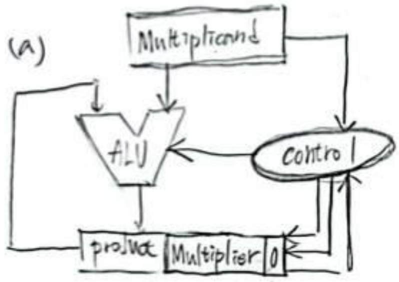
Computer Organization HW03-2 112590022

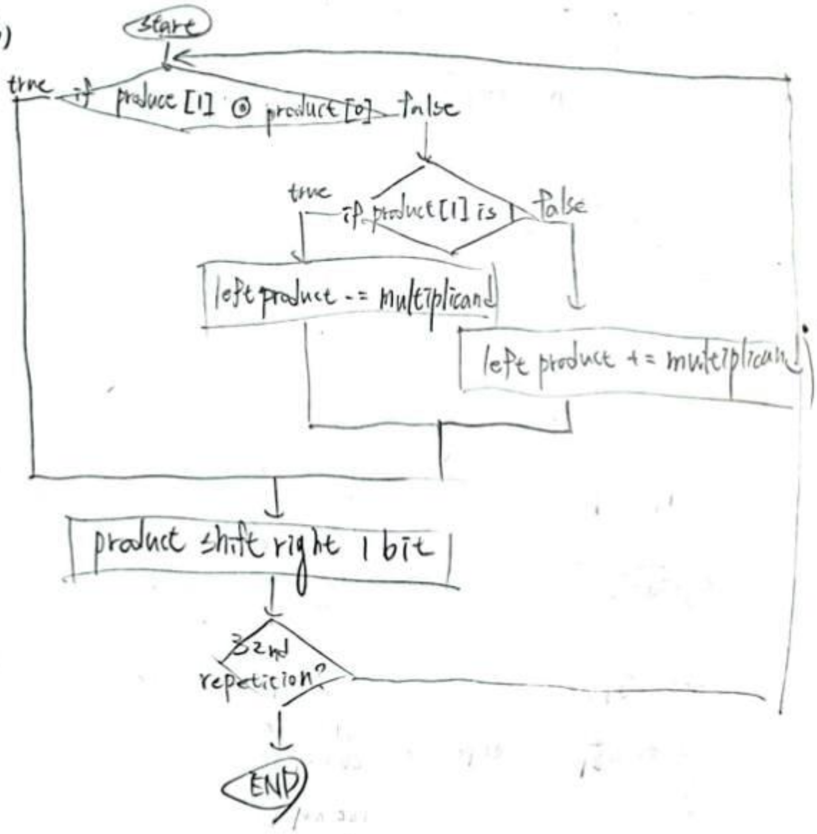
1. 147 = 10010011 28 = 00011100

Iteration	step	Remainder	Divisor
0	shift left Remainder 1 bit	00000001 00100110	00011100
1	left Remainder -= Divisor	11100101 00100110	00011100
2	if left Remainder < 0		
3	left Remainder += Divisor	00000001 00100110	
4	shift left Remainder 1 bit	00000010 01001100	
5	if not repetition 9th		
	⋮		
19	shift left Remainder 1 bit	00100100 11000000	
20	if not repetition 9th		
21	left Remainder -= Divisor	00001000 11000000	
22	if left Remainder ≥ 0		
23	shift left Remainder 1 bit and set 1	00010001 10000001	
24	if not repetition 9th		
25	left Remainder -= Divisor	11110101 10000001	
26	if left Remainder < 0		
27	left Remainder += Divisor	00010001 10000001	
28	shift left Remainder 1 bit	00100011 00000010	
29	if not repetition 9th		
30	left Remainder -= Divisor	00000111 00000010	
31	if left Remainder ≥ 0		
32	shift left Remainder 1 bit and set 1	00001110 00000101	
33	if repetition 9th		
34	left Remainder shift left 1 bit	00000111 00000101	

2. (a)



(b)



(c) in some cases,
Booth's algorithm can achieve the goal with less computation than the original multiplication method.

8.

(a) $-23.25 = 10111.01_{(2)} = 1.011101 \times 2^{(4)}$ $E = 4$

Exponent = 4 + bias = 4 + 127 = 131 = $10000011_{(2)}$

Fraction = $011101_{(2)}$

S = 1 (because −23.25 is negative number ⇒ $(-1)^1$)

$\underset{S}{1}$ $\underbrace{10000011}_{\text{Exponent}}$ $\underbrace{01110100000000000000000}_{\text{Fraction}}$ #

(b) (121) $-127 = -6$ E

$\underset{\text{Negative}}{1}$ 0111001 01101 00……

01101 → $0.25 + 0.125 + 0.03125 = 0.40625$

$-1.01101_{(2)} \times 2^{-6} = -1.40625/64 = -0.02197265625$ #

(c)

$\underset{\text{positive}}{0}$ $\underset{\text{MAX } 127+127=254}{11111110}$ 11111111 11111111 1111111 #

(d) $\underset{\text{positive}}{0}$ $\underset{\text{min } -127+127=0}{00000000}$ 000000000 000000000 001 #

(e)

sign	Exponent	Fraction	object represented
0	11111111	00000000000000000000000	$+\infty$
1	11111111	00000000000000000000000	$-\infty$
1/0	11111111	≠00000000000000000000000	NaN

(f) Negative 130

1 100 0001 0000 0010 1011 0001 1111 0010

$-1.0\ldots0 \times 2^{(130-127)} = 1.02105545997161963 \times 8 = 8.16844367980957$ #

4.

(a) $0.3125 = 0.0101_{(2)} \Rightarrow 1.0101 \times 2^0$

$1\ 01111\ 0101000000$ #

(b) $10011000.01 + 0.0101010101$

$= 1.0011000001 \times 10^7 + 1.0101010101 \times 10^{-2}$

$= 1.0011000001 \times 10^7 + 0.0000000001010101 \times 10^7$

$= 1.0011000100101010101 \times 10^7$

$\underline{0}\ \underline{00111}\ 0011000100101010\underline{1}\underline{0}\underline{1}\underline{0}$ $\left\{ \begin{array}{l} \text{Guard} = 0 \\ \text{round} = 1 \\ \text{sticky} = 0 (0 \ldots 0) \end{array} \right.$ 捨去

GRS

$\Rightarrow 0\ 00111\ 0011000100101010$ #

(c) $110.0111001 \times 1000.01101$

$= 1.100111001 \times 2^2 \times 1.00001101 \times 2^3$

$= 11010.0010111001 01 \times 2^{(2+3)}$

$= 1.1011000101110010 1 \times 2^{(5+5)}$

$0\ 01010\ 1011000101110010\underline{010}$ $\left\{ \begin{array}{l} \text{Guard} = 0 \\ \text{round} = 1 \\ \text{sticky} = 0 (0 \ldots 0) \end{array} \right.$ 捨去

GRS

$\Rightarrow 0\ 01010\ 1011000101110 01$ #