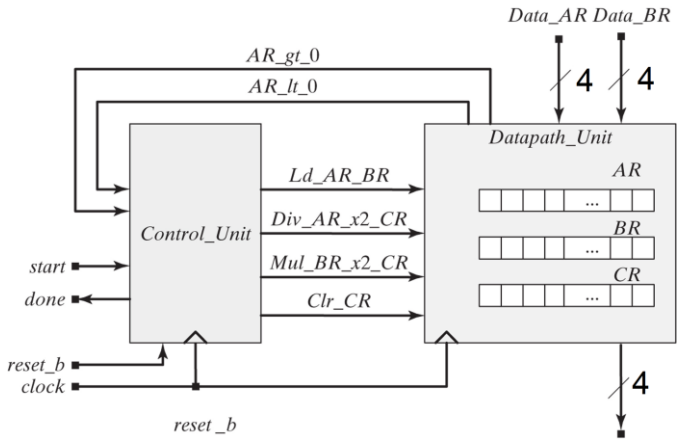
COE 405, Term 122

Design & Modeling of Digital Systems

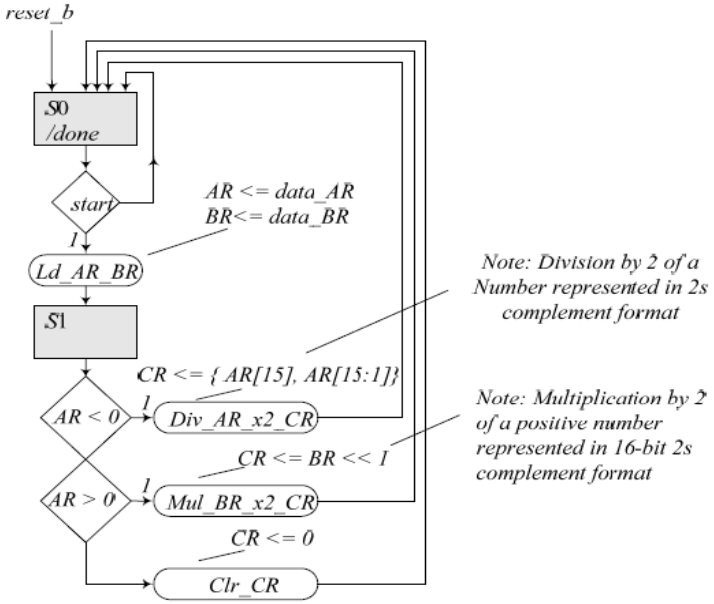
HW# 3 Solution

Due date: Saturday, March 16

Q.1. The block diagram below shows the datapath and controller for a machine that transfers two 4-bit signed numbers in 2's complement representation into registers AR and BR, divides the number in AR by 2 and transfers the result to register CR if the number in AR is negative, multiplies the number in BR by 2 and transfers the result to register CR if the number in AR is positive but non-zero, and if the number in AR is zero, clears register CR to 0.

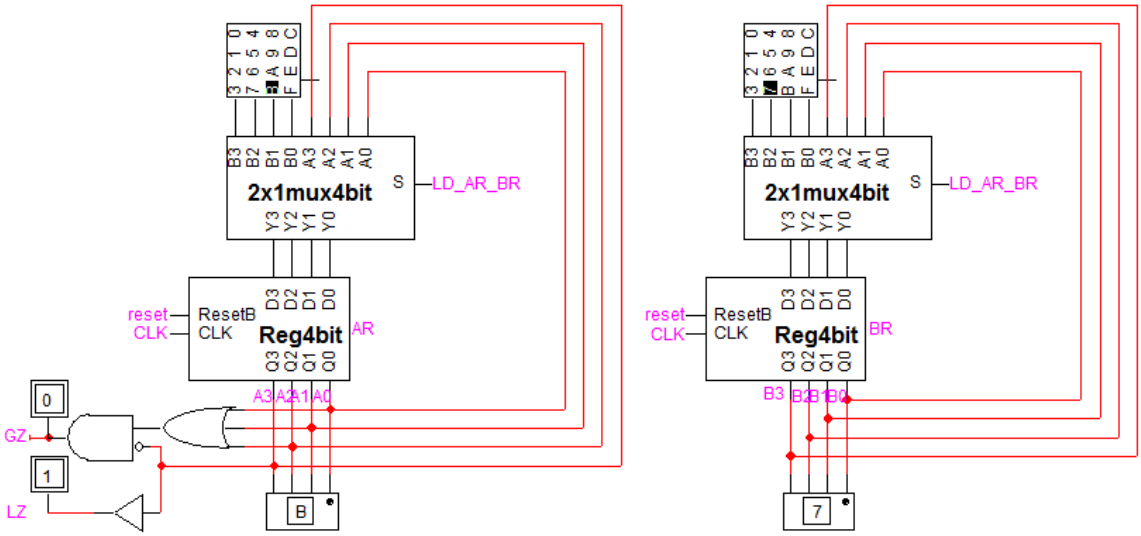


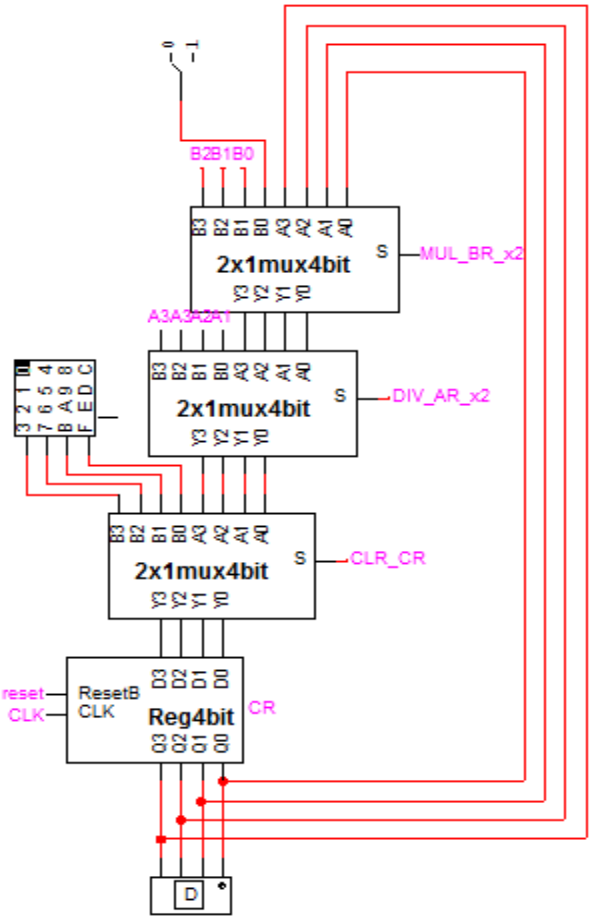
(i) Develop an ASMD chart for the machine.



(ii) Show the design of the data-path and control unit of the machine.

Data Path:

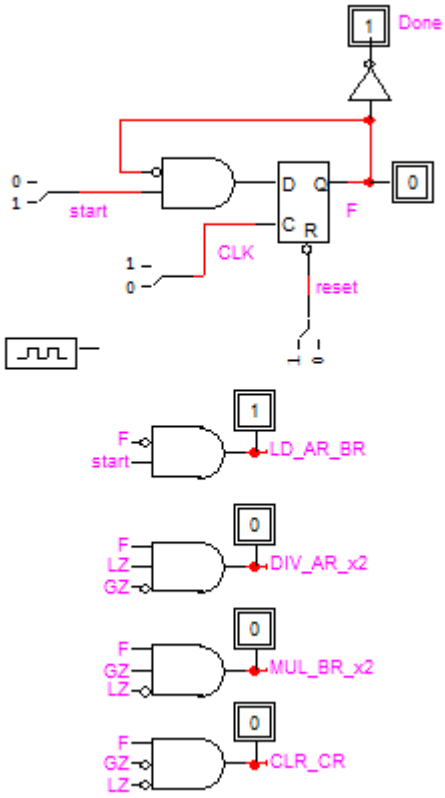




Control Unit:

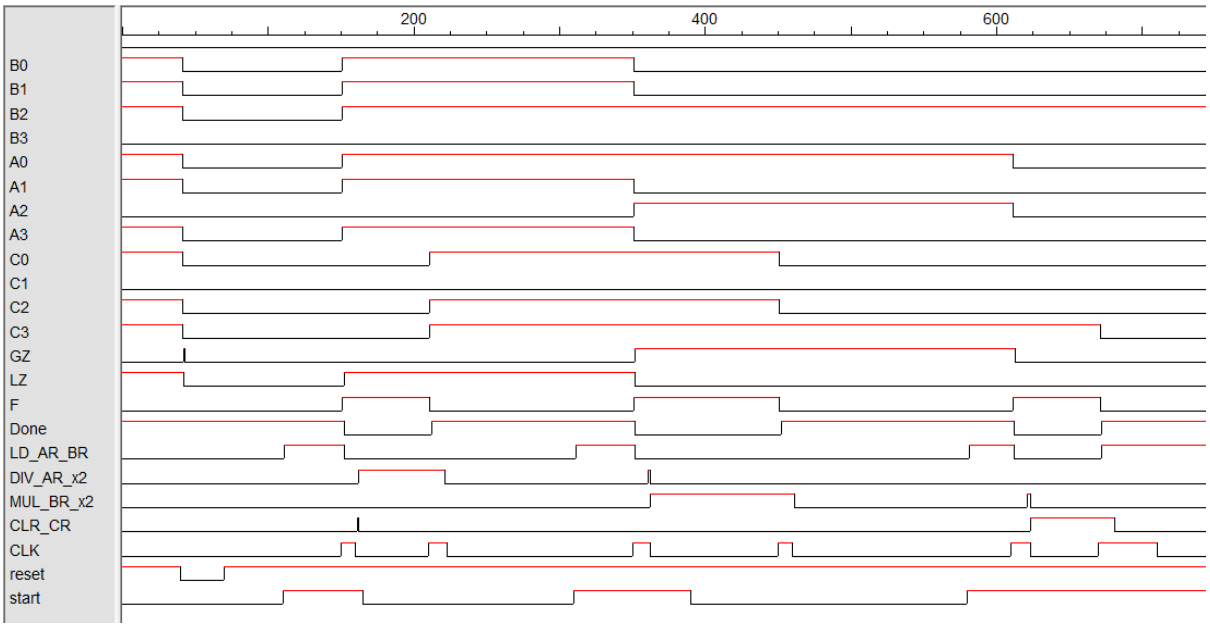
C.S.	Input			N.S.	Output				
	start	AR_gt_0	AR_lt_0		Done	LD_AR_BR	Div_AR_x2	Mul_AR_x2	Clr_CR
S0	0	x	x	S0	1	0	0	0	0
S0	1	x	x	S1	1	1	0	0	0
S1	x	0	1	S0	0	0	1	0	0
S1	x	1	0	S0	0	0	0	1	0
S1	x	0	0	S0	0	0	0	0	1

We assume the state assignment S0=0 and S1=1.



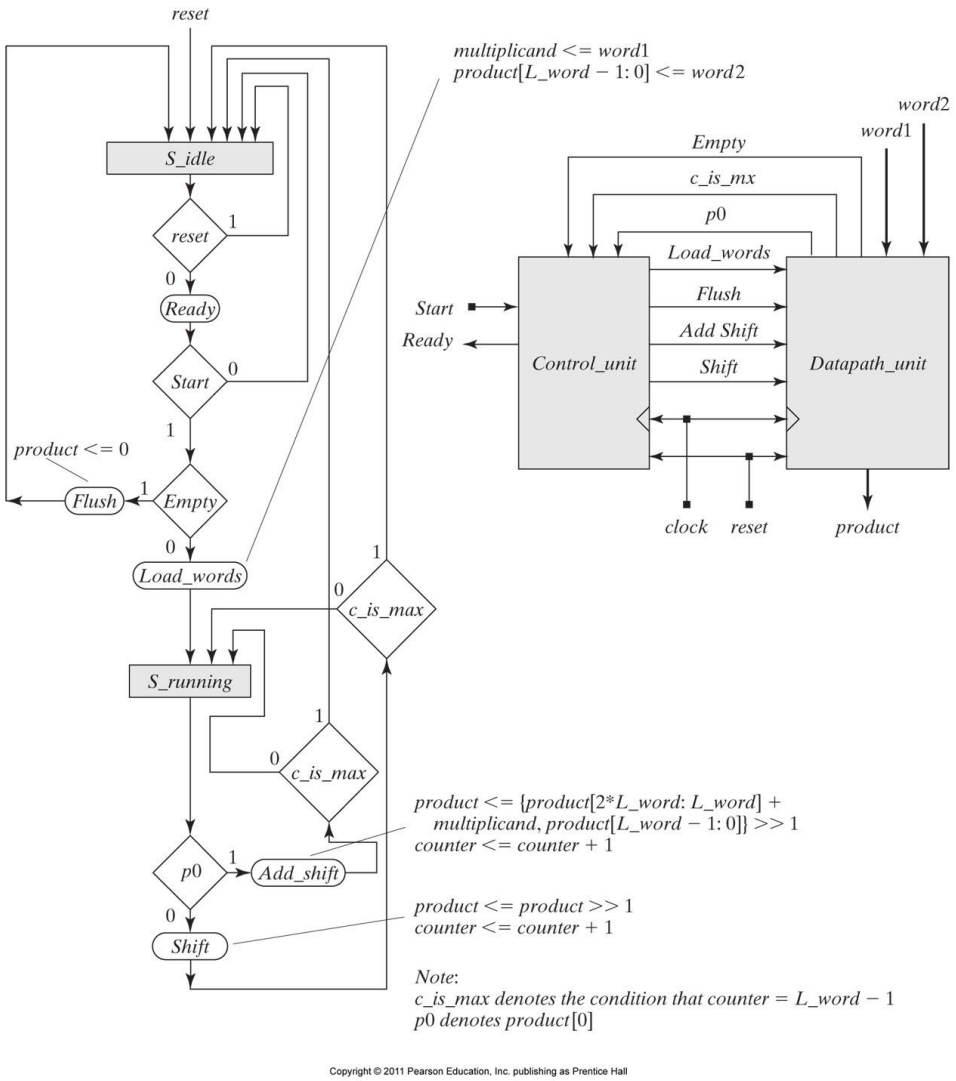
(iii) Implement the machine and verify its correct functionality by simulation.

Silmulation Results:



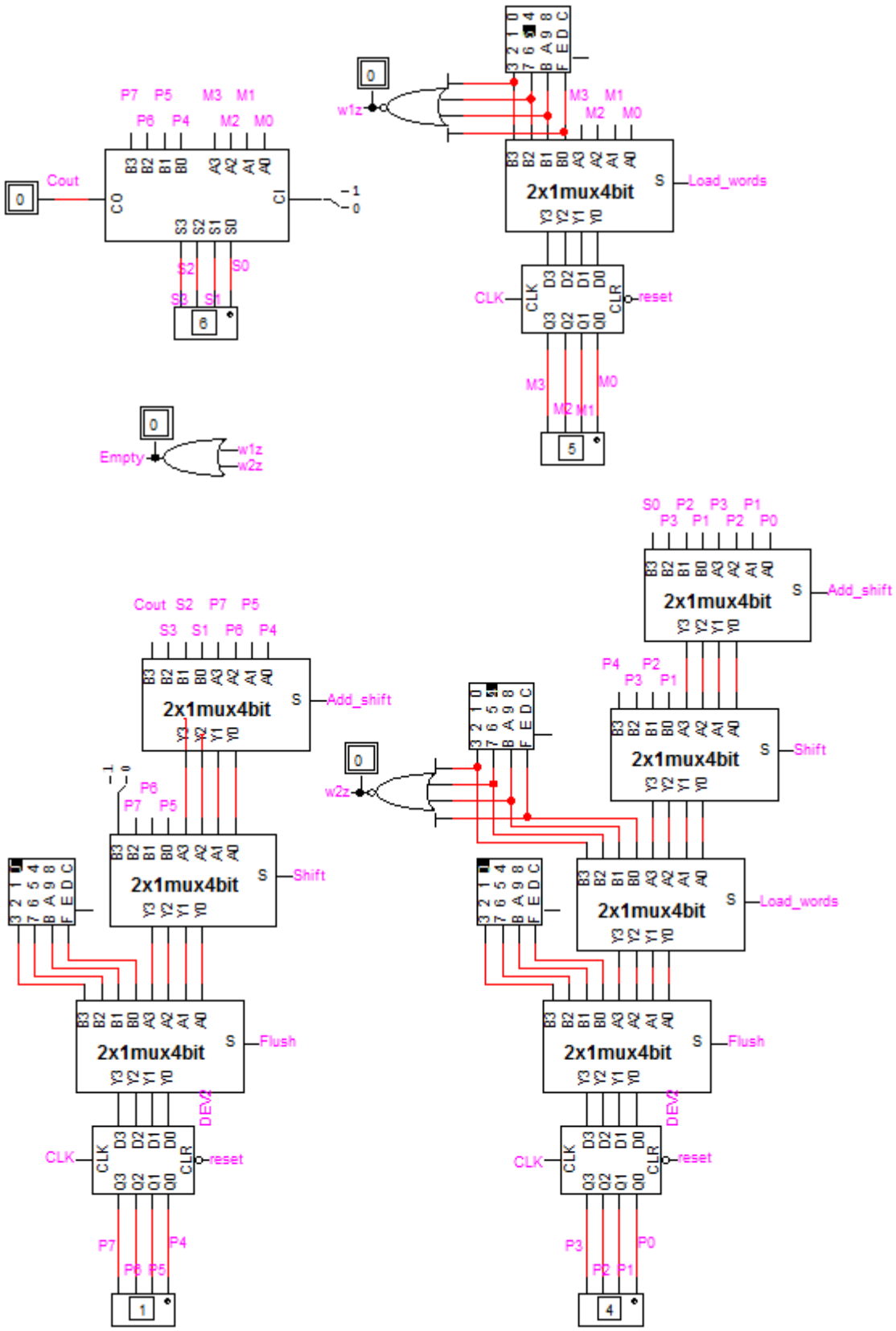
Simulation results above test the three cases and demonstrate correct operation of the circuit.

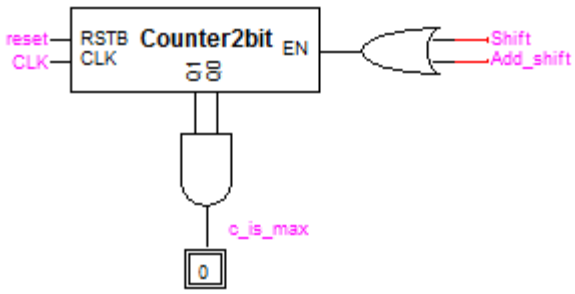
Q.2. It is required to design an unsigned 4-bit sequential multiplier. The multiplier is assumed to have an 8-bit register to hold the result, a 4-bit register to hold the multiplicand and a 3-bit counter. When reset is 1, the multiplicand, the product and counter registers are reset. When operation is started, the multiplicand register is loaded with word1 while the least significant 4-bits of the product are loaded with word2. The block diagram and the ASMD chart of the sequential multiplier are given below. *Ready* signals that the unit is ready to accept a command to multiply. If word1 or word2 are 0, *Empty* is set to 1. *Flush* loads the product with 0. *C_is_mx* is set when the counter is equal to 3. *P0* is the least significant bit of the product i.e. product[0].



(i) Show the design of the data-path and control unit of the 4-bit sequential multiplier.

Data Path:

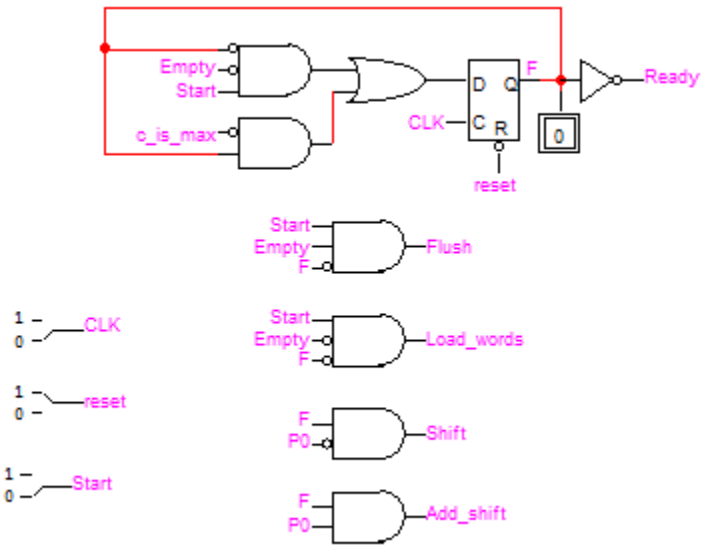

2x1mux4bit
Load_words
Add_shift
Shift
Flush
reset
CLK
Cout
Empty
w1z
w2z
DEM2



Control Unit:

C.S.	Input				N.S.	Output				
	Start	Empty	P0	c_is_max		Ready	Flush	Load_words	Shift	Add_shift
S_idle	0	x	x	X	S_idle	1	0	0	0	0
S_idle	1	1	x	X	S_idle	1	1	0	0	0
S_idle	1	0	x	x	S_running	1	0	1	0	0
S_running	x	x	0	0	S_running	0	0	0	1	0
S_running	x	x	0	1	S_idle	0	0	0	1	0
S_running	x	x	1	0	S_running	0	0	0	0	1
S_running	x	x	1	1	S_idle	0	0	0	0	1

We assume the state assignment S_idle =0 and S_running =1.



(ii) Implement the machine and verify its correct functionality by simulation.

As shown above multiplying 4x5 produces the correct result of 20=’h14.

Also, multiplying 10*6=60=’h3C.