Computer Organization Homework 1

陈贲(12212231)

Problem 1. Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (classes A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2. Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D.

(1) What is the global CPI for each implementation?
(2) FInd the clock cycles required in both cases.
(3) Which implementation is faster and why?

Solution. The global CPI for P1 and P2 can be calculated as follows:

$$CPI_{P1} = 1 \times 10\% + 2 \times 20\% + 3 \times 50\% + 3 \times 20\% = 2.6$$
$$CPI_{P2} = 2 \times 10\% + 2 \times 20\% + 2 \times 50\% + 2 \times 20\% = 2$$

and the clock cycles required in both cases are:

$$CC_{P1} = 1.0 \times 10^6 \times 2.6 = 2.6 \times 10^6$$
$$CC_{P2} = 1.0E6 \times 2 = 2.0 \times 10^6$$

and the time required in both cases are:

$$\text{CPU Time}_{P1} = \frac{2.6 \times 10^6 CPI}{2.5 \times 10^9 Hz} = 1.04ms$$
$$\text{CPU Time}_{P2} = \frac{2.0 \times 10^6 CPI}{3.0 \times 10^9 Hz} = 0.67ms$$

Therefore, P2 is faster for its higher clock rate and less CPI, which results in less CPU time.

Problem 2. Assume that registers x5 and x6 hold the values 0x80000000 and 0xD0000000, respectively.

(1) What is the value of x30 for the following assembly code?

```
add x30, x5, x6
```

Is the result in x30 the desired result, or has there been overflow? Explain it.

(1) What is the value of x30 for the following assembly code?

```
sub x30, x5, x6
```

Is the result in x30 the desired result, or has there been overflow? Explain it.

Solution. The result of the addition is

$$\text{x30} = 80000000 + D0000000 = 150000000 = 50000000$$

which is not the desired result since there's one bit overflow for a signed 32-bit value. It is abnormal that two negative number makes a positive number. And the result of the subtraction is

$$\text{x30} = 80000000 - D0000000 = 80000000 + 30000000 = B0000000$$

which is the desired result and there's no overflow since the opreand is the smallest negative number and minus a negative number, which is the same as adding a 32-bit positive number.

Problem 3.

(1) Assume 23 and 112 are signed 8-bit decimal integers stored in two's complement format. Calculate 23 + 112 using saturating arithmetic. The result should be written in decimal. Show the steps for calculation

(2) Assume 23 and 112 are signed 8-bit decimal integers stored in two's complement format. Calculate 23 - 112 using saturating arithmetic. The result should be written in decimal. Show the steps for calculation

Solution. The result of saturating addition is

$$23 + 112 = 127(135)$$

since the range of 8-bit signed integer is -128 to 127, the result is saturated to 127. And the result of saturating subtraction is

$$23 - 112 = -89$$

there's no overflow since the result is in the range of 8-bit signed integer.

Problem 4. Calculate the product of the hexadecimal unsigned 8-bit integers 62 and 14 using the hardware described below. You should show the binary contents of each register on each step.

Solution. The result of $62 \times 14 = 1101100100$. Steps:

Step	Action	Multiplier	Multiplicand	Product
0	Initial Values	1110	0000 0011 1110	0000 0000 0000
1	No Action	1110	0000 0011 1110	0000 0000 0000
	Lshift Mcand	1110	0000 0111 1100	0000 0000 0000
	Rshift Mplier	0111	0000 0111 1100	0000 0000 0000
2	Prod+=Mcant	0111	0000 0111 1100	0000 0111 1100
	Lshift Mcand	0110	0000 1111 1000	0000 0111 1100
	Rshift Mplier	0011	0000 1111 1000	0000 0111 1100
3	Prod+=Mcant	0011	0000 1111 1000	0001 0111 0100
	Lshift Mcand	0011	0001 1111 0000	0001 0111 0100
	Rshift Mplier	0001	0001 1111 0000	0001 0111 0100
4	Prod+=Mcant	0001	0001 1111 0000	0011 0110 0100
	Lshift Mcand	0001	0011 1110 0000	0011 0110 0100
	Rshift Mplier	0000	0011 1110 0000	0011 0110 0100

Step	Action	Multiplier	Multiplicand	Product
5	Final Result	0000	0011 1110 0000	0011 0110 0100

Problem 5. Calculate unsigned 6-bit integer 62 divided by 21 using the hardware described below. You should show the binary contents of each register on each step.

Solution. The result of $62 \div 21 = 2$ and remainder 20. Steps:

Step	Action	Quotient	Divisor	Remainder
0	Initial Values	0000	1010 1000	0011 1110
1	Rem-=Div	0000	1010 1000	1001 0110
	LShift 0 to Quot	0000	1010 1000	0011 1110
	Rshift Div	0000	0101 0100	0011 1110
2	Rem-=Div	0000	0101 0100	1110 0000
	LShift 0 to Quot	0000	0101 0100	0011 1110
	Rshift Div	0000	0010 1010	0011 1110
3	Rem-=Div	0000	0010 1010	0001 0100
	LShift 1 to Quot	0001	0010 1010	0001 0100
	Rshift Div	0001	0001 0101	0001 0100
4	Rem-=Div	0001	0001 0101	0001 0100
	LShift 0 to Quot	0010	0001 0101	0001 0100
	Rshift Div	0010	0000 1010	0001 0100
5	Final Result	0010	0000 1010	0001 0100

Problem 6.

(1) What decimal number does the bit pattern 0x0C000000 represent if it is an IEEE754 single precision floating point number? Show the steps for calculation.

(2) Write down the binary representation of the decimal number 63.25 assuming the IEEE754 single precision format. Show the steps for calculation.

Solution. The bit pattern is shown below

Sign	Exponent	Fraction
0	00011000	00000000000000000000000

So the value is

$$(-1)^0 \times 2^{(24-127)} \times (1+0.0)_{Bin} = (2^{-103})_{Dec}$$

The binary representation of 63.25 is that

$$(63.25)_{Dec} = (111111.01)_{Bin} = (1+0.1111101)_{Bin} \times 2^{(132-127)}$$

so the IEEE754 single precision format is

Sign	**Exponent**	**Fraction**
0	10000100	11111010000000000000000