Homework #3 **4190.308 Computer Architecture**

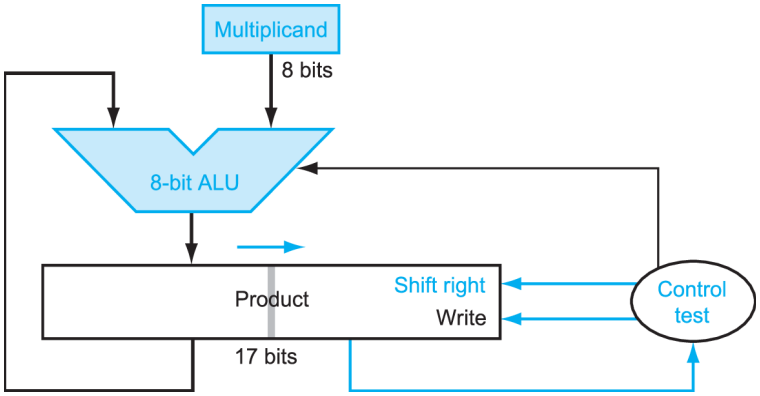
Due Date: Monday, March 29, 2023 **Sample Solution**

Submission: electronically on eTL (scan & upload)

Question 1

Integer Multiplication

For this problem we use a 16-bit version of the multiplier discussed in the book in Chapter 3.3 / Figure 3.5.

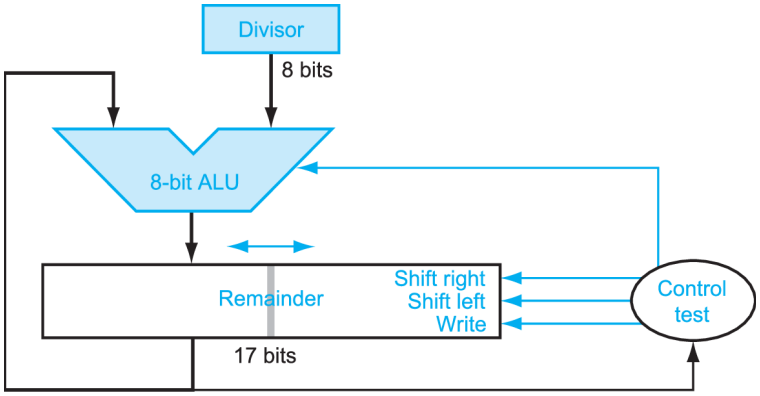


Compute the product 106 x 81 and show the values of the product register, the multiplicand register, and the action of control for each step.

Step	Product register	Multiplicand	Control
Init	0 0000 0000 0101 0001	0110 1010	**add & shift**
1	0 0011 0101 0010 1000	0110 1010	**shift**
2	0 0001 1010 1001 0100	0110 1010	**shift**
3	0 0000 1101 0100 1010	0110 1010	**shift**
4	0 0000 0110 1010 0101	0110 1010	**add & shift**
5	0 0011 1000 0101 0010	0110 1010	**shift**
6	0 0001 1100 0010 1001	0110 1010	**add & shift**
7	0 0100 0011 0001 0100	0110 1010	**shift**
8	0 0010 0001 1000 1010	0110 1010	**done**
	0x218A = 8586 = 81 x 106		

Question 2
Integer Division

For this problem we use a 16-bit version of the divider discussed in the book in Chapter 3.4 / Figure 3.11.



Compute the result of the division 97 / 5 and show the values of the remainder register, the divisor register, and the action of control for each step.

Step	Remainder register	Divisor	Control
Init	0 0000 0000 0110 0001	0000 0101	**smaller: shift in 0**
1	0 0000 0000 1100 0010	0000 0101	**smaller: shift in 0**
2	0 0000 0001 1000 0100	0000 0101	**smaller: shift in 0**
3	0 0000 0011 0000 1000	0000 0101	**smaller: shift in 0**
4	0 0000 0110 0001 0000	0000 0101	**bigger: sub, shift in 1**
5	0 0000 0010 0010 0001	0000 0101	**smaller: shift in 0**
6	0 0000 0100 0100 0010	0000 0101	**smaller: shift in 0**
7	0 0000 1000 1000 0100	0000 0101	**bigger: sub, shift in 1**
8	0 0000 0111 0000 1001	0000 0101	**bigger: sub, shift in 1**
9	0 0000 0010 0001 0011	0000 0101	**done**
	remainder quotient		
	2 19		

Note that in the last iteration *only the quotient is shifted (not the remainder)!*

Question 3

Floating Point Addition

Given is the floating point format “fp8” with the following organization.

sign	exp			frac			
7	6	5	4	3	2	1	0

The fields `exp` and `frac` are encoded in the same way as in the IEEE 754 standard. The bias is -3.

Compute the sum of a = 5.5 and b = 0.55 in fp8. Follow the steps outlined in the textbook, chapter 3.5. When rounding, use round-to-even mode.

Step 0: fp8 representation

	sign	exp				frac					
a = 5.5	0	1	0	1	1.	0	1	1	0		
b = 0.55	0	0	1	0	1.	0	0	0	1	1	0

Step 1: Align

	sign	exp				frac					
a	0	1	0	1	1.	0	1	1	0	0	0
b	0	1	0	1	0.	0	0	1	0	0	0

Step 2: Add

	sign	exp				frac					
a+b	0	1	0	1	1.	1	0	0	0	0	0

Step 3: Normalize

	sign	exp				frac				G	R
a+b	0	1	0	1	1.	1	0	0	0	0	0

Step 4: Round

	sign	exp				frac				G	R
a+b	**0**	**1**	**0**	**1**	**1.**	**1**	**0**	**0**	**0**	0	0

a+b = 6.0