Fall, 2023
Week 8
2023.10.30

組別：_____ 簽名：_____

[group4]

1. Choose the **wrong** statement:

(a) Division in MIPS will check divide by 0 case automatically, so when we write a program we don't need to double check

(b) Divisor register and remainder register has 32 bits size, while quotient register has 64 bits size

(c) In the multiplier register, when we do multiply, we need 64-bit register shift the bit left

(d) In IEEE 754 Standard, a floating pointer 01001011101010101011110010111101 is 2.224*10^9

Ans: a, b, c

(a) Division won't handle the divide by 0 case.
(b) Reminder register has 64 bit.
(c) Shift right.

[group7]

2. Assume we are designing a 16-bit MIPS CPU with 16-bit instruction words.

a. Assume the IEEE 754 floating point representation is also adjusted to 16-bit long for this 16-bit MIPS CPU. If the floating point numbers are required to represent the values within +-10^18, what are the numbers of bits of the field sign, exponent and mantissa?

b. In the IEEE-754 floating point representation, the exponent field employs the excess-N coding. What should be N in this 16-bit floating point format?

c. What is the smallest positive normalized floating number in this 16 bit floating point format?

d. Give a case which is not a number in this 16-bit floating point format.

e. True or false, if we want to increase the precision of the 16-bit floating point format without increase the total number of bits, the representation range won't change.

Ans:

a. 1 for sign, 7 for exponent (because 10^18 = 2^60 and use 6 bit to represent 60) and 8 for mantissa (because 16-1-7=8)
b. 63 because 2^7-1=63
c. 0 0000001 00000000
d. 0 1111111 none0 8 bits
e. F, The tradeoff between precision and range. Larger range -> less precision or more precision -> smaller range

[group14]

3.

(1) Change A, B to decimal.

A:

0	1000 0010	0101 0000 0000 0000 0000 000

B:

0	1000 0001	0001 0100 0000 0000 0000 000

(2) Add A, B and change it to single precision representation.

Ans:

(1) **A: 10.5**

Exponent:

$1000\ 0010_2 = 130_{10}$

Bias adjustment: 130 – 127 = 3

Significand:

$1 + 2^{-2} + 2^{-4}$ = 1.3125

Represents: $1.3125_{10} \times 2^3$ = 10.5

B: 4.3125

Exponent:

$1000\ 0001_2 = 129_{10}$

Bias adjustment: 129 – 127 = 2

Significand:

$1 + 2^{-4} + 2^{-6}$ = 1.078125

Represents: $1.078125_{10} \times 2^2$ = 4.3125

(2) $1.0101_2 \times 2^3 + 1.000101_2 \times 2^2$

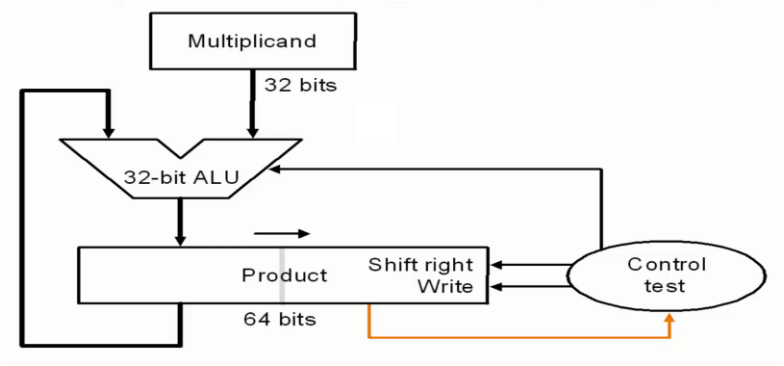
➔ $1.0101_2 \times 2^3 + 1.1000101_2 \times 2^3$

➔ 1.1101101×2^3

0	1000 0010	1101 1010 0000 0000 0000 000

[group5]

4. Given the multiplier below (unsigned), we perform 0110x0101. Please write down the value in the product for every step.



Ans:

Multiplicand	Product
0110	0000 0101
	0110 0101 (add multiplicand to hi)
	0011 0010 (right shift)
	0001 1001 (right shift)
	0111 1001 (add multiplicand to hi)
	0011 1100 (right shift)
	0001 1110 (right shift)

[group9]

5. True or false

A. For division algorithm 2, half of the bits in divisor register is always 0.

B. In division algorithm 1, the quotient register is eliminated by combining with the remainder register as shift right.

C. For division algorithm 2, shifting the left half of the remainder comes after checking the repetition.

D. Multiplication and division both use a combination of lo and hi registers.

Ans:

A. False, Algorithm 1.
B. False, Algorithm 2, shift left.
C. True.
D. True.

[group1]

6. True or false

   (1) The addition and subtraction of floating-point numbers have the associative property.

   (2) The hardware design for multiplication and division is the same.

   (3) To calculate a 4-bit divided by 4-bit equation, the remainder needs to be shifted 5 times.

   (4) Calculating a floating-point problem: 01111110000000000111010000011110 + 00000001000001111001001000010110, the answer's exponents will be 11111110.

Ans:

(1) False, in some extreme calculations, the associativity of floating-point addition and subtraction may fail.

(2) True.

(3) False, to calculate a 4-bit divided by 4-bit equation, the remainder needs to be shifted left 5 times and then shifted right 1 time for a total of 6 shifts.

(4) False, it will be 11111100.

[group3]

7. Convert the 3 “single precision” floating point into scientific notation. Write mantissa in decimal, and round to the 6th decimal place.

Sign	Exponent	Significand (mantissa)
1	1111 1111	000 1111 1111 0000 0000 0001
0	0000 0000	110 0000 0000 0000 0000 0000
1	0110 1000	101 0101 0100 0011 0100 0010

Ans:

1. Exponent=255 and mantissa≠0 ⇒ NaN
2. Exponent=0 and mantissa≠0 ⇒ denormalized

   Sign=0 ⇒ positive

   Mantissa: $2^{-1} + 2^{-2} = 0.75$

   Ans: $+0.75 \times 2^{-126}$
3. Sign=1 ⇒ negative

   Exponent: $2^3 + 2^5 + 2^6 = 104,\ 104 - 127 = -23$

   Mantissa: $1 + 2^{-1} + 2^{-3} + 2^{-5} + 2^{-7} + 2^{-9} + 2^{-14} + 2^{-15} + 2^{-17} + 2^{-22} \approx 1.666115$

   Ans: -1.666115×2^{-23}

[group6]

8. Choose the correct answer.
   A. Taking an 8-bit Wallace Tree Multiplier combing with a carry look-ahead adder, we will have at least 4 full adder delay
   B. The 64 bits register in division uses shift right mechanism, while multiplication uses shift left
   C. 1.0×10^{-11}, 2.0×10^{-7}, 35.0×10^{-3} are all normalized
   D. As we know that IEEE 754 uses bias of 127 for single precision, we can assume that
      0 0111 1110 000 0000 0000 0000 0000 0000 is bigger than
      0 1000 0000 000 0000 0000 0000 0000 0000
      If not, please explain the reason and calculate.

Ans:

   A. T
   B. F, division->left, multiplication->right.
   C. F, 35.0×10^{-3} is not.
   D. F, smaller 0111 1110 – 127(binary) and 1000 0000 – 127(binary)