CS202 · Computer Organization

Homework 3

刘乐奇 / 12011327

April 5, 2022

1 Problem 1

$50_{10} = 11\ 0010_2, 9_{10} = 00\ 1001_2$
The procedure is as below (from top to bottom):

Iter	Step	Product	Multiplicand (Mand)	Multiplier (Mer)
1	initial values	0000 0000 0000	0000 0011 0010	00 1001
2	Mer's LSB = 1 → pro = pro + Mand	0000 0011 0010	0000 0011 0010	00 1001
	Mand shift left and Mer shift right	0000 0011 0010	0000 0110 0100	00 0100
3	Mer's LSB = 0 → pro = pro	0000 0011 0010	0000 0110 0100	00 0100
	Mand shift left and Mer shift right	0000 0011 0010	0000 1100 1000	00 0010
4	Mer's LSB = 0 → pro = pro	0000 0011 0010	0000 1100 1000	00 0010
	Mand shift left and Mer shift right	0000 0011 0010	0001 1001 0000	00 0001
5	Mer's LSB = 1 → pro = pro + Mand	0001 1100 0010	0001 1001 0000	00 0001
	Mand shift left and Mer shift right	0001 1100 0010	0011 0010 0000	00 0000
6	Mer's LSB = 0 → pro = pro	0001 1100 0010	0011 0010 0000	00 0000
	Mand shift left and Mer shift right	0001 1100 0010	0110 0100 0000	00 0000
7	Mer's LSB = 0 → pro = pro	0001 1100 0010	0110 0100 0000	00 0000
	Mand shift left and Mer shift right	0001 1100 0010	1100 1000 0000	00 0000

Finally, the result is $0001\ 1100\ 0010_2 = 450_{10}$.

2 Problem 2

$98_{10} = 0110\ 0010_2, 17_{10} = 0001\ 0001_2$
The procedure is as below (from top to bottom):

Iter	Step	Product | Multiplier (P|M)	Multiplicand (Mand)
1	initial values	0000 0000 | 0001 0001	0110 0010
2	P|M's LSB = 1 → P|M's most left 8-bit += Mand	0110 0010 | 0001 0001	0110 0010
	P|M shift right	0011 0001 0 | 000 1000	
3	P|M's LSB = 0 → no change of P|M	0011 0001 0 | 000 1000	0110 0010
	P|M shift right	0001 1000 10 | 00 0100	
4	P|M's LSB = 0 → no change of P|M	0001 1000 10 | 00 0100	0110 0010
	P|M shift right	0000 1100 010 | 0 0010	
5	P|M's LSB = 0 → no change of P|M	0000 1100 010 | 0 0010	0110 0010
	P|M shift right	0000 0110 0010 | 0001	
6	P|M's LSB = 1 → P|M's most left 8-bit += Mand	0110 1000 0010 | 0001	0110 0010
	P|M shift right	0011 0100 0001 0 | 000	
7	P|M's LSB = 0 → no change of P|M	0011 0100 0001 0 | 000	0110 0010
	P|M shift right	0001 1010 0000 10 | 00	
8	P|M's LSB = 0 → no change of P|M	0001 1010 0000 10 | 00	0110 0010
	P|M shift right	0000 1101 0000 010 | 0	
9	P|M's LSB = 0 → no change of P|M	0000 1101 0000 010 | 0	0110 0010
	P|M shift right	0000 0110 1000 0010 |	

Finally, the result is $0000\ 0110\ 1000\ 0010_2 = 1666_{10}$.

3 Problem 3

$60_{10} = 11\ 1100_2, 18_{10} = 01\ 0010_2$
The procedure is as below:

Iter	Step	Quotient	Divisor	Remainder
0	initial values	000000	0100 1000 0000	0000 0011 1100
1	Rem = Rem - Div	000000	0100 1000 0000	1011 1011 1100
	Rem < 0 → + Div, shift 0 into Quo	000000	0100 1000 0000	0000 0011 1100
	shift Div right	000000	0010 0100 0000	0000 0011 1100
2	Rem = Rem - Div	000000	0010 0100 0000	1101 1111 1100
	Rem < 0 → + Div, shift 0 into Quo	000000	0010 0100 0000	0000 0011 1100
	shift Div right	000000	0001 0010 0000	0000 0011 1100
3	Rem = Rem - Div	000000	0001 0010 0000	1111 0001 1100
	Rem < 0 → + Div, shift 0 into Quo	000000	0001 0010 0000	0000 0011 1100
	shift Div right	000000	0000 1001 0000	0000 0011 1100
4	Rem = Rem - Div	000000	0000 1001 0000	1111 1010 1100
	Rem < 0 → + Div, shift 0 into Quo	000000	0000 1001 0000	0000 0011 1100
	shift Div right	000000	0000 0100 1000	0000 0011 1100
5	Rem = Rem - Div	000000	0000 0100 1000	1111 1111 0100
	Rem < 0 → + Div, shift 0 into Quo	000000	0000 0100 1000	0000 0011 1100
	shift Div right	000000	0000 0010 0100	0000 0011 1100
6	Rem = Rem - Div	000000	0000 0010 0100	0000 0001 1000
	Rem ≥ 0 → shift 1 into Quo	000001	0000 0010 0100	0000 0001 1000
	shift Div right	000001	0000 0001 0010	0000 0001 1000
7	Rem = Rem - Div	000001	0000 0001 0010	0000 0000 0110
	Rem ≥ 0 → shift 1 into Quo	000011	0000 0001 0010	0000 0000 0110
	shift Div right	000011	0000 0000 1001	0000 0000 0110

After (n+1) = 6+1 = 7 steps (where n indicates the bits representing the numbers), the result is $00\ 0011_2 = 3_{10}$.

4 Problem 4

- Since $-0.9375_{10} = -0.1111_2 = (-1)^1 \times 1.111_2 \times 2^{-1}$, the fraction is $11\ 1000\ 0000_2$ and the exponent is $-1 + bias = 14_{10} = 0\ 1110_2$. Thus, the bit pattern to represent -0.9375_{10} is 1 01110 11 1000 0000.
- For the smallest value, the exponent is 00001_2 (actually $= 1 - 15 = -14$) and the fraction is 0000000000_2 (significand = 1.0). Thus, the smallest value is $\pm 1.0 \times 2^{-14}$.
  For the largest value, the exponent is 11110_2 (actually $= 30 - 15 = 15$) and the fraction is 1111111111_2 (significand = $1.9990234375 \approx 2.0$). Thus, the largest value is $\pm 2.0 \times 2^{15}$.
  Therefore, the range is $(-2.0 \times 2^{15}, -1.0 \times 2^{-14}] \cup [1.0 \times 2^{-14}, 2.0 \times 2^{15})$.
- The relative accuracy is 2^{-10}. Solution is shown as below:

$$\begin{aligned}\frac{\Delta A}{|A|} &= \frac{2^{-10} \times 2^y}{|1.xxx \times 2^y|} \\ &\le \frac{2^{-10} \times 2^y}{|1 \times 2^y|} \\ &= 2^{-10}\end{aligned}$$

5 Problem 5

The decimal digits of precision of 16-bit half precision number is $10 \times \log_{10} 2 \approx 3$.
Step 1: compare and align with guard, round and sticky bit.

$$\begin{aligned}26.125_{10} &= 2.61250_{10} \times 10^1 \\ 0.2900390625_{10} &= 0.02901_{10} \times 10^1\end{aligned}$$

Step 2: add

$$2.61250_{10} + 0.02901_{10}(\times 10^1) = 2.64151(\times 10^1)$$

Step 3: round
With {guard bit, round bit, sticky bit} = 151, we have

$$2.64151 \times 10^1 \approx 2.64 \times 10^1$$

So the final result is 2.64×10^1.