



Πανεπιστήμιο Δυτικής Μακεδονίας  
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

---

# Ψηφιακή Σχεδίαση

**Ενότητα 11:** Μνήμη και Προγραμματίσιμη Λογική

Δρ. Μηνάς Δασυγένης

[mdasyg@ieee.org](mailto:mdasyg@ieee.org)

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής  
Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

---



# Άδειες Χρήσης

---

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Σκοπός της ενότητας

---

- Να γίνει ανάλυση των κατηγοριών της μνήμης.
- Να αναφερθούν τα χαρακτηριστικά της μνήμης.
- Σχεδίαση μνήμης σαν ψηφιακό κύκλωμα.



# Ένα πολύ σημαντικό στοιχείο κυκλώματος: Μονάδα Μνήμης

---

- Μεταφέρονται ψηφιακές πληροφορίες για αποθήκευση.
- Εξάγονται αποθηκευμένες πληροφορίες για επεξεργασία.
- Αποτελείται από πλήθος ίδιων κυκλωματικών κυττάρων.
- Δύο τύποι μνημών:
  - RAM ( γραφή + ανάγνωση )
  - ROM ( μόνο ανάγνωση )



# Μνήμη τύπου RAM

---

- RAM: Random Access Memory ( μνήμη τυχαίας προσπέλασης )
  - Λειτουργία Γραφής.
  - Λειτουργία Ανάγνωσης.
- Ο χρόνος για τη μεταφορά πληροφορίας από και προς οποιαδήποτε θέση είναι πάντα ίδιος.
- Αποθήκευση σε ομάδες από bit := words.
- Μετακίνηση words από / προς τη μνήμη.



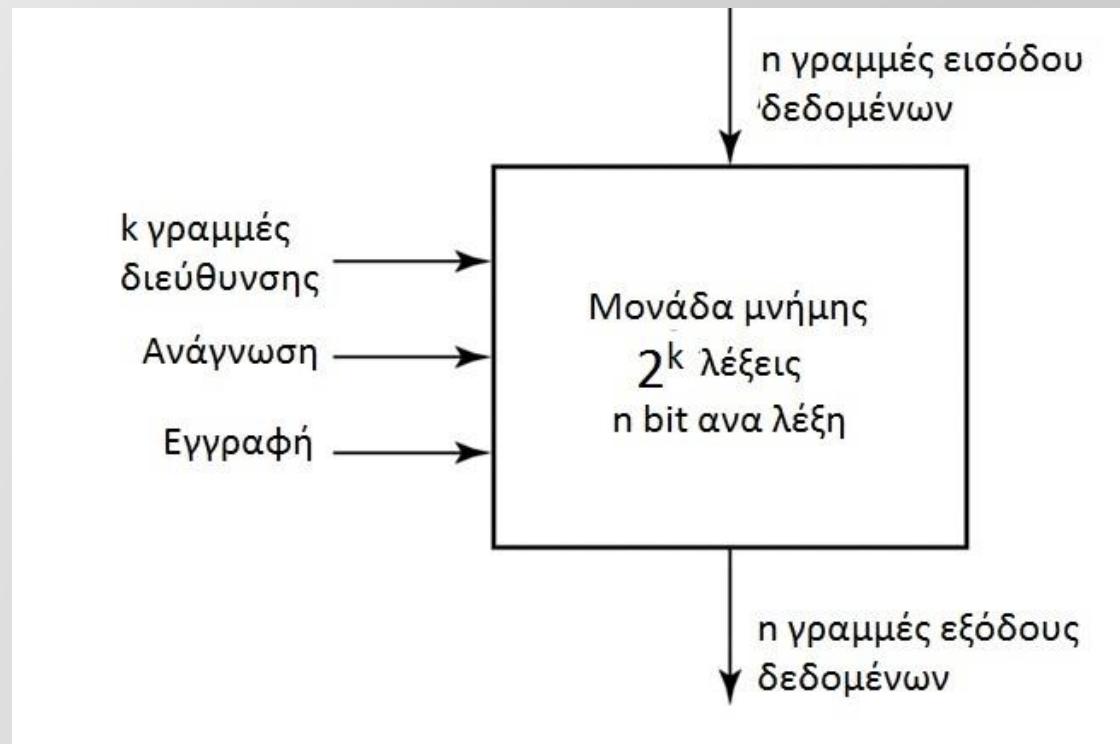
# Η RAM έχει 3 Γραμμές Επικοινωνίας

---

- Γραμμές εισόδου και εξόδου δεδομένων ( data bus ).
- Γραμμές επιλογής διεύθυνσης ( address bus ).
- Γραμμές ελέγχου ( control bus ).



# Σχηματικό Διάγραμμα της RAM





# Χαρακτηριστικά χωρητικότητας της μνήμης RAM (1)

---

- Αριθμό λέξεων.
- Αριθμό bit σε κάθε λέξη.
- $k$  bit διεύθυνσης ορίζουν  $2^k - 1$  διευθύνσεις.
- Μεγέθη μνημών:
  - KB = 1024 Bytes (  $2^{10}$  )
  - MB(  $2^{20}$  )
  - GB(  $2^{30}$  ) , 4GB :=  $2^{32}$



## Χαρακτηριστικά χωρητικότητας της μνήμης RAM (2)

Memory	# of addr	# of data lines	# of addr lines	# of total bytes
1M x 8	1,048,576	8	20	1 MB
2M x 4	2,097,152	4	21	1 MB
1K x 4	1024	4	10	512 B
4M x 32	4,194,304	32	22	16 MB
16K x 64	16,384	64	14	128 KB



# Τι δεδομένα φέρουν οι λέξεις;

---

- Μια λέξη μνήμης είναι μια ομάδα 0 , 1.
- Μπορεί να αναπαριστά:
  - Έναν αριθμό.
  - Έναν ή περισσότερους αλφαριθμητικούς χαρακτήρες.
  - Δυαδικά κωδικοποιημένη πληροφορία.



# Ενδεικτικά Περιεχόμενα μνήμης 1024x16

Πρώτη λέξη



**k bit διεύθυνσης**

**m λέξεις**

**$2^k \geq m$**

Τελευταία λέξη



Διεύθυνση μνήμης		
Δυαδικό	Δεκαδικό	Περιεχόμενα μνήμης
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
.	.	.
.	.	.
.	.	.
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101



# Λειτουργίες Γραφής & Ανάγνωσης

---

- Γραφή δεδομένων
  - Εφαρμόζεται η διεύθυνση.
  - Εφαρμόζονται τα bit δεδομένων.
  - Ενεργοποιείται η είσοδος εγγραφής.
- Ανάγνωση δεδομένων
  - Εφαρμόζεται η διεύθυνση.
  - Ενεργοποιείται η είσοδος ανάγνωσης.



# Τροποποίηση των εισόδων ελέγχου

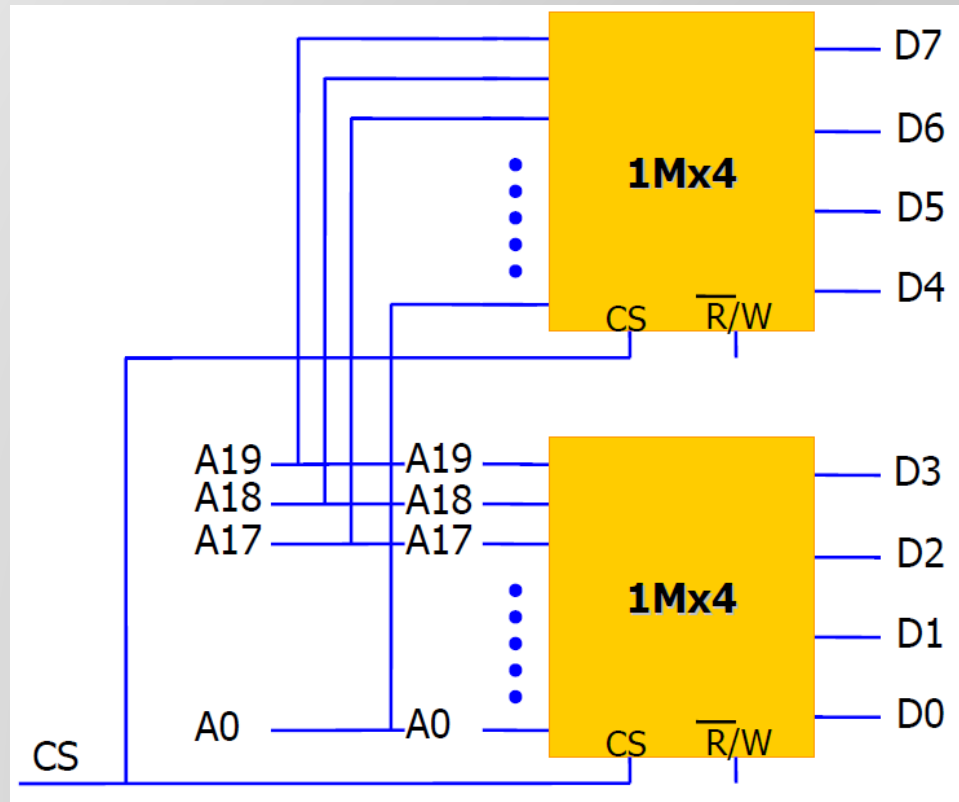
---

- Τα εμπορικά κυκλώματα μνήμης αντί για μια είσοδο ελέγχου γραφής και μια ανάγνωσης χρησιμοποιούν 2 εισόδους ελέγχου ως εξής:
  - Είσοδος Επίτρεψης Μνήμης ( chip select – cs )  
( Μόνο αν εφαρμοστεί '1' λειτουργεί η μνήμη ).
  - Είσοδος Ανάγνωσης/Γραφής ( rw )  
( Αν εφαρμοστεί 1 ανάγνωση, αν εφαρμοστεί 0 γραφή ).



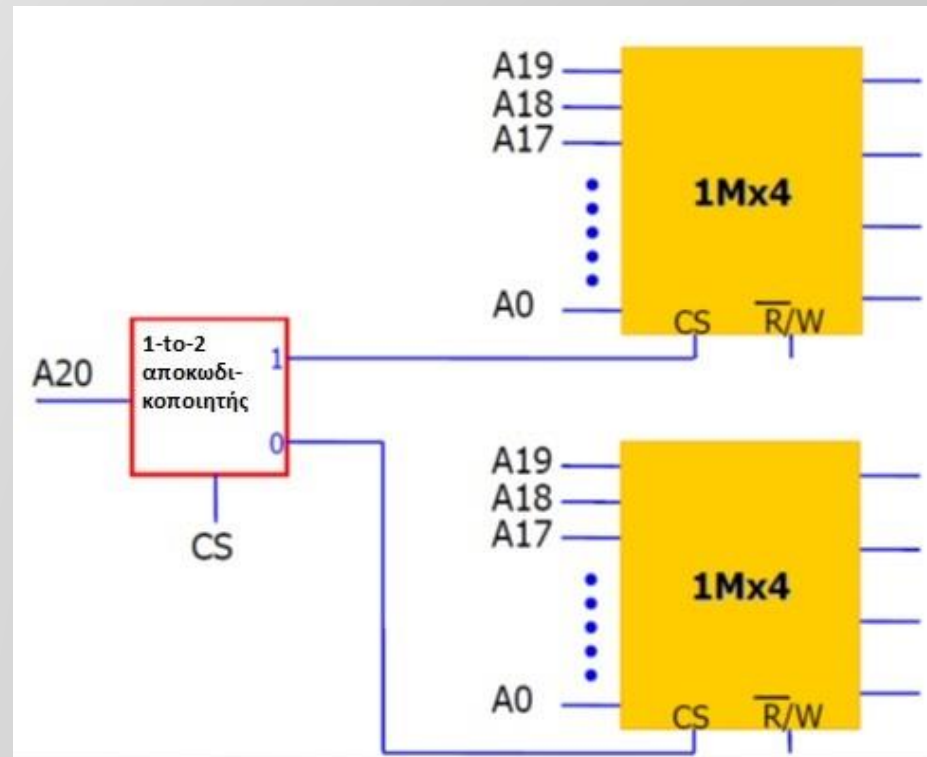
# Ιεραρχική Κατασκευή Μνήμης (1)

- Κατασκευή 1Mx8 χρησιμοποιώντας 1Mx4



# Ιεραρχική Κατασκευή Μνήμης (2)

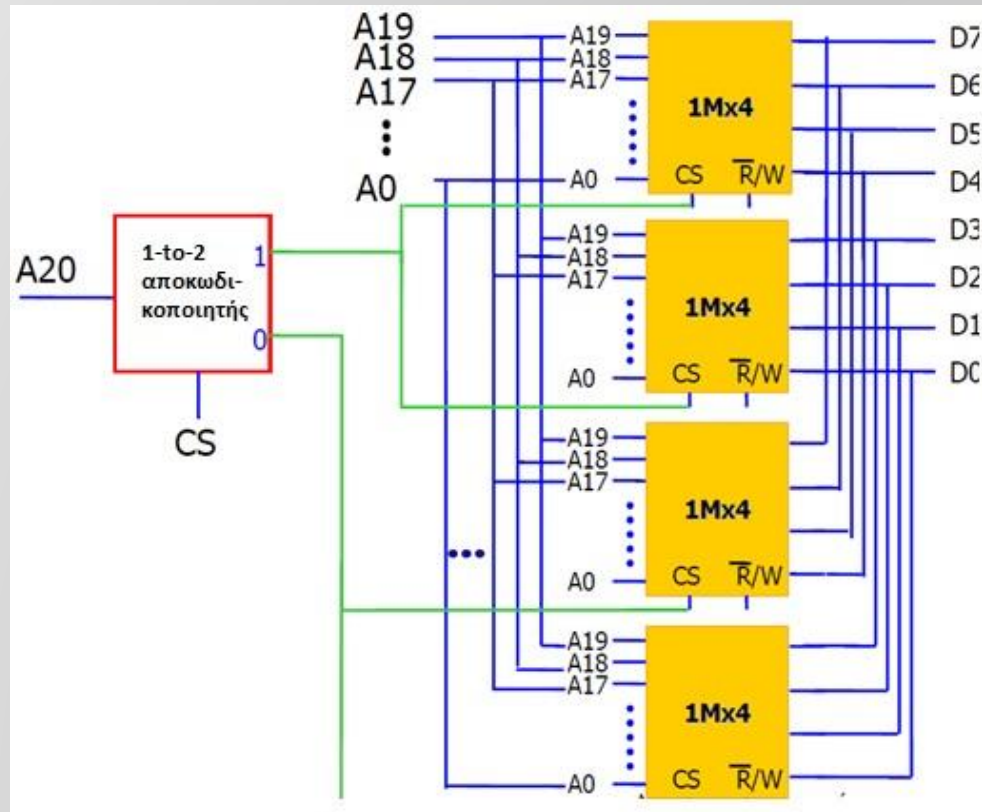
- 2Mx4 από 1Mx4





# Ιεραρχική Κατασκευή Μνήμης (3)

- 2Mx8 από 1Mx4



# Ένα στοιχείο που συνδέεται με τη μνήμη είναι η σειρά αποθήκευσης των bit της λέξης

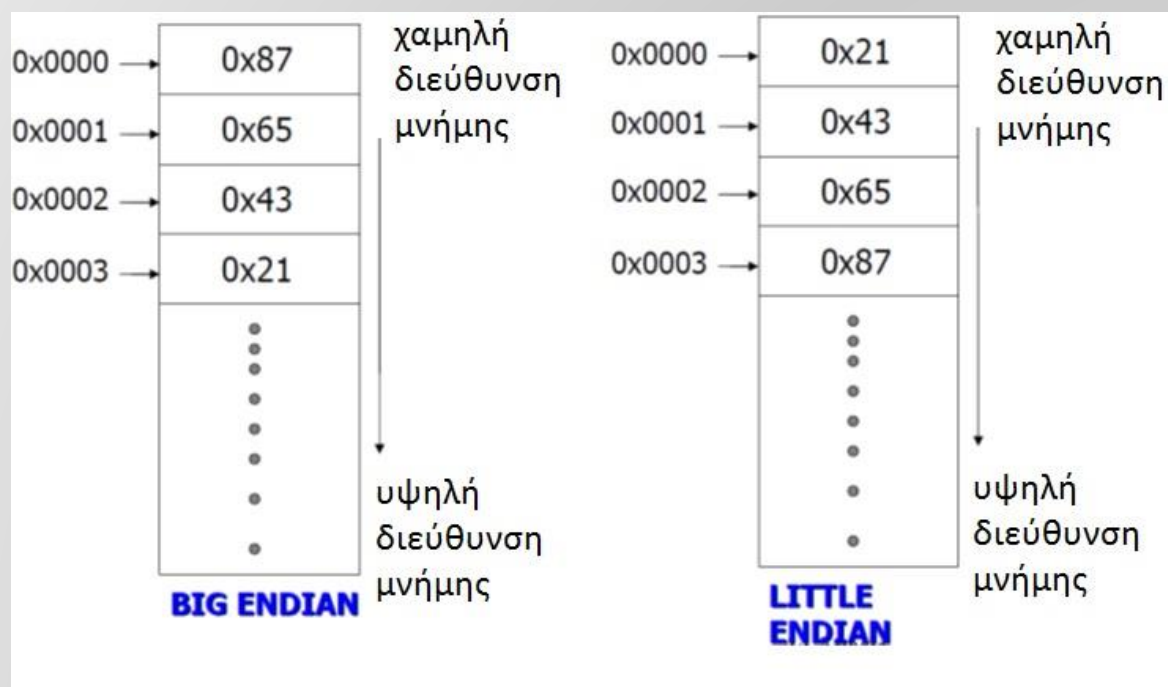
---

- Πως αποθηκεύεται μια λέξη πολλών bytes;
- Big Endian
  - Το πιο σημαντικό Byte τοποθετείται στη χαμηλότερη διεύθυνση μνήμης.
  - e.g. Sun Sparc, PowerPC
- Little Endian
  - Το πιο ασήμαντο Byte τοποθετείται στη χαμηλότερη διεύθυνση μνήμης.
  - e.g. Intel x86



# Παράδειγμα endianness

- Αποθήκευση του 0x87654321 στη διεύθυνση 0.



# Στοιχεία Χρόνου

---

- Χρόνος προσπέλασης ( access time ) της μνήμης είναι το χρονικό διάστημα που χρειάζεται η μνήμη για να επιλεγεί μια λέξη και να αναγνωστεί το αντίστοιχο περιεχόμενο.
- Χρόνος κύκλου ( cycle time ) μιας μνήμης είναι το χρονικό διάστημα που απαιτείται για να ολοκληρωθεί μια λειτουργία γραφής.  
=> απαιτείται συγχρονισμός CPU – RAM.



# Παράδειγμα Συγχρονισμού

---

- Έστω CPU clock 50 Mhz ( περίοδος 20 ns ).
- Έστω RAM cycle time 50 ns.
- Η CPU πρέπει να περιμένει 2 περιόδους μέχρι να έρθουν τα δεδομένα που έχει ζητήσει.



# Κυματομορφές Χρονισμού

Clock: Ρολόι

Memory address: Διεύθυνση μνήμης

Memory enable: Ενεργή μνήμη

Read/Write: Ανάγνωση/Εγγραφή

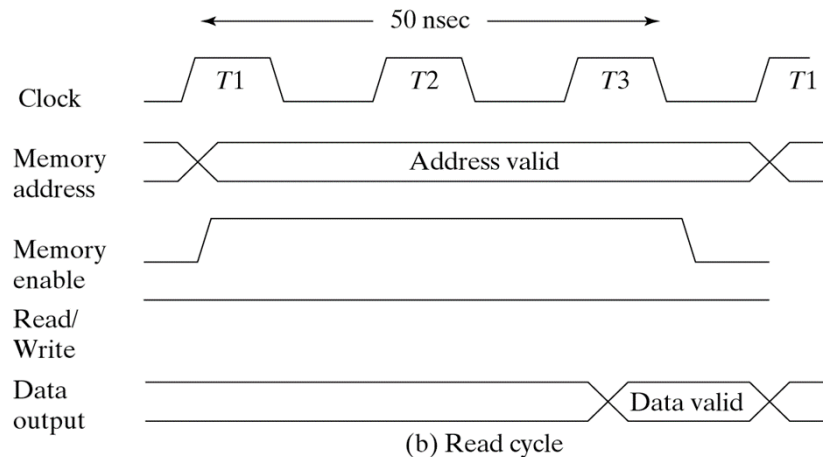
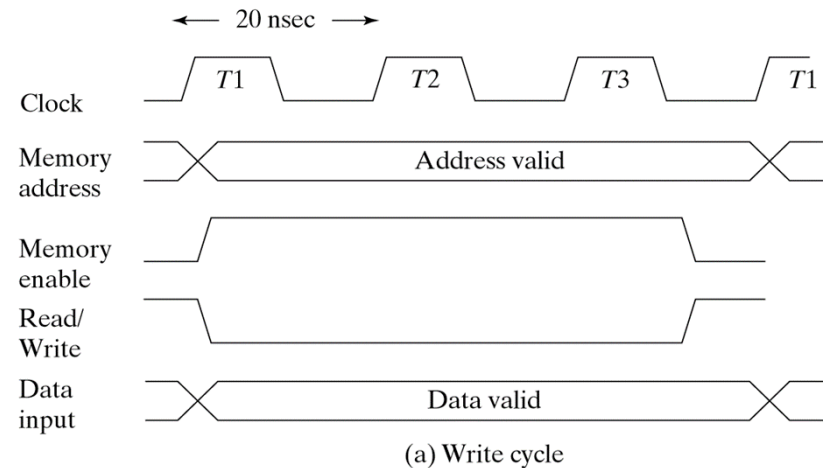
Data input: Εισαγωγή δεδομένων

Write Cycle: Περίοδος εγγραφής

Read Cycle: Περίοδος ανάγνωσης

Address valid: Έγκυρη διεύθυνση

Data valid: Έγκυρα δεδομένα



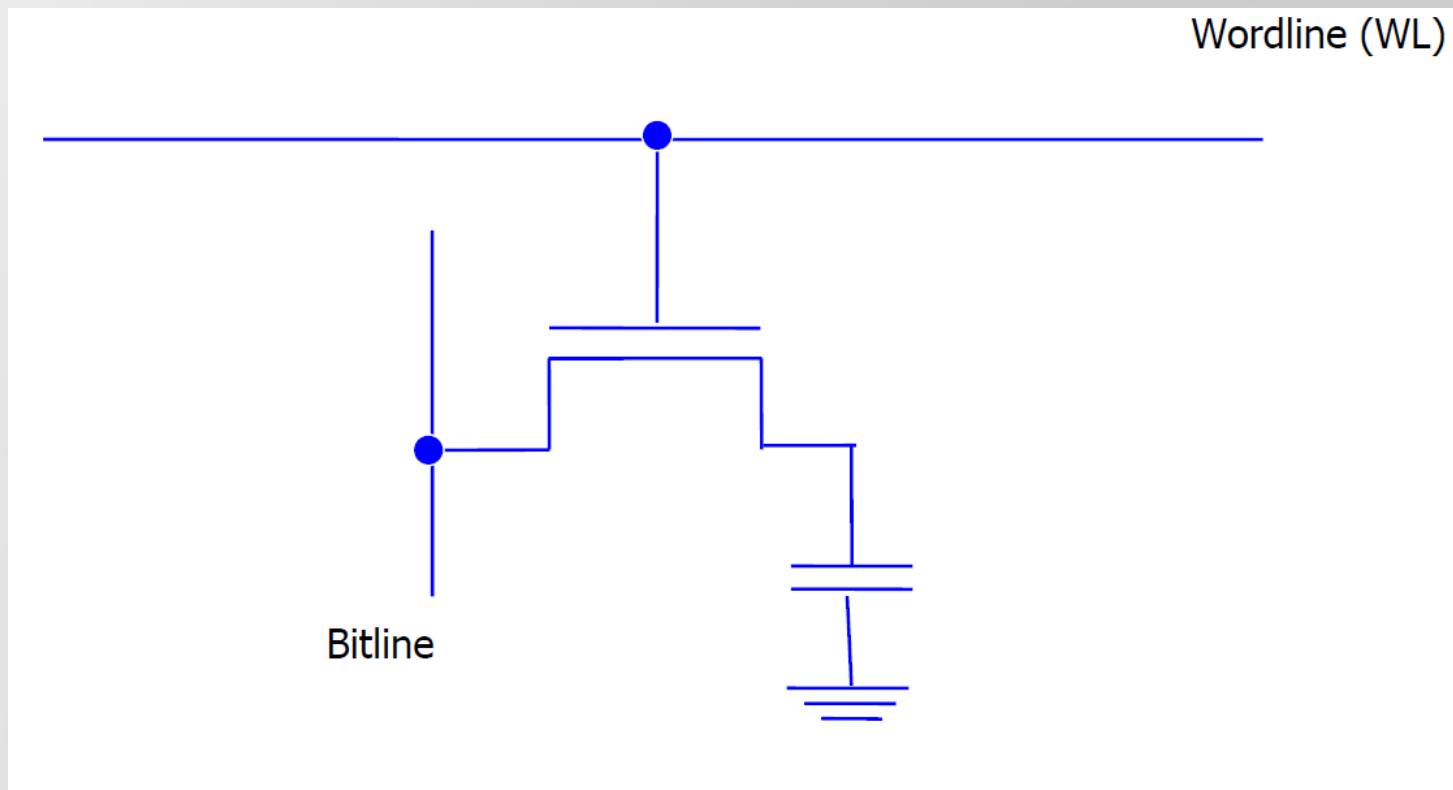
# Υπάρχουν 2 ειδών τεχνολογιών για κατασκευή μνήμης RAM

---

- Δυναμικές RAM ( DRAM ): αποθηκεύει τις δυαδικές πληροφορίες με τη μορφή ηλεκτρικών φορτίων σε πυκνωτές. Απαιτείται ανανέωση φορτίων.
- Στατικές RAM ( SRAM ): αποθηκεύει τις δυαδικές πληροφορίες με τη μορφή καταστάσεων σε μανδαλωτές.



# Η δυναμική RAM κατασκευάζεται από 1 τρανζίστορ

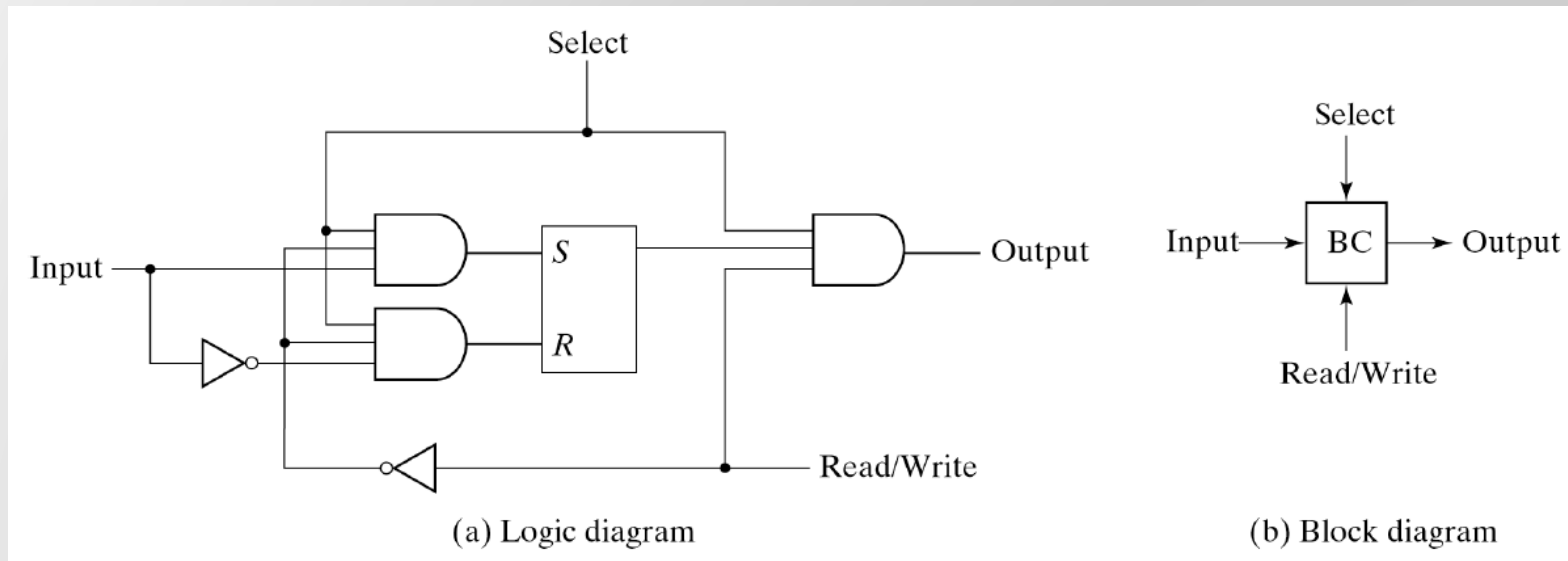




# Σημαντικό συστατικό της μνήμης το κύκλωμα αποκωδικοποίησης

- $m$  λέξεις  $n$  bit  $\Rightarrow m \times n$  δυαδικά

## Κύτταρα αποθήκευσης



Input: Είσοδος

Output: Έξοδος

Read/Write: Ανάγνωση/Εγγραφή

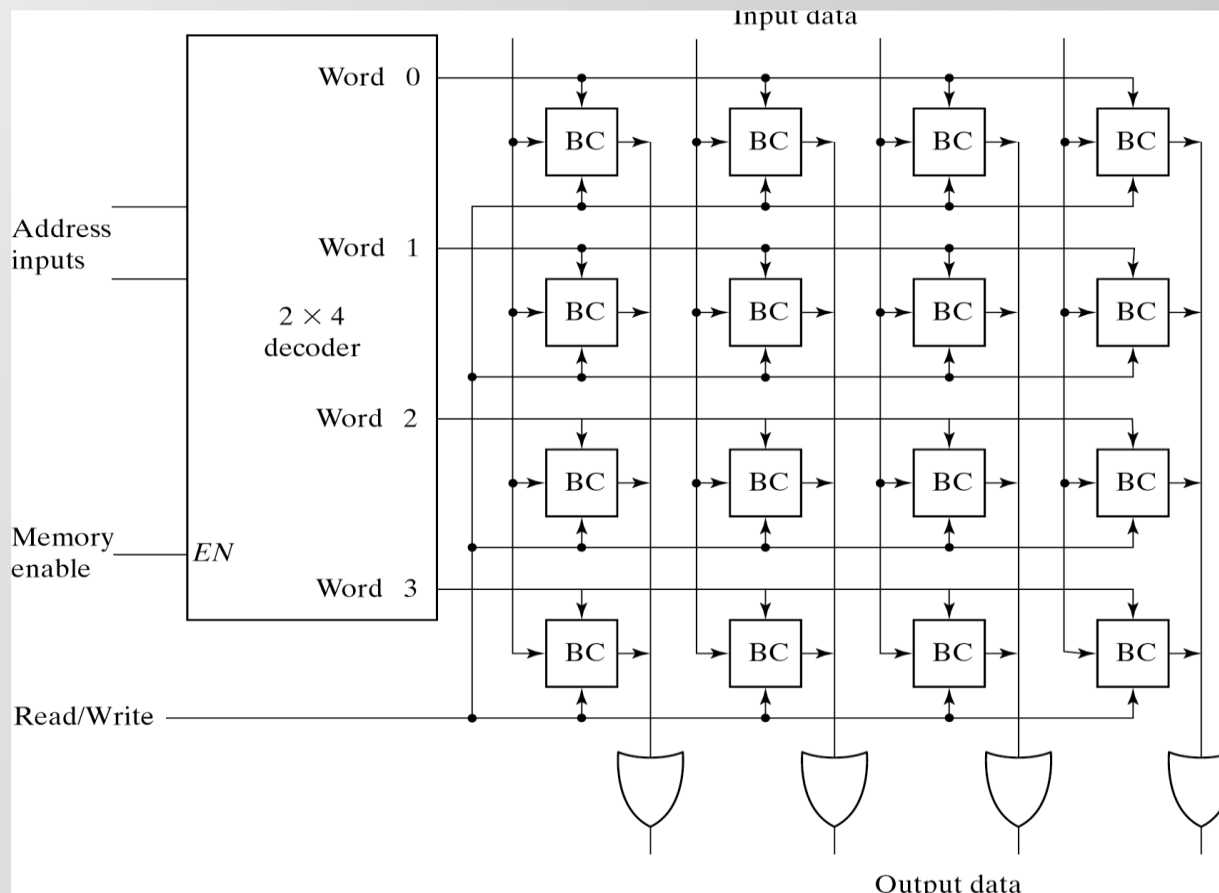
Select: Επιλογή

Logic diagram: Λογικό διάγραμμα

Block diagram: Σχηματικό διάγραμμα



# Διάγραμμα μιας RAM 4x4

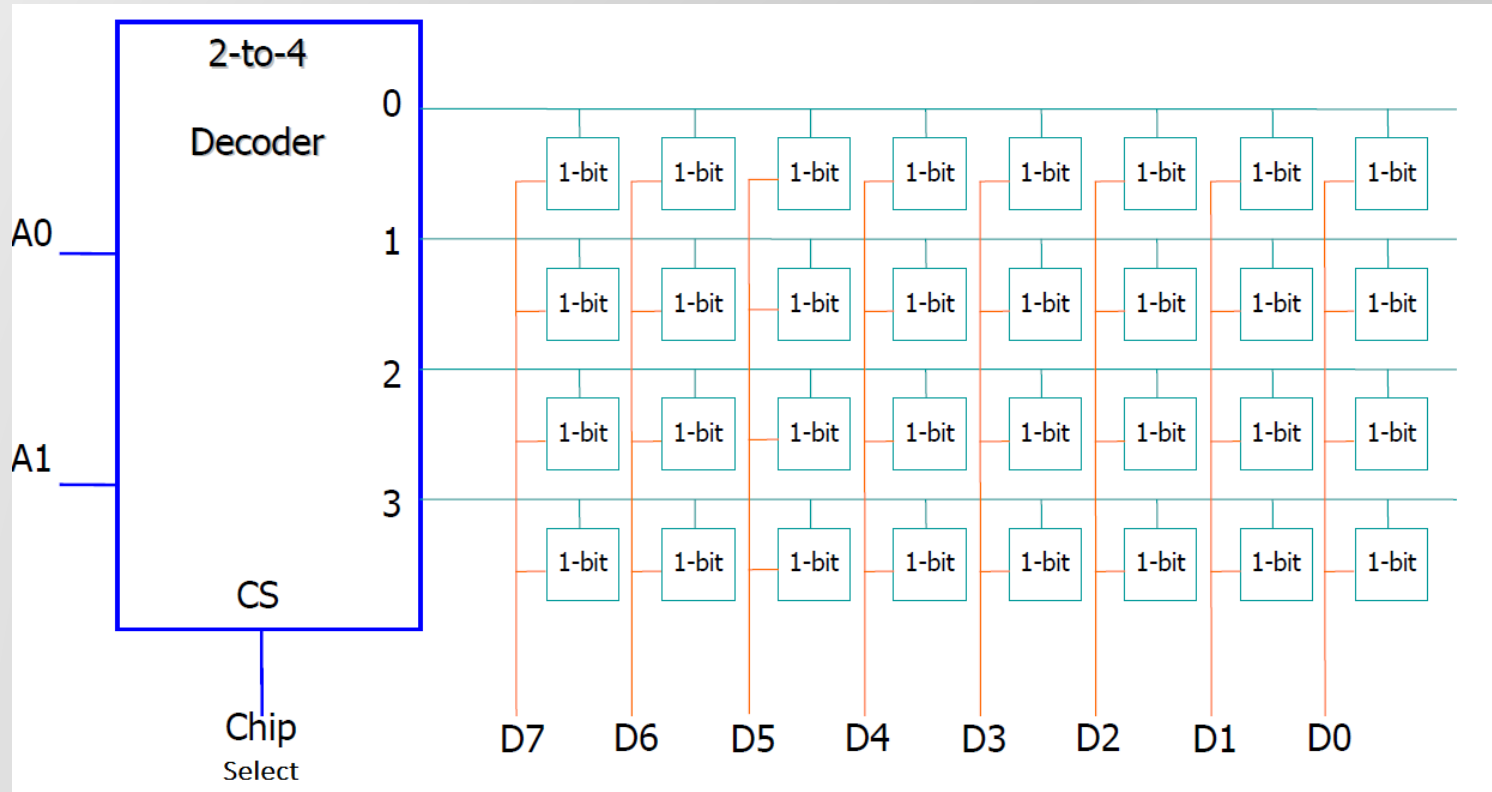


Decoder: Αποκωδικοποιητής



# Διάγραμμα μιας RAM 4x8 (1)

## 4x8 Memory



# Διάγραμμα μιας RAM 4x8 (2)



Access address: Διεύθυνση πρόσβασης



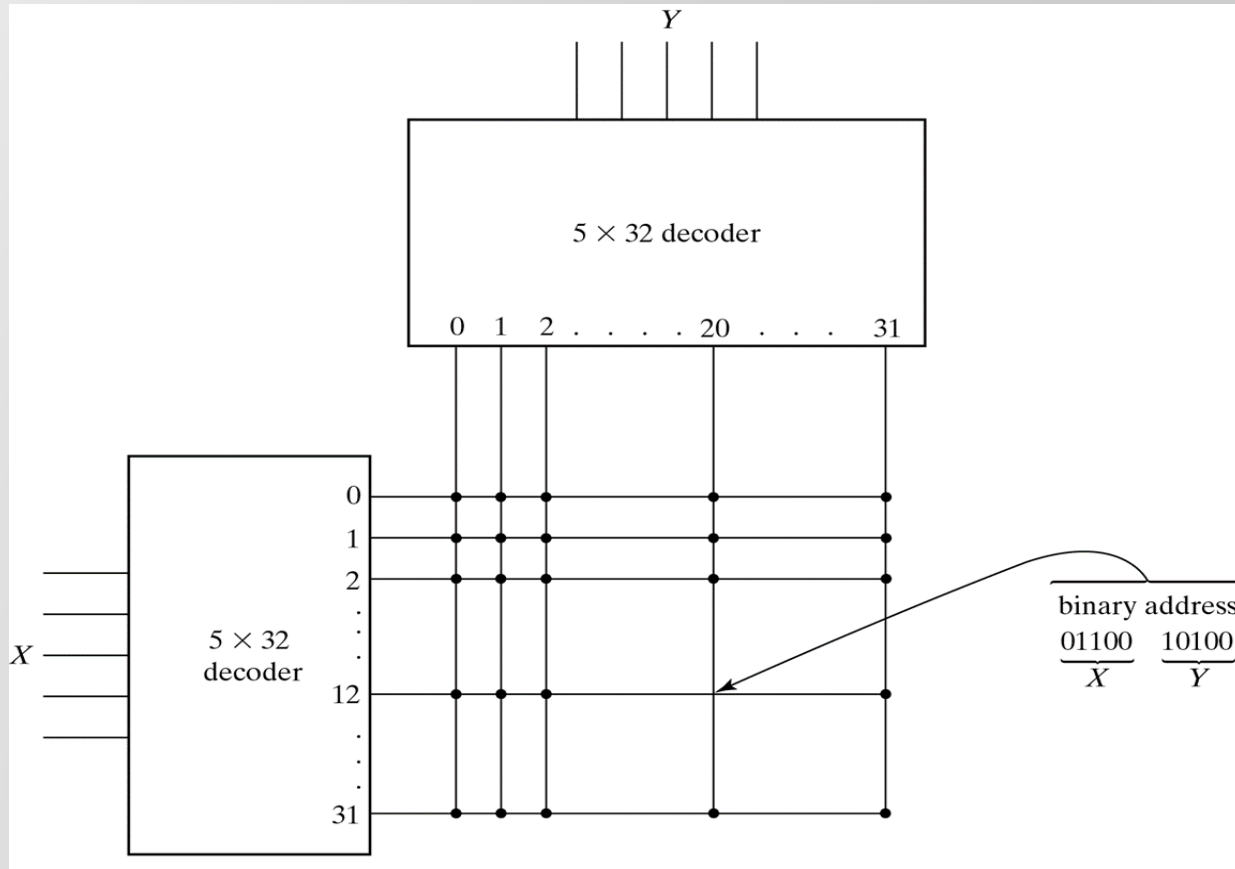
# Μπορούμε να μειώσουμε τις απαιτήσεις σε πύλες

---

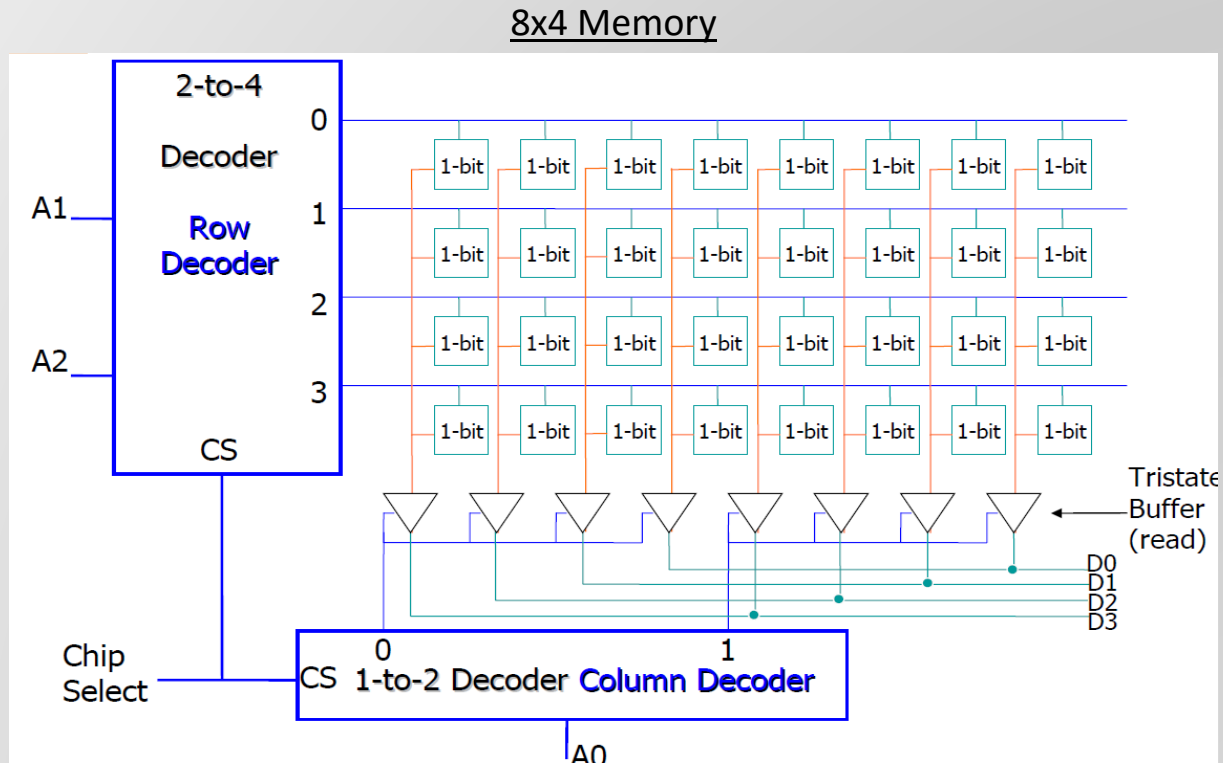
- $k$  εισοδοί απαιτούν  $2^k$  πύλες ΚΑΙ με  $k$  εισόδους ανά πύλη.
- Ο συνολικός αριθμός πυλών μπορεί να μειωθεί αν χρησιμοποιήσουμε δύο αποκωδικοποιητές σε μια δομή δύο διαστάσεων.
- Παράδειγμα: Αντί για 10-σε-1024 (1024 πύλες ΚΑΙ) μπορούμε να χρησιμοποιήσουμε 2 5-σε-32 ( 64 πυλες ΚΑΙ ) ( ... ).



# Η αποκωδικοποίηση δύο διαστάσεων είναι αποτελεσματική (1)



## Η αποκωδικοποίηση δύο διαστάσεων είναι αποτελεσματική (2)



Tristate Buffer: Τριών καταστάσεων buffer

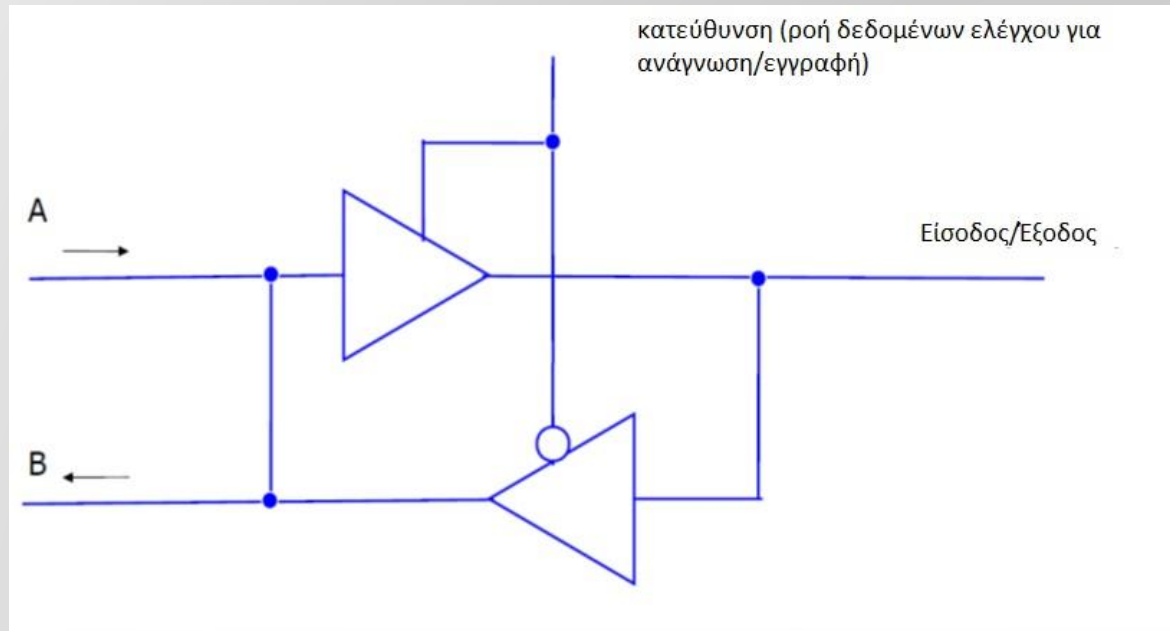
Chip Select: Επιλογή τσιπ

Row Decoder: Αποκωδικοποιητής σειράς

Column Decoder: Αποκωδικοποιητής στήλης



# Ρυθμιστής 3πλής κατάστασης ( Tristate Buffer )





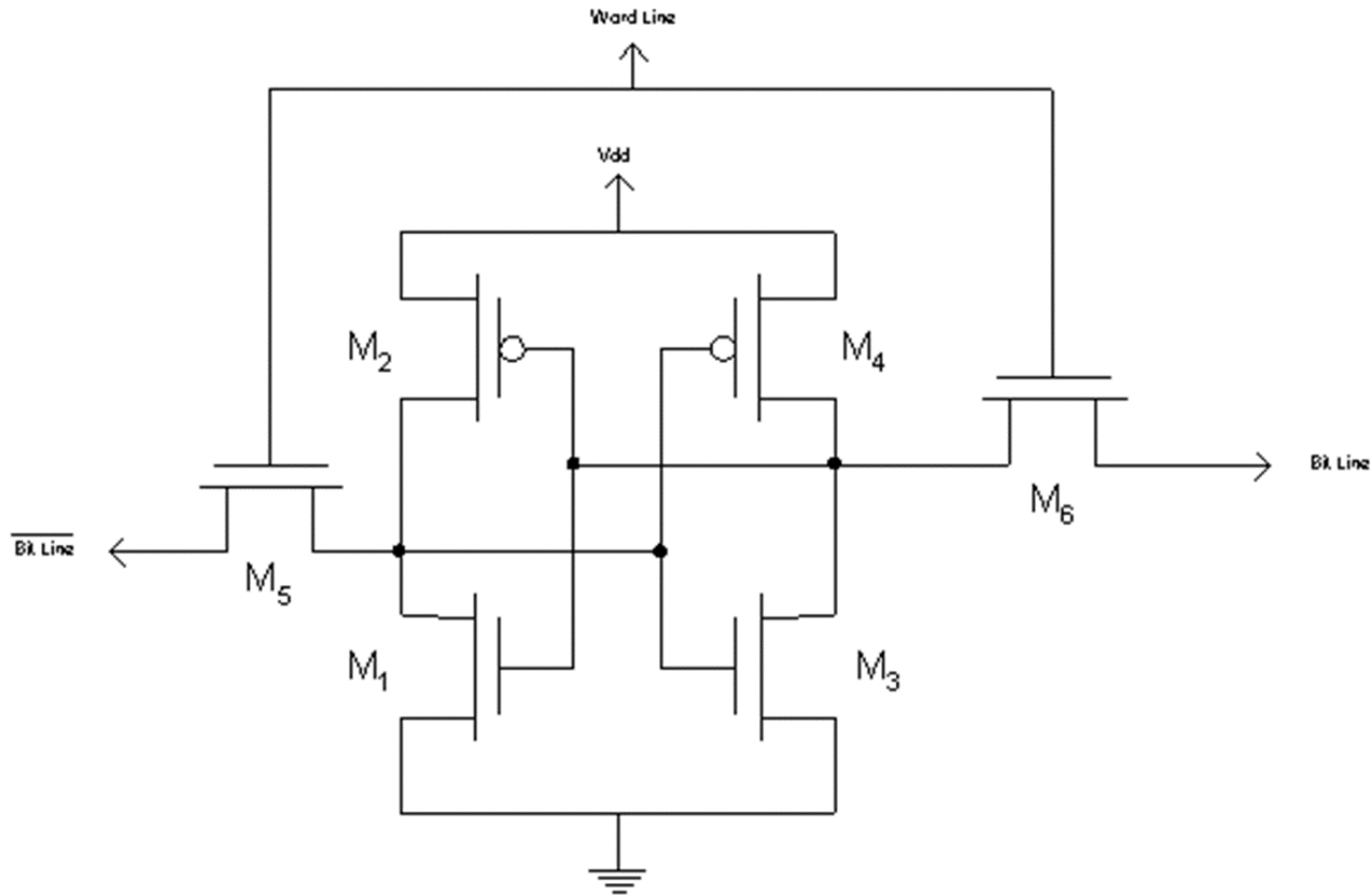
# DRAM vs SRAM

---

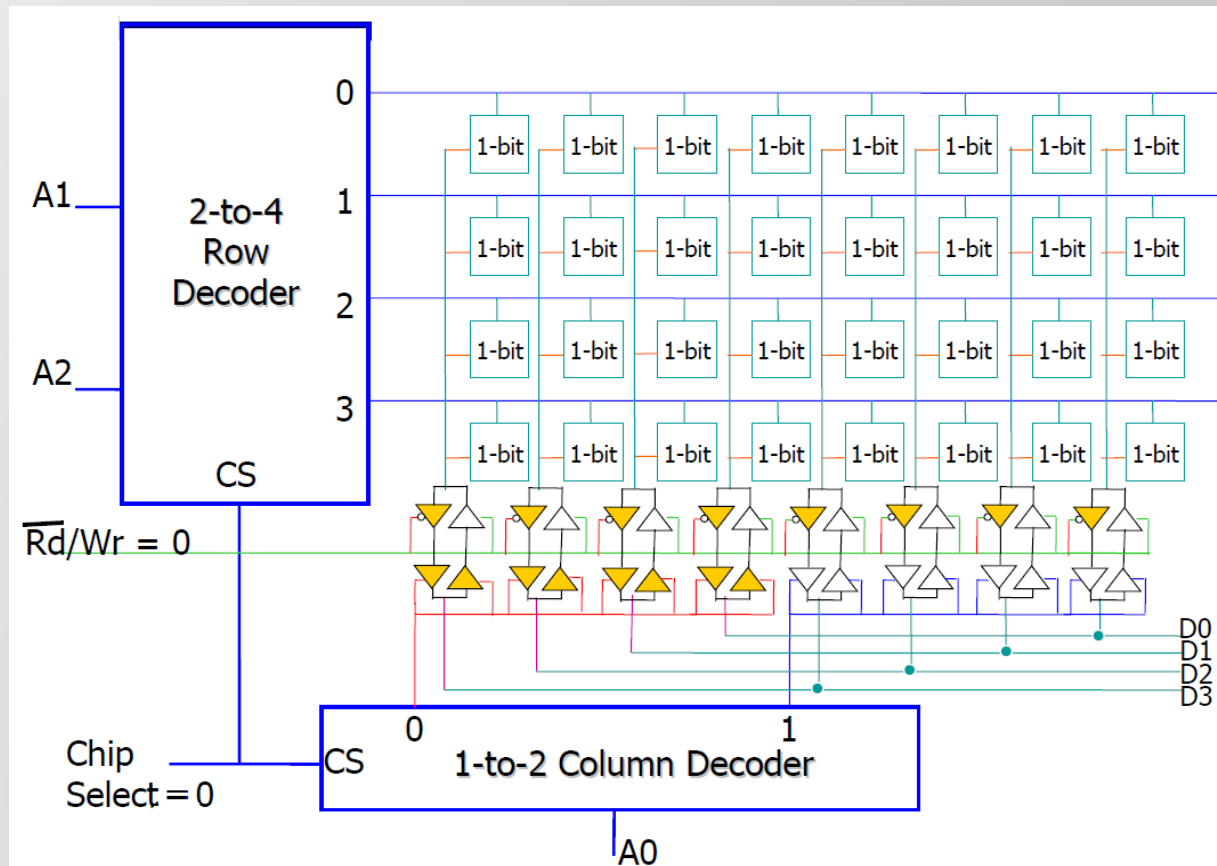
- DRAM 4 φορές την πυκνότητα της SRAM.
- DRAM κόστος αποθήκευσης 4 φορές μικρότερο από SRAM.
- SRAM Μικρότερη κατανάλωση ισχύος.
- SRAM πολύ πιο γρήγορη.



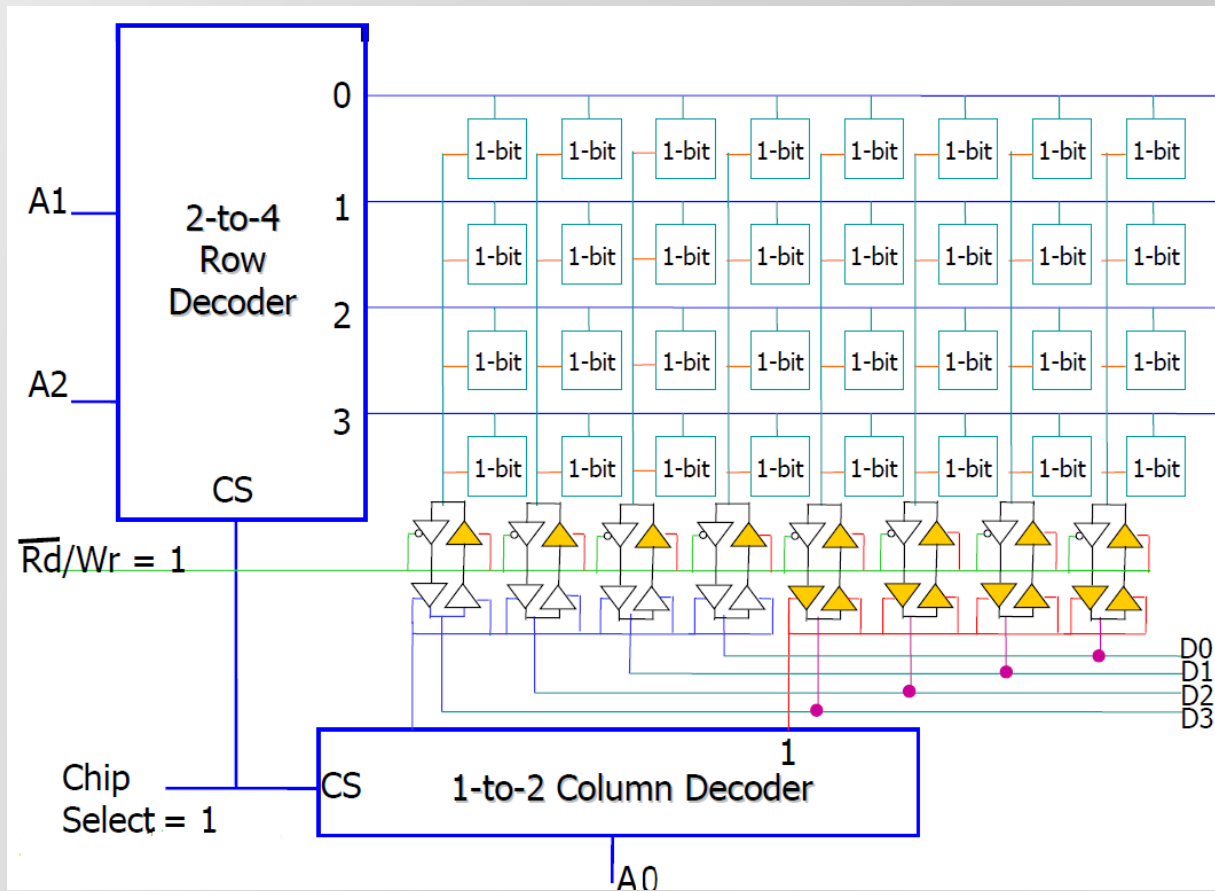
# Η στατική RAM κατασκευάζεται με 6 τρανζίστορ



# Read / Write (1)



# Read / Write (2)



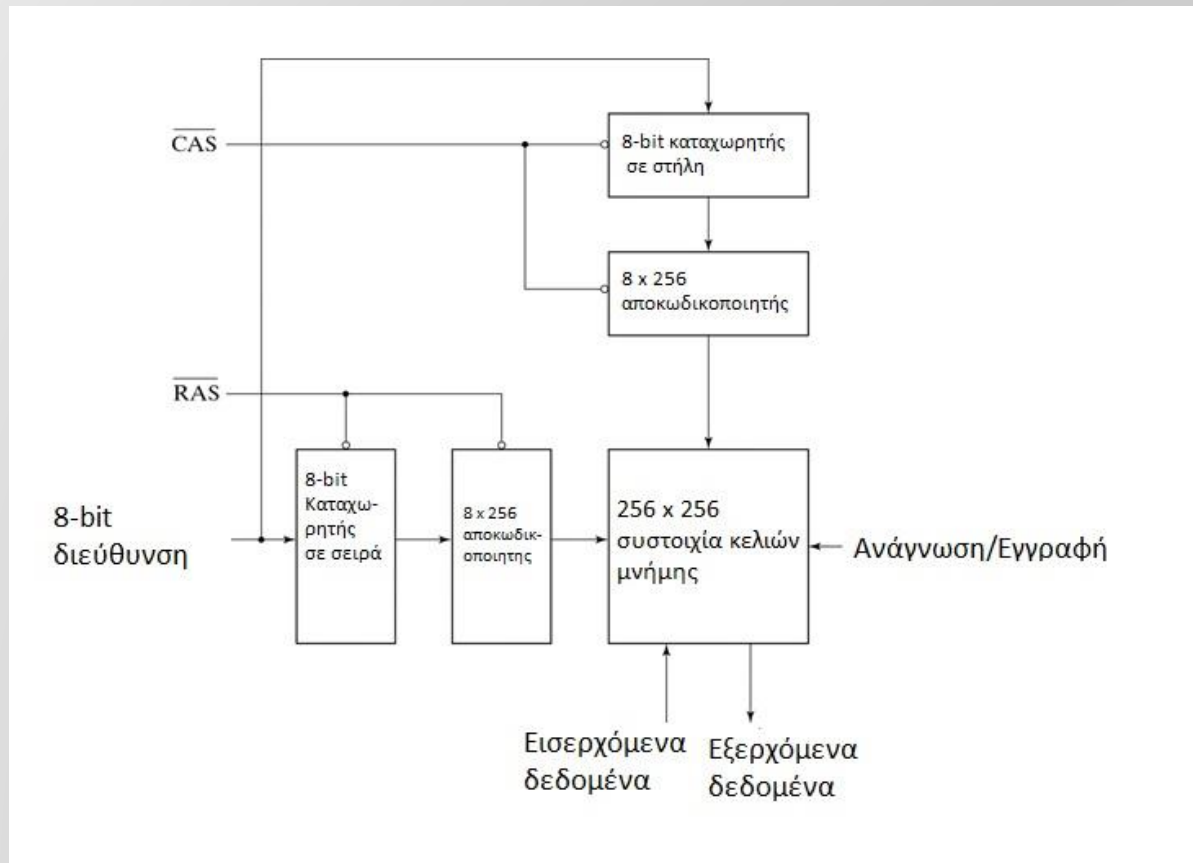
# Επιπρόσθετη Βελτιστοποίηση κατασκευής (1)

---

- Μπορεί να χρησιμοποιηθεί η πολύπλεξη διεύθυνσης, ώστε οι ίδιοι ακροδέκτες εξυπηρετούν μεγαλύτερο αριθμό bit της διεύθυνσης μνήμης.
- Για να επιτευχθεί:
  - Εφαρμόζεται κατάλληλο σήμα.
  - Εφαρμόζεται η RAS ( row address strobe ).
  - Εφαρμόζεται κατάλληλο σήμα.
  - Εφαρμόζεται η CAS ( column address strobe ).



# Επιπρόσθετη Βελτιστοποίηση κατασκευής (2)



# Ανίχνευση & Διόρθωση Λαθών

---

- Η ανίχνευση λαθών επιτυγχάνεται με το parity bit.
- Η διόρθωση λαθών παράγει πολλαπλά bit
  - Το κάθε bit ελέγχου είναι ισοτιμία ομάδας από bit.
  - Αν τα bit δεν είναι σωστά τότε παράγεται ένα πρότυπο ( syndrome ) και μπορεί να χρησιμοποιηθεί για να αναγνωριστεί το λανθασμένο bit.



# Ο πιο κοινός κώδικας διόρθωσης λαθών είναι ο Hamming

---

- $k$  bit ισοτιμίας προστίθενται σε  $n$  bit.
- Οι θέσεις των bit είναι αριθμημένες από 1 έως  $n+k$ .
- Οι θέσεις που έχουν αριθμό ίσο με δύναμη του 2. χρησιμοποιούνται για τα bit ισοτιμίας.
- Οι υπόλοιπες θέσεις είναι ο αριθμός.





# Παράδειγμα Κώδικα Hamming (1/3)

- Έστω η λέξη 11000100.
- Απαιτούνται 4 bit (επόμενος πίνακας).

Θέση bit	1	2	3	4	5	6	7	8	9	10	11	12
	P1	P2	1	P4	1	0	0	P8	0	1	0	0

- $P1 = \text{XOR}(3, 5, 7, 9, 11) = 0$
- $P2 = \text{XOR}(3, 6, 7, 10, 11) = 0$
- $P4 = \text{XOR}(5, 6, 7, 12) = 1$
- $P8 = \text{XOR}(9, 10, 11, 12) = 1$

=>η λέξη που μετατίθεται είναι: 001110010100



# Παράδειγμα Κώδικα Hamming (2/3)

- Ο παραλήπτης ελέγχει τις συγκεκριμένες θέσεις:
- $C1 = \text{XOR}( 1, 3, 5, 7, 9, 11 ) = 0$
- $C2 = \text{XOR}( 2, 3, 6, 7, 10, 11 ) = 0$
- $C4 = \text{XOR}( 4, 5, 6, 7, 12 ) = 1$
- $C8 = \text{XOR}( 8, 9, 10, 11, 12 ) = 1$
- Αν  $C_i = 0$  τότε OK.
- Αν  $C_i = 1$  τότε πρόβλημα.
- Ο κωδικός  $C = C8 C4 C2 C1$  δείχνει τη θέση του λανθασμένου bit.



# Παράδειγμα Κώδικα Hamming (3/3)

- a) Αρχική                    0 0 1 1 1 0 0 1 0 1 0 0
- b) Λαθος στο bit 1    **1** 0 1 1 1 0 0 1 0 1 0 0
- c) Λάθος στο bit 5    0 0 1 1 **0** 0 0 1 0 1 0 0

Υπολογίζοντας τα C όπως πριν:

- a) C = 0 0 0 0
- b) C = 0 0 0 1
- c) C = 0 1 0 1



# Εύρος του αριθμού των bit δεδομένων για k bit ελέγχου

---

k bit ελέγχου	n bit δεδομένων
3	2-4
4	5-11
5	12-26
6	27-57
7	58-120



# Διόρθωση λάθους 1 bit, ανίχνευση λάθους 2 bit

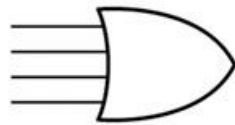
---

- Προσθέτουμε ένα ακόμη bit ισοτιμίας
- Έχουμε 4 καταστάσεις:
  - $C = 0, P = 0$  κανένα λάθος.
  - $C \neq 0, P = 1$  ένα λάθος που διορθώνεται.
  - $C \neq 0, P = 0$  λάθος σε 2 bit ( δε διορθώνεται ).
  - $C = 0, P = 1$  λάθος στο bit P13.

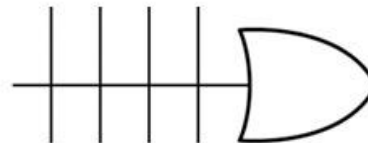


# Ένα νέο σύμβολο...παράταξης

- Στις επόμενες διαφάνειες αντί για πολλαπλές γραμμές εισόδου σε μια πύλη θα σχεδιάζουμε τη πύλη με μια μοναδική γραμμή εισόδου.



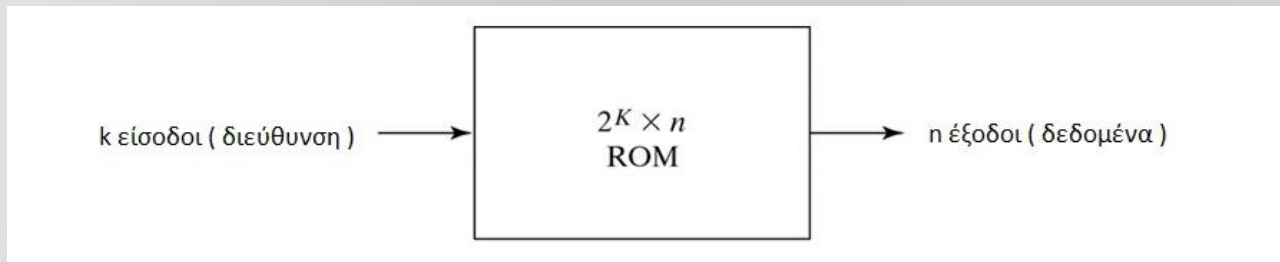
(α) Συμβατικό σύμβολο



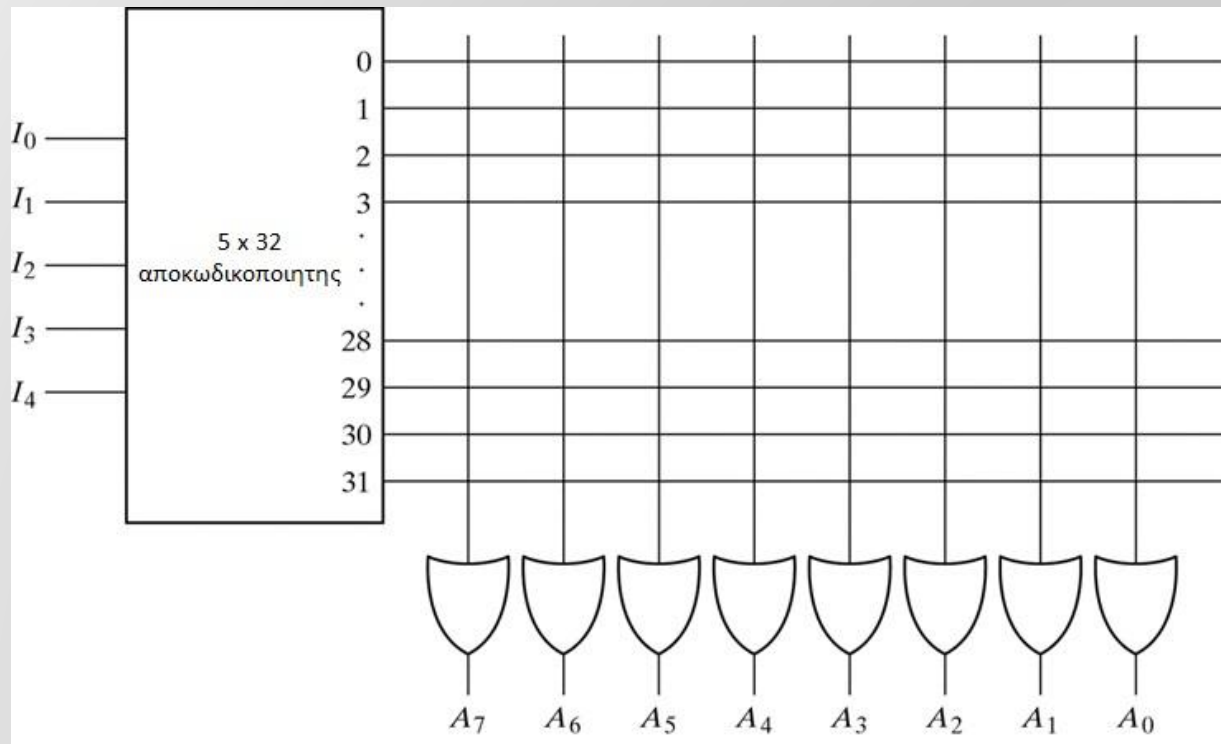
(β) Λογικό σειριακό σύμβολο

# Μνήμη ανάγνωσης μόνο

- Οι δυαδικές πληροφορίες ορίζονται από το σχεδιαστή και ενσωματώνονται στο ολοκληρωμένο κύκλωμα.
- Οι δυαδικές πληροφορίες παραμένουν ακόμη και όταν διακοπεί η παροχή ισχύος.

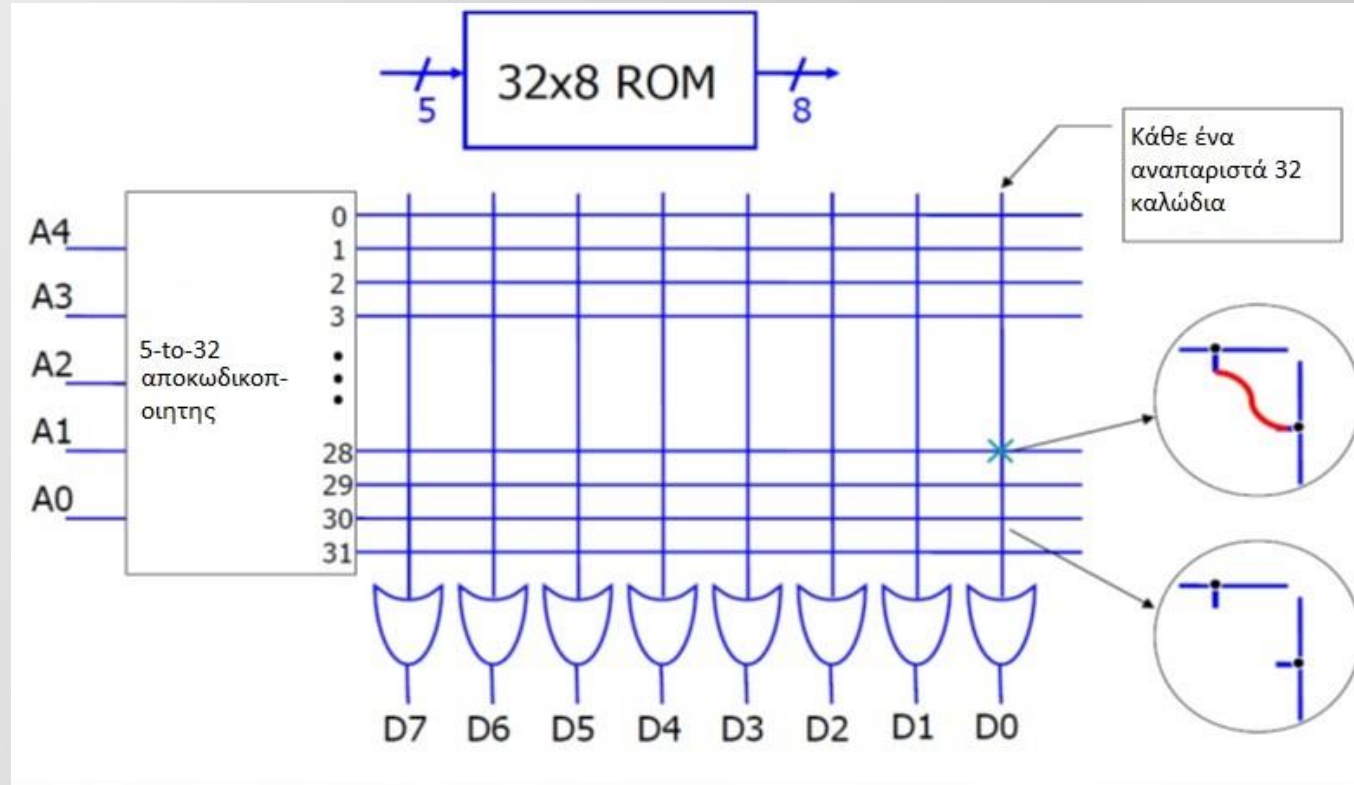


# Λογικό διάγραμμα του εσωτερικού κυκλώματος μιας ROM 32x8



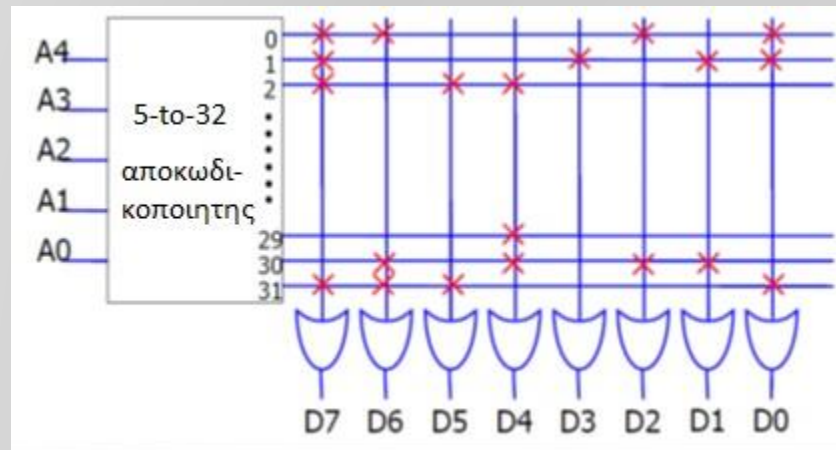


# Οι διασυνδέσεις καίγονται ( καταστρέφονται ) στον προγραμματισμό.



# Προγραμματισμός ROM

A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	1	0	0	0	1	0	1
0	0	0	0	1	1	0	0	0	1	0	1	1
0	0	0	1	0	1	0	1	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
1	1	1	0	1	0	0	0	1	0	0	0	0
1	1	1	1	0	0	1	0	1	0	1	1	0
1	1	1	1	1	1	1	1	0	0	0	0	1



## 2 ερμηνείες της ROM

---

- Η λειτουργία της ROM είναι αυτή μιας μονάδας μνήμης.
- Η ROM υλοποιεί ένα συνδυαστικό κύκλωμα ( κάθε ακροδέκτης εξόδου θεωρείται ότι υλοποιεί μια συνάρτηση Boole ).
- π.χ. Στο προηγούμενο σχήμα  
 $A_7( I_4, I_3, I_2, I_1, I_0 ) = \Sigma( 0, 2, 3, \dots, 29 )$ .



# Παράδειγμα 2:ROM για υπολογισμό τετραγώνου

X	$F(X) = X^2$
0	0
1	1
2	4
3	9
4	16
5	25
6	36
7	49

X	$F(X) = X^2$
000	000000
001	000001
010	000100
011	001001
100	010000
101	011001
110	100100
111	110001



## Παράδειγμα: Σχεδιάστε συνδυαστικό κύκλωμα που υπολογίζει το τετράγωνο αριθμού εισόδου

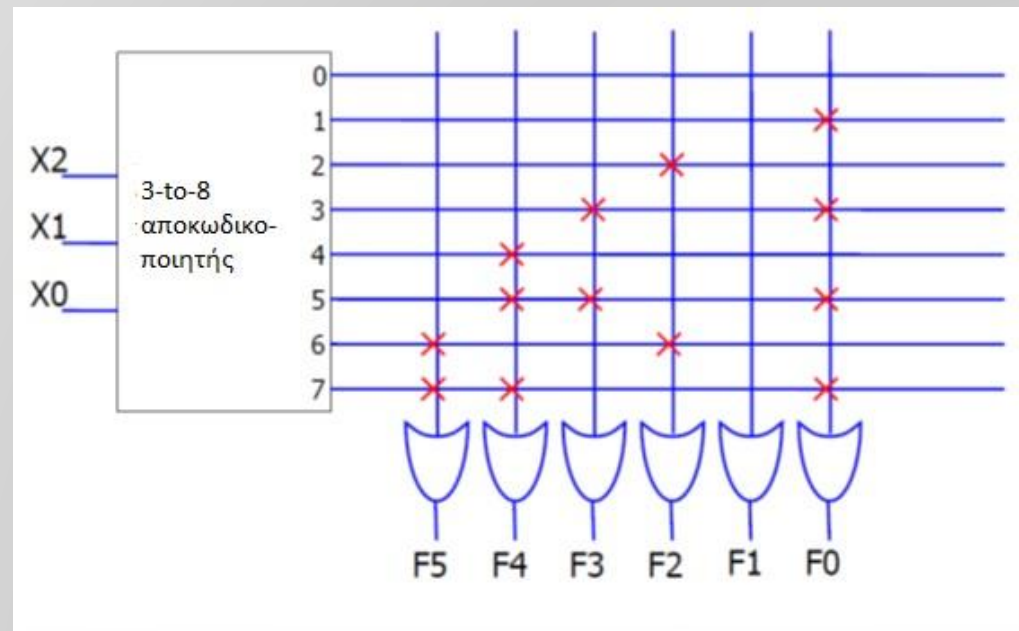
- Υπολογίζουμε τον πίνακα αληθείας.

Είσοδος			Έξοδοι						Ισοδύναμος Δεκαδικός
$A_1$	$A_1$	$A_0$	$B_5$	$B_4$	$B_3$	$B_2$	$B_1$	$B_0$	
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	0	0	0	0	0	16
1	0	1	0	1	0	0	0	1	25
1	1	0	1	0	1	1	0	0	36
1	1	1	1	1	0	0	0	1	49



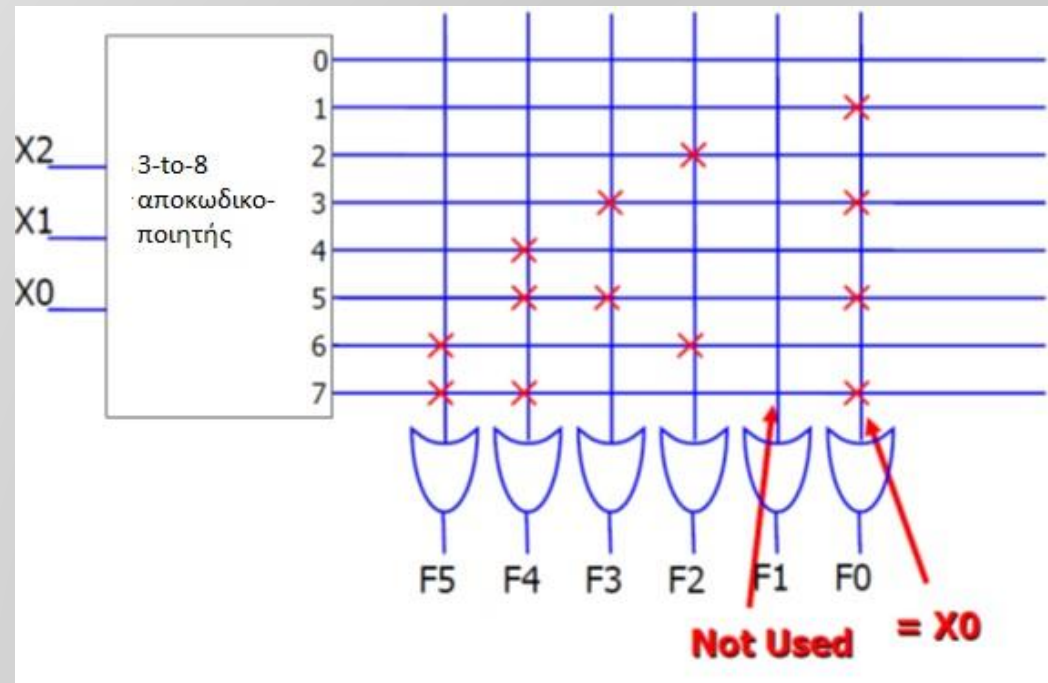
# Προγραμματισμός ROM του παραδείγματος (1)

X	F(X) = X <sup>2</sup>
000	000000
001	000001
010	000100
011	001001
100	010000
101	011001
110	100100
111	110001



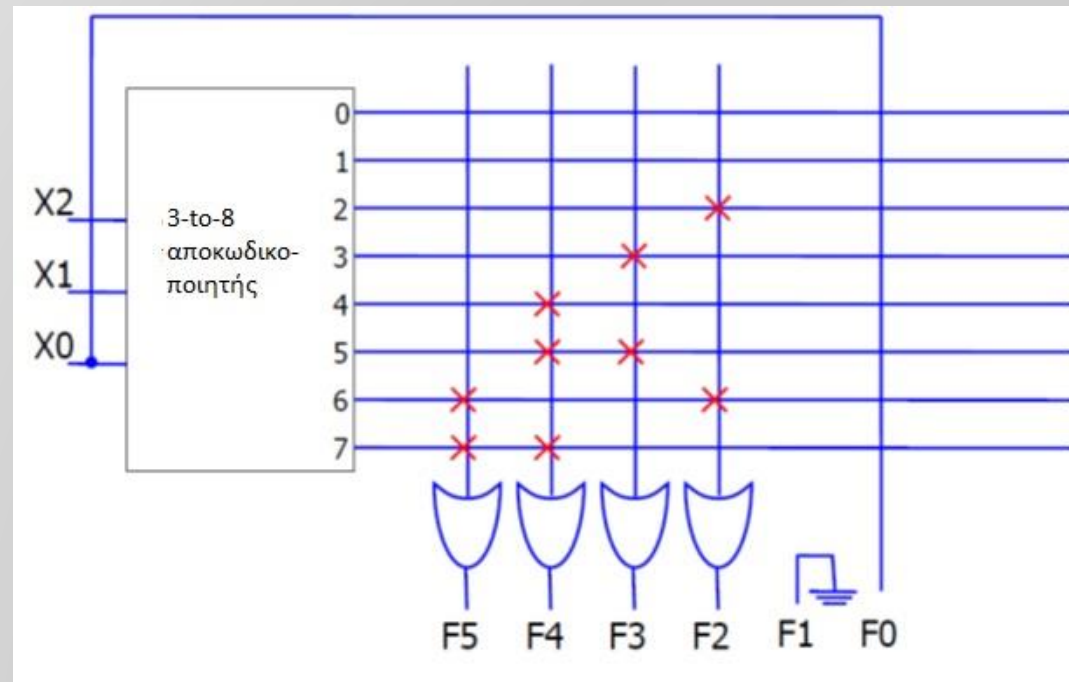
# Προγραμματισμός ROM του παραδείγματος (2)

X	F(X) = X <sup>2</sup>
000	000000
001	000001
010	000100
011	001001
100	010000
101	011001
110	100100
111	110001



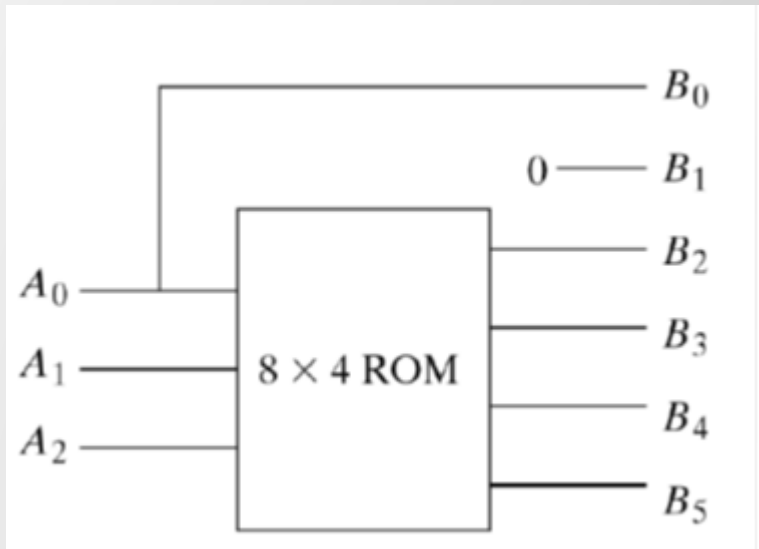
# Προγραμματισμός ROM του παραδείγματος (3)

X	$F(X) = X^2$
000	000000
001	000001
010	000100
011	001001
100	010000
101	011001
110	100100
111	110001





# Υλοποίηση του κυκλώματος



(α) σχηματικό διάγραμμα

$A_2$	$A_1$	$A_0$	$B_5$	$B_4$	$B_3$	$B_2$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

(β) πίνακας αληθείας  
ROM

# Υπάρχουν 4 τύποι ROM

---

- **ROM** ( mask programming από εταιρία, ακριβό ).
- **Programmable ROM** ( προγραμματισμός από χρήστη ).
- **Erasable PROM** ( διαγράψιμη PROM κάτω από υπεριώδες φως ).
- **Electrically EPROM** ( ηλεκτρικά διαγράψιμη PROM, προγραμματισμός χωρίς να αφαιρεθεί η ROM ).



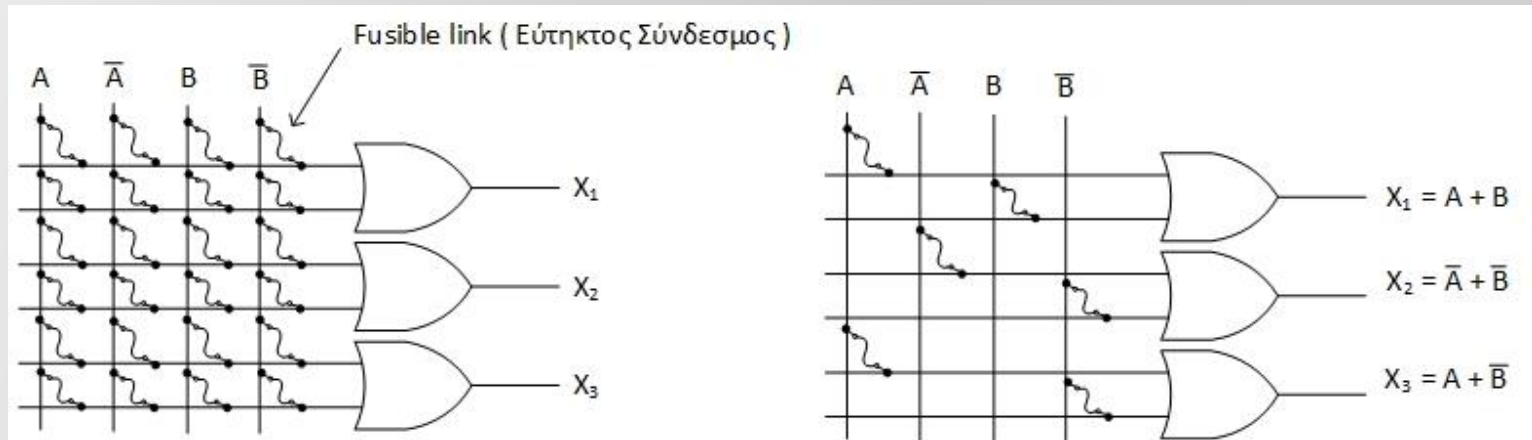
# Συσκευές Προγραμματίσιμης Λογικής

---

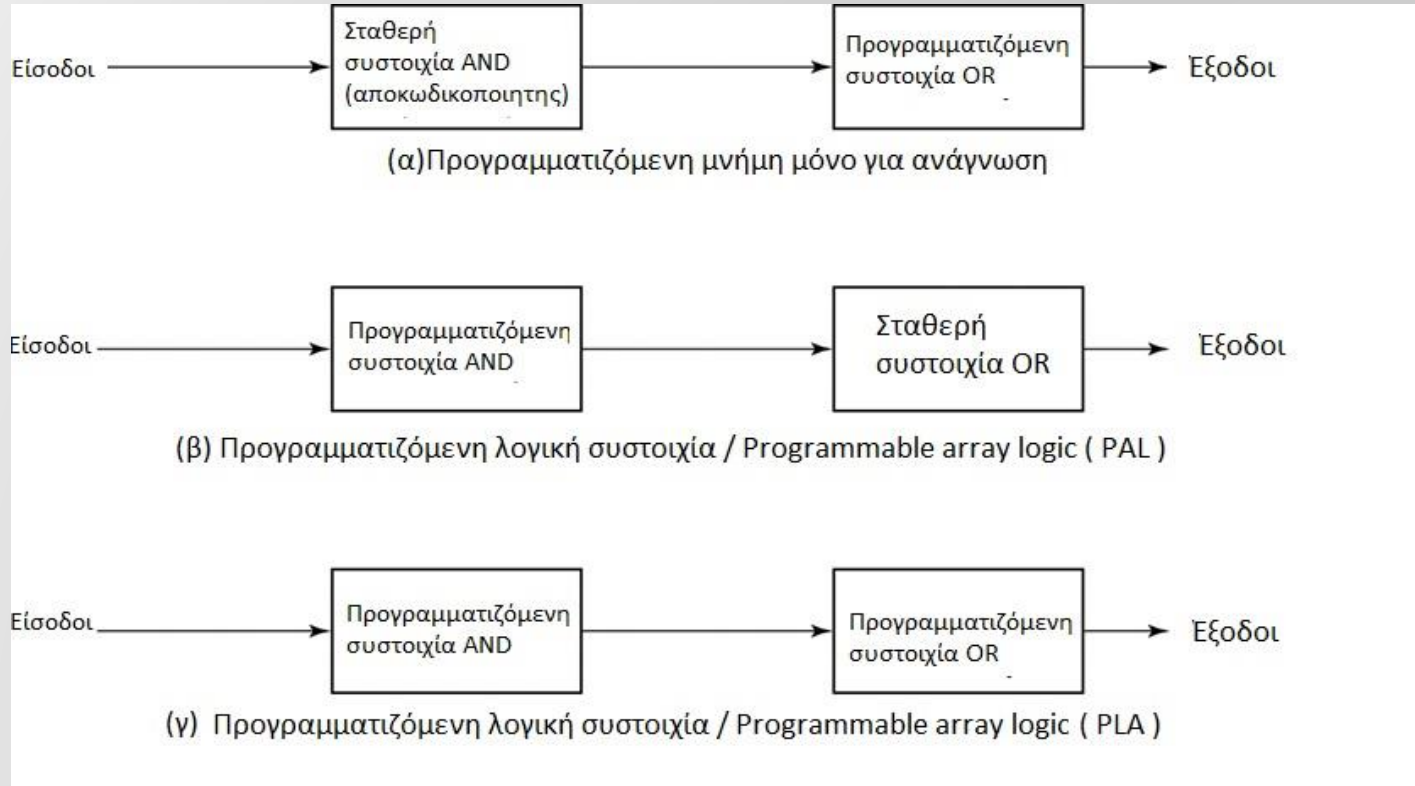
- Η PROM είναι μια συσκευή προγραμματίσιμης συνδυαστικής λογικής ( Programmable Logic Device – PLD ).
- Τα PLD χρησιμοποιούνται ευρέως αντικαθιστώντας τα SSI, MSI.
- Αποτελούνται από πίνακες με πύλες AND και OR που προγραμματίζονται για να υλοποιήσουν λογικές συναρτήσεις.
- Υπάρχουν τρεις ( 3 ) κύριοι τύποι συνδυαστικών PLD.



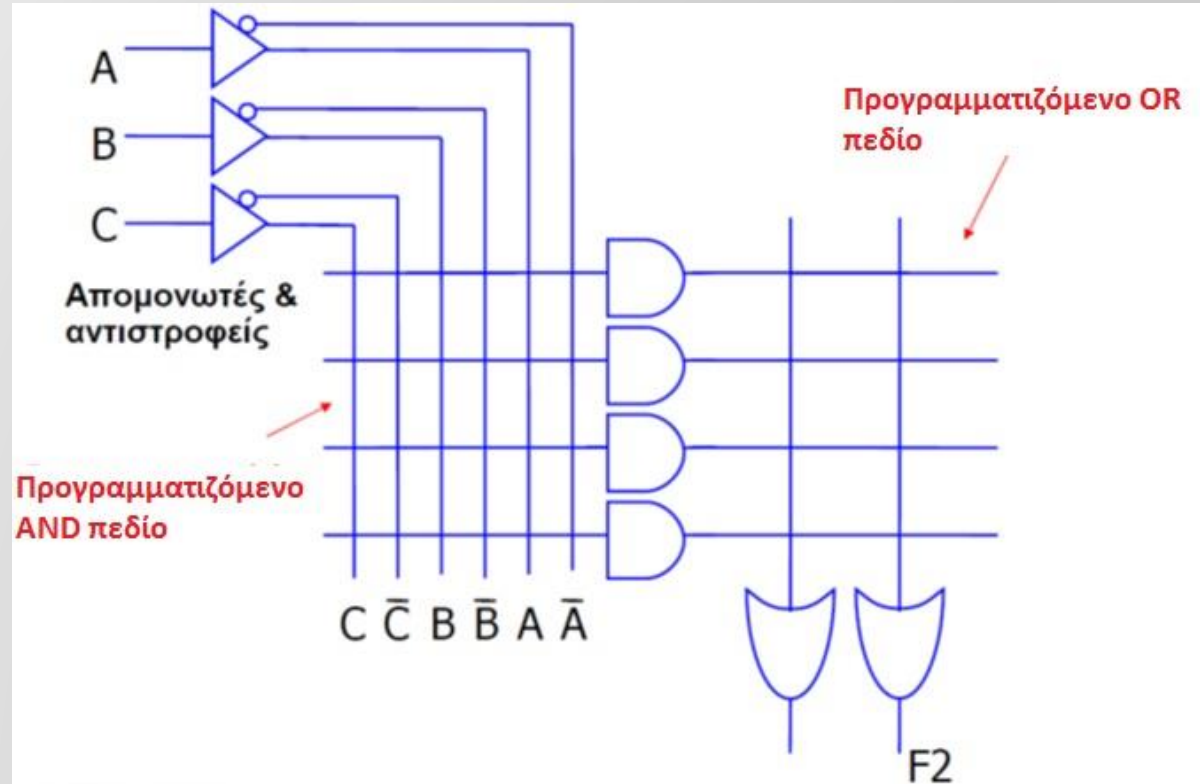
# Προγραμματιζόμενοι πίνακες ( πύλες OR/AND )



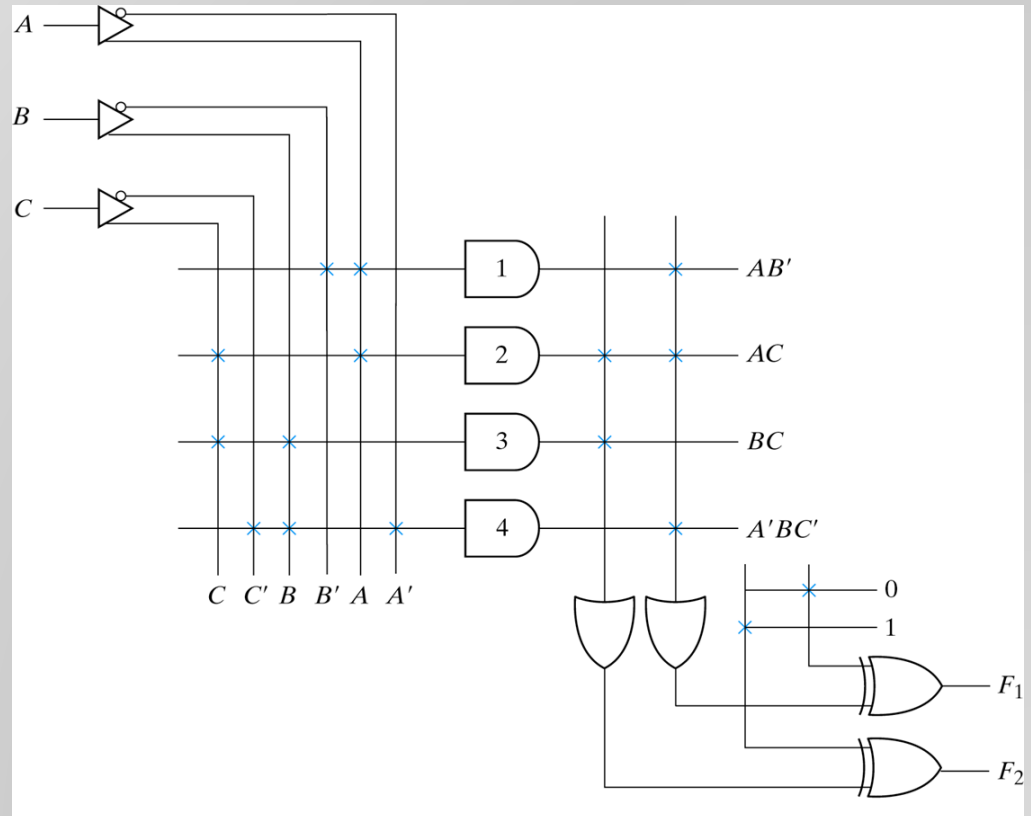
# Υπάρχουν 3 βασικοί τύποι PLD



# Programmable Logic Array ( PLA ) – Προγραμματιζόμενη Λογική Συστοιχία



# Η PLA δεν παράγει όλους τους ελαχιστόρους



- $F_1 = AB' + AC + A'BC'$
- $F_2 = (AC + BC)'$

## Ο χάρτης προγραμματισμού μπορεί να δοθεί και σε μορφή πίνακα

- Πίνακας προγραμματισμού PLA για το προηγούμενο παράδειγμα.

		Είσοδοι			Έξοδοι	
		A	B	C	(T) $F_1$	(C) $F_2$
Γινόμενο						
AB'	1	1	0	-	1	-
AC	2	1	-	1	1	1
BC	3	-	1	1	-	1
A'BC'	4	0	1	0	1	-

- Στις εισόδους αν είναι 1 τότε έχουμε κανονική μορφή, αν είναι 0 τότε συμπληρωμένη μορφή αν – τότε δεν υπάρχει σύνδεση.





# Το μέγεθος του PLA

---

- Το μέγεθος μιας PLA προσδιορίζεται από:
  - Αριθμό εισόδων.
  - Αριθμό γινομένων.
  - Αριθμό εξόδων.
- Το PLA κατασκευάζεται:
  - Από τον κατασκευαστή με μάσκες.
  - Από το χρήστη με FPLA ( field PLA ).
- Όταν υλοποιούμε ένα κύκλωμα με PLA πρέπει να ελαχιστοποιούμε τον αριθμό των διαφορετικών όρων.



# Παράδειγμα Υλοποίησης με PLA

- Υλοποιήστε τις παρακάτω λογικές συναρτήσεις χρησιμοποιώντας PLA.
  - $F1( A, B, C ) = \Sigma( 0, 1, 2, 4 )$
  - $F2( A, B, C ) = \Sigma( 0, 5, 6, 7 )$
- Απλοποιούμε τις συναρτήσεις σε μορφή αθροίσματος γινομένων....



# Λύση του προηγούμενου προβλήματος (1)

		BC		B	
		00	01	11	10
A	0	1	1	0	1
A	1	1	0	0	0
		C			

		BC		B	
		00	01	11	10
A	0	1	0	0	0
A	1	0	1	1	1
		C			

$$F_1 = A'B' + A'C' + B'C'$$

$$F_1 = (AB + AC + BC)'$$

$$F_2 = AB + AC + A'B'C'$$

$$F_2 = (A'C + A'B + AB'C')'$$

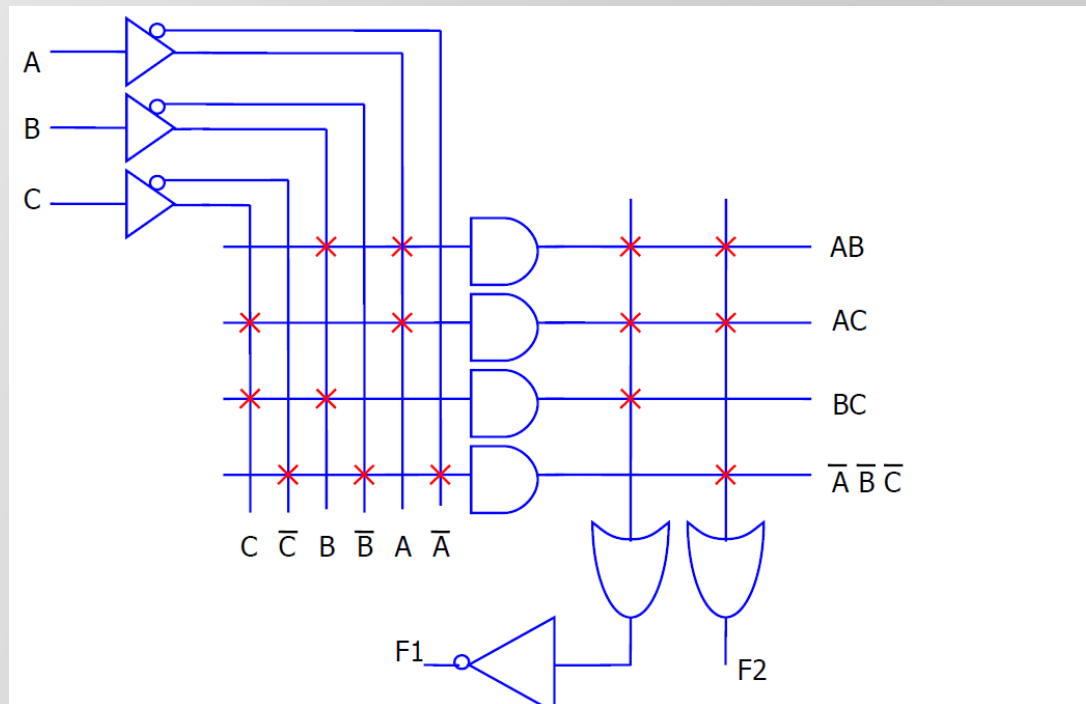
		Inputs			Outputs (C) (T)	
	Product term	A	B	C	F <sub>1</sub>	F <sub>2</sub>
AB	1	1	1	-	1	1
AC	2	1	-	1	1	1
BC	3	-	1	1	1	-
A'B'C'	4	0	0	0	-	1



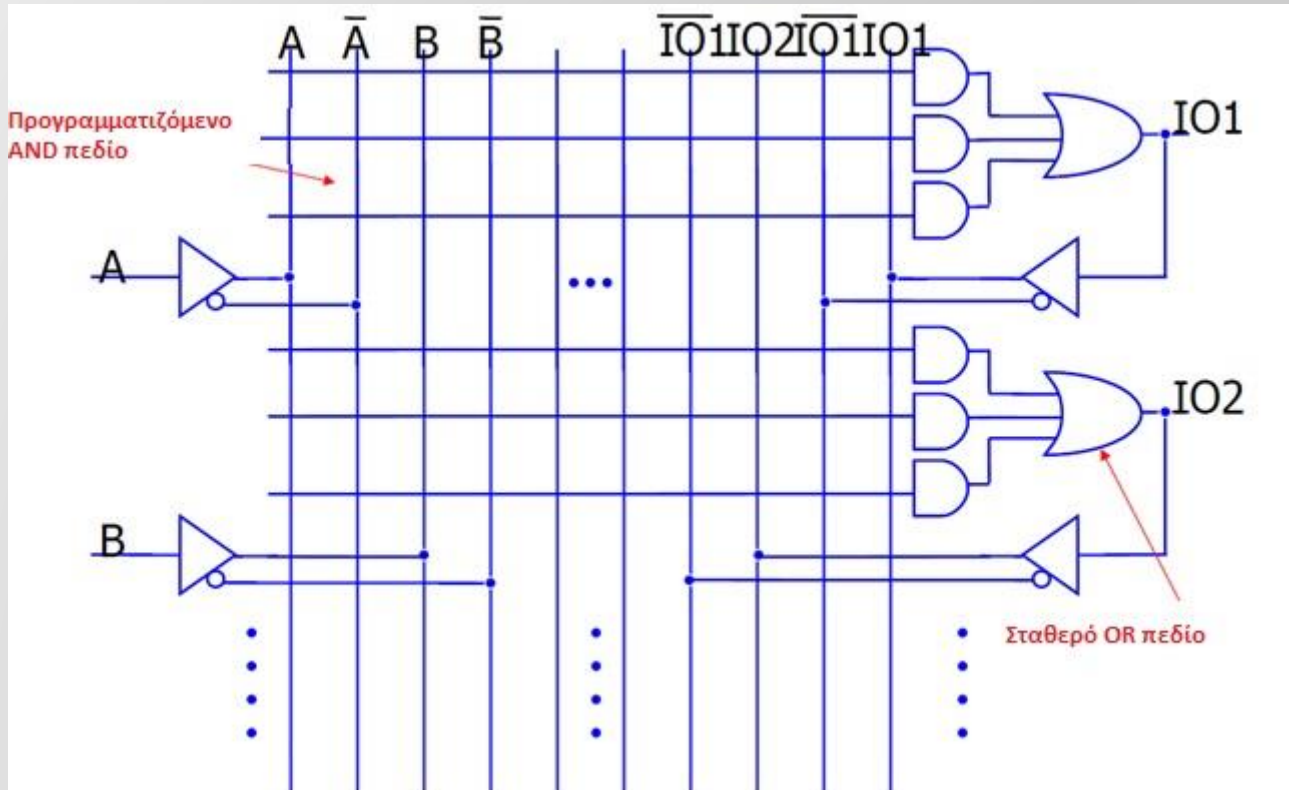
# Λύση του προηγούμενου προβλήματος (2)

- $\overline{F1} = AB + AC + BC$

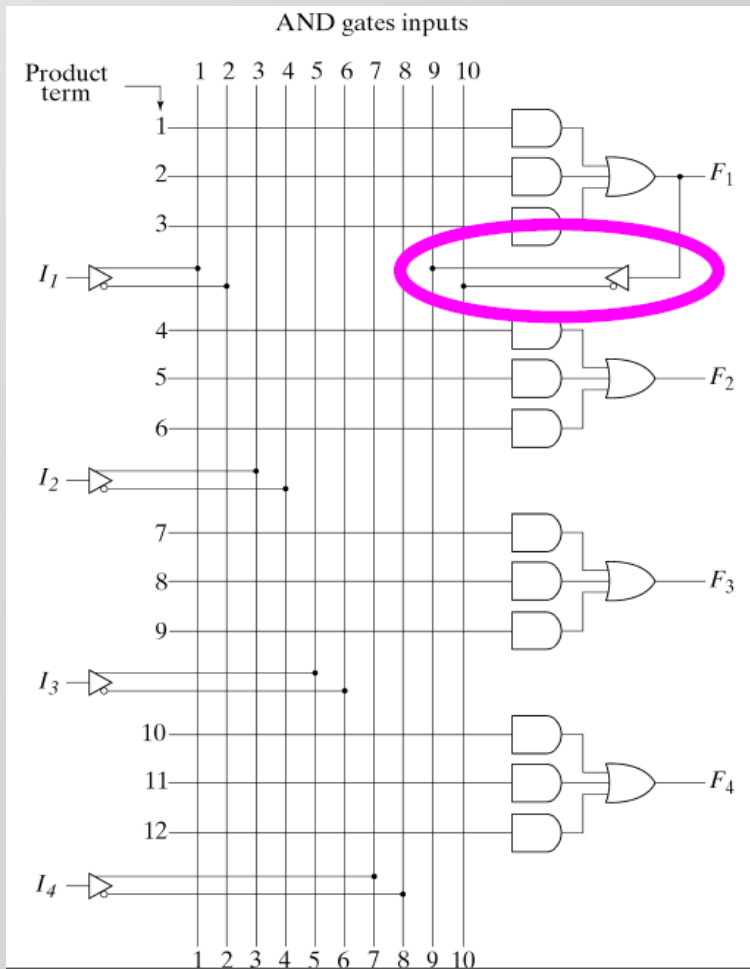
$$F2 = AB + AC + \overline{A} \overline{B} \overline{C}$$



# Programmable Array Logic ( PAL )



# Η PAL είναι πιο εύκολη στον προγραμματισμό, αλλά δεν είναι ευέλικτη



# Παράδειγμα προγραμματισμού PAL (1)

- Να υλοποιηθεί σε PAL το συνδυαστικό κύκλωμα που δίνεται από τις εξισώσεις:
  - $w(A, B, C, D) = \Sigma(2, 12, 13)$
  - $x(A, B, C, D) = \Sigma(7, 8, 9, 10, 11, 12, 13, 14, 15)$
  - $y(A, B, C, D) = \Sigma(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$
  - $z(A, B, C, D) = \Sigma(1, 2, 8, 12, 13)$
- Απλοποιώντας τις εξισώσεις:
  - $w = ABC' + A'B'CD'$
  - $x = A + BCD$
  - $y = A'B + CD + B'D'$
  - $z = w + AC'D' + A'B'CD$



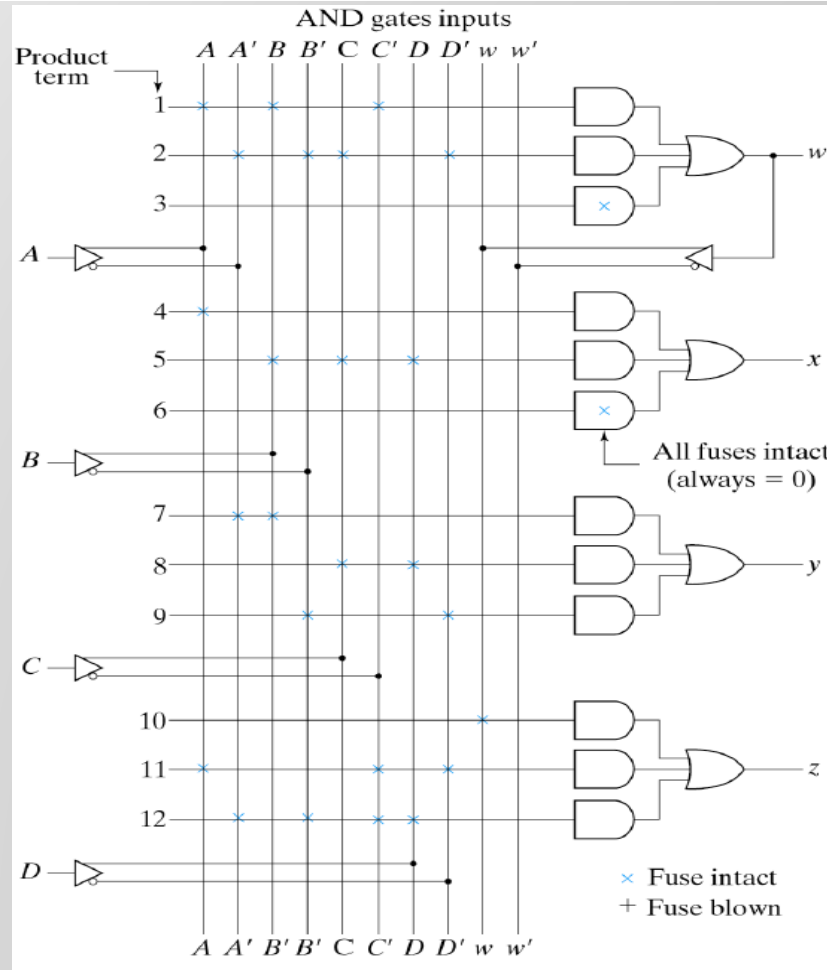
# Πίνακας Προγραμματισμού ΡΑΛ

Γινόμενο	Είσοδοι πυλών ΚΑΙ					Έξοδοι
	A	B	C	D	W	
1	1	1	0	-	-	$w = ABC' + A'B'C'D'$
2	0	0	1	0	-	
3	-	-	-	-	-	
4	1	-	-	-	-	$x = A + BCD$
5	-	1	1	1	-	
6	-	-	-	-	-	
7	0	1	-	-	-	$y = A'B + CD + B'D'$
8	-	-	1	1	-	
9	-	0	-	0	-	
10	-	-	-	-	1	$z = w + AC'D' + A'B'C'D$
11	1	-	0	0	-	
12	0	0	0	1	-	

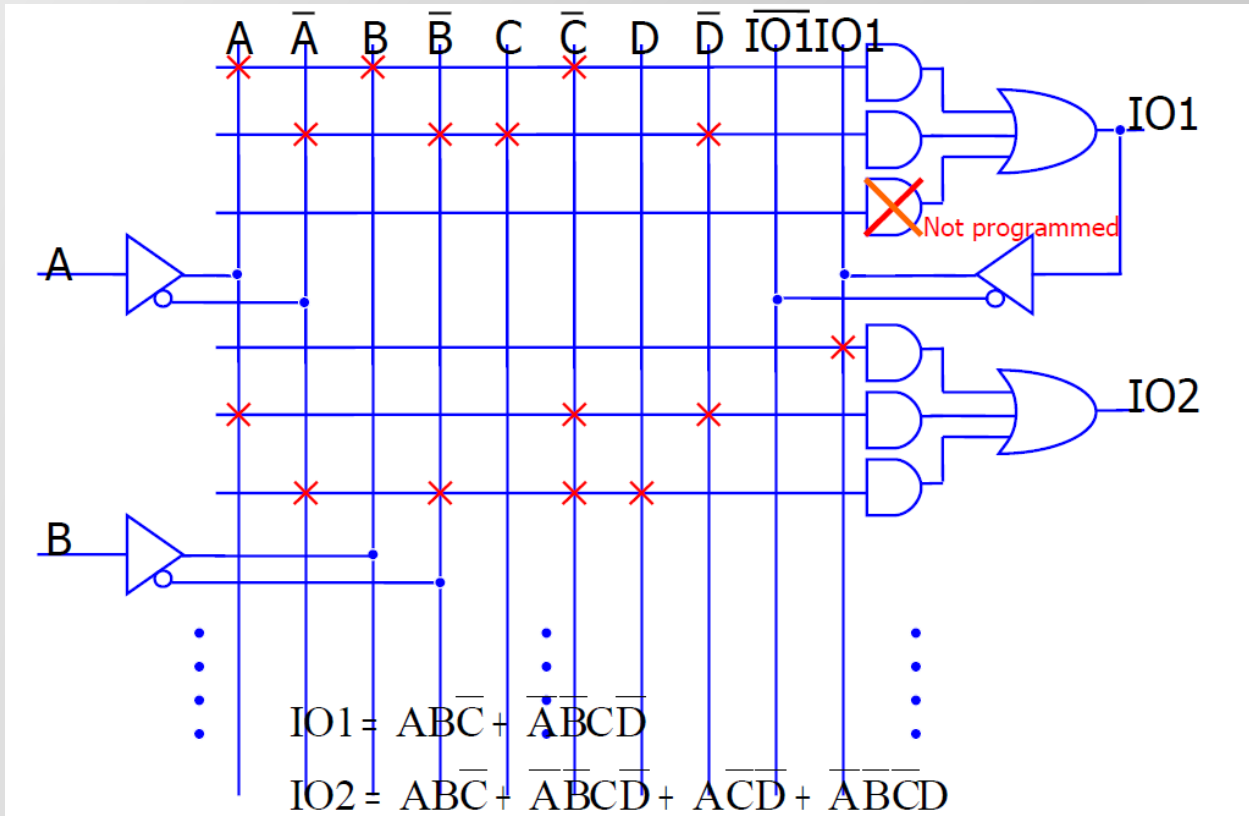




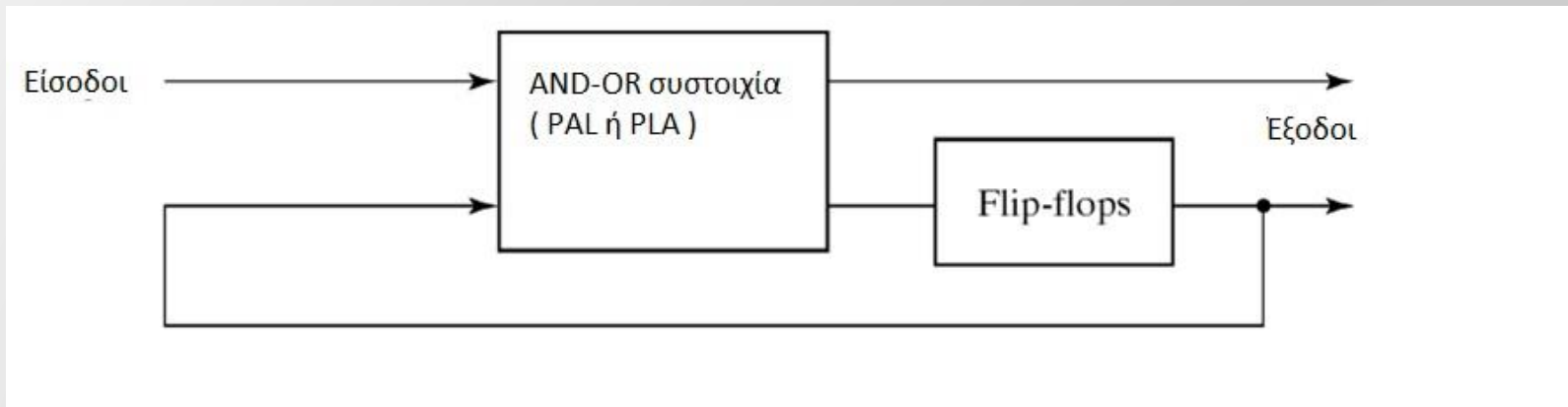
# Χάρτης διασυνδέσεων



# Παράδειγμα προγραμματισμού PAL (2)



## Κατασκευάζουμε μια ακολουθιακή PLD αν προσθέσουμε flip-flop



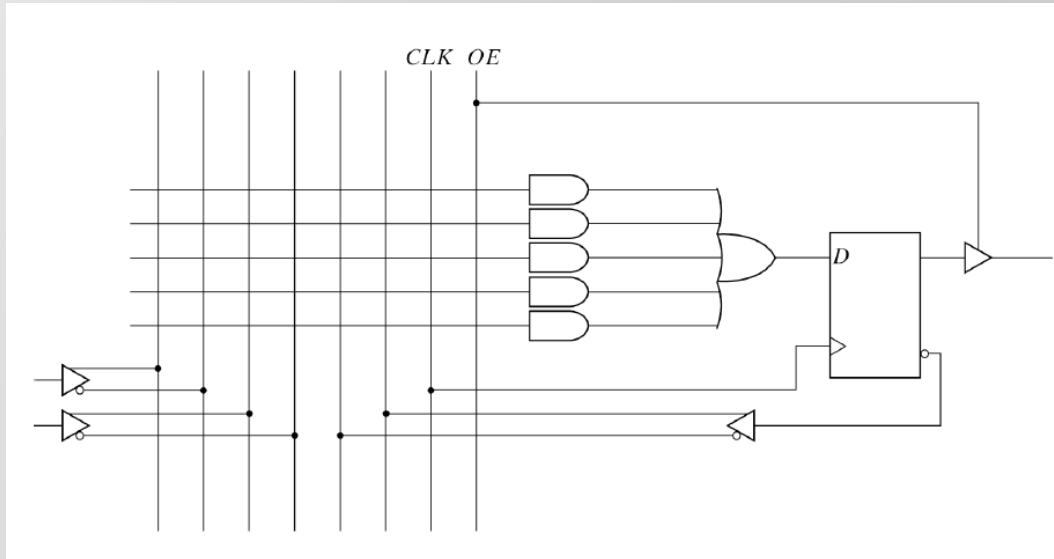
Υπάρχουν 3 κύριοι τύποι

- Ακολουθιακή ( ή απλή ) PLD –SPLD.
- Πολύπλοκη PLD –CPLD.
- Προγραμματίσιμη στο εργαστήριο διάταξη πυλών ( Field Programmable gate array – FPGA).



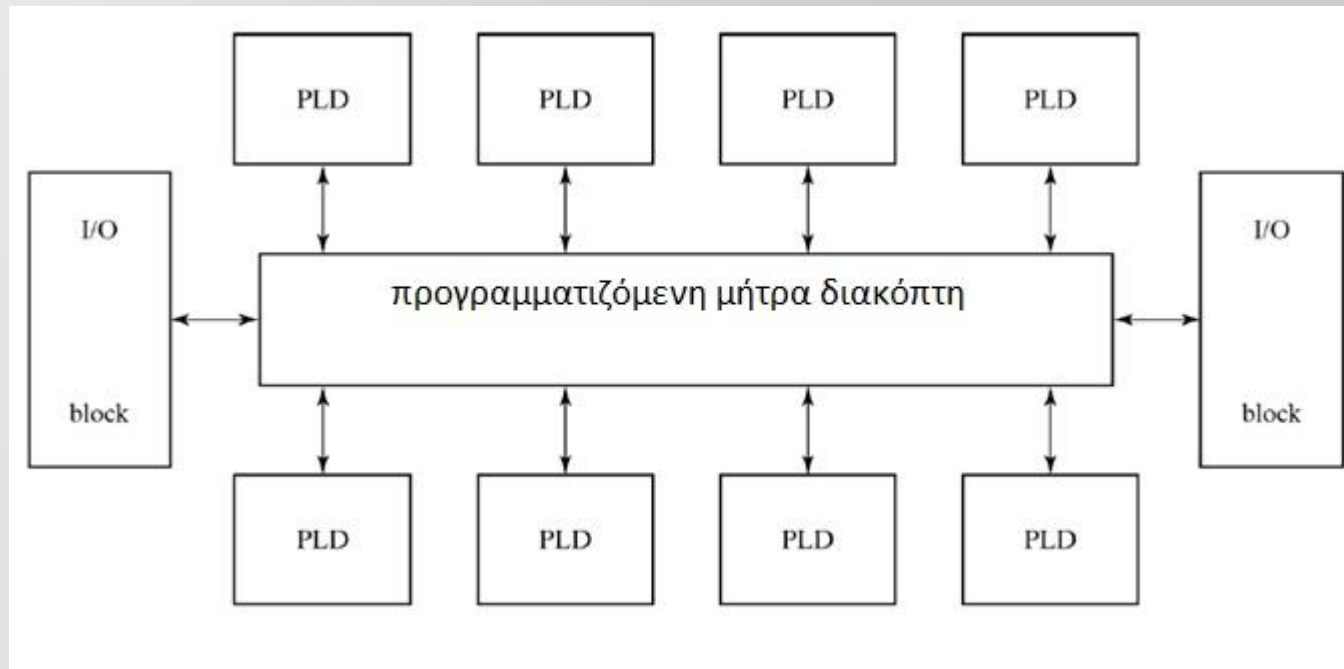
# Δομικό στοιχείο SPLD ( μακροκυψέλη )

Σήμα ελέγχου ( output enable )

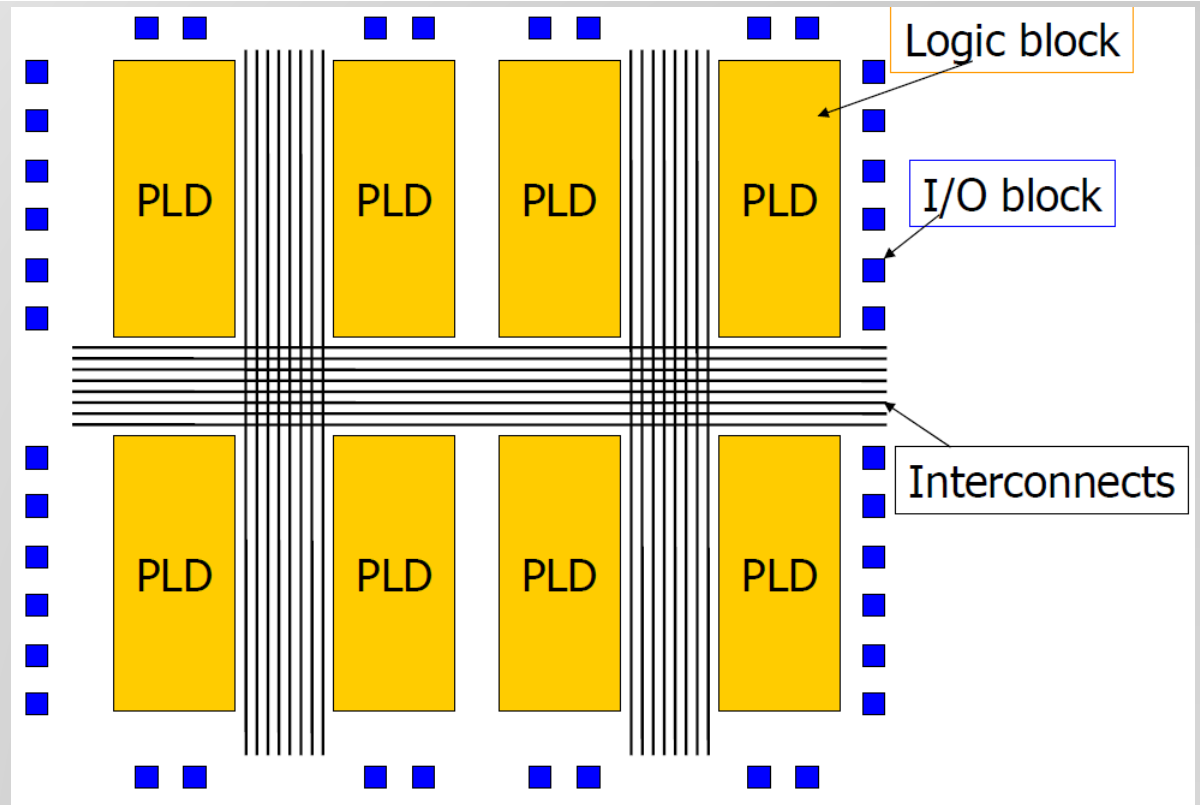


- Μπορεί να παρακαμφθεί το FF.
- Δυνατότητα για άμεση θέση / επαναφορά.
- Συμπλήρωμα της εξόδου.

# Η CPLD είναι σύνολο από PLD με προγραμματίσιμη δομή διασυνδέσεων



# Δομή CPLD



Logic block: Λογικό διάγραμμα

I/O block: Διάγραμμα εισόδου/εξόδου

Interconnects: Διασυνδέσεις



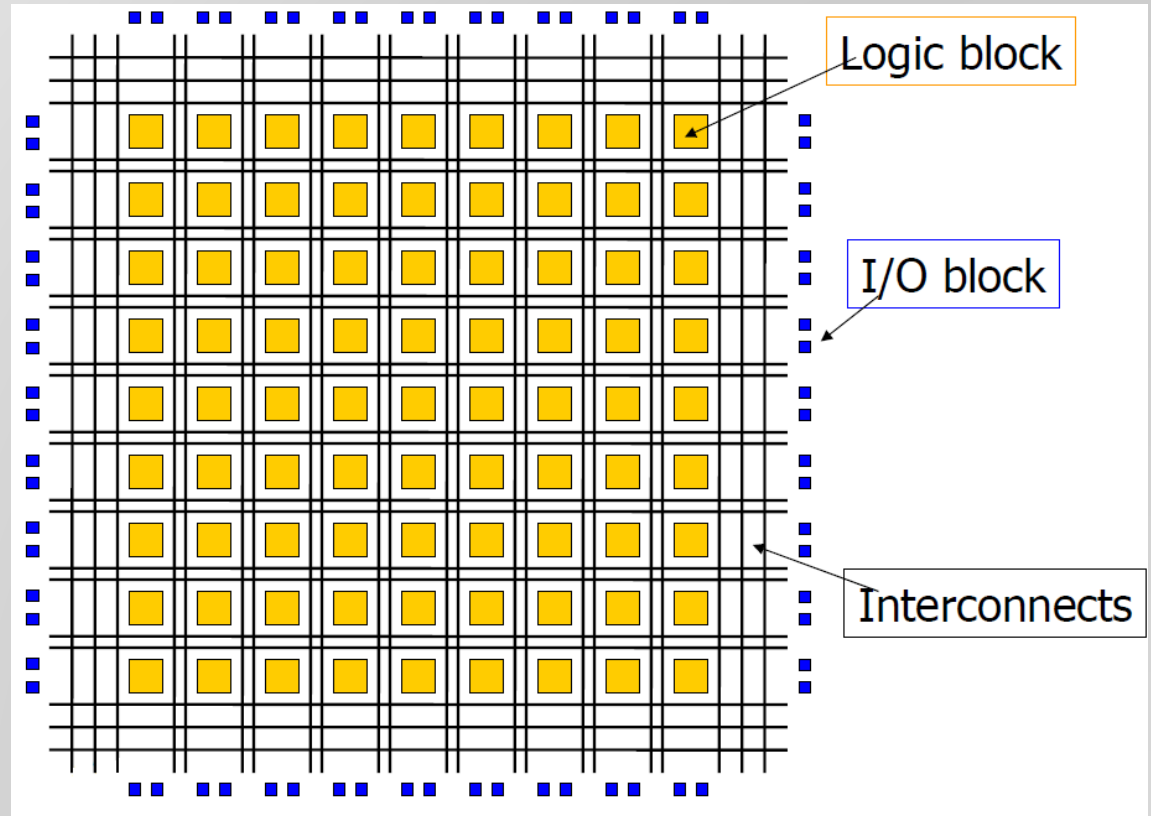
# Gate Arrays – Διατάξεις πυλών

---

- Η βασική συνιστώσα κυκλώματος είναι η διάταξη πυλών ( gate array ).
- Αποτελείται από χιλιάδες έως εκατοντάδες χιλιάδες πυλών.
- Η πιο κοινή είναι η FPGA ( field programmable gate array ). Έχει:
  - Πίνακες αναζήτησης.
  - Πολυπλέκτες.
  - Πύλες.
  - FF.



# Δομή FPGA



Logic block: Λογικό διάγραμμα

I/O block: Διάγραμμα εισόδου/εξόδου

Interconnects: Διασυνδέσεις





---

# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
ανάπτυξη στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

