

컴퓨터구조 중간고사 (73점 만점) - 답

2018. 10. 24

1. (1) (1점) $4 \times 10^8 + 3 \times 10^8 + 2 \times 10^8 = 9 \times 10^8$

(2) (1점) $\frac{3 \times 4 + 4 \times 3 + 6 \times 2}{4 + 3 + 2} = \frac{36}{9} = 4$

(3) (1점) $\frac{(9 \times 10^8) \times 4}{2 \times 10^9} = \frac{3.6 \times 10^9}{2 \times 10^9} = 1.8$ (초)

(4) (2점) $\text{CPU time}_{\text{new}} = \frac{(3 \times 4 + 4 \times 3 + 5 \times 2) \times 10^8}{2 \times 10^9} = \frac{34}{20} = 1.7$ (초)

$$\text{speedup} = \frac{1.8}{1.7} = \frac{18}{17} = 1.059$$

2. (1) (2점) 덧셈 4번, 뺄셈 0번

(2) (3점) 덧셈 1번, 뺄셈 2번

3. (1) (5점)

Iteration	Step	Divisor	Remainder
0	Initial values	00100	00000 11010
1	Shift Rem left		00001 10100
	Rem=Rem-Div		11101
	Rem<0 \Rightarrow +Div, R0=0		00001 10100
2	Shift Rem left		00011 01000
	Rem=Rem-Div		11111
	Rem<0 \Rightarrow +Div, R0=0		00011 01000
3	Shift Rem left		00110 10000
	Rem=Rem-Div		00010
	Rem \geq 0 \Rightarrow R0=1		00010 10001
4	Shift Rem left		00101 00010
	Rem=Rem-Div		00001
	Rem \geq 0 \Rightarrow R0=1		00001 00011
5	Shift Rem left		00010 00110
	Rem=Rem-Div		11110
	Rem<0 \Rightarrow +Div, R0=0		00010 00110

(2) (5점)

Iteration	Step	Divisor	Remainder
0	Initial values	00100	00000 11010
1	Shift Rem left		00001 10100
	Rem ≥ 0 ⇒ Rem=Rem-Div		11101
	Rem < 0 ⇒ R0=0		11101 10100
2	Shift Rem left		11011 01000
	Rem < 0 ⇒ Rem=Rem+Div		11111
	Rem < 0 ⇒ R0=0		11111 01000
3	Shift Rem left		11110 10000
	Rem < 0 ⇒ Rem=Rem+Div		00010
	Rem ≥ 0 ⇒ R0=1		00010 10001
4	Shift Rem left		00101 00010
	Rem ≥ 0 ⇒ Rem=Rem-Div		00001
	Rem ≥ 0 ⇒ R0=1		00001 00011
5	Shift Rem left		00010 00110
	Rem ≥ 0 ⇒ Rem=Rem-Div		11110
	Rem < 0 ⇒ R0=0		11110 00110
	Rem < 0 ⇒ Rem=Rem+Div		00010 00110

4. (5점) $28.25 = 11100.01_{\text{two}} = 1.110001_{\text{two}} \times 2^4$

지수 = $4+127 = 131 = 1000\ 0011$

0	1000 0011	110 0010 0000 0000 0000 0000
---	-----------	------------------------------

=> 41E2 0000

5. (각 2점)

(1) 00C7 2824

0	6	7	5	0	36
000000	00110	00111	00101	00000	100100

(2) 2041 0003

8	2	1	3
001000	00010	00001	0000 0000 0000 0011

(3) 0011 8140

0	0	17	16	5	0
000000	00000	10001	10000	00101	000000

(4) AC02 0007

43	0	2	7
101011	00000	00010	0000 0000 0000 0111

※ Alignment restriction 검사는 runtime에 하는 것이지 assemble time에는 하지 않음.

이 문제에서는 \$0를 사용하므로 register 값을 알 수 있지만 예외적인 경우임.

일반적으로는 register 값을 알 수 없으므로 4의 배수인지 확인할 수 없음.

(5) 1023 0003

4	1	3	3
000100	00001	00011	0000 0000 0000 0011

$$\times (216-204)/4 = 3$$

(6) $(0x03000000 - 0x03FA0200)/4$ 는 너무 커서 16 bits로 나타낼 수 없어서 assemble error!

(7) $PC+4(0x00003004)$ 와 $CC(0x70006000)$ 의 상위 4 bits가 다르다.

(8) 0800 1800

4
000010 0000 0000 0000 0110 0000 0000 00

$$\times 0x70006000 = 0b \ 0111 \ 0000 \ 0000 \ 0000 \ 0110 \ 0000 \ 0000 \ 0000$$

6. (각 2점)

(1) $\text{add } \$0, \$1, \$2 \Rightarrow$ 목적지 레지스터가 \$0이므로 아무 것도 변하지 않는다.

(2) $\text{lw } \$2, 3(\$3) \Rightarrow$ effective address = $3 \times 2 + 3 = 9$
4의 배수가 아니므로 error!

(3) $\text{lw } \$1, 2(\$5) \Rightarrow$ effective address = $5 \times 2 + 2 = 12$
 $M[12]=14=0x0E, M[13]=0x0F, M[14]=0x10, M[15]=0x11$
\$1이 0x0E0F1011이 된다.

(4) $\text{sw } \$12, 2(\$4) \Rightarrow$ effective address = $4 \times 2 + 2 = 10$
4의 배수가 아니므로 error!

(5) $\text{sw } \$2, 2(\$7) \Rightarrow$ effective address = $7 \times 2 + 16 = 10, \$2=0x0000 \ 0004$
 $M[16]=4, M[17]=0, M[18]=0, M[19]=0$ 이 된다.

(6) PC 가 0xA800 0000이 되고,
\$31=\$ra 가 0xA600 0054 가 된다.

7. (각 1점) (1) 0000 0034 (2) 3 (3) 5 (4) 0
(5) 0 (6) $3 \times 2 + 7 = D = 13_{\text{ten}}$ (7) 6 (8) $0A = 10_{\text{ten}}$
(9) $0D = 13_{\text{ten}}$ (10) $34 + 28_{\text{ten}} = 34 + 1C = 50 = 80_{\text{ten}}$ (11) 0

8. (각 1점) (1) 0 (2) 1 (3) 1 (4) 0 (5) 1 (6) 1 (7) 0 (8) X (9) 0