



National
Technical
University of
Athens

Βικέντιος Βιτάλης el18803

2^η Σειρά Ασκήσεων Αρχιτεκτονική Υπολογιστών

Ημερομηνία Παράδοσης 5/1/2021

Άσκηση 2.1

Για την επίλυση της άσκησης αυτής, δεδομένου του τελευταίο ψηφίου του Αριθμού Μητρώου(AM), έχουμε $x = 3$.

Άρα έχουμε πρότυπο κινητής υποδιαστολής:

$22 + 3 = \mathbf{25 \text{ bits}}$, κατ' αναλογία με το IEEE 754

Ο Εκθέτης έχεις μήκος $5 + 3 = \mathbf{8 \text{ bits}}$, το κλάσμα αποτελείται από **16 bits** και το πρόσημο από **1 bit**.

Υποθέτουμε την παρακάτω αρχιτεκτονική:

s	e	f
1 bit	8 bits	16 bits

Βάση της παραπάνω αρχιτεκτονικής που υποθέσαμε, θα έχουμε:

Το σύστημά μας έχει 8 bits για τον εκθέτη, επομένως το εύρος των τιμών που μπορεί να πάρει ένας προσημασμένος εκθέτης είναι: $-128 \leq e \leq 127$. Στην πραγματικότητα ο εκθέτης θα λειτουργεί ως μη προσημασμένος και θα κινείται από το 1 έως το 255.

Για την πόλωση,

$$p = 2^{e-1} - 1 = 2^{8-1} - 1 = 2^7 - 1 = 128 - 1 = 127$$

Συνεπώς η τιμή του αριθμού V σε κάθε περίπτωση ορίζεται ανάλογα την συνθήκη:

$$1. \ 0 < e < 255: \quad V = (-1)^s * 2^{e-127} * (1, f) \text{ (Normal)}$$

$$2. \ e = 255, f \neq 0: \quad V = \text{NaN}$$

$$3. \ e = 255, f = 0, s = 1: \quad V = -\infty$$

$$4. \ e = 255, f = 0, s = 0: \quad V = +\infty$$

$$5. \ e = 0, f = 0, s = 0: \quad V = +0$$

$$6. \ e = 0, f = 0, s = 1: \quad V = -0$$

$$7. \ e = 0, f \neq 0: \quad V = (-1)^s * 2^{1-p} * (0, f)$$

$$V = (-1)^s * 2^{-126} * (0, f) \text{ (Subnormal)}$$

Η ακρίβεια μεταξύ Normal:

$$D_{\text{normal}} = (1,00\dots0) * 2^{1-p} = 2^{-126}$$

Η ακρίβεια μεταξύ Subnormal

$$D_{\text{subnormal}} = (0,0\dots01) * 2^{1-p} = 2^{-126} * 2^{-16}, \text{οπότε}$$

$$D_{\text{subnormal}} = 2^{-142}$$

Μικρότερη τιμή κατά απόλυτη τιμή

Εκθέτης = 00000001 -> Πραγματικός εκθέτης = $1 - 127 = -126$

Κλάσμα: 000...00 -> Σημαντικό = 1.0

Άρα, $\pm 1.0 * 2^{-126}$

Μεγαλύτερη τιμή κατά απόλυτη τιμή

Εκθέτης = 11111110 -> Πραγματικός εκθέτης = $254 - 127 = 127$

Κλάσμα: 111...11 -> Σημαντικό ~ 2.0

Άρα, $\pm 2.0 * 2^{127}$

Άσκηση 2.2

(α) Σε ένα φύλλο Excel έχουμε τοποθετήσει τις εντολές και παρουσιάσει το διάγραμμα χρονισμού. Τα πράσινα κελιά αναπαριστούν το οριστικό τέλος μιας εντολής. Τα κουτάκια “stall” είναι τα σημεία της εντολής κατά τα οποία εισάγεται καθυστέρηση. Στις εντολές με κόκκινα γράμματα έχουμε κάνει “bold” τους καταχωρητές οι οποίοι έχουν εξάρτηση από προηγούμενη εντολή. Οι καθυστερήσεις παρατηρούνται διότι δεν έχουμε σχήματα προώθησης στη διάθεσή μας, γεγονός

που δεν μας επιτρέπει να έχουμε στην διάθεσή μας δεδομένα καταχωρητή σε επόμενη εντολή πριν το στάδιο “Write Back”.

[illegible]

Χωρίς να ζητείται από την εκφώνηση, παραθέτουμε και την εκτέλεση της δεύτερης επανάληψης για λόγους πληρότητας:

		33	34	35	36	37	38	39	40	41	42
L:	sll \$t1, \$t0, 2	EX	MEM	WB							
	add \$t2, \$t1, \$s1	IF	ID	stall	stall	EX	MEM	WB			
	lw \$t4, 0(\$t2)		IF	stall	stall	ID	stall	stall	EX	MEM	WB
	add \$t3, \$t1, \$s2					IF	stall	stall	ID	EX	MEM
	add \$t5, \$t5, \$t3								IF	ID	stall
	add \$t4, \$t4, \$t5									IF	stall
	sw \$t4, 0(\$t2)										
	addi \$t0, \$t0, 1										
	slt \$t6, \$t0, \$s3										
	bne \$t6, \$zero, L										
#L:	#sll \$t1, \$t0, 2										

43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	
WB															
stall	EX	MEM	WB												
stall	ID	stall	stall	EX	MEM	WB									
	IF	stall	stall	ID	stall	stall	EX	MEM	WB						
				IF	stall	stall	ID	EX	MEM	WB					
							IF	ID	stall	stall	EX	MEM	WB		
								IF	stall	stall	ID	EX	MEM	WB	
													IF	ID	EX

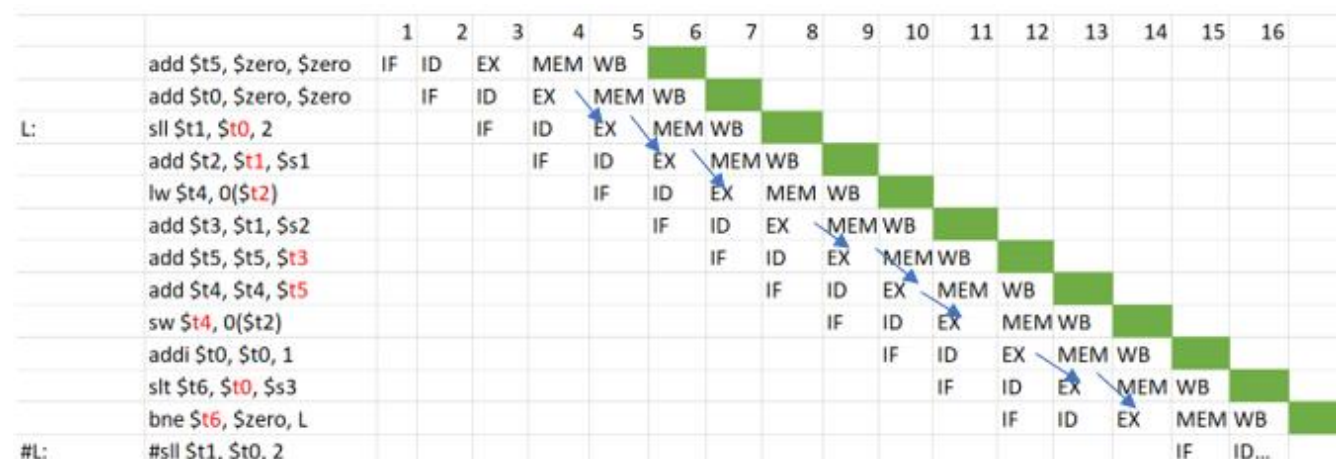
Παρατηρούμε λοιπόν, πως απαιτούνται 32 κύκλοι για την εκτέλεση της πρώτης επανάληψης και του κώδικα αρχικοποίησης. Επίσης κατά την δεύτερη επανάληψη παρατηρούμε στο δεύτερο διάγραμμα χρονισμού 25 κύκλους προκειμένου να εκτελεστεί ο κώδικας.

Ταυτόχρονα όμως έχουμε από την εκφώνηση πως ο καταχωρητής \$s3 είναι αρχικοποιημένος στη τιμή 0x100(Hex), δηλαδή 256(Decimal).

Συμπεραίνουμε λοιπόν πως απαιτούνται $32 + 25 \cdot 255$ κύκλοι ρολογιού, δηλαδή 6407cc. Οφείλουμε να αναφέρουμε πως στην τελευταία γραμμή η εντολή αρχίζει στο κύκλο 56 γιατί από την εκφώνηση της άσκησης μας δίνεται πως ο

επεξεργαστής εισάγει καθυστερήσεις (stalls) μέχρι την επίλυση η οποία πραγματοποιείται στο στάδιο EX.

(β) Έχοντας στην διάθεσή μας όλα τα σχήματα προώθησης καταλήγουμε σε ένα διάγραμμα χρονισμού της μορφής:



Παρατηρούμε ότι η “load word” δεν έχει εξάρτηση από την επόμενη εντολή, γεγονός που θα σήμαινε εισαγωγή καθυστέρησης.

Ταυτόχρονα στην αλληλουχία εντολών,

add \$t3, \$t1, \$s2

add \$t5, \$t5, \$t3

add \$t4, \$t4, \$t5

βλέπουμε πως υπάρχει κίνδυνος RAW, δηλαδή εμφανίζεται μια εντολή η οποία προσπαθεί να διαβάσει από έναν καταχωρητή πριν τον γράψει η προηγούμενη εντολή. Γνωρίζουμε πως στον MIPS ένας τέτοιος κίνδυνος είναι υπαρκτός και επιλύεται με χρήση forwarding και stalls.

Στην περίπτωση μας με την χρήση forwarding γίνεται βελτιστοποίηση της λύσης.

Παρατηρούμε λοιπόν πως για την αρχικοποίηση του κώδικα και για την συνολική εκτέλεση χρειάζονται $16 + 255 * 13 = 3331$ clock cycles (cc).

(γ) Καταλήγουμε στο ότι τα (β) και (γ) ερωτήματα ταυτίζονται διότι έχουμε ήδη πετύχει βέλτιστη λύση του προβλήματος.