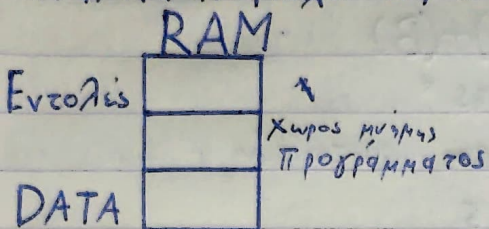


Αρχιτεκτονική Διάλεξη 6

Βασικό σύνολο καταχωρητών

1) MAR: Περιέχει τη διεύθυνση μνήμης των δεδομένων/εντολών



Εντολές και Δεδομένα έχουν διευθύνσεις

2) MDR: Μεταφέρει Δεδομένα ή Εντολές από τη μνήμη και στέλνει προς την μνήμη Δεδομένα

3) ACC: Γενικού σκοπού, συνήθως χρησιμοποιείται για πράξεις

4) Z: Υποδέχεται αποτελέσματα πράξεων από την ALU

5) PC - Program Counter - Μετρητής Προγράμματος: Κρατάει τη διεύθυνση της επόμενης εντολής που προκειται να εκτελεστεί

6) IR - Instruction Register - Καταχωρητής Εντολών: Περιέχει τον κωδικό εντολής (Operation Code - OPCODE)

OPCODE: Κωδικός που χαρακτηρίζει μια εντολή

Εστώ 8 εντολές (ADD, SUBTRACT, LOAD, STORE, INC, DIVIDE, MULTIPLY, NULL)

αρα το OPCODE = 3 $2^3 = 8$

ADD → 000	INC → 100
SUB → 001	DIV → 101
LD → 010	MUL → 110
STORE → 011	NULL → 111

ΜΟΡΦΕΣ ΕΝΤΟΛΩΝ: Κάθε εντολή έχει 1 OPCODE και ένα πλήθος διευθύνσεων. Το OPCODE δείχνει την εντολή, οι διευθύνσεις (παράγοντες) δείχνουν από ποιες διευθύνσεις μνήμης θα πάρουμε τα δεδομένα της εντολής

-A = A + B

-C = A + B

-A = A + ACC

DATA

OP	ΠΑΡΑΓΟΝΤΕΣ	Εντολές

OPCODE		0 Παράγοντες (Λειτουργίες Stack)	
OP	Ad	1 Παράγοντας $A \leftarrow A + ACC$ (Προσθέτω στην θέση A τα περιεχ. του ACC)	
OP	Ad1	Ad2	2 Παράγοντες $(A) \leftarrow (A) + (B)$
OP	Ad1	Ad2	Ad3 3 Παράγοντες $(C) \leftarrow (A) + (B)$

* Ένα σύστημα διαθέτει όλες τις μορφές *

■ Μήκος OPCODE: Εξαρτάται από το πλήθος εντολών

■ Μήκος AD(Address): Πλήθος λέξεων μνήμης

π.χ. Έστω ένα σύστημα με 16 εντολές, 2^{19} λέξεις, κάθε λέξη 2 bytes
(Έστω εντολές ενός ή παράγοντα)

1) OPCODE?

2) Address?

Αν δεν διευκρινίζεται έχω όλες τις μορφές από $4 \times 3 \times 19 = 61$

3) Ελάχιστο μήκος Address Bus, Data Bus?

2^{19} λέξεις, άρα θέλω 19bit Address Bus

και το μήκος του MAR είναι 19 bit

Ο MDR μεταφέρει εντολές κ' δεδομένα. Επειδή το μήκος μιας εντολής είναι OPCODE + Ad = 23 bit Data Bus

OP	Ad
4	19

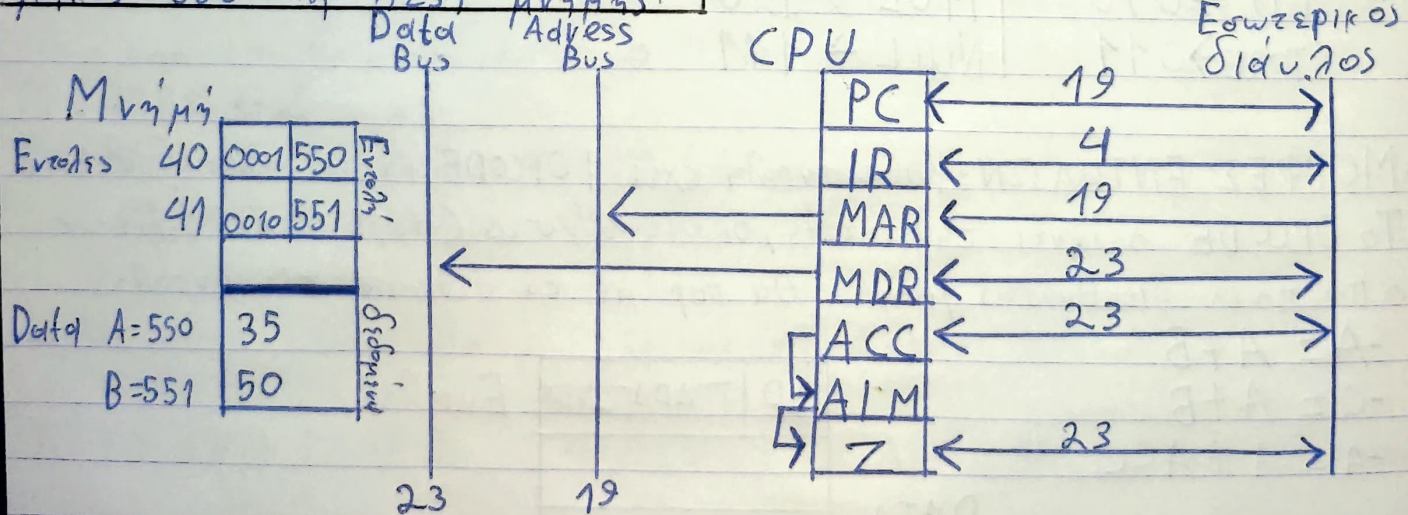
4 bit

19bit

MDR

$2^4 = 16$

Στο συγκεκριμένο παράδειγμα το μήκος του IR είναι 4 bits
όσο δηλαδή είναι το OPCODE. Ομώς πολλές φορές έχει μήκος όσο η λέξη μνήμης:



• ALM = Αριθμητική λογική μονάδα

Εσωτερικός διαύλος: χρησιμοποιείται για την επικοινωνία καταχωρητών της CPU σε κάθε χρονική στιγμή μόνο ένας μπορεί να γραφεί στον διαύλο

1) Ανάκληση: η διαδικασία με την οποία φέρνω την εντολή από την μνήμη. Μέχρις IR να διαβάσει το OPCODE η CPU δεν ξέρει ποια εντολή έρχεται. Για το λόγο αυτό η ανάκληση είναι ίδια για όλες τις εντολές!

Εκτέλεση: για κάθε εντολή είναι διαφορετική

• Πρόγραμμα

1) Φόρτωση της θέσης μνήμης $A=100$ στο ACC

2) Πρόσθεση της θέσης μνήμης $B=551$ με τον ACC

$A=550$ και αποθήκευσή στο B $B=A+B$

Inst 1: OPCODE 0001

Inst 2: OPCODE 0010

OPC Ad 1 Ad 2

41	1001	550	551
----	------	-----	-----

 → Η Ad. 2 παραχονταν με OPCODE 1001

■ Ανάκληση

Βήματα (ΜΙΚΡΟΛΕΙΤΟΥΡΓΙΕΣ ΣΕ ΚΥΚΛΟ ΡΟΛΟΓΙΟΥ) T_0, T_1, T_2 : Κύκλοι

1) Πρέπει η επόμενη εντολή να ζητηθεί και να διαβαστεί από την μνήμη $PC=40, MAR \leftarrow PC$

2) Η εντολή να μεταφερθεί (προηγείται κωδικοποίηση μνήμης) από τον MDR. $MDR \leftarrow M[MAR]$

3) Ο PC πρέπει να αυξηθεί για να περιέχει την επόμενη εντολή (στο παράδειγμα μας από 40 → 41) $PC \leftarrow PC+1$

4) IR πρέπει να διαβάσει το OPCODE: $IR \leftarrow MDR[OPCODE]$

$T_0: MAR \leftarrow PC, Z \leftarrow PC+1 \quad MAR \leftarrow 40$ [τον διαύλο τον καταλαμβάνει το PC]
 $T_1: MDR \leftarrow M[MAR], PC \leftarrow Z \quad MDR \leftarrow M[40] =$

0001	550
------	-----

 $T_2: IR \leftarrow MDR[OPCODE] \quad IR \leftarrow 0001$

Εσωτερικός διαύλος
Εξωτερικός διαύλος
Εσωτερικό MDR

Ο καταχωρητής που βρίσκεται στο δεξιό μέρος της σχέσης ← . Καταλαμβάνει τον διαύλο. Στο παλμό T_0 , ο διαύλος καταλαμβάνεται από τον PC
 T_0 : Ο Z διαβάζει την τιμή 40 από τον διαύλο την στέλνει στην ALU και την λαμβάνει αυξημένη

Παρατήρηση: Πως ο MAR και ο Z ξέρουν οτι το θημα το της Ανάληψης πρέπει να διαβάσουν απο το διαύλο;
 Πως ο PC ξέρει οτι πρέπει να γράψει στον διαύλο;
 : Ο MAR διάβασε απο τον PC την διεύθυνση της επόμενης εντολής, για να την ζητήσει απο τη μνήμη, ο MDR έλαβε την εντολή, ο PC αυξήθηκε κατά 1 μέσω του Z και ο IR πήρε το opcode

1) Η διεύθυνση A πρέπει να ζητηθεί απο την μνήμη για ανάγνωση (MAR θα την ζητήσει απο τον MDR)

2) Τα περιεχόμενα της διεύθυνσης 550 θα πάνε στον MDR

3) Ο MDR θα δώσει την τιμή στον ACC

T3: $MAR \leftarrow MDR[Address]$ $MAR \leftarrow 550$

T4: $MDR \leftarrow M[MAR]$ $MDR \leftarrow 35$

T5: $ACC \leftarrow MDR$ $ACC \leftarrow 35$

Ο PC πάντοτε είναι +1 απο αυτο που εκτελείται για αυτο γίνεται 41

1) Ζητείται η διεύθυνση B μέσω του MAR

2) Ο MDR μεταφέρει τα περιεχόμενα της διεύθυνσης 551

3) Τα περιεχόμενα του MDR προστίθενται με ειδικά του ACC και το αποτέλεσμα $\rightarrow Z$

4) Τα περιεχόμενα του Z $\rightarrow MDR$

5) Απο τον MDR $\rightarrow (B)$

T3: $MAR \leftarrow MDR[Address]$ $MAR \leftarrow 551$

T4: $MDR \leftarrow M[MAR]$ $MDR \leftarrow M[551] = 50$

T5: $Z \leftarrow ACC + MDR$ (Ο ACC τροφοδοτεί τον αθροιστή απευθείας ή παρούσα της ALU φαίνεται εδώ ο MDR μέσω διαύλου)

T6: $MDR \leftarrow Z$

T7: (Ζητείται η διεύθυνση B=551 για γράψιμο)

$M[MAR] \leftarrow MDR$ $M[551] \leftarrow 85$

SOS καθε επικοινωνία με την μνήμη βλέπει τον MAR

ΠΟΤΕ' ΔΕΝ ΓΡΑΦΩ $Z \leftarrow ALU$ ΕΠΕΙΔΗ ΔΕΝ ΕΙΝΑΙ ΚΑΤΑΧΩΡΗΤΗΣ