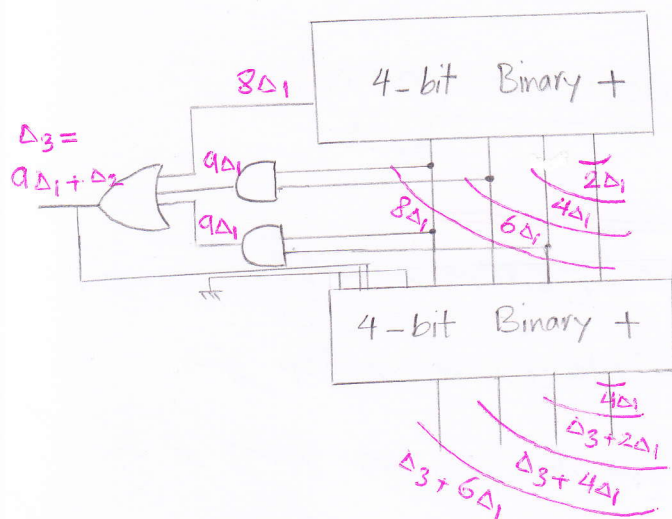


Computer Architecture - Spring 92

Midterm-Sample Solution

Problem 1.



$$\begin{cases} \Delta_1 = 7^{ns} \\ \Delta_2 = 9^{ns} \end{cases}$$

$2\Delta_1$: Full Adder (تاخیر یک NAND گیت)

$$\begin{aligned} \text{Delay}_{sum} &= \Delta_3 + 6\Delta_1 = (9\Delta_1 + \Delta_2) + 6\Delta_1 = \\ &= 15\Delta_1 + \Delta_2 = 114^{ns} \end{aligned}$$

$$\text{Delay}_{cont} = \Delta_3 = 72^{ns}$$

Problem 2.

a) $123 = 0111, 1011$
 $.75 \times 2 = 1.5 = 1 + 0.5$
 $.5 \times 2 = 1 = 1 + 0$

$\Rightarrow 123.75 = 0111, 1011.11 = 1.1110111 \times 2^6$
 (Implicit One) ←

$$S = 1$$

$$E: 2^{n-1} - 1 = 127 \Rightarrow E = 6 + 127 = 133 = 1000, 0101$$

$$M = \underbrace{1110, 1111, 00 \dots 0}_{23 \text{ bits}}$$

$52 = 0110100$
 $.125 \times 2 = 0.25 = 0 + .25$
 $.25 \times 2 = 0.5 = 0 + .5$
 $.5 \times 2 = 1 = 1 + 0$

$\Rightarrow 52.125 = 011, 0100.001 = 1.10100001 \times 2^5$

$$S = 0$$

$$E = 5 + 127 = 132 = 1000, 0100$$

$$M = \underbrace{1010, 0001, 00 \dots 0}_{23 \text{ bits}}$$

کدام سازی بایستی

b) $123.75 = 1.1110, 1111 \times 2^6$
 $52.125 = 0.1, 1010, 0001 \times 2^6$

از هم کم می کنیم:

$$\begin{array}{r} 1.1110\ 1111\ 0 \\ - 0.1101\ 0000\ 1 \\ \hline 1.0001\ 1110\ 1 \end{array}$$

$$\Rightarrow \begin{cases} S = 1 \\ E = 6 + 127 = 133 \\ M = \underbrace{0001, 1110, 1000, 0 \dots 0}_{23\ bits} \end{cases}$$

تغییرات در زنجیره بایستی که ضمیمه شده اند.

Problems 3, 4.

دستور Pop در پیاپی سازی Single-Cycle امکان پذیر نیست چون هم مقدار خوانده شده از stack را باید در رجیستر بنویسد، هم مقدار جدید \$29 را بنویسد. در واقع امکان نوشتن در دو رجیستر در یک Clock Cycle وجود ندارد.

Problem 5.

$19 = 01, 0011$
 $-13 = 11, 0011$ ← اضافه می کنیم تا هم اندازه شود

$\xrightarrow[\text{مست رات}]{\text{که چند}} \begin{array}{c} 0 \quad 0 \quad 0 \\ 1100110 \\ \hline 1 \quad 1 \quad 1 \end{array}$

$-19 = 10, 1101$

	0	1	0	0	1	1			
	0	1	0	1	0	1			
	<hr/>								
	1	1	2	2	1	1	0	1	0
	0	0	0	0	1	0	0	1	0
	0	0	0	0	1	0	0	1	0
Sign	1	1	0	1	1	0	1	0	0
Extend	1	1	0	1	1	0	1	0	0
	<hr/>								
X	1	1	1	0	0	0	0	1	0

$= -247 \checkmark$

Problem 6.

بزرگترین عدد از نظر اندازه $= 1. \overbrace{111 \dots 1}^{12\ bits} \times 2^{7-64} = 1.111 \dots 1 \times 2^{63}$

کوچکترین عدد از نظر اندازه $= 1.00 \dots 01 \times 2^{0-64} = 1.00 \dots 01 \times 2^{-64}$

Problem 7.

$$\text{CPI}_1 = 0.4 \times 1 + 0.25 \times 2 + 0.15 \times 2 + 0.2 \times 2 = 1.6$$

اگر فرض کنیم تعداد کل دستورات برنامه 100 باشد، با این دستور جدید، 15 دستور از دستورات ALU و 15 دستور از دستورات Load Word کم می شود.

$$\text{CPI}_2 = \frac{1 \times (40 - 15) + 2 \times (25 - 15) + 2 \times 15 + 3 \times 20 + 3 \times 15}{25 + 10 + 15 + 20 + 15} \approx 2.12$$

تعداد کل دستورات جدید

تغییر پیشنهاد شده مفید نیست.

Problem 8.

تغییرات در انتهاگر پاسخ

Problem 9.

$$\begin{cases} 59 = 000\ 011\ 1011 \\ 13 = \quad\quad 0\ 1101 \\ -13 = \quad\quad 1\ 0011 \end{cases}$$

	E	A	Q
init	0	00001	11011
1. shl	0	00011	10110
sub div		+ 10011	
	0	10110	
		+ 01101	
	1	00011	
2. shl	0	00111	01100
sub div		10011	
	0	11010	
		01101	
		00111	
3.	0	01110	11000
		10011	1
	1	00001	
4.	0	00011	10010
		10011	
	0	11010	
		01101	
		00011	
5.	0	00111	00100
		10011	
	0	11010	
		01101	
		00111	
		<u>00111</u>	
		R = 7	Q = 4

Problem 10.

$$\text{کوچکترین عدد قابل نمایش} = 1.0001 \times 2^{0-7} = 1.0001 \times 2^{-7}$$

$$\Rightarrow \text{دست} = 0.0001 \times 2^{-7}$$

Problem 11.*

$$2^{11} = 128 + 40 \times 2^4 + x \times 2^4 \times 2^4 \Rightarrow x = 5$$

Problem 12.

$$A = (-1)^1 \times 1.010101101000001101011000 \times 2^{\frac{26}{153-127}}$$

$$B = (-1)^0 \times 1.01100101010101101000110 \times 2^{\frac{9}{136-127}} = 0.0...01011001 \times 2^{26}$$

$$A+B = (-1)^1 \times \underbrace{1.0101011010000101111111}_{M} \times 2^{26} ; E = 26 + 127 = 153$$

Problem 13.

$$\frac{P_1}{P_2} = \frac{t_2}{t_1} = \frac{CPI_2 \times \#instr._2 \times T_2}{CPI_1 \times \#instr._1 \times T_1} = \frac{.1 \times 30 \times 2 + .15 \times 20 \times 2 + .05 \times 50 \times 2 + .7 \times 2}{.1 \times 6 + .15 \times 4 + .05 \times 20 + .7 \times 2}$$

$$= \frac{18.4}{3.6} = 5.11$$

Problem 14.

n طرز FA، در هر طرز n-1 FA

$$t_s > t_c \Rightarrow t_{d_{max}} = (\underbrace{6t_s + t_c}_{\text{طرز اول}}) + (\underbrace{6t_c + t_s}_{\text{طرز آخر}}) = 119 \text{ ns}$$

Problem 15.

	نزال ^(s)	رنگ ^(s)	ب ^(s)	ج ^(s)
ضرب	20	20/4	20	20/4
خواندن از حافظه	50	50	50/2	50/2
سایر دستورات	30	30	30	30
جمع	100	85	75	60
تسلیع بیت به حالت نزال	1	1.18	1.33	1.67

Problems 3, 4, 8.

در حل این سوالات ابتدا تغییرات نام در مسیر داده را به وجود می آوریم. پس از آن حتماً باید مقدار جدید سیگنال های کنترلی نیز تعیین شوند.

در سوال 3 می توانیم آدرس اولین byte در آخرین word stack را در instruction قرار دهیم.

پس با Sign Extend کردن این مقدار و کم کردن 29 از آن، مقدار byte های باقیمانده به دست می آید.

$$ALU_src_A = 1$$

$$ALU_src_B = 0$$

$$memToReg = 0$$

$$ALUOp = - \text{ (تفریق)}$$

$$MemWrite = 0$$

برای دستور Push (حداکثر) به تغییر در مسیر داده نیست. فقط کافی است تا عدد 4 را در بخش

immediate قرار دهیم تا با 29 جمع شده و مقدار جدید آن را بدهد.

$$ALUSrc = 1$$

$$MemWrite = 1$$

$$MemToReg = 0$$

$$ALUOp = +$$

برای دستور skip-next از Zero ALU استفاده می کنیم. با توجه به MUX رهنما

در شکل و مقدار Zero، تصمیم گیری انجام می شود.

$$ALUSrc = 0$$

$$RegWrite = 0$$

$$MemWrite = 0$$

$$PCMUX = Zero$$

در تابع سیگنال Select (هم) Zero است.

