

Assignment 3

Due Date : March 29, 2021 (day)

Name : 易 翔
Student ID : 11912013
Score : _____

Problem 1 (3.9) Score: _____

Solution: $151_{(10)} = 0_1001_0111_{(2)} = 1001_0111_{(\text{signed 2's complement})} = -105$

$214_{(10)} = 0_1101_0110_{(2)} = 1101_0110_{(\text{signed 2's complement})} = -42$

$-105 + (-42) = -147$

However, consider the answer is out of range, so the answer become -128 because the range of the signed-8-bit number is $-128 \sim 127$. ☐

Problem 2 (3.10) Score: _____

Solution: $151_{(10)} = 0_1001_0111_{(2)} = 1001_0111_{(\text{signed 2's complement})} = -105$

$214_{(10)} = 0_1101_0110_{(2)} = 1101_0110_{(\text{signed 2's complement})} = -42$

$-105 - (-42) = -63$

☐

Problem 3 (3.11) Score: _____

Solution: $151_{(10)} = 1001_0111_{(2)}$

$214_{(10)} = 1101_0110_{(2)}$

$151 + 214 = 365$

However, consider the answer is out of range, so the answer become 256 because the range of the unsigned-8-bit number is $0 \sim 256$. ☐

Problem 4 (3.13) Score: _____

Step	Action	Multiplicand	Product/Multiplier
0	Initial values	0110_0010	0000_0000_0001_0010
1	No operation	0110_0010	0000_0000_0001_0010
	Right shift Product	0110_0010	0000_0000_0000_1001
2	Product += Mcand	0110_0010	0110_0010_0000_1001
	Right shift Product	0110_0010	0011_0001_0000_0100
3	No operation	0110_0010	0011_0001_0000_0100
	Right shift Product	0110_0010	0001_1000_1000_0010
4	No operation	0110_0010	0001_1000_1000_0010
	Right shift Product	0110_0010	0000_1100_0100_0001
5	Product += Mcand	0110_0010	0110_1110_0100_0001
	Right shift Product	0110_0010	0011_0111_0010_0000
6	No operation	0110_0010	0011_0111_0010_0000
	Right shift Product	0110_0010	0001_1011_1001_0000
7	No operation	0110_0010	0001_1011_1001_0000
	Right shift Product	0110_0010	0000_1101_1100_1000
8	No operation	0110_0010	0000_1101_1100_1000
	Right shift Product	0110_0010	0000_0110_1110_0100

Solution: The answer is $0000_0110_1110_0100_{(\text{binary})} = 1764$. REcheck that $62_{(\text{hex})} \times 12_{(\text{hex})} = 98 \times 18 = 1764$. Then get the answer. ☐

Problem 5 (3.16) Score: _____

Solution: It may pass three ($\log_2 8$) level of adders. And for each level of adders it may take 4 units of time. So the answer is $3 \times 4 = 12$ time units. ☐

Problem 6 (3.18) Score: _____

Solution: 1. 将寄存器中减去除数寄存器中的内容，将结果放在余数寄存器中；

2. 测试余数

- a. 余数 ≥ 0 ，商寄存器左移，最低位设为1；
- b. 余数 < 0 ，通过给余数寄存器加上除数寄存器的内容来恢复原值，结果放在余数寄存器。商同样左移，最低为设为0。

3. 除数寄存器右移一位。

Then we obtain the answer from the table following: The quotient is 3_{hex} , and the remainder is 11_{hex} . ☐

Step	Action	Quotient	Divisor	Remainder
0	Initial values	0000.0000	0010.0001.0000.0000	0000.0000.0111.0100
1	1.	0000.0000	0010.0001.0000.0000	1101.1111.0111.0100
	2b.	0000.0000	0010.0001.0000.0000	0000.0000.0111.0100
	3.	0000.0000	0001.0000.1000.0000	0000.0000.0111.0100
2	1.	0000.0000	0001.0000.1000.0000	1110.1111.1111.0100
	2b.	0000.0000	0001.0000.1000.0000	0000.0000.0111.0100
	3.	0000.0000	0000.1000.0100.0000	0000.0000.0111.0100
3	1.	0000.0000	0000.1000.0100.0000	1111.1000.0011.0100
	2b.	0000.0000	0000.1000.0100.0000	0000.0000.0111.0100
	3.	0000.0000	0000.0100.0010.0000	0000.0000.0111.0100
4	1.	0000.0000	0000.0100.0010.0000	1111.1100.0101.0100
	2b.	0000.0000	0000.0100.0010.0000	0000.0000.0111.0100
	3.	0000.0000	0000.0010.0001.0000	0000.0000.0111.0100
5	1.	0000.0000	0000.0010.0001.0000	1111.1110.0110.0100
	2b.	0000.0000	0000.0010.0001.0000	0000.0000.0111.0100
	3.	0000.0000	0000.0001.0000.1000	0000.0000.0111.0100
6	1.	0000.0000	0000.0001.0000.1000	1111.1111.0110.1100
	2b.	0000.0000	0000.0001.0000.1000	0000.0000.0111.0100
	3.	0000.0000	0000.0000.1000.0100	0000.0000.0111.0100
7	1.	0000.0000	0000.0000.1000.0100	1111.1111.1111.0000
	2b.	0000.0000	0000.0000.1000.0100	0000.0000.0111.0100
	3.	0000.0000	0000.0000.0100.0010	0000.0000.0111.0100
8	1.	0000.0000	0000.0000.0100.0010	0000.0000.0011.0010
	2a.	0000.0001	0000.0000.0100.0010	0000.0000.0011.0010
	3.	0000.0001	0000.0000.0010.0001	0000.0000.0011.0010
9	1.	0000.0001	0000.0000.0010.0001	0000.0000.0001.0001
	2a.	0000.0011	0000.0000.0010.0001	0000.0000.0001.0001
	3.	0000.0011	0000.0000.0001.0000	0000.0000.0001.0001