CSci370 Computer Architecture: Homework 2 (double-sided)

Due date: On or before Monday, March 30, 2020 in class
Absolutely no copying others' works **Name:** ____________________________

- There are four algorithms discussed for multiplication and division. Make sure you are using the correct ones for the two questions below.
- The purpose of homeworks is for students to practice for the exams without others' help, so the penalty of mistakes will be minor.
- Without practicing for the exams properly, students would not be able to do well on the exams.

---

1. **(Refined multiplication: 50%)** Using a table similar to that shown in the Slide 8.6, calculate the product of the octal unsigned 6-bit integers 65_8 (or 110101_2) and 53_8 (or 101011_2) using the hardware and algorithm described in the figures of Slide 8.6. You should show the content of each register on each step.
Ans>

Iteration		Multiplicand	Carry	Product = HI, LO
0	Initialize (LO=Multiplier, HI=0)	110101		000000 101011
1	LO[0]=1 ⇒ Add.		0	000000 101011 + 110101 000000 110101 101011
	Shift product right by 1 bit.	110101		011010 110101
2	LO[0]=1 ⇒ Add.		1	011010 110101 + 110101 000000 001111 110101
	Shift product right by 1 bit.	110101		100111 111010
3	LO[0]=0 ⇒ Do nothing.			
	Shift product right by 1 bit.	110101		010011 111101
4	LO[0]=1 ⇒ Add.		1	010011 111101 + 110101 000000 001000 111101
	Shift product right by 1 bit.	110101		100100 011110
5	LO[0]=0 ⇒ Do nothing.			
	Shift product right by 1 bit.	110101		010010 001111
6	LO[0]=1 ⇒ Add.		1	010010 001111 + 110101 000000 000111 001111
	Shift product right by 1 bit.	110101		100011 100111

2. **(First-version division: 50%)** Using a table similar to that shown in the Slide 8.10, calculate the octal unsigned 6-bit integer 65_8 (or 110101_2) divided by another octal unsigned 6-bit integer 16_8 (or 001110_2) using the hardware and algorithm described in the figures of Slide 8.10. You should show the content of each register on each step.
†Note that you have to actually show the differences in the procedures, not just the signs.
Ans>

Iteration		Remainder	Divisor	Difference	Quotient
0	Initialize	000000 110101	001110 000000		000000
1	1: SHR, SHL, Difference	000000 110101	000111 000000	000000 110101 - 000111 000000 ⇓ 000000 110101 + 111001 000000 111001 110101	000000
	2: Diff<0 ⇒ Do Nothing				
2	1: SHR, SHL, Difference	000000 110101	000011 10000	000000 110101 - 000011 100000 ⇓ 000000 110101 + 111100 100000 111101 010101	000000
	2: Diff<0 ⇒ Do Nothing				
3	1: SHR, SHL, Difference	000000 110101	000001 110000	000000 110101 - 000001 110000 ⇓ 000000 110101 + 111110 010000 111111 000101	000000
	2: Diff<0 ⇒ Do Nothing				
4	1: SHR, SHL, Difference	000000 110101	000000 111000	000000 110101 - 000000 111000 ⇓ 000000 110101 + 111111 001000 111111 111101	000000
	2: Diff<0 ⇒ Do Nothing				
5	1: SHR, SHL, Difference	000000 110101	000000 011100	000000 110101 - 000000 011100 ⇓ 000000 110101 + 111111 100100 + 000000 011001	000000
	2: Diff≥0 ⇒ Rem=Diff, set lsb Quotient	000000 011001			000001
6	1: SHR, SHL, Difference	000000 011001	000000 001110	000000 011001 - 000000 001110 ⇓ 000000 011001 + 111111 110010 + 000000 001011	000010
	2: Diff≥0 ⇒ Rem=Diff, set lsb Quotient	000000 001011			000011