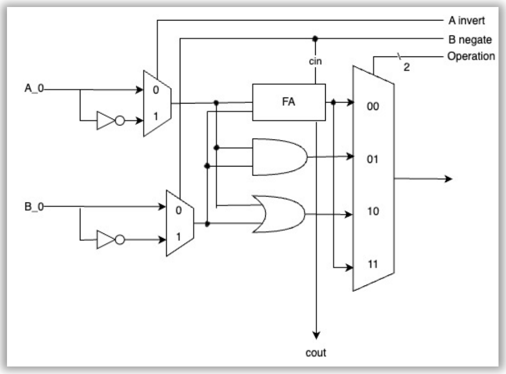
Computer Architecture HW3

111062117 黃祥陞

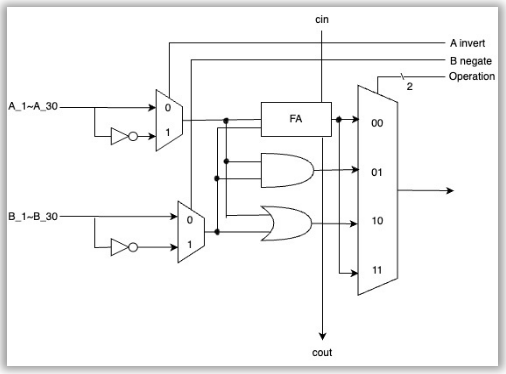
1.

A_invert	B_negate	Operation	Function
0	1	01	AND
0	1	10	OR
0	1	00	add
0	0	00	sub
1	0	01	NOR
0	1	11	add-ext
0	0	11	sub-ext

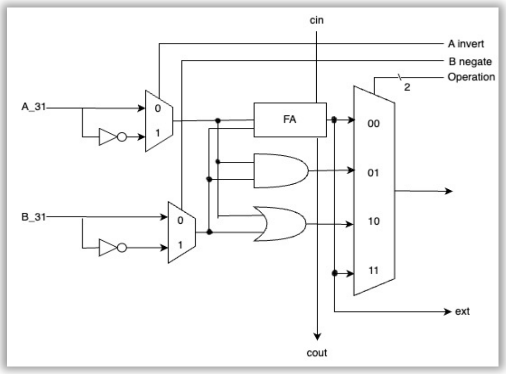
ALU0



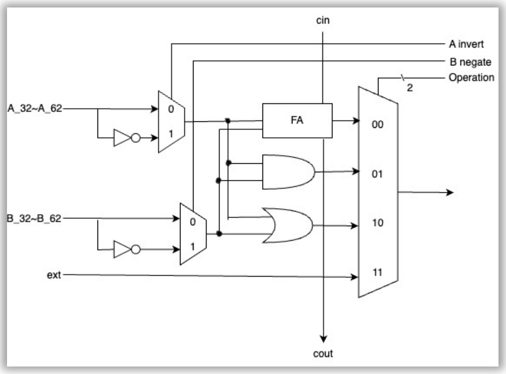
ALU1~ALU30



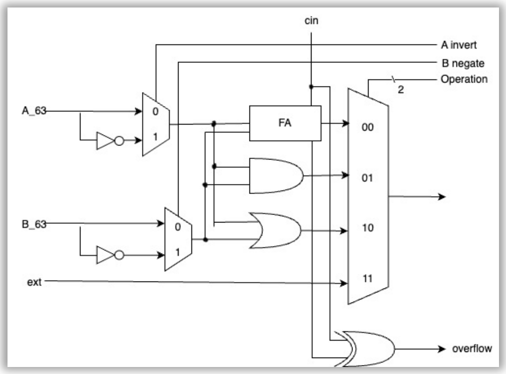
ALU31



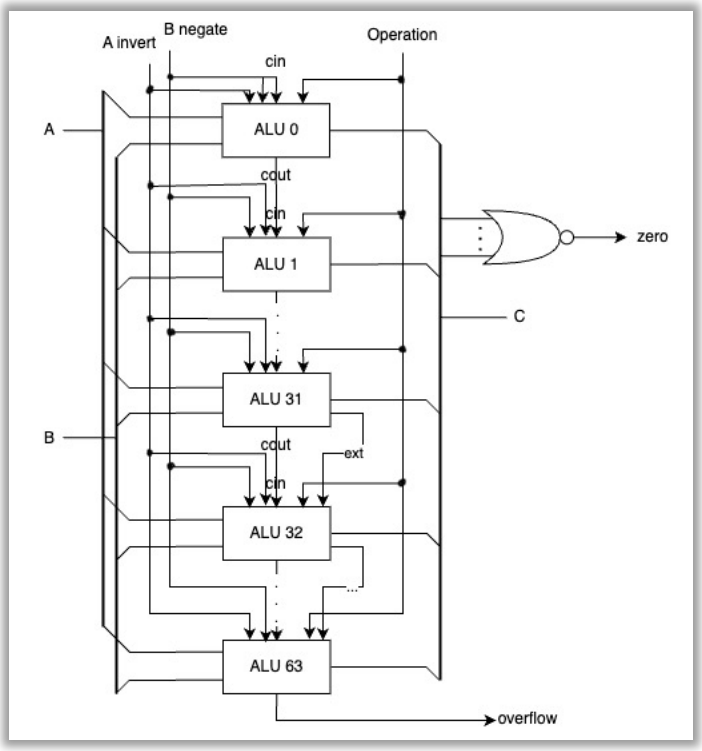
ALU32~ALU62



ALU63



64-bit ALU



2.

(a)

Iteration	step	N	M	Product
0	Initial Values	1001	0000 1110	0000 0000
1	1 → Product += M	1001	0000 1110	0000 1110
	Shift left M	1001	0001 1100	0000 1110
	Shift right N	0100	0001 1100	0000 1110
2	0 → No operation	0100	0001 1100	0000 1110
	Shift left M	0100	0011 1000	0000 1110
	Shift right N	0010	0011 1000	0000 1110
3	0 → No operation	0010	0011 1000	0000 1110
	Shift left M	0010	0111 0000	0000 1110
	Shift right N	0001	0111 0000	0000 1110
4	1 → Product += M	0001	0111 0000	0111 1110
	Shift left M	0001	1110 0000	0111 1110
	Shift right N	0000	1110 0000	0111 1110

(b)

Iteration	step	M	Product
0	Initial Values	1110	0000 1001
1	1 → prod[0:3] += M	1110	1110 1001
	Shift right Product	1110	0111 0100
2	0 → No operation	1110	0111 0100
	Shift right Product	1110	0011 1010
3	0 → No operation	1110	0011 1010
	Shift right Product	1110	0001 1101
4	1 → prod[0:3] += M	1110	1111 1101
	Shift right Product	1110	0111 1110

3.

(a)

Iteration	step	Quotient	Divisor	Remainder
0	Initial Values	0000	0101 0000	0000 0111
1	Rem -= Div	0000	0101 0000	1011 0111
	Rem<0→+Div, LSL Q	0000	0101 0000	0000 0111
	Shift Div right	0000	0010 1000	0000 0111
2	Rem -= Div	0000	0010 1000	1101 1111
	Rem<0→+Div, LSL Q	0000	0010 1000	0000 0111
	Shift Div right	0000	0001 0100	0000 0111
3	Rem -= Div	0000	0001 0100	1111 0011
	Rem<0→+Div, LSL Q	0000	0001 0100	0000 0111
	Shift Div right	0000	0000 1010	0000 0111
4	Rem -= Div	0000	0000 1010	1111 1101
	Rem<0→+Div, LSL Q	0000	0000 1010	0000 0111
	Shift Div right	0000	0000 0101	0000 0111
5	Rem -= Div	0000	0000 0101	0000 0010
	Rem≥0→LSL Q, Q0=1	0001	0000 0101	0000 0010
	Shift Div right	0001	0000 0010	0000 0010

(b)

Iteration	step	Remainder | (Quotient)	Divisor
0	Initial Values	0000 0111	0101
	Shift Remainder left	0000 1110	0101
1	Rem[0:3] -= Div	1011 1110	0101
	Rem[0:3]<0→+Div, LSL R	0001 1100	0101
2	Rem[0:3] -= Div	1100 1100	0101
	Rem[0:3]<0→+Div, LSL R	0011 1000	0101
3	Rem[0:3] -= Div	1110 1000	0101
	Rem[0:3]<0→+Div, LSL R	0111 0000	0101
4	Rem[0:3] -= Div	0010 0000	0101
	Rem[0:3]≥LSL R, R0=1	0100 0001	0101
5	Shift left Rem[0:3]	0010 0001	0101

4.

(a) For each pattern, we can transform it into 4 binary bits. 0 represents 0000, 5 represents 0101, 9 represents 1001, 4 represents 0100, 8 represents 1000, D represents 1101, E represents 1110, and C represents 1001. So it is "0000 0101 1001 0100 1000 1101 1110 1100" in binary, 93621740 in decimal in two's complement integer.

If it is an unsigned number, the result is still same since the leftmost bit is 0, which indicates that it is positive.

(b) Similarly, we will have that F represents 1111, A represents 1010, 6 represents 0110, B represents 1011, 7 represents 0111, 2 represents 0010, 1 represents 0001, and 4 represents 0100. So it is "1111 1010 0110 1011 0111 0010 0001 0100" in binary, -93621740 in decimal in two's complement integer.

If it is an unsigned number, the result is 4201345556, which is different, since the leftmost bit is 1, which indicates that it is negative.

(c)

Sign	Exponent (8 bits)	Fraction (23 bits)
0	000 0101 1	001 0100 1000 1101 1110 1100
1	111 1010 0	110 1011 0111 0010 0001 0100

i. For $05948DEC_{16}$

- Sign: positive
- Exponent: $11 - 127 = -116$
- Significand: $1 + 2^{-3} + 2^{-5} + 2^{-8} + 2^{-12} + 2^{-13} + 2^{-15} + 2^{-16} + 2^{-17} + 2^{-18} + 2^{-20} + 2^{-21} \approx 1.1605811119$

Representation: $1.1605811119 \times 2^{-116} \approx 1.396998678 \times 10^{-35}$

ii. For $FA6B7214_{16}$

- Sign: negative
- Exponent: $244 - 127 = 117$
- Significant: $1 + 2^{-1} + 2^{-2} + 2^{-4} + 2^{-6} + 2^{-7} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-13} + 2^{-18} + 2^{-20} \approx 1.8394823$

Representation: $-1.8394823 \times 2^{117} \approx -3.0563642 \times 10^{35}$

5.

(a)

i. For $\mathrm{X} = 88.4375$

We transform it into binary representation, 1011000.0111_2. Then we normalize 1011000.0111_2 to $1.0110000111_2 \times 2^6$. After that, we calculate biased exponent, $6 + 127 = 133_{10} = 10000101_2$. The fraction is 0110000111 from normalized form and padding it to 23 bits, which is 01100001110000000000000. Thus, we have

- Sign: 0
- Exponent: 10000101
- Fraction: 01100001110000000000000

In representation, $\mathrm{X} = 01000010101100001110000000000000_2$.

ii. For $\mathrm{Y} = -7.3125$

We transform 7.3125 into binary representation, 111.0101_2. Then we normalize 111.0101_2 to $1.110101_2 \times 2^2$. After that, we calculate biased exponent, $2 + 127 = 129_{10} = 10000001_2$. The fraction is 110101 from normalized form and padding it to 23 bits, which is 11010100000000000000000. Thus we have

- Sign: 1
- Exponent: 10000001
- Fraction: 11010100000000000000000

In representation, $\mathrm{Y} = 11000000111010100000000000000000_2$.

(b) For exponent, the new biased exponent is $133 + 129 - 127 = 135 = 8 + 127$. For significant, we multiply the two given significants $1.0110000111_2 \times 1.110101_2 = 10.1000011010110011_2$. Then we have $10.1000011010110011_2 \times 2^8 = 1.01000011010110011_2 \times 2^9$. Thus,

- Sign: 1
- Exponent: 10001000
- Fraction: 01000011010110011000000

In representation, $\mathrm{X} \times \mathrm{Y} = 11000100001000011010110011000000_2$.

6.

(a) For normalized number, the smallest exponent field is 1, so the biased exponent is $1-255=-254$, The fraction field is 000000 and with a hidden 1 before the binary point. So $a_0 = 1.000000 \times 2^{-254} = 1.0 \times 2^{-254}$.

(b)

i. For the largest positive denormalized number, its representation is 0000000000111111. So the scientific notation is $a_1 = 0.111111 \times 2^{-254} = 1.11111 \times 2^{-255}$.

ii. For the second largest positive denormalized number, its representation is 0000000000111110. So the scientific notation is $a_2 = 0.111110 \times 2^{-254} = 1.1111 \times 2^{-255}$.

(c)

i. Difference between a_0 and a_1 is $a_0 - a_1 = 0.000001 \times 2^{-254} = 2^{-260}$.

ii. Difference between a_1 and a_2 is $a_1 - a_2 = 0.000001 \times 2^{-254} = 2^{-260}$.

(d) For the binary pattern 1011110110100111, we have

- Sign: 1
- Exponent: $011110110_2 = 246$, so the exponent is -9 with bias adjustment.
- Fraction: 100111

So the scientific notation is -1.100111×2^{-9}.

(e) To find the nearest representation, we have to convert it to binary. For integer part, $1_{10} = 1_2$. For fraction part, $0.31_{10} = 0.01001111\ldots_2$. Its guard bit is 1, so rounding up, $0.010011111\ldots_2 \approx 0.0101_2$ So in scientific notation, $\mathrm{U} = 1.0101_2 \times 2^0$.

The actual decimal number is 1.3125.

7.

(a) Incorrect. It is correct for negative integers since we need to add bias $2^k - 1$ for effectively adjusting for the division by 2^k. But it is incorrect for positive integers.

(b) Correct. It is correct for both negative and positive integers.

(c) Incorrect. It is incorrect for negative integers. This is a counterexample with $X = -3_{10} = 1\ldots101_2$. The result of $X \gg 2$ is $1\ldots111_2 = -1_{10}$, which is inconsistent with $(X / 4) = 0$.

(d) Correct. It will check the significant bit. If it is negative, $(X \gg 31)\&3 = 3$. Otherwise $(X \gg 31)\&3 = 0$. So the result will be same as (b).