

2η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ Ακ. έτος 2018-2019, 5ο εξάμηνο

ΣΗΜΜΥ

ΜΕΡΟΣ Ι

Λυνω την ασκηση με $\chi=5$.

Συμφωνα με την εκφωνηση εχουμε

$5+6=11$ bits για τον εκθετη

20 bits για το κλασμα

1 bit προσημου

Στον υπολογιστη αυτο θα αναπαρασταθει ως

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---------------|---|---|---|---|---|---|---|---|---|---|---------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | |
| π | εκθετης 11bit | | | | | | | | | | | κλασμα 20 bit | | | | | | | | | | | | | | | | | | | |

A)

Καθε αριθμος που μπορουμε να αναπαραστησουμε με το προτυπο της κινητης υποδιαστολης ειναι της μορφης

$(-1)^{\pi} \cdot 1.xxxxxxxxxxxxxxxxxxxxx \cdot 2^{(yyyyyyyyyy - \text{Πολωση})}$

20 bit

11 bit

Οπου η πολωση ειναι μαι συμβαση που χρησημοποιουμε για να αναπαραστησουμε τους αρνητικους εκθετες μεσω θετικων αριθμων ετσι ωστε να ειναι ευκολοτερη η συγκριση τους.

Ο μεγατος προσημασμενος αριθμος που μπορουμε να αναπαραστησουμε με 11 bit ειναι το $0111111111 = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 + 2^7 + 2^8 + 2^9 + 2^{10} = 2^{11} - 2$

Θελουμε δηλαδη να αντιστοιχισουμε το ευρος $\{-2^{11-2}, 2^{11-2}\}$ στον αξονα $\{0, 2^{12-2}\}$. Αρα η πολωση μας θα ειναι το μισο του μεγατου αριθμου

Πολωση $= (2^{12-2})/2$ σε δεκαδικο 111111111 σε δυαδικο

Τωρα εαν τα bit του εκθετη παρουν την τιμη 0000000000, ο εκθετης θα γινει $0 - (2^{12-2})/2$

B)

Ο μεγαλυτερος κατα απολυτη τιμη παραστησιμος αριθμος εχει κλασμα 20 ασσους

Και εκθετη 11 ασσους

Αρα θα είναι $1,1111\dots111 = 2^0 + 2^{-1} + 2^{-2} + \dots + 2^{-19} + 2^{-20} = 2^1 - 2^{-20}$

Και εκθετης $1111\dots111 = 2^{12-2}$

Αρα ο μεγαλύτερος κατά απόλυτη τιμή παραστησιμος αριθμος είναι $(2 - 2^{-20}) * 2^{((2^{12}-2)/2)}$

Ο μικροτερος κατά απόλυτη τιμή παραστησιμος αριθμος έχει κλασμα 20 μηδενικα και εκθετη 11 μηδενικα

Κλασμα $1,000\dots000 = 1$

Εκθετης-πολωση $= 0 - (2^{12}-2)/2$

Αρα θα είναι ισος με $1 * 2^{-(2^{12}-2)/2}$

Γ)

Ο μικροτερος παραστησιμος αριθμος είναι ο $1 * 2^{-(2^{12}-2)/2}$

Και ο αμεσως επομενος είναι ο ιδιος αριθμος, αλλά στο less significant bit του κλασματος έχει ασσο ατνι για μηδεν δηλαδή

$1,000\dots0001 = 1 + 2^{-20}$

Και ίδια εκθετη

Αρα $(1 + 2^{-20}) * 2^{-(2^{12}-2)/2}$

Η ακριβεις λοιπον εινναι

$(1 + 2^{-20}) * 2^{-(2^{12}-2)/2} - 1 * 2^{-(2^{12}-2)/2} = 2^{-20} * 2^{-(2^{12}-2)/2}$

ΜΕΡΟΣ II

| CLOCK CIRCLE/εντολη | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---------------------|----|----|----|-----|-----|----|----|-----|-----|----|----|-----|----|----|-----|-----|
| Addi | IF | ID | EX | MEM | WB | | | | | | | | | | | |
| Lw | | IF | ID | EX | MEM | WB | | | | | | | | | | |
| Addi | | | IF | ID | ID | ID | EX | MEM | WB | | | | | | | |
| Lw | | | | IF | IF | IF | ID | EX | MEM | WB | | | | | | |
| Sub | | | | | | | IF | ID | ID | ID | EX | MEM | WB | | | |
| Sw | | | | | | | | IF | IF | IF | ID | ID | ID | EX | MEM | WB |
| Addi | | | | | | | | | | | IF | IF | IF | ID | EX | MEM |
| Sub | | | | | | | | | | | | | | IF | ID | ID |
| Bne | | | | | | | | | | | | | | | IF | IF |
| Lw | | | | | | | | | | | | | | | | |

| CLOCK CIRCLE/εντολη | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
|---------------------|----|-----|----|----|-----|-----|----|----|-----|----|----|-----|----|----|----|-----|----|
| Addi | | | | | | | | | | | | | | | | | |
| Lw | | | | | | | | | | | | | | | | | |
| Addi | | | | | | | | | | | | | | | | | |
| Lw | | | | | | | | | | | | | | | | | |
| Sub | EX | MEM | WB | | | | | | | | | | | | | | |
| Sw | ID | ID | ID | EX | MEM | WB | | | | | | | | | | | |
| Addi | IF | IF | IF | ID | EX | MEM | WB | | | | | | | | | | |
| Sub | | | | IF | ID | ID | ID | EX | MEM | WB | | | | | | | |
| Bne | | | | | IF | IF | IF | ID | ID | ID | EX | MEM | WB | | | | |
| Lw | | | | | | | | | | | | | IF | ID | EX | MEM | WB |

Στην ασκηση αυτη παρατηρουμε οτι μερικες εντολες για να εκτελεστουν χρειαζονται δεδομενα τα οποια κατασκευαζονται στη ακριβως επομενη εντολη

π.χ.η δευτερη addi θελει να προσθεσει $\$t0 = \$t0 + 1$, αλλα την σωστη τιμη του $\$t0$ θα την παρουμε απο την μνημη και θα την γραψουμε στους καταχωρητες στον 5^ο κυκλο, δηλαδη στο WB τμημα της load, αρα αναγκαστικα θα υπαρξει καθυστερηση και οι ID και IF των επομενων εντολων θα παραμηνουν στα σταδια τους και θα εκτελεστουν οταν τελειωσει ο πεμπτος κυκλος.

Αντιστοιχα Read After Write hazards εχουμε και στις περιπτωσεις που κανουμε

1)Lw παραγει τον $\$t0$ που χρειαζεται στην addi αμεσως μετα

2)Lw παραγει τον $\$t1$ που χρειαζεται στην sub αμεσως μετα

3)Sub παραγει τον $\$t0$ που χρειαζεται στην sw αμεσως μετα

4)Addi παραγει τον $\$t2$ που χρειαζεται στην sub αμεσως μετα

5)Sub παραγει τον $\$t4$ που χρειαζεται στην bne αμεσως μετα

Αφου καθε επαναληψη θελει 21 κυκλους εκτος απο την πρωτη φορα που θελει 22, και ο \$t2 αρχικα διαφερει απο τον \$t3 100 και αυξανεται κατα 4 καθε κυκλο, το προγραμμα θα εκτελεστει 25 φορες, και 1 κυκλο για την πρωτη addi εξω απο την λουπα

$$21 \cdot 24 + 22 + 1 = 527$$

| CLOCK CIRCLE/εντολη | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---------------------|----|----|----|-----|-----|----|-----|-----|----|-----|-----|-----|-----|-----|----|----|----|-----|
| Addi | IF | ID | EX | MEM | WB | | | | | | | | | | | | | |
| Lw | | IF | ID | EX | MEM | WB | | | | | | | | | | | | |
| Addi | | | IF | ID | ID | EX | MEM | WB | | | | | | | | | | |
| Lw | | | | IF | IF | ID | EX | MEM | WB | | | | | | | | | |
| Sub | | | | | | IF | ID | ID | EX | MEM | WB | | | | | | | |
| Sw | | | | | | | IF | IF | ID | EX | MEM | WB | | | | | | |
| Addi | | | | | | | | | IF | ID | EX | MEM | WB | | | | | |
| Sub | | | | | | | | | | IF | ID | EX | MEM | WB | | | | |
| Bne | | | | | | | | | | | IF | ID | EX | MEM | WB | | | |
| Lw | | | | | | | | | | | | | | | IF | ID | EX | MEM |

Για να αποφυγουμε τα παραπανω hazards καναμε προωθηση, καθως τα αποτελεσματα των πραξεων τα «ξερούμε» 1 η 2 κυκλους πριν γραφτουν στους καταχωρητες

Π.χ η addi παραγει το σωστο αποτελεσμα στο τελος του ex κυκλου της, αρα δεν ειναι αναγκη να περιμενουμε να γινει το wb για να συνεχισουμε

Λυσαμε τα hazards 3,4,5 με EX to EX προωθηση και ελατωσαμε τους κυκλους καθυστερησης των hazards 1 και 2 με MEM to EX προωθηση.

Τωρα πλεον απαιτουνται 13 κυκλοι για καθε εντολη, ετος απο την πρωτη που θελουμε 14

και 1 κυκλο για την πρωτη addi εξω απο την λουπα

συνολο $13 \cdot 24 + 14 + 1 = 302$ κυκλοι

Γ)

Παρατηρουμε οτι τα hazards τα οποια δεν μπορουμε να απαληψουμε με προωθηση ειναι αυτα που δημιουργουνται οταν θελουμε να χρησιμοποιησουμε μια τιμη ενος register, την οποια την φερνουμε απο την μνημε με lw στον αμεσως προηγουμενο κυκλο. Αυτο συμβαινει καθως η σωστη τιμη γινεται γνωστη στο τελος του MEM κυκλου και οχι του EX οπως για παραδειγμα στην addi.

Για να επιταχυνουμε το προγραμμα μας μπορουμε να αλλαξουμε την σειρα των εντολων ετσι ωστε να μην υπαρχει αυτο το μοτιβο.

Πιο συγκεκριμενα, εαν αλλαξουμε θεσεις στην πρωτη lw με την πρωτη addi μεσα στην L, δεν θα υπαρχει αυτο το προβλημα

addi \$t3, \$t2, 100

```
L:    lw $t0, 0($t2)
      lw $t1, 4($t2)
      addi $t0, $t0, 1
      sub $t0, $t0, $t1
      sw $t0, 0($s0)
      addi $t2, $t2, 4
      sub $t4, $t3, $t2
      bne $t4, $zero, L
```

| CLOCK CIRCLE/εντολη | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|------------------------|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|
| Addi | IF | ID | EX | MEM | WB | | | | | | | | | | | | |
| Lw | | IF | ID | EX | MEM | WB | | | | | | | | | | | |
| Lw | | | IF | ID | EX | MEM | WB | | | | | | | | | | |
| Addi | | | | IF | ID | EX | MEM | WB | | | | | | | | | |
| Sub | | | | | IF | ID | EX | MEM | WB | | | | | | | | |
| Sw | | | | | | IF | ID | EX | MEM | WB | | | | | | | |
| Addi | | | | | | | IF | ID | EX | MEM | WB | | | | | | |
| Sub | | | | | | | | IF | ID | EX | MEM | WB | | | | | |
| Bne | | | | | | | | | IF | ID | EX | MEM | WB | | | | |
| Lw | | | | | | | | | | | | | IF | ID | EX | EX | EX |

Καθως αλλαζουμε το προγραμμα αλλαζουν και οι προωθησεις που πρεπει να κανουμε για να παραμηνει σωστο, συγκεκριμενα

Πρεπει να κανουμε μια MEM to EX προωθηση για να δωσουμε στην πρωτη addi της λουπας το \$t0 και μια EX to EX για να δωσουμε στην sub παλι το \$t0

Η καθε λουπα κραται 11 κυκλους εκτος απο την ρωτη που κραται 12

και 1 κυκλο για την πρωτη addi εξω απο την λουπα

$11 \cdot 24 + 12 + 1 = 277$ κυκλους