



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ  
ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ  
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ  
ΕΥΡΩΠΑΪΚΟ ΤΑΜΕΙΟ ΠΕΡΙΦΕΡΕΙΑΚΗΣ ΑΝΑΠΤΥΞΗΣ



**ΠΑΙΔΕΙΑ ΜΠΡΟΣΤΑ**  
2<sup>ο</sup> Επιχειρησιακό Πρόγραμμα  
Εκπαίδευσης και Αρχικής  
Επαγγελματικής Κατάρτισης

ΠΡΟΓΡΑΜΜΑ ΑΝΑΜΟΡΦΩΣΗΣ ΠΡΟΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΤΜΗΜΑΤΟΣ ΜΑΘΗΜΑΤΙΚΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

Μάθημα: Εισαγωγή στην Επιστήμη των Υπολογιστών

## 7η ΕΒΔΟΜΑΔΑ - ΔΕΥΤΕΡΑ 16 ΝΟΕΜΒΡΙΟΥ 2005

### Ε. ΜΙΚΡΟΕΝΤΟΛΕΣ (Microinstructions), και ΜΙΚΡΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ (Microprogramming)

#### α. Μικροεντολές - Μικροκώδικας - Ελέγχουσα μνήμη

Στα σύγχρονα υπολογιστικά συστήματα **η εκτέλεση κάθε εντολής** από το βασικό ρεπερτόριο του συστήματος, **πραγματοποιείται με κατάλληλη επικοινωνία ενός πλήθους καταχωρητών** (registers) του συστήματος, με την οποία μεταφέρονται πληροφορίες. Πιο συγκεκριμένα, η εκτέλεση μιας σημαντικής λειτουργίας ενός Η/Υ δεν μπορεί να υλοποιηθεί με μια μόνο μεταφορά του περιεχομένου ενός καταχωρητή, αλλά απαιτεί μια σειρά από μεταφορές που θα λάβουν χώρα με έναν αριθμό παλμών του ρολογιού (clock pulser).

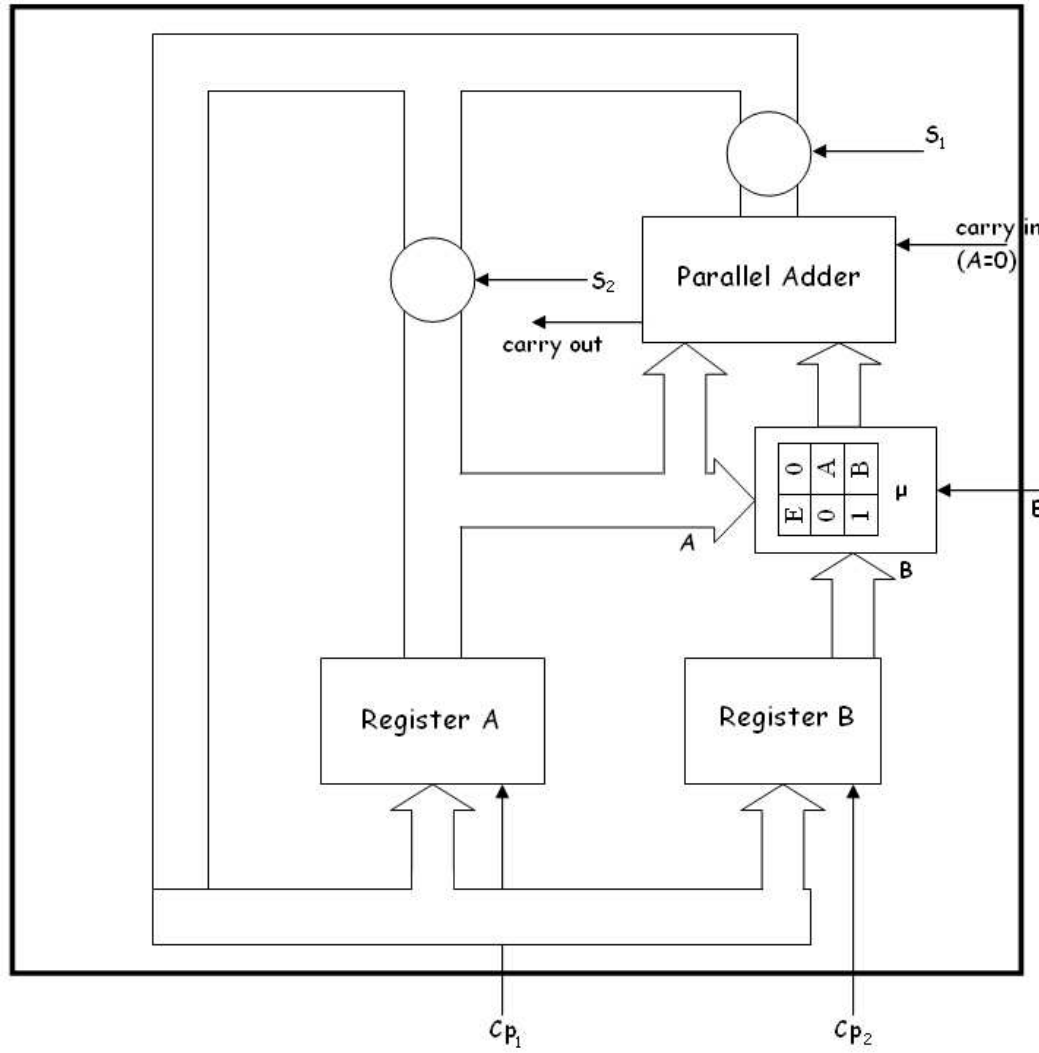
Ως παράδειγμα ας λάβουμε τη λειτουργία: «**τον δεκαπλασιασμό του περιεχομένου του καταχωρητή A**» και ας υποθέσουμε ότι η μηχανή μας διαθέτει την ALU του παρακάτω σχήματος 20, με την παρουσία και του καταχωρητή B. Η αλληλουχία των ενεργειών που μπορούν να δώσουν το ποθητό αποτέλεσμα μπορεί να είναι η ακόλουθη:

- α) Μεταφορά του περιεχομένου  $A \Rightarrow B$
- β) Πρόσθεση των περιεχομένων  $A + A \Rightarrow A(2 \cdot A)$
- γ) Πρόσθεση των περιεχομένων  $A + A \Rightarrow A(4 \cdot A)$
- δ) Πρόσθεση των περιεχομένων  $A + B \Rightarrow A(5 \cdot A)$
- ε) Πρόσθεση των περιεχομένων  $A + A \Rightarrow A(10 \cdot A)$ ,

που με τη βοήθεια των control lines  $C_{p1}$  και  $C_{p2}$  και της ελέγχουσας εισόδου E του multiplexer  $\mu$  υλοποιούνται με τη διαδοχή των παρακάτω σημάτων ελέγχου (που πραγματοποιούν τις απαιτούμενες μεταφορές):

Σχήμα 20.

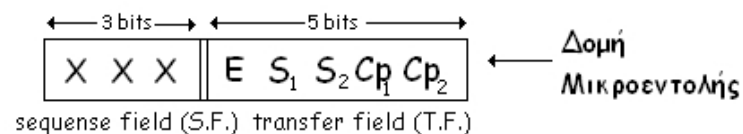
ALU



Μεταφορά	E	$S_1$	$S_2$	$C_{p_1}$	$C_{p_2}$	Αποτέλεσμα
$A \Rightarrow B$	-	0	1		$\square$	$B$ (= $A$ )
$A + B \Rightarrow A$	0	1	0	$\square$		$2 \cdot A$ (= $A$ )
$A + B \Rightarrow A$	0	1	0	$\square$		$4 \cdot A$ (= $A$ )
$A + B \Rightarrow A$	1	1	0	$\square$		$5 \cdot A$ (= $A$ )
$A + A \Rightarrow A$	0	1	0	$\square$		$10 \cdot A$ (= $A$ )

(Τα  $S_1$  και  $S_2$  αποτελούν καταλλήλους διακόπτες που επιτρέπουν δίοδο ή μη, ενώ το  $\square$  σημαίνει κατάσταση 1 του παλμού ελέγχου)

Για την εκτέλεση των παραπάνω «μεταφορών» θα πρέπει η μηχανή να εκδίδει τα κατάλληλα σήματα ελέγχου στην αριθμητική μονάδα ενώ συγχρόνως θα ακολουθεί την απαιτούμενη σειρά διαδοχής, έτσι ώστε όταν μια μεταφορά έχει ολοκληρωθεί, η κατάλληλη επόμενη να ακολουθεί. Αυτό επιτυγχάνεται με τις **μικροεντολές** (microinstructions) που αποτελούνται από **ένα σύνολο bits** που διαχωρίζονται σε **δύο τμήματα: το τμήμα διεύθυνσης της επόμενης εντολής (Sequence field) και το τμήμα υλοποίησης της μεταφοράς (Transfer field)**, όπως φαίνεται στο σχήμα που ακολουθεί:



Οι μικροεντολές αποθηκεύονται σε ειδική μνήμη υπερταχείας προσπελάσεως (ROM-Read Only Memory), που συνήθως ονομάζεται **Ελέγχουσα Μνήμη** (Control memory). **Σε κάθε κύκλο ο Η/Υ ανακαλεί μια μικροεντολή από την ελέγχουσα μνήμη και εκτελεί την μεταφορά που συνεπάγεται:** π.χ για την εκτέλεση των 5 λειτουργιών που απαιτεί ο δεκαπλασιασμός του περιεχομένου του Α με την ALU του σχήματος 20, θα αρκούσε μια ROM των 8 bytes με τα 3 πρώτα bits να δίδουν τη διεύθυνση της επόμενης εντολής (Sequence field), ενώ τα επόμενα 5 (Transfer field) θα προσδιορίζουν την απαιτούμενη διεργασία, που αποδίδεται από τον μικροκώδικα (microcode) του σχήματος 21, ενώ το σχήμα 22, δίδει την λεπτομερειακή περιγραφή του σχετικού Η/Υ.

Κοντολογής, μπορεί να λεχθεί ότι κάθε απλή μεταφορά από ένα καταχωρητή σε άλλον υλοποιείται με μια «**στοιχειώδη εντολή**», που είναι **η μικροεντολή**. Κατά συνέπεια η υλοποίηση μίας εντολής στον Η/Υ μπορεί να θεωρηθεί ότι πραγματοποιείται με την εκτέλεση ενός κατάλληλου συνδυασμού μικροεντολών του **μικροκώδικά** της.

Οι μικροκώδικες αποθηκεύονται σε μια **υπερταχεία μνήμη αναγνώσεως**, η δε πρακτική του μικροπρογραμματισμού απλοποιεί την δομή του τμήματος ελέγχου και διευρύνει το ρεπερτόριο των εντολών της γλώσσας μηχανής, έχει όμως το κόστος του μεγαλύτερου χρόνου εκτέλεσης των εντολών (πράγμα που δεν γίνεται αισθητό), ενώ παράλληλα ελαχιστοποιεί **το κόστος παραγωγής του H/W** (που έχει την καταπληκτική εξέλιξη να μειούται φοβερά το κόστος τους, με την παράλληλη **ουσιαστική βελτίωση** του προϊόντος - στην ουσία **το μόνο προϊόν που εμφανίζει αυτό το φαινόμενο**).

Τέλος, θα πρέπει να επισημανθεί **η τελείως διαφορετική φύση των μικροεντολών από τις εντολές σε γλώσσα μηχανής** παρά τις φαινομενικές ομοιότητες. Έτσι, οι μικροεντολές αποθηκεύονται στην **ελέγχουσα μνήμη** και όχι στην κυρία μνήμη του H/Υ, ενώ **ο μικροκώδικας γράφεται από τους μηχανικούς του συστήματος** και όχι από τους προγραμματιστές.

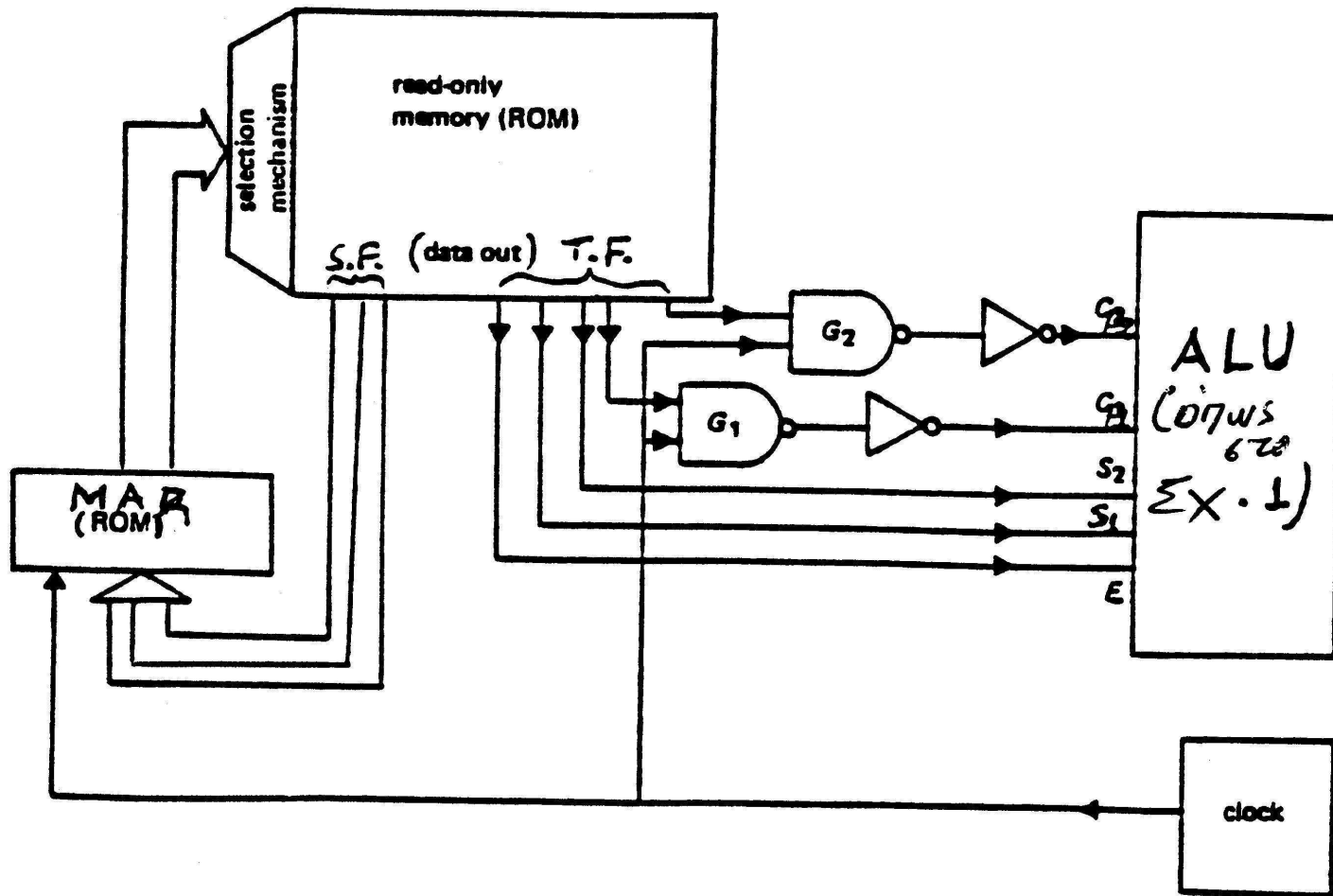
Σε κάθε κύκλο μηχανής εκτελείται μια μικροεντολή, ενώ για την εκτέλεση μιας εντολής σε γλώσσα μηχανής, απαιτούνται πολλοί κύκλοι (βλέπε κύκλους μηχανής στην προηγούμενη παράγραφο), παράλληλα **οι μικροεντολές έχουν διαφορετική δομή** από τις εντολές σε γλώσσα μηχανής και ο μικροκώδικας δεν εκτελείται με την διαδοχική δομή του προγράμματος σε γλώσσα μηχανής δηλαδή, δεν υπάρχει καταχωρητής PC που το περιεχόμενό του να αυξάνεται κατά 1 κάθε φορά στην περίπτωση της ελέγχουσας μνήμης, απ' όπου ανακαλώνται **οι μικροεντολές**, που δεν είναι ένα διαφορετικό είδος εντολής της γλώσσας μηχανής, αλλά **αποτελούν το μηχανισμό με τον οποίο υλοποιούνται οι εντολές της γλώσσας μηχανής.**



Σχήμα 21.

Διεύθυνση Μικροεντολής στην ROM	Μικροκώδικας «10 · A»		Λειτουργία	Αποτέλεσμα
	S.F. (Επόμενη Εντολή)	T.F. (Μεταφορά)		
000	001	00101	$A \Rightarrow B$	$B(= A)$
001	010	01010	$A + A \Rightarrow A$	$2A(\Rightarrow A)$
010	011	01010	$A + A \Rightarrow A$	$4A(\Rightarrow A)$
011	100	11010	$A + B \Rightarrow A$	$5A(\Rightarrow A)$
100	101	01010	$A + A \Rightarrow A$	$10A(\Rightarrow A)$
101	...	...		
110	...	...		
111	...	...		

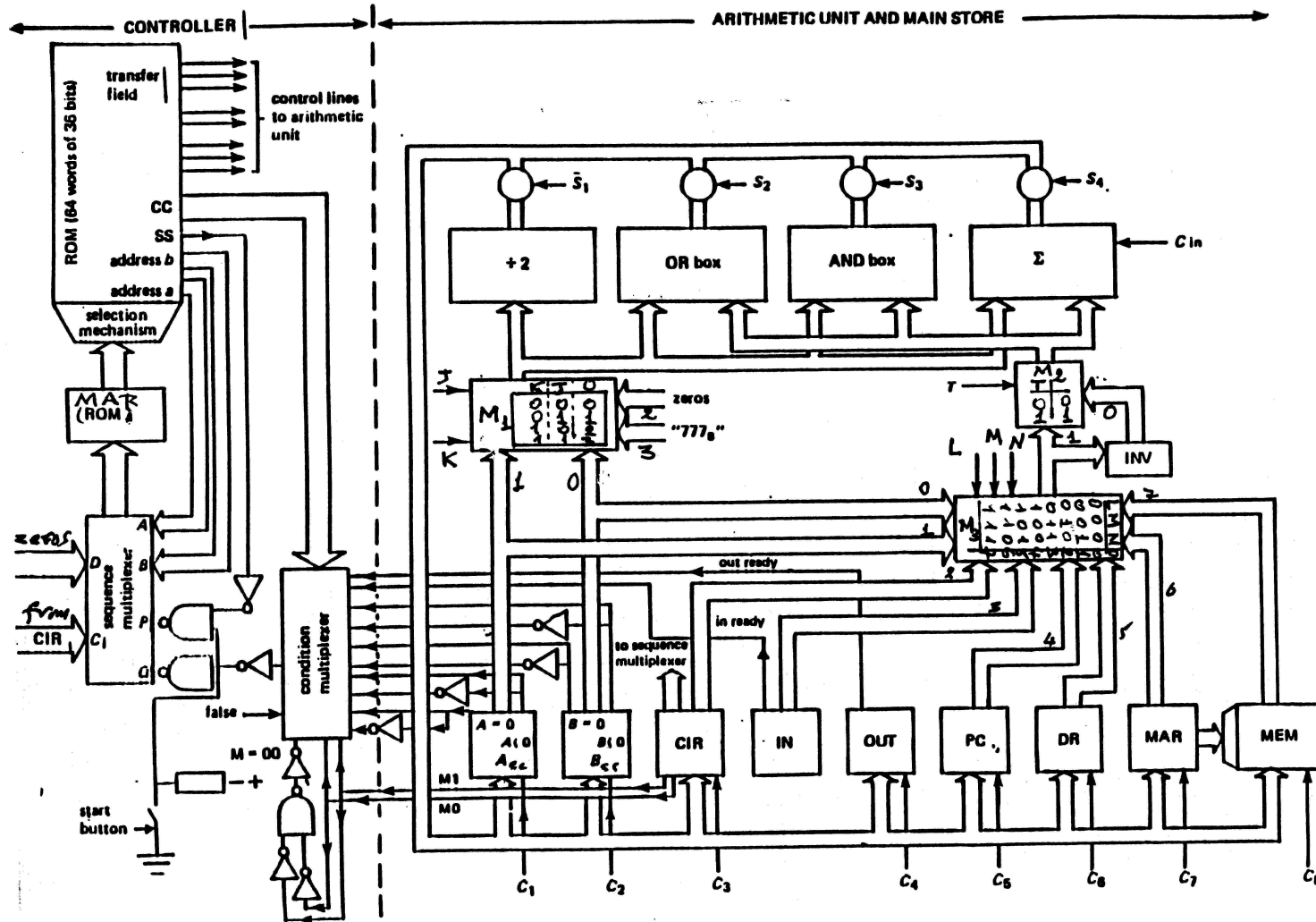
Σχήμα 22



### **γ. Διάγραμμα ενός εκπαιδευτικού Η/Υ**

Τέλος, το σχήμα 23 παρουσιάζει το λεπτομερειακό διάγραμμα ενός εκπαιδευτικού Η/Υ (Ε. Η. Υ.) με ελέγχουσα μνήμη των 64 λέξεων των 36 bits και λεπτομερειακή δομή των μικροεντολών που περιγράφονται στον πίνακα 33.

Σχήμα 23 Υπόδειγμα Ε.Η.Υ.



### Πίνακας 33. Δομή και Ενέργειες μικροεντολών

Διεύθυνση εντολής ROM	Address a	Address b	SS	Condition control	Ενέργεια	Σήματα						Ελέγχος				
						a	b	SS	CC	M1	M2	M3	C <sub>in</sub>	S	C	
0	1	0	0	False	PC and 0 ⇒ PC	000001	000000	0	0000	10	100	0	0	0010	00001000	
1	2	0	0	False	PC or 777 ⇒ PC	000010	000000	0	0000	11	100	0	0	0100	00001000	
2	3	0	0	False	PC ⇒ MAR	000011	000000	0	0000	10	100	0	0	0001	00000010	
3	a	4	0	False	MEM ⇒ CIR	000100	000000	0	0000	10	111	0	0	0001	00100000	
4	b	5	0	False	PC + 1 ⇒ PC	000101	000000	0	0000	10	100	0	1	0001	00001000	
5	c	6	10(i)	0	M = 00	000110	001010	0	1011	11	010	0	0	0010	00000100	
6	d	7	0	M1	CIR and 777 ⇒ DR	001001	000111	0	1010	10	101	0	0	0001	00000010	
7	e	9(h)	7	0	M1	DR ⇒ MAR	001001	001000	0	1001	00	101	0	0	0001	00000100
8	f	9(h)	8	0	M0	DR + A ⇒ DR	001001	000000	0	0000	01	101	0	0	0001	00000100
9	g	9	0	0	False	DR + B ⇒ DR	000000	000000	1	0000	10	111	0	0	0001	00000100
10	h	-	-	1	False	MEM ⇒ DR	000000	000000	1	0000	00	000	0	0	0000	00000000
11	i	-	-	1	False	NULL	000010	000000	0	0000	10	011	0	0	0001	10000000
12	-	2	0	0	False	IN ⇒ A	000010	000000	0	0000	10	011	0	0	0001	01000000
13	-	2	0	0	False	IN ⇒ B	000010	000000	0	0000	10	000	0	0	0001	00010000
14	-	2	2	0	False	A ⇒ OUT	000010	000000	0	0000	10	000	0	0	0001	00010000
15	-	2	2	0	False	B ⇒ OUT	000010	000000	0	0000	10	001	0	0	0001	00010000
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
32	LDAA	2	0	0	False	DR ⇒ A	000010	000000	0	0000	10	101	0	0	0001	10000000
33	LDAB	2	0	0	False	DR ⇒ B	000010	000000	0	0000	10	101	0	0	0001	01000000
34	STAA	2	0	0	False	A ⇒ MEM	000010	000000	0	0000	10	000	0	0	0001	00000001
35	STAB	2	0	0	False	B ⇒ MEM	000010	000000	0	0000	10	001	0	0	0001	00000001
36	ADDA	2	0	0	False	A + DR ⇒ A	000010	000000	0	0000	00	101	0	0	0001	10000000
37	ADDB	2	0	0	False	B + DR ⇒ B	000010	000000	0	0000	01	101	0	0	0001	01000000
38	SUBA	2	0	0	False	A - DR ⇒ A	000010	000000	0	0000	00	101	1	1	0001	10000000
39	SUBB	2	0	0	False	B - DR ⇒ B	000010	000000	0	0000	01	101	1	1	0001	01000000
40	ANDA	2	0	0	False	A and DR ⇒ A	000010	000000	0	0000	00	101	0	0	0010	10000000
41	ANDB	2	0	0	False	B and DR ⇒ B	000010	000000	0	0000	01	101	0	0	0010	01000000
42	ORAA	2	0	0	False	A or DR ⇒ A	000010	000000	0	0000	00	101	0	0	0100	10000000
43	ORAB	2	0	0	False	B or DR ⇒ B	000010	000000	0	0000	01	101	0	0	0100	01000000
44	JMPA	2	0	0	False	MAR ⇒ PC	000010	000000	0	0000	10	110	0	0	0001	00001000
45	JMPB	2	0	0	False	MAR ⇒ PC	000010	000000	0	0000	10	110	0	0	0001	00001000
46	BZEA	2	44	0	A = 0	NULL	000010	101100	0	0001	00	000	0	0	0000	00000000
47	BZEB	2	44	0	B = 0	NULL	000010	101100	0	0101	00	000	0	0	0000	00000000
48	BNZA	2	44	0	A ≠ 0	NULL	000010	101100	0	0000	00	000	0	0	0000	00000000
49	BNZB	2	44	0	B ≠ 0	NULL	000010	101100	0	0110	00	000	0	0	0000	00000000
50	BMIA	2	44	0	A < 0	NULL	000010	101100	0	0011	00	000	0	0	0000	00000000
51	BMIB	2	44	0	B < 0	NULL	000010	101100	0	0111	00	000	0	0	0000	00000000
52	BPLA	2	44	0	A ≥ 0	NULL	000010	101100	0	0100	00	000	0	0	0000	00000000
53	BPLB	2	44	0	B ≥ 0	NULL	000010	101100	0	1000	00	000	0	0	0000	00000000
54	LRSA	2	0	0	False	A ÷ 2 ⇒ A	000010	000000	0	0000	00	000	0	0	1000	10000000
55	LRSB	2	0	0	False	B ÷ 2 ⇒ B	000010	000000	0	0000	01	000	0	0	1000	01000000
56	NEGA	2	0	0	False	(-A) ⇒ A	000010	000000	0	0000	10	000	1	1	0001	10000000
57	NEGB	2	0	0	False	(-B) ⇒ B	000010	000000	0	0000	10	001	1	1	0001	01000000
58	INAA	58	11	0	IN ready	NULL	111010	001011	0	1100	00	000	0	0	0000	00000000
59	INAB	59	12	0	IN ready	NULL	111011	001100	0	1100	00	000	0	0	0000	00000000
60	OUTA	60	13	0	OUT ready	NULL	111100	001101	0	1101	00	000	0	0	0000	00000000
61	OUTB	61	14	0	OUT ready	NULL	111101	001110	0	1101	00	000	0	0	0000	00000000
62	END(A)	62	0	0	False	NULL	111110	000000	0	0000	00	000	0	0	0000	00000000
63	END(B)	63	0	0	False	NULL	111111	000000	0	0000	00	000	0	0	0000	00000000

ROM  
64 θέσεων

## Παραδείγματα.

1. Η διαδικασία της ανάκλησης μιας εντολής και η σύγχρονη αύξηση των PC κατά 1 υλοποιούνται από τις ακόλουθες 3 μικροεντολές (μεταφορές των καταχωρητών):

(2)  $PC \Rightarrow MAR$  (μεταφορά του PC στο MAR για την επιλογή της θέσης μνήμης, όπου είναι αποθηκευμένη εντολή).

(3)  $MEM \Rightarrow CIR$  (μεταφορά του περιεχομένου της επιλεγμένης θέσης μνήμης στον CIR).

(4)  $PC + 1 \Rightarrow PC$  (αύξηση του περιεχομένου του PC κατά 1).

Εντός παρενθέσεως αριστερά των εντολών είναι η θέση μνήμης στην control memory όπου είναι αποθηκευμένες οι αντίστοιχες εντολές, στον πίνακα 33.

2. Η διαδικασία της εκτέλεσης μιας εντολής, δέν είναι ενιαία (βλέπε αντίστοιχη παράγραφο) αλλά εξαρτάται από το είδος της εντολής. Έτσι, λοιπόν, έχουμε:

i. **Τις εντολές που απαιτούν χρήση (διεύθυνση) μνήμης**, όπως η:

LDA, ADD, SUB, AND και ORA.

ii. **Τις εντολές που χρειάζονται κάποια διεύθυνση** όπως η: STA και τα διάφορα JUMPS.

iii. **Τις εντολές που είναι ανεξάρτητες από τα παραπάνω**, όπως η: LRS, NEG, INA, OUT, ALT, οπότε ανάλογα με το MODE της εντολής, θα έχουμε για τα διάφορα είδη, τις μικρο-εντολές:

- **CIR and  $777_8 \Rightarrow DR$**  (Επιλογή του πεδίου διεύθυνσης της εντολής και μεταφορά του στον καταχωρητή δεδομένων (data register)).

**(Άμεση μέθοδος)**

- **CIR and  $777_8 \Rightarrow MAR$**  (Αποστολή της διεύθυνσης στον (MAR)).

- **MEM  $\Rightarrow DR$**  (Μεταφορά των δεδομένων από τη μνήμη στον καταχωρητή δεδομένων (Ευθεία μέθοδος)).

Π.χ. LDA A 32, οπότε οι προηγούμενες 2 μικροεντολές ακολουθούνται από από την DR  $\Rightarrow$  A.

- **CIR and 777<sub>8</sub>  $\Rightarrow$  DR** (Αποστολή της διεύθυνσης στον (MAR).

- **DR + A  $\Rightarrow$  MAR** (Αύξηση της κατά το περιεχόμενο του συσσωρευτή A.)

- **MEM  $\Rightarrow$  DR** (Μεταφορά του περιεχομένου της θέσης μνήμης που προκύπτει στον καταχωρητή δεδεμένων).

(Δικτυωμένη με Acc A)

Π.χ. LDA B 15A, οπότε οι προηγούμενες 3 εντολές ακολουθούνται από από την DR  $\Rightarrow$  B.

- **CIR and 777<sub>8</sub>  $\Rightarrow$  DR** Όπως προηγούμενα με χρήση του Acc B.

- **DR + B  $\Rightarrow$  MAR**

- **MEM  $\Rightarrow$  DR**



(Δικτύωση με Acc B)

Π.χ. LDA A 16B, οπότε οι προηγούμενες 3 εντολές ακολουθούνται από από την DR  $\Rightarrow$  A.

### 3. Ανάλυση ροής ενός προγράμματος σε γλώσσα μηχανής, στις εκάστοτε μικροεντολές

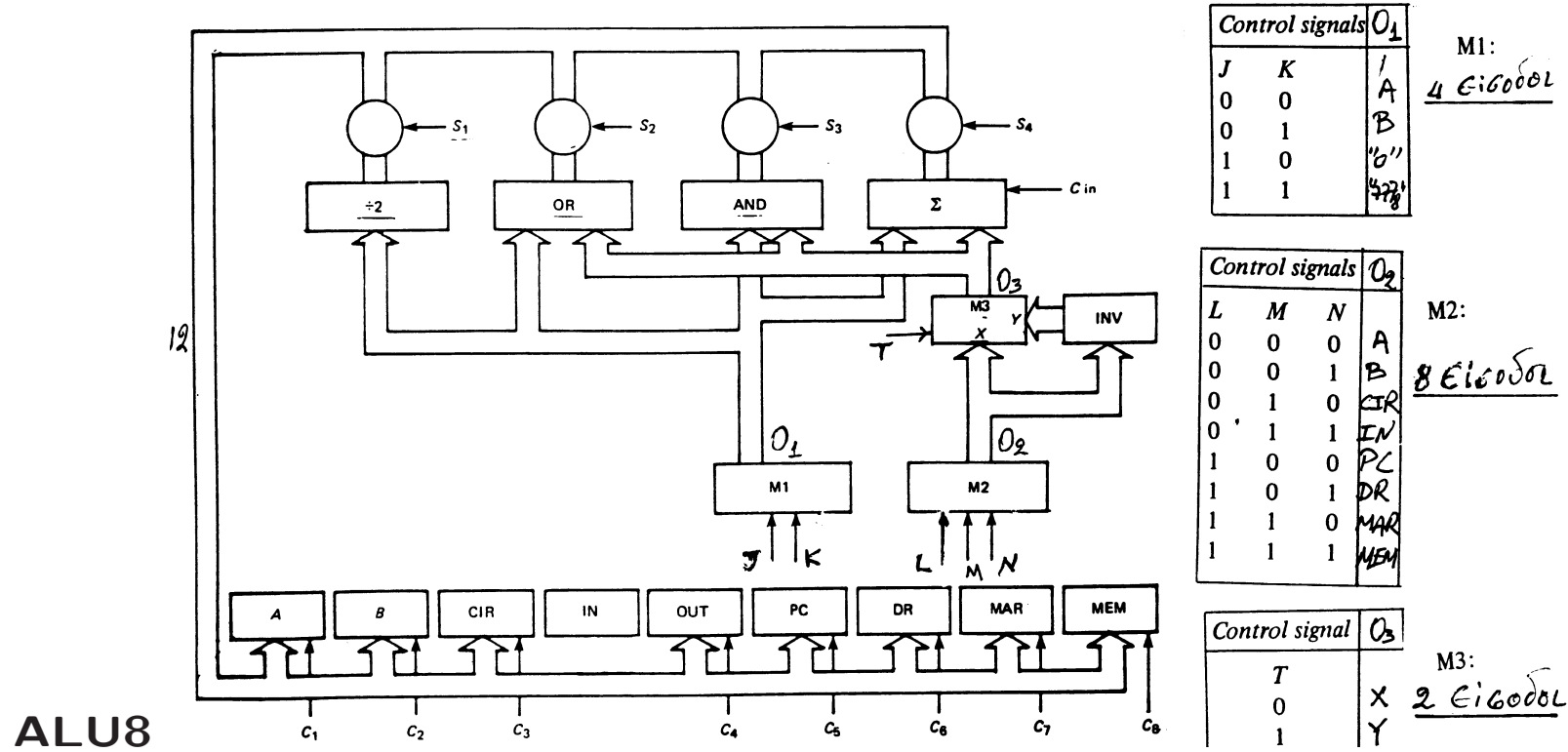
0	LDA A #3	0	LDA A #3	3		2,3,4,5,10,32
1	LDA B #0	1	LDA B #0	3	0	2,3,4,5,10,33
2	ADD B 56A	2	ADD B 56A	3	36	2,3,4,5,6,7,9,37
3	SUB A #1	3	SUB A #1	2	36	2,3,4,5,10,38
4	BNZ A 2	4	BNZ A 2	2	36	2,3,4,5,6,9,48,44
5	OUT B	2	ADD B 56A	2	43	2,3,4,5,6,7,9,37
6	END	3	SUB A #1	1	43	2,3,4,5,10,38
7	3	4	BNZ A 2	1	43	2,3,4,5,6,9,48,44
8	7	2	ADD B 56A	1	46	2,3,4,5,6,7,9,37
9	36	3	SUB A #1	0	46	2,3,4,5,10,38
		4	BNZ A 2	0	46	2,3,4,5,6,9,48
		5	OUT B	0	46	2,3,4,5,10,61,14
		6	END	0	46	2,3,4,5,10,62,62,62, .

# Εργαστήριο 13ο

1. Στην ALU του Ε.Η.Υ (βλέπε σχήμα 23) να καθοριστούν οι τιμές των σημάτων  $C_i$ , των διακοπών  $S_i$  και των γραμμών ελέγχου των πολυπλεκτών  $M_i$  για την υλοποίηση των ενεργειών του πίνακα που ακολουθεί.

ΕΝΕΡΓΕΙΑ	$S_1$	$S_2$	$S_3$	$S_4$	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$	$C_7$	$C_8$	$J$	$K$	$L$	$M$	$N$	$K$	$T$
$PC \Rightarrow MAR$																			
$MEM \Rightarrow CIR$																			
$PC + 1 \Rightarrow PC$																			
$CIR \text{ and } 777 \Rightarrow DR$																			
$DR \Rightarrow MAR$																			
$DR + A \Rightarrow MAR$																			
$DR + B \Rightarrow MAR$																			
$MEM \Rightarrow DR$																			
$DR \Rightarrow A$																			
$DR \Rightarrow B$																			
$A \Rightarrow MEM$																			
$B \Rightarrow MEM$																			
$DR + A \Rightarrow A$																			
$DR + B \Rightarrow B$																			
$A - DR \Rightarrow A$																			
$B - DR \Rightarrow B$																			
$A \text{ and } DR \Rightarrow A$																			
$B \text{ and } DR \Rightarrow B$																			
$A \text{ or } DR \Rightarrow A$																			
$B \text{ or } DR \Rightarrow B$																			
$MAR \Rightarrow PC$																			
$A \div 2 \Rightarrow A$																			
$B \div 2 \Rightarrow B$																			
$(-A) \Rightarrow A$																			
$(-B) \Rightarrow B$																			
$IN \Rightarrow A$																			
$IN \Rightarrow B$																			
$A \Rightarrow OUT$																			

**Εργαστήριο 14ο:** Δίδεται η παρακάτω βασική δομή μιας ALU8, καθώς και τα δεδομένα που πρέπει να καταλήγουν στους 3 Multiplexers της:



Δημιουργήσατε συνδέσεις μεταξύ των 8 καταχωρητών: AccA, AccB, EIR, IN, PC, DR, MAR και MEM έτσι ώστε να είναι δυνατές οι μικροεντολές που ακολουθούν και για τις οποίες δώσατε τις τιμές των σημάτων ελέγχου για την υλοποίησή τους.

## Επιθυμητές μικροεντολές για την ALU8.

PC  $\Rightarrow$  MAR

MEM  $\Rightarrow$  CIR

PC + 1  $\Rightarrow$  PC

CIR *and* 777  $\Rightarrow$

DR  $\Rightarrow$  MAR

DR + A  $\Rightarrow$  MAR

DR + B  $\Rightarrow$  MAR

MEM  $\Rightarrow$  DR

DR  $\Rightarrow$  A

DR  $\Rightarrow$  B

A  $\Rightarrow$  MEM

B  $\Rightarrow$  MEM

DR + A  $\Rightarrow$  A

DR + B  $\Rightarrow$  B

A - DR  $\Rightarrow$  A

B - DR  $\Rightarrow$  B

A *and* DR  $\Rightarrow$  A

B *and* DR  $\Rightarrow$  B

A *or* DR  $\Rightarrow$  A

B *or* DR  $\Rightarrow$  B

MAR  $\Rightarrow$  PC

A  $\div$  2  $\Rightarrow$  A

B  $\div$  2  $\Rightarrow$  B

(-A)  $\Rightarrow$  A

(-B)  $\Rightarrow$  B

IN  $\Rightarrow$  A

IN  $\Rightarrow$  B

A  $\Rightarrow$  OUT

B  $\Rightarrow$  OUT

NULL

## Βοηθητικές Ασκήσεις:

Να υπολογιστούν οι τιμές της παρακάτω συναρτήσεως:

$$f(x) = x \cdot \eta\mu x - x^2 \cdot e^{x+1} - |x^2 - 1|,$$

για τις τιμές του  $x = -5(0.5)5$ . Τέλος, **να τυπωθεί ένας πίνακας με την αντίστοιχη γραφική τους παράσταση.**

```
UNIVERSITY OF PATRAS
DEPARTMENT OF MATHEMATICS
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
FUNCTION EVALUATION
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  x                F(x)
***                *****
-5                  ...
-4.5                 ...
                    :
5                    ...
```