Homework 2

Requirement:

Please type your answers in this .docx file. Do **not** submit handwriting documents.

1. (4 points)

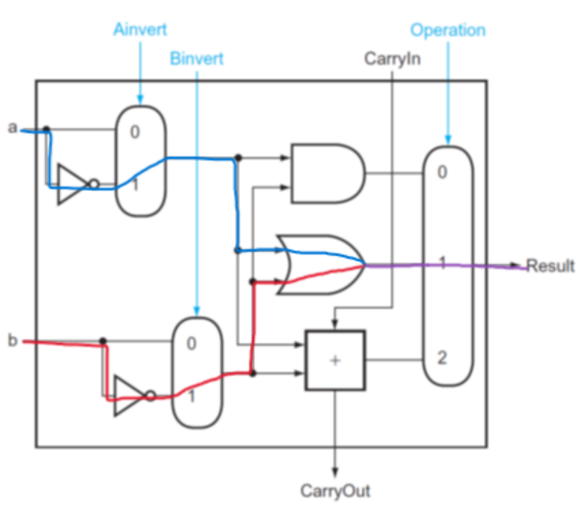
   (1). How to configure "Ainvert", "Binvert", and "Operation" to make the following 1bit ALU perform $\overline{(a+b)}$?

   Ainvert: 1

   Binvert: 1

   Operation: 00

   (2). Draw the signal route on the following figure.



2. (6 points) There are two decimal numbers A = 54, B = -77:

   (1). Compute A + B in 8-bit 2's complement. Is there any overflow? Why or why not?

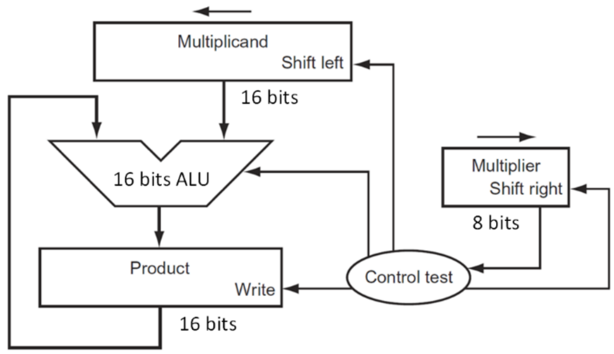
   23_{10} = 00010111_2 There is no overflow. Neither two most-significant bits are 1

   (2). Compute A – B in 8-bit 2's complement. Is there any overflow? Why or why not?

   125_{10} = 01111101_2 There is overflow. The second-most-significant bit is 1

3. (15 points) The following hardware is used to calculate the product of two unsigned 8-bit integers 0x62 and 0x12. For each step of each iteration, please fill out the following

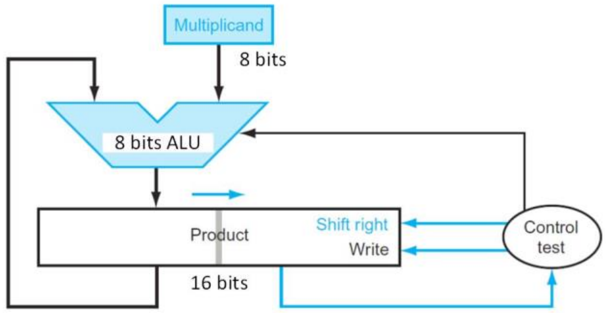
table with the values in the Multiplier register, Multiplicand register, and Product register. For "1a" step of each iteration, please specify it should be "0 -> No operation" or "1 -> Prod = Prod + Mcand" (please note: "0" and "1" are the value of the LSB of the multiplier; You can also reference the table in slide "3-3 Arithmetic for Computers" to answer this question).



Iteration	Step	Multiplier	Multiplicand	Product
0	Initial values	00111110	00001100	0000000000000000
1	1a: 0	-	-	-
	2: Shift left Multiplicand	-	00011000	-
	3: Shift right Multiplier	00011111	-	-
2	1a: 1	-	-	0000000000011000
	2: Shift left Multiplicand	-	00110000	-
	3: Shift right Multiplier	00001111	-	-
3	1a: 1	-	-	0000000001001000
	2: Shift left Multiplicand	-	01100000	-
	3: Shift right Multiplier	00000111	-	-
4	1a: 1	-	-	0000000010101000
	2: Shift left Multiplicand	-	11000000	-
	3: Shift right Multiplier	00000011	-	-
5	1a: 1	-	-	0000000101101000
	2: Shift left Multiplicand	-	10000000	-
	3: Shift right Multiplier	00000001	-	-
6	1a: 1	-	-	0000000111101000
	2: Shift left Multiplicand	-	00000000	-
	3: Shift right Multiplier	00000000	-	-
7	1a: 0	-	-	0000000111101000
	2: Shift left Multiplicand	-	00000000	-
	3: Shift right Multiplier	00000000	-	-
8	1a: 0	-	-	0000000111101000
	2: Shift left Multiplicand	-	00000000	-
	3: Shift right Multiplier	00000000	-	-

4. (15 points) The following hardware is used to calculate the product of two unsigned 8-bit integers 0x62 and 0x12. For each step of each iteration, please fill out the following table with the values in the Multiplier register and Product register. For the first step of

each iteration, please specify the product 0 value should be "0 -> No operation" or "1 -> A = A + Mcand" (please note: "A" is the left half of the product register.



Iteration	Step	Multiplicand	Product
0	Initial values	00001100	0000000000000000
1	1. Product0 =		
	2: Shift right Product		
2	1. Product0 =		
	2: Shift right Product		
3	1. Product0 =		
	2: Shift right Product		
4	1. Product0 =		
	2: Shift right Product		
5	1. Product0 =		
	2: Shift right Product		
6	1. Product0 =		
	2: Shift right Product		
7	1. Product0 =		
	2: Shift right Product		
8	1. Product0 =		
	2: Shift right Product		

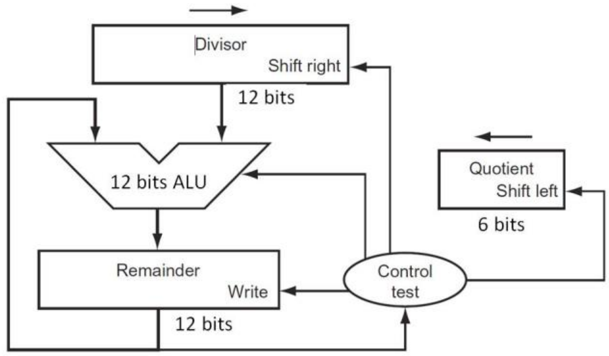
5. (8 points) Write down the binary representation of the decimal number 63.25 using the IEEE 754 single precision format. Please specify the calculation steps.
   0 10000100 11111010000000000000000

6. (8 points) What decimal number does 0X0C000000 represent if it is a floating-point number using the IEEE 754 standard? Please specify the calculation steps.
   9.86076131526e-32

7. (14 points) Write down the binary representation of 2.6125×10^{1} and $4.150390625 \times 10^{-1}$ in the IEEE 754 single precision format, then calculate the sum by hand. Round the result to the nearest, ties to even. Show all the steps.
   2.6125+0.04150390625=2.65400390625

8. (15 points) The following hardware is used to calculate the division of two unsigned 6-bit integers 0b111100 divided by 0b010001. For each step of each iteration, please fill out the following table with the values in the Divisor register, Quotient register, and

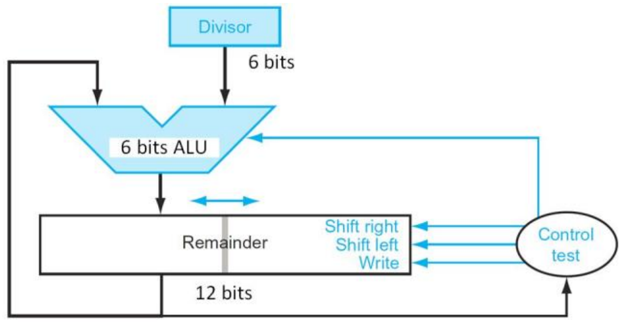
Remainder register. For the second step of each iteration, please specify it should be "Rem < 0 -> +Div, LSL Q, Q0 = 0" or "Rem >= 0 -> LSL Q, Q0 = 1", You can also reference the table in slide "3-5 Arithmetic for Computers" to answer this question).



Iteration	Step	Quotient	Divisor	Remainder
0	Initial values			
1	1: Rem = Rem - Div			
	2:			
	3: Shift right Div			
2	1: Rem = Rem - Div			
	2:			
	3: Shift right Div			
3	1: Rem = Rem - Div			
	2:			
	3: Shift right Div			
4	1: Rem = Rem - Div			
	2:			
	3: Shift right Div			
5	1: Rem = Rem - Div			
	2:			
	3: Shift right Div			
6	1: Rem = Rem - Div			
	2:			
	3: Shift right Div			
7	1: Rem = Rem - Div			
	2:			
	3: Shift right Div			

9. (15 points) The following hardware is used to calculate the division of two unsigned 6-bit integers 0b111100 divided by 0b010001. For each step of each iteration, please fill

out the following table with the values in the Divisor register and Remainder register. For the second step of each iteration, please specify it should be "Rem < 0 -> +Div, LSL Rem" or "Rem > =0, LSL Rem, Rem0 = 1".



Iteration	Step	Divisor	Remainder
0	Initial values		
1	1: Rem = Rem - Div		
	2:		
2	1: Rem = Rem - Div		
	2:		
3	1: Rem = Rem - Div		
	2:		
4	1: Rem = Rem - Div		
	2:		
5	1: Rem = Rem - Div		
	2:		
6	1: Rem = Rem - Div		
	2:		
	4: LSR LeftRem		

My work on the next page:

ECE 562 3/31

5. 63.25

0 | 10000100 | 111 11 01 00000000000000000

sign 1 exponent 8 fraction 23

63.25 = (1') ×

6.325 × 10'

$63.25_{10} \rightarrow 111111.11001$ 1.01 = 1.25

32 16 8 4 2 1 16 8 4 2 1 $\frac{1}{1}$ $\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{8}$ $\frac{1}{16}$

fraction: 111111.01 remove whole "1"

normalize: 111111.01 = 1.1111101×2^5

biased exponent: 5 + 127 = 132 bias

101 111111 10000100 $2^{(8-1)} - 1 = 127$

0 | 10000100 | 1111101 → 0

0 132 8 23

6. 0x0c000000 → 0, 00011000_8 00000000000000000000000 23

24 → 24-127 = -103 1.0

1.0×2^{-103}

7. 26.125_{10}, 0.4150390625_{10} add: 2.6540039062 5 normalize: ✓

2.6125×10^1 $4.150390625 \times 10^{-1} \rightarrow 0.04150390625 \times 10^1$

11010.001 0.0110 1 0 0 1

26 0.5 0.25 0.125 0.5 0.25 0.125 0.0625 0.03125 0.015625 0.078125

0.125 0.00390625 0.001953125 0.0009765625

fraction: 11010.001 mantissa: 0.0110 101001 exponent: $2^0 \rightarrow 0 + 127 = 127$

normalize: 1.1010001

exponent: $2^4 \rightarrow 4 + 127 = 131$

26.125 → 0 10000011 1010001 → 0

0.415... → 0 01111111 0110101001 → 0