

1. (25 pts) For the flowgraph in Figure 1, assume that the multiply operators are pipelined by 1 clock.
- a. What is the MINIMUM latency (in clocks) that can be achieved for this flowgraph? (the multiply/add operators cannot be chained). Show some work if you want partial credit.

*Minimum latency is N1 (1 clk) + N3 (2 clks) + N4 (1 clk) = 4 clks.*

- b. Show an execution unit schedule for an implementation that achieves this minimum latency with an initiation rate of 1 CLOCK.

*First, determine overlapped operations so can get execution unit count. We KNOW what the initiation rate is, it has been specified as 1 clock.*

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Clk 1	N1, N2				
Clk 2	N3	N1, N2			
Clk 3		N	N1, N2		
Clk 4	N4		N3	N1, N2	
Clk 5		N4		N3	N1, N2
Clk 6			N4		N3
				N4	

*Line in bold is generalized schedule.*

	Sample J-3	Sample J-2	Sample J-1	Sample J	
Clk I	N4 adder	N3 (2 <sup>nd</sup> clk mult)	N3 (1 <sup>st</sup> clkmult) N2 (2 <sup>nd</sup> clk mult)	N1 (input ) N2 (1 <sup>st</sup> clk mult)	

*We are starting two new multiplies, and finishing two old multiplies. This means that we only need two multipliers. We also need one Adder*

*Execution Unit schedule:*

	Mult A	Mult B	Adder
Clk I	Start N3 (sample J-1) Finish N3 (sample J-2)	Start N2 (sample J) Finish N2 (J-1)	N4 (sample J-3)

*Many of you got your operation table mixed up with your Execution Unit schedule. Some just did an operation table, some created an execution unit schedule but made it look somewhat like an operation schedule. You need to determine the operation schedule first (depends on initiation rate and if there are constraints on number of resources), and then determine your execution unit schedule. These two tables are separate entities.*

*Notice that the multipliers are working on the same node operation (Mult A does N3, Mult B does N2) - this is necessary when you have an initiation rate = 1, there is not time to switch node functions*

2. (25 pts) For the flowgraph in Figure 2, show an execution unit table that achieves the *best possible* (fewest clocks) initiation rate. Assume that execution units are NOT PIPELINED (can be completed in 1 clock cycle) and that the execution units cannot be chained.

We do NOT know the initiation rate, we need to determine what the fastest possible initiation rate is. We do NOT have a resource constraint (fixed number of multipliers or adders). Note that the longest path in the flowgraph is 3 clocks (N1, N3, N4), so one possible initiation rate is 3 as shown in the table below:

One thing to note is that Node N2 uses the previous OUTPUT value.

	Sample J	Sample J+1
Clk 1	N1, N2 (depends on Sample J-1)	
Clk 2	N3	
Clk 3	N4	
Clk 4		N1, N2 (depends on sample J)
Clk 5		N3
Clk 6		N4

This is an initiation Rate = 3. Can we do better? How about Initiation Rate = 2?


	Sample J	Sample J+1
Clk 1	N1, N2 (depends on Sample J-1)	
Clk 2	N3	
Clk 3	N4	N1, N2 (depends on sample J)
Clk 4		N3
Clk 5		N4
Clk 6		

This WILL NOT WORK. N2 under sample J+1 needs N4 from the previous Sample, but it is not ready yet!!! Move N2 to be 1 clk later.

	Sample J	Sample J+1
Clk 1	N1	
Clk 2	N3, N2 (depends on Sample J-1)	
Clk 3	N4	N1
Clk 4		N3, N2 (depends on sample J)
Clk 5		N4
Clk 6		

Now this works fine. N4 of Sample J is ready when N2 of sample J+1 needs it.

What about initiation Rate = 1 ???

	Sample J	Sample J+1
Clk 1	N1	
Clk 2	N3, N2 (depends on Sample J-1)	N1
Clk 3	N4 	N3, N2 (depends on sample J)
Clk 4		N4
Clk 5		
Clk 6		

This WILL NOT WORK. N2 (J+1) needs N4 (J) which is not ready yet. Initiation Rate =1 not possible.

So, general schedule for Initiation Rate = 2:

	Sample J-1	Sample J
Clk I	N4	N1,
Clk I+1		N3, N2 (depends on sample J)

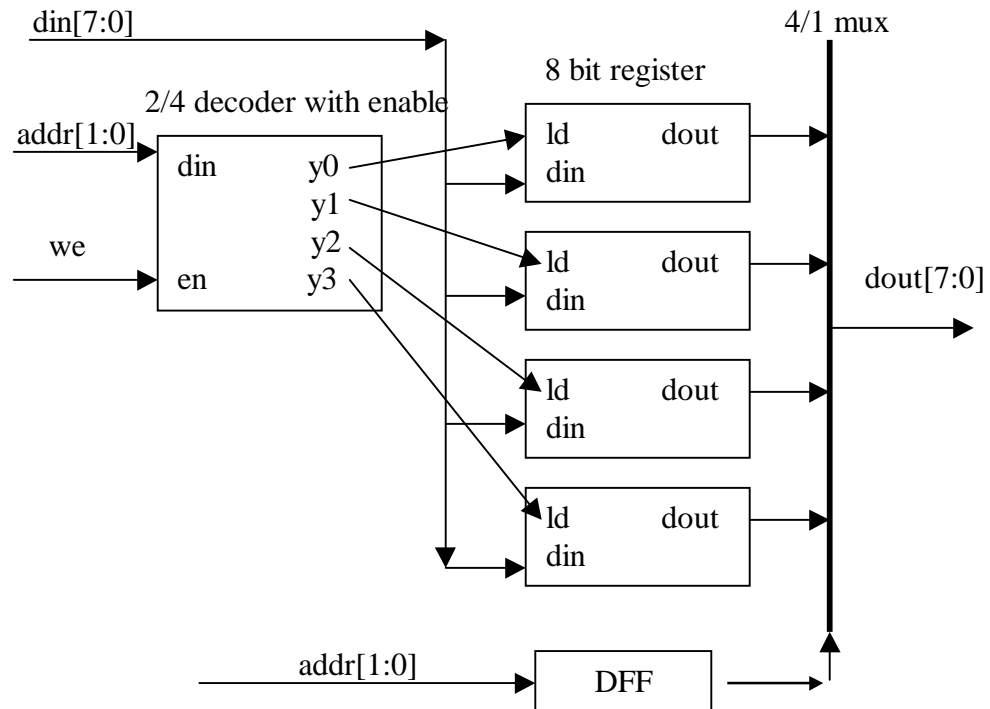
We will need one adder (Clk I), and two multipliers ( Clk I+1)

Execution Unit schedule:

	Adder	Mult A	Mult B
Clk I	N4 (J-1)	Idle	idle
Clk I+1		N3 (J)	N2(J)

3. (15 pts) Draw the block diagram of a 4 x 8 RAM built using Registers. The functionality should emulate the RAM\_DQ Altera block where all inputs are registered but the output is not.

The interface is  $\text{din}[7:0]$ ,  $\text{addr}[1:0]$ ,  $\text{we}$  (high true),  $\text{dout}[7:0]$



4. (10 pts) What is a scan path used for? Draw a diagram

*See notes.*

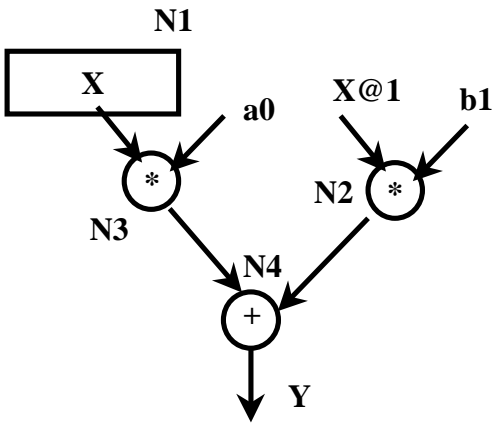
5. see notes. (15 pts) Many datasheets refer to a JTAG port or Boundary scanning. What is this? What extra pins are needed? Give their functionality.

*See notes.*

6. (10 pts) What is the relationship between a DOT clock, Horizontal Sync, and Vertical Sync in a raster scan display? What are they used for?

*See notes.*

**Figure 1**



**Figure 2**

