα υποδίκτυο.
- Το τμήμα της διεύθυνσης IP που αντιστοιχεί στους τερματικούς υπολογιστές διαιρείται σε αριθμό (αναγνωριστή) υποδικτύου και σε αναγνωριστή τερματικών υπολογιστών, έτσι ώστε να εξυπηρετηθεί το νέο επίπεδο διευθυνσιοδότησης.

Διευθυνσιοδότηση χωρίς κλάσεις και υποδίκτυα

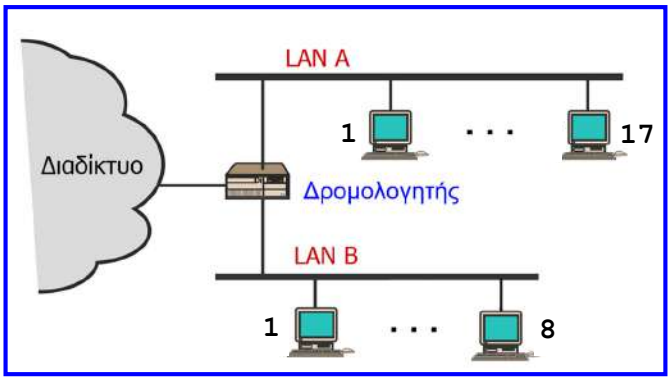
- Τα τμήματα της διεύθυνσης IP, εκτός του αναγνωριστή υπολογιστών, αναφέρονται ως πρόθεμα (prefix), ενώ ο αναγνωριστής υπολογιστών αναφέρεται ως κατάληξη (suffix).
- Οι θέσεις των bits του αναγνωριστή δικτύου και του αριθμού υποδικτύου της διεύθυνσης IP (prefix), υποδεικνύονται από αριθμό 32 bit, που αναφέρεται ως μάσκα υποδικτύου και περιέχει 1 μόνο στις θέσεις αυτές και 0 στις υπόλοιπες.
- Η λογική πράξη AND μεταξύ της μάσκας και της διεύθυνσης ενός υπολογιστή μηδενίζει τον αναγνωριστή υπολογιστή, επιτρέποντας σε έναν υπολογιστή να προσδιορίσει εάν ένα εξερχόμενο πακέτο προορίζεται για έναν υπολογιστή στο ίδιο ή σε άλλο τοπικό δίκτυο (υποδίκτυο), ώστε να το αποστείλει απευθείας ή μέσω δρομολογητή, αντίστοιχα.
- Παράδειγμα διεύθυνσης δικτύου με 64 υποδίκτυα και 1022 υπολογιστές το καθένα.



- Οι ακραίες τιμές του πεδίου υπολογιστή δεν αποδίδονται σε τερματικούς υπολογιστές αφού είναι κρατημένες για ειδικές χρήσεις. Η μέγιστη τιμή του αναγνωριστή υπολογιστή χρησιμοποιείται ως διεύθυνση ευρείας εκπομπής (broadcast address) και η μηδενική τιμή του αποτελεί το τελευταίο τμήμα της διεύθυνσης υποδικτύου (subnet address).

Παράδειγμα υποδικτύωσης

- Για τις ανάγκες ενός οργανισμού έχει εκχωρηθεί το υποδίκτυο 195.251.123.128/25, δηλαδή οι διευθύνσεις IP από 195.251.123.128 έως 195.251.123.255 ($2^{32 - 25} = 128$).
- Ο οργανισμός διαθέτει δρομολογητή συνδεδεμένο στο διαδίκτυο και σε δύο τοπικά δίκτυα: Α με 17 και Β με 8 Η/Υ.
- Ο δρομολογητής διαθέτει ξεχωριστή διεπαφή για κάθε τοπικό δίκτυο και απαιτείται η απόδοση διευθύνσεων IP και στις διεπαφές του δρομολογητή (προεπιλεγμένες πύλες).



- Χρησιμοποιώντας υποδικτύωση πρέπει επίσης να καθοριστούν οι παράμετροι IP (διεύθυνση IP, μάσκα υποδικτύου, προεπιλεγμένη πύλη), οι οποίες πρέπει να τεθούν στους υπολογιστές κάθε τοπικού δικτύου του οργανισμού.
- Το δίκτυο Α περιλαμβάνει 17 Η/Υ και τη διεπαφή του δρομολογητή, άρα χρειαζόμαστε 18 διευθύνσεις IP. Λαμβάνοντας υπόψη 2 ακόμη διευθύνσεις που δεν μπορούν να εκχωρηθούν σε Η/Υ, συγκεκριμένα η διεύθυνση ή ID υποδικτύου (όλα τα bits 0 στον αριθμό Η/Υ) και η διεύθυνση ευρείας εκπομπής (όλα τα bits 1 στον αριθμό Η/Υ), χρειαζόμαστε 20 αριθμούς υπολογιστή που απαιτούν 5 bits, οπότε η μάσκα του υποδικτύου Α είναι /27 (32 – 5 =27).

Παράδειγμα υποδικτύωσης

- Από τις 128 διευθύνσεις που εκχωρήθηκαν (μάσκα /25), δημιουργούμε το υποδίκτυο 195.251.123.128/27, που περιλαμβάνει 32 διευθύνσεις IP (από την 195.251.123.128 έως και την 195.251.123.159) με μάσκα /27 (255.255.255.224) και προεπιλεγμένη πύλη 195.251.123.129 (επιλέγουμε συνήθως την πρώτη διαθέσιμη διεύθυνση για Η/Υ).
- Η διεύθυνση IP του πρώτου Η/Υ του δικτύου Α είναι η 195.251.123.130 και του τελευταίου Η/Υ η 195.251.123.146, αφού το δίκτυο Α περιλαμβάνει 17 υπολογιστές.
- Ομοίως για το δίκτυο Β χρειαζόμαστε 11 διευθύνσεις IP (8 για τους Η/Υ, μία για τη διεπαφή του δρομολογητή, μία για το υποδίκτυο και μία για τη διεύθυνση ευρείας εκπομπής), άρα χρειαζόμαστε 4 bits για αριθμούς υπολογιστή και επομένως η μάσκα υποδικτύου του Β είναι /28.
- Δημιουργούμε λοιπόν το υποδίκτυο 195.251.123.160/28 με 16 διευθύνσεις IP ξεκινώντας από τη διεύθυνση 195.251.123.160 έως και την 195.251.123.175 με μάσκα /28 ή 255.255.255.240 και προεπιλεγμένη πύλη 195.251.123.161.
- Η διεύθυνση IP του πρώτου Η/Υ του δικτύου Β είναι η 195.251.123.162 και του τελευταίου Η/Υ η 195.251.123.169, αφού το δίκτυο Β περιλαμβάνει 8 υπολογιστές.
- Απαραίτητη προϋπόθεση για τη δημιουργία υποδικτύου είναι η πρώτη ελεύθερη διεύθυνση (ID υποδικτύου) να έχει μηδενικά τα bits που αντιστοιχούν στην διεύθυνση (ID) του υποδικτύου, προϋπόθεση που τηρείται στα δίκτυα Α και Β αφού στις πρώτες διευθύνσεις τους (195.251.123.128 και 195.251.123.160) τα τελευταία 5 και 4 bits, αντίστοιχα έχουν τιμή 0.

Παράδειγμα υποδικτύωσης

- Εάν επιθυμούσαμε να προσθέσουμε και τοπικό δίκτυο Γ με 16 Η/Υ, θα χρειαζόμασταν 19 διευθύνσεις IP (16 για τους Η/Υ, 1 για τη διεπαφή του δρομολογητή, 1 για το υποδίκτυο και 1 ως διεύθυνση ευρείας εκπομπής), επομένως απαιτούνται 5 bits για αριθμούς Η/Υ.
- Το υποδίκτυο 195.251.123.192/27 με 32 διευθύνσεις IP θα περιλάμβανε τις διευθύνσεις από 195.251.123.192 έως και 195.251.123.223 με μάσκα /27 ή 255.255.255.224.
- Εναλλακτικά, θα μπορούσαμε να δημιουργήσουμε το υποδίκτυο 195.251.123.224/27 με 32 διευθύνσεις IP (από 195.251.123.224 έως και 195.251.123.255) με μάσκα επίσης /27.
- Επισημαίνεται ότι, δεν θα μπορούσαμε να δημιουργήσουμε υποδίκτυο ξεκινώντας από τη διεύθυνση 195.251.123.176 (δηλαδή την επόμενη μετά την τελευταία διεύθυνση του υποδικτύου Β), διότι η συγκεκριμένη διεύθυνση δεν έχει τα 5 τελευταία δυαδικά ψηφία 0, όπως απαιτείται για την δημιουργία υποδικτύου με μάσκα /27.
- Κατά τη δημιουργία υποδικτύων, ξεκινάμε συνήθως από το υποδίκτυο με το μεγαλύτερο πλήθος υπολογιστών.
- Ο όρος υποδίκτυο αναφέρεται σε ένα σύνολο κόμβων που διαχωρίζεται από ένα άλλο σύνολο κόμβων με τη χρήση δρομολογητών.
- Στη γενική περίπτωση, για τη σύνδεση δρομολογητών που διαχωρίζουν υποδίκτυα, δημιουργείται ξεχωριστό υποδίκτυο.
- Ωστόσο, είναι δυνατό δύο δρομολογητές να συνδέονται στο ίδιο τοπικό δίκτυο.

Στατική δρομολόγηση

- Όταν ένα πακέτο IP λαμβάνεται σε έναν δρομολογητή, αυτός αρχικά ελέγχει εάν το πρόθεμα (prefix) της διεύθυνσης προορισμού του πακέτου ταιριάζει με το πρόθεμα κάποιας από τις διευθύνσεις προορισμού του πίνακα δρομολόγησης.
- Αυτό γίνεται μέσω λογικής πράξης AND μεταξύ της διεύθυνσης προορισμού του πακέτου και της μάσκας των διευθύνσεων προορισμού του πίνακα δρομολόγησης.
- Από το αποτέλεσμα προκύπτει εάν το πακέτο θα προωθηθεί σε δίκτυο με το οποίο συνδέεται άμεσα ο δρομολογητής ή σε άλλον δρομολογητή (next hop).
- Κανόνας ταιριάσματος μεγαλύτερου προθέματος (longest prefix match rule): κατά την αναζήτηση καταχώρησης στον πίνακα δρομολόγησης για τη διεύθυνση προορισμού ενός πακέτου, χρησιμοποιείται το μεγαλύτερο πρόθεμα διεύθυνσης προορισμού του πίνακα που ταιριάζει στη διεύθυνση προορισμού του πακέτου.
- Εάν δεν υπάρχει στον πίνακα δρομολόγησης καταχώρηση που να ταιριάζει στο πακέτο, αυτό προωθείται μέσω της προεπιλεγμένης διαδρομής (default route).
- Η προεπιλεγμένη διαδρομή δηλώνεται στον πίνακα δρομολόγησης με το υποδίκτυο προορισμού 0.0.0.0/0 και επόμενο βήμα τη διεύθυνση ενός άλλου δρομολογητή, ο οποίος επαναλαμβάνει την ίδια διαδικασία μέχρι το πακέτο IP να παραδοθεί στον προορισμό του.
- Συνήθως, τα πακέτα IP με προορισμούς εκτός του δικτύου ενός οργανισμού (δηλαδή, πακέτα της προεπιλεγμένης διαδρομής), όπως προορισμοί στο διαδίκτυο, προωθούνται σε δρομολογητή που συνδέεται στον πάροχο υπηρεσιών διαδικτύου.

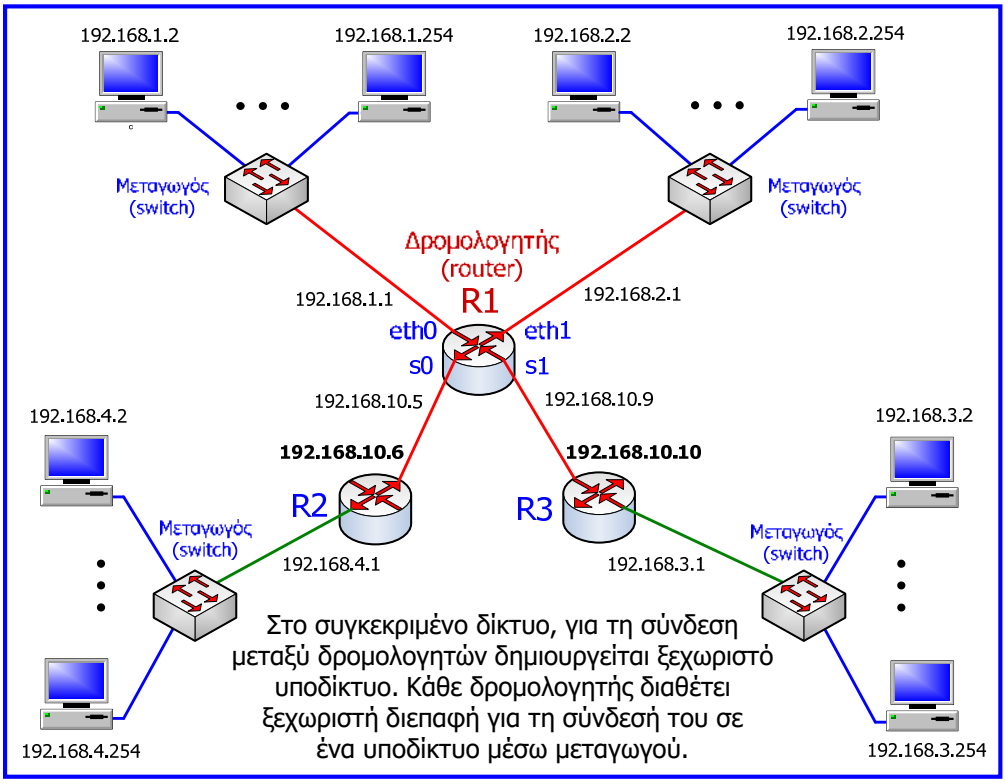
Παράδειγμα στατικής δρομολόγησης

Με δεδομένο τον παρακάτω πίνακα δρομολόγησης ενός δρομολογητή R1, καθώς και το ότι οι διευθύνσεις προορισμού του πίνακα δρομολόγησης είναι διευθύνσεις υποδικτύων που αφορούν ενσύρματα τοπικά δίκτυα, σχεδιάζουμε ένα διάγραμμα τοπολογίας του δικτύου, στο οποίο αντιστοιχεί ο πίνακας δρομολόγησης.

Διεύθυνση προορισμού (destination address)	Επόμενο βήμα (next hope)	Διεπαφή / θύρα (interface)
192.168.1.0/24	---	eth0
192.168.2.0/24	---	eth1
192.168.4.0/24	192.168.10.6	s0
192.168.3.0/24	192.168.10.10	s1

Η μη εμφάνιση διεύθυνσης επόμενου βήματος (---) ή η εμφάνιση ως διεύθυνση επόμενου βήματος του 0.0.0.0, υποδηλώνει τον ίδιο δρομολογητή.

Παράδειγμα στατικής δρομολόγησης



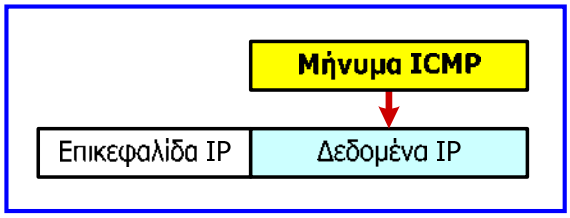
Τα πακέτα IP παραδίδονται απευθείας (όταν ο προορισμός ανήκει σε υποδίκτυο που συνδέεται άμεσα με το δρομολογητή) ή προωθούνται σε άλλον δρομολογητή για προορισμό που ανήκει σε υποδίκτυο που δεν συνδέεται άμεσα με το δρομολογητή.

Πρωτόκολλο ICMP

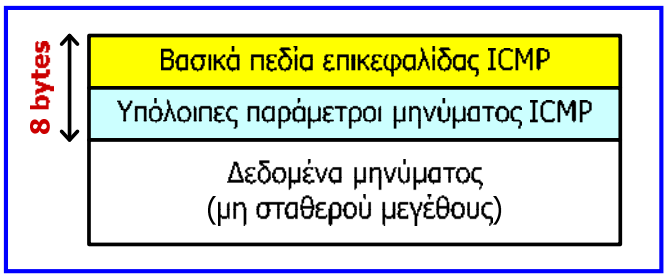
- Το πρωτόκολλο μηνυμάτων ελέγχου διαδικτύου (Internet control message protocol, ICMP) αποτελεί το μέσο για την ανταλλαγή μηνυμάτων ελέγχου μεταξύ δρομολογητών και τερματικών συστημάτων.
- Οι βασικές κατηγορίες μηνυμάτων ICMP είναι δύο: αναφοράς σφάλματος (error reporting) και άντλησης πληροφορίας (query).
- Ένα μήνυμα ICMP αναφοράς σφάλματος στέλνεται ως απόκριση στην αποστολή ενός πακέτου IP, από δρομολογητή που βρίσκεται στη διαδρομή του πακέτου αυτού ή από το τερματικό σύστημα προορισμού.
- Για παράδειγμα, όταν ένα πακέτο δεν μπορεί να φτάσει στον προορισμό του (άγνωστος προορισμός, εκπνοή διάρκειας ζωής) ή όταν ένας δρομολογητής διαπιστώσει σφάλμα στις παραμέτρους της επικεφαλίδας ενός πακέτου IP, στέλνεται μήνυμα ICMP αναφοράς σφάλματος.
- Τα μηνύματα ICMP άντλησης πληροφορίας, τα οποία υφίστανται ως ζεύγη, βοηθούν για παράδειγμα, τα τερματικά συστήματα να διαπιστώσουν την εφικτότητα επικοινωνίας με άλλα τερματικά συστήματα ή να ενημερωθούν για τους δρομολογητές του δικτύου.
- Επειδή το IP δεν εγγυάται την ασφαλή μετάδοση δεδομένων, εκτός από το πρωτόκολλο ανωτέρου επιπέδου TCP που υποστηρίζει αποτελεσματικά την ορθότητα της μετάδοσης, χρησιμοποιείται το πρωτόκολλο ICMP ως ένας μηχανισμός αναφοράς σφαλμάτων ή δυσλειτουργιών στο επίπεδο IP.

Δομή μηνυμάτων ICMP

- Αν και το ICMP θεωρείται ότι βρίσκεται στο ίδιο επίπεδο με το IP, μετά τη δημιουργία των μηνυμάτων ελέγχου από το ICMP, το IP τους προσθέτει μία επικεφαλίδα IP και κατόπιν μεταδίδει τα πακέτα που προκύπτουν, κατά το συνηθισμένο τρόπο.

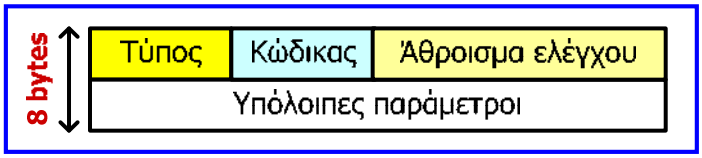


- Για το λόγο αυτό, η παράδοση των μηνυμάτων δεν εξασφαλίζεται και η χρήση τους δεν μπορεί να θεωρηθεί αξιόπιστη.
- Γενική δομή μηνύματος ICMP:



Δομή μηνυμάτων ICMP

- Ένα μήνυμα ICMP αρχίζει με μία επικεφαλίδα μεγέθους 8 bytes:



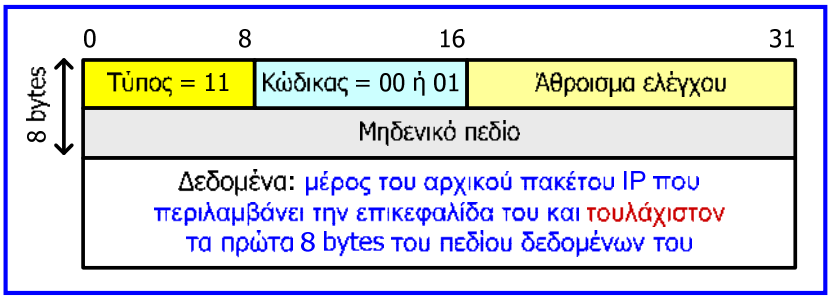
  - ✓ τύπος μηνύματος (1 byte),
  - ✓ κώδικας μηνύματος (1 byte),
  - ✓ άθροισμα ελέγχου (2 bytes) για το σύνολο του μηνύματος, δηλαδή επικεφαλίδα και δεδομένα (παράγεται με τρόπο αντίστοιχο με το άθροισμα της επικεφαλίδας IP).
  - ✓ υπόλοιπες παράμετροι του μηνύματος (4 bytes).
- Η επικεφαλίδα ακολουθείται από πεδίο δεδομένων.
- Όταν πρόκειται για μήνυμα ICMP αναφοράς σφάλματος (δηλαδή, το μήνυμα αναφέρεται σε πακέτο IP που έχει αποσταλεί), το πεδίο δεδομένων περιλαμβάνει την επικεφαλίδα IP και τουλάχιστον τα πρώτα 8 bytes του πεδίου δεδομένων του πακέτου αυτού.
- Όταν πρόκειται για μήνυμα ICMP άντλησης πληροφορίας, το περιεχόμενο και το μέγεθος του πεδίου δεδομένων καθορίζεται από την εφαρμογή που προκαλεί την αποστολή του μηνύματος (π.χ. εφαρμογές Ping και Traceroute).

Παραδείγματα τύπων μηνυμάτων ICMP

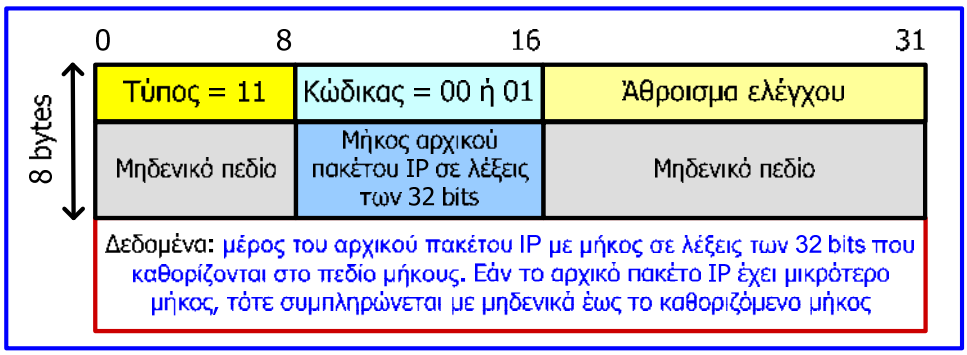
- Απροσπέλαστος προορισμός (Destination unreachable, τύπος 3): το μήνυμα επιστρέφεται από δρομολογητή, όταν δε μπορεί να προσεγγίσει ένα τερματικό σύστημα προορισμού [κώδικας 1 (απροσπέλαστο), 7 (άγνωστο)] ή ένα δίκτυο [κώδικας 0 (απροσπέλαστο), 6 (άγνωστο)] και από ένα τερματικό σύστημα όταν η θύρα προορισμού δε χρησιμοποιείται από κάποια εφαρμογή [κώδικας 3 (destination port unreachable)].
- Απαίτηση κερματισμού (Fragmentation required and DF set, τύπος 3, κώδικας 4): το μήνυμα επιστρέφεται από δρομολογητή σε τερματικό σύστημα όταν η MTU του δικτύου είναι μικρότερη από το μέγεθος του πακέτου που στάλθηκε και η σημαία DF έχει τιμή 1.
- Εκπνοή διάρκειας (Time exceeded, τύπος 11): το μήνυμα επιστρέφεται από δρομολογητή όταν η διάρκεια ζωής ενός πακέτου τελειώσει (κώδικας 0, time-to-live exceeded) ή από τερματικό σύστημα που αδυνατεί να επανασυναρμολογήσει ένα πακέτο εντός του επιτρεπόμενου χρόνου (κώδικας 1, fragment reassembly time exceeded).
- Ανακατεύθυνση (Redirect, τύπος 5, κώδικας 1): το μήνυμα στέλνεται από δρομολογητή στο τερματικό σύστημα από το οποίο έλαβε ένα πακέτο, προς ενημέρωση για την ύπαρξη συντομότερης διαδρομής (μέσω άλλου δρομολογητή) προς έναν προορισμό.
- Πρόβλημα παραμέτρων (Parameter problem, τύπος 12): ένα σφάλμα στην επικεφαλίδα IP μπορεί να προκαλέσει την επιστροφή του μηνύματος αυτού.
- Ηχώ και Απάντηση ηχούς (Echo request και Echo replay, τύπος 8 και 0, κώδικας 0): παρέχουν μηχανισμό δοκιμής επικοινωνίας μεταξύ των οντοτήτων ενός δικτύου.

Παραδείγματα δομής μηνυμάτων ICMP

- Εκπνοή διάρκειας (time exceeded):

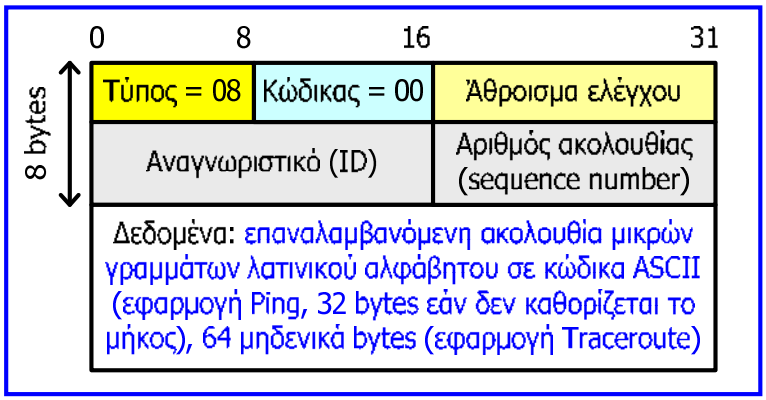


- Εκπνοή διάρκειας (time exceeded) στη διευρυμένη εκδοχή του ICMP (extended ICMP):



Παραδείγματα δομής μηνυμάτων ICMP

- Ηχώ (echo request):



Τα αναγνωριστικό (ID) πεδίο και το πεδίο αριθμού ακολουθίας (sequence number) συσχετίζονται με το μήνυμα ηχούς και πρέπει να είναι ίδια και στο μήνυμα απάντησης ηχούς (echo reply).

Το πεδίο δεδομένων πρέπει να ταιριάζει με το αντίστοιχο πεδίο του μηνύματος απάντησης ηχούς.

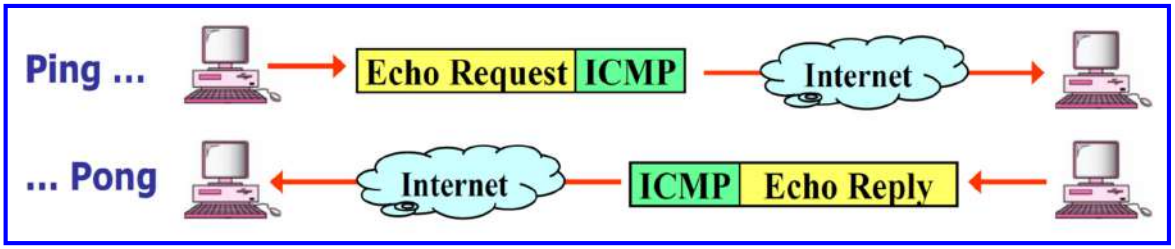
Εφαρμογή Ping

Η εφαρμογή Ping χρησιμοποιεί μηνύματα ICMP για την υλοποίηση διάφορων μηχανισμών, όπως η δοκιμή επικοινωνίας μεταξύ τερματικών συστημάτων και ο βέλτιστος προσδιορισμός της MTU (maximum transmission unit, μέγιστη μονάδα μετάδοσης) δικτύου.

```
C:\WINDOWS\system32> ping /?

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
            [-4] [-6] target_name

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet (IPv4-only).
    -i TTL         Time To Live.
    -v TOS         Type Of Service (IPv4-only. This setting has been deprecated
                   and has no effect on the type of service field in the IP
                   Header).
    -r count       Record route for count hops (IPv4-only).
    -s count       Timestamp for count hops (IPv4-only).
    -j host-list   Loose source route along host-list (IPv4-only).
    -k host-list   Strict source route along host-list (IPv4-only).
    -w timeout     Timeout in milliseconds to wait for each reply.
    -R             Use routing header to test reverse route also (IPv6-only).
                   Per RFC 5095 the use of this routing header has been
                   deprecated. Some systems may drop echo requests if
                   this header is used.
    -S srcaddr     Source address to use.
    -c compartment Routing compartment identifier.
    -p             Ping a Hyper-V Network Virtualization provider address.
    -4             Force using IPv4.
    -6             Force using IPv6.
```


Μηχανισμός δοκιμής επικοινωνίας

- Η εφαρμογή Ping χρησιμοποιείται ως μηχανισμός δοκιμής επικοινωνίας μεταξύ των δύο τερματικών συστημάτων του διαδικτύου.
- Από ένα τερματικό σύστημα πηγής, η εφαρμογή στέλνει ένα ICMP μήνυμα τύπου 8, κώδικα 0 (Ηχώ, Echo request) προς ένα τερματικό σύστημα προορισμού.
- Στην περίπτωση που το τερματικό σύστημα προορισμού είναι προσπελάσιμο, τότε επιστρέφει ένα ICMP μήνυμα τύπου 0, κώδικα 0 (Απάντηση ηχούς, Echo reply).
- Διαφορετικά, συμβαίνει εκπνοή διάρκειας ζωής του μηνύματος (timeout) του τερματικού συστήματος πηγής, το οποίο και αναφέρει την μη προσπελασιμότητα του συστήματος προορισμού.



Μηχανισμός δοκιμής επικοινωνίας

Η εφαρμογή ως προεπιλογή (default) στέλνει 4 πακέτα και παρέχει πληροφορία για το αν ελήφθη απάντηση και για την καθυστέρηση διάδοσης με επιστροφή (round trip time, RTT) για κάθε πακέτο

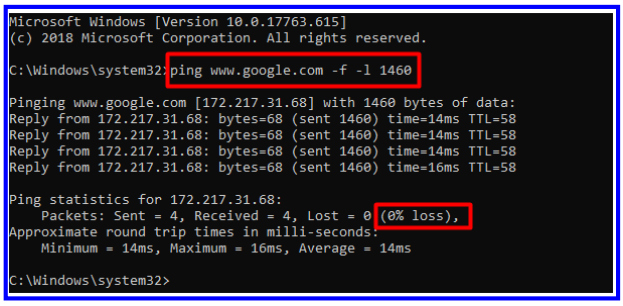
```
C:\> PING www.google.gr
Pinging www.l.google.com [216.239.59.147] with 32 bytes of data:
Reply from 216.239.59.147: bytes=32 time=181ms TTL=244
Reply from 216.239.59.147: bytes=32 time=110ms TTL=244
Reply from 216.239.59.147: bytes=32 time=111ms TTL=244
Reply from 216.239.59.147: bytes=32 time=110ms TTL=244
Ping statistics for 216.239.59.147:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 110ms, Maximum =  181ms, Average =  128ms
C:\>
```

```
C:\> PING www.in.gr
Pinging www.in.gr [194.63.247.208] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 194.63.247.208:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms
C:\>
```


Μηχανισμός βέλτιστου προσδιορισμού MTU

- Από τον κερματισμό (fragmentation) προκύπτουν επιμέρους πακέτα (fragments), με μικρότερο μέγεθος από το αρχικό πακέτο IP, ώστε αυτά να μεταδοθούν σε δίκτυο με μέγιστη μονάδα μετάδοσης (MTU) μικρότερου μεγέθους από το αρχικό πακέτο IP.
- Κερματισμός σε περισσότερα επιμέρους πακέτα από ότι απαιτείται, οδηγεί σε επιβάρυνση του δικτύου, συνεπώς είναι σημαντικός ο βέλτιστος προσδιορισμός της MTU μιας διαδρομής.
- Η εφαρμογή Ping στέλνει από ένα σύστημα πηγής, μήνυμα ICMP τύπου 8, κώδικα 0 (Ηχώ, Echo request), τέτοιο ώστε το πακέτο που θα μεταδοθεί προς ένα σύστημα τερματισμού να έχει μεγάλο μέγεθος, καθώς και τιμή 1 στη σημαία μη κερματισμού (don't fragment flag).
- Εάν κάποιος δρομολογητής στη διαδρομή του πακέτου, απορρίψει το πακέτο λόγω μικρότερου MTU του δικτύου του επόμενου άλματος, τότε επιστρέφει στο τερματικό σύστημα πηγής ένα μήνυμα ICMP τύπου 3 και κώδικα 4 (Απαίτηση κερματισμού, Fragmentation required and DF set).
- Το μήνυμα, εκτός από την επικεφαλίδα ICMP, περιλαμβάνει μέρος του πακέτου IP που στάλθηκε που περιλαμβάνει την επικεφαλίδα και τουλάχιστον τα πρώτα 8 bytes δεδομένων.
- Όταν λαμβάνει το μήνυμα ICMP, το σύστημα πηγής, ακολουθώντας την ίδια διαδικασία, μειώνει το μέγεθος του πακέτου και το ξαναστέλνει.
- Η παραπάνω διαδικασία επαναλαμβάνεται με αποστολή μικρότερων πακέτων, ώστε τελικά να επιτευχθεί βέλτιστος προσδιορισμός της μέγιστης μονάδας μετάδοσης (MTU).

Μηχανισμός βέλτιστου προσδιορισμού MTU



Μέγιστο μέγεθος πακέτου = 1460 bytes

Επικεφαλίδες IP, ICMP = 20 + 8 = 28 bytes

MTU = 1460 + 28 = 1488 bytes

Αφού εντοπίστηκε ένα μέγεθος πακέτου που δεν χρειάζεται κερματισμό, η διαδικασία θα μπορούσε να συνεχιστεί αυξάνοντας με μικρά βήματα το μέγεθος του πακέτου, ώστε να επιτευχθεί ακριβέστερος προσδιορισμός της MTU.

Εφαρμογή Traceroute

Η εφαρμογή Traceroute (σε λειτουργικό σύστημα Windows) χρησιμοποιεί μηνύματα ICMP για την ανίχνευση της διαδρομής μεταξύ δύο τερματικών συστημάτων.

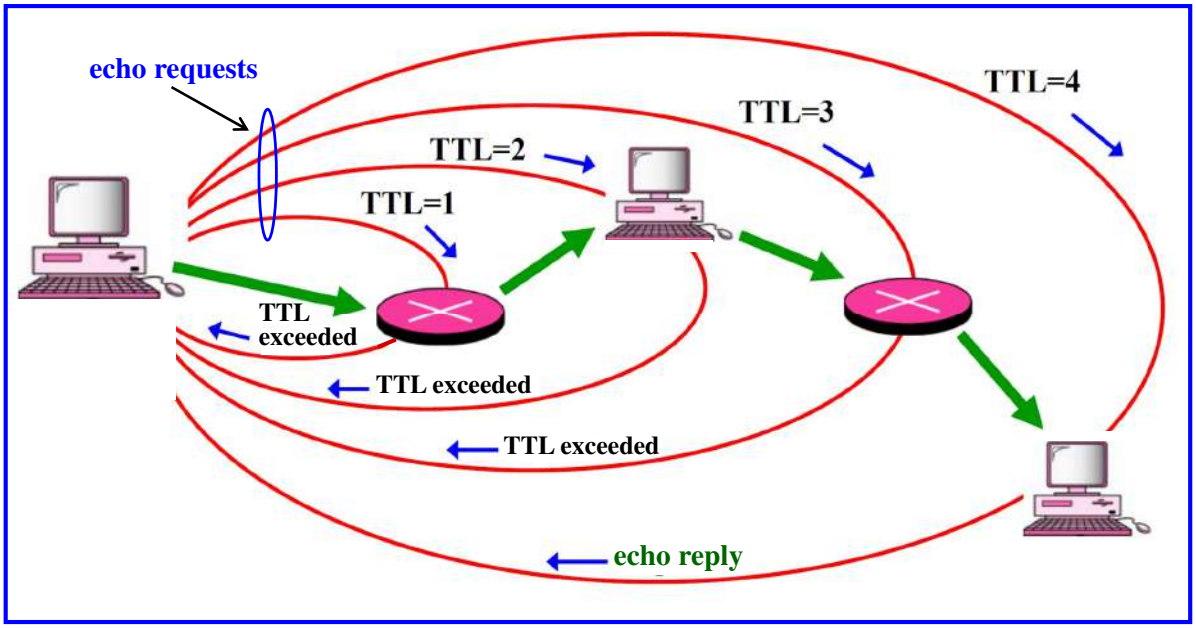
```
C:\WINDOWS\system32> tracert /?

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -d                 Do not resolve addresses to hostnames.
    -h maximum_hops    Maximum number of hops to search for target.
    -j host-list       Loose source route along host-list (IPv4-only).
    -w timeout         Wait timeout milliseconds for each reply.
    -R                 Trace round-trip path (IPv6-only).
    -S srcaddr         Source address to use (IPv6-only).
    -4                 Force using IPv4.
    -6                 Force using IPv6.
```

Μηχανισμός ανίχνευσης διαδρομής

- Η εφαρμογή Traceroute (σε λειτουργικό σύστημα Windows) μπορεί να στέλνει μηνύματα ICMP τύπου 8, κώδικα 0 (Ηχώ, Echo request) από ένα τερματικό σύστημα προς ένα άλλο τερματικό σύστημα, θέτοντας συγκεκριμένη τιμή στο πεδίο διάρκειας ζωής (TTL, time-to-live) των εξερχόμενων πακέτων IP.
- Αρχικά, στέλνει το εν λόγω μήνυμα ICMP θέτοντας την παράμετρο TTL στην τιμή 1, με αποτέλεσμα να μεταδοθεί προς το τερματικό σύστημα προορισμού ένα πακέτο IP με τιμή TTL = 1 στο αντίστοιχο πεδίο της επικεφαλίδας του, του οποίου η διάρκεια ζωής εκπνέει στον πρώτο (επόμενο) κόμβο (π.χ. δρομολογητή).
- Έτσι, ο πρώτος κόμβος απορρίπτει το πακέτο και επιστρέφει ένα μήνυμα ICMP τύπου 11 με κώδικα 0 (Εκπνοή διάρκειας ζωής, Time-to-live exceeded).
- Στη συνέχεια στέλνεται νέο μήνυμα ICMP (echo request) με TTL = 2, με αποτέλεσμα το εξερχόμενο πακέτο IP (με πεδίο TTL = 2) να απορριφθεί στον δεύτερο κόμβο, από τον οποίο επιστρέφεται ένα μήνυμα ICMP εκπνοής διάρκειας ζωής.
- Μέσω της εφαρμογής συνεχίζεται η αποστολή μηνυμάτων ICMP ηχούς με συνεχώς αυξανόμενη κατά 1 την τιμή του TTL και κάθε φορά που λαμβάνεται ICMP μήνυμα εκπνοής διάρκειας ζωής, καταγράφεται ο αποστολέας του και η αντίστοιχη καθυστέρηση διάδοσης με επιστροφή.
- Η διαδικασία ανίχνευσης διαδρομής ολοκληρώνεται όταν το σύστημα πηγής λάβει μήνυμα ICMP τύπου 0 και κώδικα 0 (Απάντηση ηχούς, Echo reply) από το σύστημα προορισμού.

Μηχανισμός ανίχνευσης διαδρομής

Traceroute: ανίχνευση διαδρομής με τη μέθοδο των επεκτεινόμενων δακτυλίων



Μηχανισμός ανίχνευσης διαδρομής

```
C:\> tracert www.google.gr
Tracing route to www.l.google.com [216.239.59.99]
over a maximum of 30 hops:
 1   10 ms   10 ms  <10 ms 10.0.0.1
 2  <10 ms  <10 ms   10 ms 10.0.0.138
 3   20 ms   30 ms   20 ms 80.76.42.179
 4   30 ms   30 ms    *    r722der_theE3.vivodi.gr [80.76.33.145]
 5   30 ms   30 ms   40 ms fe00r728der.vivodi.gr [80.76.32.30]
 6   91 ms  100 ms  100 ms se20r722lon.vivodi.gr [80.76.33.246]
 7    *     100 ms   90 ms S1-0-0.LONAR3.London.opentransit.net [193.251.252.145]
 8  100 ms  100 ms  100 ms P2-1.LONBB3.London.opentransit.net [193.251.154.93]
 9  100 ms  100 ms  100 ms So2-2-0.LONCR1.London.opentransit.net [193.251.241.105]
10  110 ms  110 ms  120 ms So1-0-0.FFTCR1.Frankfurt.opentransit.net [193.251.132.89]
11  110 ms  120 ms  120 ms P0-0.FFTCR3.Frankfurt.opentransit.net [193.251.242.154]
12  110 ms  110 ms  110 ms Google.GW.opentransit.net [193.251.249.62]
13  111 ms  110 ms    *    216.239.46.48
14  120 ms  120 ms  131 ms 64.233.175.249
15  120 ms  120 ms  120 ms 216.239.59.99
Trace complete.
C:\>
```

Στην πρώτη στήλη δίνεται το πλήθος των αλμάτων μέχρι τον κόμβο που ονοματίζεται στην τελευταία στήλη της ίδιας γραμμής. Στις επόμενες 3 στήλες δίνεται η καθυστέρηση διάδοσης με επιστροφή (round trip time, RTT) για τα τρία πακέτα (προεπιλογή) που στέλνονται κάθε φορά.

Η εφαρμογή σε Unix-like λειτουργικά συστήματα (Linux, Mac-OS κ.ά.) εκτελείται ως traceroute και λειτουργεί με παρόμοιο τρόπο εάν χρησιμοποιηθεί η επιλογή (option) –I, διαφορετικά στέλνει πακέτα επιπέδου μεταφοράς UDP (user datagram protocol, πρωτόκολλο αυτόνομου πακέτου).

Πρωτόκολλο διαδικτύου IPv6

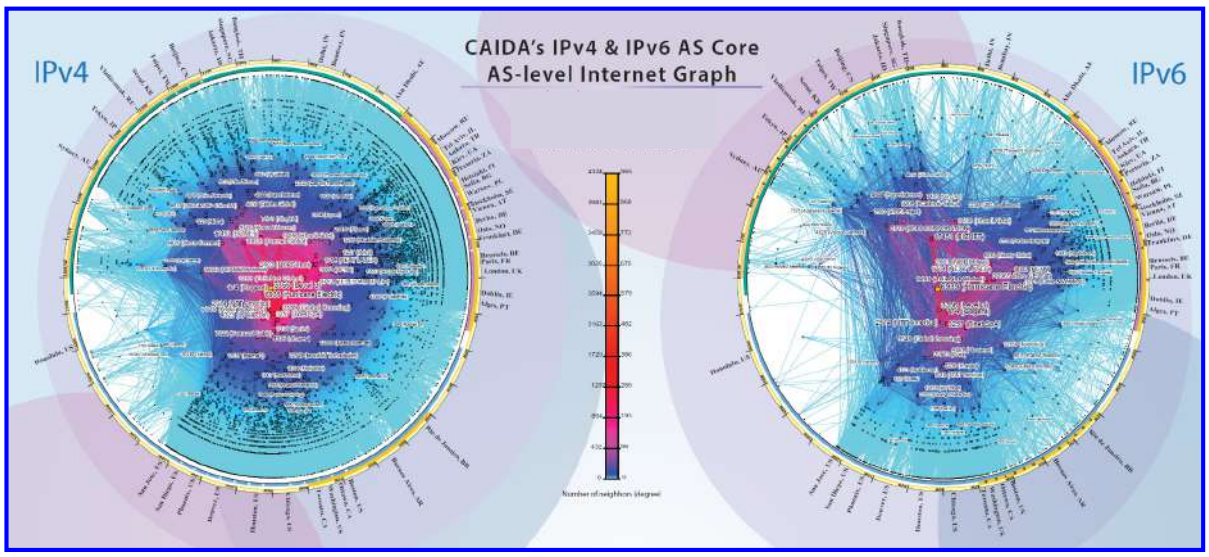
- Η έκδοση 4 του IP (IPv4) είναι αυτή που χρησιμοποιείται περισσότερο σήμερα, η έκδοση 5 προδιαγράφει ένα πρωτόκολλο διαδικτύου για λειτουργία με σύνδεση και η έκδοση 6 (IPv6) ήδη χρησιμοποιείται και σταδιακά αντικαθιστά την έκδοση 4.
- Η αντικατάσταση υπαγορεύεται από την απαίτηση για αυξημένο χώρο διευθύνσεων και από την απαίτηση για νέους τύπους υπηρεσιών.
- Ανεπάρκεια διευθύνσεων IPv4 λόγω των παρακάτω αιτιών:
  - ✓ Τα δύο πεδία των διευθύνσεων (δικτύου και υπολογιστή) βολεύουν αλλά οδηγούν σε σπατάλη του χώρου διευθύνσεων, αφού όταν εκχωρείται ένας αριθμός δικτύου, όλες οι σχετικές διευθύνσεις υπολογιστών εκχωρούνται σε αυτό το δίκτυο.
  - ✓ Οι διευθύνσεις που εκχωρούνται σε ένα δίκτυο είναι κατειλημμένες ακόμη και όταν το συγκεκριμένο δίκτυο δεν είναι συνδεδεμένο στο διαδίκτυο.
  - ✓ Εκρηκτική ανάπτυξη των επιμέρους δικτύων και του διαδικτύου.
  - ✓ Χρήση του μοντέλου TCP/IP και σε άλλους τομείς (τηλεοπτικοί δέκτες, τερματικά σημεία πώλησης κ.ά.).
  - ✓ Εκχώρηση μοναδικής διεύθυνσης IP σε κάθε υπολογιστή (πιο ευέλικτη διευθέτηση αποτελεί η δυνατότητα εκχώρησης πολλαπλών διευθύνσεων IP ανά υπολογιστή).

Βελτιώσεις στο πρωτόκολλο IPv6

- Διευθύνσεις μεγαλύτερου εύρους (128 bit) που οδηγούν σε αύξηση του χώρου διευθύνσεων.
- Βελτιωμένος μηχανισμός επιλογών: οι επιλογές του χρήστη ενσωματώνονται σε εναλλακτικές επικεφαλίδες επέκτασης που τοποθετούνται μεταξύ της επικεφαλίδας IPv6 και του τμήματος TCP και δε λαμβάνονται υπόψη από τους δρομολογητές με αποτέλεσμα την επιτάχυνση και την απλοποίηση της επεξεργασίας των πακέτων στους δρομολογητές και την ευκολότερη προσθήκη επιπλέον επιλογών.
- Δυναμική εκχώρηση διευθύνσεων.
- Αυξημένη ευελιξία διευθυνσιοδότησης που βασίζεται στη χρήση διεύθυνσης μερικής αποστολής όπου ένα πακέτο παραδίδεται σε μία ομάδα από κόμβους.
- Αποδοτικότερη κατανομή πόρων: αντί του πεδίου που διαθέτει η επικεφαλίδα IPv4 για να καθορίζει ειδική μεταχείριση ενός πακέτου (αξιοπιστία, χαμηλή καθυστέρηση κ.ά.), το IPv6 καθιστά δυνατό τον προσδιορισμό των πακέτων που ανήκουν σε μία συγκεκριμένη ροή κίνησης, για την οποία ο αποστολέας αιτείται ειδική μεταχείριση (π.χ. κινούμενη εικόνα πραγματικού χρόνου).
- Λόγω του ότι δεν είναι δυνατή η ταυτόχρονη αναβάθμιση όλων των δρομολογητών, ώστε να υποστηρίζουν το IPv6, μία λύση μετάβασης αποτελεί η ενθυλάκωση πακέτων IPv6 ως δεδομένα σε πακέτα IPv4 (packet within a packet approach), έτσι ώστε να είναι δυνατή η δρομολόγηση πακέτων IPv6 και από δρομολογητές που υποστηρίζουν μόνο το IPv4.

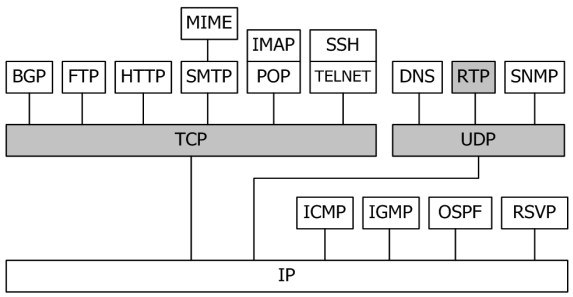
IPv4 vs IPv6

- Τα αυτόνομα συστήματα (AS) που λειτουργούν σήμερα δίκτυα IPv4 είναι περίπου 74.000, ενώ 31.000 αυτόνομα συστήματα υποστηρίζουν τη λειτουργία δικτύων IPv6. Το 2004, το ποσοστό των IPv6 αυτόνομων συστημάτων ήταν μόλις 0.5 %.
- Το πλήθος των γνωστών δικτύων IPv4 ανέρχεται σήμερα σε περίπου 935.000, ενώ το αντίστοιχο πλήθος γνωστών δικτύων IPv6 είναι περίπου 175.000.



Βασικές έννοιες πρωτοκόλλων μεταφοράς

- Τα πρωτόκολλα μεταφοράς παρέχουν στους χρήστες (εφαρμογές) υπηρεσία μεταφοράς δεδομένων από άκρο σε άκρο, απομονώνοντας τα ανώτερα επίπεδα της στοίβας από τα χαρακτηριστικά του παρεμβαλλόμενου δικτύου ή των παρεμβαλόμενων δικτύων.
- Ένα πρωτόκολλο μεταφοράς μπορεί να λειτουργεί (δηλαδή να παρέχει υπηρεσία μεταφοράς) με σύνδεση (connection-oriented) ή χωρίς σύνδεση (connectionless).
- Το πρωτόκολλο TCP παρέχει υπηρεσία με σύνδεση (σε ποικιλία εφαρμογών), ενώ για μερικά είδη εφαρμογών χρησιμοποιείται το πρωτόκολλο UDP (υπηρεσία χωρίς σύνδεση).



TCP: πρωτόκολλο ελέγχου μετάδοσης (transmission control protocol)

UDP: πρωτόκολλο αυτόνομου πακέτου (user datagram protocol)

RTP: πρωτόκολλο μεταφοράς πραγματικού χρόνου (real-time transfer protocol)

- Η πολυπλοκότητα των μηχανισμών ενός πρωτοκόλλου μεταφοράς διαφέρει ανάλογα με το αν η υπηρεσία διαδικτύου είναι αξιόπιστη ή μη αξιόπιστη (όπως συμβαίνει με το IP).

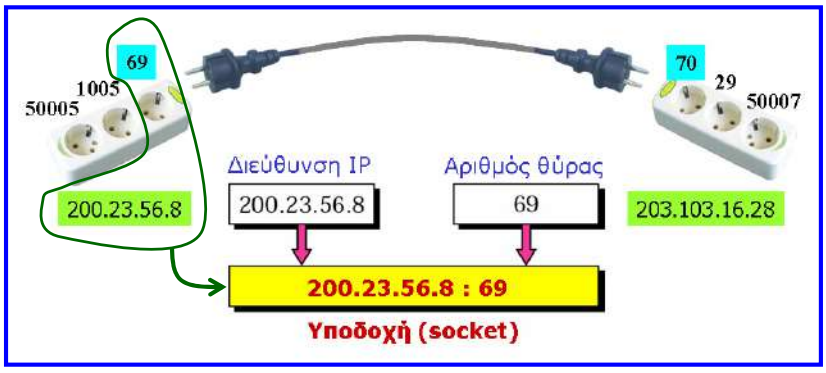
Βασικές έννοιες πρωτοκόλλων μεταφοράς

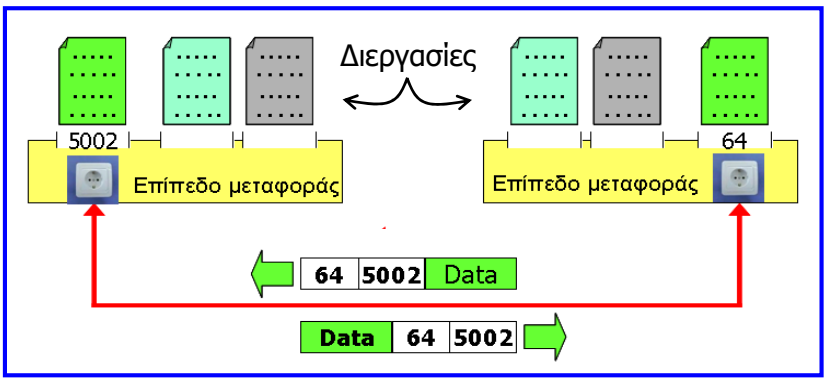
- Στην περίπτωση που η υπηρεσία δικτύου είναι αξιόπιστη (δηλαδή, παραδίδονται όλα τα πακέτα δεδομένων, χωρίς σφάλματα και με τη σωστή σειρά στον προορισμό), το πρωτόκολλο μεταφοράς είναι σχετικά απλό.
- Η αξιόπιστη υπηρεσία δικτύου χρησιμοποιεί ακολουθιακή αρίθμηση πακέτων και επιτρέπει τη χρήση απλού πρωτοκόλλου μεταφοράς, το οποίο αρκεί να υλοποιεί μηχανισμούς για διεργασίες, όπως: διευθυνσιοδότηση, πολυπλεξία, έλεγχος ροής, εγκαθίδρυση ή αποκατάσταση σύνδεσης και τερματισμός σύνδεσης.
- Η μη αξιόπιστη υπηρεσία δικτύου (όπως αυτή που παρέχει το επίπεδο IP) δημιουργεί αρκετά προβλήματα, όπως: απώλεια ή καταστροφή τμημάτων δεδομένων και άφιξη τμημάτων δεδομένων εκτός σειράς κυρίως λόγω μεταβλητών καθυστερήσεων μεταφοράς.
- Το πρωτόκολλο μεταφοράς στην περίπτωση μη αξιόπιστης υπηρεσίας δικτύου πρέπει να διευθετεί διεργασίες όπως ο έλεγχος ροής, η εγκαθίδρυση και ο τερματισμός σύνδεσης, αλλά και να λύνει τα προβλήματα που προκύπτουν λόγω αναξιοπιστίας του δικτύου, υλοποιώντας μηχανισμούς για πρόσθετες διεργασίες, όπως: διατεταγμένη παράδοση, έλεγχο για σφάλματα, αναμετάδοση και ανίχνευση αντιγράφων των τμημάτων δεδομένων.
- Στην ερώτηση γιατί χρειάζονται οι διεργασίες αυτές, αφού οι περισσότερες από αυτές περιλαμβάνονται και στο επίπεδο ελέγχου ζεύξης δεδομένων (DLC), η απάντηση είναι ότι το επίπεδο μεταφοράς παρέχει τις σχετικές υπηρεσίες από άκρο σε άκρο (δηλαδή μεταξύ τερματικών συστημάτων), ενώ το DLC παρέχει τις υπηρεσίες του ανά ζεύξη.

Διευθυνσιοδότηση

- Κατά τη μεταφορά δεδομένων μεταξύ δύο οντοτήτων μεταφοράς, η διεύθυνση του παραλήπτη καθορίζεται από τα εξής: ταυτότητα χρήστη (εφαρμογής), ταυτότητα οντότητας μεταφοράς, διεύθυνση τερματικού συστήματος και αναγωριστή δικτύου.
- Η θύρα (port) αντιπροσωπεύει ένα συγκεκριμένο χρήστη υπηρεσίας μεταφοράς (διεργασία εφαρμογής) στο καθορισμένο τερματικό σύστημα και περιλαμβάνεται στην επικεφαλίδα κάθε τμήματος δεδομένων του πρωτοκόλλου μεταφοράς.
- Στο πρωτόκολλο TCP, ο συνδυασμός θύρας και διεύθυνσης IP τερματικού συστήματος αναφέρεται ως υποδοχή (socket).
- Αν υπάρχουν περισσότερες από μία οντότητες μεταφοράς σε ένα τερματικό σύστημα, η διεύθυνση χρήστη θα πρέπει να προσδιορίζει και τον τύπο (ταυτότητα) του πρωτοκόλλου μεταφοράς (π.χ. TCP ή UDP).
- Πολλαπλοί χρήστες χρησιμοποιούν το ίδιο πρωτόκολλο μεταφοράς, δηλαδή διαφορετικές εφαρμογές χρησιμοποιούν διαφορετικές θύρες του ιδίου τερματικού συστήματος με αποτέλεσμα να είναι ενεργές ταυτόχρονα πολλαπλές συνδέσεις οντοτήτων μεταφοράς μεταξύ δύο τερματικών συστημάτων.
- Για παράδειγμα, το βασικό πρωτόκολλο μεταφοράς μηνυμάτων ηλεκτρονικού ταχυδρομείου (SMTP, θύρα 25) έχει διαφορετικό αριθμό θύρας από το πρωτόκολλο μεταφοράς υπερκειμένου (HTTP, θύρα 80)
- Ο αριθμός θύρας κάθε εφαρμογής είναι μοναδικός σε κάθε τερματικό σύστημα.

Διευθυνσιοδότηση





Θύρες (ports) TCP / UPD	
0 – 1023	Γνωστές (well-known)
1024 – 49151	Καταχωρημένες (registered)
49152 – 65535	Εφήμερες (ephemeral)

Πολυπλεξία

- Όσον αφορά τη διεπαφή του με ανώτερα επίπεδα, το πρωτόκολλο μεταφοράς εκτελεί μία λειτουργία πολυπλεξίας, δηλαδή πολλαπλοί χρήστες χρησιμοποιούν το ίδιο πρωτόκολλο μεταφοράς και διακρίνονται από αριθμούς θυρών (ports) ή σημεία πρόσβασης υπηρεσίας (service access points, SAPs).

- Μια οντότητα μεταφοράς μπορεί επίσης να εκτελέσει λειτουργία πολυπλεξίας όσον αφορά τις υπηρεσίες δικτύου (δηλαδή, του κατώτερου επιπέδου) που χρησιμοποιεί.

- Όσον αφορά λοιπόν τις υπηρεσίες δικτύου, η οντότητα μεταφοράς μπορεί για παράδειγμα να πολυπλέξει κάποιους χρήστες της σε ένα νοητό κύκλωμα του δικτύου μεταγωγής που χρησιμοποιεί (ανοδική πολυπλεξία).

- Επίσης, μπορεί να διαχωρίσει μία σύνδεση μεταφοράς μεταξύ πολλαπλών συνδέσεων χαμηλότερου επιπέδου (καθοδική πολυπλεξία ή διαχωρισμός), ώστε να αυξήσει τη διαμετακομιστική ικανότητα για τη συγκεκριμένη σύνδεση μεταφοράς.

- Ωστόσο, η διαμετακομιστική ικανότητα μίας σύνδεσης μεταφοράς δεν μπορεί να ξεπερνά το ρυθμό μετάδοσης δεδομένων της διαθέσιμης ζεύξης.

Διατεταγμένη παράδοση τμημάτων δεδομένων

- Με μία μη αξιόπιστη υπηρεσία δικτύου, ακόμη και αν παραδοθούν όλα τα τμήματα δεδομένων στο τερματικό σύστημα προορισμού, είναι πιθανό κάποια τμήματα να φτάσουν εκτός σειράς.

- Λύση στο πρόβλημα αυτό αποτελεί η διαδοχική αρίθμηση των τμημάτων και η αναδιάταξη (όταν απαιτείται) των τμημάτων δεδομένων από τον παραλήπτη.

- Το TCP αριθμεί διαδοχικά κάθε byte δεδομένων που μεταδίδεται και σε κάθε τμήμα δεδομένων (segment) αναθέτει τον αριθμό ακολουθίας ή αριθμό σειράς (SN) του πρώτου byte που αυτό περιλαμβάνει.

- Για παράδειγμα εάν κάθε τμήμα περιέχει 200 bytes δεδομένων, τότε ο αριθμός ακολουθίας κάθε τμήματος είναι κατά 200 μεγαλύτερος από εκείνον του προηγούμενου τμήματος.

- Εάν λοιπόν ένα τμήμα περιέχει 200 bytes δεδομένων και έχει αριθμό ακολουθίας (sequence number) SN = 401, τότε το επόμενο τμήμα λαμβάνει αριθμό ακολουθίας SN = 601.

Έλεγχος ροής

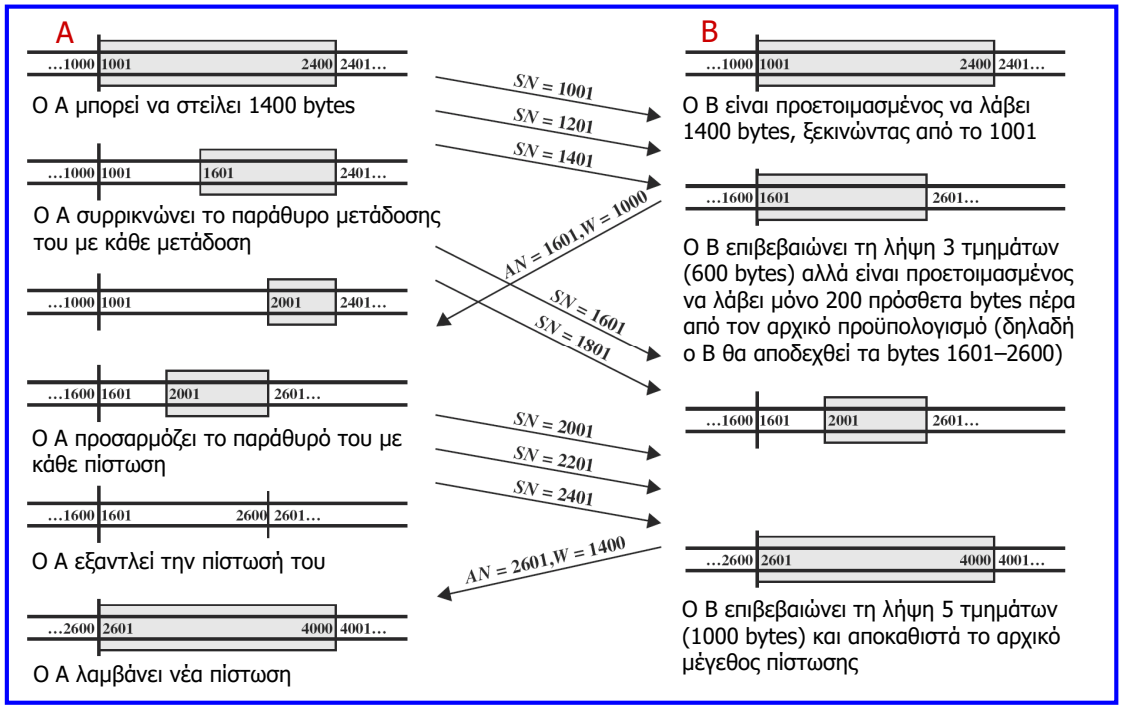
- Μια οντότητα μεταφοράς πρέπει να λαμβάνει μέτρα, έτσι ώστε να σταματάει ή να μειώνει τη ροή των δεδομένων που λαμβάνει (έλεγχος ροής), όταν υπάρχει κίνδυνος υπερχείλισης της ενδιάμεσης μνήμης που διαθέτει.
- Η λαμβάνουσα οντότητα μεταφοράς μπορεί να απορρίψει τα τμήματα που υπερχειλίζουν την ενδιάμεση μνήμη της ή να αρνηθεί τη λήψη επιπλέον τμημάτων ή να χρησιμοποιήσει την τεχνική του ολισθαίνοντος παραθύρου.
- Για μη αξιόπιστη υπηρεσία δικτύου (όπως το IP), είναι πιο αποδοτικό να αποσυνδέεται ο έλεγχος ροής από τις επιβεβαιώσεις λήψης, αφού μπορεί για έναν αριθμό τμημάτων που δεν έχει ληφθεί επιβεβαίωση, να γίνονται επαναληπτικές αναμεταδόσεις, ακόμη κι αν τα τμήματα έχουν ληφθεί αλλά έχει καθυστερήσει η επιβεβαίωσή τους.
- Έτσι, χρησιμοποιείται ένα σχήμα πίστωσης που εξοπλίζει τον παραλήπτη με μεγαλύτερο βαθμό ελέγχου της ροής δεδομένων και είναι πιο αποδοτικός για μη αξιόπιστα δίκτυα.
- Στο σχήμα αυτό, κάθε byte δεδομένων διαθέτει έναν αριθμό ακολουθίας και κάθε τμήμα που στέλνεται περιλαμβάνει στην επικεφαλίδα του τον αριθμό ακολουθίας του πρώτου byte των δεδομένων του, τον αριθμό επιβεβαίωσης λήψης και το μέγεθος παραθύρου.
- Ένα τμήμα δεδομένων μπορεί να επιβεβαιωθεί χωρίς την εκχώρηση νέας πίστωσης και αντίστροφα (δηλαδή αποσυνδέεται ο έλεγχος ροής από τις επιβεβαιώσεις).

Έλεγχος ροής

- Η οντότητα μεταφοράς προορισμού (παραλήπτης) επιβεβαιώνει τη λήψη ενός τμήματος με την επιστροφή ενός τμήματος που περιλαμβάνει τα στοιχεία AN = i και W = j.
- Τα στοιχεία i και j σημαίνουν ότι η λήψη όλων των bytes με αριθμό ακολουθίας μέχρι SN = i – 1 έχει επιβεβαιωθεί με το επόμενο αναμενόμενο byte να έχει αριθμό ακολουθίας i και ότι παρέχεται άδεια για την αποστολή ενός πρόσθετου παραθύρου μεγέθους W = j bytes, δηλαδή των bytes που αντιστοιχούν σε αριθμούς ακολουθίας από i έως (i + j – 1).
- Ο παραλήπτης (με βάση το διαθέσιμο χώρο της ενδιάμεσης μνήμης του) έχει τη δυνατότητα να αυξήσει την πίστωση κατά k (k > j), στέλνοντας (AN = i, W = k).
- Επίσης, ο παραλήπτης μπορεί να επιβεβαιώσει τη λήψη m bytes (m < j) χωρίς την παραχώρηση επιπρόσθετης πίστωσης, στέλνοντας (AN = i + m, W = j – m).
- Εάν ο παραλήπτης στείλει (AN = i, W = 0) κλείνοντας προσωρινά το παράθυρο και στη συνέχεια στείλει το τμήμα (AN = i, W = j) αλλά αυτό χαθεί, τότε ο αποστολέας δεν μπορεί να στείλει δεδομένα, ενώ ο δέκτης του έχει δώσει το δικαίωμα.
- Το αδιέξοδο ξεπερνιέται με το χρονόμετρο επιμονής (μέγιστος χρόνος μεταξύ δύο τμημάτων επιβεβαίωσης/πίστωσης), το οποίο εάν εκπνεύσει, ο αποστολέας απαιτείται να στείλει ένα τμήμα (έστω και αντίγραφο προηγούμενου) διατηρώντας την επικοινωνία ενεργή.

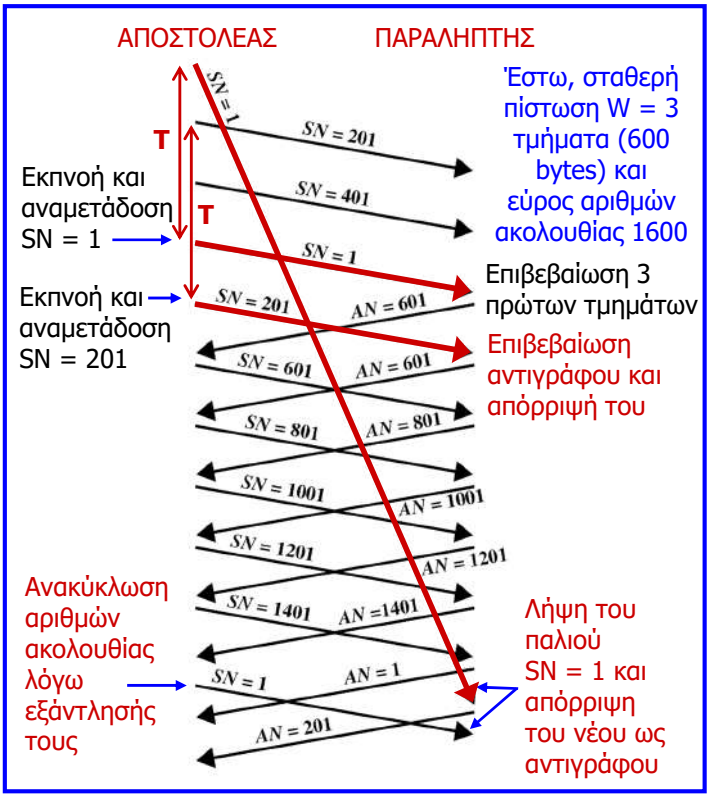
Έλεγχος ροής

Έστω, 200 bytes δεδομένων ανά τμήμα και αρχική πίστωση (διαφημιζόμενο παράθυρο) 1400 bytes στην αποστέλλουσα οντότητα, ξεκινώντας με αριθμό ακολουθίας byte 1001



Ανίχνευση αντιγράφων τμημάτων

- Εάν μία επιβεβαίωση λήψης δεν ληφθεί έγκαιρα ή χαθεί, μπορεί να φτάσουν στον παραλήπτη αντίγραφα τμημάτων.
- Όταν αποστέλλεται ένα τμήμα, αντιστοιχίζεται σε αυτό ένα χρονόμετρο αναμετάδοσης και εάν το χρονόμετρο εκπνεύσει πριν την επιβεβαίωση λήψης του τμήματος αυτού, ο αποστολέας πρέπει να το αναμεταδώσει, ανεξάρτητα από τη διαθέσιμη πίστωση.
- Εάν πρόκειται για αντίγραφο, ο παραλήπτης από τον αριθμό ακολουθίας του τμήματος, υποθέτει ότι η επιβεβαίωση λήψης του δεν έχει ληφθεί ακόμη ή έχει χαθεί και επιβεβαιώνει τη λήψη του αντιγράφου.
- Το εύρος αριθμών ακολουθίας πρέπει να είναι μεγάλο, ώστε να μην «ανακυκλώνεται» σε μικρότερο χρόνο από τη μέγιστη διάρκεια ζωής ενός τμήματος.

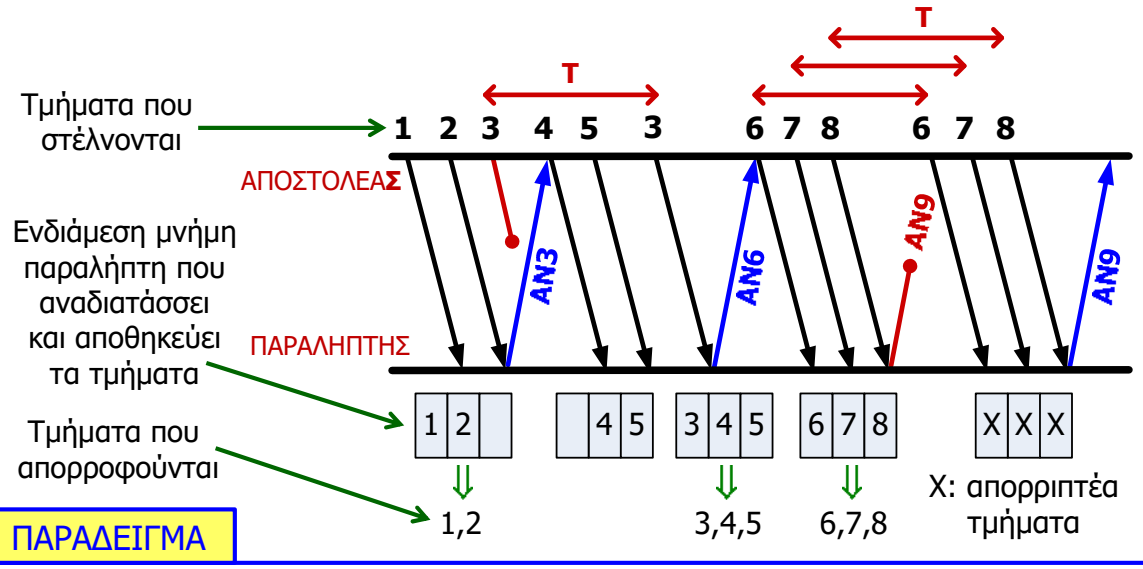


Έλεγχος σφαλμάτων και αναμετάδοση τμημάτων

- Αναμετάδοση τμημάτων επιβάλλεται όταν ένα τμήμα δεδομένων καταστραφεί κατά τη μεταφορά (αλλά έχει παραδοθεί) ή όταν ένα τμήμα δεδομένων αποτύχει να φτάσει.
- Στην πρώτη περίπτωση η λαμβάνουσα οντότητα μεταφοράς μπορεί να ανιχνεύσει το σφάλμα μέσω κατάλληλου αθροίσματος ελέγχου και να απορρίψει το τμήμα δεδομένων.
- Και στις δύο περιπτώσεις, η αποστέλλουσα οντότητα δεν γνωρίζει ότι η μετάδοση του τμήματος ήταν ανεπιτυχής, αλλά αυτό ξεπερνιέται με τη χρησιμοποίηση αθροιστικών επιβεβαιώσεων λήψης, που προαναφέρθηκε.
- Όπως είδαμε, εάν για παράδειγμα κάθε τμήμα περιέχει 200 bytes, ο δέκτης μπορεί να λάβει τα τμήματα 1, 201 και 401, αλλά να επιστρέψει μόνο την πληροφορία AN = 601, που σημαίνει ότι το τμήμα με SN = 401 και όλα τα προηγούμενα του έχουν ληφθεί επιτυχώς.
- Εάν ένα τμήμα δε φτάσει επιτυχώς, δεν θα εκδοθεί επιβεβαίωση λήψης (AN) και τότε είναι αναγκαία η επαναμετάδοσή του.
- Η κατάσταση αντιμετωπίζεται με την αντιστοίχιση ενός χρονομέτρου αναμετάδοσης σε κάθε τμήμα όταν αυτό στέλνεται και εάν το χρονόμετρο εκπνεύσει πριν την επιβεβαίωση λήψης του αντίστοιχου τμήματος, τότε ο αποστολέας πρέπει να το αναμεταδώσει.
- Η απώλεια επιβεβαιώσεων λήψης ξεπερνιέται, αφού στέλνονται μελλοντικές επιβεβαιώσεις λήψης ή το χρονόμετρο αναμετάδοσης του αποστολέα εκπνέει και αυτός αναμεταδίδει τα τμήματα δεδομένων που θεωρεί ότι δεν επιβεβαιώθηκαν.

Έλεγχος σφαλμάτων και αναμετάδοση τμημάτων

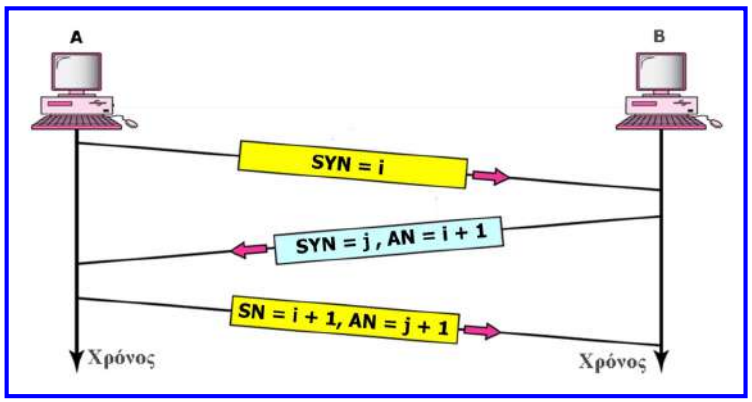
- Υποθέτουμε σταθερή πίστωση (διαφημιζόμενο παράθυρο) παραλήπτη W = 3 τμήματα.
- Ο παραλήπτης διαθέτει ενδιάμεση μνήμη αποθήκευσης και αναδιάταξης τμημάτων (re-order buffer) με μέγεθος τουλάχιστο ίσο με W.
- T: χρόνος αναμονής αναμετάδοσης (χρονόμετρο αναμετάδοσης τμήματος)



Εγκαθίδρυση και τερματισμός σύνδεσης

- Η εγκαθίδρυση (αποκατάσταση) σύνδεσης επιτρέπει σε κάθε άκρο να βεβαιωθεί ότι το άλλο υπάρχει, επιτρέπει τη διαπραγμάτευση προαιρετικών παραμέτρων (π.χ. μέγιστο μέγεθος τμήματος ή παραθύρου) και ενεργοποιεί την κατανομή πόρων της οντότητας μεταφοράς στους χρήστες (π.χ. χώρος ενδιάμεσης μνήμης).
- Η εγκαθίδρυση μιας σύνδεσης γίνεται με αμοιβαία συμφωνία ή χειραψία (handshake) των δύο άκρων (οντοτήτων μεταφοράς των τερματικών συστημάτων), αξιοποιώντας την ανταλλαγή μηνυμάτων ελέγχου (συγχρονισμού).
- Στην περίπτωση αξιόπιστης υπηρεσίας δικτύου, για την εγκαθίδρυση μιας σύνδεσης αρκεί μια χειραψία δύο βημάτων, δηλαδή η μία οντότητα μεταφοράς να στείλει ένα μήνυμα (τμήμα) συγχρονισμού SYN στην άλλη οντότητα μεταφοράς και αυτή με τη σειρά της να επιβεβαιώσει τη σύνδεση επιστρέφοντας ένα μήνυμα συγχρονισμού SYN.
- Ωστόσο, στην περίπτωση μη αξιόπιστης υπηρεσίας δικτύου, τα τμήματα SYN μπορεί να φτάσουν στον προορισμό τους άκαιρα, δηλαδή σε χρόνο που δεν χρειάζονται πλέον και να προκαλέσουν προβλήματα στην επικοινωνία των δύο οντοτήτων μεταφοράς.
- Για παράδειγμα, ένα καθυστερημένο τμήμα SYN που υπάρχει μετά τον τερματισμό μιας σύνδεσης, μπορεί να προκαλέσει τη εγκαθίδρυση μιας ανεπιθύμητης σύνδεσης.
- Έτσι, όταν πρόκειται για μη αξιόπιστη υπηρεσία δικτύου, χρησιμοποιείται η χειραψία τριών φάσεων (ή τριπλή χειραψία, 3-way handshake), για την ασφαλή εγκαθίδρυση μιας σύνδεσης.

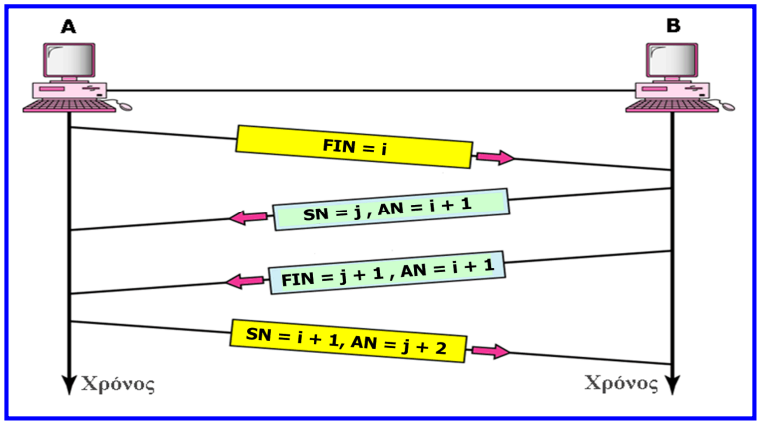
Εγκαθίδρυση και τερματισμός σύνδεσης



Η εγκαθίδρυση σύνδεσης υλοποιείται με την ανταλλαγή 3 τμημάτων (χειραψία 3 φάσεων)

- Η οντότητα μεταφοράς Α που επιθυμεί να εγκαθιδρύσει μία σύνδεση με την οντότητα Β, στέλνει αίτηση σύνδεσης στην οντότητα Β με ένα τμήμα συγχρονισμού SYN με αριθμό ακολουθίας i.
- Η οντότητα Β στέλνει στην Α ένα τμήμα SYN/AN, δηλαδή στέλνει επιβεβαίωση λήψης (AN) με αριθμό επιβεβαίωσης λήψης (i + 1) και στο ίδιο τμήμα επαναλαμβάνει την αίτηση σύνδεσης (SYN) με αριθμό ακολουθίας j.
- Η οντότητα Α επιβεβαιώνει την λήψη της αίτησης σύνδεσης, στέλνοντας επιβεβαίωση λήψης (AN) με αριθμό επιβεβαίωσης λήψης (j + 1), η οποία είναι ενσωματωμένη σε τμήμα δεδομένων της οντότητας Α με αριθμό ακολουθίας SN = i + 1.

Εγκαθίδρυση και τερματισμός σύνδεσης

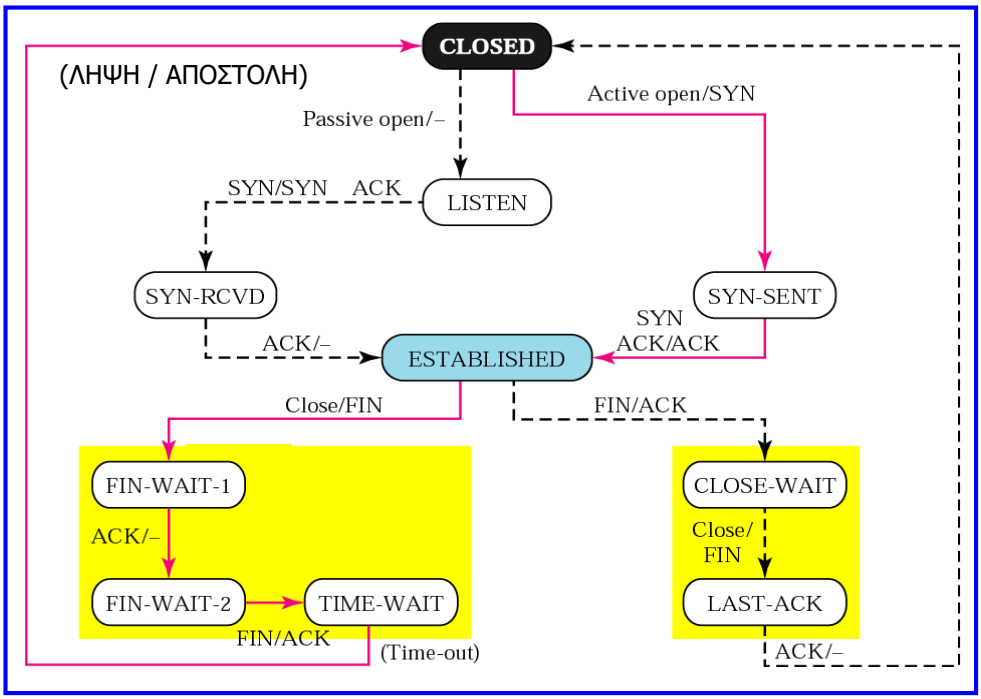


Ο τερματισμός σύνδεσης υλοποιείται με την ανταλλαγή 4 τμημάτων (χειραψία 4 φάσεων)

- Η οντότητα μεταφοράς Α που επιθυμεί να τερματίσει μία σύνδεση με την οντότητα Β, στέλνει στην Β ένα τμήμα τερματισμού FIN με αριθμό ακολουθίας i.
- Η Β στέλνει στην Α τμήμα με αριθμό ακολουθίας j και αριθμό επιβεβαίωσης λήψης (i + 1). Λόγω του ότι στο τμήμα FIN έχει ανατεθεί ο επόμενος αριθμός από αυτόν του τελευταίου byte δεδομένων, η Β μετά τη λήψη του FIN αναμένει για το τελευταίο τμήμα δεδομένων και αφού το λάβει στέλνει τμήμα FIN με αριθμό ακολουθίας (j + 1) και AN = (i + 1).
- Η Α στέλνει στη Β τμήμα με αριθμό ακολουθίας (i + 1) και αριθμό επιβεβαίωσης λήψης (j + 2), αλλά εάν δε λάβει το FIN από τη Β, αναμένει για χρόνο διπλάσιο από το χρόνο ζωής ενός τμήματος και τερματίζει τη σύνδεση.

Εγκαθίδρυση και τερματισμός σύνδεσης

Διάγραμμα καταστάσεων που περιγράφει τις διαδικασίες εγκαθίδρυσης και τερματισμού σύνδεσης στο πρωτόκολλο TCP



Οντότητα TCP A:
- CLOSED
- SYN-SENT
- ESTABLISHED
...δεδομένα...
- FIN-WAIT-1
- FIN-WAIT-2
- TIME-WAIT

Οντότητα TCP B:
- CLOSED
- LISTEN
- SYN-RCVD
- ESTABLISHED
...δεδομένα...
- CLOSE-WAIT
- LAST-ACK

Ανάκτηση από σφάλμα

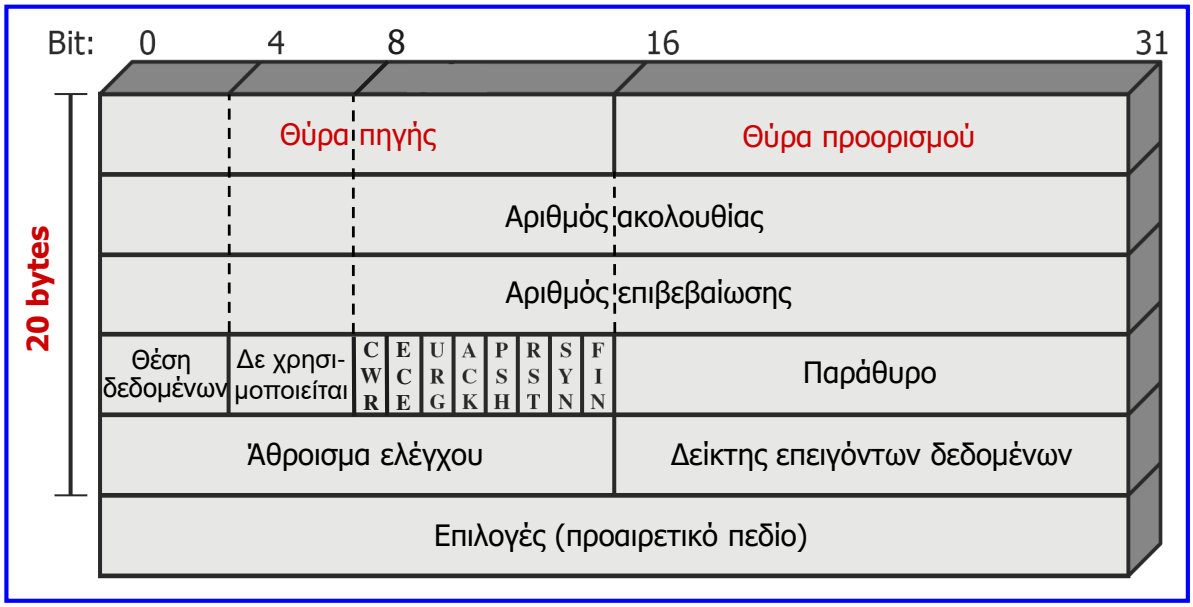
- Όταν το σύστημα στο οποίο εκτελείται μία οντότητα μεταφοράς υποστεί βλάβη και διενεργήσει επανεκκίνηση, χάνεται η πληροφορία κατάστασης όλων των ενεργών συνδέσεων και υπάρχει κίνδυνος μερικά δεδομένα χρηστών να χαθούν.
- Οι επηρεαζόμενες συνδέσεις καθίστανται μερικώς ανοικτές επειδή η πλευρά που δεν απέτυχε δεν έχει ακόμη αντιληφθεί το πρόβλημα.
- Η ενεργή πλευρά μπορεί να τερματίσει τη σύνδεση χρησιμοποιώντας ένα χρονόμετρο που μετρά το χρόνο αναμονής για επιβεβαίωση λήψης ενός τμήματος, το οποίο έχει μεταδοθεί στο μέγιστο αριθμό επαναλήψεων και να υποθέσει ότι η άλλη οντότητα μεταφοράς ή το δίκτυο έχει υποστεί βλάβη.
- Ο τερματισμός των μερικώς ανοικτών συνδέσεων μπορεί να γίνει πιο γρήγορα με τη χρήση ενός τμήματος RST.
- Η αποτυχημένη πλευρά (δηλαδή, αυτή που έχει υποστεί τη βλάβη) επιστρέφει ένα τμήμα RST με αριθμό ακολουθίας i σε κάθε τμήμα με αριθμό ακολουθίας i που λαμβάνει, η άλλη πλευρά ελέγχει τον αριθμό ακολουθίας για να επιβεβαιώσει την εγκυρότητα του τμήματος και τερματίζει τη σύνδεση.

Πρωτόκολλο ελέγχου μετάδοσης (TCP)

- Το πρωτόκολλο TCP (transmission control protocol) είναι βασικό πρωτόκολλο επιπέδου μεταφοράς και μέρος του μοντέλου TCP/IP.
- Το TCP παρέχει υπηρεσία μεταφοράς με σύνδεση (connection-oriented) και είναι ειδικά σχεδιασμένο για να λειτουργεί με το IP.
- Το TCP παρέχει αξιόπιστη επικοινωνία μεταξύ ζευγών χρηστών (διεργασιών εφαρμογών) κατά μήκος μίας σειράς από αξιόπιστα και μη αξιόπιστα δίκτυα.
- Παρέχει επίσης δύο χρήσιμες δυνατότητες περιγραφής των δεδομένων: τα κρίσιμα και τα επείγοντα δεδομένα.
- Κρίσιμα δεδομένα: ο χρήστης (εφαρμογή) πηγής μπορεί να απαιτήσει από το TCP να μεταδώσει άμεσα τα δεδομένα που υποδεικνύονται ως κρίσιμα, με ενεργοποίηση της σημαίας PSH (push) και το TCP πρέπει να παραδώσει άμεσα τα δεδομένα αυτά στο χρήστη (εφαρμογή) προορισμού, χωρίς να αναμείνει για τη λήψη και προσωρινή αποθήκευση πρόσθετων δεδομένων πριν την μετάδοσή τους.
- Επείγοντα δεδομένα: μέσω του TCP ο χρήστης (εφαρμογή) προορισμού ενημερώνεται για την ύπαρξη επειγόντων δεδομένων, που υποδείχθηκαν ως σημαντικά από τον χρήστη (εφαρμογή) πηγής και εξαρτάται από τον χρήστη προορισμού να καθορίσει την κατάλληλη ενέργεια. Τα επείγοντα δεδομένα υποδεικνύονται με ενεργοποίηση της σημαίας URG (urgent) και τοποθετούνται στην αρχή των τμημάτων TCP.

Πρωτόκολλο ελέγχου μετάδοσης (TCP)

Δομή επικεφαλίδας τμήματος TCP (TCP segment header)



- Αριθμός ακολουθίας (32 bits): αριθμός ακολουθίας πρώτου byte δεδομένων του τμήματος με εξαίρεση την περίπτωση όπου η σημαία SYN είναι 1, οπότε το πρώτο byte δεδομένων είναι αυτό με τον επόμενο αριθμό ακολουθίας από αυτόν του πεδίου.

Πρωτόκολλο ελέγχου μετάδοσης (TCP)

- Αριθμός επιβεβαίωσης λήψης (32 bits): αριθμός ακολουθίας του επόμενου byte δεδομένων που η οντότητα TCP περιμένει να λάβει.
- Θέση δεδομένων, offset (4 bits): αριθμός των λέξεων 32 bits της επικεφαλίδας.
- Σημαίες (8 bits): URG (επείγοντα δεδομένα), ACK (επιβεβαίωση λήψης), PSH (άμεση προώθηση τμήματος που περιλαμβάνει κρίσιμα δεδομένα), RST (καθαρισμός σύνδεσης), SYN (έναρξη σύνδεσης), FIN (τερματισμός σύνδεσης), ECE (αίτημα για λειτουργία ελέγχου συμφόρησης), CWR (μείωση παραθύρου αποστολής).
- Παράθυρο (16 bits): πίστωση ελέγχου ροής σε bytes, με πρώτο αριθμό byte της πίστωσης αυτόν που υποδεικνύεται στο πεδίο επιβεβαίωσης λήψης.
- Άθροισμα ελέγχου (16 bits): συμπλήρωμα ως προς 1 του αθροίσματος των λέξεων 16 bits όλου του τμήματος και μιας ψευδοεπικεφαλίδας.
- Δείκτης επειγόντων δεδομένων (16 bits): ισούται με [πλήθος των bytes του τμήματος που χαρακτηρίζονται από τον αποστολέα ως επείγοντα δεδομένα] + 1.
- Επιλογές (μεταβλητό): περιλαμβάνει επιλογές όπως: μέγιστο μέγεθος δεδομένων τμήματος (MSS) και επέκταση παραθύρου με βάση την οποία το πεδίο Παράθυρο της επικεφαλίδας ολισθαίνει προς τα αριστερά όσες θέσεις δηλώνει ο αριθμός 8 bits (με μέγιστη τιμή 14) που περιλαμβάνεται στις Επιλογές, δηλαδή εάν ο αριθμός αυτός (window scale) είναι n_{10}, τότε το μέγεθος παραθύρου επεκτείνεται και είναι ίσο με (Παράθυρο × 2^n), αφού η ολίσθηση προς τα αριστερά κατά n θέσεις ισοδυναμεί με πολλαπλασιασμό με τον αριθμό 2^n.

Πρωτόκολλο ελέγχου μετάδοσης (TCP)

- Το πεδίο αθροίσματος ελέγχου εφαρμόζεται σε ολόκληρο το τμήμα και σε μία ψευδο-επικεφαλίδα (12 bytes) που τίθεται στην αρχή της επικεφαλίδας μόνο για τον υπολογισμό του αθροίσματος ελέγχου και περιλαμβάνει τις 2 διευθύνσεις IP, 1 μηδενικό byte (00_{16}), τον αριθμό του πρωτοκόλλου TCP (06_{16}) και το μέγεθος του τμήματος TCP σε bytes (2 bytes).
- Η ψευδοεπικεφαλίδα προστατεύει το TCP από λάθος παράδοση από το IP, αφού εάν το IP παραδώσει ένα πακέτο σε λάθος τερματικό σύστημα, η λαμβάνουσα οντότητα TCP θα ανιχνεύσει το σφάλμα παράδοσης.
- Όσον αφορά τους μηχανισμούς του πρωτοκόλλου TCP, είναι κατά βάση αυτοί που αναλύθηκαν και αφορούν τις ακόλουθες βασικές διεργασίες που διευθετούνται από ένα πρωτόκολλο μεταφοράς, όταν η υπηρεσία δικτύου είναι μη αξιόπιστη (όπως του IP):
  - ✓ διευθυνσιοδότηση και πολυπλεξία,
  - ✓ διατεταγμένη παράδοση τμημάτων δεδομένων,
  - ✓ έλεγχος ροής,
  - ✓ ανίχνευση αντιγράφων τμημάτων δεδομένων,
  - ✓ έλεγχος σφαλμάτων και αναμετάδοση τμημάτων δεδομένων,
  - ✓ εγκαθίδρυση (αποκατάσταση) και τερματισμός σύνδεσης.
- Ακολουθεί η παρουσίαση των μηχανισμών που αφορούν τον έλεγχο συμφόρησης δικτύου.

Έλεγχος συμφόρησης δικτύου

- Η πιστωτική τεχνική ελέγχου ροής του TCP, χρησιμοποιείται για να επιτρέπει σε ένα τερματικό σύστημα προορισμού να περιορίζει τη ροή των τμημάτων από ένα τερματικό σύστημα πηγής, ώστε να αποφύγει την υπερχείλιση της ενδιάμεσης μνήμης του.
- Ωστόσο, όταν τα τερματικά συστήματα μεταδίδουν περισσότερα δεδομένα από όσα μπορεί να μεταφέρει το δίκτυο (δηλαδή, να προωθήσουν οι δρομολογητές), δημιουργείται συμφόρηση (congestion) δικτύου.
- Στο ξεκίνημα της εκδήλωσης της συμφόρησης, ο χρόνος μεταφοράς δεδομένων κατά μήκος του δικτύου αυξάνεται και καθώς η συμφόρηση επιδεινώνεται, απορρίπτονται τμήματα δεδομένων από τους κόμβους του δικτύου.
- Ο μηχανισμός ελέγχου ροής ανιχνεύει την έναρξη της συμφόρησης και αντιδρά σε αυτή, μειώνοντας τη ροή δεδομένων, αλλά συνήθως η αντίδραση αυτή δεν είναι αρκετή και είναι αναγκαίοι πρόσθετοι μηχανισμοί για τον έλεγχο της συμφόρησης δικτύου.
- Οι τεχνικές ελέγχου συμφόρησης διακρίνονται σε δύο κατηγορίες που βασίζονται στη διαχείριση χρονομέτρου αναμετάδοσης και στη διαχείριση παραθύρου αποστολής.
- Στις τεχνικές διαχείρισης χρονομέτρου αναμετάδοσης, ο χρόνος διάδοσης με επιστροφή (RTT) εκτιμάται ως ο μέσος όρος του RTT που παρατηρείται για πρόσφατα τμήματα και το χρονόμετρο αναμετάδοσης ρυθμίζεται σε μία τιμή λίγο μεγαλύτερη του εκτιμώμενου RTT.
- Με τις τεχνικές διαχείρισης παραθύρου αποστολής παρέχεται η δυνατότητα τροποποίησης του παραθύρου αποστολής όταν συμβαίνει συμφόρηση.

Διαχείριση χρονομέτρου αναμετάδοσης

- Απλός μέσος όρος του RTT: $ARTT\,(K)$ είναι ο εκτιμώμενος μέσος (απλός μέσος όρος) χρόνος διάδοσης με επιστροφή για τα πρώτα K τμήματα και $RTT\,(i)$ είναι ο χρόνος διάδοσης με επιστροφή που παρατηρείται για το i-οστό μεταδιδόμενο τμήμα.

$$ARTT(K+1) = \frac{1}{K+1}\sum_{i=1}^{K+1} RTT(i) \Rightarrow ARTT(K+1) = \frac{K}{K+1}\,ARTT(K) + \frac{1}{K+1}\,RTT(K+1)$$

- Εκθετικός μέσος όρος: δίνει περισσότερο βάρος σε πιο πρόσφατα δείγματα που είναι πιο πιθανό να απεικονίζουν ακριβέστερα τη μελλοντική συμπεριφορά.
- $SRTT\,(K)$: εκθετικός μέσος όρος του RTT (εκτίμηση του RTT):

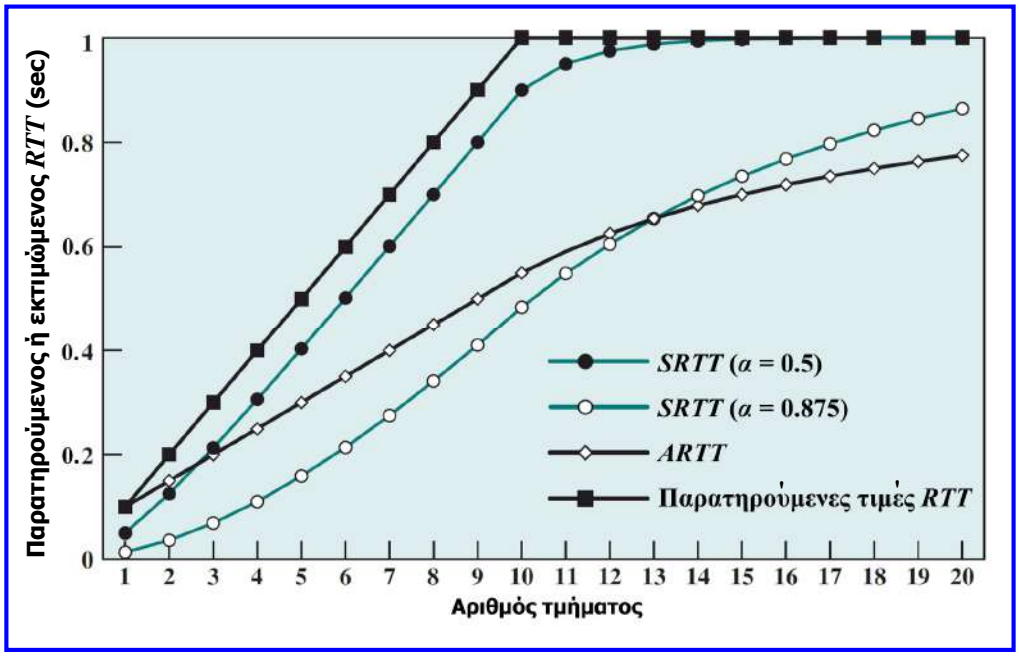
$$SRTT\,(K+1) = \alpha \cdot SRTT\,(K) + (1-\alpha)\cdot RTT\,(K+1) \Rightarrow$$
$$SRTT\,(K+1) = (1-a)\cdot RTT\,(K+1) + \alpha\cdot(1-\alpha)\cdot RTT\,(K)$$
$$+\alpha^2\cdot(1-\alpha)\cdot RTT\,(K-1) + \ldots + \alpha^K\cdot(1-a)\cdot RTT\,(1)$$

$SRRT\,(0) = 0$ και $0 < \alpha < 1$
Όσο μικρότερη είναι η τιμή του α, τόσο μεγαλύτερο βάρος δίνεται σε πιο πρόσφατες παρατηρήσεις

- Ένας τρόπος υπολογισμού της τιμής του χρονομέτρου αναμετάδοσης ή εκπνοής (RTO), είναι η πρόσθεση στον εκτιμώμενο RTT ($ARTT$ ή $SRTT$) μιας σταθεράς χρόνου Δ, ώστε το χρονόμετρο αναμετάδοσης RTO να ρυθμιστεί σε μία τιμή λίγο μεγαλύτερη της:

$$RTO\,(K+1) = [ARTT\,(K+1)\ \text{ή}\ SRTT\,(K+1)] + \Delta$$

Διαχείριση χρονομέτρου αναμετάδοσης



Η προσέγγιση εκθετικού μέσου όρου ($SRTT$) ακολουθεί ακριβέστερα τις αλλαγές των τιμών του παρατηρούμενου RTT σε σχέση με την προσέγγιση απλού μέσου όρου ($ARTT$), ιδιαίτερα όταν πρόκειται για μικρές τιμές της παραμέτρου α.

Διαχείριση χρονομέτρου αναμετάδοσης

- Το μειονέκτημα της προηγούμενης προσέγγισης είναι ότι η σταθερά Δ δεν είναι ανάλογη της εκτίμησης του RTT ($SRTT$), συνεπώς για μεγάλες τιμές του $SRTT$ η σταθερά Δ θα είναι μικρή και θα προκληθούν άσκοπες αναμεταδόσεις, ενώ για μικρές τιμές του $SRTT$ η σταθερά Δ θα είναι μεγάλη προκαλώντας καθυστερήσεις λόγω εκπνοής τμημάτων.
- Πιο αποδοτική προσέγγιση για τον υπολογισμό του RTO, αποτελεί η εκτίμηση της διακύμανσης των τιμών RTT (αλγόριθμος Jacobson), που χρησιμοποιείται στο TCP:

$$\text{Εκτίμηση } RTT: \; SRTT(K+1) = (1-g)\cdot SRTT(K) + g\cdot RTT(K+1)$$

$$\text{Διακύμανση } RTT: \; SDEV(K+1) = (1-h)\cdot SDEV(K) + h\cdot \left| RTT(K+1) - SRTT(K) \right|$$

$$\text{Χρονόμετρο αναμετάδοσης}: RTO(K+1) = SRTT(K+1) + f\cdot SDEV(K+1)$$

$$g = 0.125, \quad h = 0.25, \quad f = 4$$

- Η χρησιμοποίηση της ίδιας τιμής RTO για κάθε αναμετάδοση ενός τμήματος δε λύνει το πρόβλημα της συμφόρησης, συνεπώς μία πιο λογική προσέγγιση είναι η εκθετική υποχώρηση του RTO κατά την οποία η τιμή του RTO αυξάνεται για κάθε αναμετάδοση.
- Στην προσέγγιση αυτή, το RTO πολλαπλασιάζεται με μία σταθερή τιμή για κάθε αναμετάδοση ($RTO = q\cdot RTO$, συνήθως $q = 2$: δυαδική εκθετική υποχώρηση).

Διαχείριση χρονομέτρου αναμετάδοσης

- Ο αλγόριθμος του Karn παρέχει μια μεθοδολογία προσδιορισμού των δειγμάτων του χρόνου διάδοσης με επιστροφή (RTT) που θα πρέπει να χρησιμοποιηθούν ως είσοδοι στον αλγόριθμο Jacobson.
- Όταν ένα τμήμα εκπνέει, αναμεταδίδεται και ακολούθως λαμβάνεται μία επιβεβαίωση λήψης. Τότε υπάρχουν δύο πιθανότητες: η επιβεβαίωση να αφορά την πρώτη μετάδοση του τμήματος ή την δεύτερη μετάδοση (αναμετάδοση).
- Το TCP δεν μπορεί να διακρίνει τις περιπτώσεις και εάν είναι αληθές το δεύτερο ενδεχόμενο, το TCP θα μετρήσει από την πρώτη μετάδοση έως την επιβεβαίωση λήψης με αποτέλεσμα αρκετά αυξημένο μετρούμενο χρόνο RTT.
- Λύση δίνει ο αλγόριθμος Karn με τους ακόλουθους κανόνες:
  - ✓ Δεν χρησιμοποιεί τον μετρούμενο RTT στον αλγόριθμο Jacobson για τμήματα που αναμεταδίδονται.
  - ✓ Όταν γίνεται αναμετάδοση υπολογίζει τον RTO με δυαδική εκθετική υποχώρηση.
  - ✓ Χρησιμοποιεί τον ίδιο τρόπο για τον υπολογισμό του RTO μέχρι να φτάσει μία επιβεβαίωση λήψης για ένα τμήμα που δεν έχει αναμεταδοθεί και τότε χρησιμοποιεί τον μετρούμενο RTT για να ενημερώσει τον αλγόριθμο Jacobson και να υπολογιστούν οι μελλοντικές τιμές του RTO.

Διαχείριση παραθύρου αποστολής

- Εκτός από τεχνικές βελτίωσης αποτελεσματικότητας του χρονομέτρου αναμετάδοσης, υπάρχουν και τεχνικές διαχείρισης παραθύρου αποστολής έναντι στη συμφόρηση.
- Σε υλοποιήσεις TCP ακολουθείται συνδυασμός των τεχνικών αργής εκκίνησης και δυναμικού μεγέθους παραθύρου σε κατάσταση συμφόρησης που προτάθηκαν από τον Jacobson.
- Κατά την αργή εκκίνηση, το TCP χρησιμοποιεί παράθυρο συμφόρησης μετρούμενο σε τμήματα και η μετάδοση περιορίζεται από τη σχέση: awnd = MIN [credit, cwnd].
  - ✓ awnd: μέγεθος επιτρεπόμενου παραθύρου αποστολής σε τμήματα, δηλαδή ο αριθμός τμημάτων που επιτρέπεται να σταλούν χωρίς λήψη επιβεβαίωσης.
  - ✓ cwnd: μέγεθος παραθύρου συμφόρησης, δηλαδή του παραθύρου που χρησιμοποιεί το TCP για να μειώνει τη ροή στις περιόδους συμφόρησης.
  - ✓ credit: αριθμός τμημάτων της αχρησιμοποίητης πίστωσης που έχει παραχωρηθεί με την πιο πρόσφατη επιβεβαίωση λήψης (υπολογίζεται από το πηλίκο του πεδίου W ενός εισερχόμενου τμήματος προς το μέγεθος του τμήματος).
  - ✓ Αρχικά cwnd = 1, δηλαδή το TCP επιτρέπεται να στείλει μόνο ένα τμήμα και να περιμένει για επιβεβαίωση λήψης πριν μεταδώσει επόμενο τμήμα. Κάθε φορά που λαμβάνεται επιβεβαίωση λήψης για ένα τμήμα, το cwnd αυξάνεται κατά 1.
  - ✓ Καθώς φθάνουν οι επιβεβαιώσεις λήψης, το TCP διευρύνει το παράθυρο μέχρι η ροή να ελέγχεται από τις εισερχόμενες επιβεβαιώσεις λήψης και όχι από το cwnd.

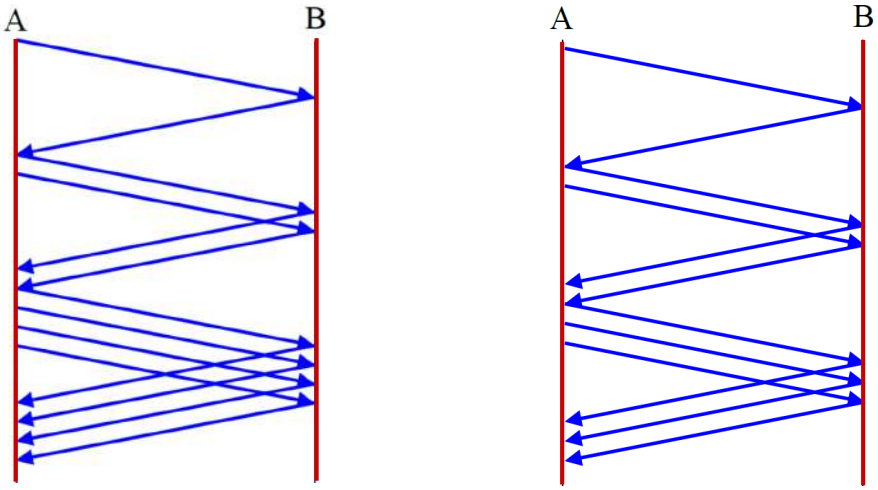
Διαχείριση παραθύρου αποστολής

- Η αργή εκκίνηση λειτουργεί αποδοτικά για την αρχικοποίηση μίας σύνδεσης, αφού επιτρέπει τον γρήγορο καθορισμό ενός λογικού μέγεθος παραθύρου.
- Ωστόσο, η αύξηση του cwnd είναι εκθετική (δηλαδή, ουσιαστικά δεν πρόκειται για αργή εκκίνηση), αφού με την 1η επιβεβαίωση λήψης γίνεται 2 και μπορούν να σταλούν 2 τμήματα και όταν αυτά επιβεβαιωθούν, το TCP αυξάνει το cwnd κατά 1 για κάθε επιβεβαιωμένο τμήμα, επομένως μπορεί να στείλει 4 τμήματα κ.ο.κ., δημιουργώντας κίνδυνο συμφόρησης.
- Αποδοτικότερη προσέγγιση αποτελεί η φάση αργής εκκίνησης στην αρχή της σύνδεσης, ακολουθούμενη από γραμμική αύξηση του cwnd (δηλαδή χρήση δυναμικού μεγέθους παραθύρου για αποφυγή συμφόρησης: φάση αποφυγής συμφόρησης).
- Στην προσέγγιση αυτή, όταν συμβεί μία εκπνοή (δηλαδή όταν χαθεί ένα τμήμα, κάτι που αποτελεί ένδειξη συμφόρησης):
  - ✓ Τίθεται κατώφλι αργής εκκίνησης ίσο με το μισό του τρέχοντος παραθύρου συμφόρησης, δηλαδή: ssthresh = cwnd / 2 (η ελάχιστη τιμή του ssthresh είναι 2).
  - ✓ Αρχικά τίθεται cwnd = 1 και εκτελείται η διαδικασία αργής εκκίνησης μέχρι cwnd = ssthresh. Σε αυτή τη φάση και για όσο ισχύει cwnd < ssthresh, το cwnd αυξάνεται κατά 1 για κάθε επιβεβαίωση λήψης τμήματος που λαμβάνεται.
  - ✓ Για cwnd ≥ ssthresh, το cwnd αυξάνεται κατά ένα για κάθε χρόνο διάδοσης με επιστροφή (*RTT*) και όταν συμβεί εκπνοή το cwnd τίθεται στην αρχική τιμή 1.

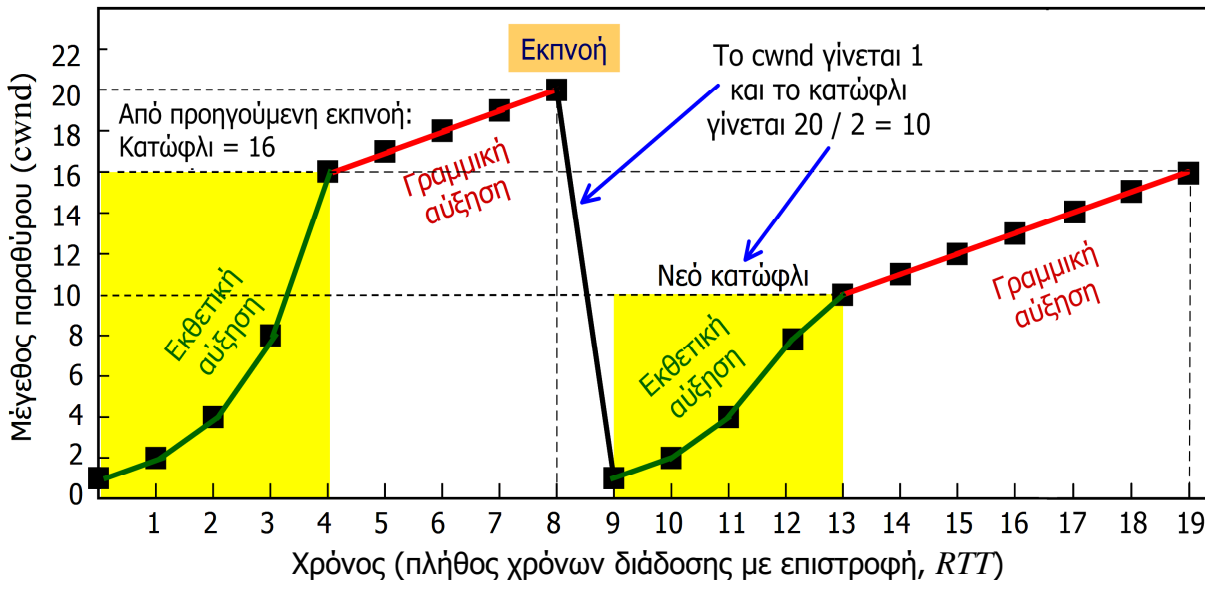
Διαχείριση παραθύρου αποστολής

Φάση αργής εκκίνησης: για κάθε επιβεβαίωση που λαμβάνεται, το παράθυρο αποστολής αυξάνεται κατά ένα

Φάση αποφυγής συμφόρησης: για κάθε παράθυρο αποστολής που λαμβάνεται επιτυχώς (δηλαδή, για κάθε χρόνο διάδοσης με επιστροφή), το παράθυρο αυξάνεται κατά ένα



Παράδειγμα διαχείρισης παραθύρου αποστολής

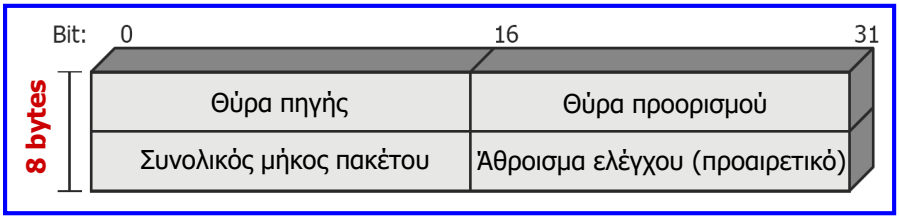


Στο παραπάνω παράδειγμα, συνέπεια της εκπνοής είναι μετά από αυτή να απαιτούνται 11 *RTTs* για ανάκαμψη στην τιμή του cwnd (16), η οποία με αργή εκκίνηση επιτυγχάνεται σε 4 μόνο *RTTs*

Πρωτόκολλο αυτόνομου πακέτου (UDP)

- Το πρωτόκολλο αυτόνομου πακέτου (user datagram protocol, UDP) αποτελεί και αυτό μέρος του μοντέλου TCP/IP και παρέχει υπηρεσία μεταφοράς χωρίς σύνδεση.
- Παρέχει μη αξιόπιστη υπηρεσία, δηλαδή η παράδοση όλων των πακέτων και η προστασία από αντίγραφα δεν είναι εξασφαλισμένες.
- Ουσιαστικά, το UDP προσθέτει μία δυνατότητα διευθυνσιοδότησης θυρών στο IP και υποστηρίζει πολυπλεξία και περιορισμένη προστασία από σφάλματα.
- Ωστόσο, σε εφαρμογές μετάδοσης σύντομων μηνυμάτων ή εφαρμογές όπου απαιτείται κυρίως μετάδοση με υψηλή ταχύτητα, παρά απολύτως αξιόπιστη μετάδοση, το UDP είναι επαρκές και η μη αξιοπιστία του μειώνει την επιβάρυνση του πρωτοκόλλου, αφού δεν απαιτεί αποκατάσταση σύνδεσης και υποστήριξη για διατεταγμένη παράδοση, έλεγχο σφαλμάτων μέσω επαναμεταδόσεων, έλεγχο ροής και συμφόρησης.
- Εφαρμογές: συλλογή δεδομένων από δίκτυα αισθητήρων, εκπομπή δεδομένων σε πολλούς χρήστες για υπηρεσίες δικτύου (π.χ. ανακοίνωση νέου κόμβου), ανταλλαγή σύντομων μηνυμάτων εξυπηρετητή με πολλούς χρήστες.
- Αξιοσημείωτο είναι ότι το πρωτόκολλο μεταφοράς πραγματικού χρόνου (real-time transport protocol, RTP) που χρησιμοποιείται για εφαρμογές πραγματικού χρόνου (μεταφορά φωνής, μετάδοση βίντεο, ψηφιακή τηλεόραση / ραδιόφωνο και άλλες εφαρμογές πολυμέσων), χρησιμοποιεί το πρωτόκολλο μεταφοράς UDP, πολυπλέκοντας διάφορες πηγές πραγματικού χρόνου σε ροή από αυτόνομα πακέτα UDP.

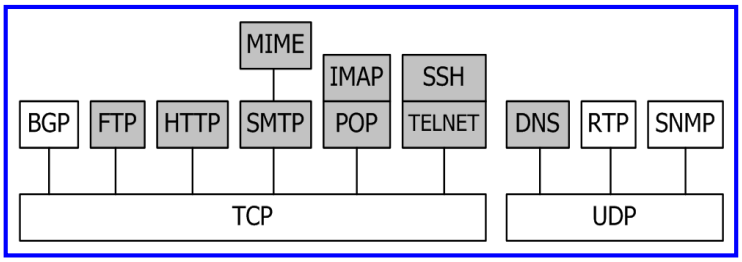
Επικεφαλίδα αυτόνομου πακέτου UDP



- Η επικεφαλίδα αυτόνομου πακέτου UDP (8 bytes) περιλαμβάνει:
  - ✓ θύρα πηγής,
  - ✓ θύρα προορισμού,
  - ✓ συνολικό μήκος αυτόνομου πακέτου (επικεφαλίδα + δεδομένα) που είναι μεταβλητό,
  - ✓ άθροισμα ελέγχου: για τον υπολογισμό του ακολουθείται ο ίδιος αλγόριθμος με εκείνον του αθροίσματος ελέγχου της επικεφαλίδας τμήματος TCP (εάν ανιχνευθεί σφάλμα, το αυτόνομο πακέτο απορρίπτεται χωρίς καμία επιπλέον ενέργεια).
- Η χρήση του αθροίσματος ελέγχου είναι προαιρετική και όταν δε χρησιμοποιείται έχει τιμή 0 και δεν υφίσταται έλεγχος της ορθότητας των δεδομένων του χρήστη ούτε στο επίπεδο μεταφοράς (UDP), ούτε στο επίπεδο δικτύου (IP).

Βασικές έννοιες εφαρμογών διαδικτύου

- Αυτοσκοπός του διαδικτύου είναι η ανταλλαγή δεδομένων και πληροφοριών μεταξύ των χρηστών και η από απόσταση εκτέλεση διάφορων εφαρμογών από τους χρήστες.
- Το επίπεδο εφαρμογής είναι αυτό που βρίσκεται πιο κοντά στον χρήστη.
- Ο χρήστης αποκτά πρόσβαση σε εφαρμογές που του επιτρέπουν να συνδεθεί στο διαδίκτυο, να ανταλλάξει μηνύματα ηλεκτρονικού ταχυδρομείου, να συλλέξει πληροφορίες από τον παγκόσμιο ιστό, να μεταφέρει αρχεία, να συμμετάσχει σε τηλεδιασκέψεις, να διασκεδάσει βλέποντας βίντεο, τηλεόραση και ακούγοντας μουσική στις διάφορες ηλεκτρονικές συσκευές του.
- Για να επιτευχθούν τα παραπάνω, έχουν αναπτυχθεί εφαρμογές, των οποίων η λειτουργία υποστηρίζεται από πρωτόκολλα που συνήθως ανήκουν στο επίπεδο εφαρμογής.



- Βασικές εφαρμογές διαδικτύου:
  - ✓ Εφαρμογή παγκόσμιου ιστού (world wide web, WWW) για την πρόσβαση μέσω διαδικτύου στη συλλογή πληροφοριών του παγκόσμιου ιστού. Υποστηρίζεται κυρίως από το πρωτόκολλο μεταφοράς υπερκειμένου (hypertext transfer protocol, HTTP).

Βασικές έννοιες εφαρμογών διαδικτύου

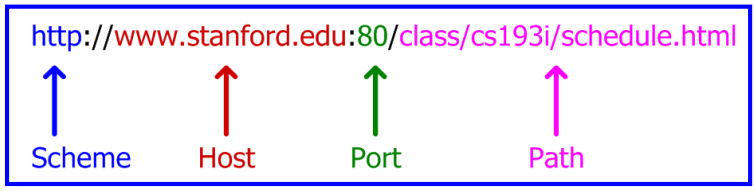
- Βασικές εφαρμογές διαδικτύου (συνέχεια):
  - ✓ Εφαρμογές απομακρυσμένης μεταφοράς αρχείων και απομακρυσμένης σύνδεσης, που υποστηρίζονται από πρωτόκολλα όπως το πρωτόκολλο μεταφοράς αρχείων (file transfer protocol, FTP), το πρωτόκολλο TELNET, το πρωτόκολλο SSH κ.ά.
  - ✓ Ηλεκτρονικό ταχυδρομείο (e-mail) που υποστηρίζεται από πρωτόκολλα όπως το πρωτόκολλο μεταφοράς απλού ταχυδρομείου (simple mail transfer protocol, SMTP), το πρωτόκολλο επέκτασης ηλεκτρονικού ταχυδρομείου πολλαπλών χρήσεων (multipurpose internet mail extension, MIME), τα πρωτόκολλα POP, IMAP κ.ά.
  - ✓ Εφαρμογές ελέγχου διαδικτύου, όπως οι εφαρμογές Traceroute και Ping, η λειτουργία των οποίων βασίζεται στο πρωτόκολλο ICMP (κεφάλαιο 5).
  - ✓ Πολυμεσικές εφαρμογές πραγματικού χρόνου που αφορούν τη μεταφορά πολυμέσων (εικόνες, ήχος και βίντεο) και υποστηρίζονται από σειρά πρωτοκόλλων και προτύπων όπως το πρωτόκολλο μεταφοράς πραγματικού χρόνου (real-time transport protocol, RTP), τα πρότυπα συμπίεσης αρχείων MPEG (Moving Picture Experts Group) κ.ά.
- Σημαντικό θέμα αποτελεί η ασφαλής μεταφορά δεδομένων, η οποία υποστηρίζεται από πρωτόκολλα που υλοποιούν το επίπεδο ασφαλούς καναλιού (secure sockets layer, SSL) ή το επίπεδο ασφαλούς μεταφοράς (transport layer security, TLS), καθώς και πρότυπα κρυπτογράφησης, όπως τα DES (data encryption standard), AES (advanced encryption standard). Τα SSL και TLS λειτουργούν μεταξύ του TCP και του επιπέδου εφαρμογής και χρησιμοποιούν το TCP για να παρέχουν υπηρεσίες ασφάλειας από άκρο σε άκρο.

Παγκόσμιος ιστός

- Το θεμελιώδες πρωτόκολλο για τη λειτουργία του παγκόσμιου ιστού (world wide web, WWW) είναι το πρωτόκολλο μεταφοράς υπερκειμένου (hypertext transfer protocol, HTTP).
- Το HTTP είναι ένα πρωτόκολλο που λειτουργεί σε σχήμα πελάτη (client) - εξυπηρετητή (server) και διαχειρίζεται ανεξάρτητες συναλλαγές δεδομένων, δημιουργώντας μια νέα σύνδεση TCP μεταξύ του πελάτη και του εξυπηρετητή για κάθε συναλλαγή.
- Οι συνδέσεις TCP χρησιμοποιούνται για την αξιοπιστία κατά τη μεταφορά των δεδομένων και τερματίζονται με την ολοκλήρωση των αντίστοιχων συναλλαγών.
- Η πιο τυπική χρήση του HTTP πραγματοποιείται μεταξύ ενός φυλλομετρητή-πελάτη (web client, browser) και ενός εξυπηρετητή διαδικτύου (web server).
- Ο πελάτης (browser, web client), δημιουργεί αίτημα (request) HTTP, το οποίο περιέχει μια εντολή (μέθοδο), έναν εντοπιστή ή αναγνωριστή διεύθυνσης που ονομάζεται URL (uniform resource locator) και τις παραμέτρους του αιτήματος, πληροφορίες για τον πελάτη και ορισμένες επιπρόσθετες πληροφορίες για το περιεχόμενο.
- Όταν ο εξυπηρετητής (web server) λάβει το αίτημα, εκτελεί την απαιτούμενη ενέργεια και επιστρέφει μια απάντηση (response) HTTP, η οποία περιλαμβάνει πληροφορίες κατάστασης, έναν κώδικα επιτυχίας ή σφάλματος και πληροφορίες για τον εξυπηρετητή, για την απάντηση και πιθανώς περιεχόμενο.

Πρωτόκολλο DNS

- Οι URL χρησιμοποιούνται ως πιο φιλικοί για το χρήστη σε σχέση με τις διευθύνσεις IP. Για παράδειγμα, αντί για τη διεύθυνση IP 193.108.160.219 ένα αίτημα HTTP περιλαμβάνει τον εντοπιστή ή αναγνωριστή διεύθυνσης URL www.eap.gr.
- Η αντιστοίχιση (μετάφραση) των URLs σε διευθύνσεις IP διενεργείται από την υπηρεσία που παρέχεται από το πρωτόκολλο DNS (domain name system).



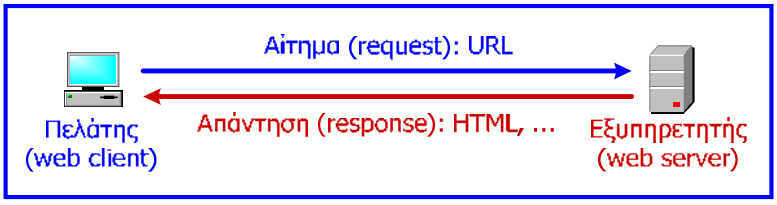
- Scheme: καθορίζει το πρωτόκολλο που θα πρέπει να χρησιμοποιηθεί για την επικοινωνία μεταξύ πελάτη και εξυπηρετητή.
- Host: προσδιορίζει τη διεύθυνση του συστήματος (πηγή) που κατέχει την πληροφορία.
- Port: προσδιορίζει τον αριθμό θύρας TCP του εξυπηρετητή, αλλά συνήθως παραβλέπεται αφού οι αριθμοί θύρας για τα πρωτόκολλα που χρησιμοποιούνται είναι γνωστοί (well-known), π.χ. HTTP: 80, HTTPS: 443.
- Path: προσδιορίζει τη θέση στον Host όπου βρίσκεται η πληροφορία στην οποία αιτείται πρόσβαση ο πελάτης (web client).

Πρωτόκολλο DNS

- Το πρωτόκολλο DNS ανήκει στο επίπεδο εφαρμογής του μοντέλου αναφοράς TCP/IP και έχει σχεδιαστεί για να χρησιμοποιεί δύο πρωτόκολλα μεταφοράς, το TCP και το UDP.
- Ωστόσο, έχει προκαθοριστεί (default) να χρησιμοποιεί το UDP (θύρα 53) για την εξυπηρέτηση των συναλλαγών του, λόγω κυρίως της μεγαλύτερης ταχύτητας που παρέχει, αφού το UDP δεν εγκαθιδρύει συνδέσεις και δεν εφαρμόζει τεχνικές που απαιτούνται για την αξιοπιστία της επικοινωνίας.
- Αυτό έχει ως αποτέλεσμα οι εξυπηρετητές DNS να μην διατηρούν συνδέσεις και τα αιτήματα DNS (αιτήματα για μετάφραση URL σε διευθύνσεις IP) που έχουν συνήθως μικρό μέγεθος να είναι αποδοτικότερο να αποστέλλονται σε αυτόνομα πακέτα UDP.
- Ωστόσο, το πρωτόκολλο UDP δεν είναι αξιόπιστο και για το λόγο αυτό η αξιοπιστία αναπληρώνεται σε κάποιο βαθμό στο επίπεδο εφαρμογής, με χρήση χρονομέτρων αναμετάδοσης και επαναμεταδόσεων πακέτων.
- Μια συναλλαγή DNS αποτελείται από ένα αυτόνομο πακέτο UDP (αίτημα) του πελάτη (client), το οποίο ακολουθείται από ένα αυτόνομο πακέτο UDP (απάντηση) του εξυπηρετητή (DNS server).
- Η εφαρμογή nslookup (command-line tool) αποτελεί τυπική επιλογή για την εύρεση της διεύθυνσης IP που αντιστοιχεί σε έναν URL.

```
C:\WINDOWS\system32> nslookup : www.eap.gr
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  193.108.160.219
```


Πρωτόκολλο HTTP



- Ο χρήστης με το φυλλομετρητή του, δηλαδή ο πελάτης, δημιουργεί μια άμεση σύνδεση TCP από σημείο σε σημείο με έναν εξυπηρετητή πηγής.
- Ο πελάτης στέλνει το αίτημα και ο εξυπηρετητής πηγής επιστρέφει την απάντηση.
- Στον εξυπηρετητή πηγής είναι αποθηκευμένη η αρχική ιστοσελίδα που αιτήθηκε ο χρήστης.
- Μετά λοιπόν από τη δημιουργία μιας συνεδρίας στο HTTP από ένα χρήστη με το φυλλομετρητή του, ο χρήστης ανακτά μια σειρά από ιστοσελίδες.
- Η ανάκτηση των ιστοσελίδων γίνεται σε μικρό χρονικό διάστημα, ενώ οι ιστοσελίδες αυτές μπορεί να βρίσκονται σε διαφορετικούς και γεωγραφικά κατανεμημένους εξυπηρετητές.
- Ο πελάτης και ο εξυπηρετητής χρησιμοποιούν προσωρινή μνήμη (cache) όπου αποθηκεύονται τα μηνύματα (αιτήματα και απαντήσεις).
- Η προσωρινή μνήμη αποθηκεύει τα μηνύματα και όταν για παράδειγμα λάβει νέο αίτημα, ίδιο με προηγούμενο, μπορεί να προσφέρει την αντίστοιχη αποθηκευμένη απάντηση, αντί να προσπελάσει την πληροφορία που δηλώνεται στον URL.

Πρωτόκολλο HTTP

Περιεχόμενα μηνύματος HTTP

Το μήνυμα μπορεί να είναι αίτημα (αίτηση, request) ή απάντηση (απόκριση, response)

Γραμμή αιτήματος ή Γραμμή κατάστασης απάντησης
Γενική επικεφαλίδα
Επικεφαλίδα αιτήματος ή Επικεφαλίδα απάντησης
Επικεφαλίδα οντότητας
Σώμα οντότητας

- Η γραμμή αιτήματος: καθορίζει τη μέθοδο του μηνύματος, το path του URL όπου βρίσκεται η ζητούμενη πληροφορία και την έκδοση (version) του πρωτοκόλλου HTTP.
- Η μέθοδος δηλώνει την εντολή του μηνύματος και οι μέθοδοι που χρησιμοποιούνται συχνά είναι η GET (ανάκτηση πληροφοριών από τον URL του μηνύματος) και η POST (για να γίνει αποδεκτή η οντότητα του αιτήματος ως νέα οντότητα στον URL του αιτήματος).
- METHOD Request-path HTTP-VersionCRLF
- Οι χαρακτήρες CR και LF, είναι οι χαρακτήρες ASCII carriage return (επιστροφή φορέα) που σηματοδοτεί την επιστροφή του σημείου εισαγωγής κειμένου (φορέα) στην αρχή της γραμμής και line feed (τροφοδοσία γραμμής) που σηματοδοτεί τη μετακίνηση στη επόμενη γραμμή. Όταν αναφέρονται μαζί σηματοδοτούν το τέλος μιας γραμμής. Ισοδύναμη γραφή του <CR><LF> είναι η \r\n.
- Παράδειγμα: GET /class/cs193i/schedule.html HTTP/1.1\r\n\

Πρωτόκολλο HTTP

- Η επικεφαλίδα αιτήματος περιλαμβάνει:
  - ✓ τη διεύθυνση της πηγής (host) από την οποία ζητείται η απάντηση,

    Παράδειγμα: Host: www.stanford.edu
  - ✓ πληροφορίες για το αίτημα και τον πελάτη που καθορίζουν παραμέτρους για την αποδεκτή απάντηση (τύπος απάντησης, γλώσσα, κωδικοποίηση μεταφοράς), παραμέτρους που καθορίζουν περιορισμούς στον εξυπηρετητή που θα εκτελέσει το αίτημα (θέτουν μια συνθήκη στη μέθοδο), παραμέτρους ασφαλείας κ.ά.

    Παραδείγματα:

    Date: Fri, 06 Nov 2020 08:49:37 GMT

    Accept: text/plain; q=0.5, text/html, text/x-dvi; q=0.8, text/x-c

    Accept-Charset: iso-8859-5, unicode-1-1; q=0.8

    Accept-Encoding: compress, gzip

    Accept-Language: en-gb, de

    If-Modified-Since: Sat 24 Apr 2021 1:00:00 GMT

    Authorization: BASIC Z3Vlc3Q6Z3Vlc3QxMjM (username:password κωδικοποιημένα σε base64).

Πρωτόκολλο HTTP

- Παράδειγμα αιτήματος με γραμμή αιτήματος και βασική επικεφαλίδα:

```
GET /class/cs193i/schedule.html HTTP/1.1\r\n\
Host: www.stanford.edu\r\n\r\n\
```

- Οι δύο ακολουθίες CRLF στο τέλος της δεύτερης γραμμής δηλώνουν το τέλος της γραμμής (η πρώτη) και μια κενή γραμμή (η δεύτερη), έτσι ώστε να ενημερώνεται ο εξυπηρετητής ότι έχει ολοκληρωθεί το μήνυμα του αιτήματος.
- Η γραμμή κατάστασης απάντησης περιλαμβάνει πληροφορίες για την κατάσταση μιας απάντησης, όπως το χρησιμοποιούμενο πρωτόκολλο, ακολουθούμενο από έναν κωδικό κατάστασης και την σχετιζόμενη με αυτόν φράση.
- HTTP-Version Status-Code Reason-PhraseCRLF
- Ο κωδικός κατάστασης αποτελείται από 3 δεκαδικά ψηφία, από τα οποία το πρώτο προσδιορίζει την κατηγορία κατάστασης.
- Για παράδειγμα, όταν το 1ο ψηφίο είναι 2 σημαίνει ότι το αίτημα έγινε αποδεκτό, όταν το 1ο ψηφίο είναι 4 σημαίνει ότι το αίτημα περιέχει συντακτικό σφάλμα ή ότι το αίτημα δεν μπορεί να διευθετηθεί, ενώ όταν το 1ο ψηφίο είναι 3 σημαίνει ότι το αίτημα δεν ικανοποιήθηκε και ο πελάτης ανακατευθύνεται ώστε να προβεί σε πρόσθετες ενέργειες.
- Τα υπόλοιπα 2 ψηφία του κωδικού κατάστασης εξειδικεύουν την κατάσταση.

Πρωτόκολλο HTTP

- Για παράδειγμα, ο κωδικός 200 με σχετιζόμενη έκφραση OK σημαίνει ότι το αίτημα ήταν επιτυχές και ότι συμπεριλαμβάνονται στην απάντηση τα αιτούμενα στοιχεία, ο κωδικός 400 με έκφραση Bad Request σημαίνει λανθασμένη σύνταξη αιτήματος, ενώ ο κωδικός 304 με έκφραση Not Modified σημαίνει ότι τα αιτούμενα στοιχεία δεν έχουν τροποποιηθεί μετά την ημερομηνία που καθορίζει ο πελάτης (δηλαδή, ο πελάτης κατευθύνεται στη χρησιμοποίηση της αποθηκευμένης έκδοσης των στοιχείων αυτών).
- Παράδειγμα γραμμής κατάστασης απάντησης: `HTTP/1.1 200 OK\r\n\`
- Η επικεφαλίδα απάντησης περιλαμβάνει πληροφορίες για την απάντηση και τον εξυπηρετητή, οι οποίες δεν περιέχονται στη γραμμή απάντησης, όπως: ακριβή θέση (location) της πληροφορίας που ζητείται στο αίτημα, η οποία χρησιμοποιείται ώστε να ανακατευθύνει τον λήπτη σε μια θέση διαφορετική από εκείνη που ορίζεται στο path του αιτήματος, το λογισμικό που χρησιμοποιείται από τον εξυπηρετητή, δήλωση εκτίμησης χρόνου για την διεκπεραίωση του αιτήματος (σε sec), δήλωση χρόνου για τον οποίο η υπηρεσία είναι μη διαθέσιμη (σε sec ή έως συγκεκριμένη ημερομηνία και ώρα), δήλωση αποδεχόμενης μονάδας μέτρησης μεγέθους των πληροφοριών που ζητούνται κ.ά.

  Παραδείγματα: Location: http://www.stanford.edu/http/cs193i/schedule.html

  Server: Apache/2.2.14 (Win32), Age: 1030

  Retry-After: Wed, 30 Jun 2021 23:59:59 GMT, Retry-After: 120,

  Accept-Ranges: bytes

Πρωτόκολλο HTTP

- Η επικεφαλίδα οντότητας περιέχει πληροφορίες για την πηγή που έχει καθοριστεί στο αίτημα, όπως τις μεθόδους που υποστηρίζονται για πρόσβαση στο path που βρίσκεται η πληροφορία που ζητήθηκε και πληροφορίες για το σώμα της οντότητας όπως τύπος, κωδικοποίηση, μέγεθος, γλώσσα, χρόνος ζωής, το χρόνο τελευταίας τροποποίησης του σώματος της οντότητας κ.ά.

  Παραδείγματα:

  Allow: GET

  Content-Encoding: gzip

  Content-Language: de, en

  Content-Length: 3495 (σε bytes)

  Content-Type: text/html ή image/jpeg κ.ά.

  Expires: Wed, 01 Dec 2021 16:00:00 GMT

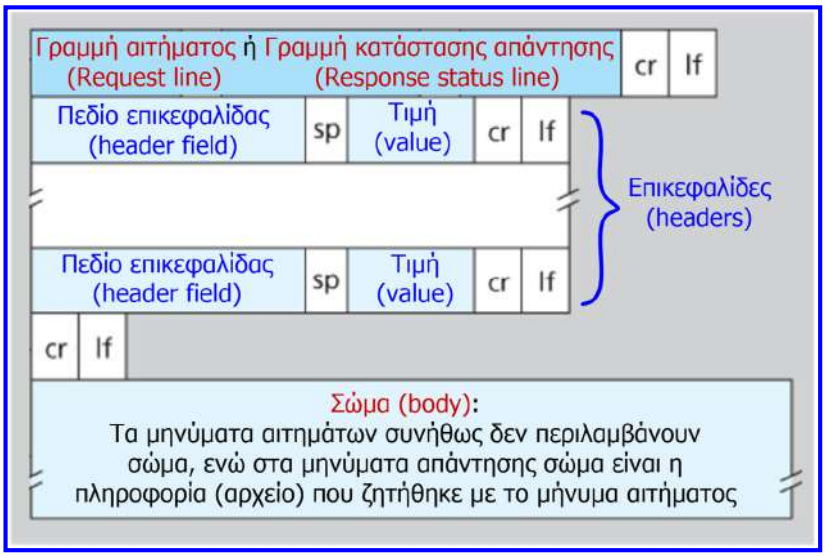
  Last-Modified: Mon, 16 Nov 2020 12:45:26 GMT

- Το σώμα οντότητας περιλαμβάνει μια ακολουθία από bytes που αποτελούν το περιεχόμενο της πληροφορίας (αρχείου) που ζητήθηκε. Το HTTP μπορεί να μεταφέρει διάφορους τύπους περιεχομένου, όπως κείμενο, δυαδικά δεδομένα, ήχο, εικόνα, βίντεο και η ερμηνεία των bytes περιεχομένου καθορίζεται από πεδία της επικεφαλίδας οντότητας.

Πρωτόκολλο HTTP

- Τέλος, η γενική επικεφαλίδα ενός μηνύματος περιέχει πεδία που ισχύουν σε μηνύματα αιτήματος και απάντησης, όπως η ημερομηνία δημιουργίας του μηνύματος, ο χρόνος ζωής του (δηλαδή ο χρόνος που ο πελάτης θα κρατήσει ανοικτή τη σύνδεση), η έκδοση MIME με την οποία συμμορφώνεται το μήνυμα, πεδίο που επισυνάπτεται από ενδιάμεσα συστήματα (proxy, gateways) στην αλυσίδα αίτησης απάντησης όπου δηλώνεται ο URL του ενδιάμεσου συστήματος κ.ά.

Δομή μηνύματος HTTP



Πρωτόκολλο HTTP

- Μια τυπική γλώσσα για τη δημιουργία ιστοσελίδων (web pages) είναι η HTML (hypertext markup language).
- Η HTML περιγράφει τη δομή μιας ιστοσελίδας και στην ουσία υποδεικνύει στον φυλλομετρητή τον τρόπο παρουσίασής της.
- Τα διάφορα στοιχεία (elements) της HTML, όπως επικεφαλίδες, παράγραφοι, πίνακες κ.ά. παριστάνονται με μια σειρά χαρακτήρων κειμένου (tags).
- Οι φυλλομετρητές δεν απεικονίζουν τα tags, αλλά τα χρησιμοποιούν για τη διαμόρφωση του περιεχομένου της ιστοσελίδας.

Παράδειγμα αρχείου HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```


Πρωτόκολλο HTTP

- Λόγω του ότι το HTTP είναι πρωτόκολλο που χρησιμοποιεί χαρακτήρες ASCII, είναι εφικτή η απομακρυσμένη πρόσβαση σε έναν εξυπηρετητή (web server) μέσω της εφαρμογής TELNET.
- Μέσω της εφαρμογής αυτής, είναι δυνατή η δημιουργία μιας σύνδεσης TCP με τη θύρα 80 του εξυπηρετητή (web server) που χρησιμοποιείται από το πρωτόκολλο HTTP.

Παράδειγμα δομής μηνυμάτων του πρωτοκόλλου HTTP

```
telnet cslibrary.stanford.edu 80
Trying 171.64.64.168...
Connected to cslibrary.stanford.edu (171.64.64.168).
Escape character is '^]'.
GET /test.html HTTP/1.1
Host: cslibrary.stanford.edu
                                  ← Κενή γραμμή μεταξύ των μηνυμάτων αιτήματος και απάντησης
HTTP/1.1 200 OK
Date: Sun, 07 Mar 2021 17:59:42 GMT
Server: Apache/1.3.26 (Unix)
Last-Modified: Thu, 25 Feb 2021 00:50:34 GMT
Accept-Ranges: bytes
Content-Length: 459
Connection: close
Content-Type: text/html
                                  ← Κενή γραμμή μεταξύ του σώματος και του υπόλοιπου μηνύματος απάντησης
<!doctype html>
<html>
<head>
<title>Test</title>
<meta http-equiv="nick-mode" content="high">
</head>
<body bgcolor="#FFFFFF">
<h1>
Test</h1>
<p>Just a little test doc.
</body>
</html>
```

Απομακρυσμένη μεταφορά αρχείων (FTP)

- Το πρωτόκολλο μεταφοράς αρχείων (file transfer protocol, FTP) χρησιμοποιείται ευρέως για την απομακρυσμένη μεταφορά αρχείων.
- Το σύστημα του πελάτη (client) που φιλοξενεί την εφαρμογή FTP μόλις συνδεθεί με τον εξυπηρετητή (server) μπορεί μέσω του πρωτοκόλλου να εκτελέσει μεταφορά αρχείων από και προς τον εξυπηρετητή κι άλλες εντολές (μετονομασία, διαγραφή αρχείων κ.ά.).
- Η μεταφορά ενός αρχείου γίνεται μέσω μιας σύνδεσης TCP δεδομένων μεταξύ μιας θύρας του πελάτη και της θύρας 21 του εξυπηρετητή.
- Πριν τη μεταφορά αρχείου προηγείται η εξακρίβωση της ταυτότητας (authentication) του πελάτη μέσω ονόματος χρήστη και κωδικού πρόσβασης.

```
ftp> help
Commands may be abbreviated.  Commands are:

!               delete          literal         prompt          send
?               debug           ls              put             status
append          dir             mdelete         pwd             trace
ascii           disconnect      mdir            quit            type
bell            get             mget            quote           user
binary          glob            mkdir           recv            verbose
bye             hash            mls             remotehelp
cd              help            mput            rename
close           lcd             open            rmdir
```

```
E:\WINNT\System32\cmd.exe - ftp gaia.cc.gatech.edu

E:\tmp>ftp gaia.cc.gatech.edu
Connected to gaia.CC.GATECH.edu.
220 gaia FTP server (SunOS 4.1) ready.
User (gaia.CC.GATECH.edu:(none)): mjd
331 Password required for mjd.
Password:
230 User mjd logged in.
ftp> get bob.c
200 PORT command successful.
150 ASCII data connection for bob.c (129.62.148.15,1282) (3 bytes).
226 ASCII Transfer complete.
4 bytes received in 0.00 seconds (4000.00 Kbytes/sec)
ftp> _
```


Απομακρυσμένη μεταφορά αρχείων (FTP)

- Λόγω του ότι στο πρωτόκολλο FTP, η σύνδεση TCP χρησιμοποιείται για την αποστολή ή λήψη των δεδομένων, το πρωτόκολλο χρησιμοποιεί μια ακόμη (παράλληλη) σύνδεση ελέγχου, για την αποστολή πληροφοριών ελέγχου από τον πελάτη στον εξυπηρετητή, όπως το όνομα χρήστη, ο κωδικός χρήστη και οι εντολές.
- Όταν ο εξυπηρετητής λαμβάνει από τη σύνδεση ελέγχου (η οποία εκκινείται από τον πελάτη) μία εντολή μεταφοράς αρχείου (από ή προς τον εξυπηρετητή), εκκινεί μια σύνδεση δεδομένων TCP προς τον πελάτη.
- Μετά τη μεταφορά του αρχείου, ο εξυπηρετητής τερματίζει τη σύνδεση δεδομένων, ενώ αν κατά τη διάρκεια της μεταφοράς του αρχείου, υπάρξει αίτημα για μεταφορά κι άλλου αρχείου, δημιουργείται νέα σύνδεση δεδομένων TCP.
- Η σύνδεση ελέγχου παραμένει ανοικτή κατά τη διάρκεια της συνόδου FTP, ενώ μια νέα σύνδεση δεδομένων δημιουργείται για κάθε μεταφορά αρχείου στην ίδια σύνοδο.
- Κάθε εντολή FTP, που στέλνεται μέσω της σύνδεσης ελέγχου και αντιστοιχεί σε μία εντολή του χρήστη, αποτελείται από 4 χαρακτήρες ASCII και συχνά περιλαμβανει όρισμα.
- Οι πιο κοινές εντολές είναι: USER username, PASS password, LIST (για αποστολή λίστας αρχείων στον τρέχοντα φάκελο), RETR filename (get), STOR filename (put).
- Κάθε εντολή ακολουθείται από απάντηση του εξυπηρετητή στον πελάτη. Οι απαντήσεις περιλαμβάνουν 3 ψηφία που συνοδεύονται από προαιρετικό μήνυμα, π.χ. 331 Password required for <username>, 125 Data connection already open; transfer starting.

Σύνδεση σε απομακρυσμένο εξυπηρετητή

- Το πρωτόκολλο TELNET επιτρέπει στο χρήστη να συνδεθεί σε έναν απομακρυσμένο εξυπηρετητή ή τερματικό σύστημα, χρησιμοποιώντας την ομώνυμη εφαρμογή που εκτελείται από τη γραμμή εντολών (command-line) του λειτουργικού συστήματος και ελέγχει το απομακρυσμένο σύστημα με εντολές κειμένου.
- Αποτελεί υπηρεσία εικονικού τερματικού (virtual terminal service) και ο χρήστης που συνδέεται μέσω της υπηρεσίας αυτής με κάποιον απομακρυσμένο εξυπηρετητή μπορεί να χρησιμοποιεί τους πόρους του (στο βαθμό που του επιτρέπεται από το διαχειριστή), με τον ίδιο τρόπο που θα τους χρησιμοποιούσε από ένα τοπικό τερματικό του εξυπηρετητή.
- Ενώ λοιπόν με τα πρωτόκολλα HTTP και FTP ο χρήστης μπορεί να αιτηθεί συγκεκριμένα αρχεία από απομακρυσμένους υπολογιστές, με το TELNET ο χρήστης μπορεί να έχει πρόσβαση στον απομακρυσμένο υπολογιστή για δεδομένα και εφαρμογές.
- Η επικοινωνία με τον απομακρυσμένο υπολογιστή, γίνεται μέσω μιας σύνδεσης TCP από μια θύρα του συστήματος χρήστη προς την θύρα 23 του απομακρυσμένου εξυπηρετητή.
- Παράδειγμα: telnet gaia.cc.gatech.edu, η απάντηση στο αίτημα είναι μια πρόσκληση προς το χρήστη για είσοδο στο απομακρυσμένο σύστημα (με όνομα και κωδικό χρήστη) και μετά τη σχετική έγκριση, αποδίδεται πρόσβαση στο απομακρυσμένο σύστημα.
- Λόγω του ότι το TELNET δεν κρυπτογραφεί τα δεδομένα που στέλνει στην απομακρυσμένη σύνδεση, προτιμάται η χρήση του πρωτοκόλλου Secure Shell, SSH (που χρησιμοποιεί την θύρα 22) και παρέχει την ίδια λειτουργικότητα, αλλά είναι πιο ασφαλές και πιο αξιόπιστο.

Ηλεκτρονικό ταχυδρομείο

- Στην εφαρμογή ηλεκτρονικού ταχυδρομείου που χρησιμοποιείται σε πολύ μεγάλο βαθμό, η μεταφορά των μηνυμάτων βασίζεται στο πρωτόκολλο μεταφοράς απλού ταχυδρομείου (simple mail transfer protocol, SMTP).
- Η απαίτηση για αποστολή μηνυμάτων που περιλαμβάνουν εκτός από απλό κείμενο και τύπους πολυμέσων, όπως φωνή, εικόνα, βίντεο, οδήγησε στην ανάπτυξη ενός νέου προτύπου ηλεκτρονικού ταχυδρομείου (που βασίζεται στο TCP) και αναφέρεται ως επέκταση ηλεκτρονικού ταχυδρομείου πολλαπλών χρήσεων (multipurpose internet mail extension, MIME).
- Ένα μήνυμα ηλεκτρονικού ταχυδρομείου δημιουργείται από ένα λογισμικό ηλεκτρονικής αλληλογραφίας (MS Outlook, Windows Mail, Mozilla ThunderBird, Eudora κ.ά.), στο οποίο ο χρήστης συντάσσει το μήνυμα που θέλει να στείλει.
- Κάθε μήνυμα που δημιουργείται έχει μια επικεφαλίδα που περιλαμβάνει τις διευθύνσεις των παραληπτών, άλλες πληροφορίες και το κυρίως σώμα που περιλαμβάνει το μήνυμα.
- Τα μηνύματα τοποθετούνται στην ουρά του αποστολέα SMTP ως μηνύματα εισόδου.
- Ο αποστολέας SMTP είναι ένα λογισμικό που λειτουργεί στον εξυπηρετητή ηλεκτρονικού ταχυδρομείου (mail server), ο οποίος είναι κόμβος στο διαδίκτυο.
- Κάθε μήνυμα της ουράς έχει 2 μέρη: κυρίως σώμα με επικεφαλίδα και λίστα διευθύνσεων παραληπτών που μετασχηματίζονται σε μορφή κατάλληλη για τον αποστολέα SMTP.

Πρωτόκολλο SMTP

- Ο αποστολέας SMTP λαμβάνει τα μηνύματα από την ουρά εξερχομένων και τα μεταδίδει προς τις κατάλληλες διευθύνσεις.
- Η μεταφορά μηνυμάτων γίνεται από κόμβο σε κόμβο μέσω SMTP συναλλαγών, οι οποίες βασίζονται σε συνδέσεις TCP.
- Συνήθως, η θύρα (port) των κόμβων που χρησιμοποιείται για τις SMTP συναλλαγές είναι η 25.
- Ένας κόμβος μπορεί να έχει πολλαπλούς αποστολείς SMTP που να λειτουργούν ταυτόχρονα, αν η ουρά των εξερχομένων είναι μεγάλη, και πολλαπλούς παραλήπτες SMTP, εάν ο ρυθμός εισερχομένων μηνυμάτων είναι μεγάλος.
- Αν ένα μήνυμα στέλνεται σε πολλαπλούς χρήστες στον ίδιο κόμβο, ο αποστολέας SMTP στέλνει το μήνυμα μόνο μια φορά.
- Αν πολλαπλά μηνύματα πρόκειται να σταλούν στον ίδιο κόμβο, ο αποστολέας SMTP δημιουργεί μια σύνδεση TCP, για να μεταφέρει όλα τα μηνύματα και μετά τερματίζει τη σύνδεση, αντί να ανοίγει και να τερματίζει μια σύνδεση για κάθε μήνυμα.
- Στις πιο πολλές περιπτώσεις, τα μηνύματα μεταφέρονται άμεσα από το σύστημα του αποστολέα στο σύστημα του παραλήπτη μέσω μίας και μοναδικής σύνδεσης TCP.
- Υπάρχουν και περιπτώσεις στις οποίες ένα μήνυμα θα προωθηθεί μέσω πολλαπλών αποστολέων SMTP και τότε το μήνυμα διατρέχει πολλαπλές συνδέσεις TCP.

Πρωτόκολλα POP και IMAP

- Ο SMTP παραλήπτης, που είναι ικανός να επαληθεύσει τους τοπικούς προορισμούς του ταχυδρομείου, δέχεται κάθε μήνυμα που φτάνει και το τοποθετεί στο «κιβώτιο» αλληλογραφίας του χρήστη (mail box) το οποίο επίσης βρίσκεται στο εξυπηρετητή ηλεκτρονικού ταχυδρομείου (mail server).
- Για τη μεταφορά των μηνυμάτων μεταξύ αποστολέα SMTP και παραλήπτη SMTP μέσω συνδέσεων TCP χρησιμοποιείται το πρωτόκολλο SMTP.
- Για τη λήψη των μηνυμάτων (από τον εξυπηρετητή) στο τερματικό σύστημα του χρήστη χρησιμοποιούνται πρωτόκολλα, όπως το POP (post office protocol) και το IMAP (Internet message access protocol).
- Τα πρωτόκολλα POP και IMAP χρησιμοποιούν τις θύρες 110 και 143, αντίστοιχα, για να εγκαθιδρύσουν μία σύνδεση TCP με τον εξυπηρετητή, ώστε να ληφθούν τα μηνύματα του χρήστη.
- Όταν τα πρωτόκολλα POP και IMAP χρησιμοποιούνται με SSL (secure socket layer), χρησιμοποιούν τις θύρες 995 και 993, αντίστοιχα.
- Μετά τη λήψη τους στο τερματικό σύστημα του χρήστη, τα μηνύματα μπορεί να παραμένουν ή να διαγράφονται από τον εξυπηρετητή.

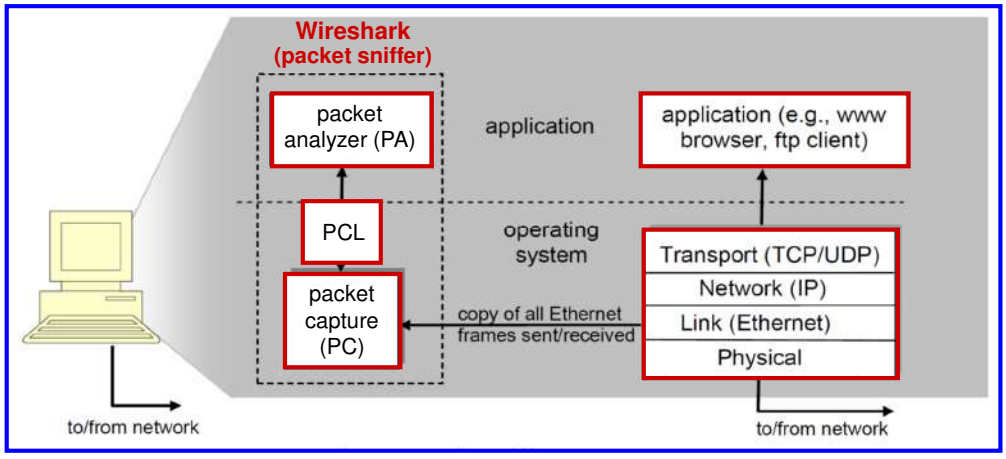
Πρωτόκολλο MIME

- Το πρωτόκολλο MIME (multipurpose internet mail extension) είναι μια επέκταση που άρει τους περιορισμούς που θέτει το SMTP στα μηνύματα ηλεκτρονικού ταχυδρομείου (π.χ. δεν μπορεί να μεταφέρει εκτελέσιμα ή άλλα δυαδικά αρχεία, απορρίπτει μηνύματα μεγαλύτερα ενός συγκεκριμένου μεγέθους, δεν μπορεί να χειριστεί δεδομένα που δεν είναι κείμενο).
- Όπως είναι φυσικό (λόγω της βασικής αποστολής του MIME) το μεγαλύτερο μέρος των προδιαγραφών του αφορά τον ορισμό διάφορων τύπων περιεχομένου μηνυμάτων.
- Το MIME καθορίζει 7 βασικούς τύπους περιεχομένου και μια σειρά από επιμέρους τύπους:
  - ✓ κείμενο,
  - ✓ πολλαπλά τμήματα (σώμα μηνύματος με πολλαπλά ανεξάρτητα τμήματα),
  - ✓ μήνυμα (σώμα μηνύματος που ενθυλακώνει άλλο μήνυμα),
  - ✓ εικόνα (απεικονίσιμη εικόνα με μορφοποίηση jpeg ή gif),
  - ✓ βίντεο (χρονικά μεταβαλλόμενη εικόνα ενδεχομένως με χρώμα και ενσωματωμένο ήχο σε μορφοποίηση mpeg),
  - ✓ ήχος (δεδομένα ήχου) και
  - ✓ εφαρμογή (άλλα είδη δεδομένων, δυαδικά δεδομένα ή δεδομένα που πρόκειται να υποστούν επεξεργασία από λογισμικό ηλεκτρονικής αλληλογραφίας).
- Το MIME χρησιμοποιεί δύο μεθόδους κωδικοποίησης των προς μεταφορά δεδομένων.

Εφαρμογή Wireshark

- Με την εφαρμογή Wireshark είναι δυνατή η παρατήρηση ανταλλαγής μηνυμάτων μεταξύ οντοτήτων πρωτοκόλλων, κατά τη διάρκεια εκτέλεσης δικτυακών εφαρμογών μεταξύ του υπολογιστή του χρήστη και ενός υπολογιστή του διαδικτύου.
- Για την παρατήρηση των μηνυμάτων που ανταλλάσσονται μεταξύ των οντοτήτων πρωτοκόλλων χρησιμοποιείται το εργαλείο σύλληψης πακέτων (packet capture, PC).
- Εκτός από τη σύλληψη και αποθήκευση πακέτων, η εφαρμογή έχει τη δυνατότητα απεικόνισης των περιεχομένων διάφορων πεδίων των μηνυμάτων που ανταλλάσσονται.
- Ο PC είναι παθητικός, δηλαδή δε στέλνει μηνύματα, απλά παρατηρεί τα μηνύματα που ανταλλάσσονται από τα πρωτόκολλα και λαμβάνει αντίγραφά τους, τα οποία αποθηκεύει στη βιβλιοθήκη σύλληψης πακέτων (packet capture library, PCL) που διαθέτει.
- Στην PCL αποθηκεύεται ένα αντίγραφο κάθε πλαισίου (frame) δεδομένων επιπέδου ζεύξης που στέλνεται ή λαμβάνεται από τον υπολογιστή του χρήστη.
- Λόγω του ότι τα μηνύματα που ανταλλάσσονται από τα πρωτόκολλα ανώτερων επιπέδων, όπως το HTTP, FTP, TCP, UDP ή το IP, τελικά ενθυλακώνονται σε πλαίσια επιπέδου ζεύξης που μεταδίδονται από το φυσικό μέσο, η σύλληψη των πλαισίων παρέχει όλα τα μηνύματα που στέλνονται ή λαμβάνονται από τα πρωτόκολλα που εκτελούνται.
- Η εφαρμογή περιλαμβάνει και τον αναλυτή πακέτων (packet analyzer, PA), ο οποίος αντιλαμβάνεται τη δομή των μηνυμάτων που ανταλλάσσονται από τα πρωτόκολλα και έχει τη δυνατότητα να απεικονίζει τα περιεχόμενα όλων των πεδίων των μηνυμάτων.

Εφαρμογή Wireshark

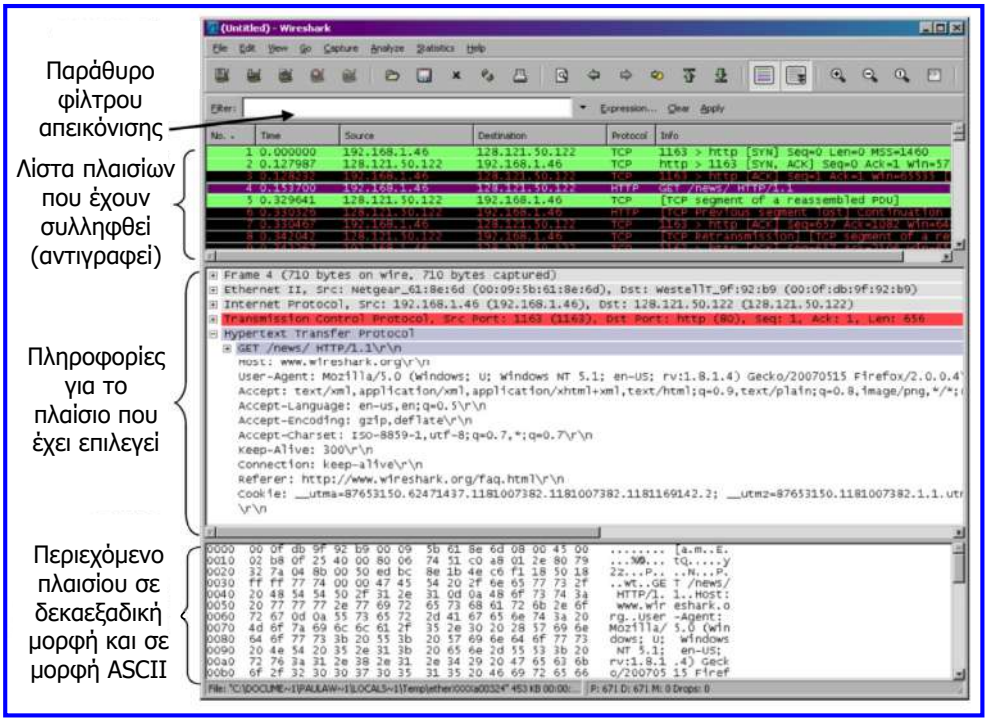


Παράδειγμα – απεικόνιση πεδίων μηνυμάτων πρωτοκόλλου HTTP:

Αφού ο PA αντιλαμβάνεται τη δομή ενός πλαισίου Ethernet, μπορεί να αναγνωρίσει ένα πακέτο IP μέσα σε αυτό. Επίσης, αφού αντιλαμβάνεται τη δομή ενός πακέτου IP, μπορεί να εξάγει ένα τμήμα TCP (ή πακέτο UDP ή μήνυμα ICMP) που ενθυλακώνεται σε αυτό και αφού αντιλαμβάνεται τη δομή ενός τμήματος TCP, μπορεί να εξάγει ένα μήνυμα HTTP που περιλαμβάνεται στο τμήμα TCP. Τέλος, αφού αντιλαμβάνεται τη δομή ενός μηνύματος του πρωτοκόλλου HTTP αναγνωρίζει τα πεδία που αυτό περιλαμβάνει (π.χ. αναγνωρίζει στα πρώτα bytes του μηνύματος HTTP την ακολουθία χαρακτήρων της μεθόδου του αιτήματος GET ή POST).

Εφαρμογή Wireshark

Διεπαφή χρήστη της εφαρμογής Wireshark



Βιβλιογραφία - Μέρος Α: Ψηφιακά συστήματα

- Λ. Μπισδούνης, Ψηφιακά συστήματα, Εκδόσεις Ε.Α.Π., 2015 (ανατύπωση 2017).
- M.M. Mano, M.D. Ciletti, Ψηφιακή σχεδίαση, Εκδόσεις Παπασωτηρίου, 2018.
- Μ. Ρουμελιώτης, Σ.Ι. Σουραβλάς, Ψηφιακή σχεδίαση: Αρχές & εφαρμογές, Εκδόσεις Τζιόλα, 2018.
- V. Nelson, H. Nagle, J. Irwin, B. Carroll, Ανάλυση και σχεδίαση κυκλωμάτων ψηφιακής λογικής, Εκδόσεις Επίκεντρο, 2007.
- J.F. Wakerly, Ψηφιακή σχεδίαση: Αρχές και πρακτικές, Εκδόσεις Κλειδάριθμος, 2019.
- Π. Λιναρδής, Ψηφιακή σχεδίαση Ι, Εκδόσεις Ε.Α.Π., 2008.
- Α. Σκόδρας, Ψηφιακή σχεδίαση ΙΙ, Εκδόσεις Ε.Α.Π., 2008.
- S. Brown, Z. Vranesic, Σχεδίαση ψηφιακών συστημάτων με τη γλώσσα VHDL, Εκδόσεις Τζιόλα, 2011.
- P.K. Lala, Principles of modern digital design, Wiley, 2007.

Βιβλιογραφία - Μέρος Β: Οργάνωση & αρχιτεκτονική Η/Υ

- C. Hammacher, Z. Vranesic, S. Zaky, Οργάνωση και αρχιτεκτονική υπολογιστών, Εκδόσεις Επίκεντρο, 2007.
- W. Stallings, Οργάνωση και αρχιτεκτονική υπολογιστών, Εκδόσεις Τζιόλα, 2011.
- D. Patterson, J. Hennessy, Οργάνωση και σχεδίαση υπολογιστών, Εκδόσεις Κλειδάριθμος, 2010.
- M.M. Mano, C.R. Kime, T. Martin, Σχεδίαση λογικών κυκλωμάτων και υπολογιστών, Εκδόσεις Τζιόλα, 2017.
- Δ. Νικολός, Αρχιτεκτονική υπολογιστών, Εκδόσεις Ε.Α.Π., 2008.
- Γ. Αλεξίου, Μικροεπεξεργαστές, Εκδόσεις Ε.Α.Π., 2008.
- A. Nagoor Kani, Microprocessor 8085 and its applications, McGraw-Hill, 2005.
- K. Kumar, B. Umashankar, The 8085 microprocessor: Architecture, programming and interfacing, Pearson, 2008.
- Intel Corp., 8085AH 8-bit HMOS microprocessors, 1987.
- Intel Corp., Intel 8080/8085 assembly language programming, 1977.
- CPU world, Microprocessor news, benchmarks, information, pictures, www.cpu-world.com.

Βιβλιογραφία - Μέρος Γ: Δίκτυα υπολογιστών

- Α. Παπαζαφειρόπουλος, Δ. Τσώλης, Δίκτυα υπολογιστών, Εκδόσεις Ε.Α.Π., 2015 (ανατύπωση 2017).
- W. Stallings, Επικοινωνίες υπολογιστών και δεδομένων, Εκδόσεις Τζιόλα, 2018.
- A.S. Tanenbaum, D.J. Wetherall, Δίκτυα υπολογιστών, Εκδόσεις Κλειδάριθμος, 2011.
- B.A. Forouzan, F. Mosharraf, Δίκτυα υπολογιστών: Προσέγγιση από πάνω προς τα κάτω, Εκδόσεις Παπασωτηρίου, 2011.
- J. Kurose, K. Ross, Δικτύωση υπολογιστών: Προσέγγιση από πάνω προς τα κάτω, Εκδόσεις Μ. Γκιούρδα, 2018.
- Γ. Φούσκας, Δίκτυα υπολογιστών, Εκδόσεις Ε.Α.Π., 2002.
- B.A. Forouzan, Πρωτόκολλο TCP/IP, Εκδόσεις Μ. Γκιούρδα, 2006.
- B.A. Forouzan, Data communications and networking, McGraw-Hill, 2013.
- L.L. Peterson, B.S. Davie, Computer networks: A systems approach, Morgan Kaufmann, Elsevier, 2007.

e-mail: bisdounis.lampros@ac.eap.gr

ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Σημειώσεις διδασκαλίας ΠΛΣ51, Λ. Μπισδούνης