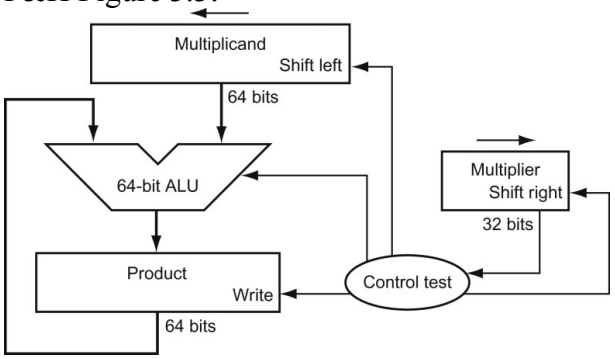
Homework 3 – Arithmetic

Out: 2.22.21
Due: 3.3.21

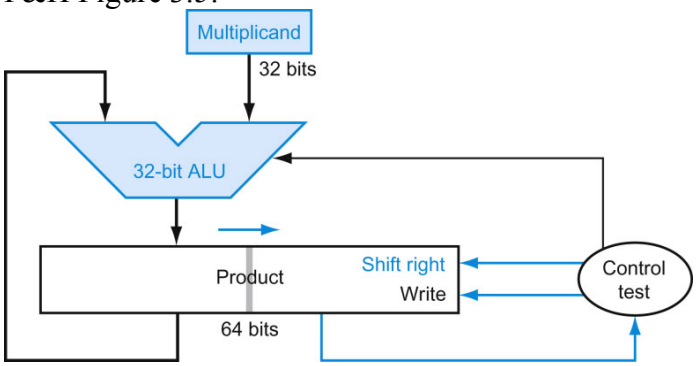
1. [Self Study]
   Practice working through the designs that we went over in class with the following test cases.
   a) 10-bit Carry Select Adder: 0111011001 + 0011000010
   b) ~~10-bit subtraction circuit: 0111011001 - 0011000010~~
   c) 4-bit multiplication -- do for "pencil-and-paper" on both multiply circuits that we saw in class (figures 3.3 and 3.5 in textbook): 1101 x 1011
   d) "unrolled" 4-bit multiplier: 1101 x 1011
   e) Carry-Save-Adder (CSA) based 4-bit multiplier -- do for both "pencil-and-paper" and the circuit

2. [Carry Select Adders]
   What is the timing of:
   a) The first 8-bit carry select adder (in the notes) built out of 3 4-bit ripple carry adders?
   b) A 12-bit adder out of 5 4-bit ripple carry adders?
   c) The best 64-bit carry-select adder you can build out of a single-size of ripple carry adder? How big are the adders? How many are there?

3. [Sequential Timing]
   Assume ripple carry adders. How long do the following circuits take to compute their respective operations? Explain.
   a) P&H Figure 3.3:

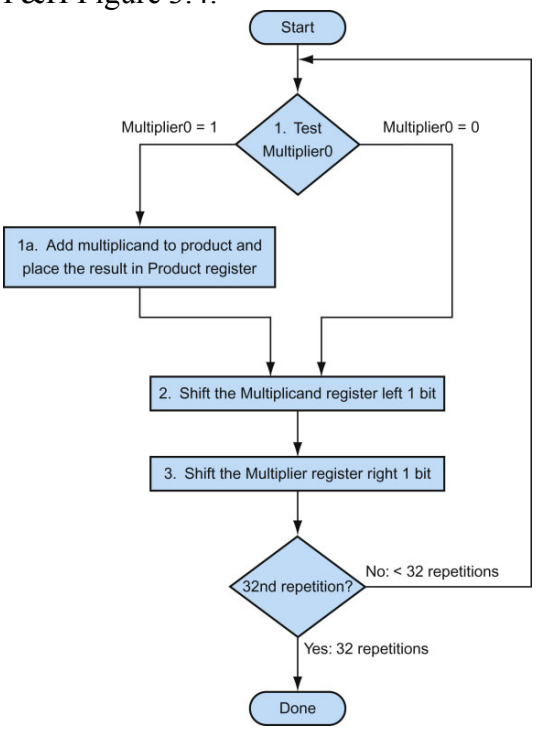


b) P&H Figure 3.5:



4. [Multiplication]
Calculate the time necessary to perform a multiply using the approach given in P&H Figures 3.3 (above) and 3.4 (below) if an integer is 8 bits wide and each step of the operation takes 4 time units. Assume that in step 1a an addition is always performed – either the multiplicand will be added, or a zero will be. Also assume that the registers have already been initialized. If this is being done in hardware, the shifts of the multiplicand and multiplier can be done simultaneously. If this is being done in software, they will have to be done one after the other. Solve for each case.

P&H Figure 3.4:



5. [Multiplication]
In class we discussed two array multipliers. How many gate delays to they take (as a function of input size N)?
a) Based on ripple carry adders only

b) Based on Carry Save Adders
c) Why is "unrolled" so much faster than the shift-and-add multiplier?

6. [Multiplication]
The CSA-based multiplier circuit (last version we saw in class) appears to be suboptimal. In particular, full adders in the first row have 0s as inputs whereas they could also have other inputs (e.g. a0 AND b2 and a1 AND b2 for the first two full adders, respectively). Doing more useful work earlier at no circuit cost should be an improvement. Try redrawing the circuit to maximize the number of full adders with three partial-product bit inputs (replacing a 0, a carry, or a sum from one or more of the full adder). Make sure it still works and that each FA has at most 3 inputs! Show your drawing. Does the timing change?

7. [Floating Point]
Write down the binary representation of the decimal number 63.25 assuming the IEEE 754 single precision format.

8. [Floating Point]
In this problem, assume integers and floats are both 9 bits. The floating point representation consists of:

1 sign bit
4 bits biased exponent, bias = 7, E_min = -6, E_max = 7
4 bits fraction
Exponents with all 0's or all 1's follow a convention analogous to the IEEE standard.
The rounding mode is IEEE nearest even.

Give answers to the following problems in the floating point representation just presented.

a) The smallest possible positive number
b) The largest possible positive number (not infinity)
c) The smallest number greater than 1
d) The largest number less than 1
e) Integer 001101010
f) Product of floating point numbers: 010100100 x 011110000
g) Product of floating point numbers: 010100100 x 001010001
h) Difference of floating point numbers: 010000011 – 001110111
i) Sum of floating point numbers: 010000000 + 000111111