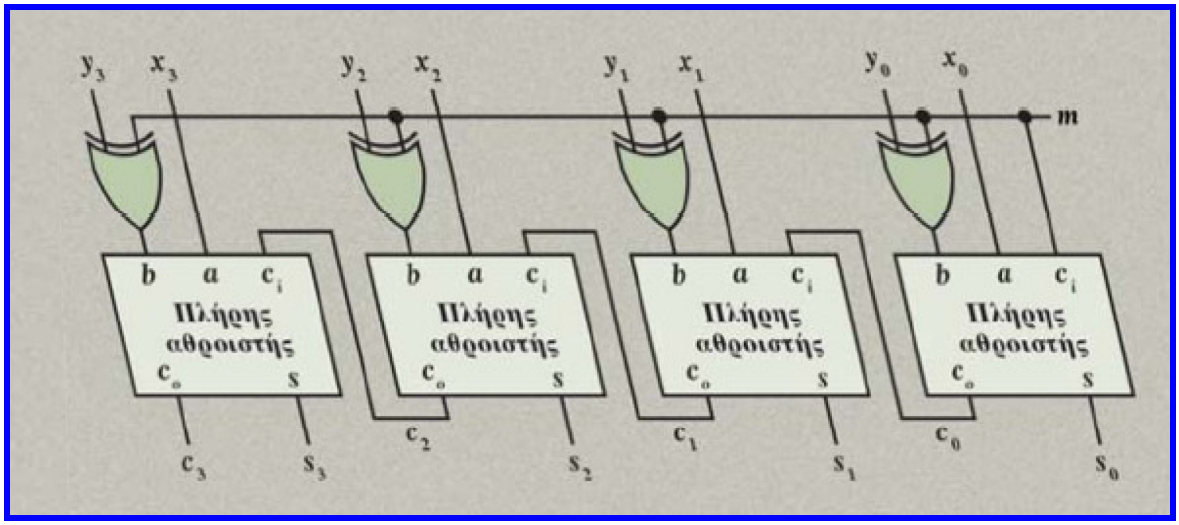
ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ: ΠΛΗΡΟΦΟΡΙΚΗ

ΘΕΜΑΤΙΚΗ ΕΝΟΤΗΤΑ:

ΠΛΗ21 - ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ



– Σημειώσεις διδασκαλίας –

Λάμπρος Μπισδούνης
Μέλος Σ.Ε.Π. Ε.Α.Π.
Καθηγητής Πανεπιστημίου Πελοποννήσου

2023

Σκοπός της θεματικής ενότητας

- Εισαγωγή στο υλικό (hardware) των υπολογιστικών συστημάτων με έμφαση στις βασικές έννοιες των αριθμητικών συστημάτων, της δυαδικής λογικής και των λογικών κυκλωμάτων, στην ανάλυση και το σχεδιασμό συνδυαστικών και ακολουθιακών ψηφιακών κυκλωμάτων, τα οποία αποτελούν τις βασικές δομικές μονάδες των ψηφιακών συστημάτων.

- Ανάλυση της δομής και λειτουργίας της κεντρικής μονάδας επεξεργασίας και του συστήματος μνήμης ενός υπολογιστικού συστήματος, καθώς και της διασύνδεσης τους με περιφερειακές μονάδες.

- Εισαγωγή στους μικροεπεξεργαστές και ενασχόληση με τη δομή και τη λειτουργία αντιπροσωπευτικών μικροεπεξεργαστών, καθώς και με τον προγραμματισμό σε επίπεδο συμβολικής γλώσσας.

Περιεχόμενα

1. Ψηφιακή σχεδίαση

- Αναλογικά και ψηφιακά σήματα, ψηφιακά συστήματα
- Παράσταση αριθμητικών δεδομένων στο δυαδικό αριθμητικό σύστημα, εκτέλεση αριθμητικών πράξεων, αριθμοί κινητής υποδιαστολής, δυαδικοί κώδικες
- Κώδικες ανίχνευσης και διόρθωσης λαθών
- Άλγεβρα Boole (ορισμοί, αξιώματα, θεωρήματα, ιδιότητες)
- Τρόποι περιγραφής λογικών παραστάσεων / συναρτήσεων
- Αντιστοίχιση βασικών λογικών συναρτήσεων σε λογικές πύλες και υλοποίηση λογικών παραστάσεων / συναρτήσεων με λογικές πύλες
- Ελαχιστοποίηση (απλοποίηση) λογικών συναρτήσεων και κυκλωμάτων με τη μέθοδο του χάρτη Karnaugh
- Λειτουργικότητα και σχεδίαση σύνθετων λογικών κυκλωμάτων που χρησιμοποιούνται στα ψηφιακά συστήματα (αθροιστές, αφαιρέτες, συγκριτές, κωδικοποιητές, αποκωδικοποιητές, πολυπλέκτες, αποπολυπλέκτες, ολισθητές)
- Βασικά στοιχεία μνήμης (μανταλωτές, flip-flops) των ψηφιακών κυκλωμάτων
- Σχεδίαση (σύνθεση) και ανάλυση σύγχρονων ακολουθιακών κυκλωμάτων
- Διάφορα είδη καταχωρητών (παράλληλοι καταχωρητές, καταχωρητές ολίσθησης, καταχωρητές με πολλαπλές λειτουργίες, κυκλικοί καταχωρητές)

2. Αρχιτεκτονική υπολογιστών

- Υπολογιστικό σύστημα (λογισμικό και υλικό), επίπεδα αρχιτεκτονικής υπολογιστών, δομή και οργάνωση υπολογιστικού συστήματος (κεντρική μονάδα επεξεργασίας, σύστημα μνήμης, μονάδες εισόδου/εξόδου).
- Τρόποι παράστασης πληροφορίας στον υπολογιστή και εκτίμηση απόδοσης υπολογιστή.
- Εντολές γλώσσας μηχανής, τρόποι διευθυνσιοδότησης της κύριας μνήμης του υπολογιστή, ταξινόμηση υπολογιστών βάσει του συνόλου εντολών.
- Κεντρική μονάδα επεξεργασίας: δομή και υλοποίηση της μονάδας επεξεργασίας δεδομένων και της μονάδας ελέγχου.
- Σύστημα μνήμης: ημιαγωγικές μνήμες, μαγνητικός δίσκος, ιεραρχία μνήμης του υπολογιστή, κρυφή (cache) μνήμη, κύρια μνήμη και διαφύλλωση κύριας μνήμης.
- Σύστημα διασύνδεσης: αρτηρίες (busses) υπολογιστή.
- Λειτουργίες εισόδου και εξόδου: διαδικασία διακίνησης πληροφορίας μεταξύ της κεντρικής μονάδας επεξεργασίας του υπολογιστή και των περιφερειακών μονάδων (μονάδων εισόδου/εξόδου).

3. Μικροεπεξεργαστές

- Εισαγωγή στα ολοκληρωμένα κυκλώματα και τους μικροεπεξεργαστές, ιστορική αναδρομή και εξέλιξη.
- Δομή (βασικές μονάδες), λειτουργία και χρονισμός εντολών των μικροεπεξεργαστών.
- Εντολές και τρόποι διευθυνσιοδότησης.
- Εκτέλεση προγράμματος σε μικροεπεξεργαστή.
- Δομή των μικροεπεξεργαστών Intel 8080/8085.
- Μικροεπεξεργαστής Intel 8085: αρχιτεκτονική (μονάδες, καταχωρητές), ακροδέκτες και σήματα, σύνολο εντολών, τρόποι διευθυνσιοδότησης, χρονισμός εντολών, κύκλοι μηχανής, λειτουργίες εισόδου/εξόδου.
- Παραδείγματα προγραμματισμού στον μικροεπεξεργαστή Intel 8085 και προσομοιωτές (simulators) της λειτουργίας του.
- Μικροεπεξεργαστής Motorola 6800: δομή, ακροδέκτες, χρονισμός εντολών, σύνολο εντολών, τρόποι διευθυνσιοδότησης, σύγκριση του Motorola 6800 με τον Intel 8085
- Μικροϋπολογιστικά συστήματα: διασύνδεση μνημών (RAM, ROM) και μονάδων εισόδου/εξόδου με μικροεπεξεργαστές.
- Εξέλιξη μικροεπεξεργαστών.

1. Ψηφιακή σχεδίαση

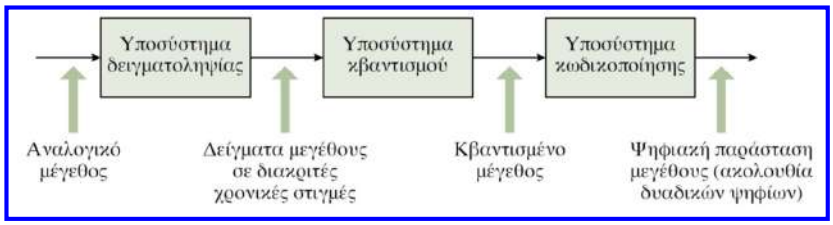
Ψηφιακά και αναλογικά σήματα

- Στις επιστήμες, την τεχνολογία, την επιχειρηματικότητα, συναντάται πολύ συχνά η ανάγκη χειρισμού ποσοτικών μεγεθών. Τα μεγέθη αυτά μετρούνται, παρακολουθούνται, συλλέγονται, επεξεργάζονται, καταγράφονται ή αποθηκεύονται και είναι σημαντική η ύπαρξη της δυνατότητας παράστασής τους 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), ροή νερού σε σωλήνα ύδρευσης (αναλογικό).
- Στην αναλογική παράσταση, ένα μέγεθος λαμβάνει συνεχείς τιμές, ενώ στην ψηφιακή παράσταση λαμβάνει διακριτές τιμές.
- Τα περισσότερα μεγέθη που μπορούν να μετρηθούν ποσοτικά, όπως θερμοκρασία, πίεση, απόσταση, ταχύτητα, ένταση ηλεκτρικού ρεύματος κ.ά., εμφανίζονται στη φύση με αναλογική μορφή.
- Επειδή τα ψηφιακά συστήματα χειρίζονται ψηφιακά σήματα, είναι απαραίτητα κατάλληλα συστήματα μετατροπής, τα οποία επιτελούν τη διαδικασία μετατροπής ενός μεγέθους που παριστάνεται με αναλογικό τρόπο σε ψηφιακή μορφή.

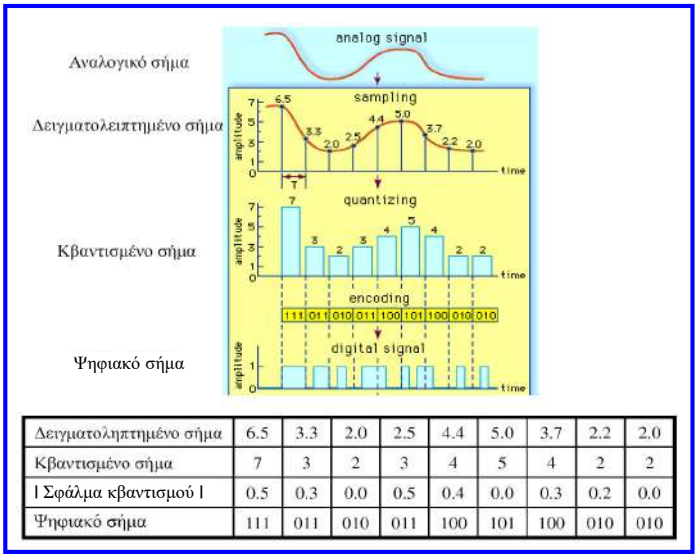
Μετατροπή αναλογικών σημάτων σε ψηφιακά

- 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 T δευτερόλεπτα: $x = x_a(nT)$.
- 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.

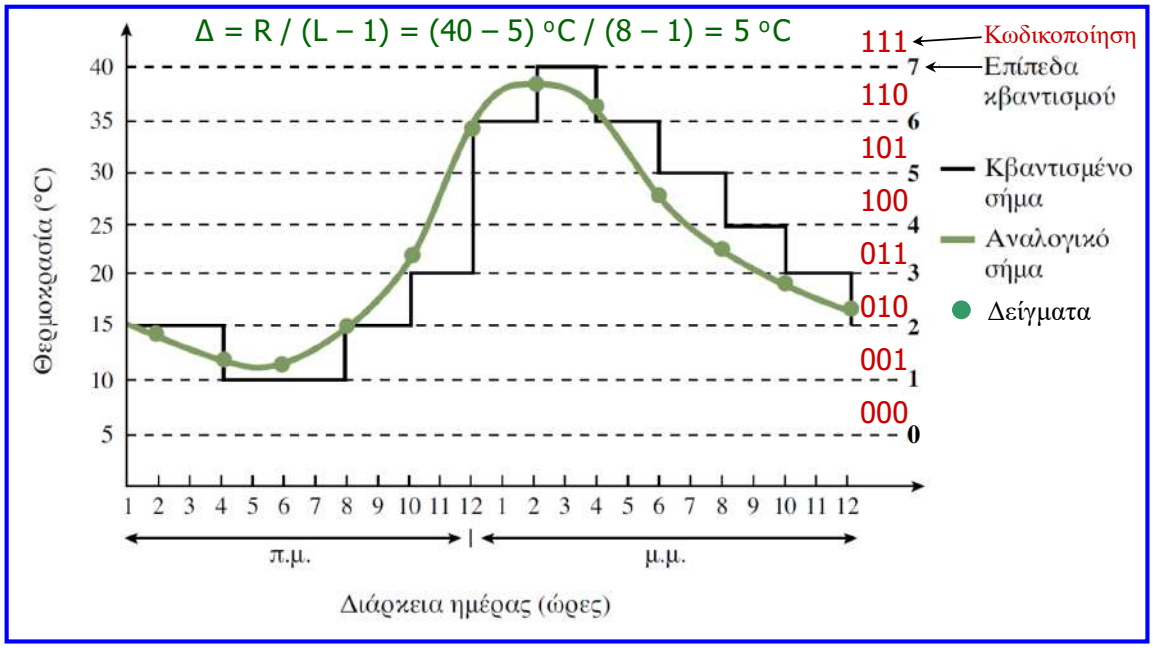
$$e_q = x - x_q$$

$$e_q \le \frac{\Delta}{2}$$

- Η αύξηση του πλήθους των επιπέδων κβαντισμού (L) οδηγεί στην μείωση του σφάλματος.

Μετατροπή αναλογικών σημάτων σε ψηφιακά

R: περιοχή τιμών μεγέθους, L: πλήθος επιπέδων κβαντισμού, Δ: απόσταση 2 επιπέδων κβαντισμού



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 significantdigit, MSD) ή περισσότερο σημαντικό δυαδικό ψηφίο (most significantbit, MSB), όταν πρόκειται για δυαδικό αριθμό, ενώ το τελευταίο αναφέρεται ως λιγότερο σημαντικό ψηφίο (least significantdigit, LSD) ή λιγότερο σημαντικό δυαδικό ψηφίο (least significantbit, 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 δεκαέξι 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 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 $\mathbf{(10110001101011.1111)_2}$ 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 $\mathbf{(306.D)_{16}}$ 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 την 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 Α 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 Α 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 Α 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

Προσημασμένοι δυαδικοί αριθμοί

- Παράδειγμα: Δεκαδικός αριθμός –13 σε παράσταση προσημασμένου μεγέθους (πρόσημο-μέτρο) με 9 δυαδικά ψηφία, σε παράσταση συμπληρώματος ως προς 1 με 9 δυαδικά ψηφία, σε παράσταση συμπληρώματος ως προς 2 με 9 δυαδικά ψηφία και σε παράσταση συμπληρώματος ως προς 8 με 3 ψηφία.
- $|-13|$ με 8 δυαδικά ψηφία = $2^3 + 2^2 + 2^0 = 00001101_2$
- Παράσταση προσημασμένου μεγέθους με 9 δυαδικά ψηφία: **1**00001101.
- Προσδιορισμός του αντίστοιχου θετικού αριθμού: +13 = **0**00001101.
- Παράσταση συμπληρώματος ως προς 1 με 9 δυαδικά ψηφία = 111110010.
- Παράσταση συμπληρώματος ως προς 2 με 9 ψηφία = 111110010 + 1 = 111110011.
- Η παράσταση συμπληρώματος ως προς 8 με 3 ψηφία προκύπτει από τη μετατροπή στο οκταδικό σύστημα της παράστασης συμπληρώματος ως 2: αφού λοιπόν 111 = 7_8, 110 = 6_8, 011 = 3_8, η ζητούμενη παράσταση είναι: 763.
- Εναλλακτικά, βρίσκουμε την οκταδική παράσταση με 3 ψηφία του +13 (= $0\times8^2 + 1\times8^1 + 5\times8^0$) που είναι ο οκταδικός αριθμός 015 και στη συνέχεια υπολογίζουμε το συμπλήρωμά του ως προς 8 (777 – 015 = 762 + 1 = 763) που είναι η ζητούμενη παράσταση του δεκαδικού αριθμού –13.

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 του δεκαδικού συστήματος, 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 και να λάβουμε υπόψη το δανειζόμενο ψηφίο στην αφαίρεση των ψηφίων της επόμενης θέσης.
- 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 (Μ) το συμπλήρωμα ως προς 2 του αφαιρετέου (Α), 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	$(+1)_{10}$

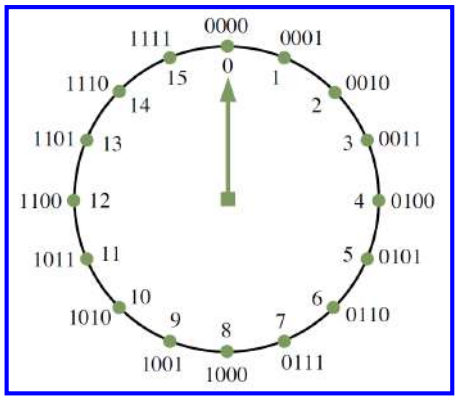
Υπερχείλιση

- 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

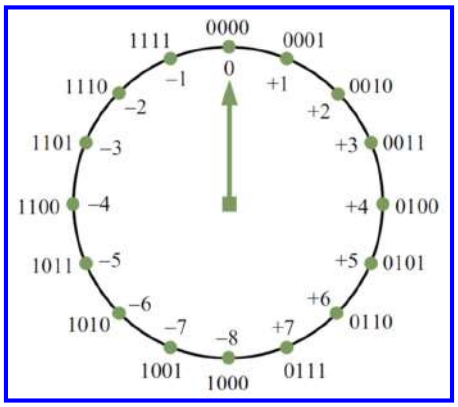
- Κατά την πρόσθεση των θετικών δυαδικών αριθμών 0111 = $(+7)_{10}$ και 0100 = $(+4)_{10}$ προκύπτει λανθασμένο αρνητικό αποτέλεσμα, δηλαδή 1011 = $(-5)_{10}$, λόγω της υπερχείλισης.
- Αυτό συμβαίνει διότι το ορθό αποτέλεσμα 01011 = $(+11)_{10}$ είναι μεγαλύτερο από το μεγαλύτερο δυνατό θετικό αριθμό 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 $(-8)_{10}$.

Κυκλική διάταξη δυαδικών αριθμών



- Για την πρόσθεση δύο αριθμών (π.χ. 3 + 9 ή 0011 + 1001), ξεκινώντας από τον αριθμό 0011 μετακινούμε το δείκτη δεξιόστροφα κατά 9 θέσεις και καταλήγουμε στο άθροισμα που είναι ο αριθμός 12 (1100).
- Εάν το ζητούμενο άθροισμα είναι μεγαλύτερο του 15 (π.χ. 3 + 14 = 17 ή 0011 + 1110 = 10001), η τελική θέση του δείκτη θα είναι ο αριθμός 0001, δηλαδή θα απουσιάζει το κρατούμενο ψηφίο που προκύπτει από την πρόσθεση στην πιο σημαντική θέση.
- Συνεπώς, ο κανόνας ύπαρξης κρατουμένου είναι η μετάβαση του δείκτη από τη θέση 15 στη θέση 0.
- Όταν πρόκειται για αφαίρεση θα πρέπει να μετακινήσουμε το δείκτη δεξιόστροφα τόσες θέσεις, όσες υποδεικνύει το συμπλήρωμα ως προς 2 του αφαιρετέου ή μπορούσε να υιοθετήσουμε αριστερόστροφη κίνηση του δείκτη κατά το πλήθος θέσεων που υποδεικνύεται από τον μειωτέο.

Κυκλική διάταξη προσημασμένων δυαδικών αριθμών



- Για την πρόσθεση δύο προσημασμένων αριθμών (π.χ. +3 και –7 ή 0011 και 1001), ξεκινώντας από τον αριθμό 0011 μετακινούμε το δείκτη δεξιόστροφα κατά 9 θέσεις (αφού 1001 = 9) και καταλήγουμε στο άθροισμα που είναι ο αριθμός –4 (1100).
- Η πράξη της αφαίρεσης προσημασμένων αριθμών ανάγεται σε πρόσθεση του μειωτέου με τον αντίθετο αριθμό του αφαιρετέου.
- Μπορούμε να παρατηρήσουμε ότι εάν προσθέσουμε θετικούς αριθμούς, υπερχείλιση συμβαίνει όταν υπάρχει διάβαση του δείκτη από τη θέση +7 στη θέση -8, ενώ εάν προσθέτετε αρνητικούς αριθμούς, υπερχείλιση συμβαίνει όταν δεν υπάρχει μετάβαση του δείκτη από τη θέση +7 στη θέση –8.
- Για παράδειγμα, κατά την πρόσθεση των +7 και +4, υπάρχει μετάβαση του δείκτη από τη θέση +7 στη θέση -8 και η τελική του θέση είναι ο αριθμός –5 (1011), ενώ το ορθό αποτέλεσμα της πρόσθεσης είναι ο αριθμός +11 (01011). Το λανθασμένο αποτέλεσμα οφείλεται στην υπερχείλιση.

Πολλαπλασιασμός και διαίρεση δυαδικών αριθμών

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

				0	1	1	0	0	$(12)_{10}$
×				1	0	1	0	1	$(21)_{10}$
				0	1	1	0	0	
			0	0	0	0	0		
		0	1	1	0	0			
	0	0	0	0	0				
+	0	1	1	0	0				
	0	1	1	1	1	1	0	0	$(252)_{10}$

$(119)_{10}$: διαιρετέος

	1	1	1	0	1	1	1	1	0	0	1	$(9)_{10}$: διαιρέτης
−	1	0	0	1	↓			1	1	0	1	$(13)_{10}$: πηλίκο
	0	1	0	1	1							
	−	1	0	0	1	↓						
		0	0	1	0	1						
			1	0	0	1	↓					
			0	1	0	1	1					
			−	1	0	0	1					
				0	0	1	0	$(2)_{10}$: υπόλοιπο				

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).

	1	1	
	4	5	6
+	1	2	2
	6	0	0

1	1	1	
	A	B	C
+	E	D	F
1	9	9	B

- 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.

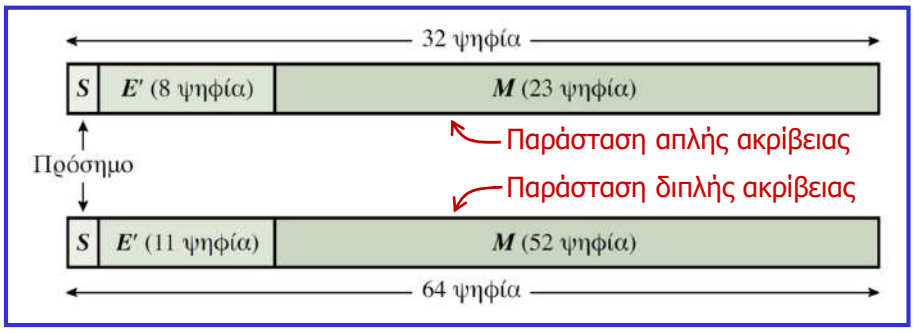
Αριθμοί κινητής υποδιαστολής

- 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 (όταν E′ = M = 0), ∞ (για E′ = 255, M = 0) και NaN ή «Not A Number» (όταν E′ = 255, M # 0).
- Το ψηφίο που βρίσκεται αριστερά της υποδιαστολής είναι πάντοτε 1 και δεν αναπαρίσταται, αλλά υπονοείται 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.

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

Κώδικας BCD

- 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.
- Κατά την πρόσθεση δύο δεκαδικών ψηφίων κωδικοποιημένων κατά BCD, όταν το δυαδικό άθροισμα είναι ίσο ή μικρότερο του 1001 (9), τότε ταυτίζεται με το άθροισμα κατά BCD.
- Όταν το δυαδικό άθροισμα είναι μεγαλύτερο του 1001 (δηλαδή δεν περιλαμβάνεται στον BCD), θα πρέπει να προσθέσουμε στο δυαδικό άθροισμα τον αριθμό 0110 (6).
- Θα προκύψει κρατούμενο δυαδικό ψηφίο, το οποίο θα πρέπει να ληφθεί υπόψη κατά την πρόσθεση των κωδικοποιημένων κατά BCD ψηφίων της πιο σημαντικής θέσης.

$$
\begin{array}{rcccccccccc}
 & 0 & 0 & 1 & 0 & & 0 & 1 & 0 & 0 & (24)_{10} \\
+ & 0 & 1 & 1 & 0 & & 0 & 1 & 1 & 1 & (67)_{10} \\
\hline
 & 1 & 0 & 0 & 0 & & 1 & 0 & 1 & 1 & \\
 & & & & & + & 0 & 1 & 1 & 0 & (6)_{10} \\
\hline
+ & & & & 1 & \leftarrow & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \\
\hline
 & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & & & & & &
\end{array}
$$

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, ο οποίος δεν ακολουθεί συντελεστές βαρύτητας.

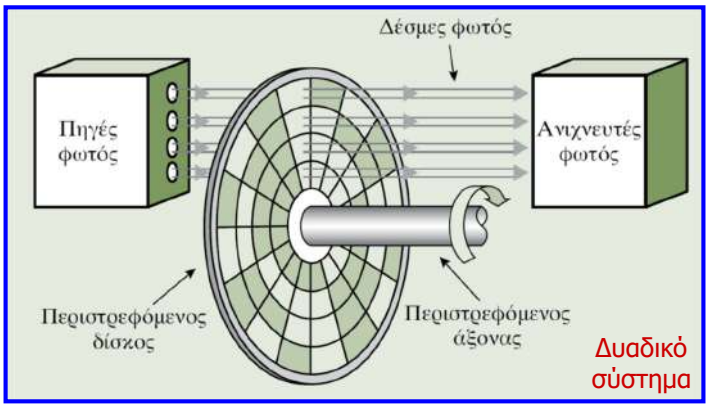
αρχή	ανάκλαση	προσθήκη	ανάκλαση	προσθήκη	ανάκλαση	προσθήκη
0	0	00	00	000	000	0000
1	1	01	01	001	001	0001
	1	11	11	011	011	0011
	0	10	10	010	010	0010
			10	110	110	0110
			11	111	111	0111
			01	101	101	0101
			00	100	100	0100
					100	1100
					101	1101
					111	1111
					110	1110
					010	1010
					011	1011
					001	1001
					000	1000

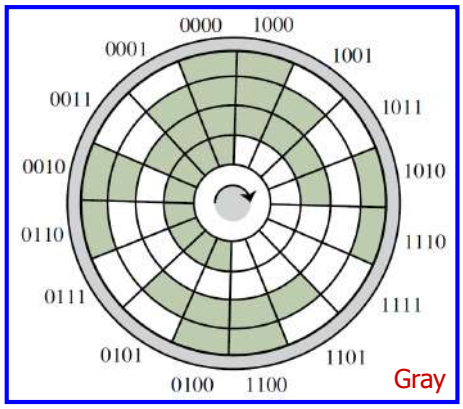
Παραγωγή του κώδικα Gray με ανάκλαση

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.

Μετατροπή μεταξύ κώδικα Gray & δυαδικού συστήματος

Μετατροπή μεταξύ του δυαδικού αριθμού $b_n b_{n-1} \ldots b_1 b_0$ και της λέξης κώδικα Gray $g_n g_{n-1} \ldots g_1 g_0$ (το σύμβολο + σε κύκλο δηλώνει πρόσθεση με αγνόηση κρατουμένου):

$$g_n = b_n$$
$$g_i = b_i \oplus b_{i+1}, \quad 0 \le i < n$$

$$\begin{aligned} g_5 &= b_5 && \Rightarrow && 1 \\ g_4 &= b_4 \oplus b_5 && \Rightarrow && 0 \oplus 1 = 1 \\ g_3 &= b_3 \oplus b_4 && \Rightarrow && 0 \oplus 1 = 1 \\ g_2 &= b_2 \oplus b_3 && \Rightarrow && 1 \oplus 1 = 0 \\ g_1 &= b_1 \oplus b_2 && \Rightarrow && 0 \oplus 1 = 1 \\ g_0 &= b_0 \oplus b_1 && \Rightarrow && 1 \oplus 0 = 1 \end{aligned}$$

Η λέξη του κώδικα Gray που αντιστοιχεί στον δυαδικό αριθμό 101101 είναι η 111011

$$b_n = g_n$$
$$b_i = g_i \oplus b_{i+1}, \quad 0 \le i < n$$

$$\begin{aligned} b_3 &= g_3 && \Rightarrow && 1 \\ b_2 &= g_2 \oplus b_3 && \Rightarrow && 1 \oplus 1 = 0 \\ b_1 &= g_1 \oplus b_2 && \Rightarrow && 0 \oplus 0 = 0 \\ b_0 &= g_0 \oplus b_1 && \Rightarrow && 1 \oplus 0 = 1 \end{aligned}$$

Η λέξη 1101 του κώδικα Gray αντιστοιχεί στον δυαδικό αριθμό 1001

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, στων οποίων την πιο σημαντική θέση 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.

Κώδικας Hamming

- Όλοι οι κώδικες διόρθωσης / ανίχνευσης σφαλμάτων προσθέτουν πλεονασματική πληροφορία στα δεδομένα που αποστέλλονται.
- Πληρέστερος κώδικας ανίχνευσης σφαλμάτων είναι ο κώδικας Hamming, ο οποίος έχει 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 ελέγχου (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:

$\mathbf{C_1}$ = {0,1,0,1,1,1} = **0**
$\mathbf{C_2}$ = {1,1,0,1,0,1} = **0**
$\mathbf{C_4}$ = {1,0,0,1,0} = **0**
$\mathbf{C_8}$ = {0,1,0,1,0} = **0**

$C_8C_4C_2C_1$ = 0000 ⇒ σωστή λήψη

$\mathbf{C_1}$ = {0,1,0,1,1,1} = **0**
$\mathbf{C_2}$ = {1,1,1,1,0,1} = **1**
$\mathbf{C_4}$ = {1,0,1,1,0} = **1**
$\mathbf{C_8}$ = {0,1,0,1,0} = **0**

→ 2 + 4 = 6

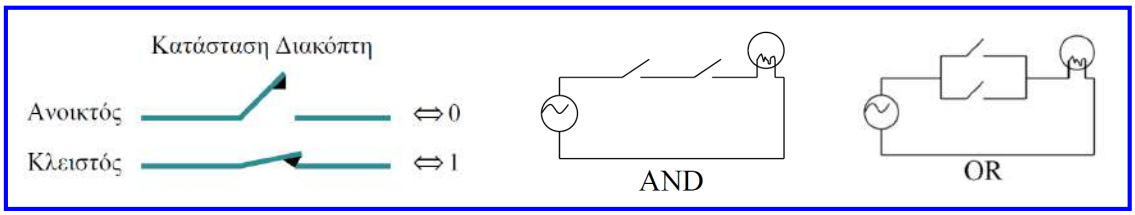
$C_8C_4C_2C_1 = 0110 = 6_{10}$ ⇒ σφάλμα στη θέση 6

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.** Θεώρημα 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 $x \cdot (x + y) = x$.
- **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 εξοικείωσή 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:

$$x'y' + y'z + xz + xy + yz' = x'y' + y'z(x + x') + xz + xy + yz'$$
$$= x'y' + xy'z + x'y'z + xz + xy + yz' = x'y'(1 + z) + xz(y' + 1) + xy + yz'$$
$$= x'y'1 + xz1 + xy(z + z') + yz' = x'y' + xz + xyz + xyz' + yz'$$
$$= x'y' + xz(1 + y) + yz'(1 + x) = x'y' + xz1 + yz'1 = x'y' + xz + yz'$$

- Για την απόδειξη του πρώτου τμήματος του θεωρήματος ομοφωνίας, θα πρέπει να πάλι το αξίωμα (x + x′ = 1), προκειμένου 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:

$$xy + x'z + yz = xy + x'z + yz(x + x') = xy + x'z + yzx + yzx'$$
$$= xy(1 + z) + x'z(1 + y) = xy1 + x'z1 = xy + x'z$$

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. 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:

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

- Στην τελευταία ισότητα εφαρμόστηκε το θεώρημα ομοφωνίας.

Ελάχιστοι και μέγιστοι όροι λογικών συναρτήσεων

- Ένα λογικό γινόμενο, στο οποίο συμμετέχουν όλες οι μεταβλητές μιας συνάρτησης (στην κανονική μορφή τους ή στη μορφή συμπληρώματος) μία φορά, ονομάζεται ελάχιστος όρος ή ελαχιστόρος (minterm) ενώ ένα λογικό άθροισμα μεταβλητών με τα ίδια χαρακτηριστικά ονομάζεται μέγιστος όρος ή 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 ελάχιστος όρος και ένας μέγιστος όρος, 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 7 δυνατών συνδυασμών τιμών των 3 μεταβλητών και των 7 ελαχίστων όρων.
- Όταν η τιμή μιας μεταβλητής είναι 0, τότε στον αντίστοιχο ελάχιστο όρο συμμετέχει το συμπλήρωμα της μεταβλητής αυτής, ενώ όταν η τιμή της μεταβλητής είναι 1, τότε στον αντίστοιχο ελάχιστο όρο συμμετέχει η μεταβλητή αυτή με την κανονική μορφή της.

Ελάχιστοι και μέγιστοι όροι λογικών συναρτήσεων

- Οι ελάχιστοι όροι ονομάζονται ως m_0 έως m_7 (ο δείκτης συμπίπτει με το δεκαδικό αριθμό που αντιστοιχεί σε κάθε ελάχιστο όρο). Οι μέγιστοι όροι (M_0 έως M_7) είναι τα συμπληρώματα των αντίστοιχων ελαχίστων όρων και προκύπτουν εύκολα από τους ελάχιστους όρους με εφαρμογή του θεωρήματος De Morgan.
- Για n μεταβλητές, σχηματίζονται 2^n ελάχιστοι όροι και 2^n μέγιστοι όροι.

x	y	z	Ελάχιστοι όροι		Μέγιστοι όροι	
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	$x'yz'$	(m_2)	$x+y'+z$	(M_2)
0	1	1	$x'yz$	(m_3)	$x+y'+z'$	(M_3)
1	0	0	$xy'z'$	(m_4)	$x'+y+z$	(M_4)
1	0	1	$xy'z$	(m_5)	$x'+y+z'$	(M_5)
1	1	0	xyz'	(m_6)	$x'+y'+z$	(M_6)
1	1	1	xyz	(m_7)	$x'+y'+z'$	(M_7)

Κανονικές & πρότυπες μορφές λογικής συνάρτησης

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

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

- Η F' μπορεί να εκφραστεί σε μορφή αθροίσματος ελαχίστων όρων, εάν προσθέσουμε τους ελάχιστους όρους που αντιστοιχούν στους συνδυασμούς τιμών των μεταβλητών, για τους οποίους η συνάρτηση λαμβάνει τιμή 0, που είναι οι ελάχιστοι όροι 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 μέγιστων όρων. Στη μορφή αυτή συμμετέχουν οι μέγιστοι όροι που αντιστοιχούν στους συνδυασμούς τιμών των μεταβλητών, για τους οποίους η συνάρτηση λαμβάνει λογική τιμή 0.
- Οι αλγεβρικές εκφράσεις μιας λογικής συνάρτησης σε μορφή αθροίσματος ελαχίστων όρων και γινομένου μέγιστων όρων, αναφέρονται ως κανονικές μορφές (canonical forms).

Κανονικές & 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 ελαχίστων όρων ή των μεγίστων όρων που απουσιάζουν από την αρχική κανονική μορφή.
- Για να προκύψουν οι δείκτες αυτοί, θα πρέπει να λάβουμε υπόψη ότι ο αριθμός των ελαχίστων ή μεγίστων όρων για συναρτήσεις n μεταβλητών ανέρχεται σε 2^n.
- Το άθροισμα ελαχίστων όρων και το γινόμενο μεγίστων όρων μιας λογικής συνάρτησης αναφέρονται και ως Σ( ), Π( ), αντίστοιχα, θέτοντας εντός των παρενθέσεων τους δείκτες των ελαχίστων όρων ή των μεγίστων όρων, αντίστοιχα.
- Οι κανονικές μορφές μιας λογικής συνάρτησης προκύπτουν απευθείας από τον πίνακα αλήθειας και λόγω του ότι στους ελάχιστους και στους μέγιστους όρους 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 & 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 ελαχίστων όρων.
- 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 μεγίστων όρων.
- Η μετατροπή μεταξύ πρότυπων μορφών γίνεται εύκολα με χρήση του αξιώματος της επιμεριστικότητας.

Κανονικές & πρότυπες μορφές λογικής συνάρτησης

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

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

- 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 μεγίστων όρων 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 μέγιστων όρων:

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

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).

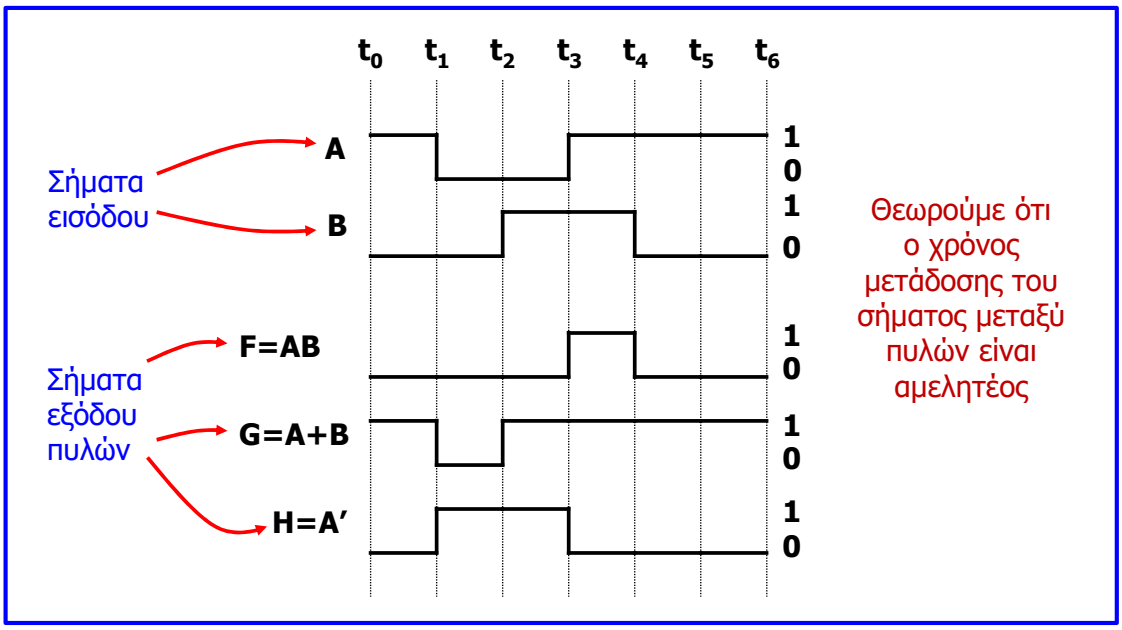
Λογικές πύλες

- 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	x ή y — x' ή y'
$F_1 = xy$	Λογικό γινόμενο	AND	x, y — xy
$F_7 = x + y$	Λογικό άθροισμα	OR	x, y — $x + y$
$F_{14} = (xy)'$	Συμπλήρωμα λογικού γινομένου	NAND	x, y — $(xy)'$
$F_8 = (x + y)'$	Συμπλήρωμα λογικού αθροίσματος	NOR	x, y — $(x + y)'$
$F_6 = xy' + x'y = x \oplus y$	Αποκλειστικό OR (exclusive OR)	XOR	x, y — $x'y + xy'$
$F_9 = xy + x'y' = (x \oplus y)' = x \odot y$	Ισοδυναμία	XNOR	x, y — $xy + x'y'$

Λογικές πύλες

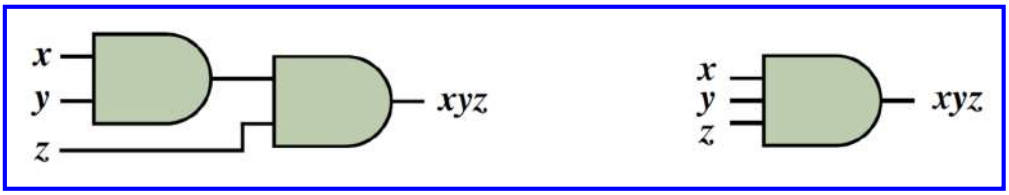
Χρονικά διαγράμματα (κυματομορφές)



Θεωρούμε ότι ο χρόνος μετάδοσης του σήματος μεταξύ πυλών είναι αμελητέος

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.



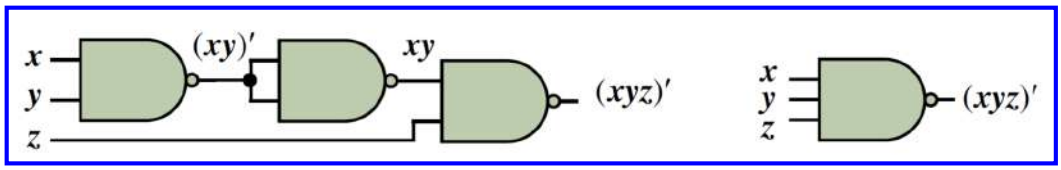
- Η πύλη OR παράγει στην έξοδό της λογική τιμή 1, όταν έστω μία από τις δύο εισόδους της λαμβάνει λογική τιμή 1, ενώ όταν όλες οι είσοδοι λαμβάνουν τιμή 0, τότε παράγει στην έξοδό της λογική τιμή 0.
- Η πύλη OR μπορεί, επίσης, να επεκταθεί σε τρεις ή περισσότερες εισόδους.

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)’ χρησιμοποιώντας δύο πύλες 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

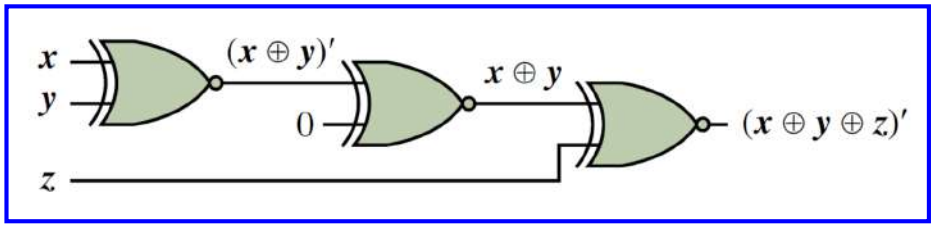
- Για την υλοποίηση μιας πύλης NAND τριών εισόδων, απαιτούνται τρεις πύλες NAND δύο εισόδων.
- Όταν θέτουμε στις δύο εισόδους μιας πύλης NAND την ίδια μεταβλητή εισόδου, τότε η πύλη λειτουργεί ως αντιστροφέας, αφού (AA)′ = A′.



- 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

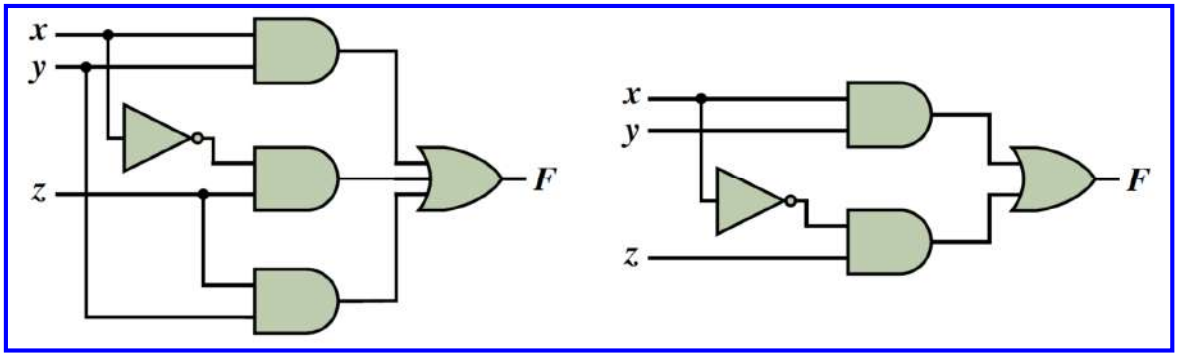
- Η πύλη XNOR δύο εισόδων παράγει ως έξοδο το συμπλήρωμα της εξόδου της πύλης XOR, αφού οι συναρτήσεις που επιτελούνται από τις πύλες XOR και XNOR είναι συμπληρωματικές μεταξύ τους.
- Για περισσότερες από δύο εισόδους, η έξοδος μιας πύλης XNOR λαμβάνει λογική τιμή 1, όταν άρτιος αριθμός εισόδων λαμβάνει 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 $(x \oplus y \oplus z)'$ χρησιμοποιώντας μια ακολουθία δύο πυλών XNOR δύο εισόδων.
- Για την υλοποίηση μιας πύλης XNOR τριών εισόδων, απαιτούνται τρεις πύλες XNOR δύο εισόδων.



Λογικά κυκλώματα με πύλες NOT, AND και OR

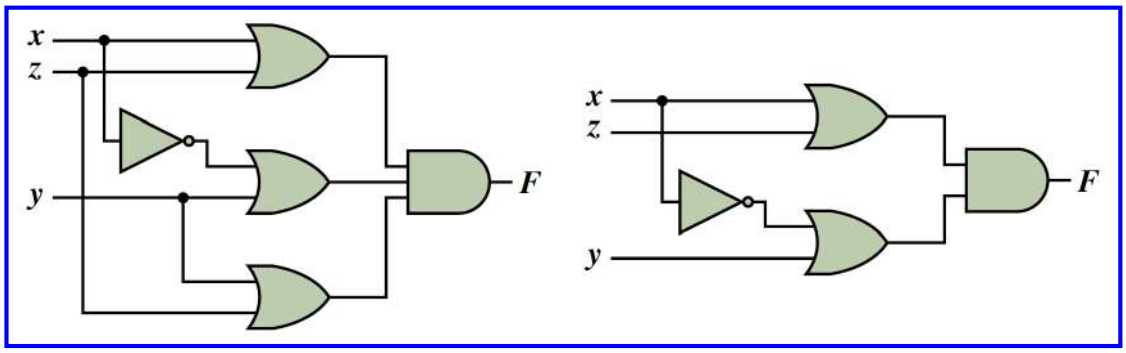
- Ένα λογικό κύκλωμα αποτελείται από γραμμές διασύνδεσης λογικών πυλών που αντιστοιχούν στις διαδρομές των δυαδικών σημάτων και από λογικές πύλες που επιτελούν την επεξεργασία μεταξύ των σημάτων, η οποία δηλώνεται στη λογική συνάρτηση που επιτελείται από κάθε πύλη.
- Τα τρία λογικά γινόμενα που περιλαμβάνονται 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 στην αρχική συνάρτηση, καταλήγουμε εύκολα στην ισοδύναμη συνάρτηση F(x,y,z) = (x + z)(x′ + y)(y + z), η οποία έχει πλέον 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 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 σημαντικός στόχος των σχεδιαστών είναι η μείωση του κόστους υλοποίησης, μια ένδειξη του οποίου είναι το άθροισμα του αριθμού των λογικών πυλών και του αριθμού 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:

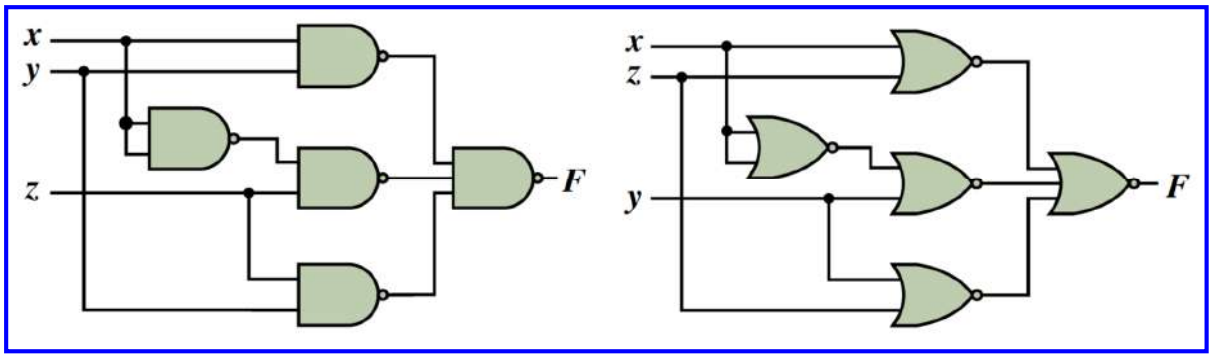
$$F(x,y,z) = [(xy + x'z + yz)']' = [(xy)'(x'z)'(yz)']'$$

- 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



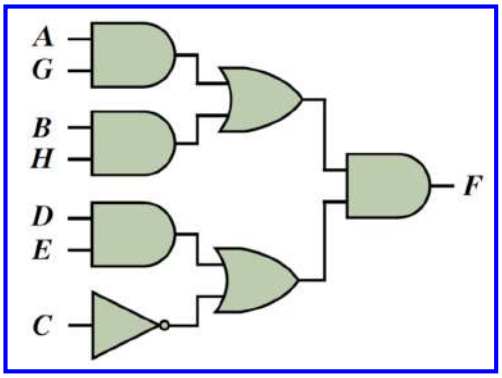
- Συμπεραίνουμε ότι:
  - ✓ οποιοδήποτε λογικό κύκλωμα διάταξης AND-OR μπορεί να μετατραπεί σε λογικό κύκλωμα διάταξης NAND-NAND με την ίδια τοπολογία,
  - ✓ οποιοδήποτε λογικό κύκλωμα διάταξης OR-AND μπορεί να μετατραπεί σε λογικό κύκλωμα διάταξης NOR-NOR ίδιας τοπολογίας.

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.

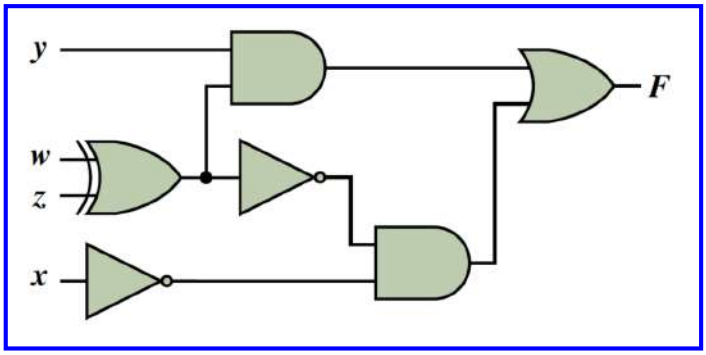
Λογικά κυκλώματα πολλών επιπέδων

- Σε αρκετές περιπτώσεις, η χρήση λογικών κυκλωμάτων με περισσότερα από δύο επίπεδα πυλών μπορεί να οδηγήσει και σε μικρότερο πλήθος πυλών.
- Η μέθοδος της παραγοντοποίησης (αξίωμα επιμεριστικότητας) μπορεί να μας προσφέρει τη δυνατότητα να χρησιμοποιήσουμε λογικές πύλες XOR ή XNOR, ώστε να μειώσουμε το κόστος της υλοποίησης μιας λογικής συνάρτησης.
- Επίσης, μία ή περισσότερες πύλες ή, στη γενικότερη περίπτωση, ένα ή περισσότερα υποκυκλώματα μπορούν να υλοποιούν συναρτήσεις που χρησιμοποιούνται περισσότερες από μία φορές στο συνολικό λογικό κύκλωμα.
- Η συνάρτηση F(x,y,z,w) = x'z'w' + yz'w + x'zw + yzw' απαιτεί για την υλοποίησή της 3 αντιστροφείς για την παραγωγή των συμπληρωματικών μορφών των μεταβλητών εισόδου, 4 πύλες AND 3 εισόδων για την παραγωγή των λογικών γινομένων και μία πύλη OR 4 εισόδων για την παραγωγή του συνολικού λογικού αθροίσματος.
- Χρησιμοποιώντας τη μέθοδο της παραγοντοποίησης, μπορούμε να καταλήξουμε στην ακόλουθη ισοδύναμη λογική συνάρτηση:

$$F = x'z'w' + yz'w + x'zw + yzw' = x'(z'w' + zw) + y(z'w + zw')$$
$$= x'(z \oplus w)' + y(z \oplus w)$$

Λογικά κυκλώματα πολλών επιπέδων

- Η υλοποίηση της συνάρτησης που προέκυψε μπορεί να επιτευχθεί με 4 επίπεδα πυλών, οι είσοδοι των συμμετεχουσών πυλών δεν είναι σε καμία περίπτωση περισσότερες από δύο και η έξοδος της πύλης XOR χρησιμοποιείται τόσο στην υλοποίηση της λογικής συνάρτησης αποκλειστικού OR, όσο και στην υλοποίηση της λογικής ισοδυναμίας των μεταβλητών z και w, με τη συνδρομή ενός αντιστροφέα.



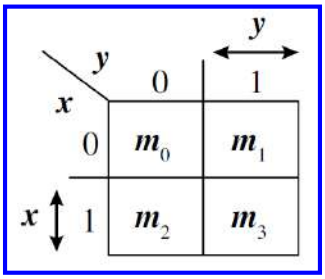
- Η υλοποίηση λογικών συναρτήσεων με πολλά επίπεδα πυλών περιορίζει τον αριθμό των εισόδων των πυλών ή προσφέρει τη δυνατότητα χρησιμοποίησης μικρότερου πλήθους λογικών πυλών, οδηγεί ωστόσο σε μεγαλύτερη καθυστέρηση απόκρισης των λογικών κυκλωμάτων, λόγω των πολλαπλών επιπέδων πυλών που παρεμβάλλονται μεταξύ των εισόδων και της εξόδου του κυκλώματος.

Ελαχιστοποίηση (απλοποίηση) λογικών συναρτήσεων

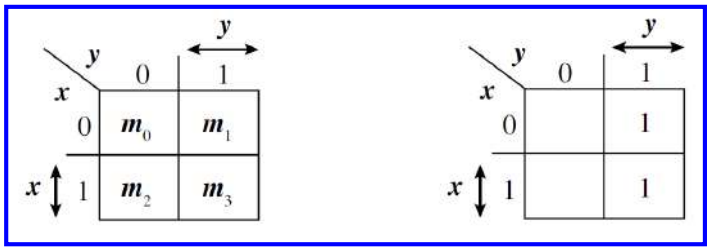
- Οι αλγεβρικοί μετασχηματισμοί που βασίζονται στα αξιώματα και τα θεωρήματα της άλγεβρας Boole μπορούν να οδηγήσουν σε απλοποιημένες μορφές λογικών συναρτήσεων, έτσι ώστε να μπορεί να επιτευχθεί η υλοποίησή τους με μικρότερο αριθμό λογικών πυλών.
- Ωστόσο, η χρήση αλγεβρικών μετασχηματισμών δεν είναι 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 specificintegrated 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 ελάχιστο όρο της συνάρτησης.
- Για την περιγραφή μιας συνάρτησης 2 μεταβλητών απαιτείται χάρτης Karnaugh με 4 τετράγωνα, δηλαδή όσοι είναι και οι ελάχιστοι όροι που μπορούν να συμμετέχουν στη συνάρτηση.



Περιγραφή λογικών συναρτήσεων με χάρτες Karnaugh

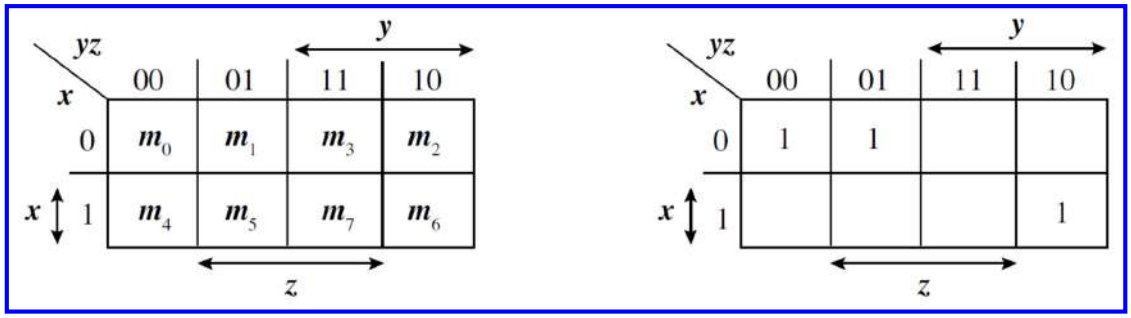


F(x,y) = x'y + xy

- 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 ελάχιστους όρους 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 ελάχιστους όρους που συμμετέχουν στη συνάρτηση.-

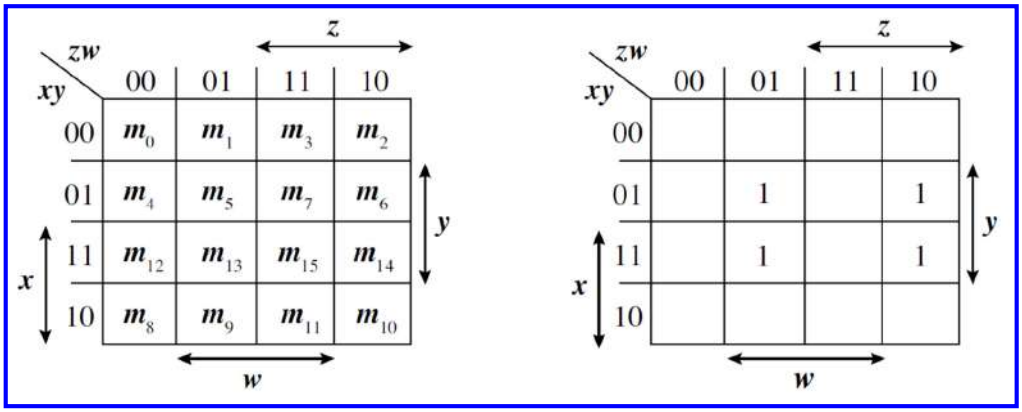
Περιγραφή λογικών συναρτήσεων με χάρτες Karnaugh

- Για την περιγραφή μιας λογικής συνάρτησης 3 μεταβλητών θα πρέπει να χρησιμοποιηθεί χάρτης Karnaugh οκτώ τετραγώνων, αφού τρεις μεταβλητές συνιστούν 8 ελάχιστους όρους.
- 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 ελάχιστος όρος 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:



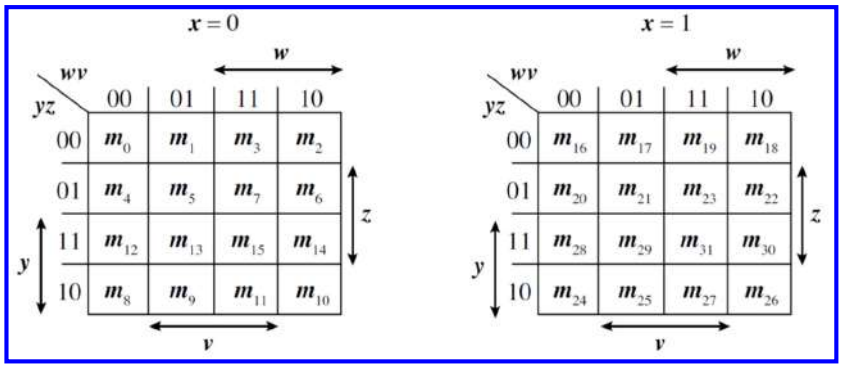
Περιγραφή λογικών συναρτήσεων με χάρτες Karnaugh

- Για την περιγραφή μιας λογικής συνάρτησης 4 μεταβλητών χρησιμοποιείται χάρτης 16 τετραγώνων.
- 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 ελάχιστος όρος που αντιστοιχεί σε κάθε τετράγωνο λαμβάνει λογική τιμή 1.
- $F(x,y,z,w) = \Sigma(5, 6, 13, 14)$



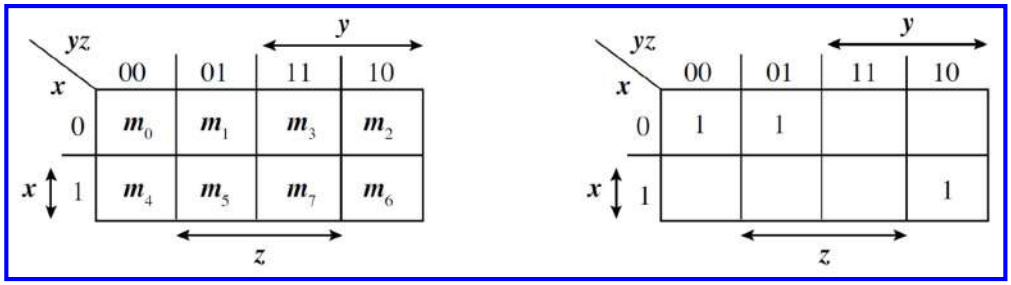
Περιγραφή λογικών συναρτήσεων με χάρτες Karnaugh

- 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 ελάχιστους όρους με τη μεταβλητή x' και τα τετράγωνα του 2ου χάρτη αντιστοιχούν στους ελάχιστους όρους με την x.



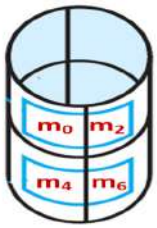
Ελαχιστοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

- Οι ελάχιστοι όροι 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 ελάχιστο όρο που αντιστοιχεί στο ένα τετράγωνο και με την κανονική της μορφή στον ελάχιστο όρο που αντιστοιχεί στο γειτονικό του τετράγωνο.
- Στον χάρτη 3 μεταβλητών, οι ελάχιστοι όροι 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: $m_0 + m_1 = x'y'z' + x'y'z = x'y'(z' + z) = x'y'$. Αυτό σημαίνει ότι κατά τη λογική άθροιση ελαχίστων όρων που αντιστοιχούν σε γειτονικά τετράγωνα, απαλείφεται η μεταβλητή κατά την οποία αυτοί διαφέρουν.



Ελαχιστοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

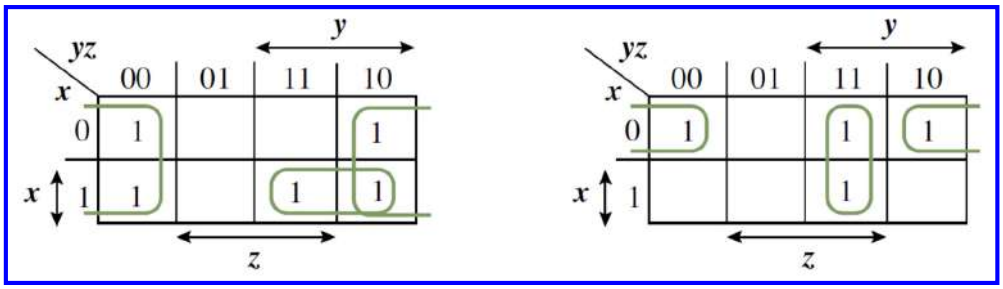
- Στο χάρτη Karnaugh 3 μεταβλητών εκτός των γειτονικών τετραγώνων συναντώνται και άλλα τετράγωνα των οποίων οι αντίστοιχοι ελαχιστόροι διαφέρουν κατά μία μεταβλητή.
- Τέτοια ζεύγη τετραγώνων είναι εκείνα που αντιστοιχούν στους ελαχιστόρους m_0, m_2 και στους m_4, m_6.
- Τα τετράγωνα καθενός από τα ζεύγη 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

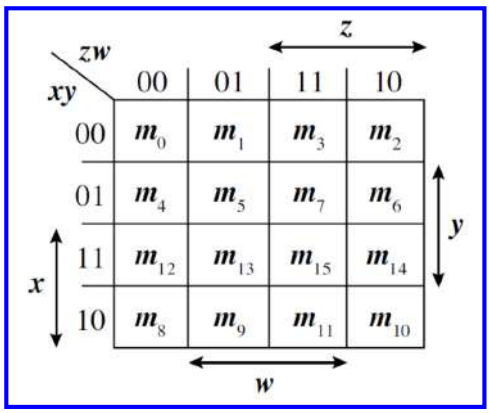
- Το πλήθος των τετραγώνων μιας ομάδας θα πρέπει να ισούται με δύναμη του 2 και κάθε ομάδα είναι επιθυμητό να περιλαμβάνει τον μέγιστο δυνατό αριθμό τετραγώνων που περιέχουν μονάδα.
- Βασικός στόχος της διαδικασίας ελαχιστοποίησης είναι η συμπερίληψη όλων των τετραγώνων που περιέχουν μονάδες σε μία τουλάχιστον 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 $F_2(x,y,z) = x'y'z' + xyz + x'yz + x'yz'$:

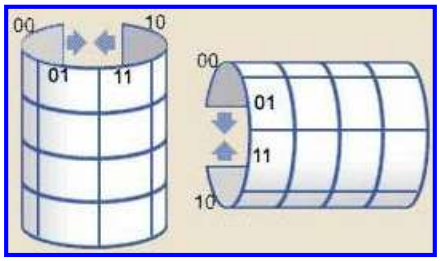


$F_1(x,y,z) = z' + xy$ $F_2(x,y,z) = yz + x'z'$

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.





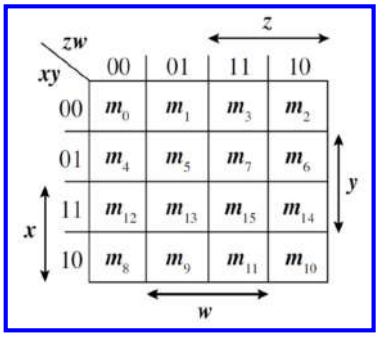
Ελαχιστοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

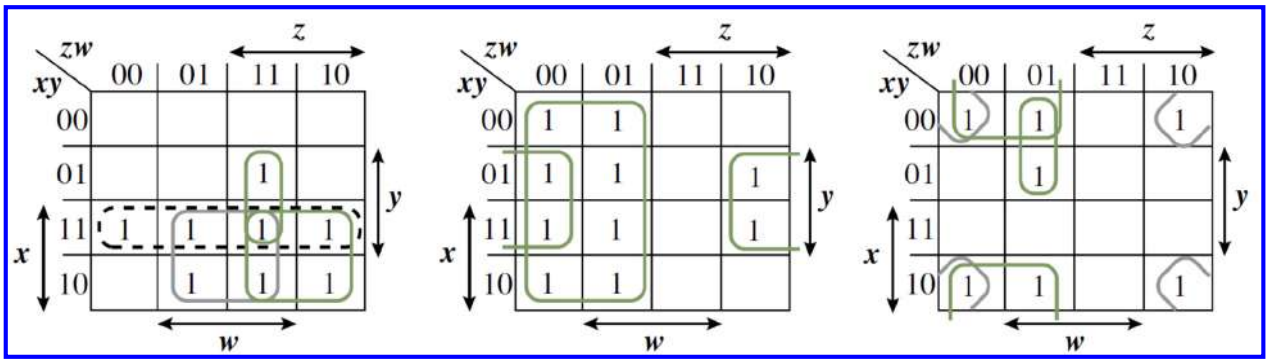
Ελαχιστοποίηση των συναρτήσεων:

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

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

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



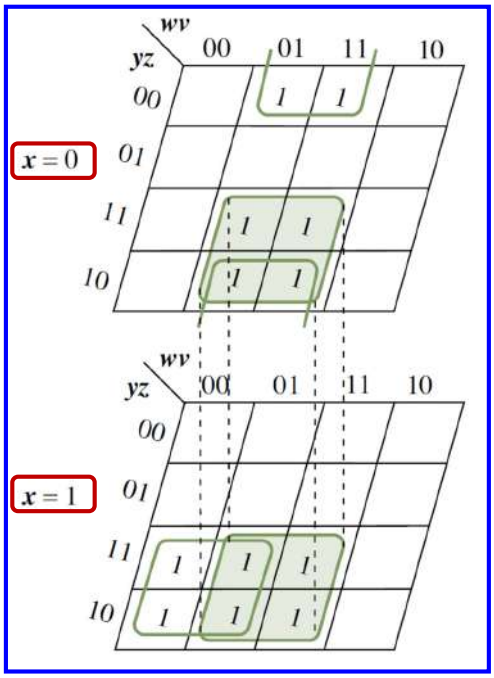


Ελαχιστοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

- 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 ελάχιστο όρο της λογικής συνάρτησης.
- Για κάθε τετράγωνο ενός χάρτη Karnaugh n μεταβλητών υπάρχουν n γειτονικά τετράγωνα και κάθε ζεύγος τετραγώνων αντιστοιχεί σε ελάχιστους όρους 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 ελάχιστο όρο 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.

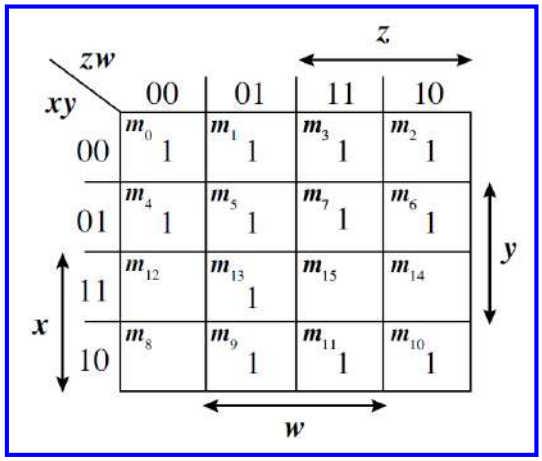
Χάρτης Karnaugh συνάρτησης πρότυπης μορφής

- Για το σχηματισμό του χάρτη Karnaugh μιας συνάρτησης που είναι σε πρότυπη μορφή αθροίσματος γινομένων (περιλαμβάνει και γινόμενα στα οποία δε συμμετέχουν όλες οι μεταβλητές), λαμβάνουμε υπόψη ότι ότι κάθε γινόμενο αντιστοιχεί σε ένα, δύο ή περισσότερα τετράγωνα του χάρτη, ανάλογα με τις μεταβλητές που συμμετέχουν σε αυτό.
- Σημειώνοντας, μονάδες στα κατάλληλα τετράγωνα, μπορούμε εύκολα να εξαγάγουμε την κανονική μορφή αθροίσματος ελαχίστων όρων της συνάρτησης.
- $F(x,y,z,w) = xz'w + y'z + x' = \Sigma(0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13)$

$m_9 + m_{13}$

$m_2 + m_3 + m_{10} + m_{11}$

$m_0 + m_1 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7$



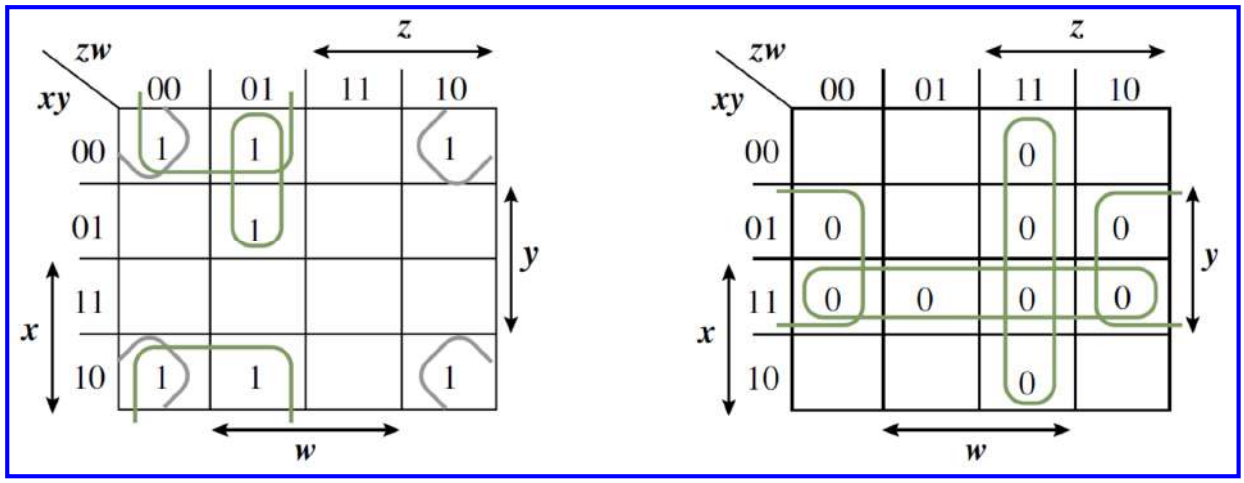
Υλοποίηση ελαχιστοποιημένων συναρτήσεων

- 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 ελάχιστους όρους που δεν περιλαμβάνονται στη συνάρτηση.
- Το άθροισμα των ελαχίστων όρων 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

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

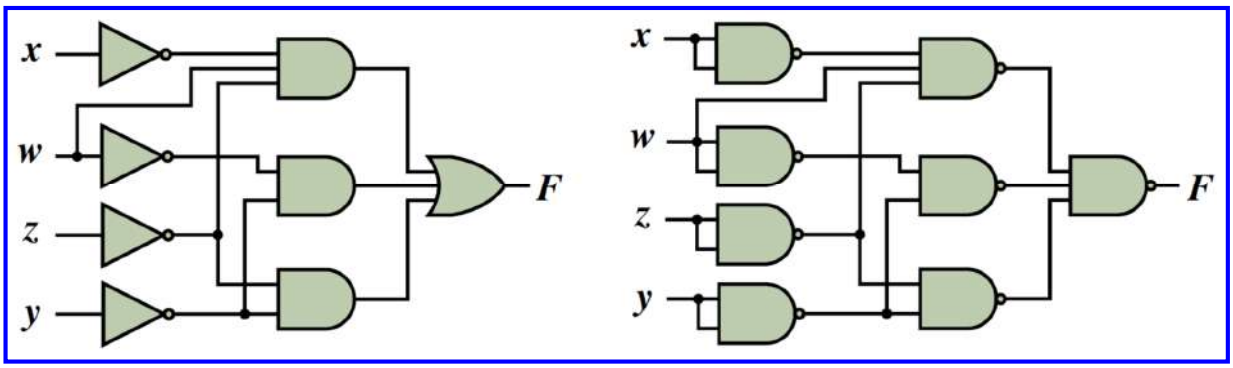
$F'(x,y,z,w) = xy + zw + yw'$



Υλοποίηση ελαχιστοποιημένων συναρτήσεων

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

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



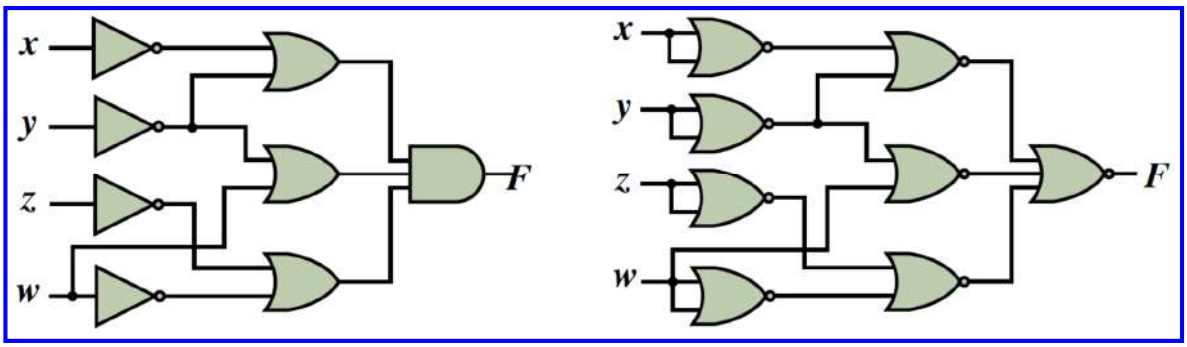
AND-OR NAND-NAND

Υλοποίηση ελαχιστοποιημένων συναρτήσεων

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

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

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



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 ελάχιστοι όροι 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).

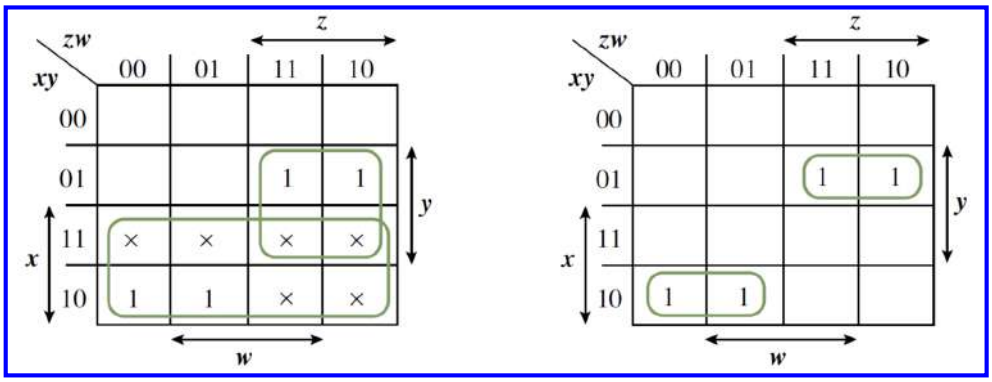
Μερικώς καθορισμένες συναρτήσεις

- 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 $F(x,y,z) = \Sigma(1, 2, 5) + d(0, 3, 4)$ ή $F(x,y,z) = \Sigma(1, 2, 5) + \times(0, 3, 4)$, περιλαμβάνει τους ελάχιστους όρους m_1, m_2, m_5 και ως αδιάφορους όρους τα γινόμενα x'y'z', x'yz, xy'z'.

Μερικώς καθορισμένες συναρτήσεις

- Σχεδίαση λογικού κυκλώματος που λαμβάνει στις εισόδους του x, y, z, w, 4 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 ελάχιστους όρους 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 στην 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 & ουσιώδεις 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 ελάχιστους όρους 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.

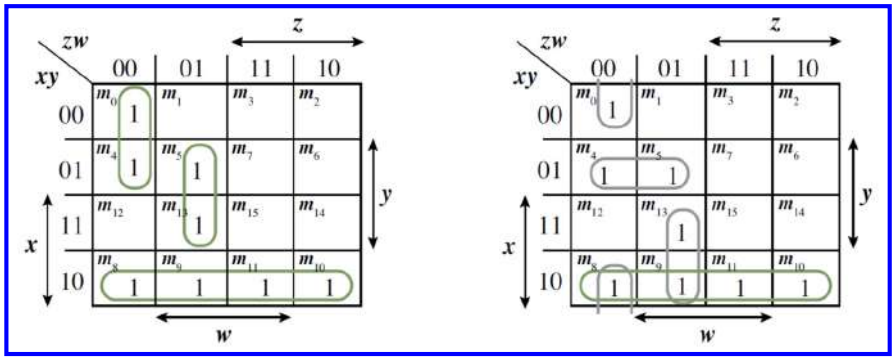
Πρώτοι & ουσιώδεις πρώτοι συνεπαγωγοί συνάρτησης

- Οι συνεπαγωγοί αυτοί εκφράζουν ή καλύπτουν τους ελάχιστους όρους της 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 ελάχιστους όρους, οι οποίοι δεν εκφράζονται από άλλον πρώτο συνεπαγωγό.
- Οι πρώτοι συνεπαγωγοί, οι οποίοι εκφράζουν έναν τουλάχιστον ελάχιστο όρο μιας συνάρτησης που δεν εκφράζεται από άλλον πρώτο συνεπαγωγό, αναφέρονται ως ουσιώδεις (ή βασικοί) πρώτοι συνεπαγωγοί (essential prime implicants) της συνάρτησης.
- Μετά την επιλογή των ουσιωδών πρώτων συνεπαγωγών, θα πρέπει να επιλέγονται οι πρώτοι συνεπαγωγοί που εκφράζουν ή καλύπτουν τους υπόλοιπους ελάχιστους όρους, 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 & ουσιώδεις πρώτοι συνεπαγωγοί συνάρτησης

F(x,y,z,w) =

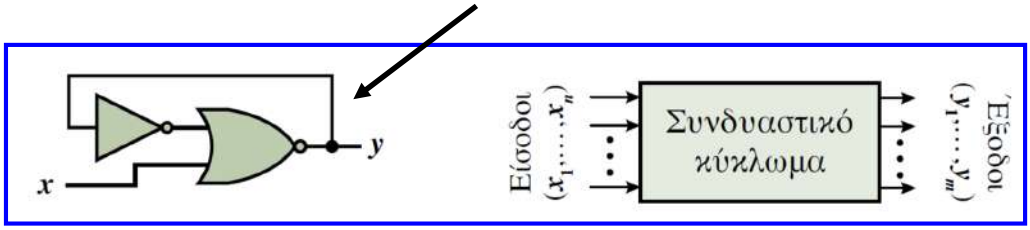
Σ(0, 4, 5, 8, 9, 10, 11, 13)



- 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 ουσιώδης πρώτος συνεπαγωγός, αφού είναι ο μοναδικός που καλύπτει τους ελάχιστους όρους m_{10} και m_{11}.
- Θα πρέπει να επιλέξουμε από τους 5 πρώτους συνεπαγωγούς, εκείνους που εκφράζουν όλους τους ελάχιστους όρους 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).
- 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.

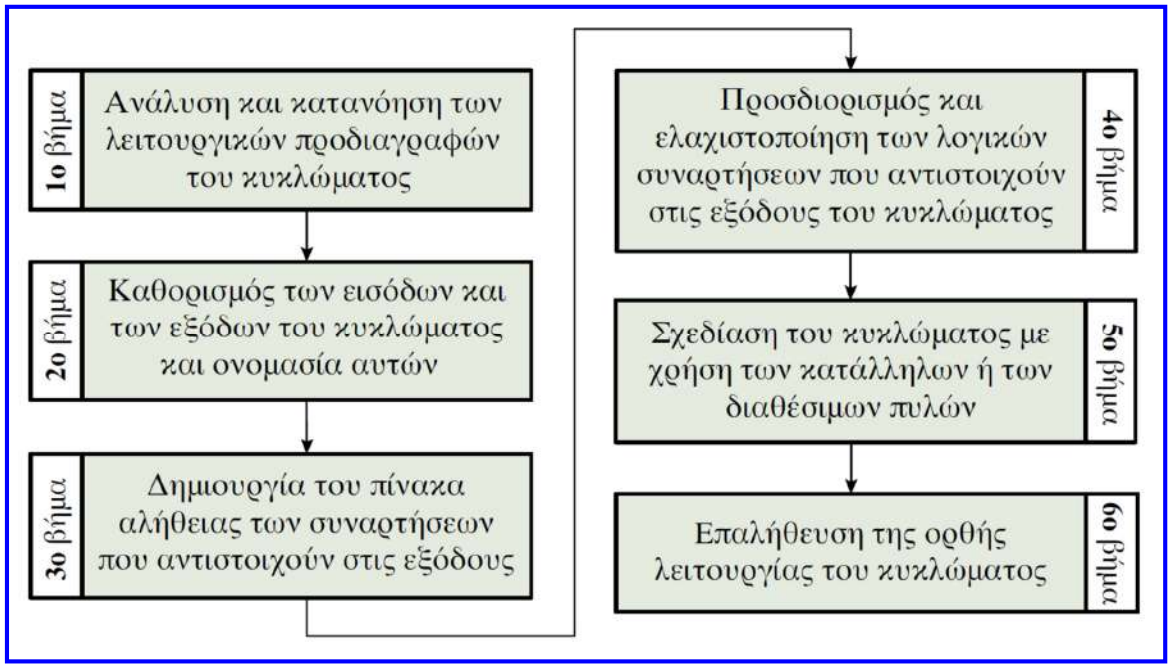
Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

- 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 A θα ενεργοποιεί το συναγερμό της πινακοθήκης και η B θα ενεργοποιεί τη φωτεινή ένδειξη στο υποκατάστημα της εταιρείας φύλαξης.
- Θεωρούμε, επίσης, ότι οι έξοδοι A και B 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.

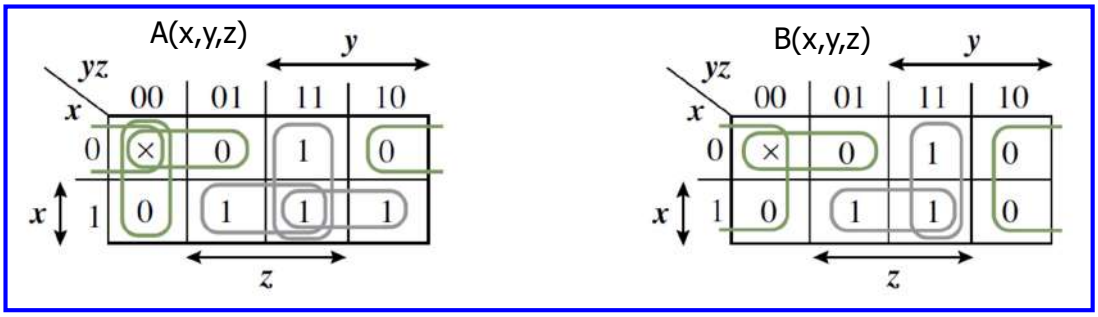
$$A(x,y,z) = x'yz + xy'z + xyz' + xyz$$
$$B(x,y,z) = x'yz + xy'z + xyz$$

- 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

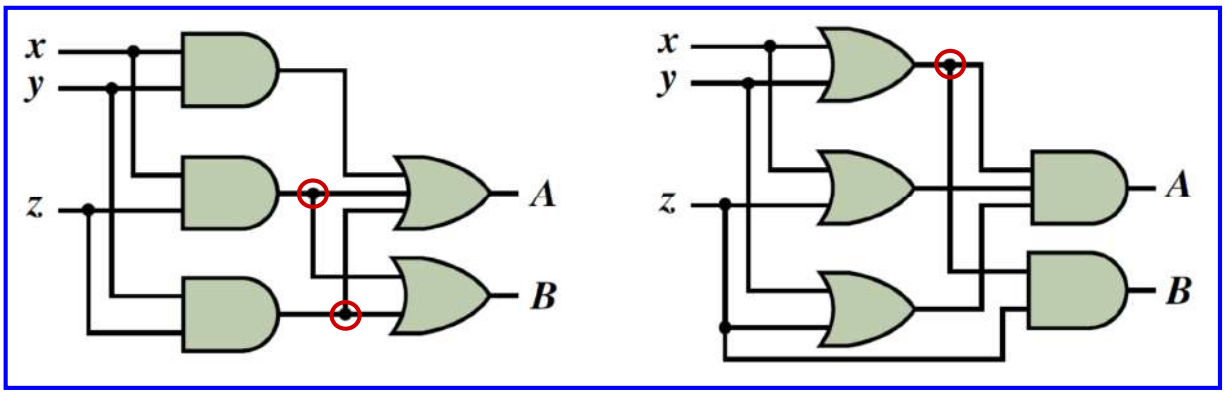


$$A(x,y,z) = xy + xz + yz$$
$$B(x,y,z) = xz + yz$$

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

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

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



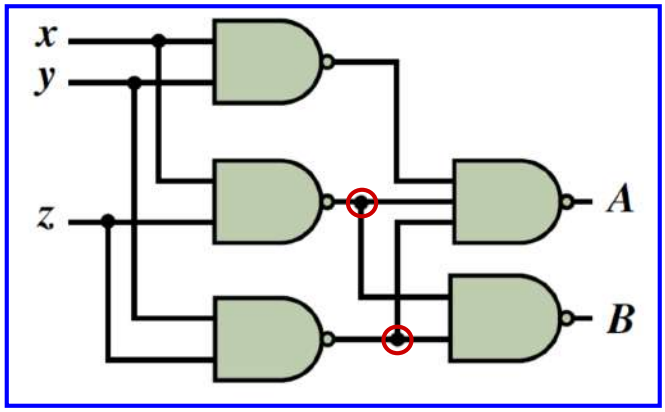
Κατά τη σύνθεση ενός συνδυαστικού κυκλώματος με πολλαπλές εξόδους, ενδέχεται οι λογικές συναρτήσεις δύο ή περισσότερων εξόδων να περιλαμβάνουν ορισμένους κοινούς όρους (λογικά γινόμενα ή αθροίσματα). Στην περίπτωση αυτή, οι κοινοί όροι υλοποιούνται μία φορά και επαναχρησιμοποιούνται για την παραγωγή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

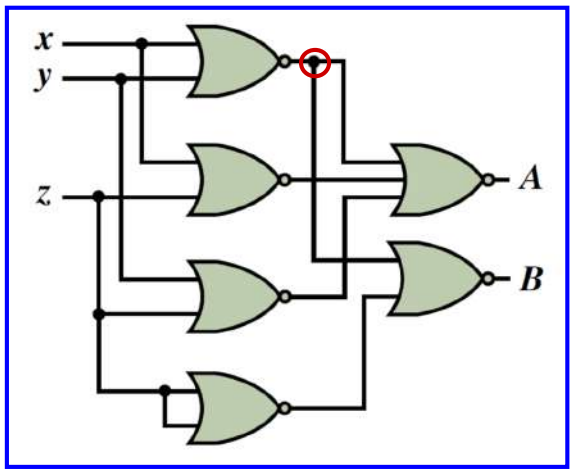


Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

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:

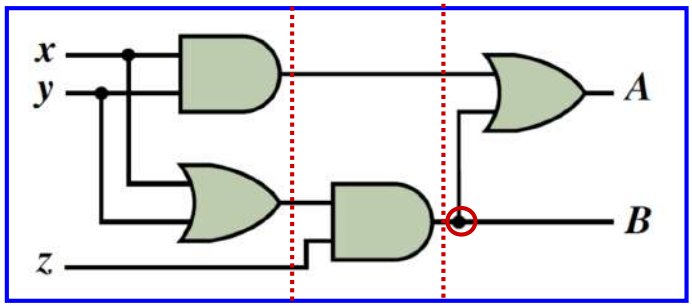


Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

- 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:

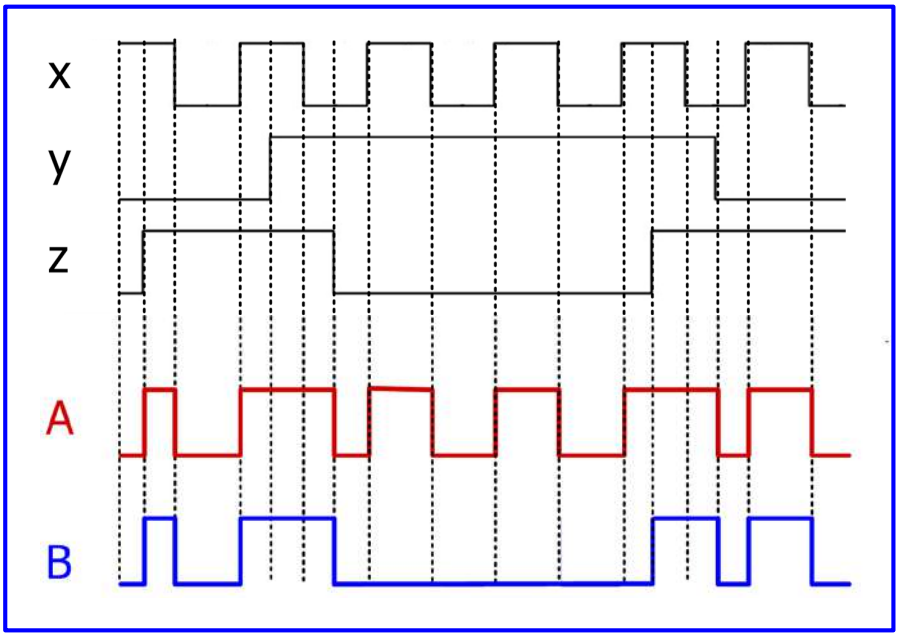
  $$A(x,y,z) = (x + y)z + xy \quad \text{και} \quad B(x,y,z) = (x + y)z$$

  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.



Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Χρονικά διαγράμματα εισόδων και εξόδων
(θεωρώντας αμελητέα καθυστέρηση πυλών)



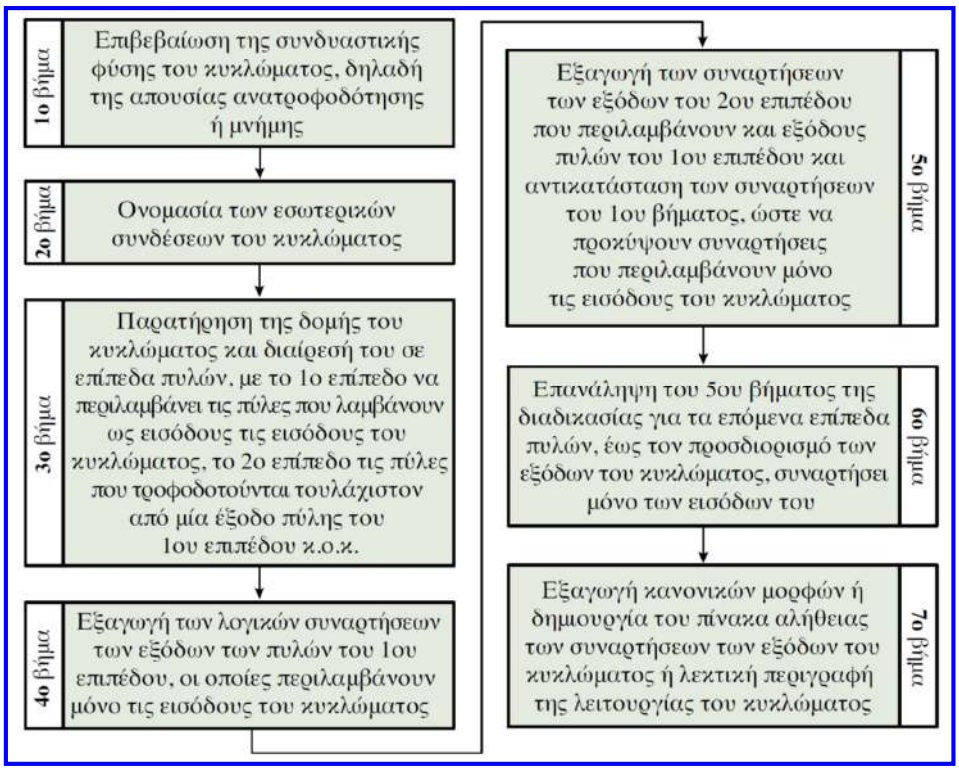
Ανάλυση συνδυαστικών κυκλωμάτων

- 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.

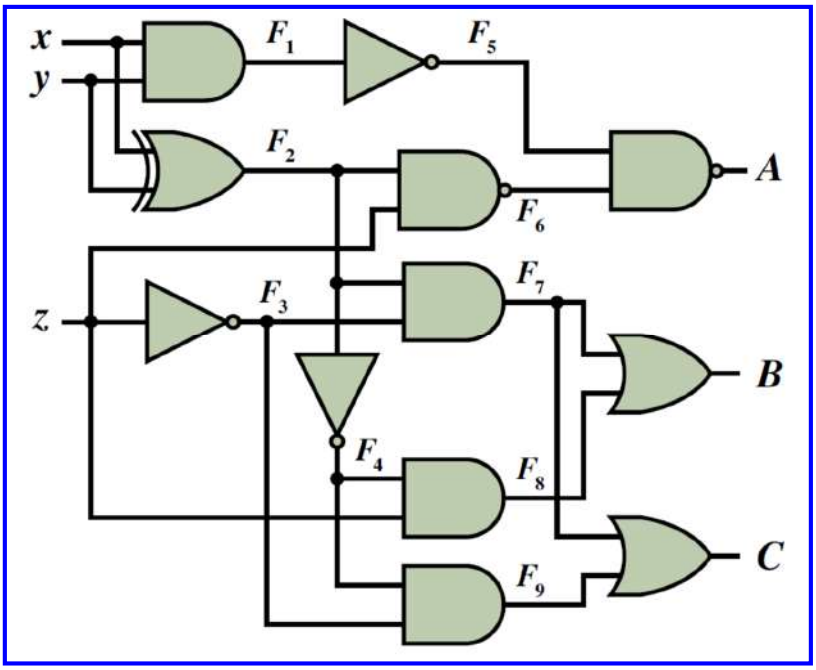
- Προκειμένου, να διαπιστωθεί η λειτουργική ισοδυναμία δύο ή περισσότερων συνδυαστικών κυκλωμάτων, θα πρέπει αυτά να αναλυθούν, ώστε να διαπιστωθεί εάν αντιστοιχούν στον ίδιο πίνακα αλήθειας ή στις ίδιες κανονικές αλγεβρικές μορφές

- 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 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 ελαχίστων όρων 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.

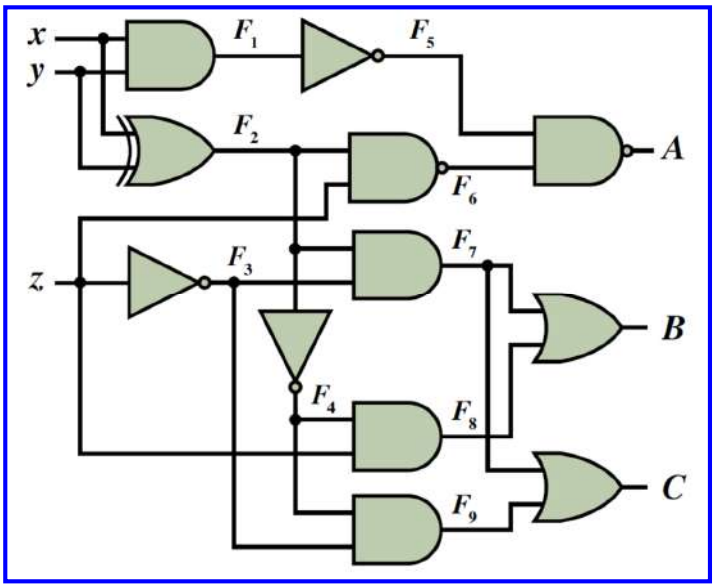
Ανάλυση συνδυαστικών κυκλωμάτων

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

Καθυστέρηση συνδυαστικών κυκλωμάτων

- Τα σήματα περνόντας από τις λογικές πύλες υφίστανται αναπόφευκτες καθυστερήσεις.
- Οι καθυστερήσεις αυτές συνιστούν την καθυστέρηση διάδοση του κυκλώματος.
- Θεώρηση: όλες οι λογικές πύλες παρουσιάζουν όμοια καθυστέρηση D.

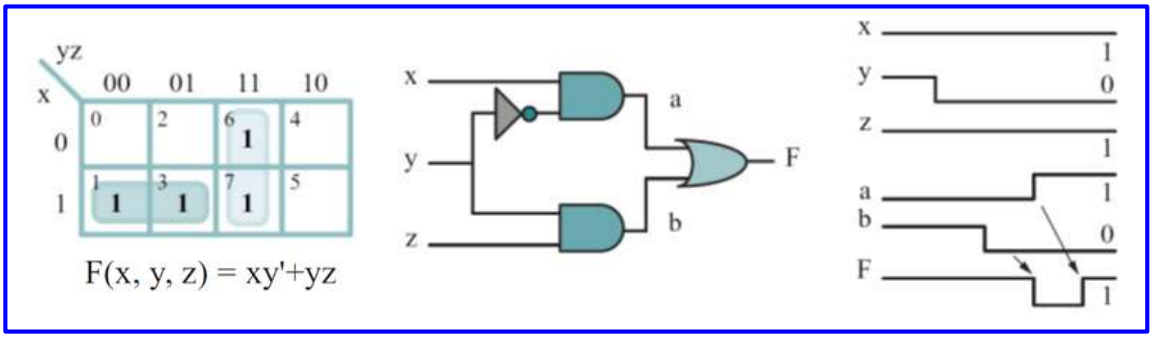


από την είσοδο	στην έξοδο	μέγιστη καθυστέρηση
x	A	3xD
x	B	4xD
x	C	4xD
y	A	3xD
y	B	4xD
y	C	4xD
z	A	2xD
z	B	3xD
z	C	3xD

Η μέγιστη καθυστέρηση του κυκλώματος είναι 4 x D και η μέγιστη συχνότητας λειτουργίας f = 1 / (4 x D)

Κίνδυνοι (hazards) στα συνδυαστικά κυκλώματα

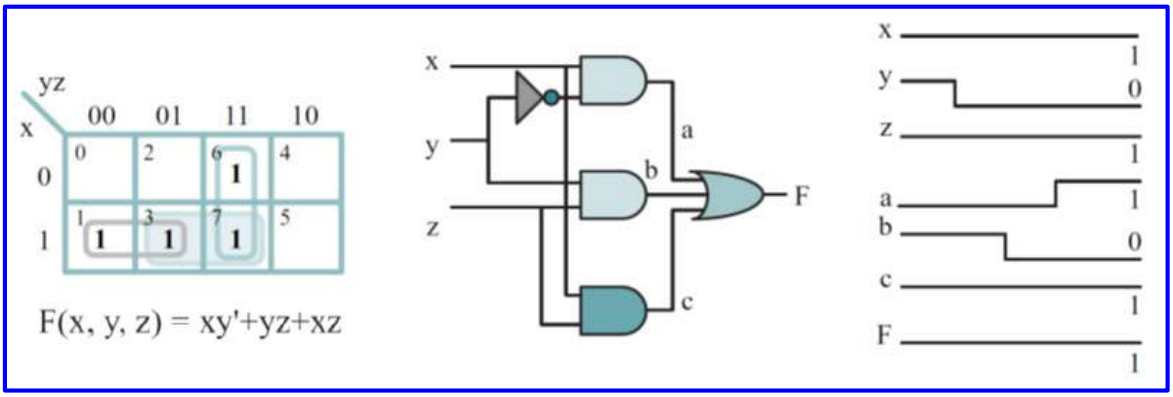
- Οι κίνδυνοι (hazards) σε ένα λογικό κύκλωμα προκαλούνται όταν ένα σήμα διχάζεται σε κάποιον κόμβο του κυκλώματος και κατόπιν, ακολουθώντας διαφορετικές διαδρομές μέσα στο κύκλωμα, τα δύο παράγωγα σήματά του καταλήγουν στην ίδια λογική πύλη.



- Στο σήμα y εισάγεται καθυστέρηση δύο πυλών στην διαδρομή a, αλλά μόνο μιας πύλης στην διαδρομή b.
- Ο απρόβλεπτος παλμός 101 που δημιουργείται στην έξοδο F (η οποία θα πρέπει να είναι συνεχώς 1 για τις δεδομένες τιμές εισόδων), αναφέρεται ως κίνδυνος (hazard).
- Τέτοιοι κίνδυνοι συμβαίνουν όταν η λογική συνάρτηση του κυκλώματος περιλαμβάνει γειτονικούς ελάχιστους όρους (m_3, m_7), οι οποίοι όμως ανήκουν σε διαφορετική ομάδα του χάρτη Karnaugh.

Κίνδυνοι (hazards) στα συνδυαστικά κυκλώματα

- Η συμπεριφορά του κυκλώματος διορθώνεται με προσθήκη μιας πλεονάζουσας ομάδας μονάδων στο χάρτη Karnaugh, η οποία καλύπτει τους γειτονικούς ελάχιστους όρους που ανήκουν σε διαφορετικές ομάδες.



- Με την προσθήκη του όρου xz που καλύπτει τους γειτονικούς ελάχιστους όρους m_3, m_7 που ανήκουν σε διαφορετικές ομάδες, δημιουργείται ένα λογικά ισοδύναμο κύκλωμα, το οποίο δεν είναι μεν το ελάχιστο δυνατό, αλλά συμπεριφέρεται καλύτερα, αφού αποφεύγεται ο κίνδυνος (hazard).
- Οι κίνδυνοι δεν είναι τόσο κρίσιμοι στα συνδυαστικά κυκλώματα, αφού μετά την πάροδο μικρού χρόνου, η έξοδός τους σταθεροποιείται στην ορθή λογική κατάσταση. Ωστόσο, μπορούν να αποτελέσουν πρόβλημα για τα ακολουθιακά κυκλώματα.

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.

Αριθμητικά συνδυαστικά κυκλώματα

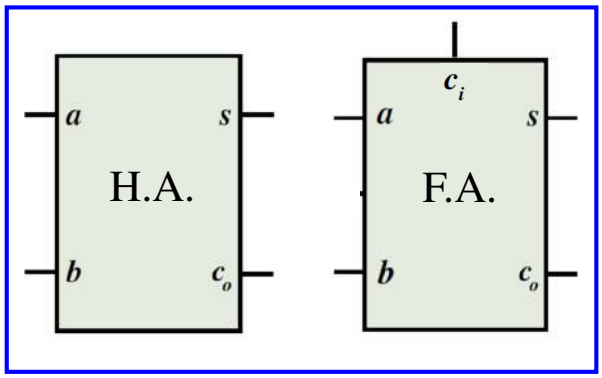
- Στα ψηφιακά συστήματα χρησιμοποιούνται ευρέως συνδυαστικά κυκλώματα που επιτελούν βασικές αριθμητικές λειτουργίες (πρόσθεση, αφαίρεση, σύγκριση δυαδικών αριθμών κ.ά.).
- Η πρόσθεση δύο δυαδικών ψηφίων υλοποιείται από ένα συνδυαστικό κύκλωμα δύο εισόδων, δηλαδή των ψηφίων που προστίθενται, και δύο εξόδων, 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, H.A.), ενώ το δεύτερο ως πλήρης αθροιστής (full adder, F.A.).

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

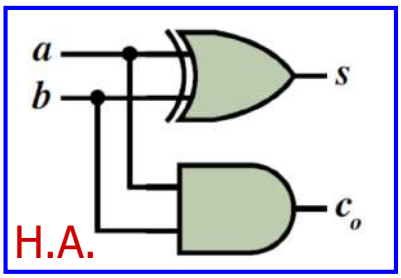


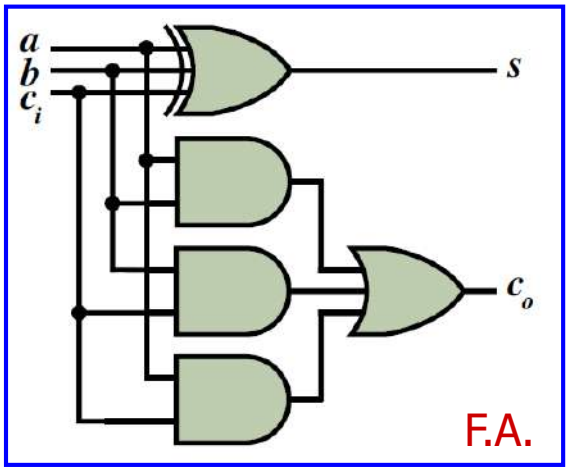
$$s = a'b'c_i + a'bc_i' + ab'c_i' + abc_i = (ab' + a'b)c_i' + (ab + a'b')c_i = a \oplus b \oplus c_i$$

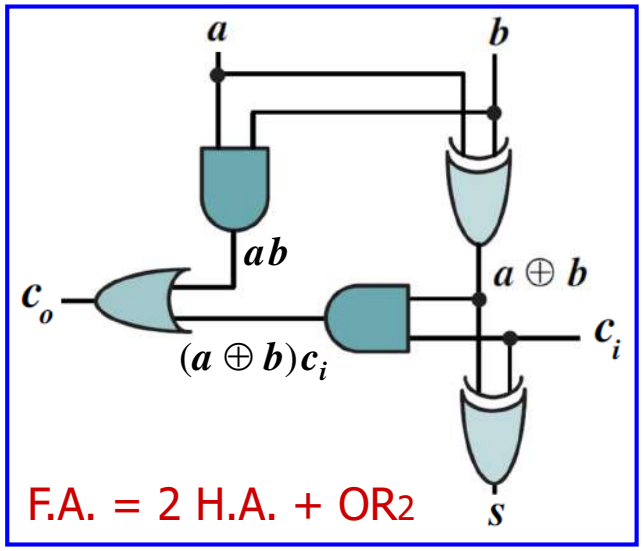
$$c_o = a'bc_i + ab'c_i + abc_i' + abc_i = a'bc_i + ab'c_i + abc_i' + abc_i + abc_i + abc_i$$
$$= ab(c_i + c_i') + bc_i(a + a') + ac_i(b + b') = ab + bc_i + ac_i$$

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

Ημιαθροιστής και πλήρης αθροιστής





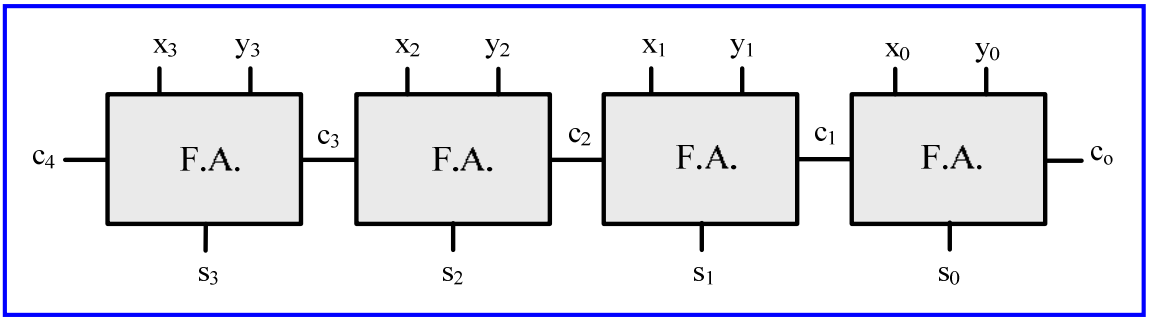


Παράλληλος αθροιστής

- Χρησιμοποιώντας ως δομικό στοιχείο τον πλήρη αθροιστή, μπορούμε να συνθέσουμε συνδυαστικά κυκλώματα παράλληλων αθροιστών για αριθμούς με περισσότερα ψηφία.
- Το αποτέλεσμα της πρόσθεσης 2 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 2 δυαδικών αριθμών n ψηφίων προκύπτει εύκολα, εάν συνδέσουμε n πλήρεις αθροιστές.
- Κάθε πλήρης αθροιστής δέχεται ένα ζεύγος ψηφίων των 2 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 2 αριθμών λαμβάνεται στην έξοδο 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

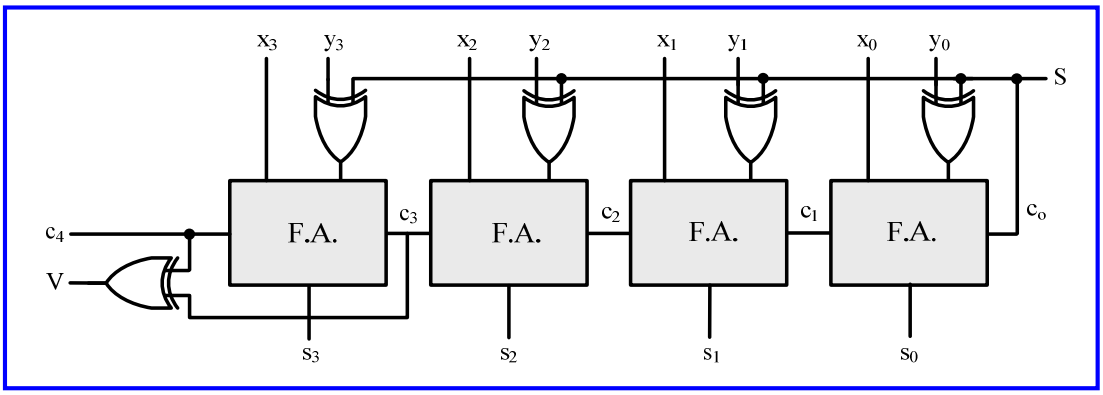
Παράλληλη πρόσθεση τετραψήφιων δυαδικών αριθμών



Παράλληλος αθροιστής / αφαιρέτης

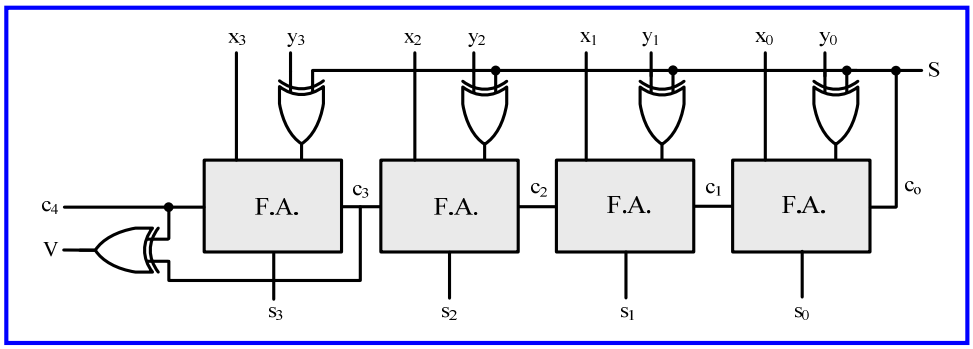
- Η αφαίρεση δυαδικών αριθμών ανάγεται σε πρόσθεση του μειωτέου με τον αντίθετο αριθμό του αφαιρετέου, ο οποίος μπορεί να παρασταθεί με το συμπλήρωμα ως προς 2 του αφαιρετέου.
- Το συμπλήρωμα ως προς 2 ενός δυαδικού αριθμού προκύπτει με πρόσθεση μιας μονάδας στο συμπλήρωμά του ως προς 1 και ότι το συμπλήρωμα 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 2 εισόδων παράγει στην έξοδό της τιμή 0, όταν οι είσοδοι λαμβάνουν την ίδια τιμή, και τιμή 1, όταν οι λογικές τιμές των εισόδων είναι διαφορετικές.
- 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 S = 1, οι πλήρεις αθροιστές τροφοδοτούνται με το συμπλήρωμα των ψηφίων του αριθμού Y (δηλαδή με τον Y′) 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 Y και κατά συνέπεια για την εκτέλεση της αφαίρεσης X – Y.
- Όταν S = 0, οι πλήρεις αθροιστές τροφοδοτούνται με τα ψηφία του αριθμού Y και η λειτουργία του κυκλώματος είναι όμοια με εκείνη του παράλληλου αθροιστή.

Παράλληλος αθροιστής / αφαιρέτης

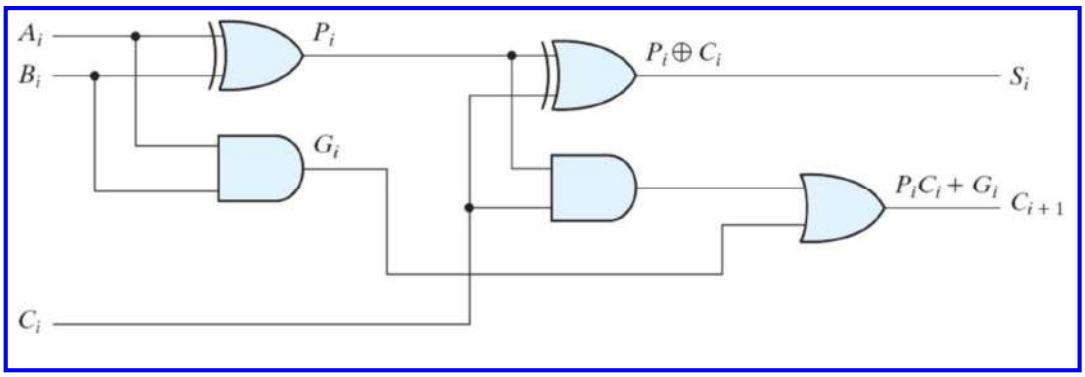


- Η έξοδος V υποδεικνύει την υπερχείλιση κατά την πρόσθεση ή αφαίρεση, δηλ. τη μετατόπιση του ψηφίου-προσήμου του αποτελέσματος από την αναμενόμενη (πιο σημαντική) θέση.
- Υπερχείλιση κατά την πρόσθεση δύο προσημασμένων αριθμών συμβαίνει όταν οι αριθμοί είναι ομόσημοι και το αποτέλεσμα που προκύπτει έχει διαφορετικό πρόσημο από αυτούς.
- Αφού η αφαίρεση ανάγεται σε πρόσθεση του μειωτέου με τον αντίθετο του αφαιρετέου, υπερχείλιση κατά την αφαίρεση 2 προσημασμένων αριθμών (X – Y) συμβαίνει όταν ο X είναι θετικός, ο Y είναι αρνητικός και το αποτέλεσμα αρνητικό, καθώς και όταν ο X είναι αρνητικός, ο Y 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 εξαγωγής του κρατουμένου ενός πλήρους αθροιστή εισάγει καθυστέρηση 2 πυλών ή 3 πυλών, ανάλογα με την υλοποίηση του.
- Έτσι, η καθυστέρηση διάδοσης κρατουμένου ενός παράλληλου αθροιστή n ψηφίων με την καθυστέρηση 2 × n ή 3 × n 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

$C_1 = G_0 + P_0C_0$

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

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

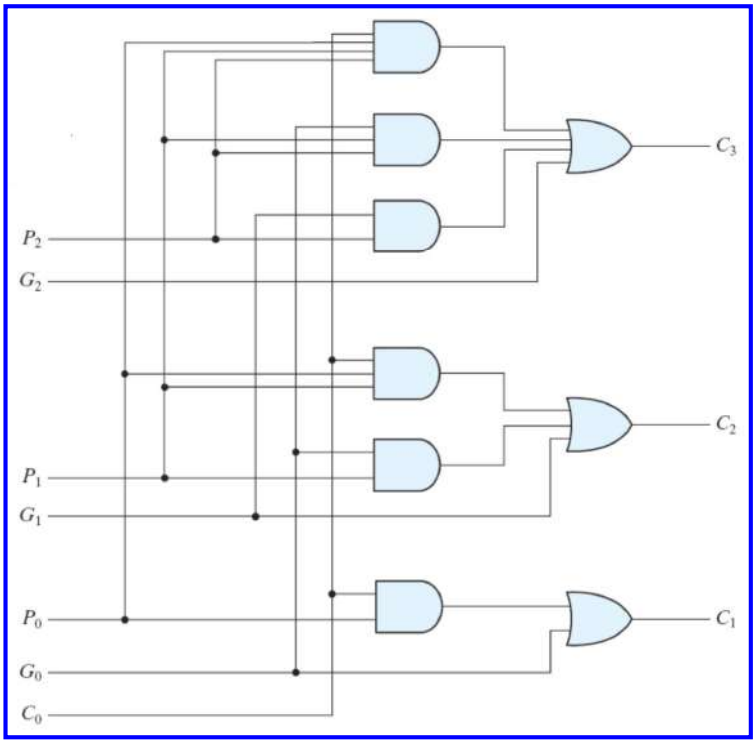
Γεννήτρια πρόβλεψης κρατουμένου

$C_1 = G_0 + P_0C_0$

$C_2 = G_1 + P_1G_0 + P_1P_0C_0$

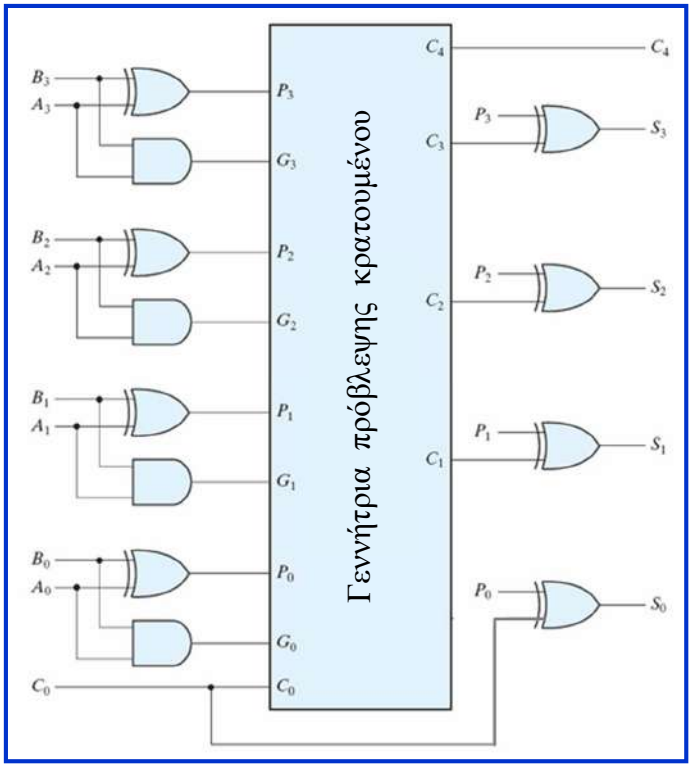
$C_3 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$

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



Αθροιστής πρόβλεψης κρατουμένου

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



$S_i = P_i \oplus C_i$

Συγκριτές

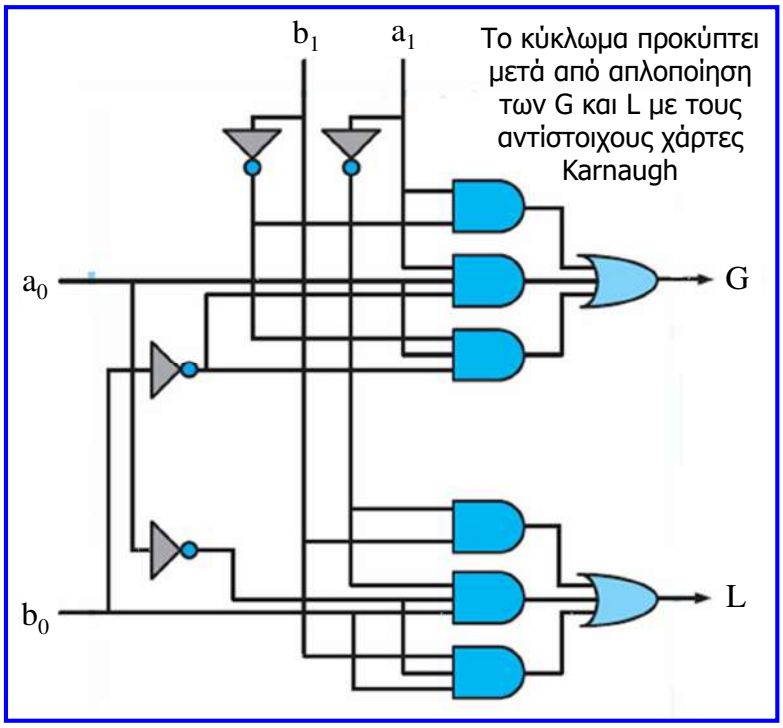
- Ο συγκριτής είναι ένα κύκλωμα που συγκρίνει δύο αριθμούς (X, Y) και αποτυπώνει τη σχέση τους (X < Y, X = Y, X > Y) σε δύο εξόδους (G, L):

G L	Σχέση X,Y
0 0	X=Y
0 1	X<Y
1 0	X>Y

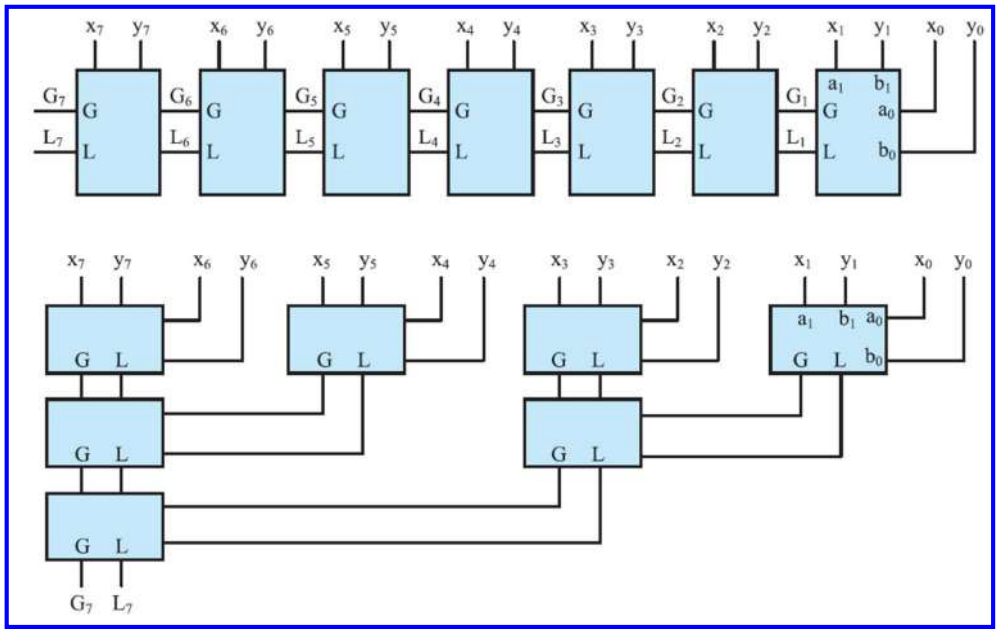
- Για τη σύγκριση του μεγέθους μη προσημασμένων δυαδικών αριθμών, χρησιμοποιούνται συνδυαστικά κυκλώματα που βασίζονται στον έλεγχο των ψηφίων τους ανά ζεύγη.
- Η σύγκριση δύο προσημασμένων δυαδικών αριθμών μπορεί να γίνει μέσω ελέγχου του αποτελέσματος της αφαίρεσής τους.
- Μια τεχνική σύνθεσης συγκριτών μεγέθους δυαδικών αριθμών είναι η σχεδίαση ενός βασικού συγκριτή αριθμών με δύο δυαδικά ψηφία και η χρησιμοποίησή του για σύνθεση συγκριτών για αριθμούς με περισσότερα δυαδικά ψηφία, ξεκινώντας από το ζεύγος ψηφίων της λιγότερο σημαντικής θέσης.

Συγκριτής διψήφιων δυαδικών αριθμών

a_1	b_1	a_0	b_0	G	L
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	0	0



Σειριακός και παράλληλος συγκριτής



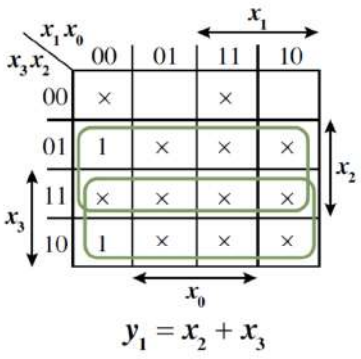
Ο σειριακός συγκριτής n-ψήφιων αριθμών αποτελείται από n – 1 διψήφιους συγκριτές και έχει καθυστέρηση ανάλογη του n – 1. Ο παράλληλος συγκριτής, αν και αποτελείται από το ίδιο πλήθος διψήφιων συγκριτών, έχει μικρότερη καθυστέρηση, ανάλογη του $\log_2 n$.

Κωδικοποιητές

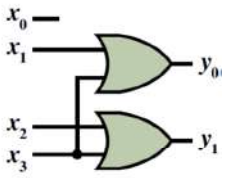
- Τα ψηφιακά συστήματα έχουν τη δυνατότητα να χειρίζονται διακριτά στοιχεία πληροφορίας που ανήκουν σε ένα πεπερασμένο σύνολο, αφού το καθένα από αυτά μπορεί να παρασταθεί με μία ακολουθία δυαδικών ψηφίων, ακολουθώντας τους κανόνες ενός δυαδικού κώδικα.
- Κάθε διαφορετικό στοιχείο πληροφορίας αντιστοιχίζεται 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.

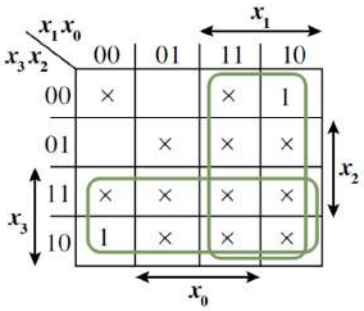
Απλός κωδικοποιητής 4 σε 2

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	×	×

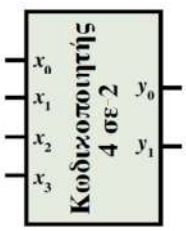


$$y_1 = x_2 + x_3$$





$$y_0 = x_1 + x_3$$

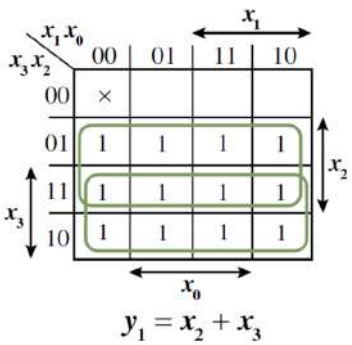


Κωδικοποιητής προτεραιότητας

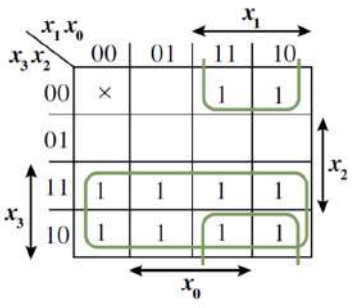
- Στη λειτουργία του απλού κωδικοποιητή εντοπίζονται δύο προβλήματα.
- Το πρώτο αφορά το γεγονός ότι οι έξοδοι του κυκλώματος μηδενίζονται, όταν όλες οι είσοδοι έχουν λογική τιμή 0, με αποτέλεσμα αυτός ο συνδυασμός τιμών των εισόδων να μην μπορεί να διακριθεί από την περίπτωση όπου μόνο η είσοδος x_0 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 (S) 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 S, 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 να καθορίζεται από την είσοδο με τη μεγαλύτερη προτεραιότητα.
- Για παράδειγμα, μπορεί να καθοριστεί ως είσοδος με τη μεγαλύτερη προτεραιότητα η x_3 και να ακολουθούν σε σειρά προτεραιότητας οι x_2, x_1 και x_0.

Κωδικοποιητής προτεραιότητας

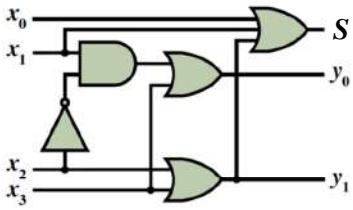
x_3	x_2	x_1	x_0	y_1	y_0	S
0	0	0	0	×	×	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	1	0	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1



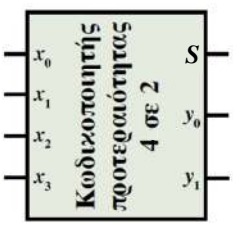
$$y_1 = x_2 + x_3$$



$$y_0 = x_3 + x_1x_2'$$



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

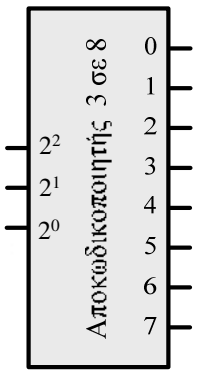


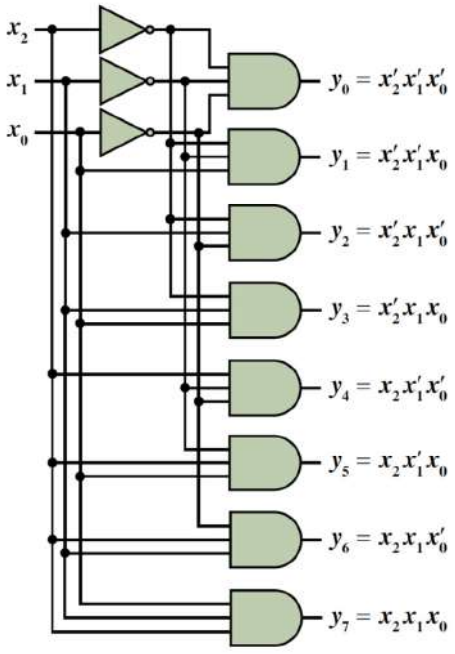
Αποκωδικοποιητές

- 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 ελαχίστων όρων, αφού κάθε έξοδος αντιστοιχεί σε έναν ελάχιστο όρο 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 ελάχιστων όρων.

Αποκωδικοποιητής 3 σε 8

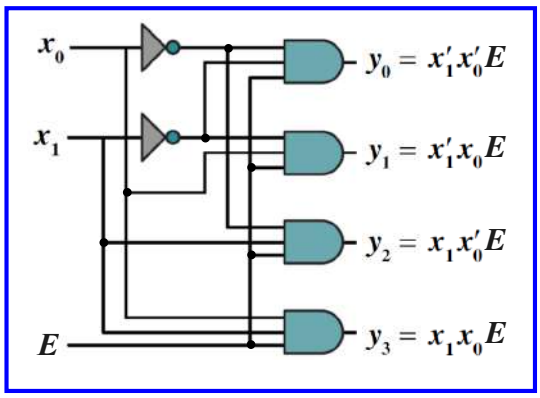
x_2	x_1	x_0	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

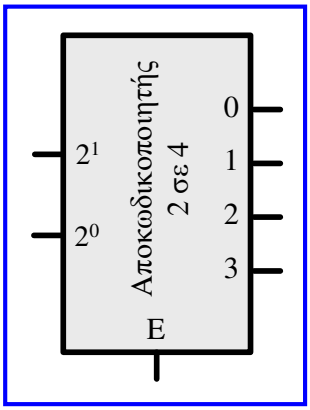




Αποκωδικοποιητής 2 σε 4 με είσοδο ενεργοποίησης

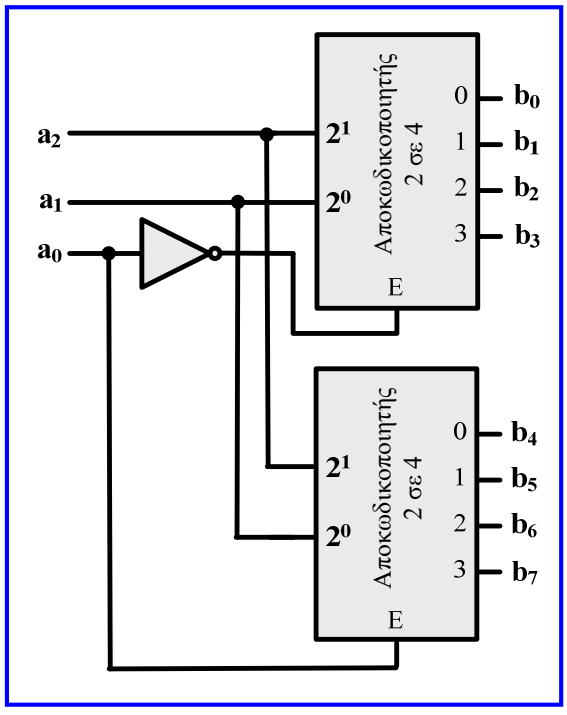
- Εκτός από τα ψηφία εισόδου που πρόκειται να αποκωδικοποιηθούν, το κύκλωμα ενός αποκωδικοποιητή μπορεί να περιλαμβάνει μία ακόμη είσοδο ενεργοποίησης (enable input), ανάλογα με την τιμή της οποίας να επιτρέπεται ή όχι η αποκωδικοποίηση.
- Η προσθήκη αυτή προϋποθέτει τη χρήση πυλών AND με μία επιπλέον 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.





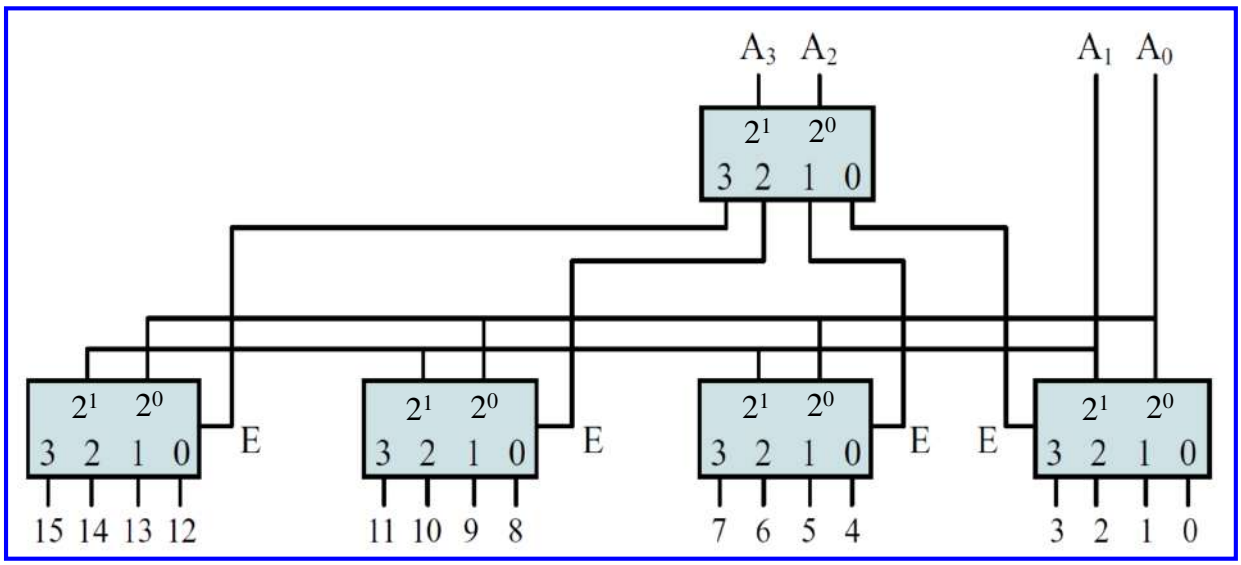
Σύνθεση πολύπλοκων αποκωδικοποιητών

- Η είσοδος E παρέχει τη δυνατότητα συνδυασμού αποκωδικοποιητών με στόχο τη σύνθεση αποκωδικοποιητών με περισσότερες εισόδους.
- Στη σχεδίαση ενός αποκωδικοποιητή 3 σε 8 με 2 αποκωδικοποιητές 2 σε 4:
  - ✓ όταν $a_0 = 0$, ενεργοποιείται ο πρώτος αποκωδικοποιητής 2 σε 4 και παράγει στις εξόδους b_0 έως b_3 τους ελάχιστους όρους m_0 έως m_3, αντίστοιχα,
  - ✓ όταν $a_0 = 1$, ενεργοποιείται ο δεύτερος αποκωδικοποιητής 2 σε 4 και παράγει στις εξόδους b_4 έως b_7 τους ελάχιστους όρους m_4 έως m_7, αντίστοιχα.



Σύνθεση πολύπλοκων αποκωδικοποιητών

Σχεδίαση αποκωδικοποιητή 4 σε 16 με 5 αποκωδικοποιητές 2 σε 4

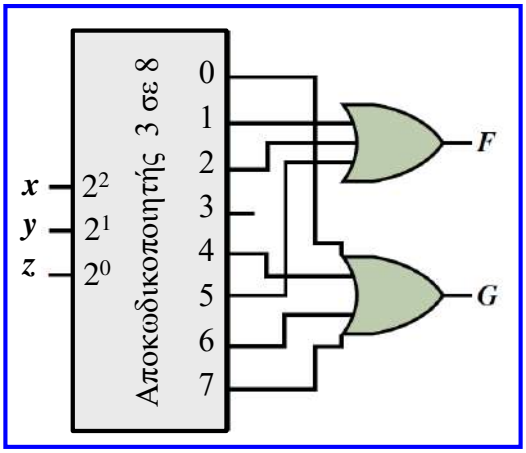


Υλοποίηση λογικών συναρτήσεων με αποκωδικοποιητή

- Αφού ένας αποκωδικοποιητής n σε 2^n αποτελεί και γεννήτρια ελαχίστων όρων, 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 ελαχίστων όρων.
- 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 ελαχίστων όρων που συμμετέχουν στη συνάρτηση.

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

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

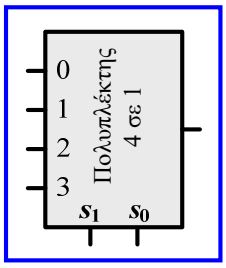


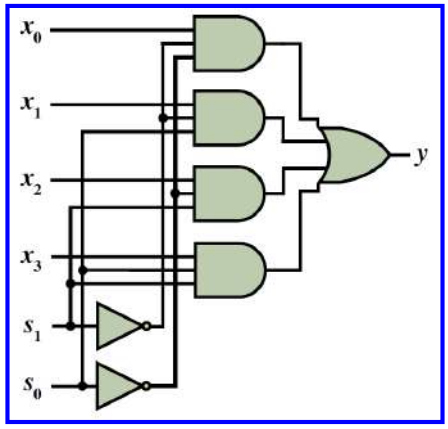
Πολυπλέκτες

- 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





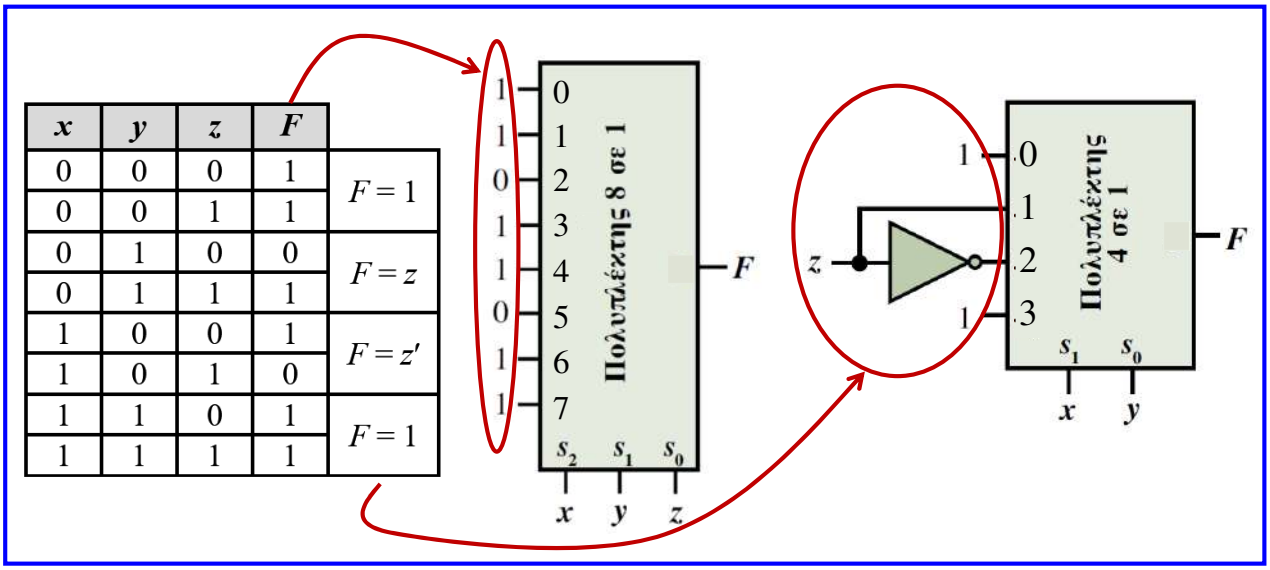
Υλοποίηση λογικών συναρτήσεων με πολυπλέκτες

- 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 ελαχίστων όρων των μεταβλητών που αντιστοιχούν στις εισόδους επιλογής, θέτοντας λογική τιμή 1 στις εισόδους δεδομένων που αντιστοιχούν στους ελάχιστους όρους που συμμετέχουν στο επιθυμητό άθροισμα και λογική τιμή 0 στους ελάχιστους όρους 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.
- 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 την υλοποίηση λογικής συνάρτησης n μεταβλητών με πολυπλέκτη 2^{n-1} σε 1, ενδέχεται να απαιτηθεί η χρησιμοποίηση ενός αντιστροφέα, έτσι ώστε να λαμβάνεται η συμπληρωματική μορφή της μεταβλητής που συμμετέχει στην τροφοδότηση των εισόδων δεδομένων του πολυπλέκτη.

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



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:

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

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

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

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:

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

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

- 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

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

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

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

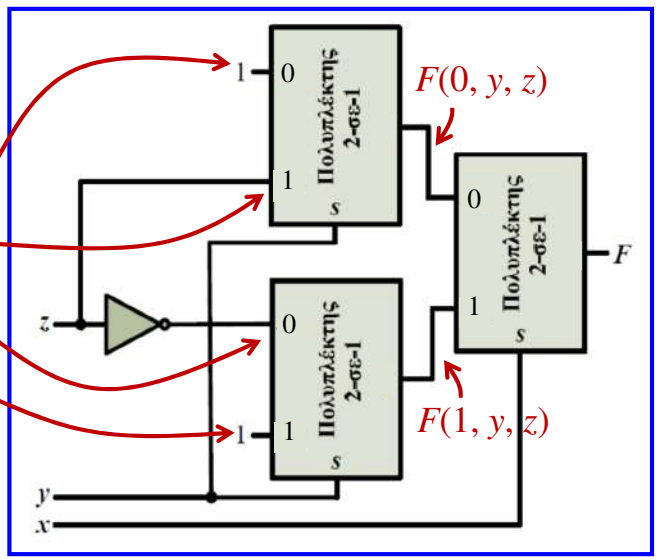
x	y	z	F	
0	0	0	1	$F = 1$
0	0	1	1	
0	1	0	0	$F = z$
0	1	1	1	
1	0	0	1	$F = z'$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	

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

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

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

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



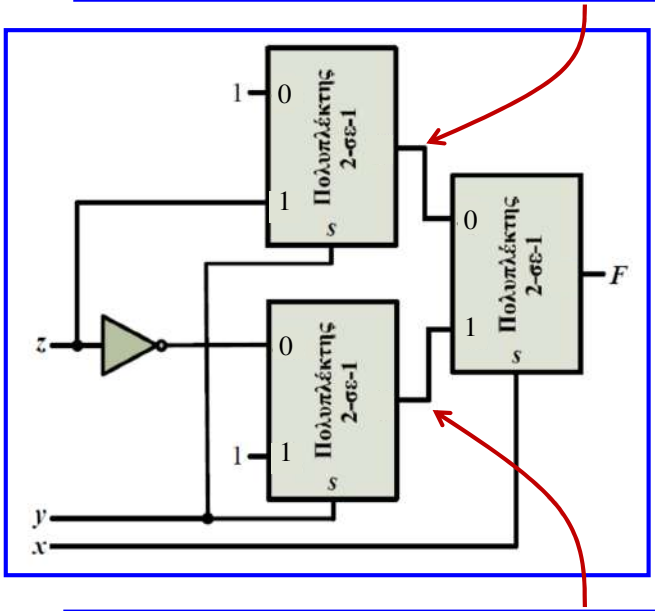
$$F(x,y,z) = x(y\mathbf{1} + y'\mathbf{z'}) + x'(y\mathbf{z} + y'\mathbf{1})$$

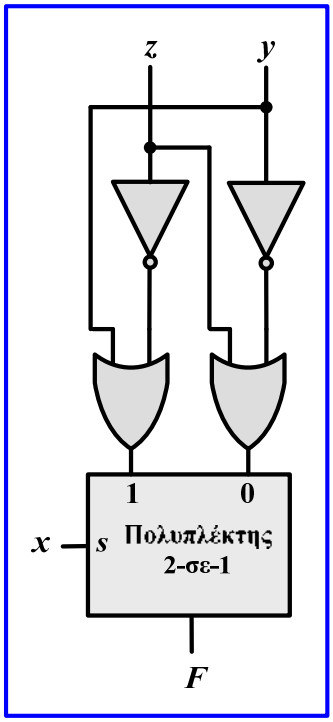
Υλοποίηση συναρτήσεων με πολυπλέκτες 2 σε 1

- 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.
- Σε απλούστερο κύκλωμα υλοποίησης μπορούμε να καταλήξουμε και στις περιπτώσεις όπου είσοδοι δεδομένων των πολυπλεκτών του πρώτου επιπέδου τροφοδοτούνται με λογικές τιμές 0 ή 1, αντικαθιστώντας πολυπλέκτες με λογικές πύλες, εφόσον αυτές είναι διαθέσιμες.

Υλοποίηση συναρτήσεων με πολυπλέκτες 2 σε 1

$$yz + y'1 = (y + y')(z + y') = y' + z = F(0, y, z)$$





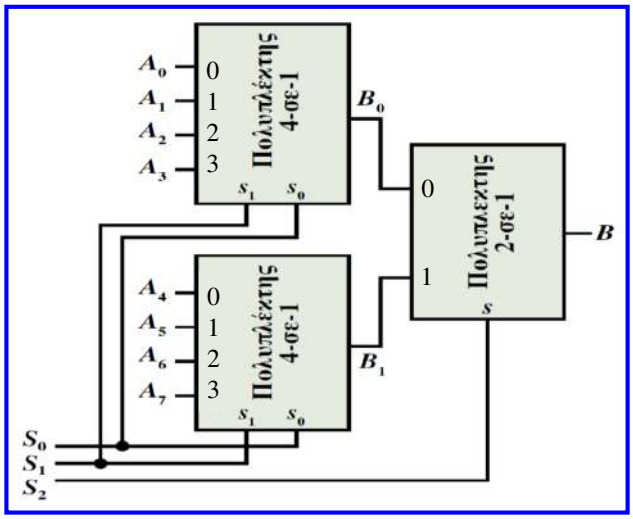
$$y1 + y'z' = (y + y')(y + z') = y + z' = F(1, y, z)$$

Σύνθεση πολύπλοκων πολυπλεκτών

Είναι δυνατό να συνθέσουμε πολυπλέκτες πολλών εισόδων με μικρότερους πολυπλέκτες.

Έξοδος πολυπλέκτη 8-σε-1

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

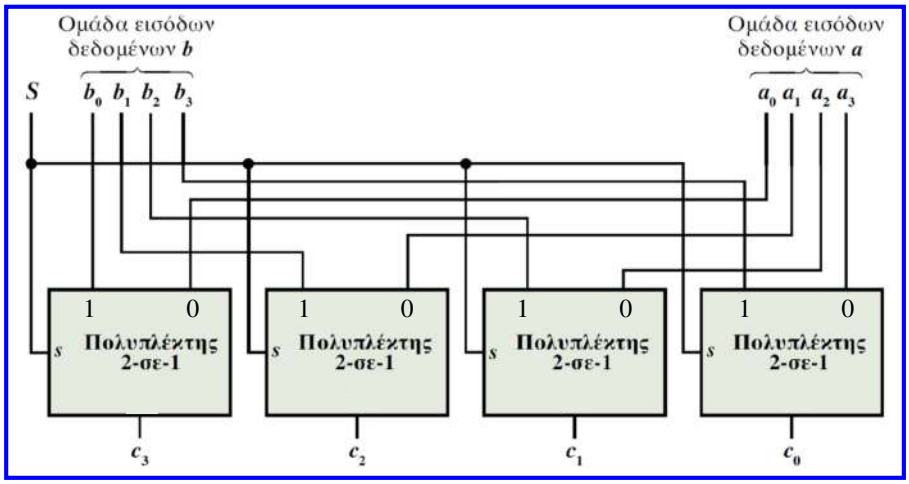


Πολυπλέκτες επιλογής πολλαπλών εισόδων

- 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.

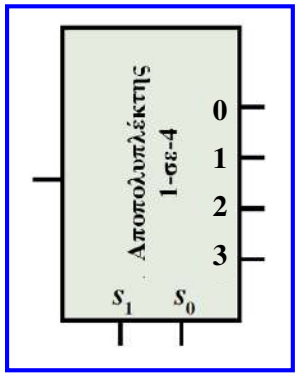
Παράδειγμα πολλαπλού πολυπλέκτη:

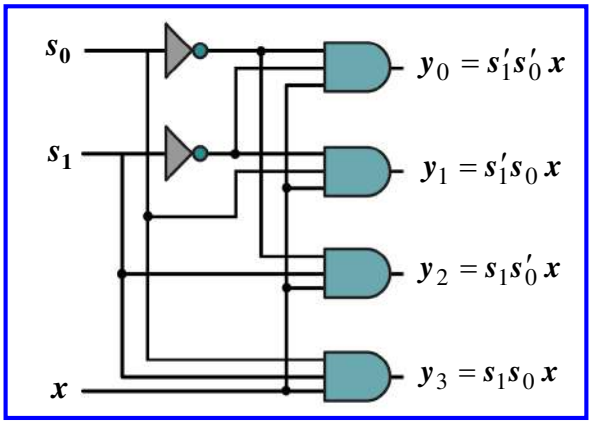
Τετραπλός πολυπλέκτης 2-σε-1



Αποπλέκτες ή αποπολυπλέκτες

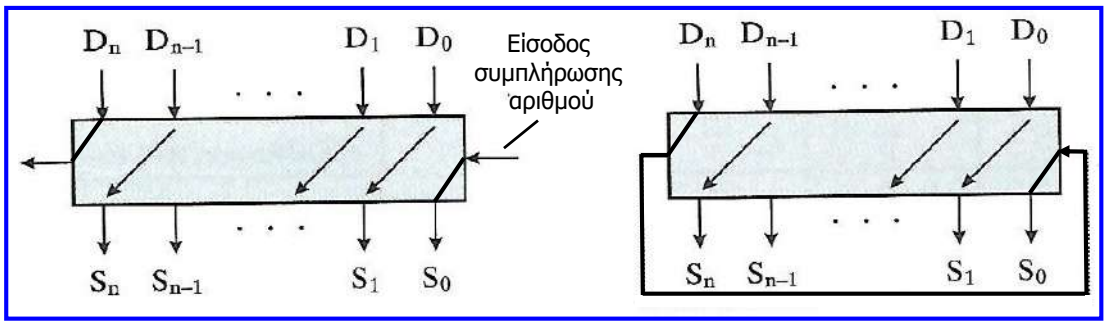
- Οι αποπολυπλέκτες (demultiplexers) ή αποπλέκτες επιτελούν την αντίστροφη λειτουργία από εκείνη των πολυπλεκτών.
- Διαθέτουν μία είσοδο δεδομένων και 2^n εξόδους δεδομένων, σε μία από τις οποίες μεταφέρεται η είσοδος δεδομένων, ανάλογα με το συνδυασμό τιμών των n εισόδων επιλογής.
- Ο αποκωδικοποιητής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





Ολισθητές

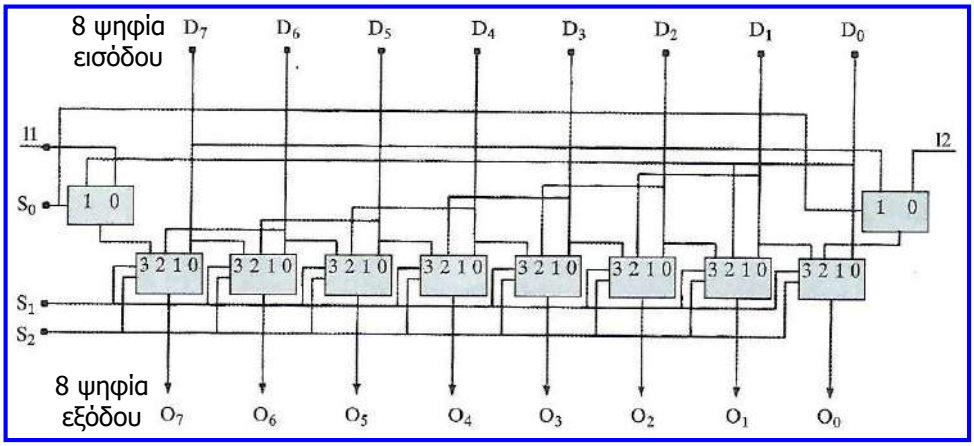
- Κατά τον πολλαπλασιασμό δυαδικών αριθμών απαιτούνται διαδοχικές μετατοπίσεις των ψηφίων του πολλαπλασιαστέου κατά μία θέση αριστερά, ενώ κατά τη διαίρεση απαιτούνται διαδοχικές μετατοπίσεις των ψηφίων του διαιρέτη κατά μία θέση δεξιά.
- Είναι χρήσιμα λοιπόν συνδυαστικά κυκλώματα ολισθητών, τα οποία δέχονται στην είσοδό τους έναν δυαδικό αριθμό και στην έξοδό τους λαμβάνεται μια εκδοχή του αριθμού εισόδου μετατοπισμένη κατά μία ή περισσότερες θέσεις προς τα δεξιά ή προς τα αριστερά.
- Χρησιμοποιούνται επίσης, οι κυκλικοί ολισθητές, στους οποίους τα ακραία ψηφία του αριθμού εισόδου τροφοδοτούν τις εκ διαμέτρου αντίθετες από αυτά εξόδους.



Ολισθητής αριστερής ολίσθησης κατά μία θέση

Αριστερόστροφος κυκλικός ολισθητής μίας θέσης

Ολισθητής πολλαπλών λειτουργιών με πολυπλέκτες



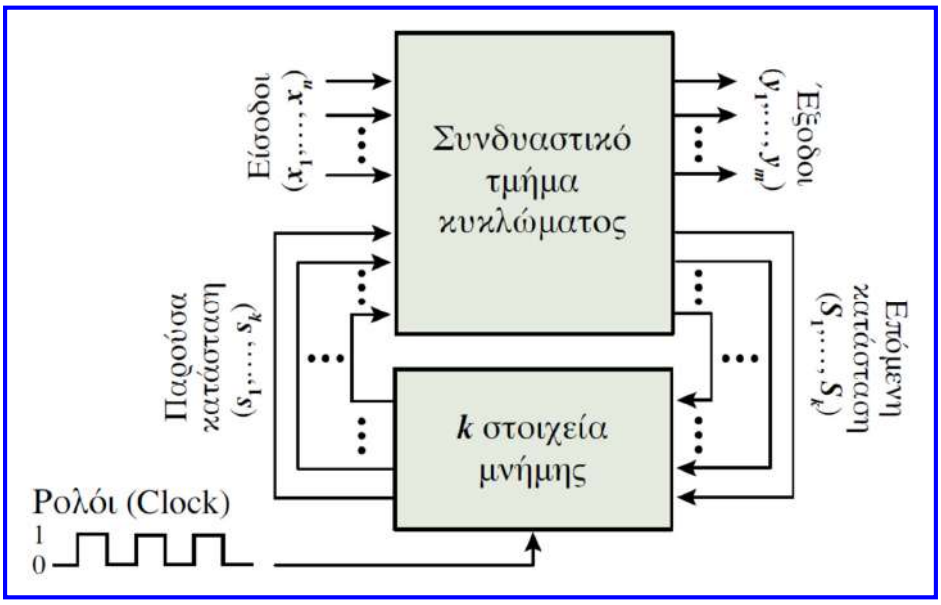
Οι πλάγιες είσοδοι I1 και I2 δίνουν τη δυνατότητα εισαγωγής συμπληρωματικών ψηφίων κατά την ολίσθηση

S_2	S_1	S_0	Λειτουργία	
0	0	X	$O_i = D_i$ (i=0... 7)	Ουδεμία μεταβολή
0	1	X	Δεν χρησιμοποιείται	
1	0	0	$O_i = D_{i-1}$, O_0 = I2 (i=1... 7)	Ολίσθηση αριστερά, συμπλήρωση από είσοδο I2
1	0	1	$O_i = D_{i-1}$, $O_0 = D_7$ (i=1... 7)	Περιστροφή αριστερά
1	1	0	$O_i = D_{i+1}$, O_7 = I1 (i=0... 6)	Ολίσθηση δεξιά, συμπλήρωση από είσοδο I1
1	1	1	$O_i = D_{i+1}$, $O_7 = D_0$ (i=0... 6)	Περιστροφή δεξιά

Ακολουθιακά κυκλώματα

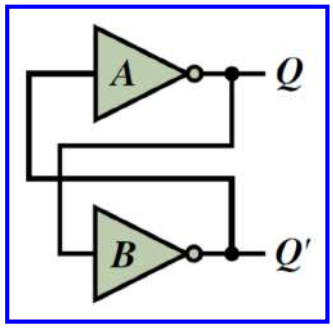
- Στα συνδυαστικά κυκλώματα, η λογική τιμή της εξόδου ή των εξόδων τους, κάθε χρονική στιγμή, εξαρτάται μόνο από τη λογική τιμή των εισόδων που εφαρμόζεται σε αυτά την ίδια χρονική στιγμή.
- Τα ακολουθιακά κυκλώματα (sequential circuits), εκτός από ένα συνδυαστικό τμήμα, περιλαμβάνουν στοιχεία (ή κυκλώματα) μνήμης που αναφέρονται ως μανταλωτές (latches) και flip-flop.
- 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 κατάσταση (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 που αναφέρεται ως ρολόι (clock).

Βασική δομή σύγχρονου ακολουθιακού κυκλώματος



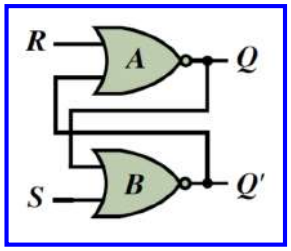
Κυκλώματα μνήμης ακολουθιακών κυκλωμάτων

- Οι μανταλωτές, και τα απλά flip-flop 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 έχει τιμή 1).
- Η μεταβολή των εξόδων των ακμοπυροδοτούμενων flip-flop 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 Β θα λάβει τιμή 1 (δηλ. Q'), αφού η έξοδος Q του αντιστροφέα Α συνδέεται στην είσοδο του αντιστροφέα Β.
- Η έξοδος του αντιστροφέα Β συνδέεται στην είσοδο του αντιστροφέα Α, με αποτέλεσμα Q = 0, γεγονός που είναι σύμφωνο με την υπόθεσή μας.
- Παρομοίως, αν υποθέσουμε ότι Q = 1, τότε η έξοδος του αντιστροφέα Β λαμβάνει τιμή 0 (δηλ. ξανά Q').



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

Μανταλωτής SR

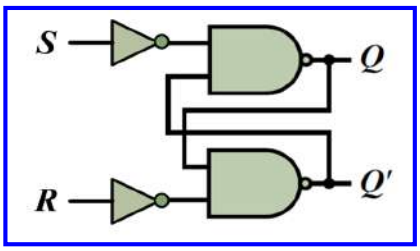


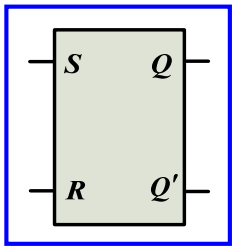
S	R	Q_{t+1}	Q'_{t+1}	Παρατηρήσεις
0	0	Q_t	Q'_t	Αμετάβλητη κατάσταση
0	1	0	1	Κατάσταση μηδενισμού
1	0	1	0	Κατάσταση θέσης
1	1	0	0	Απροσδιόριστη κατάσταση

- 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.

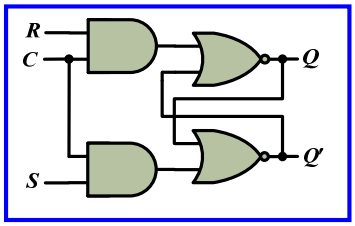
Μανταλωτής SR

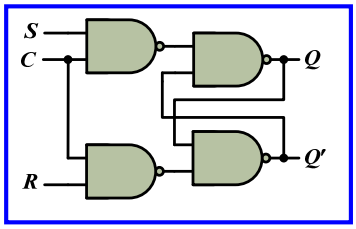
- Ο μανταλωτής SR μπορεί επίσης να υλοποιηθεί με δύο πύλες NAND, τροφοδοτούμενες με τις συμπληρωματικές μορφές των εισόδων. Όταν S = R = 1, οι έξοδοι των πυλών NAND λαμβάνουν τιμή 0 και δεν είναι συμπληρωματικές μεταξύ τους (απροσδιόριστη κατάσταση).

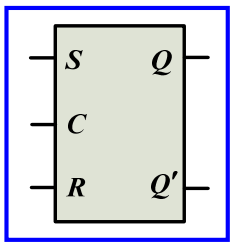




- Επειδή, ο μανταλωτής SR με ένα επίπεδο πυλών είναι εξαιρετικά ευαίσθητος σε ανεπιθύμητους παλμούς (glitches) των εισόδων του, προσθέτουμε ένα επίπεδο πυλών και μια είσοδο ενεργοποίησης (C), έτσι ώστε μόνο όταν C = 1, επιτρέπεται οι τιμές των S, R να περάσουν στο δεύτερο επίπεδο πυλών.

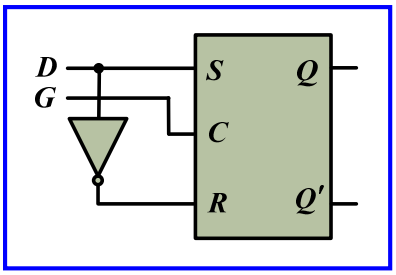




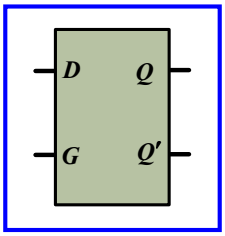


Μανταλωτής D

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

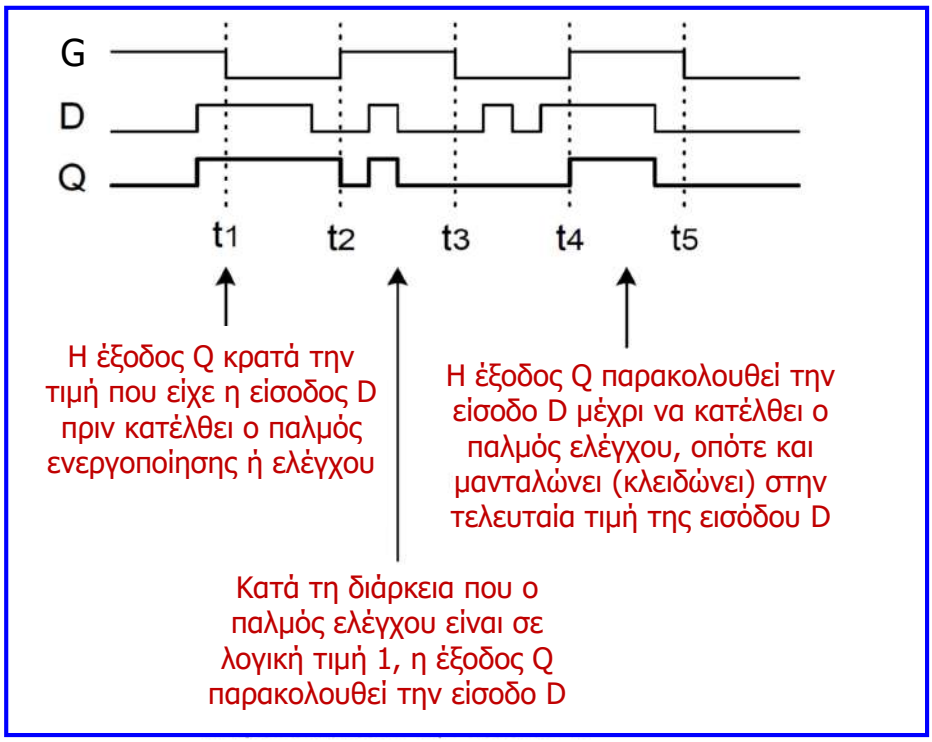


G	D	Q_{t+1}	Q'_{t+1}	Παρατηρήσεις
0	0	Q_t	Q'_t	Αμετάβλητη κατάσταση
0	1	Q_t	Q'_t	Αμετάβλητη κατάσταση
1	0	0	1	Κατάσταση μηδενισμού
1	1	1	0	Κατάσταση θέσης



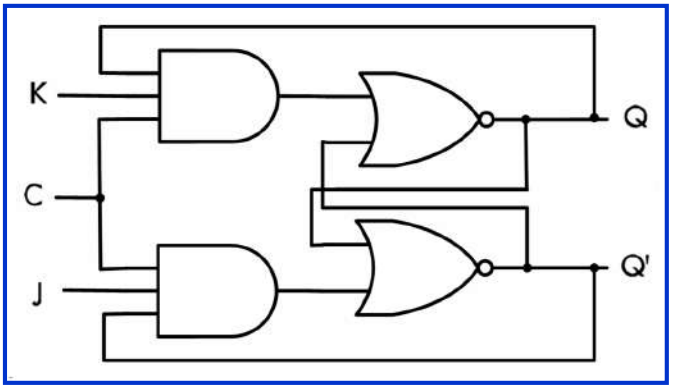
$$Q_{t+1} = GD + G'Q_t$$

Μανταλωτής D



Απλό JK flip-flop

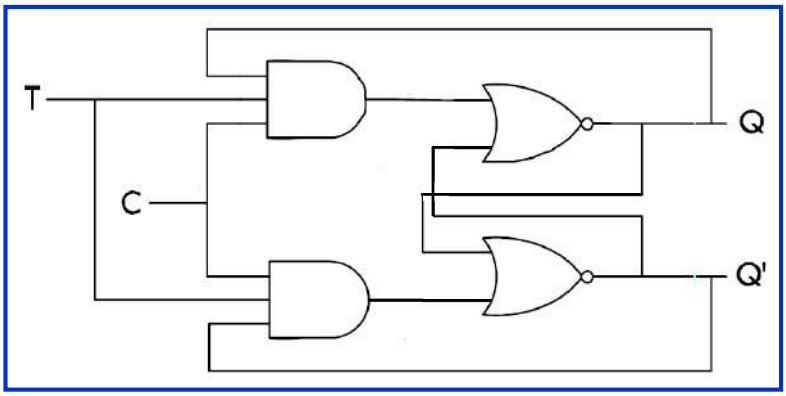
- Η απροσδιόριστη κατάσταση του μανταλωτή SR μπορεί να αρθεί με τροποποίηση που συνίσταται στην ανατροφοδότηση των εξόδων στις εισόδους του, που οδηγεί στη δημιουργία του απλού JK flip-flop.
- Μόνο όταν C = 1, οι είσοδοι επηρεάζουν την πληροφορία που αποθηκεύεται.
- Όταν J = K = 1 (και C = 1), τότε η κατάσταση του flip-flop αντιστρέφεται.
- Στις υπόλοιπες περιπτώσεις τιμών των εισόδων, η λειτουργία του είναι αντίστοιχη με εκείνη του μανταλωτή SR με είσοδο ενεργοποίησης.



J	K	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	Q'_t	Q_t	Αντιστροφή

Απλό T flip-flop

- Πρόκειται για μια παραλλαγή του απλού JK flip-flop με συνδεδεμένες τις δύο εισόδους του.
- Μόνο όταν C = 1, η είσοδος T επηρεάζει την πληροφορία που αποθηκεύεται.
- Όταν T = 0 (και C = 1) το flip-flop παραμένει στην ίδια κατάσταση, ενώ όταν T = 1 (και C = 1), τότε η κατάσταση του flip-flop αντιστρέφεται.



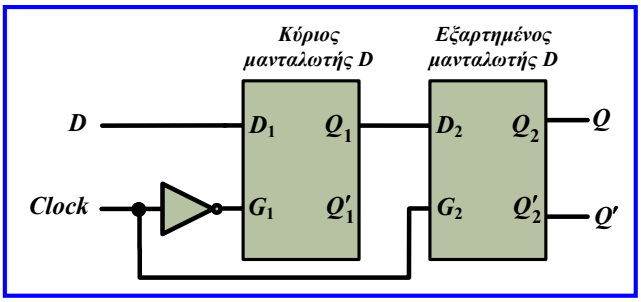
T	Q_{t+1}	Q'_{t+1}	Παρατηρήσεις
0	Q_t	Q'_t	Αμετάβλητη κατάσταση
1	Q'_t	Q_t	Αντιστροφή

Ακμοπυροδοτούμενα flip-flops

- Εάν στην είσοδο ελέγχου (G) 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 flip-flops να αλλάζει μόνο σε διακριτές χρονικές στιγμές.
- Για το λόγο αυτόν έχουν αναπτυχθεί flip-flops 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 (edge-triggered flip-flop).
- Ένα ακμοπυροδοτούμενο flip-flop μπορεί να αλλάξει κατάσταση κατά την ανερχόμενη ή την κατερχόμενη ακμή του σήματος ρολογιού.
- Η βασική διαφορά αυτών των flip-flop με τους μανταλωτές με είσοδο ενεργοποίησης ή τα απλά flip-flop έγκειται στο ότι μεταβολή της τιμής των εξόδων τους συμβαίνει μόνο σε διακριτές χρονικές στιγμές, οι οποίες καθορίζονται από ένα σήμα ρολογιού.
- Τα ακμοπυροδοτούμενα flip-flop χρησιμοποιούνται ως στοιχεία μνήμης στα ΣΑΚ.
- Η βασική διαφορά μεταξύ των διάφορων τύπων flip-flop (D, JK, T), αφορά τον τρόπο με τον οποίο οι είσοδοι δεδομένων τους επηρεάζουν τη δυαδική πληροφορία που αποθηκεύεται σε αυτά, δηλαδή την κατάστασή τους.

Ακμοπυροδοτούμενο D flip-flop

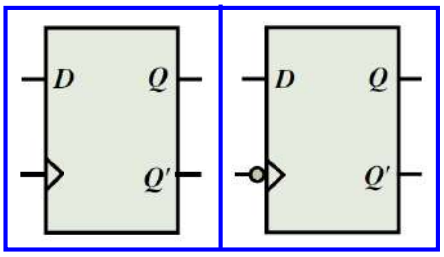
- Ένα ακμοπυροδοτούμενου στην ανερχόμενη ακμή D flip-flop, συνδυάζει δύο μανταλωτές D και αναφέρεται ως κύριο-εξαρτημένο flip-flop (master-slave flip-flop).
- Η είσοδος D μεταφέρεται στην έξοδο του κύριου μανταλωτή, όταν Clock = 0.
- Στο διάστημα κατά το οποίο Clock = 0, η έξοδος του κύριου και, κατά συνέπεια, η είσοδος του εξαρτημένου 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 flip-flop μπορεί να αλλάξει μόνο στην ανερχόμενη ακμή του Clock.
- Για flip-flop D με αλλαγή κατάστασης στην κατερχόμενη ακμή του Clock τροφοδοτούμε τον κύριο μανταλωτή με την κανονική μορφή του Clock και τον εξαρτημένο με Clock’.



Ακμοπυροδοτούμενο D flip-flop

- Για την παρουσίαση της λειτουργίας των ακμοπυροδοτούμενων flip-flop 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 flip-flop να προσδιορίσουμε τις τιμές της εισόδου ή των εισόδων του που προκαλούν τη σχετική μετάβαση.
- Αυτό επιτυγχάνεται με τον πίνακα διέγερσης (excitation table) των flip-flops.

Σύμβολα του D flip-flop



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

Χαρακτηριστική εξίσωση

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

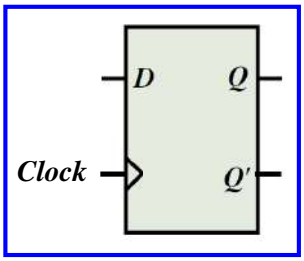
D	Q_{t+1}
0	0
1	1

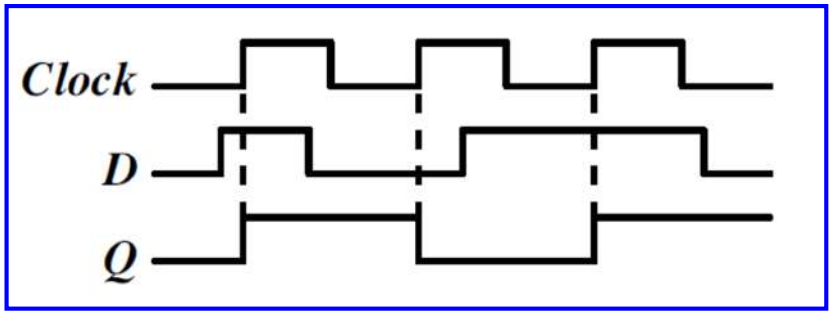
Πίνακες λειτουργίας ή χαρακτηριστικοί πίνακες

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

Πίνακας διέγερσης

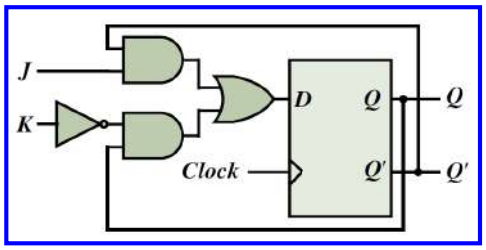
Ακμοπυροδοτούμενο D flip-flop



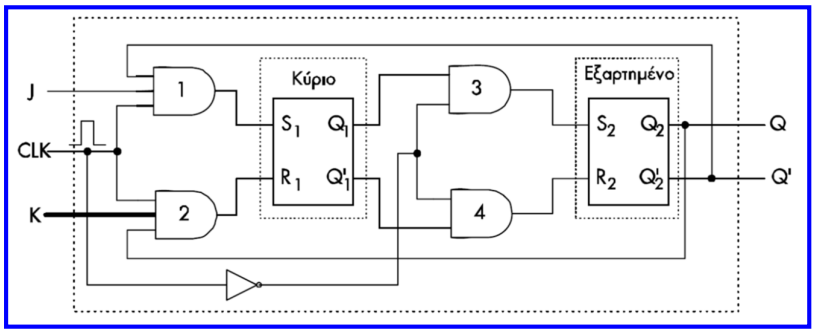


Ακμοπυροδοτούμενο JK flip-flop

- Με το D flip-flop 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 ως εξής:



$$D = JQ' + K'Q$$



Ακμοπυροδοτούμενο JK flip-flop

Πίνακες λειτουργίας

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

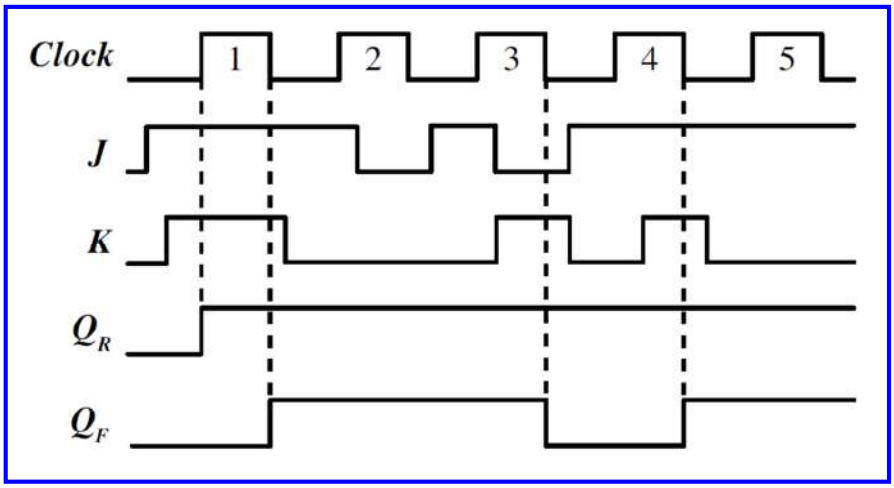
J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t

Πίνακας διέγερσης

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

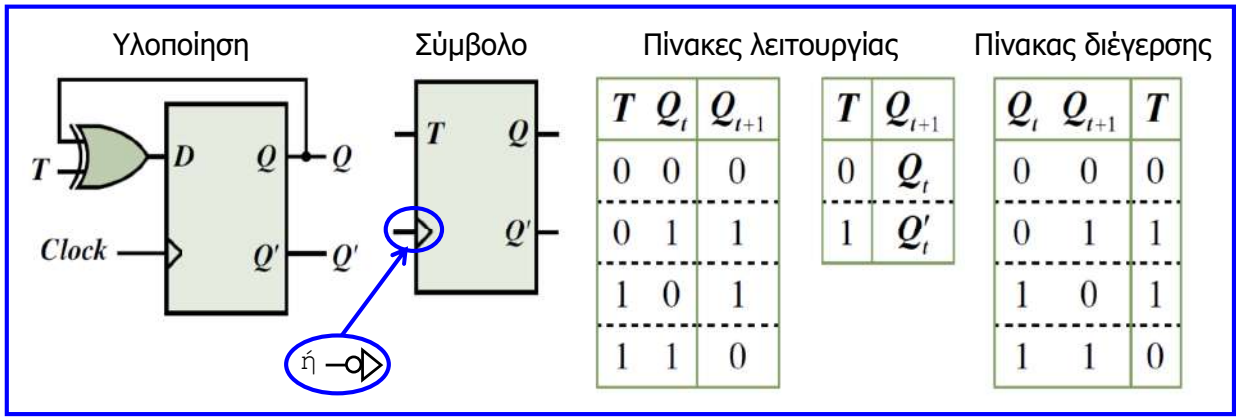
Χαρακτηριστική εξίσωση $Q_{t+1} = JQ'_t + K'Q_t$

Ακμοπυροδοτούμενο JK flip-flop



Ακμοπυροδοτούμενο T flip-flop

Το T flip-flop περιλαμβάνει μία είσοδο δεδομένων (T) και υλοποιείται εύκολα με συνδυασμό ενός ακμοπυροδοτούμενου D flip-flop και μιας πύλης XOR δύο εισόδων



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

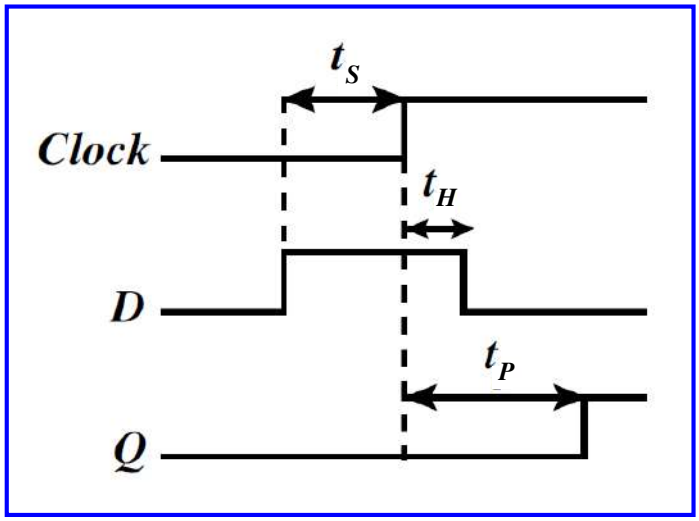
Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

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

Χρονικοί περιορισμοί στα ακμοπυροδοτούμενα flip-flop

- Η αλλαγή κατάστασης ενός ακμοπυροδοτούμενου flip-flop γίνεται ακριβώς τη χρονική στιγμή έλευσης της ακμής του σήματος ρολογιού, στην οποία πυροδοτείται το flip-flop.
- 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 flip-flop, 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 D flip-flop είναι το ελάχιστο χρονικό διάστημα πριν από την έλευση της ακμής του σήματος ρολογιού, στην οποία πυροδοτείται το flip-flop, κατά το οποίο η είσοδος δεδομένων θα πρέπει να διατηρείται σταθερή στην επιθυμητή λογική τιμή, ώστε αυτή να μεταφερθεί στην έξοδο του flip-flop.
- Χρόνος παραμονής είναι το ελάχιστο χρονικό διάστημα μετά την έλευση της ακμής του σήματος ρολογιού στην οποία πυροδοτείται το flip-flop, κατά το οποίο η είσοδος δεδομένων θα πρέπει να παραμείνει σταθερή στην επιθυμητή λογική τιμή.

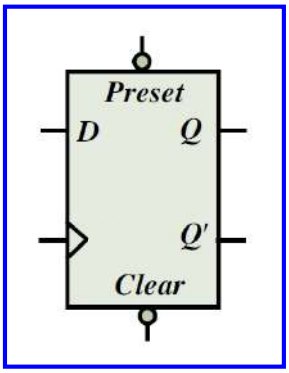
Χρονικοί περιορισμοί στα ακμοπυροδοτούμενα flip-flop

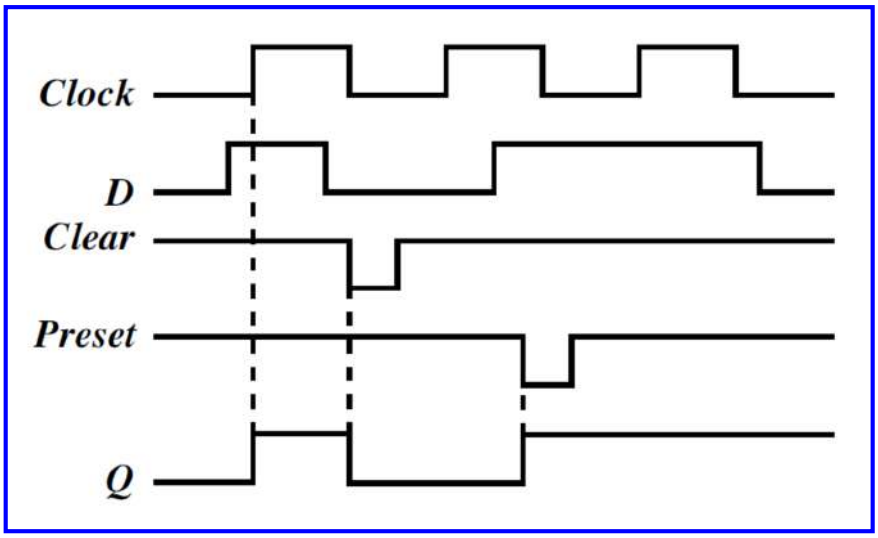


Ασύγχρονες είσοδοι στα ακμοπυροδοτούμενα flip-flop

- Η είσοδος δεδομένων (D) του D flip-flop αποτελεί σύγχρονη είσοδο (synchronous input), αφού η επίδρασή της στη δυαδική πληροφορία που αποθηκεύεται στο flip-flop είναι συγχρονισμένη με τις ακμές του σήματος ρολογιού στις οποίες πυροδοτείται το στοιχείο.
- Ωστόσο, στα flip-flops 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 flip-flop 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 μηδενισμού ενεργοποιούνται όταν λαμβάνουν λογική τιμή 0, και η περίπτωση ταυτόχρονης ενεργοποίησής τους θα πρέπει να αποφεύγεται, αφού οδηγεί το flip-flop σε απροσδιόριστη κατάσταση.

Ασύγχρονες είσοδοι στα ακμοπυροδοτούμενα flip-flop





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:

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

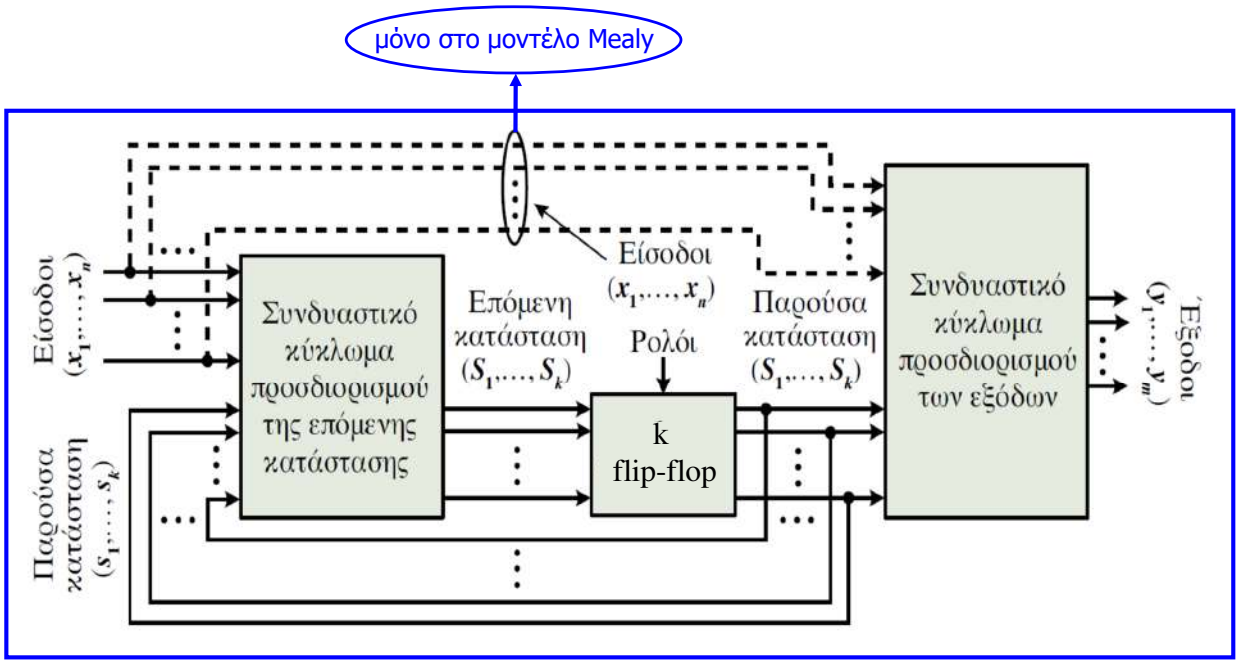
- Γενικά, η λειτουργία ενός σύγχρονου ακολουθιακού κυκλώματος (ΣΑΚ) με n εισόδους (x_1 έως x_n), m εξόδους (y_1 έως y_m) και k στοιχεία μνήμης (flip-flops), στα οποία αντιστοιχούν οι μεταβλητές κατάστασης, με τις μεταβλητές από s_1 έως s_k και από S_1 έως S_k να εκφράζουν την παρούσα και την επόμενη 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 κατάστασή 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 κατάστασή του, τότε το ΣΑΚ περιγράφεται με το μοντέλο Moore.

Σύγχρονα ακολουθιακά κυκλώματα (ΣΑΚ)



Περιγραφή λειτουργίας ΣΑΚ

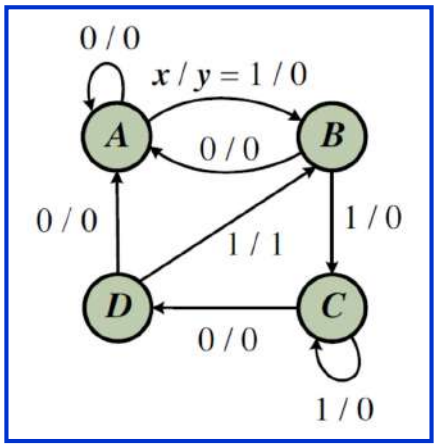
- Ένας «φιλικός» και παραστατικός τρόπος περιγραφής των λειτουργικών σχέσεων ανάμεσα στις μεταβλητές που συμμετέχουν σε ένα ΣΑΚ, είναι το διάγραμμα καταστάσεων (state diagram).
- Οι διαφορετικές καταστάσεις του κυκλώματος παριστάνονται με κύκλους και οι μεταβάσεις από μια κατάσταση στην επόμενη παριστάνονται 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 (flip-flop) του ακολουθιακού κυκλώματος.
- Έτσι, για να εκφράσουμε 2 καταστάσεις απαιτείται 1 μεταβλητή και το ΣΑΚ περιλαμβάνει 1 flip-flop, από 3 έως 4 καταστάσεις απαιτούνται 2 μεταβλητές και 2 flip-flop, από 5 έως 8 καταστάσεις απαιτούνται 3 μεταβλητές και 3 flip-flop, ενώ για να εκφράσουμε από 9 έως 16 καταστάσεις απαιτούνται 4 μεταβλητές και 4 flip-flop, 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.
- Πρόκειται για ανιχνευτή ακολουθίας (sequence recognizer) που ενεργοποιεί την έξοδό του, όταν λαμβάνει στην είσοδο 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.

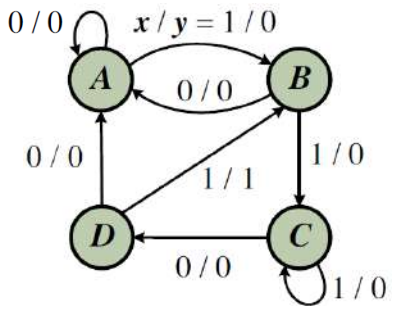
A: αρχική κατάσταση



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



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

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

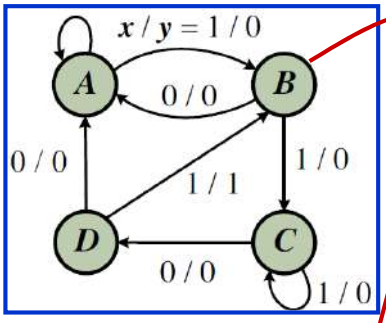
Περιγραφή λειτουργίας ΣΑΚ

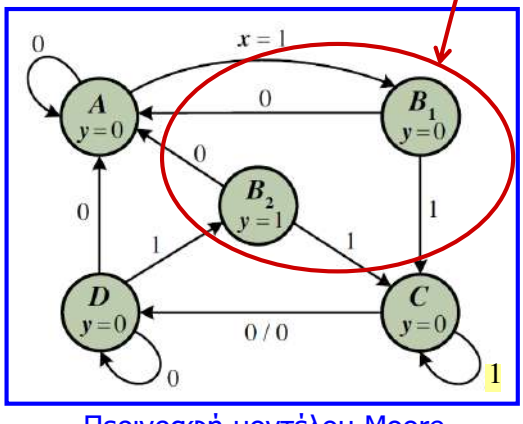
- Λόγω του ότι στις μεταβάσεις προς μία κατάσταση μοντέλου Mealy μπορούν να αντιστοιχούν διαφορετικές τιμές εξόδου, η περιγραφή μοντέλου Moore απαιτεί συνήθως περισσότερες καταστάσεις από εκείνη του μοντέλου Mealy με ισοδύναμη συμπεριφορά.
- Μια κατάσταση μοντέλου Mealy που αντιστοιχεί σε διαφορετικές 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.

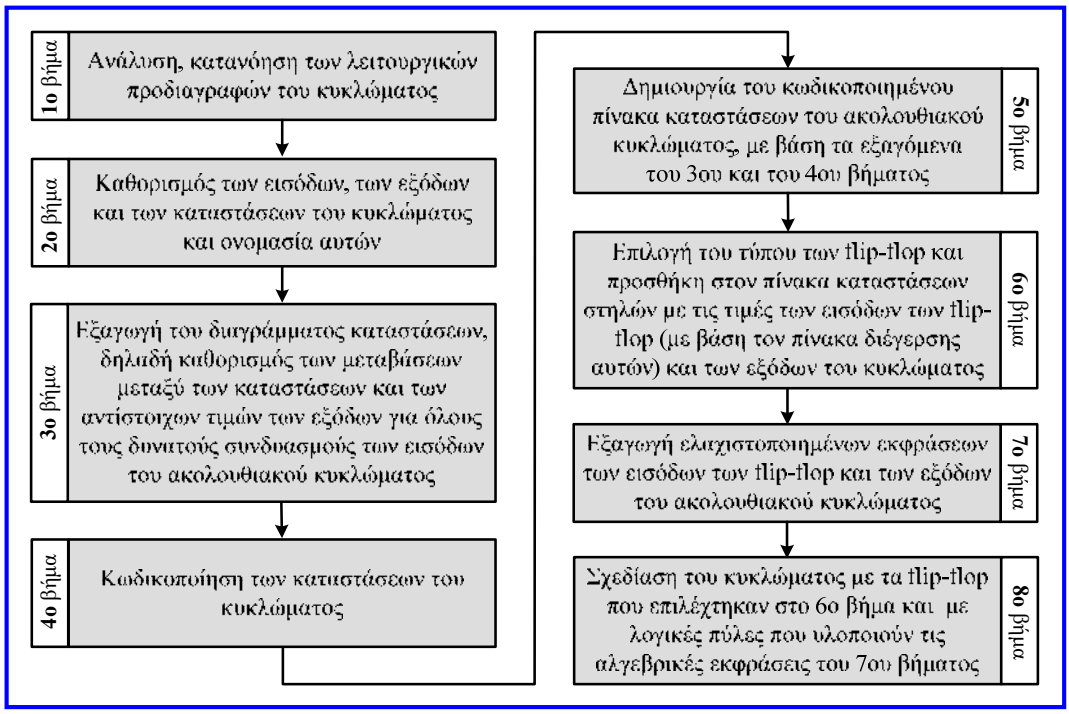
Περιγραφή μοντέλου Mealy





Περιγραφή μοντέλου Moore

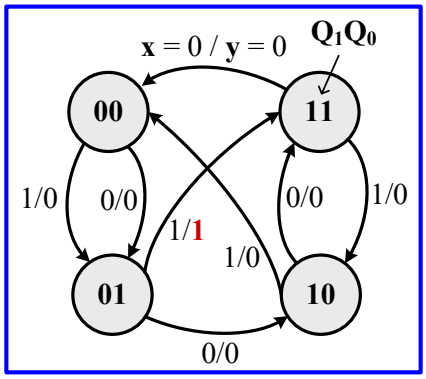
Σχεδίαση (σύνθεση) ΣΑΚ



Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 1ο

- Επιθυμούμε να συνθέσουμε ΣΑΚ με D flip-flop, 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 (y = 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 flip-flop.

Διάγραμμα καταστάσεων (μοντέλο Mealy):



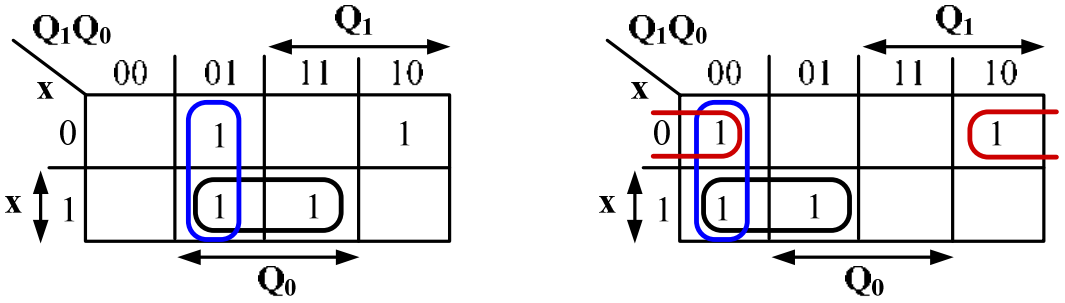
Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 1ο

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

Είσοδος	Παρούσα κατάσταση		Επόμενη κατάσταση		Είσοδοι των flip-flop		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

Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 1ο

Ελαχιστοποιημένες εκφράσεις των εισόδων των flip-flop:



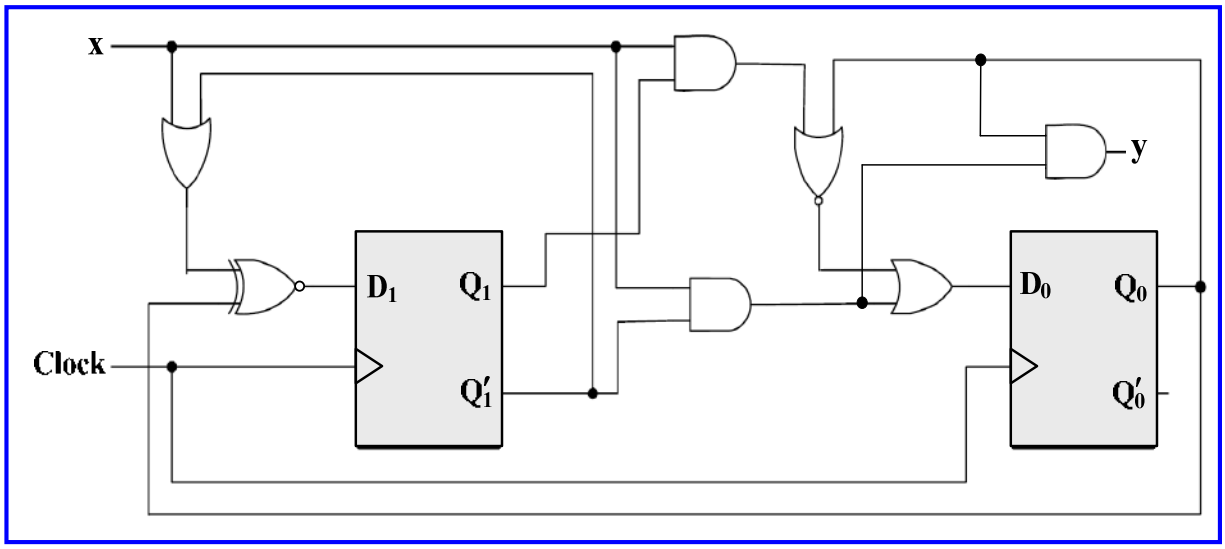
$$D_1 = Q'_1Q_0 + xQ_0 + x'Q_1Q'_0$$
$$= Q_0(Q'_1 + x) + Q'_0(Q'_1 + x)'$$
$$= [Q_0 \oplus (Q'_1 + x)]'$$

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

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

Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 1ο

Λογικό διάγραμμα του ΣΑΚ με D flip-flop:

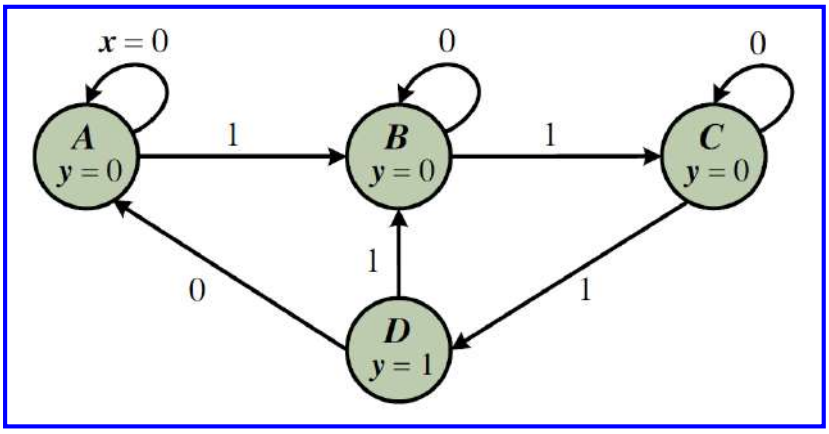


Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 2ο

- 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.

Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 2ο

Διάγραμμα καταστάσεων (μοντέλο Moore)



Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 2ο

- Αφού το κύκλωμα διατρέχει 4 καταστάσεις, απαιτούνται 2 flip-flop.
- 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 flip-flop 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

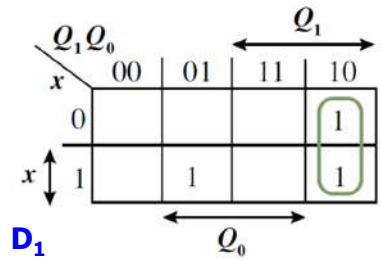
Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 2ο

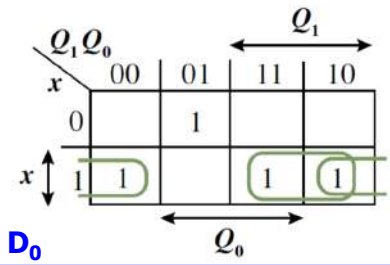
$Q_{t+1} = D$

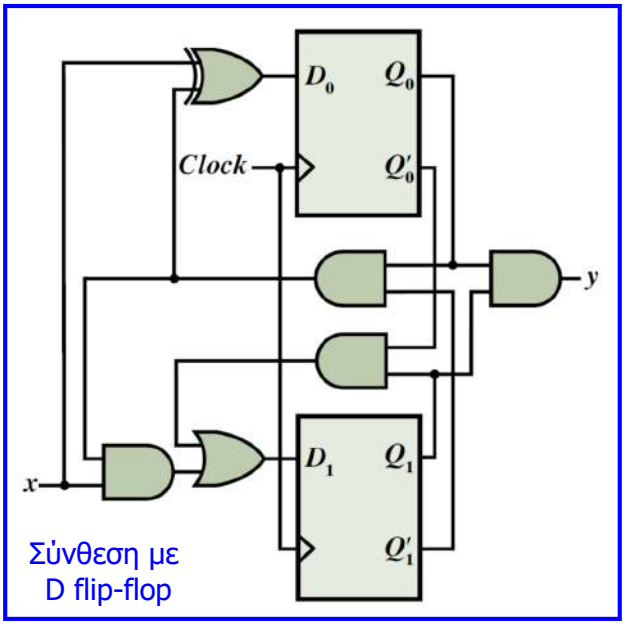
$$D_1 = xQ_1'Q_0 + Q_1Q_0'$$

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

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







Σύνθεση με D flip-flop

Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 2ο

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

Είσοδος	Παρούσα κατάσταση		Επόμενη κατάσταση		Είσοδοι των flip-flop						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 flip-flop

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

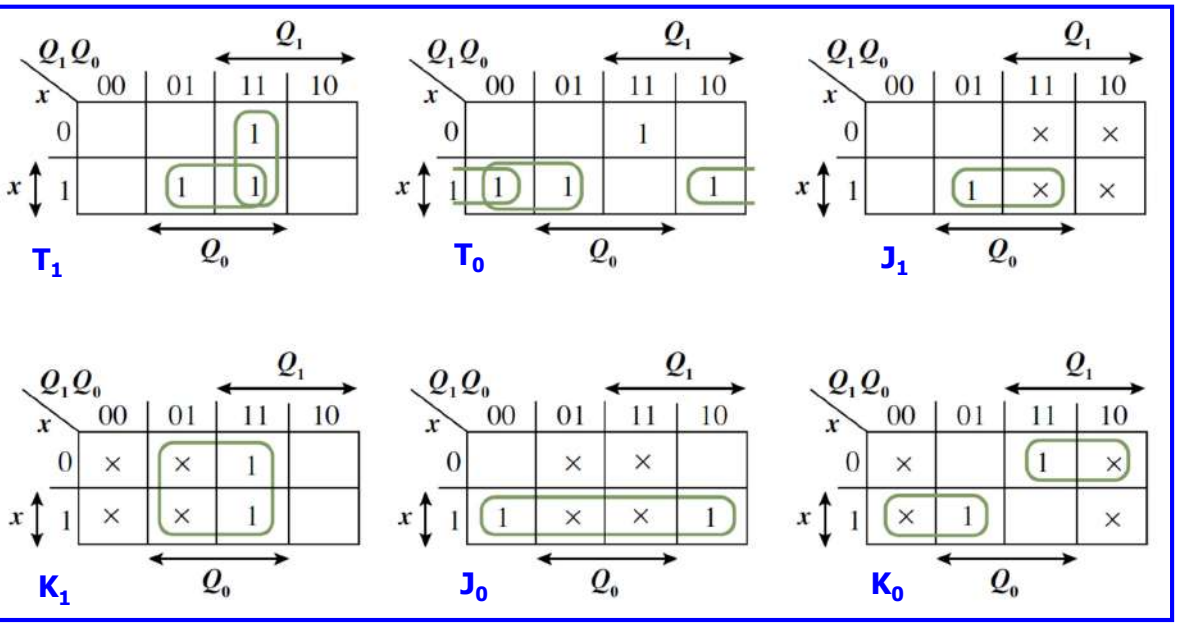
Q_t	Q_{t+1}	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 2ο

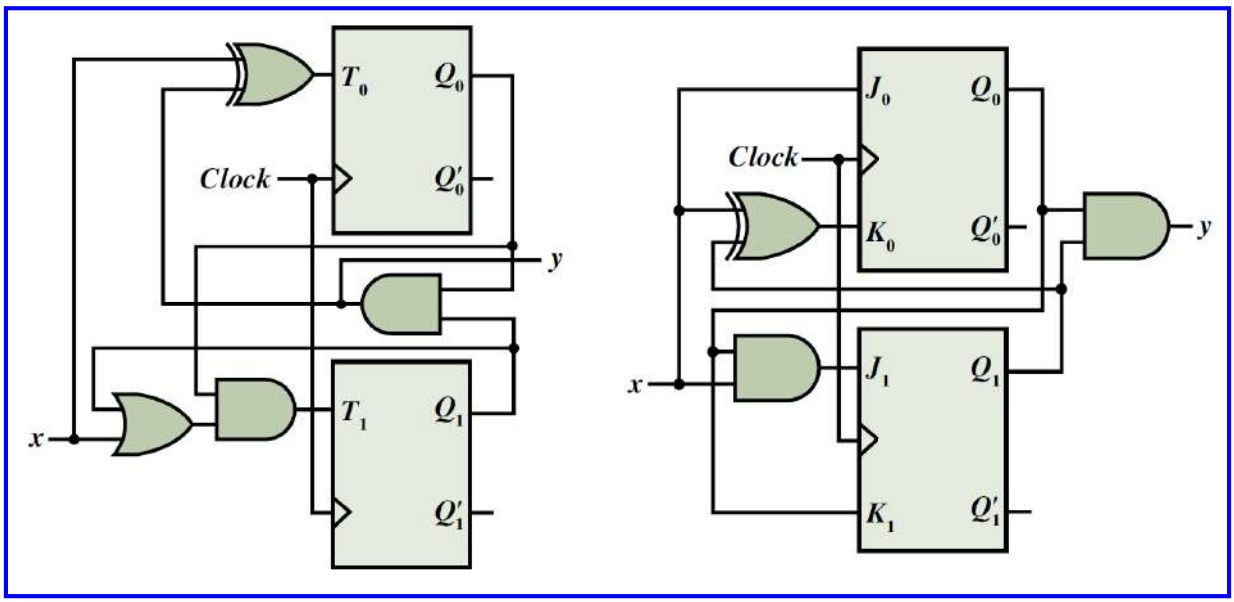
$$T_1 = xQ_0 + Q_1Q_0 = (x + Q_1)Q_0$$

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

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



Σχεδίαση (σύνθεση) ΣΑΚ – Παράδειγμα 2ο



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

Μερικώς καθορισμένα ΣΑΚ

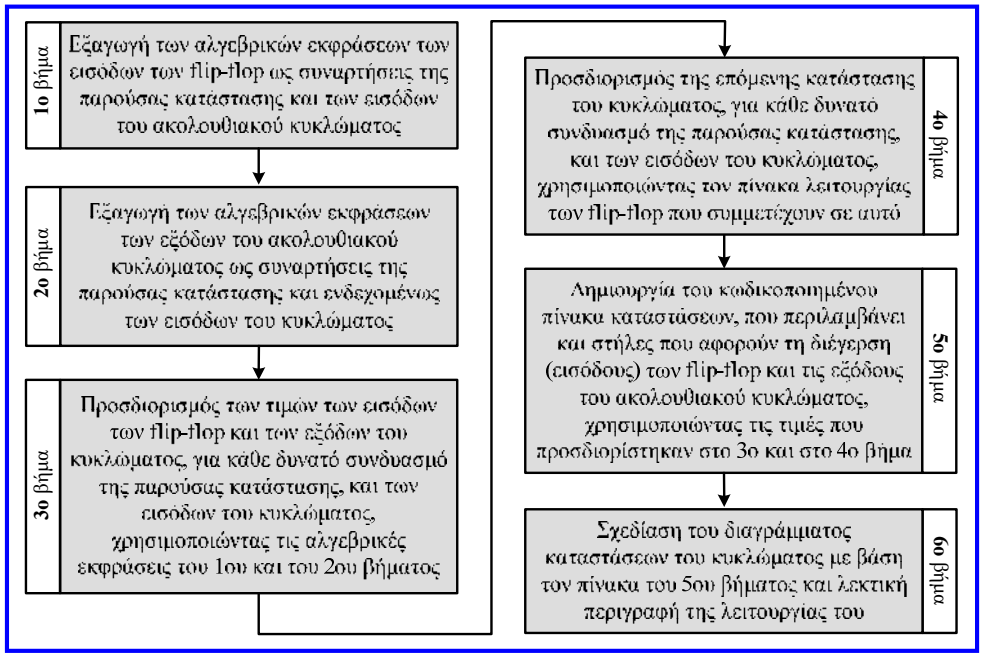
- Στο ΣΑΚ που υλοποιήσαμε για κάθε συνδυασμό κατάστασης και τιμής εισόδου καθορίζεται ένα ζεύγος επόμενης κατάστασης και τιμής εξόδου (πλήρως καθορισμένο κύκλωμα, completely specified circuit).

- 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, με αποτέλεσμα την εξαγωγή απλούστερων εκφράσεων προσδιορισμού των εισόδων των flip-flop και των εξόδων του κυκλώματος.

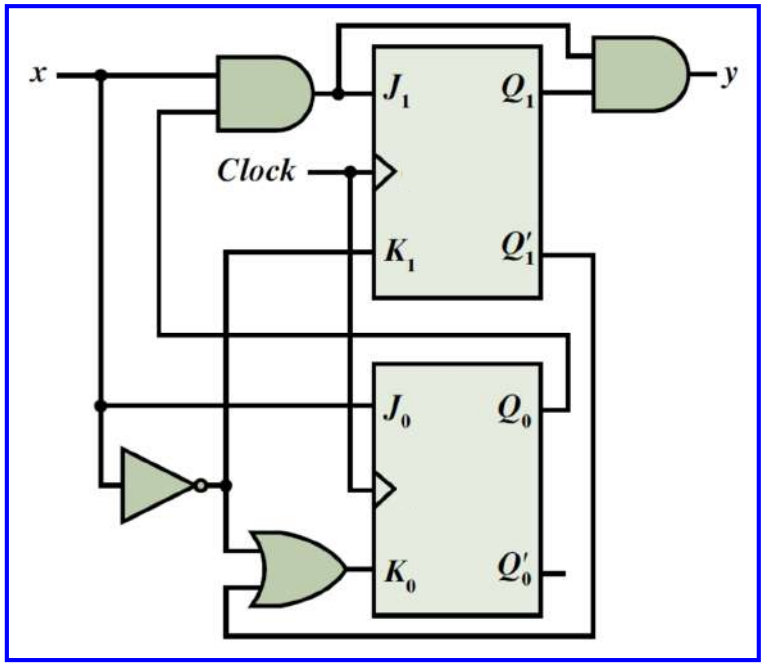
Ανάλυση ΣΑΚ

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 flip-flop με βάση τις εκφράσεις τους για κάθε συνδυασμό τιμών εισόδου και παρούσας κατάστασης και στη συνέχεια, με βάση τον πίνακα λειτουργίας του JK flip-flop, 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 flip-flop				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

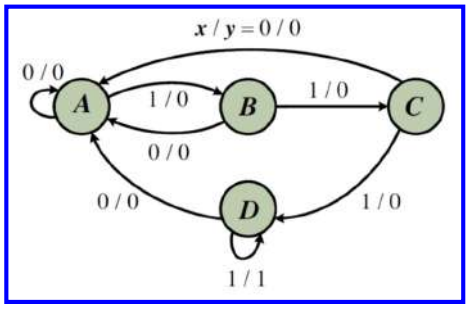
Πίνακας λειτουργίας JK flip-flop

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



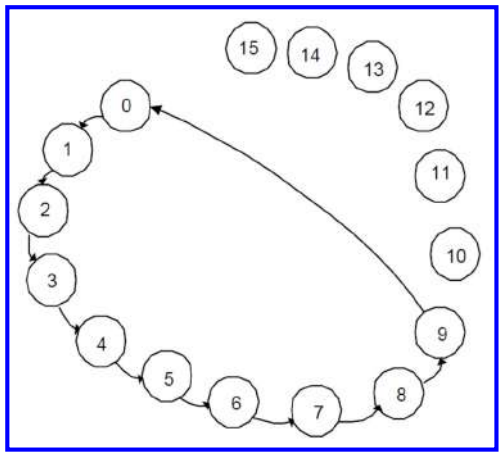
- Διαπιστώνουμε ότι το κύκλωμα, ξεκινώντας από την κατάσταση A, διατρέχει κατά σειρά τις καταστάσεις B, C και D, όταν η ακολουθία των δυαδικών ψηφίων που λαμβάνεται στην είσοδο x αποτελείται από 3 μονάδες.
- Στις περιπτώσεις που υπάρχει παρεμβολή ψηφίου εισόδου με τιμή 0, το κύκλωμα επιστρέφει στην κατάσταση A.
- Η έξοδος y λαμβάνει τιμή 1, μόνο όταν το κύκλωμα βρίσκεται στην κατάσταση D και ληφθεί μία ακόμη μονάδα στην είσοδο. Στην περίπτωση αυτή, 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).

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 flip-flops, 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 τις τιμές 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-flop 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-flop.
- 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 flip-flop			
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

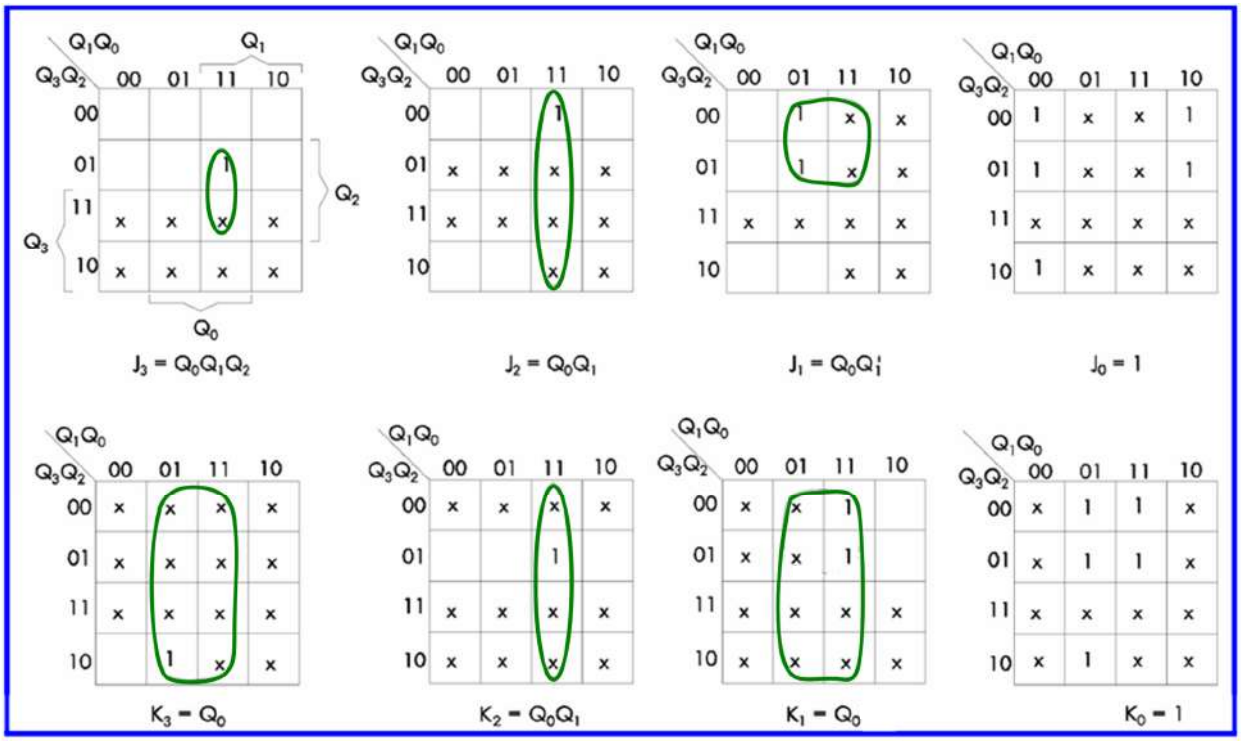
Πίνακας καταστάσεων

Πίνακας διέγερσης JK flip-flop

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

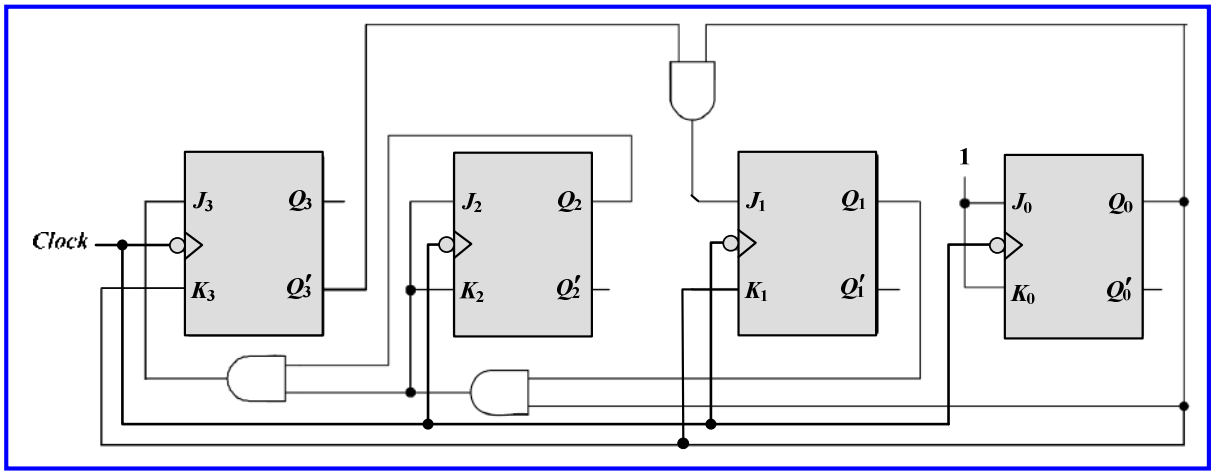
ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 1ο

Προσδιορισμός των συναρτήσεων εισόδων των flip-flop



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 flip-flops				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

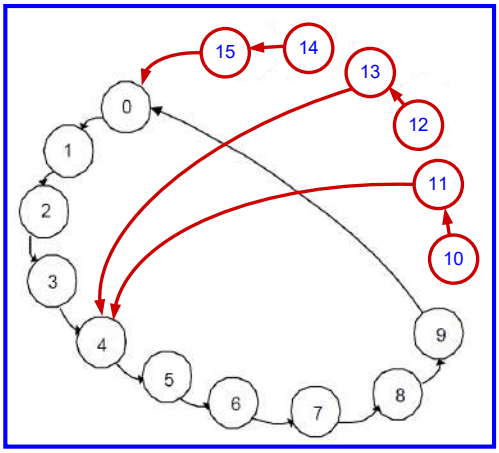
Αχρησιμοποίητες καταστάσεις

Πίνακας λειτουργίας JK flip-flop

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

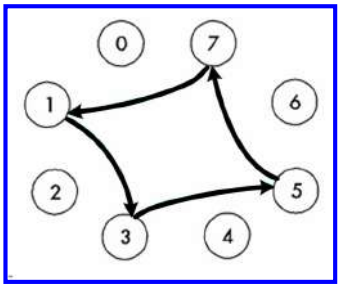


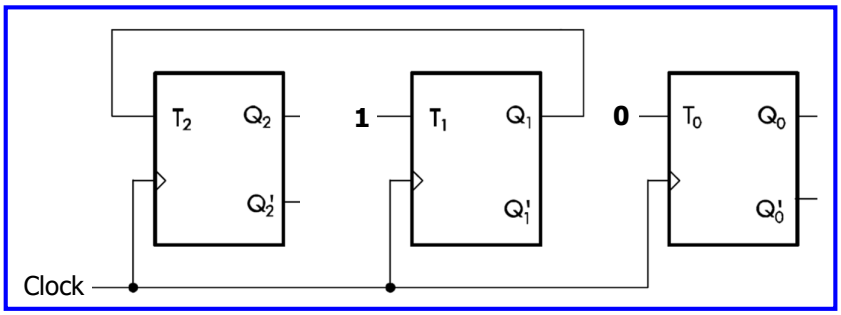
Από την ανάλυση του κυκλώματος διαπιστώνουμε ότι εάν το ΣΑΚ βρεθεί σε μία από τις αχρησιμοποίητες καταστάσεις, το αργότερο μετά από 2 παλμούς ρολογιού, επιστρέφει σε μία από τις έγκυρες καταστάσεις (κύκλωμα με αυτόματη διόρθωση).

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 T flip-flop, 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.

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

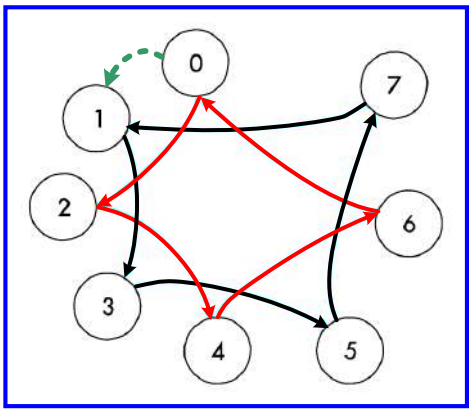
Πίνακας λειτουργίας JK flip-flop

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

Παρούσα κατάσταση			Είσοδοι flip-flops			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.

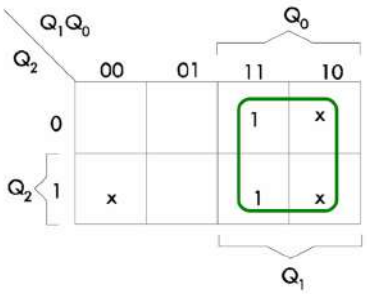


Πίνακας διέγερσης T flip-flop

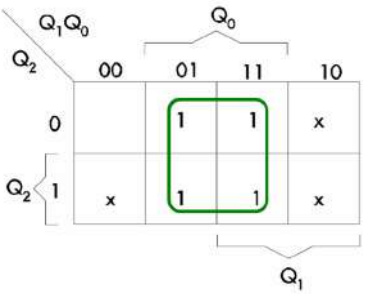
Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Παρούσα κατάσταση			Επόμενη κατάσταση			Είσοδοι flip-flops		
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

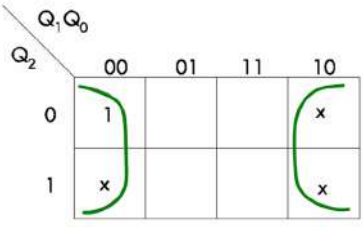
ΣΑΚ με αχρησιμοποίητες καταστάσεις – Παράδειγμα 2ο



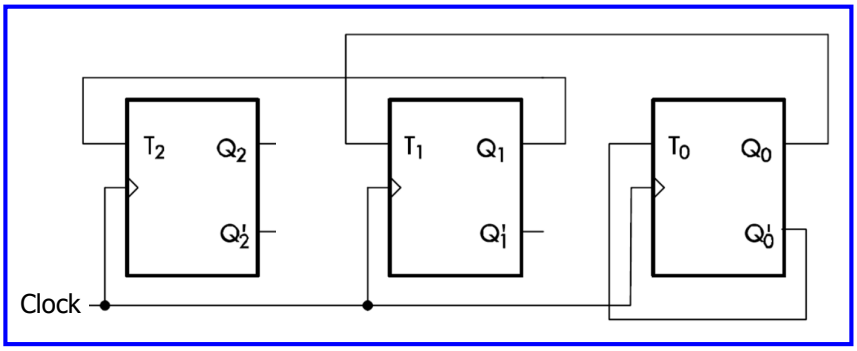
$T_2 = Q_1$



$T_1 = Q_0$



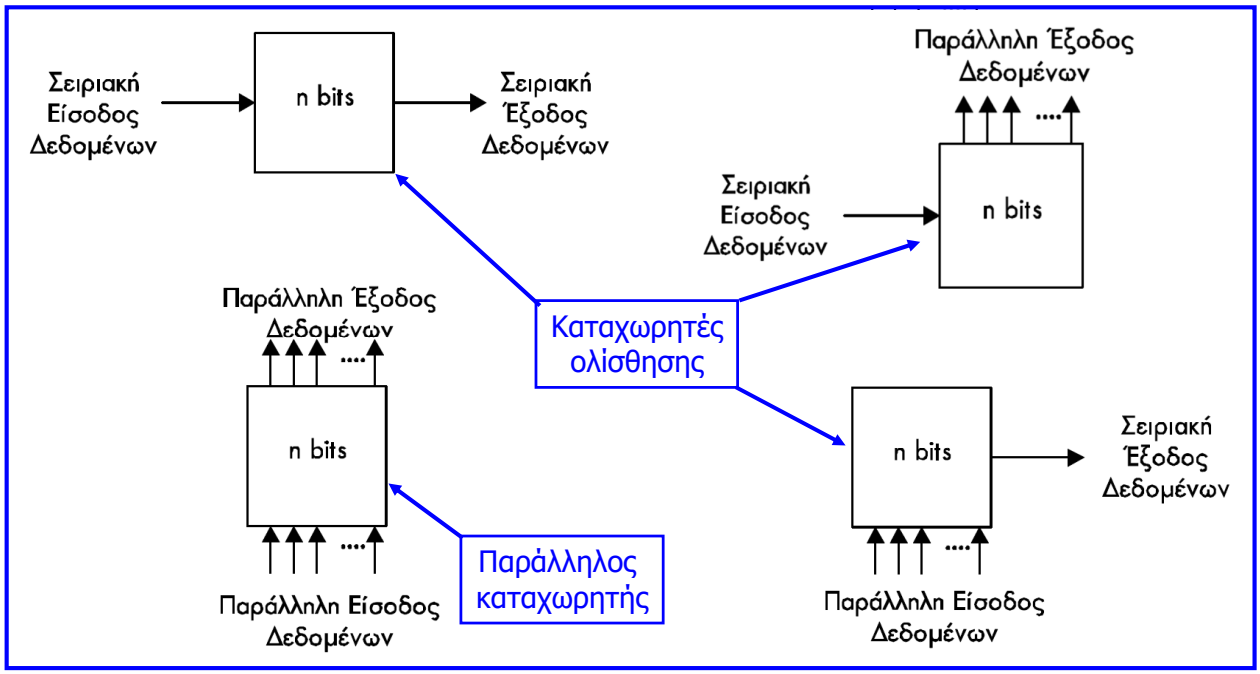
$T_0 = Q'_0$



Καταχωρητές

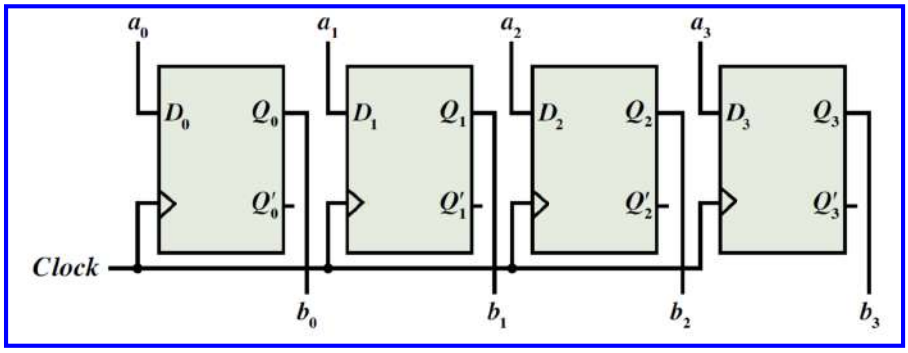
- Οι καταχωρητές (registers) είναι σύγχρονα ακολουθιακά κυκλώματα τα οποία χρησιμοποιούνται στα ψηφιακά συστήματα για την προσωρινή αποθήκευση δυαδικών δεδομένων, ώστε να διευκολυνθεί η επεξεργασία τους.
- Περιλαμβάνουν συνήθως ένα συνδυαστικό τμήμα και flip-flop.
- Με δεδομένο ότι ένα flip-flop μπορεί να αποθηκεύσει πληροφορία ενός δυαδικού ψηφίου, προκύπτει εύκολα ότι ένας καταχωρητής θα πρέπει να περιλαμβάνει τόσα flip-flops, όσα είναι τα δυαδικά ψηφία που πρόκειται να αποθηκευτούν σε αυτόν.
- Η φόρτωση (loading) ή είσοδος των δυαδικών δεδομένων σε έναν καταχωρητή μπορεί να γίνει παράλληλα (δηλ. ταυτόχρονη φόρτωση του συνόλου των δυαδικών ψηφίων σε έναν παλμό του σήματος ρολογιού) ή σειριακά (δηλ. φόρτωση ενός δυαδικού ψηφίου σε κάθε παλμό του σήματος ρολογιού).
- Η ανάγνωση (reading) ή έξοδος των δυαδικών δεδομένων από έναν καταχωρητή μπορεί να γίνεται με τους ίδιους τρόπους.
- 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 flip-flops των καταχωρητών.

Καταχωρητές



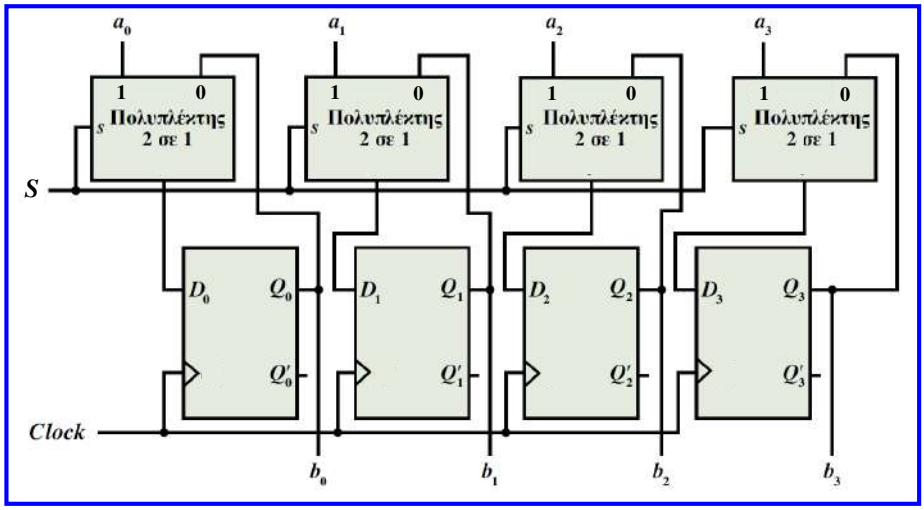
Απλός παράλληλος καταχωρητής

- 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 flip-flop.
- Η προσθήκη δυνατότητας μηδενισμού της πληροφορίας του καταχωρητή μπορεί να επιτευχθεί με τη χρησιμοποίηση flip-flop με ασύγχρονη είσοδο καθαρισμού (clear).



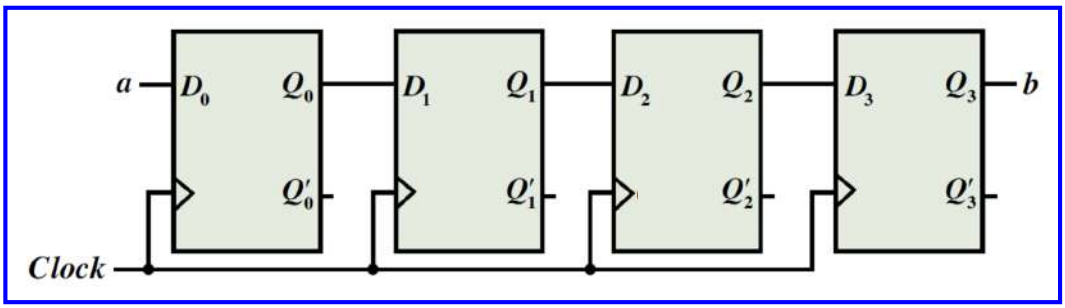
Παράλληλος καταχωρητής με είσοδο φόρτωσης

- 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 flip-flop, η είσοδος επιλογής των οποίων τροφοδοτείται με μία επιπλέον είσοδο S του καταχωρητή, που αναφέρεται ως είσοδος ελέγχου παράλληλης εισόδου ή είσοδος φόρτωσης.
- Όταν S = 0, η είσοδος δεδομένων κάθε flip-flop τροφοδοτείται με την τιμή της παρούσας κατάστασής του (τα δεδομένα του καταχωρητή παραμένουν αναλλοίωτα).
- Όταν S = 1, στις εισόδους δεδομένων των flip-flop τροφοδοτούνται τα δυαδικά ψηφία εισόδου του καταχωρητή (νέα δεδομένα στη ανερχόμενη ακμή του σήματος ρολογιού).



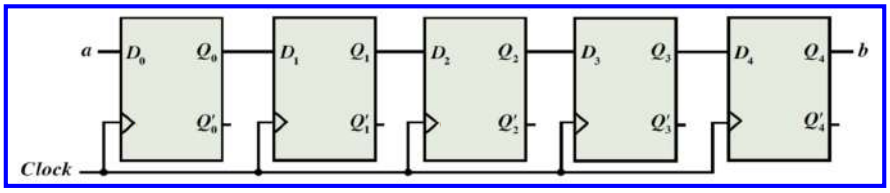
Καταχωρητής ολίσθησης με σειριακή είσοδο & έξοδο

- Ο καταχωρητής ολίσθησης με σειριακή είσοδο και έξοδο είναι μία αλυσίδα από flip-flop, 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 flip-flop, ύστερα από n παλμούς ρολογιού, το δυαδικό ψηφίο που εισήλθε πρώτο θα είναι αποθηκευμένο στο τελευταίο flip-flop της αλυσίδας, ενώ το δυαδικό ψηφίο που εισήλθε τελευταίο θα είναι αποθηκευμένο στο πρώτο flip-flop.

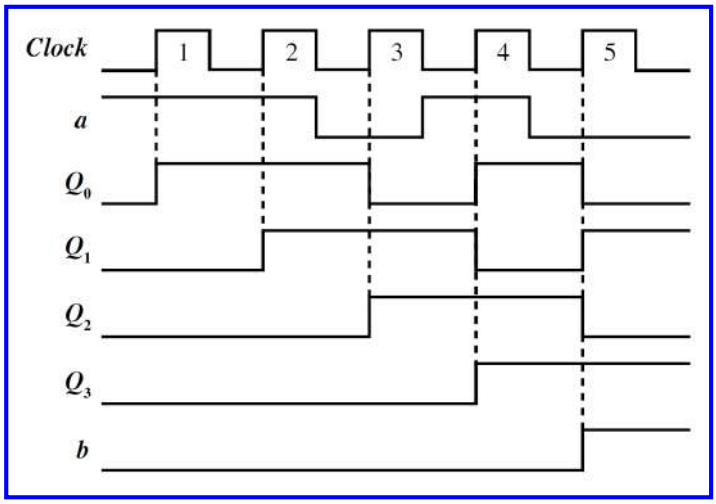


Καταχωρητής ολίσθησης με σειριακή είσοδο & έξοδο

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

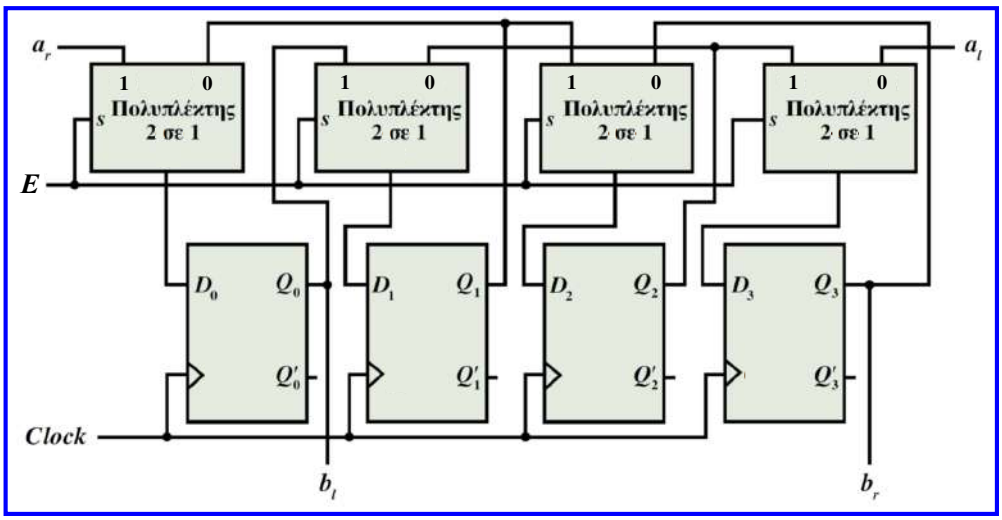


Μετά από 5 παλμούς του σήματος ρολογιού, το περιεχόμενο του καταχωρητή ολίσθησης (δηλ. οι τιμές των εξόδων των flip-flop) είναι η ακολουθία $bQ_3Q_2Q_1Q_0$ = 11010, η οποία συμπίπτει με την εισερχόμενη ακολουθία



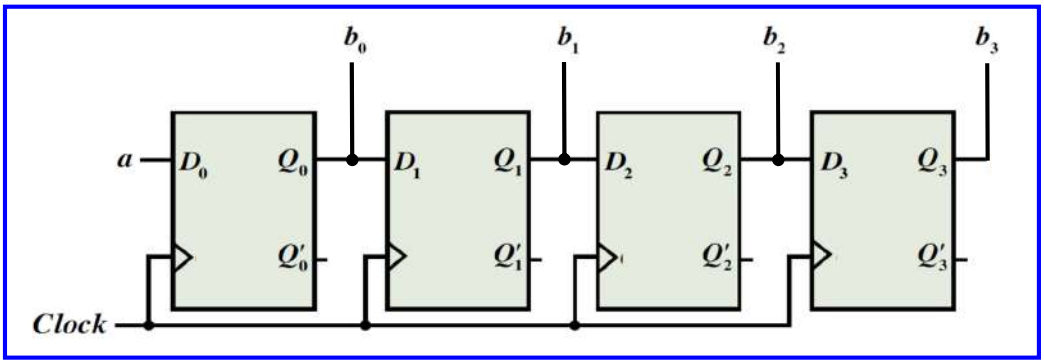
Αμφίδρομος καταχωρητής ολίσθησης

- 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 flip-flop.
- Η είσοδος E ελέγχει τη φορά της ολίσθησης.
- Όταν E = 1, στην είσοδο κάθε flip-flop συνδέεται η έξοδος του προηγούμενου και εκτελείται ολίσθηση προς τα δεξιά, με σειριακή είσοδο a_r και σειριακή έξοδο b_r.
- Όταν E = 0, στην είσοδο κάθε flip-flop συνδέεται η έξοδος του επόμενου και εκτελείται ολίσθηση προς τα αριστερά, με σειριακή είσοδο a_l και σειριακή έξοδο b_l.



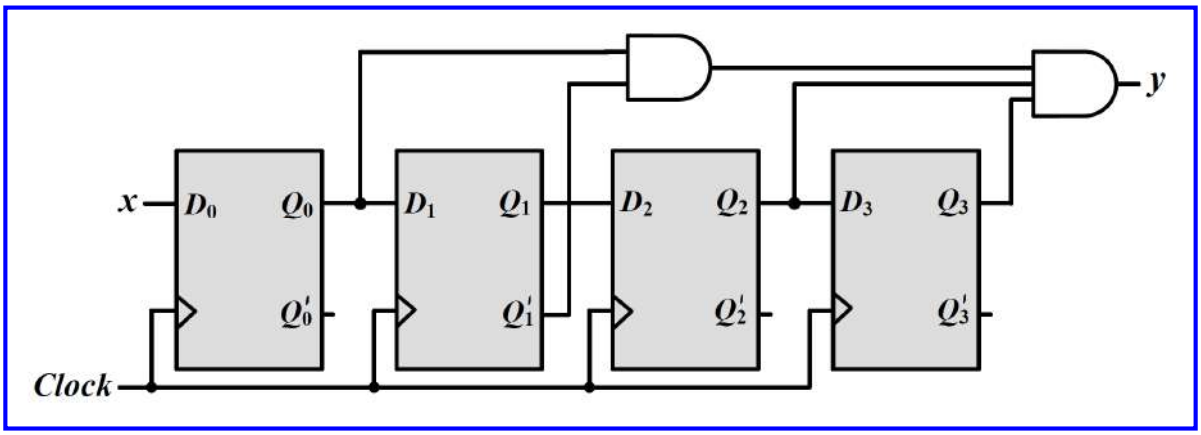
Καταχωρητής με σειριακή είσοδο & παράλληλη έξοδο

- 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-flop που συμμετέχουν στο καταχωρητή.



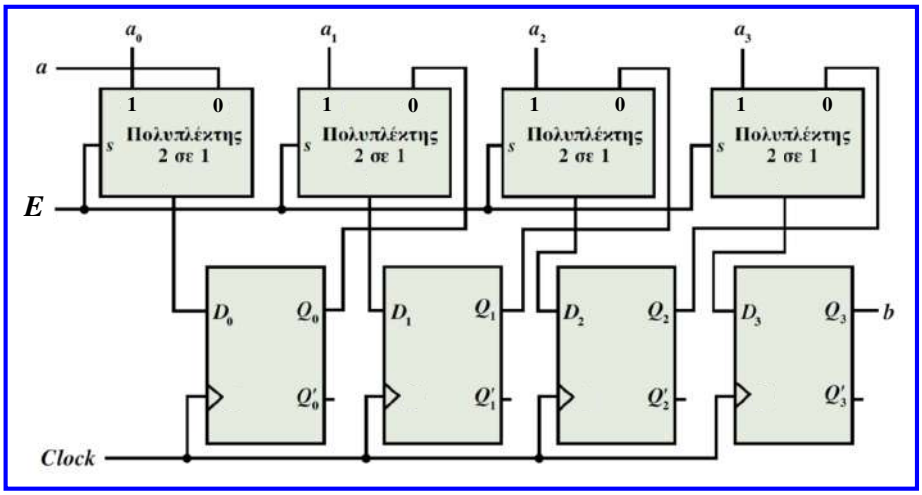
Καταχωρητής με σειριακή είσοδο & παράλληλη έξοδο

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 E ελέγχει την παράλληλη είσοδο (φόρτωση) δεδομένων.
- Όταν E = 0, τότε η λειτουργία του κυκλώματος είναι όμοια με εκείνη του καταχωρητή ολίσθησης με σειριακή είσοδο (a) και σειριακή έξοδο (b).
- Όταν E = 1, τότε ενεργοποιείται η παράλληλη φόρτωση των δεδομένων, μέσω των γραμμών εισόδου a_0, a_1, a_2 και a_3.

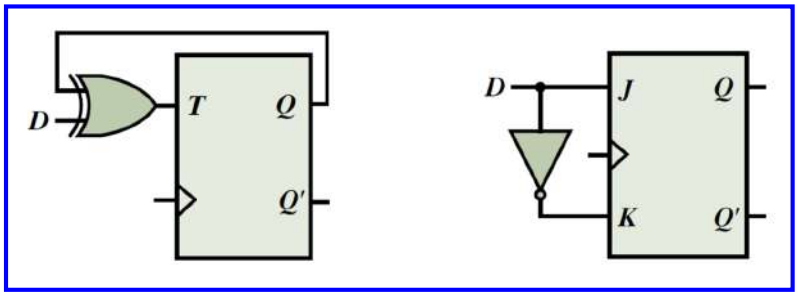


Υλοποίηση καταχωρητών με τα υπόλοιπα flip-flops

- Για τη σύνθεση των διάφορων παραλλαγών καταχωρητών μπορούν να χρησιμοποιηθούν JK ή T flip-flop αντί για D flip-flop με βάση απλές μετατροπές μεταξύ των flip-flop.
- Για να μετατρέψουμε ένα T ή JK σε D flip-flop, 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 D flip-flop.

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

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

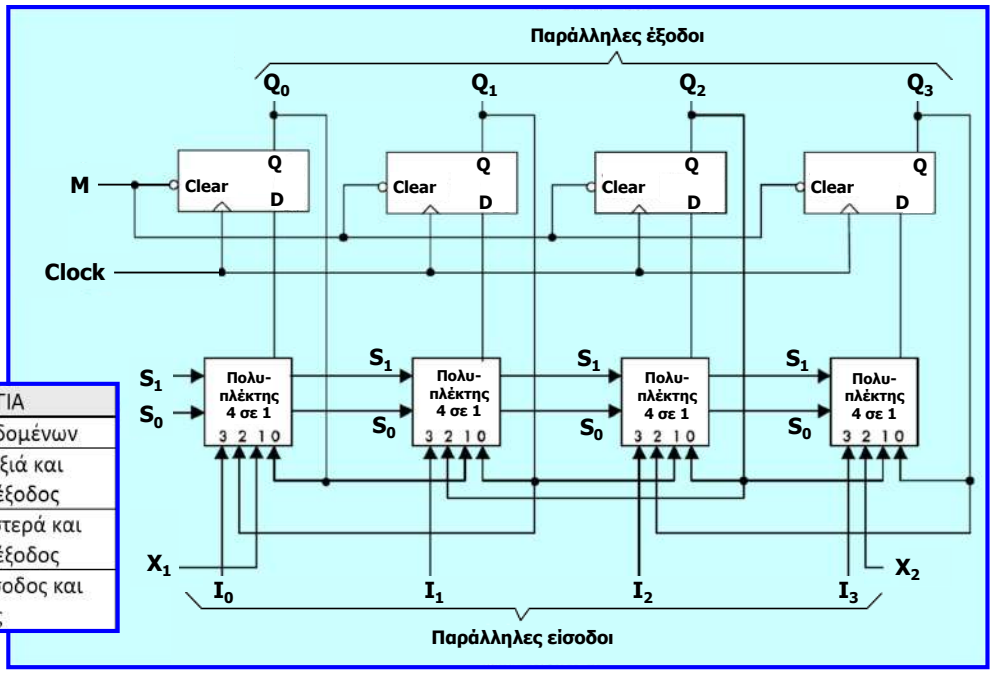


Υλοποίηση σύνθετων καταχωρητών

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

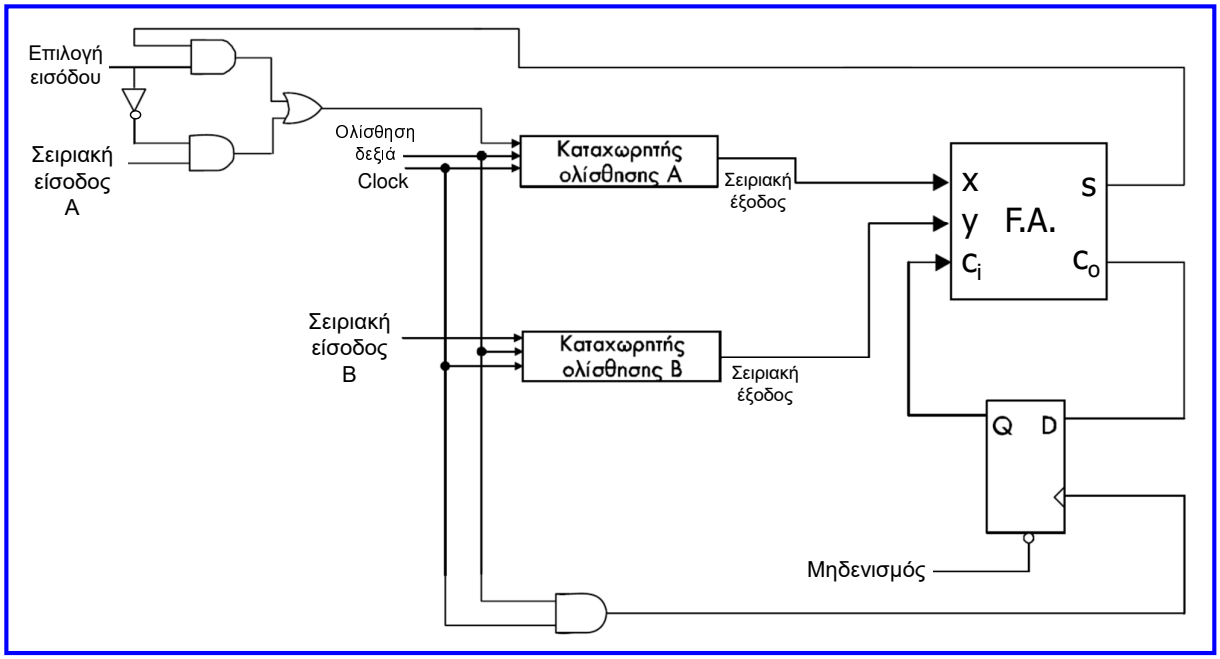
Αμφίδρομος καταχωρητής παράλληλης εισόδου και εξόδου με δυνατότητα σειριακής ολίσθησης προς τα δεξιά και προς τα αριστερά

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



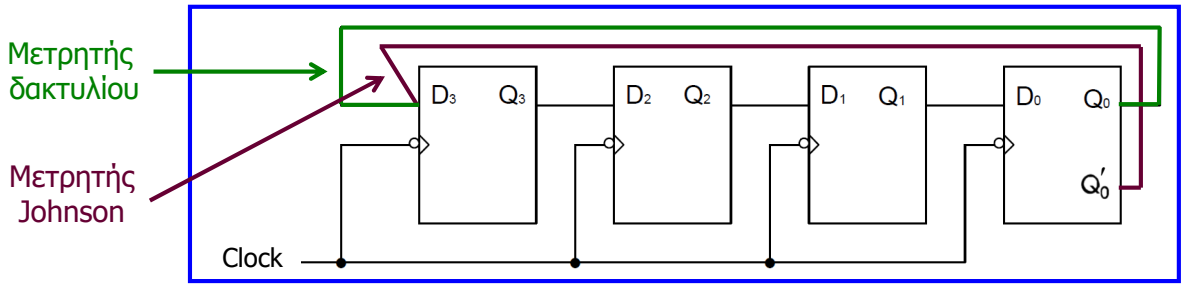
Σειριακός αθροιστής με καταχωρητές ολίσθησης

Οι προσθετέοι αποθηκεύονται σειριακά (με πρώτο το LSB) στους καταχωρητές A και B, το άθροισμα αποθηκεύεται ένα ψηφίο ανά παλμό ρολογιού στον καταχωρητή A και το κρατούμενο κάθε θέσης αποθηκεύεται προσωρινά στο D flip-flop



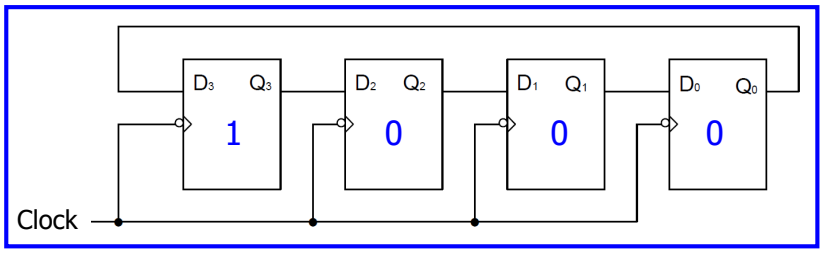
Κυκλικοί καταχωρητές

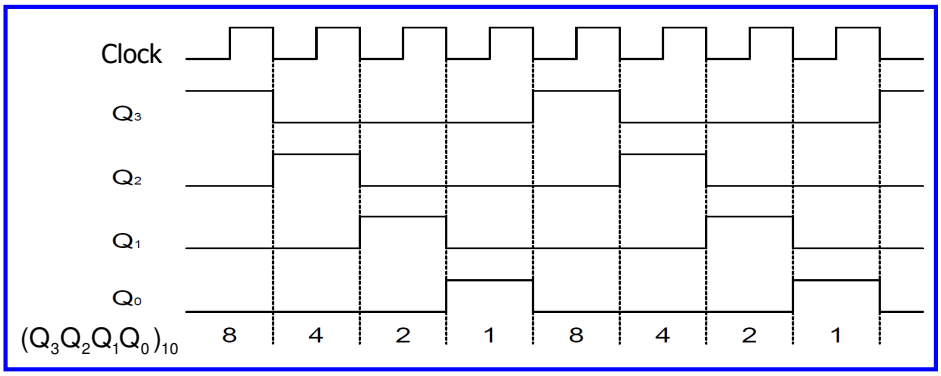
- Για τη δημιουργία των σημάτων χρονισμού που επιτελούν τον έλεγχο ψηφιακών συστημάτων και καθορίζουν τη χρονική ακολουθία των λειτουργιών τους, χρησιμοποιούνται σύγχρονα ακολουθιακά κυκλώματα που αναφέρονται ως κυκλικοί καταχωρητές (circular registers) ή μετρητές ολίσθησης (shift counters) σε συνδυασμό με λογικές πύλες αποκωδικοποίησης.
- Τα πιο συνηθισμένα από αυτά είναι ο μετρητής δακτυλίου (ring counter) και ο ανεστραμμένος μετρητής δακτυλίου (twisted ring counter) που αναφέρεται και ως μετρητής Johnson.
- Ο πρώτος είναι ένας καταχωρητής ολίσθησης του οποίου η σειριακή έξοδος συνδέεται στη σειριακή είσοδό του, ενώ στον δεύτερο η συμπληρωματική σειριακή έξοδός του συνδέεται στη σειριακή είσοδό του.



Μετρητής δακτυλίου

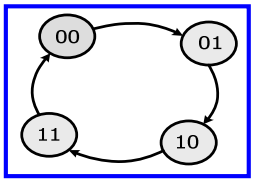
- Αρχικά, στο πρώτο flip-flop φορτώνεται η κατάσταση 1, ενώ στα υπόλοιπα η κατάσταση 0.
- Το περιεχόμενο κάθε flip-flop μεταφέρεται κυκλικά στο επόμενο σε κάθε κατερχόμενη ακμή του σήματος ρολογιού.
- Για τη δημιουργία n διαφορετικών σημάτων χρονισμού, απαιτούνται n flip-flop.

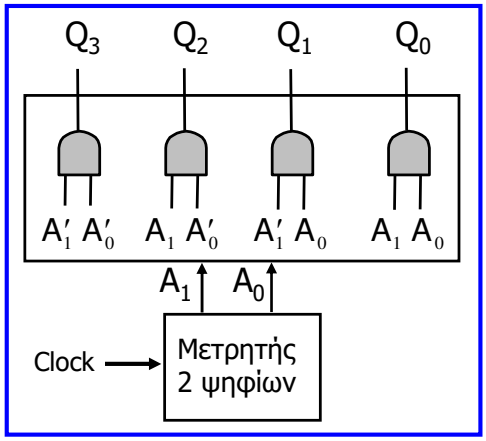


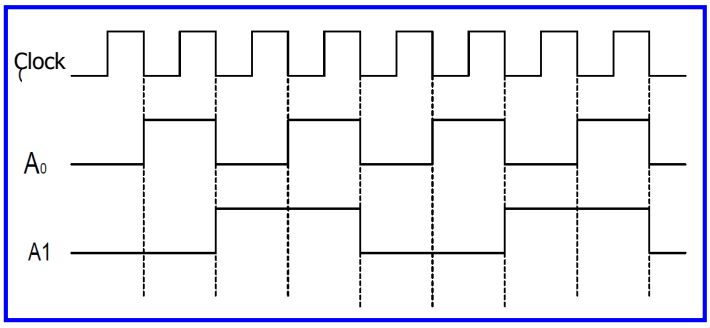


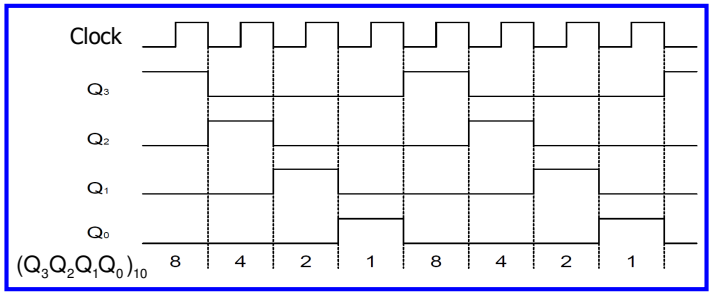
Μετρητής δακτυλίου

Τα ίδια σήματα χρονισμού μπορούν παραχθούν και από μετρητή 2 ψηφίων (δηλαδή, ένα σύγχρονο ακολουθιακό κύκλωμα με 2 flip-flop, που διατρέχει επαναλαμβανόμενα τις καταστάσεις 0 έως 3), εάν οι έξοδοί του τροφοδοτήσουν 4 πύλες AND (αποκωδικοποίησης):



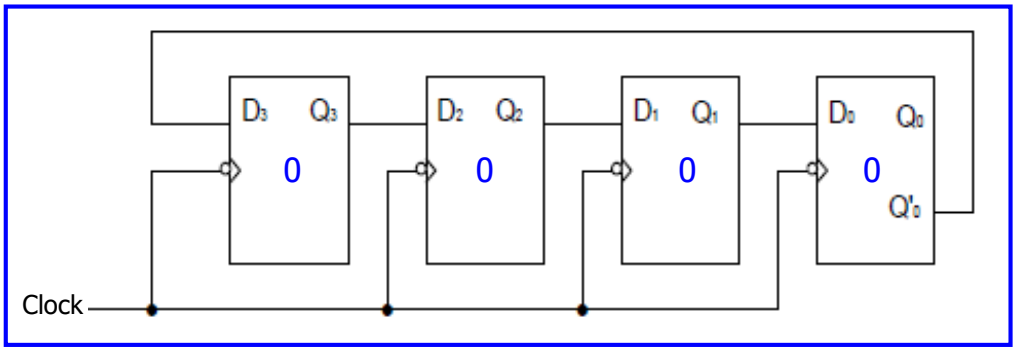




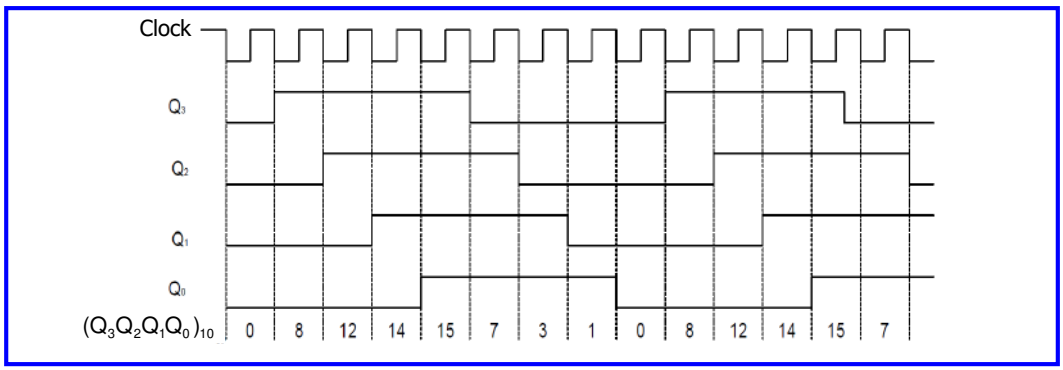


Μετρητής Johnson

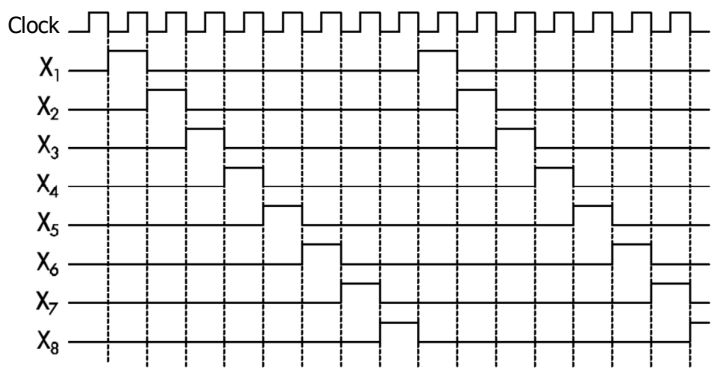
- Ο αριθμός των διαφορετικών καταστάσεων της ακολουθίας που παράγει ένας μετρητής δακτυλίου μπορεί να διπλασιαστεί, εάν συνδεθεί ως μετρητής Johnson (δηλ. με σύνδεση της συμπληρωματικής εξόδου του τελευταίου flip-flop με την είσοδο του 1ου flip-flop).
- Ο μετρητής αυτός προκαλεί ολίσθηση των περιεχομένων του μία θέση δεξιά σε κάθε κατερχόμενη ακμή του ρολογιού και ταυτόχρονα η συμπληρωματική τιμή του τελευταίου flip-flop μεταφέρεται στο πρώτο flip-flop.
- Ένας μετρητής Johnson n ψηφίων δημιουργεί ακολουθία 2n σημάτων χρονισμού με χρήση 2n πυλών AND αποκωδικοποίησης.



Μετρητής Johnson

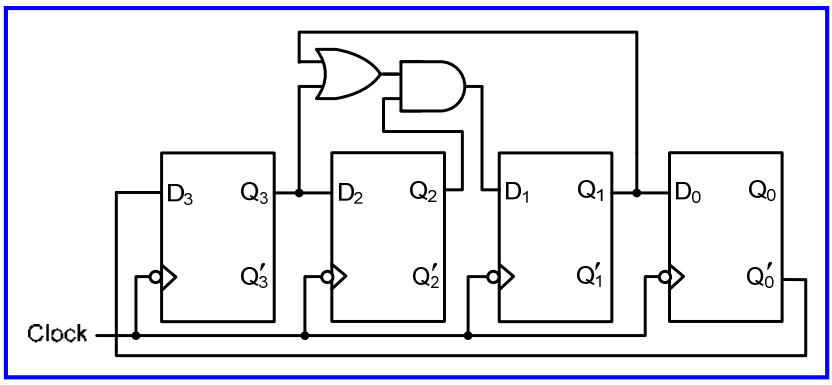


Q_3	Q_2	Q_1	Q_0	Πύλη AND για την αποκωδικοποίηση
0	0	0	0	$X_1 = Q'_3\ Q'_0$
1	0	0	0	$X_2 = Q_3\ Q'_2$
1	1	0	0	$X_3 = Q_2\ Q'_1$
1	1	1	0	$X_4 = Q_1\ Q'_0$
1	1	1	1	$X_5 = Q_3\ Q_0$
0	1	1	1	$X_6 = Q'_3\ Q_2$
0	0	1	1	$X_7 = Q'_2\ Q_1$
0	0	0	1	$X_8 = Q'_1\ Q_0$



Μετρητής Johnson

- Από τις 16 δυνατές καταστάσεις που μπορούν να δημιουργηθούν από τις 4 μεταβλητές κατάστασης (Q_3, Q_2, Q_1, Q_0) ο μετρητής Johnson διατρέχει μόνο τις 8 καταστάσεις.
- Από την ανάλυση του κυκλώματος προκύπτει ότι εάν αυτό βρεθεί σε μια αχρησιμοποίητη κατάσταση, θα επιμείνει στη μετάβαση από τη μία αχρησιμοποίητη κατάσταση στην άλλη και δεν θα επιστρέψει ποτέ σε μία από τις 8 επιτρεπτές καταστάσεις.
- Για να αρθεί η δυσλειτουργία, μία λύση είναι να παρέμβουμε στο κύκλωμα και να το μετατρέψουμε αποσυνδέοντας την έξοδο του 2ου flip-flop από την είσοδο του 3ου flip-flop και θέτοντας ως είσοδο στο 3ο flip-flop τη συνάρτηση $(Q_3 + Q_1)Q_2$.



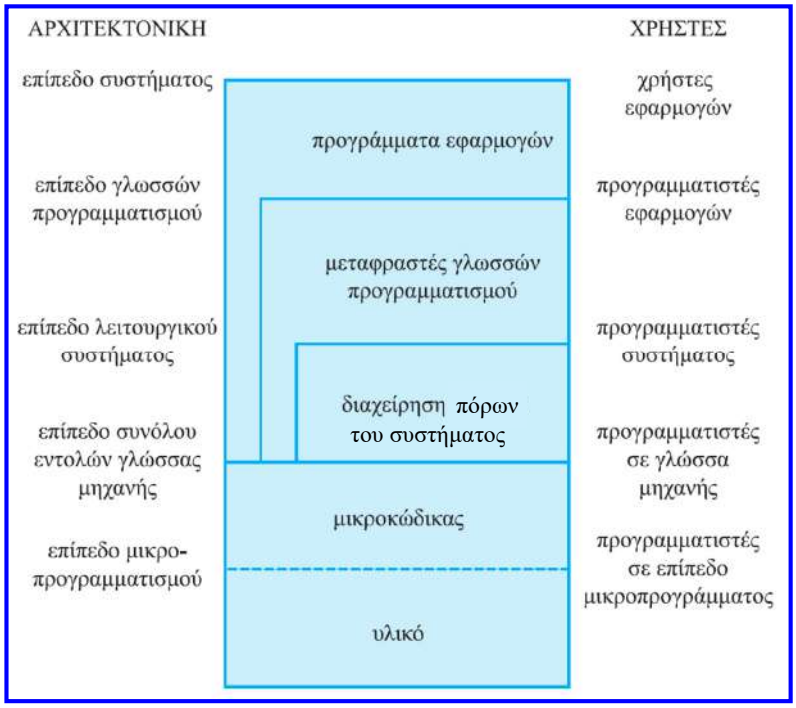
2. Αρχιτεκτονική υπολογιστών

Υπολογιστικό σύστημα

- Υπολογιστικό σύστημα (ή υπολογιστής) είναι ένα σύστημα επεξεργασίας πληροφοριών που ως είσοδο λαμβάνει δεδομένα προς επεξεργασία και προγράμματα που καθορίζουν το είδος της επεξεργασίας και παράγει ως έξοδο τα αποτελέσματα της επεξεργασίας.
- Κάθε υπολογιστικό σύστημα αποτελείται από υλικό (hardware) και λογισμικό (software).
- Το υλικό περιλαμβάνει τις μονάδες που συνιστούν το υπολογιστικό σύστημα (κεντρική μονάδα επεξεργασίας, μνήμες, περιφερειακές μονάδες).
- Το λογισμικό περιλαμβάνει το σύνολο των προγραμμάτων που εκτελούνται στο υπολογιστικό σύστημα και συνίσταται από:
  - ✓ το λογισμικό συστήματος (system software) που περιλαμβάνει το λειτουργικό σύστημα (operating system) που είναι υπεύθυνο για την καλύτερη κατανομή και εκμετάλλευση του υλικού και βοηθητικά προγράμματα (utilities),
  - ✓ το διαγνωστικό λογισμικό (diagnostic software) που περιλαμβάνει προγράμματα ελέγχου της ορθής λειτουργίας των μονάδων του υπολογιστή και
  - ✓ το λογισμικό εφαρμογών (application software) που περιλαμβάνει τα προγράμματα που αφορούν οποιαδήποτε εφαρμογή των χρηστών (βάσεις δεδομένων, CAD, εφαρμογές γραφείου και πολυμέσων, δικτυακές εφαρμογές κ.ά).

Αρχιτεκτονική υπολογιστικού συστήματος

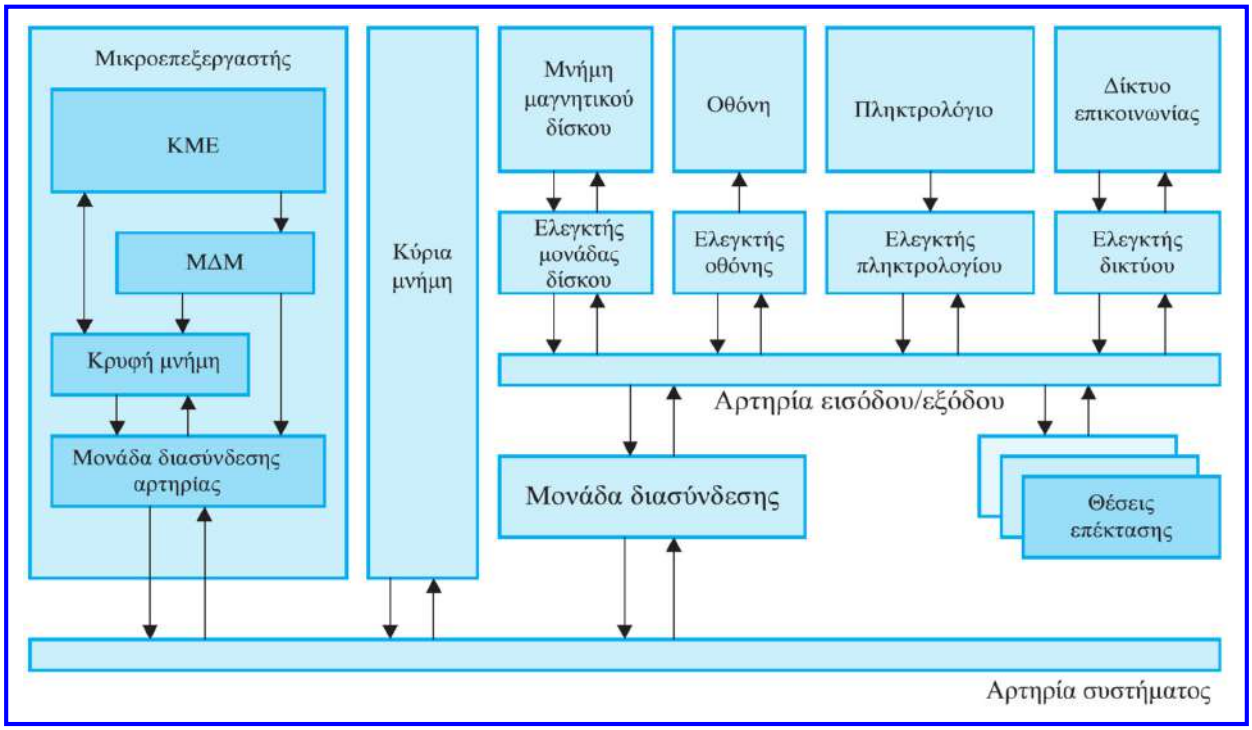
- Ένα υπολογιστικό σύστημα μπορεί να περιγραφεί σε διάφορα επίπεδα.
- Αρχιτεκτονική υπολογιστών σε ένα επίπεδο είναι η συμπεριφορά, οι ιδιότητες και οι δυνατότητές του στο επίπεδο αυτό, που καθορίζονται από τη δομή και την οργάνωσή του κάτω από το εν λόγω επίπεδο.
- Η δομή περιγράφει τις μονάδες που απαρτίζουν το σύστημα και τον τρόπο διασύνδεσής τους, ενώ η οργάνωση περιγράφει την αλληλεπίδραση και τη διαχείριση των μονάδων.



Αρχιτεκτονική υπολογιστικού συστήματος

- Η αρχιτεκτονική σε επίπεδο συνόλου εντολών γλώσσας μηχανής αποτελεί το όριο διασύνδεσης υλικού και λογισμικού ενός υπολογιστικού συστήματος.
- Η αρχιτεκτονική στο επίπεδο αυτό, περιγράφει την οργάνωση της κύριας μνήμης του συστήματος, τους καταχωρητές, τα διαθέσιμα είδη δεδομένων, τον τρόπο κωδικοποίησης και παράστασής τους, τη μορφή και το σύνολο των εντολών και τους τρόπους προσπέλασης δεδομένων (τρόποι διευθυνσιοδότησης, addressing modes).
- Στη συνέχεια θα ασχοληθούμε με την αρχιτεκτονική του υπολογιστικού συστήματος που περικλείει την αρχιτεκτονική σε επίπεδο συνόλου εντολών γλώσσας μηχανής, τη δομή, την οργάνωση, την υλοποίηση και την απόδοση.
- Η δομή περιγράφει τις μονάδες υλικού του συστήματος και τον τρόπο διασύνδεσής τους.
- Η οργάνωση περιγράφει την αλληλεπίδραση μεταξύ των μονάδων αυτών.
- Η υλοποίηση αφορά τον ακριβή σχεδιασμό των μονάδων του συστήματος και την τεχνολογία που χρησιμοποιείται.
- Η απόδοση ενός υπολογιστικού συστήματος καθορίζεται από τη δομή, την οργάνωση και την υλοποίησή του.

Δομή προσωπικού υπολογιστή



Κεντρική μονάδα επεξεργασίας (ΚΜΕ)

- Η ΚΜΕ που υλοποιείται σε ολοκληρωμένο κύκλωμα και αναφέρεται ως επεξεργαστής (processor), αποτελείται από την μονάδα επεξεργασίας δεδομένων (data path unit) και τη μονάδα ελέγχου (control unit).
- Στο ίδιο ολοκληρωμένο κύκλωμα μπορεί να συνυπάρχουν η κρυφή μνήμη (cache memory) και η μονάδα διαχείρισης μνήμης (ΜΔΜ, memory management unit).
- Η μονάδα επεξεργασίας δεδομένων αποτελείται από καταχωρητές και κυκλώματα εκτέλεσης λογικών και αριθμητικών πράξεων και συνήθως περιλαμβάνει υπομονάδες επεξεργασίας δεδομένων σταθερής και κινητής υποδιαστολής.
- Οι καταχωρητές διακρίνονται σε καταχωρητές γενικού και ειδικού σκοπού.
- Οι καταχωρητές γενικού σκοπού χρησιμοποιούνται για διάφορες λειτουργίες ανάλογα με την επιθυμία του προγραμματιστή, όπως για παράδειγμα η αποθήκευση δεδομένων ή η αποθήκευση ενδιάμεσων αποτελεσμάτων αριθμητικών ή λογικών πράξεων.
- Καταχωρητές ειδικού σκοπού είναι ο μετρητής προγράμματος (program counter) που περιέχει τη διεύθυνση της θέσης μνήμης της επόμενης προς εκτέλεση εντολής, ο καταχωρητής κατάστασης (state register) που αποθηκεύει ειδικά στοιχεία υπολογισμών (για παράδειγμα υποδεικνύει το πρόσημο του αποτελέσματος ή την ύπαρξη υπερχείλισης σε μια πράξη), οι καταχωρητές δείκτη (index registers) που χρησιμοποιούνται για τον καθορισμό διευθύνσεων κ.ά.

Κεντρική μονάδα επεξεργασίας (ΚΜΕ)

- Το πλήθος και το μέγεθος των καταχωρητών ποικίλει από επεξεργαστή σε επεξεργαστή.
- Η μονάδα ελέγχου αναλύει την εντολή που πρόκειται να εκτελεστεί και παράγει ακολουθίες σημάτων ελέγχου που έχουν ως συνέπεια τη μεταφοράς δεδομένων μεταξύ των μονάδων του υπολογιστή και την εκτέλεση συγκεκριμένων πράξεων.
- Η ΚΜΕ εκτελεί μια εντολή ακολουθώντας στη γενική περίπτωση μια σειρά βημάτων:
  1. Προσκόμιση στην ΚΜΕ την εντολή που είναι αποθηκευμένη στη θέση μνήμης που δείχνει ο μετρητής του προγράμματος.
  2. Αλλαγή του περιεχομένου του μετρητή προγράμματος, ώστε να περιέχει τη θέση μνήμης στην οποία είναι αποθηκευμένη η επόμενη προς εκτέλεση εντολή.
  3. Ανάλυση της εντολής και έλεγχος εάν η εντολή απαιτεί δεδομένα από τη μνήμη και αν ναι, προσδιορισμός της διεύθυνσης που είναι αποθηκευμένα.
  4. Προσκόμιση των δεδομένων σε κάποιους από τους καταχωρητές της ΚΜΕ.
  5. Εκτέλεση της εντολής.
  6. Αποθήκευση των αποτελεσμάτων.
  7. Επιστροφή στο πρώτο βήμα, ώστε να αρχίσει η εκτέλεση της επόμενης εντολής.

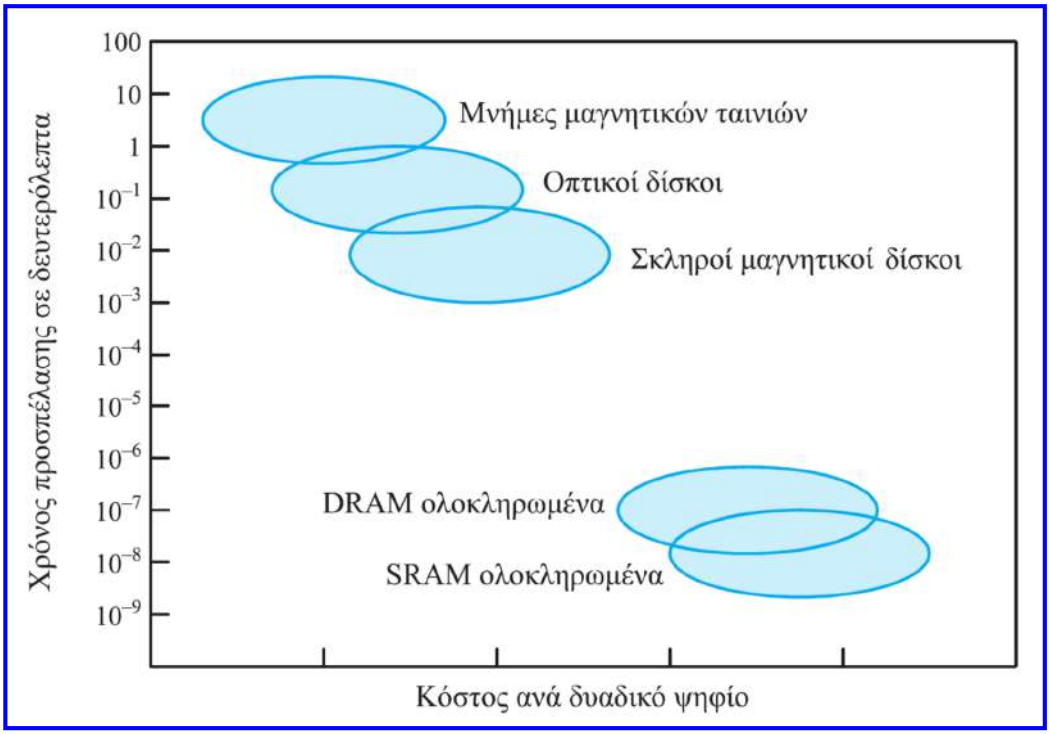
Σύστημα μνήμης

- Η κύρια μνήμη αποτελείται από έναν αριθμό θέσεων που σε καθεμία από αυτές αποθηκεύεται ένα τμήμα πληροφορίας (εντολή, τμήμα εντολής ή δεδομένων) με τη μορφή ακολουθίας δυαδικών ψηφίων.
- Το πλήθος των δυαδικών ψηφίων που μπορεί να αποθηκευθεί σε μία θέση μνήμης διαφέρει από υπολογιστή σε υπολογιστή.
- Σε κάθε θέση μνήμης αντιστοιχεί ένας αριθμός που αναφέρεται ως φυσική διεύθυνση της θέσης μνήμης και χρησιμοποιείται για να είναι δυνατή η αναφορά στη θέση μνήμης.
- Χρόνος προσπέλασης (access time) είναι ο χρόνος από τη λήψη μιας απαίτησης ανάγνωσης ή εγγραφής μέχρι τη χρονική στιγμή που η απαιτούμενη πληροφορία είναι διαθέσιμη στις εξόδους ή εισόδους της μνήμης.
- Ο χρόνος προσπέλασης ανάγνωσης ή εγγραφής καθορίζει την ταχύτητα της κύριας μνήμης και εξαρτάται από την τεχνολογία κατασκευής, το μέγεθος και τον τρόπο σχεδιασμού της.
- Χρόνος κύκλου μνήμης (memory cycle time) είναι ο χρόνος από την έναρξη μιας προσπέλασης μέχρι τη χρονική στιγμή που επιτρέπεται η έναρξη νέας προσπέλασης.
- Ρυθμός μεταφοράς δεδομένων (data transfer rate) είναι το μέγιστο ποσό πληροφορίας που μπορεί να μεταφερθεί προς ή από την κύρια μνήμη κάθε δευτερόλεπτο, ισούται με το αντίστροφο του χρόνου κύκλου μνήμης και μετριέται σε bytes ανά sec

Σύστημα μνήμης

- Η ΚΜΕ επικοινωνεί με την κύρια μνήμη μέσω ενός καταχωρητή διευθύνσεων και ενός καταχωρητή δεδομένων, οι οποίοι μπορούν να είναι καταχωρητές ειδικού ή γενικού σκοπού, ανάλογα με τον επεξεργαστή.
- Όταν πρόκειται να γίνει ανάγνωση μιας λέξης από τη μνήμη, η αντίστοιχη διεύθυνση τοποθετείται στον καταχωρητή διευθύνσεων και παράγεται ένα σήμα από τη μονάδα ελέγχου προς την μνήμη που δηλώνει την απαίτηση ανάγνωσης.
- Μετά από έναν χρόνο προσπέλασης, η λέξη μεταφέρεται στον καταχωρητή δεδομένων.
- Αντίστοιχη είναι και η διαδικασία εγγραφής μιας λέξης στην κύρια μνήμη.
- Εάν το πλήθος των ψηφίων του καταχωρητή διευθύνσεων είναι n, τότε το μέγιστο πλήθος των θέσεων μνήμης στις οποίες μπορεί να διενεργηθεί άμεση αναφορά είναι 2^n.
- Η βοηθητική μνήμη χρησιμοποιείται για την αποθήκευση προγραμμάτων και δεδομένων που δεν χρειάζονται συνεχώς από την ΚΜΕ ή ως μνήμη υπερχείλισης για τις περιπτώσεις όπου ξεπερνιέται η χωρητικότητα της κύριας μνήμης.
- Η πληροφορία που είναι αποθηκευμένη στη βοηθητική μνήμη, δεν είναι άμεσα προσπελάσιμη από την ΚΜΕ, αλλά θα πρέπει πρώτα να μεταφερθεί στην κύρια μνήμη.
- Για την κύρια μνήμη χρησιμοποιούνται ημιαγωγικές μνήμες άμεσης προσπέλασης (RAM, random access memories), ενώ για τη βοηθητική μνήμη χρησιμοποιούνται μαγνητικά και οπτικά μέσα αποθήκευσης.

Σύστημα μνήμης

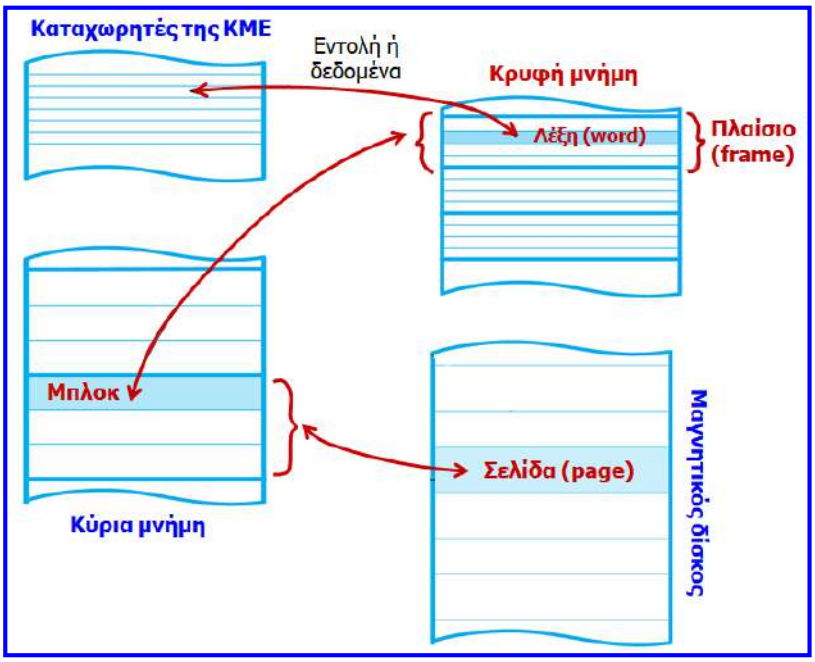


Σύστημα μνήμης

- Η κύρια μνήμη είναι σχετικά αργή προκαλώντας αργοπορία στην ΚΜΕ με την κατάσταση να γίνεται χειρότερη όταν η απαιτούμενη πληροφορία πρέπει να μεταφερθεί στην κύρια μνήμη από τη βοηθητική.
- Για τη βελτίωση της ταχύτητας, χρησιμοποιείται η κρυφή μνήμη (cache memory) που είναι σχετικά μικρής χωρητικότητας και χρησιμοποιείται για την αποθήκευση πληροφορίας που αναμένεται ότι θα χρησιμοποιηθεί άμεσα ή με μεγάλη συχνότητα στο μέλλον.
- Η βελτίωση επέρχεται διότι, η πλειονότητα των προγραμμάτων χαρακτηρίζεται από τοπικότητα (locality) αναφορών, που σημαίνει ότι η πληροφορία (εντολές και δεδομένα) που χρησιμοποιήθηκε πρόσφατα είναι πιθανόν να ξαναχρησιμοποιηθεί στο άμεσο μέλλον, καθώς και ότι η πληροφορία που βρίσκεται κοντά στην πληροφορία που χρησιμοποιείται, είναι πιθανό να χρησιμοποιηθεί στο άμεσο μέλλον.
- Η κρυφή μνήμη συνήθως υλοποιείται στο ίδιο ολοκληρωμένο κύκλωμα με την ΚΜΕ.
- Στην κρυφή μνήμη αποθηκεύονται αντίγραφα μέρους της πληροφορίας της κύριας μνήμης και όταν η απαιτούμενη από την ΚΜΕ πληροφορία βρίσκεται στην κρυφή μνήμη, αυτή προσπελαύνεται πολύ γρήγορα, αποφεύγοντας την καθυστέρηση της προσπέλασης της κύριας μνήμης.
- Εάν η ζητούμενη πληροφορία δεν υπάρχει στην κρυφή μνήμη, ένα τμήμα (block) πληροφορίας που περιέχει τη ζητούμενη πληροφορία μεταφέρεται στην κρυφή μνήμη.

Σύστημα μνήμης

- Το ποσοστό προσπελάσεων που ικανοποιούνται από την κρυφή μνήμη, χωρίς προσπέλαση της κύριας, αναφέρεται ως λόγος ή ποσοστό επιτυχίας.
- Η κρυφή μνήμη μπορεί να είναι ενιαία (unified cache) για δεδομένα και εντολές ή ξεχωριστή για την αποθήκευση δεδομένων (data cache) και εντολών (instruction cache).



Σύστημα μνήμης

- Η ιδεατή μνήμη (virtual memory) δημιουργεί στον χρήστη την αίσθηση ότι η κύρια μνήμη και ένα μέρος της μνήμης του δίσκου αποτελούν ενιαία μνήμη, έτσι ώστε να παρέχεται ικανοποίηση των αυξημένων απαιτήσεων των εφαρμογών των χρηστών και περιορισμός του κόστους που επιφέρει η αύξηση χωρητικότητας της κύριας μνήμης.
- Στην περίπτωση χρήσης της τεχνικής της ιδεατής μνήμης, οι διευθύνσεις που παράγονται από την ΜΚΕ αναφέρονται ως λογικές διευθύνσεις (logical addresses).
- Η μονάδα διαχείρισης μνήμης (ΜΔΕ, memory management unit, MMU) λαμβάνει τις λογικές διευθύνσεις και παράγει φυσικές διευθύνσεις.
- Όταν η ζητούμενη από την ΚΜΕ πληροφορία βρίσκεται στην κύρια μνήμη, προσπελαύνεται άμεσα.
- Όταν η ζητούμενη από την ΚΜΕ πληροφορία δεν βρίσκεται στην κύρια μνήμη, μεταφέρεται με ευθύνη του λειτουργικού συστήματος ένα τμήμα πληροφορίας που περιλαμβάνει τη ζητούμενη πληροφορία από τη μνήμη δίσκου στην κύρια μνήμη.
- Ο χρήστης (προγραμματιστής εφαρμογών) δεν αντιλαμβάνεται τη διαφορά, αφού η ιδεατή μνήμη είναι αντιληπτή μόνο από το λογισμικό συστήματος.
- Ωστόσο, η μεταφορά πληροφορίας από τη μνήμη του δίσκου στην κύρια μνήμη συνεπάγεται καθυστέρηση στο χρόνο προσπέλασης της πληροφορίας.

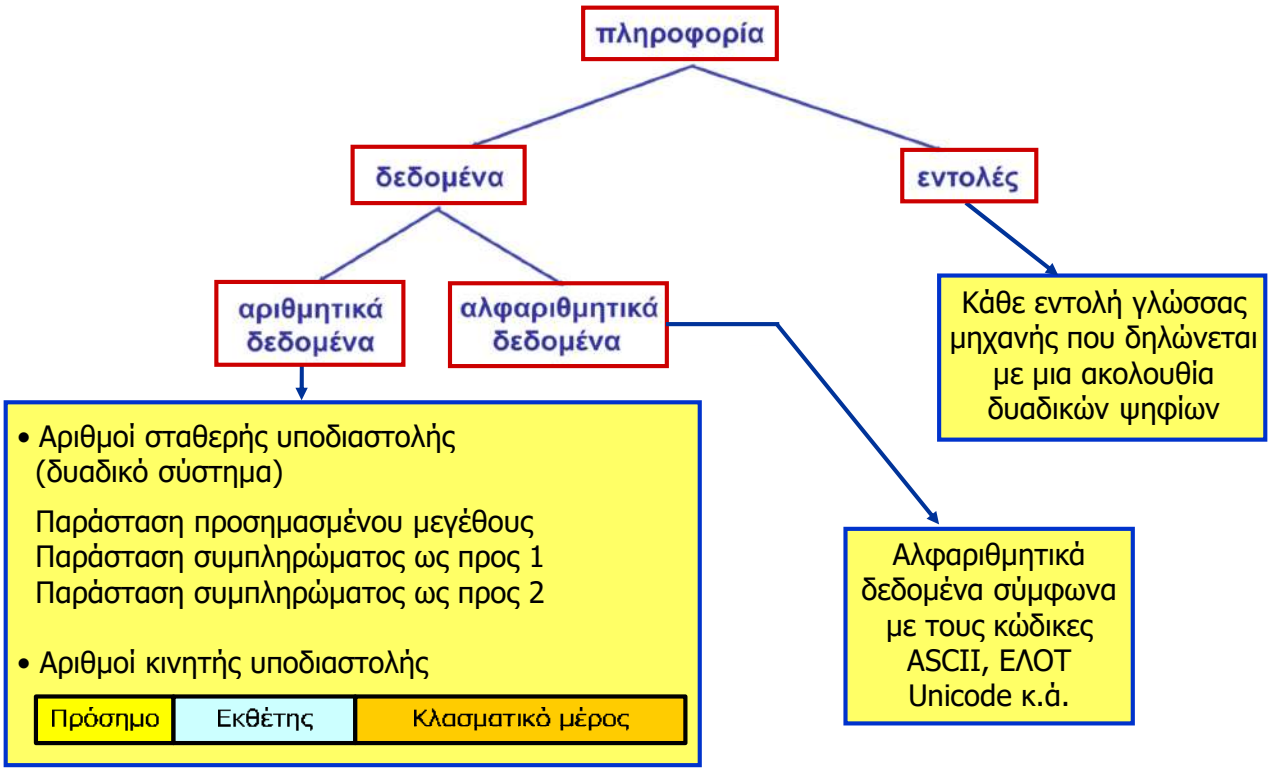
Μονάδες εισόδου-εξόδου

- Οι μονάδες εισόδου-εξόδου χρησιμοποιούνται για την επικοινωνία του υπολογιστή με το περιβάλλον του και διακρίνονται σε:
  - ✓ μονάδες εισόδου (π.χ. πληκτρολόγιο, μικρόφωνο, σαρωτής),
  - ✓ μονάδες εξόδου (π.χ. εκτυπωτής, οθόνη, μεγάφωνο),
  - ✓ μονάδες εισόδου και εξόδου (π.χ. ελεγκτής δικτύου),
  - ✓ και μονάδες αποθήκευσης πληροφορίας (π.χ. μαγνητικός δίσκος, οπτικός δίσκος).
- Οι μονάδες αυτής χρησιμοποιούνται συνήθως για την επικοινωνία χρήστη και υπολογιστή, καθώς και για την επικοινωνία μεταξύ υπολογιστών.
- Τα λειτουργικά χαρακτηριστικά των μονάδων εισόδου-εξόδου, διαφέρουν σημαντικά, ωστόσο το κοινό χαρακτηριστικό τους είναι η σχετικά μικρή ταχύτητα προσπέλασής τους, αν και αυτή διαφέρει σημαντικά από μονάδα σε μονάδα.

Απόδοση υπολογιστών

- Η απόδοση ενός υπολογιστή σχετίζεται με την ταχύτητα εκτέλεσης προγραμμάτων και εξαρτάται από την ταχύτητα του επεξεργαστή, αλλά και από το σύστημα μνήμης και το σύστημα διασύνδεσης των μονάδων του υπολογιστή.
- Ο όρος MIPS δηλώνει τα εκατομμύρια εντολών που εκτελούνται ανά δευτερόλεπτο.
- Ωστόσο, λόγω του ότι ο χρόνος εκτέλεσης μιας εντολής εξαρτάται από την πολυπλοκότητά της και το πλήθος εντολών ενός προγράμματος εξαρτάται από το διαθέσιμο σύνολο εντολών του υπολογιστή, το μέτρο αυτό δεν είναι αξιόπιστο για τη σύγκριση υπολογιστών με διαφορετικό σύνολο εντολών.
- Ο όρος MFLOPS δηλώνει τα εκατομμύρια πράξεων κινητής υποδιαστολής που εκτελούνται ανά δευτερόλεπτο.
- Το μέτρο αυτό, επίσης, δεν είναι αξιόπιστο για τη σύγκριση υπολογιστών, αφού αλλάζει ανάλογα με το ποσοστό των πράξεων κινητής υποδιαστολής σε σχέση με τις πράξεις σταθερής υποδιαστολής, αλλά και ανάλογα με το ποσοστό αργών πράξεων κινητής υποδιαστολής σε σχέση με τις γρήγορες.
- Ο ασφαλέστερος τρόπος εκτίμησης της απόδοσης ενός υπολογιστή, είναι η χρήση αντιπροσωπευτικών προγραμμάτων από διάφορα πεδία εφαρμογών, τα οποία αναφέρονται ως μετροπρογράμματα (benchmarks).
- Η πιο δημοφιλής συλλογή μετροπρογραμμάτων είναι αυτή που επιλέγεται από τον οργανισμό SPEC (system performance evaluation corporation).

Οργάνωση πληροφορίας στον υπολογιστή



Σφάλματα στην παράσταση αριθμητικών δεδομένων

- Τα σφάλματα κατά την παράσταση αριθμητικών δεδομένων οφείλονται στο ότι για την παράσταση των αριθμών είναι διαθέσιμος πεπερασμένος αριθμός δυαδικών ψηφίων.
- Η διαδικασία κατά την οποία παραλείπουμε τα λιγότερο σημαντικά ψηφία, όταν προκύπτει υπέρβαση του διαθέσιμου πλήθους ψηφίων, αναφέρεται ως αποκοπή (truncation).
- Το μέγεθος του σφάλματος εξαρτάται από τον αριθμό των απορριφθέντων ψηφίων.
- Το σφάλμα περικοπής μπορεί να μειωθεί με τη μέθοδο της στρογγυλοποίησης (rounding).
- Ένας τρόπος στρογγυλοποίησης είναι να προσθέσουμε στον αριθμό την ποσότητα $b^r / 2$, όπου b^r είναι το βάρος του λιγότερο σημαντικού ψηφίου που διατηρείται και b η βάση του αριθμητικού συστήματος και στη συνέχεια να διενεργήσουμε περικοπή.
- Το σφάλμα στρογγυλοποίησης (round-off error) προσδιορίζεται ως η διαφορά του αρχικού από τον στρογγυλοποιημένο αριθμό και περιορίζεται όταν οι επιτρεπτές παραστάσεις των αριθμών έχουν επαρκή ακρίβεια.

Παράδειγμα: Για τη στρογγυλοποίηση του δεκαδικού αριθμού 0.3729836 σε 4 δεκαδικά ψηφία, προσθέτουμε στον αριθμό την ποσότητα (10^{-4} / 2) και λαμβάνουμε 0.3730336. Περικόπτοντας τα τελευταία 3 ψηφία καταλήγουμε στον αριθμό 0.3730, ο οποίος είναι πιο κοντά στον αρχικό αριθμό από ότι ο αριθμός 0.3729 που προκύπτει με άμεση περικοπή 3 ψηφίων. Σφάλμα στρογγυλοποίησης = $|0.3729836 - 0.3730| = 1.64 \times 10^{-5}$.

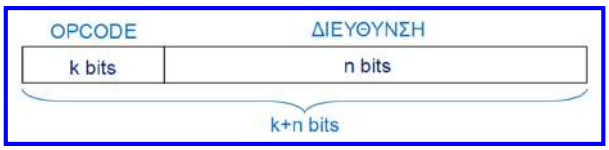
Εντολές γλώσσας μηχανής

- Το σύνολο των εντολών που αναγνωρίζονται και εκτελούνται άμεσα από τα κυκλώματα (μονάδες, υπομονάδες) ενός υπολογιστή αποτελούν τη γλώσσα μηχανής (machine language) του υπολογιστή.
- Κάθε εντολή καθορίζει τη λειτουργία (operation) που εκτελεί και τα τελούμενα (operands) που θα χρησιμοποιηθούν.
- Η λειτουργία και τα δεδομένα περιγράφονται από συγκεκριμένα πεδία της εντολής.
- Το πεδίο περιγραφής της λειτουργίας αναφέρεται ως κωδικός λειτουργίας (operation code).
- Τα πεδία των τελούμενων ή τελεστέων (operands) μπορεί να περιέχουν τα ίδια τα δεδομένα που θα χρησιμοποιηθούν ή διευθύνσεις θέσεων μνήμης και καταχωρητών, ώστε να προσδιοριστεί η θέση στην οποία βρίσκονται τα δεδομένα που θα χρησιμοποιηθούν.
- Σε κάθε εντολή διατίθενται πρόσθετη πληροφορία που καθορίζει τον τρόπο με τον οποίο θα χρησιμοποιηθεί η πληροφορία των τελούμενων, έτσι ώστε να προσδιοριστούν τα δεδομένα που θα χρησιμοποιηθούν.
- Οι εντολές διακρίνονται σε: εντολές μεταφοράς δεδομένων (για μεταφορά πληροφορίας μεταξύ των καταχωρητών της ΚΜΕ ή μεταξύ αυτών και της κύριας μνήμης), αριθμητικές εντολές, εντολές λογικών πράξεων (πράξεις άλγεβρας Boole, ολίσθηση), εντολές ελέγχου ροής προγράμματος ή διακλάδωσης (με ή χωρίς συνθήκη) και εντολές εισόδου-εξόδου.
- Μία εντολή μπορεί να ανήκει σε περισσότερες από μία κατηγορίες.

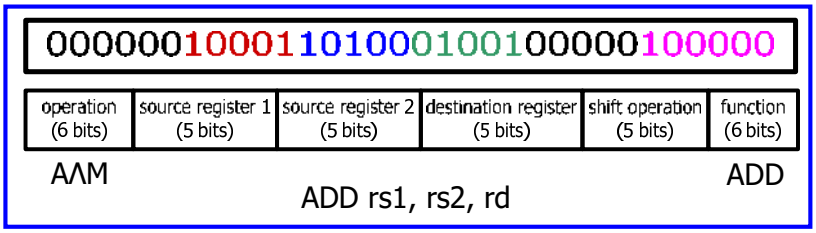
Εντολές γλώσσας μηχανής

Μορφές κωδικοποίησης εντολών γλώσσας μηχανής

Παράδειγμα 1: εντολή που επιτρέπει την εκτέλεση 2^k διαφορετικών πράξεων και τη δήλωση 2^n διευθύνσεων μνήμης:

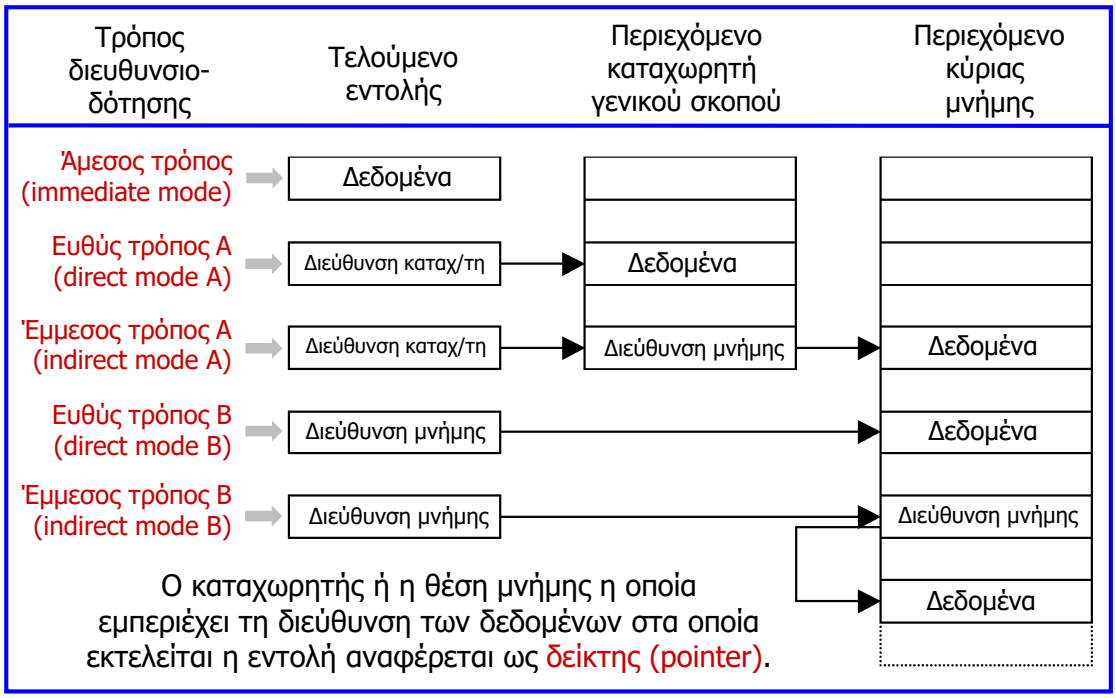


Παράδειγμα 2: εντολή 32 ψηφίων του επεξεργαστή MIPS I που προσθέτει τα περιεχόμενα των καταχωρητών 17 & 20 και τοποθετεί το άθροισμα στον καταχωρητή 9:



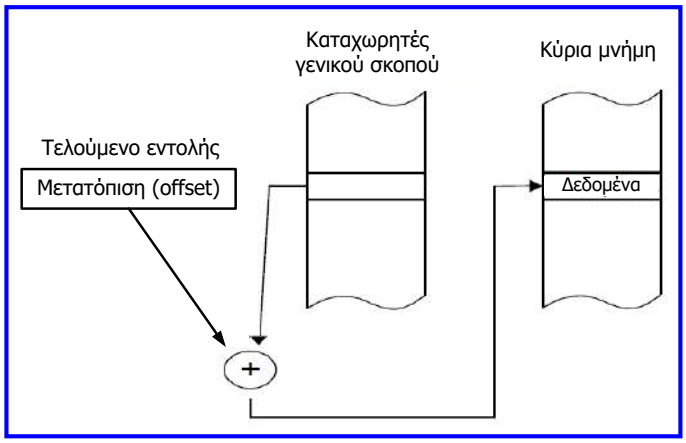
Βασικοί τρόποι διευθυνσιοδότησης

Τρόποι διευθυνσιοδότησης είναι οι τεχνικές που χρησιμοποιούνται και δηλώνονται στις εντολές του επεξεργαστή για την προσπέλαση των δεδομένων.



Σχετική διευθυνσιοδότηση

- Στη σχετική διευθυνσιοδότηση (relative addressing mode) το πεδίο του τελούμενου περιέχει μια σχετική διεύθυνση που αναφέρεται ως μετατόπιση (offset).
- Στην εντολή ορίζεται επίσης η διεύθυνση ενός καταχωρητή γενικού σκοπού, το άθροισμα του περιεχομένου του οποίου με τη μετατόπιση, αποτελεί τη διεύθυνση κύριας μνήμης όπου είναι αποθηκευμένα τα δεδομένα που θα χρησιμοποιηθούν.



- Η διεύθυνση του καταχωρητή μπορεί να μη δηλώνεται ρητά στην εντολή, αλλά να υπονοείται (για παράδειγμα σε κάποιες εντολές προστίθεται στη μετατόπιση το περιεχόμενο του μετρητή προγράμματος).

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 σωρού (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 τελούμενο, ενώ υπονοείται ότι το άλλο τελούμενο 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 τελούμενο είναι αποθηκευμένο σε μία διεύθυνση κύριας μνήμης, ενώ το άλλο τελούμενο 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

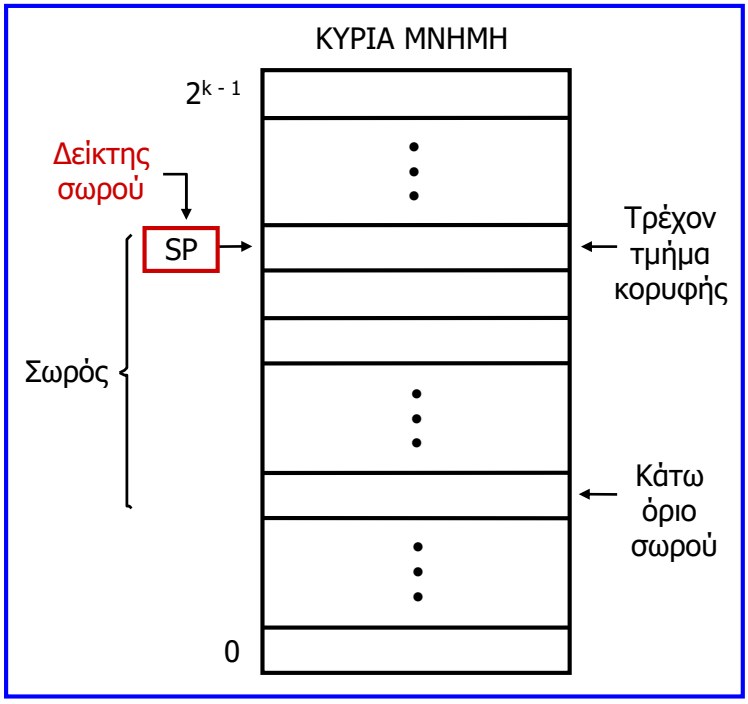
- Στις εντολές πράξεων των αρχιτεκτονικών καταχωρητή-καταχωρητή δηλώνονται 3 ή 2 καταχωρητές γενικού σκοπού, από τους οποίους οι δύο περιέχουν τα τελούμενα και ο τρίτος (ή ο δεύτερος) χρησιμοποιείται για την αποθήκευση του αποτελέσματος.
- Για την αντιγραφή των περιεχομένων της κύριας μνήμης σε καταχωρητές 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 τελούμενα 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

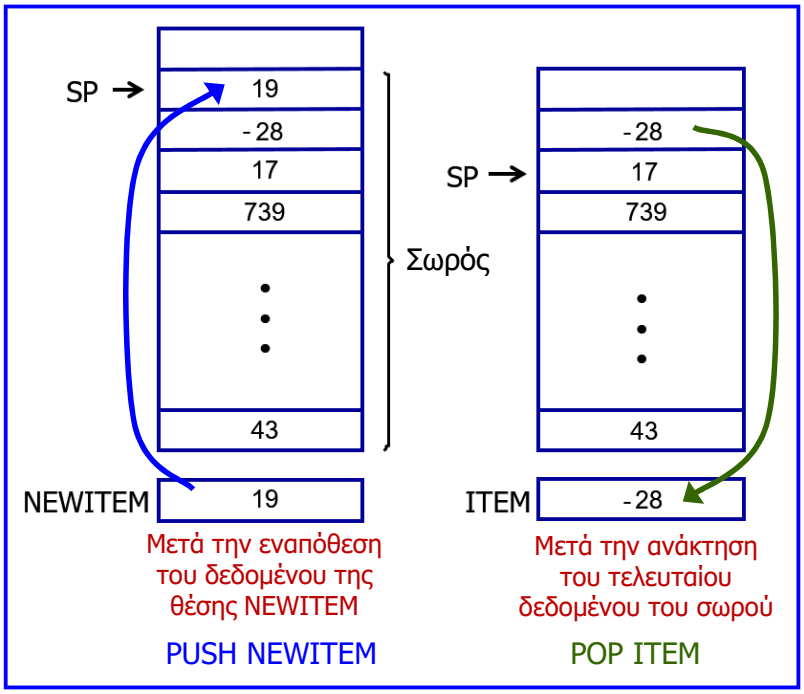
LOAD R1, A	Μεταφορά στον καταχωρητή R1 του περιεχομένου της θέσης μνήμης A
LOAD R2, B	Μεταφορά στον καταχωρητή R2 του περιεχομένου της θέσης μνήμης B
LOAD R3, C	Μεταφορά στον καταχωρητή R3 του περιεχομένου της θέσης μνήμης C
MUL R4, R2, R3 (ή MUL R2, R3)	Πολλαπλασιασμός του περιεχομένου των καταχωρητών 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

Αρχιτεκτονική μηχανισμού σωρού

- Ο σωρός (stack) αποτελείται από τμήματα δεδομένων που είναι αποθηκευμένα σε διαδοχικές θέσεις της κύριας μνήμης.
- Το τμήμα δεδομένων που τοποθετήθηκε τελευταίο στον σωρό αποτελεί την κορυφή του σωρού.
- Χρησιμοποιείται ένας καταχωρητής που περιέχει πάντα τη διεύθυνση της κορυφής του σωρού και αναφέρεται ως δείκτης σωρού (stack pointer).
- Για ανάκτηση (pop) δεδομένων από τη σωρό και για εναπόθεση (push) δεδομένων στον σωρό, οι επεξεργαστές διαθέτουν ειδικές εντολές.



Αρχιτεκτονική μηχανισμού σωρού



Αρχιτεκτονική μηχανισμού σωρού

Στις αρχιτεκτονικές μηχανισμού σωρού:

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

Αρχιτεκτονική μηχανισμού σωρού

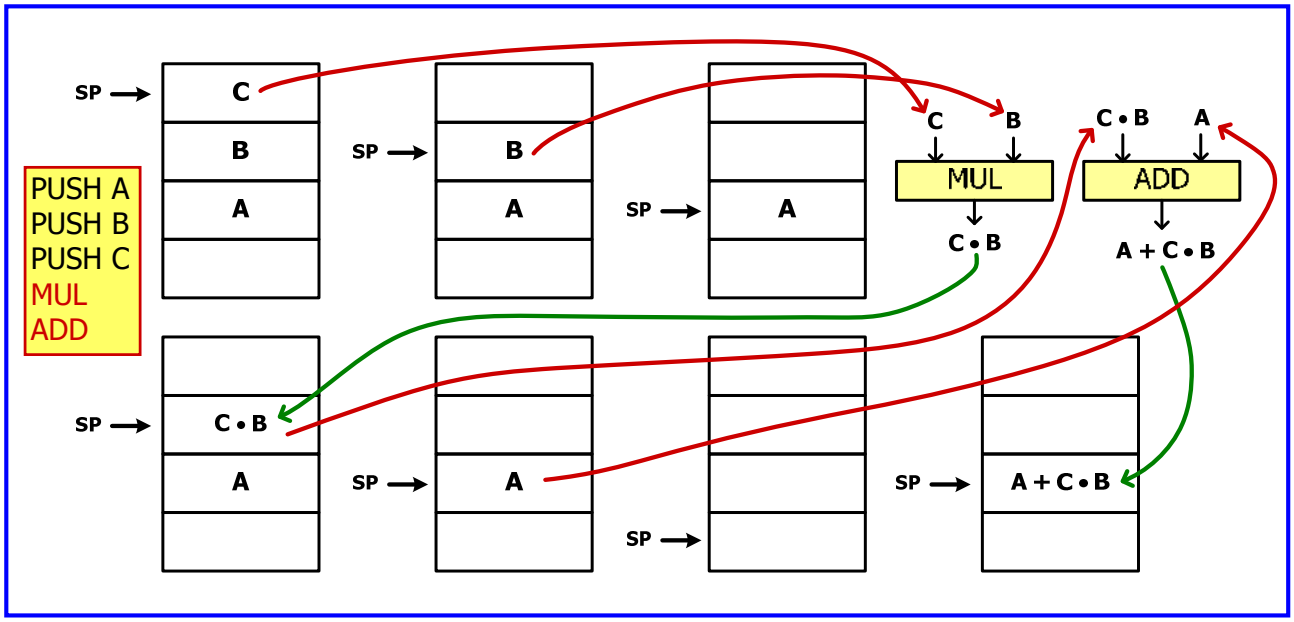
- Η δημιουργία συμβολικού κώδικα (Assembly) για αρχιτεκτονικές μηχανισμού σωρού, διευκολύνεται εάν γράψουμε την προς υπολογισμό παράσταση σε σημειογραφία επιθέματος (postfix notation).
- Σε αυτή δεν χρησιμοποιούνται παρενθέσεις, οι τελεστές ακολουθούν τα τελούμενα (operands) και εάν υπάρχουν πολλαπλοί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

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

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

Αρχιτεκτονική μηχανισμού σωρού

Στην αρχιτεκτονική σωρού, οι πράξεις δύο τελούμενων εκτελούνται μεταξύ των περιεχομένων των δύο κορυφαίων θέσεων του σωρού, σε τρία στάδια και το αποτέλεσμα αποθηκεύεται στην κορυφή του σωρού.



Υπολογιστές CISC και RISC

- CISC (complex instruction set computers): υπολογιστές πολύπλοκου συνόλου εντολών.
- RISC (reduced instruction set computers): υπολογιστές απλού συνόλου εντολών.
- Το σύνολο των εντολών σε επίπεδο γλώσσας μηχανής περιλαμβάνει:
  - ✓ στους υπολογιστές CISC: μεγάλο πλήθος κωδικών λειτουργίας, τρόπων διευθυνσιοδότησης και ειδών δεδομένων, πολύπλοκες και πανίσχυρες εντολές που εννοιολογικά βρίσκονται κοντά στις εντολές προγραμματισμού υψηλού επιπέδου
  - ✓ στους υπολογιστές RISC: απλές εντολές τύπου καταχωρητή-καταχωρητή, μικρό πλήθος απλών τρόπων διευθυνσιοδότησης, πολλοί καταχωρητές γενικού σκοπού.
- Χρόνος εκτέλεσης προγράμματος: (N x S) / R, N: πλήθος εντολών προγράμματος, S ή CPI: μέσος αριθμός βημάτων (κύκλων ρολογιού της ΚΜΕ) για την εκτέλεση μιας εντολής, R: συχνότητα λειτουργίας ΚΜΕ. Η απόδοση είναι αντιστρόφως ανάλογη του χρόνου εκτέλεσης.
- Οι απλές εντολές (RISC) απαιτούν μικρό αριθμό βημάτων για να εκτελεστούν, αλλά ένας υπολογιστής με απλό σύνολο εντολών χρειάζεται μεγάλο αριθμό εντολών για να εκτελέσει ένα πρόγραμμα (μεγάλο N, μικρό S ή CPI).
- Οι πολύπλοκες εντολές (CISC) εμπεριέχουν μεγαλύτερο αριθμό βημάτων, αλλά ένας υπολογιστής με εντολές που επιτελούν πιο σύνθετες πράξεις χρειάζεται λιγότερες εντολές για να εκτελέσει ένα πρόγραμμα (μικρό N, μεγάλο S ή CPI).

Υπολογιστές CISC και RISC

- Αρχιτεκτονική Pentium (CISC):

  - ✓ Πλήθος εντολών: 235
  - ✓ Μορφή εντολών: μεταβλητή (8 έως 88 bits)
  - ✓ Πλήθος τρόπων διευθυνσιοδότησης: 11
  - ✓ Πλήθος καταχωρητών γενικού σκοπού: 8

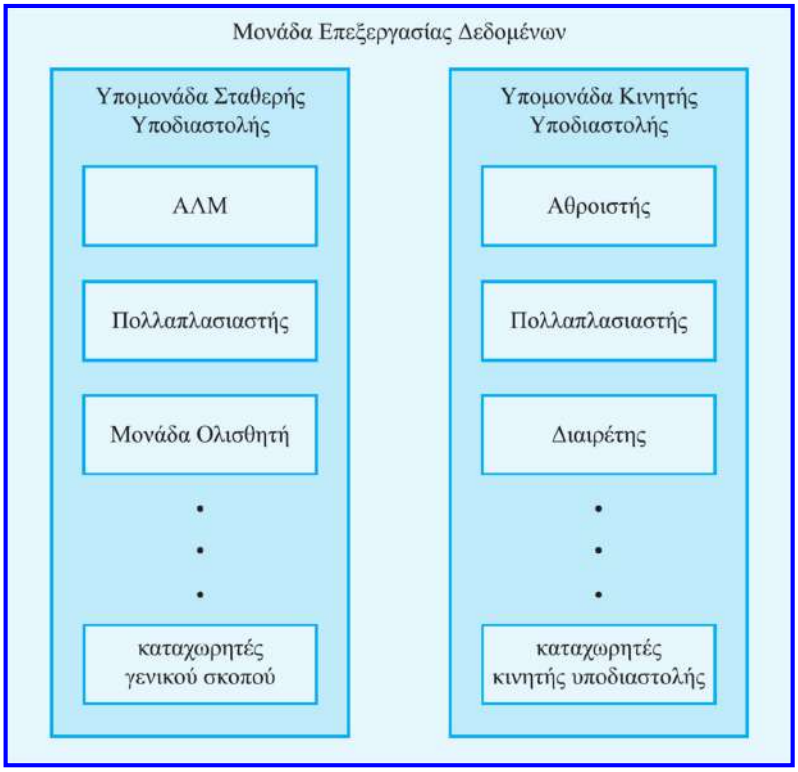
- Αρχιτεκτονική MIPS I (RISC):

  - ✓ Πλήθος εντολών: 70
  - ✓ Μορφή εντολών: σταθερή (32 bits)
  - ✓ Πλήθος τρόπων διευθυνσιοδότησης: 1
  - ✓ Πλήθος καταχωρητών γενικού σκοπού: 64

Κεντρική μονάδα επεξεργασίας (ΚΜΕ)

- Όπως προαναφέρθηκε, η ΚΜΕ αποτελείται από τη μονάδα επεξεργασίας δεδομένων (data path unit) και τη μονάδα ελέγχου (control unit).

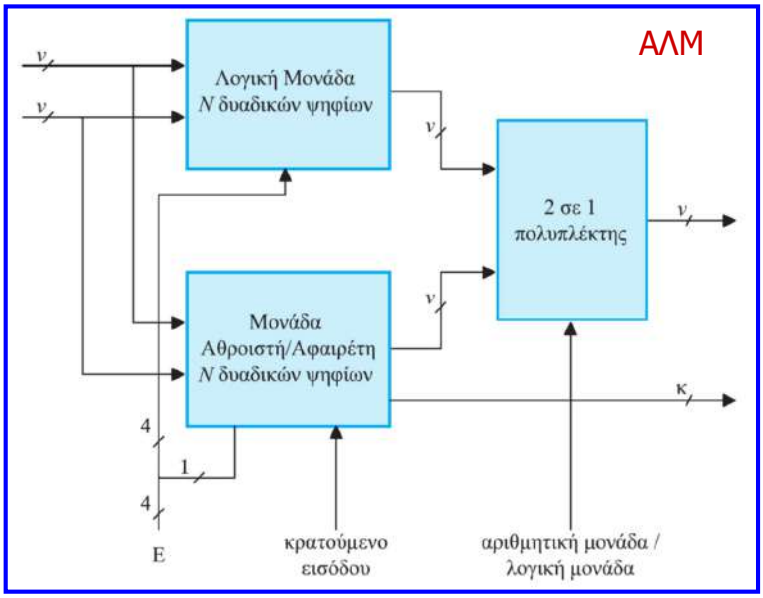
- Η μονάδα επεξεργασίας δεδομένων στους σύγχρονους υπολογιστές περιλαμβάνει υπομονάδες επεξεργασίας δεδομένων σταθερής και κινητής υποδιαστολής, στις οποίες συμμετέχουν καταχωρητές.



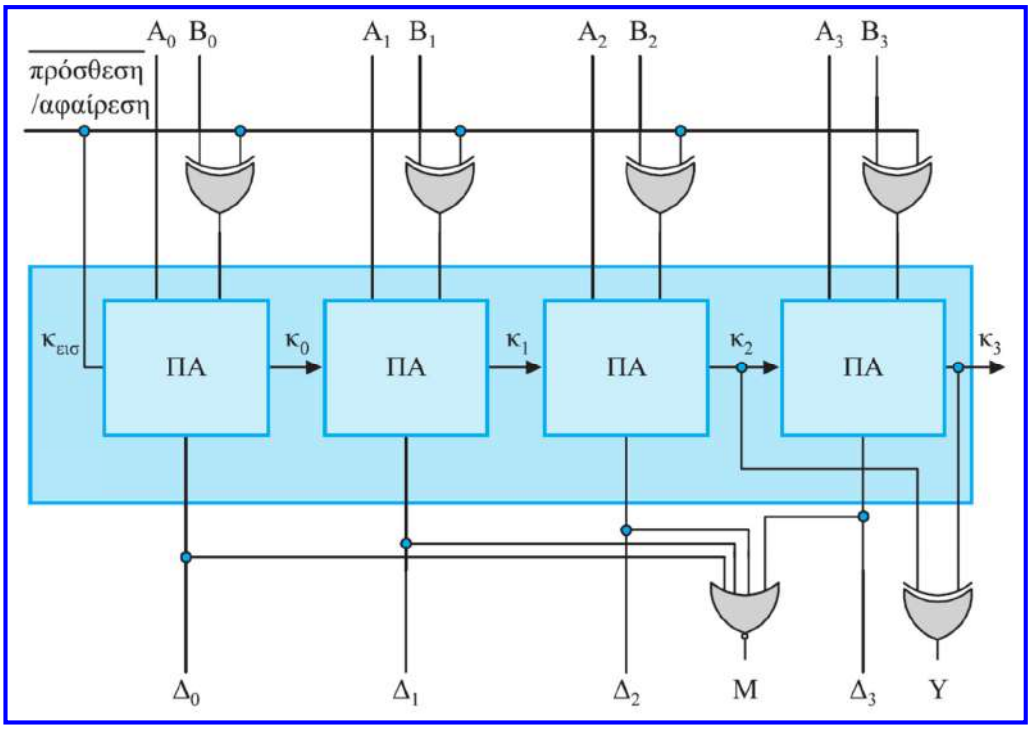
Μονάδα επεξεργασίας δεδομένων σταθερής υποδιαστολής

- Η μονάδα επεξεργασίας δεδομένων σταθερής υποδιαστολής περιλαμβάνει την αριθμητική λογική μονάδα (ΑΛΜ), μια μονάδα ολίσθησης, καταχωρητές γενικού σκοπού και συνήθως πολλαπλασιαστή και διαιρέτη.
- Η ΑΛΜ αποτελείται από μια υπομονάδα πρόσθεσης και αφαίρεσης και μια υπομονάδα εκτέλεσης λογικών πράξεων.

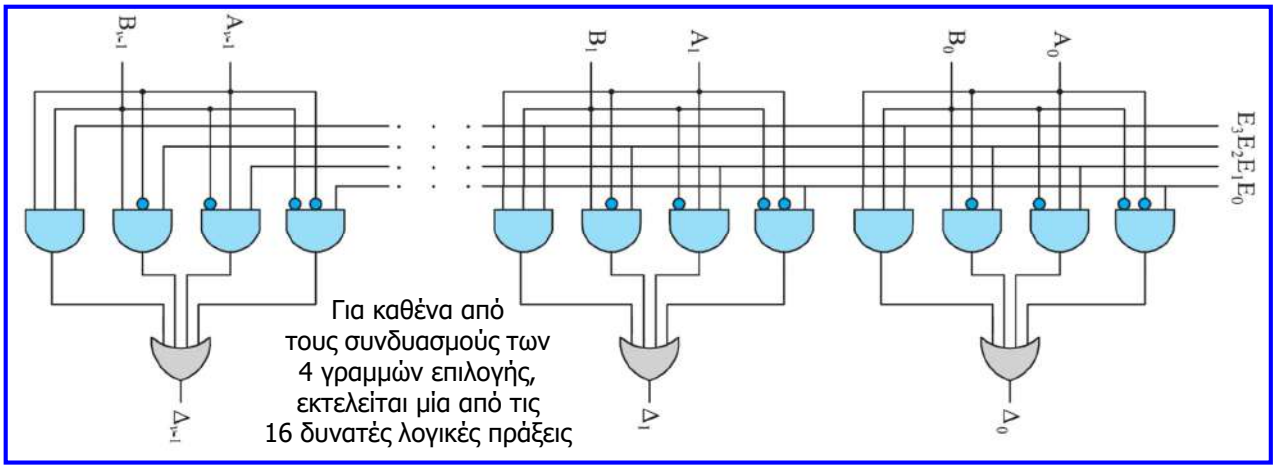
Οι αριθμοί που περιλαμβάνουν κλασματικό μέρος και η θέση της υποδιαστολής σε αυτού είναι προκαθορισμένη, αναφέρονται ως αριθμοί σταθερής υποδιαστολής (fixed point numbers).



Υπομονάδα πρόσθεσης και αφαίρεσης



Υπομονάδα εκτέλεσης λογικών πράξεων



Πίνακες αλήθειας συναρτήσεων δύο μεταβλητών

A	B	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

Παράδειγμα:

Για $E_3E_2E_1E_0 = 1110$ υλοποιείται η πράξη OR (F_7), αφού:

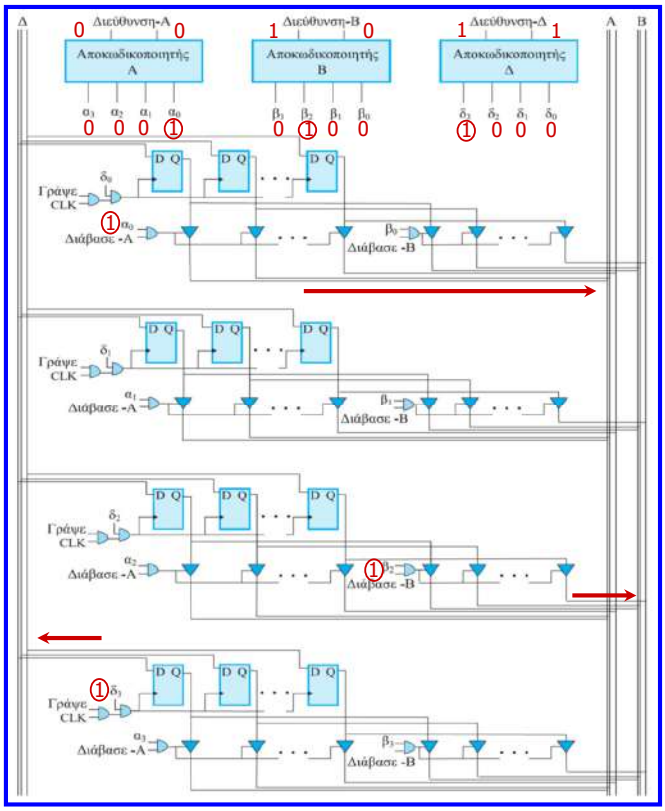
$\Delta = AB + A'B + AB' =$
$(A + A')B + AB' =$
$B + AB' = A + B$

Καταχωρητές γενικού σκοπού

Μονάδα 4 καταχωρητών
(4-register file)
με 2 πόρτες ανάγνωσης
και 1 πόρτα εγγραφής

Τα δεδομένα που διαβάζονται από τους καταχωρητές μεταφέρονται μέσω των αρτηριών Α και Β, ενώ αυτά που αποθηκεύονται στους καταχωρητές μεταφέρονται μέσω της αρτηρίας Δ.

Παράδειγμα: για την ανάγνωση του περιεχομένου του 1ου και του 3ου καταχωρητή και την εγγραφή περιεχομένου στον 4ο καταχωρητή, η μονάδα ελέγχου στέλνει στις εισόδους των αποκωδικοποιητών Α, Β και Δ τις διευθύνσεις 0, 2, 3, αντίστοιχα και θέτει τις γραμμές ελέγχου ανάγνωσης Α & Β και εγγραφής στη λογική τιμή 1.



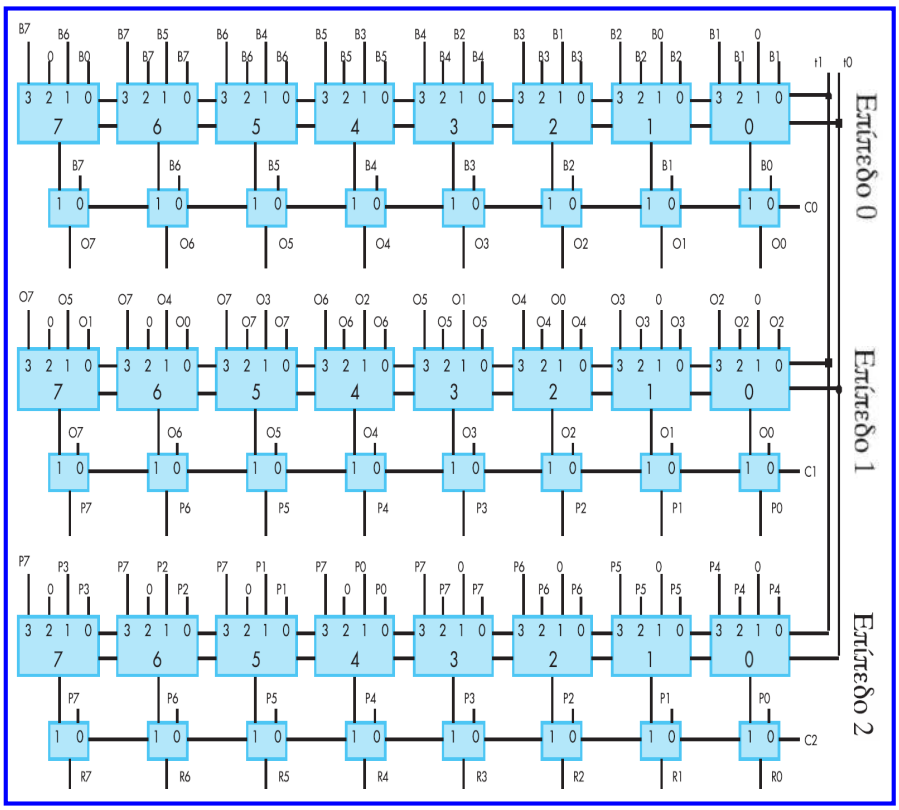
Μονάδα ολίσθησης

Συνδυαστικός ολισθητής 8 ψηφίων με 3 επίπεδα

Κάθε επίπεδο περιλαμβάνει μια σειρά 8 πολυπλεκτών 4 σε 1 και μια σειρά 8 πολυπλεκτών 2 σε 1.

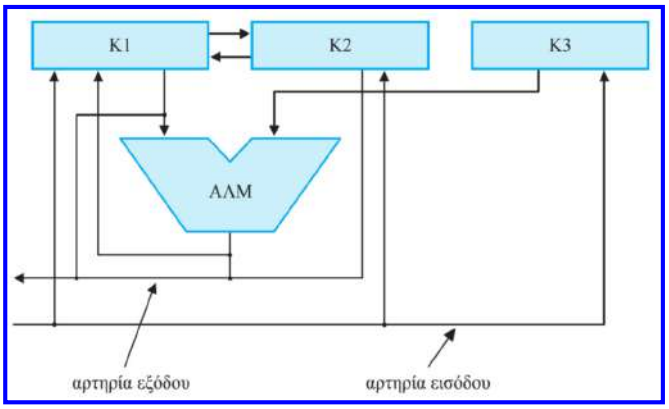
Ο δυαδικός αριθμός που σχηματίζουν οι είσοδοι C_2, C_1, C_0 αποτελεί το πλήθος των θέσεων ολίσθησης.

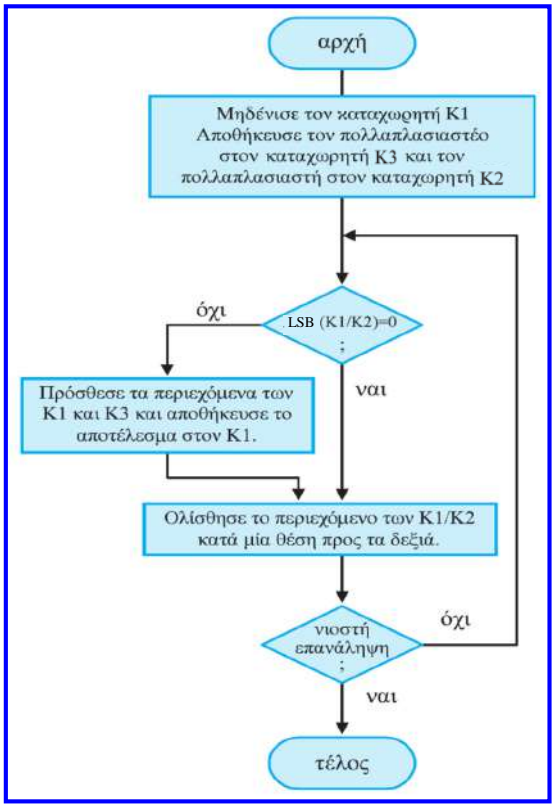
Στη γενική περίπτωση για δεδομένα $n = 2^κ$ ψηφίων, ο ολισθητής αποτελείται από κ επίπεδα.



t_1t_0	Πράξη
00	Κυκλική ολίσθηση προς τα δεξιά
01	Λογική ολίσθηση προς τα αριστερά
10	Λογική ολίσθηση προς τα δεξιά
11	Αριθμητική ολίσθηση προς τα δεξιά

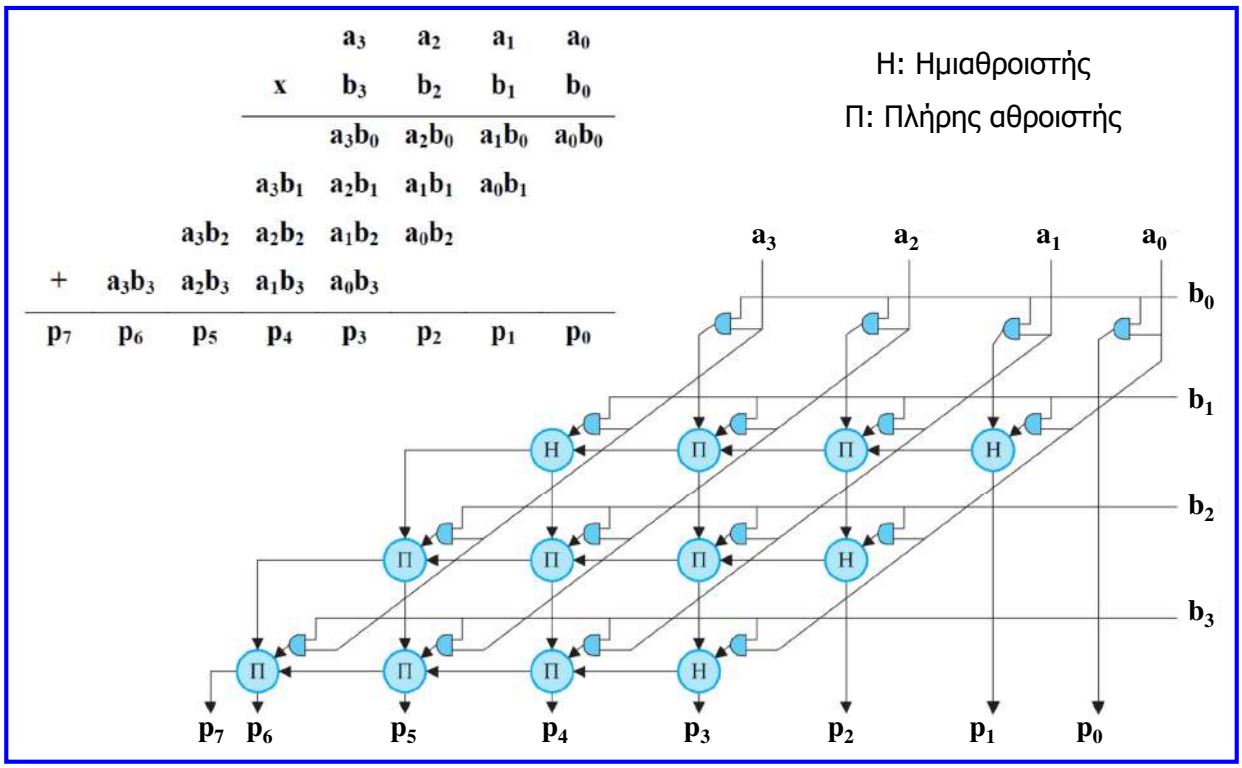
ΑΛΜ με δυνατότητα εκτέλεσης πολλαπλασιασμού





Παράδειγμα:										
	K3	1	1	0	0					$(12)_{10}$
×	K2	0	1	0	1					$(5)_{10}$
	K1/K2	0	0	0	0	0	1	0	**1**	
+	K3	1	1	0	0					
	K1/K2	1	1	0	0	0	1	0	1	
ΟΛΙΣΘΗΣΗ ΔΕΞΙΑ	K1/K2	0	1	1	0	0	0	1	**0**	
ΟΛΙΣΘΗΣΗ ΔΕΞΙΑ	K1/K2	0	0	1	1	0	0	0	**1**	
+	K3	1	1	0	0					
	K1/K2	1	1	1	1	0	0	0	1	
ΟΛΙΣΘΗΣΗ ΔΕΞΙΑ	K1/K2	0	1	1	1	1	0	0	**0**	
ΟΛΙΣΘΗΣΗ ΔΕΞΙΑ	K1/K2	0	0	1	1	1	1	0	0	$(60)_{10}$

Συνδυαστικό κύκλωμα πολλαπλασιασμού



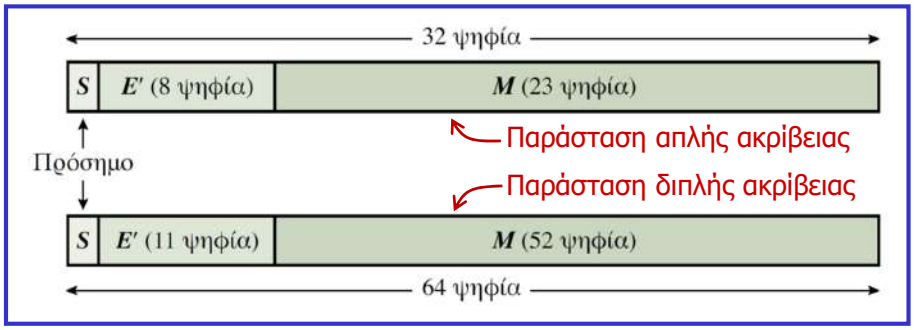
Αριθμοί κινητής υποδιαστολής

- Όπως προαναφέρθηκε, οι αριθμοί που περιλαμβάνουν κλασματικό μέρος και η θέση της υποδιαστολής σε αυτού είναι προκαθορισμένη, 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, 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) και περιλαμβάνονται σε αυτά μονάδες για την εκτέλεση πράξεων μεταξύ αριθμών κινητής υποδιαστολής.

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.

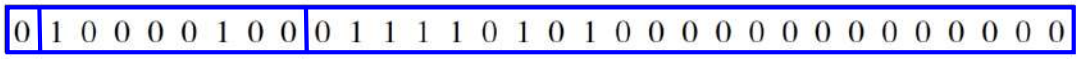
Αριθμοί κινητής υποδιαστολής



- Η κωδικοποίηση που χρησιμοποιείται στον εκθέτη, αναφέρεται ως παράσταση πόλωσης, 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 (όταν E′ = M = 0), ∞ (για E′ = 255, M = 0) και NaN ή «Not A Number» (όταν E′ = 255, M # 0).
- Το ψηφίο που βρίσκεται αριστερά της υποδιαστολής είναι πάντοτε 1 και δεν αναπαρίσταται, αλλά υπονοείται 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.

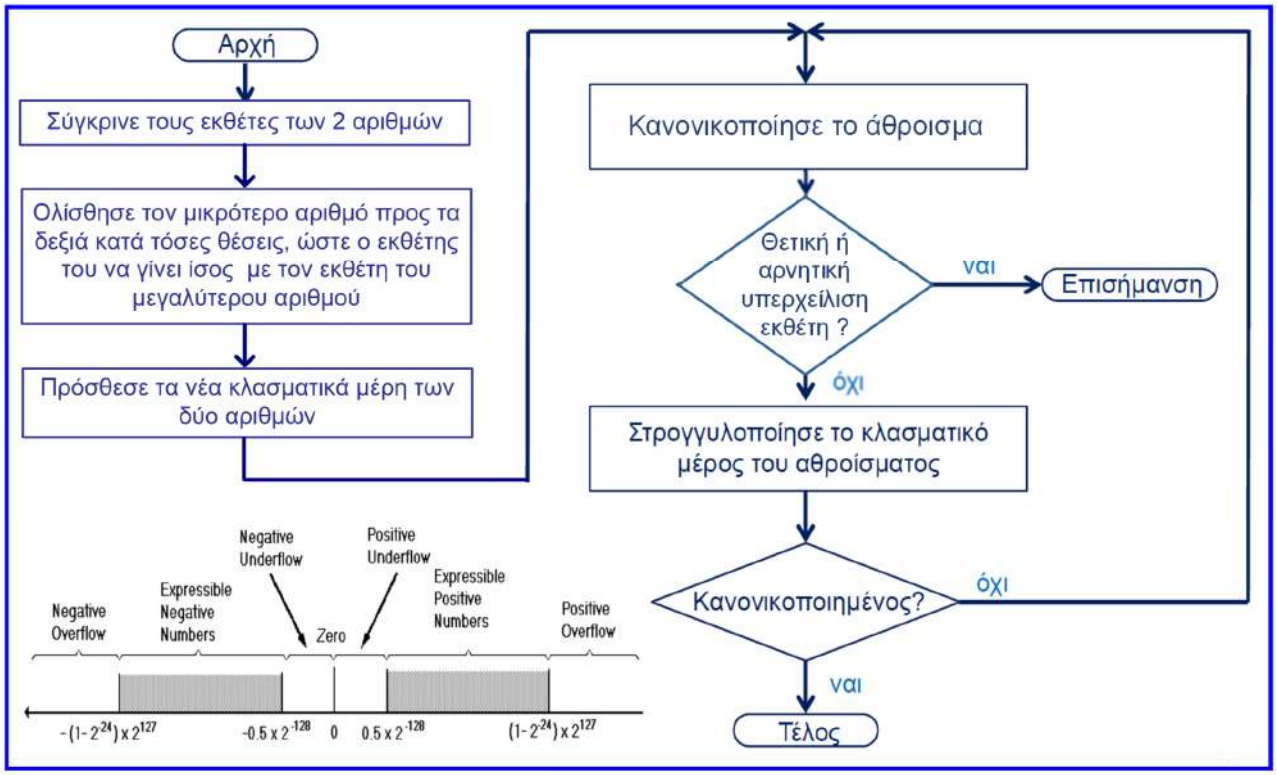
Αριθμοί κινητής υποδιαστολής

- 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.



- 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.

Πρόσθεση αριθμών κινητής υποδιαστολής



Πρόσθεση & αφαίρεση αριθμών κινητής υποδιαστολής

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

- Δυαδική παράσταση: X = 110101.101 = 110101.101 × 2^0, Y = – 11010111.01 × 2^0.
- Κανονικοποίηση αριθμών, δηλ. μετατόπιση υποδιαστολής προς τα αριστερά τόσες θέσεις όσες απαιτούνται σε κάθε αριθμό, έτσι ώστε να είναι μονάδα το πιο σημαντικό ψηφίο τους, με αντίστοιχη αύξηση των εκθετών των αριθμών: X = 1.10101101 × 2^5 και Y = – 1.101011101 × 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: X μετά από ολίσθηση κατά 2 θέσεις δεξιά = 0.0110101101.
- Για τον υπολογισμό του αθροίσματος X + Y, υπολογίζουμε το συμπλήρωμα ως προς 2 του αρνητικού 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

Πρόσθεση & αφαίρεση αριθμών κινητής υποδιαστολής

- Επειδή Y > 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	0

- Για τον υπολογισμό της διαφοράς X – Y αρκεί να αθροίσουμε κλασματικό μέρος του μετατοπισμένου αριθμού X με τον αντίθετο του Y (1.101011101):

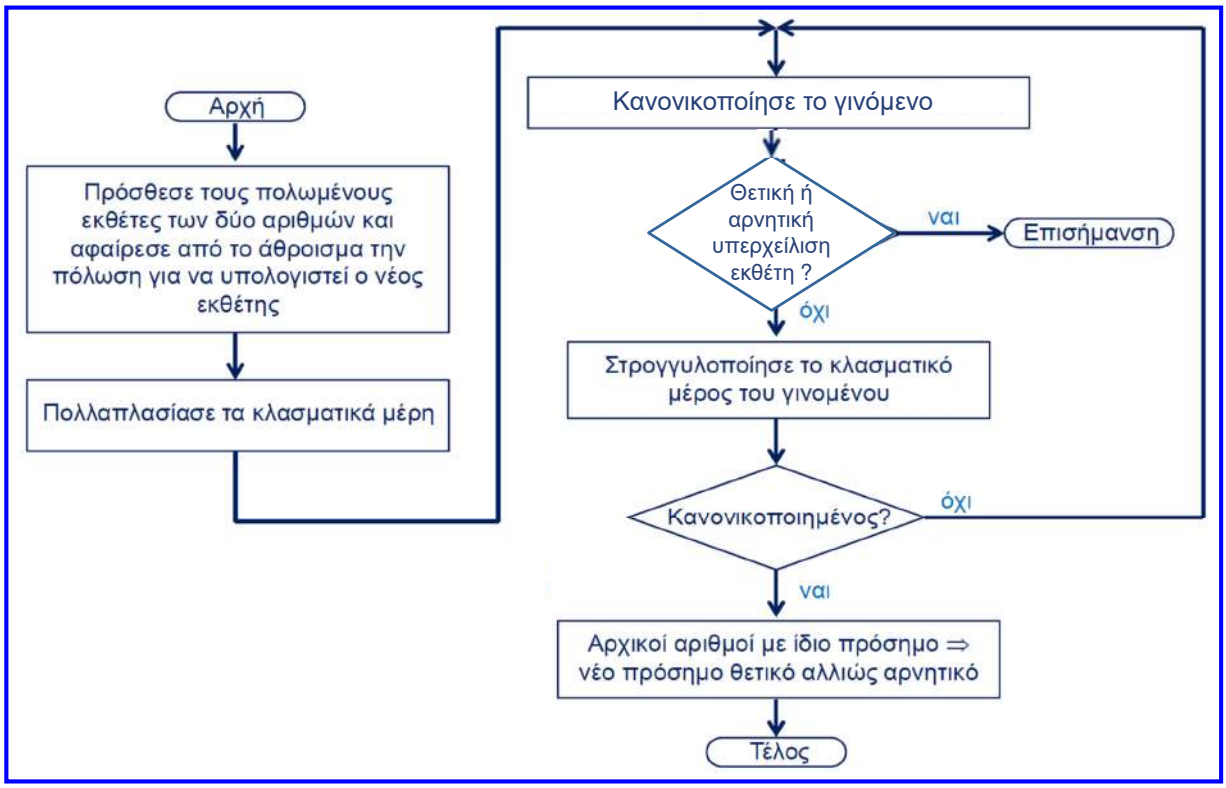
	0.	0	1	1	0	1	0	1	1	0	1
+	1.	1	0	1	0	1	1	1	0	1	0
1	0.	0	0	0	1	1	0	0	1	1	1

- Τέλος, κανονικοποιούμε το αποτέλεσμα, μετατοπίζοντας την υποδιαστολή μία θέση προς τα αριστερά με παράλληλη αύξηση του εκθέτη κατά μία μονάδα:

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

- Στην περίπτωση που προκύπτει υπέρβαση του διαθέσιμου πλήθους ψηφίων, απαιτείται στρογγυλοποίηση του κλασματικού μέρους του αποτελέσματος, η οποία μπορεί να διενεργηθεί με τη μέθοδο που αναφέρθηκε στη σελίδα 19 (δηλαδή πρόσθεση στον αριθμό του 2^{-23} / 2 = 2^{-24} σε δυαδική μορφή και περικοπή των πλεοναζόντων ψηφίων).

Πολλαπλασιασμός αριθμών κινητής υποδιαστολής



Πολλαπλασιασμός αριθμών κινητής υποδιαστολής

Παράδειγμα: Εκτέλεση πράξης A x B με A = 106.625 και B = – 0.203125

- Δυαδική παράσταση και κανονικοποίηση: $A = 1100110.101 \times 2^0 = 1.100110101 \times 2^6$, $B = -0.001101 = -0.001101 \times 2^0 = -1.101 \times 2^{-3}$.
- Υπολογισμός εκθετών: $E_A = 6 + 127 = 133$ ή 10000101, $E_B = -3 + 127 = 124$ ή 01111100.

A:	0	1	0	0	0	0	1	0	1	1	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B:	1	0	1	1	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Πρόσθεση των εκθετών και αφαίρεση του 127 από το αποτέλεσμα: 133 + 124 – 127 = 130 ή 10000010, ώστε να προκύψει ο εκθέτης του γινομένου.
- Πολλαπλασιασμός των κλασματικών μερών:

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

Πολ/σμός & διαίρεση αριθμών κινητής υποδιαστολής

- Το αποτέλεσμα δεν είναι κανονικοποιημένο, οπότε θα πρέπει να κανονικοποιηθεί μετατοπίζοντας την υποδιαστολή μία θέση αριστερά (1.0100110110001) και αυξάνοντας τον εκθέτη κατά 1, ώστε να προκύψει ο τελικός πολωμένος εκθέτης του γινομένου (131).

A x B: | 1 | 1 0 0 0 0 0 1 1 | 0 1 0 0 1 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 |

- Αφού οι Α και Β είναι ετερόσημοι, το γινόμενο λαμβάνει αρνητικό πρόσημο (1).

- Κατά την εκτέλεση της διαίρεσης μεταξύ αριθμών κινητής υποδιαστολής, διενεργούμε αρχικά αφαίρεση των εκθετών και πρόσθεση του 127 (πόλωσης) στο αποτέλεσμα.

- Στη συνέχεια διαιρούμε τα κλασματικά μέρη, καθορίζουμε το πρόσημο του αποτελέσματος και εάν απαιτείται διενεργούμε στρογγυλοποίηση και κανονικοποίηση του πηλίκου.

- Επισημαίνεται ότι κατά την εκτέλεση όλων των πράξεων μεταξύ των κλασματικών μερών, σε αυτές συμμετέχει και το ψηφίο που βρίσκεται αριστερά της υποδιαστολής.

Μονάδα ελέγχου

- Όπως προαναφέρθηκε στη σελίδα 9, η ΚΜΕ εκτελεί μια εντολή ακολουθώντας στη γενική περίπτωση μια σειρά από 7 βήματα, τα οποία αποτελούν τον κύκλο εντολής.

- Ο κύκλος εντολής περιλαμβάνει τη φάση προσκόμισης της εντολής και τη φάση εκτέλεσης της εντολής.

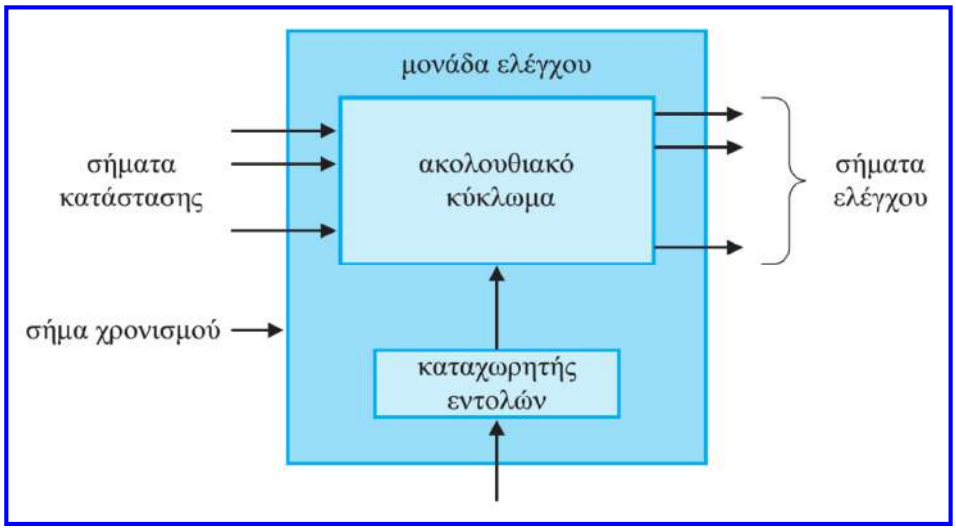
- Η μονάδα ελέγχου παράγει σήματα ελέγχου που στην πρώτη φάση έχουν σαν συνέπεια την προσκόμιση της εντολής, ενώ στη δεύτερη φάση η μονάδα ελέγχου αποκωδικοποιεί (ερμηνεύει) την εντολή και ανάλογα με το περιεχόμενο αυτής, παράγει σήματα ώστε να κατευθυνθούν τα δεδομένα στις κατάλληλες λειτουργικές μονάδες και να επιλεγούν οι λειτουργίες που θα εκτελεστούν με κατάλληλη χρονική σειρά.

- Η επιλογή της σειράς εκτέλεσης των εντολών βασίζεται στη χρήση του μετρητή προγράμματος (program counter, PC), ο οποίος περιέχει τη διεύθυνση της θέσης μνήμης όπου είναι αποθηκευμένη η επόμενη προς εκτέλεση εντολή.

- Η σειριακή εκτέλεση των εντολών μπορεί να μεταβληθεί, όταν η εκτελούμενη εντολή είναι εντολή άλματος (jump), διακλάδωσης (branch), κλήσης υποπρογράμματος ή υπορουτίνας (subroutine call) ή έχει συμβεί ειδική περίπτωση (exception) ή έχει ληφθεί σήμα διακοπής (interrupt) και θα πρέπει να μεταφερθεί ο έλεγχος του προγράμματος στην ρουτίνα εξυπηρέτησης του σήματος διακοπής.

Υλοποίηση μονάδας ελέγχου με ακολουθιακό κύκλωμα

- Μία μέθοδος υλοποίησης της μονάδας ελέγχου, βασίζεται στη σχεδίαση κατάλληλου ακολουθιακού κυκλώματος για την παραγωγή των απαιτούμενων σημάτων ελέγχου.
- Πλεονέκτημα: λειτουργεί με υψηλές ταχύτητες.
- Μειονεκτήματα: απαιτεί κυκλώματα σύνθετης λογικής που αυξάνουν το κόστος, περιορισμένη ευελιξία (δυσκολία προσθήκης νέων εντολών).

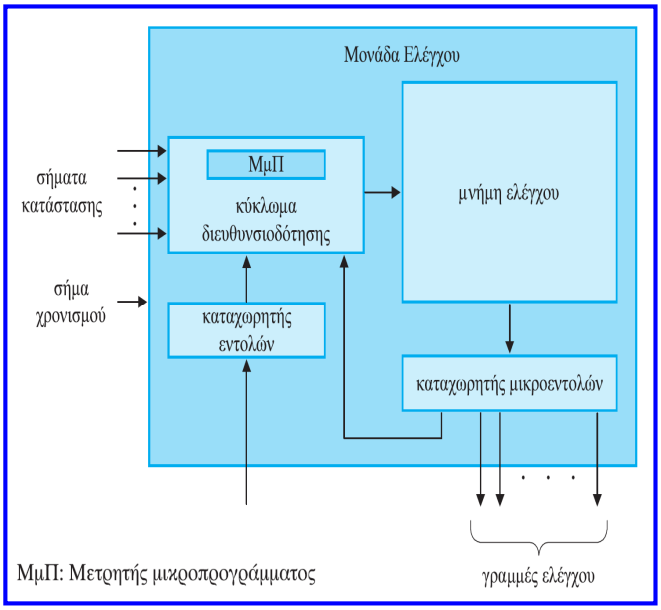


Μικροπρογραμματισμός

- Μικροπρογραμματισμός είναι μια μέθοδος σχεδιασμού μονάδας ελέγχου που οδηγεί σε μικροπρογραμματιζόμενη μονάδα ελέγχου (microprogrammed control unit).
- Η πληροφορία που αφορά την επιλογή των σημάτων ελέγχου και ένα μέρος της πληροφορίας που αφορά τη σειρά παραγωγής τους, αποθηκεύεται στη μνήμη ελέγχου (control memory).
- Τα σήματα ελέγχου καθορίζονται από τα ψηφία μιας ψηφιολέξης που αναφέρεται ως μικροεντολή (microinstruction) και προσκομίζεται από τη μνήμη ελέγχου.
- Οι μικρολειτουργίες που ενεργοποιούνται από μία μικροεντολή εκτελούνται σε ένα κύκλο ρολογιού.
- Η ακολουθία μικροεντολών που πρέπει να προσκομιστούν για να εκτελεστεί μια εντολή, αποτελεί το μικροπρόγραμμα (microprogram) της εντολής.
- Κάθε εντολή εκτελείται προσκομίζοντας από τη μνήμη ελέγχου τις μικροεντολές που αποτελούν το μικροπρόγραμμά της.
- Πλεονεκτήματα: συστηματική μέθοδος αφού οργανώνει τα σήματα ελέγχου σε καθορισμένες μικροεντολές και ευέλικτη αφού υλοποιείται με λογισμικό και είναι εύκολες οι διορθώσεις ή τροποποιήσεις.
- Μειονεκτήματα: αργή μέθοδος λόγω προσκόμισης μικροεντολών από τη μνήμη ελέγχου, απαίτηση για εξατομικευμένους μεταγλωττιστές.

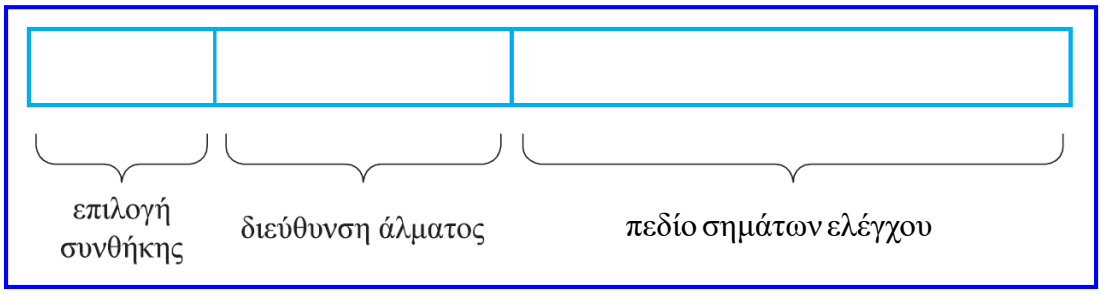
Μικροπρογραμματιζόμενη μονάδα ελέγχου

- Το κύκλωμα διευθυνσιοδότησης από τον κωδικό λειτουργίας της εντολής που βρίσκεται στον καταχωρητή εντολών, προσδιορίζει τη διεύθυνση θέσης μνήμης ελέγχου που περιέχει την 1η μικροεντολή του μικροπρογράμματος της τρέχουσας εντολής και την αποθηκεύει στον ΜμΠ.
- Η μικροεντολή προσκομίζεται στον καταχωρητή μικροεντολών και τα ψηφία αυτής που έχουν τιμή 1 ενεργοποιούν τα αντίστοιχα σήματα ελέγχου.
- Ταυτόχρονα, το περιεχόμενο του ΜμΠ αυξάνεται κατά 1, ώστε να περιέχει τη διεύθυνση της επόμενης μικροεντολής του μικροπρογράμματος, εκτός κι αν πρόκειται για μικροεντολή άλματος.



- Η τελευταία μικροεντολή κάθε μικροπρογράμματος είναι μικροεντολή άλματος χωρίς συνθήκη, που περιέχει τη διεύθυνση του μικροπρογράμματος που υλοποιεί την προσκόμιση νέας εντολής, το οποίο είναι κοινό για όλες τις εντολές.

Γενική μορφή μικροεντολής

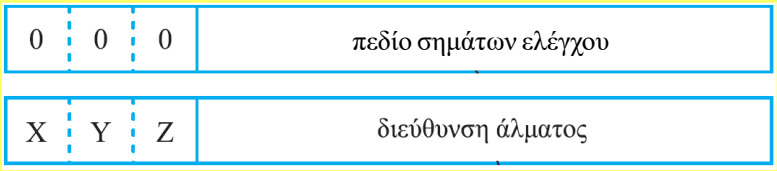


- Το πεδίο επιλογής συνθήκης, καθορίζει τη συνθήκη που θα πρέπει να ελεγχθεί (π.χ. ύπαρξη μηδενικού αποτελέσματος, δηλαδή 1 στο αντίστοιχο ψηφίο του καταχωρητή κατάστασης), στις μικροεντολές άλματος υπό συνθήκη.
- Το πεδίο διεύθυνσης άλματος περιέχει τη διεύθυνση της θέσης στη μνήμη ελέγχου που περιέχει την μικροεντολή που πρέπει να προσκομιστεί στον καταχωρητή μικροεντολών, εάν ικανοποιείται η συνθήκη άλματος.
- Στο πεδίο σημάτων ελέγχου έχουν τιμή 1 μόνο τα δυαδικά ψηφία που αντιστοιχούν στα σήματα που πρέπει να ενεργοποιηθούν όταν η συγκεκριμένη μικροεντολή προσκομιστεί στον καταχωρητή μικροεντολών.

Μείωση χωρητικότητας μνήμης ελέγχου

- Βασικές τεχνικές που χρησιμοποιούνται για τη μείωση της απαιτούμενης χωρητικότητας της μνήμης ελέγχου είναι:
  - ✓ η χρησιμοποίηση περισσότερων από μία μορφών μικροεντολής,
  - ✓ και η κωδικοποίηση του πεδίου σημάτων ελέγχου.
- Σχετικά με την πρώτη τεχνική, αφού στις μικροεντολές άλματος δεν χρησιμοποιείται το πεδίο σημάτων ελέγχου, μπορεί η μορφή τους να περιλαμβάνει μόνο τα πεδία επιλογής συνθήκης και διεύθυνσης άλματος.
- Παρομοίως, αφού στις υπόλοιπες μικροεντολές σε καμία περίπτωση δεν χρησιμοποιείται το πεδίο διεύθυνσης άλματος, αυτό μπορεί να παραληφθεί.

Παράδειγμα: εάν έχουμε 5 μικροεντολές άλματος υπό συνθήκη, 1 εντολή άλματος χωρίς συνθήκη και 80 σήματα ελέγχου σε κάθε άλλη μικροεντολή, τότε όταν το πρώτο πεδίο περιέχει 000, το δεύτερο πεδίο καθορίζει τα σήματα ελέγχου, ενώ 6 άλλοι συνδυασμοί του πρώτου πεδίου αντιστοιχούν στις 6 μικροεντολές άλματος.

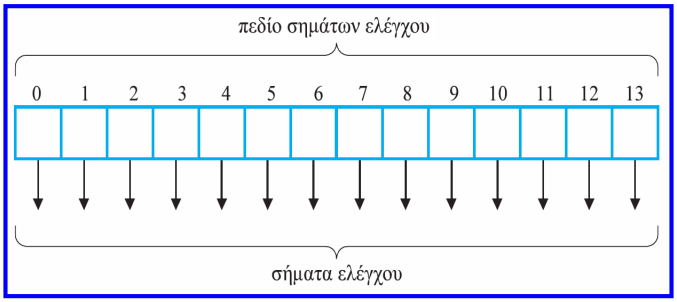


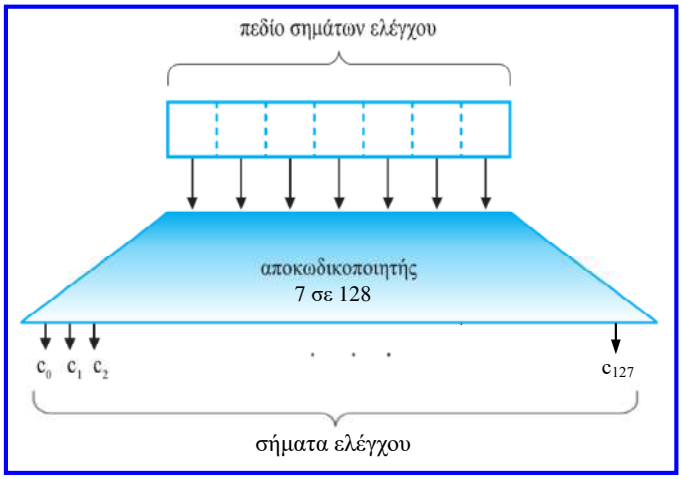
Μορφή μικροεντολής σημάτων ελέγχου

Μορφή μικροεντολής άλματος

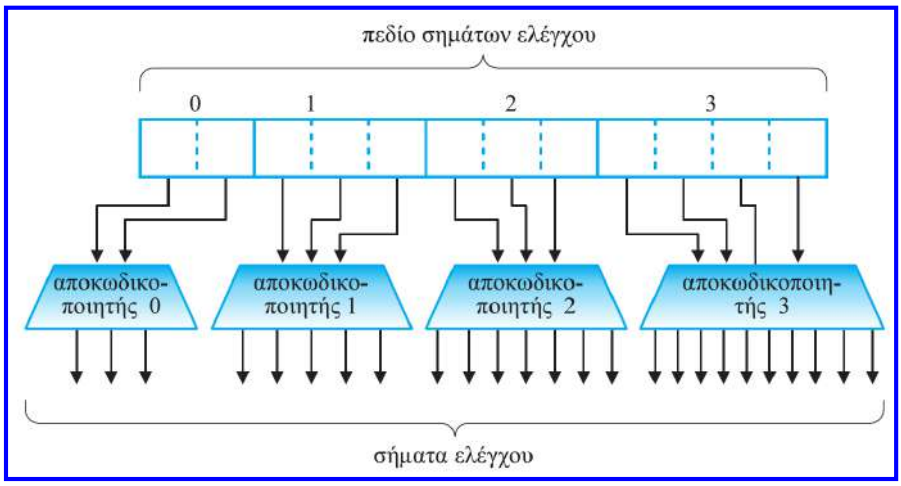
Κωδικοποίηση πεδίου σημάτων ελέγχου

- Όταν σε κάθε σήμα ελέγχου αντιστοιχεί ένα ψηφίο μικροεντολής, τότε κάθε μικροεντολή μπορεί να ενεργοποιήσει οποιοδήποτε πλήθος σημάτων ελέγχου.
- Αρκεί βέβαια οι μικρολειτουργίες που προκαλούνται από τα σήματα ελέγχου να μπορούν να εκτελεστούν παράλληλα, γεγονός που εξαρτάται από τη δομή του επεξεργαστή.
- Εάν αυτός έχει σχεδιαστεί έτσι ώστε σε κάθε χρονική περίοδο να ενεργοποιείται μόνο ένα σήμα ελέγχου (δηλαδή, να εκτελείται μία μικρολειτουργία), τότε είναι δυνατή η μείωση του μήκους μικροεντολών με κατάλληλη κωδικοποίηση του πεδίου των σημάτων ελέγχου και χρήση ενός αποκωδικοποιητή για την παραγωγή των σημάτων ελέγχου.





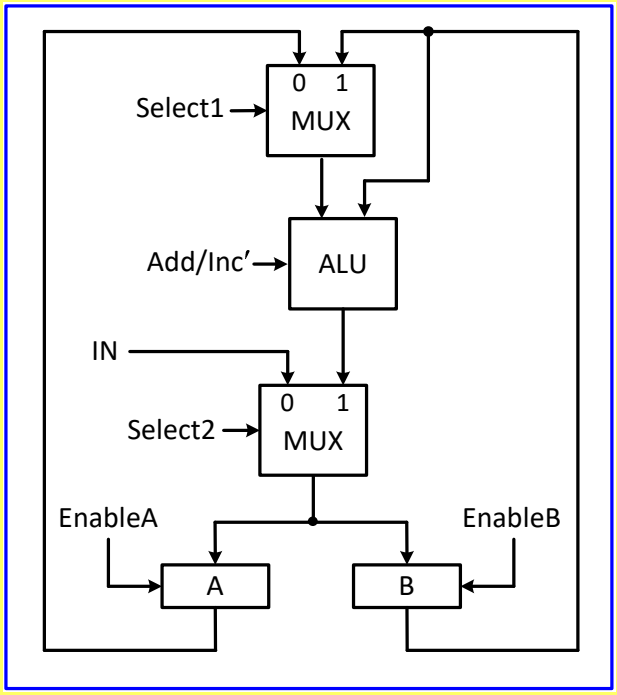
Κωδικοποίηση πεδίου σημάτων ελέγχου



- Συνήθως στις υλοποιήσεις μικροεπεξεργαστών, υπάρχουν μικρολειτουργίες που δεν μπορούν να εκτελεστούν παράλληλα.
- Επομένως, είναι δυνατός ο διαχωρισμός των σημάτων ελέγχου σε ομάδες, έτσι ώστε σε κάθε ομάδα να ανήκουν σήματα που δεν είναι ενεργά την ίδια χρονική περίοδο.
- Στην περίπτωση αυτή για κάθε ομάδα σημάτων, χρησιμοποιείται κι ένας αποκωδικοποιητής.

Παράδειγμα μικροπρογράμματος

- Ο υποτιθέμενος επεξεργαστής περιλαμβάνει δομή με 2 καταχωρητές A και B, 2 πολυπλέκτες (MUX) και ALU.
- Η συγκεκριμένη ALU είναι συνδυαστικό κύκλωμα που δεν διαθέτει εσωτερικές μονάδες αποθήκευσης και έχει τη δυνατότητα εκτέλεσης δύο πράξεων: πρόσθεση των τελούμενων που εφαρμόζονται στις 2 εισόδους της ή αύξηση κατά 1 του τελούμενου που εφαρμόζεται στην δεξιά είσοδό της, ανάλογα με το αν το σήμα ελέγχου Add/Inc' λαμβάνει τιμή 1 ή 0, αντίστοιχα
- Όταν τα σήματα επίτρεψης εγγραφής EnableA και EnableB λαμβάνουν τιμή 1, τότε εγγράφεται στον αντίστοιχο καταχωρητή η είσοδός του.

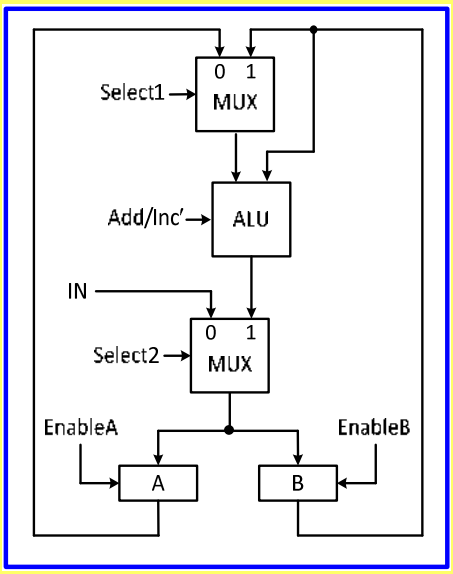


Παράδειγμα μικροπρογράμματος

Μικροπρόγραμμα της εντολής A ← IN + B + 1, η οποία προσθέτει τα περιεχόμενα της εισόδου IN και του B, αυξάνει το άθροισμα κατά 1 και αποθηκεύει το αποτέλεσμα στον A:

Μικροεντολές	Περιγραφή
A ← IN	Μεταφορά της εισόδου IN στον καταχωρητή A
B ← A + B	Πρόσθεση των περιεχομένων των καταχωρητών A και B και αποθήκευση του αθροίσματος στον B
A ← B + 1	Αύξηση κατά 1 του περιεχομένου του καταχωρητή B και αποθήκευση του αποτελέσματος στον A

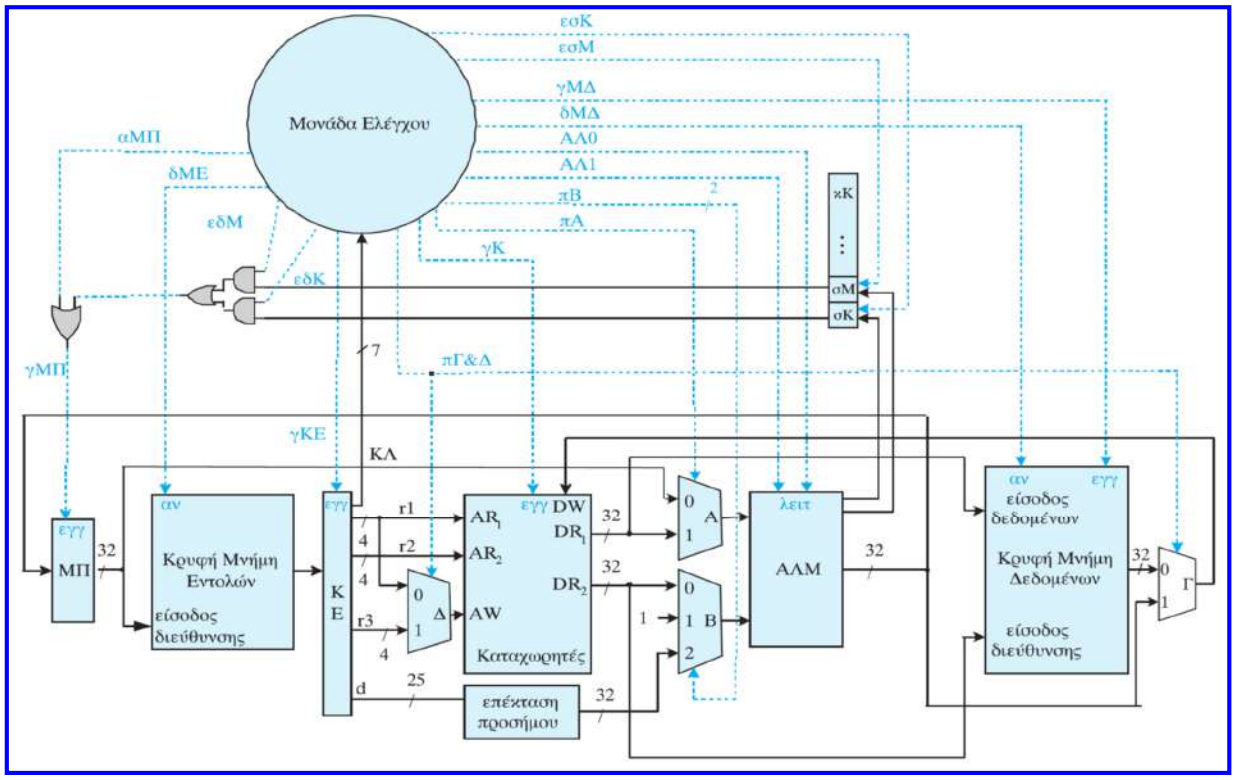
Μικροεντολές	Τιμές σημάτων ελέγχου του επεξεργαστή				
	Select1	Select2	Add/Inc'	EnableA	EnableB
A ← IN	X	0	X	1	0
B ← A + B	0	1	1	0	1
A ← B + 1	X	1	0	1	0



ΚΜΕ δεδομένων σταθερής υποδιαστολής

- Στο Παράρτημα Β του τόμου Αρχιτεκτονική Υπολογιστών Ι, περιγράφεται αναλυτικά η λειτουργία μιας κεντρικής μονάδας επεξεργασίας (ΚΜΕ) δεδομένων σταθερής υποδιαστολής, η οποία βασίζεται στη φιλοσοφία των επεξεργαστών RISC (απλές εντολές σταθερού μεγέθους, απλοί τρόποι διευθυνσιοδότησης και εντολές εκτέλεσης πράξεων μόνο μεταξύ δεδομένων που είναι αποθηκευμένα σε καταχωρητές).
- Από την εν λόγω αναλυτική περιγραφή μπορείτε να κατανοήσετε τη δομή της ΚΜΕ δεδομένων σταθερής υποδιαστολής, το σχεδιασμό και τα αναλυτικά βήματα εκτέλεσης εντολών, όπως: εντολές αριθμητικών πράξεων (ADD/SUB), εντολές ανάγνωσης και φόρτωσης δεδομένων (LOAD/STORE), καθώς και εντολές διακλάδωσης (BRZ/BRC).
- Επίσης, μπορείτε να κατανοήσετε τις ενέργειες που λαμβάνουν χώρα στην μονάδα επεξεργασίας των δεδομένων, με βάση τα σήματα ελέγχου που παράγει η μονάδα ελέγχου σε κάθε κύκλο εντολής.
- Τα παραπάνω θα σας βοηθήσουν να εμπεδώσετε και να επεκτείνετε τις γνώσεις που αφορούν όλα όσα προαναφέρθηκαν.

ΚΜΕ δεδομένων σταθερής υποδιαστολής

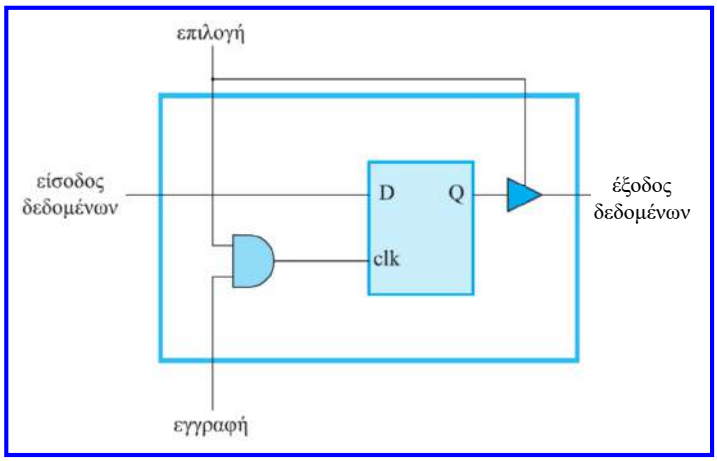


Σύστημα μνήμης

- Κύριος σκοπός στο σχεδιασμό ενός συστήματος μνήμης είναι η παροχή επαρκούς χωρητικότητας αποθήκευσης, διατηρώντας ένα αποδεκτό επίπεδο απόδοσης και χαμηλό μέσο κόστος ανά δυαδικό ψηφίο.
- Ο σκοπός αυτός επιτυγχάνεται κυρίως με τεχνικές, όπως:
  - ✓ χρησιμοποίηση ενός αριθμού από διαφορετικές μονάδες μνήμης με διαφορετικό λόγο κόστους / απόδοσης που οργανώνονται κατά τέτοιο τρόπο (ιεραρχία μνήμης) ώστε να παρέχουν υψηλή μέση απόδοση με χαμηλό μέσο κόστος,
  - ✓ ανάπτυξη ιδεατής-μνήμης (virtual-memory) για να απελευθερωθεί ο χρήστης από την διαχείριση της μνήμης και να γίνουν τα προγράμματα ανεξάρτητα από την διαμόρφωση της φυσικής μνήμης.
- Για την υλοποίηση του συστήματος μνήμης έχουν αναπτυχθεί τεχνολογίες για υλοποίηση:
  - ✓ ημιαγωγικών μνημών: άμεσης προσπέλασης μέσω διευθύνσεων (random access memories, RAM) που χρησιμοποιούνται ως μνήμες προσωρινής αποθήκευσης,
  - ✓ ηλεκτρικά απαλείψιμων επαναπρογραμματιζόμενων ROM (electrically erasable programable ROM, EEPROM) που διαγράφονται, διαβάζονται και εγγράφονται με εφαρμογή διαφορετικών σταθμών τάσης στις κυψελίδες τους και στην ίδια τεχνολογία βασίζονται οι μνήμες Flash και SSD (solid-state drives),
  - ✓ μαγνητικών μνημών (μαγνητικοί δίσκοι) και οπτικών μνημών (CDROM, DVD).

Ημιαγωγικές μνήμες RAM

- Οι ημιαγωγικές μνήμες RAM αποτελούνται από:
  - ✓ κυψελίδες (cells) καθεμία από τις οποίες αποθηκεύει ένα δυαδικό ψηφίο,
  - ✓ αποκωδικοποιητές και
  - ✓ κυκλώματα τριών καταστάσεων, ενίσχυσης σημάτων.



Κυψελίδα RAM

επιλογή = 1, εγγραφή = 0 ⇒ Ανάγνωση περιεχομένου

επιλογή = 1, εγγραφή = 1 ⇒ Εγγραφή περιεχομένου

Ημιαγωγικές μνήμες RAM

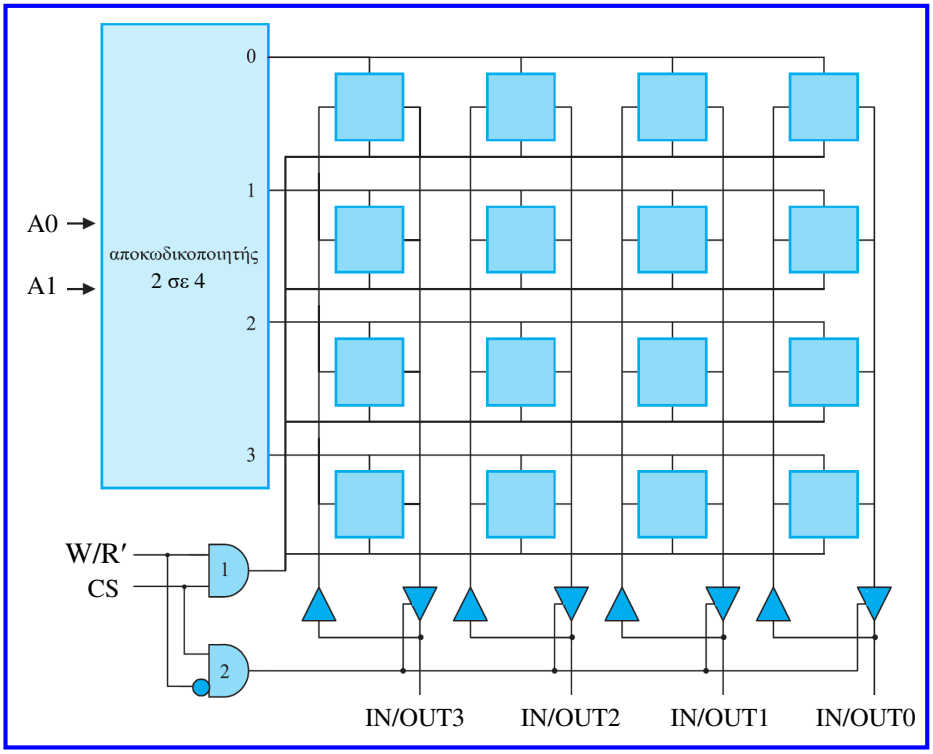
RAM 16 ψηφίων

Σε κάθε εγγραφή ή ανάγνωση, αποθηκεύεται ή διαβάζεται μια λέξη 4 ψηφίων.

2 ψηφία (A0, A1) για διευθυνσιοδότησή που τίθενται ως είσοδοι σε αποκωδικοποιητή 2 σε 4.

CS (chip select) = 0 ⇒ καμία ενέργεια.

CS (chip select) = 1 ⇒ ανάγνωση ή εγγραφή μιας γραμμής, ανάλογα με τιμές των σημάτων W/R', A0, A1.

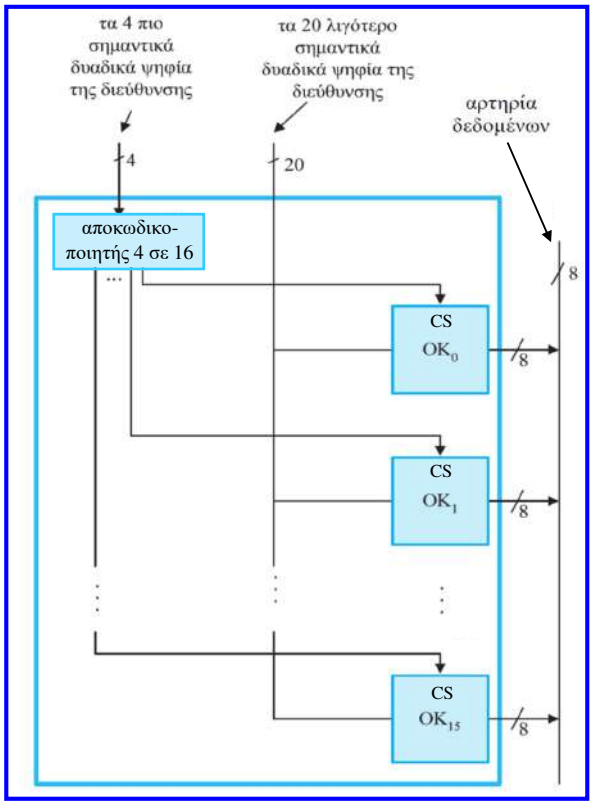


Ημιαγωγικές μνήμες RAM

- Επειδή συνήθως η χωρητικότητας της απαιτούμενης μνήμης είναι μεγάλη, δεν μπορεί να υλοποιηθεί σε ένα ολοκληρωμένο κύκλωμα (ΟΚ, integrated circuit, IC, chip) μνήμης.
- Επομένως, απαιτείται η χρησιμοποίηση περισσότερων ΟΚ μνήμης.

Παράδειγμα: με διαθέσιμα ΟΚ μνήμης 1 Mbyte με εσωτερική οργάνωση 1 byte ανά θέση μνήμης (δηλαδή, συνολικά 2^{20} bytes), για να σχεδιάσουμε ένα σύστημα μνήμης 16 Mbytes με οργάνωση 1 byte ανά θέση μνήμης, μπορούμε να χρησιμοποιήσουμε 16 ΟΚ από τα διαθέσιμα.

Απαιτείται διεύθυνση 24 ψηφίων: 4 με την αποκωδικοποίηση των οποίων επιτυγχάνεται η επιλογή κάθε ΟΚ μνήμης και 20 για την επιλογή ενός εκ των 2^{20} bytes εντός του επιλεγμένου ΟΚ μνήμης.

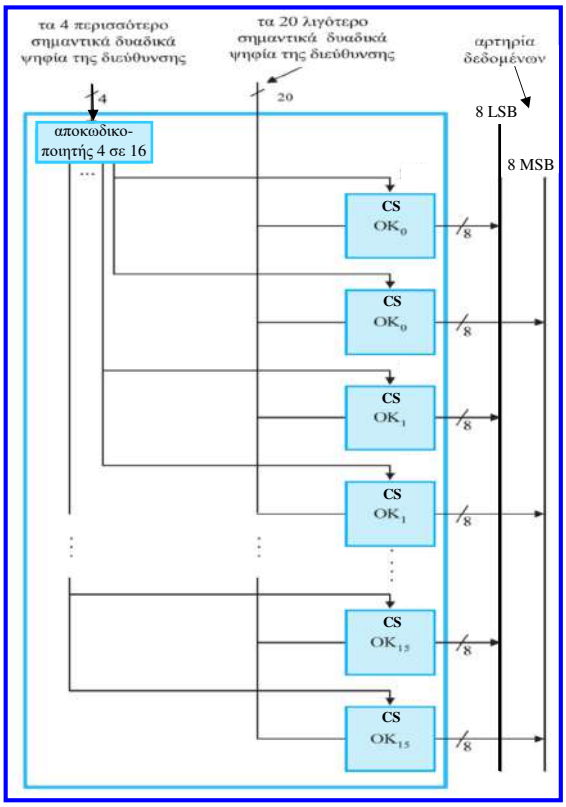


Ημιαγωγικές μνήμες RAM

Υπάρχουν περιπτώσεις όπου διαθέτουμε ΟΚ μνήμης με πλήθος ψηφίων ανά θέση μνήμης μικρότερου του απαιτούμενου

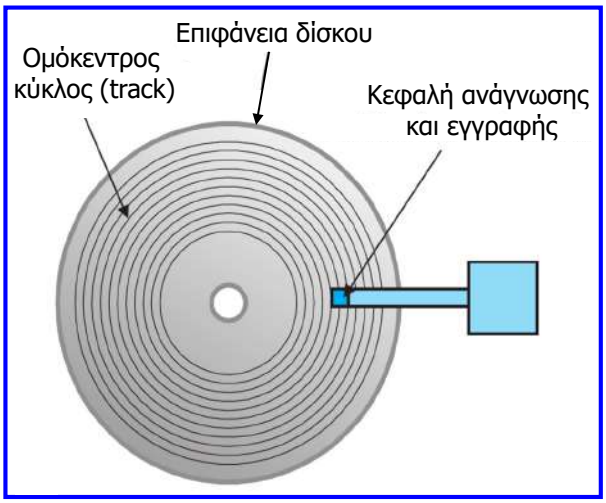
Παράδειγμα: με διαθέσιμα ΟΚ μνήμης 1 Mbyte με εσωτερική οργάνωση 1 byte ανά θέση μνήμης, για να σχεδιάσουμε ένα σύστημα μνήμης 32 Mbytes με οργάνωση μία λέξη των 2 bytes ανά θέση μνήμης, μπορούμε να χρησιμοποιήσουμε 16 ζεύγη από τα διαθέσιμα ΟΚ.

Απαιτείται διεύθυνση 24 ψηφίων: 4 με την αποκωδικοποίηση των οποίων επιτυγχάνεται η επιλογή κάθε ζεύγους ΟΚ μνήμης και 20 για την επιλογή ενός εκ των 2^{20} λέξεων εντός του επιλεγμένου ζεύγους ΟΚ μνήμης. Το ένα ΟΚ κάθε ζεύγους αποθηκεύει το λιγότερο σημαντικό byte, ενώ το άλλο αποθηκεύει το περισσότερο σημαντικό byte κάθε λέξης.



Μονάδα μαγνητικού δίσκου

- Η αποθήκευση πληροφορίες γίνεται σε έναν ή περισσότερες δίσκους, η επιφάνεια των οποίων επικαλύπτεται με μαγνητικό υλικό.
- Η εγγραφή της πληροφορίας γίνεται κατά μήκος ομόκεντρων κύκλων (tracks).
- Κάθε επιφάνεια έχει χιλιάδες ομόκεντρους κύκλους και κάθε ομόκεντρος κύκλος διαιρείται σε τμήματα ή τομείς (sectors), σε καθένα από τα οποία αποθηκεύεται μια σταθερή ποσότητα πληροφορίας.



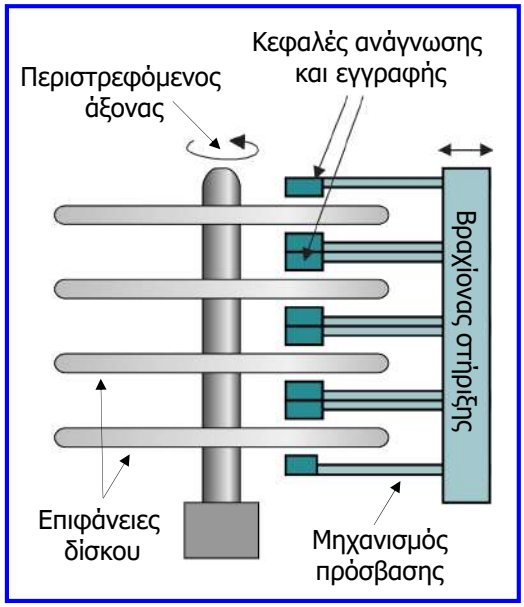
Παράδειγμα: μαγνητικός δίσκος διαμέτρου 2.5 ιντσών με 28 επιφάνειες, 31,330 ομόκεντρους κύκλους ανά επιφάνεια, 765 τμήματα ανά ομόκεντρο κύκλο και δυνατότητα αποθήκευσης 512 bytes ανά τμήμα.

Χωρητικότητα: 28 x 31,330 x 765 x 512 = 343,597,363,000 bytes ÷ 2^{30} = 320 GBytes.

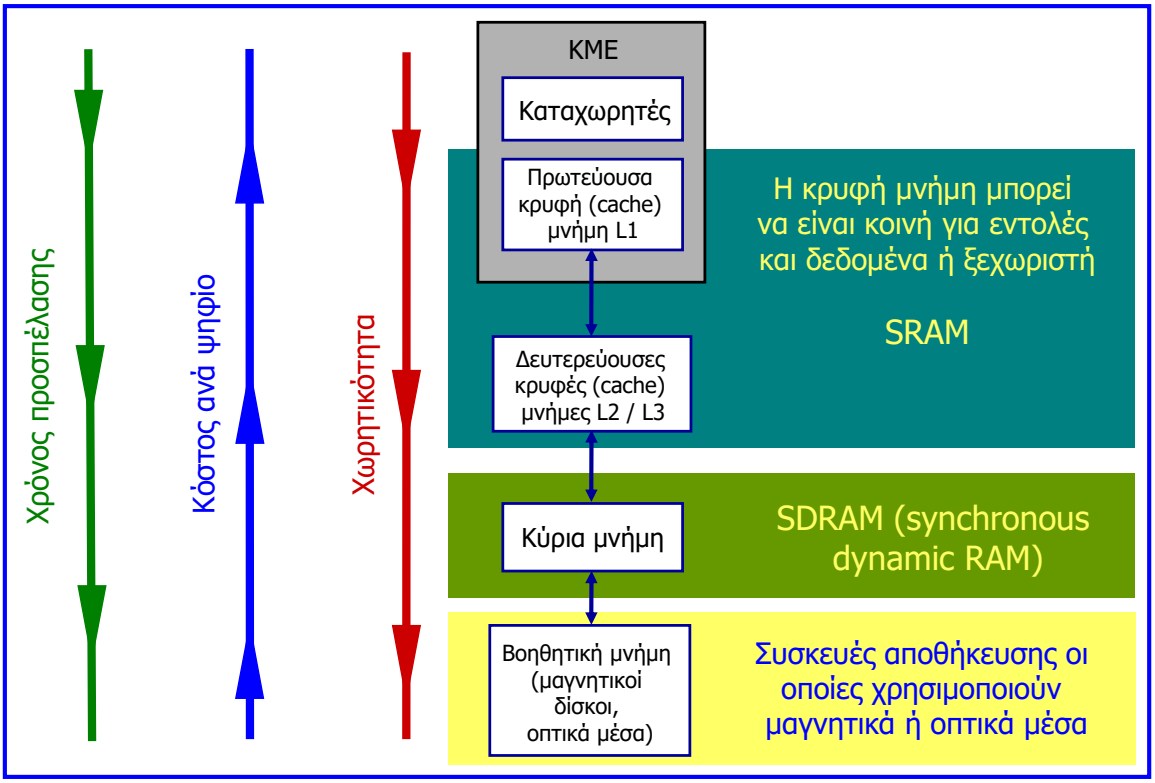
Μονάδα μαγνητικού δίσκου

- Η λειτουργία του συστήματος ελέγχεται από κύκλωμα (ελεγκτής δίσκου, disk controller).
- Για τη μεταφορά πληροφορίας από τον δίσκο στον καταχωρητή του ελεγκτή δίσκου με προορισμό την κύρια μνήμη, δηλώνεται η διεύθυνση κύριας μνήμης όπου θα αποθηκευτεί το πρώτο τμήμα πληροφορίας και η διεύθυνση από την οποία αρχίζει η πληροφορία στο δίσκο (επιφάνεια, ομόκεντρος κύκλος, τμήμα).
- Η καθυστέρηση μεταξύ παραλαβής μίας εντολής ανάγνωσης από το δίσκο και έναρξης μεταφοράς δεδομένων προς τον καταχωρητή του ελεγκτή είναι ο χρόνος προσπέλασης δίσκου (disk access time) και περιλαμβάνει δύο συνιστώσες:
  - ✓ Χρόνος αναζήτησης (seek time): χρόνος μετακίνησης κεφαλής στον επιθυμητό ομόκεντρο κύκλο.
  - ✓ Χρόνος αναμονής (latency, rotational delay): χρόνος έως ότου η αρχή του επιθυμητού τμήματος να βρεθεί κάτω από την κεφαλή.

Οι επιφάνειες της μονάδας περιστρέφονται με σταθερή ταχύτητα και ο βραχίονας στήριξης μετακινείται ώστε οι κεφαλές να τοποθετούνται στους ομόκεντρους κύκλους με την ίδια ακτίνα



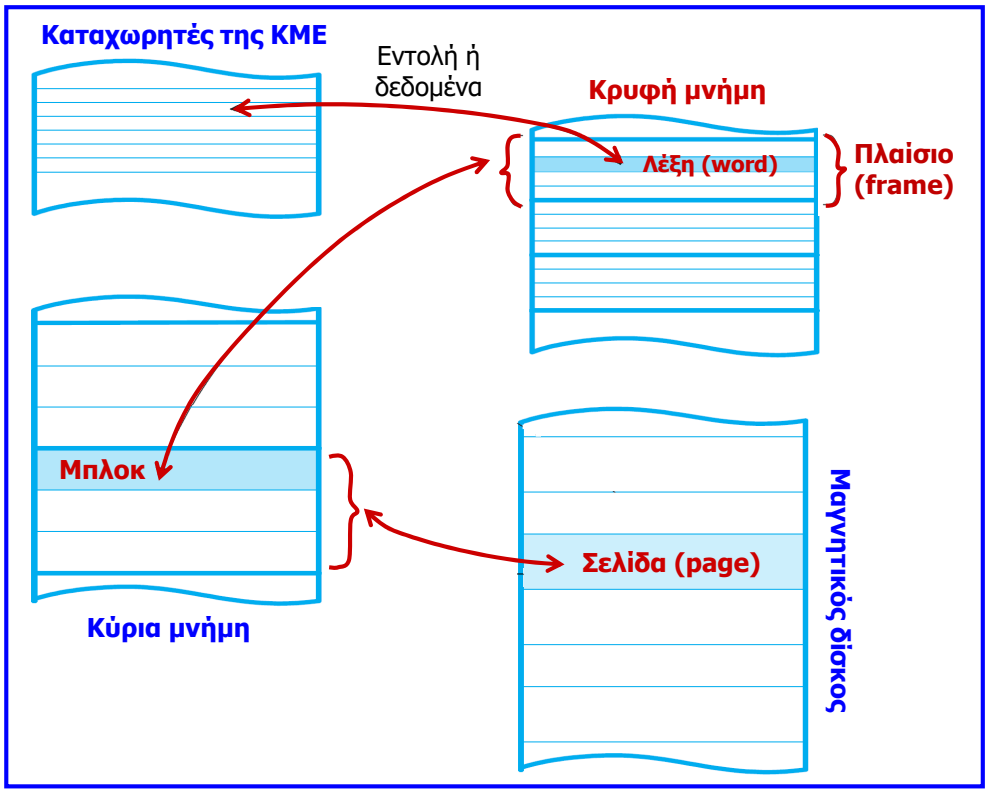
Ιεραρχία συστήματος μνήμης



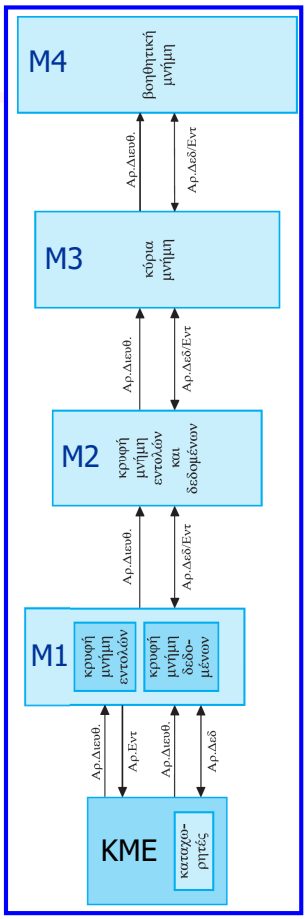
Ιεραρχία συστήματος μνήμης

- Η κύρια μνήμη είναι σχετικά αργή προκαλώντας αργοπορία στην ΚΜΕ με την κατάσταση να γίνεται χειρότερη όταν η απαιτούμενη πληροφορία πρέπει να μεταφερθεί στην κύρια μνήμη από τη βοηθητική.
- Για τη βελτίωση της ταχύτητας, χρησιμοποιείται η κρυφή μνήμη (cache memory) που είναι σχετικά μικρής χωρητικότητας και χρησιμοποιείται για την αποθήκευση πληροφορίας που αναμένεται ότι θα χρησιμοποιηθεί άμεσα ή με μεγάλη συχνότητα στο μέλλον.
- Η βελτίωση επέρχεται διότι, η πλειονότητα των προγραμμάτων χαρακτηρίζεται από τοπικότητα (locality) αναφορών, που σημαίνει ότι η πληροφορία (εντολές και δεδομένα) που χρησιμοποιήθηκε πρόσφατα είναι πιθανόν να ξαναχρησιμοποιηθεί στο άμεσο μέλλον, καθώς και ότι η πληροφορία που βρίσκεται κοντά στην πληροφορία που χρησιμοποιείται, είναι πιθανό να χρησιμοποιηθεί στο άμεσο μέλλον.
- Στην κρυφή μνήμη αποθηκεύονται αντίγραφα (πλαίσια) μέρους της πληροφορίας της κύριας μνήμης και όταν η απαιτούμενη από την ΚΜΕ πληροφορία βρίσκεται στην κρυφή μνήμη, αυτή προσπελαύνεται πολύ γρήγορα, αποφεύγοντας την καθυστέρηση της προσπέλασης της κύριας μνήμης.
- Εάν η ζητούμενη πληροφορία δεν υπάρχει στην κρυφή μνήμη, ένα τμήμα (block) πληροφορίας που περιέχει τη ζητούμενη πληροφορία μεταφέρεται στην κρυφή μνήμη.

Ιεραρχία συστήματος μνήμης



Απόδοση συστήματος ιεραρχικής μνήμης



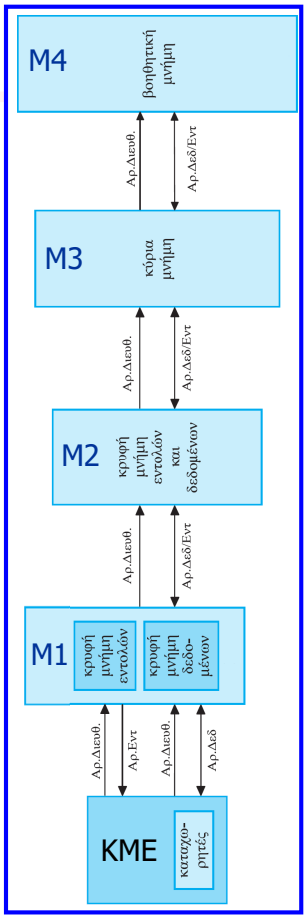
- Σκοπός του σχεδιασμού ιεραρχίας μνήμης είναι να επιτευχθεί:
  - ✓ απόδοση που να προσεγγίζει όσο γίνεται περισσότερο την πιο γρήγορη διάταξη μνήμης M1 και
  - ✓ κόστος ανά δυαδικό ψηφίο μνήμης που να προσεγγίζει όσο γίνεται περισσότερο τη φθηνότερη διάταξη μνήμης Mn.
- Η απόδοση της ιεραρχικής μνήμης εξαρτάται από:
  - ✓ την στατιστική των αναφερόμενων διευθύνσεων, δηλαδή τη σειρά και τη συχνότητα με την οποία τα προγράμματα που τρέχουν στο συγκεκριμένο υπολογιστικό σύστημα παράγουν τις διάφορες λογικές διευθύνσεις,
  - ✓ το χρόνο προσπέλασης από την ΚΜΕ κάθε επιπέδου μνήμης,
  - ✓ τη χωρητικότητα αποθήκευσης κάθε επιπέδου μνήμης,
  - ✓ το μέγεθος των μπλοκ πληροφορίας που μεταφέρονται μεταξύ διαδοχικών επιπέδων μνήμης και
  - ✓ τη στρατηγική (allocation algorithm) που χρησιμοποιείται για τον προσδιορισμό των περιοχών της Mi στις οποίες θα μεταφερθούν μπλοκ πληροφορίας από την Mi+1.

Κόστος συστήματος ιεραρχικής μνήμης

Μέσο κόστος ανά δυαδικό ψηφίο συστήματος ιεραρχικής μνήμης:

$$C = \frac{C_1 S_1 + C_2 S_2 + \ldots + C_n S_n}{S_1 + S_2 + \ldots + S_n}$$

C_i: κόστος ανά δυαδικό ψηφίο της μνήμης Mi
S_i: Χωρητικότητα σε δυαδικά ψηφία της μνήμης Mi

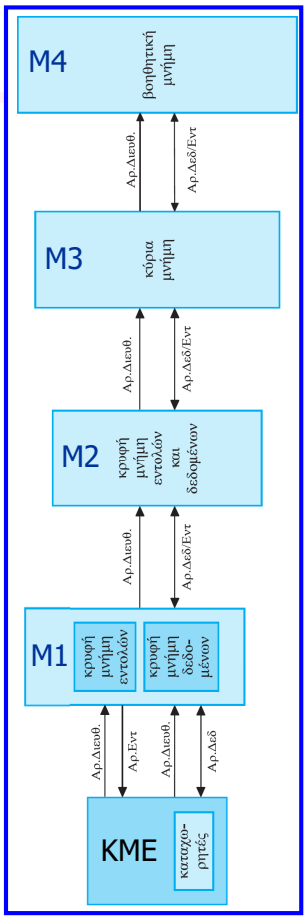


Λόγος ή ρυθμός επιτυχίας

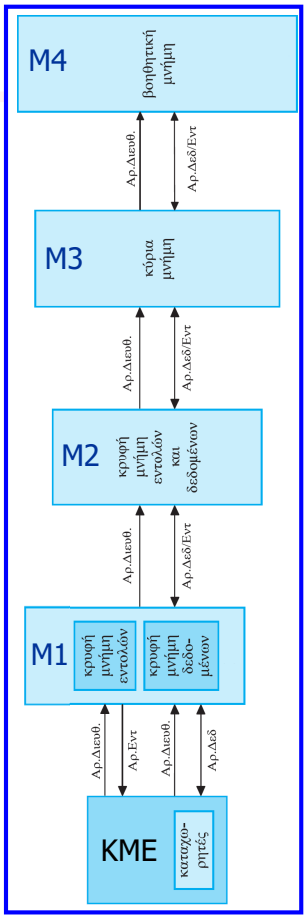
Λόγος επιτυχίας (hit ratio, rate) E_i της μονάδας μνήμης του επιπέδου i είναι ως η πιθανότητα η ζητούμενη πληροφορία από την ΚΜΕ να βρίσκεται στην μνήμη του επιπέδου i:

$$E_i = \frac{N_1 + N_2 + \ldots + N_{i-1} + N_i}{N}$$

$N_1, N_2, \ldots, N_{i-1}, N_i$: πλήθος προσπελάσεων που ικανοποιούνται από τα επίπεδα μνήμης $1, 2, \ldots, i-1$,
N: σύνολο αναφορών στην ιεραρχία μνήμης.



Χρόνος προσπέλασης



- Χρόνος προσπέλασης από την ΚΜΕ μιας λέξης που βρίσκεται στο επίπεδο μνήμης i+1:

$$T_{i+1} = \tau_1 + \tau_{B_2} + \tau_{B_3} + \ldots + \tau_{B_{i+1}} = \tau_1 + \sum_{j=2}^{i+1} \tau_{B_j} \qquad T_1 = \tau_1$$

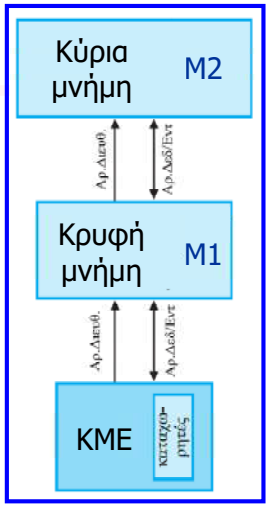
τ_{Bi+1}: χρόνος μεταφοράς ενός μπλοκ πληροφορίας από το επίπεδο μνήμης i+1 στο επίπεδο μνήμης i, τ_1: χρόνος προσπέλασης από την ΚΜΕ της μνήμης Μ1.

- Μέσος χρόνος προσπέλασης πληροφορίας που βρίσκεται σε ιεραρχία μνημών με ν επίπεδα:

$$T = \sum_{i=1}^{v} (E_i - E_{i-1}) T_i \qquad E_0 = 0,\ E_v = 1$$

Ο όρος $E_i - E_{i-1}$ δηλώνει την πιθανότητα η απαίτηση να ικανοποιηθεί από το επίπεδο μνήμης i και όχι από το επίπεδο μνήμης i-1.

Κρυφή μνήμη & ενημέρωση κύριας μνήμης

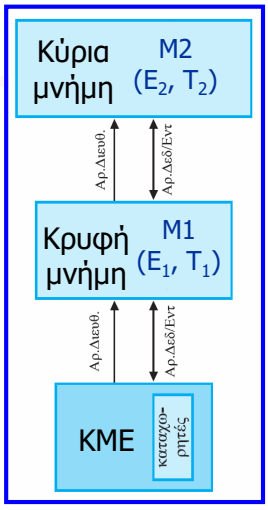


- Θεωρούμε ότι στο ιεραρχικό σύστημα μνήμης υπάρχει μόνο ένα επίπεδο κρυφής μνήμης μεταξύ της ΚΜΕ και της κύριας μνήμης.
- Όταν πρόκειται για λειτουργία ανάγνωσης και η επιθυμητή από την ΚΜΕ λέξη βρίσκεται στην κρυφή μνήμη, το πλαίσιο που περιέχει τη λέξη αυτή προωθείται στην ΚΜΕ και τότε έχουμε επιτυχία (hit) ανάγνωσης.
- Όταν πρόκειται για λειτουργία εγγραφής και το μπλοκ στο οποίο αντιστοιχεί η λέξη, που πρόκειται να εγγραφεί από την ΚΜΕ στη μνήμη, βρίσκεται στην κρυφή μνήμη, έχουμε επιτυχία εγγραφής και πρέπει να γίνει ενημέρωση της κύριας μνήμης, η οποία μπορεί να επιτευχθεί με δύο τακτικές:
- Τακτική άμεσης ενημέρωσης (write through): Η θέση στην κρυφή μνήμη και η αντίστοιχη θέση στην κύρια μνήμη ενημερώνονται ταυτόχρονα, δηλαδή η λέξη γράφεται ταυτόχρονα στο πλαίσιο της κρυφής μνήμης και στο μπλοκ της κύριας μνήμης.
- Τακτική τελικής ενημέρωσης (write back): Ενημερώνεται η θέση (δηλ. γράφεται η λέξη) στην κρυφή 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 πλαίσιο της κρυφής μνήμης που περιλαμβάνει τη λέξη αυτή πρόκειται να απομακρυνθεί από την κρυφή μνήμη, ώστε να δημιουργηθεί χώρος για νέο μπλοκ κύριας μνήμης. Στην περίπτωση αυτή γίνεται ενημέρωση (εγγραφή) στην κύρια μνήμη όλου του πλαισίου της κρυφής μνήμης που περιλαμβάνει τη λέξη.

Κρυφή μνήμη & ενημέρωση κύριας μνήμης

- Όταν πρόκειται για λειτουργία ανάγνωσης και το μπλοκ που περιλαμβάνει την επιθυμητή λέξη δεν βρίσκεται στην κρυφή μνήμη, έχουμε αποτυχία (miss) ανάγνωσης.
- Τότε, το μπλοκ στο οποίο αντιστοιχεί η επιθυμητή λέξη, αντιγράφεται από την κύρια μνήμη στην κρυφή μνήμη.
- Όταν το μπλοκ φορτωθεί στην κρυφή μνήμη, η επιθυμητή λέξη προωθείται στην ΚΜΕ.
- Όταν πρόκειται για λειτουργία εγγραφής και το μπλοκ στο οποίο αντιστοιχεί η λέξη που πρόκειται να εγγραφεί από την ΚΜΕ στη μνήμη δεν βρίσκεται στην κρυφή μνήμη, έχουμε αποτυχία εγγραφής και πρέπει να γίνει ενημέρωση της κύριας μνήμης, η οποία μπορεί να επιτευχθεί με δύο τακτικές:
- Τακτική προσκόμισης κατά την εγγραφή (write allocate): Το μπλοκ στο οποίο αντιστοιχεί η επιθυμητή λέξη αντιγράφεται από την κύρια στην κρυφή μνήμη και ακολουθείται η τακτική ενημέρωσης της κύριας μνήμης που χρησιμοποιεί η κρυφή μνήμη στην περίπτωση επιτυχίας εγγραφής. Η επιθυμητή λέξη γράφεται ταυτόχρονα στο πλαίσιο της κρυφής μνήμης και στο μπλοκ της κύριας μνήμης (write through) ή η θέση της επιθυμητής λέξης στην κύρια μνήμη ενημερώνεται μόνο όταν το πλαίσιο της κρυφής μνήμης που περιλαμβάνει τη λέξη, πρόκειται να απομακρυνθεί από την κρυφή μνήμη (write back), με εγγραφή στην κύρια μνήμη ολόκληρου του αντίστοιχου πλαισίου της κρυφής μνήμης.
- Τακτική μη προσκόμισης κατά την εγγραφή (write around ή no write allocate): Η επιθυμητή λέξη εγγράφεται κατευθείαν στο μπλοκ της κύριας μνήμης που αντιστοιχεί.

Κρυφή μνήμη & ενημέρωση κύριας μνήμης



- Μέσος χρόνος λειτουργίας ανάγνωσης του συστήματος μνήμης, όπως τον αντιλαμβάνεται η ΚΜΕ.

$$T = \sum_{i=1}^{2}(E_i - E_{i-1}) = (E_1 - E_0)\cdot T_1 + (E_2 - E_1)\cdot T_2 = (E_1 - 0)\cdot T_1 + (1 - E_1)\cdot T_2$$

$$T = E_C \cdot T_C + (1 - E_C)\cdot T_M$$

E_C: λόγος επιτυχίας κρυφής μνήμης

T_C: χρόνος ανάγνωσης πληροφορίας κρυφής μνήμης από την ΚΜΕ

T_M: χρόνος ανάγνωσης πληροφορίας κύριας μνήμης από την ΚΜΕ

- Σε περίπτωση επιτυχίας ο χρόνος ανάγνωσης είναι T_C, ενώ σε περίπτωση αποτυχίας είναι T_M (ο χρόνος T_M περιλαμβάνει τον χρόνο μεταφοράς ενός μπλοκ δεδομένων από την κύρια μνήμη στην κρυφή μνήμη και τον χρόνο προσπέλασης της κρυφής μνήμης από την ΚΜΕ).
- Σε αρκετά συστήματα, η αναφορά στην κύρια μνήμη γίνεται παράλληλα με την προσπέλαση της κρυφής μνήμης, έτσι ώστε αν ο έλεγχος της κρυφής μνήμης είναι ανεπιτυχής (αποτυχία), να έχει ήδη αρχίσει ο κύκλος προσπέλασης της κύριας μνήμης.
- Όταν πρόκειται για λειτουργία εγγραφής, για τον υπολογισμό του μέσου χρόνου προσπέλασης του συστήματος μνήμης, πρέπει να λαμβάνεται υπόψη η τακτική ενημέρωσης της κύριας μνήμης που ακολουθείται από το σύστημα μνήμης.

Κρυφή μνήμη & ενημέρωση κύριας μνήμης

Παράδειγμα 1: Κρυφή μνήμη με πλαίσια 8 λέξεων, T_C = 1 κύκλος ρολογιού και E_C = 80%. T_M = 1 κύκλος για μεταφορά της εναρκτήριας διεύθυνσης από την ΚΜΕ στην κύρια μνήμη, 10 κύκλοι για προσπέλαση της 1ης λέξης, 4 κύκλοι ανά λέξη για την προσπέλαση των επόμενων λέξεων του μπλοκ. Η μεταφορά μιας λέξης από την κύρια στην κρυφή μνήμη απαιτεί 1 κύκλο, όσο και η προσπέλαση / μεταφορά μιας λέξης από την κρυφή μνήμη στην ΚΜΕ. Η αναφορά στην κύρια μνήμη γίνεται παράλληλα με την προσπέλαση της κρυφής.

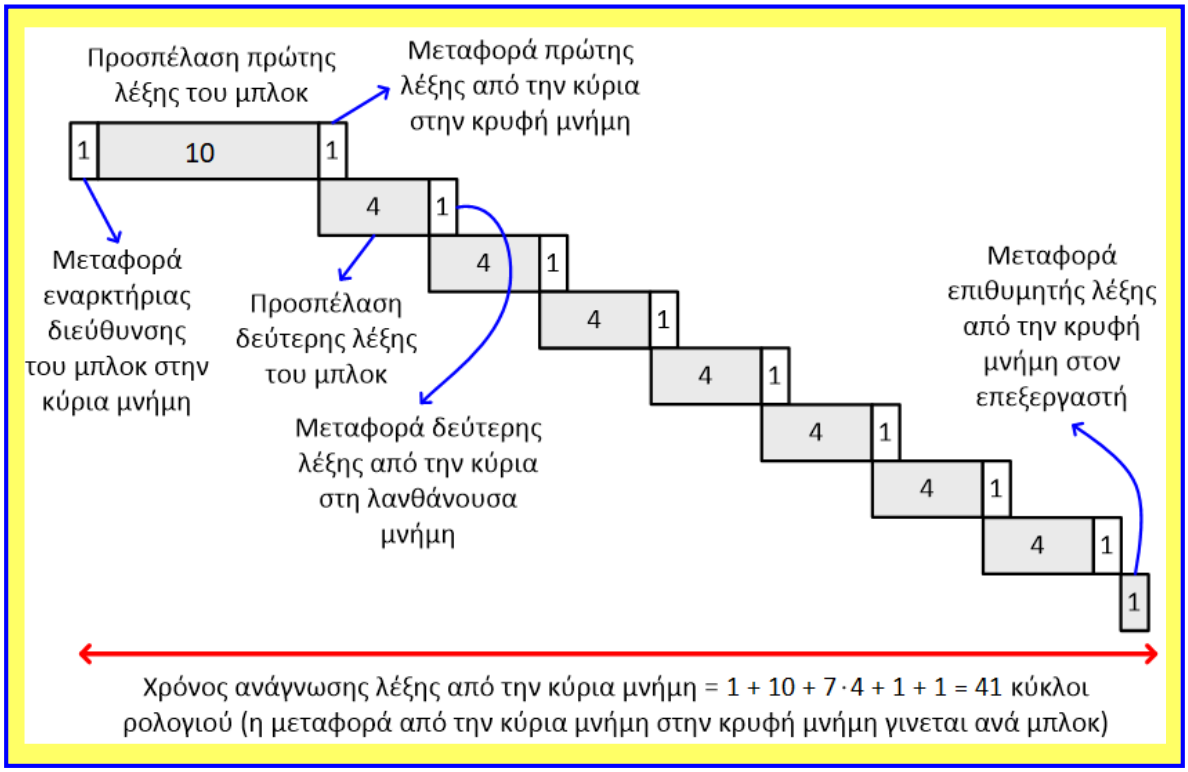
- Χρόνος ανάγνωσης λέξης από την κρυφή μνήμη: T_C = 1 κύκλος ρολογιού.
- Χρόνος ανάγνωσης λέξης από την κύρια μνήμη: $T_M = 1 + 10 + 7 \times 4 + 1 + 1 = 41$ κύκλοι ρολογιού (η μεταφορά από την κύρια στην κρυφή μνήμη γίνεται ανά μπλοκ).
- Μέσος χρόνος ανάγνωσης συστήματος μνήμης:

$$T = E_C \cdot T_C + (1 - E_C) \cdot T_M = 0.8 \cdot 1 + 0.2 \cdot 41 = \text{9 κύκλοι ρολογιού}$$

Παράδειγμα 2: Στην παραπάνω κρυφή μνήμη σε περίπτωση επιτυχίας εγγραφής ακολουθείται η τακτική άμεσης ενημέρωσης (write through), ενώ σε περίπτωση αποτυχίας εγγραφής ακολουθείται η τακτική μη προσκόμισης κατά την εγγραφή (write around).

- Χρόνος εγγραφής μιας λέξης στην κύρια μνήμη: 1 + 10 = 11 κύκλοι ρολογιού.
- Αφού με το συνδυασμό των δύο τακτικών κάθε λέξη εγγράφεται στην κρυφή και στην κύρια μνήμη και η αναφορά στην κύρια μνήμη γίνεται παράλληλα με την εγγραφή στην κρυφή, ο μέσος χρόνος εγγραφής στο σύστημα μνήμης είναι 11 κύκλοι ρολογιού.

Κρυφή μνήμη & ενημέρωση κύριας μνήμης

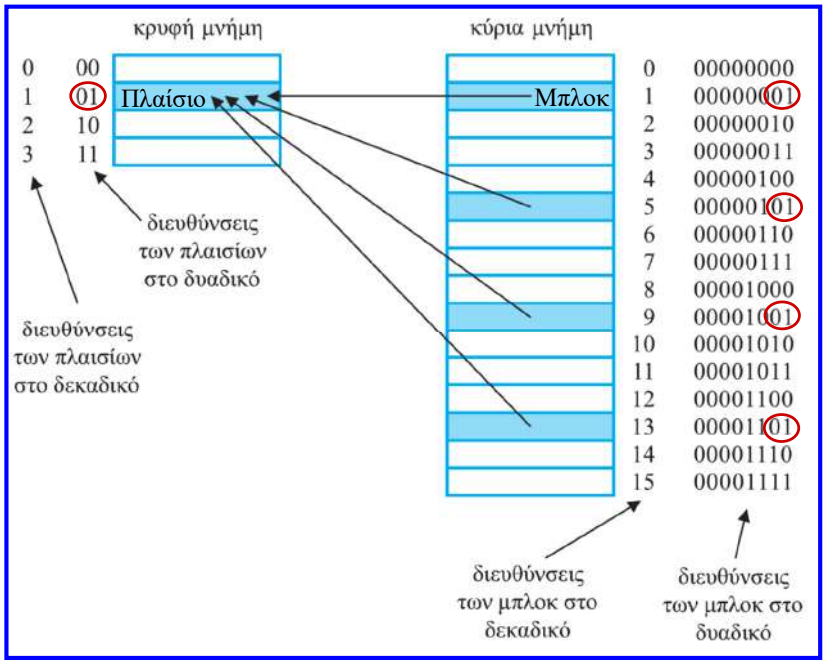


Οργάνωση κρυφής μνήμης

- Θεωρούμε ξανά ότι στο ιεραρχικό σύστημα μνήμης υπάρχει μόνο ένα επίπεδο κρυφής μνήμης (cache memory) μεταξύ της ΚΜΕ και της κύριας μνήμης.
- Το πλήθος των μπλοκ κύριας μνήμης υπερβαίνει σημαντικά το πλήθος πλαισίων κρυφής μνήμης, αφού το μέγεθος της κρυφής μνήμης είναι περίπου 3 τάξεις μεγέθους μικρότερο.
- Ο τρόπος απεικόνισης (placement policy) των μπλοκ της κύριας μνήμης σε πλαίσια της κρυφής μνήμης, καθορίζει την οργάνωση της κρυφής μνήμης:
  - ✓ άμεση οργάνωση (direct mapping),
  - ✓ οργάνωση πλήρους συσχέτισης (fully associative) και
  - ✓ οργάνωση τ-τρόπων συνόλου συσχέτισης (t-way set associative).
- Το πλήθος λέξεων ανά πλαίσιο κρυφής μνήμης, το πλήθος πλαισίων της κρυφής μνήμης και το πλήθος μπλοκ κύριας μνήμης είναι δυνάμεις του 2.
- Το πλαίσιο κρυφής μνήμης και το μπλοκ κύριας μνήμης περιέχουν το ίδιο πλήθος λέξεων.
- Ωστόσο, ένα πλαίσιο κρυφής μνήμης εκτός από τις λέξεις δεδομένων (που περιέχονται στο μπλοκ κύριας μνήμης) περιλαμβάνει και έναν αριθμό δυαδικών ψηφίων που αναφέρεται ως κατάλογος ή μνήμη ετικετών (tag directory).
- Η ετικέτα (tag) κάθε πλαισίου είναι ένα τμήμα της διεύθυνσης της κύριας μνήμης που υποδεικνύει ποιο μπλοκ κύριας μνήμης βρίσκεται κάθε στιγμή αποθηκευμένο στο πλαίσιο.

Κρυφή μνήμη άμεσης οργάνωσης

- Η διεύθυνση του πλαισίου της κρυφής μνήμης, στο οποίο θα τοποθετηθεί το μπλοκ της κύριας μνήμης με διεύθυνση Μ, δίνεται από το υπόλοιπο της διαίρεσης του Μ δια του πλήθους των πλαισίων της κρυφής μνήμης.
- Σε κάθε πλαίσιο μπορεί κάθε χρονική στιγμή να βρίσκεται μόνο ένα μπλοκ της κύριας μνήμης από το σύνολο των μπλοκ που το υπόλοιπο της διαίρεσης των διευθύνσεών τους δια του πλήθους των πλαισίων δίνει τον αριθμό του πλαισίου.

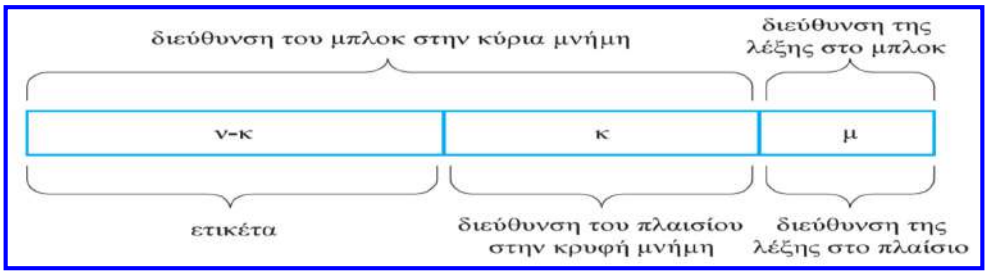


- Η διεύθυνση του πλαισίου ισούται με το δύο λιγότερο σημαντικά ψηφία της διεύθυνσης των μπλοκ που μπορούν να τοποθετηθούν στο πλαίσιο αυτό.

Κρυφή μνήμη άμεσης οργάνωσης

- Πλήθος μπλοκ κύριας μνήμης: 2^{v}
- Πλήθος λέξεων ανά μπλοκ κύριας μνήμης και ανά πλαίσιο κρυφής μνήμης : $2^{μ}$
- Μέγεθος κύριας μνήμης: $2^{v+μ}$ θέσεις (λέξεις)
- Πλήθος πλαισίων κρυφής μνήμης: $2^{κ}$
- Μέγεθος της κρυφής μνήμης: $2^{κ+μ}$ θέσεις (λέξεις)

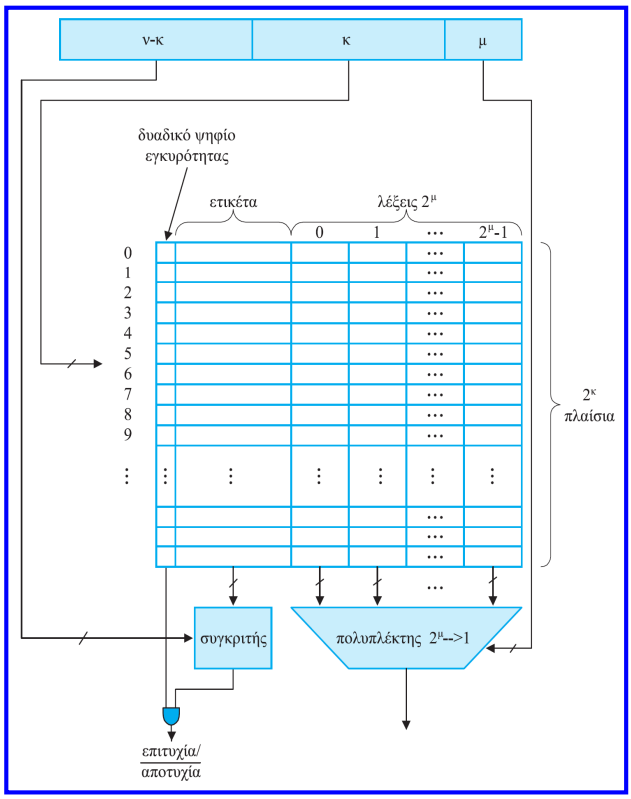
Τρόπος χρήσης της διεύθυνσης που παράγει η ΚΜΕ
για προσπέλαση κρυφής μνήμης με άμεση οργάνωση



- Όταν ένα μπλοκ κύριας μνήμης υπάρχει σε ένα πλαίσιο, η ετικέτα που συνδέεται με το πλαίσιο αυτό περιέχει τα v-k περισσότερο σημαντικά ψηφία της διεύθυνσης του μπλοκ στην κύρια μνήμη.

Κρυφή μνήμη άμεσης οργάνωσης

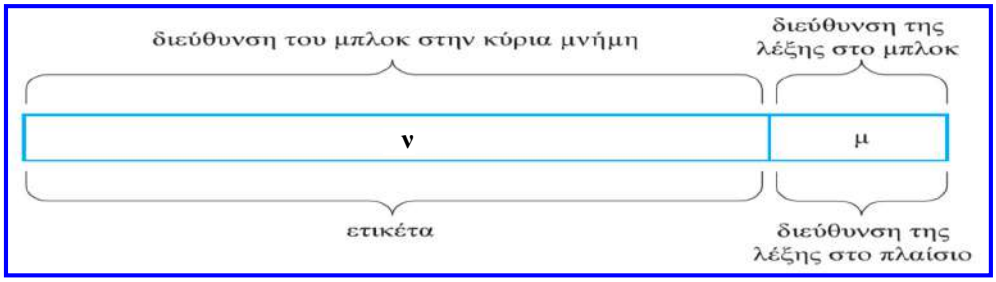
- Τα ψηφία της ετικέτας που είναι αποθηκευμένα στη κρυφή μνήμη (στη θέση που υποδεικνύουν τα κ ψηφία της διεύθυνσης) συγκρίνονται με τα ψηφία του πεδίου της ετικέτας της διεύθυνσης.
- Επειδή κατά την 1η προσπέλαση ενός πλαισίου, η επιθυμητή ετικέτα μπορεί να είναι ίδια με την ετικέτα που είναι αποθηκευμένη στην κρυφή μνήμη και με την έναρξη εκτέλεσης ενός προγράμματος η κρυφή μνήμη έχει τυχαία πληροφορία, μπορεί η τυχαία πληροφορία να ληφθεί ως επιθυμητή.
- Έτσι, κατά την έναρξη εκτέλεσης ενός προγράμματος όλα τα ψηφία εγκυρότητας γίνονται 0 και όταν προσκομίζεται ένα μπλοκ σε ένα πλαίσιο, τότε στο αντίστοιχο ψηφίο εγκυρότητας γίνεται 1.
- Εάν οι ετικέτες που συγκρίθηκαν είναι ίδιες και το ψηφίο εγκυρότητας είναι 1, τότε συμβαίνει επιτυχία.



Κρυφή μνήμη με οργάνωση πλήρους συσχέτισης

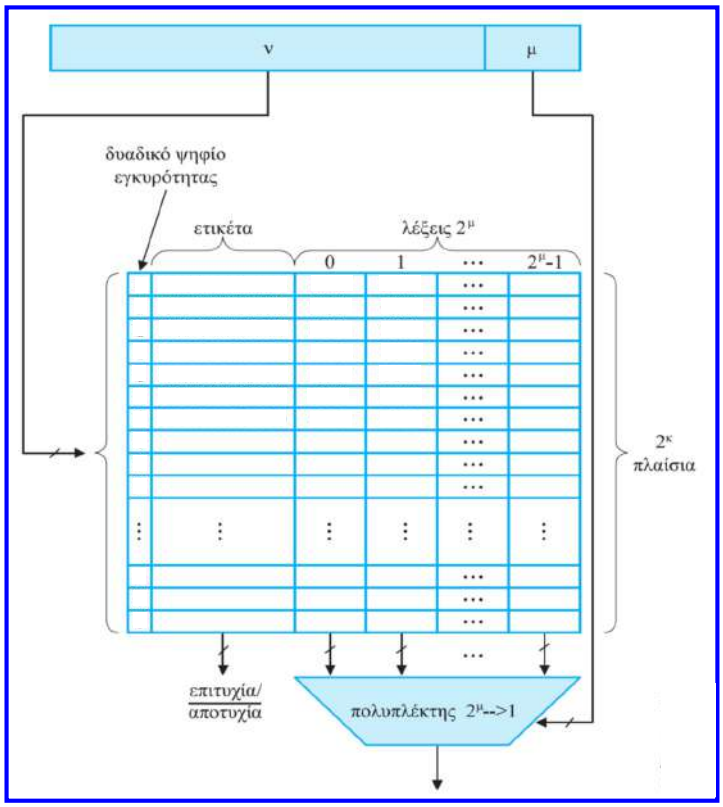
- Κάθε μπλοκ κύριας μνήμης μπορεί να τοποθετηθεί σε οποιοδήποτε πλαίσιο κρυφής μνήμης.
- Πλήθος μπλοκ κύριας μνήμης: 2^{v}
- Πλήθος λέξεων ανά μπλοκ κύριας μνήμης και ανά πλαίσιο κρυφής μνήμης : $2^{μ}$
- Μέγεθος κύριας μνήμης: $2^{v+μ}$ θέσεις (λέξεις)
- Πλήθος πλαισίων κρυφής μνήμης: $2^{κ}$
- Μέγεθος της κρυφής μνήμης: $2^{κ+μ}$ θέσεις (λέξεις)

Τρόπος χρήσης της διεύθυνσης που παράγει η ΚΜΕ
για προσπέλαση κρυφής μνήμης με οργάνωση πλήρους συσχέτισης



Κρυφή μνήμη με οργάνωση πλήρους συσχέτισης

- Το πεδίο της ετικέτας της διεύθυνσης που παράγει η ΚΜΕ συγκρίνεται ταυτόχρονα με όλες τις ετικέτες που είναι αποθηκευμένες στην κρυφή μνήμη.
- Εάν το πεδίο της ετικέτας είναι ίδιο με κάποια από τις αποθηκευμένες ετικέτες και το αντίστοιχο ψηφίο εγκυρότητας έχει τιμή 1, τότε συμβαίνει επιτυχία.
- Η πληροφορία του πλαισίου που συνδέεται με αυτή την ετικέτα εμφανίζεται στις εισόδους του πολυπλέκτη και με βάση το πεδίο διεύθυνσης της λέξης στο πλαίσιο, λαμβάνεται στην έξοδο του πολυπλέκτη η επιθυμητή λέξη.



Κρυφή μνήμη με οργάνωση πλήρους συσχέτισης

Παράδειγμα:

Κύρια μνήμη: 64K λέξεις

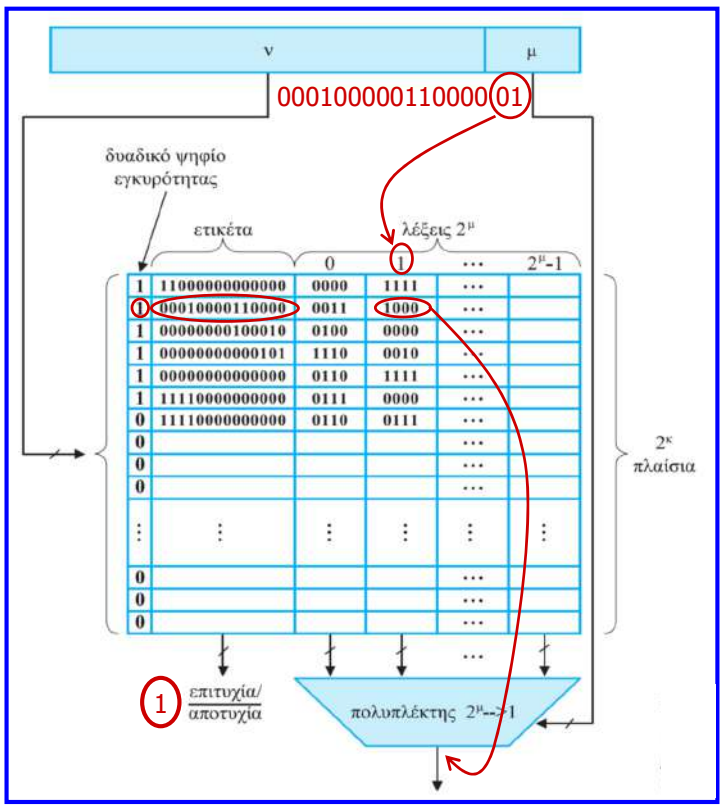
Κρυφή μνήμη: 64 λέξεις

Πλαίσιο: 4 λέξεις

Επομένως, η κρυφή μνήμη έχει 64 / 4 = 16 πλαίσια και αφού πλαίσιο και μπλοκ έχουν τον ίδιο αριθμό λέξεων, η κύρια μνήμη αποτελείται από 16K μπλοκ.

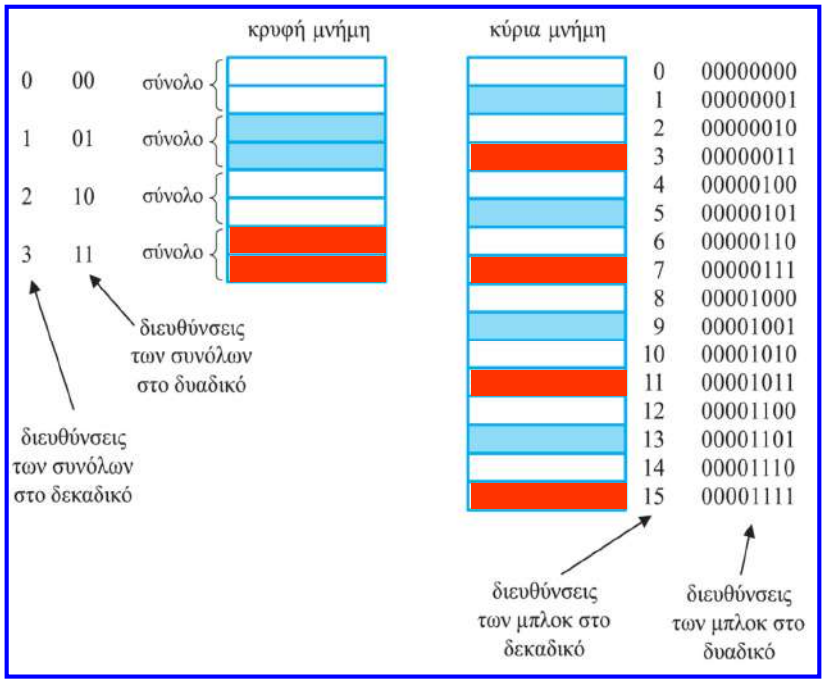
4 λέξεις ανά πλαίσιο = $2^2 \Rightarrow \mu = 2$

16K μπλοκ = $2^4 \cdot 2^{10} = 2^{14} \Rightarrow v = 14$



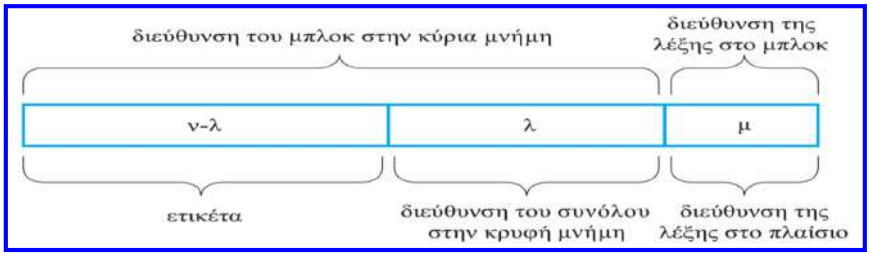
Κρυφή μνήμη τ-τρόπων συνόλου συσχέτισης

- Η κρυφή μνήμη θεωρείται ότι αποτελείται από ομάδες τ πλαισίων που λέγονται σύνολα (sets).
- Κάθε μπλοκ μπορεί να τοποθετηθεί σε οποιοδήποτε πλαίσιο ενός συγκεκριμένου συνόλου.
- Σε κάθε σύνολο αντιστοιχεί μία διεύθυνση και η διεύθυνση συνόλου στο οποίο μπορεί να τοποθετηθεί το μπλοκ της κύριας μνήμης με διεύθυνση Μ, δίνεται από το υπόλοιπο της διαίρεσης του Μ δια του πλήθους των συνόλων της κρυφής μνήμης.



Κρυφή μνήμη τ-τρόπων συνόλου συσχέτισης

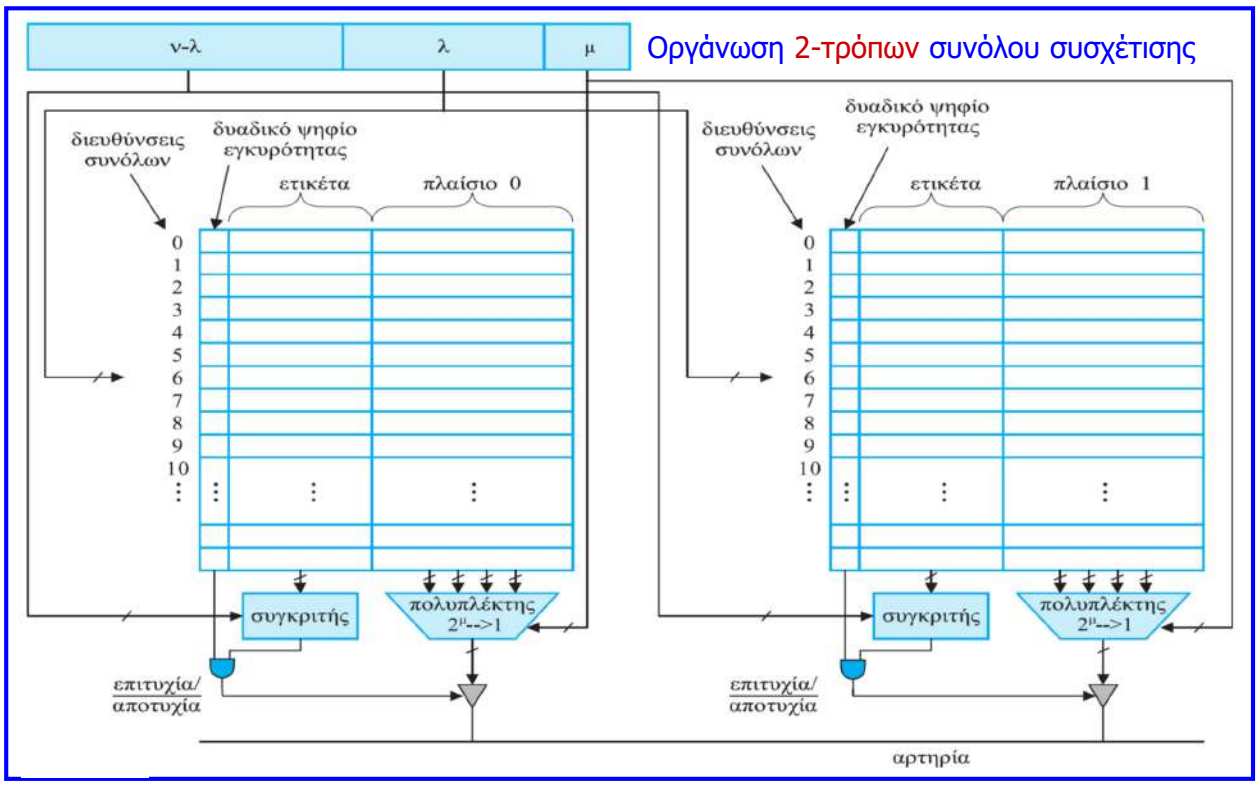
- Πλήθος μπλοκ κύριας μνήμης: 2^{ν}
- Πλήθος λέξεων ανά μπλοκ κύριας μνήμης και ανά πλαίσιο κρυφής μνήμης : 2^{μ}
- Μέγεθος κύριας μνήμης: $2^{\nu+\mu}$ θέσεις (λέξεις)
- Πλήθος συνόλων κρυφής μνήμης: 2^{λ}
- Πλήθος πλαισίων ανά σύνολο: $\tau = 2^{\xi}$
- Μέγεθος της κρυφής μνήμης: $2^{\lambda+\xi+\mu}$ θέσεις (λέξεις)



Τρόπος χρήσης της διεύθυνσης που παράγει η ΚΜΕ για προσπέλαση κρυφής μνήμης με οργάνωση τ-τρόπων συνόλου συσχέτισης

- Τα ψηφία του πεδίου της ετικέτας της διεύθυνσης συγκρίνονται με τις τ ετικέτες καθεμία από τις οποίες είναι αποθηκευμένη σε μία από τις τ RAM που συνιστούν την κρυφή μνήμη (στη θέση που υποδεικνύουν τα λ ψηφία της διεύθυνσης) .
- Εάν μία από τις συγκρίσεις έχει θετικό αποτέλεσμα και το αντίστοιχο ψηφίο εγκυρότητας έχει τιμή 1, τότε συμβαίνει επιτυχία.

Κρυφή μνήμη τ-τρόπων συνόλου συσχέτισης



Σύγκριση τρόπων οργάνωσης κρυφής μνήμης

- Τα πλεονεκτήματα της άμεσης οργάνωσης κρυφής μνήμης είναι ο μικρός χρόνος προσπέλασης και το χαμηλό κόστος υλοποίησης.
- Αντιθέτως, τα μειονεκτήματα της οργάνωσης πλήρους συσχέτισης είναι ο μεγάλος χρόνος προσπέλασης και το υψηλό κόστος υλοποίησης.
- Το γεγονός ότι στην άμεση οργάνωση, δύο ή περισσότερα μπλοκ της κύριας μνήμης που αντιστοιχούν στο ίδιο πλαίσιο της κρυφής μνήμης δεν μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη, έχει σαν αποτέλεσμα υψηλότερο ρυθμό αποτυχιών.
- Αντιθέτως, στην οργάνωση πλήρους συσχέτισης, επειδή η κρυφή μνήμη μπορεί να περιέχει οποιοδήποτε συνδυασμό από μπλοκ κύριας μνήμης, η οργάνωση αυτή παρέχει μικρότερο ρυθμό αποτυχιών.
- Η οργάνωση τ-τρόπων συνόλου συσχέτισης για $\tau = 1$ ισοδυναμεί με άμεση οργάνωση, ενώ όταν το πλήθος πλαισίων ανά σύνολο γίνεται ίσο με το πλήθος των πλαισίων της κρυφής μνήμης, τότε ισοδυναμεί με οργάνωση πλήρους συσχέτισης.
- Επομένως, η οργάνωση τ-τρόπων συνόλου συσχέτισης αποτελεί μια λύση εξισορρόπησης των μειονεκτημάτων και των πλεονεκτημάτων των δύο άλλων τρόπων οργάνωσης της κρυφής μνήμης.

Στρατηγικές απελευθέρωσης πλαισίων κρυφής μνήμης

- Όταν η ΚΜΕ ζητήσει πληροφορία από την κρυφή μνήμη και αυτή δεν είναι διαθέσιμη, τότε θα πρέπει το μπλοκ κύριας μνήμης που περιέχει την ζητούμενη πληροφορία να προσκομιστεί από την κύρια μνήμη.
- Εάν η κρυφή μνήμη είναι γεμάτη, τότε το μπλοκ που θα προσκομιστεί θα πρέπει να αντικαταστήσει κάποιο από τα αποθηκευμένα στην κρυφή μνήμη.
- Οι στρατηγικές απελευθέρωσης (replacement policies) πλαισίων κρυφής μνήμης για την προσκόμιση μπλοκ κύριας μνήμης καθορίζουν το πλαίσιο που θα αντικατασταθεί.
- Στην άμεση οργάνωση κρυφής μνήμης, αφού κάθε μπλοκ της κύριας μνήμης μπορεί να τοποθετηθεί μόνο σε ένα πλαίσιο κρυφής μνήμης, η στρατηγική απελευθέρωσης είναι δεδομένη και εύκολα υλοποιήσιμη.
- Στην οργάνωση πλήρους συσχέτισης ή στην οργάνωση συνόλου συσχέτισης, αφού ένα μπλοκ κύριας μνήμης μπορεί να τοποθετηθεί σε περισσότερα από ένα πλαίσια κρυφής μνήμης, θα πρέπει να επιλεγεί για αντικατάσταση ένα από αυτά τα πλαίσια.

Στρατηγικές απελευθέρωσης πλαισίων κρυφής μνήμης

- Τρεις (3) βασικές στρατηγικές για τον προσδιορισμό του πλαισίου που θα αντικατασταθεί:
  - ✓ Τυχαία επιλογή (random policy).
  - ✓ Επιλογή του μη χρησιμοποιηθέντος πρόσφατα πλαισίου (least-recently used policy, LRU): η στρατηγική βασίζεται στην τοπικότητα των αναφορών μνήμης και αντικαθιστά το πλαίσιο του οποίου δεν έχει γίνει χρήση (ανάγνωση ή εγγραφή) το μεγαλύτερο χρονικό διάστημα, έτσι ώστε να ελαττωθεί η πιθανότητα απομάκρυνσης πληροφορίας που θα χρειαστεί στο άμεσο μέλλον.
  - ✓ Επιλογή του πλαισίου που προσκομίστηκε πρώτο στην κρυφή μνήμη (first-in / first-out policy, FIFO).
- Για να μπορεί να υλοποιηθεί η στρατηγική LRU, η κρυφή μνήμη (μέσω του ελεγκτή της) κρατάει πληροφορία (δηλαδή ενημερώνει την τιμή ενός μετρητή ανά πλαίσιο) που αφορά τις προσπελάσεις (αναγνώσεις και εγγραφές) που γίνονται σε κάθε πλαίσιο.
- Για να μπορεί να υλοποιηθεί η στρατηγική FIFO, η κρυφή μνήμη (μέσω του ελεγκτή της) κρατάει πληροφορία που αφορά τη σειρά προσκόμισης (εγγραφής) των πλαισίων.

Κρυφή 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 κρυφή μνήμη με άμεση οργάνωση και χωρητικότητα 64 λέξεων με 8 λέξεις ανά πλαίσιο, θα υπολογίσουμε το λόγο επιτυχίας και το συνολικό πλήθος των bytes που διαβάζονται από την κύρια μνήμη.
- Θεωρούμε ότι αρχικά η κρυφή μνήμη είναι κενή ή ότι δεν περιέχει έγκυρα δεδομένα.

Κρυφή μνήμη: παραδείγματα τρόπων απεικόνισης

- Η κρυφή μνήμη χρησιμοποιεί άμεση οργάνωση και έχει χωρητικότητα 64 λέξεων με 8 λέξεις ανά πλαίσιο, επομένως περιλαμβάνει 8 πλαίσια με 8 λέξεις το καθένα.
- Έτσι, από τη διεύθυνση των 8 bits, χρησιμοποιούνται 3 bits για τον καθορισμό της λέξης μέσα στο πλαίσιο, 3 bits για τον καθορισμό του πλαισίου και 8 – 3 – 3 = 2 bits για ετικέτα.

Διεύθυνση Μνήμης

Ετικέτα	Πλαίσιο	Λέξη
2 bits	3 bits	3 bits

Μπλοκ κύριας μνήμης: Μ (διεύθυνση πρώτης λέξης – διεύθυνση τελευταίας λέξης)

Διεύθυνση μνήμης	Ετικέτα-Πλαίσιο-Λέξη	Επιτυχία / Αποτυχία	Μπλοκ κύριας μνήμης → πλαίσιο λανθάνουσας μνήμης
16	00-010-110	Αποτυχία	Μ(10-17) → Π2
8C	10-001-100	Αποτυχία	Μ(88-8F) → Π1
14	00-010-100	Επιτυχία	Π2
03	00-000-011	Αποτυχία	Μ(0-7) → Π0
12	00-010-010	Επιτυχία	Π2
1B	00-011-011	Αποτυχία	Μ(18-1F) → Π3
90	10-010-000	Αποτυχία	Μ(90-97) → Π2, **αντικατάσταση**
8A	10-001-010	Επιτυχία	Π1

Λόγος επιτυχίας: 3/8, Ποσοστό επιτυχίας: 37.5%

Κρυφή μνήμη: παραδείγματα τρόπων απεικόνισης

- Αφού ο λόγος επιτυχίας για την κρυφή 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 αποτυχίας πρέπει από την κύρια μνήμη να μεταφερθεί ένα μπλοκ, 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.

Κρυφή μνήμη: παραδείγματα τρόπων απεικόνισης

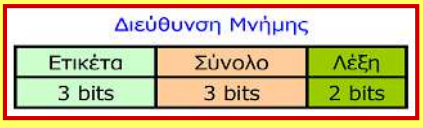
Παράδειγμα 2

- 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 κρυφή μνήμη με οργάνωση 2-τρόπων συνόλου συσχέτισης και χωρητικότητα 64 λέξεων και 4 λέξεις ανά πλαίσιο, θα υπολογίσουμε τον λόγο επιτυχίας και το συνολικό πλήθος των bytes που διαβάζονται από την κύρια μνήμη.
- Θεωρούμε ότι αρχικά η κρυφή μνήμη είναι κενή ή δεν περιέχει έγκυρα δεδομένα και ότι χρησιμοποιεί τον αλγόριθμο αντικατάστασης τμημάτων LRU.

Κρυφή μνήμη: παραδείγματα τρόπων απεικόνισης

- Η κρυφή μνήμη χρησιμοποιεί οργάνωση 2-τρόπων συνόλου συσχέτισης (2 πλαίσια ανά σύνολο) και έχει χωρητικότητα 64 λέξεων με 4 λέξεις ανά πλαίσιο, επομένως περιλαμβάνει 16 / 2 = 8 σύνολα με 2 πλαίσια το καθένα.
- Έτσι, από τη διεύθυνση των 8 bits, χρησιμοποιούνται 2 bits για τον καθορισμό της λέξης μέσα στο πλαίσιο, 3 bits για καθορισμό του συνόλου και 8 – 2 – 3 = 3 bits για ετικέτα.



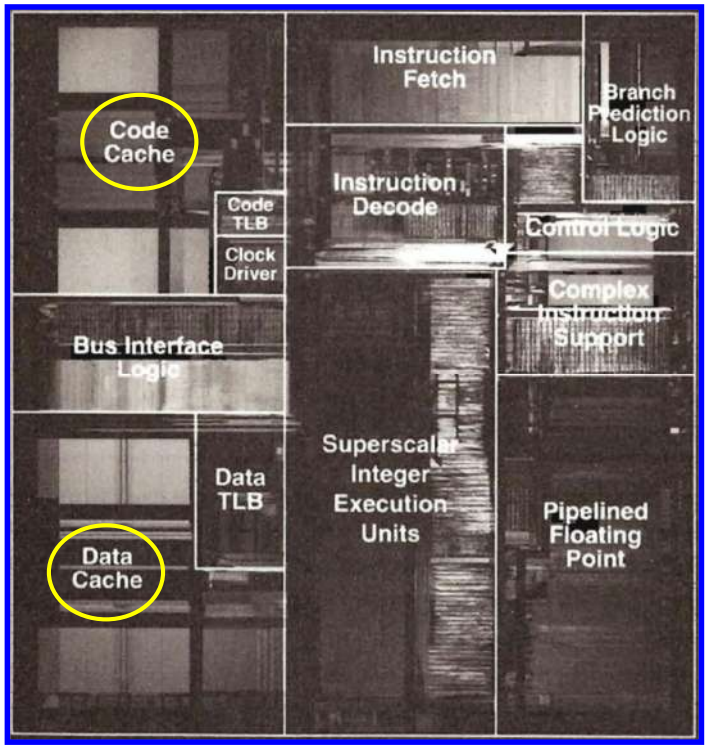
Διεύθυνση μνήμης	Ετικέτα – Σύνολο – Λέξη	Επιτυχία / Αποτυχία	Μπλοκ κύριας μνήμης → Σύνολο και πλαίσιο λανθάνουσας μνήμης
16	000-101-10	Αποτυχία	Μ(14-17) → (Σ5, Π0)
8C	100-011-00	Αποτυχία	Μ(8C-8F) → (Σ3, Π0)
14	000-101-00	Επιτυχία	(Σ5, Π0)
03	000-000-11	Αποτυχία	Μ(0-3) → (Σ0, Π0)
37	001-101-11	Αποτυχία	Μ(34-37) → (Σ5, Π1)
75	011-101-01	Αποτυχία	Μ(74-77) → (Σ5, Π0), **αντικατάσταση**
8E	100-011-10	Επιτυχία	(Σ3, Π0)
0C	000-011-00	Αποτυχία	Μ(0C-0F) → (Σ3, Π1)
2F	001-011-11	Αποτυχία	Μ(2C-2F) → (Σ3, Π0), **αντικατάσταση**

Λόγος επιτυχίας: 2/9, Ποσοστό επιτυχίας: 22.2%

Κρυφή μνήμη: παραδείγματα τρόπων απεικόνισης

- Αφού ο λόγος επιτυχίας για την κρυφή μνήμη με οργάνωση 2-τρόπων συνόλου συσχέτισης 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 μπλοκ, 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.

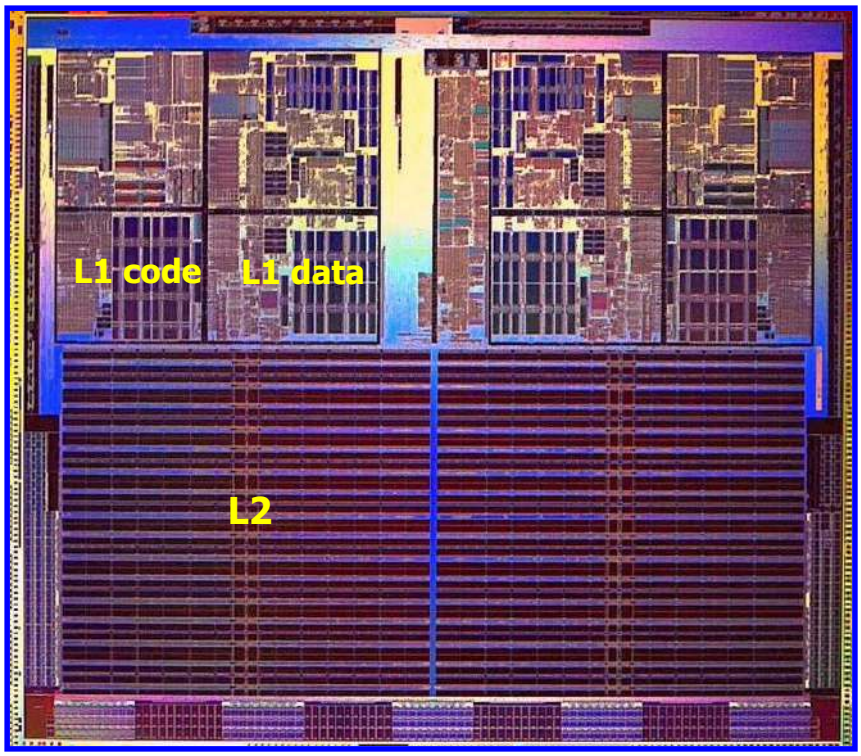
Λανθάνουσα μνήμη του Intel Pentium (1993)



8 Kbytes L1 δεδομένων και 8 Kbytes L1 εντολών με οργάνωση 2-τρόπων συνόλου συσχέτισης (2 πλαίσια ανά σύνολο) και αλγόριθμο αντικατάστασης τμημάτων LRU.

Χρησιμοποιεί δύο εναλλακτικές τακτικές ενημέρωσης της μνήμης ανώτερου επιπέδου: write-through και write-back

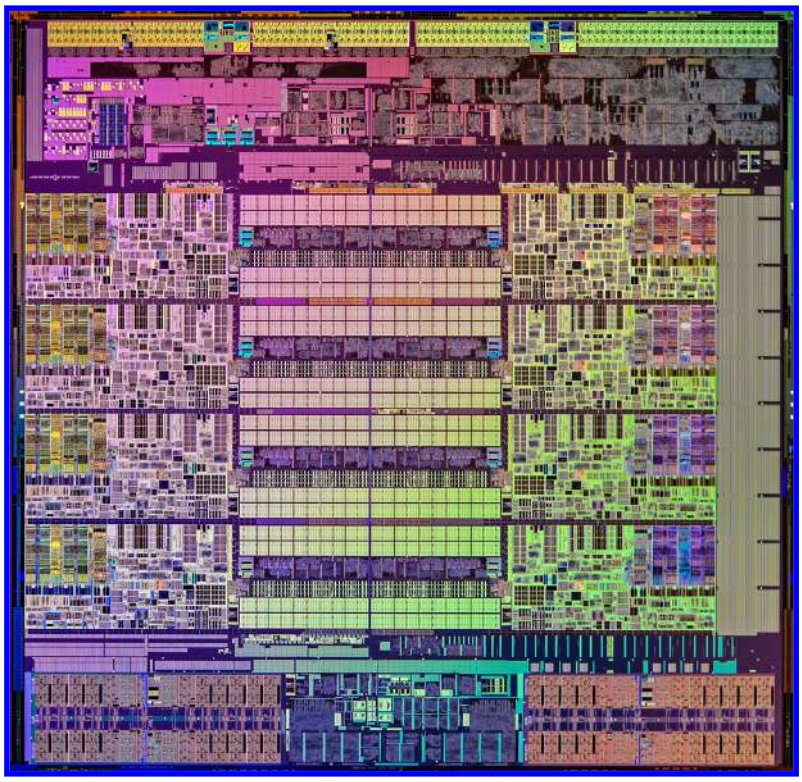
Λανθάνουσα μνήμη διπύρηνου AMD Athlon 64 (2005)



64 Kbytes L1 δεδομένων και 64 Kbytes L1 εντολών ανά πυρήνα με οργάνωση 2-τρόπων συνόλου συσχέτισης (2 πλαίσια ανά σύνολο).

512 Kbytes L2 ανά πυρήνα με οργάνωση 16-τρόπων συνόλου συσχέτισης (16 πλαίσια ανά σύνολο).

Λανθάνουσα μνήμη οκταπύρηνου Intel Core i7 (2014)



32 Kbytes L1 δεδομένων και 32 Kbytes L1 εντολών, ανά πυρήνα με οργάνωση 8-τρόπων συνόλου συσχέτισης (8 πλαίσια ανά σύνολο).

256 Kbytes L2 ανά πυρήνα με οργάνωσης 8 τρόπων συνόλου συσχέτισης (8 πλαίσια ανά σύνολο).

20 Mbytes L3, κοινή για τους 8 πυρήνες με οργάνωση 20-τρόπων συνολου συσχέτισης (20 πλαίσια ανά σύνολο).

Κύρια μνήμη

- Βασικό χαρακτηριστικό της κύριας μνήμης είναι ο ρυθμός μεταφοράς δεδομένων (data transfer rate, bandwidth), δηλαδή ο αριθμός δυαδικών ψηφίων που μπορούν να προσπελαστούν ανά δευτερόλεπτο.

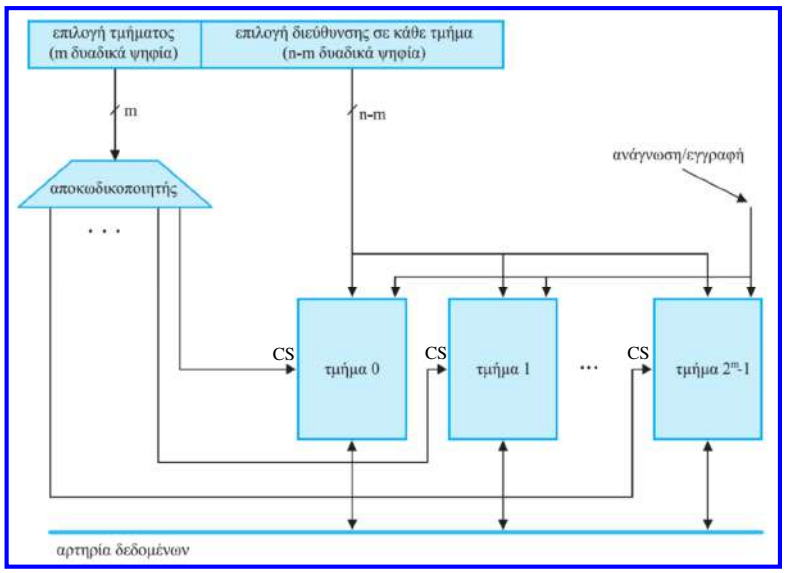
Παράδειγμα: εάν ο χρόνος κύκλου μνήμης είναι 50 ns και μπορούν να προσπελαστούν 32 ψηφία ανά κύκλο, τότε ο ρυθμός μεταφοράς δεδομένων είναι 32 bits / (50 x 10^{-9}) s = 640 x 10^3 bits / s = 640 Mbits / s.

- Οι κύριοι παράγοντες που επηρεάζουν το ρυθμό μεταφοράς είναι η διαμόρφωση μνήμης (memory configuration) και τα χαρακτηριστικά των ολοκληρωμένων κυκλωμάτων (τμημάτων) που την απαρτίζουν.

- Η διαμόρφωση μνήμης χαρακτηρίζεται από τον αριθμό των τμημάτων της, τον τρόπο κατανομής διευθύνσεων μεταξύ αυτών και το εύρος αρτηρίας δεδομένων.
- Βασικά χαρακτηριστικά των τμημάτων μνήμης είναι το εύρος τους, ο χρόνος προσπέλασης (χρόνος από αίτημα προσπέλασης μέχρι παραλαβή περιεχομένου) και ο χρόνος κύκλου μνήμης (χρόνος μεταξύ δύο διαδοχικών προσπελάσεων).
- Δύο τρόποι κατανομής διευθύνσεων στα τμήματα μνήμης:
  - ✓ οργάνωση υψηλής τάξης διαφύλλωσης (high-order interleaving) και
  - ✓ οργάνωση N-τρόπων χαμηλής τάξης διαφύλλωσης (N-way low-order interleaving)

Υψηλής τάξης διαφύλλωση μνήμης

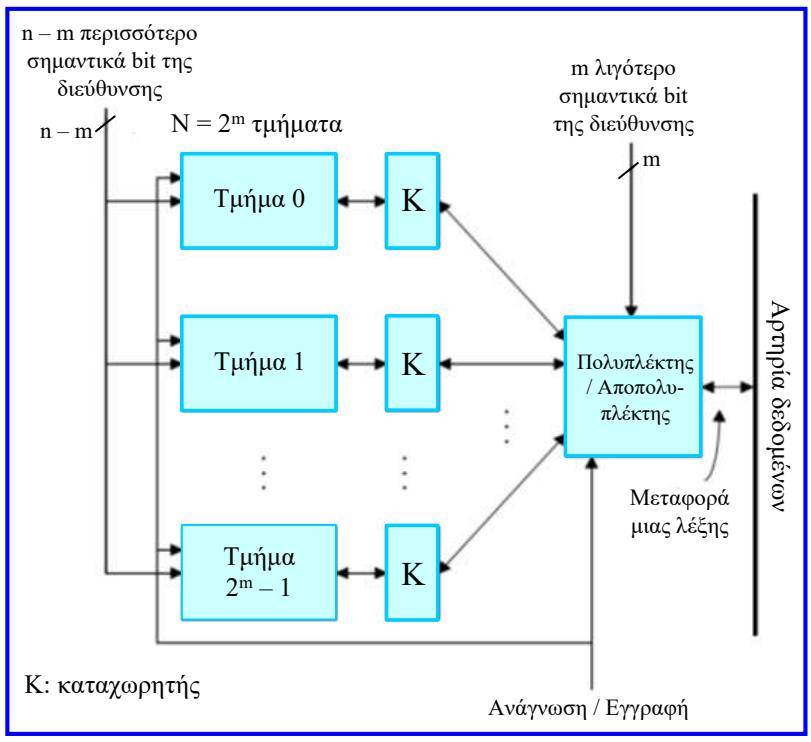
- Χωρητικότητα κύριας μνήμης: N = 2^n λέξεις
- Διεύθυνση: n δυαδικά ψηφία
- Τμήματα μνήμης: M = 2^m
- Διευθύνσεις τμήματος: 2^n / 2^m = 2^{n-m}
- Το 1ο τμήμα περιέχει τις διευθύνσεις από 0 έως 2^{n-m} –1, το 2ο τμήμα τις διευθύνσεις από 2^{n-m} έως 2x2^{n-m} –1, κ.ο.κ.
- Γενικά, το τμήμα i περιέχει τις διευθύνσεις από ix2^{n-m} μέχρι (i+1)x2^{n-m} –1.



- Η κατανομή αυτή συνεπάγεται ότι τα m περισσότερο σημαντικά ψηφία της διεύθυνσης χρησιμοποιούνται για την επιλογή τμήματος μνήμης, ενώ τα υπόλοιπα (n – m) λιγότερο σημαντικά ψηφία της διεύθυνσης χρησιμοποιούνται για την επιλογή της διεύθυνσης μέσα στο τμήμα.
- Πλεονέκτημα: εύκολη επεκτασιμότητα μνήμης με προσθήκη τμημάτων.

N-τρόπων χαμηλής τάξης διαφύλλωση μνήμης

- Οι διευθύνσεις κατανέμονται έτσι ώστε διαδοχικές διευθύνσεις να ανήκουν σε διαδοχικά τμήματα.
- Η κατανομή αυτή συνεπάγεται ότι τα m λιγότερο σημαντικά ψηφία της διεύθυνσης επιλέγουν το τμήμα μνήμης, από το σύνολο των N = 2^m τμημάτων.
- Τα υπόλοιπα n – m περισσότερο σημαντικά ψηφία επιλέγουν τη διεύθυνση μέσα σε κάθε τμήμα.
- Πλεονέκτημα: τα περιεχόμενα N = 2^m θέσεων μνήμης με διαδοχικές διευθύνσεις μπορούν να προσπελαστούν ταυτόχρονα.



Παράδειγμα σχεδιασμού κύριας μνήμης

Σχεδιασμός κύριας μνήμης με χωρητικότητα 64 Kbytes

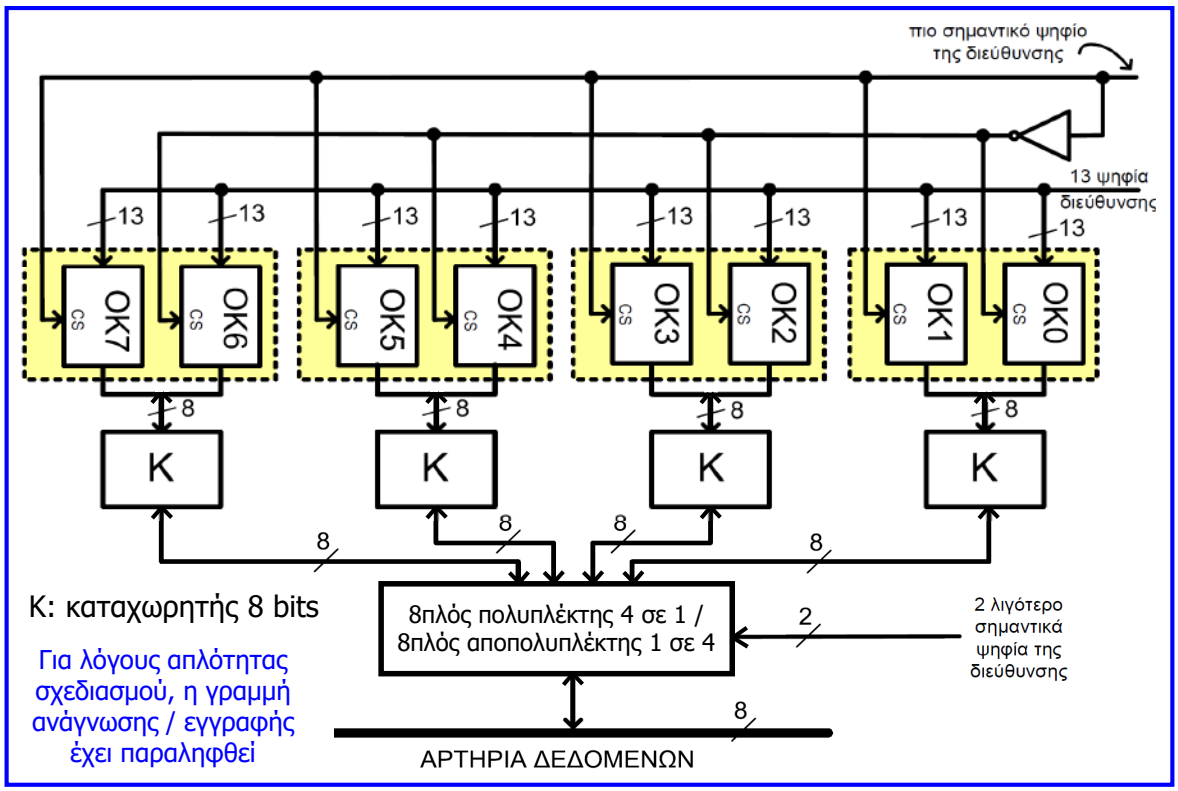
Διαθέσιμη μνήμη: ολοκληρωμένα κυκλώματα (ΟΚ) με χωρητικότητα 8 Kbytes το καθένα

Οργάνωση μνήμης: 4-τρόπων χαμηλής τάξης διαφύλλωση

Μήκος λέξεων: 8 δυαδικά ψηφία (1 byte)

- Αφού η χωρητικότητα της ζητούμενης μνήμης είναι 64 Kbytes = 64 x 1024 = 2^6 x 2^{10} = 2^{16} bytes και κάθε λέξη έχει μήκος 8 bits = 1 byte, το μήκος διευθύνσεων μνήμης είναι 16 bits.
- Για το σχεδιασμό μνήμης 64 Kbytes με οργάνωση 4-τρόπων χαμηλής τάξης διαφύλλωσης απαιτούνται N = 4 τμήματα μνήμης με χωρητικότητα 64 / 4 = 16 Kbytes το καθένα.
- Επειδή τα διαθέσιμα ΟΚ έχουν χωρητικότητα 8 Kbytes το καθένα, θα χρησιμοποιήσουμε 8 από αυτά, 2 για κάθε τμήμα της κύριας μνήμης.
- Αφού N = 2^m και N = 4, δηλαδή m = 2, συνεπάγεται ότι τα 2 λιγότερο σημαντικά ψηφία της διεύθυνσης, επιλέγουν το τμήμα της μνήμης.
- Τα υπόλοιπα 16 – 2 = 14 ψηφία της διεύθυνσης επιλέγουν τη διεύθυνση μέσα σε κάθε τμήμα.
- Επειδή όμως κάθε τμήμα αποτελείται από 2 ΟΚ, το πιο σημαντικό ψηφίο της διεύθυνσης επιλέγει το ένα από τα 2 ΟΚ μέσα σε κάθε τμήμα, ενώ τα υπόλοιπα 13 ψηφία επιλέγουν τη διεύθυνση μέσα σε κάθε ΟΚ.

Παράδειγμα σχεδιασμού κύριας μνήμης



Αρτηρίες του υπολογιστή

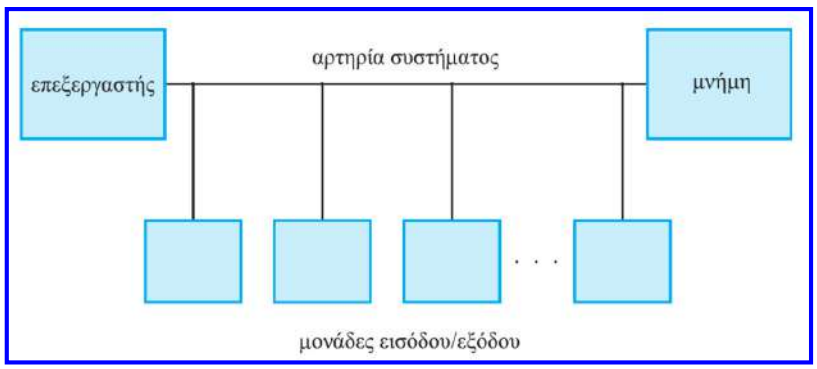
- Για να είναι δυνατή η εκτέλεση προγραμμάτων και η επεξεργασία της πληροφορίας, οι μανάδες του υπολογιστή πρέπει να επικοινωνούν μεταξύ τους (π.χ. επικοινωνία ΚΜΕ με μνήμη και μονάδες εισόδου / εξόδου).
- Η επικοινωνία αυτή γίνεται μέσω των αρτηριών.
- Αρτηρία (bus) είναι το σύνολο των γραμμών διασύνδεσης που χρησιμοποιούνται για την επικοινωνία δύο ή περισσότερων μονάδων, η οποία διέπεται από σύνολο κανόνων που αναφέρεται ως πρωτόκολλο (protocol) αρτηρίας.
- Οι γραμμές διασύνδεσης περιλαμβάνουν:
  - ✓ γραμμές ελέγχου για να δηλώνεται η απαίτηση επικοινωνίας, η αναγνώριση απαίτησης επικοινωνίας και το είδος πληροφορίας στις γραμμές δεδομένων,
  - ✓ γραμμές δεδομένων που μεταφέρουν δεδομένα, εντολές και διευθύνσεις
- Πιο συνηθισμένος τρόπος περιγραφής πρωτοκόλλου αρτηρίας είναι τα χρονικά διαγράμματα των σημάτων ελέγχου και των γραμμών δεδομένων και διευθύνσεων, με βάση τα οποία διενεργείται η επικοινωνία των μονάδων του υπολογιστή.

Αρτηρίες αποκλειστικής χρήσης

- Αρτηρία αποκλειστικής χρήσης (dedicated bus, point-to-point) είναι αυτή που χρησιμοποιείται αποκλειστικά για να μεταφέρει πληροφορία μεταξύ δύο μόνο μονάδων.
- Για την επικοινωνία n μονάδων, ανά δύο, με όλους τους δυνατούς τρόπους, απαιτούνται n x (n-1) / 2 αρτηρίες αποκλειστικής χρήσης.
- Όταν οι μονάδες συνδέονται με αρτηρίες αποκλειστικής χρήσης περιορίζονται οι καθυστερήσεις επικοινωνίας, αφού η αρτηρία αποκλειστικής χρήσης με την οποία συνδέεται ένα ζεύγος μονάδων δεν είναι ποτέ απασχολημένη από άλλες μονάδες.
- Ωστόσο, το κύριο μειονέκτημα είναι το υψηλό κόστος υλοποίησης και αποτελεί την αιτία που αυτός ο τρόπος διασύνδεσης συνήθως δεν χρησιμοποιείται.

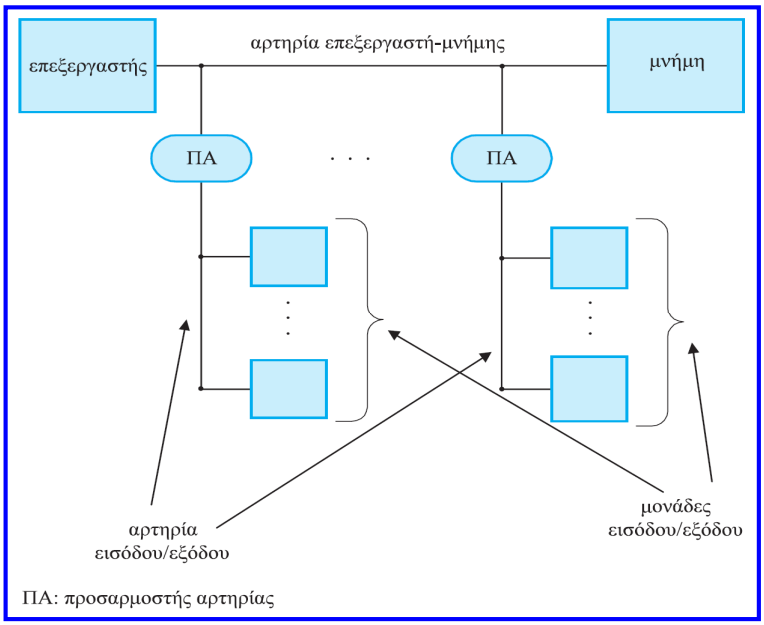
Αρτηρίες κοινής χρήσης

- Αρτηρία κοινής χρήσης (shared bus) είναι αυτή με την οποία συνδέονται όλες οι μονάδες του υπολογιστικού συστήματος.
- Κάθε χρονική στιγμή μόνο δύο μονάδες μπορούν να επικοινωνούν μεταξύ τους.
- Επομένως, απαιτείται μηχανισμός ελέγχου και διαιτησίας ώστε να επιβλέπεται το χρονικό μοίρασμα της χρήσης της αρτηρίας μεταξύ των μονάδων.
- Οι αρτηρίες κοινής χρήσης χρησιμοποιούνται ευρέως στα υπολογιστικά συστήματα.
- Πλεονεκτήματα: χαμηλό κόστος υλοποίησης, ευκολία σύνδεσης νέων μονάδων στο σύστημα.
- Μειονεκτήματα: καθυστερήσεις στην επικοινωνία, ευαισθησία σε βλάβες.



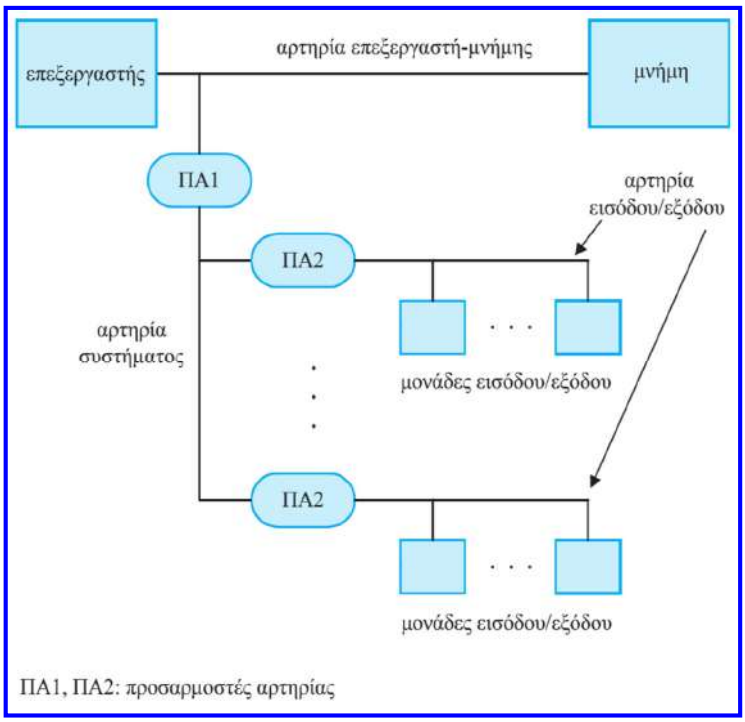
Αρτηρίες κοινής χρήσης

- Οι δομές επικοινωνίας των μονάδων σε αρκετά υπολογιστικά συστήματα, χρησιμοποιούν περισσότερες από μία αρτηρίες κοινής χρήσης.
- Με τις δομές αυτές είναι δυνατή την ίδια χρονική στιγμή η επικοινωνία περισσοτέρων του ενός ζευγών μονάδων



Αρτηρίες κοινής χρήσης

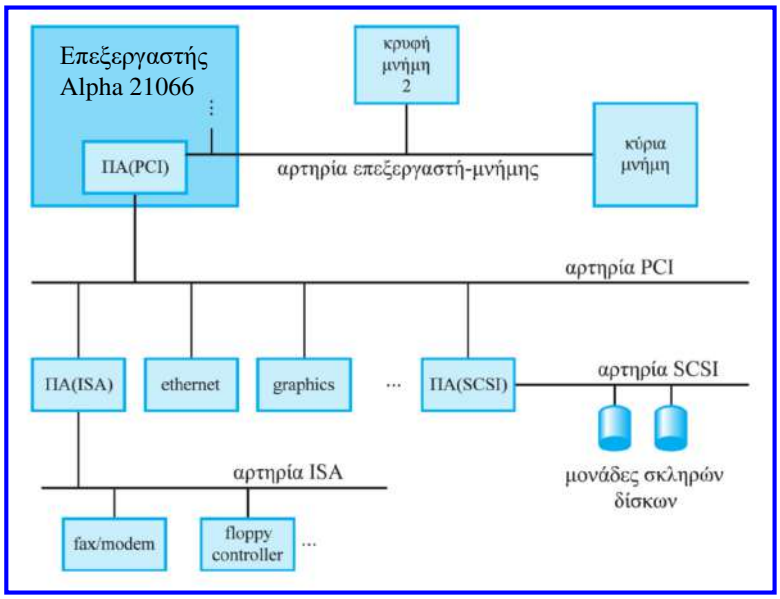
- Αρτηρία επεξεργαστή-μνήμης:
  - ✓ Μικρού μήκους.
  - ✓ Υψηλής ταχύτητας.
  - ✓ Ταιριάζει με τα χαρακτηριστικά επεξεργαστή και μνήμης με σκοπό την μεγιστοποίηση του ρυθμού μεταφοράς πληροφορίας μεταξύ επεξεργαστή και μνήμης.
- Αρτηρία συστήματος: συνδέει την αρτηρία επεξεργαστή-μνήμης με τις αρτηρίες εισόδου/εξόδου.
- Αρτηρίες εισόδου-εξόδου:
  - ✓ Μεγαλύτερου μήκους.
  - ✓ Μπορούν να συνδεθούν διάφορα είδη μονάδων εισόδου/εξόδου.



Παραδείγματα αρτηριών κοινής χρήσης

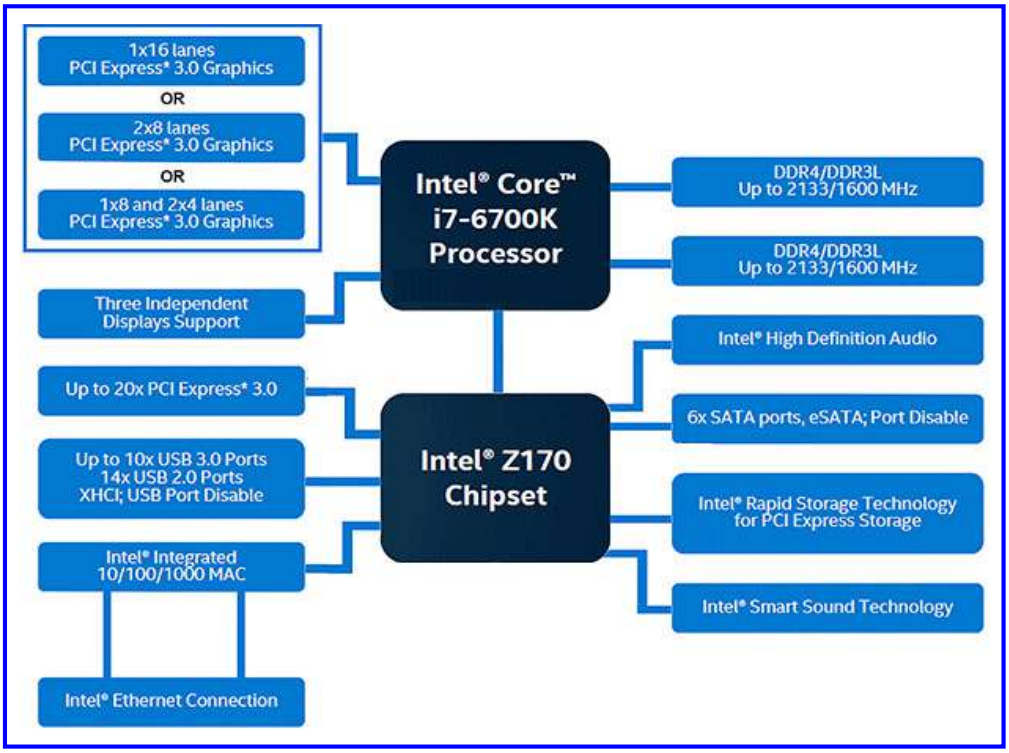
- Αρτηρίες συστήματος: Peripheral Component Interconnect (PCI).
- Αρτηρίες εισόδου/εξόδου: Industry Standard Architecture (ISA), Extended ISA (EISA), Micro-Channel, EIDE, SCSI.

Τυπικό υπολογιστικό σύστημα με αρτηρίες κοινής χρήσης βασισμένο στον επεξεργαστή Alpha 21066 (Digital Equipment, 1994)



Παραδείγματα αρτηριών κοινής χρήσης

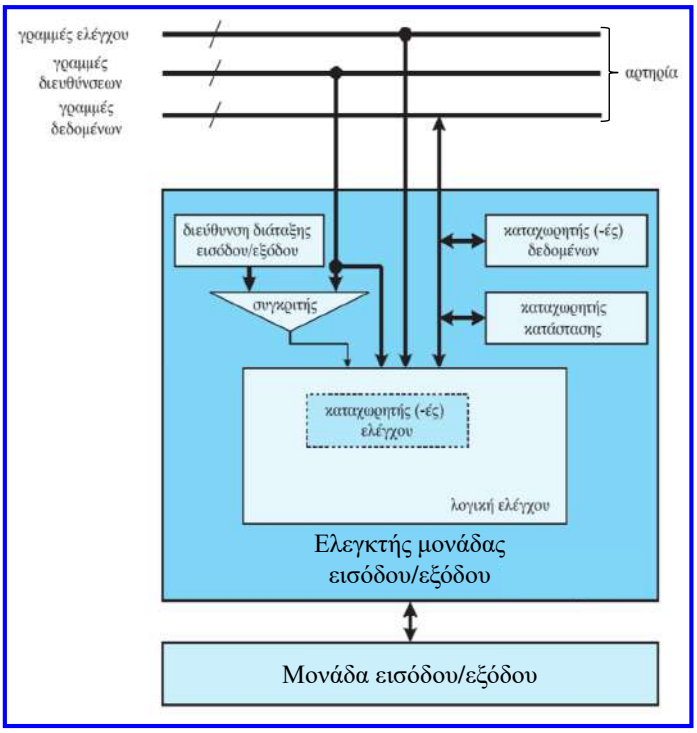
Υπολογιστικό σύστημα με αρτηρίες κοινής χρήσης βασισμένο στον επεξεργαστή i7 (Intel, 2015)



Ελεγκτές μονάδων εισόδου/εξόδου

Σκοπός του ελεγκτή μονάδας εισόδου/εξόδου είναι:

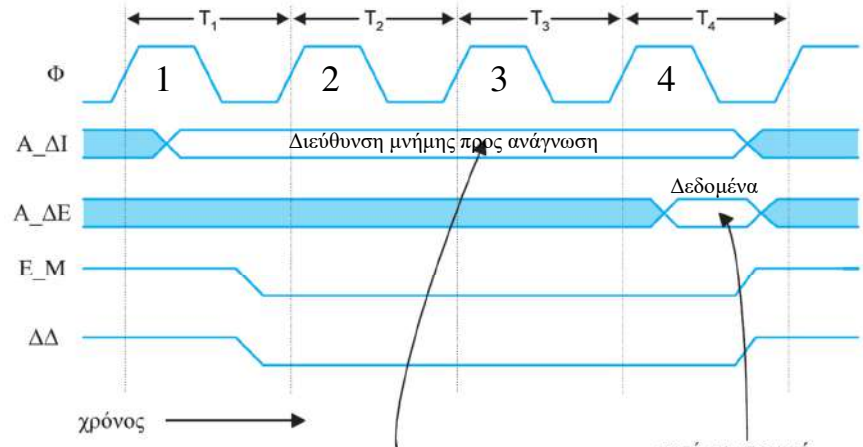
- να αποδεσμεύσει την ΚΜΕ και την κύρια μνήμη από τις λεπτομέρειες που αφορούν τη λειτουργία των μονάδων εισόδου/εξόδου,
- να αποδεσμεύσει την αρτηρία από τις απαιτήσεις διασύνδεσης της συγκεκριμένης μονάδας εισόδου εξόδου,
- να διαθέσει χώρο ενδιάμεσης αποθήκευσης πληροφορίας που αφορά την επικοινωνία με τη μονάδα εισόδου/εξόδου,
- να μετατρέπει τη μορφή ή την κωδικοποίηση των δεδομένων της μονάδας εισόδου/εξόδου, ώστε αυτή να συμμορφώνεται με το πρωτόκολλο της αρτηρίας στην οποία συνδέεται.



Σύγχρονες αρτηρίες

Στις σύγχρονες αρτηρίες μία από τις γραμμές ελέγχου χρησιμοποιείται για τη μετάδοση ενός περιοδικού σήματος παλμών (σήμα χρονισμού ή ρολόι αρτηρίας), στο οποίο βασίζεται το πρωτόκολλο επικοινωνίας, δηλ. όλες οι ενέργειες γίνονται σε σχέση με αυτό το σήμα.

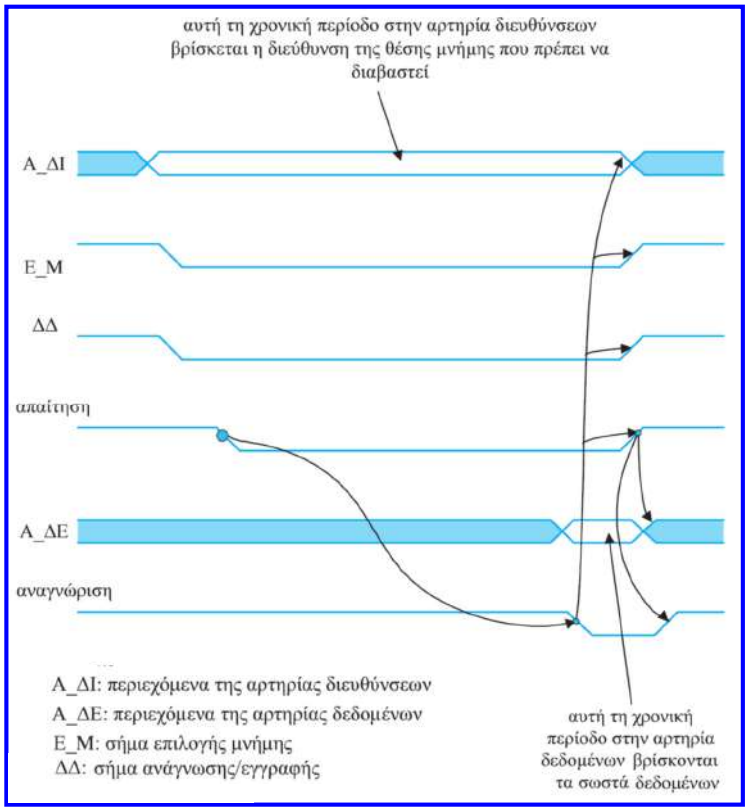
Ανάγνωση δεδομένων από τη μνήμη (προσπέλαση της μνήμης σε 4 κύκλους του σήματος χρονισμού)



A_ΔΙ: περιεχόμενα της αρτηρίας διευθύνσεων
A_ΔΕ: περιεχόμενα της αρτηρίας δεδομένων
E_M: σήμα επιλογής μνήμης
ΔΔ: σήμα ανάγνωσης/εγγραφής

Ασύγχρονες αρτηρίες

- Στις ασύγχρονες αρτηρίες, ο συγχρονισμός μεταξύ της μονάδας-πομπού και της μονάδας-δέκτη επιτυγχάνεται με τη χρήση ενός πρωτοκόλλου χειραψίας (handshaking protocol).
- Οι μονάδες που επικοινωνούν χρησιμοποιούν τα σήματα απαίτησης (request) και αναγνώρισης (acknowledge).
- Η απενεργοποίηση της απαίτησης από τη μία μονάδα γίνεται μετά την ενεργοποίηση της αναγνώρισης από την άλλη.
- Η απενεργοποίηση της αναγνώρισης από τη μία μονάδα γίνεται μετά την απενεργοποίηση της απαίτησης από την άλλη.



Απόδοση αρτηρίας

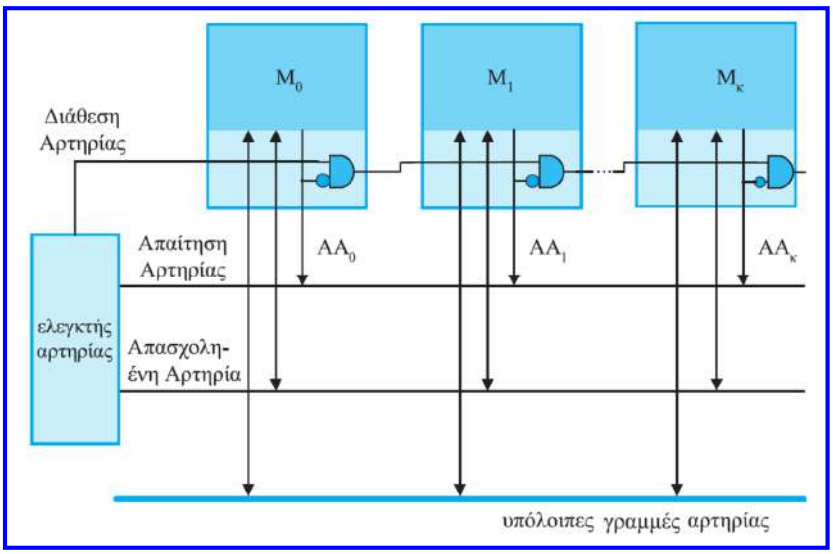
- Η μέγιστη ταχύτητα μιας αρτηρίας περιορίζεται από το μήκος της αρτηρίας και το πλήθος των μονάδων που είναι συνδεδεμένες σε αυτή.
- Δύο μέτρα απόδοσης αρτηρίας:
  - ✓ καθυστέρηση (latency) με την οποία μεταφέρονται τα δεδομένα,
  - ✓ ρυθμός μεταφοράς (transfer rate) των δεδομένων, ο οποίος εξαρτάται από:
    - το εύρος της αρτηρίας,
    - την ύπαρξη διακριτών γραμμών δεδομένων και διευθύνσεων και
    - την δυνατότητα μεταφοράς συνόλου λέξεων, χωρίς τη μεσολάβηση αποστολής διεύθυνσης για την αποστολή κάθε λέξης.
- Οι παράγοντες που αυξάνουν το ρυθμό μεταφοράς της αρτηρίας (δηλαδή, αύξηση του εύρους, διακριτές γραμμές διευθύνσεων και δεδομένων, δυνατότητα αποστολής συνόλου λέξεων), αυξάνουν και το κόστος υλοποίησης της αρτηρίας.

Διαιτησία αρτηρίας

- Όταν μια μονάδα που συνδέεται σε μια αρτηρία επιθυμεί να την χρησιμοποιήσει, θα πρέπει να ανιχνεύσει αν η αρτηρία είναι ελεύθερη (δηλαδή δεν χρησιμοποιείται ήδη από άλλες μονάδες).
- Επίσης, όταν δύο μονάδες αποφασίσουν την ίδια χρονική στιγμή να χρησιμοποιήσουν την αρτηρία, τότε καθεμία από αυτές θα ανιχνεύσει την αρτηρία ελεύθερη και θα προσπαθήσει να διαβιβάσει μέσω της αρτηρίας τα σήματα που επιθυμεί, με αποτέλεσμα τα σήματα των δύο μονάδων να αναμειχθούν.
- Ο μηχανισμός που καθορίζει ποια μονάδα του υπολογιστικού συστήματος θα αποκτήσει τον έλεγχο της αρτηρίας, αναφέρεται ως διαιτησία (arbitration) αρτηρίας.
- Σχήματα διαιτησίας:
  - ✓ διαιτησία με χρήση αλυσίδας (daisy chain arbitration),
  - ✓ κεντρική παράλληλη διαιτησία (centralized parallel arbitration) και
  - ✓ κατανεμημένη διαιτησία (distributed arbitration).

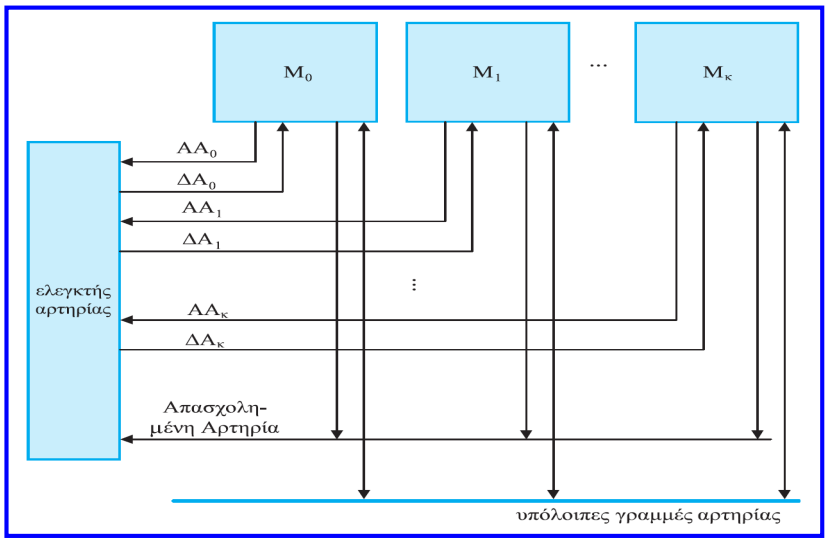
Διαιτησία με χρήση αλυσίδας

- Η γραμμή διάθεσης της αλυσίδας περνάει διαμέσου όλων των μονάδων κατά σειρά προτεραιότητας, η οποία καθορίζεται από τη θέση της μονάδας (η M_0 προηγείται).
- Μια μονάδα υψηλής προτεραιότητας που δέχεται την διάθεση αρτηρίας, ενεργοποιώντας την απαίτηση αρτηρίας δεν της επιτρέπει να διαδοθεί στις υπόλοιπες μονάδες.
- Πλεονεκτεί λόγω απλότητας σχεδιασμού, μειονεκτεί λόγω έλλειψης δικαιοσύνης.



Κεντρική παράλληλη διαιτησία

- Κάθε μονάδα διαθέτει τα δικά της σήματα απαίτησης αρτηρίας και διάθεσης αρτηρίας.
- Ο ελεγκτής αρτηρίας επιλέγει μία από τις μονάδες που απαιτούν την αρτηρία και της δίνει την κυριότητα της αρτηρίας όταν αυτή είναι ελεύθερη.
- Για την επιλογή, ο ελεγκτής ακολουθεί σταθερή ή εναλλασσόμενη προτεραιότητα (rotating priority) κατά την οποία μπορεί να αλλάξει την προτεραιότητα των μονάδων, ανάλογα με το ιστορικό εξυπηρέτησής τους.



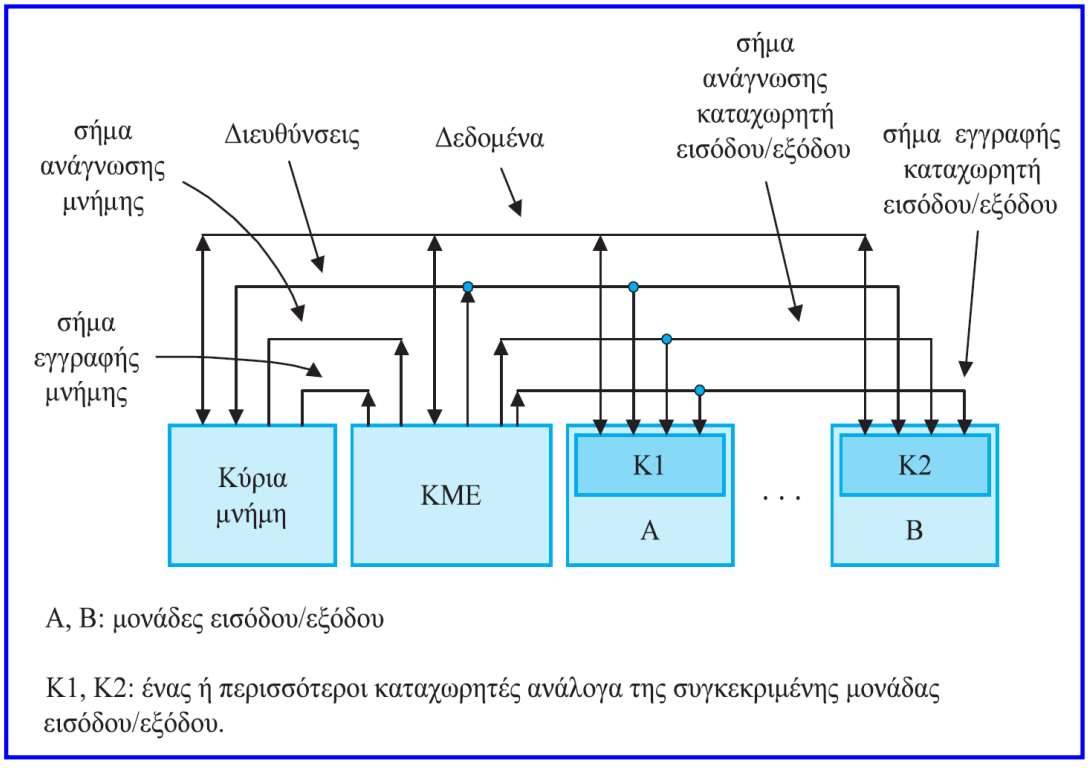
Κατανεμημένη διαιτησία

- Στην κατανεμημένη διαιτησία με αυτοεπιλογή (self-selection), η αρτηρία διαθέτει γραμμές διαιτησίας.
- Σε κάθε μονάδα αντιστοιχεί ένας αριθμός που δηλώνει την προτεραιότητά της.
- Κάθε μονάδα που απαιτεί τη χρήση της αρτηρίας, θέτει τον αριθμό προτεραιότητάς της στις γραμμές διαιτησίας της αρτηρίας.
- Εάν έχει τεθεί αριθμός μεγαλύτερης προτεραιότητας, η μονάδα αποσύρει τον αριθμό της.
- Μετά από κάποιο χρονικό διάστημα στις γραμμές διαιτησίας υπάρχει μόνο ο αριθμός της μονάδας με τη μεγαλύτερη προτεραιότητα μεταξύ αυτών που απαιτούσαν την αρτηρία.
- Η μονάδα που ανιχνεύει τον αριθμό προτεραιότητάς της στις γραμμές διαιτησίας μπορεί να χρησιμοποιήσει την αρτηρία.
- Στην κατανεμημένη διαιτησία με ανίχνευση σύγκρουσης (collision detection), κάθε μονάδα που επιθυμεί να χρησιμοποιήσει την αρτηρία γράφει σε αυτή, ενώ ταυτόχρονα διαβάζει την αρτηρία για να διαπιστώσει αν η πληροφορία έχει διαφοροποιηθεί (δηλαδή εάν συνέβη σύγκρουση με εγγραφή από άλλη μονάδα,
- Στην περίπτωση σύγκρουσης, οι μονάδες ξαναπροσπαθούν μετά από τυχαία χρονικά διαστήματα.
- Οι μονάδες που λαμβάνουν τη πληροφορία που είναι αποτέλεσμα σύγκρουσης, ανιχνεύουν τη σύγκρουση και αγνοούν την πληροφορία.

Διακίνηση πληροφορίας μεταξύ ΚΜΕ & μονάδων Ε/Ε

- Η διακίνηση πληροφορίας μεταξύ ΚΜΕ και μονάδων εισόδου/εξόδου (Ε/Ε), συνίσταται στην αποστολή εντολών από την ΚΜΕ στις μονάδες Ε/Ε, και στην υλοποίηση των λειτουργιών εισόδου/εξόδου.
- Δύο προσεγγίσεις χρησιμοποιούνται για την αποστολή εντολών από την ΚΜΕ στις μονάδες Ε/Ε.
- Στην πρώτη προσέγγιση, το σύνολο εντολών της ΚΜΕ διαθέτει ειδικές εντολές εισόδου/εξόδου και χρησιμοποιώντας αυτές, η ΚΜΕ μπορεί να γράψει στους καταχωρητές του ελεγκτή μιας μονάδας Ε/Ε εντολές και δεδομένα.
- Στην περίπτωση αυτή ο χώρος διευθύνσεων των μονάδων Ε/Ε και της κύριας μνήμης είναι διαφορετικοί.
- Στην δεύτερη προσέγγιση, ένα μέρος των διευθύνσεων μνήμης συνδέεται με καταχωρητές του ελεγκτή της μονάδας Ε/Ε (memory mapped I/O).
- Στην περίπτωση αυτή, οι εντολές που χρησιμοποιούνται για ανάγνωση ή εγγραφή σε θέση κύριας μνήμης, χρησιμοποιούνται και για τις αντίστοιχες λειτουργίες στους καταχωρητές της μονάδας Ε/Ε, γράφοντας την εντολή στον καταχωρητή ελέγχου του ελεγκτή της μονάδας, στον οποίο αντιστοιχεί μια διεύθυνση κύριας μνήμης.

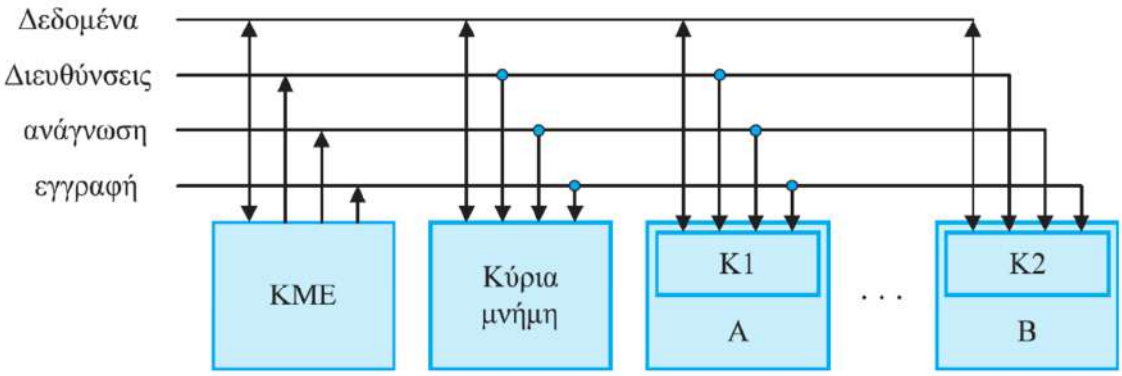
Διακριτοί χώροι διευθύνσεων μνήμης & μονάδων Ε/Ε



A, B: μονάδες εισόδου/εξόδου

K1, K2: ένας ή περισσότεροι καταχωρητές ανάλογα της συγκεκριμένης μονάδας εισόδου/εξόδου.

Ενιαίος χώρος διευθύνσεων μνήμης & μονάδων Ε/Ε



A, B: μονάδες εισόδου/εξόδου

K1, K2: ένας ή περισσότεροι καταχωρητές ανάλογα της συγκεκριμένης μονάδας εισόδου/εξόδου.

Παράδειγμα

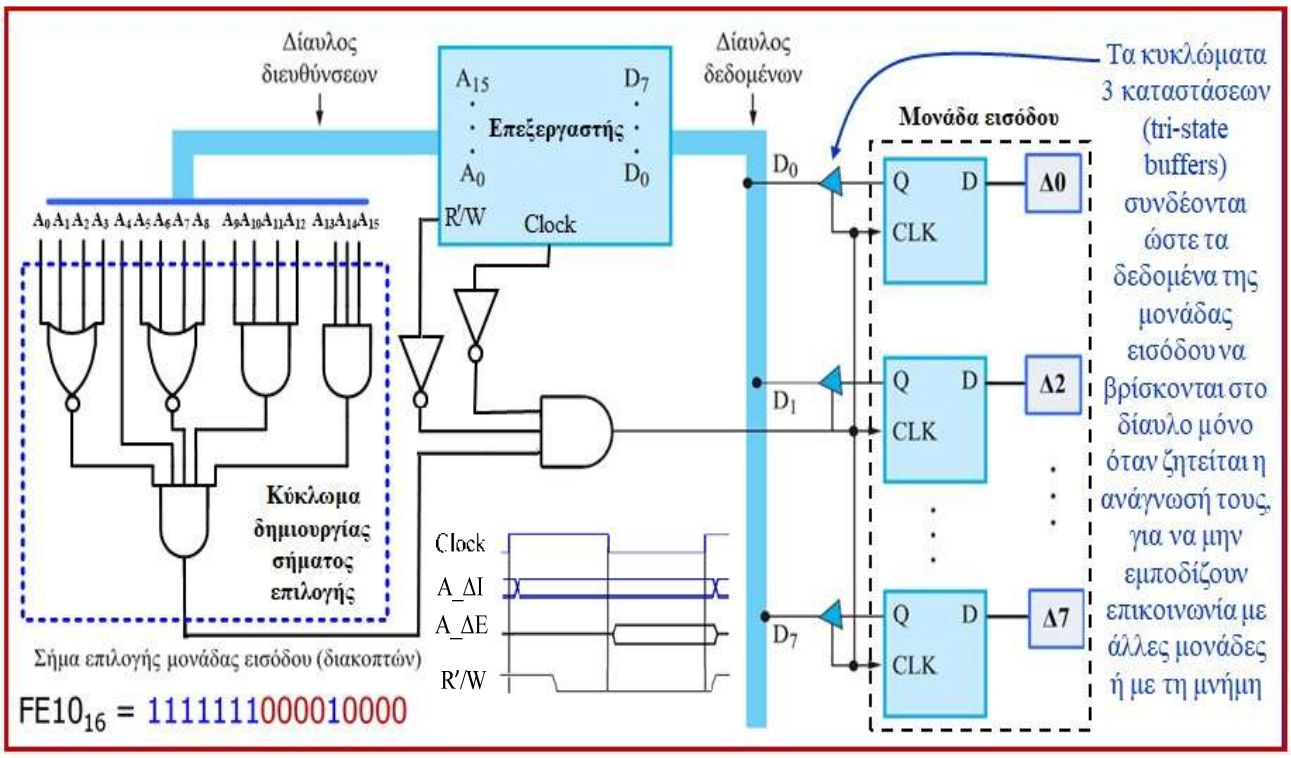
Ένα υπολογιστικό σύστημα με χώρο διευθύνσεων μνήμης 2^{16} λέξεων των 8 δυαδικών ψηφίων χρησιμοποιεί την προσέγγιση «memory-mapped I/O», δηλαδή μέρος των διευθύνσεων μνήμης αξιοποιείται για την προσπέλαση μονάδων εισόδου-εξόδου.

Επιθυμούμε να σχεδιάσουμε κύκλωμα διασύνδεσης για την ανάγνωση της κατάστασης (0 ή 1) οκτώ διακοπτών, με τη μορφή ενός byte στη διεύθυνση $FE10_{16}$.

Το υπολογιστικό σύστημα διαθέτει σύγχρονη αρτηρία ενός κύκλου ρολογιού.

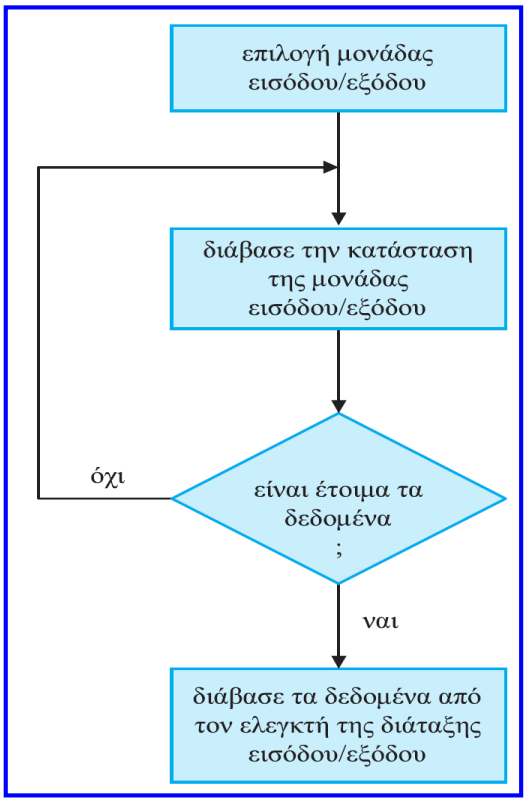
- Κάθε διεύθυνση αποτελείται από 16 bit, και η διεύθυνση $FE10_{16}$ στο δυαδικό σύστημα έχει ως εξής: 1111111000010000.
- Θα πρέπει λοιπόν να σχεδιαστεί κατάλληλο συνδυαστικό κύκλωμα που να ενεργοποιεί την ανάγνωση της κατάστασης των οκτώ διακοπτών, όταν ζητηθεί πρόσβαση στην διεύθυνση αυτή.
- Το κύκλωμα θα πρέπει να περιλαμβάνει ένα υποκύκλωμα αποκωδικοποίησης διεύθυνσης και όταν μέσω αυτού του υποκυκλώματος ενεργοποιηθεί η ανάγνωση των δεδομένων της μονάδας εισόδου (διακοπτών), τότε η πληροφορία εισόδου (κατάσταση διακοπτών) οδηγείται στις γραμμές της αρτηρίας δεδομένων του υπολογιστικού συστήματος.

Παράδειγμα



Διακίνηση πληροφορίας μεταξύ ΚΜΕ & μονάδων Ε/Ε

- Τρεις βασικές προσεγγίσεις χρησιμοποιούνται για την υλοποίηση λειτουργιών εισόδου/εξόδου.
- Στην προγραμματισμένη διαδικασία εισόδου/εξόδου, όταν η ΚΜΕ θέλει να διαβάσει δεδομένα από μία μονάδα Ε/Ε, αρχικά στέλνει το σχετικό αίτημα στον αντίστοιχο ελεγκτή.
- Η ΚΜΕ μέσω ενός βρόχου επανάληψης διαβάζει περιοδικά τον καταχωρητή κατάστασης του ελεγκτή μιας μονάδας Ε/Ε για να προσδιορίσει την κατάστασή της.
- Η διαδικασία αυτή αναφέρεται ως χρονοπρογραμματιζόμενος έλεγχος (polling) της μονάδας Ε/Ε από την ΚΜΕ.
- Η ΚΜΕ αναμένει και όταν διαπιστώσει ότι τα δεδομένα της μονάδας είναι έτοιμα προς μεταφορά και μια μονάδα δεδομένων έχει μεταφερθεί στον καταχωρητή δεδομένων του ελεγκτή της μονάδας, ξεκινά την μεταφορά των δεδομένων σε έναν καταχωρητή της και κατόπιν στην κύρια μνήμη.



Παράδειγμα

- Θεωρήστε υπολογιστή με επεξεργαστή συχνότητας λειτουργίας 1 GHz, ο οποίος, εκτός των άλλων, χρησιμοποιείται για τον έλεγχο λειτουργίας 1000 μονάδων Ε/Ε μέσω προγραμματιζόμενης διαδικασίας Ε/Ε με χρονοπρογραμματισμό. 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.

Διακίνηση πληροφορίας μεταξύ ΚΜΕ & μονάδων Ε/Ε

- Βασικό μειονέκτημα της προηγούμενης μεθόδου είναι ότι κατά τη διαδικασία του ελέγχου των μονάδων Ε/Ε (polling), η ΚΜΕ δεν εκτελεί άλλες χρήσιμες λειτουργίες.
- Μια προσέγγιση ώστε να αυξηθεί η απόδοση του συστήματος, είναι η διαδικασία εισόδου/εξόδου με χρήση σημάτων διακοπής (interrupts).
- Η μονάδα Ε/Ε για να απαιτήσει εξυπηρέτηση στέλνει μέσω της αρτηρίας ένα σήμα διακοπής προς την ΚΜΕ (όταν είναι έτοιμη για μεταφορά δεδομένων) και η ΚΜΕ το αναγνωρίζει με αποστολή σήματος επιβεβαίωσης διακοπής (interrupt acknowledge).
- Τότε η ΚΜΕ σταματάει την εκτέλεση του τρέχοντος προγράμματος (αφού ολοκληρώσει την εκτελούμενη εντολή) για να εκτελέσει ένα υποπρόγραμμα (υποπρόγραμμα εξυπηρέτησης διακοπής, interrupt service routine, ISR), το οποίο εξυπηρετεί τη συγκεκριμένη απαίτηση διακοπής (π.χ. υποπρόγραμμα ή υπορουτίνα print).
- Η ΚΜΕ αποθηκεύει στην κύρια μνήμη το περιεχόμενο των καταχωρητών που χρησιμοποιούνταν πριν την εκτέλεση του ISR.
- Ο χρόνος που απαιτείται για την αποθήκευση του περιεχομένου των καταχωρητών, μαζί με την καθυστέρηση μεταξύ της λήψης αίτησης διακοπής και της έναρξης εξυπηρέτησής της, συνιστούν την καθυστέρηση διακοπών (interrupt latency).
- Μετά την εκτέλεση του υποπρογράμματος εξυπηρέτησης διακοπής, η ΚΜΕ συνεχίζει την εκτέλεση του προγράμματος που εκτελούσε όταν έλαβε το σήμα διακοπής.

Παράδειγμα

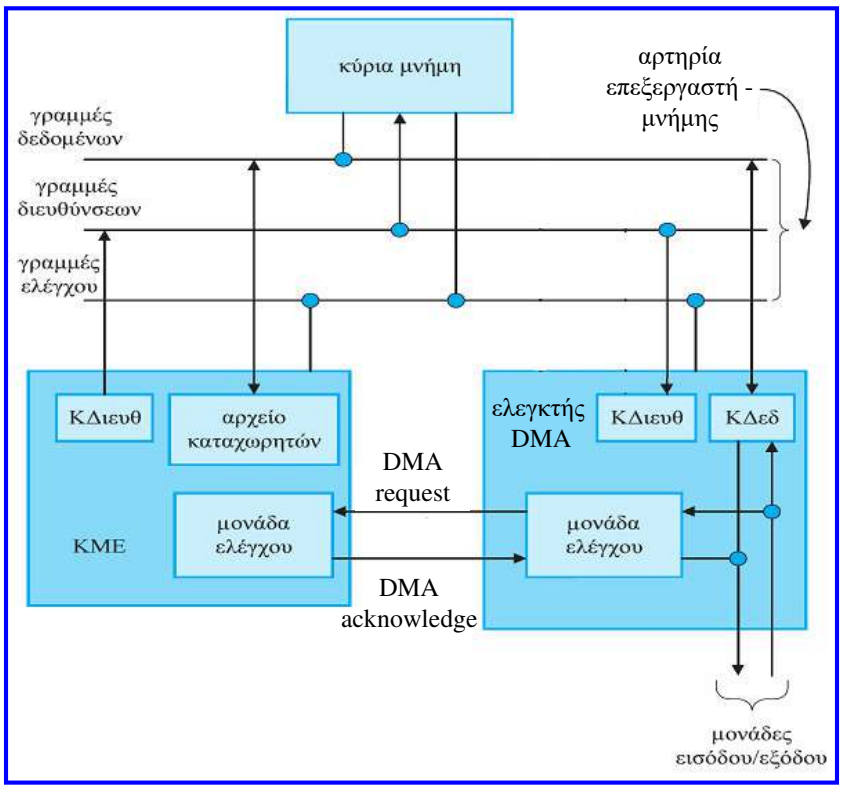
- Θεωρήστε υπολογιστή με επεξεργαστή συχνότητας λειτουργίας 1 GHz, που εκτός των άλλων, χρησιμοποιείται για τον έλεγχο λειτουργίας 1000 μονάδων Ε/Ε, οι οποίες έχουν τη δυνατότητα αποστολής σημάτων διακοπής, ώστε να εξυπηρετηθούν από την ΚΜΕ Το κόστος διακοπής λειτουργίας της ΚΜΕ για την εξυπηρέτηση της 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 της ΚΜΕ 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%.

Άμεση προσπέλαση μνήμης από μονάδες Ε/Ε

- Μεγαλύτερος βαθμός παραλληλίας μεταξύ της λειτουργίας της ΚΜΕ και της διαδικασίας εισόδου/εξόδου, επιτυγχάνεται όταν ο ελεγκτής της μονάδας Ε/Ε έχει τη δυνατότητα να μεταφέρει δεδομένα μεταξύ της μονάδας Ε/Ε και της κύριας μνήμης χωρίς την ουσιαστική παρεμβολή της ΚΜΕ (άμεση προσπέλαση μνήμης, direct memory access, DMA).
- Στην περίπτωση αυτή, πρόκειται για ελεγκτή DMA (άμεσης προσπέλασης μνήμης).
- Η ΚΜΕ παρέχει την εναρκτήρια διεύθυνση της κύριας μνήμης, τον αριθμό των λέξεων, την κατεύθυνση μεταφοράς των δεδομένων και τη διεύθυνση των δεδομένων στη μονάδα Ε/Ε και αρχικοποιεί τη διαδικασία εισόδου/εξόδου εκτελώντας μια ειδική εντολή.
- Κατά τη διάρκεια της μεταφοράς δεδομένων, η ΚΜΕ είναι ελεύθερη να εκτελεί άλλες λειτουργίες, με συνέπεια τη βελτίωση της απόδοσης του συστήματος.
- Όταν η μεταφορά των δεδομένων ολοκληρωθεί, ο ελεγκτής DMA ειδοποιεί την ΚΜΕ, στέλνοντας ένα σήμα διακοπής.
- Ο ελεγκτής DMA έχει πρόσβαση στην αρτηρία επεξεργαστή-μνήμης, τη χρήση της οποίας απαιτεί μέσω της ενεργοποίησης μιας γραμμής απαίτησης άμεσης προσπέλασης μνήμης (DMA request).
- Η ΚΜΕ όταν λάβει μια απαίτηση DMA παραχωρεί τον έλεγχο της αρτηρίας στον ελεγκτή DMA και τον ειδοποιεί ενεργοποιώντας μια γραμμή αναγνώρισης άμεσης προσπέλασης μνήμης (DMA acknowledge).

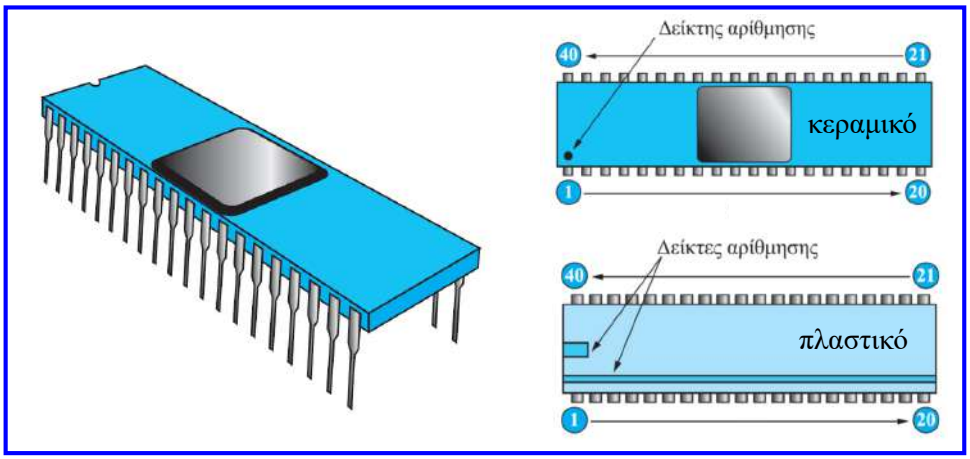
Άμεση προσπέλαση μνήμης από μονάδες Ε/Ε



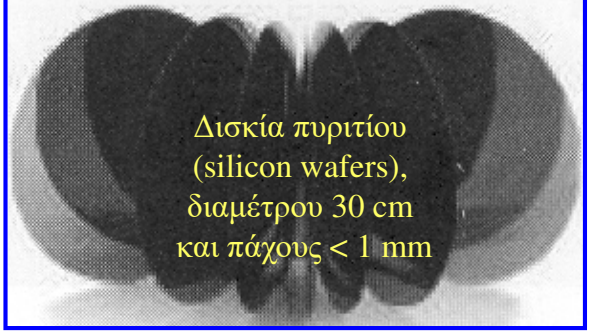
3. Μικροεπεξεργαστές

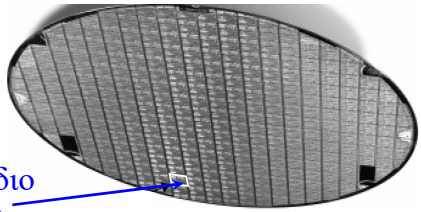
Ολοκληρωμένα κυκλώματα

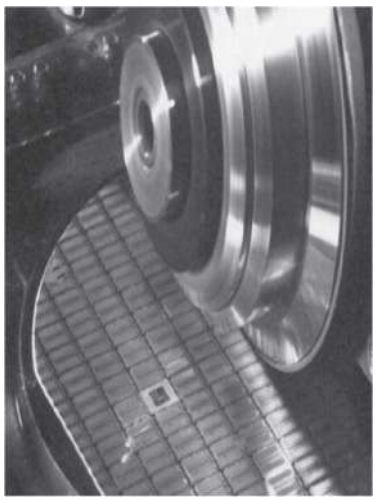
- Τα λογικά κυκλώματα σήμερα κατασκευάζονται με τη μορφή ολοκληρωμένων κυκλωμάτων (integrated circuits, ICs).
- Τα ολοκληρωμένα κυκλώματα αποτελούνται από ένα κομμάτι αγώγιμου υλικού (πυριτίου, silicon), στο οποίο ενσωματώνονται χιλιάδες ή εκατομμύρια ηλεκτρονικών στοιχείων (κυρίως τρανζίστορς, transistors), με τα οποία δημιουργούνται τα επιθυμητά κυκλώματα.
- Τα ολοκληρωμένα κυκλώματα προστατεύονται από πλαστικό ή κεραμικό περίβλημα (packaging), ενώ οι είσοδοι και οι έξοδοι τους καταλήγουν σε μεταλλικές επαφές εκτός του περιβλήματος, που αναφέρονται ως ακροδέκτες (pins).



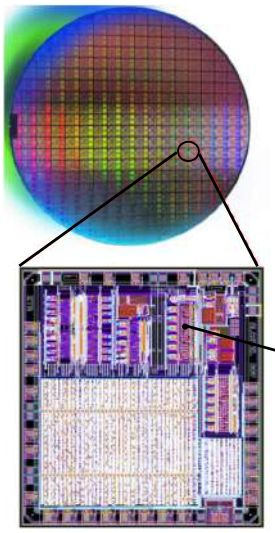
Ολοκληρωμένα κυκλώματα

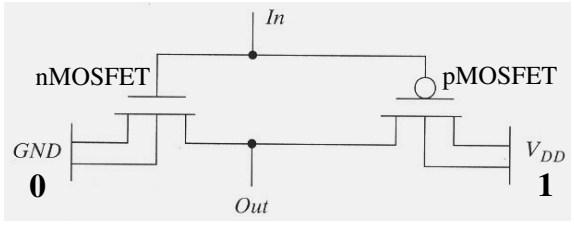


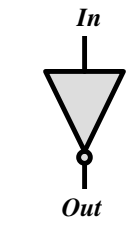


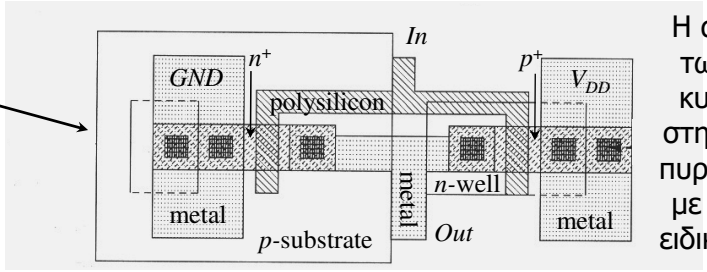


Ολοκληρωμένα κυκλώματα

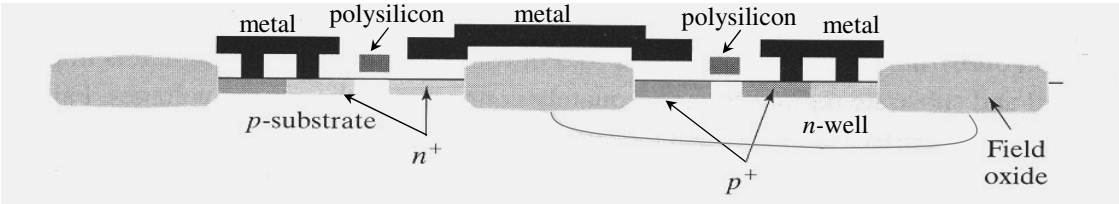




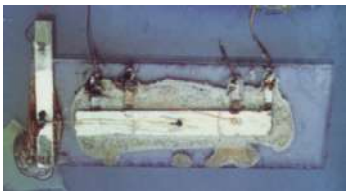


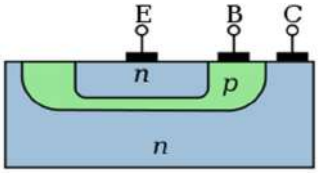


Η αποτύπωση των λογικών κυκλωμάτων στην επιφάνεια πυριτίου γίνεται με εναπόθεση ειδικών μασκών

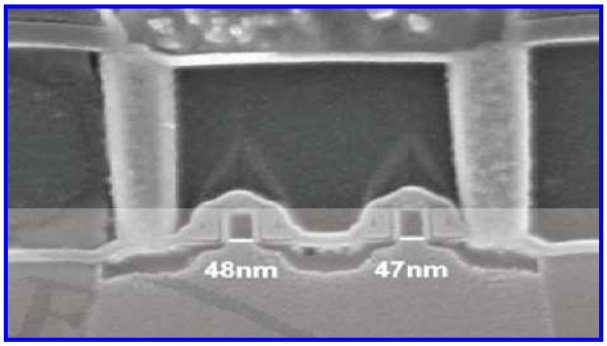


Ολοκληρωμένα κυκλώματα





Το πρώτο επίπεδο τρανζίστορ (1960)



MOSFET τρανζίστορς (2009)

Τα ολοκληρωμένα κυκλώματα πλεονεκτούν λόγω του μικρού μεγέθους τους, του χαμηλού κόστους, της χαμηλής κατανάλωσης ενέργειας και της υψηλής αξιοπιστίας.

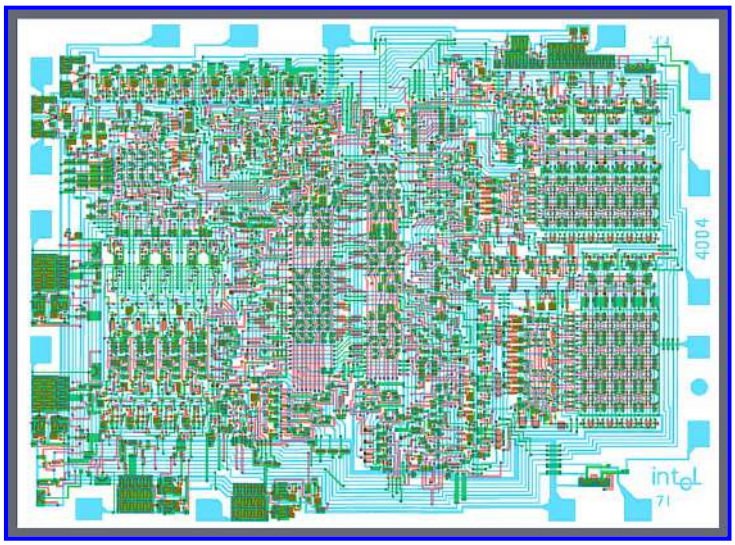
Ολοκληρωμένα κυκλώματα

- Τα ολοκληρωμένα κυκλώματα διακρίνονται σε κατηγορίες, ανάλογα με το πλήθος των τρανζίστορς που περιλαμβάνουν:
  - ✓ κυκλώματα μικρής κλίμακας ολοκλήρωσης (small scale integration, SSI), που περιλαμβάνουν μερικά μόνο τρανζίστορς,
  - ✓ κυκλώματα μεσαίας κλίμακας ολοκλήρωσης (medium scale integration, MSI), που περιλαμβάνουν από μερικές δεκάδες μέχρι εκατοντάδες τρανζίστορς.
  - ✓ κυκλώματα μεγάλης κλίμακας ολοκλήρωσης (large scale integration, LSI), που περιλαμβάνουν από μερικές εκατοντάδες έως μερικές χιλιάδες τρανζίστορς,
  - ✓ κυκλώματα πολύ μεγάλης κλίμακας ολοκλήρωσης (very large scale integration, VLSI), που περιλαμβάνουν από μερικές χιλιάδες έως εκατοντάδες χιλιάδες τρανζίστορς και
  - ✓ κυκλώματα πάρα πολύ μεγάλης κλίμακας ολοκλήρωσης (ultra large scale integration, ULSI), που περιλαμβάνουν πάνω από ένα εκατομμύριο τρανζίστορς.
- Καθώς ο αριθμός των τρανζίστορς, που περικλείονται στην επιφάνεια ενός ολοκληρωμένου κυκλώματος, έχει γίνει τρομακτικά μεγάλος, επιβάλλεται ο σχεδιασμός των σύγχρονων ολοκληρωμένων κυκλωμάτων με τη χρήση εργαλείων σχεδιασμού με υπολογιστή (computer-aided design).

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 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 αρκετά 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

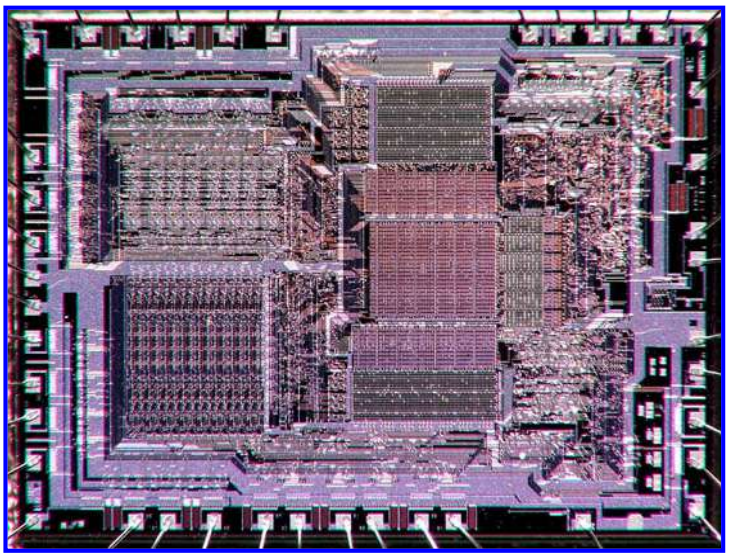
Πλήθος τρανζίστορς: 2300

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)



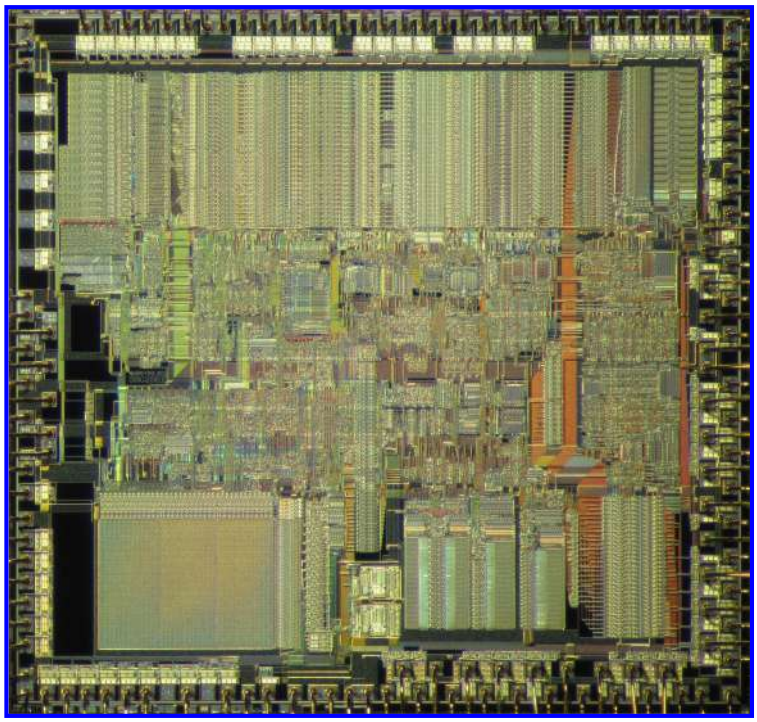
Τεχνολογία κατασκευής: 3 μm, NMOS

Επιφάνεια: 20 mm^2

Πλήθος τρανζίστορς: 6500

Συχνότητα λειτουργίας: 3 MHz

Μικροεπεξεργαστής Intel 80386 (1985)



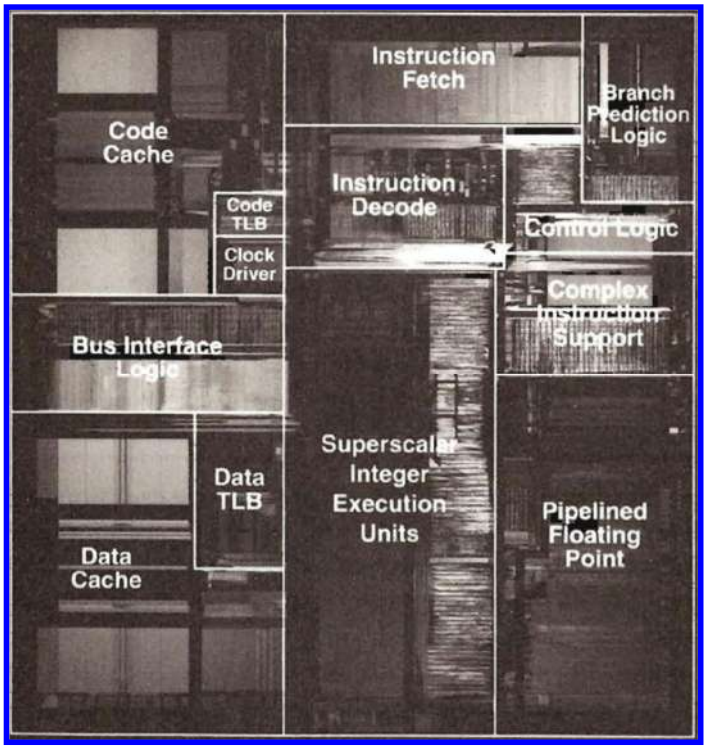
Τεχνολογία κατασκευής: 1.5 μm
CMOS

Επιφάνεια: 104 mm^2

Πλήθος τρανζίστορς: 275000

Συχνότητα λειτουργίας: 16 MHz

Μικροεπεξεργαστής Intel Pentium (1993)



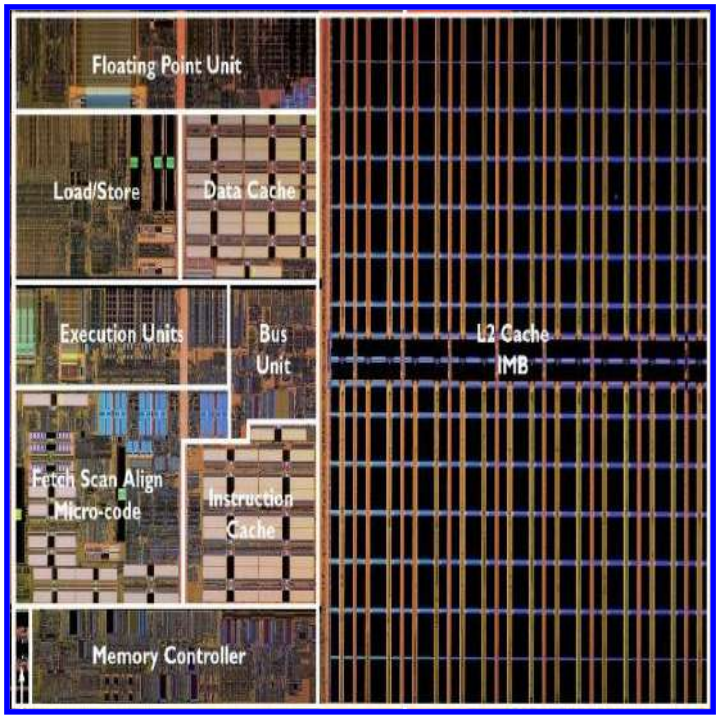
Τεχνολογία κατασκευής: 0.8 μm

Επιφάνεια: 294 mm^2

Πλήθος τρανζίστορς:
3.1 εκατομμύρια

Συχνότητα λειτουργίας: 66 MHz

Μικροεπεξεργαστής AMD Athlon 64 (2004)



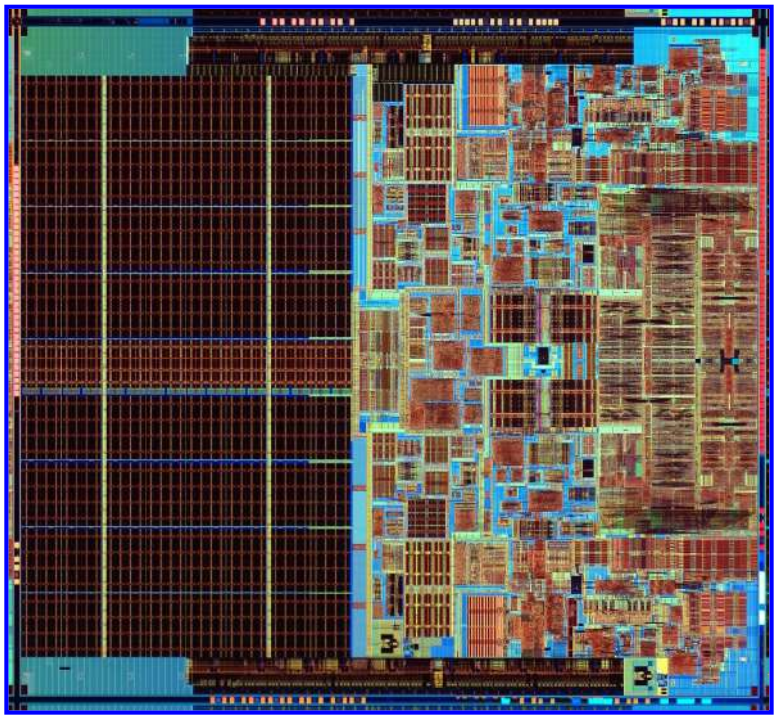
Τεχνολογία κατασκευής: 0.13 μm

Επιφάνεια: 193 mm^2

Πλήθος τρανζίστορς: 106 εκατομμύρια

Συχνότητα λειτουργίας: 1.6 GHz

Διπύρηνος μικροεπεξεργαστής Intel Core 2 Duo (2007)



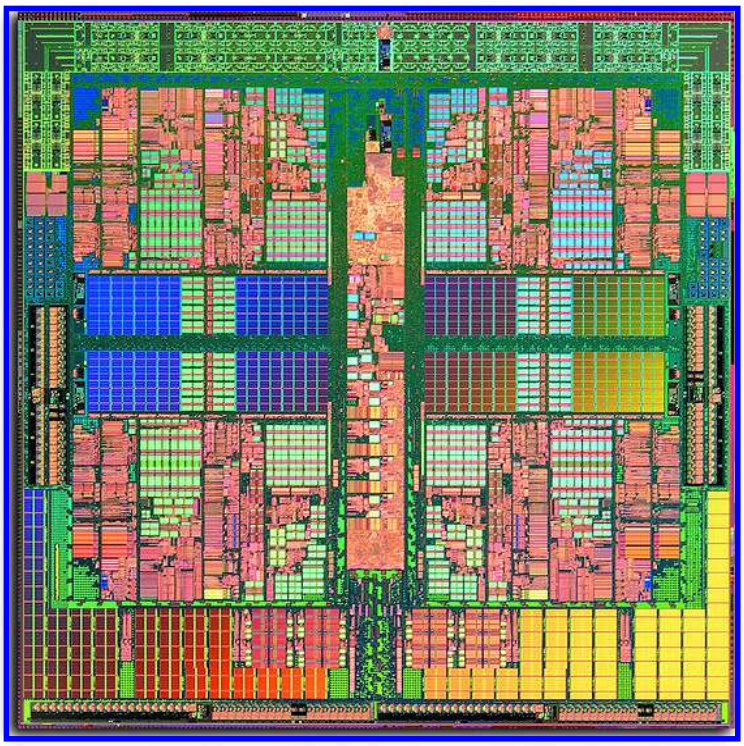
Τεχνολογία κατασκευής: 65 nm

Επιφάνεια: 111 mm^2

Πλήθος τρανζίστορς: 169 εκατομμύρια

Συχνότητα λειτουργίας: 1.8 GHz

Τετραπύρηνος μικροεπεξεργαστής AMD Opteron (2008)



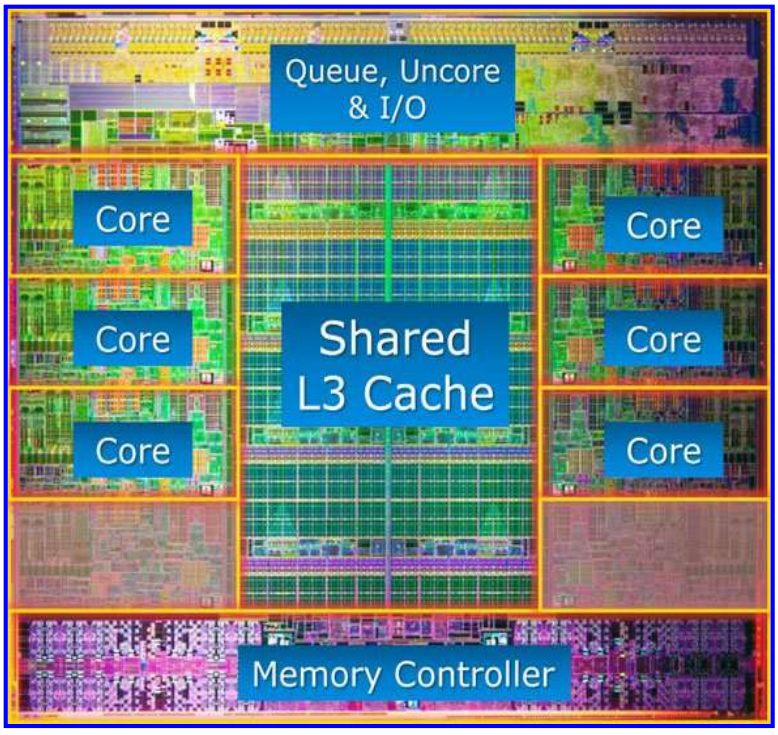
Τεχνολογία κατασκευής: 65 nm

Επιφάνεια: 283 mm^2

Πλήθος τρανζίστορς:
463 εκατομμύρια

Συχνότητα λειτουργίας: 2 GHz

Εξαπύρηνος μικροεπεξεργαστής Intel Core i7 (2011)



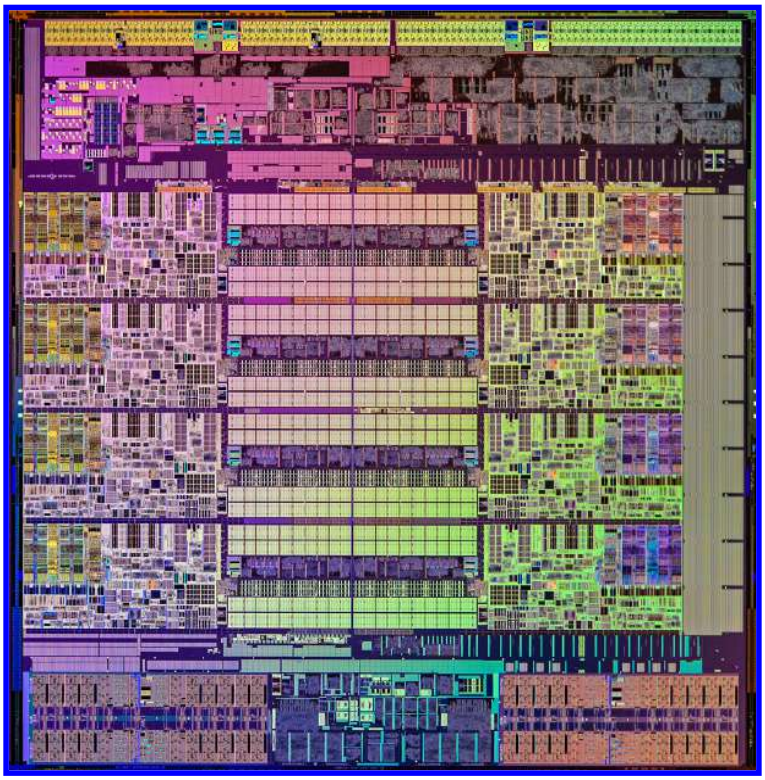
Τεχνολογία κατασκευής: 32 nm

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...

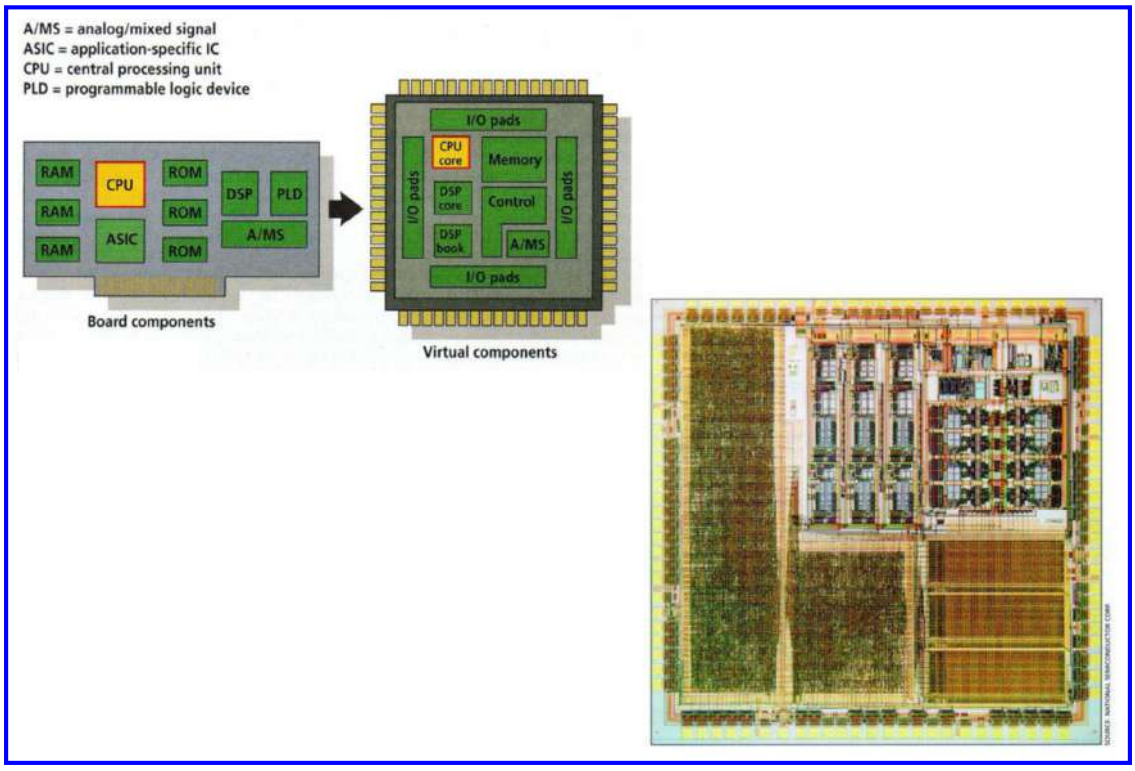
- Την διετία 2017-2018, η Intel ανέπτυξε τη σειρά μικροεπεξεργαστών Core i9 σε τεχνολογία κατασκευής 14 nm, που περιλαμβάνουν από 8 έως 14 πυρήνες με περισσότερα από 5 δισεκατομμύρια τρανζίστορς.

- 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.

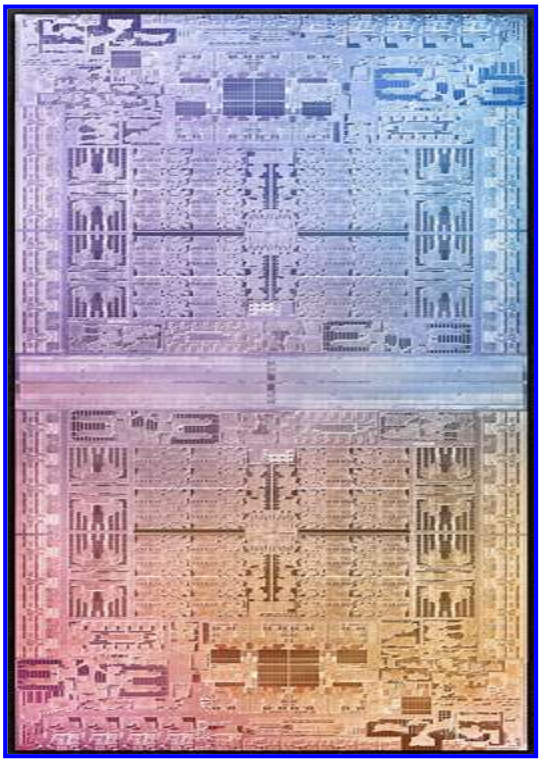
- Την διετία 2017-2018, η AMD ανέπτυξε τη σειρά μικροεπεξεργαστών Ryzen σε τεχνολογία 14 nm, που περιλαμβάνουν έως 16 πυρήνες, ενώ πρόσφατα ανέπτυξε την 3η γενιά μικροεπεξεργαστών Ryzen σε τεχνολογία 7 nm που περιλαμβάνει έως 32 πυρήνες, βελτιώνοντας τις επιδόσεις των μικροεπεξεργαστών της.

- 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.

Σύστημα σε ολοκληρωμένο κύκλωμα (SoC)



Apple M1 Ultra SoC (2022)



ARM processor-based system-on-chip

Τεχνολογία κατασκευής: 5 nm

Πλήθος τρανζίστορς:
114 δισεκατομμύρια

Δύο ψηφίδες (dies) συνδεδεμένες στην ίδια συσκευασία (package)

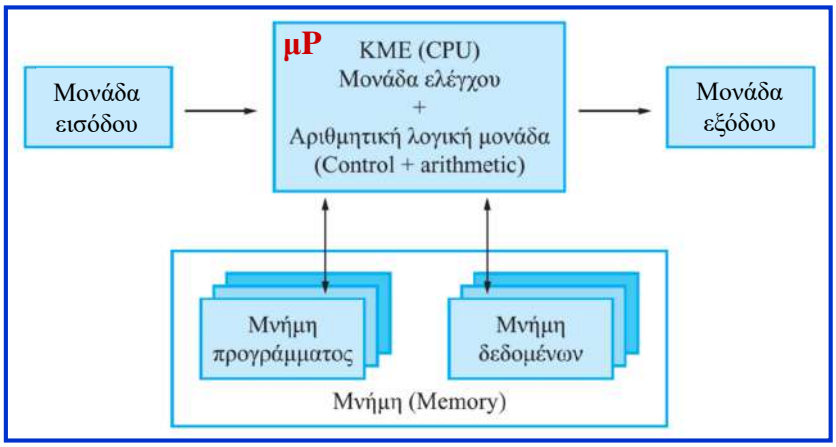
20 πυρήνες CPU

64 πυρήνες GPU (graphics processing unit για επιτάχυνση λειτουργιών γραφικών)

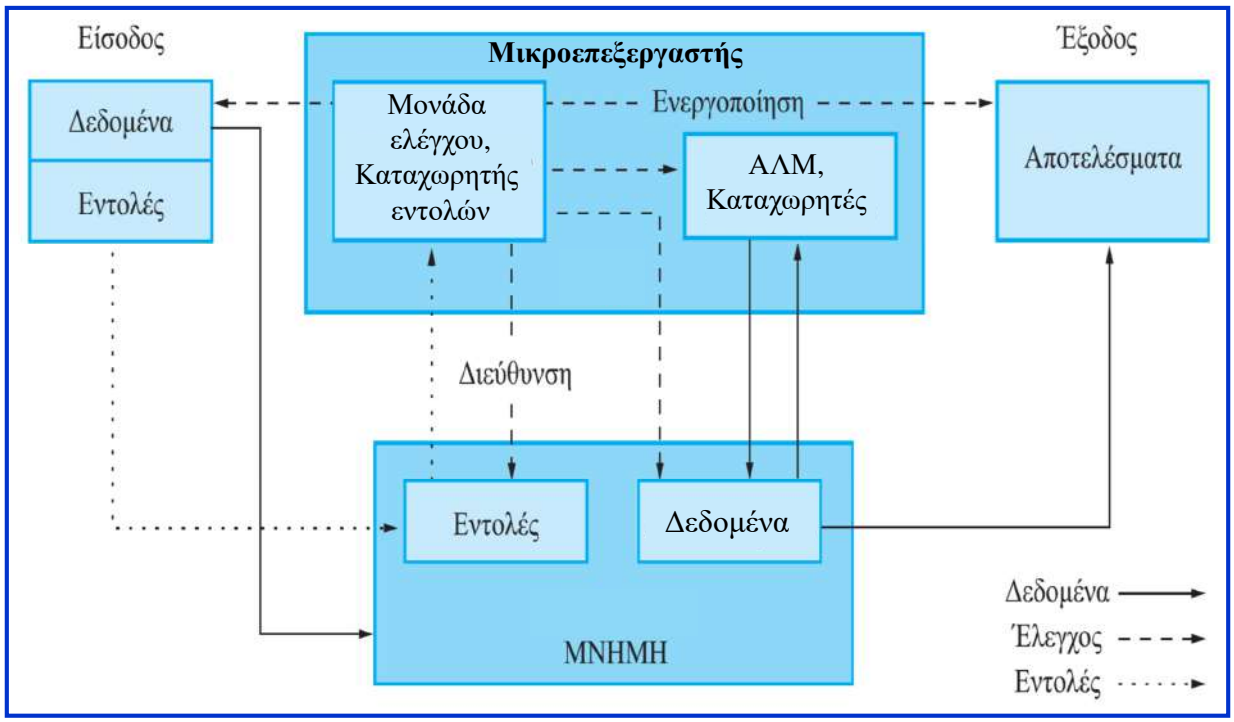
32 πυρήνες NPU (neural processing unit για επιτάχυνση λειτουργιών τεχνητής νοημοσύνης και μηχανικής μάθησης)

Βασικές έννοιες μικροεπεξεργαστών

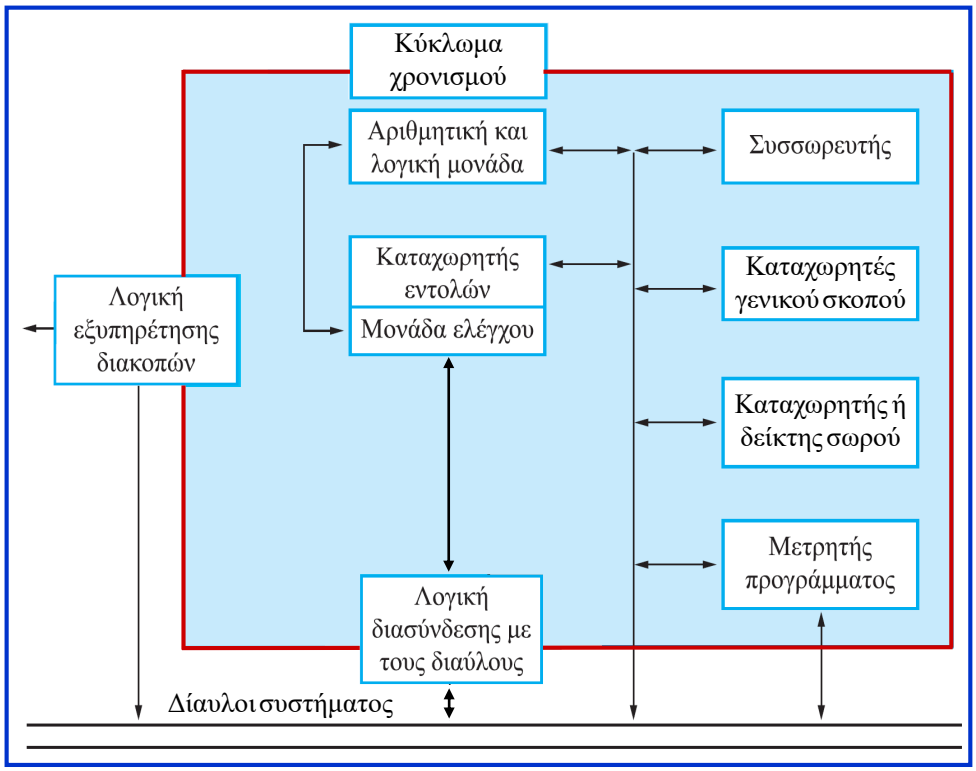
- Μικροεπεξεργαστής είναι ένα προγραμματιζόμενο ακολουθιακό λογικό σύστημα με δυνατότητα εκτέλεσης προγραμμάτων, το οποίο συνεργάζεται με μνήμες και περιφερειακές μονάδες τις οποίες συγχρονίζει.
- Πρόγραμμα είναι ένα σύνολο εντολών που αποσκοπούν στην επίτευξη μίας λειτουργίας.
- Εντολές είναι οι προκαθορισμένες ενέργειες με μεταβλητά δεδομένα (όπως αριθμητικές και λογικές εντολές, εντολές μετακίνησης δεδομένων, εντολές διακλάδωσης).
- Μικροϋπολογιστικό σύστημα είναι κάθε σύστημα που βασίζεται σε μικροεπεξεργαστή.



Λειτουργικό διάγραμμα μικροϋπολογιστικού συστήματος



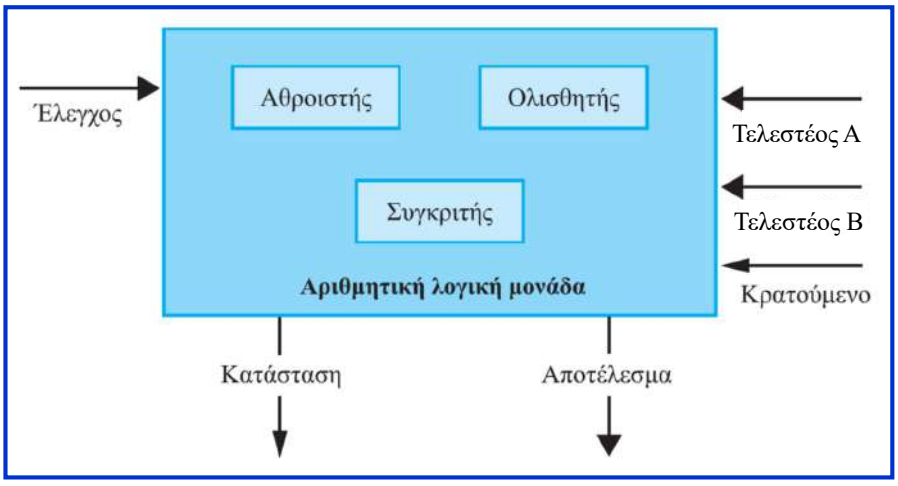
Δομή μικροεπεξεργαστή



Μερικές λειτουργίες (εξυπηρέτηση διακοπών, διασύνδεση με διαύλους ή αρτηρίες) επιτελούνται σε συνεργασία με άλλα ολοκληρωμένα κυκλώματα.

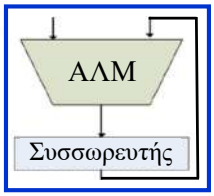
Επειδή η ενεργοποίηση των λειτουργιών γίνεται σε προκαθορισμένα χρονικά διαστήματα είναι αναγκαίο ένα κύκλωμα χρονισμού (εντός ή εκτός του μικροεπεξεργαστή) για την παραγωγή σήματος ρολογιού.

Αριθμητική λογική μονάδα (ΑΛΜ) μικροεπεξεργαστή



- Η ΑΛΜ εκτελεί λογικές και αριθμητικές πράξεις σε έναν ή δύο τελεστέους (τελούμενα ή έντελα), όπως: πρόσθεση και αφαίρεση δύο αριθμών, αύξηση ή ελάττωση αριθμού κατά ένα, σύγκριση δύο αριθμών, ολίσθηση ή περιστροφή αριθμού, υπολογισμός συμπληρώματος αριθμού και λογικές πράξεις AND, OR, NOT, NAND, NOR, XOR, XNOR.
- Όταν η ΑΛΜ δεν έχει τη δυνατότητα εκτέλεσης σύνθετων πράξεων, αυτές αναλύονται σε συνδυασμό βασικών πράξεων (για παράδειγμα, ο πολλαπλασιασμός αναλύεται σε διαδοχικές προσθέσεις και ολισθήσεις).

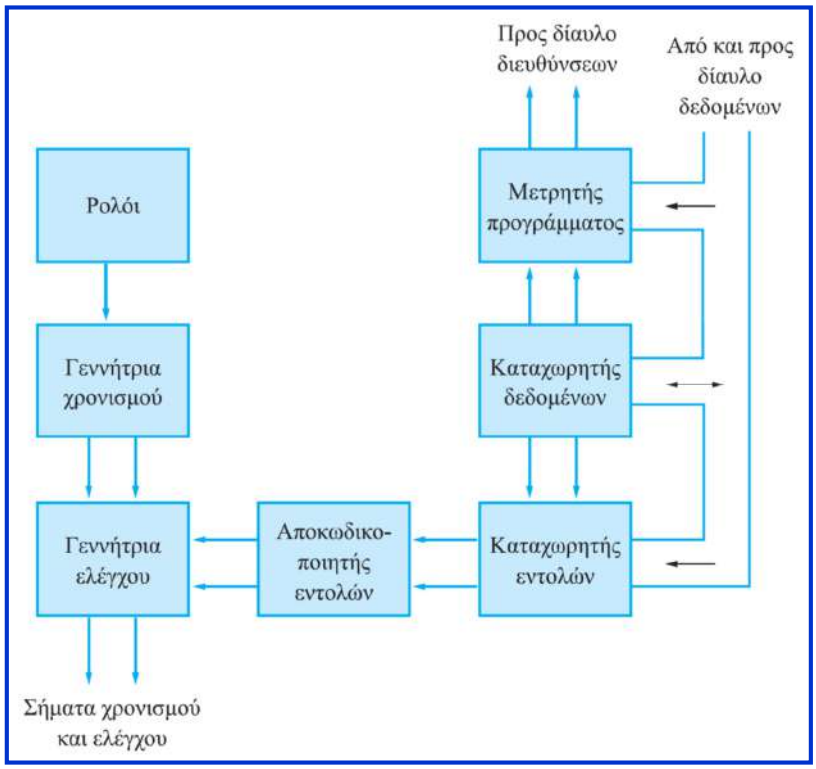
Καταχωρητές μικροεπεξεργαστή



Καταχωρητές (registers), μονάδες αποθήκευσης προσωρινών δεδομένων:

- συσσωρευτής (accumulator): αποθηκεύει τον έναν αριθμό για να εκτελεστεί μια πράξη στην ΑΛΜ και στη συνέχεια το αποτέλεσμα πράξης,
- καταχωρητές γενικού σκοπού (general purpose registers): είναι περισσότεροι από ένας και αποθηκεύουν τον άλλο αριθμό για να εκτελεστεί μια πράξη (όταν υπάρχει συσσωρευτής, διαφορετικά ενδέχεται να αποθηκεύουν και τους δύο αριθμούς στους οποίους θα εκτελεστεί μια πράξη), ενδιάμεσα αποτελέσματα, την διεύθυνση μνήμης όπου βρίσκονται οι αριθμοί στους οποίους θα εκτελεστεί μια πράξη ή την διεύθυνση μνήμης όπου θα αποθηκευτεί το αποτέλεσμα μιας πράξης,
- καταχωρητής κατάστασης επεξεργαστή (processor status register): αποτελείται από σύνολο δυαδικών ψηφίων (σημαίες κατάστασης, flags), που υποδεικνύουν την παρούσα κατάσταση του μικροεπεξεργαστή, π.χ. μηδενικό ή αρνητικό αποτέλεσμα στην τελευταία πράξη, ύπαρξη κρατούμενου ή υπερχείλισης,
- καταχωρητής εντολών (instruction register): αποθηκεύει τον κώδικα λειτουργίας της προς εκτέλεση εντολής, ενώ αυτή αποκωδικοποιείται για να εκτελεστεί στη συνέχεια,
- μετρητής προγράμματος (program counter): αποθηκεύει την διεύθυνση θέσης μνήμης, όπου είναι αποθηκευμένη η επόμενη προς εκτέλεση εντολή,
- δείκτης σωρού (stack pointer): αποθηκεύει την διεύθυνση θέσης μνήμης της κορυφής σωρού, η οποία περιέχει την τελευταία πληροφορία που τοποθετήθηκε στον σωρό.

Μονάδα ελέγχου μικροεπεξεργαστή



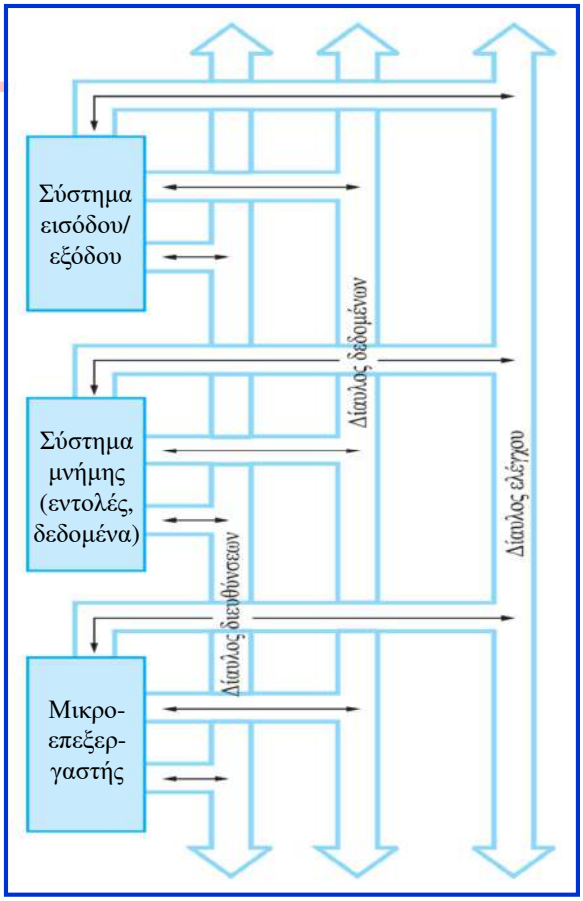
Η μονάδα ελέγχου συντονίζει την ΑΛΜ, τη μνήμη και τις μονάδες εισόδου/εξόδου, έτσι ώστε να ενεργοποιούνται στα κατάλληλα χρονικά διαστήματα για να εκτελεστεί με ορθό τρόπο μια εντολή.

Ο κώδικας λειτουργίας της εντολής (που αποθηκεύεται στον καταχωρητή εντολών) αποκωδικοποιείται (αναγνωρίζεται) από τον αποκωδικοποιητή εντολών, έτσι ώστε να παραχθούν από τη γεννήτρια ελέγχου τα κατάλληλα σήματα ενεργοποίησης με την ορθή σειρά των υπόλοιπων μονάδων του συστήματος.

Δίαυλοι ή αρτηρίες (buses)

Οι δίαυλοι είναι σύνολο από γραμμές που συνδέουν τα τμήματα του μικροϋπολογιστικού συστήματος:

- δίαυλος δεδομένων (data bus): δίαυλος δύο κατευθύνσεων (bidirectional) που μεταφέρει εντολές προς αποκωδικοποίηση και δεδομένα προς επεξεργασία,
- δίαυλος διευθύνσεων (address bus): δίαυλος μοναδικής κατεύθυνσης (unidirectional) που μεταφέρει την διεύθυνση μνήμης από την οποία θα διαβαστούν ή στην οποία θα γραφτούν δεδομένα,
- δίαυλος ελέγχου (control bus): δίαυλος μοναδικής κατεύθυνσης που μεταφέρει πληροφορίες σχετικά με τη λειτουργία που πρόκειται να εκτελεστεί, π.χ. ανάγνωση/εγγραφή μνήμης, ενεργοποίηση μονάδας εισόδου/εξόδου.

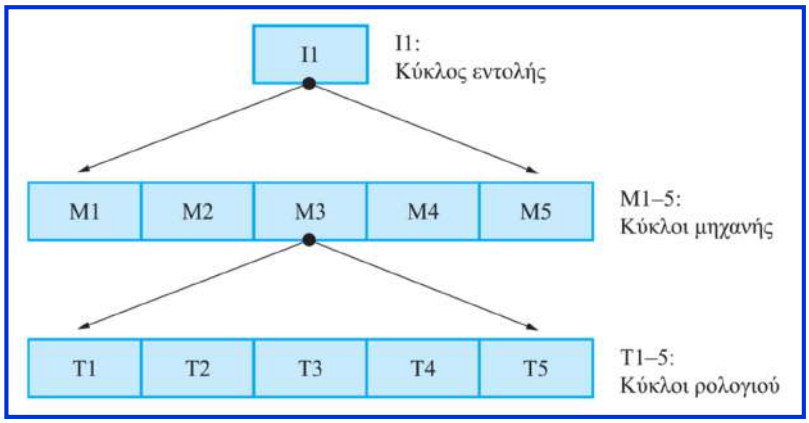


Προσκόμιση, αποκωδικοποίηση και εκτέλεση εντολής

Ο μικροεπεξεργαστής εκτελεί μια εντολή επαναλαμβάνοντας στη γενική περίπτωση μια σειρά επτά βημάτων:

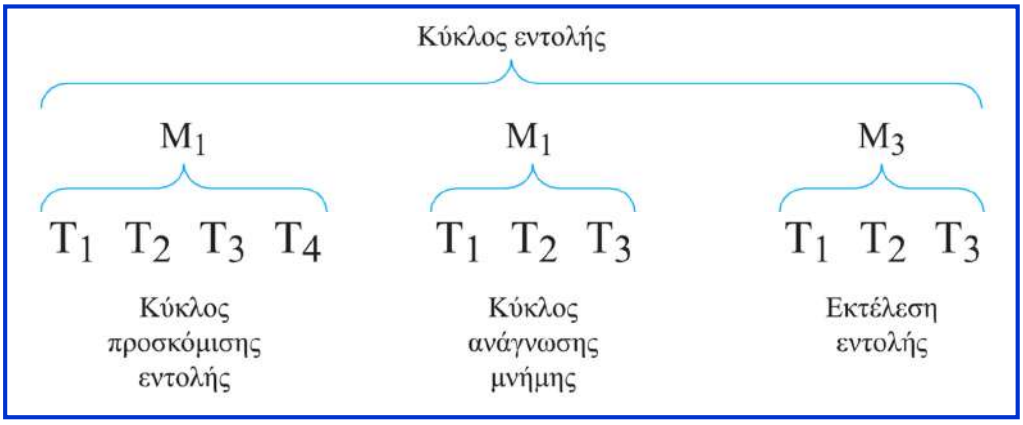
1. Προσκόμιση της εντολής που είναι αποθηκευμένη στη θέση μνήμης που δείχνει ο μετρητής του προγράμματος και αποθήκευσή της στον καταχωρητή εντολών.
2. Αλλαγή (αύξηση) του περιεχομένου του μετρητή προγράμματος, ώστε να περιέχει τη θέση μνήμης στην οποία είναι αποθηκευμένη η επόμενη προς εκτέλεση εντολή.
3. Αποκωδικοποίηση (αναγνώριση του τύπου) της εντολής και έλεγχος εάν η εντολή απαιτεί δεδομένα από τη μνήμη και αν ναι, προσδιορισμός της διεύθυνσης που είναι αποθηκευμένα.
4. Προσκόμιση των δεδομένων σε κάποιους από τους καταχωρητές του μικροεπεξεργαστή.
5. Εκτέλεση της εντολής.
6. Αποθήκευση των αποτελεσμάτων.
7. Επιστροφή στο πρώτο βήμα, ώστε να αρχίσει η εκτέλεση της επόμενης εντολής.

Χρονισμός των εντολών



- Κύκλος εντολής: απαραίτητος χρόνος για την ολοκλήρωση των βημάτων μιας εντολής.
- Ο κύκλος εντολής μπορεί να διαιρείται σε κύκλους μηχανής (συνήθως 1 έως 5).
- Βασικοί κύκλοι μηχανής: κύκλος προσκόμισης εντολής, κύκλος ανάγνωσης από τη μνήμη ή από μονάδα εισόδου/εξόδου, κύκλος εγγραφής στη μνήμη ή σε μονάδα εισόδου/εξόδου, κύκλος εκτέλεσης εσωτερικής λειτουργίας του μικροεπεξεργαστή, δηλαδή κάθε κύκλος μηχανής είναι ο απαραίτητος χρόνος για την εκτέλεση μιας βασικής λειτουργίας.
- Κάθε κύκλος μηχανής μπορεί να διαιρείται σε κύκλους ρολογιού (συνήθως 3 ως 6) και ένας κύκλος ρολογιού ισούται με την περίοδο του σήματος ρολογιού του μικροεπεξεργαστή.

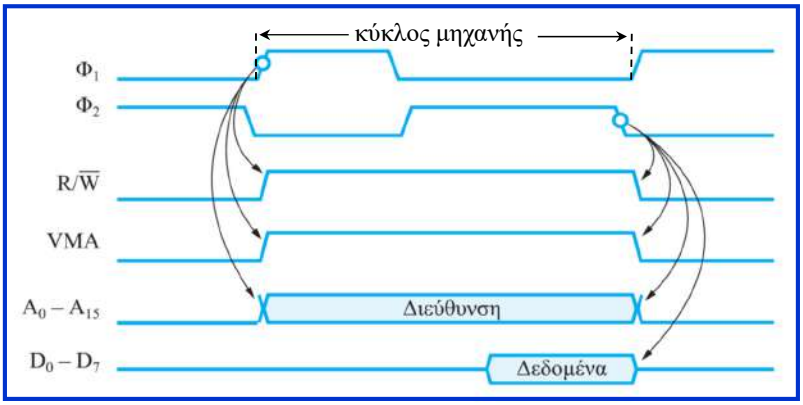
Χρονισμός των εντολών



- Παράδειγμα: εντολή πρόσθεσης δύο αριθμών (ADD), ο ένας εκ των οποίων είναι αποθηκευμένος στον συσσωρευτή και ο δεύτερος σε μια θέση μνήμης που υποδεικνύεται στην εντολή. Το αποτέλεσμα της πρόσθεσης αποθηκεύεται στον συσσωρευτή.
- Η εντολή αποτελείται από τρεις (3) κύκλους μηχανής: κύκλος προσκόμισης της εντολής από τη μνήμη κατά τον οποίο ο κωδικός λειτουργίας της εντολής μεταφέρεται στον αποκωδικοποιητή εντολών και αποκωδικοποιείται, κύκλος ανάγνωσης από τη μνήμη της τιμής που θα προστεθεί στο περιεχόμενο του συσσωρευτή και κύκλος εκτέλεσης της λειτουργίας της πρόσθεσης και αποθήκευσης του αποτελέσματος στον συσσωρευτή.

Χρονισμός των εντολών

Παράδειγμα διαγραμμάτων χρονισμού κύκλου ανάγνωσης δεδομένων από τη μνήμη, σε μικροϋπολογιστικό σύστημα με μικροεπεξεργαστή Motorola 6800



- Ο 6800 χρησιμοποιεί δύο συμπληρωματικά σήματα ρολογιού (Φ_1, Φ_2) για το χρονισμό των λειτουργιών στο μικροϋπολογιστικό σύστημα (όταν $\Phi_1 = 1$ τότε $\Phi_2 = 0$ και αντίστροφα).
- Κατά τη μετάβαση του Φ_1 στην τιμή 1, το σήμα R/W′ μεταβαίνει στην τιμή 1 υποδεικνύοντας λειτουργία ανάγνωσης από τη μνήμη και το σήμα VMA μεταβαίνει στην τιμή 1, δηλώνοντας ότι η διεύθυνση όπου βρίσκονται τα δεδομένα, έχει τοποθετηθεί στον δίαυλο διευθύνσεων. Τα σήματα R/W′, VMA παραμένουν σε τιμή 1 μέχρι το τέλος του κύκλου μηχανής.
- Η μνήμη τοποθετεί τα δεδομένα στον δίαυλο δεδομένων πριν το τέλος του κύκλου μηχανής και ο μικροεπεξεργαστής μεταφέρει τα δεδομένα στους εσωτερικούς καταχωρητές του με την κατερχόμενη ακμή του Φ_2.

Εντολές μικροεπεξεργαστή

- Οι εντολές που μπορεί να εκτελέσει ένας μικροεπεξεργαστής αποτελούν το σύνολο εντολών του (instruction set).
- Το μήκος κάθε εντολής εξαρτάται από τον αριθμό των τελούμενων ή τελεστέων ή έντελων (instruction operands) που περιλαμβάνει και συνήθως είναι μεταξύ ενός και τριών bytes στην 1η γενιά επεξεργαστών, ενώ είναι περισσότερα bytes στους σημερινούς επεξεργαστές.

Εντολή ενός byte	Κωδικός λειτουργίας		
Εντολή δύο bytes	Κωδικός λειτουργίας	Έντελο 1 (δεδομένα ή διεύθυνση)	
Εντολή τριών bytes	Κωδικός λειτουργίας	Έντελο 1 (δεδομένα ή διεύθυνση)	Έντελο 2 (δεδομένα ή διεύθυνση)

Κατηγορίες εντολών μικροεπεξεργαστή

- Λογικές εντολές: εντολές λογικών πράξεων (AND, OR, NOT, NAND, NOR, XOR, XNOR), εντολές ολίσθησης αριθμού και εντολές περιστροφής αριθμού.
- Αριθμητικές εντολές: εντολές αριθμητικών πράξεων (πρόσθεση, αφαίρεση, πολλαπλασιασμός διαίρεση), εντολές αύξησης ή ελάττωσης αριθμού κατά ένα, εντολές υπολογισμού συμπληρώματος αριθμού, εντολές σύγκρισης δύο αριθμών.
- Εντολές μετακίνησης δεδομένων: εντολές φόρτωσης αριθμού στον συσσωρευτή ή σε καταχωρητές γενικού σκοπού από τη μνήμη (load), εντολές αποθήκευσης αριθμού στην κύρια μνήμη (store), εντολές μετακίνησης δεδομένων μεταξύ θέσεων της μνήμης, εντολές ανάγνωσης ή εγγραφής δεδομένων σε μονάδες εισόδου/εξόδου.
- Εντολές διακλάδωσης ή άλματος: εντολές (jump) που τροποποιούν τη σειρά εκτέλεσης των εντολών σε σχέση με τη σειριακή εκτέλεση των εντολών (που καθορίζεται από τον μετρητή προγράμματος) και μπορεί να είναι χωρίς συνθήκη μεταφέροντας την εκτέλεση των εντολών στο σημείο που περιγράφει το έντελό τους ή με συνθήκη, όπου μετά από έλεγχο της συνθήκης τους (π.χ. συνθήκη που αφορά το αποτέλεσμα της τελευταίας πράξης που εκτελέστηκε) καθορίζεται εάν θα αλλάξει η σειριακή εκτέλεση των εντολών.
- Εντολές κλήσης υπορουτίνας ή επιστροφής από υπορουτίνα (call, return).
- Γενικές εντολές: εντολές διαχείρισης σωρού (push, pop), εντολές διακοπής λειτουργίας (halt, break), εντολή nop (no operation) για εισαγωγή καθυστέρησης, κ.ά.

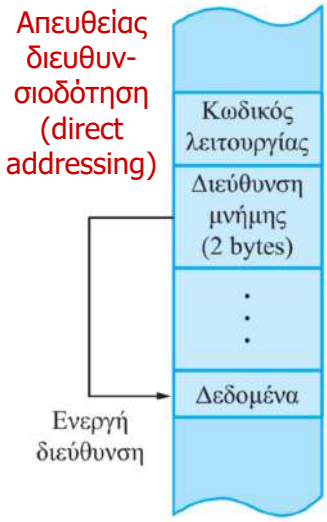
Τρόποι διευθυνσιοδότησης

Τρόποι διευθυνσιοδότησης (addressing modes) είναι οι τρόποι με τους οποίους ο μικροεπεξεργαστής υποδεικνύει τη θέση που βρίσκονται τα δεδομένα που απαιτούνται από μία εντολή για την εκτέλεση της λειτουργίας που καθορίζεται σε αυτή.

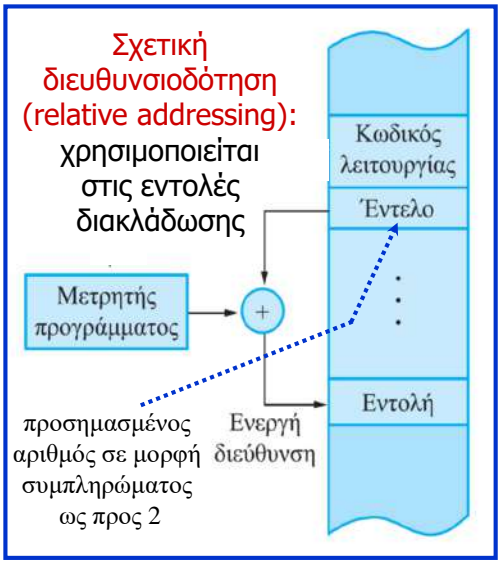
Διευθυνσιοδότηση καταχωρητή (register addressing): δεδομένα σε καταχωρητή που προσδιορίζεται στον κωδικό λειτουργίας

Άμεση διευθυνσιο-δότηση (immediate addressing)

Απευθείας διευθυν-σιοδότηση (direct addressing)

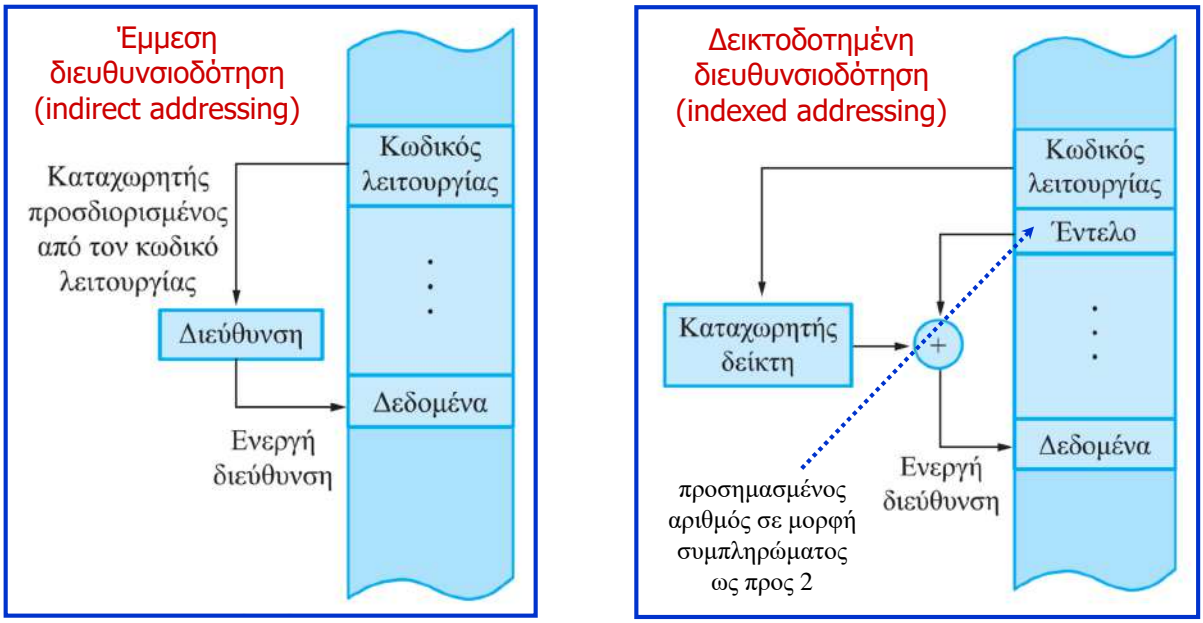


Σχετική διευθυνσιοδότηση (relative addressing): χρησιμοποιείται στις εντολές διακλάδωσης



Διευθυνσιοδότηση μηδενικής σελίδας (zero page addressing): απευθείας διευθυνσιοδότηση με διεύθυνση μνήμης 1 byte (διευθυνσιοδοτούνται μόνο οι θέσεις από 0000 έως 00FF).

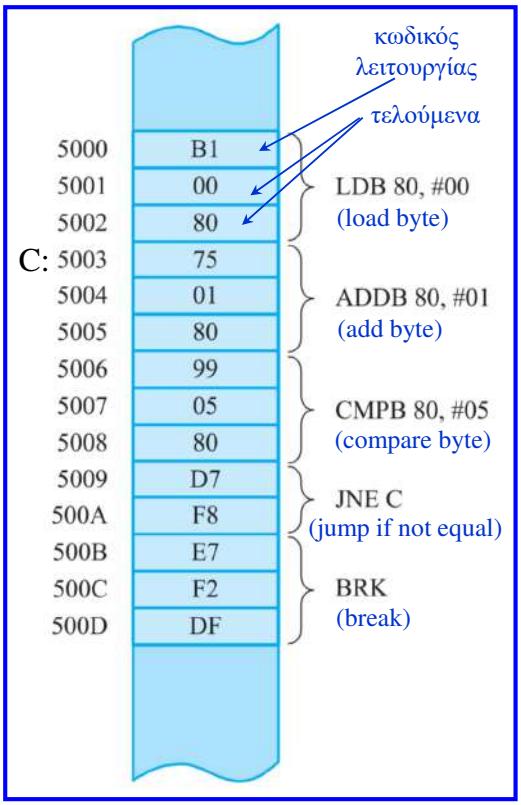
Τρόποι διευθυνσιοδότησης



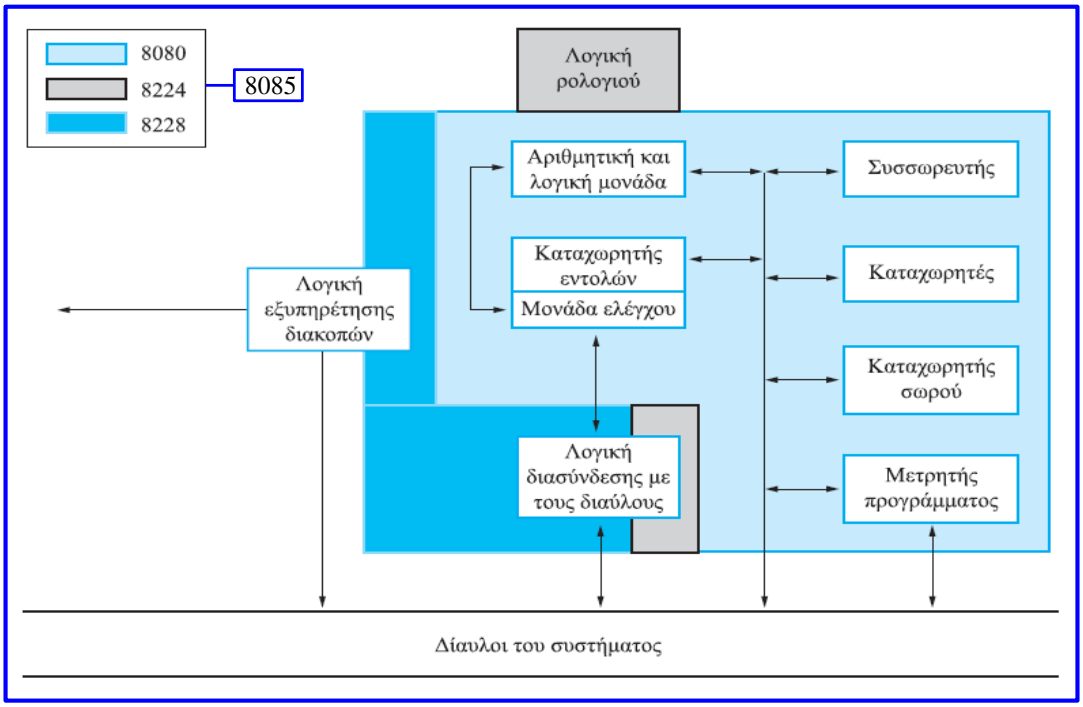
Ανάλογα με τον τρόπο διευθυνσιοδότησης που χρησιμοποιείται, μια εντολή έχει διαφορετικούς κωδικούς λειτουργίας

Εκτέλεση προγράμματος σε μικροεπεξεργαστή

- Το πρόγραμμα (Intel 80196) που αποθηκεύει διαδοχικά στον καταχωρητή 80_{16} τις τιμές από 1_{16} έως 5_{16}, αποθηκεύεται στην κύρια μνήμη.
- Η εκτέλεση ξεκινάει από τη θέση μνήμης 5000_{16} που περιέχεται στον μετρητή προγράμματος.
- Ο μικροεπεξεργαστής διαβάζει το 1ο byte (κωδικός λειτουργίας), το αποκωδικοποιεί και αντιλαμβάνεται ότι ακολουθούν 2 bytes με τα τελούμενα εντολής.
- Οι δεκαεξαδικοί αριθμοί που περιγράφονται στην εντολή με # αφορούν δεδομένα, ενώ χωρίς # αφορούν διεύθυνση καταχωρητή.
- Στην εντολή διακλάδωσης JNE, το τελούμενο $F8_{16}$ αναπαριστά τον αριθμό -8_{10} σε μορφή συμπληρώματος ως προς 2, με αποτέλεσμα εάν δεν ισχύει η συνθήκη της ισότητας στην εντολή σύγκρισης, τότε προστίθεται στον μετρητή προγράμματος (που περιέχει τη διεύθυνση της επόμενης εντολής, $500B_{16}$) ο αριθμός -8_{10}, έτσι ώστε αυτός να δείχνει την διεύθυνση 5003_{16}.



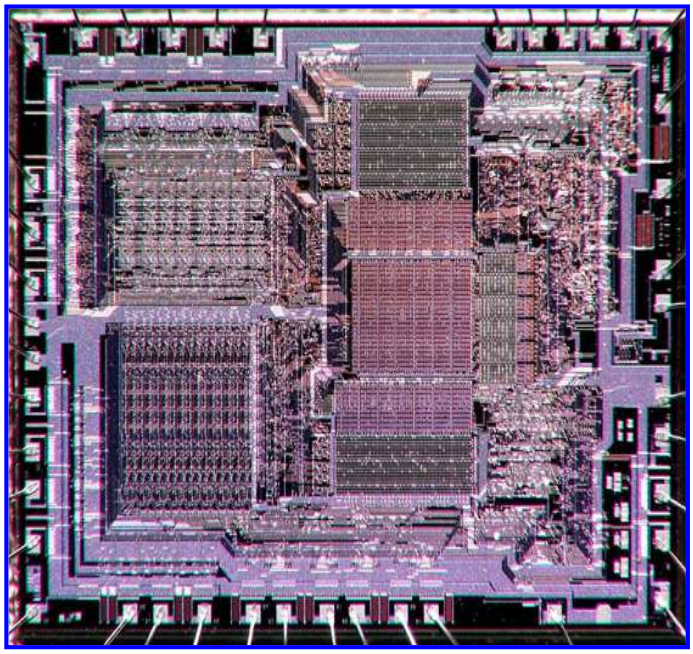
Δομή των μικροεπεξεργαστών Intel 8080 / 8085



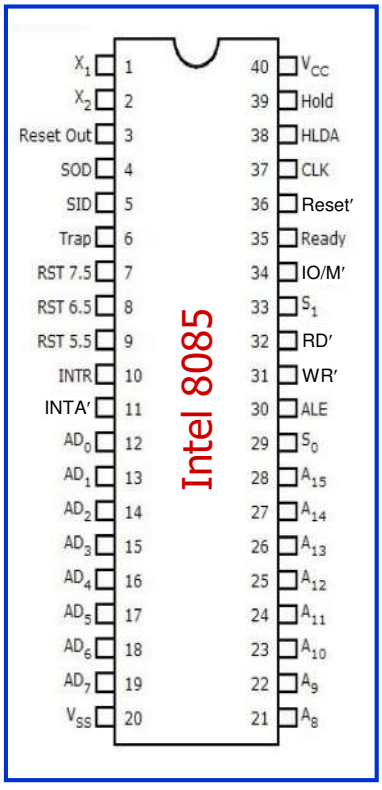
Δομή των μικροεπεξεργαστών Intel 8080 / 8085

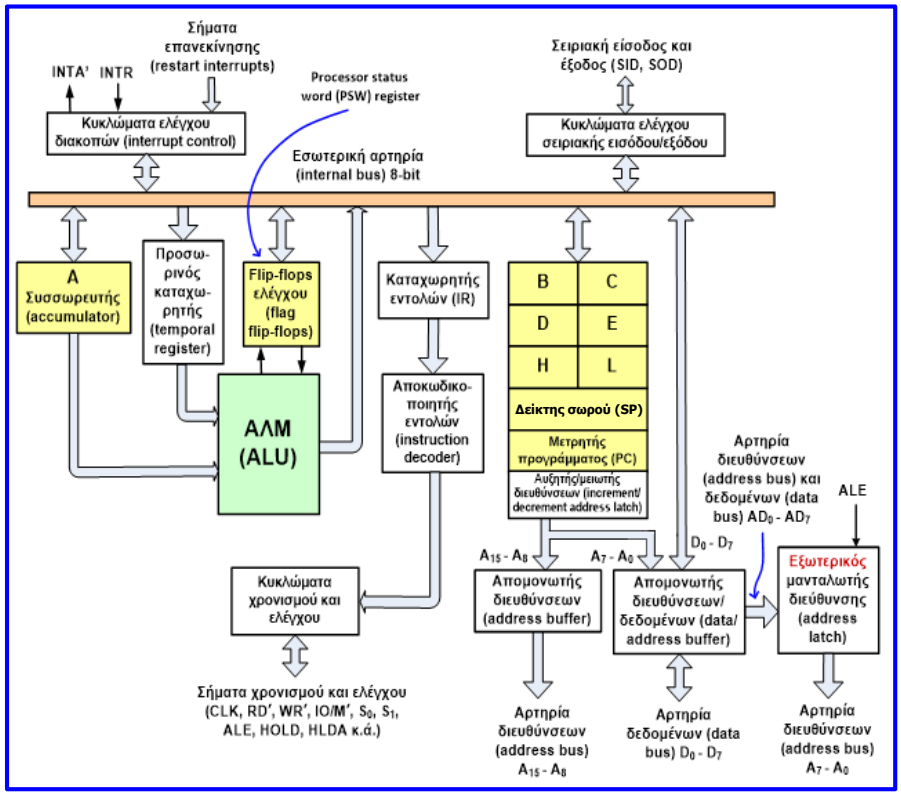
- Ο μικροεπεξεργαστής Intel 8080 χρησιμοποιεί τη βασική δομή μικροεπεξεργαστή που προαναφέρθηκε και υποστηρίζεται από 2 βασικά ολοκληρωμένα κυκλώματα: 8224, 8228.
- Το ολοκληρωμένο κύκλωμα 8224 αποτελεί το κύκλωμα χρονισμού του μικροεπεξεργαστή, δηλαδή τη γεννήτρια των δύο συμπληρωματικών σημάτων ρολογιού (Φ_1, Φ_2) που είναι απαραίτητα για τη λειτουργία του.
- Το ολοκληρωμένο κύκλωμα 8228 περιλαμβάνει κυκλώματα οδήγησης του διαύλου δεδομένων, παραγωγής των απαραίτητων σημάτων ελέγχου και εξυπηρέτησης των διακοπών.
- Ο μικροεπεξεργαστής 8085 αποτελεί βελτιωμένη έκδοση του 8080.
- Ενσωματώνει τις λειτουργίες της γεννήτριας σήματος ρολογιού (χρησιμοποιεί ένα σήμα ρολογιού), της παραγωγής των σημάτων ελέγχου, της οδήγησης του διαύλου δεδομένων και βελτιώνει τη διαδικασία διαχείρισης των διακοπών.
- Είναι 100% συμβατός με τον Intel 8080 σε ότι αφορά τον προγραμματισμό του.
- Υλοποιεί την πολυπλεξία μέρους του διαύλου διευθύνσεων με το δίαυλο δεδομένων, με αποτέλεσμα την μείωση του πλήθος των ακροδεκτών του.
- Τέλος, διαθέτει ακροδέκτες εισόδου και εξόδου για τη σειριακή λήψη και μετάδοση δεδομένων.

Μικροεπεξεργαστής Intel 8085



Ακροδέκτες και αρχιτεκτονική του Intel 8085



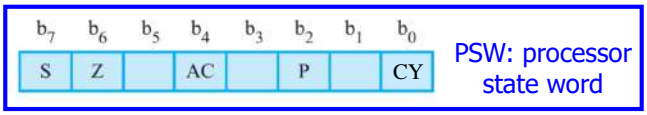


Καταχωρητές του Intel 8085

- 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 σωρού (stack pointer)	16 bits
PSW	Καταχωρητής κατάστασης	8 bits (5 σημαίες ελέγχου)

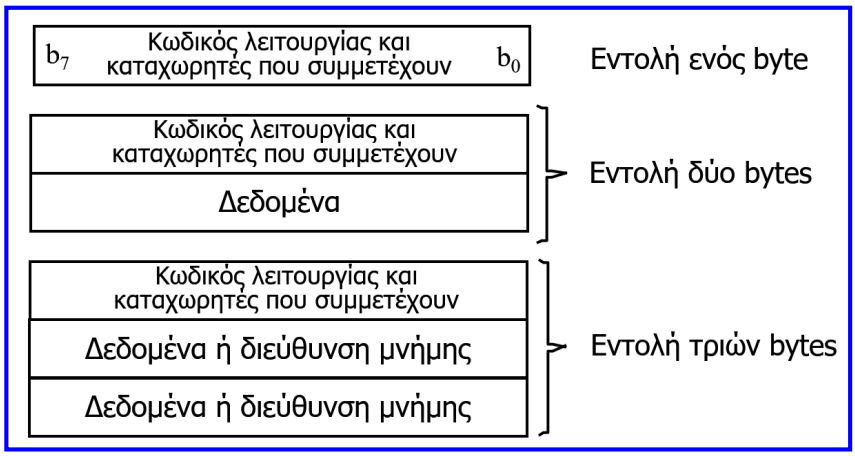
- Σημαίες κατάστασης ή ελέγχου (flags):



  - ✓ 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 (το λιγότερο σημαντικό byte αποθηκεύεται στη μικρότερη διεύθυνση).
- 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 συνήθως επηρεάζουν τις σημαίες ελέγχου (κατάστασης).
- 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: εναπόθεση (PUSH) στον σωρό και ανάκτηση (POP) από τον σωρό. Χρησιμοποιούνται όταν οι καταχωρητές του επεξεργαστή δεν επαρκούν για την αποθήκευση ενδιάμεσων αποτελεσμάτων, καθώς και για αποθήκευση και αποκατάσταση της κατάστασης του επεξεργαστή (τρέχουσες τιμές καταχωρητών και σημαιών ελέγχου) πριν και μετά την εκτέλεση υπορουτίνας (συνάρτησης), αντίστοιχα.
- Εντολές για ανάγνωση δεδομένων από θύρα εισόδου (IN) και εγγραφή δεδομένων σε θύρα 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 επεξεργαστή 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 (άλματος) 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.

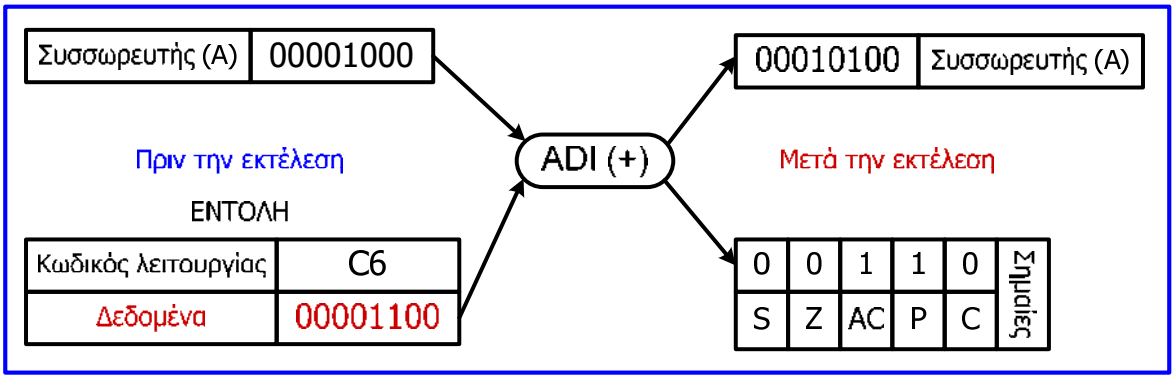
Εντολές διαχείρισης στοίβας του Intel 8085

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

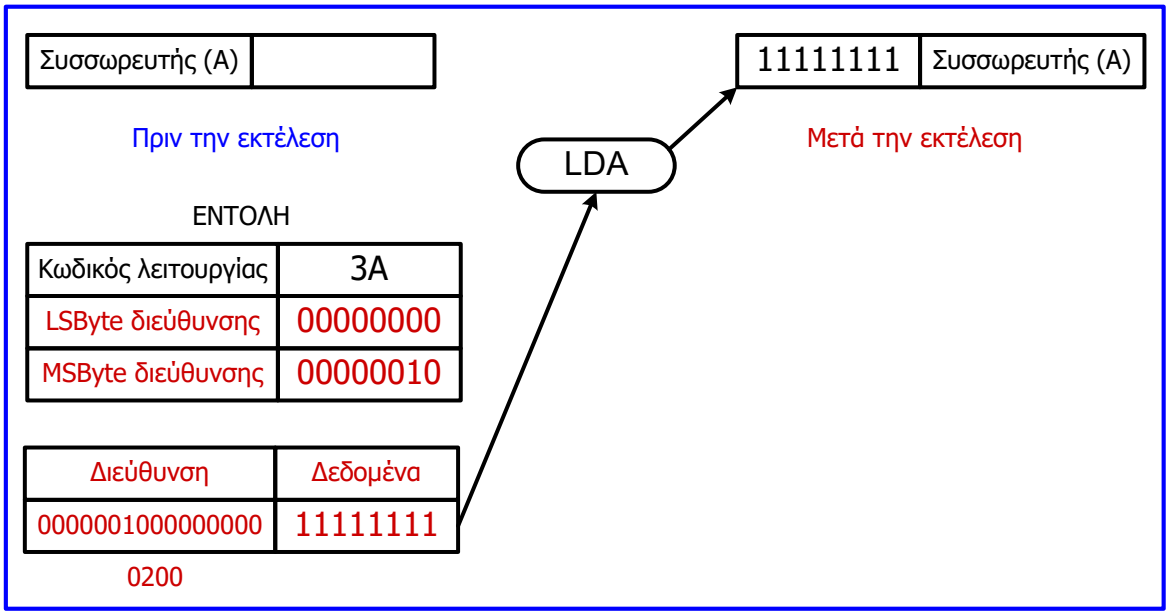
Τρόποι διευθυνσιοδότησης στον Intel 8085

- 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.



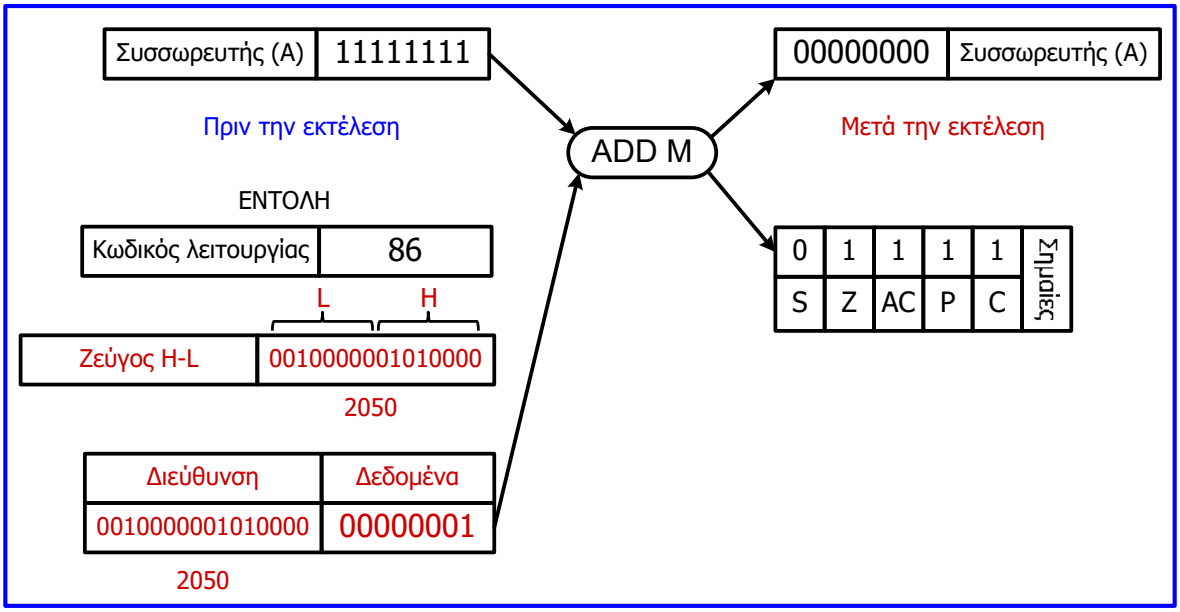
Τρόποι διευθυνσιοδότησης στον Intel 8085

- 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.



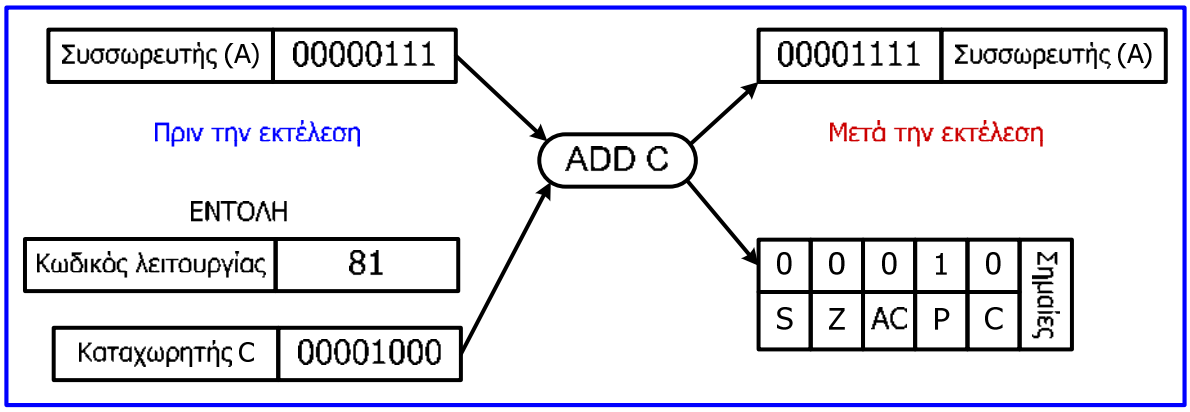
Τρόποι διευθυνσιοδότησης στον Intel 8085

- 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

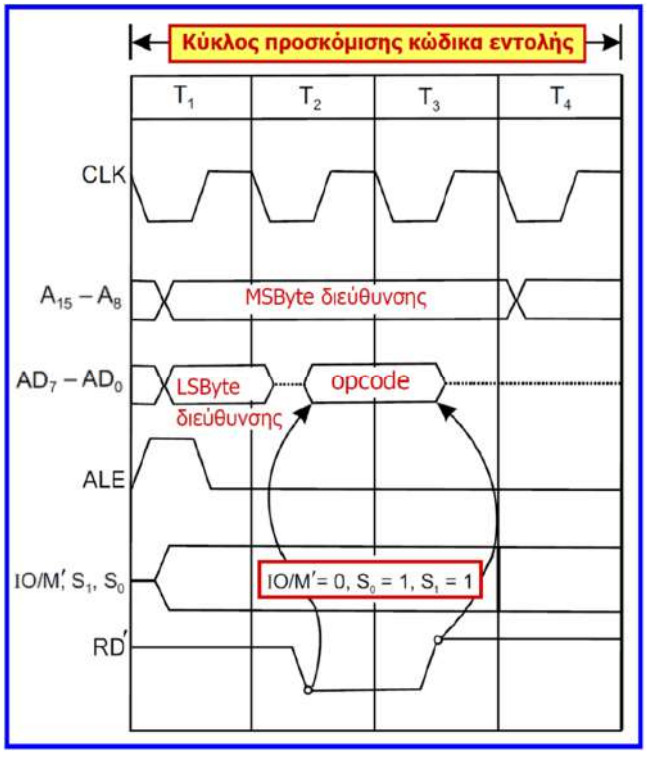
- Ο χρόνος που απαιτείται για την ολοκλήρωση μιας εντολής εξαρτάται από τον τύπο της εντολής, δηλαδή από τις επιμέρους λειτουργίες που περιλαμβάνει. Ο χρόνος αυτός, στον μικροεπεξεργαστή Intel 8085, αποτελείται από κύκλους μηχανής (machine cycles).
- Κάθε κύκλος μηχανής μπορεί να αποτελείται από 3 έως 5 κύκλους ρολογιού.
- 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.

Χρονισμός των εντολών του Intel 8085

- Η εκτέλεση των εντολών με μέγεθος ενός byte (π.χ. ADD B, MOV C,B, RRC), ολοκληρώνεται κατά τη 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 = 3 + 4 = 7 κύκλοι ρολογιού) όταν δεν ακολουθείται η διακλάδωση και 3 κύκλους μηχανής (opcode fetch + 2 × memory read = 4 + 2 × 3 = 10 κύκλοι ρολογιού) όταν ακολουθείται η διακλάδωση.

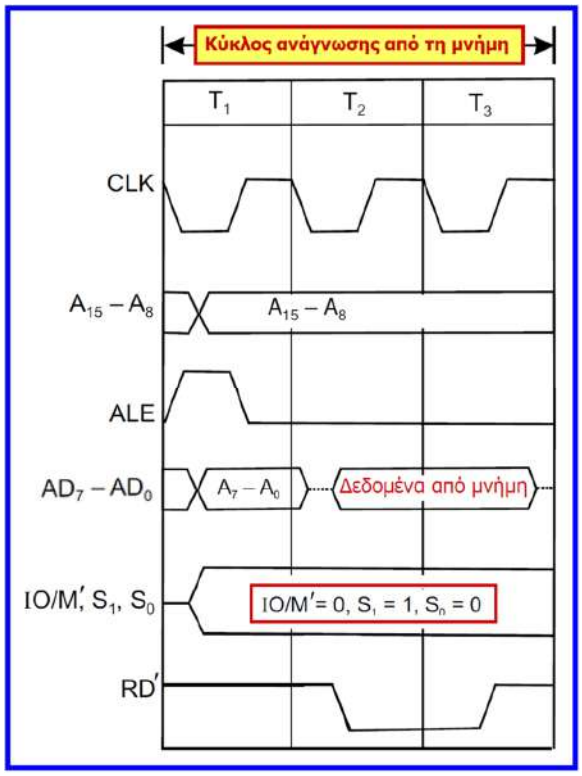
Βασικοί κύκλοι μηχανής του Intel 8085

- 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.



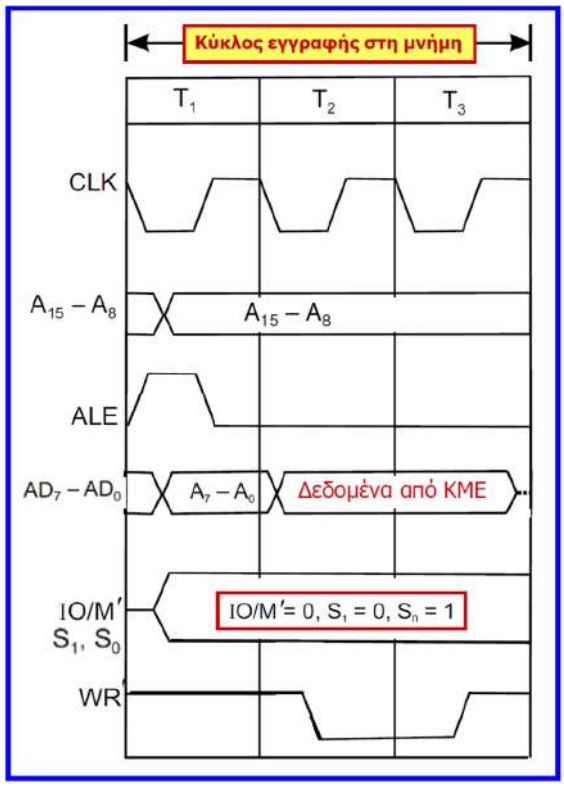
Βασικοί κύκλοι μηχανής του Intel 8085

- 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 E/E.
- 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

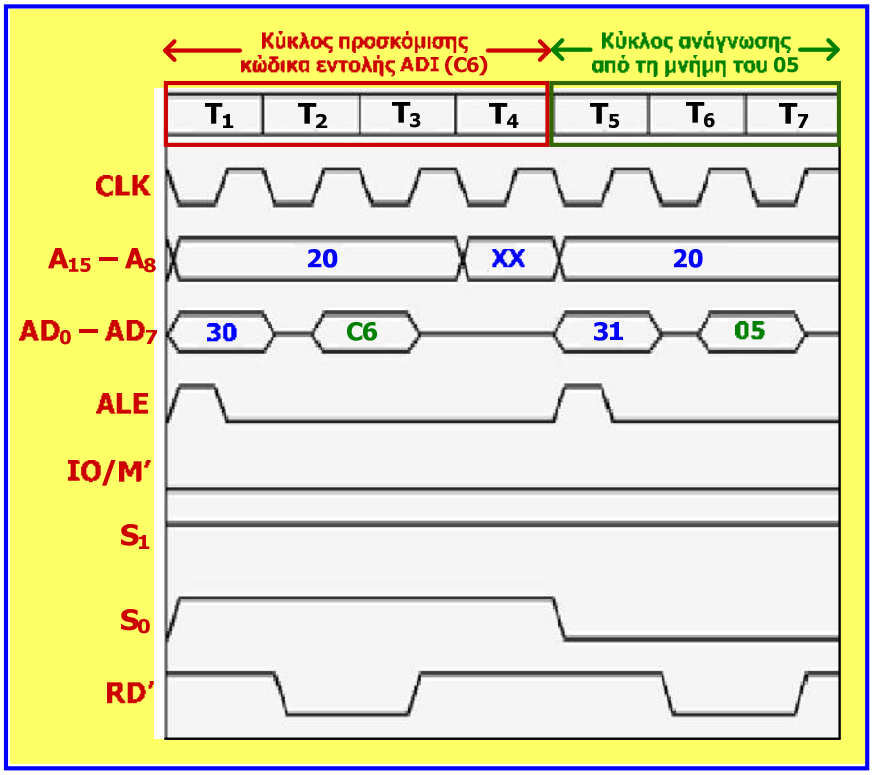
- Στη διάρκεια του κύκλου ρολογιού T_2 ενός κύκλου προσκόμισης κώδικα εντολής ή ενός κύκλου ανάγνωσης, ενεργοποιείται (γίνεται 0) από τον μικροεπεξεργαστή το σήμα RD' και εκτελείται λειτουργία ανάγνωσης κώδικα εντολής ή δεδομένων από τη μνήμη (ή από καταχωρητή μονάδας E/E).
- Στη διάρκεια του κύκλου T_2 ενός κύκλου εγγραφής, ενεργοποιείται το σήμα WR' και εκτελείται λειτουργία εγγραφής δεδομένων στη μνήμη ή σε καταχωρητή μονάδας E/E.
- 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

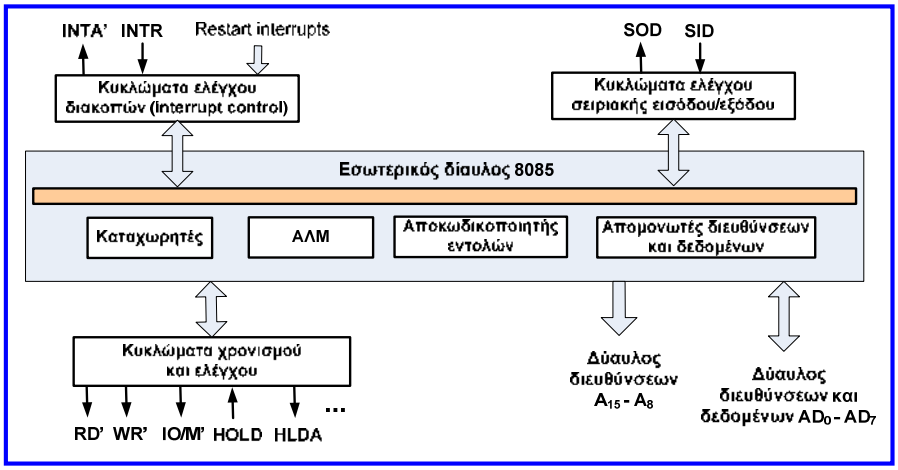
- Εντολή: 2030 ADI 05 (πρόσθεση της τιμής 5 στο περιεχόμενο του συσσωρευτή)
- Για να ολοκληρωθεί η παραπάνω εντολή, χρειάζονται 2 κύκλοι μηχανής: κύκλος προσκόμισης κώδικα εντολής (ADI = C6) που διαρκεί 4 κύκλους ρολογιού και κύκλος ανάγνωσης ορίσματος (05) από τη μνήμη που διαρκεί 3 κύκλους ρολογιού.
- 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



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 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 I/O ή peripheral-mapped I/O).
- Κατά την εκτέλεση των ειδικών εντολών IN και OUT, 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 (bit 7) του συσσωρευτή Α μεταφέρεται στη γραμμή SOD (όταν το bit 6 ισούται με 1), 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) και 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.

Παραδείγματα προγραμματισμού του Intel 8085

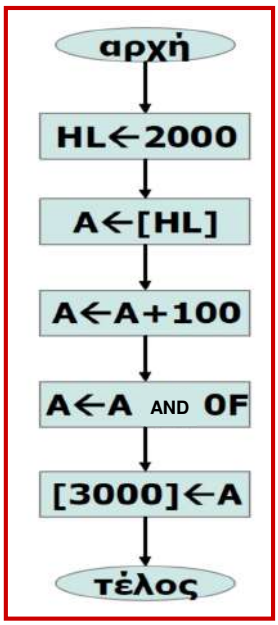
Τα παραδείγματα 1 – 3 προγραμματισμού του επεξεργαστή Intel 8085 που ακολουθούν είναι διαθέσιμα σε βιντεοδιάλεξη διάρκειας 30 λεπτών, η οποία είναι προσβάσιμη στο σύνδεσμο:

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.

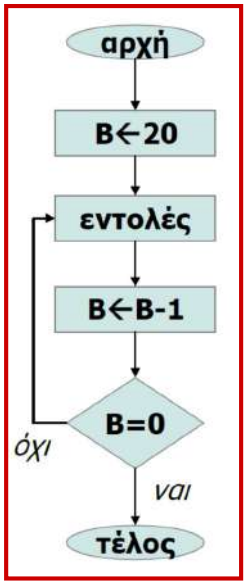
Παράδειγμα 1ο



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

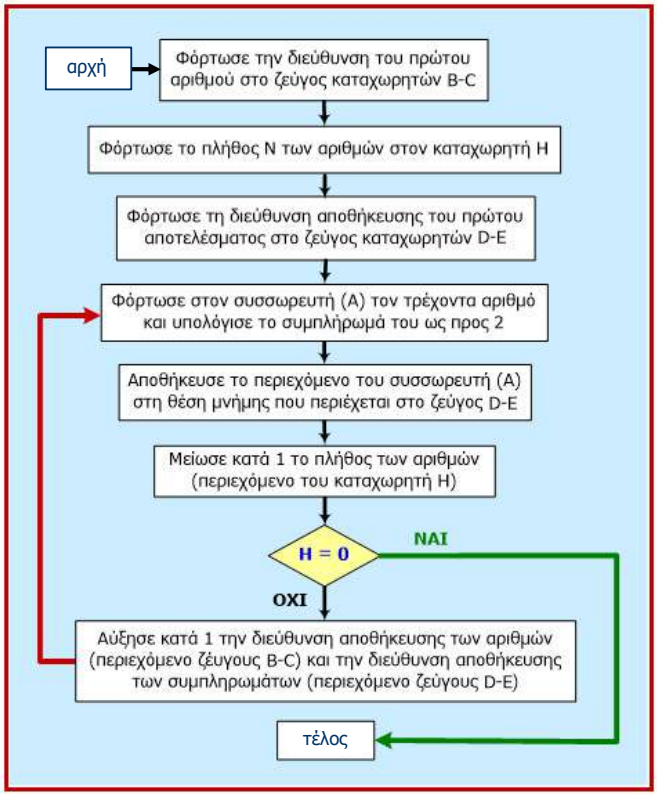


	MOV B,14	ο καταχωρητής B θα είναι ο μετρητής μας ($20_{10} = 00010100_2 = 14_{16}$)
Loop:	… … … …	
	Ακολουθία εντολών … … … …	
	DCR B	μείωση του μετρητή κατά 1
	JNZ Loop	εάν ο μετρητής μας δεν μηδενίστηκε, επιστροφή στην διεύθυνση Loop
	HLT	διαφορετικά τέλος και αδρανοποίηση του μικροεπεξεργαστή

Παράδειγμα 3ο

Σύνταξη προγράμματος υπολογισμού του συμπληρώματος ως προς 2, Ν 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

ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
482
Σημειώσεις διδασκαλίας ΠΛΗ21, Λ. Μπισδούνης

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

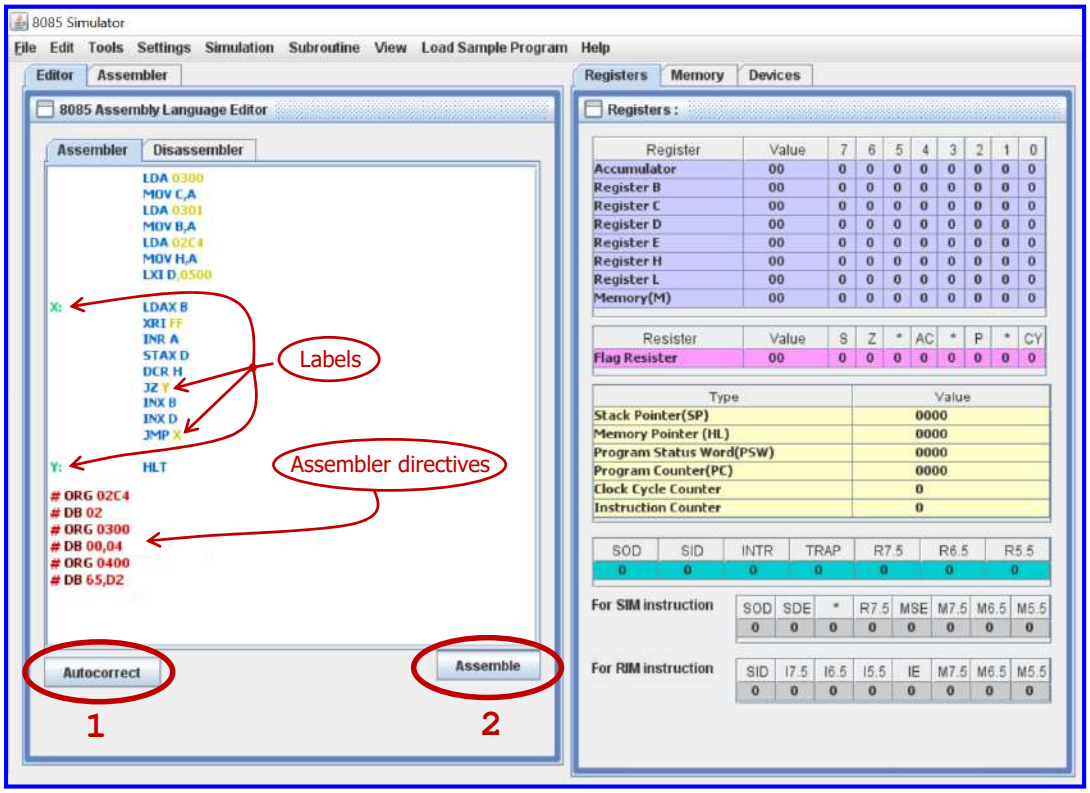
```
# ORG 02C4
# DB 02
# ORG 0300
# DB 00,04
# ORG 0400
# DB 65,D2
```

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.

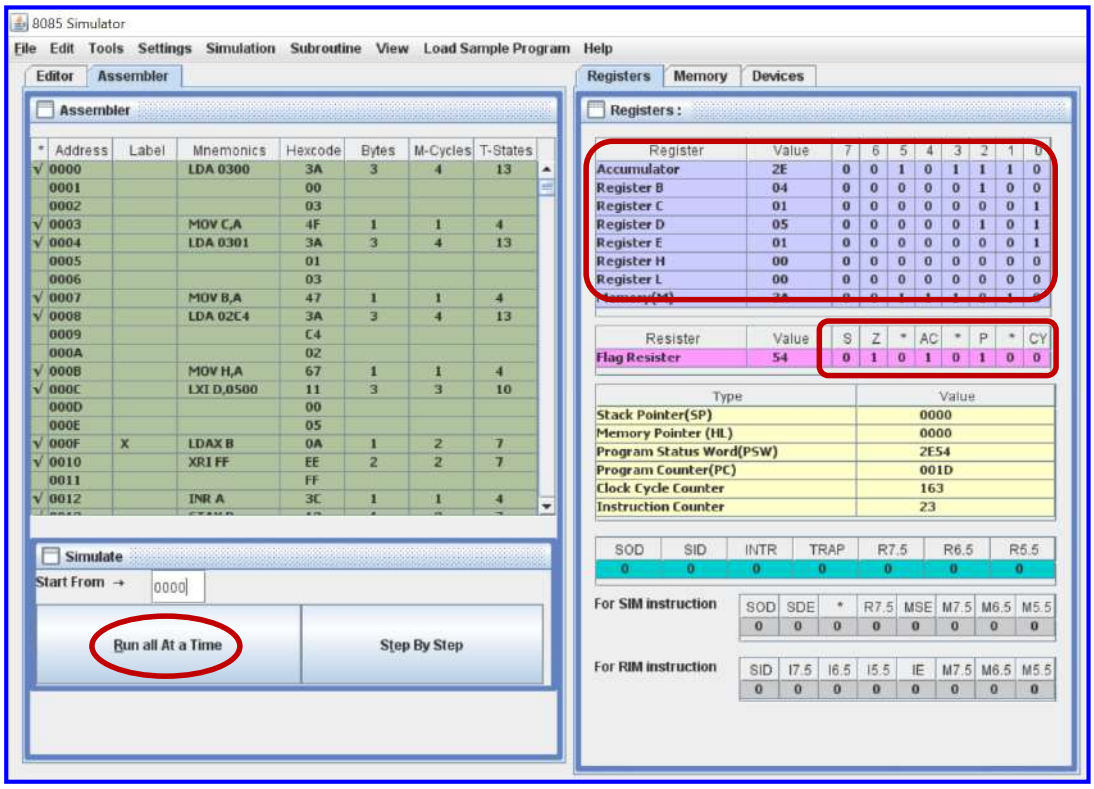
Παράδειγμα 3ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra



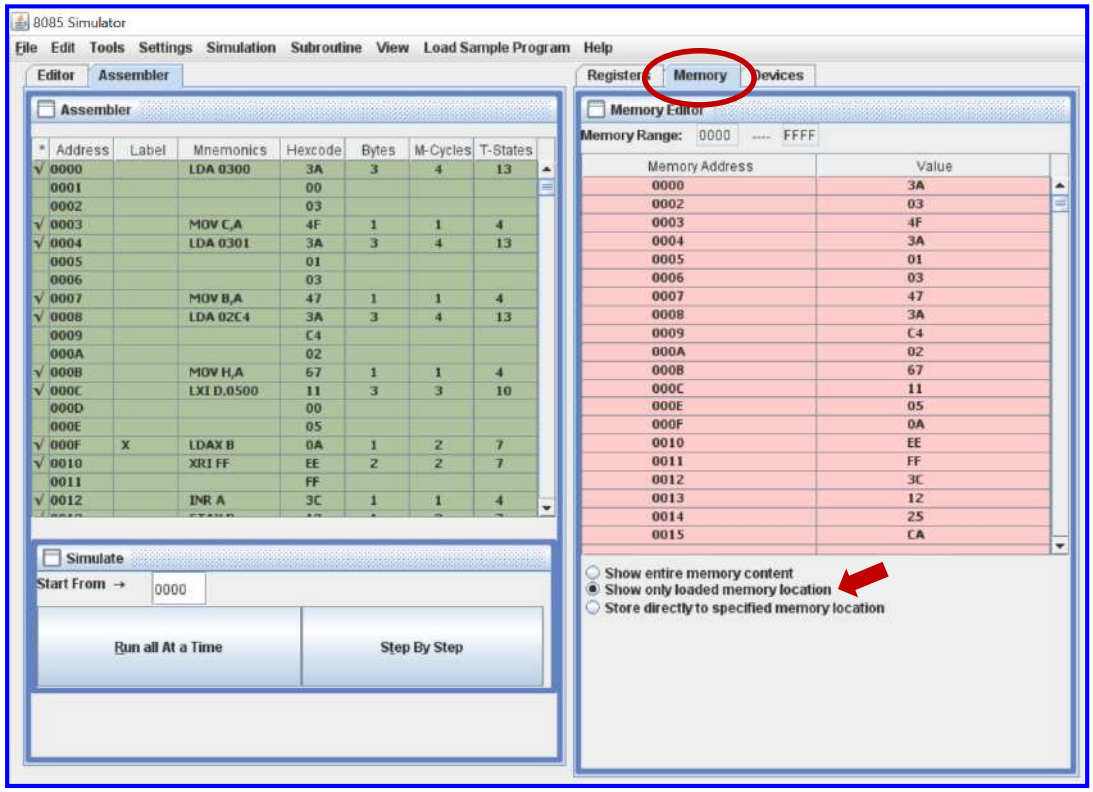
Παράδειγμα 3ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra



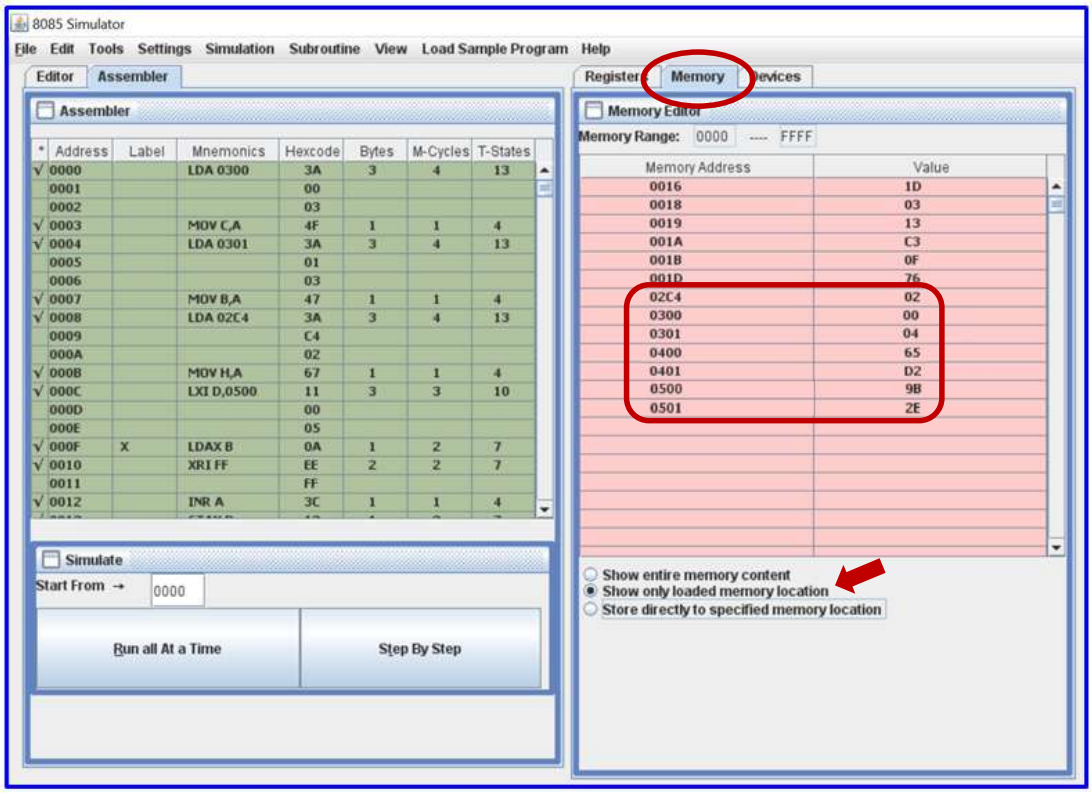
Παράδειγμα 3ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra



Παράδειγμα 3ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra

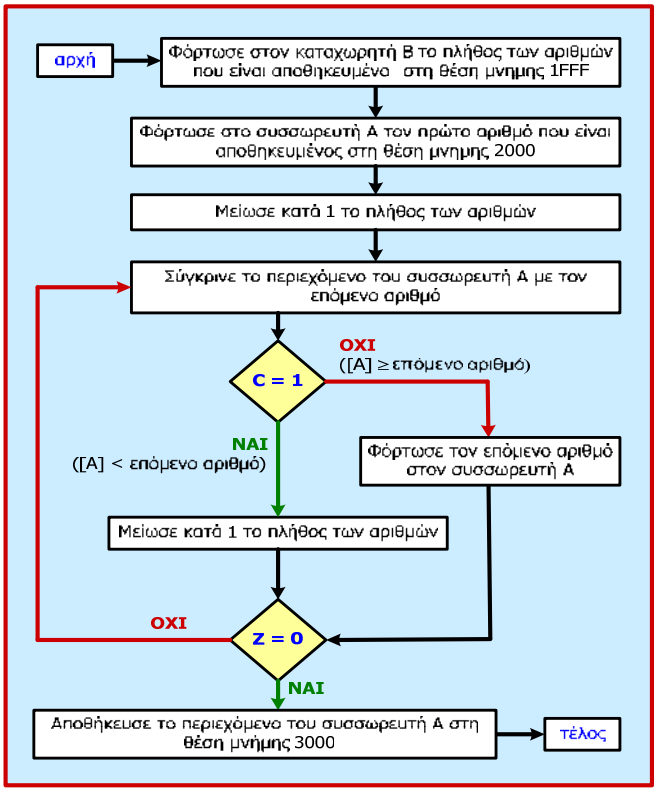


Παράδειγμα 4ο

Σύνταξη προγράμματος υπολογισμού του ελάχιστου μεταξύ Ν θετικών 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 4ο: διάγραμμα ροής

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



Παράδειγμα 4ο: 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

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 X 0 0

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 X 0 0
09	02 XX	XX XX	20 01	0 X 0 0

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

$A = 09_{16}$
$09_{16} - 05_{16} = 04_{16}$
$= 00000100$
$09 \geq 05 \Rightarrow C = 0$

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 X 0 0
09	02 XX	XX XX	20 01	0 X 0 0
09	02 XX	XX XX	20 01	0 0 0 0

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
05	02 XX	XX XX	20 01	0 0 0 0

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
05	02 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 01	0 0 0 0

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
05	02 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 01	0 0 0 0

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
05	02 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 02	0 0 0 0

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

$A = 05_{16}$
$05_{16} - 71_{16} =$
$00000101 - 01110001$
$00000101 + 10001111 =$
10010100
$05 < 71 \Rightarrow C = 1$

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
05	02 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 02	0 0 0 0
05	01 XX	XX XX	20 02	0 1 1 0

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
05	02 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 02	0 0 0 0
05	01 XX	XX XX	20 02	0 1 1 0

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
05	02 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 02	0 0 0 0
05	01 XX	XX XX	20 02	0 1 1 0
05	00 XX	XX XX	20 02	1 1 0 1

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	XX

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
05	02 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 02	0 0 0 0
05	01 XX	XX XX	20 02	0 1 1 0
05	00 XX	XX XX	20 02	1 1 0 1

Παράδειγμα 4ο: εκτέλεση

```
0000  LXI H,1FFF
0003  MOV B,M
0004  INX H
0005  MOV A,M
0006  DCR B
0007  INX H
0008  CMP M
0009  JC 000D
000C  MOV A,M
000D  DCR B
000E  JNZ 0007
0011  STA 3000
0014  HLT
```

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71
3000	05

Καταχωρητές				Σημαίες
A	B C	D E	H L	Z CY S P
XX	XX XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	1F FF	X X X X
XX	03 XX	XX XX	20 00	X X X X
09	03 XX	XX XX	20 00	X X X X
09	02 XX	XX XX	20 00	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
09	02 XX	XX XX	20 01	0 0 0 0
05	02 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 01	0 0 0 0
05	01 XX	XX XX	20 02	0 0 0 0
05	01 XX	XX XX	20 02	0 1 1 0
05	00 XX	XX XX	20 02	1 1 0 1
05	00 XX	XX XX	20 02	1 1 0 1

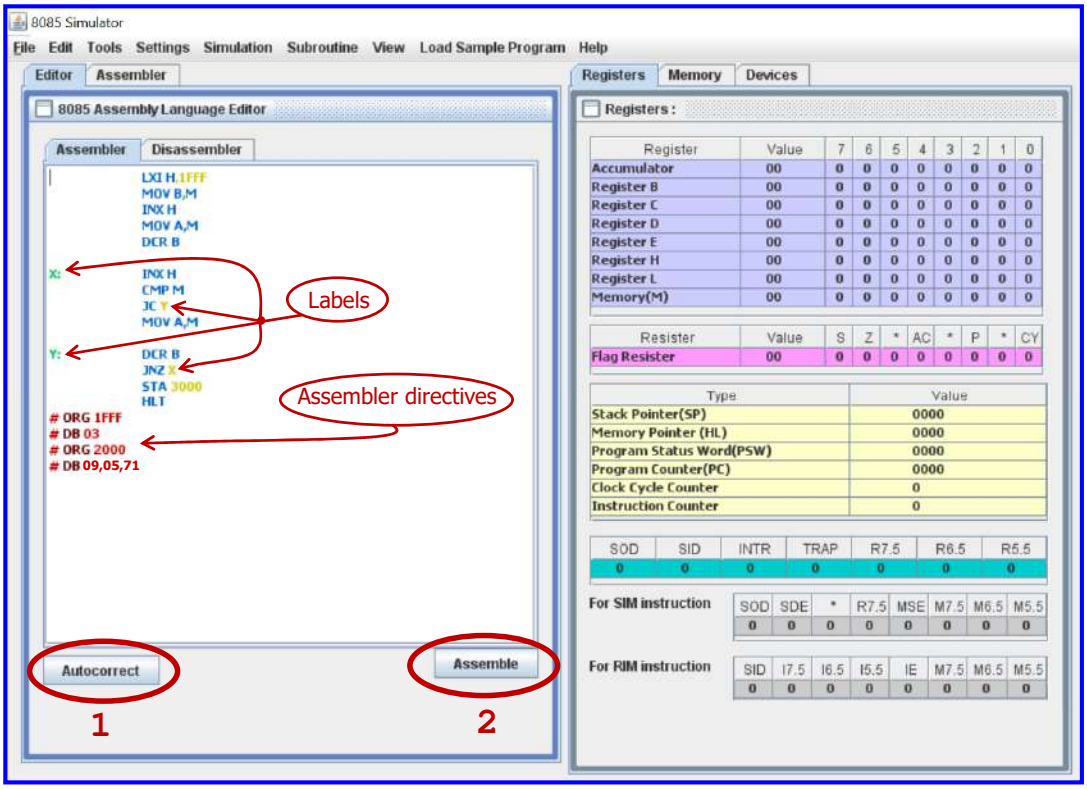
Παράδειγμα 4ο: οδηγίες στον συμβολομεταφραστή

Μνήμη	
Διεύθυνση	Περιε-χόμενο
1FFF	03
2000	09
2001	05
2002	71

```
# ORG 1FFF
# DB 03
# ORG 2000
# DB 09,05,71
```

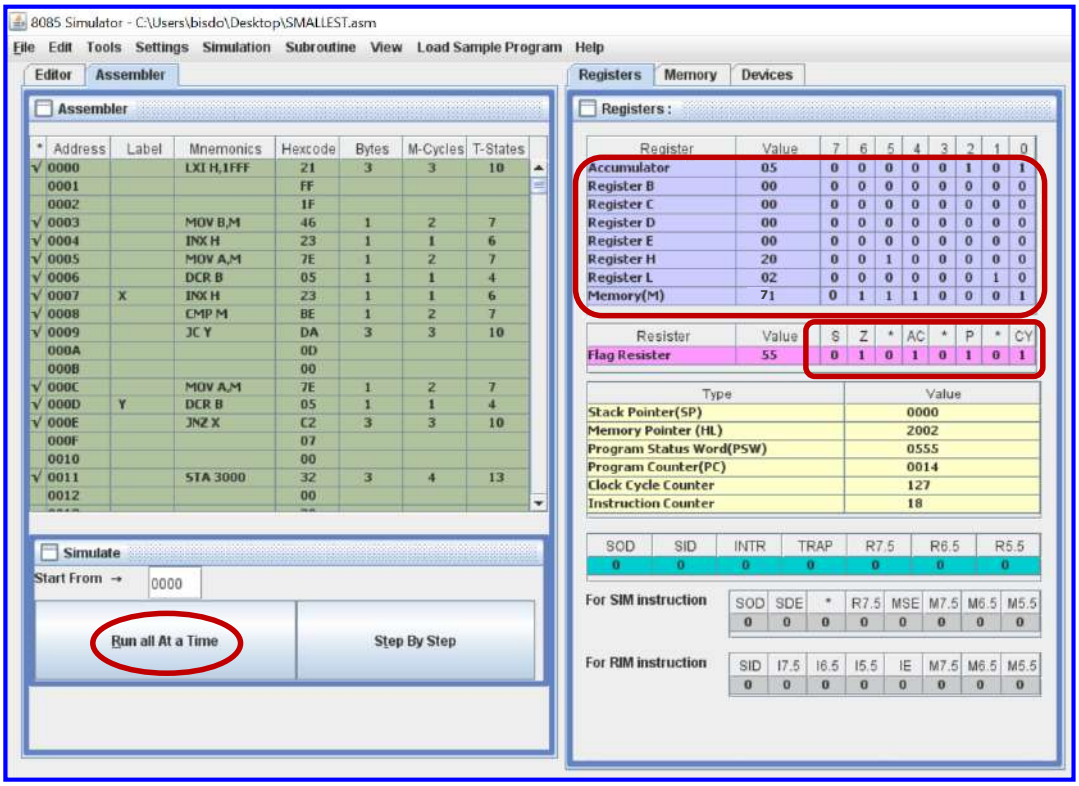
Παράδειγμα 4ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra



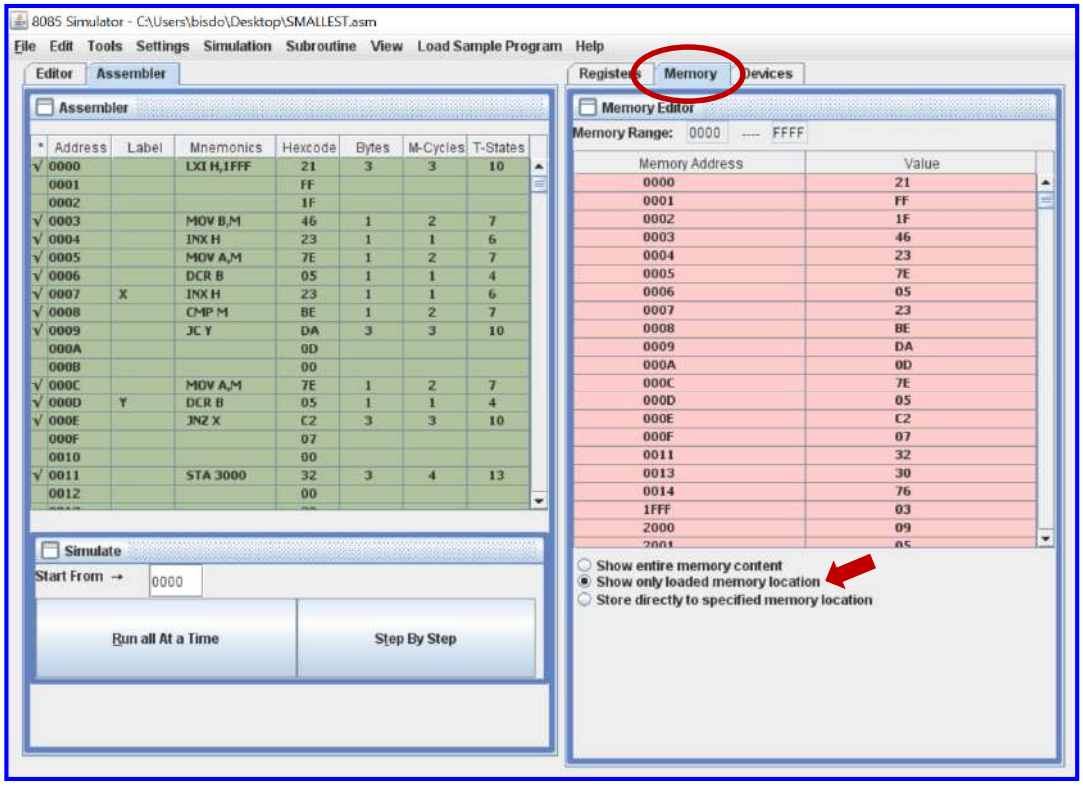
Παράδειγμα 4ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra



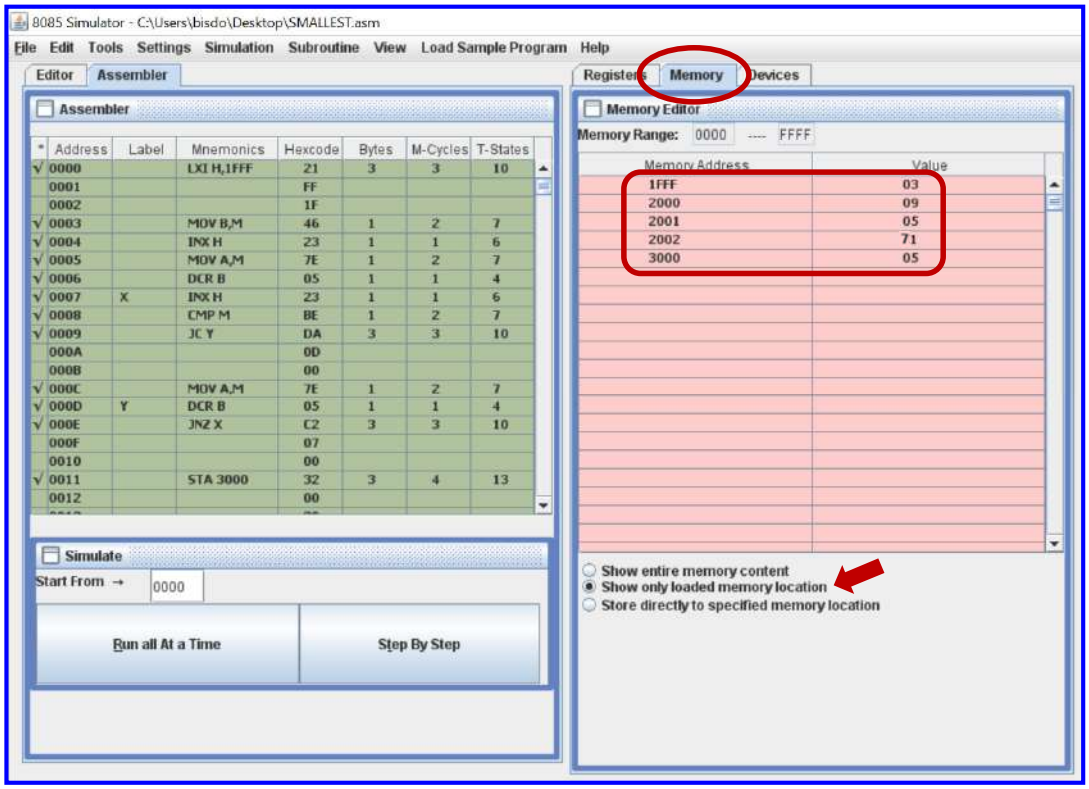
Παράδειγμα 4ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra

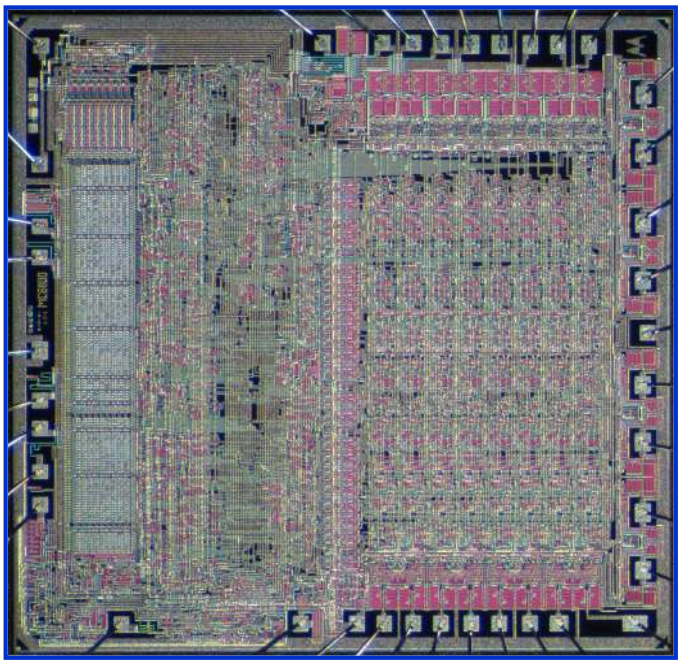


Παράδειγμα 4ο: προσομοίωση

Jubin's 8085 Simulator
© Jubin Mitra

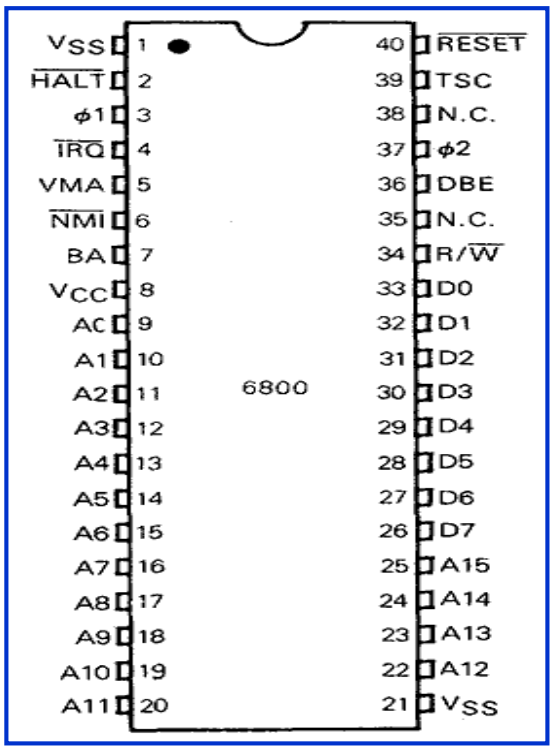


Μικροεπεξεργαστής Motorola 6800



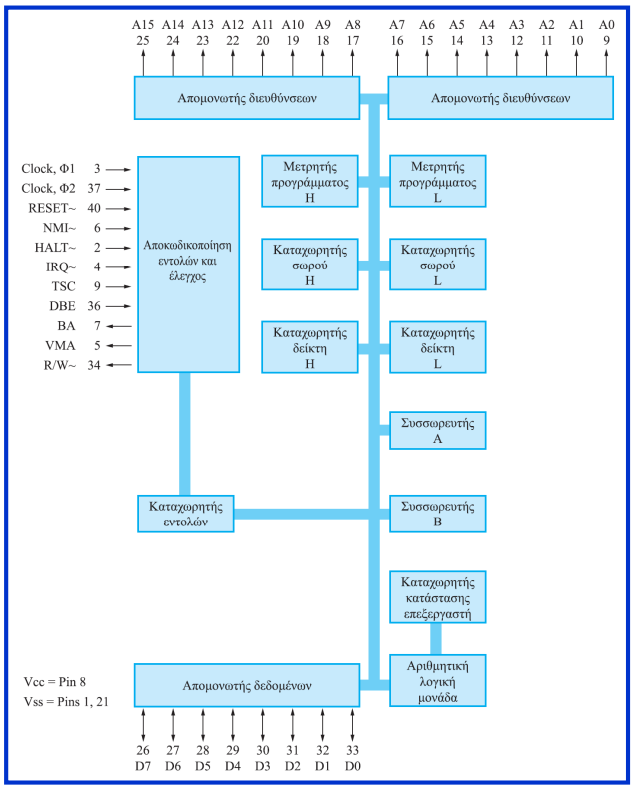
1974

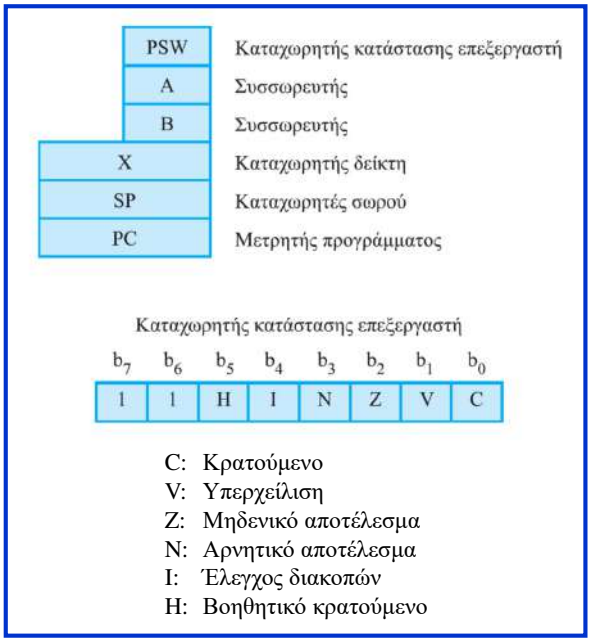
Ακροδέκτες και σήματα του Motorola 6800



Ακροδέκτης		*Περιγραφή*
A_0–A_{15}	(OUT)	Γραμμές διευθύνσεων
D_0–D_7	(BI)	Γραμμές δεδομένων
$\overline{HALT}$	(IN)	Σήμα στάσης επεξεργασίας
TSC	(IN)	Σήμα ελέγχου δίαυλου διευθύνσεων
R/$\overline{W}$	(OUT)	Σήμα ανάγνωσης/εγγραφής
VMA	(OUT)	Σήμα έγκυρης διεύθυνσης
DBE	(IN)	Σήμα ελέγχου δίαυλου δεδομένων
BA	(OUT)	Σήμα κατάστασης διαύλων
$\overline{IRQ}$	(IN)	Σήμα ενεργοποίησης διακοπής IRQ
$\overline{RESET}$	(IN)	Σήμα αρχικοποίησης
$\overline{NMI}$	(IN)	Ενεργοποίηση διακοπής NMI
Φ_1, Φ_2	(IN)	Σήματα χρονισμού
V_{SS}, V_{CC}	(IN)	Σήματα τροφοδοσίας και γείωσης

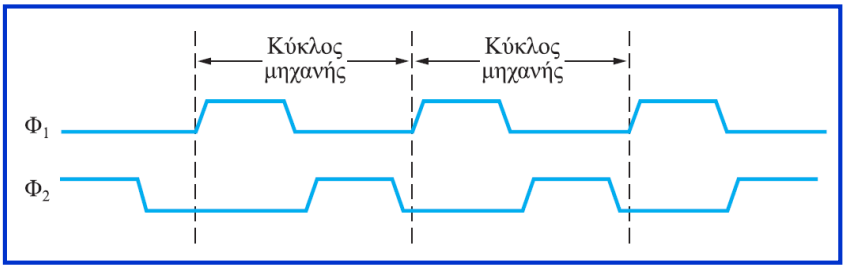
Δομή και καταχωρητές του Motorola 6800





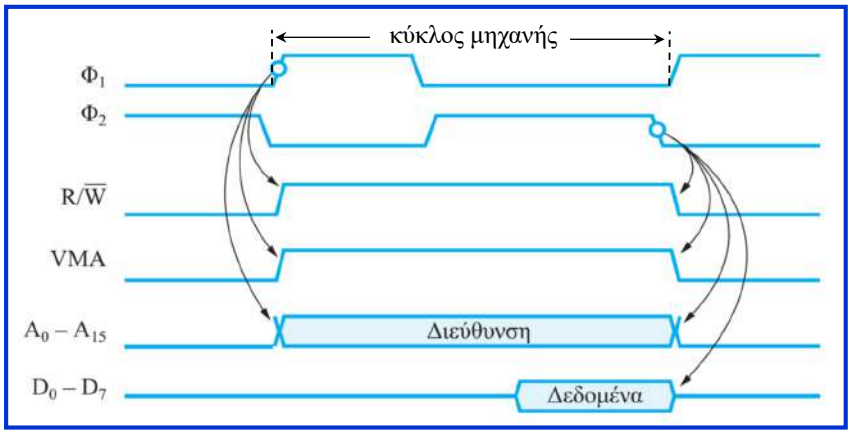
Χρονισμός εντολών στον Motorola 6800

- Ο 6800 χρησιμοποιεί δύο συμπληρωματικά σήματα ρολογιού (Φ_1, Φ_2) για το χρονισμό των λειτουργιών στο μικροϋπολογιστικό σύστημα (όταν Φ1 = 1 τότε Φ2 = 0 και αντίστροφα).



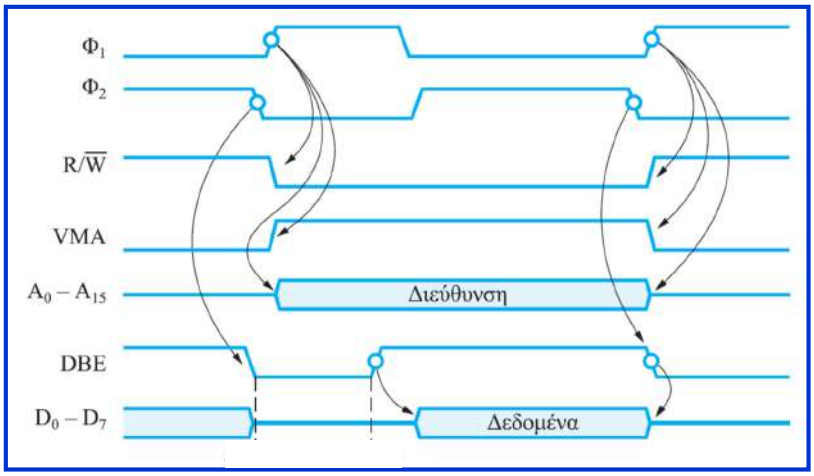
- Οι εντολές του Motorola 6800 απαιτούν από 2 έως 8 κύκλους μηχανής για να εκτελεσθούν.
- Υπάρχουν 3 τύποι κύκλων μηχανής που μπορούν να λάβουν χώρα κατά την εκτέλεση μιας εντολής:
  - ✓ Κύκλος ανάγνωσης: ένα byte δεδομένων μεταφέρεται από τη μνήμη ή μία μονάδα εισόδου/ εξόδου προς το μικροεπεξεργαστή.
  - ✓ Κύκλος εγγραφής: ένα byte δεδομένων μεταφέρεται από το μικροεπεξεργαστή προς τη μνήμη ή μία μονάδα εισόδου/εξόδου.
  - ✓ Κύκλος εσωτερικής λειτουργίας: αφορά τη διενέργεια εσωτερικών λειτουργιών του μικροεπεξεργαστή, χωρίς δραστηριότητα στους διαύλους του συστήματος.

Κύκλος ανάγνωσης στον Motorola 6800



- Κατά τη μετάβαση του Φ1 στην τιμή 1, το σήμα R/W′ μεταβαίνει στην τιμή 1 υποδεικνύοντας λειτουργία ανάγνωσης από τη μνήμη και το σήμα VMA μεταβαίνει στην τιμή 1, δηλώνοντας ότι η διεύθυνση όπου βρίσκονται τα δεδομένα, έχει τοποθετηθεί στον δίαυλο διευθύνσεων.
- Τα σήματα R/W′, VMA παραμένουν σε τιμή 1 μέχρι το τέλος του κύκλου μηχανής.
- Η μνήμη τοποθετεί τα δεδομένα στον δίαυλο δεδομένων πριν το τέλος του κύκλου μηχανής και ο μικροεπεξεργαστής μεταφέρει τα δεδομένα στους εσωτερικούς καταχωρητές του με την κατερχόμενη ακμή του Φ2.

Κύκλος εγγραφής στον Motorola 6800



- Κατά τη μετάβαση του Φ1 στην τιμή 1, το σήμα R/W′ μεταβαίνει στην τιμή 0 υποδεικνύοντας λειτουργία εγγραφής στη μνήμη ή σε μονάδα εισόδου/εξόδου και το σήμα VMA μεταβαίνει στην τιμή 1, δηλώνοντας ότι η διεύθυνση όπου θα αποθηκευθούν τα δεδομένα, έχει τοποθετηθεί στον δίαυλο διευθύνσεων.
- Όταν το σήμα DBE μεταβεί στην τιμή 1, τα δεδομένα τοποθετούνται στον δίαυλο δεδομένων από τον μικροεπεξεργαστή και παραμένουν μέχρι το τέλος του κύκλου.
- Στο χρόνο αυτό, τα δεδομένα αποθηκεύονται στην μνήμη ή σε μονάδα εισόδου/εξόδου.

Εντολές και διευθυνσιοδότηση στον Motorola 6800

- Οι εντολές του Motorola 6800 αποθηκεύονται σε bytes στην κύρια μνήμη και αποτελούνται από 1, 2 ή 3 bytes.
- Η Motorola ομαδοποιεί τις εντολές του 6800 σε 4 κατηγορίες:
  - ✓ Εντολές συσσωρευτή και μνήμης: αριθμητικές και λογικές εντολές, εντολές ελέγχου και μεταφοράς δεδομένων
  - ✓ Εντολές καταχωρητή δείκτη και δείκτη σωρού: εντολές αύξησης ή ελάττωσης, φόρτωσης ή αποθήκευσης και μεταφοράς που χρησιμοποιούν τον καταχωρητή δείκτη και τον δείκτη σωρού.
  - ✓ Εντολές αλμάτων και διακλαδώσεων που χρησιμοποιούνται για τον έλεγχο της μεταφοράς της εκτέλεσης του προγράμματος από ένα σημείο σε κάποιο άλλο. Όλες οι εντολές διακλάδωσης χρησιμοποιούν σχετική διευθυνσιοδότηση.
  - ✓ Εντολές καταχωρητή κατάστασης επεξεργαστή με τις οποίες γίνεται δυνατή η απευθείας διαχείριση των σημαιών του καταχωρητή κατάστασης επεξεργαστή.
- Τρόποι διευθυνσιοδότησης στον Motorola 6800: διευθυνσιοδότηση καταχωρητή, άμεση διευθυνσιοδότηση, διευθυνσιοδότηση μηδενικής σελίδας, απευθείας διευθυνσιοδότηση, δεικτοδοτημένη διευθυνσιοδότηση και σχετική διευθυνσιοδότηση.

Σύγκριση του Motorola 6800 με τον Intel 8085

- Ο Motorola 6800 έχει απλούστερο χρονισμό εντολών από τον Intel 8085.
- Ο κύκλος ρολογιού του Motorola 6800 ταυτίζεται με τον κύκλο μηχανής, ενώ στον Intel 8085 κάθε κύκλος μηχανής αποτελείται από 3 έως 5 κύκλους ρολογιού.
- Ο Intel 8085 ειδικές εντολές για προσπέλαση μονάδων εισόδου/εξόδου, ενώ στον Motorola 6800 οι μονάδες εισόδου-εξόδου είναι προσπελάσιμες μόνο ως θέσεις μνήμης.
- Ο Motorola 6800 έχει απλούστερο σύνολο σημάτων ελέγχου και δεν πολυπλέκει τον δίαυλο δεδομένων με των διευθύνσεων, όπως γίνεται στον Intel 8085 για λιγότερους ακροδέκτες.
- Ο Motorola 6800 διαθέτει λιγότερους βασικούς τύπους εντολών και περισσότερους τρόπους διευθυνσιοδότησης από τον Intel 8085.
- Το σύνολο εντολών του Motorola 6800 απαιτεί εκτεταμένη χρήση της μνήμης, λόγω των λιγότερων καταχωρητών και της έλλειψης δυνατότητας μετακίνησης δεδομένων μεταξύ των καταχωρητών του.
- Το κύκλωμα χρονισμού στον Intel 8085 είναι ενσωματωμένο στο ίδιο ολοκληρωμένο κύκλωμα, ενώ στον Motorola 6800 απαιτείται εξωτερικό κύκλωμα χρονισμού.
- Για τον Intel 8085 έχουν αναπτυχθεί αρκετά κυκλώματα υποστήριξης για την υλοποίηση μικροϋπολογιστικών συστημάτων, κάτι που δεν ισχύει για τον Motorola 6800.
- Ο Intel 8085 έχει δυνατότητα σειριακής λήψης και μετάδοσης δεδομένων.

Μικροϋπολογιστικά συστήματα

- Ένα τυπικό μικροϋπολογιστικό σύστημα αποτελείται από μικροεπεξεργαστή, μνήμη ROM, στην οποία είναι αποθηκευμένο το πρόγραμμα που εκτελείται στον μικροεπεξεργαστή, μνήμη RAM για την αποθήκευση των προσωρινών δεδομένων και περιφερειακές μονάδες εισόδου/εξόδου, όπως πληκτρολόγιο, μονάδα απεικόνισης δεδομένων, μονάδα δίσκου κ.ά.

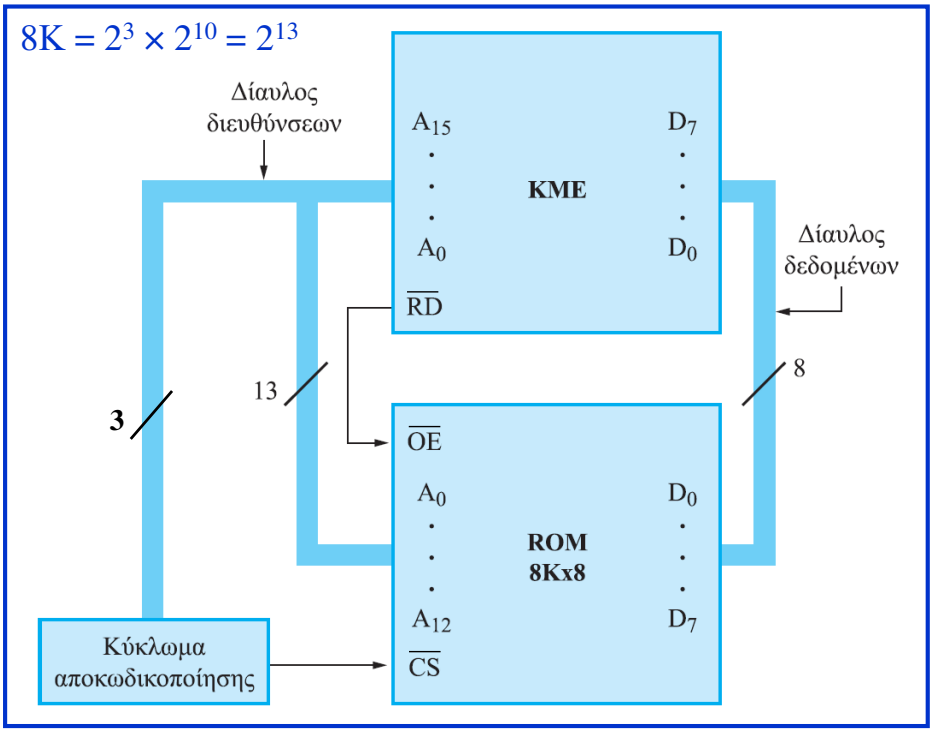
- Οι κατασκευαστές μικροεπεξεργαστών παρέχουν συνήθως ολοκληρωμένα κυκλώματα ειδικού σκοπού, τα οποία υποστηρίζουν τη διασύνδεση των μικροεπεξεργαστών με μονάδες εισόδου-εξόδου (κυκλώματα διασύνδεσης και επιλογής μονάδων εισόδου/εξόδου, κυκλώματα απόδοσης προτεραιότητας σε μονάδες εισόδου/εξόδου, ελεγκτές DMA κ.ά.) και αναφέρονται ως κυκλώματα διασύνδεσης (interface circuits).

- Τα τμήματα ενός τυπικού μικροϋπολογιστικού συστήματος διασυνδέονται μεταξύ τους μέσω τριών βασικών διαύλων ή αρτηριών:
  - ✓ δίαυλος διευθύνσεων για τη μεταφορά των διευθύνσεων μνήμης ή μονάδων εισόδου/εξόδου,
  - ✓ δίαυλος δεδομένων για τη μεταφορά των δεδομένων από και προς τον μικροεπεξεργαστή και
  - ✓ δίαυλος ελέγχου για την διακίνηση των διάφορων σημάτων ελέγχου (χρονισμού) του συστήματος.

Διασύνδεση μνημών ROM

- Τα περιεχόμενα των μνημών ROM (read only memories) είναι μόνιμα αποθηκευμένα και δεν είναι δυνατή η διαγραφή τους ή η μεταβολή τους.
- Χρησιμοποιούνται συνήθως για την αποθήκευση των προγραμμάτων που εκτελούνται στον μικροεπεξεργαστή.
- Οι μνήμες ROM είναι ολοκληρωμένα κυκλώματα που περιλαμβάνουν σήματα διευθύνσεων, σήματα δεδομένων, σήμα ενεργοποίησης (CS′, chip select, το οποίο όταν είναι ενεργό, δηλαδή σε λογική τιμή 0 ενεργοποιείται η μνήμη) και σήμα επίτρεψης εξόδου (OE′, output enable, το οποίο όταν είναι ενεργό, δηλαδή σε λογική τιμή 0 τα δεδομένα μπορούν να τοποθετηθούν στο δίαυλο δεδομένων).
- Κατά τη διασύνδεση μνημών ROM με μικροεπεξεργαστή, τα λιγότερο σημαντικά ψηφία του διαύλου διευθύνσεων του μικροεπεξεργαστή συνδέονται με τα αντίστοιχα σήματα διευθύνσεων της μνήμης.
- Τα σήματα δεδομένων της μνήμης συνδέονται με τα αντίστοιχα ψηφία του διαύλου δεδομένων του μικροεπεξεργαστή.
- Το σήμα OE′ συνδέεται στο σήμα ανάγνωσης δεδομένων (RD′) του μικροεπεξεργαστή.
- Το σήμα CS′ δημιουργείται από τα περισσότερο σημαντικά σήματα του διαύλου διευθύνσεων με τη βοήθεια κατάλληλου κυκλώματος αποκωδικοποίησης διευθύνσεων.

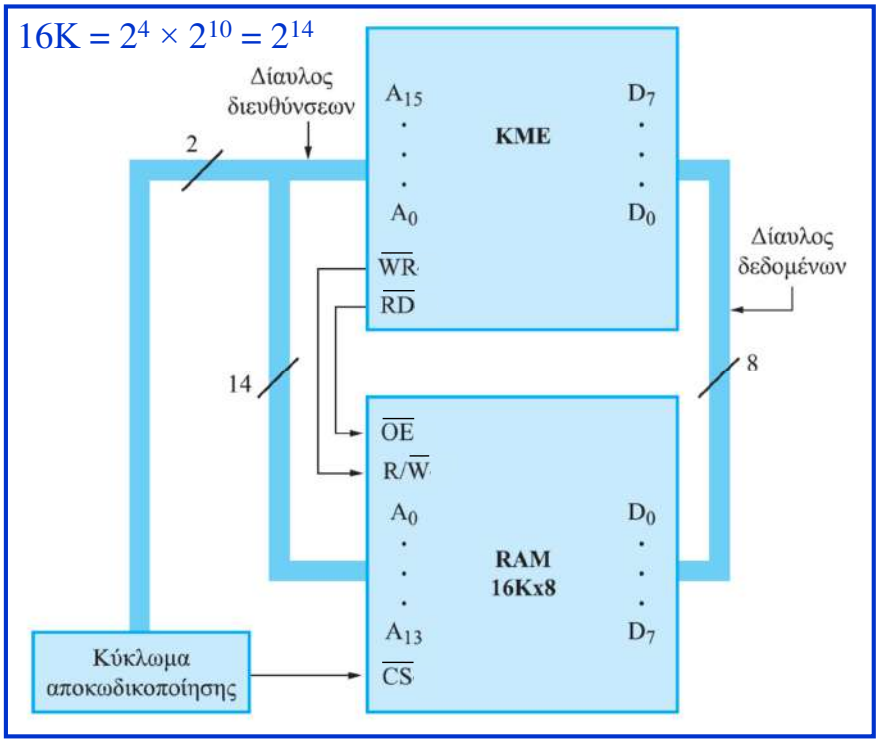
Διασύνδεση μνημών ROM



Διασύνδεση μνημών RAM

- Στις μνήμες RAM (random access memories), εκτός από την ανάγνωση είναι δυνατή και η εγγραφή δεδομένων.
- Χρησιμοποιούνται από τα προγράμματα για την ανάγνωση και αποθήκευση προσωρινών δεδομένων.
- Εκτός από τα σήματα διευθύνσεων, δεδομένων, CS′, OE′, οι μνήμες RAM περιλαμβάνουν το σήμα ανάγνωσης/εγγραφής (R/W′), το οποίο καθορίζει εάν πρόκειται για λειτουργία ανάγνωσης ή εγγραφής.
- Κατά τη διασύνδεση μνημών RAM με μικροεπεξεργαστή, ακολουθούνται οι ίδιοι κανόνες με εκείνους της διασύνδεσης μνημών ROM.
- Το σήμα εγγραφής μνήμης του μικροεπεξεργαστή (WR′ ή MEMW′) οδηγεί το σήμα ανάγνωσης/εγγραφής (R/W′) των μνημών RAM. Όταν R/W′ = 0 επιλέγεται λειτουργία εγγραφής, ενώ όταν R/W′ = 1 επιλέγεται λειτουργία ανάγνωσης.
- Όταν ο μικροεπεξεργαστής επιθυμεί την ανάγνωση ή εγγραφή δεδομένων στη μνήμη, τα λιγότερο σημαντικά ψηφία της διεύθυνσης καθορίζουν τη θέση μνήμης που θα γίνει η ανάγνωση ή η εγγραφή.
- Τα περισσότερο σημαντικά ψηφία της διεύθυνσης δημιουργούν το σήμα CS′, μέσω κατάλληλου κυκλώματος αποκωδικοποίησης διευθύνσεων.

Διασύνδεση μνημών RAM



Παράδειγμα δημιουργίας σημάτων επιλογής μνημών

- Σε μικροϋπολογιστικό σύστημα με δίαυλο διευθύνσεων 16 ψηφίων, επιθυμούμε την κάλυψη της περιοχής διευθύνσεων από 0000 έως 57FF με 3 ολοκληρωμένα κυκλώματα (ΟΚ) μνημών ROM1 16K, RAM1 4K και RAM2 2K.
- Για τη δημιουργία σημάτων επιλογής μνημών, αρχικά καταστρώνουμε το χάρτη της μνήμης του συστήματος, με βάση τις ζητούμενες θέσεις μνήμης και το μέγεθος των ολοκληρωμένων κυκλωμάτων μνημών που συμμετέχουν στο σύστημα.

Μνήμη	Μέγεθος (bytes)	Ψηφία (N) Διεύθυνσης	Πεδίο Διευθύνσεων	Διευθύνσεις σε δυαδική μορφή $A_{15}\ A_{14}\ \ldots\ A_2\ A_1\ A_0$
ROM1	16K	14 ($16K = 2^{14}$)	0000 – 3FFF	0000 0000 0000 0000 0011 1111 1111 1111
RAM1	4K	12 ($4K = 2^{12}$)	4000 – 4FFF	0100 0000 0000 0000 0100 1111 1111 1111
RAM2	2K	11 ($2K = 2^{11}$)	5000 – 57FF	0101 0000 0000 0000 0101 0111 1111 1111

Για τον υπολογισμό της τελευταίας διεύθυνσης κάθε ΟΚ, προσθέτουμε 2^N-1 στην πρώτη διεύθυνση, δηλαδή τον αριθμό με τα N λιγότερο σημαντικά ψηφία 1 και τα υπόλοιπα ψηφία 0.

- Στο παράδειγμα, το σήμα επιλογής κάθε ΟΚ μνήμης προκύπτει από τα δυαδικά ψηφία των διευθύνσεων του ΟΚ που έχουν την ίδια τιμή στον χάρτη του συστήματος μνήμης.

$$CS'_{ROM1} = (A'_{15} \cdot A'_{14})'$$
$$CS'_{RAM1} = (A'_{15} \cdot A_{14} \cdot A'_{13} \cdot A'_{12})'$$
$$CS'_{RAM2} = (A'_{15} \cdot A_{14} \cdot A'_{13} \cdot A_{12} \cdot A'_{11})'$$

Παράδειγμα διασύνδεσης μνήμης στον Motorola 6800

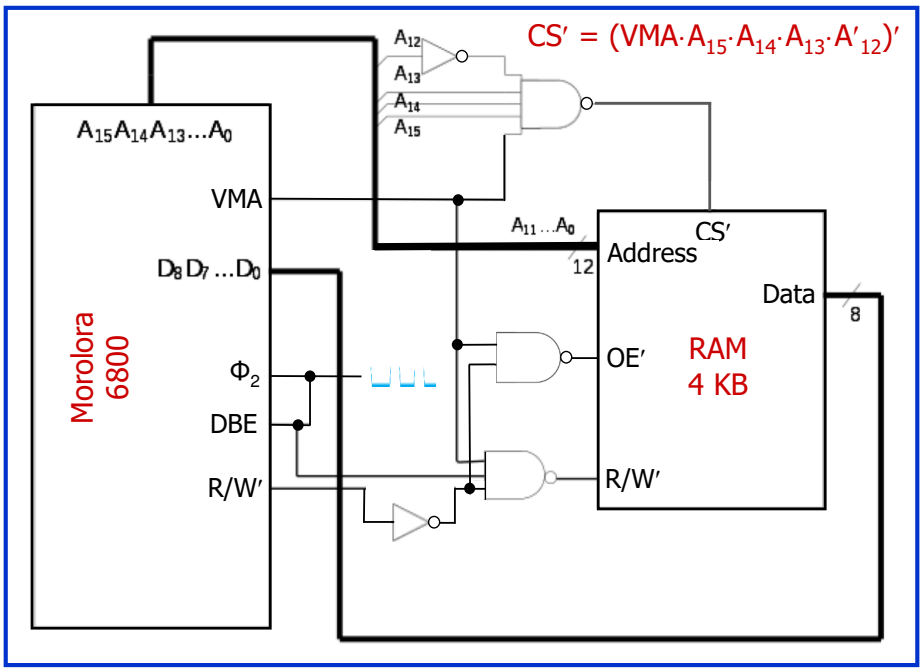
Διασύνδεση ολοκληρωμένου κυκλώματος μνήμης RAM 4KB για διευθύνσεις από E000 έως EFFF.

Πεδίο διευθύνσεων:
E000 = 11100...0 έως
11101...1 = EFFF.

Στον Motorola 6800, κατά την ανάγνωση δεδομένων πρέπει R/W′ = 1, VMA = 1.

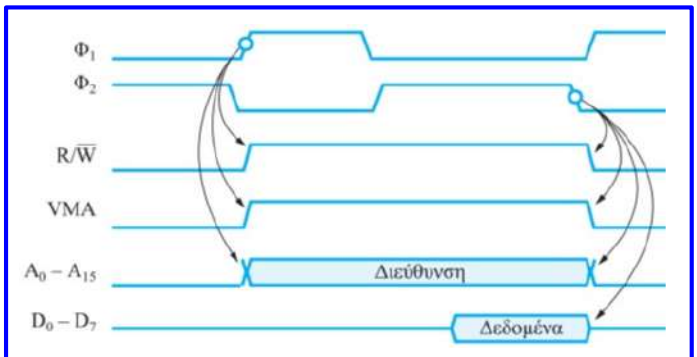
Κατά την εγγραφή δεδομένων πρέπει R/W′ = 0, VMA = 1.

Στον ακροδέκτη DBE τροφοδοτείται το σήμα χρονισμού Φ_2.

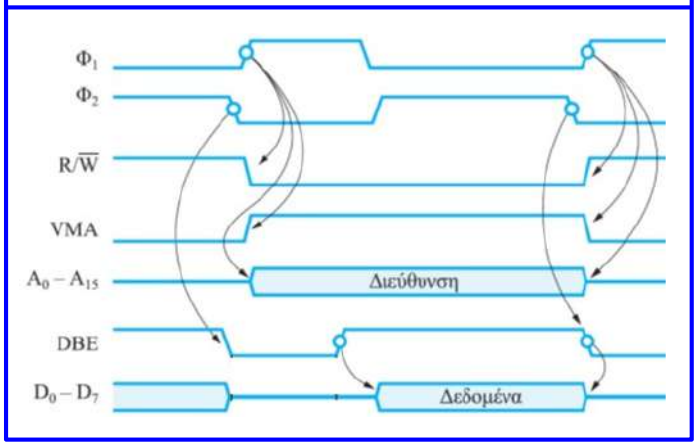


Παράδειγμα διασύνδεσης μνήμης στον Motorola 6800

Διαγράμματα χρονισμού κύκλου ανάγνωσης



Διαγράμματα χρονισμού κύκλου εγγραφής



Παράδειγμα διασύνδεσης συστήματος μνήμης

- Μικροϋπολογιστικό σύστημα με δίαυλο δεδομένων 8 ψηφίων και δίαυλο διευθύνσεων 16 ψηφίων. Οι διευθύνσεις μνήμης 0000 - 1FFF καλύπτονται από μνήμη ROM1, ενώ οι διευθύνσεις 4000 - 47FF και 5000 - 5FFF από δύο μνήμες RAM1 και RAM2, αντίστοιχα.
- Υπολογισμός της χωρητικότητας (μεγέθους) της μνήμης του συστήματος σε Kbits :
  - ✓ Αφού ο δίαυλος δεδομένων είναι 8 ψηφίων, δηλαδή 1 Byte, αυτό είναι και το μήκος λέξης κάθε κυκλώματος μνήμης.
  - ✓ Η μνήμη ROM1 καλύπτει τις διευθύνσεις 0000 - 1FFF, συνεπώς διαθέτει χωρητικότητα 2^{13} Bytes, δηλαδή $2^{13} = 2^3 \times 2^{10}$ = 8 KBytes ή 64 Kbits.
  - ✓ Η μνήμη RAM1 καλύπτει τις διευθύνσεις 4000 - 47FF που αντιστοιχούν σε 2^{11} Bytes, δηλαδή διαθέτει χωρητικότητα $2^{11} = 2 \times 2^{10}$ = 2 KBytes ή 16 Kbits.
  - ✓ Η μνήμη RAM2 καλύπτει τις διευθύνσεις 5000 - 5FFF που αντιστοιχούν σε 2^{12} Byte, δηλαδή διαθέτει χωρητικότητα $2^{12} = 2^2 \times 2^{10}$ = 4 KBytes ή 32 Kbits.
  - ✓ Συνολικά, η μνήμη του συστήματος έχει χωρητικότητα 112 Kbits.

Παράδειγμα διασύνδεσης συστήματος μνήμης

- Διαθέτουμε τα ακόλουθα ολοκληρωμένα κυκλώματα μνήμης και επιθυμούμε να επεκτείνουμε το σύστημα μνήμης, έτσι ώστε να καλυφθεί πλήρως όλο το πεδίο διευθύνσεων. Σύμφωνα με τις προδιαγραφές της εν λόγω επέκτασης:
  - ✓ η μνήμη ROM θα πρέπει να τοποθετηθεί στις αρχικές διευθύνσεις,
  - ✓ για την δημιουργία των σημάτων επιλογής των μνημών είναι διαθέσιμοι δύο αποκωδικοποιητές 2 σε 4 με είσοδο επίτρεψης, ένας αποκωδικοποιητής 3 σε 8 με είσοδο επίτρεψης και λογικές πύλες,
  - ✓ τα σήματα επιλογής όλων των μνημών ενεργοποιούνται στην λογική τιμή 1.

ΟΚ μνήμης	Χωρητικότητα (Kbits)	Οργάνωση (bits/διεύθυνση)
ROM2, ROM3	32	8
RAM3, RAM4	8	4
RAM5	64	8
RAM6	256	8

Παράδειγμα διασύνδεσης συστήματος μνήμης

Κατάστρωση του χάρτη του συστήματος μνήμης (για ευκολότερη υλοποίηση των κυκλωμάτων διασύνδεσης, η τοποθέτηση των μνημών στο χάρτη θα πρέπει να γίνεται έτσι ώστε να δημιουργούνται ομάδες με χωρητικότητες δυνάμεων του 2):

Μνήμη	Μέγεθος (λέξεις)	Ψηφία Διεύθυνσης	Πεδίο Διευθύνσεων	Διευθύνσεις σε δυαδική μορφή $A_{15}\,A_{14}\ \ldots\ A_2\,A_1\,A_0$
ROM1	8K	13	0000 – 1FFF	0000 0000 0000 0000 0001 1111 1111 1111
ROM2	4K	12	2000 – 2FFF	0010 0000 0000 0000 0010 1111 1111 1111
ROM3	4K	12	3000 – 3FFF	0011 0000 0000 0000 0011 1111 1111 1111
RAM1	2K	11	4000 – 47FF	0100 0000 0000 0000 0100 0111 1111 1111
RAM3, RAM4	2K	11	4800 – 4FFF	0100 1000 0000 0000 0100 1111 1111 1111
RAM2	4K	12	5000 – 5FFF	0101 0000 0000 0000 0101 1111 1111 1111
RAM5	8K	13	6000-7FFF	0110 0000 0000 0000 0111 1111 1111 1111
RAM6	32K	15	8000-FFFF	1000 0000 0000 0000 1111 1111 1111 1111

Παράδειγμα διασύνδεσης συστήματος μνήμης

$CS_{ROM1} = A'_{15}\,A'_{14}\,A'_{13}$

$CS_{ROM2} = A'_{15}\,A'_{14}\,A_{13}\,A'_{12}$

$CS_{ROM3} = A'_{15}\,A'_{14}\,A_{13}\,A_{12}$

$CS_{RAM1} = A'_{15}\,A_{14}\,A'_{13}\,A'_{12}\,A'_{11}$

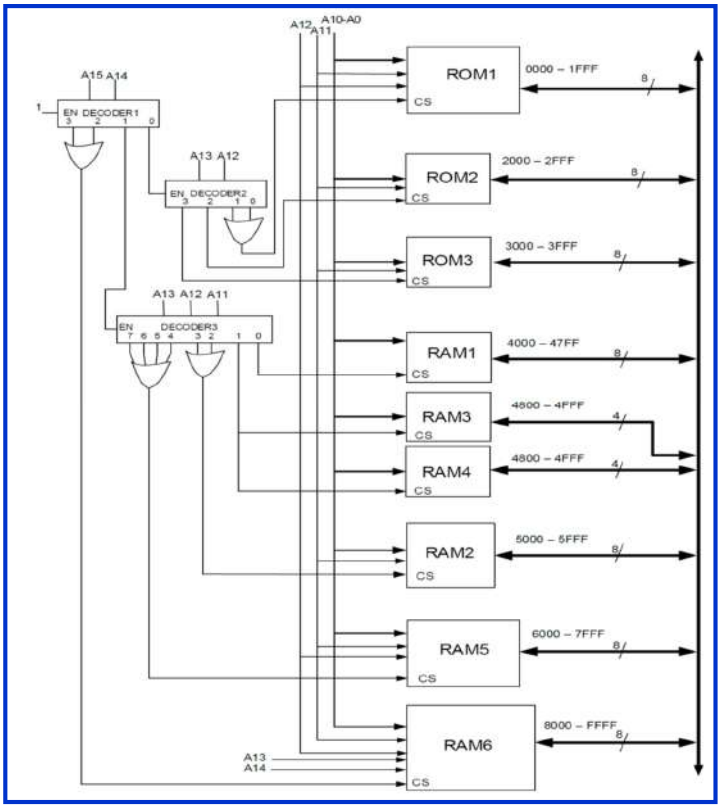
$CS_{RAM3/4} = A'_{15}\,A_{14}\,A'_{13}\,A'_{12}\,A_{11}$

$CS_{RAM2} = A'_{15}\,A_{14}\,A'_{13}\,A_{12}$

$CS_{RAM5} = A'_{15}\,A_{14}\,A_{13}$ $\quad CS_{RAM6} = A_{15}$

Από τις διευθύνσεις των μνημών σε δυαδική μορφή, προκύπτει η συνάρτηση των σημάτων επιλογής τους, τα οποία δημιουργούνται με τη χρήση των τριών διαθέσιμων αποκωδικοποιητών με είσοδο επίτρεψης και λογικών πυλών OR.

Για λόγους απλότητας της σχεδίασης δεν περιλαμβάνονται τα σήματα ανάγνωσης και εγγραφής.

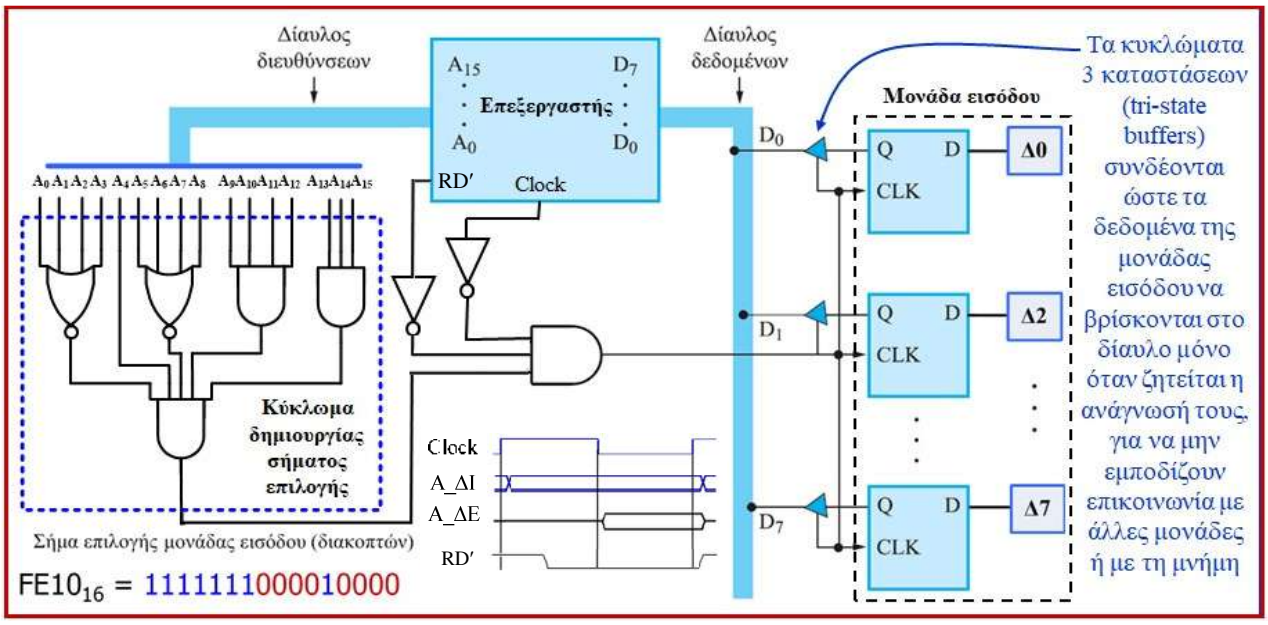


Διασύνδεση μονάδων εισόδου/εξόδου

- Για τη διασύνδεση μονάδων εισόδου/εξόδου με μικροεπεξεργαστή, υπάρχουν 2 τεχνικές.
- Η πρώτη τεχνική βασίζεται σε διακριτούς χώρους διευθύνσεων μνήμης και μονάδων εισόδου/εξόδου (I/O-mapped I/O) και η μεταφορά δεδομένων προς και από τις μονάδες εισόδου/εξόδου γίνεται με ειδικές εντολές εισόδου/εξόδου.
- Κατά την εκτέλεση των εντολών εισόδου/εξόδου, τίθεται τιμή 1 στα σήματα ανάγνωσης από μονάδα εισόδου (I/OR′) ή εγγραφής σε μονάδα εξόδου (I/OW′) ή τίθεται τιμή 1 στο σήμα IO/M′ και τιμή 0 στα σήματα RD′ ή WR′, ανάλογα με τον μικροεπεξεργαστή.
- Η δεύτερη τεχνική βασίζεται σε ενιαίο χώρο διευθύνσεων μνήμης και μονάδων εισόδου/εξόδου (memory-mapped I/O), στην οποία ένα μέρος των διευθύνσεων μνήμης αντιστοιχεί στους καταχωρητές των μονάδων εισόδου/εξόδου.
- Στην περίπτωση αυτή χειριζόμαστε τις μονάδες εισόδου/εξόδου όπως τη μνήμη και μπορούν να χρησιμοποιηθούν όλες οι εντολές και οι τρόποι διευθυνσιοδότησης του επεξεργαστή που χρησιμοποιούνται για τη μνήμη.
- Η διασύνδεση των συσκευών εισόδου/εξόδου γίνεται με τη βοήθεια του σήματος R/W′ ή των σημάτων RD′ και WR′ ή των σημάτων MEMR′ και MEMW′, ανάλογα με τον μικροεπεξεργαστή του συστήματος.
- Η διασύνδεση πολύπλοκων μονάδων εισόδου/εξόδου επιτυγχάνεται με ειδικά ολοκληρωμένα κυκλώματα διασύνδεσης.

Παράδειγμα διασύνδεσης μονάδας εισόδου

Διασύνδεση μονάδας εισόδου (8 διακόπτες), με μικροεπεξεργαστή που χρησιμοποιεί memory-mapped I/O και διαβάζει την κατάσταση των διακοπτών (0 ή 1) με τη μορφή ενός byte στη διεύθυνση FE10. Το υπολογιστικό σύστημα διαθέτει σύγχρονη αρτηρία ενός κύκλου ρολογιού.



Εξέλιξη μικροεπεξεργαστών

- Από τους πρώτους μικροεπεξεργαστές με μήκος λέξης 8 bits, έχουμε φτάσει σήμερα σε μικροεπεξεργαστές με μήκος λέξης 32, 64 ή ακόμη και 128 bits.
- Οι σημερινοί μικροεπεξεργαστές υποστηρίζουν την συνύπαρξη και συνλειτουργία πολλαπλών πυρήνων (cores), έχοντας φτάσει σε μικροεπεξεργαστές με δεκάδες πυρήνων.
- Το πλήθος των τρανζίστορς που περιέχονται σε έναν μικροεπεξεργαστή φθάνει σήμερα σε αρκετά δισεκατομμύρια, ενώ την δεκαετία του 1970 ήταν μόνο μερικές χιλιάδες.
- Η εξέλιξη της τεχνολογίας έδωσε τη δυνατότητα για μεγαλύτερες ταχύτητες χρονισμού των μικροεπεξεργαστών (σημερινές ταχύτητες χρονισμού της τάξης μερικών GHz).
- Έχει αυξηθεί ιδιαίτερα το μέγεθος της μνήμης (δυνατότητα διευθυνσιοδότησης πολλών GBytes) και ο αριθμός των διαθέσιμων καταχωρητών.
- Οι σύγχρονοι μικροεπεξεργαστές υποστηρίζουν συστήματα ιεραρχίας μνήμης και ενσωματώνουν έως και τρία επίπεδα κρυφής μνήμης.
- Υποστηρίζεται αυξημένη παραλληλία στα στάδια εκτέλεσης των διάφορων λειτουργιών (αλυσιδωτή επεξεργασία ή διοχέτευση, pipelining) και εφαρμογή εντολών σε πολλαπλά δεδομένα (SIMD, single instruction multiple data).
- Εκτός της δυνατότητας παράλληλης λειτουργίας διαφορετικών πυρήνων του ίδιου μικροεπεξεργαστή, έχει αναπτυχθεί η δυνατότητα εκτέλεσης πολλαπλών «νημάτων» (multi-threads) στον ίδιο πυρήνα, δηλ. παράλληλη εκτέλεση τμημάτων του ίδιου κώδικα.

Βιβλιογραφία

Ψηφιακή σχεδίαση

- Λ. Μπισδούνης, Ψηφιακά συστήματα, Εκδόσεις Ε.Α.Π., 2015 (ανατύπωση 2017).
- Π. Λιναρδής, Ψηφιακή σχεδίαση Ι, Εκδόσεις Ε.Α.Π., 2008.
- Α. Σκόδρας, Ψηφιακή σχεδίαση ΙΙ, Εκδόσεις Ε.Α.Π., 2008.
- Μ. Ρουμελιώτης, Σ.Ι. Σουραβλάς, Ψηφιακή σχεδίαση: Αρχές & εφαρμογές, Εκδόσεις Τζιόλα, 2018.
- M.M. Mano, M.D. Ciletti, Ψηφιακή σχεδίαση, Εκδόσεις Παπασωτηρίου, 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.
- S. Brown, Z. Vranesic, Σχεδίαση ψηφιακών συστημάτων με τη γλώσσα VHDL, Εκδόσεις Τζιόλα, 2011.
- P.K. Lala, Principles of modern digital design, Wiley, 2007.

Αρχιτεκτονική υπολογιστών

- Δ. Νικολός, Αρχιτεκτονική υπολογιστών Ι, Εκδόσεις Ε.Α.Π., 2008.
- Δ. Νικολός, Αρχιτεκτονική υπολογιστών, Έκδοση Π. Παπακωνσταντίνου, 2017.
- C. Hammacher, Z. Vranesic, S. Zaky, Οργάνωση και αρχιτεκτονική υπολογιστών, Εκδόσεις Επίκεντρο, 2007.
- W. Stallings, Οργάνωση και αρχιτεκτονική υπολογιστών, Εκδόσεις Τζιόλα, 2020.
- D. Patterson, J. Hennessy, Αρχιτεκτονική υπολογιστών, Εκδόσεις Κλειδάριθμος, 2020.
- A.S. Tanenbaum, Η αρχιτεκτονική των υπολογιστών: Μια δομημένη προσέγγιση, Εκδόσεις Κλειδάριθμος, 2000.
- M. Burrell, Αρχιτεκτονική υπολογιστών, Εκδόσεις Πανεπιστημίου Μακεδονίας, 2006.
- M.M. Mano, C.R. Kime, T. Martin, Σχεδίαση λογικών κυκλωμάτων και υπολογιστών, Εκδόσεις Τζιόλα, 2017.

Μικροεπεξεργαστές

- Γ. Αλεξίου, Μικροεπεξεργαστές, Εκδόσεις Ε.Α.Π., 2008.
- Ν. Πετρέλλης, Γ. Αλεξίου, Μικροεπεξεργαστές και σχεδιασμός μικροϋπολογιστικών συστημάτων, Εκδόσεις Κλειδάριθμος, 2012.
- Π. Παπάζογλου, Μικροεπεξεργαστές: Αρχές και εφαρμογές, Εκδόσεις Τζιόλα, 2022.
- 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.

e-mail: bisdounis.lampros@ac.eap.gr