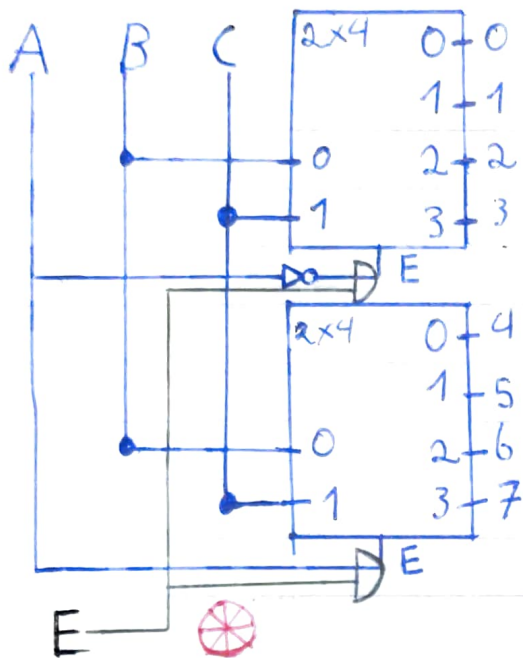


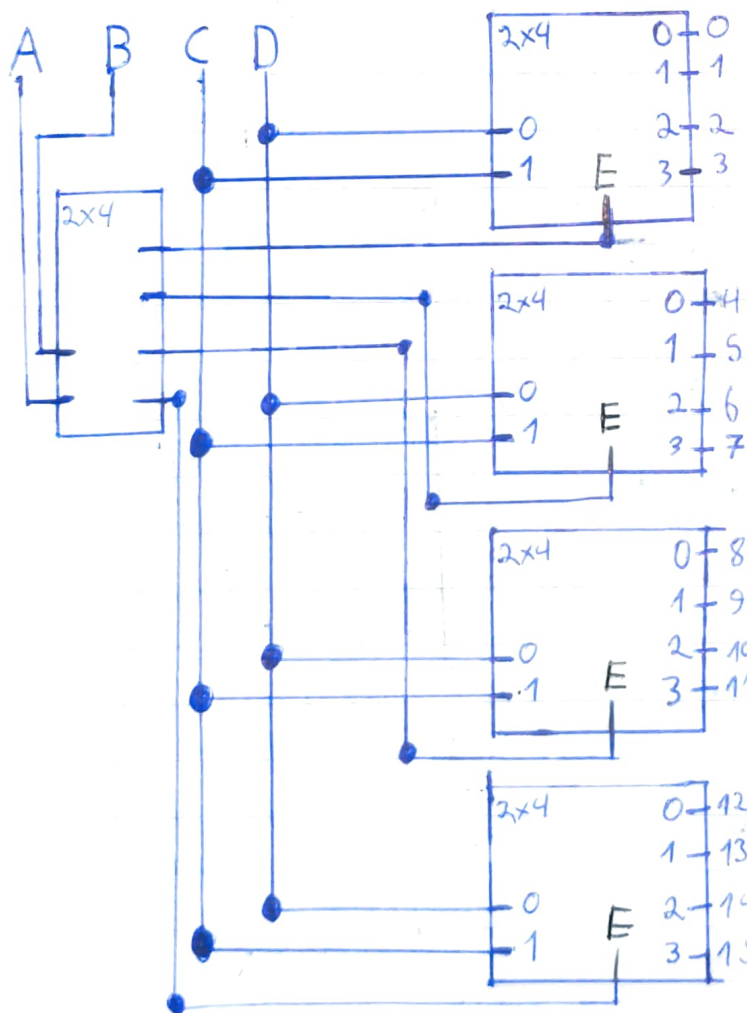
Αρχιτεκτονική Διάλεξη 4

Συνδυαστικά κυκλώματα



• Όταν το E είναι 0, δεν δουλεύει ο Decoder
Όταν το E είναι 1, δουλεύει ο Decoder

• Αν το $A=0$ μιλάμε για το πρώτο Decoder
• Αν το $A=1$ μιλάμε για το δεύτερο Decoder



← Τα 2 πρώτα bits $AB=00$

← Τα 2 πρώτα bits $AB=01$

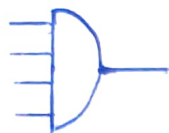
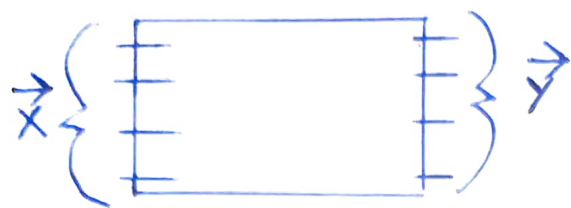
← Τα 2 πρώτα bits $AB=10$

← Τα 2 πρώτα bits $AB=11$

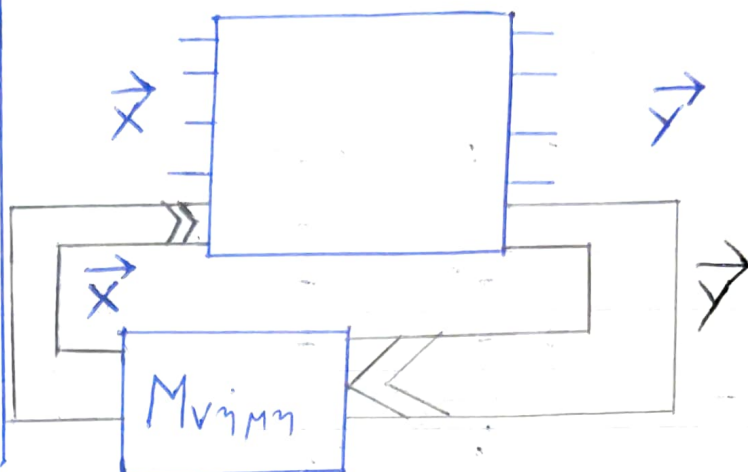
Μια έξοδος θα είναι 1 (η πρώτη/Α) αφού θα δουλέψει ο 3ος Decoder

Θα μπορούσα να ~~εξάγω~~ ~~εξάγω~~ 1 και Decoder 1x2 στο πρώτο κύκλωμα

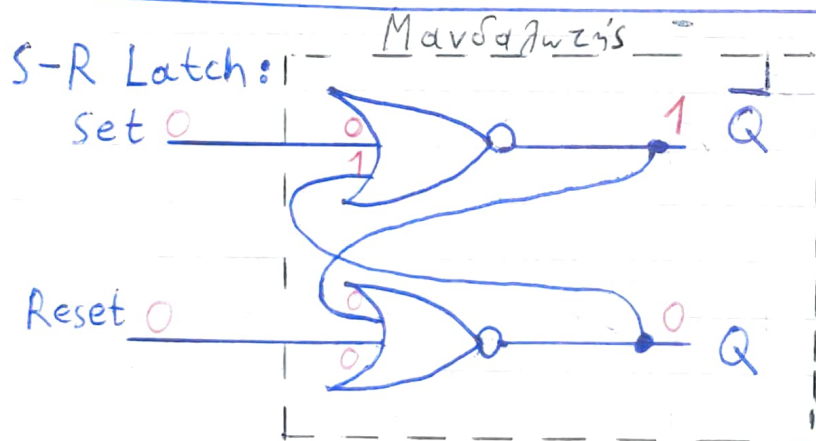
Συνδυαστικά κυκλώματα



Ακολουθιακά κυκλώματα



- 1) Μανδαλωτές (latch)
- 2) Flip-Flop (μανδαλωτής που έχει clock είσοδο)
clock : Προσδιορίζει ποτε αλλάζει
- 3) Καταχωρητές



Σε κανονική λειτουργία και τα 2 είναι 0 στο Input

Για να αλλάξω την τιμή του Μανδαλωτή κενά ή το Set ή το Reset από 0 σε 1

Πίνακας αληθείας για SR

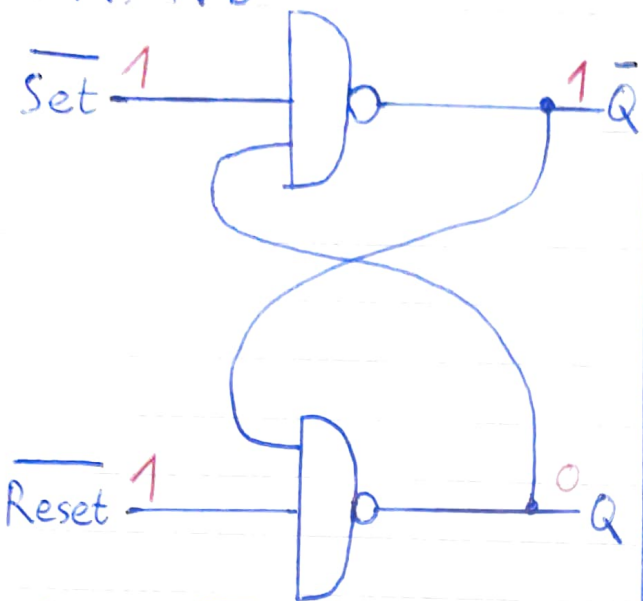
S	R	Q+
0	0	Q+
1	0	1
0	1	0
1	1	Απροσβ.

Q = τιμή ίδια με αυτήν που είχε πριν μπει

Εάν, $S=R=1$
τότε τα Q δεν είναι άλλο αντιστροφή οπότε Απροσδιοριστό (γιατί $Q=\bar{Q}$)

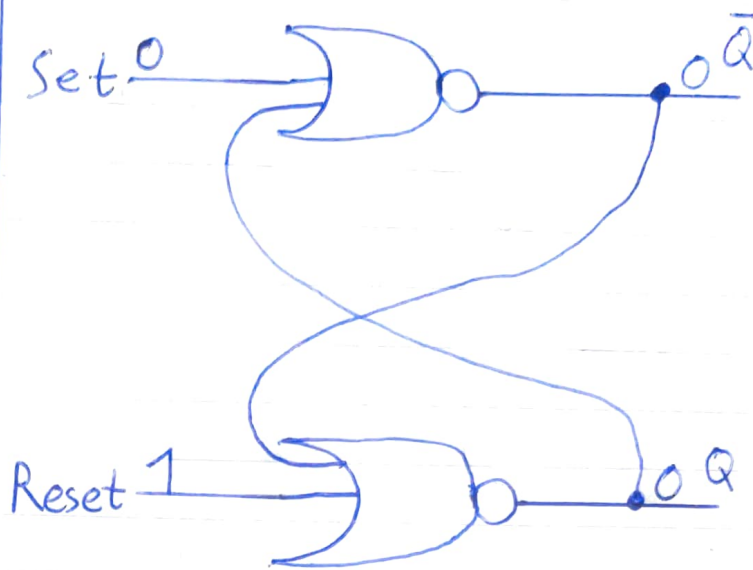
Παραδείγματα με NAND και NOR

NAND

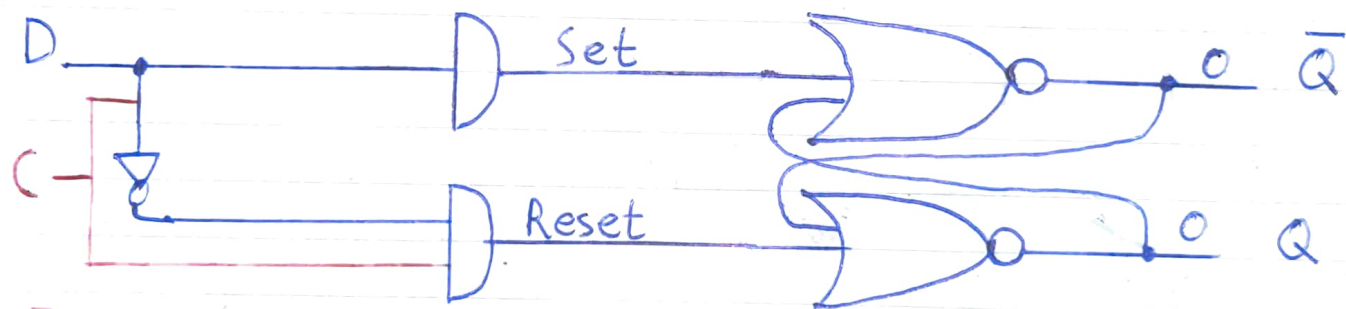


Αφού $\overline{set}=1$ και $\overline{reset}=1$
 τότε $Set=0$ και $reset=0$

NOR



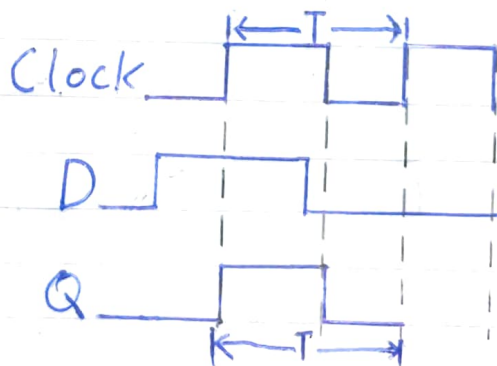
- Πως να αποφύγουμε την απροσδιοριστή μορφή: με clock
 Συγχρονό (ακολουθούν αλλαγές ρολογιού)

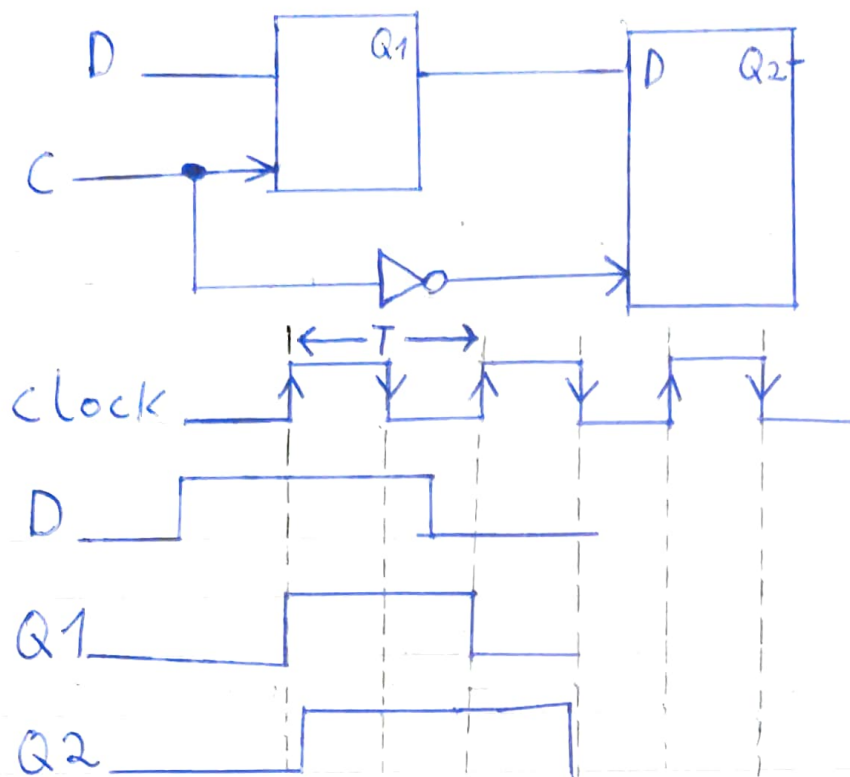


Για να έχουμε Input 0-0 βάζουμε ένα clock. Έχουμε
 πάει σε Flip-Flop δεν είναι άλλο μονοκύτταρο

- Χρονοδιάγραμμα clock:

$$f = \frac{1}{T}$$



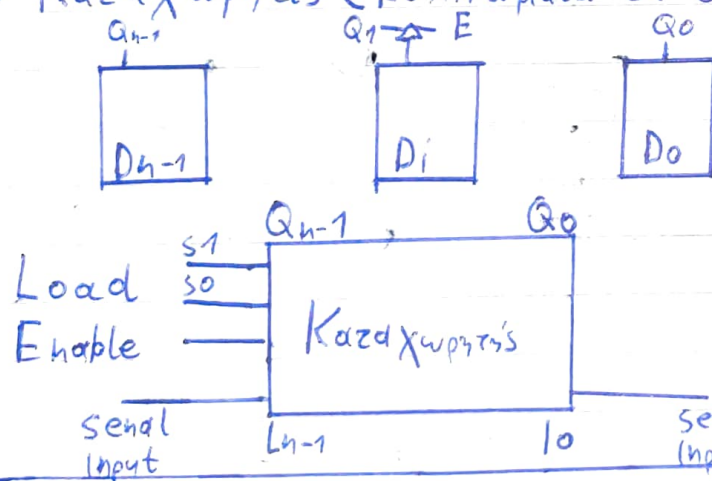


$\begin{array}{ c } \hline F \\ \hline \end{array}$	$\begin{array}{ c } \hline Q \\ \hline \end{array}$
$\begin{array}{ c } \hline K \\ \hline \end{array}$	$\begin{array}{ c } \hline \bar{Q} \\ \hline \end{array}$

D	Q
0	0
1	1

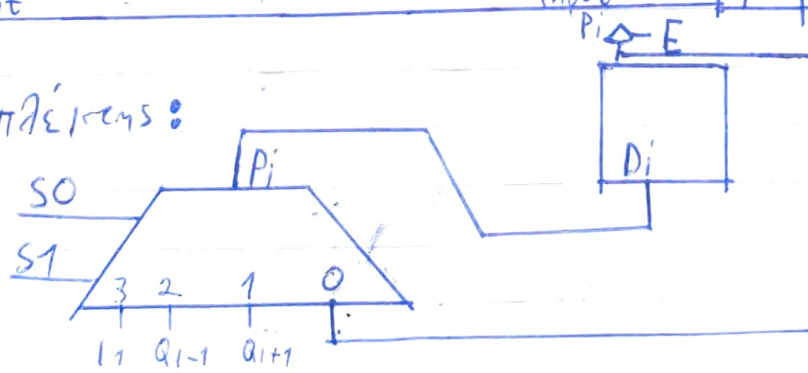
S	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	\bar{Q}_t

- Καταχωρητές (κυκλώματα τα οποία αποτελούνται από πολλά Flip-Flops)



S1	S0	
0	0	Δεν αλλάζει
0	1	Μετατόπιση δεξιά
1	0	Μετατόπιση αριστερά
1	1	Ρύθμιση

Πολυπλέκτες:



Πολυπλέκτες

- Πολυπλέξια: Επιλέχουμε από πολλές γραμμές εισόδου και βγάζουμε μια έξοδο

- Πολυπλέκτης: Έχει ενσωματωμένο αποκωδικοποιητή

Μεγεθος Πολυπλέκτη: $2^n \times 1$ όπου:

n = Γραμμές Επιλογής } $n \times 2^n$ Αποκωδικοποιητής για επιλογή
 2^n = Γραμμές Εισόδου

Παράδειγμα: - Πολυπλέκτης $4 \times 1 = 2^2 \times 1$

2 Γραμμές Επιλογής } Για να επιλέξω
 4 Γραμμές Εισόδου } χρειαζομαι DEC 2×4

- Πολυπλέκτης $8 \times 1 = 2^3 \times 1$

3 Γραμμές Επιλογής } Για να επιλέξω
 8 Γραμμές Εισόδου } χρειαζομαι DEC 3×8

■ Πίνακας Αληθείας

Έστω ο MUX 4×1

- 4 Γραμμές εισόδου

- 1 Έξοδος

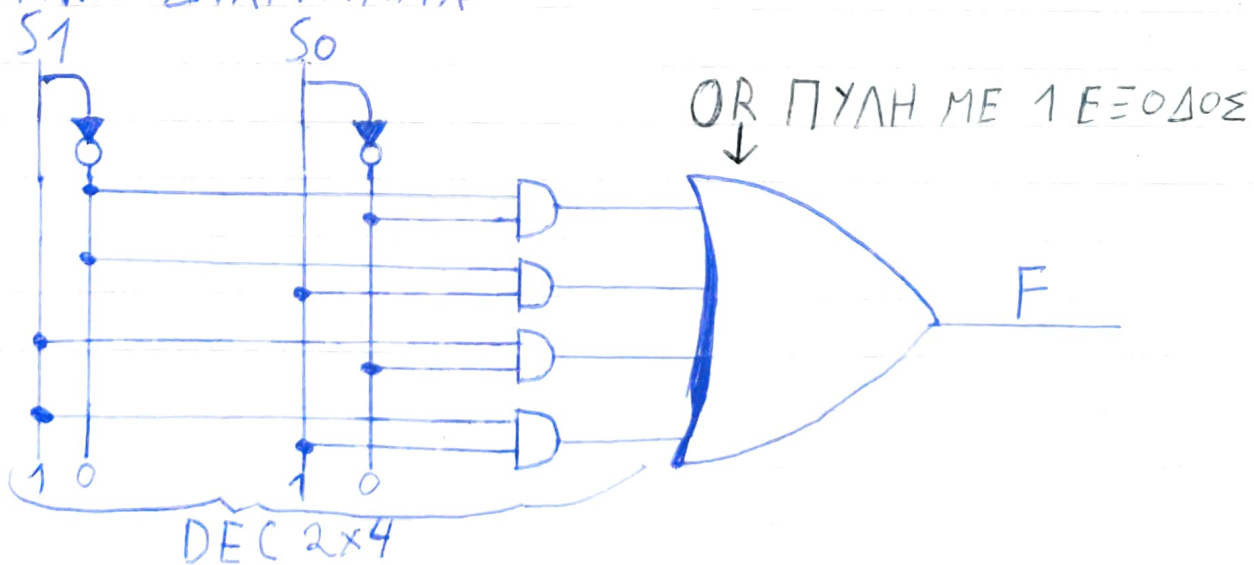
- 2 Γραμμές Επιλογής (S_1, S_0)

Χρησιμοποιούνται για αποκωδικοποίηση δηλαδή επιλογή της γραμμής που θα βγει στην έξοδο.

- Οι Γραμμές επιλογής αποκωδικοποιούν το σήμα εισόδου I_i που θα βγει στην F

• ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ

ΓΡΑΜΜΕΣ ΕΠΙΛΟΓΗΣ	ΕΞΟΔΟΣ
S_1	S_0
0	0
0	1
1	0
1	1
	I_0
	I_1
	I_2
	I_3



Στον Αποκωδικοποίηση, κάθε σελιμή μια απο τις πύλες 0-3 δίνει έξοδο 1. Οι άλλες δίνουν 0.

- Έστω $S_1, S_0 = 0, 0 \Rightarrow$ Η Πύλη 0 έχει δύο σήματα εισόδου ίσα με 1. Η έξοδος της εξαρτάται από το $I_0 (1 \cdot 1 \cdot I_0 = I_0)$

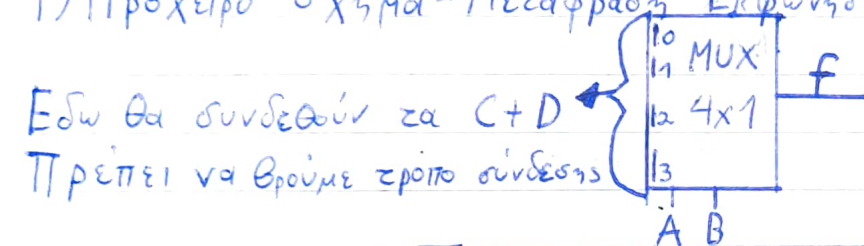
Οι πύλες 1-3 έχουν έξοδο 0. Η OR έχει εισόδους $I_0 + 0 + 0 + 0 = I_0$

Παράδειγμα 1:

• Δίνετε η $F(A, B, C, D) = \Sigma(0, 1, 2, 3, 5, 6, 7, 13, 15)$

Να υλοποιηθεί με πολυπλέκτη 4×1 όπου τα σήματα A, B συνδέονται με τις γραμμές επιλογής του πολυπλέκτη και τα C, D με τις γραμμές εισόδου Βήματα:

1) Προχειρό σχήμα - Μετάφραση Εισαγωγής



2) Πινάκας Αληθείας	Είσοδοι				Έξοδος	
	A	B	C	D	F	
$I_0 = 1$	0	0	0	0	1	$A, B = 00$ $F = I_0$
	0	0	0	1	1	
	0	0	1	0	1	
	0	0	1	1	1	
$I_1 = C + D$	0	1	0	0	0	$A, B = 01$ $F = I_1$
	0	1	0	1	1	
	0	1	1	0	1	
	0	1	1	1	1	
$I_2 = 0$	1	0	0	0	0	$AB = 10$ $F = I_2$
	1	0	0	1	0	
	1	0	1	0	0	
	1	0	1	1	0	
$I_3 = D$	1	1	0	0	0	$AB = 11$ $F = I_3$
	1	1	0	1	1	
	1	1	1	0	0	
	1	1	1	1	1	

- $A=0, B=0$

Όταν $AB=00$, τότε $F=I_0$. Όμως το I_0 συνδέεται με τα C, D . Αρα πρέπει να συνδέσουμε το I_0 με έναν συνδυασμό πυλών που θα έχουν είσοδο τα C, D

• Βλέπουμε ότι όταν $AB=00 \Rightarrow F=1$ για κάθε συνδυασμό των C, D . Αν συνδέσω το I_0 με 1 τότε $AB=00, F=1$

Συμπέρασμα: $I_0=1$

- $A=0, B=1$

Όταν $AB=01$, τότε $F=I_1$. Επομένως πρέπει να βρούμε ποιον λογικό συνδυασμό των C και D θα συνδεθούμε στην I_1

Δηλαδή αν $AB=01$ τότε $F=1$ όταν $C, D=011$ $C, D=0111$ $CD=1011$

Απλοποίηση Karnaugh:

$C \backslash D$	0	1
0		1
1	1	1

$\Rightarrow C+D$ Διοτι Ομάδα 1: D
Ομάδα 2: C

- $A=1, B=1$

Όταν $AB=11$ τότε $F=I_3$ Αν $AB=11$ τότε $F=1$ όταν $CD=0111$ $CD=11$

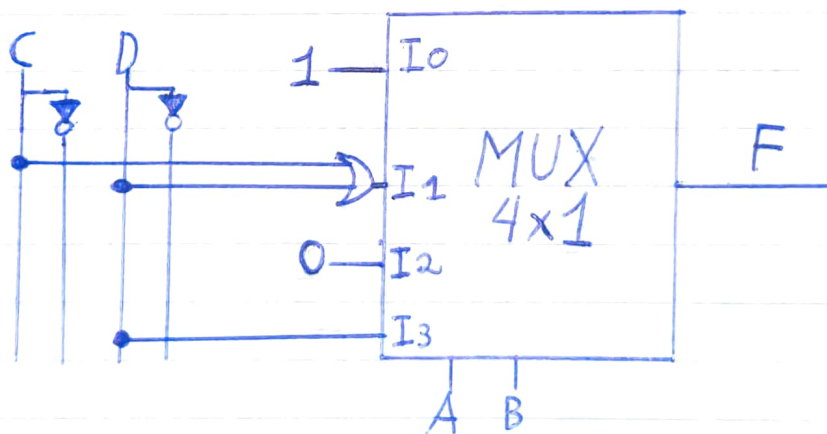
$C \backslash D$

$C \backslash D$	0	1
0		1
1		1

$\Rightarrow D$ ή $C'D+CD=D(C'+C)=D$

Όπως έχουμε ζανα αναφέρει αφού στην ομάδα μόνο το $D=1$ υπάρχει
θα ζουμε D . Αν ήταν $D=0$ θα είζουμε D' . Εκεί η ομάδα θα ζαν 1 σε ηλί αρα

3) Σχήμα:



Παράδειγμα 2

Δίνεται $F(ABCD) = \Sigma(0, 1, 2, 3, 5, 6, 7, 13, 15)$. Τώρα να υλοποιηθεί με MUX 8x1

όπου A, B, C είναι Γραμμές Επιλογής και D είναι Γραμμή Εισόδου

Είσοδοι				Έξοδος
A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

$$I_0 = 1$$

$$I_1 = 1$$

$$I_2 = D \quad F(=D)$$

$$I_3 = 1$$

$$I_4 = 0$$

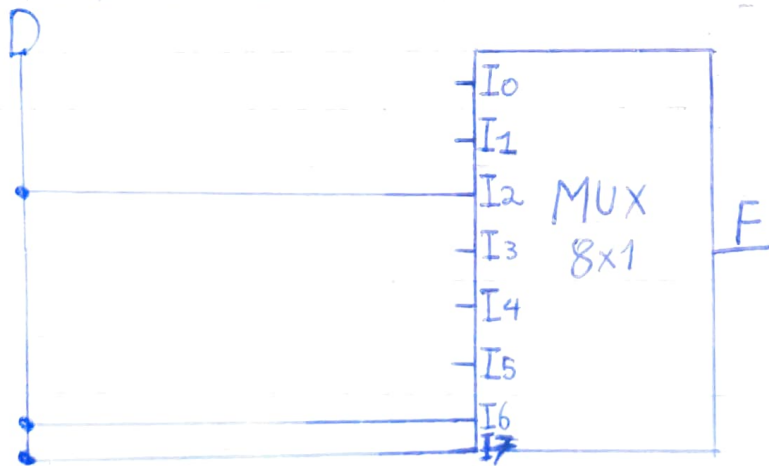
$$I_5 = 0$$

$$I_6 = D$$

$$I_7 = D$$

■ Αλλάζοντας την
σχεδίαση και θάβοντας
το D στις εισόδους
(Μονάδα D) αναγκαστικά
τα A, B, C πάνε στις γραμμές
επιλογής. Άρα έχω 3 γραμμές
επιλογής \Rightarrow MUX 8x1
(Είσοδοι $I_7 - I_0$)

■ Για κάθε περίπτωση
συγκρίνουμε την F με την D



■ ROM

• Οργάνωση Μνήμης:

- 1) Αποκωδικοποίηση
- 2) Εσωτερική Οργάνωση με Flip-Flop
- 3) Επικοινωνία με CPU και Register

• Μέγεθος Μνήμης

- 1) Πλήθος λέξεων ανά Memory Chip
- 2) Μέγεθος λέξεων
- 3) Πλήθος Chip

π.χ. Μια μνήμη μεγέθους 1 Kbyte χωρίζεται σε 4 chip των 256 byte ($256 = 2^8$) με μέγεθος λέξης 1 byte και 256 λέξεις/chip:

$$4 \text{ chip} \times 256 \text{ λέξεις} \times 1 \text{ byte/λέξη} = 2^2 \times 2^8 \times 2^0 = 2^{10} \text{ byte} = 1 \text{ Kbyte}$$

► Το ίδιο μέγεθος μνήμης (1Kb) με 128 λέξεις/chip:

α) Διπλασιάζω τα chip:

$$\left. \begin{array}{l} 8 \text{ chip} = 2^3 \\ 128 \text{ λέξεις} = 2^7 \\ 1 \text{ byte/λέξη} = 2^0 \end{array} \right\} 2^3 \times 2^7 \times 2^0 = 2^{10} \text{ byte} = 1 \text{ Kbyte}$$

β) Διπλασιάζω το Μέγεθος λέξης:

$$\left. \begin{array}{l} 4 \text{ chip} = 2^2 \\ 128 \text{ λέξεις} = 2^7 \\ 2 \text{ byte/λέξη} = 2^1 \end{array} \right\} 2^2 \times 2^7 \times 2^1 = 2^{10} \text{ byte} = 1024 \text{ byte} = 1 \text{ Kbyte}$$

Παράδειγμα 1:

Δίνεται μια μνήμη 1Kb οργάνωση σε 16 Chip. Το μέγεθος της λέξης/chip είναι 2 byte

- 1) Πόσες λέξεις υπάρχουν σε κάθε chip;
- 2) Ποιο είναι το μέγεθος των αποκωδικοποιητών CS, WS;
- 3) Πώς επιλέγεται η διεύθυνση μνήμης 132;
- 4) Πώς θα προστεθούν τα δεδομένα των θέσεων μνήμης 132 και 133; Έστω ότι η θέση 132 περιέχει τον αριθμό 50 και η θέση μνήμης 133 τον αριθμό 7

Λύσεις:

- 1) Μέγεθος λέξης 2 byte. Η μνήμη έχει μέγεθος 1Kb από ερώτη $2^{10} / 2^1 = 2^9 = 512$ λέξεις των 2 byte
Έχουμε 16 Chip, επομένως $\frac{512}{16} = \frac{2^9}{2^4} = 2^5 = 32$ λέξεις/chip
κάθε chip έχει μέγεθος $32 \times 2 = 64$ byte

Λύση 2)

CS = Chip Select:

- Το Chip Select χρησιμοποιεί όταν δοθεί μια διεύθυνση από την CPU προς τη μνήμη για να βρεθεί το Chip στο οποίο υπάρχει η λέξη
- Μέγεθος: Αν έχω N Chip είναι $\log_2(N)$ Είσοδοι x Ν Εξόδους. 16 Chip \Rightarrow CS έχει 16 Εξόδους. Άρα CS: 4x16
- Τα πιο σημαντικά bit της διεύθυνσης πάνε στο CS
- Η μνήμη έχει 1 CS

WS = Wordselect:

- Υπάρχει 1 WS για κάθε chip. Εσωτερικά σε κάθε chip. (10 WS)
- Έχει τόσες Εξόδους όσες είναι και οι λέξεις του chip. 32 λέξεις. Άρα 5x32 (Επιλογές λέξεων)
- Είσοδοι = \log_2 (Πλήθος Εξόδων)

Λύση 3)

Πόσα bit χρειαζονται για την διεύθυνσιοδότηση της μνήμης; 512 λέξεις. Άρα $\log_2(512) = 9 \text{ bit}$

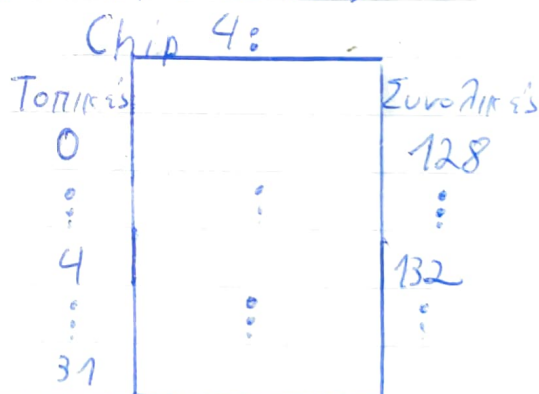
► I8 - I5: Είσοδοι στο CS

► I4 - I0: Είσοδοι κοινές σε όλα τα WS.

• Η CPU ζητάει την διεύθυνση 132. Επειδή έχουμε 512 λέξεις, η διεύθυνση είναι μήκους 9 bit. $132 = 010000100$ (* $132 = 128 + 4 = 2^7 + 2^2$)

$\begin{matrix} I_8 & I_7 & I_6 & I_5 & I_4 & I_3 & I_2 & I_1 & I_0 \\ \hline \text{Chip 4} & & & & & & & & \\ \text{(CS)} & & & & & & & & \end{matrix}$

$\begin{matrix} \text{Λέξη 4} \\ \text{(WS)} \end{matrix}$



ΒΗΜΑΤΑ

1) ΒΡΙΣΚΩ ΤΟ ΠΛΗΘΟΣ ΤΩΝ ΛΕΞΕΩΝ

2) ΜΕΓΕΘΟΣ ΔΙΕΥΘΥΝΣΕΩΝ ΠΟΥ ΖΗΤΑ Η CPU

3) ΧΩΡΙΖΩ ΤΗ ΔΙΕΥΘΥΝΣΗ bit 1 στα bit του Chipselect και του Wordselect

SOS