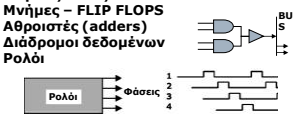
Η εκτέλεση των αλγορίθμων (Ο υπολογιστής)

HARDWARE AND COMPUTER ARCHITECTURE

ΥΛΙΚΟ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

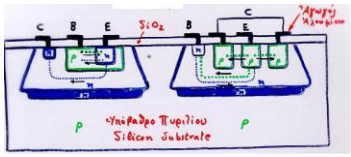
ΥΛΙΚΟ HARDWARE

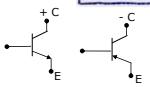
- Ημιαγωγοί – τρανζίστορς
- Πύλες OR, AND, NOT
- Εξαρτήματα / μονάδες
  - Λογικές πύλες
  - Μνήμες – FLIP FLOPS
  - Αθροιστές (adders)
  - Διάδρομοι δεδομένων
  - Ρολόι



  - Λογική Ελέγχου
    - Ενσωματωμένη λογική ελέγχου (hardwired) RISC
    - Μικροπρογραμματιζόμενη λογική (microprogram) CISC.

Διπολικά τρανζίστορς (bipolar)





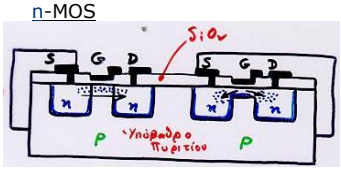
npn pnp

..... **ηλεκτρόνια** n πυρίτιο με πρόσμιξη { αρσενικού 5 / φωσφόρου

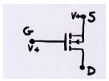
..... οπές p πυρίτιο με πρόσμιξη { βόριου 3 / αλουμινίου

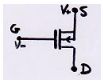
Μονοπολικά τρανζίστορς MOS FET

n-MOS



n φώσφορος
p βόριο
ηλεκτρόνια

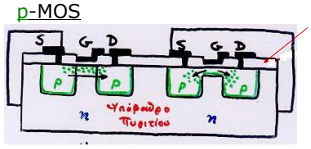




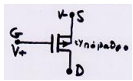
Source
Drain
Gate

MOS τρανζίστορς FET

p-MOS



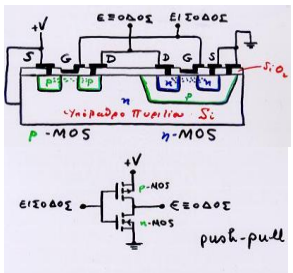
n φώσφορος
p βόριο
οπές



Enhancement mode

Depletion mode

Συμπληρωματικά MOS τρανζίστορς C-MOS



Boolean άλγεβρα (B,+,·,-,0,1)

- Αξιώματα (Huntington)
  - Κλειστότητα του B ως προς +, · .
    Για κάθε x,y ∈ B x·y ∈ B και x+y ∈ B
  - Ύπαρξη ουδέτερων στοιχείων 0,1 ∈ B
    Για κάθε x ∈ B x+0 = x και x·1 = x.
  - Αντιμεταθετική ιδιότητα (commutativity)
    Για κάθε x,y ∈ B x+y = y+x x·y = y·x.
  - Επιμεριστική ιδιότητα (distributivity)
    Για κάθε x,y,z ∈ B x+(y·z) = (x+y)·(x+z)
    x·(y+z) = (x·y)+(x·z)
  - Ύπαρξη συμπληρώματος (complement)

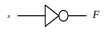
$$\forall x \in B \ \exists \bar{x} \in B \text{ με } x+\bar{x}=1,\ x\cdot\bar{x}=0$$

  - Υπάρχουν τουλάχιστον δύο στοιχεία x,y ∈B με x ≠ y.

Θεωρήματα

- **Αρχή Δυισμού <B,+,·,-,0,1> <B, ·,+,-,1,0>**
- **Θ1. Τα στοιχεία 1 και 0 είναι μοναδικά. Το συμπλήρωμα $\bar{x}$ του x είναι μοναδικό.**
- **Θ2. Για κάθε x∈B**
  x+x = x
  x·x = x
- **Θ3. Για κάθε x ∈ B**
  x+1 = 1 και x·0 = 0
- **Θ4. Απορρόφησης. Για κάθε x,y ∈ B.**
  x+x·y = x και x·(x+y)=x
- **Θ5. Τα στοιχεία 0 και 1 είναι διαφορετικά μεταξύ τους και ισχύει $\bar{1}=0$.**
- **Θ6. Για κάθε x ∈ B ισχύει $\bar{\bar{x}}=x$.**
- **Θ7. Προσεταιρισμός**
  x+(y+z) = (x+y)+z και x·(y · z) = (x · y) · z
- **Θ8. De Morgan** $\overline{x+y}=\bar{x}\cdot\bar{y}$ $\overline{x\cdot y}=\bar{x}+\bar{y}$
- **Θ9. Επιμερισμός** $x+\bar{x}\cdot y=x+y$ $x\cdot(\bar{x}+y)=x\cdot y$

Λογικές πύλες

όνομα	Συμβολισμός	Συνάρτηση	Πίνακας αληθείας
AND		$F = xy$	x y F: 0 0 0; 0 1 0; 1 0 0; 1 1 1
OR		$F = x+y$	x y F: 0 0 0; 0 1 1; 1 0 1; 1 1 1
NOT		$F = \bar{x}$	x F: 0 1; 1 0

Λογικές πύλες

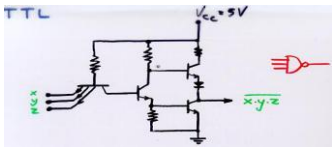
όνομα	Συμβολισμός	Συνάρτηση	Πίνακας αληθείας
NAND		$F = \overline{xy}$	x y F: 0 0 1; 0 1 1; 1 0 1; 1 1 0
NOR		$F = \overline{x+y}$	x y F: 0 0 1; 0 1 0; 1 0 0; 1 1 0
XOR		$F = x \oplus y$	x y F: 0 0 0; 0 1 1; 1 0 1; 1 1 0

Λογικές Πύλες

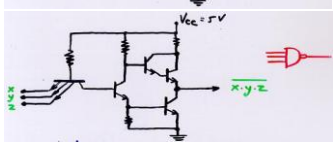
όνομα	Συμβολισμός	Συνάρτηση	Πίνακας αληθείας
XNOR ή EQUI		$F = x \otimes y$	x y F: 0 0 1; 0 1 0; 1 0 0; 1 1 1
Buffer (ενισχυτής)		$F = x$	x F: 0 0; 1 1

TTL

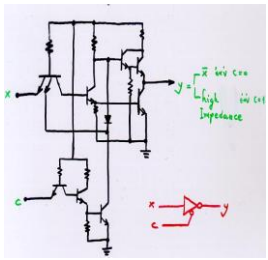
Standard or Low –Power NAND πύλη

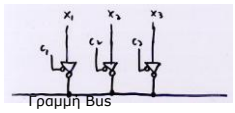


High speed TTL NAND



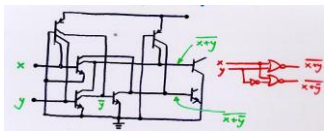
Three-state TTL πύλη



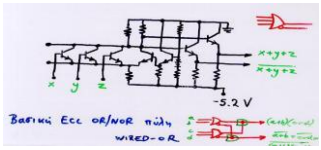


I^2L – ECL

I^2L (Integrated Injection Logic)

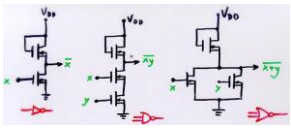


ECL (Emitter-Coupled Logic)

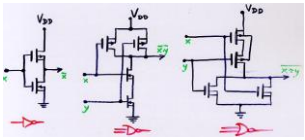


MOS

n-MOS πύλες



c-mos πύλες

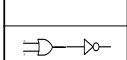
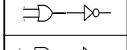
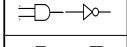


Συναρτήσεις των δύο μεταβλητών

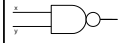
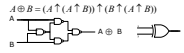
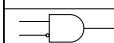
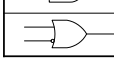
x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

$f_0 = 0$	σταθερά0	$f_8 = \overline{x+y} = \overline{x} \cdot \overline{y}$	$x \downarrow y$ NOR
$f_1 = xy$	AND x και y	$f_9 = \overline{xy} + xy$	$x \otimes y$ ισοδυναμία x ίσον y
$f_2 = x\overline{y}$	x/y x αλλά όχι y απαγόρευση	$f_{10} = \overline{y}$	NOT y
$f_3 = x$	μεταφορά - προβολή x	$f_{11} = x + \overline{y}$	$x \subset y$ συνέπεια αν $y \rightarrow x$
$f_4 = \overline{x}y$	y/x y αλλά όχι x	$f_{12} = \overline{x}$	NOT x
$f_5 = y$	προβολή y	$f_{13} = \overline{x} + y$	$x \supset y$ συνέπεια αν $x \rightarrow y$
$f_6 = x\overline{y} + \overline{x}y$	$x \oplus y$ XOR x ή y αλλά όχι μαζί	$f_{14} = \overline{xy} = \overline{x} + \overline{y}$	$x \uparrow y$ NAND
$f_7 = x + y$	OR x ή y	$f_{15} = 1$	σταθερά 1

Πρωτόγονοι τελεστές

	Πρωτόγονοι σύμβολα	Παράγωγα
	+ και -	$A \cdot B = \overline{\overline{A} + \overline{B}}$
	· και -	$A + B = \overline{\overline{A} \cdot \overline{B}}$
	⊕ και ·	$A + B = (A \oplus B) \oplus (A \cdot B)$ $\overline{A} = A \oplus 1$
	⊗ και ·	$A + B = (A \otimes B) \otimes (A \cdot B)$ $\overline{A} = A \otimes 0$

Καθολικοί τελεστές

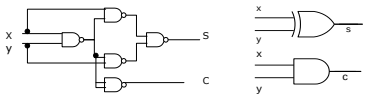
	Πρωτόγονα σύμβολα	Παράγωγα
	NOR $x \downarrow y$	$A + B = (A \downarrow B) \downarrow (A \downarrow B)$ $A \cdot B = (A \downarrow A) \downarrow (B \downarrow B)$ $\overline{A} = A \downarrow 0$
	NAND $x \uparrow y$	$A + B = (A \uparrow A) \uparrow (B \uparrow B)$ $A \cdot B = (A \uparrow B) \uparrow (A \uparrow B)$ $\overline{A} = A \uparrow 1$ 
	$x \,/\, y$	$A + B = 1/((1/A)/B)$ $\overline{A} = 1/A$ $A \cdot B = A/(1/B)$
	$x \subset y$	$A + B = A \subset (0 \subset B)$ $\overline{A} = 0 \subset A$ $A \cdot B = 0 \subset ((0 \subset A) \subset B)$

Αθροιστές

1. **Ημιαθροιστής (half adder)**

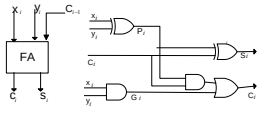
x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = \bar{x}y + x\bar{y} = x \oplus y$$
$$C = xy$$



Αθροιστές

2. **Πλήρης αθροιστής – full adder**



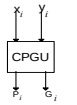
X_i	Y_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S_i = x_i \oplus y_i \oplus c_{i-1}$$
$$c_i = x_i y_i + (x_i \oplus y_i) c_{i-1}$$

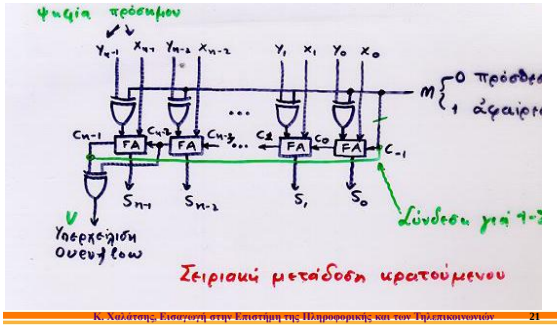
$G_i = x_i y_i$ Συνάρτηση γέννησης κρατουμένου

$P_i = x_i \oplus y_i$ Συνάρτηση μετάδοσης κρατουμένου

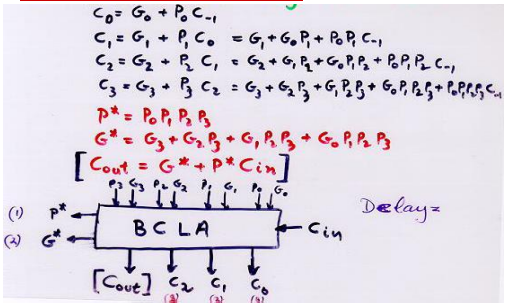
$$S_i = P_i \oplus C_{i-1}$$
$$C_i = G_i + P_i C_{i-1}$$



Αθροιστής/Αφαιρέτης δύο n-ψήφιων αριθμών στο 2-Σ

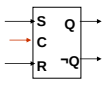


Αθροιστής με πρόβλεψη κρατούμενου κατά ομάδες



Flip–flops – Μνημονικά στοιχεία

☐ **Set-Reset flip-flop S-R**

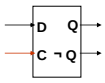


S	R	Q	Q'
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

Q'= S + ¬RQ
SR = 0

Q' είναι η επομένη κατάσταση

☐ **Delay flip-flop D**



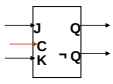
D	Q	Q'
0	0	0
0	1	0
1	0	1
1	1	1

Q´=D

- Level Triggered – Διέγερση επιπέδου
- Master – Slave
- Edge Triggered – Μετωπική διέγερση
- Preset – Clear είσοδοι

Flip–flops – Μνημονικά στοιχεία

☐ **J - K flip-flop**

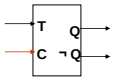


J	K	Q	Q'
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Q´=J¬Q + ¬KQ

Q' είναι η επομένη κατάσταση

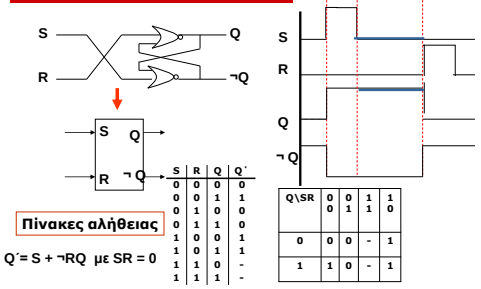
☐ **T flip-flop**



T	Q	Q'
0	0	0
0	1	1
1	0	1
1	1	0

Q´=Q¬T + T¬Q

Δικατάστατο Μνημονικό στοιχείο: Set – Reset flip-flop - 1 bit



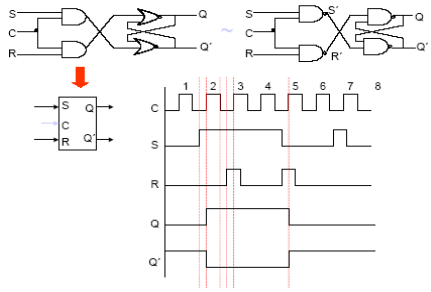
S	R	Q	Q΄
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

Πίνακες αλήθειας

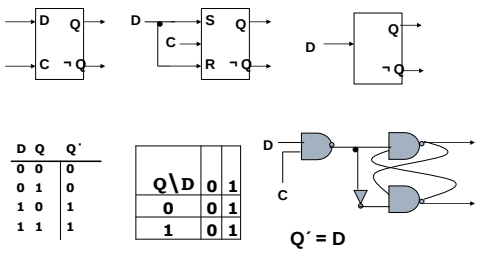
Q΄= S + ¬RQ με SR = 0

Q\SR	0 0	0 1	1 1	1 0
0	0	0	-	1
1	1	0	-	1

S – R flip –flop με ρολόι



Delay flip-flop



D	Q	Q΄
0	0	0
0	1	0
1	0	1
1	1	1

Q\D	0	1
0	0	1
1	0	1

Q΄ = D

Αρχιτεκτονική Υπολογιστή

VON NEUMANN

ΤΙ ΚΑΝΕΙ ΕΝΑΣ ΥΠΟΛΟΓΙΣΤΗΣ;

ΠΑΙΡΝΕΙ ΠΛΗΡΟΦΟΡΙΕΣ ΑΠΟ ΤΟΝ ΕΞΩΤΕΡΙΚΟ ΚΟΣΜΟΣ (ΜΕΣΩ ΜΟΝΑΔΩΝ ΕΙΣΟΔΟΥ), ΤΙΣ ΕΠΕΞΕΡΓΑΖΕΤΑΙ ΑΡΙΘΜΗΤΙΚΑ-ΛΟΓΙΚΑ (ΜΕΣΩ ΤΟΥ ΕΠΕΞΕΡΓΑΣΤΗ ΚΑΙ ΤΗΣ ΜΝΗΜΗΣ) ΚΑΙ ΠΑΡΑΓΕΙ ΑΠΟΤΕΛΕΣΜΑΤΑ-ΠΛΗΡΟΦΟΡΙΕΣ ΓΙΑ ΤΟΝ ΕΞΩΤΕΡΙΚΟ ΚΟΣΜΟΣ (ΜΕΣΩ ΤΩΝ ΜΟΝΑΔΩ ΕΞΟΔΟΥ)

ΚΥΡΙΕΣ ΣΥΝΙΣΤΩΣΕΣ

- **ΕΠΕΞΕΡΓΑΣΤΗΣ (Ρ)**

  Είναι η καρδιά του υπολογιστή. Περιέχει στοιχεία για την εκτέλεση αριθμητικών και λογικών πράξεων σύμφωνα με κάποια καθορισμένη ακολουθία
- **ΜΝΗΜΗ (Μ)**

  Είναι ένα σύνολο καταχωρητών (Δυαδικές λέξεις) που περιέχουν το πρόγραμμα (σειρά εντολών) και δεδομένα (DATA) για τον επεξεργαστή.
- **ΜΟΝΑΔΕΣ ΕΙΣΟΔΟΥ/ΕΞΟΔΟΥ (Ι/Ο)**

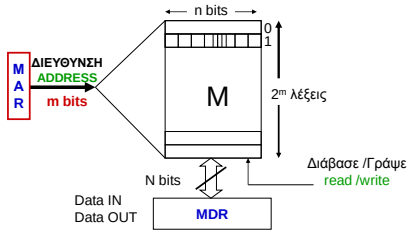
  Παρέχουν στον υπολογιστή τα δεδομένα που θα επεξεργαστεί από τον εξωτερικό κόσμο και μεταφέρουν έξω τα αποτελέσματα.

ΜΝΗΜΗ (Μ)

- **Η ΜΝΗΜΗ περιέχει**
  - **ΕΝΤΟΛΕΣ (instructions) και**
  - **ΕΝΤΕΛΑ (operands)**
- **Η μνήμη περιέχει καταχωρητές (θέσεις-locations).**
- **Κάθε θέση μνήμης χαρακτηρίζεται από:**
  - **ΔΙΕΥΘΥΝΣΗ: προσδιορίζει πια θέση πρόκειται να διαβάσεις ή να γράψεις**
  - **ΠΕΡΙΕΧΟΜΕΝΟ: η πληροφορία που είναι φυσικά φυλαγμένη - γραμμένη στη θέση μνήμης.**

Δομή της Μνήμης

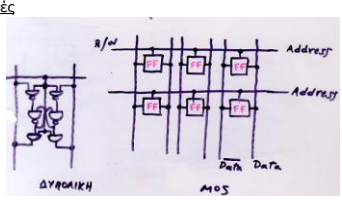
MAR=Memory Address Register, Καταχωρητής Διευθύνσεων Μνήμης
MDR= Memory Data Register, Καταχωρητής Δεδομένων Μνήμης



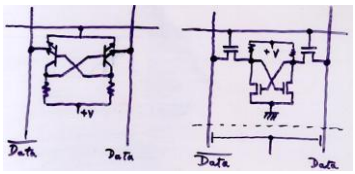
Τύποι μνημών RAM

- Τύποι RAM
  - Στατικές – static
  - Δυναμικές – dynamic (refresh αναζωογόνηση)

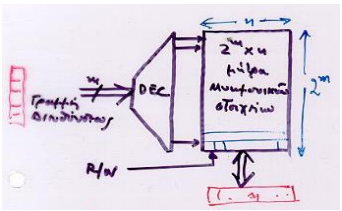
Στατικές



Τύποι μνημών RAM

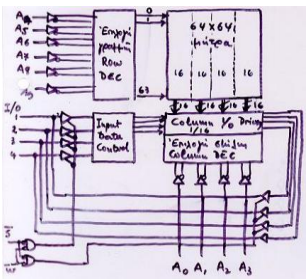


RAMs Μνήμες Τυχαίας Προσπέλασης

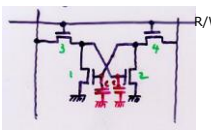


Παράδειγμα: 1024 words x 4 bits/word RAM

RAMs Μνήμες Τυχαίας Προσπέλασης



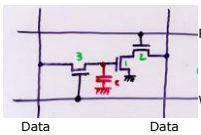
Δυναμικές RAMs



R/W word select

Κυψέλη 4-1-2

Data Lines
Address
Αριθμός τρανζίστορς



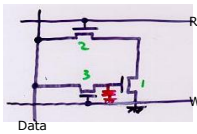
Read select

Write select

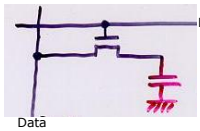
Κυψέλη 3-2-2

Data Data

Δυναμικές RAMs



Κυψέλη 3-2-1



Κυψέλη 1-1-1

Μνήμη

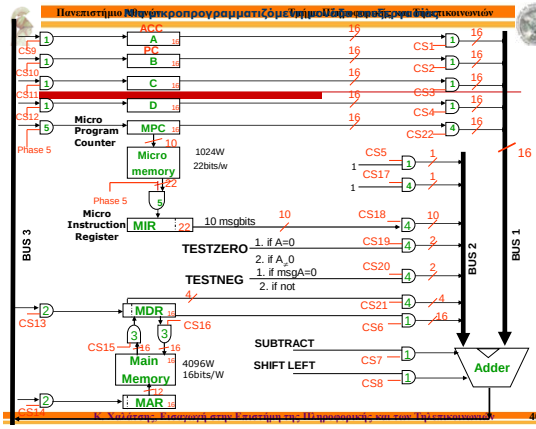
- **Η πληροφορία που είναι γραμμένη σε μια θέση μνήμης με τη μορφή bits (0,1) – γνωστή σαν ΛΕΞΗ. Ο αριθμός των bits (n) ανά λέξη είναι καθοριστική παράμετρος του όλου υπολογιστή και είναι γνωστή σαν ΜΗΚΟΣ ΛΕΞΗΣ (word length). Συνήθεις τιμές είναι n= 4, 8, 12, 16, 24, 32 bits**
- **Το μέγεθος της μνήμης καθορίζει την ισχύ του υπολογιστή όσον αφορά το πλήθος των DATA και των εντολών που μπορούν να αποθηκευτούν. Εάν η ΔΙΕΥΘΥΝΣΗ είναι m bits τότε το μέγιστο μέγεθος είναι 2^m ΛΕΞΕΙΣ.**
  Συνήθεις τιμές του m :
  m = 12, 16, 20, 24, 32 bits
- **ΕΙΔΗ Μνήμης**
  - **RAM ή RWM ανάγνωσης/εγγραφής**
  - **ROM μόνο ανάγνωσης**

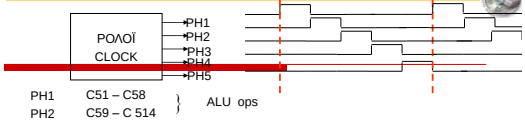
ΠΡΟΓΡΑΜΜΑ – ΕΝΤΟΛΕΣ – ΕΝΤΕΛΑ
PROGRAM – INSTRUCTIONS – OPERANDS

ΠΡΟΓΡΑΜΜΑ: σύνολο εντολών που έχουν δημιουργηθεί από τον προγραμματιστή και έχουν φορτωθεί στην μνήμη. Οι εντολές εκτελούνται μια μια από τον επεξεργαστή. Το πρόγραμμα καθορίζει την εργασία που κάνει ο υπολογιστής.

Εντολή: Μια διαταγή που εκτελείται από τον επεξεργαστή.

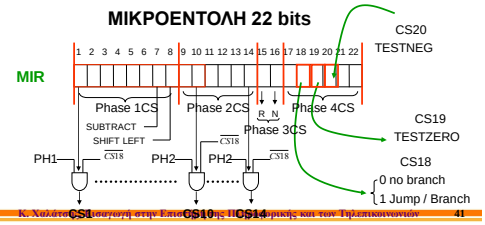
- **ΕΝΤΟΛΕΣ ΕΠΕΞΕΡΓΑΣΙΑΣ**
  - **Αριθμητικές**
  - **Μεταφοράς**
  - **Λογικές**
- **ΕΝΤΟΛΕΣ ΕΛΕΓΧΟΥ**
  Τροποποιούν την κανονική ροή εκτέλεσης των εντολών
- **ΕΝΤΕΛΟ - ΔΕΔΟΜΕΝΟ**
  Μονάδα πληροφορίας φυλαγμένη στη μνήμη πάνω στην οποία επενεργεί ο επεξεργαστής όταν εκτελεί μια ΕΝΤΟΛΗ





PH1	C51 – C58	ALU ops
PH2	C59 – C 514	
PH3	C515 – C516	Read / Write in Main Memory
PH4	C517 – C522	Υπολογισμός του MPC, διεύθυνση επομένης μικροεντολής
PH5	Phase 5	Read Μικροεντολή

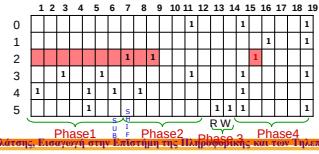
ΜΙΚΡΟΕΝΤΟΛΗ 22 bits



Μικροπρόγραμμα : Πολλαπλασιασμός αργός

```
module multiply
        { C*A→M(1) }
        set MDR = 0*
        repeat A times
                add C to MDR
        move 1 to MAR & write to memory
```

MPC (micro memory address)	Μικροεντολή	Σχόλιο
0	0+0→ MDR ; MPC + 1 → MPC	Initialize MDR
1	MPC + TESTZERO → MPC	Check if A=0
2	5 → MPC	Yes ; exit
3	C + MDR → MDR; MPC +1 → MPC	No ; add C to MDR
4	A – 1 → A ; 0 + 1→MPC	Decrement A; go to top of loop
5	0 + 1 → MAR, Write ; MPC+1→MPC	Write result into M(1)



Μικροπρόγραμμα : Πολλαπλασιασμός Ταχύς

```
   000111   C
x  000110   A
   000000
    000111     C shifted once
   000111      C shifted twice
  000000
 000000
000000
101010
```

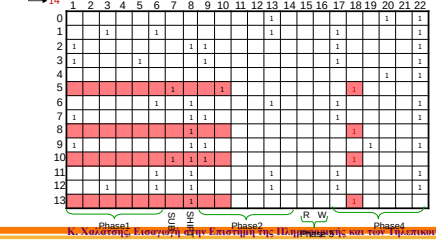
module multiply
{ C x A →MDR }
1 set MDR = 0
2 if leftmost bit of A = 1
then add C to MDR
3 shift A left one position
4 add 1 to A {σημάδι}
5 while A has some bit other than the leftmost = 1 do
5.1 shift MDR left one position
5.2 if left most bit of A=1
then add C to MDR
5.3 shift A left one position

Βήματα εκτέλεσης του μικροπρογράμματος ταχύ πολλαπλασιασμού για το παράδειγμα μας

Βήμα	A	C	MDR
0	000110	000111	
1			000000
2			
3	001100		
4	001101 add 1		
5			
51			000000 ←shift
52			
53	011010 ←shift		
5			
51			000000 ←shift
52			
53	110100 ←shift		
5			
51			000000 ←shift
52			000111 MDR+C
53	101000 ←shift		
5			
51			001110 ←shift
52			010101 MDR+C
53	010000 ←shift		
5			
51			101010 ←shift
52			
53	100000 ←shift		101010
END			ΑΠΑΝΤΕΧΟΜΕ


```
0   0+0 → MDR;        MPC+TESTNEG→MPC   Set MDR=0.  test if msA = 1
1   C+MDR → MDR;      MPC+1→MPC         (Yes) add C to MDR
2   A+ 0  shr A;      MPC+1→MPC         (No) Shift A
3   A+1 → A;          MPC+1→MPC         set END marker
4      MPC+TESTNEG→MPC                  check if msA=1
5      9 → MPC                          Yes, jump ahead
6   0+MDR shl MDR;    MPC+1→MPC         No, shift MDR
7   A+0  shr A;       MPC+1→MPC         shift A
8      4→MPC                            loop
9   A+0  shr A;       MPC+TESTZERO→MPC  shift A; test if A=0;
10     14→MPC                           (Yes) exit
11  0+MDR  shl MDR;   MPC+1→MPC         (No) shift MDR
12  C+MDR → MDR;      MPC+1→MPC         Add C to MDR
13     4→MPC                            loop
14
```

msA+1
exit



Εκτέλεση του Μικροπρογράμματος C = 000111, A = 000110, 7 x 6 = 42

Αρ.	μικροεντολή
0	MDR = 000000 & Test msA
	No
2	← A = 001101
3	A = 001101 add marker
4	Test msA
	No
6	← MDR = 000000
7	← A = 011010
8	GO TO 4
4	Test msA
	No
6	← MDR = 000000
7	← A = 110100
8	GO TO 4
4	Test msA
	Yes
5	GO TO 9
9	← A = 101000 Test A=0
	No

Αρ.	μικροεντολή
11	← MDR = 000000
12	C + MDR = 000111
13	GO TO 4
4	Test msA
	Yes
5	GO TO 9
9	← A = 010000,Test A=0
	No
11	← MDR = 001110
12	C + MDR = 010101
13	GO TO 4
4	Test msA
6	← MDR = 101010
7	← A = 100000
8	GO TO 4
4	Test msA
	Yes
5	GO TO 9
9	← A = 000000 , Test A=0
	Yes
10	GO TO 14 EXIT

ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ

(της μικροπραγραμματιζόμενης μηχανής)

- **Μορφή εντολής (τύπου accumulator/συσσωρευτή)**

OP CODE	M address
4 bits	12 bits

  - **Δηλαδή, οι αριθμητικές πράξεις γίνονται μεταξύ του συσσωρευτή και κάποιας θέσης μνήμης και το αποτέλεσμα πηγαίνει στον συσσωρευτή**
- **ACCUMULATOR / ΣΥΣΣΩΡΕΥΤΗΣ είναι ο καταχωρητής Α**
- **Απαριθμητής εντολών (Program Counter) είναι ο καταχωρητής Β**

ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ
Ρεπερτόριο εντολών

0001	LOAD M	M→ ACC
0010	STORE M	ACC → M
0011	ADD M	ACC + M → ACC
0100	SUBTRACT M	ACC - M → ACC
0101	MULTIPLY M	ACC * M → ACC
0110	DIVIDE M	ACC / M → ACC
0111	JUMP M	Jump to M
1000	JUMPZERO M	Jump to M if ACC= 0
1001	JUMPMSB M	Jump to M if msbACC = 1
1010	JUMPSUB M	Jump to SUBROUTINE at M
1011	RETURN M	Return from subroutine at M

- **ACCUMULATOR / ΣΥΣΣΩΡΕΥΤΗΣ είναι ο καταχωρητής Α**
- **Απαριθμητής εντολών (Program Counter) είναι ο καταχωρητής Β**

Παράδειγμα προγράμματος y = 2^x

Διεύθυνση	Εντολή	Σχόλια
946	LOAD 958	Set y = 1 A ←1
947	STORE 957	y ←A
948	LOAD 956	x → A
949	JUMPZERO 959	if x = 0 finish
950	SUBTRACT 958	A - 1 → A
951	STORE 956	x: = x – 1
952	LOAD 957	y → A
953	ADD 957	A + A → A
954	STORE 957	y: = y + y double
955	JUMP 948	repeat loop
956	x	
957	y	
958	1	
959	JUMP 959	

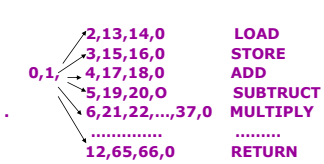
ΚΥΚΛΟΣ ΜΗΧΑΝΗΣ

- Ανάκληση εντολής από την κύρια μνήμη
- Αποκωδικοποιήση εντολής
- Εκτέλεση εντολής
  - [ανάκληση δεδομένων]
  - πράξη / εκτέλεση
  - [αποθήκευση αποτελέσματος]

- module interpreter
  - repeat
    - fetch next machine language instruction
    - add 1 to B (PC +1→ PC)
    - decode instruction
    - execute instruction
  - forever

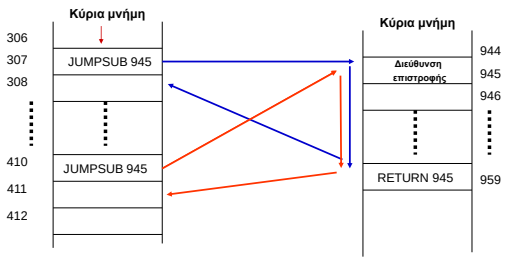
Εκτέλεση Εντολής

Σε κάθε εντολή μηχανής αντιστοιχεί ένα μικροπρόγραμμα

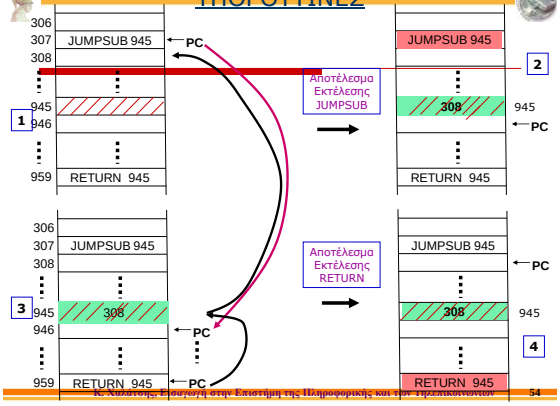


Micro address	Μικροεντολή	Σχόλια
0	B+0→MAR; Read; MPC+1→MPC	Next I → MDR
1	B+1→B; MPC+MDR→MPC	PC+1→PC; Decode
2	13 → MPC	Load
3	15 → MPC	Store
4	17 → MPC	Add
5	19 → MPC	Sub
6	21 → MPC	Mult
7	38 → MPC	Div
8	55 → MPC	Jump
9	56 → MPC	Jump Z
10	59 → MPC	Jump msb
11	62 → MPC	Jump sub
12	65 → MPC	Return
13	0 + MDR → MAR; Read ; MPC+1 →MPC	Load: M →MDR
14	0+ MDR →A; 0+0 →MPC	MDR →Acc
15	0+MDR →MAR; MPC+1 →MPC	Store: M, →MAR
16	A+0 →MDR; Write; 0+0 →MPC	Acc →M
17	0+MDR →MAR; Read; MPC+1 →MPC	Add: M →MDR
18	A+MDR →A; 0+0 →MPC	Acc + M → Acc
19	0+MDR → MAR; Read; MPC+1 → MPC	Sub: M →MDR
20	A- MDR → A; 0+0 →MPC	Acc-M →Acc
21	.....	Mult
...	.....	.....
38	.....	Div
...	.....	.....
55	0+MDR →B; 0+0 →MPC	Jump: M →PC
56	MPC+TESTZERO →MPC	JumpZ:
57	0+MDR→ B; 0+0 →MPC	A=0 Jump M
58	0+0 →MPC	A≠0 Continue
59	MPC+TESTNEG →MPC	Jumpmsb:
60	0+ MDR → B; 0+0 → MPC	Amsb =1 Jump M
61	0+0 →MPC	else Continue
62	0+MDR →MAR → C; MPC+1 →MPC	JumpSub: M →C
63	B+0 → MDR; Write; MPC+1 →MPC	Save Return address →M
64	C+1 → B; 0+0 →MPC	Jump M
65	0+MDR →MAR; Read; MPC +1 →MPC	Return: Return address →MAR
66	0+MDR → B; 0+0 →MPC	Return address →PC

Κλήση ενός "module" από δύο διαφορετικές θέσεις (παράδειγμα)

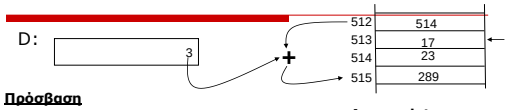


ΥΠΟΡΟΥΤΙΝΕΣ



ΒΕΛΤΙΩΣΕΙΣ ΤΗΣ ΜΗΧΑΝΗΣ

1) ΤΡΟΠΟΙ ΠΡΟΣΒΑΣΗΣ = addressing modes
καταχωρητής δείκτη



Πρόσβαση			Accumulator
Απ' ευθείας	Immediate	LOAD #512	512
Άμεση	Direct	LOAD 512	514
Έμμεση	Indirect	LOAD 512, I	23
Με δείκτη	Indexed	LOAD 512, X	289

2) Καταχωρητές εργασίας

Μια ομάδα καταχωρητών στον επεξεργαστή που κρατούν ενδιάμεσα αποτελέσματα (και όχι μόνο ένας ACC)

3) Pipelining - Σωλήνωση

Επιτρέπει την ταυτόχρονη εκτέλεση περισσοτέρων της μιας εντολής (η κάθε εκτελούμενη εντολή είναι σε διαφορετική φάση εκτέλεσης