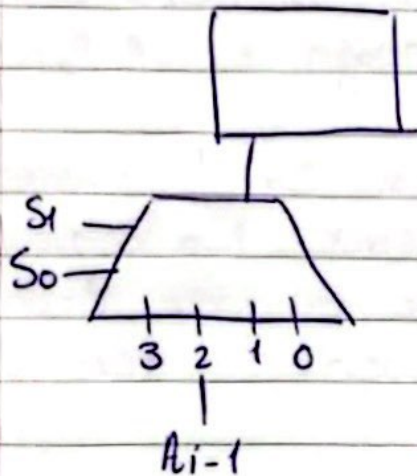


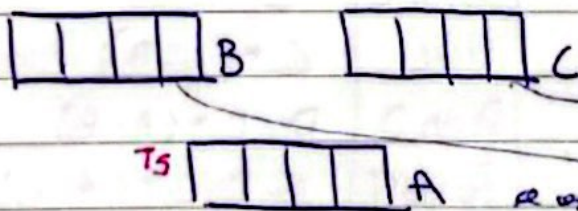
Λογικασταθμική Υπολογιστών (Ενδιάμεση)

Ανασχεδιασμός οχι βιβλίο

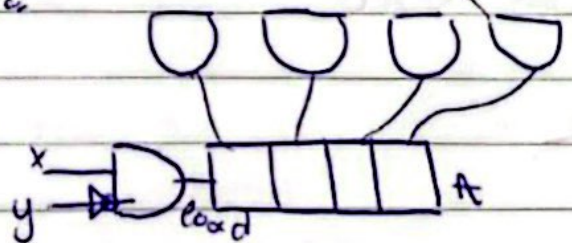


→ 0 καταχωρητές αλλιώς είναι έγχρωμοι να κάνει 4 βήματα

Να δώσω RTL καταχωρητές και να ηρθεί να υπονοή-
σουν το hardware: $x_{i+1}^A \leftarrow B \wedge C$
 T_5

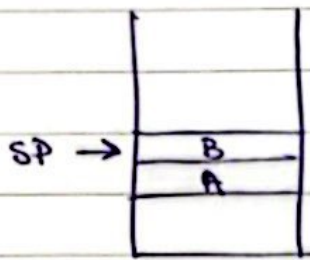


ή αν μας έδινε συνάρτηση



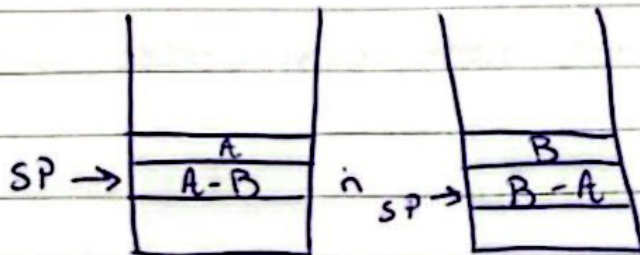
και τείνει να γράφει

Στα stack αυτή την έκφραση $(A-B)*C+D*(F-H)$ θα πρέπει να το φέρουμε σε μορφή RPN, δηλ. $AB-C*...$
 Το δεύτερο εγχείρημα που πρέπει να προσέξουμε στο stack:



Θα πρέπει να μας δώσει τη σειρά στην οποία θα γίνεται η πράξη π.χ. $(A-B)$ ή $B-A$

Αν το γινόμενο θα γινε φέρουμε και το stack. Να δείχνουμε τι υπάρχει σε κάθε βήμα στην κορυφή του stack



* το B είναι στην κορυφή και μετά το A

Push A	A
Push B	B, A *
SUB	A-B
Push C	C, A-B <small>είναι το C στην κορυφή</small>
MUL	$C*(A-B)$ <small>είναι όλο</small>
Push D	D, $C*(A-B)$ <small>κορυφή</small>
Push G	G, D, $C*(A-B)$
MUL	$G*D, C*(A-B)$

...



αυτό θα χρειάζεται μόνο για τα stack για να μπορούμε να δείχνουμε πώς είναι η κορυφή

→

opcode	R	R	Address	Port
--------	---	---	---------	------

Μπορεί να μην μας δίνεται το opcode, αλλά θα μπορούμε να το βρούμε από το πόσες εντοχές έχει ο καταχωρ.

Αν έχει 8 καταχωρ. → 4 bits

Πιθανόν να προϋποθέτει και κάποια διεύθυνση που θα πρέπει να μας δώσει πόσο θα είναι

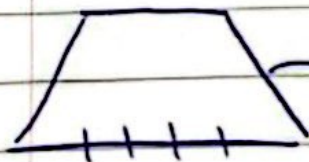
Το μέγεθος της εντοχής θα πρέπει να είναι πολύ/λίγο του 8 bytes. Αν δεν είναι για να συμπληρωθεί τότε θα βρούμε εμείς (Port)

→ Πολυηλέκτες

Ανοδιωτιστές

Αθροιστές

Πολυηλέκτες

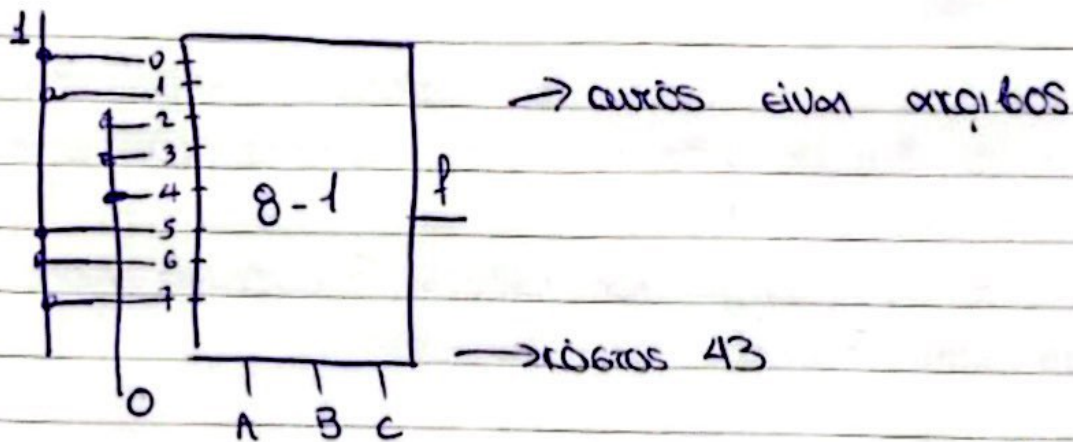


→ αυτοί είναι οι βασικοί πολυηλέκτες.

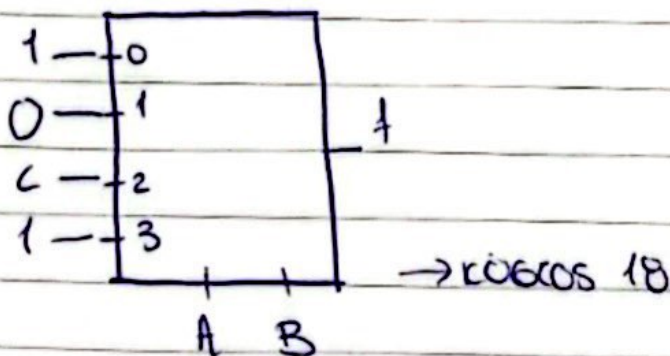
Θα μπορούσαμε να τον χαρακτηρίσουμε και για να βρούμε μια ευχέρηση

π.χ. $f(A, B, C) = \sum m(0, 1, 5, 6, 7)$

Αν πάρω 8 σε 1 που είναι ο μεγαλύτερος που μπορούμε να πάρουμε, 625 ελεύθετος θα βάρουμε 0 ή 1



Αν κάνουμε έναν 4-6ε-1



	A	B	C	$f(A,B,C)$	
0	0	0	0	1	} n 6w είναι 1 620 00
1	0	0	1	1	
2	0	1	0	0	} n 6w είναι 0
3	0	1	1	0	
4	1	0	0	0	} ίδιο με C
5	1	0	1	1	
6	1	1	0	1	} n ευχαρ είναι 1
7	1	1	1	1	

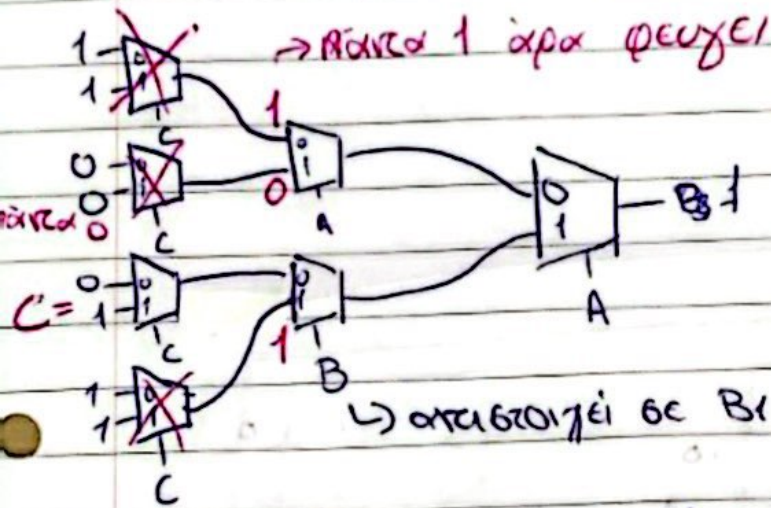
Εκ μπορούμε να μας δίνει ελεύθετος πύλων

\Rightarrow → αριθμός 2

το αριθμός το δίνει ο αριθμός των ελεύθετων

\Rightarrow → αριθμός 3

$$f(A, B, C) = \sum m(0, 1, 5, 6, 7)$$



το A χωρίζει τα πιθανά
αποτελέσματα σε 2 κομμάτια

→ απελευθεύει σε B1 άφου το A=1

→ Αν μας ζητήσει να το κάνουμε μόνο
με πολλαπλασιαστές $2^{\text{bit}} - 1$

Μπορεί να μας ζητήσει ο ίδιος με τι πολλαπλασιαστή να
το κάνουμε $8^{\text{bit}} - 1$, $4^{\text{bit}} - 1$ ή $2^{\text{bit}} - 1$

Μπορεί να μας ζητήσει να το κάνουμε με τον πιο φθηνό
τρόπο, άρα αποβάλλεται το $8^{\text{bit}} - 1$ και μένουν τα
άλλα 2 αναλόγως, επειδή μπορεί να χρειάζεται να
προσθέσουμε πύλες στο $4^{\text{bit}} - 1$

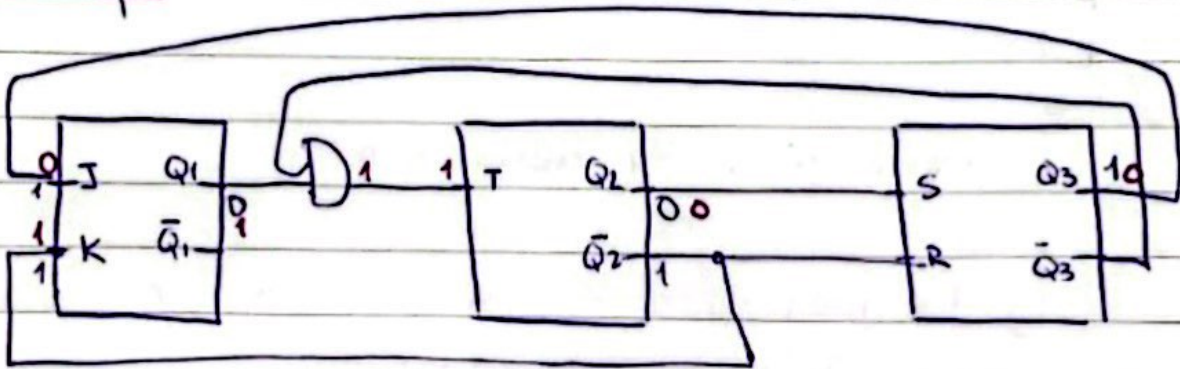
βλέπουμε αμέσως κατασκευάζουν εκτός - ύλης

Οι αποδοκονοιντες χρησιμοποιουσε περικοπο εως μινιμους

Δεν εα ημερεει

Οι αποδοκονοιντες ειναι για την υλοποιηση εναρξησεων γιατι ειναι ποτι ακριβοι και αργοι

flip-flop



0 0 1
1 0 0 ←
0 1 0
↖
reset

Το JK μετρεει με το SR

το SR αν εχει 2 αρεους δεν

ζεπει τι να βγαλει, ενω το JK με 2 αρεους αλλαζει τιμη, ηταν 0 αρα το κανει 1

Το Π οταν ειναι 1 εα αλλαζει τιμη αρα αν το 0 ειναι 1

Οταν εχω J=0 και K=1 γινεται reset

Δεν πρεπει να γεραφτε κατι ξεχωριστο για τους μονδαριους

Multiplex

ROM1	4KB
ROM2	4KB
RAM1	2KB
RAM2	2KB
	<u>12KB</u>

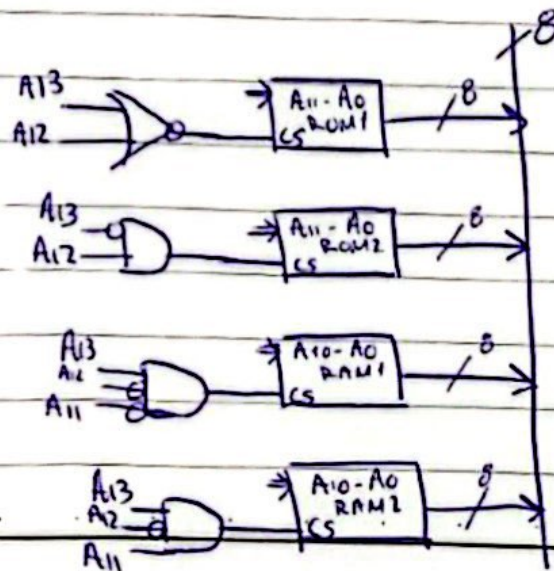
Address 14 bits

		$A_{13} A_{12}$			
0-(4K-1)	ROM 1	00	000000... 0	0000-0FFF	} A_{13} n ROM Sourceu Ödev eivon 00
		00	1111... 1		
4-(8K-1)	ROM 2	01	00... 0	1000-1FFF	
		01	11... 1		
	RAM 1	100	00... 0	2000-27FF	
		100	111... 1		
	RAM 2	101	00... 0	2800-2FFF	
		101	11... 1		

A B C D E F
10 11 12 13 14 15

4K/8K/16K

Adresler → 16 bit



Cache

$$C = K \times N \times L$$

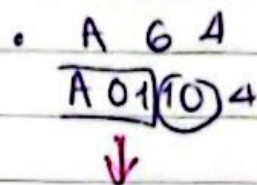
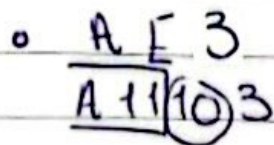
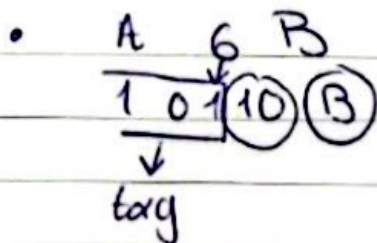
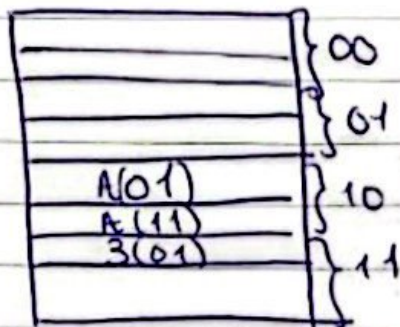
K = number of bytes in each set

N = sets

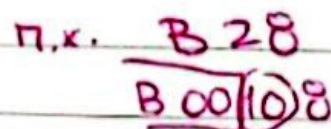
L = bits per byte

$$128 = 2 \times 4 \times 16$$

tag	Set	offset
6	2	4



αφού υπάρχει ήδη
 αν ήταν διαφορετικό
 αλλά έπρεπε να μπει
 στο 10-set τότε θα
 το αντικαθιστούμε με το παλιό



θα μας δώσει πιο
 αλγόριθμο να χειρισθούμε
 για την αντικατάσταση