

Δ. ΤΜΗΜΑ ΕΛΕΓΧΟΥ (Control), ΚΑΤΑΧΩΡΗΤΕΣ (Registers), ΔΙΑΓΡΑΦΟΙ (Buses), ΚΥΚΛΟΙ ΜΗΧΑΝΗΣ (Machine Cycles) και ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ (Machine Language)

1. Τμήμα Ελέγχου - Καταχωρητές - Διάλογοι - Κύκλοι Μηχανής

Το τμήμα ελέγχου αποτελεί τον «νέο» του Η.Υ. που κατευθύνει το όλο σύστημα για την ορθή εκτέλεση των εντολών που του δίδονται. Έτσι, στο τμήμα ελέγχου αποκωδικοποιούνται οι διάφορες εντολές (η μία μετά την άλλη διαδοχικά βάζει της σειράς που προβλέπεται) και δημιουργούνται κατάλληλα σήματα ελέγχου για τη μεταφορά των δεδομένων και τον απαιτούμενο συγχρονισμό στην εκτέλεση των εντολών κ.λ.π. Φυσικά οι εντολές είναι γραμμένες σε γλώσσα Μηχανής (την μόνη που καταλαβαίνει ο Η.Υ.).

Ρολοί της μηχανής.

Θεμελιώδη ρόλο στην όλη λειτουργία του τηλέωτος ελέγχου έχουν οι δύο καταχωρητές **CIR**(Current Instruction Register) και **PC/IC** (Program/Instruction Counter) του τηλέωτος, στους οποίους αποθηκεύονται η θέση εντολής και η θέση εντολής όπου έχει αποθηκευτεί η επόμενη προς εκτέλεση εντολή (βλέπε το κλασικό διάγραμμα ενός ψηφιακού υπολογιστή του Neumann, στο σχήμα 1). Βέβαια, οι δύο παραπάνω καταχωρητές μαζί με περιικούς άλλους (όπως π.χ. ο **MAR** και ο **Data/Buffer Register - DR**) και με τη βοήθεια των τριών διαύλων επικοινωνίας (Busses) που ενοπάρχουν σε κάθε Η.Υ. (Ο διαύλος μεταφοράς των δεδομένων - **Data Bus**, ο διαύλος μεταφοράς της διεύθυνσης μνήμης - **Address bus** και ο διαύλος μεταφοράς των σημάτων ελέγχου - **Control bus**), επιτυγχάνεται η αποτελεσματική επικοινωνία των ζώτων επιτυχών μηχανών των Η.Υ. και η βέλτιστη απόδοση της λειτουργίας του. Τέλος, οι βασικές λειτουργίες ενός Η.Υ. εκτελούνται σε καθορισμένα χρονικά διαστήματα που προσδιορίζονται από τους τακτικούς του ρολοί.

κτήρηση των βασικών λειτουργιών της εντολής.
 Ονομάζεται **κύκλος εκτέλεσης** (execution cycle), και την ε-
 δος της και ο τρόπος επεξεργασίας της. Ο δεύτερος κύκλος, που
 στην μονάδα ελέγχου, όπου εξετάζεται για να καθοριστεί το εί-
κλος ανάληψης (fetch cycle), ανακάλυψε από την μνήμη μια εντολή
μηχανής (Machine Cycles). Ο πρώτος κύκλος, που ονομάζεται **κ-
 δυ υποσύνολα βασικών λειτουργιών**, που ονομάζονται **κύκλοι
 λήψης εντολής** και **επεξεργασίας εντολής** ακολουθούν
 της πολλαπλότητας (πρέπει να θυμάται η μνήμη της πρώτης προς εκτέ-
 μνήμη, οπότε ο καταχωρητής IC (Απομνηστεύτης εντολών ή μίτρι-
 σης των εντολών του πολλαπλότητας και των δεδομένων του στην
 είναι σαφές ότι μετά την **προκαταρκτική επεξεργασία** - της τοποθέτη-
 Από το διάγραμμα της διαδοχής των λειτουργιών (βλέπε σχ. 16)

λειτουργιών μιας εντολής εξαρτάται απ' αυτή την ίδια την εντολή.
 Σε ένα σταθερό πάγιο, ενώ το πάγιο των βασικών λει-
 Έτσι, η χρονική διάρκεια ενός βασικού κύκλου μηχανής αντιστοιχεί

(αριθμούνται διαδοχικά), που είναι:

Εργασία του Διαβάνου για την ανάκτηση της εντολής
IO STG απαιτούνται, όπως και οι άλλοι, στο πακέτο σχήμα 17, απαιτούνται οι
 η στη θέση 1 της μνήμης του H.T., θα πρέπει ο καταχωρητής IC να
 φουσκά για την εκτέλεση της, που υποτίθεται ότι είναι αποθηκευμένη-

δεν είναι 0 (=00000000) έως την 1023 (=11111111).
 σε μνήμη με MAR των 10 bits (άρα με μνήμη 1024 θέσεων) από τη

SUB 32

λειτουργία της συσκευής (στη μνήμη) (Στη μνήμη) :
 ένα από αυτούς τους δύο κύκλους, για την περίπτωση της εκτέ-
 τομεία και τις βασικές λειτουργίες του Διαβάνου σε κάθε
 την σφαιρική της όλης διαδικασίας θα παραβέσονται πα-
 της εκτέλεσης (ανάλογα με την εντολή που εκτελείται). Τέλος, για
 της μνήμης. ο κύκλος της ανάκτησης της εντολής και ο κύκλος
 Εξάλλου στο σχήμα 16 είναι σαφές και ξεχωριστοί οι δύο κύκλοι

1°: Απόστολή της διεύθυνσης από τον PC στον MAR.

2°: Ο Decoder αποκωδικοποιεί τη θέση μνήμης 1.

3°: Μεταφορά του περιεχομένου της θέσης μνήμης 1 στον DR.

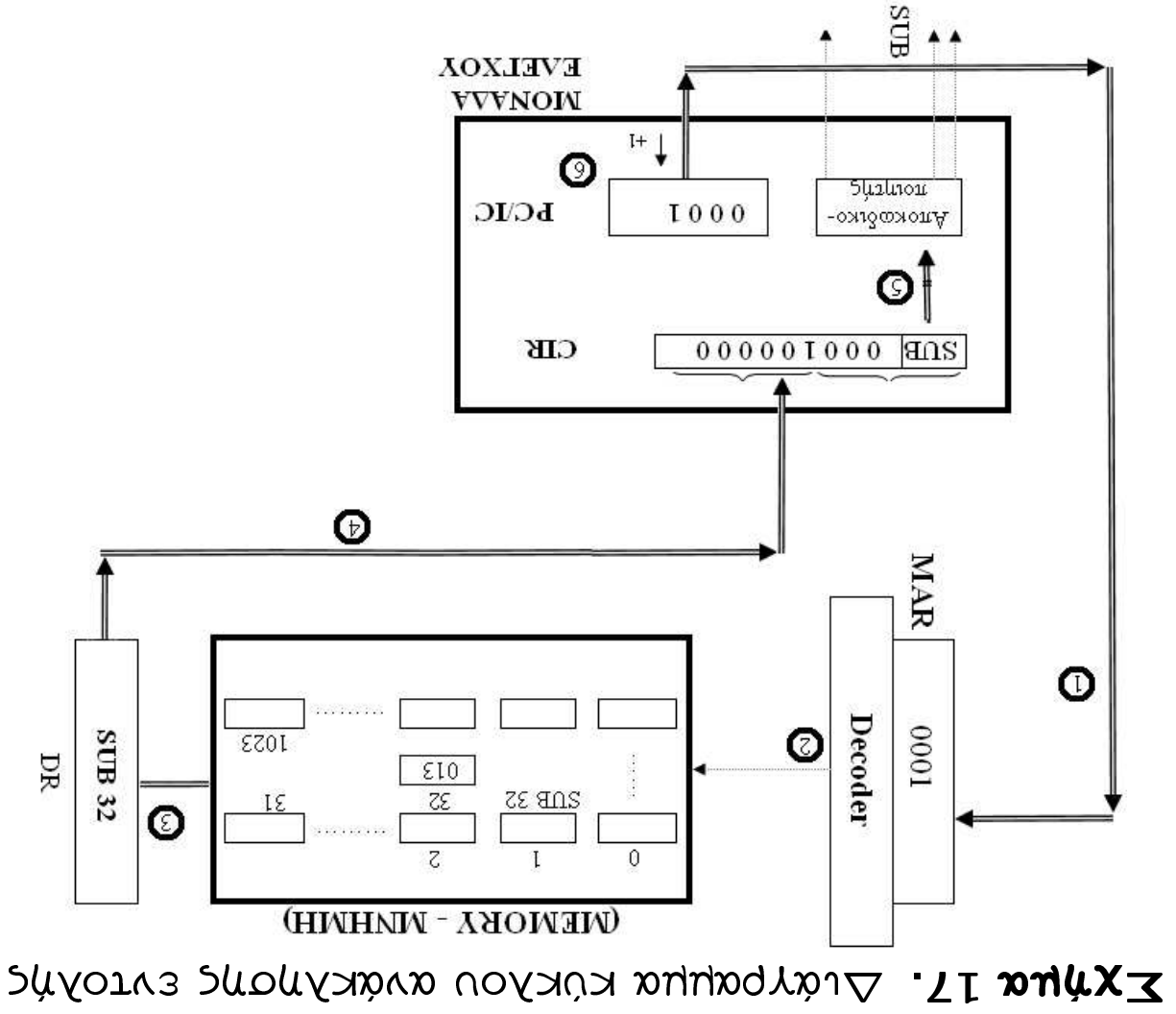
4°: Μεταβίβαση του περιεχομένου του DR στον CIR για αποκ-

ωδικοποίηση.

5°: Αποκωδικοποίηση της εντολής και ετοιμασία της διαδικασίας

για SUB.

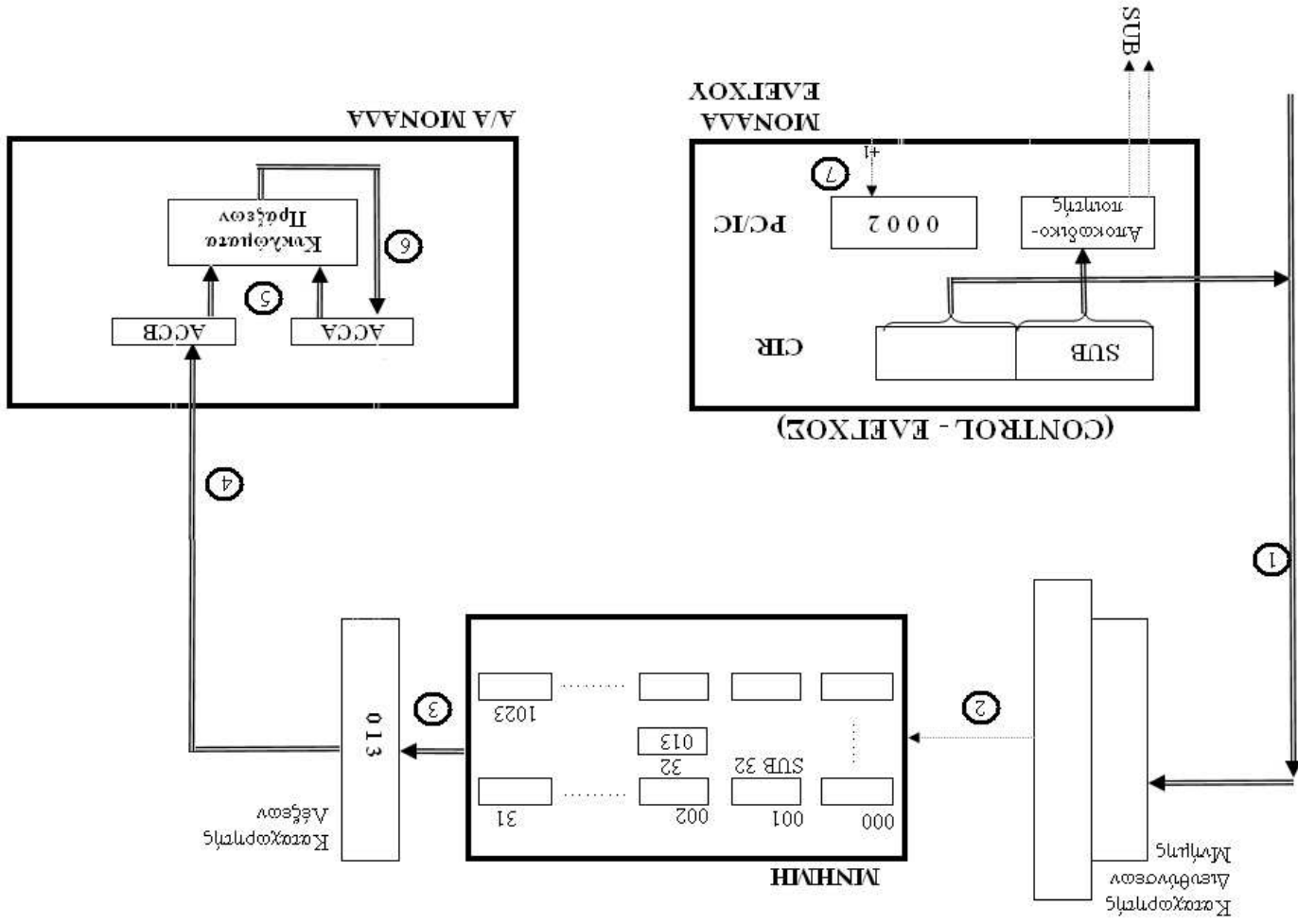
6°: Το περιεχόμενο του καταχωρητή PC αυξάνεται κατά 1.



Σχήμα 17. Διαγράμμα κύκλου ανακλήσης ερωτήσης

- Στη συνέχεια ξεκινά ο κύκλος της εκτέλεσης, οπότε λαμβάνουν χώρα οι παρακάτω λειτουργίες (εντολή με χρήση μνήμης), όπως απ-υθμούνται, στο σχήμα 18:
- 1°: Από τον CIR αποστέλλεται η διεύθυνση μνήμης που εμπλέκεται στον MAR.
- 2°: Ο Decoder ενεργοποιεί τη θέση μνήμης 32.
- 3°: Το περιεχόμενο της θέσης μνήμης 32 αποστέλλεται στον DR.
- 4°: Το περιεχόμενο του DR αποστέλλεται στην ALU για επεξεργασία.
- 5°: Εκτελείται η πράξη της αφαίρεσης.
- 6°: Η σχηματιζόμενη διαφορά αποθηκεύεται στον ACCA.

Σχήμα 18. Διαγράμμα κύκλου εκτέλεσης της εντολής



λειτουργία της εντολής.

μζου της πύλης ζ μεταφζρται στην ALU για να λάβει μζρος στην ταφζρται στον καταχωρητή δεδομένων (Data register), όπου δια-decoder προσδιορίζεται η θζση μνήμης που το περιεχόμενζ της με-μα (bb bbb bbb) μεταφζρται στον MAR, όπου με τη βοήθεια του θβζ στο αποκωδικοποιητή εντολών, ενζ συγχρόνως το δεύτρω τιμή-ζφφζ (rrr: bbbbbb) να πζρξουν και το μζν πζρωτο (rrr) να μεταφζρ-εντολή στην πύλη 4 να αφήσει τα δευτερευόντα του καταχωρητή εντολών στα ενεργειακά κζκλζ της ζκζζζζζζ της εντολής, δίδοντας του φλπ-φλοπ, που τίθεται στην κατάσταση P (Reset) με αποτζζε-παλμός του ρολογιου του Η.Υ. προκαλεί αλλαγή της κατάστασης Μόλις ο προηγούμενος κύκλος της ανάκλησης ολοκληρωθζ, ένας μνήμη στο καταχωρητή (CIR).

εντολή (της πζρωτης), με ενεργαποίηση των πύλων 1 και 2 από τη και συνεπώς γίνεται υλοποίηση της μεταφοράς της **πζρος ζκζζζζζζ** και φλπ-φλοπ (φ-φ.) ζίναι στην κατάσταση ζ (set) σχήμα 19, ένας σύντομος σχολιασμός του οποίου ακολουθεί.

Flipping την ενεργαποίηση των 2 κύκλων αποδίδεται από το παρκατάω-Τζλος, το σύνθετο διάγραμμα των λειτουργιών με τη χρήση Flip-

και ως παρακωλονησουμε την πορεία της εκτέλεσης των σχήμα

Θέση Μητρώου	Εντολή Μηχανής	Η εντολή συμβολικά
1	0001 0000 010000	(LDA A 16)
2	0010 0000 010011	(ADD A 19)
3	0011 0000 010000	(STA A 16)

θέση οι εντολές συμβολικά).

Ας υποθέσουμε το παρακάτω πρόγραμμα σε γλώσσα μηχανής, που είναι αποθηκευμένο στις αναρραφόμενες θέσεις μνήμης (σε παρ-
-εξαι 30) στην εντολή 30.

Παραδείγματα:

εντολής.

Τέλος, κατά τη διάρκεια της εκτέλεσης, η διεύθυνση του κατα-
γραφέα PC αυξάνεται κατά 1, έτσι ώστε όταν ο κύκλος της μεταφοράς
αρχίσει να υπάρχει στον PC η διεύθυνση της επόμενης για εκτέλεση

φυσικά, στην αρχή, στο βήμα 1, τίθεται η τιμή 1 στον PC (θέση
 μνήμης της πρώτης εκτέλεσης εντολής). Το φ.φ. τίθεται στην
 κατάσταση Σ, πράγμα που ενργοποιεί την μεταφορά της διεύθυν-
 σης 1 δια της πύλης 1 στον MAR και στον αποκωδικοποιητή (βήμα
 2) που υποκινεί τη μνήμη να θήσει το περιεχόμενό της θέση 1 στον
 καταχωρητή δεδομένων (βήμα 3) που μεταφέρει το περιεχόμενό του
 δια της πύλης 2 στον καταχωρητή εντολών (βήμα 4), που περιέχει
 τώρα την εντολή «0001 0000 010000». Στην συνέχεια το φ.φ. ε-
 νργοποιεί την κατάσταση Ρ για την εκτέλεση της εντολής, οπότε οι
 πύλες 3 και 4 αφήνουν τη διεύθυνση του «0001» στον αποκωδικοποι-
 ητή εντολών (βήμα 5) και του περιεχομένου «000010000» στον
 MAR, οπότε ενεργοποιείται ο αποκωδικοποιητής (βήμα 6) και υπο-
 κινεί την μνήμη να θήσει το περιεχόμενό της θέση 16 (≡0000010000)
 στον ALU και εκτελείται η εντολή με αποτέλεσμα ο υποσώρευτής Α
 να περιέχει το περιεχόμενό της θέσης μνήμης 16.

Μια τζελική παρατήρηση από την προηγούμενη διαδίκασια. **Όταν** γίνεται προσπέλαση της μνήμης με διεύθυνση που **ποτέ** **δεν** **χρησιμοποιείται** (κατά τον PC) (κατά τη διεύθυνση της επόμενης προς κτέ- ληση εντολής) **ή** **απλά** **απενεργοποιείται** (απενεργοποιείται σε εντολή της μηχανής και **οδενεί** **στον** **δομέων** (DR) αντιστοιχεί σε εντολή της μηχανής **που** **εξάγεται** **στον** **κατάχωρητη** **δε-** **κατάχωρητη** **εντολών** (CIR). **Ενώ** **όταν** **γίνεται** **προσπέλαση** **μνή-** **μης** **από** **τον** **κατάχωρητη** **των** **εντολών** (κατά την εκτέλεση) **τότε** **το** **παραχόμενο** **της** **λέξης** **μνήμης** **οδενεί** **στην** **ALU**, **γιατί** **είναι** **δε-** **δομένο** (data).

Το βασικό επίπεδο είναι οι διάφορες λογικές πύλες (ή μάλλον, η μία η NAND) και τα διάφορα κυκλώματα, με τα οποία επιτυγχάνονται η επεξεργασία και η αποθήκευση των πληροφοριών. Αυτό το επίπεδο είναι εκείνο που ονομάζουμε HARDWARE του συστήματος (H/W). **Μόνο του το H/W αδυνατεί** να κάνει οποιαδήποτε εργασία, όπως και κάθε άλλη πληροφοριακή μηχανή.

Το οποίοδηποτε υπολογιστικό σύστημα, που είναι αφιερωμένο για την εξυπηρέτηση ενός ειδικού έργου (όπως π.χ. οι κεραιές δε-σών για τις αεροπορικές εταιρείες, ή η παρακολούθηση ασθένων σε μονάδες εντατικής θεραπείας) είναι σχεδιασμένο ως μια ολόκληρη, που αποτελείται από διάφορα «στρώματα» (layers) - επίπεδα.

α. Εισαγωγή

2. Γνώσα Μηχανής

Η μηχανή αποτελείται από μία CPU με 2 16-bit συσσωρευτές (accumulators), τον A και B, ένα 9-bit μετρητή εντολών (Program/Instruction counter - PC/IC) και μία μνήμη με μέγεθος 512 λέξεων των 16 bits η κάθε μία.

β. Βασικά χαρακτηριστικά του Miracle

Βασικά χαρακτηριστικά.

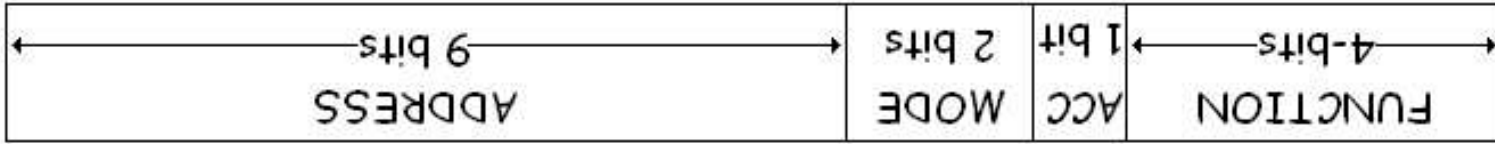
Εάν όμως πλαισιωθεί το H/W με το κατάλληλο σύνολο προ-γραμματών (συλλογής συνόλων κατάλληλων εντολών, του SOFTWARE - λογισμικού), τότε το σύστημα H/W - S/W, αξιοποιώντας την τεράστια ταχύτητα που διαθέτει, καθίσταται το τεχνολογικό θαύμα της εποχής μας. Σε αυτό λοιπόν το πλαίσιο και στην ειδική μορφή του προγραμματισμού σε γλώσσα μηχανής θα σκεφτούμε τώρα το ενδιαφέρον μας, γιατί είναι πολύ στένα συνδεμένη με το H/W του υπολογιστή. Έτσι, λοιπόν, θα υποθέσουμε ότι έχουμε έναν υποθετικό, εκταυθευτικό Η.Υ., τον «Miracle» με τα ακόλουθα

! Αιτιότητες με διεύθυνση (Κάθε μία επιλέγει κάποιον σωσ-

2. Οι εντολές της γλώσσας του Miracle

αντιπροσωπεύουν:

(μεταφορές δεδομένων). Πιο συγκεκριμένα έχουμε τις εντολές που
σε εντολές αλλαγής της διεύθυνσης της διεύθυνσης του ποδοκλάματος
χωρίς διεύθυνση, που επιλέκουν μόνο κάποιο σωσώπετη και
εντολές σε εντολές με διεύθυνση (address) μηήης, σε εντολές
Τέλος, οι 16 εντολές που μπορεί να εκτελέσει ο MIRACLE διακρί-



addresses):

Η CPU έχει ένα σύνολο 16 εντολών που εκτελεί (instruction reper-
tory), με την εξής δομή (4 εντολές είναι χωρίς διεύθυνση μηήης -

Εντολές Μεταφοράς Ελέγχου

CODE	Συμβολισμός	Λειτουργία (FUNCTION, τα 4 αρχικά bits)	ΠΑΡΑΔΕΙΓΜΑ
0000	LDA	Βάζε το περιεχόμενο της θέσης μνήμης που ακολούθει στον συσσωρευτή που αναγράφεται	(π.χ. LDA A 2)
0001	STA	Αποθηκεύσε το περιεχόμενο του συσσωρευτή που αναγράφεται στη θέση μνήμης που ακολούθει	(π.χ. STA B 48)
0010	ADD	Πρόσθεσε στον συσσωρευτή που αναγράφεται το περιεχόμενο της θέσης μνήμης που ακολούθει	(π.χ. ADD A 49)
0011	SUB	Αφάρσε από τον συσσωρευτή που αναγράφεται το περιεχόμενο της θέσης μνήμης που αναγράφεται	(π.χ. SUB B 25)
0100	ORA	Εκτέλεσε την πράξη OR μεταξύ του συσσωρευτή που αναγράφεται και της θέσης μνήμης που ακολούθει	(π.χ. ORA A 15)
0101	AND	Όπως προηγούμενα για την πράξη της AND	(π.χ. AND A 17)
0110	JMP	Πήγαινε και εκτέλεσε την εντολή που βρίσκεται στη θέση μνήμης που ακολούθει (Πλήρητα Χώρις Όρους)	(π.χ. JMP 13)
0111	BZE	Πήγαινε στην εντολή της θέσης μνήμης που ακολούθει όταν το περιεχόμενο του συσσωρευτή είναι μηδέν	(π.χ. BZE A 15)
1000	BNZ	Όπως προηγούμενα, αλλά με περιεχόμενο συσσωρευτή δεν μηδέν (Non Zero)	(π.χ. BNZ B 16)
1001	BMI	Όπως προηγούμενα, αλλά με περιεχόμενο συσσωρευτή αρνητικό (Minus)	(π.χ. BMI B 25)
1010	BPL	Όπως προηγούμενα, αλλά με περιεχόμενο συσσωρευτή θετικό (Plus)	(π.χ. BPL A 14)

!!. Αιτιότητες Χωρίς Σειρήνες

1011	LRS	Μετατόπιση το περιεχόμενο του συσσωρευτή που αναγράφεται μία θέση δεξιά. (π.χ. LRS A)
1100	NEG	Αντιστροφή του προσήμου του συσσωρευτή που αναγράφεται. (π.χ. NEG B)
1101	INA	Εισαγωγή ενός αριθμού από τη μονάδα εισόδου στον αναφερόμενο συσσωρευτή. (π.χ. INA A) Προσοχή, τα δεδομένα εισάγονται στον Η.Υ. δια μέσου συσσωρευτή και όχι άμεσα στη μνήμη του.
1110	OUT	Εκτύπωση του περιεχομένου του αναφερόμενου συσσωρευτή στον εκτυπωτή. π.χ. OUT B
1111	HLT	Τέλος του προγράμματος

!!!. Το 50 bit καθορίζει τον **Συσσωρευτή** που εμπλέκεται στη λειτουργία της εντολής. Εάν είναι **0** εμπλέκεται ο **A**, εάν είναι **1** εμπλέκεται ο **B**.

iv. Το **radio mode** (60 και 70 bit) καθορίζει τον τρόπο υλοποίησης της λειτουργίας. Έτσι, αν είναι:

00 Άμεση πρόσβαση
 LDA A # 1 (Immediate mode)
 αναφέρεται ο αριθμός είναι ο operand, δηλαδή ο αριθμός που άμεσα εισάγεται στον αναφέρεται υποσώρευτη.

01 Ευθεία πρόσβαση
 LDA A 35 (Direct mode)
 αναφέρεται ο αριθμός 35 είναι η διεύθυνση μνήμης που βρίσκεται ο operand.

10 Index by accumulator A
 STA B 16A
 αναφέρεται ο αριθμός 16A που είναι η διεύθυνση μνήμης όπου θα αποθηκευτεί το περιεχόμενο του B.

11 Index by accumulator B
 SUB A 25B
 αναφέρεται ο αριθμός 25B που είναι η διεύθυνση στο περιεχόμενο του B δίνει τη διεύθυνση μνήμης το περιεχόμενο της οποίας θα αφαιρεθεί από τον υποσώρευτη A.

V. Τέλος, τα τελεντοα 9 bits (Address) δίνουν τη δυαδική διεύθυνση, της ισοδύναμης με τον αριθμό που αναγράφεται και πρέπει να είναι μεταξύ του 0 και 511 (για τον Miracle).

BNZ A 15

δηλαδή, την

$1000 \equiv BNZ$, $0 \equiv ACCA$, $01 \equiv$ ευθεία πρόσβαση, $000001111 \equiv 15$,

εσωτερική μορφή: **1000001000001111** εντολής παριστά:

πως αποθηκεύεται στη μνήμη είναι: **0010 0 01 000110001**, ενώ η

Παραδείγματα: Η εσωτερική μορφή της εντολής **ADD A 49 0-**

Παράδειγμα Χρήσης του MIRACLE

1)

0	INA A	↓	Διάβασε τον πρώτο αριθμό
1	STA A 50	↓	Τοποθέτησε τον στη θέση 50
2	INA A	↓	Διάβασε τον δεύτερο αριθμό
3	STA A 51	↓	Τοποθέτησε τον στη θέση 51
4	LDA A 50	↓	Βρες το A - B
5	SUB A 51		
6	BZE A 13	↓	Έλεγε: if A=B πήγαινε στην 13
7	BMI A 10	↓	Έλεγε: if A<B πήγαινε στην 10
8	STA A 50	↓	Τοποθέτησε τον A-B στον A
9	JMP 4	↓	Πήδα στην 4, για επανάληψη
10	NEG A	↓	Τοποθέτησε το B-A στον Acc A
11	STA A 51	↓	Τοποθέτησε το B-A στον B
12	JMP 4	↓	Πήδα στην 4, για επανάληψη
13	LDA A 50	↓	Βάλε στον Acc A το περιεχόμενο της 50
14	OUT A	↓	Βγάλε έξω τον M.K.Δ.
15	HLT	↓	Σταμάτα

Miracle (στο ουαϊκό).

Άσκηση: Να γράφει το παραπάνω πρόγραμμα σε γλώσσα μηχανής

Σχόλια:
 Το πρόγραμμα βρίσκει τον M.K.Δ. των αριθμών A και B με τον αλγόριθμο του Ευκλείδη, και με τη βοήθεια των εζής του σχήματος: 20. Άλλως, αφαιρέσε από τον μεγαλύτερο τον μικρότερο και επανέλαβε το βήμα 1. 10. Εάν οι δύο αριθμοί είναι ίσοι, τότε ο M.K.Δ. είναι η κοινή τους τιμή.

20. Άλλως, αφαιρέσε από τον μεγαλύτερο τον μικρότερο και επανέλαβε το βήμα 1.

→ ο M.K.Δ.=4.
 20 8, 12 8, 4 8, 4 4
 B=8
 Π.Χ., εάν A=20 και

Άσκηση: Να γράφει το παρακάτω πρόγραμμα σε γλώσσα Miracle.

0	LDA A #0	↓	Μηδένισε τον Acc A
1	LDA B #50	↓	Τοποθέτησε στον Acc B το 50
2	STA A 19B	↓	Μηδένισε την 19 + B
3	SUB B #1	↓	Αφαίρεσε 1 από Acc B
4	BNZ B 2	↓	Εφες την επόμενη θέση μηδένισμου
5	HLT	↓	Σταμάτα

Σχόλιο: Το πρόγραμμα μηδενίζει τις θέσεις 19, 20, 21, ..., 69.

2)

3) Το παρακάτω πρόγραμμα διαβάζει ένα σύνολο N αριθμους (του ακολουθου), τους ταξινομει κατά τάξη μετεθους και τέλος τους εκτυπώνει. Ο N αποθηκευεται στη θέση 49, ενώ στην 48 u-παρχει ένα «flag». τέλος, οι αριθμοί αποθηκευονται από τη θέση 50 και πέρα.

0	INA A			
1	STA A 49			
2	BZE A 47			
3	INA B			
4	STA B 49A			
5	SUB A #1			
6	JMP 2			
7	LDA A #0			
8	STA A 48			
9	LDA A #2			
10	LDA B 49A			
11	SUB B 48A			
12	BPL B 20			
13	STA B 48			
14	LDA B 49A			
15	STA B 47			
16	LDA B 48A			
17	STA B 49A			
18	LDA B 47			
19	STA B 48A			
20	SUB A 49			
21	BZE A 25			
22	ADD A 49			
23	ADD A #1			
24	JMP 10			
25	LDA B 48			
26	BNZ A 7			
27	LDA A 49			
28	LDA B 49A			
29	OUT B			
30	SUB A #1			
31	BNZ A 28			
32	HLT			

↓ STOP

↓ OUTPUT RESULTS

↓ SEE IF ARRAY EXHAUSTED

Ασκήσεις:

1) Να γραφεί ένα πρόγραμμα Miracle, που να βρίσκει το άθροισμα:

$$1 + 2 + 3 + \dots + 50.$$

2) Να γραφεί ένα πρόγραμμα Miracle, που να βρίσκει το $7!$ (τόσο χωράει για θέση μνήμης του $i \dots 2^{16} - 1$).

3) Τι θα εκτυπώσει με το ακόλουθο πρόγραμμα:

```
0 LDA B #3
1 ADD B #5
2 SUB B #2
3 OUT B
4 HLT
```

4) Ομοίως για το πρόγραμμα:

0 LDA A 35

1 LDA B #17

2 OUT A

3 OUT B

4 HLT

17 + 63

35 + 39.

5) Ομοίως για το πρόγραμμα:

0 LDA A 5

1 STA A 2

2 LDA A #3

3 OUT A

4 HLT

5 + 2100

6) Ποιο πρέπει να είναι το input του προγράμματος που ακολουθεί για να εκτυπωθεί το 102;

0	INA	A
1	STA	A 50
2	ADD	A 50
3	OUT	A
4	HLT	

7) Να γραφεί ένα πρόγραμμα Miracle, που να διαβάσει 100 αριθμούς και να εκτυπώνει τον μεγαλύτερο τους.