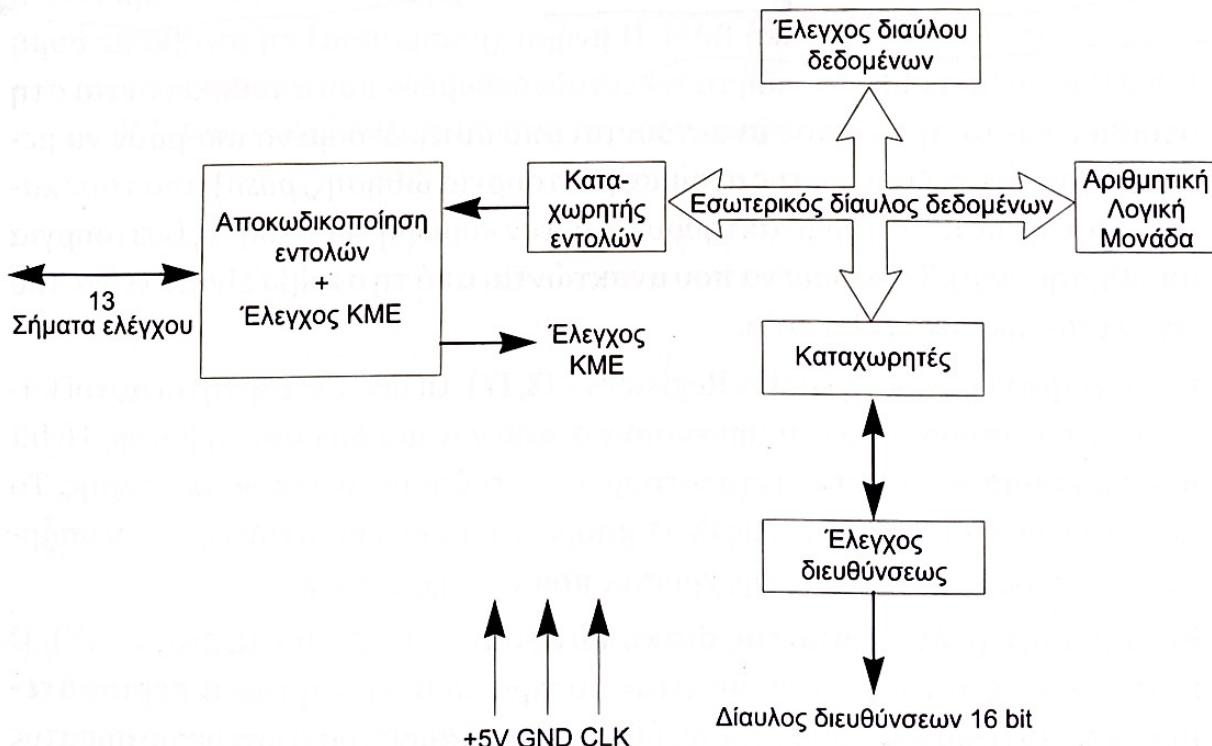


# Ο μικροεπεξεργαστής Z80

## 2.1 Εισαγωγή

Ο Z80 υλοποιήθηκε αρχικά με την τεχνολογία NMOS και το ολοκληρωμένο κύκλωμα του διαθέτει 40 ακροδέκτες που είναι συμβατοί με την τεχνολογία TTL. Λειτουργεί με μία μόνο τάση τροφοδοσίας 5 Volt και στο εμπόριο κυκλοφόρησε αρχικά με συχνότητα ρυθμού 2,5 MHz. Στο Σχήμα 2-1 παρουσιάζεται η βασική του αρχιτεκτονική.

Μέσα στην Κεντρική Μονάδα Επεξεργασίας (KME) υπάρχει ένας εσωτερικός δίαυλος δεδομένων για την εσωτερική επικοινωνία του ελεγκτή του διαύλου δεδομένων (data bus control), της αριθμητικής και λογικής μονάδας (ALM), του καταχωρητή εντολών (instruction register) και των καταχωρητών της KME (CPU registers).



Σχήμα 2-1. Δομή του Z80  
(Πηγή: Zilog Z80 Family CPU User Manual).

ters). Η εντολή που είναι αποθηκευμένη στον καταχωρητή εντολών αποκωδικοποιείται με τον ορισμό των κατάλληλων σημάτων ελέγχου. Επιπλέον, εσωτερικά οι καταχωρητές της KME διασυνδέονται με τον **ελεγκτή του διαύλου διευθύνσεων**. Το εύρος του διαύλου αυτού είναι 16 bit. Ο δίαυλος δεδομένων έχει εύρος 8 bit.

## 2.2 Προγραμματιζόμενοι καταχωρητές του Z80

### 2.2.1 Καταχωρητές ειδικής χρήσης

Οι καταχωρητές ειδικής χρήσης του Z80 είναι: ο μετρητής προγράμματος (program counter - PC) 16 bit, ο δείκτης στοίβας (stack pointer - SP) 16 bit, δύο συσσωρευτές 8 bit, οι 16 bit καταχωρητές IX και IY, ο καταχωρητής πίνακα διακοπών (I) 8 bit, ο καταχωρητής ανανέωσης μνήμης (memory refresh register) 8 bit και, τέλος, δύο καταχωρητές κατάστασης (status registers) των 8 bit ο καθένας.

- **Μετρητής προγράμματος** (Program Counter - PC): Ο μετρητής προγράμματος αποθηκεύει τη διεύθυνση 16 bit της πιο πρόσφατης εντολής που προσκομίστηκε από τη μνήμη. Ο μετρητής προγράμματος αυξάνεται αυτόματα όταν τα περιεχόμενά του μεταφερθούν στις γραμμές διευθύνσεων. Όταν υπάρχουν εντολές άλματος και διακλάδωσης, η νέα τιμή διεύθυνσης φορτώνεται αυτόματα στον μετρητή προγράμματος.
- **Δείκτης στοίβας** (Stack Pointer - SP): Ο δείκτης στοίβας αποθηκεύει τα 16 bit της διεύθυνσης της κορυφής της στοίβας (stack) η οποία μπορεί να βρίσκεται στους δύπτες στην εσωτερική RAM. Η μνήμη χρησιμοποιεί τη στοίβα με δομή LIFO (last in first out), δηλαδή τα τελευταία δεδομένα που αποθηκεύονται στη στοίβα είναι τα πρώτα που ανακτούνται από αυτή. Δεδομένα μπορούν να μεταφερθούν στην κορυφή της στοίβας (λειτουργία ώθησης, push) από τους καταχωρητές της KME ή να ανακτηθούν από την κορυφή της στοίβας (λειτουργία απώθησης, pop). Τα δεδομένα που ανακτώνται από τη στοίβα είναι εκείνα που είχαν αποθηκευτεί τελευταία.
- **Καταχωρητές δείκτη** (Index Registers - IX, IY): Οι δύο ανεξάρτητοι αυτοί καταχωρητές μπορούν να αποθηκεύσουν ο καθένας μια διεύθυνση βάσης 16 bit που χρησιμοποιείται σε δεικτοδοτούμενους τρόπους διευθυνσιοδότησης. Το περιεχόμενό κάποιου από τους IX, IY μπορεί να προστεθεί στην τιμή που υπάρχει στο πεδίο μετατόπισης της χρησιμοποιούμενης εντολής.
- **Καταχωρητής διανύσματος διακοπών** (Interrupt Vector Register - IV): Ο καταχωρητής αυτός χρησιμοποιείται για την αποθήκευση των 8 περισσότερων σημαντικών ψηφίων της διεύθυνσης χειρισμού του εξωτερικού σήματος

ΣΥΜΒΟΛΩΝ IV	ΑΝΑΖΩΓΟΝΗΣΗ ΜΝΗΜΗΣ R
	ΚΑΤΑΧ. ΔΕΙΚΤΗ IX
	ΚΑΤΑΧ. ΔΕΙΚΤΗ IY
	ΔΕΙΚΤΗΣ ΣΤΟΙΒΑΣ SP
	ΜΕΤΡΗΤΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ PC

Σχήμα 2-2. Ειδικοί καταχωρητές

διακοπής (Interrupt Service Routine – ISR). Τα 8 λιγότερο σημαντικά ψηφία της διεύθυνσης αυτής παρέχονται από την περιφερειακή συσκευή που ζητά εξυπηρέτηση με τη συγκεκριμένη διακοπή.

- **Καταχωρητής ανανέωσης μνήμης** (Memory Refresh Register – R): Στον καταχωρητή αυτόν αποθηκεύεται ένας μετρητής υπεύθυνος για την ανανέωση (refresh) δυναμικών μνημών. Με αυτόν τον καταχωρητή, η σύνδεση μιας δυναμικής μνήμης γίνεται το ίδιο απλή με εκείνη μιας στατικής. Τα 7 bit αυτού του καταχωρητή αυξάνονται αυτόματα μετά από κάθε προσκόμιση εντολής. Το όγδοο bit παραμένει όπως αρχικά προγραμματίστηκε. Ο μετρητής αυτός μπορεί να φορτωθεί και να χρησιμοποιηθεί από τον προγραμματιστή για την ανανέωση της μνήμης χωρίς να απασχολείται ο Z80 με τη διαδικασία αυτή.
- **Καταχωρητής κατάστασης** (Status Register – F): Ο καταχωρητής κατάστασης περιέχει διάφορες σημαίες (ενδείξεις), όπως για παράδειγμα αν το αποτέλεσμα είναι 0, αρνητικό, αν έχει προκύψει κρατούμενο/δανεικό κ.ά. Στο Σχήμα 2-3 παρουσιάζονται τα πεδία του καταχωρητή κατάστασης.

S	Z	X	H	X	P/V	N	C
---	---	---	---	---	-----	---	---

S: Σημαία προσήμου (sign flag)

Z: Σημαία μηδενικού αποτελέσματος (zero flag)

X: Δεν χρησιμοποιείται

H: H BCD ένδειξη κρατουμένου (half carry)

P/V: Σημαία ισοτιμίας (parity) ή υπερχείλισης (overflow)

N: Σημαία αφαίρεσης (negative flag)

C: Σημαία κρατουμένου (carry flag)

Σχήμα 2-3. Ο καταχωρητής κατάστασης F

## 2.2.2 Καταχωρητές γενικής χρήσης

Ο Z80 έχει 14 καταχωρητές γενικής χρήσης των 8 bit κατανεμημένους σε 2 σύνολα: A, B, C, D, E, H, L και A', B', C', D', E', H', L'. Σε κάθε χρονική στιγμή μόνο ένα σύνολο μπορεί να χρησιμοποιηθεί μαζί με τους αντίστοιχους καταχωρητές κατάστασης F και F'. Ο καταχωρητής A (καθώς και ο αντίστοιχός του A') θεωρείται ο **συσσωρευτής** του Z80 παρότι και οι υπόλοιποι καταχωρητές μπορούν να χρησιμοποιηθούν στις περισσότερες περιπτώσεις για την αποθήκευση τόσο του ορίσματος όσο και του αποτελέσματος μιας πράξης. Κάθε καταχωρητής μπορεί να χρησιμοποιηθεί αυτόνομα (8 bit) ή σε ζευγάρι με κάποιον άλλο ως καταχωρητής 16 bit. Οι συνδυασμοί που προκύπτουν παρουσιάζονται στο Σχήμα 2-4.

Η αρχιτεκτονική των εσωτερικών καταχωρητών του Z80 έχει τα εξής πλεονεκτήματα:

- Υπάρχει διαθέσιμος αρκετός χώρος αποθήκευσης ενδιάμεσων αποτελεσμάτων μέσα στην KME (λόγω μεγάλου πλήθους καταχωρητών) πράγμα που έχει αποτέλεσμα την αυξημένη ταχύτητα αφού η προσπέλαση σε καταχωρητές της KME γίνεται ταχύτερα απ' ότι η προσπέλαση σε δεδομένα της RAM.
- Υπάρχει η δυνατότητα γρήγορης αποθήκευσης του περιβάλλοντος στην περίπτωση διακοπών (interrupts) με την αυτόματη αποθήκευση των δεδομένων στο εναλλακτικό σύνολο καταχωρητών (swap)
- Οι αριθμητικές και λογικές πράξεις μπορούν να εκτελεστούν σε συγκεκριμένο καταχωρητή ή σε συνδυασμό με κάποιον άλλον ή με μια διεύθυνση μνήμης. Καταχωρητές σε οριζόντια ζεύγη του Σχήματος 2-4 χρησιμοποιούνται για την αποθήκευση διευθύνσεων. Π.χ., στο BC το B περιέχει τα 8 περισσότερο σημαντικά κάψηφία της διεύθυνσης και το C τα 8 λιγότερο σημαντικά.

ΚΥΡΙΟ ΣΥΝΟΛΟ ΚΑΤΑΧΩΡΗΤΩΝ		ΕΝΑΛΛΑΚΤΙΚΟ ΣΥΝΟΛΟ ΚΑΤΑΧΩΡΗΤΩΝ	
ACCUMULATOR	FLAGS	ACCUMULATOR	FLAGS
A	F	A'	F'
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'

Σχήμα 2-4. Καταχωρητές γενικού σκοπού

## 2.3 Αριθμητική και λογική μονάδα

Οι αριθμητικές και λογικές εντολές 8 bit της KME εκτελούνται από τη μονάδα αυτή. Εσωτερικά, η αριθμητική και λογική μονάδα (ALU) επικοινωνεί με τους καταχωρητές και τον δίσυλο δεδομένων. Τα κυκλώματα της αριθμητικής και λογικής μονάδας επιλέγονται και συγχρονίζονται από τα σήματα ελέγχου που παράγει η αποκωδικοποίηση μιας εντολής. Οι λειτουργίες που εκτελούνται από την ALU περιλαμβάνουν:

1. Πρόσθεση	ADD
2. Αφαίρεση	SUBTRACT
3. Λογικό ΚΑΙ	AND
4. Λογικό Ή	OR
5. Λογικό αποκλειστικό Ή	XOR
6. Σύγκριση	COMPARE
7. Αριστερή/Δεξιά μετατόπιση	LEFT/RIGHT SHIFT/ROTATE
8. Αύξηση κατά 1	INCREMENT
9. Μείωση κατά 1	DECREMENT
10. Ενεργοποίηση/απενεργοποίηση bit	SET/RESET BIT
11. Έλεγχος τιμής bit	TEST BIT

## 2.4 Τρόποι διευθυνσιοδότησης

- **Απόλυτη (Absolute):** Είναι ο απλούστερος τρόπος διευθυνσιοδότησης. Σε άλλους επεξεργαστές αυτός ο τρόπος διευθυνσιοδότησης ονομάζεται **Direct**. Οι εντολές που χρησιμοποιούν αυτή τη μέθοδο χρειάζονται τρεις διαδοχικές θέσεις μνήμης. Η πρώτη είναι ο κωδικός εντολής, ενώ οι υπόλοιπες δύο περιέχουν τη διεύθυνση μνήμης όπου βρίσκονται τα δεδομένα.  
Π.χ., LD B, (4F00): Φόρτωσε τον B με τα δεδομένα από τη διεύθυνση 4F00.
- **Άμεση (Immediate):** Υπάρχουν δύο τρόποι άμεσης διευθυνσιοδότησης, ανάλογα με το αν χρησιμοποιούνται καταχωρητές 8 ή 16 bit. Έτσι το μήκος των εντολών μπορεί να είναι δύο ή τρία byte. Στο πρώτο byte περιέχεται ο κωδικός λειτουργίας της εντολής ενώ στο δεύτερο, ή κατά περίπτωση και στο τρίτο, περιέχονται τα ίδια τα δεδομένα που θα χειριστεί η εντολή.  
Π.χ., LD A,n: Φόρτωσε τον A με την τιμή n.
- **Υπονοούμενη (Implied):** Στις εντολές όπου χρησιμοποιείται αυτού του είδους η διευθυνσιοδότηση το όρισμα υπονοείται. Οι εντολές αυτές εκτελούν συνήθως απλές λειτουργίες και δεν διαθέτουν εναλλακτικές μεθόδους διευθυνσιοδότησης.

Π.χ., ADD A,B: Πρόσθεσε την τιμή του B στον συσσωρευτή και αποθήκευσε το αποτέλεσμα στον συσσωρευτή.

- **Σχετική (Relative):** Η μέθοδος αυτή χρησιμοποιείται κυρίως από εντολές διακλάδωσης υπό συνθήκη. Το πεδίο της εντολής περιέχει έναν προσημασμένο αριθμό, δηλαδή μια μετατόπιση (offset) η οποία προστίθεται στον μετρητή προγράμματος για τον υπολογισμό της νέας διεύθυνσης από την οποία θα συνεχιστεί η εκτέλεση του προγράμματος εφόσον η συνθήκη είναι αληθής διαφορετικά, ακολουθείται η κανονική ροή του προγράμματος.
- Π.χ., JR CC,NN: Αν ισχύει η συνθήκη CC, μετάβαση στη διεύθυνση PC+NN.
- **Δεικτοδοτούμενη (Indexed):** Στο πεδίο του ορίσματος υπάρχει μια διεύθυνση η οποία αναφέρεται ως **βασική** και στην οποία προστίθεται η **μετατόπιση** (offset) για να προκύψει η τελική διεύθυνση των δεδομένων που θα χειριστεί η εντολή.
- Π.χ., LD A,(BC+offset): Φόρτωσε τον A με τα περιεχόμενα της μνήμης BC+offset
- **Έμμεση με χρήση καταχωρητή (Indirect):** Κάθε ένας από τους BC, DE, HL μπορεί να χρησιμοποιηθεί για την αποθήκευση μιας διεύθυνσης μνήμης.
- Π.χ., LD A,(HL): Φόρτωσε τον A με τα περιεχόμενα της θέσης μνήμης HL
- **Διεύθυνση bit (Bit address):** Η μέθοδος αυτή χρησιμοποιείται από εντολές που επιστρέφουν 1 ή 0 ή εξετάζουν κάποιο μεμονωμένο bit καταχωρητή ή θέσης μνήμης.

## 2.5 Ακροδέκτες και χρονισμοί

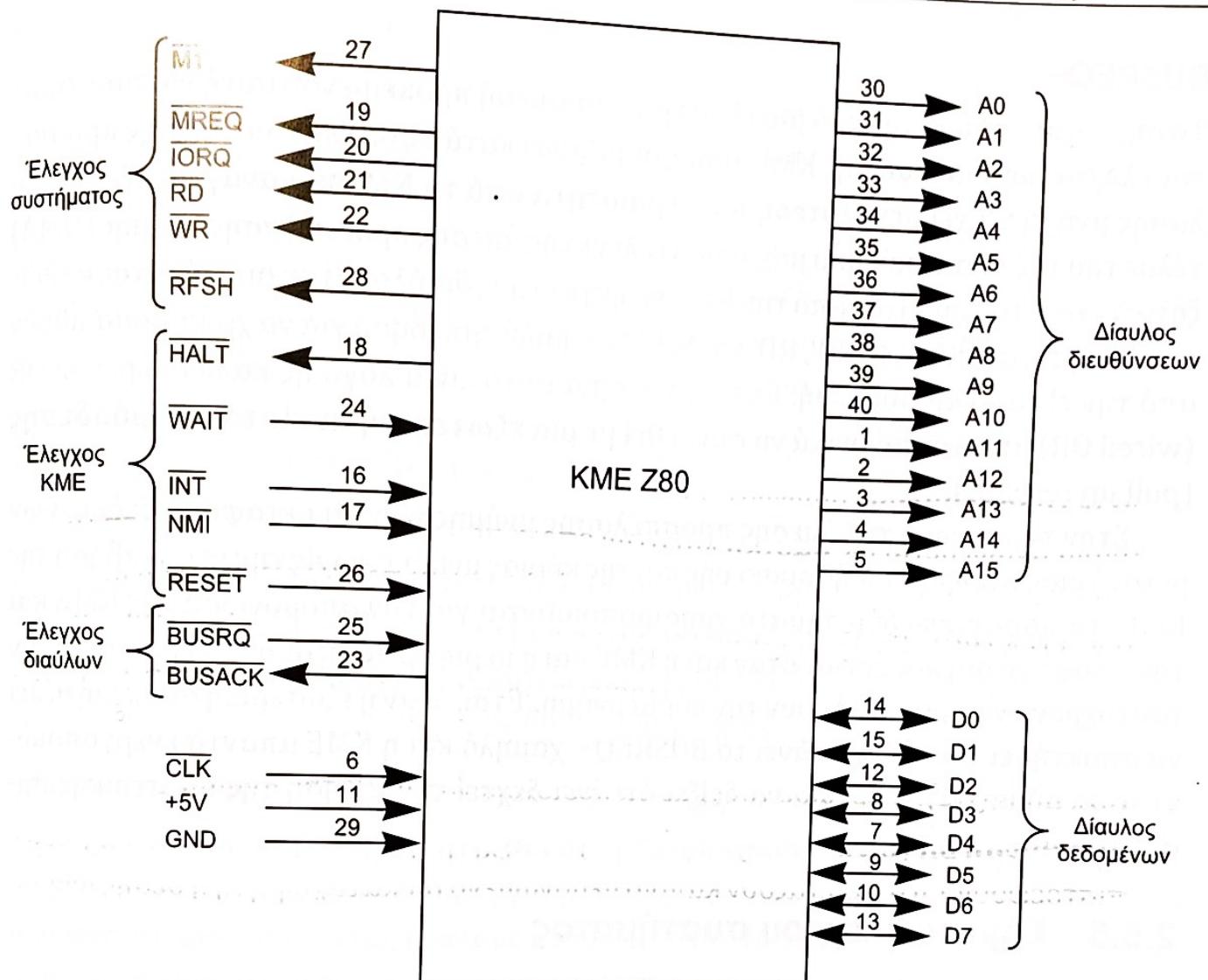
### 2.5.1 Εισαγωγή

Ο Z80 έχει 40 ακροδέκτες που παρουσιάζονται στο Σχήμα 2-5. Οι ακροδέκτες στο σχήμα είναι ομαδοποιημένοι σύμφωνα με τη λειτουργία τους και όχι σύμφωνα με τη φυσική τους διάταξη στο ολοκληρωμένο κύκλωμα.

### 2.5.2 Δίαυλος διευθύνσεων

#### A15-A0

Ο δίαυλος διευθύνσεων είναι 16 bit και αποτελεί έξοδο τριών καταστάσεων από τον επεξεργαστή. Οι διευθύνσεις που αυτός μεταφέρει αναφέρονται είτε σε **κύρια μνήμη** (μέχρι 64 Kbytes) είτε σε **διευθύνσεις περιφερειακών** (χώρος διευθυνσιοδότησης εισόδου/εξόδου). Το περισσότερο σημαντικό ψηφίο της διεύθυνσης κύριας μνήμης είναι το A15 και το λιγότερο σημαντικό το A0. Οι διευθύνσεις του χώρου E/E έχουν μήριζεται αυτή τη διακοπή σχηματίζεται ως εξής: περισσότερο σημαντικό byte λαμβάνεται από το Διάνυσμα Διακοπών (**Interrupt Vector-IV**) και το λιγότερο σημαντικό byte λαμβάνεται από το περιφερειακό.



Σχήμα 2-5. Ακροδέκτες του Z80

(Πηγή: Zilog Z80 Family CPU User Manual).

### 2.5.3 Δίαυλος δεδομένων

#### D7-D0

Ο δίαυλος δεδομένων έχει εύρος 8 bit και διπλή κατεύθυνση, όπως έχουμε εξηγήσει στην Ενότητα 1.3.

### 2.5.4 Σήματα ελέγχου διαύλων

#### BUSACK~

Το σήμα αυτό είναι ένα σήμα εξόδου και είναι ενεργό όταν είναι χαμηλό. Δείχνει στη συσκευή που ζητά τον έλεγχο του συστήματος για άμεση προσπέλαση μνήμης ότι ο δίαυλος διευθύνσεων, ο δίαυλος δεδομένων και τα σήματα MREQ', IORQ', RD' και WR' είναι σε υψηλή εμπέδηση, άρα μπορούν να οδηγηθούν από την εξωτερική συσκευή.

## BUSREQ~

Το σήμα αυτό οδηγείται από μια εξωτερική συσκευή προκειμένου αυτή να αποκτήσει τον έλεγχο των διαύλων της KME, όπως συμβαίνει κατά τη διαδικασία άμεσης προσπέλασης μνήμης. Έχει μεγαλύτερη προτεραιότητα από το NMI' και αναγνωρίζεται στο τέλος του τρέχοντος κύκλου μηχανής. Ο ελεγκτής άμεσης προσπέλασης μνήμης (DMA) ζητάει μέσω του σήματος από την KME να φέρει τους διαύλους του συστήματος καθώς και τα σήματα MREQ', IORQ', RD' και WR' σε υψηλή εμπέδηση για να χρησιμοποιηθούν από την εξωτερική συσκευή. Επειδή το σήμα αυτό είναι λογικής καλωδιωμένου OR (wired OR) απαιτεί κανονικά να συνδεθεί με μια **εξωτερική αντίσταση πρόσδεσης** (pull up resistor).

Στην περίπτωση της άμεσης προσπέλασης μνήμης γίνεται μεταφορά δεδομένων μεταξύ μιας περιφερειακής συσκευής και της κύριας μνήμης χωρίς την μεσολάβηση της KME. Τα παραπάνω δύο σήματα χρησιμοποιούνται για την απομόνωση της KME και την αποφυγή συγκρούσεων όταν και η KME και η περιφερειακή συσκευή προσπαθούν ταυτόχρονα να προσπελάσουν την κύρια μνήμη. Έτσι, όταν η εξωτερική συσκευή θέλει να αποκτήσει τον έλεγχο κάνει το BUSREQ~ χαμηλό και η KME απαντά ενεργοποιώντας το σήμα BUSACK~ για να δείξει ότι έχει δεχτεί την αίτηση της εξωτερικής συσκευής.

## 2.5.5 Σήματα ελέγχου συστήματος

### M1~

Σήμα εξόδου το οποίο, μαζί με το σήμα MREQ~, δηλώνει ότι ο τρέχον κύκλος μηχανής αφορά **προσκόμιση του κωδικού της εντολής**. Σε συνδυασμό όμως με το σήμα IORQ~ μπορεί να δώσει ένδειξη αποδοχής διακοπής (interrupt acknowledge) από την πλευρά του Z80.

### MREQ~

Έξοδος τριών καταστάσεων. Το σήμα αυτό δείχνει ότι ο δίαυλος διευθύνσεων μεταφέρει διεύθυνση για λειτουργία ανάγνωσης ή εγγραφής στην κύρια μνήμη.

### IORQ~

Είναι ένα σήμα εισόδου/εξόδου τριών καταστάσεων που δείχνει, όταν είναι ενεργό, ότι τα 8 λιγότερο σημαντικά bit του διαύλου διευθύνσεων περιέχουν μια διεύθυνση περιφερειακής συσκευής για ανάγνωση ή εγγραφή σε αυτήν.

### RD~

Σήμα εξόδου τριών καταστάσεων. Το σήμα αυτό δείχνει ότι η KME θέλει να διαβάσει δεδομένα από τη μνήμη ή από κάποια περιφερειακή συσκευή.

**WR~**

Σήμα εξόδου τριών καταστάσεων. Το σήμα αυτό δείχνει ότι ο δίαυλος δεδομένων περιέχει δεδομένα προς αποθήκευση.

**RFSH~**

Αυτό το σήμα εξόδου, μαζί με το MREQ~, δείχνει ότι τα 7 λιγότερο σημαντικά ψηφία του διαύλου διευθύνσεων του συστήματος μπορούν να χρησιμοποιηθούν σαν διεύθυνση αναζωγόνησης στις δυναμικές μνήμες του συστήματος.

## 2.5.6 Σήματα ελέγχου KME

**HALT~**

Είναι ένα σήμα εξόδου που είναι ενεργό όταν είναι χαμηλό και δείχνει ότι ο Z80 έχει εκτελέσει μια εντολή «παύσης» και βρίσκεται σε αναμονή κάποιας διακοπής πριν συνεχιστεί η λειτουργία. Σε αυτή την κατάσταση, ο Z80 εκτελεί εντολές **nop** για να διατηρήσει τη λειτουργία ανανέωσης της δυναμικής μνήμης.

**WAIT~**

Το σήμα εισόδου αυτό δείχνει στο Z80 ότι η διευθυνσιοδοτημένη μνήμη ή οι περιφερειακές συσκευές δεν είναι έτοιμες για μεταφορά δεδομένων. Ο Z80 παραμένει στην ίδια κατάσταση εισάγοντας κύκλους αναμονής στον τρέχοντα κύκλο μηχανής όσο το σήμα αυτό είναι ενεργό.

**INT~ (Interrupt Request)**

Αυτό το σήμα εισόδου οδηγείται από τις περιφερειακές συσκευές. Η KME επιτρέπει τη διακοπή εφόσον είναι ενεργή η κατάσταση του εσωτερικού προγραμματιζόμενου δισταθούς (flip-flop, IFF1) που δείχνει αν επιτρέπονται οι διακοπές. Το σήμα αυτό πρέπει κανονικά να συνδεθεί σε μια εξωτερική αντίσταση πρόσδεσης (pull-up resistor).

**NMI~ (Non Maskable Interrupt)**

Είναι σήμα εισόδου για διακοπές μεγαλύτερης προτεραιότητας από το INT~ το οποίο ελέγχεται στο τέλος της εκτέλεσης κάθε εντολής ανεξάρτητα από την κατάσταση του IFF1, που δείχνει ότι οι διακοπές είναι ενεργές. Επιβάλει στο Z80 να εκτελέσει άλμα στη διεύθυνση 0066h όπου υπάρχει η ρουτίνα εξυπηρέτησης της διακοπής.

**RESET~**

Το σήμα αυτό είναι είσοδος και αρχικοποιεί το Z80: μηδενίζει το IFF1, τον μετρητή προγράμματος, και τους καταχωρητές διανύσματος διακοπών (I) και ανανέωσης της μνήμης (R), ενώ θέτει την κατάλληλη τιμή στην κατάσταση των διακοπών. Κατά τη

διάρκεια αυτής της αρχικοποίησης οι δίαυλοι δεδομένων και διευθύνσεων μεταβαίνουν σε υψηλή εμπέδηση και όλα τα σήματα ελέγχου σε ανενεργές καταστάσεις. Απαιτούνται τουλάχιστον 3 κύκλοι μηχανής για να ολοκληρωθεί η διαδικασία αρχικοποίησης του σήματος Reset.

## CLK

Είναι το μονοφασικό ρολόι συστήματος.

### 2.5.7 Χρονισμοί

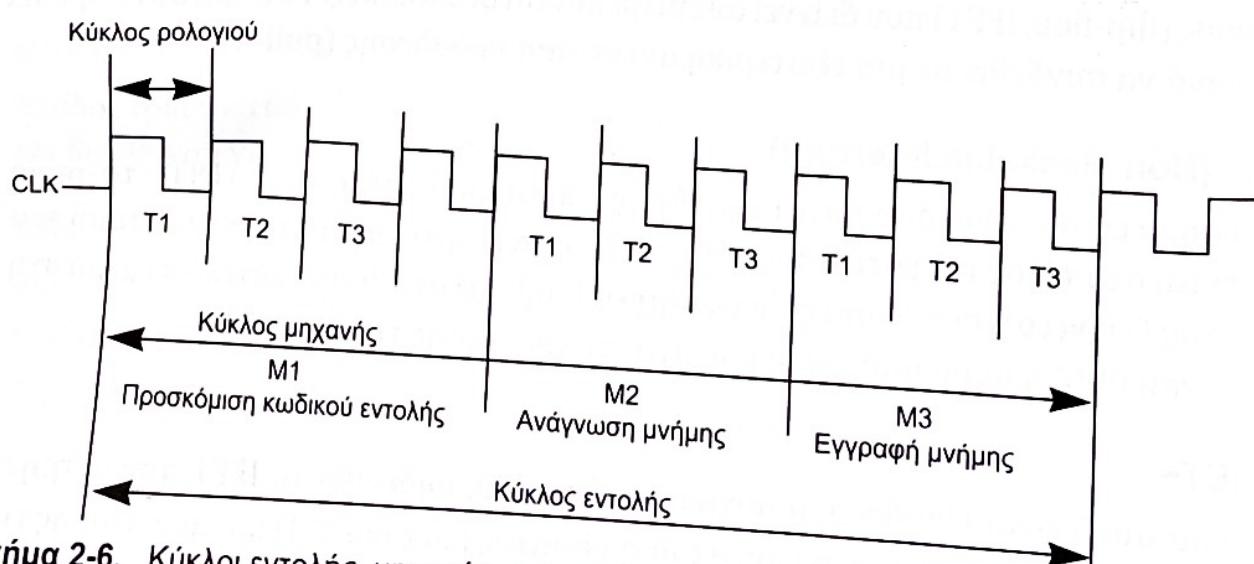
#### 2.5.7.1 Γενικά

Η αλληλεπίδραση του Z80 με τα εξωτερικά κυκλώματα περιλαμβάνουν μερικές πολύ απλές λειτουργίες όπως:

- ανάγνωση ή εγγραφή κύριας μνήμης,
- ανάγνωση ή εγγραφή περιφερειακής συσκευής,
- αναγνώριση διακοπής (interrupt acknowledge).

Όλες οι εντολές περιλαμβάνουν μία ή περισσότερες από τις παραπάνω λειτουργίες. Κάθε μία από τις λειτουργίες αυτές μπορεί να διαρκέσει από τρεις μέχρι έξι περιόδους ρολογιού. Αυτός ο χρόνος μπορεί να επεκταθεί ώστε ο Z80 να υποστηρίξει την ταχύτητα των αργών περιφερειακών συσκευών ή μνήμης.

Οι βασικές περίοδοι ρολογιού αναφέρονται ως T (κύκλοι ρολογιού – clock cycles) και οι βασικές λειτουργίες που προαναφέραμε ως M (κύκλοι μηχανής – machine cycles). Στο Σχήμα 2-6 φαίνεται πως μια τυπική εντολή είναι απλά μια σειρά συγκεκριμένων M και T κύκλων. Η εντολή του σχήματος αποτελείται από τρεις κύκλους μηχανής M.



**Σχήμα 2-6.** Κύκλοι εντολής, μηχανής, ρολογιού  
(Πηγή: Zilog Z80 Family CPU User Manual)

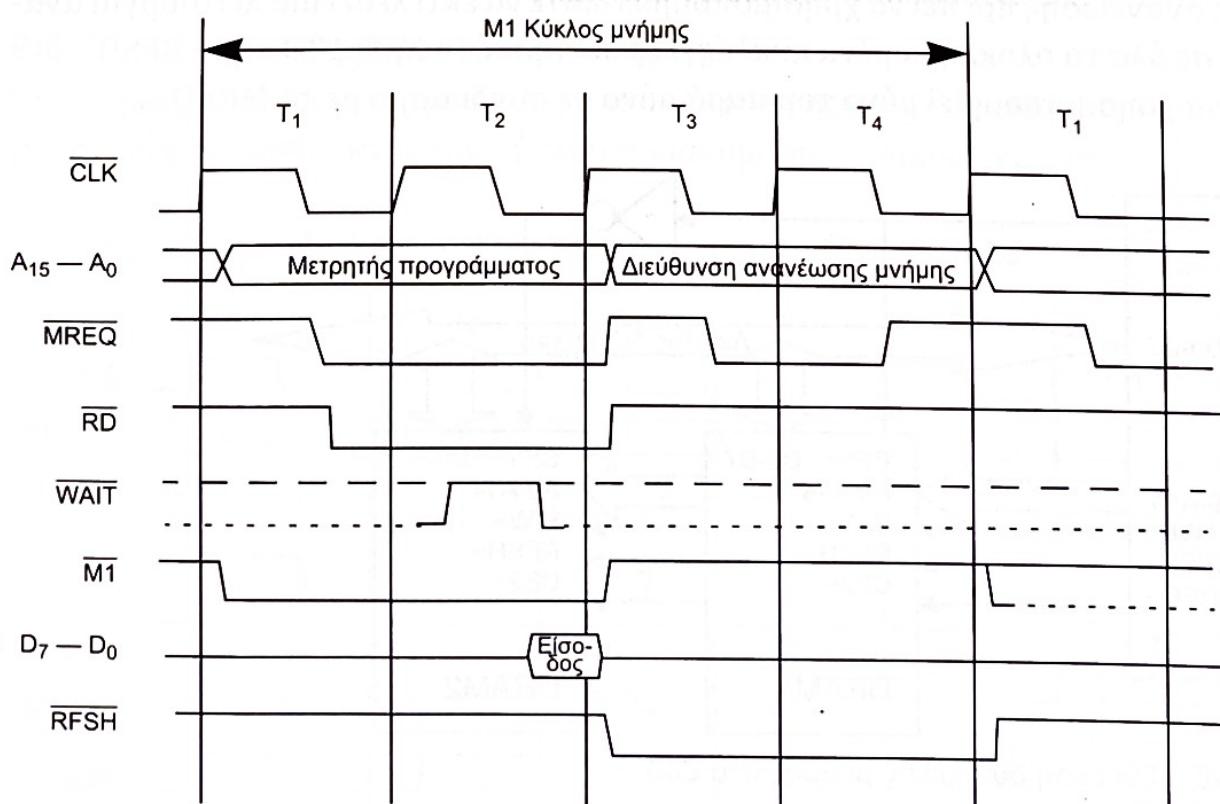
Ο πρώτος κύκλος M κάθε εντολής είναι ένας κύκλος προσκόμισης κωδικού εντολής (operational code fetch), ο οποίος αποτελείται από 4, 5 ή 6 κύκλους T, εκτός αν ο χρόνος αυτός παραταθεί από ένα σήμα **αναμονής** (wait) όπως θα εξηγηθεί παρακάτω.

Ο κύκλος προσκόμισης M1 χρησιμοποιείται για την προσκόμιση του κωδικού λειτουργίας (OP code) της επόμενης εντολής που πρόκειται να εκτελεστεί. Οι επόμενοι κύκλοι μηχανής μετακινούν (αντιγράφουν) δεδομένα ανάμεσα στο Z80 και τη μνήμη ή τις περιφερειακές συσκευές και μπορούν να περιλαμβάνουν από τρεις έως πέντε κύκλους T. Ομοίως, και αυτοί οι κύκλοι μηχανής μπορεί να παραταθούν με καταστάσεις αναμονής για την υποστήριξη αργών περιφερειακών συσκευών.

Στη συνέχεια θα περιγράψουμε τα διαγράμματα χρονισμού των βασικών κύκλων μηχανής. Κατά τη διάρκεια του T2 και κάθε διαδοχικού κύκλου αναμονής Tw, ο Z80 δειγματοληπτεί τη γραμμή ελέγχου WAIT κατά την καθοδική ακμή του ρολογιού. Αν η γραμμή WAIT είναι ενεργή θα εισαχθεί νέα κατάσταση αναμονής κατά τη διάρκεια του επομένου κύκλου. Με τη χρήση αυτής της τεχνικής, η λειτουργία ανάγνωσης μπορεί να επιμηκυνθεί έτσι ώστε να συμφωνεί με τον χρόνο προσπέλασης κάθε τύπου μνήμης ή περιφερειακής συσκευής.

### 2.5.7.2 Προσκόμιση εντολής

Το Σχήμα 2-7 δείχνει τον χρονισμό κατά τη διάρκεια ενός κύκλου M1 (προσκόμιση του κωδικού OP).



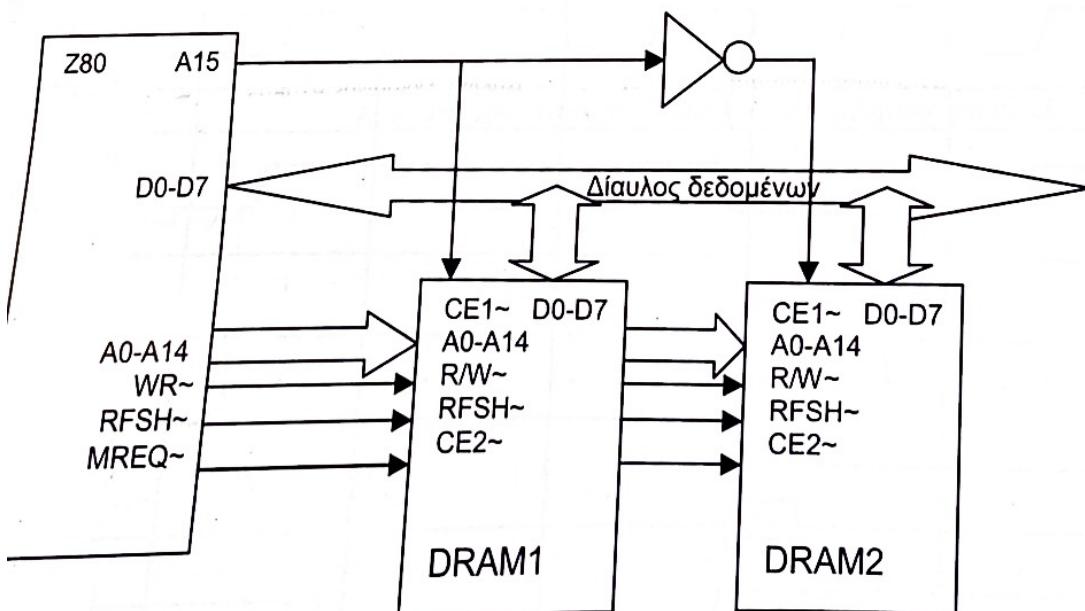
Σχήμα 2-7. Προσκόμιση εντολής  
(Πηγή: Zilog Z80 Family CPU User Manual).

Ο μετρητής προγράμματος τοποθετείται στον δίαυλο διεύθυνσεων στην αρχή του κύκλου M1. Μισό κύκλο ρολογιού αργότερα ενεργοποιείται το σήμα MREQ~. Εκείνη τη στιγμή, η διεύθυνση στη μνήμη έχει ήδη σταθεροποιηθεί και έτσι η αρνητική ακμή του MREQ~ μπορεί να χρησιμοποιηθεί απευθείας ως ρολόι ενεργοποίησης ολοκληρωμένου για τις δυναμικές μνήμες. Η γραμμή RD~ επίσης ενεργοποιείται για να δείξει ότι τα προς ανάγνωση δεδομένα της μνήμης είναι έτοιμα στον δίαυλο δεδομένων του Z80.

Ο Z80 δειγματοληπτεί τα δεδομένα από τη μνήμη στον δίαυλο δεδομένων στη θετική ακμή του ρολογιού του κύκλου T3 και η ίδια ακμή χρησιμοποιείται από την KME ληφία των δεδομένων από το Z80 προτού το σήμα RD~ απενεργοποιηθεί. Οι κύκλοι T3 και T4 ενός κύκλου προσκόμισης χρησιμοποιούνται για την ανανέωση δυναμικών μνημών (η KME αξιοποιεί αυτό τον χρόνο για να αποκωδικοποιήσει και να εκτελέσει την ήδη προσκομισμένη εντολή).

Κατά τη διάρκεια των κύκλων T3 και T4 τα επτά λιγότερο σημαντικά ψηφία του διαύλου δεδομένων περιέχουν μια διεύθυνση ανανέωσης μνήμης και το σήμα RFSH~ γίνεται ενεργό για να ωθήσει τις δυναμικές μνήμες να πραγματοποιήσουν τη λειτουργία αυτή στη συγκεκριμένη περίοδο.

Πρέπει να σημειωθεί ότι το σήμα RD~ δεν ενεργοποιείται κατά τη διάρκεια της διαδικασίας ανανέωσης προκειμένου να αποτρέψει δεδομένα από διαφορετικά τμήματα μνήμης να τοποθετηθούν στον δίαυλο δεδομένων. Το σήμα MREQ~ κατά τη διάρκεια της ανανέωσης πρέπει να χρησιμοποιηθεί ώστε να εκτελέσει μια λειτουργία ανανέωσης σε όλα τα ολοκληρωμένα κυκλώματα δυναμικής μνήμης. Το σήμα RFSH~ δεν μπορεί να χρησιμοποιηθεί μόνο του παρά μόνο σε συνδυασμό με το MREQ~.



μα 2-8. Σύνδεση δυναμικής μνήμης στο Z80

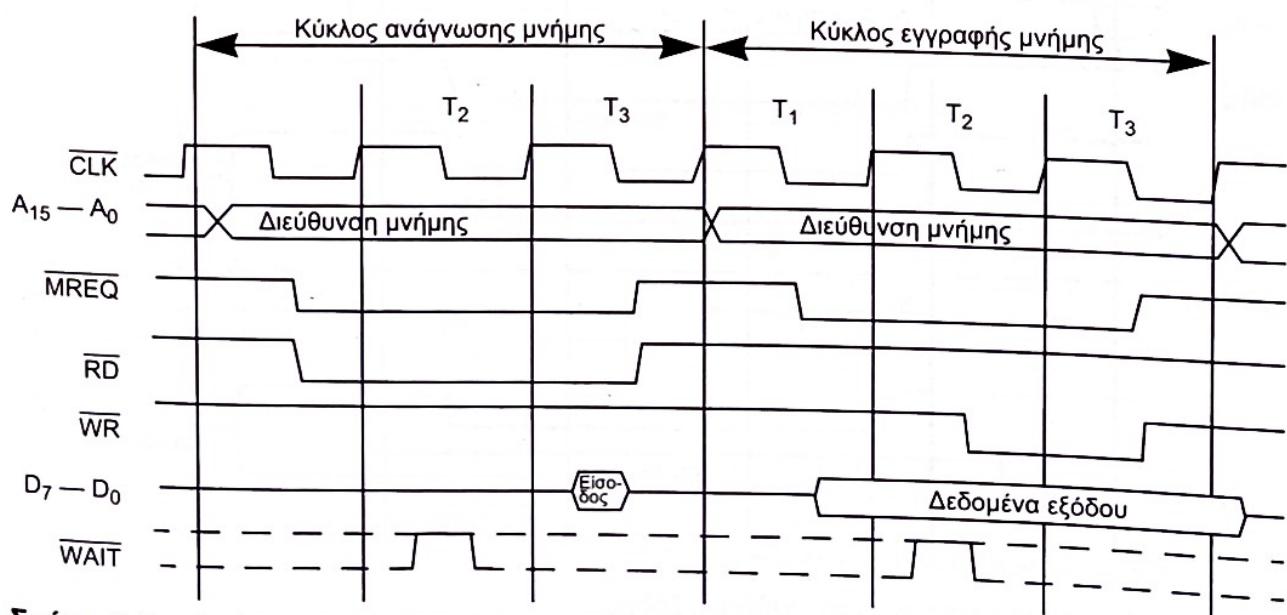
## Παράδειγμα 2.1

Στο Σχήμα 2-8 παρουσιάζεται η σύνδεση δύο ολοκληρωμένων κυκλωμάτων δυναμικής μνήμης των 32KB η κάθε μία. Και οι δύο αυτές μνήμες διαθέτουν δύο εισόδους ενεργοποίησης (Chip Enable 1 και 2) οι οποίες πρέπει να είναι και οι δύο ενεργές για να επιλεγεί το συγκεκριμένο ολοκληρωμένο κύκλωμα. Οι είσοδοι CE1~ έχουν χρησιμοποιηθεί για την αποκωδικοποίηση και ενεργοποιούνται με κάποια συγκεκριμένη τιμή του A15. Οι είσοδοι CE2~ συνδέονται με το σήμα MREQ~ ώστε η ενεργοποίηση να πραγματοποιείται μόνο κατά την προσπέλαση της κύριας μνήμης.

Η DRAM1 ενεργοποιείται όταν το A15 είναι 0 ενώ η DRAM2 όταν το A15 είναι 1. Με άλλα λόγια, η πρώτη μνήμη καταλαμβάνει τις διευθύνσεις 0-7FFFh, ενώ η δεύτερη τις 8000h-FFFFh. Όμως, σε κάθε κύκλο προσκόμισης εντολής, και ανεξάρτητα από ποια συγκεκριμένη διεύθυνση μεταξύ 0 και FFFFh έχει προσπελάσει ο Z80 για την ανάκληση της εντολής, και οι δύο μνήμες ανανεώνονται στην διεύθυνση που δείχνουν οι 8 λιγότερο σημαντικές γραμμές του διαύλου διευθύνσεων. Η διεύθυνση ανανέωσης είναι η τρέχουσα τιμή του καταχωρητή Refresh (ο οποίος αυξάνεται αυτόματα) και εμφανίζεται στον δίαυλο διευθύνσεων στους κύκλους T3 και T4. Η τιμή του διαύλου δεδομένων την περίοδο αυτή αγνοείται. Όπως γίνεται φανερό από τα παραπάνω, η ενεργοποίηση του σήματος RFSH~ υπερισχύει των CE1~ και CE2~ αφού η ανανέωση πραγματοποιείται ακόμα κι αν αυτά τα σήματα είναι ανενεργά. Με τη μέθοδο αυτή ανανεώνονται πολλές θέσεις μνήμης ταυτόχρονα σε διαφορετικά ολοκληρωμένα κυκλώματα.

### 2.5.7.3 Εγγραφή-Ανάγνωση μνήμης

Στο Σχήμα 2-9 φαίνεται ο χρονισμός των σημάτων της μνήμης κατά τους κύκλους ανάγνωσης ή εγγραφής εκτός του κύκλου προσκόμισης κωδικού εντολής.



Σχήμα 2-9. Ανάγνωση και εγγραφή μνήμης  
(Πηγή: Zilog Z80 Family CPU User Manual).

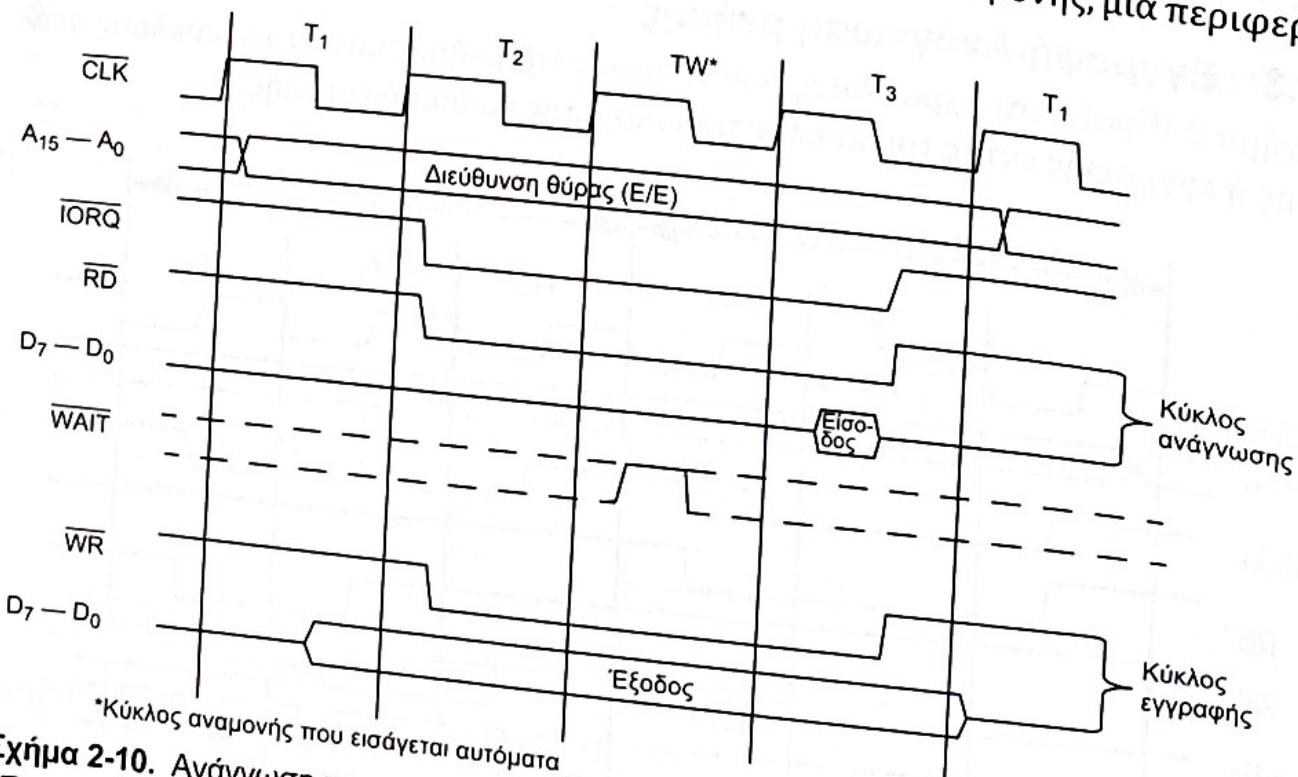
Αυτοί οι κύκλοι γενικά έχουν διάρκεια τρεις περιόδους ρολογιού εκτός αν απαιτούνται καταστάσεις αναμονής από τη μνήμη μέσω του σήματος WAIT~.

Τα σήματα MREQ~ και RD~ χρησιμοποιούνται όπως και στον κύκλο προσκόμισης. Στην περίπτωση του κύκλου εγγραφής στη μνήμη, το σήμα MREQ~ γίνεται επίσης ενεργό όταν ο δίαυλος δεδομένων βρίσκεται σε μια σταθερή κατάσταση έτσι ώστε να μπορεί να χρησιμοποιηθεί άμεσα ως σήμα ενεργοποίησης για ολοκληρωμένα κυκλώματα δυναμικών μνημών. Η γραμμή WR~ είναι ενεργή όταν τα δεδομένα στον δίαυλο δεδομένων είναι σταθερά έτσι ώστε να μπορούν να χρησιμοποιηθούν άμεσα ως παλμός για εγγραφή ή ανάγνωση σε κάθε τύπο μνήμης. Επιπλέον, το σήμα WR~ απενεργοποιείται στη μέση του T2 προτού τα περιεχόμενα των διαύλων δεδομένων και διευθύνσεων μεταβληθούν έτσι ώστε να ικανοποιούνται οι απαιτήσεις κάθε μνήμης.

#### 2.5.7.4 Κύκλοι εισόδου-εξόδου

Το Σχήμα 2-10 παρουσιάζει μια λειτουργία ανάγνωσης ή εγγραφής περιφερειακής συσκευής εισόδου-εξόδου.

Παρατηρούμε ότι κατά τη διάρκεια της λειτουργίας εισόδου-εξόδου εισάγεται αυτόματα ένας κύκλος αναμονής (Wait State). Αυτό συμβαίνει διότι, κατά τη διάρκεια των λειτουργιών εισόδου/εξόδου περιφερειακών συσκευών, ο σύντομος χρόνος από τη στιγμή που το σήμα IORQ~ ενεργοποιείται ως τη στιγμή που ο Z80 πραγματοποιεί δειγματοληψία στη γραμμή WAIT~ δεν είναι συχνά αρκετός για να ανταποκριθεί μια περιφερειακή συσκευή. Χωρίς αυτόν τον επιπλέον χρόνο αναμονής, μια περιφερειακή



Σχήμα 2-10. Ανάγνωση και εγγραφή εισόδου/εξόδου  
(Πηγή: Zilog Z80 Family CPU User Manual).

συσκευή δεν προλαβαίνει να αποκωδικοποιήσει την γραμμή WAIT~ για να διακρίνει αν χρειάζεται εισαγωγή σε κατάσταση αναμονής (wait state). Επίσης, χωρίς αυτή την κατάσταση αναμονής θα ήταν δύσκολο να σχεδιαστούν συσκευές εισόδου/εξόδου τεχνολογίας MOS οι οποίες να μπορούν να λειτουργήσουν με τη μέγιστη ταχύτητα του Z80. Κατά τη διάρκεια λειτουργίας ανάγνωσης περιφερειακής συσκευής E/E, η γραμμή RD~ χρησιμοποιείται για να επιτρέψει στη διευθυνσιοδοτούμενη συσκευή να τροφοδοτήσει δεδομένα στον αντίστοιχο δίαυλο.

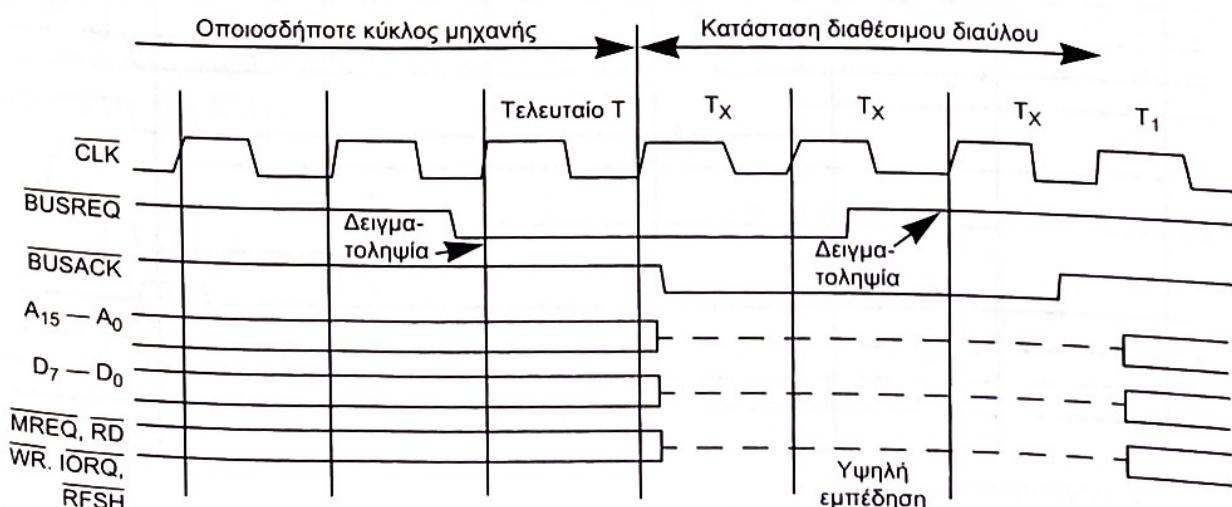
Για λειτουργίες εγγραφής από περιφερειακές συσκευές, η γραμμή WR~ χρησιμοποιείται ως ρολόι στη συσκευή εισόδου/εξόδου.

### 2.5.7.5 Απαίτηση διαύλου-κύκλος ανάγνωσης

Το Σχήμα 2-11 δείχνει τον χρονισμό της απαίτησης διαύλου (Bus Request) για άμεση προσπέλαση μνήμης (DMA). Το σήμα BUSREQ~ δειγματοληπτείται από τον Z80 με τη θετική ακμή της τελευταίας περιόδου του ρολογιού. Αν το σήμα BUSREQ~ είναι ενεργό, τότε ο Z80 θα φέρει τη διεύθυνση, τα δεδομένα, και τα σήματα ελέγχου τριών καταστάσεων σε κατάσταση υψηλής εμπέδησης, με τη θετική ακμή του ρολογιού. Από εκείνη τη στιγμή κάθε εξωτερική συσκευή μπορεί να οδηγήσει τους διαύλους για μεταφορά δεδομένων ανάμεσα σε μνήμες και περιφερειακές συσκευές.

Ο μέγιστος χρόνος που χρειάζεται ο Z80 για να απαντήσει σε μία απαίτηση διαύλου είναι ίσος με έναν κύκλο μηχανής και ένας ελεγκτής άμεσης προσπέλασης μνήμης που ελέγχει εξωτερικά τον δίαυλο μπορεί να διατηρήσει τον έλεγχο για όσους κύκλους επιθυμεί. Παρόλα αυτά πρέπει να σημειωθεί ότι, αν πραγματοποιηθεί άμεση προσπέλαση μνήμης μεγάλης διάρκειας, ο ελεγκτής DMA θα πρέπει να αναλάβει τη διαδικασία ανανέωσης των δυναμικών μνημών.

Κατά τη διάρκεια ενός κύκλου απαίτησης διαύλου, ο Z80 δεν μπορεί να διακοπεί από σήματα NMI~ και INT~.



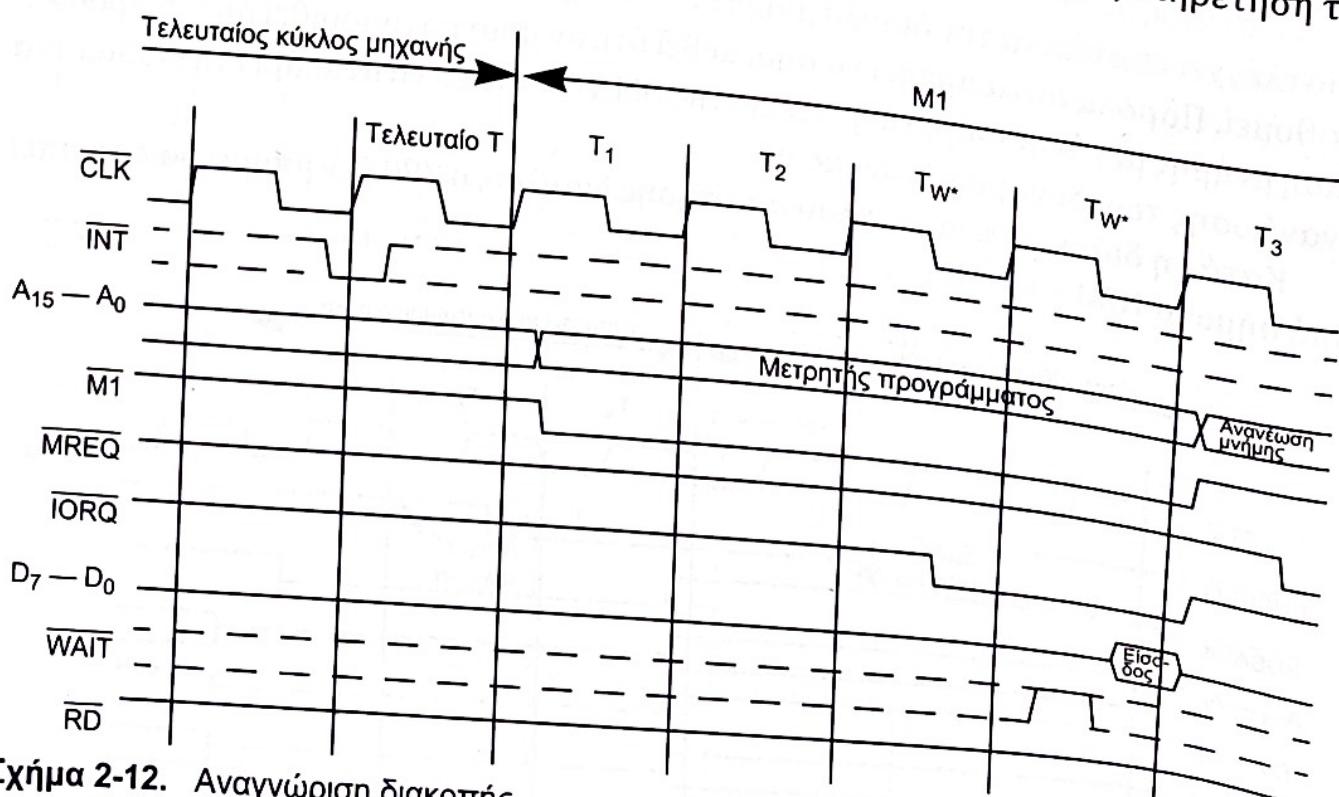
Σχήμα 2-11. Απαίτηση διαύλου  
(Πηγή: Zilog Z80 Family CPU User Manual).

### 2.5.7.6 Απαίτηση διακοπής – κύκλος ανάγνωσης

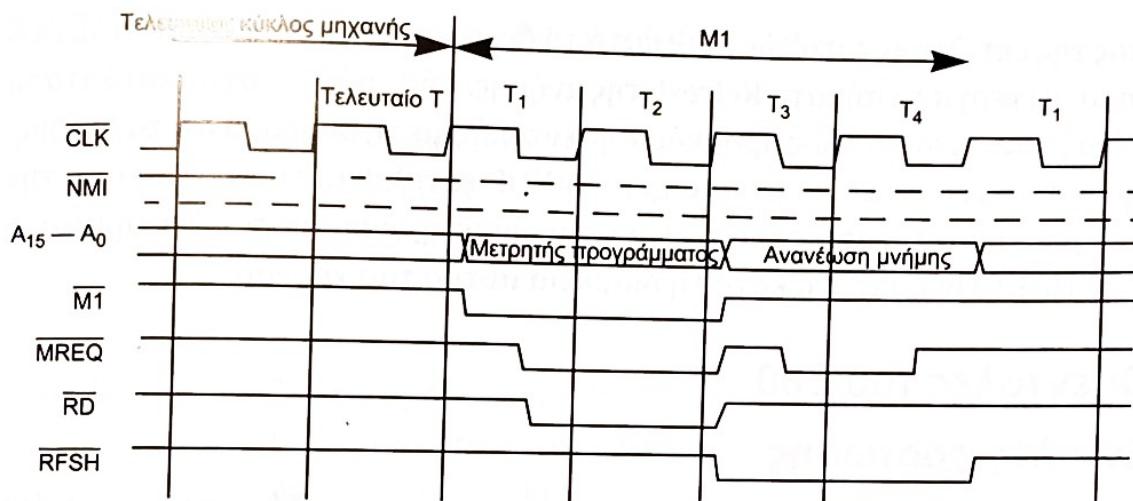
Το σήμα διακοπής INT~ δειγματοληπτείται από την KME με τη θετική ακμή του κάθε ρολογιού στο τέλος κάθε εντολής. Το σήμα δεν θα γίνει αποδεκτό αν το εσωτερικό δι-σταθές (flip flop) της KME που επιτρέπει ή δεν επιτρέπει τις διακοπές βρίσκεται στην κατάσταση 0 ή αν το σήμα BUSREQ~ είναι ενεργό. Όταν το σήμα γίνει αποδεκτό, εμφανίζεται ένας ειδικός κύκλος μηχανής. Κατά τη διάρκεια αυτού του ειδικού κύκλου το σήμα IORQ~ γίνεται ενεργό αντί του MREQ~ έτσι ώστε να δείξει ότι η συσκευή που προκάλεσε τη διακοπή μπορεί να τοποθετήσει τα 8 λιγότερο σημαντικά ψηφία της δι-μειωθεί ότι σε αυτό τον κύκλο προστίθενται αυτόματα δύο καταστάσεις αναμονής (wait states). Οι δύο καταστάσεις αναμονής δίνουν άνεση χρόνου ώστε να αναγνωρι-της διεύθυνσης της ρουτίνας εξυπηρέτησης διακοπής. Περισσότερες λεπτομέρειες θα δοθούν στο κεφάλαιο όπου περιγράφονται οι διακοπές του Z80.

### 2.5.7.7 Απόκριση σε διακοπή τύπου NMI (Non Maskable Interrupt)

Το Σχήμα 2-13 παρουσιάζει τον κύκλο απαίτησης και αναγνώρισης για τη διακοπή NMI~. Το σήμα δειγματοληπτείται την ίδια στιγμή με τη γραμμή διακοπών INT~, αλλά αυτή η γραμμή έχει προτεραιότητα σε σχέση με την κανονική και η εξυπηρέτησή της δεν μπορεί να απαγορευτεί από το λογισμικό.



Σχήμα 2-12. Αναγνώριση διακοπής  
(Πηγή: Zilog Z80 Family CPU User Manual).



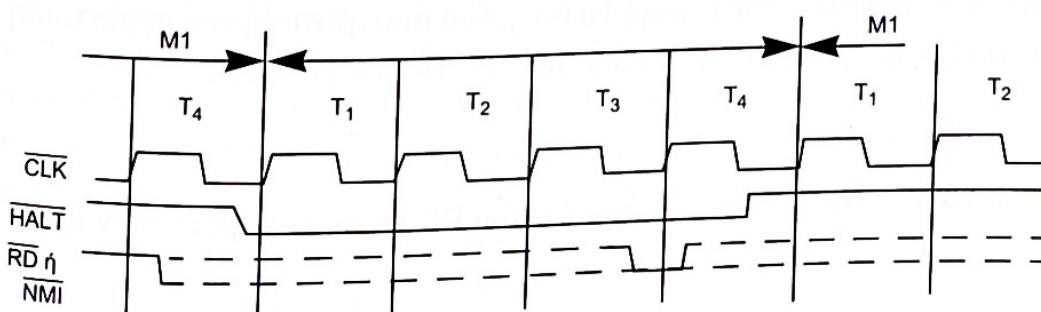
**Σχήμα 2-13.** Αναγνώριση NMI  
(Πηγή: Zilog Z80 Family CPU User Manual)

Η συνήθης χρήση του είναι να προκαλεί άμεση αντίδραση σε σημαντικά σήματα όπως, για παράδειγμα, μια πτώση της τροφοδοσίας. Η απόκριση του Z80 σε μια τέτοια αίτηση διακοπής είναι παρόμοια με μια κανονική λειτουργία ανάγνωσης μνήμης. Η μόνη διαφορά είναι ότι το περιεχόμενο του διαύλου δεδομένων αγνοείται, ενώ αυτόματα ο επεξεργαστής αποθηκεύει τον μετρητή προγράμματος στον εξωτερικό σωρό και πραγματοποιεί άλμα στη διεύθυνση 0066h. Η ρουτίνα εξυπηρέτησης για μια τέτοια διακοπή πρέπει να αρχίζει σε αυτή τη διεύθυνση.

### 2.5.7.8 Έξοδος από την εντολή HALT

Αφού εκτελεστεί μια εντολή HALT, ο Z80 εκτελεί εντολές NOP μέχρι να δεχτεί μια διακοπή. Οι δύο γραμμές διακοπών (INT~, NMI~) δειγματοληπτούνται με την ανοδική ακμή του ρολογιού κατά τη διάρκεια του κύκλου T<sub>4</sub>, όπως φαίνεται στο Σχήμα 2-14.

Όταν υπάρχει μια αποδεκτή αίτηση διακοπής, τότε η KME θα εξέλθει από την κατάσταση HALT στην επόμενη ανοδική ακμή του ρολογιού. Ο επόμενος κύκλος θα είναι ένας κύκλος αναγνώρισης της διακοπής ο οποίος θα συμφωνεί με τον τύπο διακοπής που δέχτηκε η KME.



**Σχήμα 2-14.** Εκτέλεση εντολής Halt  
(Πηγή: Zilog Z80 Family CPU User Manual).

Ο σκοπός της εκτέλεσης εντολών NOP κατά τη διάρκεια της κατάστασης HALT είναι να διατηρούν ενεργά τα σήματα Refresh της μνήμης. Κάθε κύκλος στην κατάσταση HALT είναι ένας κανονικός κύκλος προσκόμισης εντολής, με τη διαφορά ότι τα δεδομένα από τη μνήμη αγνοούνται και ότι μια εντολή NOP εκτελείται εσωτερικά από την KME και δεν προσκομίζεται από κάποια περιοχή μνήμης προγράμματος. Το σήμα αναγνώρισης του HALT είναι ενεργό κατά τη διάρκεια αυτού του χρόνου.

## 2.6 Οι εντολές του Z80

### 2.6.1 Εντολές φόρτωσης

Τις εντολές της κατηγορίας αυτής μπορούμε να τις χωρίσουμε σε δύο υποκατηγορίες: σε αυτή που περιλαμβάνει τις εντολές φόρτωσης 8 bit και σε αυτή που περιλαμβάνει τις εντολές φόρτωσης 16 bit. Οι εντολές που ανήκουν στην πρώτη κατηγορία είτε αποφορτώνουν κάποιο καταχωρητή του Z80 σε κάποιον άλλο καταχωρητή, είτε αποθηκεύουν το περιεχόμενο ενός καταχωρητή του Z80 σε κάποιον άλλο καταχωρητή, είτε χόμενο ενός καταχωρητή στη μνήμη.

#### Παραδείγματα

- LD (DE),C: αποθήκευσε το περιεχόμενο του καταχωρητή C στη διεύθυνση της κύριας μνήμης που είναι αποθηκευμένη στο ζεύγος DE.
- LD (address),A: αποθήκευσε το περιεχόμενο του συσσωρευτή στη θέση μνήμης address.
- LD (address), HL: αποθήκευσε το περιεχόμενο των καταχωρητών H και L στη θέση κύριας μνήμης address+1 και address αντίστοιχα.
- LD A,(address): φόρτωσε τον συσσωρευτή με το περιεχόμενο της θέσης κύριας μνήμης address.
- LD (DE),A: αποθήκευσε το περιεχόμενο του συσσωρευτή στη θέση κύριας μνήμης που ορίζει το ζεύγος DE.

Τα ίδια ισχύουν και για τις εντολές φόρτωσης 16 bit, μόνο που σε αυτή την περίπτωση χρησιμοποιούνται τα εξής ζεύγη καταχωρητών: BC, DE, HL, IX, IY και SP.

#### Παραδείγματα

- LD DE,HL: φόρτωσε στο ζεύγος καταχωρητών DE το περιεχόμενο των καταχωρητών HL.

## 2.6.2 Εντολές ανταλλαγής, μαζικής μεταφοράς, και αναζήτησης

Οι εντολές ανταλλαγής επιτρέπουν την ανταλλαγή των τιμών δύο ζευγών καταχωρητών, αρκεί αυτοί να ανήκουν στο ίδιο σύνολο (δηλαδή στο σύνολο των κύριων ή εναλλακτικών καταχωρητών του Σχήματος 2-4)

### Παράδειγμα

EX HL,DE: Το περιεχόμενο του D ανταλλάσσεται με αυτό του καταχωρητή H. Ομοίως και το περιεχόμενο του E ανταλλάσσεται με το περιεχόμενο του καταχωρητή L.

Οι άλλες τρεις εντολές ανταλλαγών χρησιμοποιούν τον καταχωρητή SP ως δείκτη στοίβας. Η τιμή του SP δεν επηρεάζεται από την εκτέλεση των εντολών ανταλλαγής. Έτσι, ως παράδειγμα αναφέρονται οι εντολές EX (SP),HL, η EX (SP),IY καθώς και η EX (SP),IX οι οποίες ανταλλάσσουν το περιεχόμενο των ζευγών HL, IY και IX αντίστοιχα, με τις τιμές που βρίσκονται στην κορυφή της στοίβας.

Οι εντολές LDI, LDD, LDIR, LDDR είναι εντολές μαζικής μεταφοράς που χρησιμοποιούν τα ζεύγη DE, BC, HL. Οι εντολές αυτές πραγματοποιούν μαζικές μεταφορές πληροφοριών από τη μία περιοχή μνήμης στην άλλη. Το ζεύγος DE πρέπει να φορτωθεί με τη διεύθυνση της μνήμης όπου θα καταλήξουν τα δεδομένα, το ζεύγος BC πρέπει να φορτωθεί με το πλήθος των δεδομένων και το HL με τη διεύθυνση από την οποία θα ληφθούν τα δεδομένα. Η εκτέλεση της LDI περιλαμβάνει αρχικά τη μεταφορά ενός byte από τη διεύθυνση στην οποία δείχνει το HL στη διεύθυνση όπου δείχνει το DE. Στη συνέχεια, οι διευθύνσεις στα HL, DE αυξάνονται κατά 1 και η τιμή του BC μειώνεται κατά 1. Αν η τιμή του BC δεν έχει φτάσει στο 0 τότε ο δείκτης P/V του καταχωρητή κατάστασης F είναι 1.

Η LDD είναι όμοια με την LDI με τη διαφορά ότι οι τιμές των HL, DE μειώνονται αντί να αυξάνονται. Με άλλα λόγια, στην LDD η μεταφορά των δεδομένων ξεκινάει από το τέλος της περιοχής και φτάνει στην αρχή, ενώ στην LDI συμβαίνει το αντίστροφο. Οι εντολές LDIR, LDDR είναι όμοιες με τις LDI, LDD με τη διαφορά ότι η διαδικασία επαναλαμβάνεται μέχρις ότου η τιμή του BC μηδενιστεί. Ουσιαστικά οι LDIR, LDDR χρησιμοποιούνται για τη μαζική μεταφορά ενός μπλοκ δεδομένων ενώ οι LDI, LDD χρησιμοποιούνται σε έναν αντίστοιχο βρόχο που όμως δεν θέλουμε να τερματίσει (μόνο) στην περίπτωση που το BC μηδενιστεί. Για παράδειγμα, μπορεί να επιθυμούμε τον τερματισμό της μεταφοράς και όταν βρεθεί μια αναζητούμενη τιμή.

Οι εντολές αναζήτησης (searching) περιλαμβάνουν τις εντολές CPIR, CPDR οι οποίες ερευνούν κάποια περιοχή μνήμης για να δουν αν εμφανίζεται κάποια προκαθορισμένη τιμή. Χρησιμοποιούν το ζεύγος HL για την τρέχουσα διεύθυνση μέσα στην περιοχή που ερευνάται, τον BC για το πλήθος των θέσεων που θα εξεταστούν, και τον συσσωρευτή A για την αποθήκευση της τιμής που αναζητείται.

Η CPI διαβάζει το επόμενο byte από τη θέση στην οποία δείχνει ο HL, αυξάνει του HL κατά 1, και μειώνει τον BC κατά 1. Το byte που διαβάστηκε συγκρίνεται με την τιμή του A. Ανάλογα με το αν συμπίπτει η συγκρινόμενη τιμή και αν ο BC έχει φτάσει το 0, τίθενται ή καθαρίζονται οι κατάλληλες σημαίες του καταχωρητή F. Η CPD είναι όμοια με την CPI με τη διαφορά ότι το περιεχόμενο του HL μειώνεται. Οι εντολές CPIR, CPDR είναι όμοιες με τις CPI, CPD αντίστοιχα, με τη διαφορά ότι η διαδικασία επαναλαμβάνεται μέχρις ότου ο BC φτάσει στο 0 (εφόσον βέβαια δεν βρεθεί η αναζητούμενη τιμή, διαφορετικά τερματίζονται αυτόματα).

### 2.6.3 Αριθμητικές και λογικές εντολές

Υπάρχουν δύο είδη αριθμητικών και λογικών εντολών: εκείνες που δέχονται ορίσματα των 8 bit και εκείνες των 16 bit. Οι εντολές των 8 bit (πρόσθεση, αφαίρεση, λογικό ΚΑΙ, λογικό Ή, αποκλειστικό Ή και σύγκριση) χρησιμοποιούν ως ένα όρισμα την τιμή του συσσωρευτή A. Το δεύτερο όρισμα μπορεί να βρίσκεται είτε σε κάποιο άλλο καταχωλεσμά μιας τέτοιας πράξης αποθηκεύεται πάλι στον συσσωρευτή A.

Υπάρχουν δύο εντολές αριθμητικής πρόσθεσης: η ADD και η ADC. Η πρώτη έχει την μορφή  $ADD\ A,s$  και προσθέτει την τιμή του A στην τιμή του s ενώ το αποτέλεσμα αποθηκεύεται στον A. Το αποτέλεσμα της πρόσθεσης δύο αριθμών 8 bit είναι το πολύ 9 bit. Το 9ο ψηφίο αποθηκεύεται στον δείκτη κρατουμένου C του καταχωρητή κατάστασης F. Η δεύτερη εντολή  $ADC\ A,s$  λαμβάνει υπόψη της και την αρχική τιμή του κρατουμένου C, δηλαδή υπολογίζει την ποσότητα  $A+s+C$  και το αποτέλεσμα αποθηκεύεται στον A και στον C (το 9ο πιο σημαντικό ψηφίο).

Η αριθμητική αφαίρεση υλοποιείται με δύο εντολές, την SUB και την SBC. Όμοια με τις ADD και SBC αντίστοιχα οι παραπάνω εντολές αφαιρούν την τιμή του s από τον A και αποθηκεύουν το αποτέλεσμα στον A, ενώ αν προκύψει «δανεικό» αυτό αποθηκεύεται στο C του καταχωρητή F. Η SBC λαμβάνει υπόψη της πιθανό «δανεικό» από προηγούμενη αφαίρεση.

Σε μια πρόσθεση ή αφαίρεση προσημασμένων αριθμών είναι πιθανόν να προκύψει υπερχείλιση αν το αποτέλεσμα υπερβαίνει την τιμή +127 (7Fh) ή -127 (80h). Ενα τέτοιο γεγονός μπορεί να ανιχνευθεί με έλεγχο της τιμής P/V του καταχωρητή F.

Στο σύνολο των αριθμητικών εντολών ανήκουν και οι INC s/DEC s, οι οποίες αυξάνουν και μειώνουν κατά 1 αντίστοιχα την τιμή ενός καταχωρητή της KME, ενεργοποιώντας ορισμένους δείκτες του F.

Οι λογικές πράξεις των 8 bit έχουν παρόμοια δομή με εκείνες της αριθμητικής πρόσθεσης. Υπάρχουν 3 λογικές πράξεις: το λογικό ΚΑΙ (AND), το λογικό Ή (OR) και το Αποκλειστικό Ή (XOR). Κάθε λογική πράξη εφαρμόζεται στα αντίστοιχα bit των δύο ορισμάτων και κατά συνέπεια δεν υπάρχει η έννοια του κρατουμένου. Ως γνωστόν, με την AND

μπορούν να γίνουν. Ο κάποια μεμονωμένα bit μιας τιμής, με την OR να γίνουν 1, και με την XOR να αντιστραφεί η τιμή τους αφήνοντας ανέπαφες τις τιμές των υπολοίπων bit.

Οι αριθμητικές εντολές 16 bit εφαρμόζονται μεταξύ ζευγών καταχωρητών ή των καταχωρητών 16 bit της KME: IX, IY και SP. Σε αυτές τις εντολές ο HL είναι ο βασικός αθροιστής για τις αριθμητικές και λογικές πράξεις των 16 bit. Το περιεχόμενο των καταχωρητών BC, DE, HL και SP μπορεί να προστεθεί στο περιεχόμενο του HL ή να αφαιρεθεί από αυτό.

## 2.6.4 Εντολές περιστροφής και ολίσθησης

Οι εντολές της κατηγορίας αυτής είναι οι RLCA, RLA, RRCA, και RRA, οι οποίες περιστρέφουν (rotate) το περιεχόμενο του συσσωρευτή A. Το δεύτερο γράμμα κάθε εντολής αυτού του είδους αναφέρεται σε αριστερή (L) ή δεξιά (R) μετατόπιση. Η εντολή RLA μετατοπίζει το περιεχόμενο του A αριστερά τοποθετώντας το πιο σημαντικό ψηφίο της αρχικής τιμής στο κρατούμενο (C). Η προηγούμενη τιμή του κρατουμένου τοποθετείται στη λιγότερο σημαντική θέση του A μετά την ολίσθηση. Η RRA είναι παρόμοια, μόνο που η ολίσθηση γίνεται δεξιά. Οι RLCA και RRCA μετατοπίζουν το περιεχόμενο του A κατά μία θέση αριστερά ή δεξιά. Στην RLCA, το πιο σημαντικό ψηφίο της αρχικής τιμής αποθηκεύεται στο λιγότερο σημαντικό καθώς και στο κρατούμενο. Στην RRCA, το λιγότερο σημαντικό ψηφίο της αρχικής τιμής αποθηκεύεται στο περισσότερο σημαντικό ψηφίο καθώς και στο κρατούμενο μετά την ολίσθηση.

Οι RLD και RRD επιδρούν στο περιεχόμενο μιας θέσης μνήμης της οποίας η διεύθυνση βρίσκεται στον HL. Η τιμή στη διεύθυνση αυτή μετακινείται κατά 4 θέσεις αριστερά ή δεξιά αντίστοιχα ενώ στην κυκλική περιστροφή συμμετέχουν τα 4 λιγότερο σημαντικά ψηφία του A. Χρησιμοποιούνται για αριθμητικές πράξεις BCD.

Οι εντολές που αρχίζουν από S είναι εντολές ολίσθησης (shift) και όχι περιστροφής. Αυτό σημαίνει ότι το bit που «περισσεύει» μετά την ολίσθηση δεν ανακυκλώνεται δηλαδή δεν αποθηκεύεται στην περισσότερο ή λιγότερο σημαντική θέση του αποτελέσματος, όπως συνέβη με τις RRA, RLA. Στις εντολές αριθμητικής ολίσθησης (SRA, SLA) διατηρείται το πρόσημο. Στην ολίσθηση προς τα αριστερά αυτό δεν δημιουργεί καμία διαφορά αφού στο λιγότερο σημαντικό ψηφίο απλώς εισέρχονται μηδενικά. Όμως, στη δεξιά αριθμητική ολίσθηση (SRA), η τιμή του πιο σημαντικού ψηφίου είναι ίδια με εκείνη που υπήρχε στην ίδια θέση πριν τη μετατόπιση. Η λογική δεξιά ολίσθηση (SRL), αντίθετα, εισάγει μηδενικά στην πιο σημαντική θέση ανεξάρτητα από το πρόσημο.

Οι RL, RR, RLC, RRC είναι όμοιες με τις RLA, RRA, RLCA, RRCA με τη διαφορά ότι εφαρμόζονται σε θέσεις μνήμης που μπορεί να υποδεικνύονται π.χ., από την τιμή του HL. Με τις εντολές της κατηγορίας αυτής μπορούν να πραγματοποιηθούν πράξεις πολλαπλασιασμού ή διαίρεσης με το 2 ή να ελεγχθούν οι τιμές των bit κάποιων καταχωρητών ή της μνήμης.

### 2.6.5 Εντολές χειρισμού bit

Οι εντολές της κατηγορίας αυτής δρουν σε συγκεκριμένα bit μνήμης ή καταχωρητών. Μπορούν να θέσουν ένα μεμονωμένο bit στην τιμή 1 ή 0: η εντολή SET b,reg θέτει το bit b του καταχωρητή reg στην τιμή 1, ενώ αντίστοιχα η RESET το κάνει 0. Η εντολή BITI reg κάνει τη σημαία μηδέν (Zero flag) ίση με την αντίστροφη τιμή του bit b του καταχωρητή reg. Οι εντολές αυτές μπορούν να εφαρμοστούν και με υπονοούμενη ή σχετική διεύθυνσιο δότηση. Υπάρχει επίσης δυνατότητα συνδυασμού φόρτωσης και κάλυψης μεμονωμένων πεδίων bit (load/masking).

### 2.6.6 Εντολές áλματος

Οι εντολές της κατηγορίας αυτής περιλαμβάνουν τις JUMP, CALL και RETURN.

Οι εντολές JUMP μεταφέρουν την εκτέλεση του προγράμματος από μια περιοχή σε μια άλλη αλλάζοντας το περιεχόμενο του μετρητή προγράμματος χωρίς να λαμβάνεται πρόνοια για τη διάσωση της διεύθυνσης της εντολής που προηγήθηκε της JUMP. Οι εντολές CALL χρησιμοποιούνται για áλμα σε υπορουτίνες óμοια με τις JUMP αλλά αποθηκεύουν και τη διεύθυνση της εντολής που ακολουθεί την CALL στη στοίβα ἐτσι ώστε να είναι δυνατή η επιστροφή από την υπορουτίνα στη θέση óπου αυτή κλήθηκε. Αυτό συμβαίνει όταν στην υπορουτίνα εκτελεστεί η εντολή RETURN η οποία επιστρέφει τον έλεγχο στην επόμενη εντολή της CALL προσδιορίζοντας τη διεύθυνσή της από την τιμή που είναι αποθηκευμένη στη στοίβα.

Υπάρχουν δύο κατηγορίες εντολών JUMP: υπό συνθήκη και χωρίς συνθήκη.

Για παράδειγμα, η JP NN αναγκάζει την KME να πραγματοποιήσει áλμα στη διεύθυνση NN, ενώ η JP CC,NN θα πραγματοποιήσει áλμα στη διεύθυνση NN αλλά μόνο όταν η συνθήκη CC είναι αληθής. Αντίστοιχα ισχύουν και για τις CALL εντολές.

Τέλος, οι RETI και RETN χρησιμοποιούνται για επιστροφή από διακοπή τύπου INT και NMI, αντίστοιχα.

### 2.6.7 Εντολές εισόδου και εξόδου

Ο Z80 διαθέτει ένα μεγάλο πλήθος εντολών εισόδου/εξόδου που επιτρέπουν τη μεταφορά δεδομένων μεταξύ θέσεων μνήμης περιφερειακών (θύρες E/E) και καταχωρητών της KME ή της κύριας μνήμης. Διατίθενται ακόμα και εντολές μεταφοράς μπλοκ για την ταχεία αντιγραφή πολλών δεδομένων από περιφερειακά κυκλώματα με μία εντολή.

Η εντολή IN A,port διαβάζει από τη θύρα (port) ένα byte το οποίο μεταφέρει στον καταχωρητή A, ενώ η OUT port,A γράφει στη θύρα τα περιεχόμενα του συσσωρευτή A. Η IN reg,(C) διαβάζει στον καταχωρητή (reg) την τιμή από τη θύρα στην οποία δείχνει η τιμή του καταχωρητή C. Η OUT (C),reg γράφει την τιμή του καταχωρητή (reg) στη θύρα όπου δείχνει ο καταχωρητής C.

Η εντολή INI έχει υπονοούμενα ορίσματα, και μεταφέρει ένα byte από τη θύρα όπου δείχνει ο C στη θέση μνήμης της οποίας τη διεύθυνση αποθηκεύει ο HL. Ο καταχωρητής B που χρησιμοποιείται ως μετρητής μειώνεται κατά 1 και η διεύθυνση του HL αυξάνεται κατά 1. Με την εντολή αυτή μπορεί να μεταφερθεί ένα πλήθος byte από διαδοχικές θέσεις μνήμης τις οποίες διατρέχει ο HL στη θύρα όπου δείχνει ο C. Μετά από κάθε εντολή INI θα πρέπει να ελέγχεται ο καταχωρητής κατάστασης F για τη συνθήκη B=0. Μια παραλλαγή της INI είναι η INIR, στην οποία η μεταφορά επαναλαμβάνεται μέχρις ότου ο B γίνει 0.

Οι εντολές IND και INDR είναι όμοιες με τις INI και INIR, αντίστοιχα, με τη διαφορά ότι ο HL μειώνεται μετά από κάθε μεταφορά byte.

Αντίστοιχη λειτουργία με αντίθετη κατεύθυνση μεταφοράς έχουν οι εντολές OUTI, OUTD, OUTIR, OUTDR. Η διαφορά τους έγκειται στο ότι το byte που μεταφέρεται λαμβάνεται από τη διεύθυνση στην οποία δείχνει ο HL και αποθηκεύεται στη θύρα όπου δείχνει ο C.

## 2.6.8 Εντολές γενικής χρήσης

Εδώ συγκαταλέγονται εντολές αριθμητικής και ελέγχου. Η εκτέλεση των εντολών αυτών απαιτεί έναν ή περισσότερους τελεστές. Οι CPL και NEG επιδρούν στον συσσωρευτή A. Η πρώτη αντιστρέφει το περιεχόμενό του, δηλαδή κάνει τους άσους μηδενικά και αντίστροφα, ενώ η δεύτερη δίνει την αρνητική τιμή του A (σε μορφή συμπληρώματος ως προς δύο). Η εντολή SCF δίνει 1 και η CCF αντιστρέφει τον δείκτη κρατουμένου (zero flag) αντίστοιχα.

Η εντολή NOP δεν κάνει τίποτα και εκτελείται όταν ο Z80 βρίσκεται σε κατάσταση HALT έτσι ώστε να ανανεώνεται η δυναμική μνήμη (refresh). Οι εντολές EI και DI επιτρέπουν ή απαγορεύουν, αντίστοιχα, τις διακοπές. Οι εντολές IM 0, IM 1, IM 2 καθορίζουν τον τρόπο εξυπηρέτησης των διακοπών (δείτε την Ενότητα 2.7). Τέλος, η DAA προσαρμόζει το αποτέλεσμα των εντολών ADD, ADL, INC, SBC, SUB, DEC, NEG από δυαδική σε αναπαράσταση BCD.

## Παράδειγμα 2.2

- Η εντολή LD (DE),C απαιτεί ένα byte από τη μνήμη προγράμματος αφού τα ορίσματα DE και C μπορούν να αποθηκευτούν μαζί με τον κωδικό της εντολής στο ίδιο byte. Χρειάζεται 2 κύκλους μηχανής: έναν για την προσκόμιση της εντολής από τη μνήμη προγράμματος και έναν κύκλο εγγραφής στη μνήμη για την αποθήκευση του byte δεδομένων στη διεύθυνση που είναι αποθηκευμένη στο ζεύγος DE.
- Η εντολή ADC A,3000h, όπως προαναφέρθηκε, προσθέτει τα περιεχόμενα της θέσης 3000h με την τιμή του συσσωρευτή και την τιμή του κρατουμένου, και το αποτέλεσμα αποθηκεύεται στον συσσωρευτή και το κρατούμενο. Χρειάζεται 3 byte από τη μνήμη προγράμματος εκ των οποίων το ένα για την αποθήκευση του κωδικού της

εντολής και τα άλλα δύο για την αποθήκευση της διεύθυνσης 3000h. Τέσσερις κύκλοι μηχανής είναι αρκετοί για την εκτέλεση της εντολής: ένας για την προσκόμιση του κωδικού, άλλοι δύο για την προσκόμιση της τιμής 3000h από τη μνήμη προγράμματος, και ένας για την ανάγνωση της τιμής που βρίσκεται στην θέση 3000h της μνήμης δεδομένων. Η εκτέλεση της πρόσθεσης γίνεται στη συνέχεια εσωτερικού και δεν χρειάζεται ιδιαίτερο κύκλο μηχανής.

- Η εντολή OUT (46h). Α αποθηκεύει την τιμή του καταχωρητή A στη θύρα E/E 46h. Χρειάζεται 2 byte από τη μνήμη προγράμματος, το ένα για τον κωδικό της εντολής και το δεύτερο για την αποθήκευση της τιμής 46h. Η εντολή εκτελείται σε 3 κύκλους μηχανής: δύο για την προσκόμιση του κωδικού και της διεύθυνσης 46h και έναν για την εγγραφή σε χώρο E/E.
- Η εντολή DEC IY μειώνει την τιμή του IY καταχωρητή δείκτη κατά 1, το αποτέλεσμα αποθηκεύεται σε ένα byte στη μνήμη προγράμματος, και εκτελείται σε έναν μόνο κύκλο μηχανής (προσκόμιση εντολής).

Στη συνέχεια θα χρησιμοποιήσουμε τις εντολές του Z80 για να δώσουμε μερικά απλά παραδείγματα προγραμματισμού σε συμβολική γλώσσα.

### Παράδειγμα 2.3

Η απλή πρόσθεση δύο αριθμών 16 bit θα πρέπει να πραγματοποιηθεί με τη χρήση της εντολής πρόσθεσης δύο φορές, μια για το λιγότερο και μια για το περισσότερο σημαντικό τμήμα των ορισμάτων. Στην πρόσθεση των περισσότερο σημαντικών byte θα πρέπει να ληφθεί υπόψη και πιθανό κρατούμενο από την πρόσθεση των λιγότερο σημαντικών byte. Ας υποθέσουμε ότι το πρώτο όρισμα είναι αποθηκευμένο στις θέσεις 3000h-3001h, το δεύτερο στις θέσεις 3002h-3003h, και ότι το αποτέλεσμα θα αποθηκευτεί στις θέσεις 3004h-3005h. Αν ακολουθείται αρχιτεκτονική Little Endian τότε το λιγότερο σημαντικό τμήμα κάθε ορίσματος και του αποτελέσματος αποθηκεύεται στη χαμηλότερη διεύθυνση. Ο κώδικας που εκτελεί την παραπάνω διαδικασία μπορεί να είναι ο ακόλουθος (μετά το ερωτηματικό ακολουθούν σχόλια μέχρι το τέλος της γραμμής):

```

LD A, (3000h); A←το λιγότερο σημαντικό τμήμα του 1ου ορίσματος
LD B, (3002h); B←το λιγότερο σημαντικό τμήμα του 2ου ορίσματος
ADD A,B      ; A←A+B
LD (3004h),A; Αποθήκευση του λιγότερο σημαντικού τμήματος
                ; του αποτελέσματος στη 3006h
LD A, (3001h); A←το περισσότερο σημαντικό τμήμα του 1ου ορίσματος
LD B, (3003h); B←το περισσότερο σημαντικό τμήμα του 2ου ορίσματος
ADC A,B      ; A←A+B+Κρατούμενο
LD (3005h),A; Αποθήκευση του πιο σημαντικού τμήματος
                ; του αποτελέσματος στο 3007h

```

Προσέξτε ότι κατά τη δεύτερη πρόσθεση λαμβάνεται υπόψη το κρατούμενο από την πρώτη αφού οι εντολές LD που παρεμβάλλονται δεν επηρεάζουν τον δείκτη κρατούμένου (δείτε τον πίνακα εντολών στην Ενότητα 2.8).

### Παράδειγμα 2.4

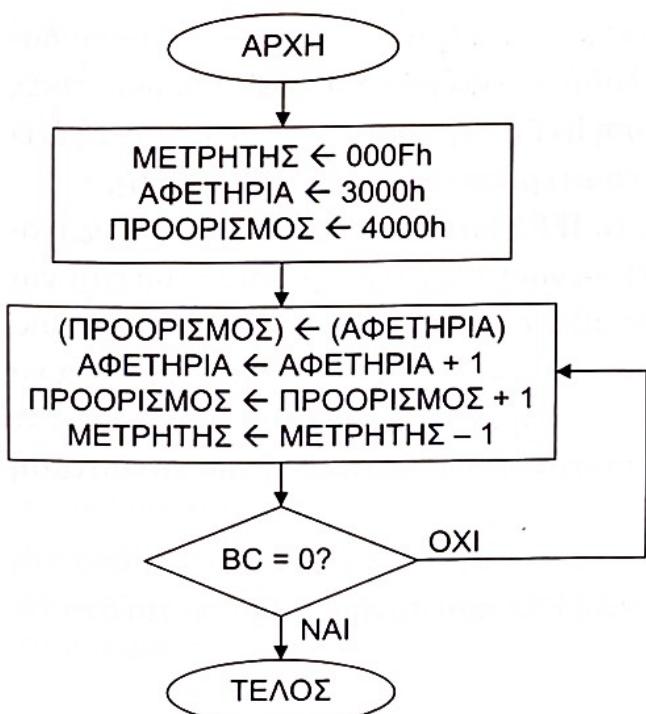
Ανταλλαγή τις τιμής των καταχωρητών E και C χωρίς τη χρήση εντολών LD, με τη βοήθεια της στοίβας και συγκεκριμένα με τις εντολές:

```
PUSH BC
PUSH DE
POP BC
POP DE
```

Προσέξτε ότι οι παραπάνω εντολές συντάσσονται με ζεύγη καταχωρητών και χρησιμοποιούν τη στοίβα, η οποία έχει δομή LIFO (Last In First Out). Έτσι η ανταλλαγή τιμών γίνεται πράγματι στους καταχωρητές E και C αλλά αναγκαστικά ανταλλάσσουν τιμές και οι B και D.

### Παράδειγμα 2.5

Στο παράδειγμα αυτό θα αντιγράψουμε τα περιεχόμενα των θέσεων 3000h-300Fh στις θέσεις 4000h-400Fh, αντίστοιχα. Όπως παρατηρούμε, το πλήθος των στοιχείων είναι 15. Για να περιγράψουμε τον κώδικα μπορούμε να χρησιμοποιήσουμε ένα διάγραμμα ροής όπως το παρακάτω:



Σχήμα 2-15. Διάγραμμα ροής που περιγράφει το πρόγραμμα αντιγραφής δεδομένων

LD BC, 000Fh ; το ζεύγος BC είναι ο μετρητής  
 LD HL, 3000h ; το ζεύγος HL περιέχει τη διεύθυνση αφετηρίας  
 LD DE, 4000h ; το ζεύγος DE περιέχει τη διεύθυνση προορισμού

LOOP:

```

LD A, (HL)      ; (ΠΡΟΟΡΙΣΜΟΣ) ← (ΑΦΕΤΗΡΙΑ)
LD (DE), A
INC HL          ; ΑΦΕΤΗΡΙΑ ← ΑΦΕΤΗΡΙΑ + 1
INC DE          ; ΠΡΟΟΡΙΣΜΟΣ ← ΠΡΟΟΡΙΣΜΟΣ + 1
DEC BC          ; ΜΕΤΡΗΤΗΣ ← ΜΕΤΡΗΤΗΣ - 1
JR NZ, Loop    ; ΑΝ ΜΕΤΡΗΤΗΣ ≠ 0 ΠΗΓΑΙΝΕ ΣΤΟ LOOP

```

Στον Z80, το παραπάνω πρόγραμμα μπορεί να υλοποιηθεί με πολύ λιγότερες εντολές με τη βοήθεια των εντολών μετακίνησης μπλοκ δεδομένων. Έτσι όλες οι εντολές κάτω από την ετικέτα LOOP μπορούν να αντικατασταθούν από την LDIR:

```

LD BC, 000Fh
LD HL, 3000h
LD DE, 4000h
LDIR

```

## 2.7 Διακοπές

### 2.7.1 Γενικά

Υπάρχουν δύο ακροδέκτες διακοπών στον Z80: η Non Maskable Interrupt (NMI~) και γραμμή INT~. Όταν συμβεί η πρώτη διακοπή (NMI~), αναγκάζει τον επεξεργαστή να πραγματοποιήσει άλμα στη διεύθυνση 0066h ενώ η δεύτερη μπορεί να διευθυνσιοδοτήσει μέχρι 127 διανυσματικές διακοπές, δηλαδή να οδηγήσει σε 127 διαφορετικές ρουτίνες εξυπηρέτησης της διακοπής. Η διακοπή INT αναγνωρίζεται μόνο αν αυτή έχει επιτραπεί με ενεργοποίηση του αντίστοιχου εσωτερικού δισταθούς (flip flop).

Στον Z80 υπάρχουν δύο τέτοια δισταθή: το IFF1 και το IFF2. Το πρώτο ενεργοποιεί ή απενεργοποιεί τις διακοπές τύπου INT~ ενώ το δεύτερο χρησιμοποιείται για προσωρινή αποθήκευση του πρώτου όταν συμβεί διακοπή NMI~. Όταν συμβεί μια διακοπή NMI~, το IFF1 γίνεται 0 και αδρανοποιούνται διακοπές τύπου INT~ για να δοθεί προτεραιότητα στην NMI~. Μετά την εκτέλεση της εντολής RETN της ρουτίνας την οποία είχε αποθηκεύσει το IFF2.

Η συνύπαρξη των INT~ και NMI~ επιτρέπεται αν εκτελεστεί η εντολή EI μέσα στη ρουτίνα εξυπηρέτησης της NMI~ μετά την εντολή EXX που συνήθως βρίσκεται στο τέλος της ρουτίνας εξυπηρέτησης.

### 2.2.2 Διακοπές τύπου INT~ (Normal External Interrupt)

Αν ο Z80 εκτελέσει την εντολή **IM 0**, τίθεται σε κατάσταση διακοπών 0. Στην κατάσταση αυτή το εξωτερικό κύκλωμα που προκάλεσε τη διακοπή θα πρέπει να παράσχει μια πλήρη εντολή διακλάδωσης στη ρουτίνα εξυπηρέτησης της διακοπής. Αυτό είναι μια πολύπλοκη διαδικασία από άποψη χρονισμών σημάτων ελέγχου και διαύλων και για τον λόγο αυτό δεν χρησιμοποιήθηκε ευρέως. Δίνει όμως συμβατότητα με τον τρόπο χειρισμού διακοπών από παλαιότερους επεξεργαστές, όπως ο 8080A.

Οι εντολές **IM 1** και **IM 2** θέτουν τον Z80 σε καταστάσεις διακοπών 1 και 2, αντίστοιχα. Στην κατάσταση τύπου 1, όταν συμβεί διακοπή (με INT~ χαμηλό) ο Z80 θα μεταβεί στη διεύθυνση 0038h, ενώ αν βρίσκεται στην κατάσταση 2 θα μεταβεί σε μια διεύθυνση η οποία καθορίζεται:

1. από το περιεχόμενο του καταχωρητή IV του Z80 (περισσότερο σημαντικό byte)
2. από τη διεύθυνση η οποία παρέχεται από τη συσκευή εισόδου/εξόδου που δημιούργησε τη διακοπή (λιγότερο σημαντικό byte).

Ο καταχωρητής IV αποθηκεύει τα 8 σημαντικότερα ψηφία της διεύθυνσης 16 bit ενός πίνακα διανυσμάτων διακοπής και η τιμή του ορίζεται συνήθως κατά την αρχικοποίηση του Z80. Τα 8 λιγότερο σημαντικά ψηφία παρέχονται από τη συσκευή που ζητά εξυπηρέτηση. Το ποια τιμή θα έχουν τα 8 αυτά ψηφία για μια συγκεκριμένη πηγή διακοπής καθορίζεται κατά την αρχικοποίηση κάθε περιφερειακής συσκευής. Μπορούν έτσι να οριστούν μέχρι 128 θέσεις στον πίνακα διανυσμάτων και κάθε θέση έχει εύρος 2 byte (το λιγότερο σημαντικό bit των 8 ψηφίων είναι πάντα 0). Τα δύο αυτά byte περιέχουν τη διεύθυνση της κάθε ρουτίνας εξυπηρέτησης.

Ο αλγόριθμος που ακολουθείται από την KME για διακοπή τύπου 2 προϋποθέτει ότι

- έχει εκτελεστεί η εντολή IM 2
- έχει ενεργοποιηθεί το IFF1
- ενεργοποιείται το INT'

Ο Z80 αναγνωρίζει τη διακοπή στον επόμενο κύκλο M1. Η συσκευή που δημιούργησε τη διακοπή στέλνει μέσω του διαύλου δεδομένων τα 8 λιγότερο σημαντικά ψηφία τα οποία συνδυάζονται με τα 8 περισσότερο σημαντικά από τον IV. Ο Z80 τοποθετεί το περιεχόμενο του μετρητή προγράμματος στη στοίβα. Η διεύθυνση που σχηματίστηκε προηγουμένως χρησιμοποιείται για να προσπελαστεί ο πίνακας διακοπών και να ληφθεί η διεύθυνση της ρουτίνας εξυπηρέτησης διακοπής στην οποία πραγματοποιεί τελικά άλμα ο Z80.

### 2.7.3 NMI~ (Non Maskable Interrupt)

Όταν προκύψει ένα σήμα NMI~, η διακοπή αναγνωρίζεται στο τέλος της εντολής που εκτελείται εκείνη τη στιγμή. Η KME ενεργεί ως εξής:

1. αποθηκεύει στη στοίβα τον μετρητή προγράμματος PC που περιέχει τη διεύθυνση της επόμενης προς εκτέλεση εντολής
2. εκτελεί άλμα στη διεύθυνση 0066h

Συνοψίζοντας μπορούμε να πούμε ότι με τις διακοπές σταματά προσωρινά η εκτέλεση του τρέχοντος προγράμματος από την KME και αρχίζει η εκτέλεση ενός άλλου τοποίου ονομάζεται **ρουτίνα εξυπηρέτησης διακοπής** (Interrupt Service Routine - ISR).

Μετά την ολοκλήρωση της ρουτίνας ISR πρέπει να επανέλθει ο έλεγχος στο πρόγραμμα που διακόπηκε και από το σημείο όπου σταμάτησε. Για να διεκπεραιωθεί σωστά αυτή η διαδικασία, θα πρέπει να διασωθεί το περιβάλλον του προγράμματος (δηλαδή οι χρησιμοποιούμενοι καταχωρητές) πριν την εκτέλεση της ISR.

Αυτό μπορεί να επιτευχθεί αν οι αρχικές εντολές της ISR αποθηκεύονται καταχωρητές όπως γίνεται με τις εντολές:

EX AF, AF'

EXX

οι οποίες ανταλλάσσουν τους καταχωρητές χωρίς τόνο με τους τονισμένους καταχωρητές. Για παράδειγμα, το περιεχόμενο του D θα αποθηκευθεί στον D' και αντίστροφα. Μέσα στην ISR, ο D μπορεί να χρησιμοποιηθεί για την αποθήκευση νέων τιμών και στο τέλος της ρουτίνας αυτής θα πρέπει να εκτελεστούν πάλι οι παραπάνω εντολές ώστε, όταν επιστραφεί ο έλεγχος στο πρόγραμμα που διακόπηκε, οι καταχωρητές να έχουν τις τιμές που είχαν όταν προκλήθηκε η διακοπή.

### 2.8 Σύνοψη συνόλου εντολών

Στη δεύτερη στήλη του πίνακα εντολών που παρατίθεται στο τέλος αυτής της παραγράφου χρησιμοποιούνται τα παρακάτω σύμβολα που αντιστοιχούν σε σημαίες του καταχωρητή κατάστασης F. Αν κάτω από κάθε ένα από τα ονόματα των σημαιών εμφανίζεται το σύμβολο '-' , '\*' , '0' , '1' ή '?' σημαίνει ότι με την εκτέλεση της συγκεκριμένης εντολής, η αντίστοιχη σημαία μένει ανεπηρέαστη, επηρεάζεται (μπορεί να γίνει '1' ή '0' ανάλογα με την περίπτωση), γίνεται οπωσδήποτε '0', γίνεται οπωσδήποτε '1' ή είναι άγνωστο σε ποια τιμή θα καταλήξει αντίστοιχα

Σύμβολο Σημαίας	Θέση στον F	Επεξήγηση
S	S	Σημαία προσήμου (Bit 7)
Z	Z	Σημαία μηδενικού (Bit 6)
HC	H	Σημαία κρατουμένου τεσσάρων bits(Bit 4)
P/V	P	Σημαία ισοτιμίας/υπερχείλισης (Bit 2, V=overflow)
N	N	Σημαία πρόσθεσης/αφαίρεσης (Bit 1)
CY	C	Σημαία κρατουμένου (Bit 0)

Τα σύμβολα που επεξηγούνται στον παρακάτω πίνακα χρησιμοποιούνται για την περιγραφή της σύνταξης και της λειτουργίας μιας εντολής στον πίνακα εντολών που ακολουθεί.

N	Άμεση διευθυνσιοδότηση - Immediate addressing
Nn	Επέκταση άμεσης διευθυνσιοδότησης - Immediate extended Σχετική διευθυνσιοδότηση-Relative addressing ( $PC=PC+2+offset$ )
E	Εκτεταμένη διευθυνσιοδότηση - Extended addressing
(nn)	Δεικτοδοτούμενη διευθυνσιοδότηση-Indexed addressing
(xx+d)	Διευθυνσιοδότηση καταχωρητή-Register addressing
R	Έμμεση μέσω καταχωρητή-Register indirect addressing
(rr)	Υπονοούμενη διευθυνσιοδότηση - Implied addressing
B	Διευθυνσιοδότηση bit - Bit addressing
P	Τροποποιημένη μηδενικής σελίδας - Modified page zero (RST)
A B C D E	Καταχωρητές (8-bit)
AF BC DE HL	Ζεύγη καταχωρητών (16-bit)
F	Καταχωρητής κατάστασης (8-bit)
I ÍV	Πιο σημαντικό byte διεύθυνσης εξυπηρέτησης διακοπών (8-bit)
IX IY	Καταχωρητές δείκτη (16-bit)
PC	Μετρητής προγράμματος (16-bit)
R	Καταχωρητής αναζωογόνησης μνήμης
SP	Δείκτης στοίβας (16-bit)
b	Ένα bit (θέση 0 έως 7)
cc	Συνθήκη (C, M, NC, NZ, P, PE, PO, Z)
d	Έκφραση ενός byte (-128 μέχρι +127)
dst	Προορισμός s, ss, (BC), (DE), (HL), (nn)
e	Επέκταση ενός byte (-126 μέχρι +129)
m	Οποιοσδήποτε καταχωρητής r, (HL) ή (xx+d)

n	Επέκταση ενός byte (0 μέχρι 255)
nn	Έκφραση δύο bytes (0 μέχρι 65535)
pp	Ζεύγος καταχωρητών BC, DE, IX ή SP
qq	Ζεύγος καταχωρητών AF, BC, DE ή HL
qq'	Εναλλακτικό ζεύγος των AF, BC, DE ή HL
r	Καταχωρητής A, B, C, D, E, H ή L
rr	Ζεύγος καταχωρητών BC, DE, IY ή SP
s	Οποιοσδήποτε καταχωρητής r, τιμή n, (HL) ή (xx+d)
src	Πηγή δεδομένων s, ss, (BC), (DE), (HL), nn, (nn)
ss	Ζεύγος καταχωρητών BC, DE, HL ή SP
xx	Καταχωρητής δείκτη IX ή IY
+ - * / ^	Πρόσθεση/Αφαίρεση/Πολλαπλασιασμός/Διαιρέση/Εκθέτης
& ~ v x	Λογικό KAI/OXI/Η/Αποκλειστικό Η
<- ->	Περιστροφή Αριστερά / Δεξιά
( )	Εμμεση Διευθυνσιοδότηση
( )+ -( )	Άμεση Διευθυνσιοδότηση με αυτόματη αύξηση ή μείωση κατά 1
{ }	Συνδυασμός Τελεστών
#	.. επίσης BC=BC-1, DE=DE-1
##	Χρησιμοποιούνται μόνο τα 4 λιγότερο σημαντικά bits του A

Ακολουθεί ο πίνακας εντολών του Z80 με τα σύμβολα που εξηγήθηκαν εν συντομίᾳ προηγουμένως:

Κωδικός Εντολής	SZHPNC	Περιγραφή	Λειτουργία
ADC A,s	***V0*	Πρόσθεση με κρατούμενο	A=A+s+CY
ADC HL,ss	**?V0*	Πρόσθεση με κρατούμενο	HL=HL+ss+CY
ADD A,s	***V0*	Πρόσθεση	A=A+s
ADD HL,ss	--?-0*	Πρόσθεση	HL=HL+ss
ADD IX,pp	--?-0*	Πρόσθεση	IX=IX+pp
ADD IY,rr	--?-0*	Πρόσθεση	IY=IY+rr
AND s	***P00	Λογικό KAI	A=A&s
BIT b,m	***P00	Έλεγχος bit	m&{2^b}
CALL cc,nn	-----	Κλήση υπό συνθήκη	If cc CALL
CALL nn	-----	Κλήση χωρίς συνθήκη	-(SP)=PC, PC=nn
CCF	--?-0*	Αντιστροφή Κρατουμένου	CY=~CY

Κωδικός Εντολής	SZHPNC	Περιγραφή	Λειτουργία
CP s	***V1*	Σύγκριση	A-s
CPD	****1-	Σύγκριση και μείωση	A- (HL), HL=HL-1, BC=BC-1
CPDR	****1-	Επαναλαμβανόμενη σύγκριση και μείωση	CPD till A=(HL) or BC=0
CPI	****1-	Σύγκριση και αύξηση	A- (HL), HL=HL+1, BC=BC-1
CPIR	****1-	Επαναλαμβανόμενη σύγκριση και αύξηση	CPI till A=(HL) or BC=0
CPL	--1-1-	Συμπλήρωμα ως προς 2	A=~A
DAA	***P-*	Προσαρμογή δεκαδικού (BCD)	A=BCD format
DEC s	***V1-	Μείωση κατά 1	s=s-1
DEC xx	-----	Μείωση κατά 1	xx=xx-1
DEC ss	-----	Μείωση κατά 1	ss=ss-1
DI	-----	Απαγόρευση διακοπών	
DJNZ e	-----	Μείωση κατά 1, άλμα αν ≠ 0	B=B-1 till B=0
EI	-----	Ενεργοποίηση διακοπών	
EX (SP), HL	-----	Ανταλλαγή	(SP)<->HL
EX (SP), xx	-----	Ανταλλαγή	(SP)<->xx
EX AF, AF'	-----	Ανταλλαγή	AF<->AF'
EX DE, HL	-----	Ανταλλαγή	DE<->HL
EXX	-----	Ανταλλαγή	qq<->qq' (except AF)
HALT	-----	Ακινητοποίηση	
IM n	-----	Ορισμός τρόπου χειρισμού διακοπών	(n=0,1,2)
IN A, (n)	-----	Είσοδος από περιφερειακή συσκευή	A=(n)
IN r, (C)	***P0-	Είσοδος από περιφερειακή συσκευή	r=(C)
INC r	***V0-	Αύξηση κατά 1	r=r+1
INC (HL)	***V0-	Αύξηση κατά 1	(HL)=(HL)+1
INC xx	-----	Αύξηση κατά 1	xx=xx+1
INC (xx+d)	***V0-	Αύξηση κατά 1	(xx+d)=(xx+d)+1
INC ss	-----	Αύξηση κατά 1	ss=ss+1
IND	?*?1-	Είσοδος και μείωση	(HL)=(C), HL=HL-1, B=B-1
INDR	?1??1-	Επαναλαμβανόμενη είσοδος και μείωση	IND till B=0
INI	?*?1-	Είσοδος και αύξηση	(HL)=(C), HL=HL+1, B=B-1
INIR	?1??1-	Επαναλαμβανόμενη είσοδος και Αύξηση	INI till B=0
JP (HL)	-----	Άλμα χωρίς συνθήκη	PC=(HL)
JP (xx)	-----	Άλμα χωρίς συνθήκη	PC=(xx)
JP nn	-----	Άλμα χωρίς συνθήκη	PC=nn

Κωδικός Εντολής	SZHPNC	Περιγραφή	Λειτουργία
JP cc,nn	-----	Άλμα με συνθήκη	If cc JP
JR e	-----	Σχετικό άλμα χωρίς συνθήκη	PC=PC+e
JR cc,e	-----	Σχετικό άλμα με συνθήκη	If cc JR(cc=C,NC,NZ,Z)
LD dst,src	-----	Φόρτωση	dst=src
LD A,i	**0*0-	Φόρτωση	A=i (i=I,R)
LDD	--0*0-	Φόρτωση και μείωση	(DE)=(HL), HL=HL-1, #
LDDR	--000-	Επαναλαμβανόμενη φόρτωση και μείωση	LDD till BC=0
LDI	--0*0-	Φόρτωση και αύξηση	(DE)=(HL), HL=HL+1, #
LDIR	--000-	Επαναλαμβανόμενη φόρτωση και αύξηση	LDI till BC=0
NEG	***V1*	Αντιστροφή πρόσημου	A=-A
NOP	-----	Καμία ενέργεια	
OR s	***P00	Λογικό Η	A=Avs
OTDR	?1??1-	Επαναλαμβανόμενη έξοδος και μείωση	OUTD till B=0
OTIR	?1??1-	Επαναλαμβανόμενη έξοδος και αύξηση	OUTI till B=0
OUT (C),r	-----	Έξοδος	(C)=r
OUT (n),A	-----	Έξοδος	(n)=A
OUTD	?*??1-	Έξοδος και μείωση	(C)=(HL), HL=HL-1, B=B-1
OUTI	?*??1-	Έξοδος και αύξηση	(C)=(HL), HL=HL+1, B=B-1
POP xx	-----	Ανάκτηση από στοίβα	xx=(SP) +
POP qq	-----	Ανάκτηση από στοίβα	qq=(SP) +
PUSH xx	-----	Αποθήκευση σε στοίβα	-(SP)=xx
PUSH qq	-----	Αποθήκευση σε στοίβα	-(SP)=qq
RES b,m	-----	Καθαρισμός bit	m=m&{~2^b}
RET	-----	Επιστροφή	PC=(SP) +
RET cc	-----	Επιστροφή υπό συνθήκη	If cc RET
RETI	-----	Επιστροφή από διακοπή INT	PC=(SP) +
RETN	-----	Επιστροφή από διακοπή MN1	PC=(SP) +
RL m	**0P0*	Περιστροφή αριστερά	m={CY,m}<-
RLA	--0-0*	Περιστροφή αριστερά μέσω A	A={CY,A}<-
RLC m	**0P0*	Περιστροφή αριστερά, κυκλικά	m=m<-
RLCA	--0-0*	Περιστροφή αριστερά, κυκλικά μέσω A	A=A<-

Κωδικός Εντολής	SZHPNC	Περιγραφή	Λειτουργία
RLD	**0P0-	Περιστροφή αριστερά, 4 bit	{A, (HL)}={A, (HL)}<- ##
RR m	**0P0*	Περιστροφή δεξιά	m=->{CY,m}
RRA	--0-0*	Περιστροφή δεξιά μέσω A	A=->{CY,A}
RRC m	**0P0*	Περιστροφή δεξιά, κυκλικά	m=->m
RRCA	--0-0*	Περιστροφή δεξιά, κυκλικά	A=->A
RRD	**0P0-	Περιστροφή δεξιά, 4 bit	{A, (HL)}=->{A, (HL)} ##
RST p	-----	Επανεκκίνηση από τη θέση p	(p=0H, 8H, 10H, ..., 38H)
SBC A,s	***V1*	Αφαίρεση με δανεικό	A=A-s-CY
SBC HL,ss	**?V1*	Αφαίρεση με δανεικό	HL=HL-ss-CY
SCF	--0-01	Θέσε στο κρατούμενο τιμή 1	CY=1
SET b,m	-----	Θέσε 1 στο bit b στη θέση m	m=mv{2^b}
SLA m	**0P0*	Αριθμητική αριστερή ολίσθηση	m=m*2
SRA m	**0P0*	Αριθμητική δεξιά ολίσθηση	m=m/2
SRL m	**0P0*	Λογική δεξιά ολίσθηση	m=->{0,m,CY}
SUB s	***V1*	Αφαίρεση	A=A-s
XOR s	***P00	Λογικό αποκλειστικό Ή	A=Axs

## Ερωτήσεις - Ασκήσεις

1. Είναι έγκυρη η εντολή LD A,D';
2. Αν γνωρίζετε ότι στο HL έχει αποθηκευτεί η διεύθυνση 30BFh, τι τιμή έχει ο H και ο L;
3. Ποια λειτουργία επιτελούν, ποια μέθοδο διευθυνσιοδότησης ακολουθούν, πόσα byte μνήμης προγράμματος και ποιους κύκλους μηχανής χρειάζονται οι παρακάτω εντολές;
  - α. LD HL,4f28h
  - β. SBC (HL)
  - γ. RLC (IX+4)
  - δ. OUT (C),B
  - ε. INDR
  - στ. CPIR
  - ζ. BIT 3,(HL)