Computer Architecture

Fall, 2020

Week 8

2020.11.2

Group:

組員簽名：_____

[group10] (對抗賽)

1. According to the floating-point addition, there's something wrong in the following hardware design. Please corrects it. Moreover, please fills the blank of (1), (2), …, (6) using Table 1.

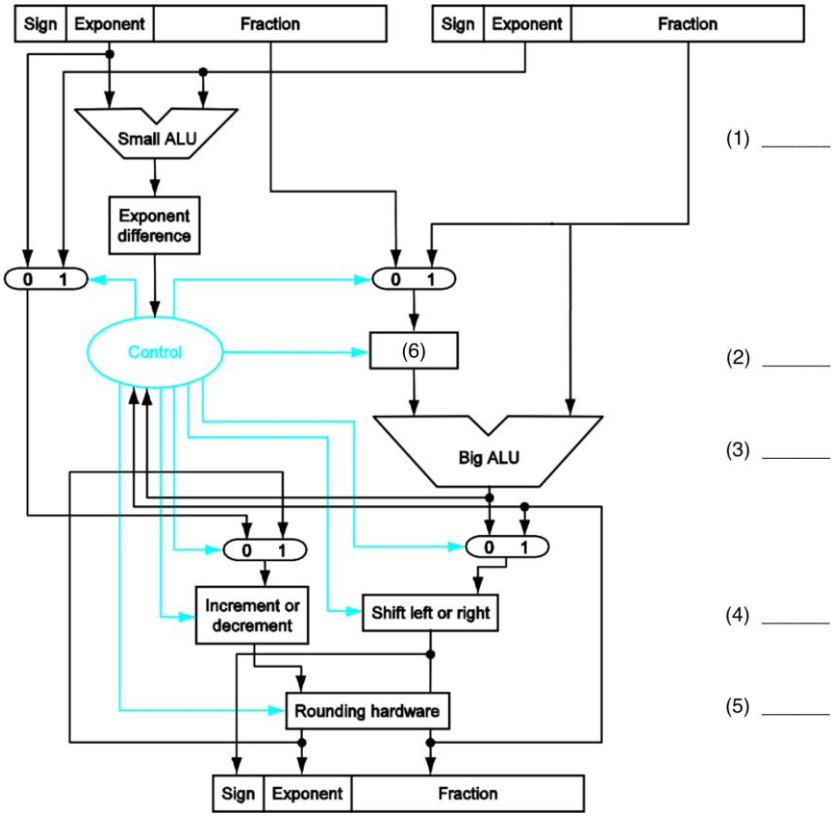
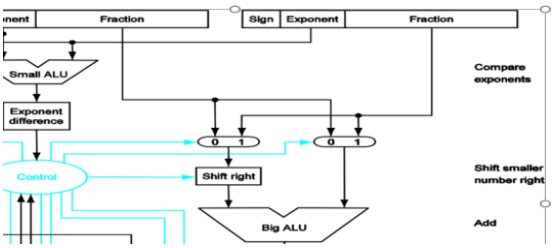


Table 1.

A. Normalize	B. Shift smaller number right	C. Add
D. Round	E. Shift right	F. Compare exponents

Ans : (1) F (2) B (3) C (4) A (5) D (6) E



[group9] (對抗賽)

2. Please find whether the followings are true or false, if false, correct it.
   (A) Compare the Divide Hardware v1 and v2, we can save our time S to S/2
   (B) Since both HI/LO are registers, we can access the value inside it directly.
   (C) When we do signed multiply, what we do on sign bits of multiplier is different with other bits of multiplier.
   (D) HI/LO will always store most-significant 32 and least-significant 32 bits.
   (E) In IEEE 754, if a digit's exponent is 0110 1000 (104), then it means its real exponent is 104.

Ans:

(A) False, by using Divide Hardware v2, we can save our "space" S to S/2
(B) False, we have to access HI/LO by mfhi, mflo
(C) True.
(D) False, division use HI/LO as quotient and remainder.
(E) False, real exponent is 104 – 127 = -23

[group3] (對抗賽)

3. Please do the unsigned multiplication of 1010 and 0011 by filling the form. (Hint: use the multiply algorithm version 2)

Multiplicand	Product

Ans:

Multiplicand	Product
1010	0000 0011
	1010 0011
1010	0101 0001
	1111 0001
1010	0111 1000
1010	0011 1100
1010	0001 1110

[group14]

4. Using IEEE 754 single precision format to express following numbers.
   (A) +0
   (B) -0
   (C) +∞
   (D) -∞
   (E) Largest positive normalized number
   (F) Smallest positive normalized number
   (G) Largest negative normalized number
   (H) Smallest negative normalized number
   (I) NaN (Not a number)

Ans:

(A) 0 00000000 0~0
(B) 1 00000000 0~0
(C) 0 11111111 0~0
(D) 1 11111111 0~0
(E) 0 11111110 1~1
(F) 0 00000001 0~0
(G) 1 00000001 0~0
(H) 1 11111110 1~1
(I) {0,1} 11111111 Nonzero

[group5] (對抗賽)

5. Which statement is incorrect about the floating point in IEEE 754 Standard?
   (A) Double precision has greater accuracy than single precision due to larger significand.
   (B) The range of significand is: 0<=significand<=1.
   (C) The number - 0.75 in decimal converting to floating point representation for single precision is: 1 0111 1110 100 0000 0000 0000 0000 0000.
   (D) For single precision,when the exponent is 255_{10} and significand is nonzero,the number is "+/- infinity".
   (E) In floating-point addition,we will left shift the smaller number to add two floating number.

Ans:

(B) 0<significand<1.
(D) the exponent is 255_{10} and significand is "zero"
(E) we will "right" shift the smaller number

[group4] (對抗賽)

6. In Python, is the the conditional 0.1 + 0.2 == 0.3 true or false? Please explain your answer. (Hint. IEEE

754 rounding error)

ANS: We use IEEE 754 to represent float point. If the floating point cannot be represented as sum of power of 2, there will be an error between the original floating point and the number which represent in IEEE 754.
In base 2, 1/10 is the infinitely repeating fraction
0.0001100110011001100110011001100110011001100110011...
So, if you transfer the number into IEEE 754 and then convert it to decimal, the number will be
0.1000000000000000055511151231257827021181583404541015625.
For the same reason, 0.2 will generate error when transfer to IEEE 754. Thus, if 0.1 + 0.2, the answer will be 0.30000000000000004, which is not 0.3.

```
In [2]:  1 0.1+0.2==0.3
Out[2]: False
```

[group12] (對抗賽)

7. Consider the IEEE 754 single precision floating-point (FP) format concept to solve (a ~ d) and then consider the given point operand format as follows to solve (e ~ h). Which of the following statements are correct?

Sign(s)	Exponent (e)	Fraction (f)
1 bit	8 bits	23 bits

(A) The FP number representation is $(-1)^{Sign}$ x (significand) x $2^{exponent - bias}$
(B) The smallest positive normalized number is: 1.0 x 2^{-125}
(C) The smallest positive denormalized number is: 1.0 x 2^{-149}.
(D) Converting an integer variable to a single precision FP number will lose range.
(E) The decimal number -13.5 is represented as 1 10000011 11011000000000000000000
(F) The largest positive number normalized number is + $(1 - 2^{-23}) * 2^{+127}$.
(G) The smallest positive denormalized number is + 2^{-149}.
(H) Exponents with all 1's are reserved for ± ∞ and NaN.

ANS: (a), (c), (d), (g), (h)
(a): $(-1)^{sign}$ X (1.fraction) $2^{exponent - bias}$ or $(-1)^{sign}$ X (significand) x $2^{exponent - bias}$
(b): It should be 1.0 x 2^{-126}
(e): $-13.5_{10} = -1101.1_2 = -1.1011_2 * 2^3 \rightarrow$ 1 10000010 10110000000000000000000
(f): The largest positive number = $(1 + 1 - 2^{-23}) \times 2^{127} = (1 - 2^{-24}) \times 2^{128}$

[group2] (對抗賽)

8. (Hint. floating point addition) 將 0.5 與-0.43751。以二進位相加,並假設有 4 個精確位數。

ANS:

首先將這兩個數字轉換為以正規化表示的二進位形式：

$0.5_{10} = 1/2_{10} = 0.1_2 = 1.000_2 \times 2^{-1}$

$-0.437510 = -7/2^4{}_{10} = -0.0111_2 = -1.110_2 \times 2^{-2}$

步驟一：將指數較小的數$(-1.110_2 \times 2^{-2})$的有效數字右移直到它的指數部份與較大的數一致為止:

$-1.110_2 \times 2^{-2} = -0.111_2 \times ^2-1$

步驟二：將有效數字相加：$1.000_2 \times 2^{-1} + (-0.111_2 \times 2^{-1})$

$= 0.001_2 \times 2^{-1}$

步驟三：將總和正規化並檢查是否有發生溢位或下溢的情況：

$0.0012 \times 2^{-1} = 1.000_2 \times 2^{-4}$(因為 127>-4>-126,所以沒有溢位或下溢的情形)。

步驟四：將總和四捨五入：$1.000_2 \times 2^{-4}$

該結果已經是四個位數,所以無需四捨五入。

最後的結果為 $1.000_2 \times 2^{-4} = 0.0001_2 = 1/2^4 = 1/16_{10} = 0.0625_{10}$

[group8]

9. Multiply two four-bit 2's complement numbers a= -3 and b= -7

(a). please fill the blanks

				1	1	0	1
				1	0	0	1

(b.) Do the same problem by using Booth's algorithm again

				1	1	0	1

ANS:

1	1	1	1	1	1	0	1
0	0	0	0	0	0	0	
0	0	0	0	0	0		
1	1	1	0	1			
0	0	0	1	0	1	0	1

b.

				1	1	0	1
				-1	0	1	-1

0	0	0	0	0	0	1	1
1	1	1	1	1	0	1	
0	0	0	1	1			
0	0	0	1	0	1	0	1