

Αρχιτεκτονική Υπολογιστών

Τμήμα Ρουμελιώτη

Παλαιά Θέματα-Λύσεις
+Σύντομο Cheat Sheet

ΠΕΡΙΕΧΟΜΕΝΑ:

Cheat sheet - Μετατροπές, συμπλήρωμα, πρόσθεση, RTL, διευθυνσιοδότηση

Ιούνιος 2021 - Θέματα

Σεπτέμβριος 2021 - Θέματα/Λύσεις

Ιούνιος 2022 - Θέματα/Λύσεις

Σεπτέμβριος 2023 - Θέματα

Ιούνιος 2023 - Θέματα

Φεβρουάριος 2024 - Θέματα/Λύσεις

Ιούνιος 2024 - Θέματα/Λύσεις

*δεν είναι όλες οι λύσεις δικές μου, το παρόν
αρχείο αποτελεί επίσης αποτέλεσμα έρευνας

Δεκαδικοί	Δυαδικοί	Δεκαεξαδικοί	Οκταδικοί
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

Συμπλήρωμα:
Για να επιτύχει το συμπλήρωμα ως προς το 2 ,πρέπει να έχουμε κάνει συμπλήρωμα ως προς το 1.
Συμπλήρωμα ως προς το 1:Αντιστρέφω τον αριθμό.πχ: 1001 0101 ,θα γίνει 0110 1010
Συμπλήρωμα ως προς το 2:Αφού γίνει το συμπλήρωμα ως προς το 1 θα προσθέσουμε τον αριθμό 1. πχ: 1001 0101 : 0110 1010
Πρόσθεση bit
1+1=0 ,με κρατούμενο 1
0+0=0
0+1=1
1+0=1

+0000 0001
0110 1011

ΠΧ:
74622061₈ = 111 100 110 010 010 000 110 001 (1 ψηφίο οκταδικού: 3 bits)
473A85₁₆ = 0100 0111 0011 1010 1000 0101 (1 ψηφίο δεκαεξαδικού: 4 bits)

RTL:

Fetch (υπάρχει πάντα)

Execution (αλλάζει ανα εντολή)

Fetch:

MAR \rightarrow PC

MDR \leftarrow M[MAR]; PC \leftarrow PC + 1

IR \leftarrow MDR(opcode)

LDA:

1. MAR \leftarrow PC, PC \leftarrow PC + 1

2. MDR \leftarrow M[MAR]

3. IR \leftarrow MDR(opcode)

4. F \leftarrow 1

1. MAR \leftarrow MDR(address)

2. MDR \leftarrow M[MAR]

3. AC \leftarrow MDR

ADD:

1. MAR \leftarrow PC, PC \leftarrow PC + 1

2. MDR \leftarrow M[MAR]

3. IR \leftarrow MDR(opcode)

4. F \leftarrow 1

1. MAR \leftarrow MDR(address)

2. MDR \leftarrow M[MAR]

3. AC \leftarrow MDR + AC

4. F \leftarrow 0

STA:

1. MAR \leftarrow PC, PC \leftarrow PC + 1

2. MDR \leftarrow M[MAR]

3. IR \leftarrow MDR(opcode)

4. F \leftarrow 1

1. MAR \leftarrow MDR(address)

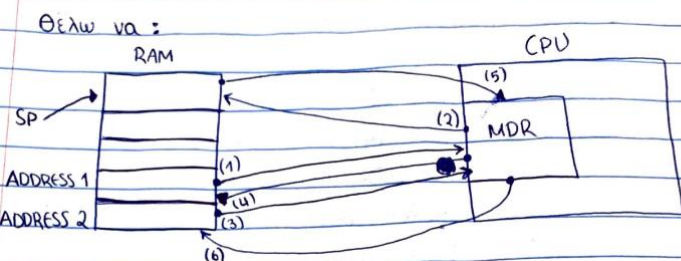
2. MDR \leftarrow AC

3. M[MAR] \leftarrow MDR

4. F \leftarrow 0

ΑΣΚΗΣΗ 4

Εκτέλεση SWAP-STACK-AB. Εναλλαγή δεδομένων θέσεων μνήμης A, B χωρίς καταχώριση για ενδιάμεση αποθήκευση.



T0: MAR \leftarrow PC, PC \leftarrow PC + 1

T1: MDR \leftarrow M[MAR], Z \leftarrow PC

T2: IR \leftarrow MDR

T3: MAR \leftarrow IR(operand1) } (1) τιμή ADDRESS 1 μεταφέρεται στον MDR

T4: MDR \leftarrow M[MAR]

T5: MAR \leftarrow SP

T6: M[MAR] \leftarrow MDR } (2) ο MDR περνάει την τιμή της ADDRESS 1 στον SP

T7: MAR \leftarrow IR(operand2) } (3) τιμή ADDRESS 2 μεταφέρεται στον MDR

T8: MDR \leftarrow M[MAR]

T9: MAR \leftarrow IR(operand1) } (4) ο MDR περνάει την τιμή της ADDRESS 2 στον SP

T10: MDR \leftarrow M[MAR]

T11: MAR \leftarrow SP

T12: MDR \leftarrow M[MAR]

T13: MAR \leftarrow IR(operand2) } (5) η τιμή του SP μεταφέρεται στον MDR

T14: MDR \leftarrow M[MAR]

Διακίνηση Στοιβάς: Με την χρήση του καταχωρητή stack pointer ο SP δείχνει στην κορυφή της στοιβάς, στην τελευταία θέση που έχουν γραφτεί δεδομένα. Εμφανίζει στη στοιβά οπότε ο SP μειώνεται κατά 1. Ανάγνωση από την στοιβά σημαίνει αύξηση του SP. Ο MAR λαμβάνει τιμές από τον SP.

Η ανάγνωση του 50 σημαίνει ότι ο SP θα αυξηθεί κατά 1 όταν θα γίνει νέα εμφάνιση, η τιμή 50 θα διαγραφεί.

ΑΣΚΗΣΗ 5

Σχεδιασμός εντολής PUSH A: τοποθετώ δεδομένα της θέσης μνήμης A στην κορυφή της στοιβάς. Η θέση κορυφής στοιβάς υπάρχει στον SP, ο οποίος μειώνεται κατά 1 όταν τοποθετηθεί ο παράγοντας στη στοιβά.

T0: MAR \leftarrow PC, Z \leftarrow PC + 1

T1: MDR \leftarrow M[MAR], PC \leftarrow Z

T2: IR \leftarrow MDR

T3: MAR \leftarrow IR(operand)

T4: MDR \leftarrow M[MAR]

T5: MAR \leftarrow SP, Z \leftarrow SP - 1

T6: M[MAR] \leftarrow MDR

T7: SP \leftarrow Z

- ΑΝΑΚΛΗΣΗ -

T3: MAR \leftarrow SP

T4: MDR \leftarrow M[MAR]

T5: ACC \leftarrow MDR

T6: MAR \leftarrow IR(opcode)

T7: MDR \leftarrow M[MAR]

T8: MAR \leftarrow SP

T9: M[MAR] \leftarrow MDR

T10: MDR \leftarrow ACC

T11: MAR \leftarrow IR(opcode)

T12: M[MAR] \leftarrow MDR

Διευθυνσιοδότηση

1. Απόλυτη Διευθυνση

Absolute EA=A (effective address)

MOVE R5,A ,στο A είναι αποθηκευένος ο παράγοντας
MOVE R5,700

2. Άμεση Διευθυνση

Immediate EA=- (effective address)

MOVE R5,#X
MOVE R5,#7

3. Καταχωρητή: Register

Absolute EA=R3 (effective address)

MOVE R5,R3

4. Έμμεση Address

Indirect EA=(A) ,στο A είναι αποθηκευμένη η
διευθυνση του παράγοντα

MOVE R5,(A)
MOVE R5,(700)

5. Δείκτη (Ri)

Index EA:(R3)
MOVE R5,(R3)

6. Δείκτη με μετατόπιση (Ri+x)

Index with offset EA:(R3+x)
MOVE R5,x(R3)

7. Base Register EA:(R3+R2) MOVE R5,(R3,R2)

8. Base offset Register EA:(R3+R2+x) MOVE R5,(R3,R2)x

9. Σχετική Relative PC MOVE R5,x(PC) EA=(PC+x)

10. Αυτοαυξηση AutoIncrement Ri MOVE R5,(R3)+ EA:(R3)

11. Αυτομείωση AutoDecrement Ri MOVE R5,-(R3) EA:(R3-1)

Τα πιο σύνηθες ↑

Θέμα 1^ο (15 μονάδες)

Σε κάποιο σημείο εκτέλεσης ενός προγράμματος, η μνήμη, οι καταχωρητές R0,... R6, και ο Program Counter έχουν τα παρακάτω περιεχόμενα (στο δεκαδικό):

Memory		Registers		Z	Y
				(προτελευταίο ψηφίο)	(τελευταίο ψηφίο)
.	.	R0	512		
.	.	R1			
512	28	R2	256		
513	215	R3			
514	761	R4	710		
.		R5			
.		R6	Z		
.					
760	512	PC	688		
761	99				
762	100				
763	101				
764	102				
765	103				
766	104				
767	105				
768	106				
769	107				
770	108				

Λύση:

- a) MOVE R7,512: **28**
- b) MOVE R7,(514): **99**
- c) MOVE R7,50(R4): **512**
- d) MOVE R7,R0: **512**
- e) MOVE R7,72(PC,R6): **105**

Προσοχή: Κάθε λέξη μνήμης περιέχει 16 bits, δηλαδή η λέξη αυτού του υπολογιστή είναι 16 bits.

Δώστε την τιμή που θα μεταφερθεί στον καταχωρητή R7, όταν εκτελούνται οι παρακάτω εντολές, με τις ίδιες αρχικές τιμές για κάθε εντολή:

- a. MOVE R7, 512
- b. MOVE R7, (514)
- c. MOVE R7, 50(R4)
- d. MOVE R7, R0
- e. MOVE R7, 72(PC, R6)

Θέμα 2^ο (20 μονάδες)

Σας δίνονται: α) 1 chip μνήμης χωρητικότητας 16 Megabits με 8 bits ανά θέση μνήμης και β) 2 chips μνήμης χωρητικότητας 512 Mbytes το καθένα με 4 bits ανά θέση μνήμης. Με τα chips αυτά θέλουμε να κατασκευάσουμε μια μνήμη 2M θέσεων με 12 bits ανά θέση μνήμης. Δώστε το λογικό διάγραμμα της μνήμης αυτής, δείχνοντας πως θα χρησιμοποιηθούν τα bits της διεύθυνσης και προσθέτοντας ό,τι άλλο τυχόν λογικό κύκλωμα χρειάζεται.

Λύση:

Ιούνιος 2021

Θέμα 3^ο (15 μονάδες)

Υλοποιείστε την παρακάτω συνάρτηση χρησιμοποιώντας μόνο πύλες NAND δύο εισόδων και inverters. Θα πρέπει να την ελαχιστοποιήσετε πρώτα χρησιμοποιώντας έναν Karnaugh map. Ο όρος Y μπορεί να συμπέσει με κάποιον άλλο, αλλά αυτό δεν έχει καμία σημασία.

$$f(A, B, C, D) = \sum m(0, 1, 2, 6, 7, 8, 14) + \sum d(9, 10, 11, 15)$$

Λύση:

Θέμα 4^ο (15 μονάδες)

Ένας υπολογιστής του ενός εσωτερικού διαύλου (bus) έχει εντολές της παρακάτω μορφής:

OPCODE	ADDRESS
--------	---------

Δώστε την περιγραφή της συνολικής εκτέλεσης της εντολής ISNZ (increment and skip if not zero). Η εντολή αυτή αυξάνει τον accumulator κατά ένα και αν το τελικό αποτέλεσμα είναι μηδέν, τότε θα εκτελεστεί η επόμενη εντολή, αλλιώς η εκτέλεση θα συνεχιστεί με την μεθεπόμενη χωρίς να εκτελεστεί η επόμενη. Δώστε την περιγραφή σε RTL.

Λύση:

Θέμα 5^ο (20 μονάδες)

Θεωρήστε δύο κρυφές μνήμες, μια άμεσης οργάνωσης και μία με οργάνωση 2-δρόμων συνόλου συσχέτισης. Κάθε μία έχει χωρητικότητα 8 KBytes και 16 bytes ανά γραμμή ενώ είναι προσπελάσιμη με διευθύνσεις των 32 bits. Σε κάθε byte του συστήματος μνήμης αντιστοιχεί μία φυσική διεύθυνση.

1. Να αναφέρετε τα πεδία από τα οποία θεωρούμε ότι αποτελείται η διεύθυνση που παράγει ο επεξεργαστής, το εύρος κάθε πεδίου σε bits, καθώς και τι σημασία κάθε πεδίου στην περίπτωση της κρυφής μνήμης:

- i. άμεσης οργάνωσης
- ii. δύο τρόπων συνόλου συσχέτισης.

2. Δίνονται οι διευθύνσεις στο δεκαεξαδικό

- α. EY79467B
- β. EY794678
- γ. EY79F672
- δ. EY79D674

όπου το ψηφίο Y είναι το τελευταίο ψηφίο τους αριθμού μητρώου σας.

2i. Για την κρυφή μνήμη άμεσης οργάνωσης να αναφέρετε και να δικαιολογήσετε εάν τα περιεχόμενα των διευθύνσεων:

α και β,
γ και δ,
α, β και δ
είναι δυνατόν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη.

2ii. Για την κρυφή μνήμη με οργάνωση 2-τρόπων συνόλου συσχέτισης να αναφέρετε και να δικαιολογήσετε εάν τα περιεχόμενα των διευθύνσεων:

α, β και δ
α, β, γ και δ
είναι δυνατόν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη.

Λύση:

Μετατρέπω σε δυαδικό:

E279467B = 1110 0010 0111 1001 0100 0110 0111 1011
E2794678 = 1110 0010 0111 1001 0100 0110 0111 1000
E279F672 = 1110 0010 0111 1001 1111 0110 0111 0010
E279D674 = 1110 0010 0111 1001 1101 0110 0111 0100

Η διεύθυνση αποτελείται απο τα πεδία: tag(υπολοιπο),set (log2(N),offset log2(L)
Και έχει μήκος 32 bits με βάση την εκφώνηση

i) N=C/L=512 (8196/16)

19	9	4
----	---	---

Tag set offset

Αβ είναι στο ίδιο πλαίσιο αρα πάντα μαζί.γ,δ στο ίδιο set αρα δεν μπορούν να είναι μαζί.Μπορούν να βρίσκονται στην κρυφή μνημη ταυτόχρονα ή α,β,γ ή α,β,δ

	tag	set	offset
α	(E279)010	0(67)	B
β	(E279)010	0(67)	8
γ	(E279)111	1(67)	2
δ	(E279)110	1(67)	4

ii) N=C/L*K=256 (8196/32)

19	8	5
----	---	---

Tag set offset

Αβ είναι στο ίδιο πλαίσιο αρα πάντα μαζί.γ,δ έχουν ίδιο set αλλά το κάθε set δεν έχει μονο 1 tag αρα μπορούν να είναι μαζί.Μπορούν να βρίσκονται στην κρυφή μνημη ταυτόχρονα όλες οι διευθύνσεις και α,β,γ,δ και α,β,δ

	tag	set	offset
α	(E279)010	0(6)011	1(B)
β	(E279)010	0(6)011	1(8)
γ	(E279)111	1(6)011	1(2)
δ	(E279)110	1(6)011	1(4)

Ιούνιος 2021

Θέμα 6^ο (15 μονάδες)

Ένας τυπικός επεξεργαστής RISC χρειάζεται 4 κύκλους (ανάκληση εντολής, υπολογισμό διεύθυνσης, load, αποθήκευση σε καταχωρητή) για τις εντολές LOAD και STORE, και 3 κύκλους (ανάκληση εντολής, εκτέλεση, αποθήκευση σε καταχωρητή) για κάθε μία από τις άλλες εντολές. Εξετάστε το παρακάτω πρόγραμμα:

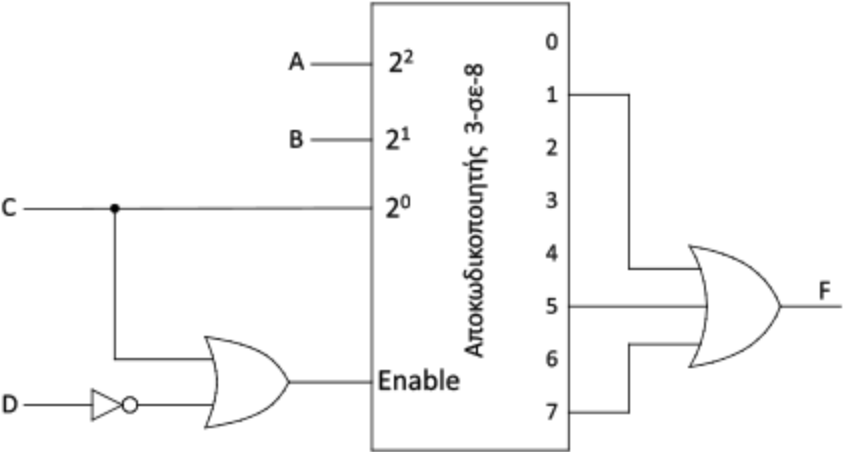
```
LOAD:      R1 ← X
SUBTRACT:  R1 ← R1-R2
STORE:     X ← R1
ADD:       R2 ← R1+R2
```

Σχεδιάστε ένα χρονικό διάγραμμα που να δείχνει την εκτέλεση αυτών των τεσσάρων εντολών στην διασωλήνωση, και υπολογίστε πόσο χρόνο χρειάζεται ο επεξεργαστής για την εκτέλεσή τους. Να υποθέσετε μέγιστη επικάλυψη των εντολών.

Λύση:

Θέμα 1^ο (15 μονάδες)

α) Να εκφράσετε τη συνάρτηση F που υλοποιεί το παρακάτω κύκλωμα στην απλούστερη δυνατή της μορφή αθροίσματος γινομένων.



Λύση:

Θέμα 1. Μονάδες 15

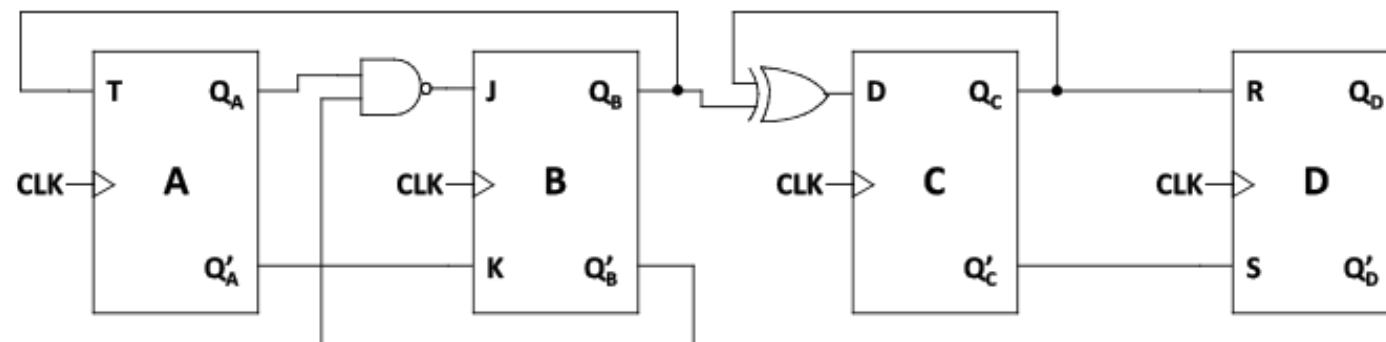
$$F = (C + D')A'B'C + (C + D')AB'C + (C + D')ABC$$

CD \ AB	00	01	11	10
00			1	1
01				
11			1	1
10			1	1

Από τον Karnaugh: $F = AC + B'C$

Θέμα 2^ο (15 μονάδες)

Δίνεται το παρακάτω ακολουθιακό κύκλωμα που αποτελείται από 4 Flip-Flops Η αρχική τιμή των FF είναι το δυαδικό ισοδύναμο τους τελευταίου ψηφίου του αριθμού μητρώου σας (Y). Π.χ. αν το Y=6 τα Flip-Flops έχουν αρχικά την τιμή ABCD=0110.



Να δώσετε τι τιμές των Flip-Flops μετά από έναν παλμό του ρολογιού, μετά από δύο και μετά από τρεις.

Λύση:

Θέμα 2. Μονάδες 15 (Y τελευταίο ψηφίο AM)

Αρχική κατάσταση = Y (τελευταίο ψηφίο AM)

Y	Αρχική	Πρώτος παλμός	Δεύτερος παλμός	Τρίτος παλμός
0	0000	0101	1011	1010
1	0001	0101	1011	1010
2	0010	0110	1000	1001
3	0011	0110	1000	1001
4	0100	1011	1010	1010
5	0101	1011	1010	1010
6	0110	1000	1001	1001
7	0111	1000	1001	1001
8	1000	1001	1001	1001
9	1001	1001	1001	1001

Θέμα 3° (15 μονάδες)

Μια μηχανή RISC έχει περίοδο ρολογιού 50ns. 20% των εντολών της είναι LOAD και STORE εντολές. Στις θυρίδες καθυστέρησης αυτών των εντολών τοποθετούνται κατά μέσο όρο 50+Z% NO-OP εντολές και 50-Z% χρήσιμες εντολές. Στο καινούργιο μοντέλο της μηχανής που βγαίνει στην αγορά η περίοδος έχει μειωθεί στα 45ns. Το κόστος όμως αυτής της μείωσης είναι να χρειάζονται πλέον δύο θυρίδες καθυστέρησης για κάθε εντολή μνήμης, και μόνο το 20+Z% από όλες τις θυρίδες καθυστέρησης γεμίζει με χρήσιμες εντολές. Πια μηχανή είναι πιο γρήγορη και κατά πόσο;

Λύση:

Θέμα 3. Μονάδες 15 (Z = προτελευταίο ψηφίο AM)
Η A είναι πάντα πιο γρήγορη. Αν T_a και T_b είναι ο χρόνος εκτέλεσης μιας εντολής τότε:

Z	T_a/T_b	A πιο γρήγορη
0	0.925926	7.4%
1	0.930429	7.0%
2	0.934959	6.5%
3	0.939517	6.0%
4	0.944104	5.6%
5	0.948718	5.1%
6	0.953361	4.7%
7	0.958032	4.2%
8	0.962733	3.7%
9	0.967463	3.3%

Σεπτέμβριος 2021

Θέμα 4^ο (22 μονάδες)

Μία μνήμη cache είναι οργανωμένη σε 4 γραμμές των 32 bytes. Η κυρίως μνήμη έχει μέγεθος 64 KBytes. Ο χρόνος προσπέλασης της κυρίως μνήμης είναι 200 ns ενώ της cache είναι 20 ns.

α) Δώστε το μέγεθος του tag σε bits.

β) Ποιός είναι ο μέσος χρόνος προσπέλασης αν το hit ratio είναι $80+Z\%$?

γ) Υποθέστε ότι αρχικά η cache είναι άδεια και χρησιμοποιείται ο αλγόριθμος FIFO για την αντικατάσταση των γραμμών. Η CPU παράγει διαδοχικά την σειρά των διευθύνσεων που δίνεται παρακάτω. Για κάθε μία από αυτές τις διευθύνσεις σημειώστε αν έχουμε hit ή miss. Δώστε τις τελικές τιμές των tags (στο δυαδικό) στο τέλος της σειράς αυτής των διευθύνσεων. (Υ είναι το τελευταίο ψηφίο του AM σας)

113A

3B4F

FA44

11Y3

FAY8

1152

3BY5

1166

Λύση:

Θέμα 4. Μονάδες 22

α) Μέγεθος tag = 11 bits

β) (Z = προτελευταίο ψηφίο AM)

Z	0	1	2	3	4	5	6	7	8	9
T	56ns	54.2ns	52.4ns	50.6ns	48.8ns	47ns	45.2ns	43.4ns	41.6ns	39.8ns
Εναλλακτικά	60ns	58ns	56ns	54 ns	52ns	50ns	48ns	46ns	44ns	42ns

γ) Τα δύο πρώτα ψηφία του tag είναι στο δεκαεξαδικό και τα τρία επόμενα στο δυαδικό. Τα τελικά περιεχόμενα φαίνονται με κόκκινο. (Y = τελευταίο ψηφίο AM)

[illegible]

Θέμα 5^ο (15 μονάδες)

Θεωρήστε ένα υπολογιστή που βασίζεται στη χρήση καταχωρητών γενικού σκοπού και του οποίου το σύνολο των εντολών σε επίπεδο γλώσσας μηχανής αποτελείται από 200 εντολές, το μήκος κάθε εντολής είναι ακέραιο πολλαπλάσιο του οκτώ, και διαθέτει 8 καταχωρητές με μήκος 32 δυαδικών ψηφίων ο καθένας. Η αρτηρία διευθύνσεων είναι των 32 δυαδικών ψηφίων.

α. Για κάθε μία από τις ακόλουθες εντολές σε συμβολική γλώσσα να σχεδιάσετε τη μορφή των εντολών σε επίπεδο γλώσσας μηχανής (δηλαδή να δώσετε τα πεδία από τα οποία αποτελείται κάθε εντολή σε επίπεδο γλώσσας μηχανής, το εύρος κάθε πεδίου και τι δηλώνει κάθε πεδίο) ώστε να μπορεί να διευθυνσιοδοτηθεί ολόκληρη η κύρια μνήμη.

- α) LOAD R, A ; $R \leftarrow A$, όπου A διεύθυνση θέσης μνήμης (κατ' ευθείαν τρόπος διευθυνσιοδότησης).
- β) LOAD R1, (R2) ; $R1 \leftarrow M(R2)$, έμμεσος (indirect) τρόπος διευθυνσιοδότησης με χρήση καταχωρητή.
- γ) LOAD R, #D ; άμεσος (immediate) τρόπος διευθυνσιοδότησης, D είναι αριθμός 4+Z δυαδικών ψηφίων (όπου Z το προτελευταίο ψηφίο του AM σας).

Λύση:

$2^8=256>200$ αρα 8bit opcode. $2^3=8$ αρα 3bit registers

1)	8bit opcode	3bit Register	32bit Address	5bit useless	Μήκος:6bytes
2)	8bit opcode	3bit Register	3bit Register	2bit useless	Μήκος:2bytes
3)	8bit opcode	3bit Register	11bit offset	2bit useless	Μήκος:3bytes

Θέμα 6^ο (18 μονάδες)

Μια μικροαρχιτεκτονική έχει μικροεντολές μήκους $90+Y$ bits. Το αποθηκευμένο μικροπρόγραμμα απαιτεί 1800 θέσεις μνήμης ελέγχου (μROM). Ο σκοπός μας είναι να μειώσουμε το μέγεθος της μνήμης ελέγχου με μία από τις παρακάτω τρεις τεχνικές:

- 1. Κωδικοποιούμε μερικά από τα πεδία των μικροεντολών έτσι ώστε το μήκος τους να μειωθεί στα $60+Y$ bits.
- 2. Χρησιμοποιούμε μια τεχνική αποφυγής της επανάληψης κάποιων bits που είναι κοινά σε μια σειρά μικροεντολών. Έτσι, η πρώτη μικροεντολή της σειράς περιλαμβάνει όλα τα $90+Y$ bits ενώ κάθε επόμενη μικροεντολή της σειράς περιλαμβάνει μόνο τα bits που δεν είναι κοινά. Υποθέστε λοιπόν ότι τα μη κοινά bits είναι $35+Y$ (επομένως κάθε μικροεντολή της σειράς μετά την πρώτη έχει $65-Y$ bits). Υποθέστε επίσης ότι η κάθε σειρά έχει κατά μέσο όρο 9 μικροεντολές.
- 3. Η τελευταία τεχνική είναι να χρησιμοποιήσουμε nanomemory λαμβάνοντας υπ' όψιν ότι υπάρχουν $290+Y$ διαφορετικές μεταξύ τους μικροεντολές.

Ποιά από τις τρεις τεχνικές είναι ελαχιστοποιεί το μέγεθος της μνήμης ελέγχου;

Λύση: **Θέμα 6. Μονάδες 18 (Y = τελευταίο ψηφίο AM)**

1)

Y	0	1	2	3	4	5	6	7	8	9
Μήκος	60	61	62	63	64	65	66	67	68	69
μROM (bits)	108000	109800	111600	113400	115200	117000	118800	120600	122400	124200

2)

Y	0	1	2	3	4	5	6	7	8	9
πρώτη	90	91	92	93	94	95	96	97	98	99
8 επόμενες	35	36	37	38	39	40	41	42	43	44
μROM (bits)	74000	75800	77600	79400	81200	83000	84800	86600	88400	90200

3) Σε κάθε περίπτωση η τρίτη μέθοδος ελαχιστοποιεί το συνολικό μέγεθος της μνήμης ελέγχου

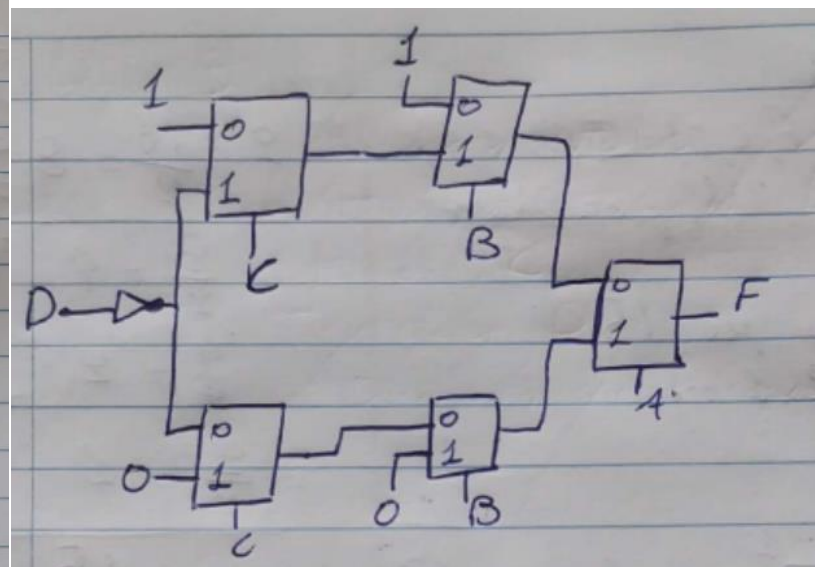
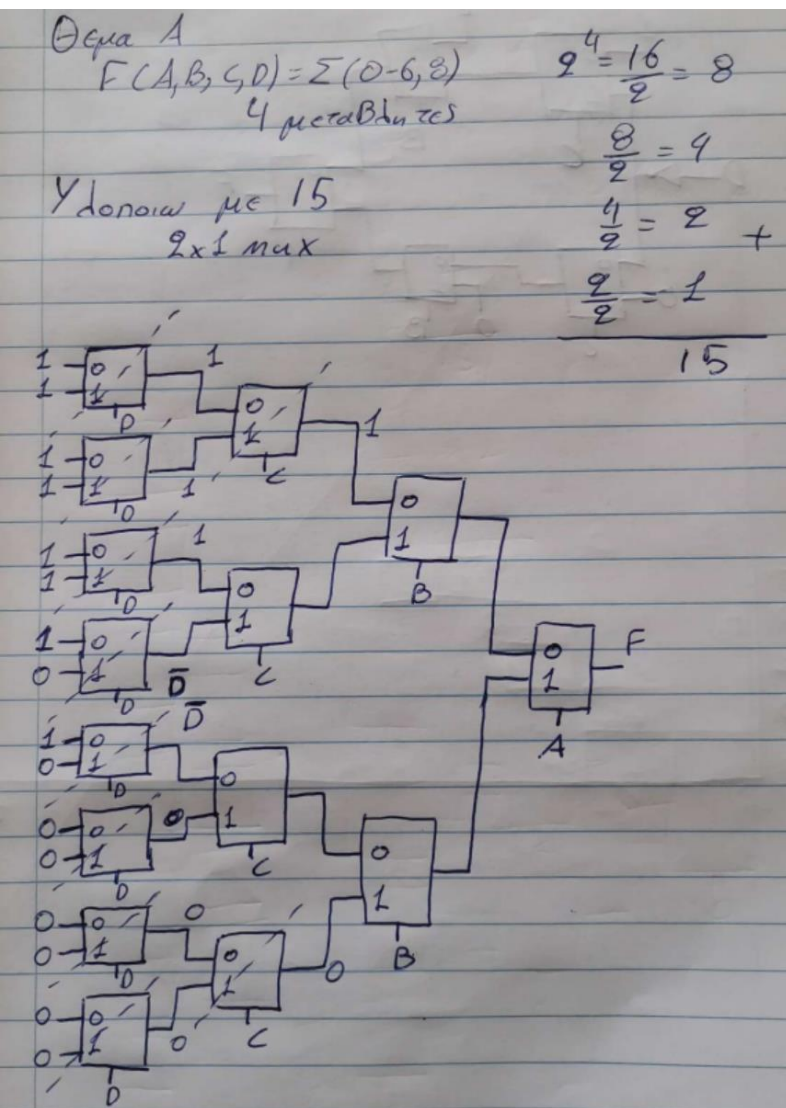
Y	0	1	2	3	4	5	6	7	8	9
μROM	16200	16200	16200	16200	16200	16200	16200	16200	16200	16200
nROM	26100	26481	26864	27249	27636	28025	28416	28809	29204	29601
Σύνολο	42300	42681	43064	43449	43836	44225	44616	45009	45404	45801

Ιούνιος 2022

Θέμα 1ο (15 μονάδες)

Δίνεται η συνάρτηση $f(A,B,C,D) = \Sigma(0-6,8)$. Σχεδιάστε το κύκλωμα αυτό χρησιμοποιώντας αποκλειστικά 5 πολυπλέκτες 2-σε-1 και μια πύλη NOT.

Λύση:

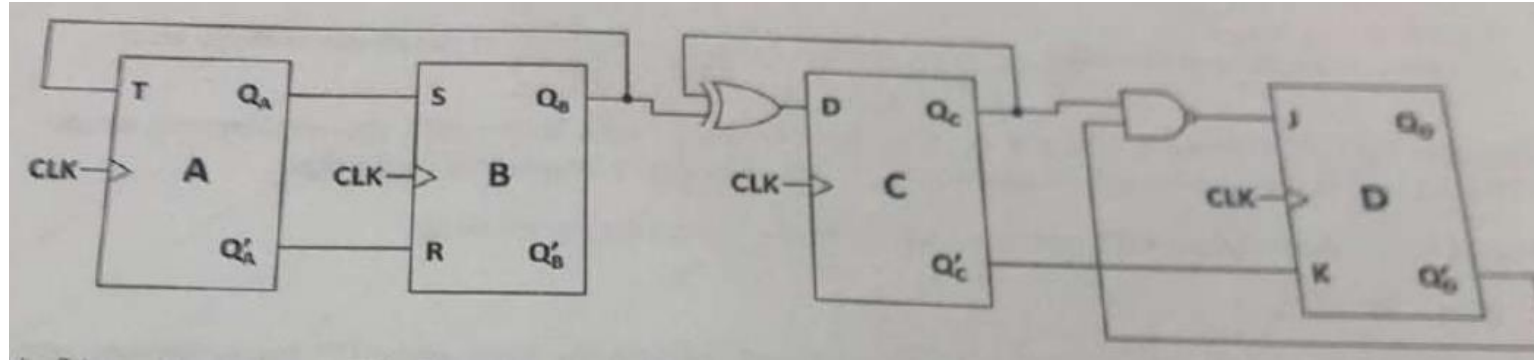


A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

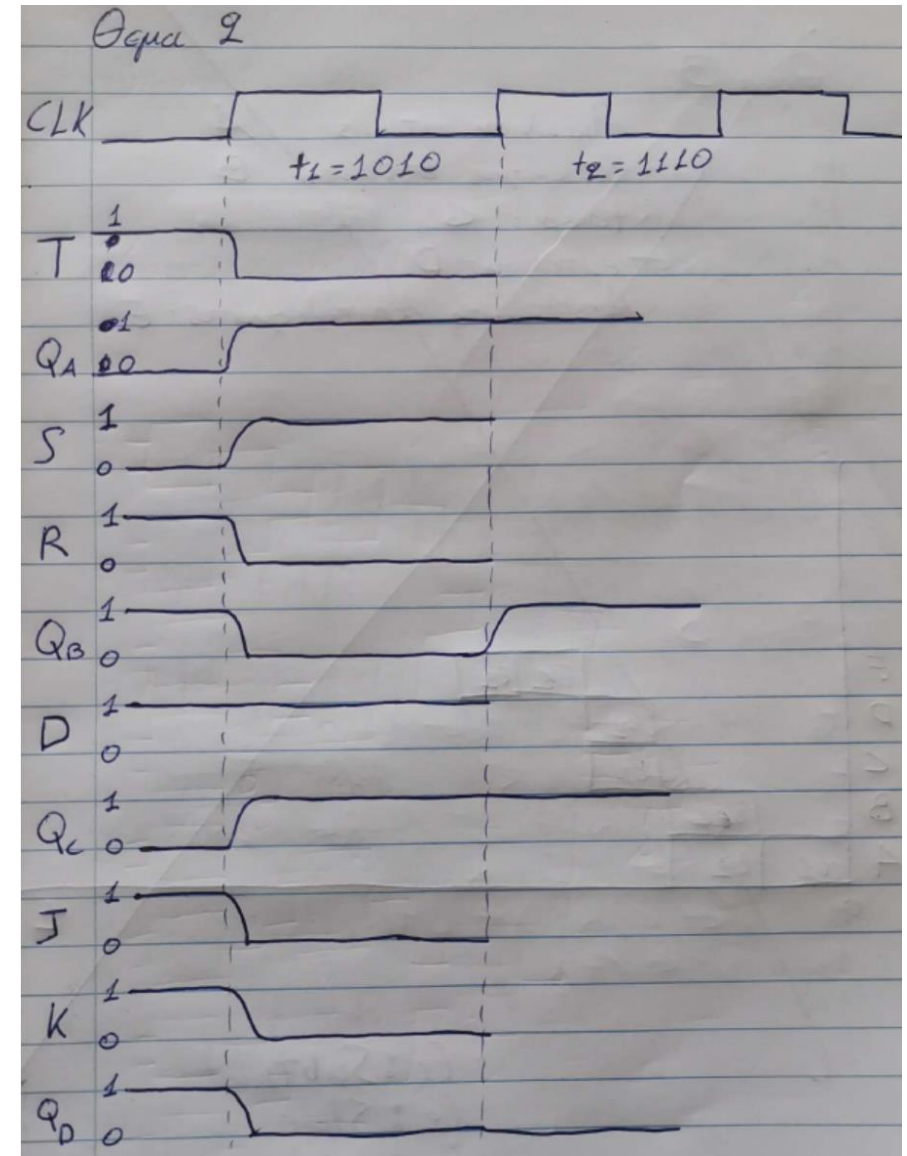
D	F
0	1
1	0

Θέμα 2ο (20 μονάδες)

Δίνεται το παρακάτω ακολουθιακό κυκλωμα που αποτελείται απο 4 flip-flops. Η αρχική τιμή του FF είναι ABCD=0101. Να δώσετε τις τιμες των flip-flops μετα απο εναν παλμο του ρολογιού, και μετα απο δύο.



Λύση:



Θέμα 3° (20 μονάδες)

Ένας τυπικός επεξεργαστής RISC χρειάζεται 4 στάδια (ανάκληση εντολής, υπολογισμός διεύθυνσης, load, αποθήκευση αποτελέσματος) για τις εντολές LOAD και STORE, και 3 στάδια (ανάκληση εντολής, εκτέλεση, αποθήκευση αποτελέσματος) για κάθε μία από τις άλλες εντολές. Εξετάστε το παρακάτω πρόγραμμα:

SUBTRACT: $R1 \leftarrow R1 - R2$

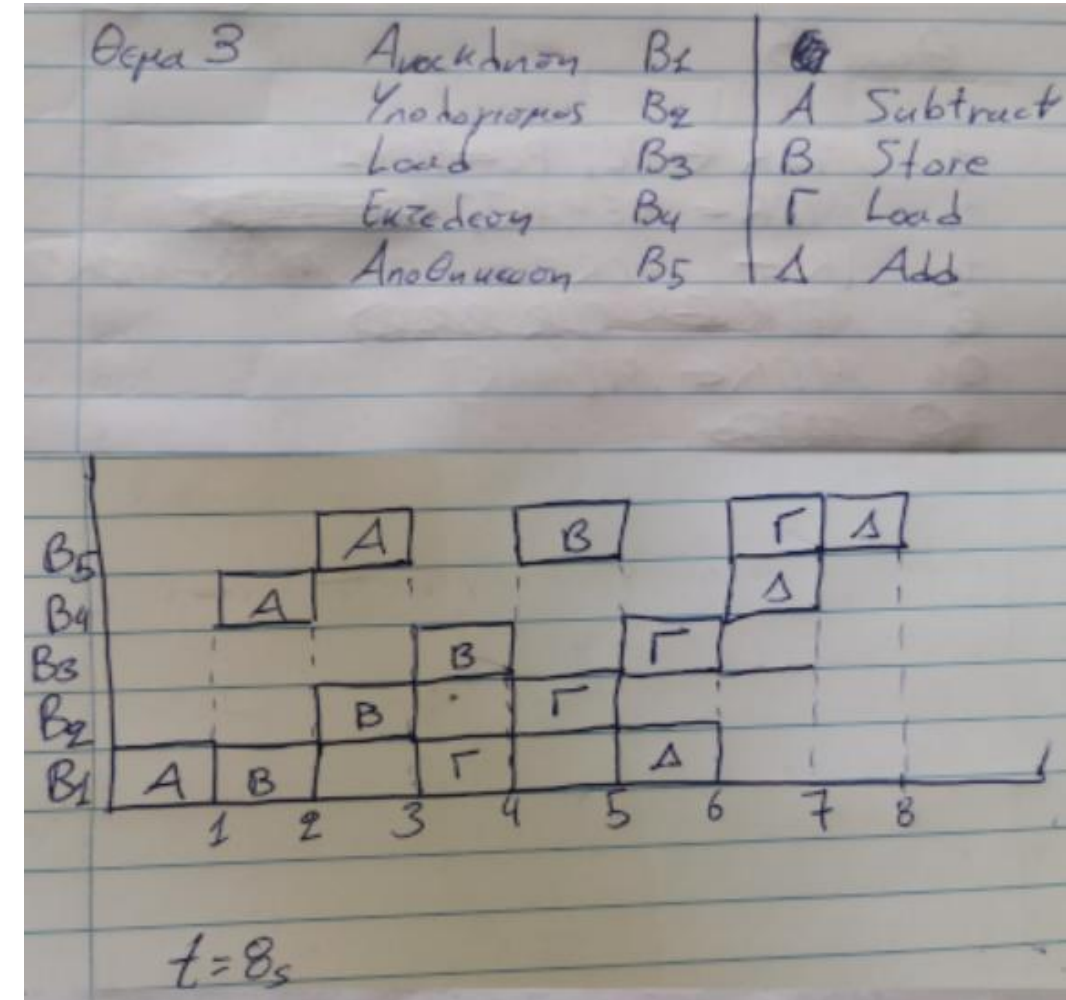
STORE: $X \leftarrow R1$

LOAD: $R1 \leftarrow X$

ADD: $R2 \leftarrow R1 + R2$

Σχεδιάστε ένα χρονικό διάγραμμα που να δείχνει την εκτέλεση αυτών των τεσσάρων εντολών, και υπολογίστε πόσο χρόνο χρειάζεται ο επεξεργαστής για την εκτέλεσή τους. Να υποθέσετε μέγιστη επικάλυψη των εντολών.

Λύση:



Ιούνιος 2022

Θέμα 4ο (24 μονάδες)

Δίνεται η κρυφή μνήμη των 64Kbytes με γραμμές των 32bytes. Ο επεξεργαστής παράγει διευθύνσεις των 20 δυαδικών ψηφίων. Δίνονται οι παρακάτω διευθύνσεις:

Δ1: 6F36A

Δ2: 87373

Δ3: 5F374

Δ4: 6F37B

Για κάθε μια από τις επόμενες δύο οργανώσεις κρυφής μνήμης:

A) Άμεση οργάνωση (direct mapped)

B) Οργάνωση 2-τροπων συνόλου συσχέτισης (2-way associative)

1) Μπορούν οι διευθύνσεις Δ1, Δ3, Δ4 να βρίσκονται ταυτόχρονα στην cache (κρυφή μνήμη)

1) Μπορούν οι διευθύνσεις Δ1, Δ2, Δ4 να βρίσκονται ταυτόχρονα στην cache (κρυφή μνήμη)

Λύση:

6F36A = 0110 1111 0011 0110 1010

87373 = 1000 0111 0011 0111 0011

5F374 = 0101 1111 0011 0111 0100

6F37B = 0110 1111 0011 0111 1011

A) $N = C/L = 2048 \text{ (64K/32)}$

4	11	5
---	----	---

Tag set offset

Δ1, Δ4 στο ίδιο tag αρα πάντα μαζί. Δ1, Δ4, Δ3 στο ίδιο σετ αρα δεν μπορούν να είναι μαζί. Άρα έχουμε Δ1, Δ2, Δ4 ή Δ2, Δ3 στην cache

	tag	set	offset
Δ1	(6)	(F3)011	0(A)
Δ2	(8)	(73)011	1(3)
Δ3	(5)	(F3)011	1(4)
Δ4	(6)	(F3)011	1(B)

B) $N = C/L * K = 1024 \text{ (64K/64)}$

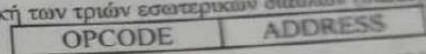
4	10	6
---	----	---

Tag set offset

Δ1, Δ4 στο ίδιο tag αρα πάντα μαζί. Δ1, Δ4, Δ3 στο ίδιο σετ αλλα μπορούν να είναι μαζί λόγω 2-way associative. Άρα έχουμε όλες τις διευθύνσεις ταυτόχρονα στην cache χωρίς πρόβλημα.

	tag	set	offset
Δ1	(6)	(F3)01	10(A)
Δ2	(8)	(73)01	11(3)
Δ3	(5)	(F3)01	11(4)
Δ4	(6)	(F3)01	11(B)

Θέμα 5^ο (21 μονάδες)
Δίδεται η παρακάτω αρχιτεκτονική των τριών εσωτερικών διαύλων (buses). Η μισή των απαντήσεων είναι:

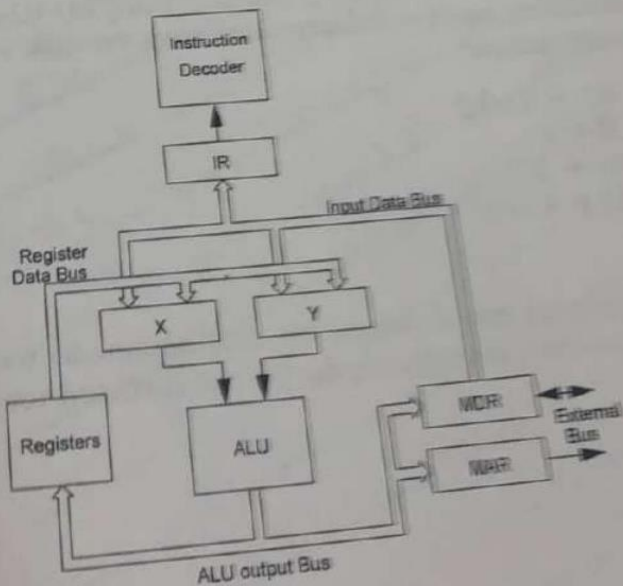


όπου το OPCODE είναι 8 bits και το ADDRESS είναι 24 bits. Κάθε λέξη που υπολογιστεί είναι 32 bits. Η εσωτερική αρχιτεκτονική της CPU δίνεται στο παρακάτω διάγραμμα.

Δώστε τον πλήρη κώδικα RTL για την ανάκτηση και εκτέλεση της εντολής

ADD D4, (D3)

όπου τα περιεχόμενα της θέσης μνήμης, η διεύθυνση της οποίας είναι στην D3 προστίθενται στα περιεχόμενα του D4 και το αποτέλεσμα πηγαίνει πάλι στον καταχωρητή D4. Υποθέστε ότι ο PC είναι συνδεδεμένος όπου χρειάζεται και περιλαμβάνει όλα τα hardware που απαιτούνται για τη διαχείρισή του. Σημ.: Για πλήρεις μονάδες πρέπει να δώσετε τον ελάχιστο αριθμό βημάτων.



Λύση:

Θέμα 5

$t_0: MAR \leftarrow PC; PC \leftarrow PC + 1$

$t_1: MDR \leftarrow M[MAR]$

$t_2: IR \leftarrow MDR[OPcode]; MAR \leftarrow MDR[Addr]$

$t_3: MDR \leftarrow M[MAR]$

$t_4: Y \leftarrow MDR; X \leftarrow D4$

$t_5: D4 \leftarrow Y + X$

Θέμα 1^ο (15 μονάδες)

Για καθεμία από τις παρακάτω αρχιτεκτονικές να γραφεί πρόγραμμα σε επίπεδο συμβολικής γλώσσας, με το ελάχιστο πλήθος εντολών, για τον υπολογισμό της έκφρασης:

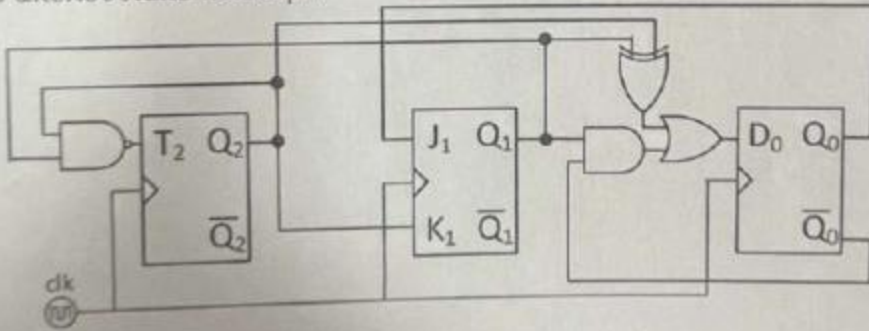
$$F = \frac{A - 3 \times B + C \times D}{C - B}$$

- α. Αρχιτεκτονική που βασίζεται στη χρήση συσσωρευτή.
- β. Αρχιτεκτονική που βασίζεται στη χρήση μηχανισμού στοίβας (stack). Σημειώνεται ότι στην αρχιτεκτονική αυτή, οι εντολές αριθμητικών πράξεων εκτελούνται επί των δύο κορυφαίων στοιχείων της στοίβας με δεύτερο τελεστέο κάθε πράξης αυτόν που βρίσκεται στην κορυφή του σωρού και πρώτο τον επόμενο μετά το κορυφαίο στοιχείο. Δηλαδή, για σωρό με περιεχόμενο $K_{n-1}, K_{n-2}, \dots, K_0$ όπου το K_{n-1} είναι το κορυφαίο στοιχείο (αυτό που εισήχθη στο σωρό πλέον πρόσφατα), η εκτέλεση της εντολής "OPER", που αντιστοιχεί στην υλοποίηση της πράξης "OPERATION", θα επιστρέψει το αποτέλεσμα (K_{n-2} OPERATION K_{n-1}).
- γ. Αρχιτεκτονική καταχωρητή-καταχωρητή (load/store) που διαθέτει εντολές με δύο τελούμενα.

Λύση:

Θέμα 2^ο (15 μονάδες)

Δίνεται το παρακάτω ακολουθιακό κύκλωμα.



Αρχικά οι τιμές των flip-flops είναι $Q_2Q_1Q_0=001$. Δώστε τις τιμές των flip-flops μετά από ένα, δύο και τρεις παλμούς του ρολογιού.

Λύση:

Θέμα 3° (20 μονάδες)
Ένα υπολογιστικό σύστημα έχει δίαυλο διευθύνσεων των 24 bits, δίαυλο δεδομένων των 8 bits και το σύστημα μνήμης του είναι οργανωμένο με ένα byte ανά θέση (δηλαδή σε κάθε byte αντιστοιχεί μια διεύθυνση). Για τις ακόλουθες περιπτώσεις κρυφών μνημών :

α. Μεγέθους 8 Kbytes με γραμμές των 16 bytes και άμεση οργάνωση, και
β. Μεγέθους 64Kbytes με γραμμές των 32 bytes και 2-τρόπων συνόλου συσχέτισης (2-way set associative) οργάνωση.

να καθορίσετε:

1. Τα πεδία από τα οποία θεωρούμε ότι αποτελείται η διεύθυνση του επεξεργαστή για την προσπέλαση της κρυφής μνήμης.
2. Ποια από τα περιεχόμενα των θέσεων μνήμης με διευθύνσεις
a: 52622060_h, d: 73512073_h,
b: 10603164_h, e: 52622077_h και
c: 66522077_h, f: 66554426_h,
είναι δυνατόν να βρίσκονται ταυτόχρονα σε κάθε κρυφή μνήμη. Να δικαιολογήσετε την απάντησή σας.

- a. 52622060 = 101 010 110 010 010 000 110 000
- b. 10603164 = 001 000 110 000 011 001 110 100
- c. 66522077 = 110 110 101 010 010 000 111 111
- d. 73512073 = 111 011 101 001 010 000 111 011
- e. 52622077 = 101 010 110 010 010 000 111 111
- f. 66554426 = 110 110 101 101 100 100 010 110

Λύση:

α) $N=C/L=128(8K/16)$

13	7	4
----	---	---

Tag set offset

ae ίδιο tag παντα μαζί,a,c,d,e
ίδιο set αρα δεν μπορούν να
είναι μαζί.Αρα στην cache θα
είναι ή ae,b,f ή c,b,f ή d,b,f

	tag	set	offset
a	(5262)0	10(2)11	0(0)
b	(1060)0	11(3)11	0(4)
c	(6652)0	10(2)11	1(7)
d	(7351)0	10(2)11	1(3)
e	(5262)0	10(2)11	1(7)
f	(6655)1	00(4)01	0(6)

β) $N=C/L*K=1024(64K/64)$

8	10	6
---	----	---

Tag set offset

ae ίδιο tag αρα πάντα μαζί,c,f
ίδιο tag αρα παντα
μαζί.Μπορούν όλες μαζί να
βρίσκονται στην cache χωρίς
πρόβλημα

	tag	set	offset
a	(52)11	0(220)	(60)
b	(10)11	0(031)	(64)
c	(66)10	1(220)	(77)
d	(73)10	1(120)	(73)
e	(52)11	0(220)	(77)
f	(66)10	1(544)	(6)

Θέμα 4^ο (20 μονάδες)

(α) Διαθέτετε Ολοκληρωμένα Κυκλώματα μνήμης ROM (ΟΚ₁) με οργάνωση 2048 θέσεων και 4 δυαδικά ψηφία ανά θέση. Χρησιμοποιώντας όσα αντίγραφα του ΟΚ₁ χρειαστείτε και συνδυαστικά κυκλώματα, να συνθέσετε μνήμη ROM χωρητικότητας 8 KBytes με οργάνωση ενός byte ανά θέση.

(β) Θεωρώντας την ROM που συνθέσατε στο (α) υποερώτημα ως έτοιμο ΟΚ₁ να την διασυνδέσετε σε έναν μικροεπεξεργαστή που διαθέτει δίαυλο δεδομένων 8 δυαδικών ψηφίων και δίαυλο διευθύνσεων 16 δυαδικών ψηφίων. Η ROM θα πρέπει να αποκρίνεται στο χώρο διευθύνσεων που ξεκινά από τη διεύθυνση 1000₁₆. Να δώσετε το σχέδιο της διασύνδεσης και να το αιτιολογήσετε συνοπτικά.

Θέμα 5^ο (15 μονάδες)

Λύση:

Θέμα 5^ο (15 μονάδες)

Μια μηχανή RISC έχει περίοδο ρολογιού 90ns. 25% των εντολών της είναι LOAD και STORE εντολές. Στις θυρίδες καθυστέρησης αυτών των εντολών τοποθετούνται κατά μέσο όρο 40% NO-OP εντολές και 60% χρήσιμες εντολές. Στο καινούργιο μοντέλο της μηχανής που βγαίνει στην αγορά η περίοδος έχει μειωθεί στα 70ns. Το κόστος όμως αυτής της μείωσης είναι να χρειάζονται πλέον δύο θυρίδες καθυστέρησης για κάθε εντολή μνήμης, και μόνο το ένα τέταρτο από όλες τις θυρίδες καθυστέρησης γεμίζει με χρήσιμες εντολές. Πια μηχανή είναι πιο γρήγορη και κατά πόσο;

Λύση:

Θέμα 6^ο (15 μονάδες)

Μια CPU έχει εντολές μήκους 12 bits. Το μέγεθος ενός πεδίου διεύθυνσης είναι 4 bits. Είναι δυνατόν να έχουμε:

14	εντολές δύο διευθύνσεων
29	εντολές μίας διεύθυνσης
48	εντολές μηδενικής διεύθυνσης

χρησιμοποιώντας αυτή τη μορφή εντολών;
Δικαιολογήστε λεπτομερώς την απάντησή σας.

Λύση:

Σεπτέμβριος 2023

Λύση:

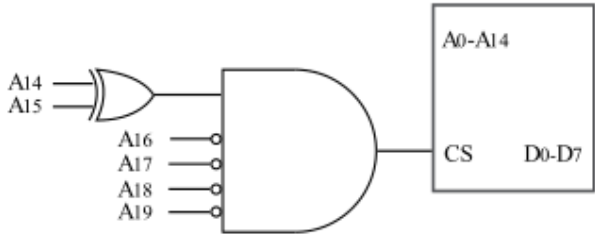
Θέμα 1^ο (20 μονάδες)

- α. Διαθέτετε ολοκληρωμένα κυκλώματα μνήμης ROM (OK1) μεγέθους 16Kbits με 4 δυαδικά ψηφία (bits) ανά θέση. Χρησιμοποιώντας όσα αντίγραφα του OK1 χρειαστείτε και συνδυαστικά κυκλώματα, να συνθέσετε μνήμη ROM χωρητικότητας 32 KBytes με οργάνωση ενός byte (8 bits) ανά θέση.
- β. Θεωρώντας την ROM που συνθέσατε στο (α) υποερώτημα ως έτοιμο OK, να την διασυνδέσετε σε έναν μικροεπεξεργαστή που διαθέτει δίαυλο δεδομένων 8 δυαδικών ψηφίων και δίαυλο διευθύνσεων 20 δυαδικών ψηφίων. Η ROM θα πρέπει να αποκρίνεται στο χώρο διευθύνσεων που ξεκινά από τη διεύθυνση 04000₁₆. Να δώσετε το σχέδιο της διασύνδεσης.

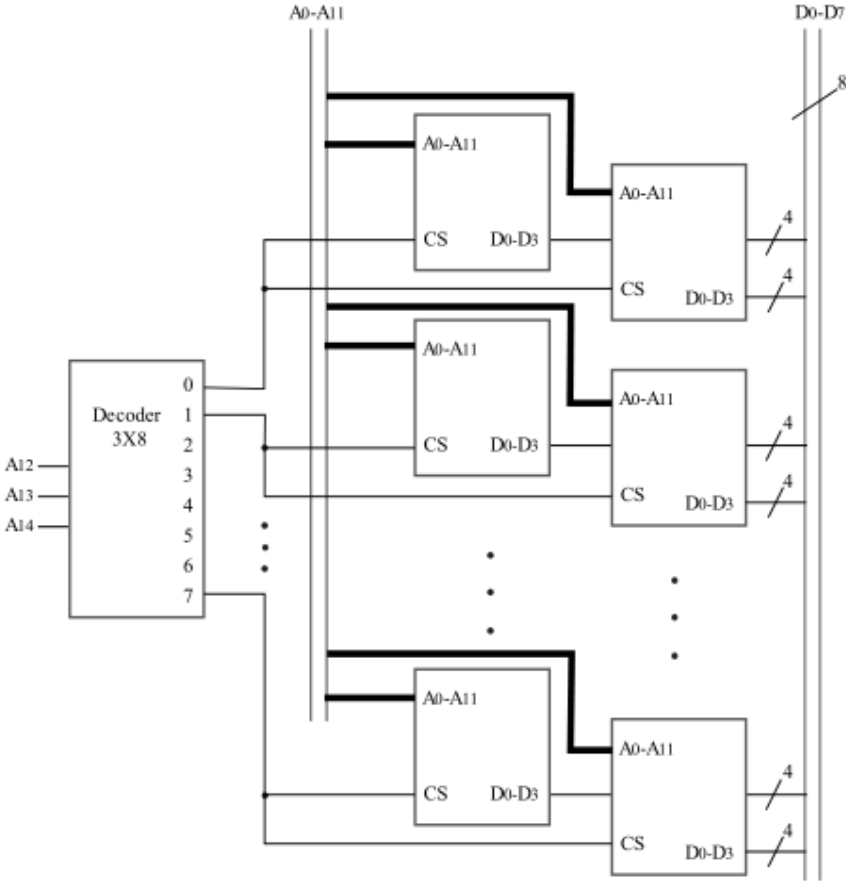
Λύση:

b) Χάρτης μνήμης:

04000	0000 0100 0000 0000 0000
07FFF	0000 0111 1111 1111 1111
08000	0000 1000 0000 0000 0000
0BFFF	0000 1011 1111 1111 1111



a)



Θέμα 2^ο (20 μονάδες)

Ένα υπολογιστικό σύστημα έχει αρτηρία διευθύνσεων των 24 δυαδικών ψηφίων, αρτηρία δεδομένων των 8 δυαδικών ψηφίων και το σύστημα μνήμης του είναι οργανωμένο με ένα byte ανά θέση (δηλαδή σε κάθε byte αντιστοιχεί μια διεύθυνση). Για τις ακόλουθες περιπτώσεις κρυφών μνημών :

- α. Μεγέθους 8 Kbytes με πλαίσια των 16 bytes και άμεση οργάνωση, και
- β. Μεγέθους 64Kbytes με πλαίσια των 32 bytes και 2-τρόπων συνόλου συσχέτισης (2-way set associative) οργάνωση, να καθορίσετε :
 1. Τα πεδία από τα οποία θεωρούμε ότι αποτελείται η διεύθυνση του επεξεργαστή για την προσπέλαση της κρυφής μνήμης.
 2. Ποια από τα περιεχόμενα των θέσεων μνήμης με διευθύνσεις
 - a: 74622061₈ d: 53512074₈,
 - b: 12603165₈ e: 74622070₈ και
 - c: 35522070₈ f: 35554427₈,

είναι δυνατόν να βρίσκονται ταυτόχρονα σε κάθε κρυφή μνήμη. Να δικαιολογήστε την απάντησή σας.

Λύση:

α1. $N=C/L=512$

11	9	4
tag	set	offset

- α2.
- a. (746) 01|0 (20) 11|0 (1)
 - b. (126) 00|0 (31) 11|0 (5)
 - c. (355) 01|0 (20) 11|1 (0)
 - d. (535) 00|1 (20) 11|1 (4)
 - e. (746) 01|0 (20) 11|1 (0)
 - f. (355) 10|1 (44) 01|0 (7)
- ae είναι στο ίδιο πλαίσιο άρα πάντα μαζί. c είναι στο ίδιο set άρα δεν μπορεί να είναι μαζί με ae. Επομένως ή **(ae)**, **b,d,f** ή **c,b,d,f**.

β1. $N=C/(KXL)=1024$

9	10	5
tag	set	offset

- β2.
- a. (746) | (220) 1|10 (1)
 - b. (126) | (031) 1|10 (5)
 - c. (355) | (220) 1|11 (0)
 - d. (535) | (120) 1|11 (4)
 - e. (746) | (220) 1|11 (0)
 - f. (355) | (544) 0|10 (7)
- ae είναι στο ίδιο πλαίσιο άρα πάντα μαζί. c είναι στο ίδιο set αλλά το set έχει 2 πλαίσια αρα μπορεί να είναι μαζί με (ae). Επομένως **όλες** μπορούν να είναι ταυτόχρονα στην cache.

Θέμα 3^ο (15 μονάδες)

Ένας υπολογιστής των δύο εσωτερικών διαύλων (buses) έχει εντολές της παρακάτω μορφής:

OPCODE	ADDRESS
--------	---------

Η διεύθυνση ADDRESS δεν είναι απόλυτη διεύθυνση αλλά σχετική. Αυτό σημαίνει ότι προστίθεται στον PC για να υπολογιστεί η απόλυτη διεύθυνση του παράγοντα.

Δώστε σε RTL περιγραφή της συνολικής εκτέλεσης της εντολής ADD A. Η εντολή αυτή προσθέτει τον παράγοντα στο συσσωρευτή (accumulator) και βάζει το αποτέλεσμα πάλι στον συσσωρευτή.

Λύση:

Χρειάζεται βοηθητικός καταχωρητής (έστω Y) ανάμεσα στο bus εξόδου καταχωρητών και την ALU.

- 1. MAR←PC
- 2. MDR←M[MAR];PC←PC+1
- 3. IR←MDR(opcode)
- 4. TMP←PC
- 5. MAR←TMP+MDR(address)
- 6. MDR←M[MAR]
- 7. AC←AC+MDR;end

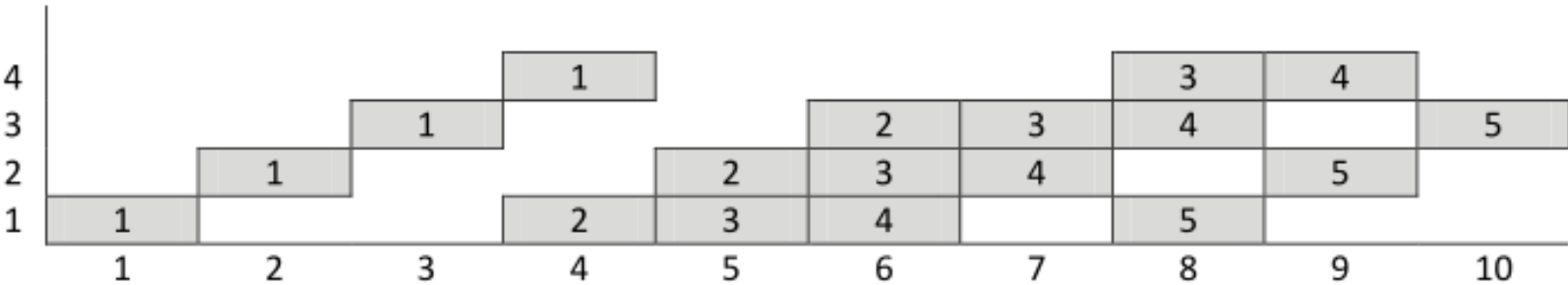
Θέμα 4^ο (15 μονάδες)

Ένας τυπικός επεξεργαστής RISC χρειάζεται 4 στάδια (ανάκληση εντολής, υπολογισμό διεύθυνσης, load, αποθήκευση αποτελέσματος) για τις εντολές LOAD και STORE, και 3 στάδια (ανάκληση εντολής, εκτέλεση, αποθήκευση αποτελέσματος) για κάθε μία από τις άλλες εντολές. Εξετάστε το παρακάτω πρόγραμμα:

```
LOAD:      R1 ← X
SUBTRACT:  R1 ← R1-R2
LOAD:      R2 ← X
STORE:     X ← R1
ADD:       R2 ← R1+R2
```

Λύση:

Η εξάρτηση των δεδομένων των καταχωρητών οδηγεί στο ακόλουθο χρονοδιάγραμμα (Τα βήματα δεν αντιστοιχούν επακριβώς σε λειτουργίες αλλά οι χρόνοι είναι σωστοί).



Συνολικός χρόνος εκτέλεσης 10 παλμοί.

Φεβρουάριος 2024

Θέμα 5^ο (15 μονάδες)

Διαθέτετε πλήρεις αθροιστές του ενός δυαδικού ψηφίου, μία AND πύλη και 3 inverters (αντιστροφείς). Να σχεδιάσετε με αυτά ένα κύκλωμα που να εκτελεί τις λειτουργίες του ακόλουθου πίνακα, όπου X ένας ακέραιος αριθμός των 4 δυαδικών ψηφίων και S_1, S_0 σήματα επιλογής της επιθυμητής λειτουργίας.

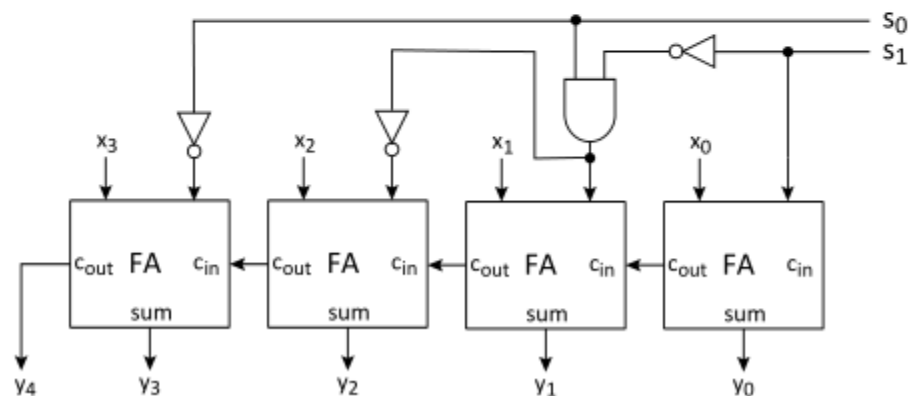
$S_1 S_0$	Αειτουργία
00	X - 4
01	X + 2
10	X - 3
11	X + 5

Θεωρήστε ότι η είσοδος X και η έξοδος του κυκλώματός σας είναι κωδικοποιημένες σε κώδικα "συμπλήρωμα ως προς 2" και αγνοήστε την περίπτωση υπεργείλισης.

Λύση: Για τους αρνητικούς αριθμούς προστίθεται το συμπλήρωμα ως προς 2:

S_1S_0	Προσθετέος
00	1100
01	0010
10	1101
11	0101

Και το κύκλωμα είναι:



Θέμα 6^ο (15 μονάδες)

Θεωρήστε ένα υπολογιστή που βασίζεται στη χρήση καταχωρητών γενικού σκοπού και του οποίου το σύνολο των εντολών σε επίπεδο γλώσσας μηχανής αποτελείται από 190 εντολές, το μήκος κάθε εντολής είναι ακέραιο πολλαπλάσιο του οκτώ, και διαθέτει 16 καταχωρητές με μήκος 32 δυαδικών ψηφίων ο καθένας. Η αρτηρία διευθύνσεων είναι των 32 δυαδικών ψηφίων.

Για κάθε μία από τις ακόλουθες εντολές σε συμβολική γλώσσα να σχεδιάσετε τη μορφή των εντολών σε επίπεδο γλώσσας μηχανής (δηλαδή να δώσετε τα πεδία από τα οποία αποτελείται κάθε εντολή σε επίπεδο γλώσσας μηχανής, το εύρος κάθε πεδίου και τι δηλώνει κάθε πεδίο) ώστε να μπορεί να διευθυνσιοδοτηθεί ολόκληρη η κύρια μνήμη.

- 1) LOAD R2, A ; R ← A, όπου A διεύθυνση θέσης μνήμης (κατ' ευθείαν τρόπος διευθυνσιοδότησης)
- 2) LOAD R1, (R2) ; R1 ← M(R2), έμμεσος (indirect) τρόπος διευθυνσιοδότησης με χρήση καταχωρητή
- 3) LOAD R3, #D ; άμεσος (immediate) τρόπος διευθυνσιοδότησης, D είναι αριθμός 12 bits.

Λύση:
2^8=256>190 αρα 8 bit opcode.2^4=16 αρα 4 bits καταχωρητες.

1)	8bit opcode	4bit Register	32bit Address	4bit useless	Μήκος:6Bytes
2)	8bit opcode	4bit Register	4bit Register		Μήκος:2Bytes
3)	8bit opcode	4bit Register	12bit offset		Μήκος:3Bytes

Πρόβλημα 1 (Μονάδες 15)

Για καθεμία από τις παρακάτω αρχιτεκτονικές να γραφεί πρόγραμμα σε επίπεδο συμβολικής γλώσσας, με το ελάχιστο πλήθος εντολών, για τον υπολογισμό της έκφρασης:

$$F = \frac{A - B/C + D \times (E - F)}{B - A}$$

- α. Αρχιτεκτονική που βασίζεται στη χρήση συσσωρευτή.
- β. Αρχιτεκτονική που βασίζεται στη χρήση μηχανισμού στοίβας (stack). Σημειώνεται ότι στην αρχιτεκτονική αυτή, οι εντολές αριθμητικών πράξεων εκτελούνται επί των δύο κορυφαίων στοιχείων του σωρού με πρώτο τελεστέο κάθε πράξης αυτόν που βρίσκεται στην κορυφή του σωρού και δεύτερο τον επόμενο μετά το κορυφαίο στοιχείο. Δηλαδή, για σωρό με περιεχόμενο $K_{n-1}, K_{n-2}, \dots, K_0$ όπου το K_{n-1} είναι το κορυφαίο στοιχείο (αυτό που εισήχθη στο σωρό πλέον πρόσφατα), η εκτέλεση της εντολής "OPER", που αντιστοιχεί στην υλοποίηση της πράξης "OPERATION", θα επιστρέψει το αποτέλεσμα $(K_{n-1} \text{ OPERATION } K_{n-2})$.
- γ. Αρχιτεκτονική που βασίζεται σε καταχωρητές γενικού σκοπού και διαθέτει εντολές με δύο τελούμενα.

Λύση:

Θέμα 2 (20 μονάδες)
Σε ένα σύστημα επεξεργαστή με κρυφή μνήμη, σε κάθε byte αντιστοιχεί μία διεύθυνση (έχουμε 1 byte ανά θέση μνήμης) και οι διευθύνσεις που παράγει ο επεξεργαστής είναι των 24 δυαδικών ψηφίων. Η κρυφή μνήμη είναι χωρητικότητας 4KBytes και με 32-bytes ανά πλαίσιο. Δίνονται οι παρακάτω διευθύνσεις μνήμης:

α. 473A85
β. 473A9F
γ. 7F2A98
δ. 7F2A9A
ε. AEC7EB

και οι παρακάτω διατάξεις κρυφής μνήμης:

Δ1: Άμεσης οργάνωσης
Δ2. 2-τρόπων οργάνωσης συνόλων

Για κάθε μία από τια διατάξεις αυτές, να δώσετε την ανάλυση της διεύθυνσης και να εξετάσετε αν όλες οι παραπάνω διευθύνσεις μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη.

Λύση:

Δ1: $N=C/L=128$ (4096/32)

12	7	5
----	---	---

Tag(υπολοιπο) set logN offsetLogL

12+7+5=24 που δινεται στην εκφώνηση πως είναι το μέγεθος της διευθυνσης

L:μεγεθος γραμμης
N:αριθμος συνολου
K:γραμμες/set
C:K*N*L

TAG	SET	OFFSET
(473)	(A)100	0(5)
(473)	(A)100	1(F)
(7F2)	(A)100	1(8)
(7F2)	(A)100	1(A)
(AEC)	(7)111	0(B)

γ,δ στο ίδιο πλαίσιο (tag) αρα πάντα μαζί,αβ στο ίδιο πλαίσιο άρα πάντα μαζί.αβ ίδιο set με γ,δ άρα δεν μπορούν να είναι μαζί.Επομένως έχουμε γ,δ,ε ή α,β,ε

Μετατρέπω σε δυαδικό συστημα:
473A85:0100 0111 0011 1010 1000 0101
473A9F:0100 0111 0011 1010 1001 1111
7F2A98:0111 1111 0010 1010 1001 1000
7F2A9A:0111 1111 0010 1010 1001 1010
AEC7EB:1010 1110 1100 0111 1110 1011

Δ2: $N=C/L*K=64$

13	6	5
----	---	---

Tag(υπολοιπο) set logN offsetLogL

γ,δ στο ίδιο πλαίσιο (tag) αρα πάντα μαζί,αβ στο ίδιο πλαίσιο αρα πάντα μαζί.αβ ίδιο set με γ,δ αλλά το set έχει δύο πλαίσια άρα μπορεί να είναι μαζί .Άρα μπορούν να είναι όλες μαζί στην μνημη cache.

TAG	SET	OFFSET
(473)1	010100	00101
(473)1	010100	11111
(7F2)1	010100	00011
(7F2)1	010100	11010
(AEC)0	111111	01011

Θέμα 3 (15 μονάδες)

Έχετε στη διάθεση σας έναν αποκωδικοποιητή με τρεις εισόδους, έναν πολυπλέκτη με δύο εισόδους ελέγχου και μία λογική πύλη (όποια επιθυμείτε). Να υλοποιήσετε τη συνάρτηση $F = f_1(x, y, z) + f_2(A, B, C)$ με $f_1(x, y, z) = xy' + xyz'$ και $f_2(A, B, C) = A'BC + AB' + ABC'$

Να θεωρήσετε ότι είναι διαθέσιμες και οι συμπληρωματικές τιμές των εισόδων.

Λύση:

Θέμα 4 (20 μονάδες)

Έχετε στη διάθεσή σας ολοκληρωμένα κυκλώματα μνήμης OK1, με χωρητικότητα 16 Mbits και 8 δυαδικά ψηφία ανά θέση μνήμης καθώς και ολοκληρωμένα κυκλώματα μνήμης OK2 με χωρητικότητα 512 KBytes και 4 δυαδικά ψηφία ανά θέση μνήμης. Επίσης έχετε στη διάθεσή σας ένα αποκωδικοποιητή (όποιον επιθυμείτε) και δύο πύλες (και πάλι όποιες επιθυμείτε). Να σχεδιάσετε σύστημα μνήμης με χωρητικότητα 36 Mbits και 12 δυαδικά ψηφία ανά θέση μνήμης χρησιμοποιώντας το ελάχιστο πλήθος ολοκληρωμένων κυκλωμάτων. Η μικρότερη διεύθυνση του συστήματος μνήμης που θα σχεδιάσετε να είναι η μηδενική. Οι διευθύνσεις που παράγει ο επεξεργαστής είναι των 23 δυαδικών ψηφίων.

- α) Πόσα OK κάθε τύπου θα χρησιμοποιήσετε;
- β) Δώστε το σχεδιάγραμμα του κυκλώματος

Λύση:

Θέμα 5 (15 μονάδες)
Ένας υπολογιστής των 2 εσωτερικών διαύλων (buses) έχει εντολές της παρακάτω μορφής:

OPCODE	ADDRESS
--------	---------

Δώστε την εκτέλεση της εντολής POP A. Η εντολή αυτή εξάγει την τιμή που υπάρχει στην κορυφή της στοίβας (stack) και την τοποθετεί στην θέση μνήμης A. Η κορυφή της στοίβας περιέχεται στον καταχωρητή SP ο οποίος θα πρέπει να μειωθεί κατά 1 αφού εξαχθεί ο παράγοντας. Χρησιμοποιείστε RTL για να δείξετε την εκτέλεση της εντολής.

Λύση:

Θέμα 6 (15 μονάδες) Μόνο για το Τμήμα Β'
Το Τμήμα Β' (Κουλ.-Πετ.) λύνει και αυτή την άσκηση.
(Η άσκηση ΔΕΝ θα βαθμολογηθεί για το Τμήμα Γ' (Πλ.Ω))

Η μνήμη ενός διασωληνωμένου (pipelined) διανυσματικού υπολογιστή (vector processor) έχει 8 αρθρώματα (modules). Το διάβασμα από τη μνήμη διαρκεί 3 clock cycles και το γράψιμο στην μνήμη διαρκεί 2 clock cycles, ενώ το pipeline χρειάζεται 4 clock cycles. Θέλουμε να εκτελέσουμε την πράξη: $C=A+B$ όπου A , B , και C είναι διανύσματα (vectors). Τα δεδομένα είναι αποθηκευμένα στην μνήμη με τον ακόλουθο τρόπο:

Module 0	A[0]	A[8]			B[6]			C[4]		
Module 1	A[1]	A[9]			B[7]			C[5]		
Module 2	A[2]				B[0]	B[8]		C[6]		
Module 3	A[3]				B[1]	B[9]		C[7]		
Module 4	A[4]				B[2]			C[0]	C[8]	
Module 5	A[5]				B[3]			C[1]	C[9]	
Module 6	A[6]				B[4]			C[2]		
Module 7	A[7]				B[5]			C[3]		

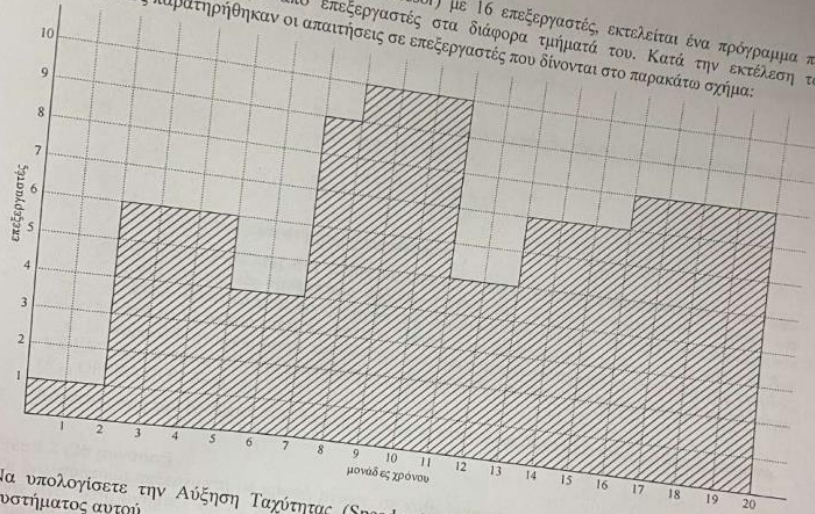
Χρησιμοποιήστε πίνακα, για να δείξετε πώς θα διαβαστούν οι πίνακες A και B από την μνήμη και πώς θα γραφεί ο πίνακας C στην μνήμη. Δώστε επίσης την καθυστέρηση (delay) των vectors που χρειάζεται για σωστή είσοδο στο pipeline καί γράψιμο στην μνήμη.

Λύση:

Ιούνιος 2024

Θέμα 6 (15 μονάδες) Μόνο για το Τμήμα Γ'
Το Τμήμα Γ' (Πλ.-Ω), λύνει και αυτή την άσκηση.
(Η άσκηση ΔΕΝ θα βαθμολογηθεί για το Τμήμα Β' (Κοιλ.-Πετ.))

Σε έναν παράλληλο επεξεργαστή (multiprocessor) με 16 επεξεργαστές, εκτελείται ένα πρόγραμμα που απαιτεί διαφορετικό αριθμό από επεξεργαστές στα διάφορα τμήματά του. Κατά την εκτέλεση του προγράμματος παρατηρήθηκαν οι απαιτήσεις σε επεξεργαστές που δίνονται στο παρακάτω σχήμα:



Να υπολογίσετε την Αύξηση Ταχύτητας (Speed-up) και τον Βαθμό Χρήσης (Efficiency) του συστήματος αυτού.

Λύση: