

5 Οργάνωση Υπολογιστών

Εισαγωγή στην Επιστήμη των Υπολογιστών © Εκδόσεις Κλειδάριθμος

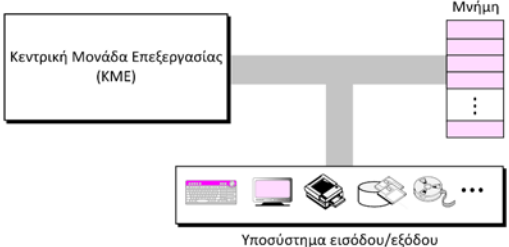


Στόχοι

Μετά την ολοκλήρωση αυτού του κεφαλαίου, ο σπουδαστής θα είναι σε θέση:

- Να περιγράφει τα τρία υποσυστήματα ενός υπολογιστή.
- Να περιγράφει τον ρόλο της κεντρικής μονάδας επεξεργασίας (ΚΜΕ).
- Να περιγράφει τις φάσεις ανάκλησης, αποκωδικοποίησης, και εκτέλεσης ενός κύκλου.
- Να περιγράφει την κύρια μνήμη και τον χώρο διεύθυνσών της.
- Να ορίζει το υποσύστημα εισόδου/εξόδου.
- Να κατανοεί τον τρόπο διασύνδεσης των υποσυστημάτων.
- Να περιγράφει διάφορες μεθόδους διεθυνσιοδότησης εισόδου/εξόδου.
- Να διακρίνει τις δύο βασικές τάσεις στον σχεδιασμό υπολογιστών.
- Να κατανοεί τον τρόπο με τον οποίο βελτιώνεται η διεκπεραιωτική ικανότητα ενός υπολογιστή με τη χρήση διαχέτευσης και παράλληλης επεξεργασίας.

Τα μέρη που αποτελούν έναν υπολογιστή μπορούν να χωριστούν σε τρεις βασικές κατηγορίες ή υποσυστήματα: την **κεντρική μονάδα επεξεργασίας (ΚΜΕ)**, την **κύρια μνήμη**, και το **υποσύστημα εισόδου/εξόδου**.



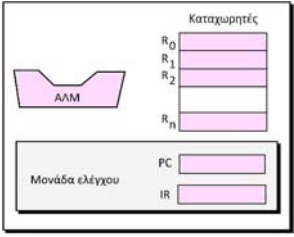
Εικόνα 5.1 Υλικό υπολογιστών (υποσυστήματα)

5-1 ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ

Η **κεντρική μονάδα επεξεργασίας** ή ΚΜΕ (central processing unit, CPU) εκτελεί λειτουργίες στα δεδομένα. Στις περισσότερες αρχιτεκτονικές αποτελείται από τρία τμήματα: μια **αριθμητική και λογική μονάδα (ΑΛΜ)**, μια **μονάδα ελέγχου**, και ένα σύνολο **καταχωρητών**, δηλαδή θέσεις γρήγορης αποθήκευσης.

Η αριθμητική και λογική μονάδα (ΑΛΜ)

Η αριθμητική και λογική μονάδα ή ΑΛΜ (arithmetic logic unit, ALU) εκτελεί αριθμητικές και λογικές πράξεις, και πράξεις μετατόπισης στα δεδομένα.



Εικόνα 5.2 Κεντρική μονάδα επεξεργασίας (ΚΜΕ)

Καταχωρητές

Οι καταχωρητές είναι αυτόνομες θέσεις γρήγορης αποθήκευσης στις οποίες διατηρούνται δεδομένα προσωρινά. Για να είναι εφικτή η λειτουργία της ΚΜΕ απαιτούνται πολλοί καταχωρητές. Ορισμένοι από αυτούς παρουσιάζονται στην Εικόνα 5.2.

- Καταχωρητές δεδομένων
- Καταχωρητές εντολών
- Μετρητής προγράμματος

Η μονάδα ελέγχου

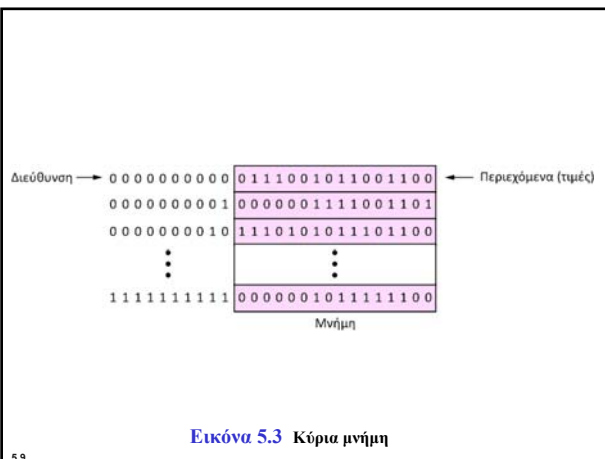
Το τρίτο μέρος της ΚΜΕ είναι η μονάδα ελέγχου. Η μονάδα ελέγχου ελέγχει τη λειτουργία του κάθε υποσυστήματος. Ο έλεγχος πραγματοποιείται μέσω σημάτων που στέλνονται από τη μονάδα ελέγχου στα άλλα υποσυστήματα.

5.7

5-2 ΚΥΡΙΑ ΜΝΗΜΗ

Η **κύρια μνήμη** είναι το δεύτερο βασικό υποσύστημα σε έναν υπολογιστή (Εικόνα 5.3). Αποτελείται από μια συλλογή θέσεων αποθήκευσης, κάθε μία από τις οποίες διαθέτει ένα μοναδικό αναγνωριστικό που ονομάζεται **διεύθυνση**. Τα δεδομένα μεταφέρονται από και προς τη μνήμη σε ομάδες bit που ονομάζονται **λέξεις**. Μια λέξη μπορεί να είναι μια ομάδα από 8 bit, 16 bit, 32 bit, ή 64 bit (και αυξάνονται συνεχώς). Αν η λέξη είναι 8 bit, τότε αναφέρεται ως **byte**. Ο όρος byte είναι τόσο συνηθισμένος στην επιστήμη των υπολογιστών, που κάποιες φορές μια λέξη 16 bit αναφέρεται ως λέξη 2 byte, ή μια λέξη 32 bit αναφέρεται ως λέξη 4 byte.

5.8



Εικόνα 5.3 Κύρια μνήμη

5.9

Χώρος διευθύνσεων

Για την προσπέλαση μιας λέξης στη μνήμη χρειάζεται ένα αναγνωριστικό. Αν και οι προγραμματιστές χρησιμοποιούν ονόματα για τον προσδιορισμό των λέξεων (ή συλλογών από λέξεις), σε επίπεδο υλικού κάθε λέξη προσδιορίζεται από μια διεύθυνση. Ο συνολικός αριθμός των μοναδικά προσδιορισμένων θέσεων στη μνήμη ονομάζεται **χώρος διευθύνσεων** (address space). Για παράδειγμα, μια μνήμη 64 kilobyte με μέγεθος λέξης 1 byte διαθέτει χώρο διευθύνσεων με εύρος από 0 έως 65.535.

5.10

Πίνακας 5.1 Μονάδες μνήμης

Μονάδα	Ακριβές πλήθος byte	Προσέγγιση
Kilobyte	2^{10} (1024) byte	10^3 byte
Megabyte	2^{20} (1.048.576) byte	10^6 byte
Gigabyte	2^{30} (1.073.741.824) byte	10^9 byte
Terabyte	2^{40} byte	10^{12} byte



Οι διευθύνσεις μνήμης καθορίζονται με τη χρήση μη προσημασμένων δυαδικών ακεραίων.

5.11

Παράδειγμα 5.1

Ένας υπολογιστής έχει 32 MB (megabyte) μνήμης. Πόσα bit χρειάζονται για τη διευθυνσιοδότηση κάθε bit της μνήμης;

Λύση

Ο χώρος διευθύνσεων της μνήμης είναι 32 MB, ή 2^{25} ($2^5 \times 2^{20}$). Αυτό σημαίνει ότι για τη διευθυνσιοδότηση κάθε byte απαιτούνται $\log_2 2^{25}$, ή **25 bit**.

Παράδειγμα 5.2

Ένας υπολογιστής έχει 128 MB μνήμης. Κάθε λέξη σε αυτόν τον υπολογιστή είναι οκτώ byte. Πόσα bit χρειάζονται για τη διευθυνσιοδότηση κάθε λέξης της μνήμης;

Λύση

Ο χώρος διευθύνσεων της μνήμης είναι 128 MB, δηλαδή 2^{27} . Όμως κάθε λέξη είναι οκτώ (2^3) byte, το οποίο σημαίνει ότι έχουμε 2^{24} λέξεις. Επομένως, για τη διευθυνσιοδότηση κάθε λέξης απαιτούνται $\log_2 2^{24}$, ή **24 bit**.

5.12

Τύποι μνήμης

Υπάρχουν δύο κύριοι τύποι μνήμης: η **RAM** και η **ROM**.

Μνήμη τυχαίας προσπέλασης (RAM)

- ❑ Στατική RAM (SRAM)
- ❑ Δυναμική RAM (DRAM)

Μνήμη μόνο για ανάγνωση (ROM)

- ❑ Προγραμματιζόμενη μνήμη μόνο για ανάγνωση (PROM)
- ❑ Διαγράψιμη προγραμματιζόμενη μνήμη μόνο για ανάγνωση (EPROM)
- ❑ Ηλεκτρικά διαγράψιμη προγραμματιζόμενη μνήμη μόνο για ανάγνωση (EEPROM)

5.13

Ιεραρχία μνήμης

Οι χρήστες υπολογιστών χρειάζονται μεγάλες ποσότητες μνήμης, ιδιαίτερα εκείνης που είναι πολύ γρήγορη και πολύ φτηνή. Αυτή η απαίτηση δεν είναι δυνατόν να ικανοποιηθεί πάντοτε, αφού η πολύ γρήγορη μνήμη συνήθως δεν είναι φθηνή, οπότε πρέπει να γίνεται κάποιος συμβιβασμός. Τη λύση αποτελούν τα **ιεραρχικά** επίπεδα μνήμης.

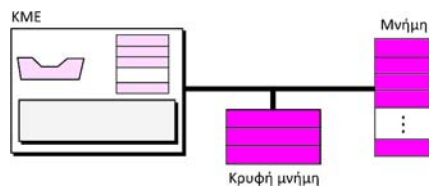


Εικόνα 5.4 Ιεραρχία μνήμης

5.14

Κρυφή μνήμη

Η κρυφή μνήμη (cache memory) είναι γρηγορότερη από την κύρια μνήμη αλλά πιο αργή από την ΚΜΕ και τους καταχωρητές της. Η κρυφή μνήμη, η οποία συνήθως έχει μικρό μέγεθος, μεσολαβεί μεταξύ της ΚΜΕ και της κύριας μνήμης (Εικόνα 5.5).



Εικόνα 5.5 Κρυφή μνήμη

5.15

5-3 ΥΠΟΣΥΣΤΗΜΑ ΕΙΣΟΔΟΥ/ΕΞΟΔΟΥ

Το τρίτο βασικό υποσύστημα ενός υπολογιστή είναι η ομάδα συσκευών που αναφέρεται ως υποσύστημα εισόδου/εξόδου (E/E). Αυτό το υποσύστημα επιτρέπει σε έναν υπολογιστή να επικοινωνεί με τον έξω κόσμο και να διατηρεί προγράμματα και δεδομένα ακόμα και όταν είναι κλειστός. Οι συσκευές εισόδου/εξόδου χωρίζονται σε δύο μεγάλες κατηγορίες: στις **μη αποθηκευτικές** και στις **αποθηκευτικές**.

5.16

Μη αποθηκευτικές συσκευές

Οι **μη αποθηκευτικές συσκευές** επιτρέπουν στην ΚΜΕ και τη μνήμη να επικοινωνούν με τον έξω κόσμο, αλλά δεν μπορούν να αποθηκεύσουν πληροφορίες.

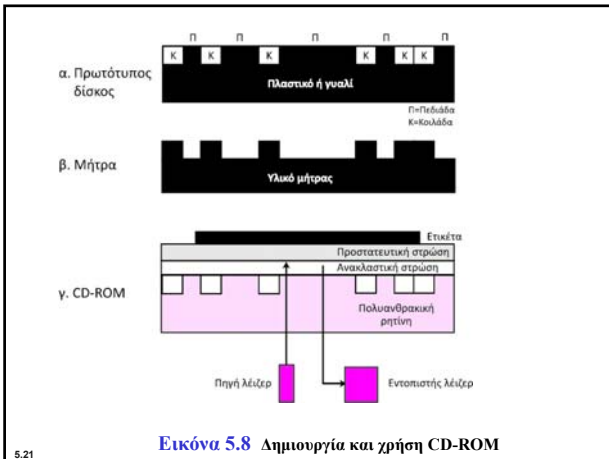
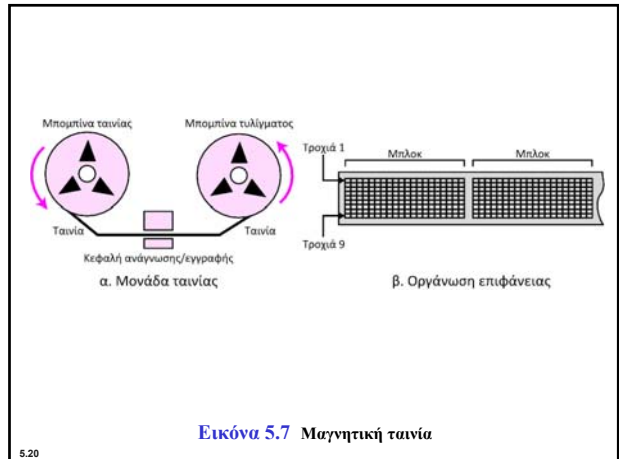
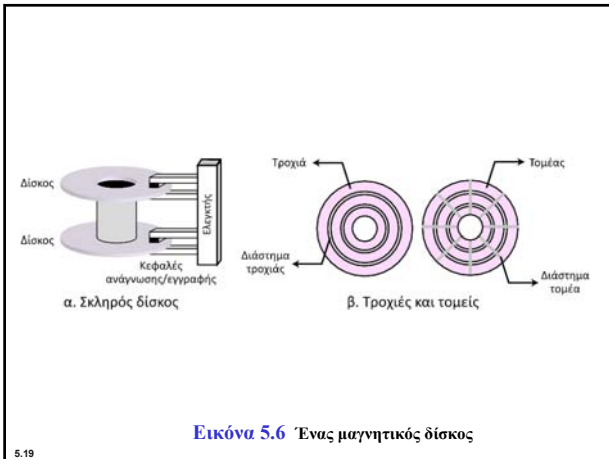
- ❑ Πληκτρολόγιο και οθόνη
- ❑ Εκτυπωτής

5.17

Αποθηκευτικές συσκευές

Οι **αποθηκευτικές συσκευές**, αν και θεωρούνται συσκευές E/E, μπορούν να φιλοξενήσουν μεγάλες ποσότητες πληροφοριών για μελλοντική ανάκτηση. Είναι φτηνότερες από την κύρια μνήμη, και τα περιεχόμενά τους είναι μη πτητικά, δηλαδή δεν διαγράφονται με τη διακοπή παροχής ρεύματος. Μερικές φορές αναφέρονται ως βοηθητικές αποθηκευτικές συσκευές. Χωρίζονται σε **μαγνητικές** και **οπτικές**.

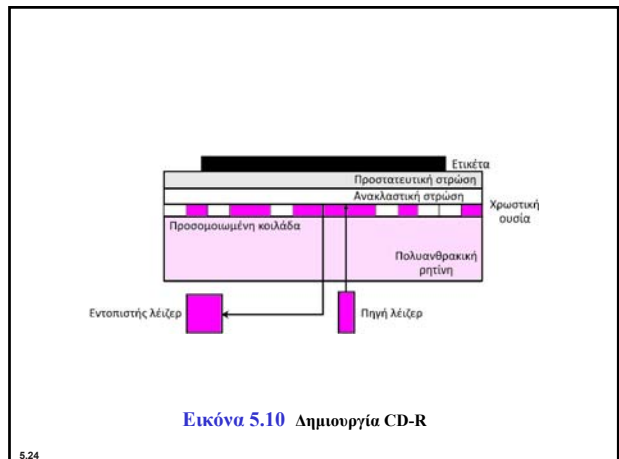
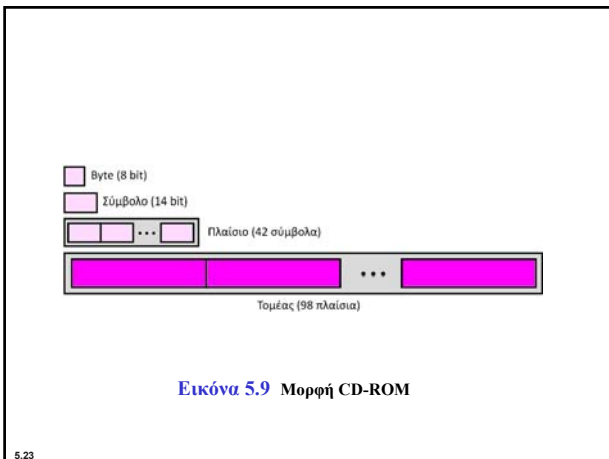
5.18

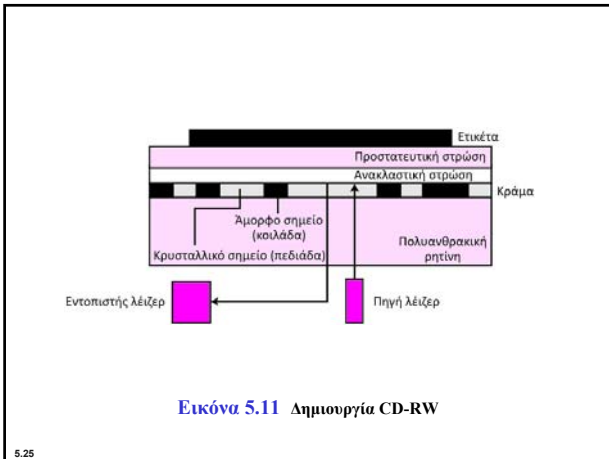


Πίνακας 5.2 Ταχύτητες CD-ROM

Ταχύτητα	Ρυθμός ανάγνωσης δεδομένων	Προσέγγιση
1x	153.600 byte ανά δευτερόλεπτο	150 KB/s
2x	307.200 byte ανά δευτερόλεπτο	300 KB/s
4x	614.400 byte ανά δευτερόλεπτο	600 KB/s
6x	921.600 byte ανά δευτερόλεπτο	900 KB/s
8x	1.228.800 byte ανά δευτερόλεπτο	1,2 MB/s
12x	1.843.200 byte ανά δευτερόλεπτο	1,8 MB/s
16x	2.457.600 byte ανά δευτερόλεπτο	2,4 MB/s
24x	3.688.400 byte ανά δευτερόλεπτο	3,6 MB/s
32x	4.915.200 byte ανά δευτερόλεπτο	4,8 MB/s
40x	6.144.000 byte ανά δευτερόλεπτο	6 MB/s

5.22



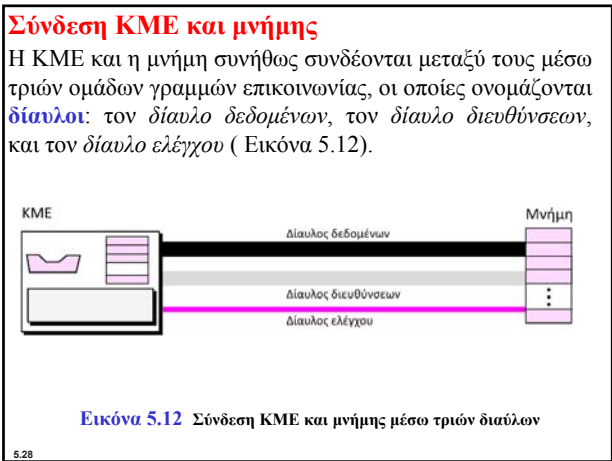


Πίνακας 5.3 Χωρητικότητες DVD

Είδος	Χωρητικότητα
μονής πλευράς, μονής στρώσης	4,7 GB
μονής πλευράς, διπλής στρώσης	8,5 GB
διπλής πλευράς, μονής στρώσης	9,4 GB
διπλής πλευράς, διπλής στρώσης	17 GB

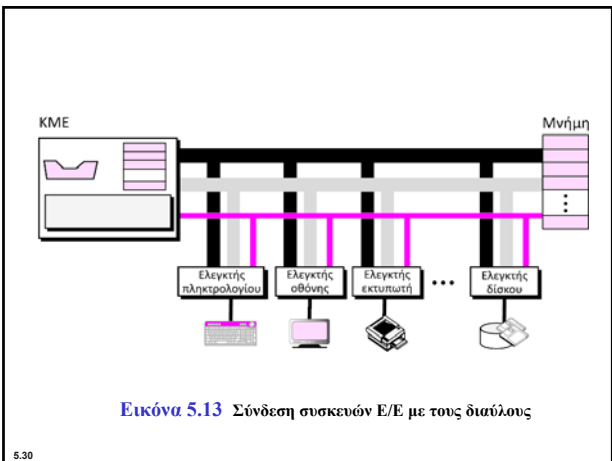
5-4 ΔΙΑΣΥΝΔΕΣΗ ΥΠΟΣΥΣΤΗΜΑΤΩΝ

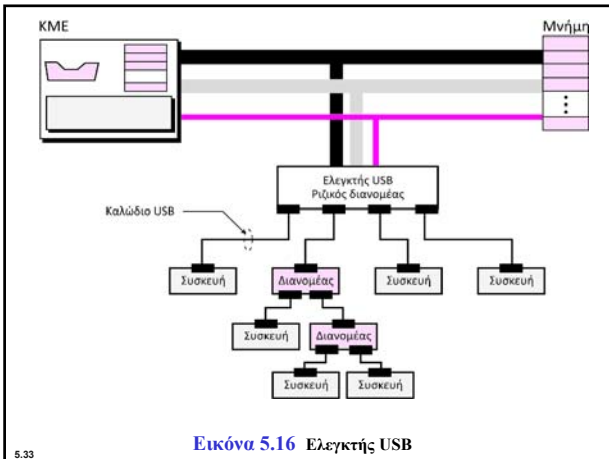
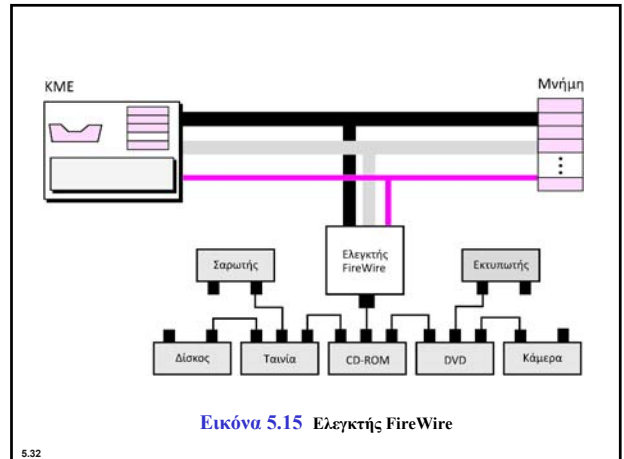
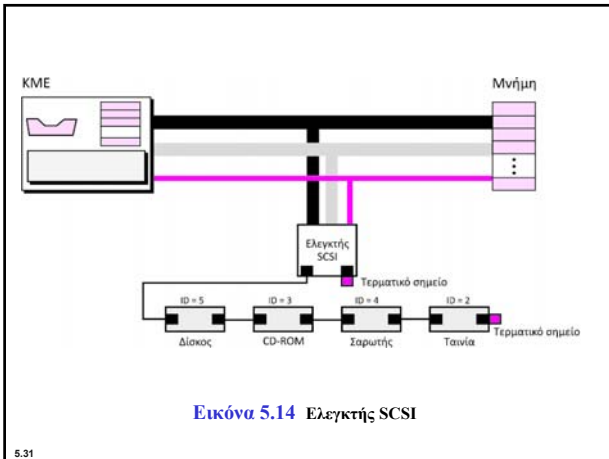
Στις προηγούμενες ενότητες περιγράφηκαν τα χαρακτηριστικά των τριών υποσυστημάτων (ΚΜΕ, κύρια μνήμη, και E/E) σε έναν αυτόνομο υπολογιστή. Σε αυτή την ενότητα θα δούμε τον τρόπο με τον οποίο διασυνδέονται τα τρία αυτά υποσυστήματα. Η διασύνδεση παίζει σημαντικό ρόλο, επειδή τα τρία υποσυστήματα πρέπει να ανταλλάσσουν πληροφορίες.



Σύνδεση συσκευών E/E

Οι συσκευές E/E δεν είναι δυνατό να συνδεθούν απευθείας στους διαύλους που συνδέουν την ΚΜΕ με τη μνήμη, επειδή η φύση των συσκευών αυτών είναι διαφορετική από τη φύση της ΚΜΕ και της μνήμης. Οι συσκευές E/E είναι ηλεκτρομηχανικές, μαγνητικές, ή οπτικές, ενώ η ΚΜΕ και η μνήμη είναι ηλεκτρονικές συσκευές. Οι συσκευές E/E λειτουργούν σε αρκετά μικρότερη ταχύτητα από ό,τι η ΚΜΕ και η μνήμη, με αποτέλεσμα να είναι απαραίτητη η ύπαρξη μιας ενδιάμεσης συσκευής για τον χειρισμό αυτής της διαφοράς. Για τον λόγο αυτό, οι συσκευές E/E συνδέονται με τους διαύλους μέσω μιας συσκευής που ονομάζεται ελεγκτής εισόδου/εξόδου ή διασύνδεση (interface). Κάθε συσκευή εισόδου/εξόδου έχει τον δικό της ελεγκτή (Εικόνα 5.13).

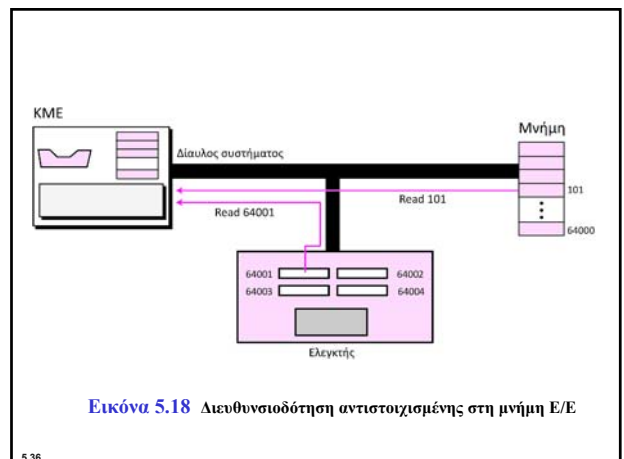
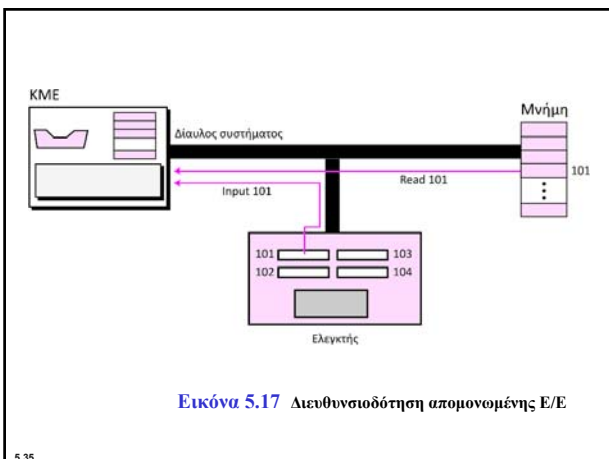




Διευθυνσιοδότηση συσκευών εισόδου/εξόδου

Η ΚΜΕ συνήθως χρησιμοποιεί τον ίδιο δίαυλο για την ανάγνωση ή την εγγραφή δεδομένων στην κύρια μνήμη και τις συσκευές Ε/Ε. Η μοναδική διαφορά είναι η εντολή. Αν η εντολή αναφέρεται σε κάποια λέξη της κύριας μνήμης, η μεταφορά δεδομένων γίνεται μεταξύ κύριας μνήμης και ΚΜΕ. Αν η εντολή προσδιορίζει κάποια συσκευή Ε/Ε, η μεταφορά δεδομένων γίνεται μεταξύ της συσκευής Ε/Ε και της ΚΜΕ. Για τον χειρισμό της διευθυνσιοδότησης των συσκευών Ε/Ε υπάρχουν δύο μέθοδοι: η απομονωμένη Ε/Ε και η αντιστοιχισμένη στη μνήμη Ε/Ε.

5.34



5-5 ΕΚΤΕΛΕΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ

Σήμερα, η επεξεργασία των δεδομένων από τους **υπολογιστές γενικής χρήσης** γίνεται με τη χρήση ενός συνόλου εντολών που ονομάζεται **πρόγραμμα**. Ο υπολογιστής εκτελεί το πρόγραμμα για να δημιουργήσει δεδομένα εξόδου χρησιμοποιώντας δεδομένα εισόδου. Τόσο το πρόγραμμα όσο και τα δεδομένα αποθηκεύονται στη μνήμη.

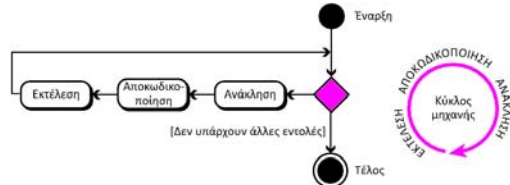


Στο τέλος αυτού του κεφαλαίου παρέχονται ορισμένα παραδείγματα σχετικά με τον τρόπο εκτέλεσης ενός προγράμματος από έναν υποθετικό απλό υπολογιστή.

5.37

Κύκλος μηχανής

Η ΚΜΕ χρησιμοποιεί επαναλαμβανόμενους **κύκλους μηχανής** (machine cycles) για να εκτελέσει τις εντολές ενός προγράμματος μία προς μία, από την αρχή μέχρι το τέλος. Ένας απλοποιημένος κύκλος αποτελείται από τρεις φάσεις: **ανάκληση** (fetch), **αποκωδικοποίηση** (decode), και **εκτέλεση** (execute) — Εικόνα 5.19.



Εικόνα 5.19 Τα βήματα ενός κύκλου μηχανής

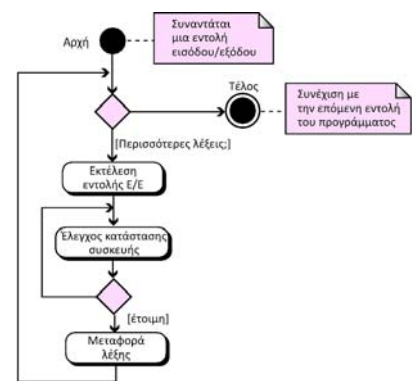
5.38

Λειτουργία εισόδου/εξόδου

Για τη μεταφορά δεδομένων από συσκευές E/E στην ΚΜΕ και τη μνήμη απαιτούνται εντολές. Επειδή οι συσκευές εισόδου/εξόδου λειτουργούν σε πολύ μικρότερες ταχύτητες από την ΚΜΕ, η λειτουργία της ΚΜΕ πρέπει με κάποιον τρόπο να συγχρονίζεται με τη δική τους. Για τον συγχρονισμό αυτό έχουν αναπτυχθεί τρεις μέθοδοι: η προγραμματισμένη E/E, η οδηγούμενη από διακοπές E/E, και η άμεση προσπέλαση μνήμης (DMA).

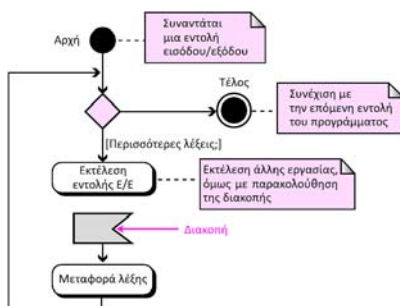
- Προγραμματισμένη E/E
- Οδηγούμενη από διακοπές E/E
- Άμεση προσπέλαση μνήμης (DMA)

5.39



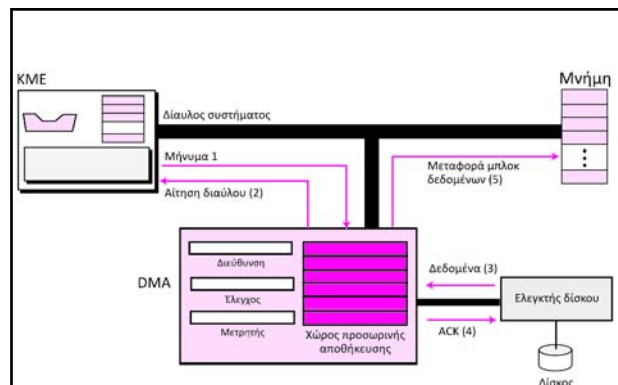
Εικόνα 5.20 Προγραμματισμένη E/E

5.40



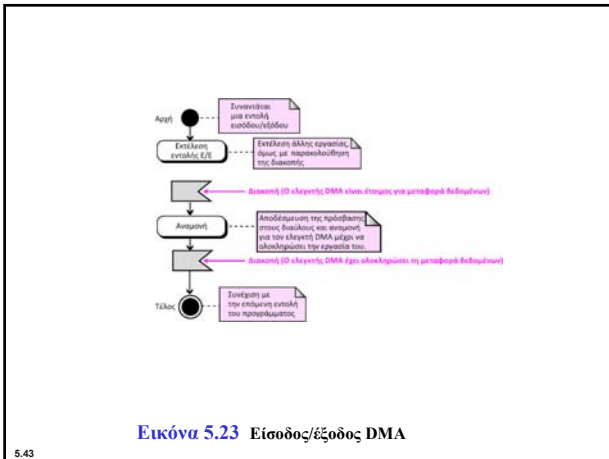
Εικόνα 5.21 Οδηγούμενη από διακοπές E/E

5.41



Εικόνα 5.22 Σύνδεση DMA στον γενικό διάλυο

5.42



Εικόνα 5.23 Είσοδος/έξοδος DMA

5-6 ΔΙΑΦΟΡΕΤΙΚΕΣ ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ

Η αρχιτεκτονική και η οργάνωση των υπολογιστών έχουν περάσει από πολλές αλλαγές κατά τη διάρκεια των τελευταίων δεκαετιών. Σε αυτή την ενότητα θα περιγράψουμε ορισμένες αρχιτεκτονικές και τρόπους οργάνωσης που διαφέρουν από την απλή αρχιτεκτονική υπολογιστών την οποία περιγράψαμε νωρίτερα.

CISC

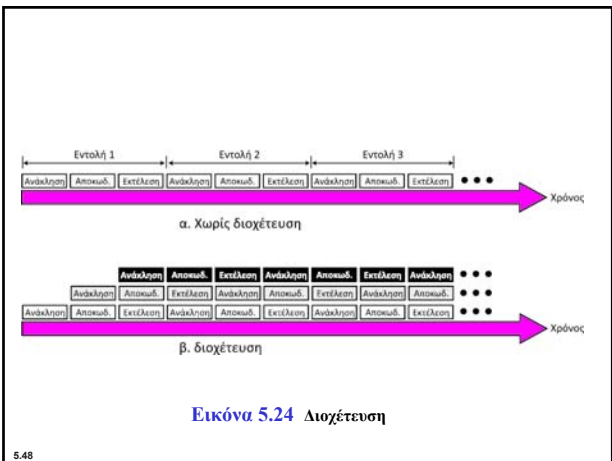
Το ακρωνύμιο CISC (προφέρεται *σισκ*) προέρχεται από τη φράση complex instruction set computer, δηλαδή υπολογιστής σύνθετου συνόλου εντολών. Η στρατηγική στην οποία βασίζονται οι αρχιτεκτονικές CISC είναι η ύπαρξη ενός μεγάλου συνόλου εντολών, στο οποίο συμπεριλαμβάνονται και σύνθετες. Ο προγραμματισμός σε CISC είναι πιο εύκολος από ό,τι σε άλλους σχεδιασμούς, επειδή υπάρχει μία εντολή για κάθε απλή ή σύνθετη εργασία. Συνεπώς, οι προγραμματιστές δεν χρειάζεται να γράφουν σύνολα εντολών για να πραγματοποιήσουν κάποια σύνθετη εργασία.

RISC

Το ακρωνύμιο RISC (προφέρεται *ρισκ*) προέρχεται από τη φράση reduced instruction set computer, δηλαδή υπολογιστής περιορισμένου συνόλου εντολών. Η στρατηγική στην οποία βασίζεται η αρχιτεκτονική RISC είναι η ύπαρξη ενός μικρού συνόλου εντολών, οι οποίες πραγματοποιούν ένα ελάχιστο πλήθος απλών λειτουργιών. Οι σύνθετες εντολές προσομοιώνονται με τη χρήση ενός υποσυνόλου απλών εντολών. Ο προγραμματισμός σε RISC είναι πιο δύσκολος και χρονοβόρος από ό,τι στον άλλο σχεδιασμό, επειδή οι πιο σύνθετες εντολές προσομοιώνονται από απλές.

Διοχέτευση

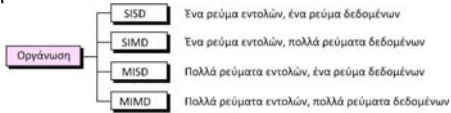
Έχουμε μάθει ότι, για κάθε εντολή, ο υπολογιστής χρησιμοποιεί τις τρεις φάσεις της ανάκλησης, αποκωδικοποίησης, και εκτέλεσης. Στους πρώτους υπολογιστές, οι φάσεις αυτές έπρεπε να ολοκληρώνονται με τη σειρά για κάθε εντολή. Με άλλα λόγια, πριν ξεκινήσουν οι φάσεις της εντολής $n + 1$ πρέπει να ολοκληρωθούν και οι τρεις φάσεις της εντολής n . Στους σύγχρονους υπολογιστές χρησιμοποιείται μια τεχνική που ονομάζεται **διοχέτευση** (pipelining) η οποία βελτιώνει τη διεκπεραιωτική ικανότητά τους (δηλαδή ο συνολικός αριθμός των εντολών που μπορούν να εκτελεστούν σε κάθε χρονική περίοδο). Η τεχνική αυτή βασίζεται στην ιδέα ότι, αν η μονάδα ελέγχου μπορεί να διεκπεραιώνει δύο από αυτές τις φάσεις ή και τις τρεις, τότε η επόμενη εντολή μπορεί να ξεκινήσει πριν ολοκληρωθεί η προηγούμενη.



Εικόνα 5.24 Διοχέτευση

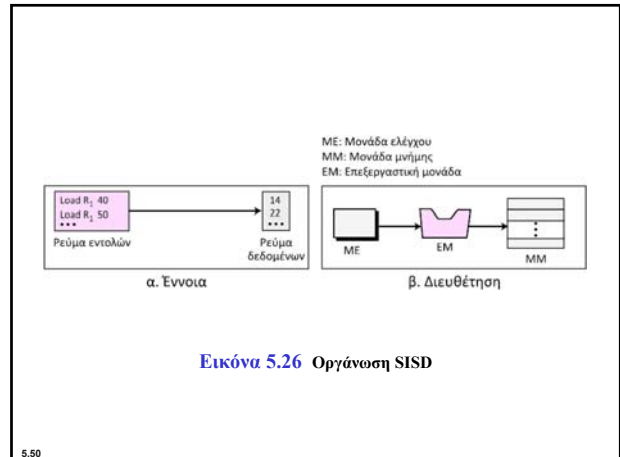
Παράλληλη επεξεργασία

Παραδοσιακά, οι υπολογιστές διέθεταν μία μονάδα ελέγχου, μία αριθμητική και λογική μονάδα, και μία μονάδα μνήμης. Με την εξέλιξη της τεχνολογίας και την πτώση του κόστους του υλικού υπολογιστών, σήμερα υπάρχουν διαθέσιμοι υπολογιστές με πολλές μονάδες ελέγχου, πολλές αριθμητικές και λογικές μονάδες, και πολλές μονάδες μνήμης. Η ιδέα αυτή αναφέρεται ως **παράλληλη επεξεργασία**. Όπως και η διοχέτευση, η παράλληλη επεξεργασία μπορεί να βελτιώσει τη διεκπεραιωτική ικανότητα.



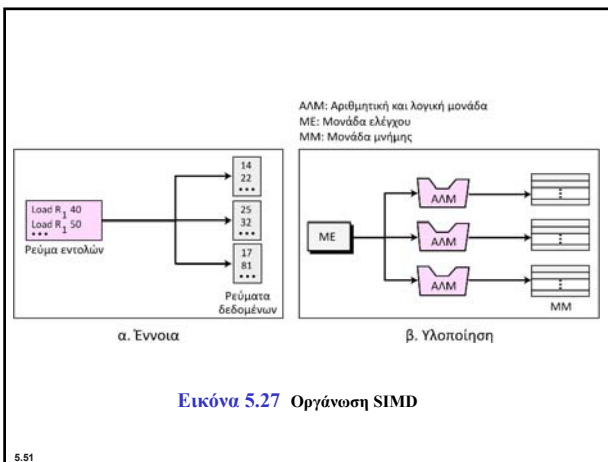
Εικόνα 5.24 Ταξινόμηση της οργάνωσης υπολογιστών

5.49



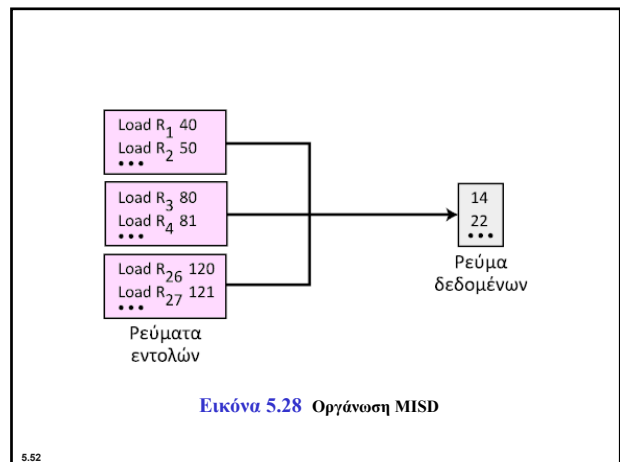
Εικόνα 5.26 Οργάνωση SIMD

5.50



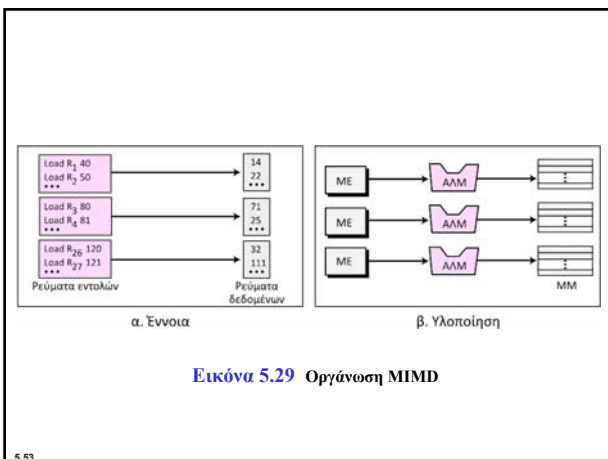
Εικόνα 5.27 Οργάνωση SIMD

5.51



Εικόνα 5.28 Οργάνωση MISD

5.52



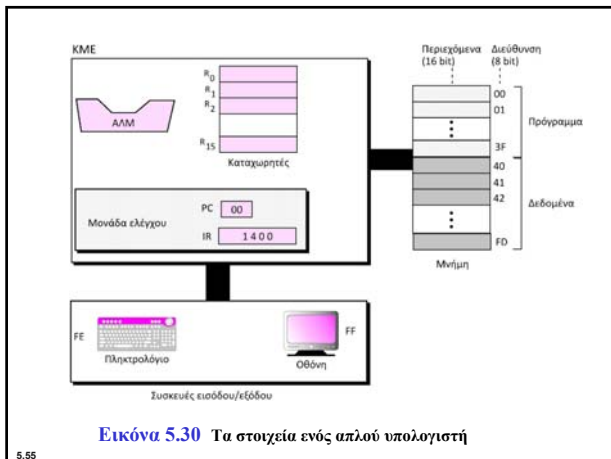
Εικόνα 5.29 Οργάνωση MIMD

5.53

5-7 ΕΝΑΣ ΑΠΛΟΣ ΥΠΟΛΟΓΙΣΤΗΣ

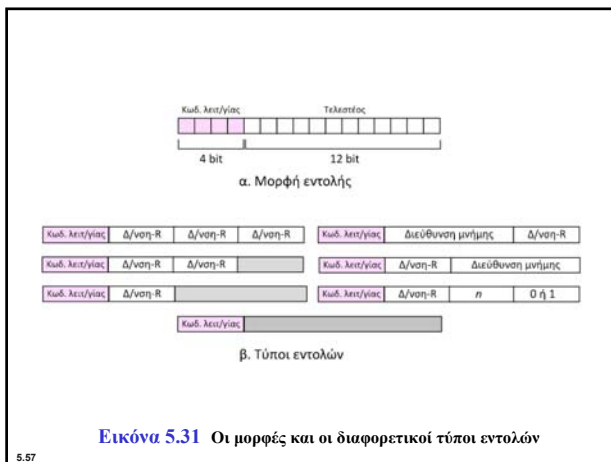
Για να εξηγήσουμε την αρχιτεκτονική των υπολογιστών, καθώς και τον τρόπο επεξεργασίας εντολών, θα χρησιμοποιήσουμε έναν απλό (υποθετικό) υπολογιστή, ο οποίος παρουσιάζεται στην Εικόνα 5.30. Ο απλός υπολογιστής μας αποτελείται από τρία στοιχεία: ΚΜΕ, μνήμη, και υποσύστημα εισόδου/εξόδου.

5.54



Σύνολο εντολών

Ο απλός υπολογιστής μας διαθέτει ένα σύνολο δεκαέξι εντολών, παρόλο που χρησιμοποιούμε μόνο δεκατέσσερις από αυτές. Κάθε εντολή υπολογιστή αποτελείται από δύο μέρη: τον **κωδικό λειτουργίας** (operation code, ή για συντομία opcode) και τον **τελεστέο** (ή τελεστέους, operands). Ο κωδικός λειτουργίας καθορίζει τον τύπο της λειτουργίας που θα εκτελεστεί στον τελεστέο ή τους τελεστέους. Κάθε εντολή αποτελείται από δεκαέξι bit που χωρίζονται σε τέσσερα πεδία των 4 bit. Το αριστερότερο πεδίο περιέχει τον κωδικό λειτουργίας και τα άλλα τρία πεδία περιέχουν τον τελεστέο ή τη διεύθυνση του τελεστέου (ή των τελεστέων), όπως μπορείτε να δείτε στην Εικόνα 5.31.



Επεξεργασία των εντολών

Ο απλός υπολογιστής μας, όπως και οι περισσότεροι υπολογιστές, χρησιμοποιεί κύκλους μηχανής. Κάθε κύκλος αποτελείται από τρεις φάσεις: ανάκληση, αποκωδικοποίηση, και εκτέλεση. Κατά τη φάση της ανάκλησης, ανακαλείται από τη μνήμη η εντολή με τη διεύθυνση που καθορίζεται από τον μετρητή προγράμματος (PC), και φορτώνεται στον καταχωρητή εντολών (IR). Στη συνέχεια ο μετρητής προγράμματος αυξάνεται κατά ένα ώστε να δείχνει στην επόμενη εντολή. Κατά τη φάση της αποκωδικοποίησης, αποκωδικοποιείται η εντολή στον καταχωρητή εντολών και ανακαλούνται από τον καταχωρητή ή από τη μνήμη οι τελεστέοι. Κατά τη φάση της εκτέλεσης, η εντολή εκτελείται και τα αποτελέσματα τοποθετούνται στην κατάλληλη θέση μνήμης ή στον κατάλληλο καταχωρητή. Μετά την ολοκλήρωση της τρίτης φάσης, η μονάδα ελέγχου ξεκινά πάλι τον κύκλο, όμως τώρα ο μετρητής προγράμματος δείχνει στην επόμενη εντολή. Η διαδικασία συνεχίζεται μέχρι η ΚΜΕ να συναντήσει μια εντολή τερματισμού (HALT).

Πίνακας 5.4 Λίστα εντολών για τον απλό υπολογιστή μας

Εντολή	Κωδ.	Τελεστέοι				Ενέργεια
		d_1	d_2	d_3	d_4	
HALT	0					Διακόπται την εκτέλεση του προγράμματος
LOAD	1	R_0	M_5			$R_0 \leftarrow M_5$
STORE	2	M_0	R_5			$M_0 \leftarrow R_5$
ADDI	3	R_0	R_{51}	R_{52}		$R_0 \leftarrow R_{51} + R_{52}$
ADDF	4	R_0	R_{51}	R_{52}		$R_0 \leftarrow R_{51} + R_{52}$
MOVE	5	R_0	R_5			$R_0 \leftarrow R_5$
NOT	6	R_0	R_5			$R_0 \leftarrow \neg R_5$
AND	7	R_0	R_{51}	R_{52}		$R_0 \leftarrow R_{51} \text{ AND } R_{52}$
OR	8	R_0	R_{51}	R_{52}		$R_0 \leftarrow R_{51} \text{ OR } R_{52}$
XOR	9	R_0	R_{51}	R_{52}		$R_0 \leftarrow R_{51} \text{ XOR } R_{52}$
INC	A	R				$R \leftarrow R + 1$
DEC	B	R				$R \leftarrow R - 1$
ROTATE	C	R	n	0 ή 1		$\text{Rot}_n R$
JUMP	D	R	n			Αν $R_0 \neq R$ τότε $PC = n$, διαφορετικά συνέχεια

Υπόμνημα
 R_0, R_{51}, R_{52} : Δεκαεξαδική διεύθυνση των καταχωρητών προέλευσης
 R_5 : Δεκαεξαδική διεύθυνση του καταχωρητή προορισμού
 M_5 : Δεκαεξαδική διεύθυνση της θέσης μνήμης προέλευσης
 M_0 : Δεκαεξαδική διεύθυνση της θέσης μνήμης προορισμού
n: δεκαεξαδικός αριθμός
 d_1, d_2, d_3, d_4 : 10, 20, 30, και 40 δεκαεξαδικό ψηφίο

Ένα παράδειγμα

Ας εξετάσουμε πώς ο απλός υπολογιστής μας προσθέτει δύο ακεραίους A και B και παράγει το αποτέλεσμα C. Υποθέτουμε ότι οι ακεραίοι βρίσκονται σε μορφή συμπληρώματος ως προς δύο. Μαθηματικά, η πράξη εκφράζεται ως εξής:

$$C = A + B$$

Θεωρούμε ότι οι πρώτοι δύο ακεραίοι αποθηκεύονται στις θέσεις μνήμης $(40)_{16}$ και $(41)_{16}$ και το αποτέλεσμα θα αποθηκευτεί θέση μνήμης $(42)_{16}$. Για να πραγματοποιήσει την πρόσθεση ένα απλό πρόγραμμα χρειάζονται πέντε εντολές, οι οποίες παρουσιάζονται παρακάτω:

1. Φόρτωση των περιεχομένων του M_{40} στον καταχωρητή R_0 ($R_0 \leftarrow M_{40}$).
2. Φόρτωση των περιεχομένων του M_{41} στον καταχωρητή R_1 ($R_1 \leftarrow M_{41}$).
3. Πρόσθεση των περιεχομένων των R_0 και R_1 και τοποθέτηση του αποτελέσματος στον R_2 ($R_2 \leftarrow R_0 + R_1$).
4. Αποθήκευση των περιεχομένων του R_2 στη θέση μνήμης M_{42} ($M_{42} \leftarrow R_2$).
5. Διακοπή.

Στη γλώσσα του απλού υπολογιστή μας, οι πέντε αυτές εντολές κωδικοποιούνται ως εξής:

Κωδικός	Ερμηνεία
$(1040)_{16}$	1: LOAD 0: R_0 40: M_{40}
$(1141)_{16}$	1: LOAD 1: R_1 41: M_{41}
$(3201)_{16}$	3: ADDI 2: R_2 0: R_0 1: R_1
$(2422)_{16}$	2: STORE 42: M_{42} 2: R_2
$(0000)_{16}$	0: HALT

5.61

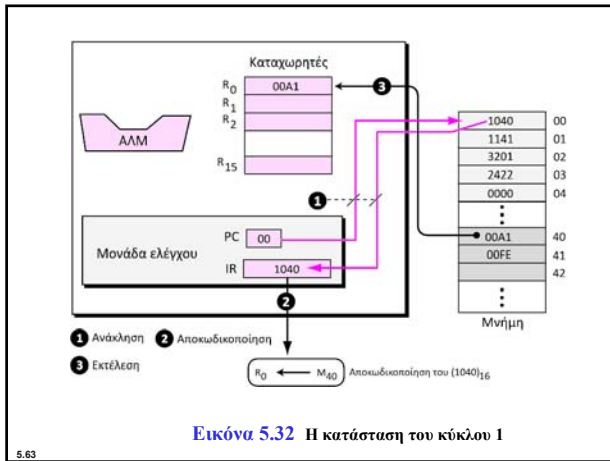
Αποθήκευση του προγράμματος και των δεδομένων

Μπορούμε να αποθηκεύσουμε το πρόγραμμα των πέντε γραμμών στη μνήμη ξεκινώντας από τη θέση $(00)_{16}$ και φτάνοντας μέχρι τη θέση $(04)_{16}$. Γνωρίζουμε ήδη πως τα δεδομένα πρέπει να αποθηκευτούν στις θέσεις μνήμης $(40)_{16}$, $(41)_{16}$, και $(42)_{16}$.

Κύκλοι

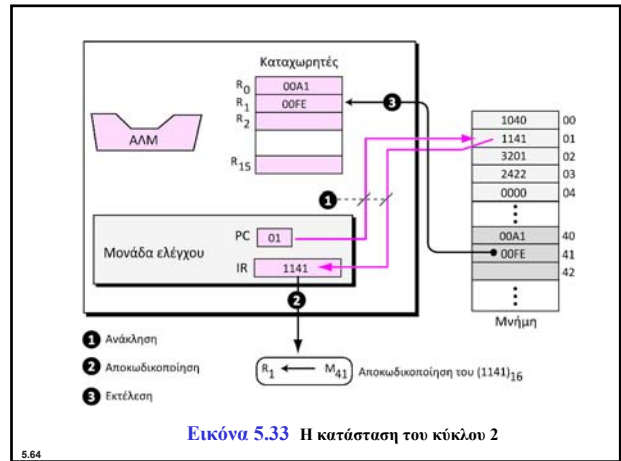
Ο υπολογιστής μας χρησιμοποιεί έναν κύκλο ανά εντολή. Επομένως, για ένα μικρό πρόγραμμα με πέντε εντολές χρειαζόμαστε πέντε κύκλους. Γνωρίζουμε επίσης πως κάθε κύκλος κανονικά αποτελείται από τρία βήματα: ανάκληση, αποκωδικοποίηση, και εκτέλεση. Προς το παρόν, ας υποθέσουμε ότι θέλουμε να εκτελέσουμε την πρόσθεση $161 + 254 = 415$. Στη μνήμη, οι αριθμοί εμφανίζονται σε δεκαεξαδική μορφή, δηλαδή $(00A1)_{16}$, $(00FE)_{16}$, και $(019F)_{16}$.

5.62



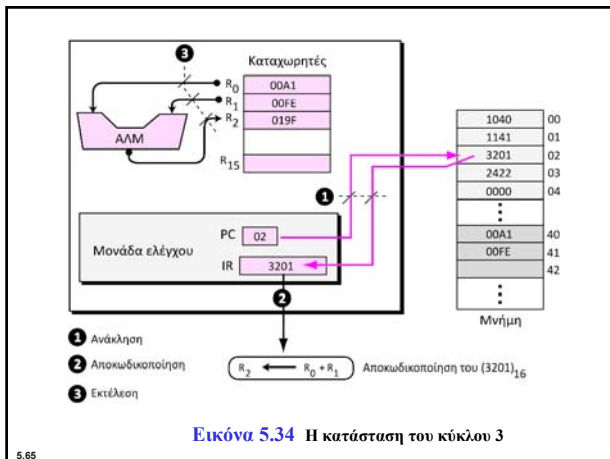
Εικόνα 5.32 Η κατάσταση του κύκλου 1

5.63



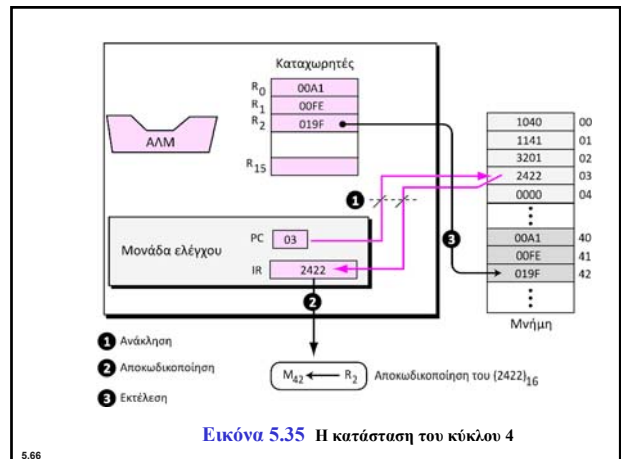
Εικόνα 5.33 Η κατάσταση του κύκλου 2

5.64



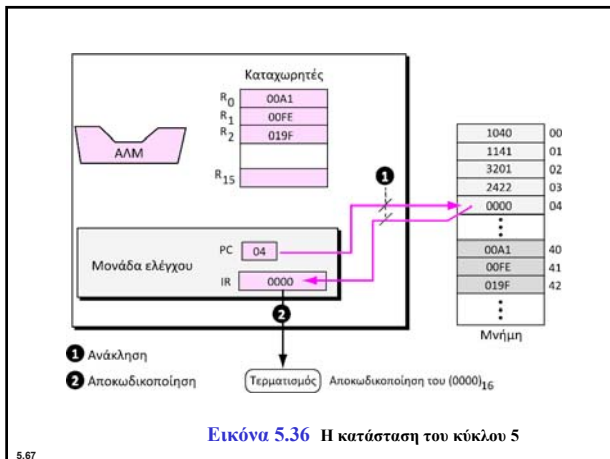
Εικόνα 5.34 Η κατάσταση του κύκλου 3

5.65



Εικόνα 5.35 Η κατάσταση του κύκλου 4

5.66



Άλλο ένα παράδειγμα

Στο προηγούμενο παράδειγμα υποθέσαμε ότι οι δύο ακεραίοι που θα προσθέταμε βρίσκονταν ήδη στη μνήμη. Επίσης υποθέσαμε ότι το αποτέλεσμα της πρόσθεσης θα φιλοξενούνταν στη μνήμη. Τώρα αναρωτηθείτε πώς μπορούμε να αποθηκεύσουμε στη μνήμη τους δύο ακεραίους που θέλουμε να προσθέσουμε, ή πώς μπορούμε να χρησιμοποιήσουμε το αποτέλεσμα μετά την αποθήκευσή του στη μνήμη. Σε ένα πραγματικό σενάριο, οι δύο ακεραίοι θα καταχωρίζονταν στη μνήμη με τη χρήση μιας συσκευής εισόδου, όπως ένα πληκτρολόγιο, και ο τρίτος ακεραίος θα μπορούσε να εμφανιστεί μέσω μιας συσκευής εξόδου, όπως μια οθόνη. Η λήψη δεδομένων μέσω μιας συσκευής εισόδου συνήθως ονομάζεται λειτουργία ανάγνωσης, ενώ η αποστολή δεδομένων σε μια συσκευή εξόδου συνήθως αναφέρεται ως λειτουργία εγγραφής. Για να κάνουμε πιο πρακτικό το προηγούμενο πρόγραμμα, θα πρέπει να το τροποποιήσουμε ως εξής:

5.68

1. Ανάγνωση ενός ακεραίου που βρίσκεται στη θέση μνήμης M_{40} .
2. $R_0 \leftarrow M_{40}$.
3. Ανάγνωση ενός ακεραίου που βρίσκεται στη θέση μνήμης M_{41} .
4. $R_1 \leftarrow M_{41}$.
5. $R_2 \leftarrow R_0 + R_1$.
6. $M_{42} \leftarrow R_2$.
7. Εγγραφή του ακεραίου από τη θέση μνήμης M_{42} .
8. Διακοπή.

Σε αυτόν τον υπολογιστή μπορούμε να προσομοιώσουμε τις λειτουργίες ανάγνωσης και εγγραφής χρησιμοποιώντας τις εντολές LOAD και STORE. Επιπλέον, οι εντολές LOAD και STORE διαβάζουν δεδομένα που τοποθετούνται στην ΚΜΕ και γράφουν δεδομένα τα οποία ανακτούν από αυτήν. Για την ανάγνωση δεδομένων στη μνήμη ή την εγγραφή δεδομένων από τη μνήμη χρειαζόμαστε δύο εντολές. Η λειτουργία ανάγνωσης είναι η εξής:

5.69

$R \leftarrow M_{FE}$ Επειδή το πληκτρολόγιο υποτίθεται ότι αποτελεί τη θέση μνήμης $(FE)_{16}$
 $M \leftarrow R$

Η λειτουργία εγγραφής είναι η εξής:

$R \leftarrow M$
 $M_{FF} \leftarrow R$ Επειδή η οθόνη υποτίθεται ότι αποτελεί τη θέση μνήμης $(FF)_{16}$

Η λειτουργία εισόδου πρέπει πάντα να διαβάζει δεδομένα από μια συσκευή εισόδου στη μνήμη· η λειτουργία εξόδου πρέπει πάντα να γράφει δεδομένα από τη μνήμη σε μια συσκευή εξόδου.

5.70

Το πρόγραμμα κωδικοποιείται ως εξής:

1	$(1FFE)_{16}$	5	$(1040)_{16}$	9	$(1F42)_{16}$
2	$(240F)_{16}$	6	$(1141)_{16}$	10	$(2FFF)_{16}$
3	$(1FFE)_{16}$	7	$(3201)_{16}$	11	$(0000)_{16}$
4	$(241F)_{16}$	8	$(2422)_{16}$		

Οι λειτουργίες 1 έως 4 αφορούν την είσοδο και οι λειτουργίες 9 και 10 χρησιμοποιούνται για την έξοδο. Όταν εκτελεστεί αυτό το πρόγραμμα, θα περιμένει από τον χρήστη να καταχωρίσει δύο ακεραίους με το πληκτρολόγιο και να πατήσει το πλήκτρο Enter. Στη συνέχεια, το πρόγραμμα υπολογίζει το άθροισμα και εμφανίζει το αποτέλεσμα στην οθόνη.

5.71