Computer Organization HW2 Answer

> 2024.4.7

Question 1(20分)

a)(6分=3+3)

- **Method 1**

Class A: 10^5 instr. Class B: 2×10^5 instr. Class C: 5×10^5 instr. Class D: 2×10^5 instr.

Time = No. instr. × CPI/clock rate

Total time P1 = $(10^5 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 3 + 2 \times 10^5 \times 3)/(2.5 \times 10^9) = 10.4 \times 10^{-4}$ s

Total time P2 = $(10^5 \times 2 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 2 + 2 \times 10^5 \times 2)/(3 \times 10^9) = 6.66 \times 10^{-4}$ s

CPI(P1) = $10.4 \times 10^{-4} \times 2.5 \times 10^9/10^6 = 2.6$ **(3分)**

CPI(P2) = $6.66 \times 10^{-4} \times 3 \times 10^9 /10^6 = 2.0$ **(3分)**

- **Method 2**

$$\text{Global CPI}_1 = \frac{1}{\text{total IC}} \sum_{k\in\{A,B,C,D\}} \text{IC}_k \times \text{CPI}_{1k} = 1 \times 0.1 + 2 \times 0.2 + 3 \times 0.5 + 3 \times 0.2 = 2.6$$

$$\text{Global CPI}_2 = \frac{1}{\text{total IC}} \sum_{k\in\{A,B,C,D\}} \text{IC}_k \times \text{CPI}_{2k} = 2 \times 0.1 + 2 \times 0.2 + 2 \times 0.5 + 2 \times 0.2 = 2$$

b)(6分=3+3)

- **Method 1**

clock cycles(P1) = $10^5 \times 1 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 3 + 2 \times 10^5 \times 3 = 2.6 \times 10^6$ **(3分)**

clock cycles(P2) = $10^5 \times 2 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 2 + 2 \times 10^5 \times 2 = 2.0 \times 10^6$ **(3分)**

- **Method 2**

$$\text{Clock Cycles}_1 = \text{Global CPI}_1 \times \text{total IC} = 2.6 \times 10^6$$
$$\text{Clock Cycles}_2 = \text{Global CPI}_2 \times \text{total IC} = 2 \times 10^6$$

c)(8分=3+3+2)

$$T_1 = \frac{\text{Clock Cycles}_1}{f_1} = \frac{2.6 \times 10^6}{2.5 \times 10^9} s = 1.04 \times 10^{-3} s$$
$$T_2 = \frac{\text{Clock Cycles}_2}{f_2} = \frac{2 \times 10^6}{3 \times 10^9} s = 6.67 \times 10^{-4} s$$

Since $T_1 > T_2$, Cpu time is shorter, P2 is better.（2分）

> 每个计算的答案均为3分，c问中最后解释答案2分

Question 2(14分)

a)(7分=3+2+2)

```
x5 := 0x80000000
x6 := 0xD0000000
x5 + x6 = 0x150000000
x30 = 0x50000000 //(3分)
>>> overflow //(2分)
```

The sum of two negative number turns out to be positive. **(2分)**(言之有理即可)

b)(7分=3+2+2)

```
x5 := 0x80000000
x6 := 0xD0000000
-x6 = 0x2FFFFFFFF + 1 = 0x30000000
x5 - x6 = 0xB0000000
x30 = 0xB0000000 //或者是对符号位进行了判断(3分)
>>> no overflow/ correct/ desired //(2分)
```

Substraction between two numbers with same sign will not cause overflow. **(2分)**(言之有理即可)

> 注意：risc-v的add，sub指令是对有符号数做处理，因此这里的操作数是有符号数。如果只用无符号数来判断不得分，如果分两种情况讨论，每问扣两分。b问subtract也可以不需要写计算过程，只需要写出对符号位的判断就够。

Question 3(10分)

a)(5分)

- **Method 1**

```
  00010111 (23)
+ 01110000 (112)
-----------
  10000111 (-121)
 saturate >>> 127
```

- **Method 2**

$$23 + 112 = 135 > 127.$$

$$\text{Hence } 23 + 112 = 127$$

b)(5分)

- **Method 1**

```
112 = 01110000
-112 = 10001111 + 1 = 10010000
  00010111 (23)
+ 10010000 (-112)
------------
  10100111 (-89)
```

- **Method 2**

$$23 - 112 = -89 > -128$$

$$\text{Hence } 23 - 112 = -89$$

> 要写出与127/-128等边界相关，或者是有直接分步计算过程。过程3分，答案2分

Question 4(20分)

Step	Multiplicand	Product
Initial	0110_0010	0000_0000_0001_0100
1	0110_0010	0000_0000_0000_1010
2	0110_0010	0000_0000_0000_0101
3	0110_0010	0011_0001_0000_0010
4	0110_0010	0001_1000_1000_0001
5	0110_0010	0011_1101_0100_0000
6	0110_0010	0001_1110_1010_0000
7	0110_0010	0000_1111_0101_0000
8	0110_0010	0000_0111_1010_1000

0x62 x 0x14 = 0x7A8 = $1960_{(10)}$

> Multiplicand(8分)+ Product(8分)+最后答案(4分)。初始step错一个扣2分，中间step错一个扣1分

Question 5(25分)

Step	Divisor	Remainder	Quotient
Initial	0101_0100_0000	0000_0011_1110	00_0000
1	0010_1010_0000	0000_0011_1110	00_0000
2	0001_0101_0000	0000_0011_1110	00_0000
3	0000_1010_1000	0000_0011_1110	00_0000
4	0000_0101_0100	0000_0011_1110	00_0000
5	0000_0010_1010	0000_0011_1110	00_0000
6	0000_0001_0101	0000_0001_0100	00_0001
7	0000_0000_1010	0000_0001_0100	00_0010

62 = 21 x 2 + 20

> Divisor(7分)+ Remainder(7分)+Quotient(7分)+最后答案商和余数各两分(4分=2+2)。初始step错一个扣2分，中间step错一个扣1分

Question 6(16分)

a)(8分)

0x0C000000 = $0000\_1100\_0000\_0000\_0000\_0000\_0000\_0000_{(2)}$ **(2分)**

sign bit: 0 **(1分)**

exponential: $0001\_1000_{(2)}$ = 24

fraction: $0000\_0000\_0000\_0000\ 0000\_000_{(2)}$ **(1分)**

exponential - bias = 24 - 127 = -103 **(1分)**

num: 2^{-103} **(3分)**

b)(8分)

63.25 = 111111.01 = $1.1111101 * 2^5$ **(2分)**

exponential = 5 + 127 = 132 = $1000\_0100_{(2)}$ **(1分)**

sign bit: 0 **(1分)**

fraction: $111\_1101\_0000\_0000\_0000\_0000_{(2)}$ **(1分)**

num: $0100\_0010\_0111\_1101\_0000\_0000\_0000\_0000_{(2)}$ = 0x427D_0000 **(3分)**