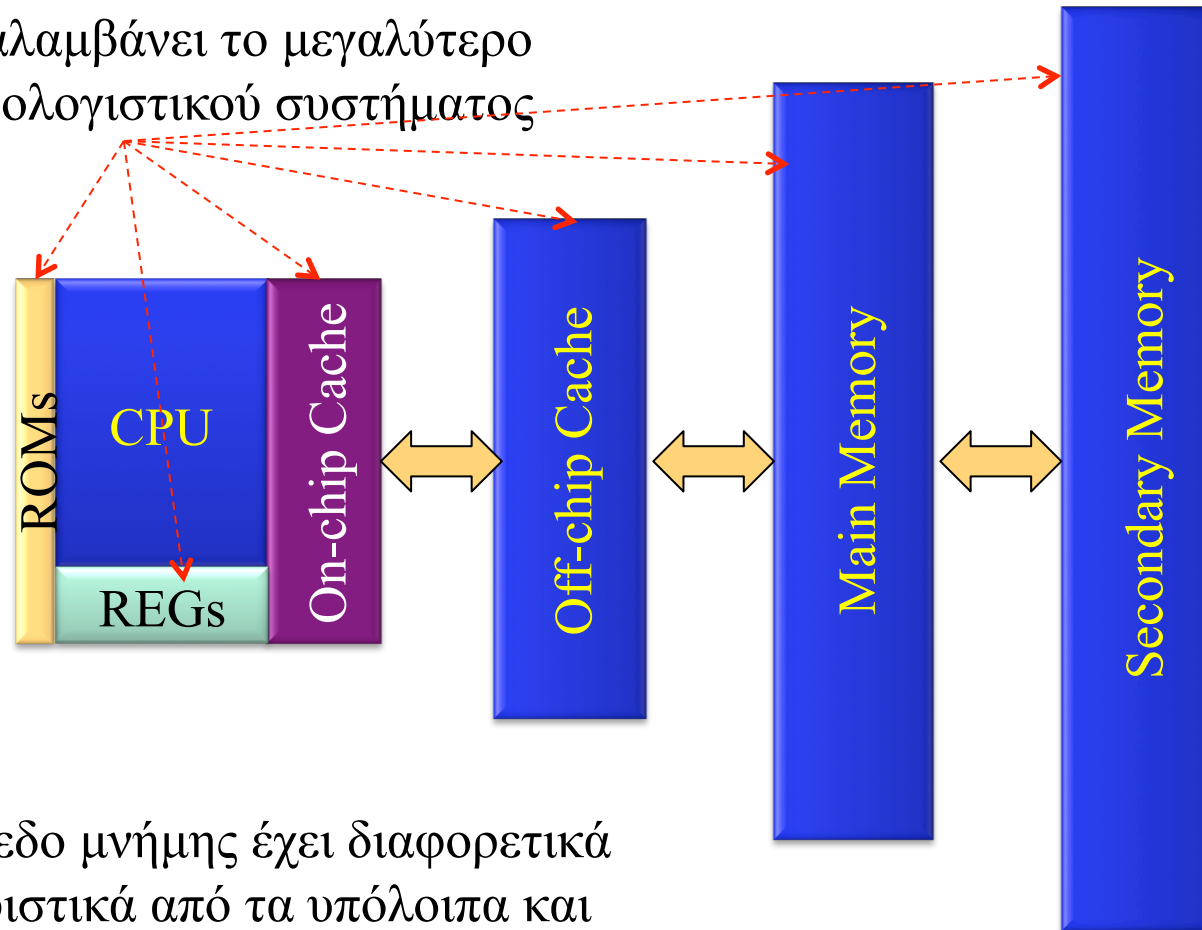

*5^η Θεματική Ενότητα : Μνήμη &
Προγραμματιζόμενη Λογική*

Επιμέλεια διαφανειών:
Χρ. Καβουσιανός

Μνήμη

Η μνήμη καταλαμβάνει το μεγαλύτερο μέρος ενός υπολογιστικού συστήματος

Δύο τύποι:
ROM - RAM



Κάθε επίπεδο μνήμης έχει διαφορετικά χαρακτηριστικά από τα υπόλοιπα και σχεδιάζεται διαφορετικά

Τρόποι Διαχωρισμού Μνημών



1. Μόνο ανάγνωσης - Ανάγνωσης & Εγγραφής

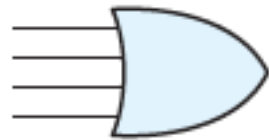
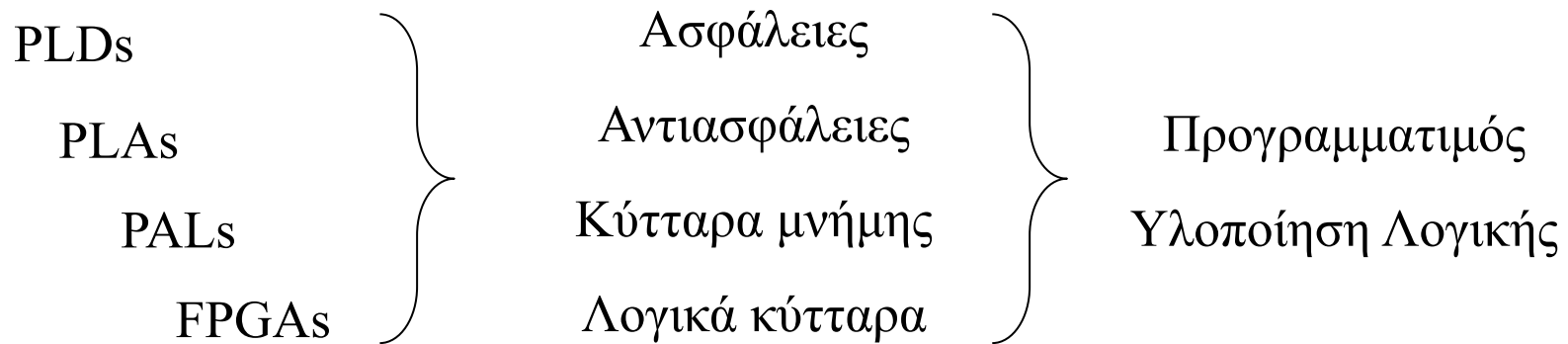
2. Στατικές - Δυναμικές

3. Τυχαίας - Σειριακής Προσπέλασης

4. Πρόσκαιρες (volatile) ή Μη (non-volatile)



Προγραμματίσιμη Λογική



(a) Conventional symbol



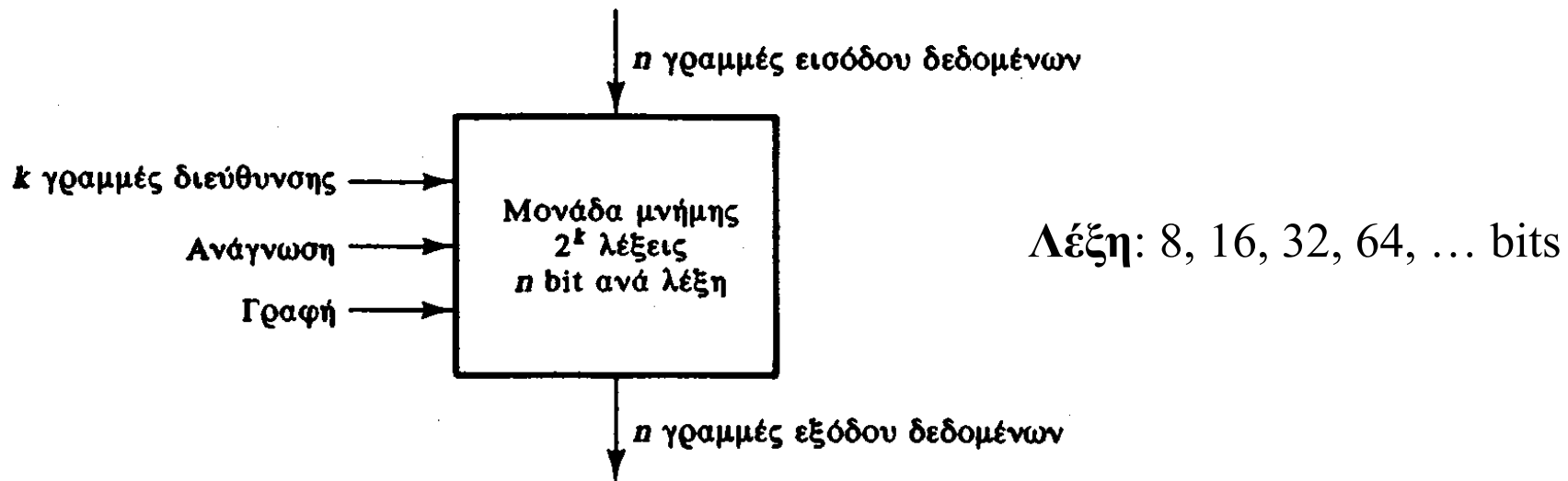
(b) Array logic symbol

Μνήμη Τυχαίας Προσπέλασης (RAM)

Μονάδα Μνήμης: Ένα σύνολο από κύτταρα αποθήκευσης μαζί με τα απαραίτητα κυκλώματα για τη μεταφορά πληροφοριών.

RAM: η μεταφορά μπορεί να γίνει από και προς οποιαδήποτε τυχαία θέση σε σταθερό χρόνο.

Οι δυαδικές πληροφορίες αποθηκεύονται ομαδοποιημένες σε **λέξεις** (**words**).



Παράδειγμα: Περιεχόμενα Μνήμης 1024 x 16

Memory address		Memory content
Binary	Decimal	
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

1 Kbyte = 2¹⁰ bytes
1 Mbyte = 2²⁰ bytes
1 Gbyte = 2³⁰ bytes
1 Tbyte = 2⁴⁰ bytes

10 bit διεύθυνση
 16 bits λέξη

Λειτουργίες Γραφής/Ανάγνωσης

Εγγραφή Μνήμης.

1. Μεταφορά της δυαδικής διεύθυνσης της λέξης στις γραμμές διεύθυνσης.
2. Μεταφορά των bits δεδομένων στις γραμμές δεδομένων
3. Ενεργοποίηση της εισόδου ελέγχου γραφής.

Ανάγνωση Μνήμης.

1. Μεταφορά της δυαδικής διεύθυνσης της λέξης στις γραμμές διεύθυνσης.
2. Ενεργοποίηση της εισόδου ελέγχου ανάγνωσης.

Memory Enable	Read/Write	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

Κυματομορφές Χρονισμού

Η μνήμη "χρονίζεται" από τον επεξεργαστή.

Χρόνος Προσπέλασης: χρόνος για την επιλογή μίας διεύθυνσης και την ανάγνωση του περιεχομένου της

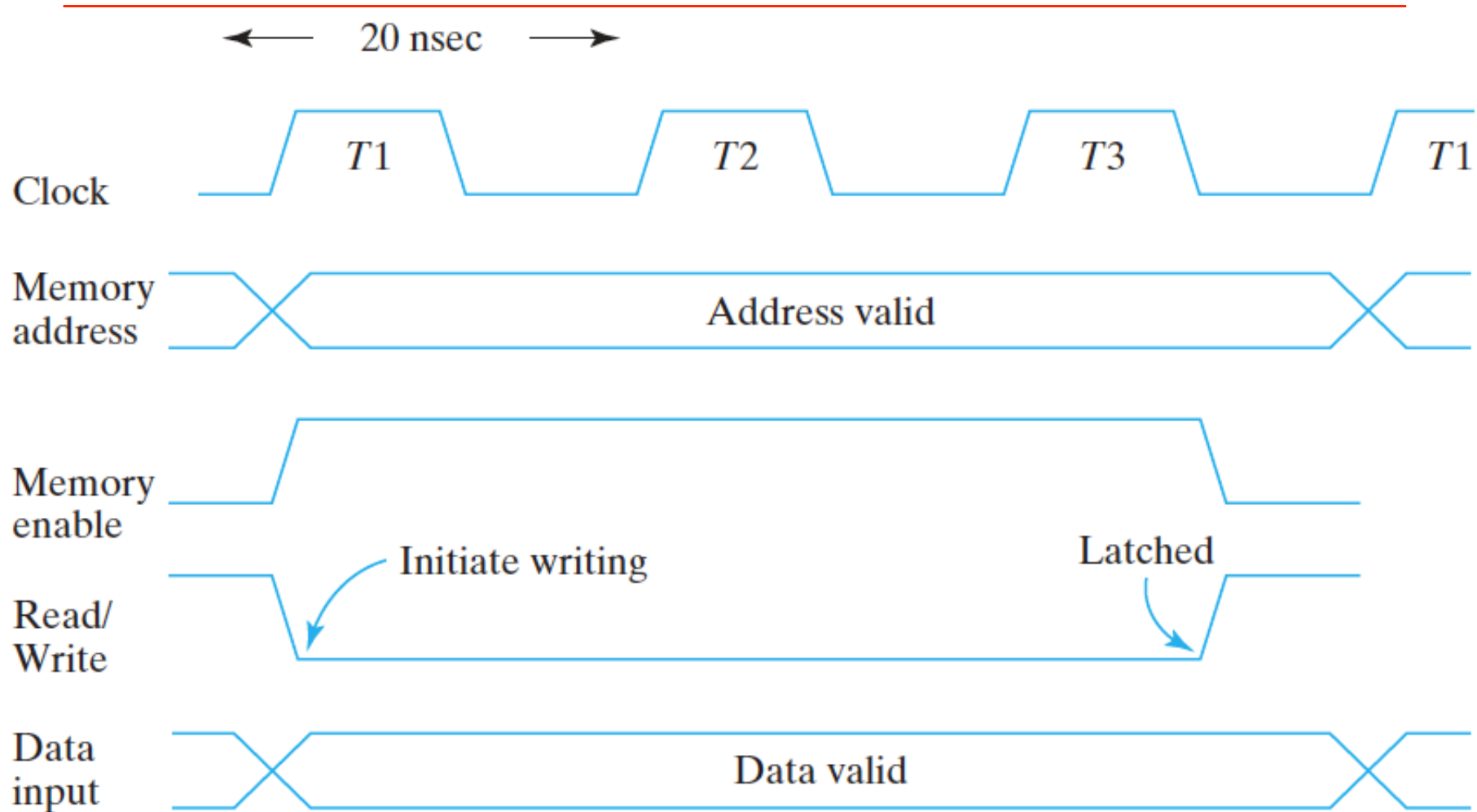
Χρόνος Κύκλου: χρόνος για την ολοκλήρωση μίας λειτουργίας εγγραφής

Η μνήμη ανταποκρίνεται σε **πολλαπλούς** κύκλους ρολογιού, ανάλογα με την ταχύτητα απόκρισης της



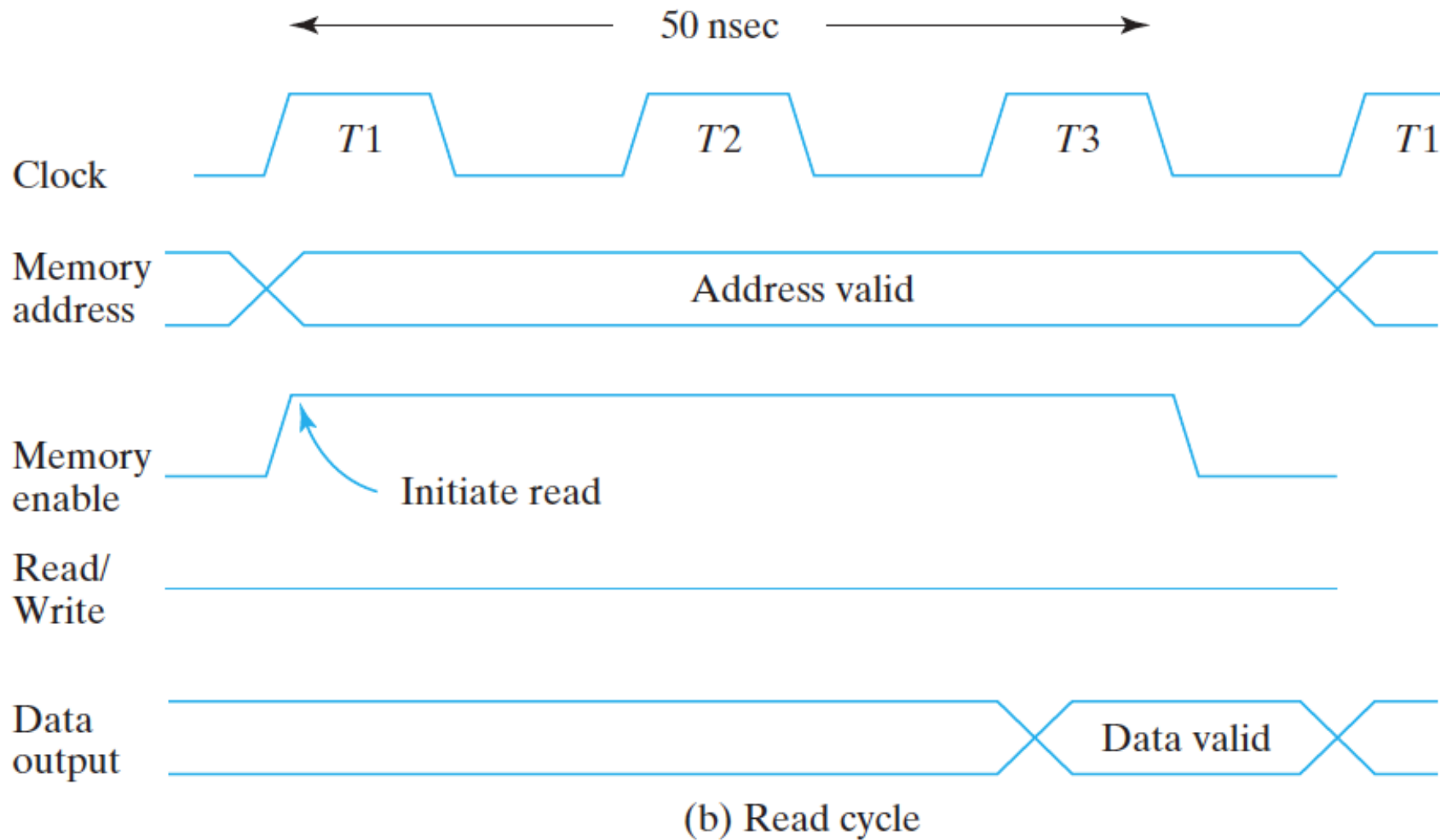
Ο επεξεργαστής αναμένει επαρκή αριθμό κύκλων για να απαντήσει η μνήμη

Κυματομορφή Εγγραφής



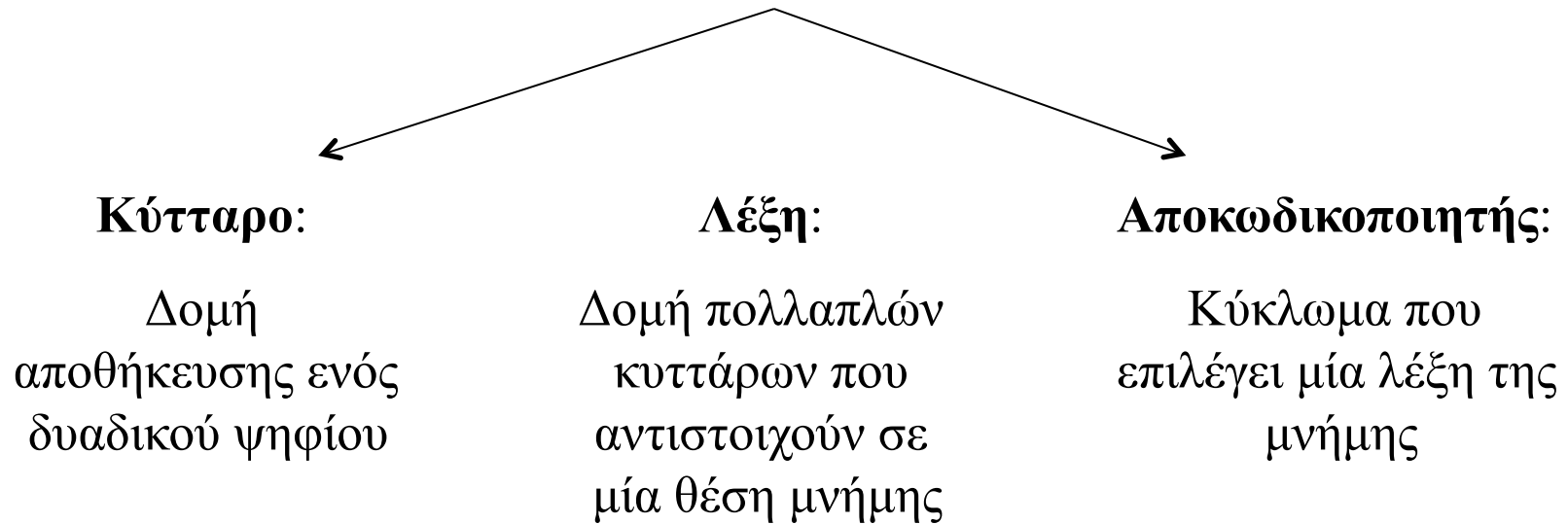
(a) Write cycle

Κυματομορφή Ανάγνωσης

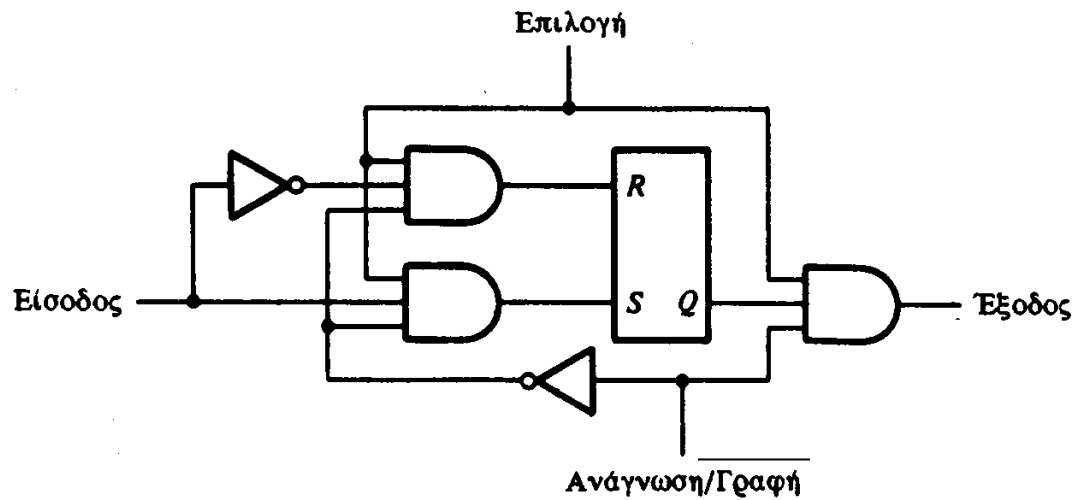


Αποκωδικοποίηση Μνήμης

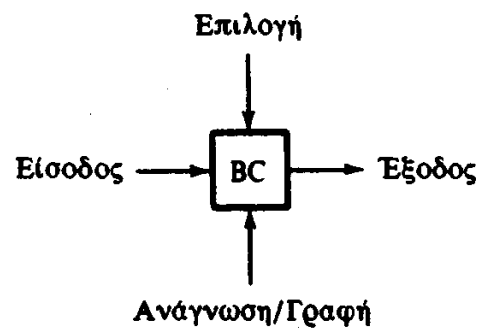
Αποκωδικοποίηση είναι η διαδικασία επιλογής της θέσης μνήμης την οποία θα γράψουμε ή θα διαβάσουμε.



Ένα κύτταρο μνήμης

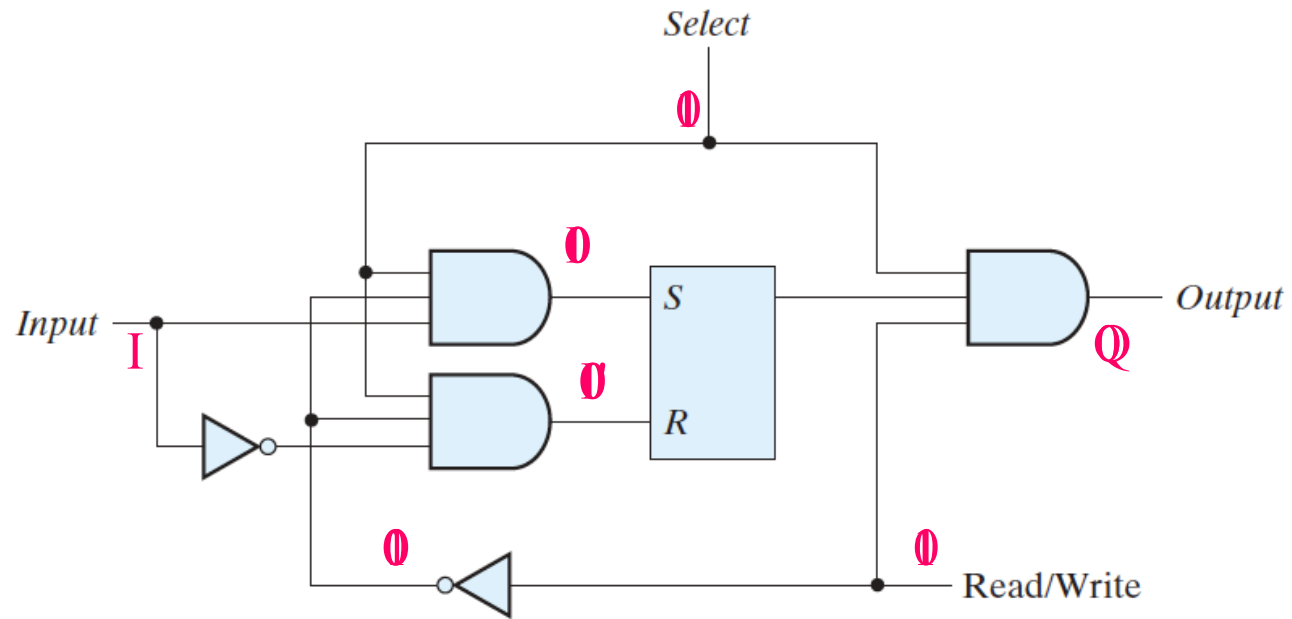


(α) Λογικό διάγραμμα



(β) Συμβολικό διάγραμμα

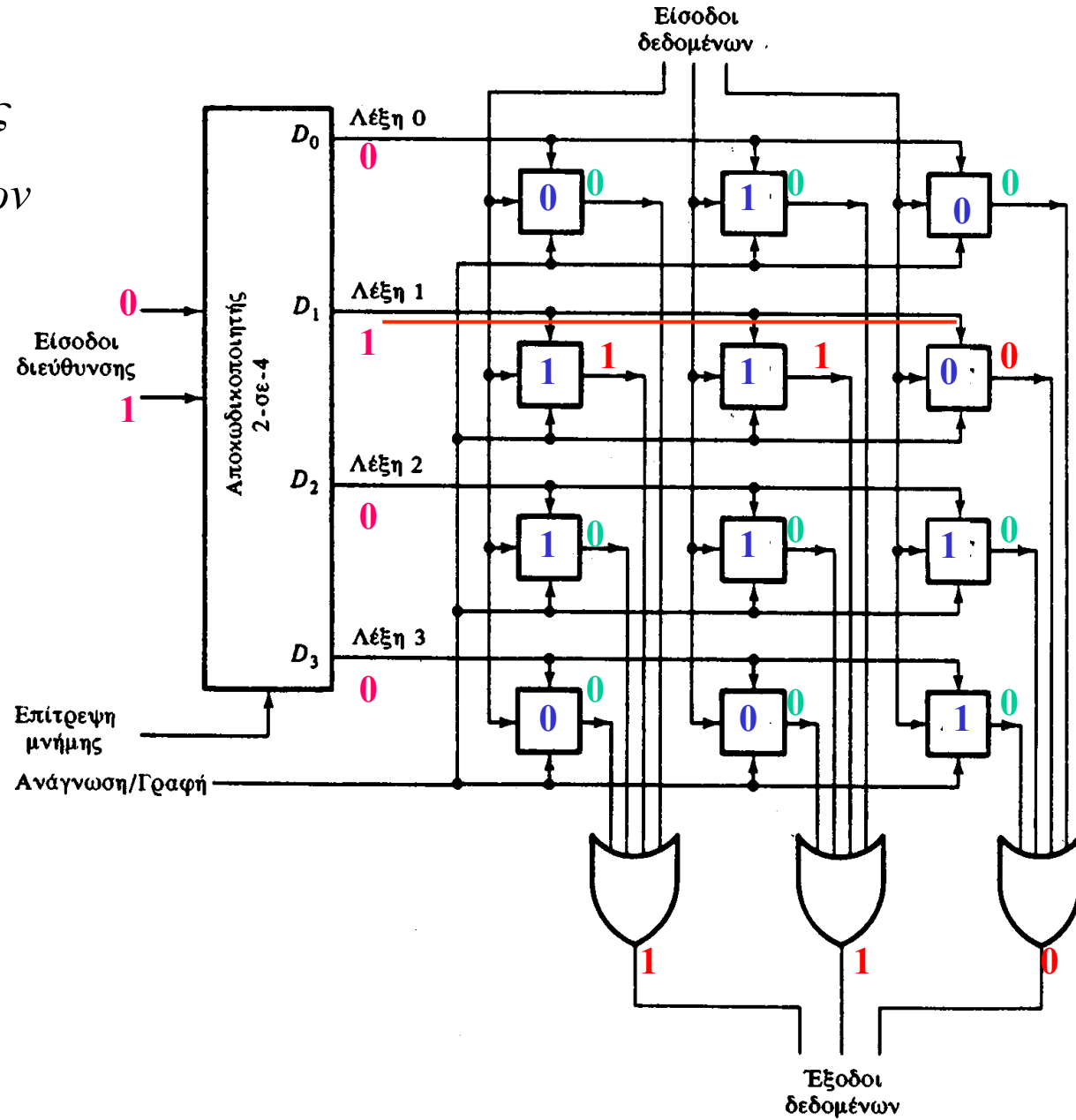
Ένα κύτταρο μνήμης



(a) Logic diagram

Παράδειγμα Μνήμης
4x3 δυαδικών ψηφίων

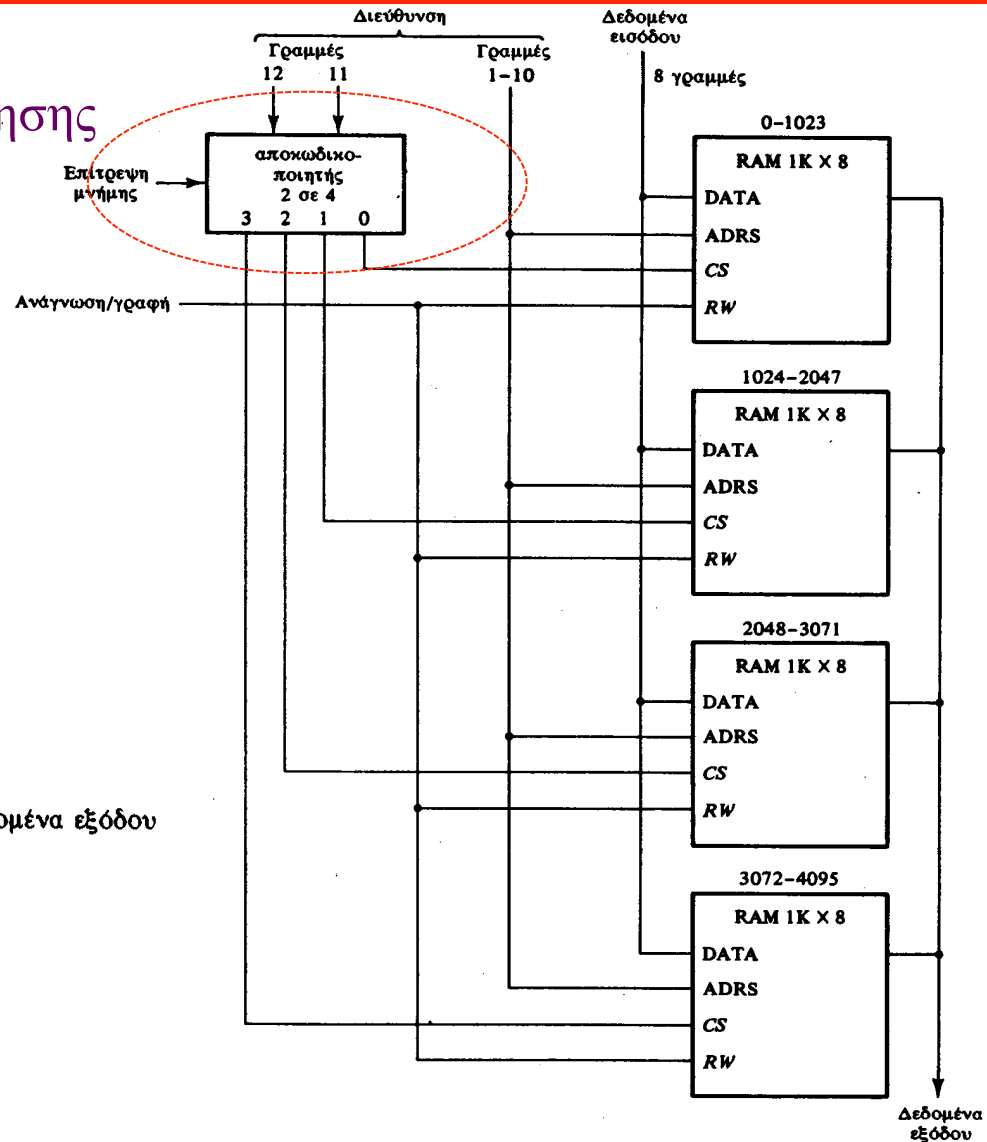
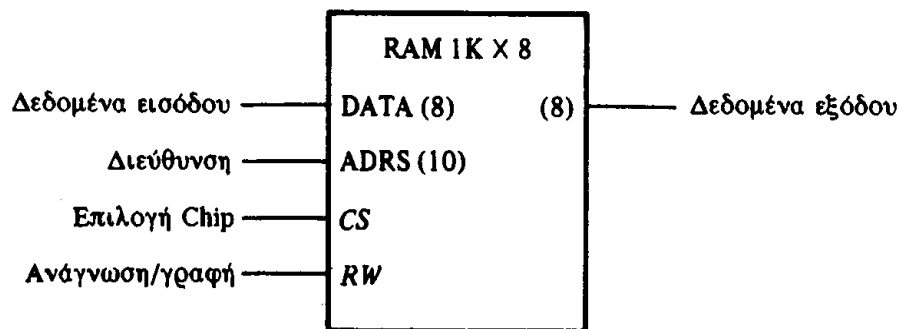
Εσωτερική Οργάνωση Μνήμης



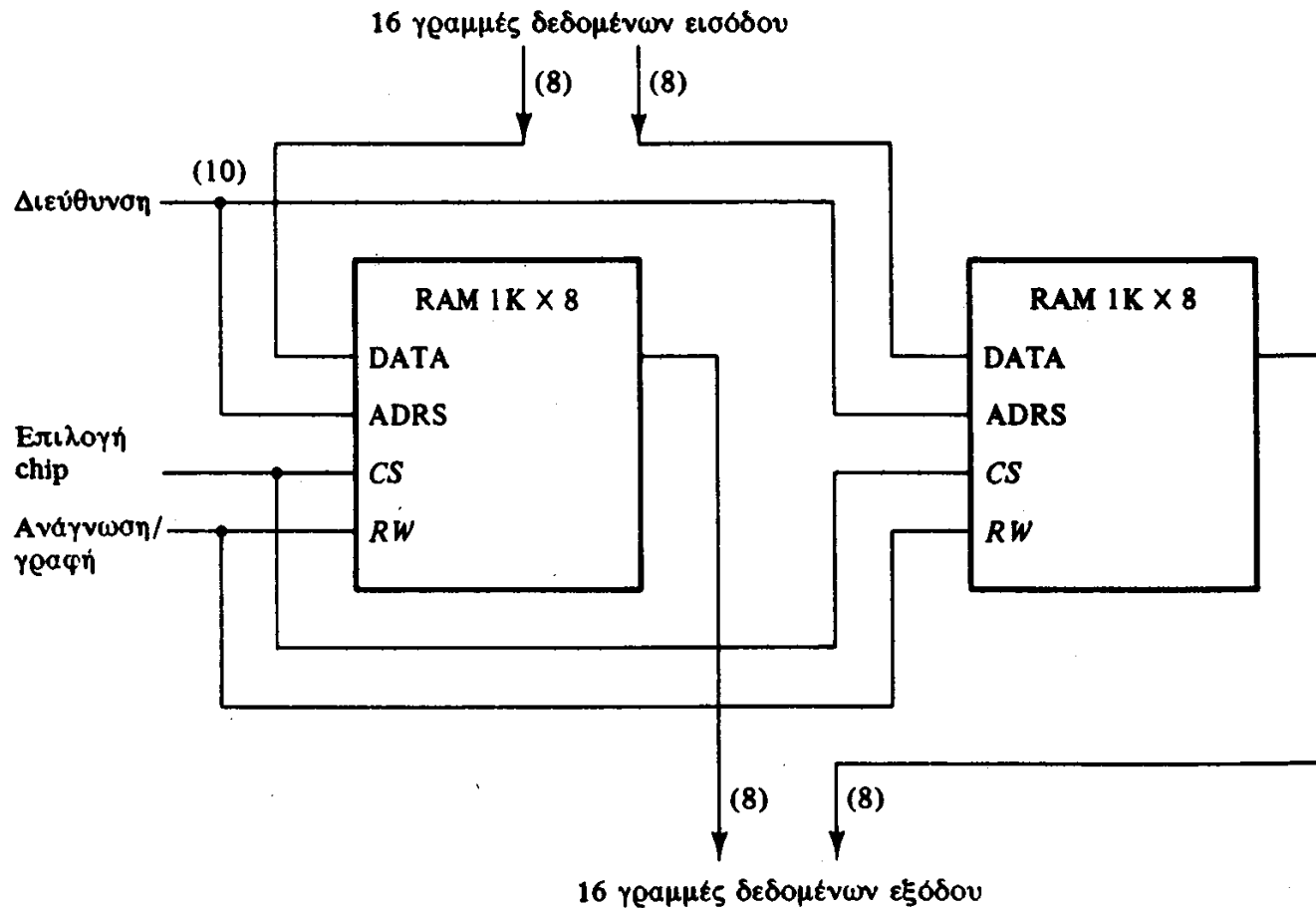
Επέκταση μνήμης

2^ο επίπεδο αποκωδικοποίησης

Χρησιμοποιούνται πολλαπλά ολοκληρωμένα μνήμης για την δημιουργία μεγαλύτερων (σε χωρητικότητα) μνημών



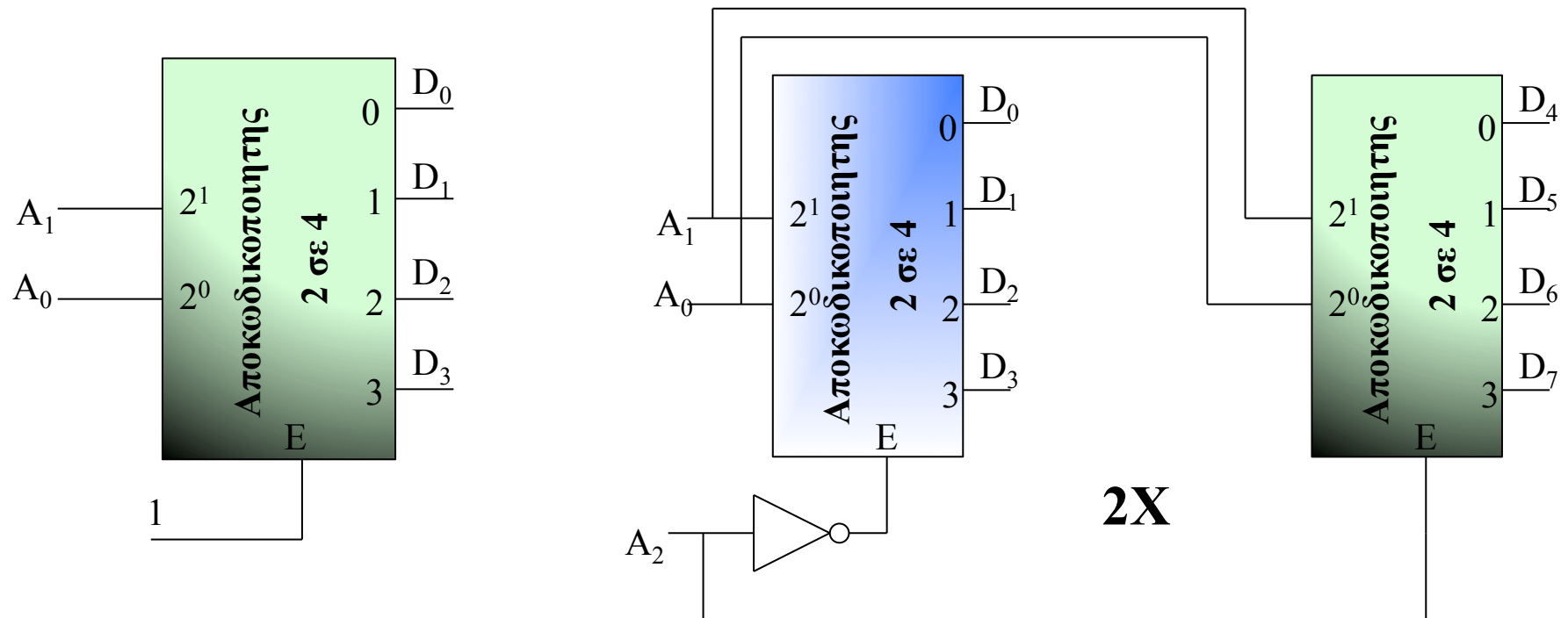
Παράταξη με RAM chips



Οι εισοδοι δεδομένων ταυτίζονται με τις εξόδους αποτελεσμάτων. Αυτό οδηγεί στην χρήση αμφίδρομων στοιχείων (bidirectional).

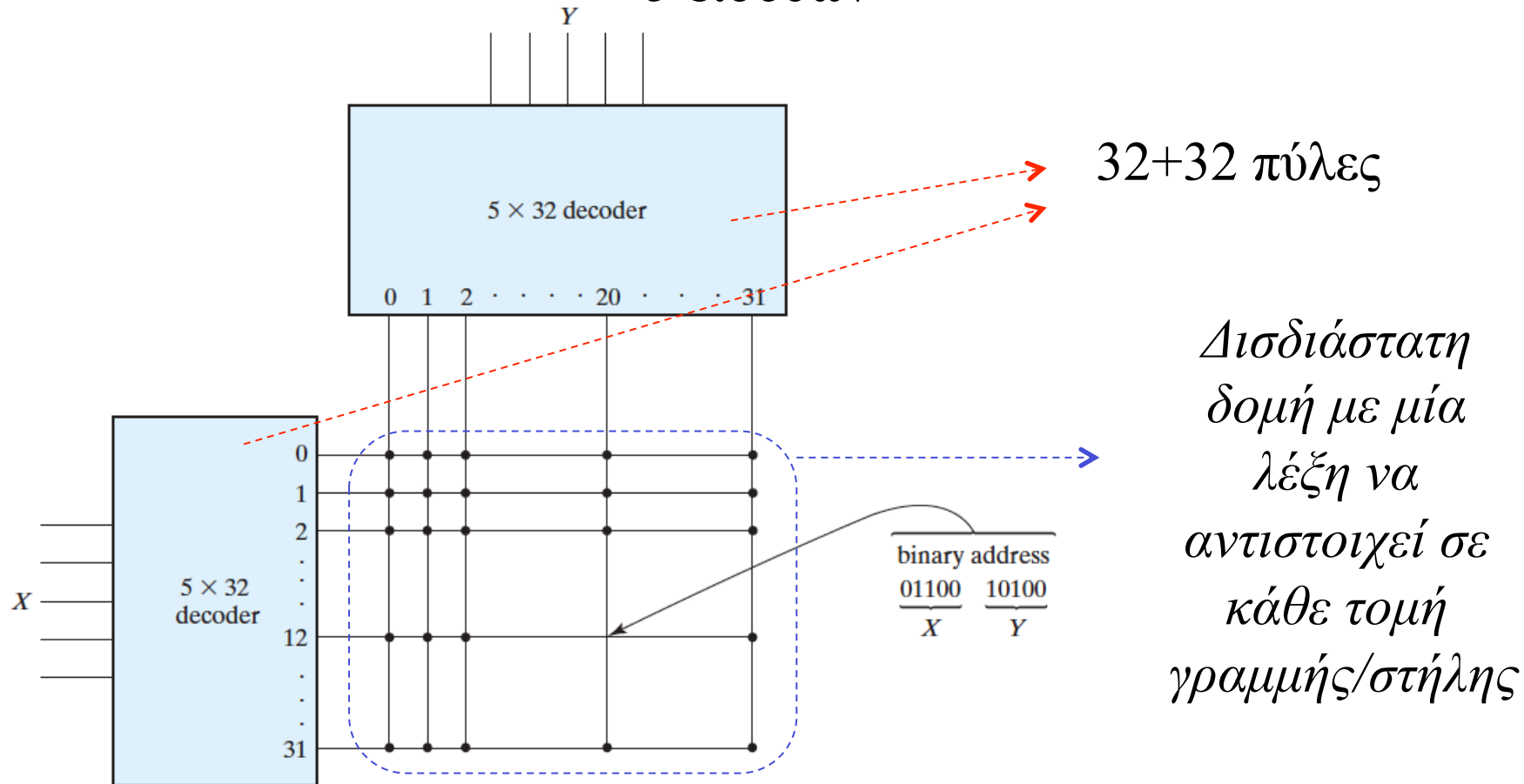
Κόστος Αποκωδικοποίησης

Ο αποκωδικοποιητής αυξάνει εκθετικά σε μέγεθος με γραμμική αύξηση του αριθμού των διευθύνσεων.



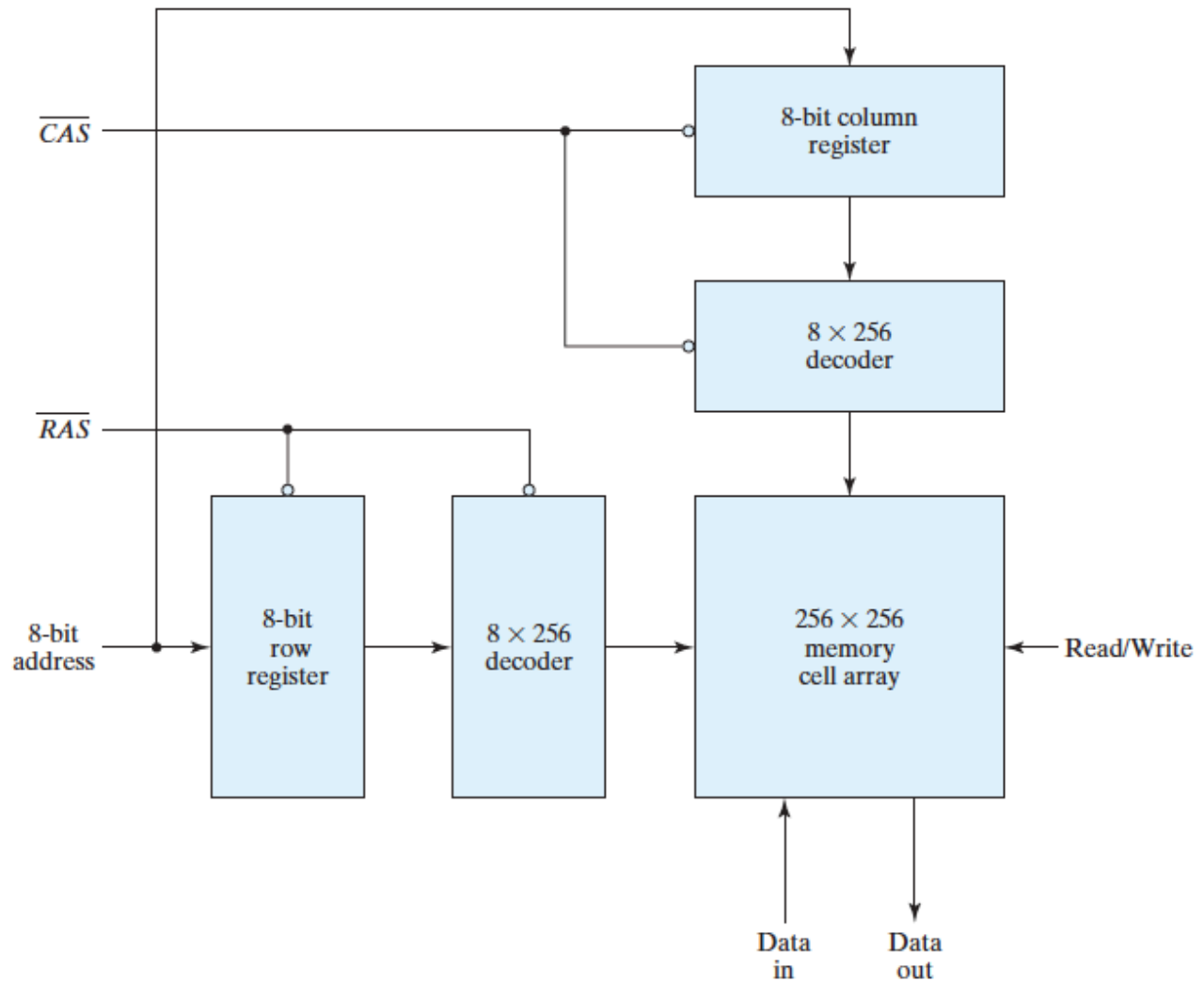
Ταυτόχρονη Αποκωδικοποίηση

Διεύθυνση 10 δυαδικών ψηφίων απαιτεί $2^{10} = 1024$ πύλες ΚΑΙ
5 εισόδων



Πολύπλεξη Διευθύνσεων

Για μείωση ακροδεκτών και διαύλων, η διεύθυνση αποκωδικοποιείται σε γραμμές-στήλες με πολύπλεξία στον χρόνο



Θόρυβος & Ακτινοβολία



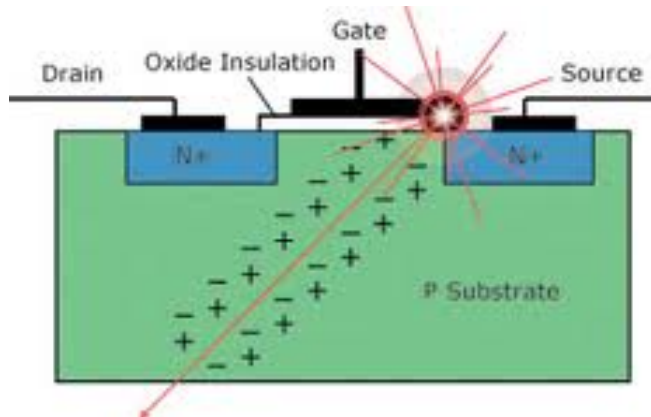
Οι μνήμες καταλαμβάνουν την μεγαλύτερη επιφάνεια του ολοκληρωμένου και δέχονται ακτινοβολία.



Οι μικρές διαστάσεις στις υπομικρονικές τεχνολογίες οδηγεί σε καταστροφή δεδομένων



Απαιτείται προστασία δεδομένων ειδικά σε διαστημικές εφαρμογές



Κώδικες Ανίχνευσης/Διόρθωσης Λαθών

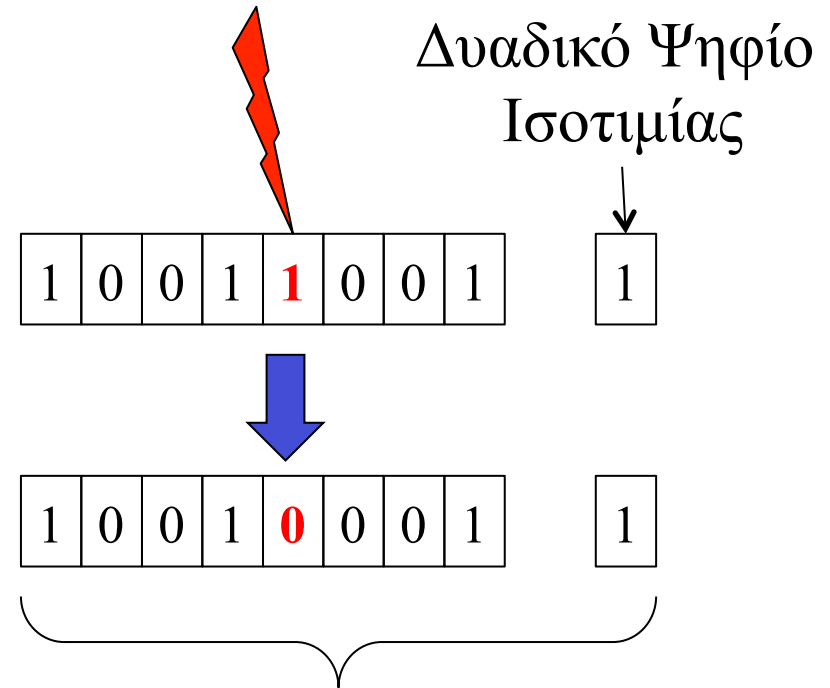
Ο απλούστερος κώδικας είναι ο κώδικας ισοτιμίας



Ανίχνευση λαθών



Η διόρθωση λαθών απαιτεί πλεονασμό πληροφορίας ώστε να εντοπιστεί το λανθασμένο ψηφίο



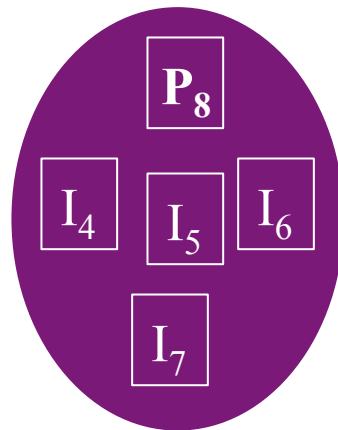
Κώδικας Hamming

C_8	C_4	C_2	C_1	
0	0	0	0	-
0	0	0	1	P_1
0	0	1	0	P_2
0	0	1	1	I_0
0	1	0	0	P_4
0	1	0	1	I_1
0	1	1	0	I_2
0	1	1	1	I_3
1	0	0	0	P_8
1	0	0	1	I_4
1	0	1	0	I_5
1	0	1	1	I_6
1	1	0	0	I_7

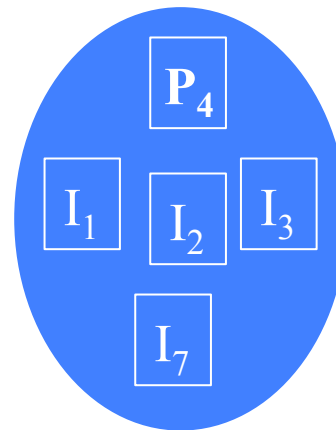
I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7



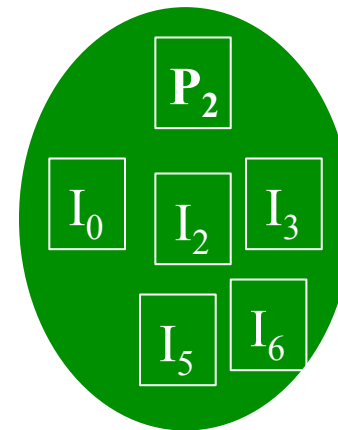
P_1 P_2 I_0 P_4 I_1 I_2 I_3 P_8 I_4 I_5 I_6 I_7



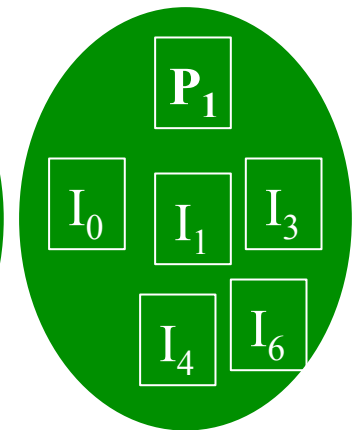
C_8



C_4

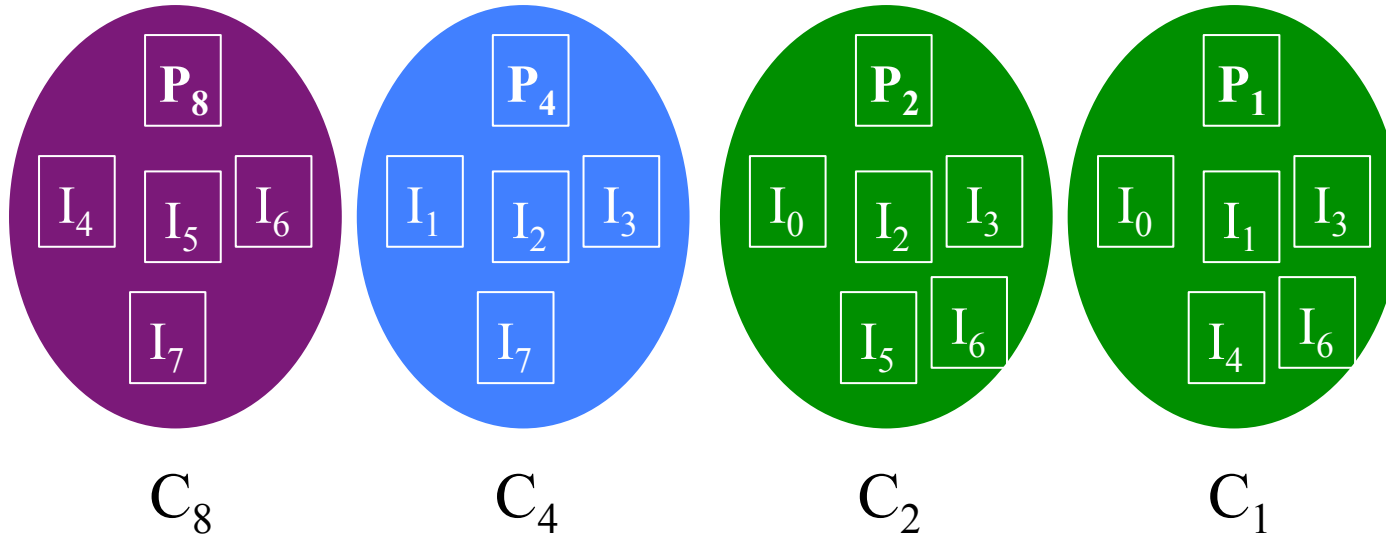


C_2



C_1

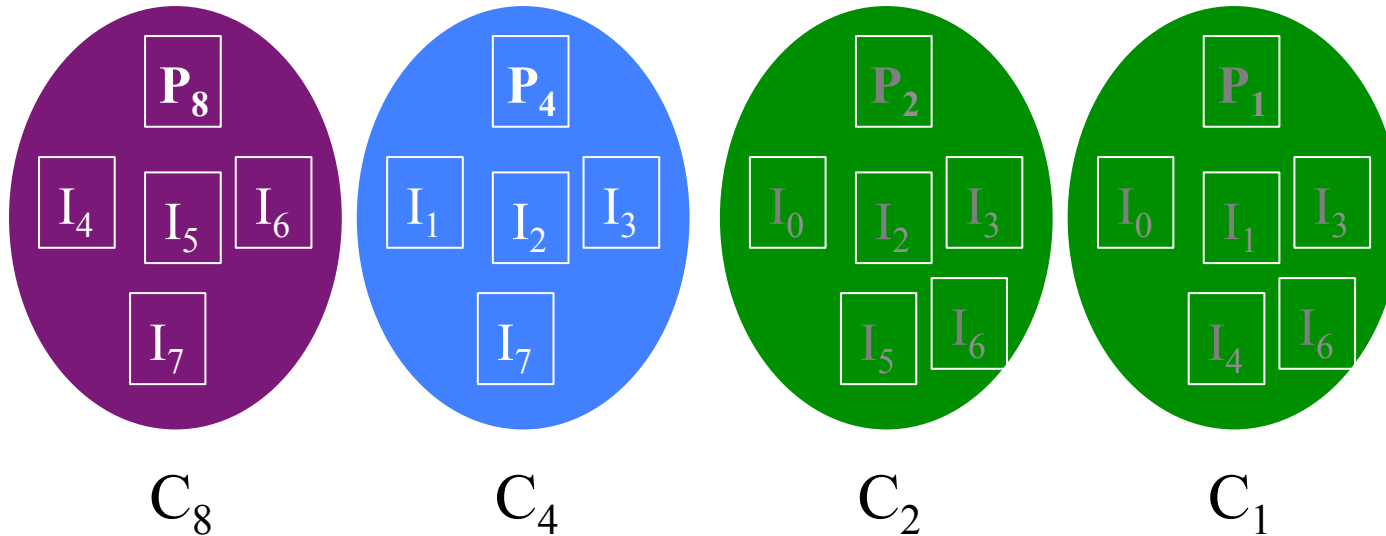
Κώδικας Hamming



Η κάθε ομάδα ελέγχεται για την ισοτιμία της.

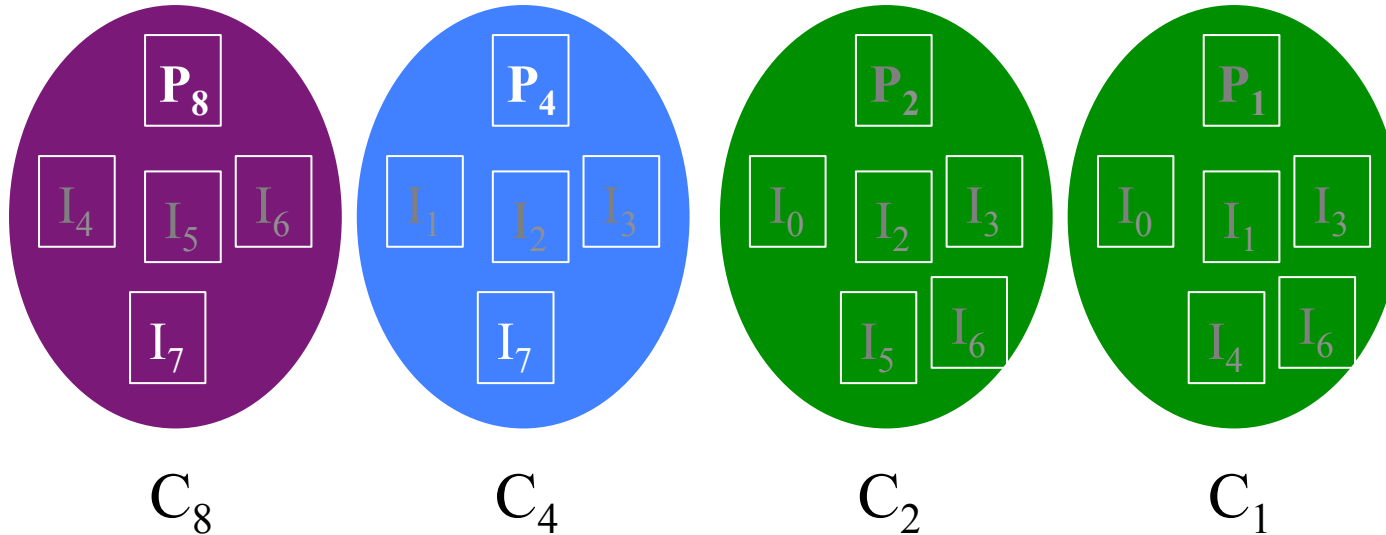
Έστω η ομάδα C_8 και η C_4 έχουν λάθος ισοτιμία και οι υπόλοιπες σωστή

Κώδικας Hamming



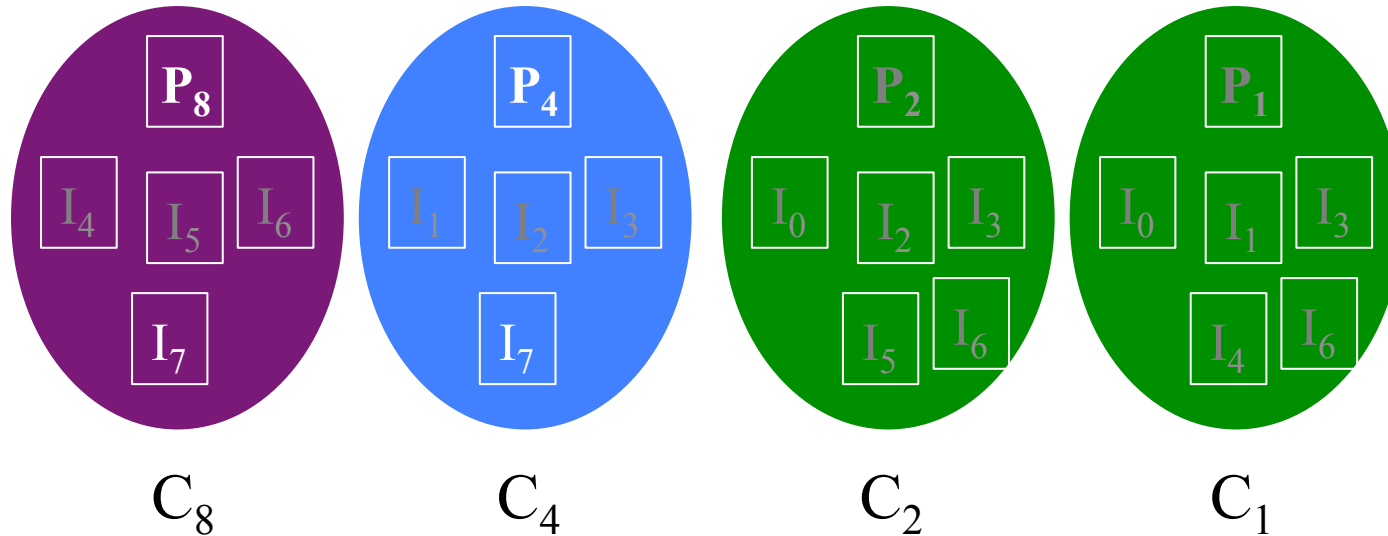
1. Το λάθος δεν μπορεί να είναι ούτε στην C_1 , ούτε στην C_2 .

Κώδικας Hamming



1. Το λάθος δεν μπορεί να είναι ούτε στην C_1 , ούτε στην C_2 .
2. Τα ψηφία που απαλλάσσονται διαγράφονται και από τις υπόλοιπες μονάδες.

Κώδικας Hamming



1. Το λάθος δεν μπορεί να είναι ούτε στην C_1 , ούτε στην C_2 .
2. Τα ψηφία που απαλλάσσονται διαγράφονται παντού.
3. Το λάθος επηρεάζει και την C_8 και την C_4 . Το μόνο κοινό τους που μένει είναι το I_7 , το οποίο είναι τελικά το λάθος.

Κώδικας Hamming

Παρατηρήστε ότι αν θέσουμε $C_8=C_4=1$ που αντιστοιχούν στις λανθασμένες ομάδες, τότε ο δυαδικός αριθμός

$$C_8C_4C_2C_1 = 1100$$

υποδεικνύει την λανθασμένη θέση (12)

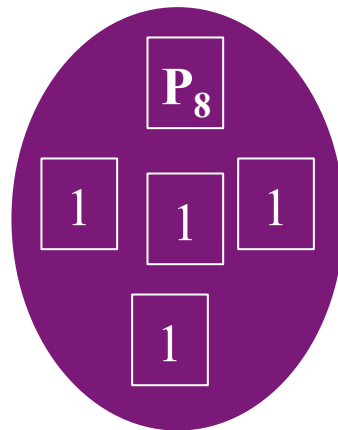
1	2	3	4	5	6	7	8	9	10	11	12
P₁	P₂	I ₀	P₄	I ₁	I ₂	I ₃	P₈	I ₄	I ₅	I ₆	I₇

Κωδικοποίηση (περιττή ισοτιμία)

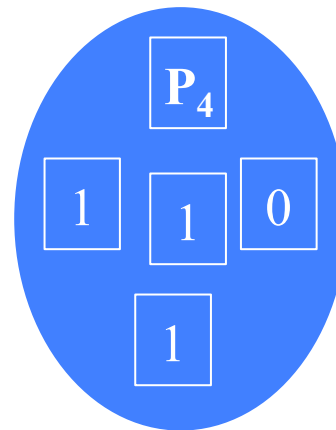
C_8	C_4	C_2	C_1	
0	0	0	0	-
0	0	0	1	P_1
0	0	1	0	P_2
0	0	1	1	I_0
0	1	0	0	P_4
0	1	0	1	I_1
0	1	1	0	I_2
0	1	1	1	I_3
1	0	0	0	P_8
1	0	0	1	I_4
1	0	1	0	I_5
1	0	1	1	I_6
1	1	0	0	I_7

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
0	1	1	0	1	1	1	1

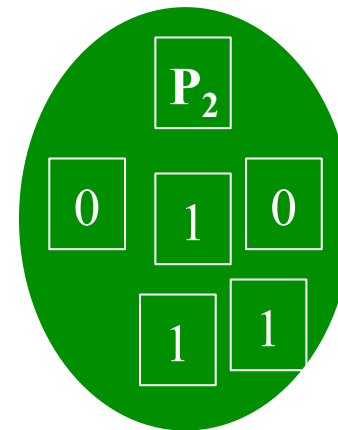
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
P_1	P_2	0	P_4	1	1	0	P_8
1	1	1	1	1	1	1	1



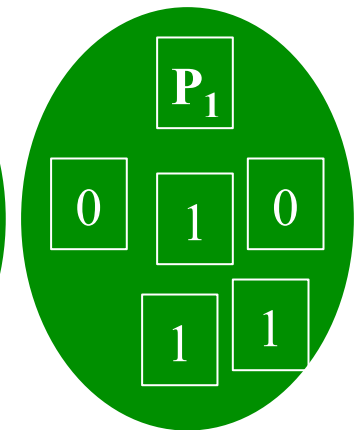
$$P_8 = 1$$



$$P_4 = 0$$



$$P_2 = 0$$



$$P_1 = 0$$

Αποκωδικοποίηση (περιττή ισοτιμία)

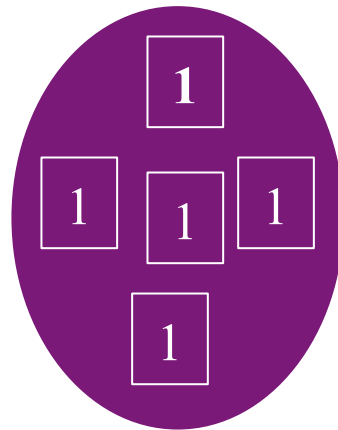
C_8	C_4	C_2	C_1	
0	0	0	0	-
0	0	0	1	P_1
0	0	1	0	P_2
0	0	1	1	I_0
0	1	0	0	P_4
0	1	0	1	I_1
0	1	1	0	I_2
0	1	1	1	I_3
1	0	0	0	P_8
1	0	0	1	I_4
1	0	1	0	I_5
1	0	1	1	I_6
1	1	0	0	I_7

0 1 1 0 1 1 1 1

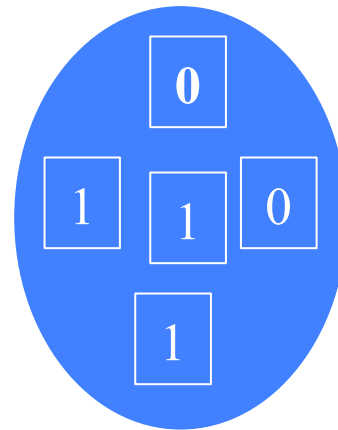


P_1 P_2 I_0 P_4 I_1 I_2 I_3 P_8 I_4 I_5 I_6 I_7

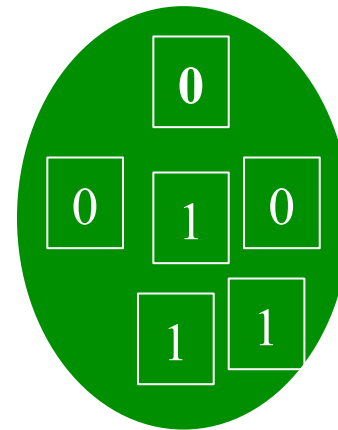
1 0 0 0 1 1 0 0 1 1 1 1



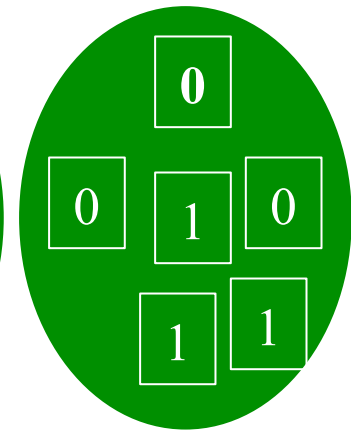
$$C_8 = 0$$



$$C_4 = 0$$



$$C_2 = 0$$



$$C_1 = 0$$

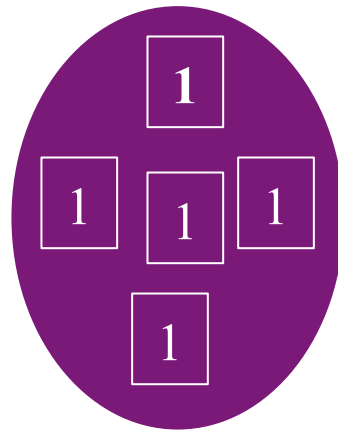
Αποκωδικοποίηση (περιττή ισοτιμία)

C_8	C_4	C_2	C_1	
0	0	0	0	-
0	0	0	1	P_1
0	0	1	0	P_2
0	0	1	1	I_0
0	1	0	0	P_4
0	1	0	1	I_1
0	1	1	0	I_2
0	1	1	1	I_3
1	0	0	0	P_8
1	0	0	1	I_4
1	0	1	0	I_5
1	0	1	1	I_6
1	1	0	0	I_7

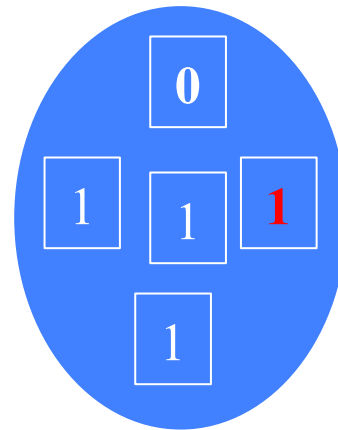
0	1	1	0	1	1	1	1
---	---	---	---	---	---	---	---



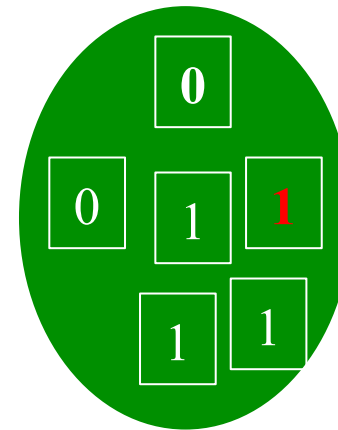
P_1	P_2	I_0	P_4	I_1	I_2	I_3	P_8	I_4	I_5	I_6	I_7
1	0	0	0	1	1	1	0	1	1	1	1



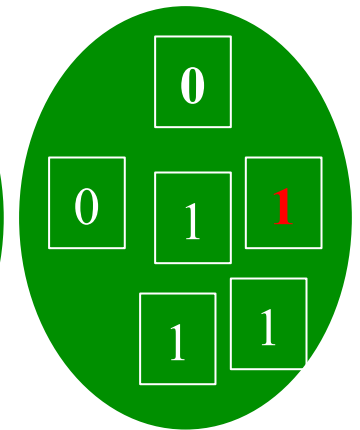
$$C_8 = 0$$



$$C_4 = 1$$



$$C_2 = 1$$



$$C_1 = 1$$

Κώδικας Hamming

Ο δυαδικός αριθμός

$$C_8C_4C_2C_1 = 0111$$

υποδεικνύει την λαθασμένη θέση (7)

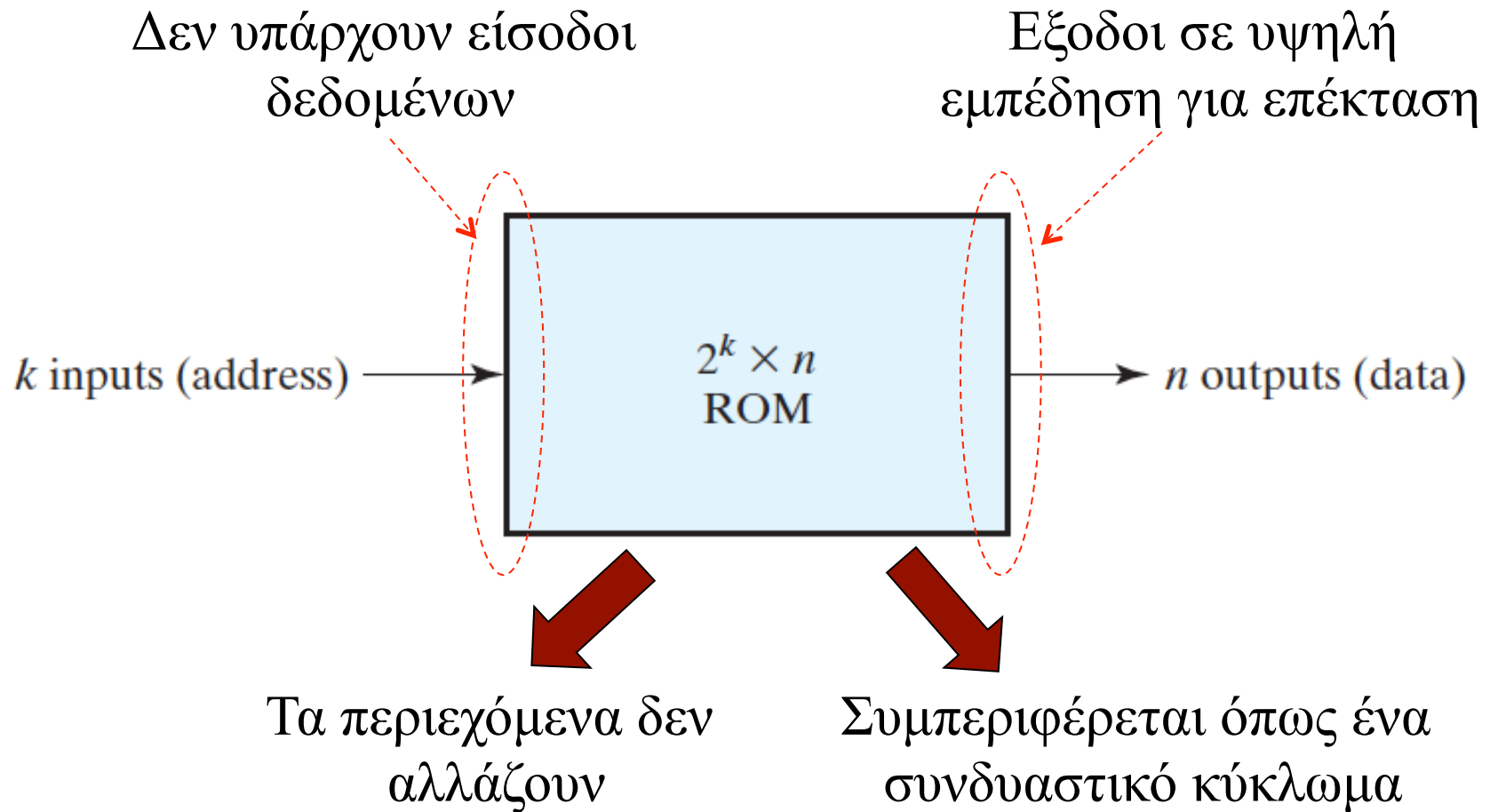
1	2	3	4	5	6	7	8	9	10	11	12
P₁	P₂	I ₀	P₄	I ₁	I ₂	I₃	P₈	I ₄	I ₅	I ₆	I ₇



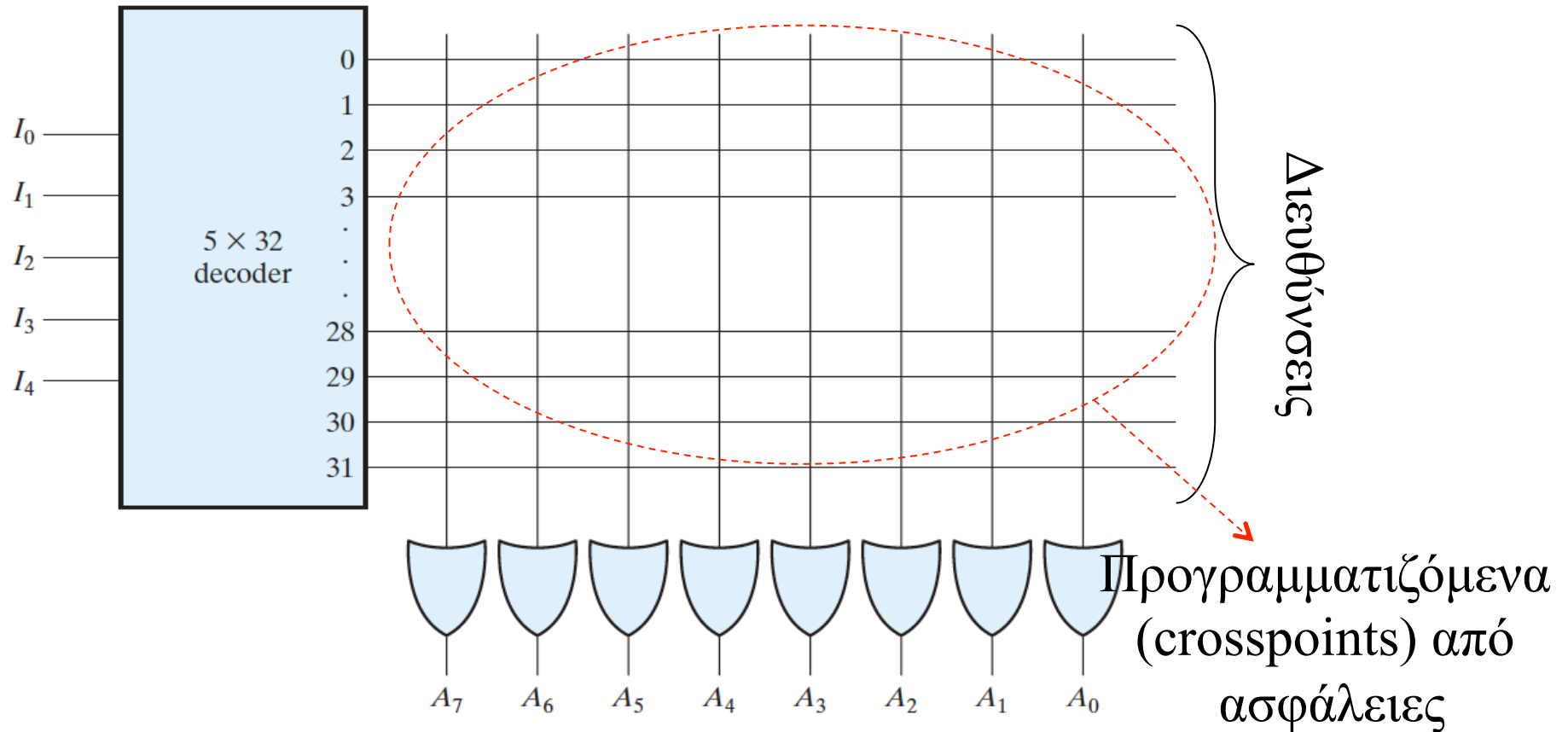
P₁	P₂	0	P₄	1	1	0	P₈	1	1	1	1
----------------------	----------------------	---	----------------------	---	---	----------	----------------------	---	---	---	---

Διόρθωση του λάθους

Μνήμες ROM



Μνήμη ROM 32x8



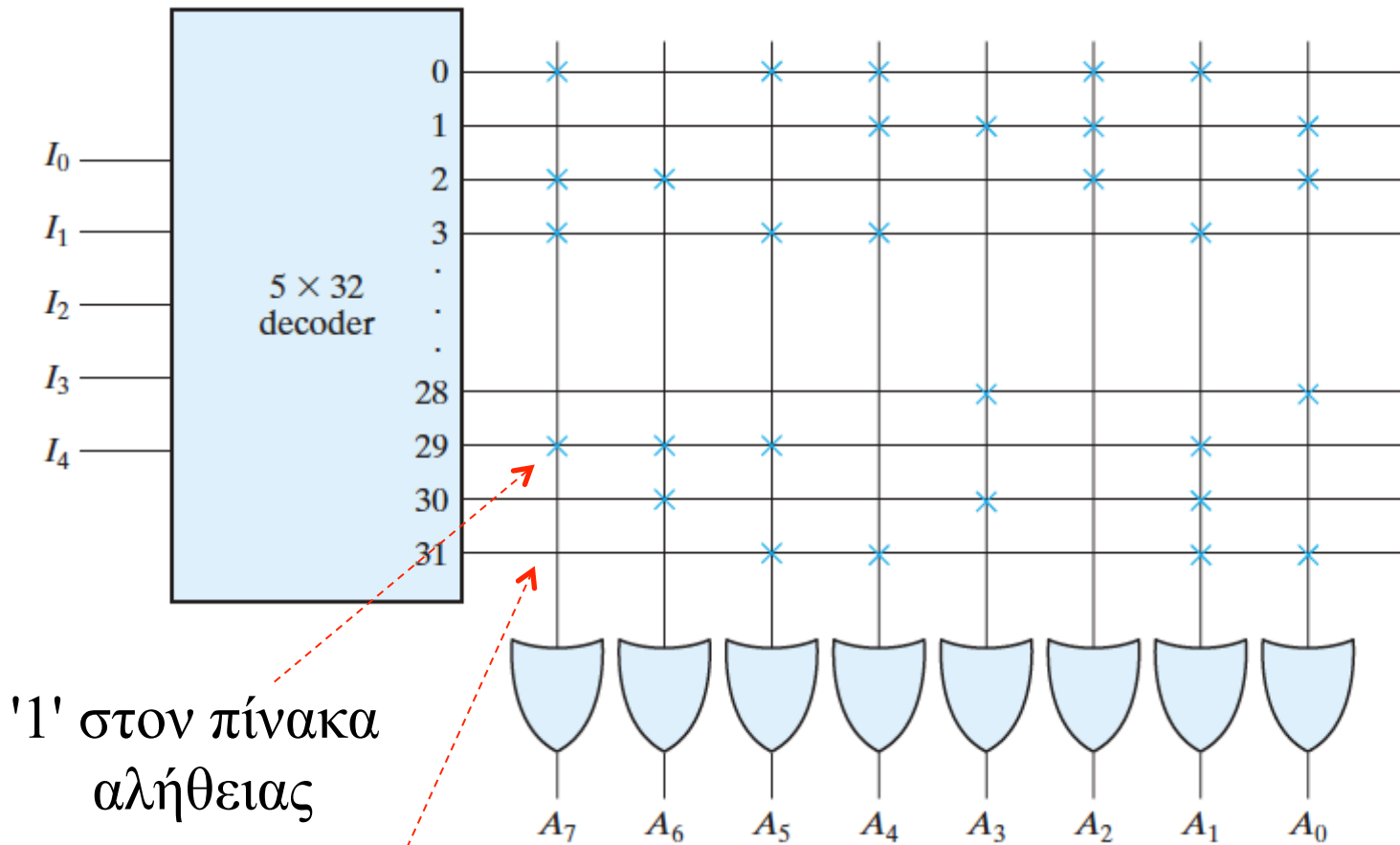
Οι συνδέσεις των πυλών Η καθορίζουν
τα περιεχόμενα των θέσεων

Μνήμη ROM 32x8

Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮							⋮			
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

Πίνακας αλήθειας μνήμης ROM

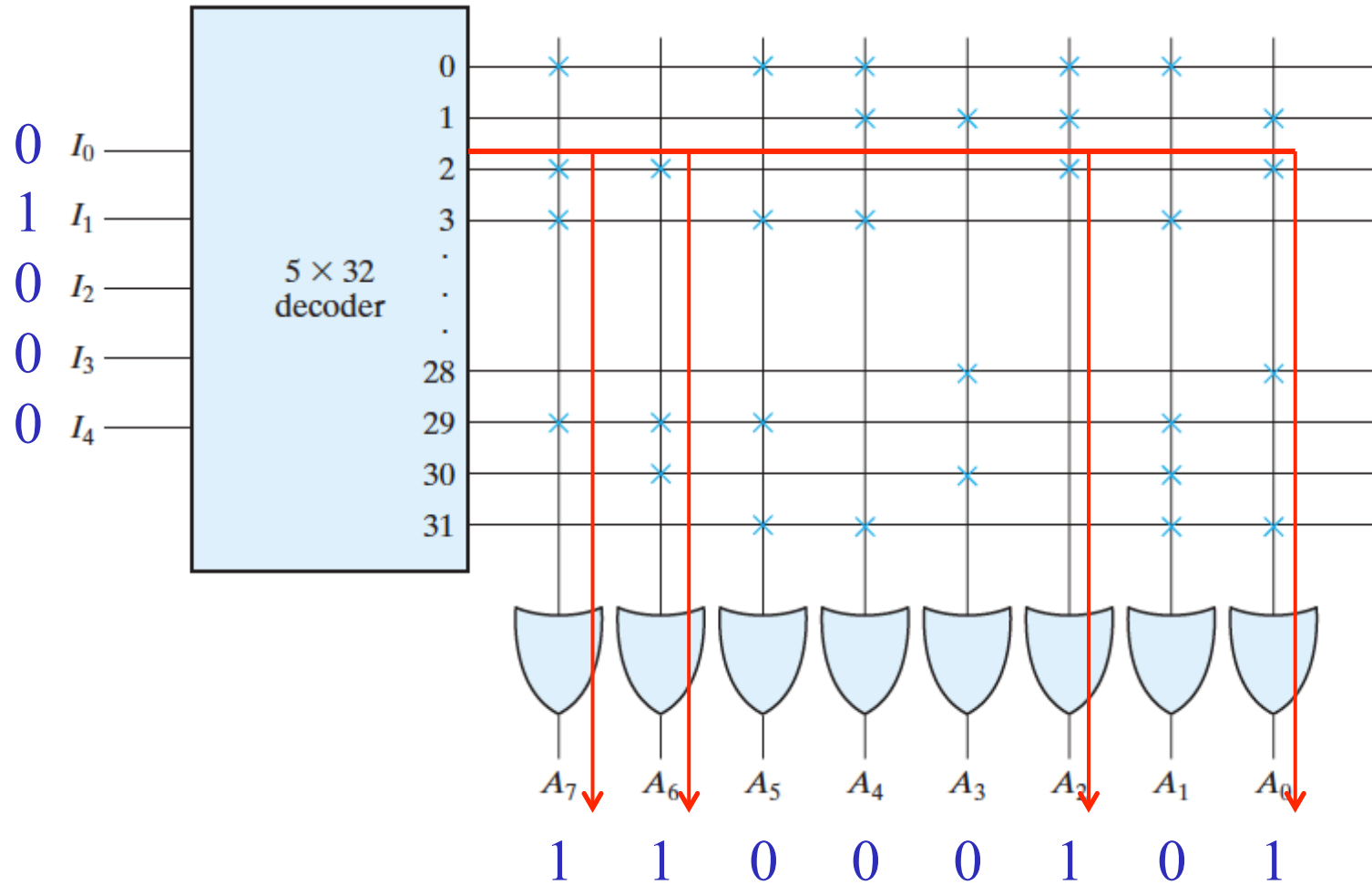
Μνήμη ROM 32x8



'1' στον πίνακα
αλήθειας

'0' στον πίνακα αλήθειας
(κάψιμο ασφάλειας)

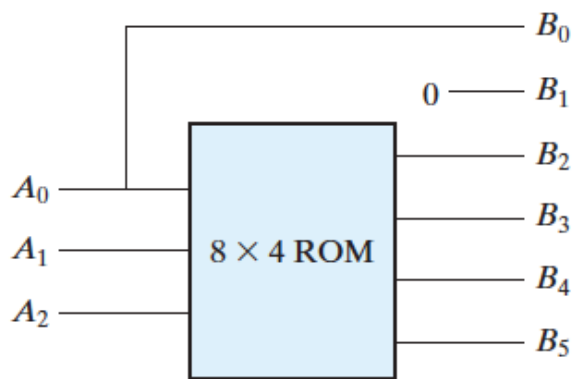
Μνήμη ROM 32x8



Υλοποίηση Συνδυαστικού Κυκλώματος

Inputs			Outputs						Decimal
A_2	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	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	1	0	0	1	49

Παράδειγμα:
κύκλωμα
τετραγωνιστή
3 bits

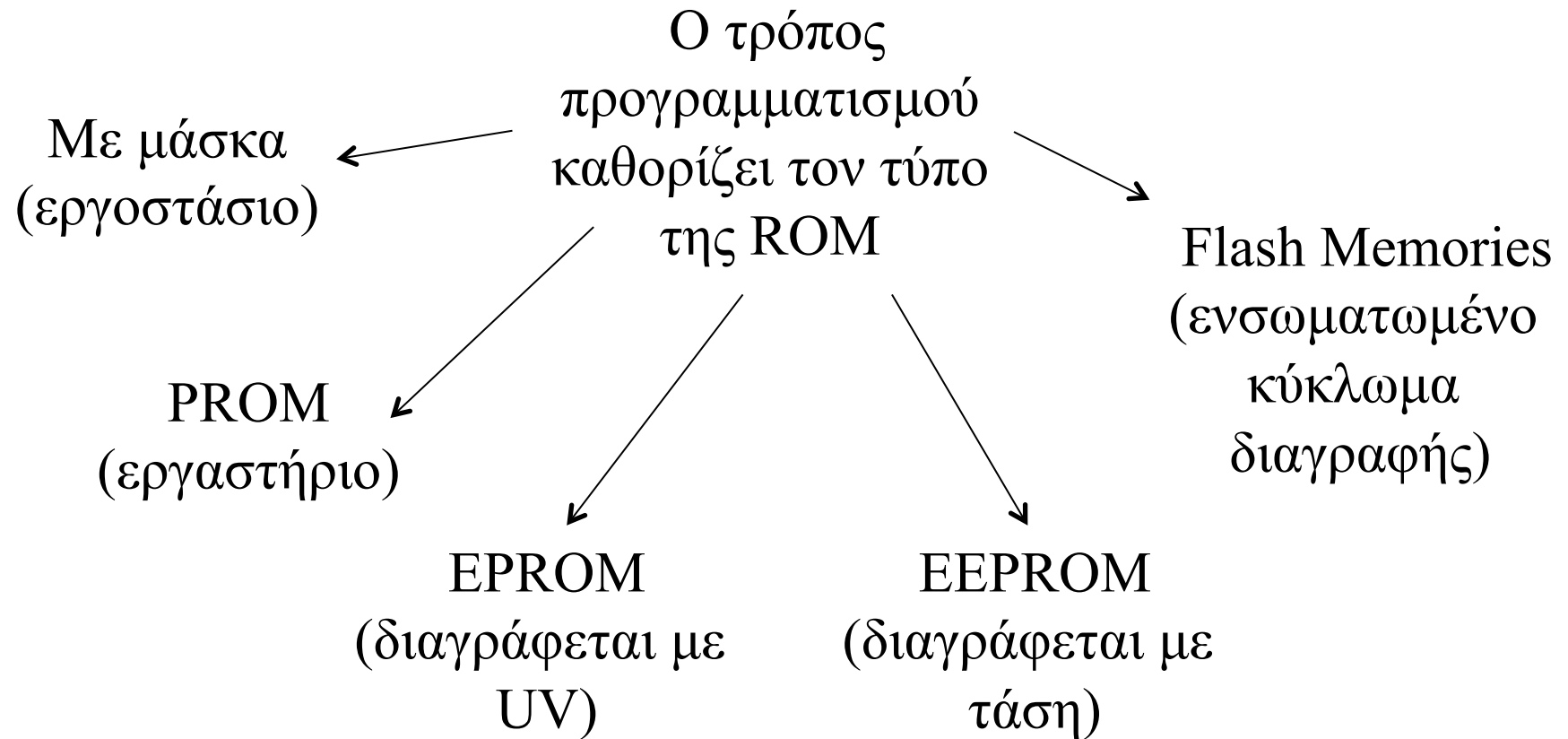


(a) Block diagram

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

(b) ROM truth table

Τύποι ROM



Συνδυαστικά PLDs



(a) Programmable read-only memory (PROM)

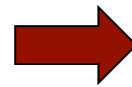


(b) Programmable array logic (PAL)



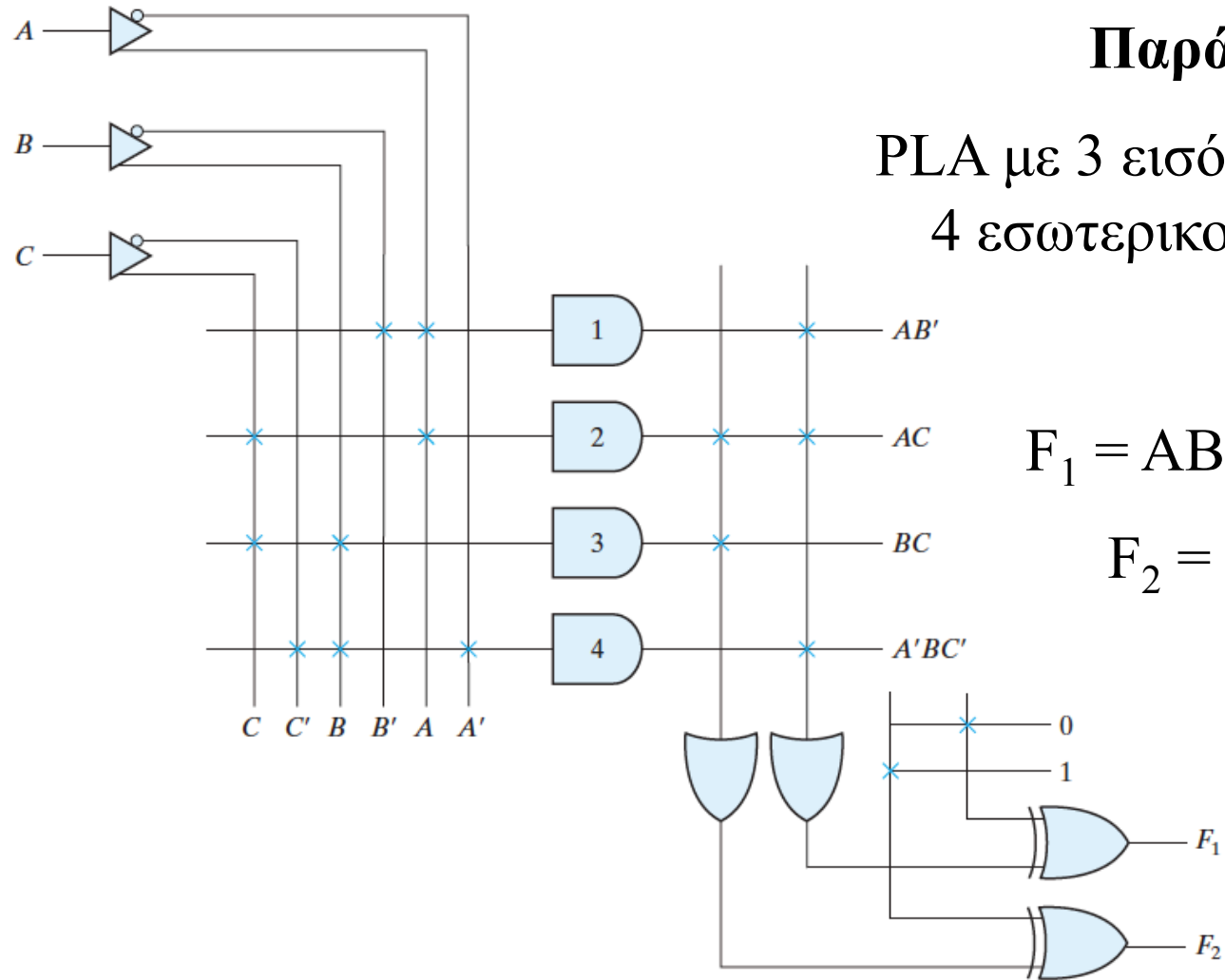
(c) Programmable logic array (PLA)

Όλοι οι ελαχιστόροι δεν
είναι απαραίτητοι



Μείωση κόστους με
PLDs

Programmable Logic Arrays



Παράδειγμα:

PLA με 3 εισόδους, 2 εξόδους,
4 εσωτερικούς παράγοντες

$$F_1 = AB' + AC + A'BC'$$

$$F_2 = (AC + BC)'$$

Πίνακας Προγραμματισμού PLA

	Product Term	Inputs			Outputs	
		A	B	C	(T)	(C)
					F ₁	F ₂
<i>AB'</i>	1	1	0	—	1	—
<i>AC</i>	2	1	—	1	1	1
<i>BC</i>	3	—	1	1	—	1
<i>A'BC'</i>	4	0	1	0	1	—

$$F_1 = AB' + AC + A'BC'$$

$$F_2 = (AC + BC)'$$

Παράδειγμα

$$w(A, B, C, D) = \Sigma(2, 12, 13)$$

$$x(A, B, C, D) = \Sigma(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$y(A, B, C, D) = \Sigma(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$z(A, B, C, D) = \Sigma(1, 2, 8, 12, 13)$$

$$w = ABC' + A'B'CD'$$

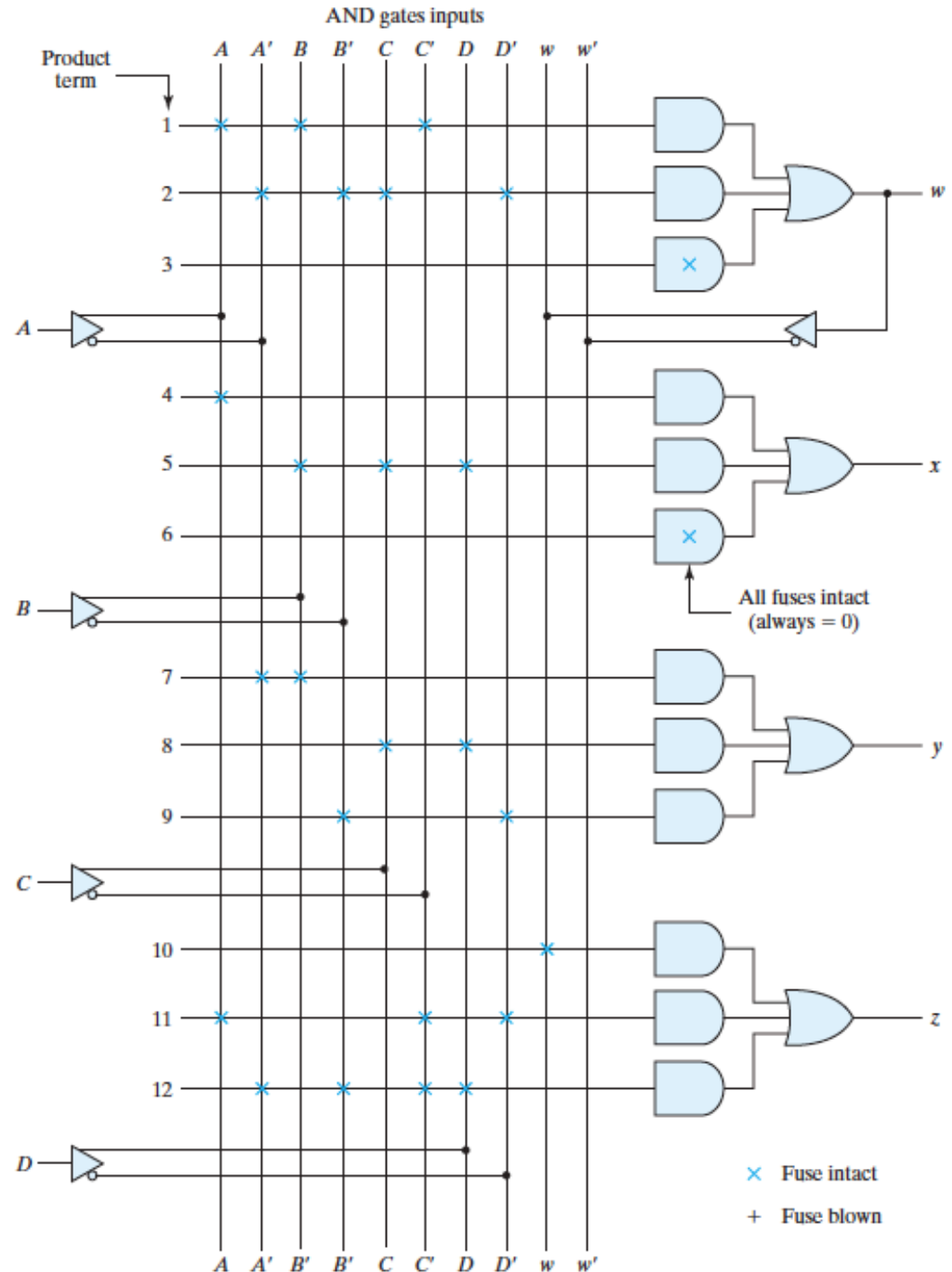
$$x = A + BCD$$

$$y = A'B + CD + B'D'$$

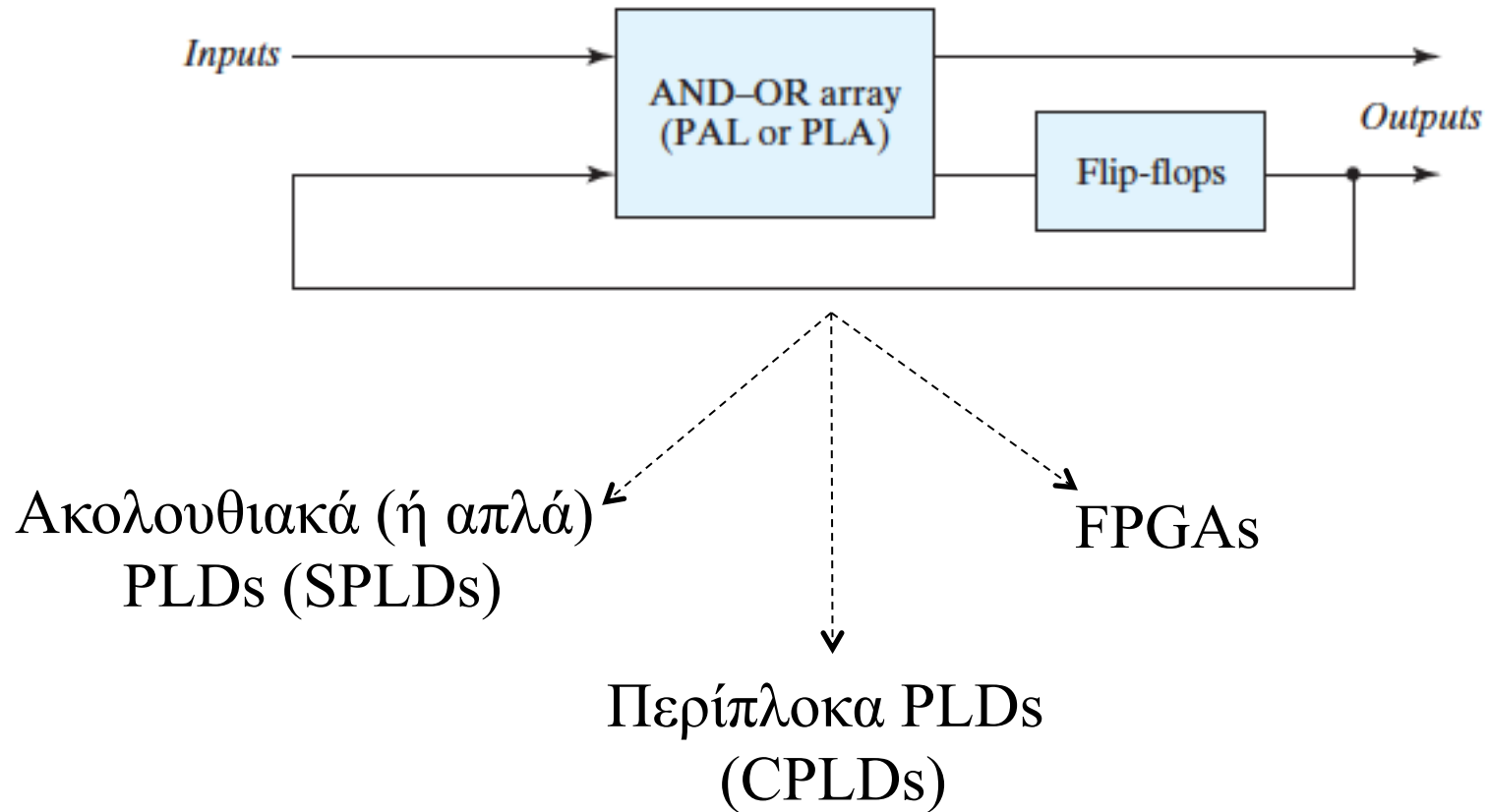
$$z = ABC' + A'B'CD' + AC'D' + A'B'C'D$$

$$= w + AC'D' + A'B'C'D$$

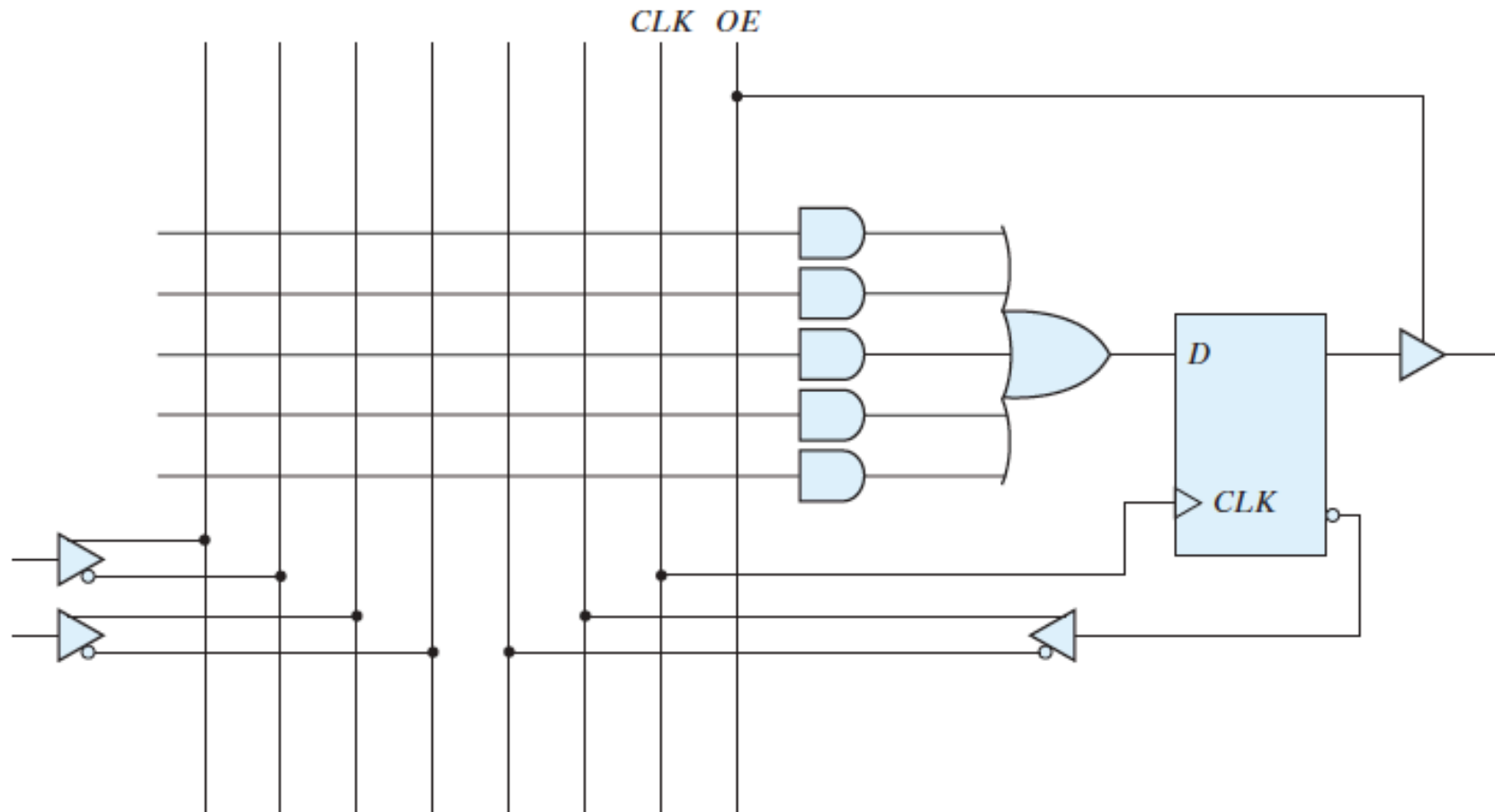
Product Term	AND Inputs					Outputs
	A	B	C	D	w	
1	1	1	0	-	-	$w = ABC' + A'B'CD'$
2	0	0	1	0	-	
3	-	-	-	-	-	
4	1	-	-	-	-	$x = A + BCD$
5	-	1	1	1	-	
6	-	-	-	-	-	
7	0	1	-	-	-	$y = A'B + CD + B'D'$
8	-	-	1	1	-	
9	-	0	-	0	-	
10	-	-	-	-	1	$z = w + AC'D' + A'B'C'D$
11	1	-	0	0	-	
12	0	0	0	1	-	



Ακολουθιακές Προγραμματιζόμενες Συσκευές

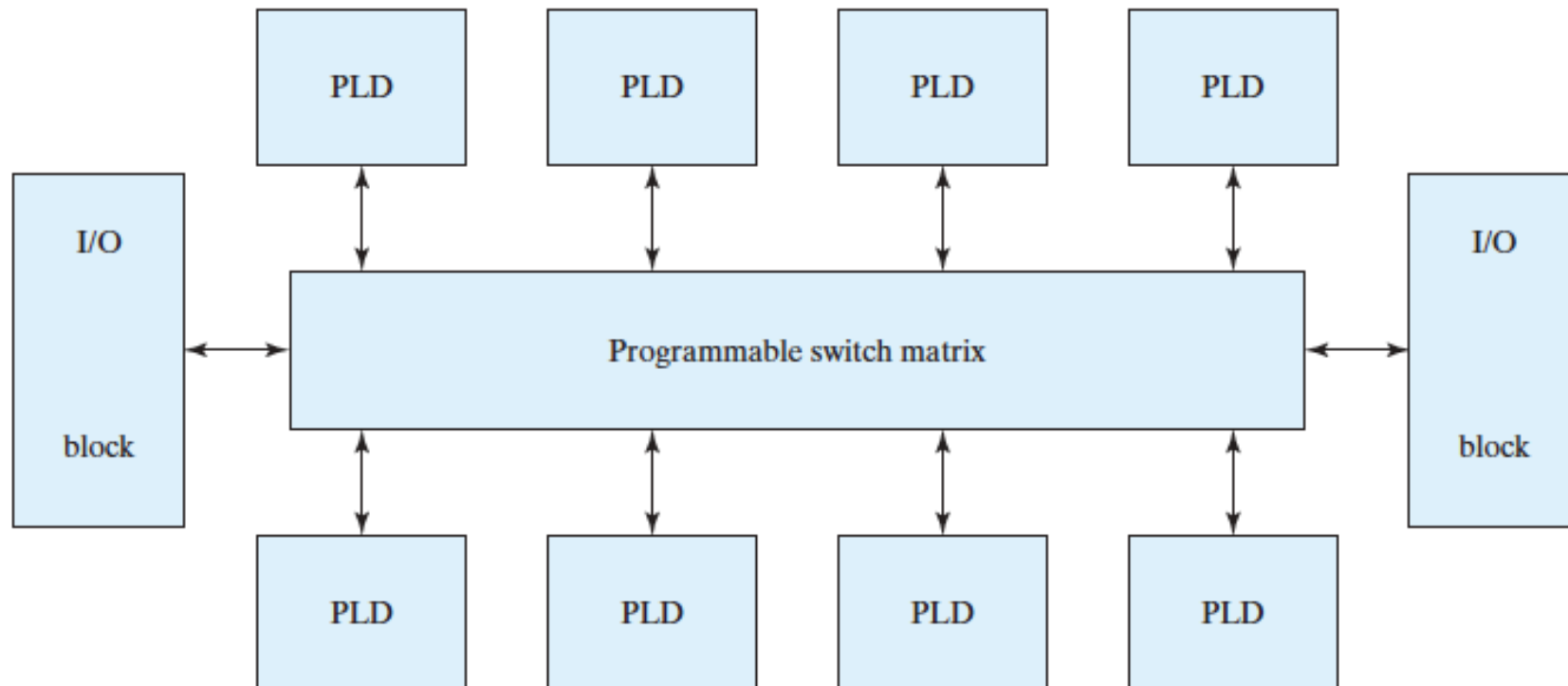


SPLDs (macrocell)



Ένα τυπικό SPLD έχει 8-10 macrocells

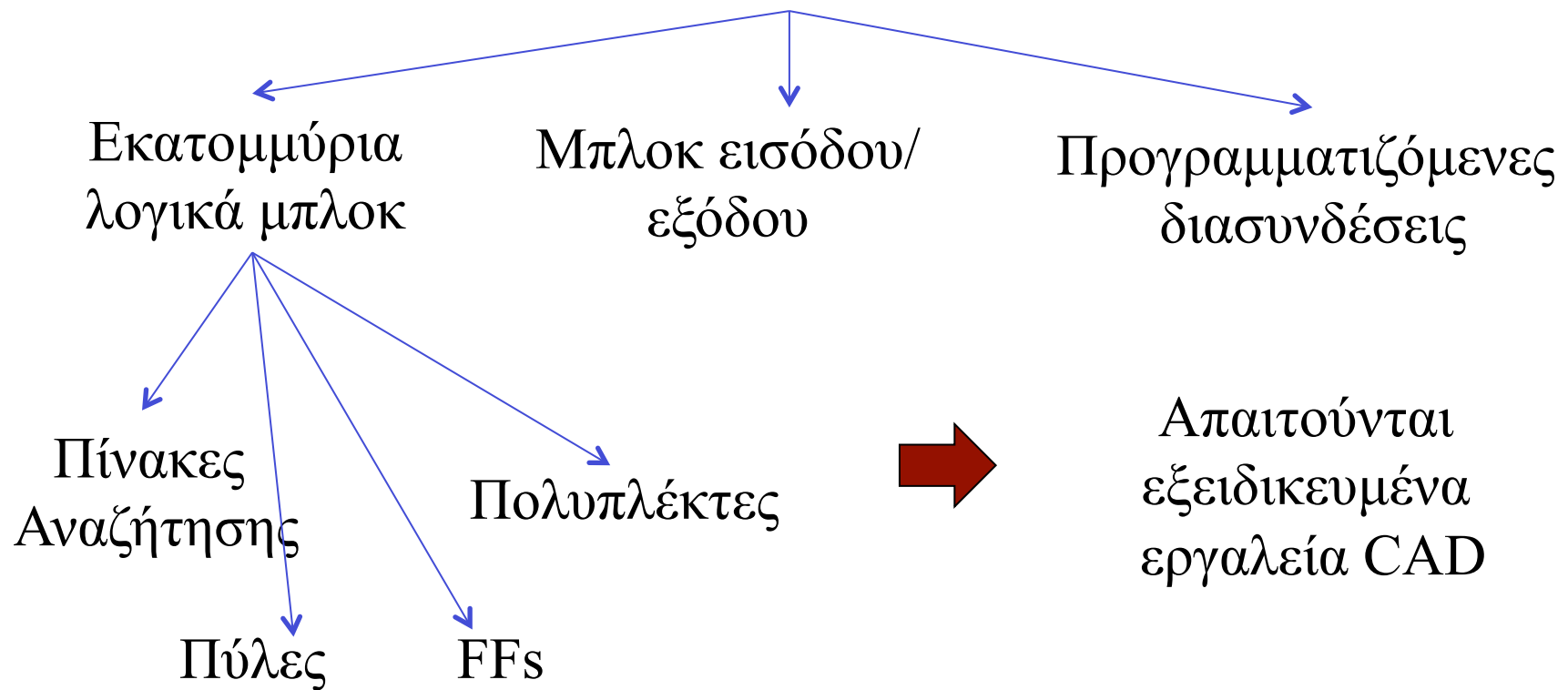
CPLDs



Παρέχουν περισσότερες δυνατότητες προγραμματισμού
λογικής

FPGAs

Αποτελούν VLSI κυκλώματα που προγραμματίζονται στο πεδίο εφαρμογής

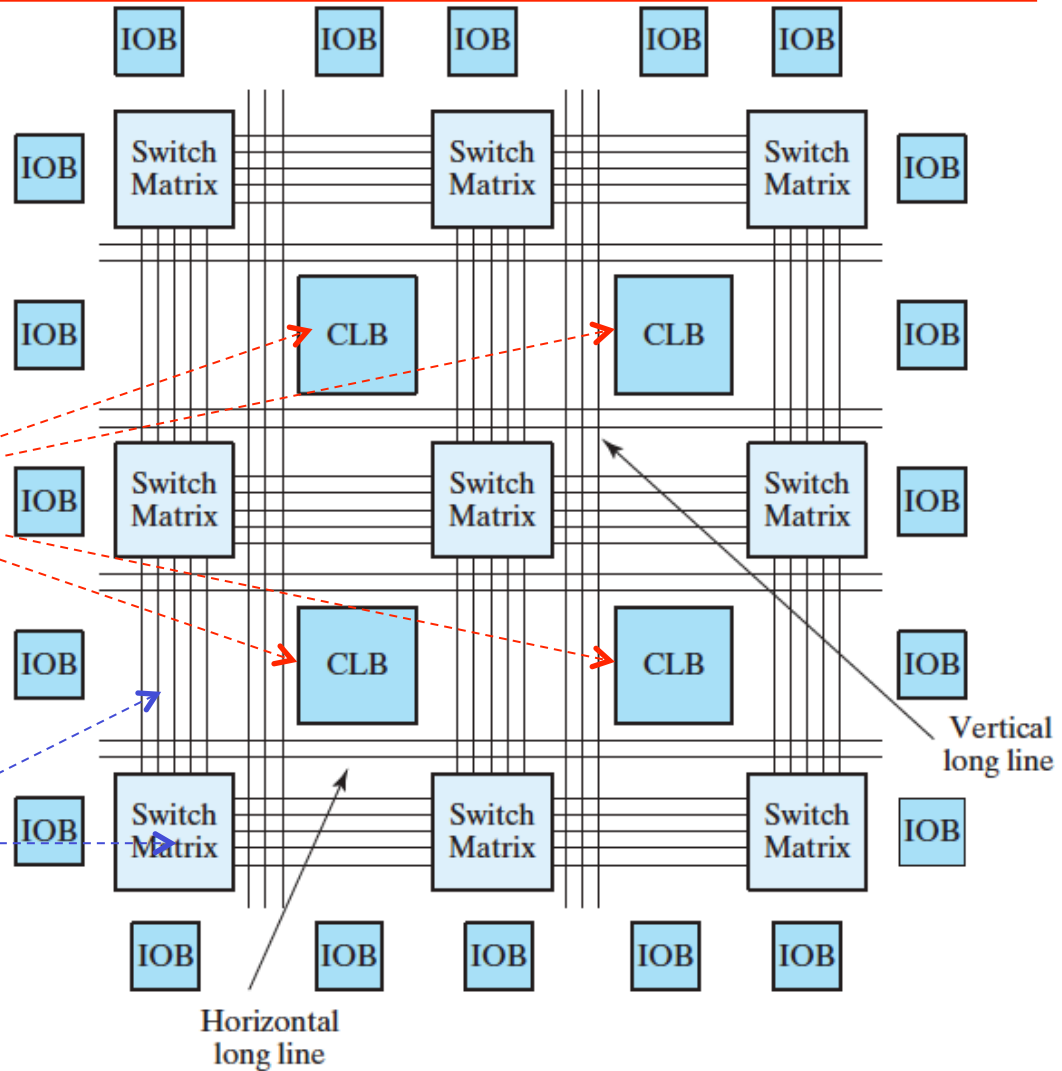


Xilinx FPGAs

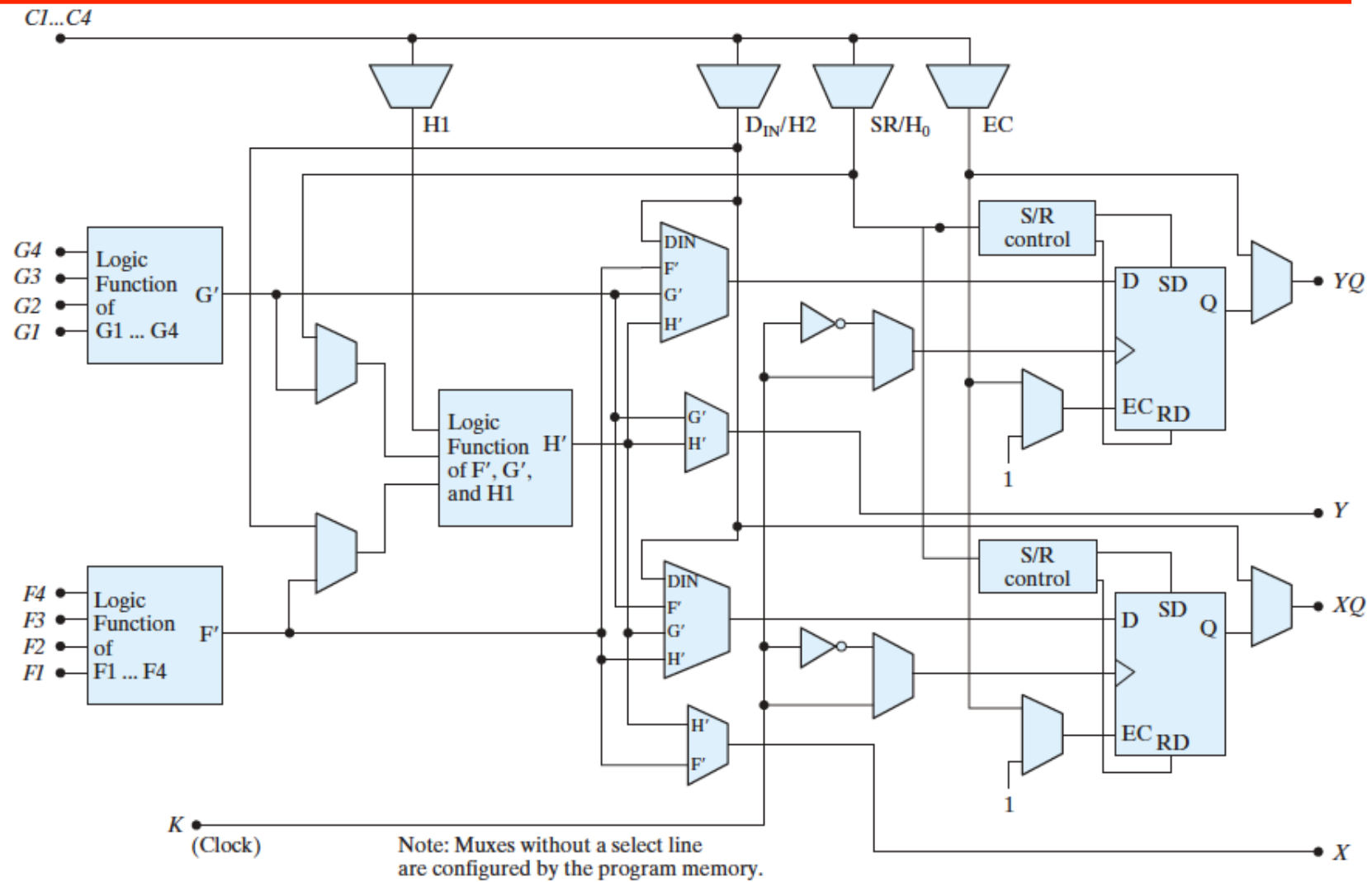
*Η τελευταία
οικογένεια είναι
η Spartan*

Επεξεργασία
δεδομένων

Διασύνδεση
επεξεργαστικών
μονάδων

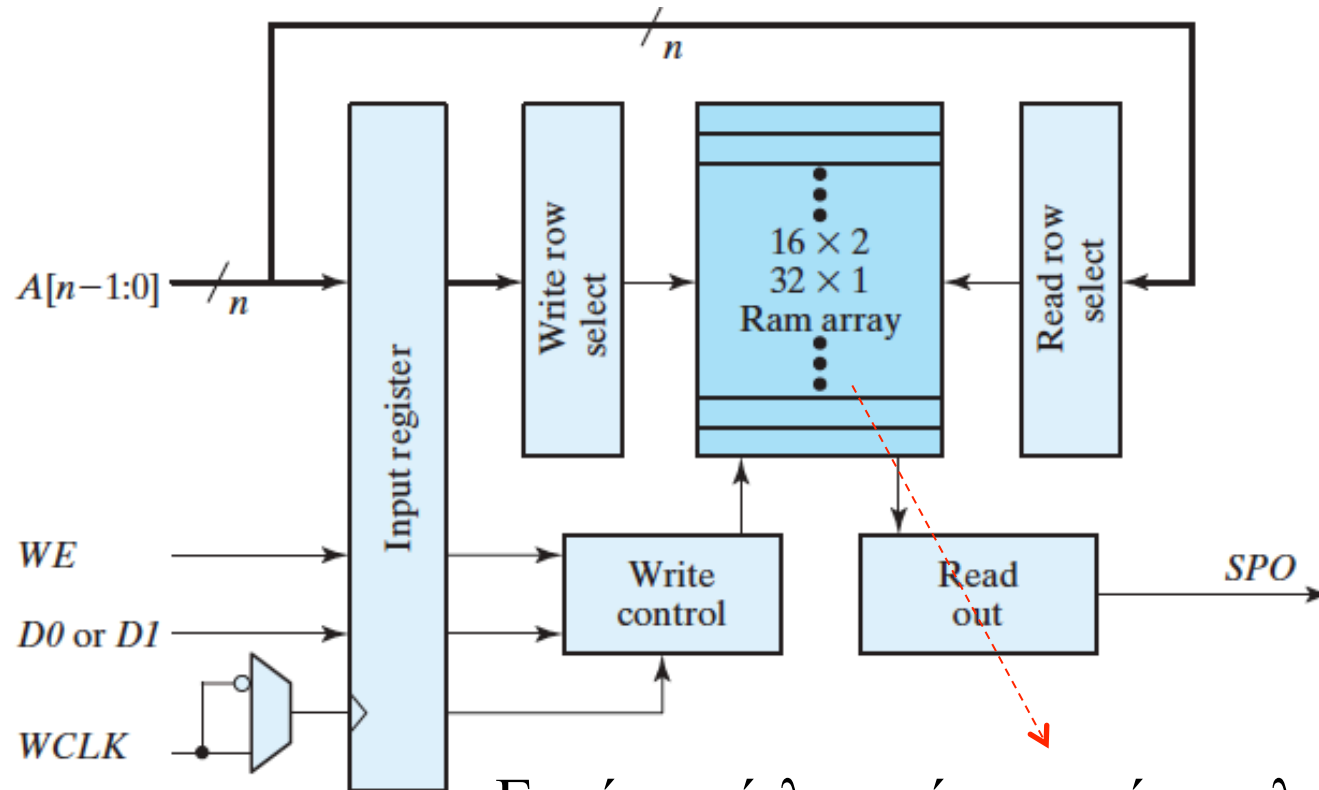


Configurable Logic Block



Configurable Logic Block

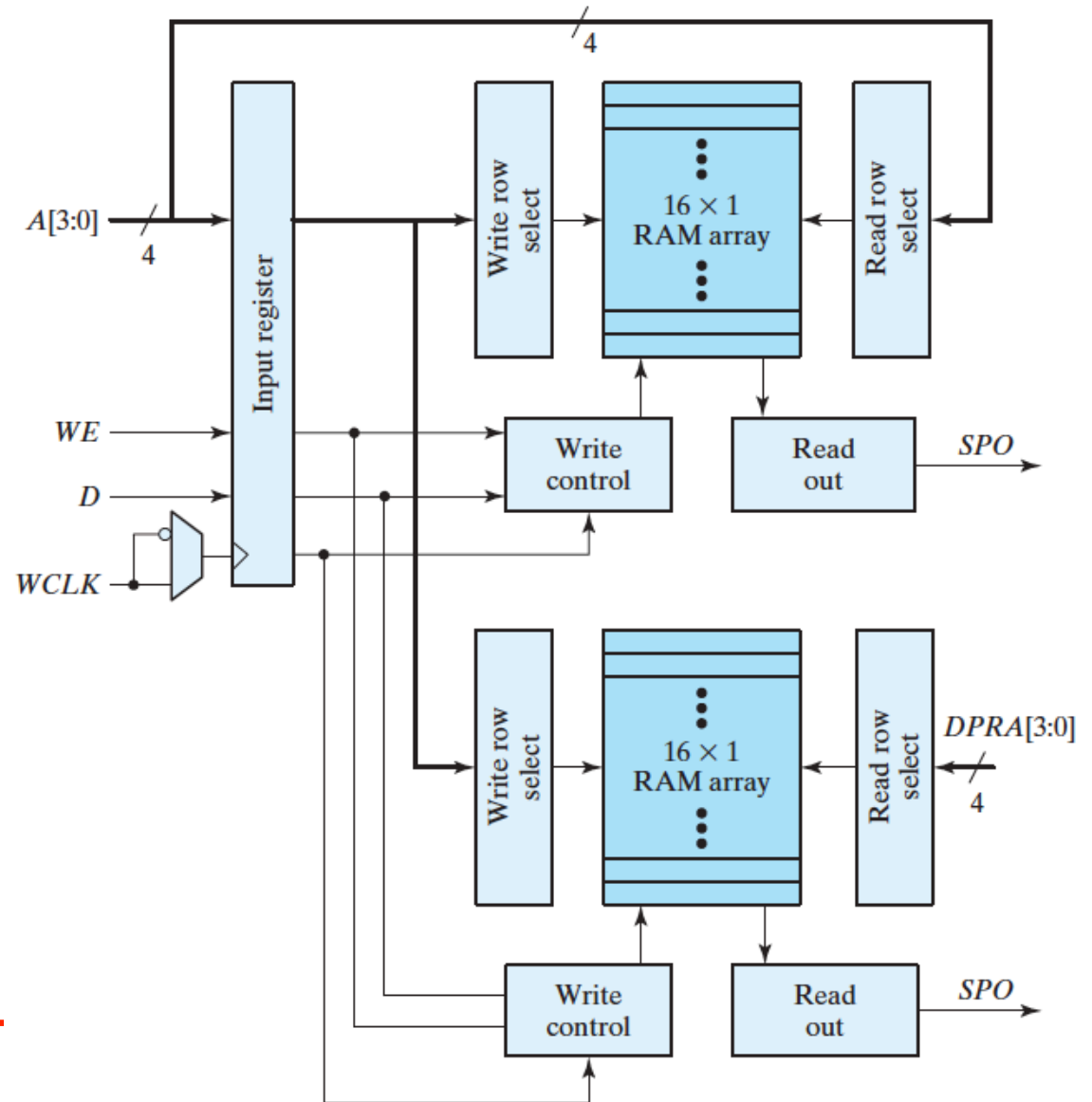
Διαμόρφωση CLBs για δημιουργία SRAMs



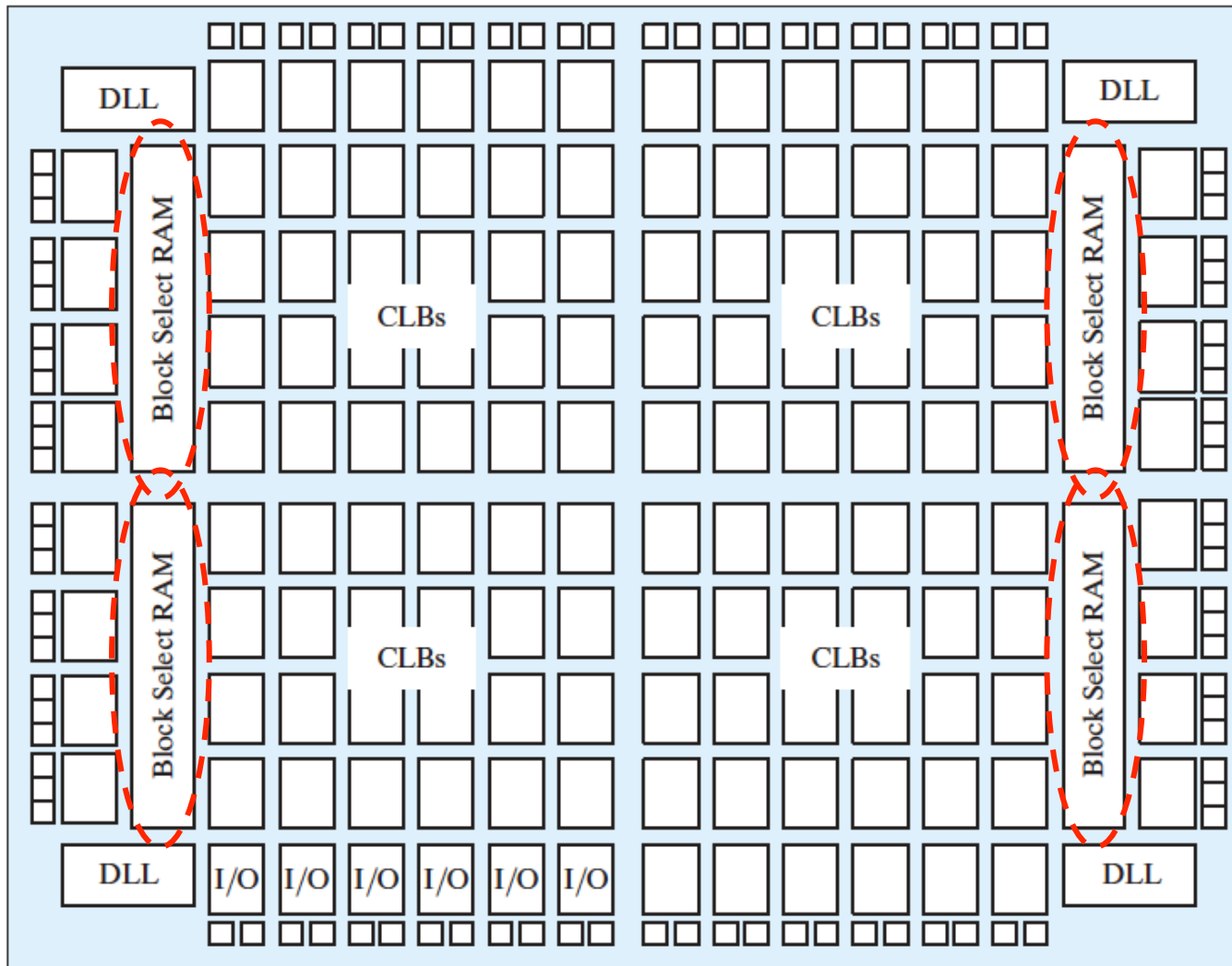
Εκτός από λογική μπορεί να υλοποιήσει και μνήμη με τα κατάλληλα περιφερειακά

Configurable Logic Block

Διαμόρφωση CLBs
για δημιουργία Dual-
Port SRAMs



Αρχιτεκτονική Spartan II



Xilinx Virtex

Κάθε επόμενη
οικογένεια
προσθέτει
δυνατότητες

