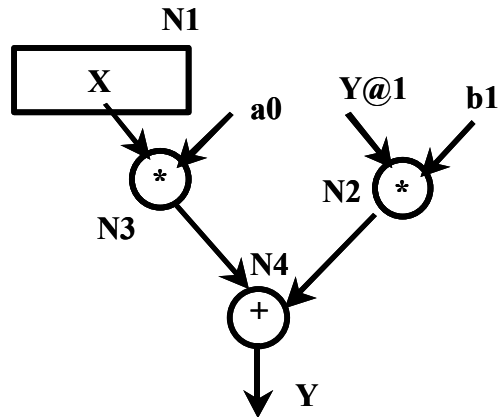


For any partial credit, you must show your work.

1. (15 pts) For the flowgraph below, show a schedule that will use one multiplier, one adder with the minimum number of clock cycles. Do not use any overlapped computations. You may not need all the clock cycles shown in the table. Show register allocation!!

a. How many registers do you need? (do not count a0, b1 registers) *2 Registers*

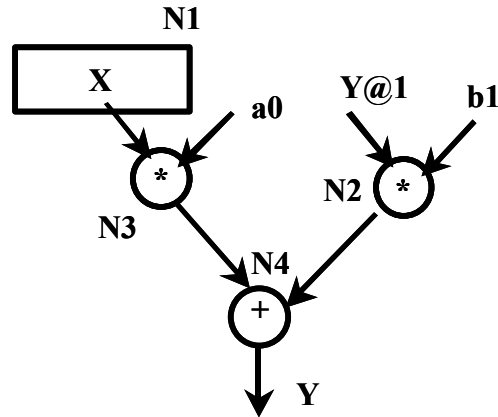
**Figure 1**



	Multiplier	Adder	Register transfer ops
Clk 1	N2 (RA=Y@1 * b1)		RA ← X RB ← N2
Clk 2	N3 (RA=X * a0)		RA ← N3
Clk 3		N4 (RB=N2 + RA=N3)	RA ← N4

2. (15 pts) For the flowgraph below, show a schedule that will use one multiplier, one adder with the minimum number of clock cycles. Assume the multiplier has one pipeline stage. Do not use any overlapped computations. You may not need all the clock cycles shown in the table. Show register allocation!!
- a. How many registers do you need? (do not count a0, b1 registers) *2 registers*

**Figure 1**



	Multiplier	Adder	Register transfer ops
Clk 1	N2 (RA=Y@1 * b1)		RA ← X
Clk 2	N3 (RA=X * a0)		RB ← N2
Clk 3			RA ← N3
Clk 4		N4 (RB=N2 + RA=N3)	RA ← N4
Clk 5			

3. (20 pts) For the flowgraph in problem 2, used overlapped computations to achieve an initiation rate faster than what can be achieved without overlapped computations. Assume all multipliers have a pipeline stage. Indicate what clocks form the ‘generalized’ schedule You may not need all of the samples/clocks shown. You do not have to show register allocation.

a. What is the latency of your schedule? 5

b. What is initiation rate? 3

c. How many multipliers are needed? 1

d. How many adders are needed? 1

*A solution that only used 1 multiplier but had 5 clocks of latency.*

	Sample J	Sample J+1	Sample J+2	Sample J+3	Sample J+4
Clk 1	N1				
Clk 2	N3				
Clk 3	N2				
Clk 4		N1			
Clk 5	N4	N3			
Clk 6		N2			
Clk 7					
Clk 8		N4			
Clk 9					
Clk 10					
Clk 11					
Clk 12					
Clk 13					
Clk 14					

- 3 (20 pts) For the flowgraph in problem 2, used overlapped computations to achieve an initiation rate faster than what can be achieved without overlapped computations. Assume all multipliers have a pipeline stage. Indicate what clocks form the 'generalized' schedule. You may not need all of the samples/clocks shown. You do not have to show register allocation.

e. What is the latency of your schedule? 4

f. What is initiation rate? 3

g. How many multipliers are needed? 2

h. How many adders are needed? 1

*A solution that used 2 multipliers but reduced latency to 4 clocks.*

	Sample J	Sample J+1	Sample J+2	Sample J+3	Sample J+4
Clk 1	N1				
Clk 2	N2, N3				
Clk 3					
Clk 4	N4	N1			
Clk 5		N2, N3			
Clk 6					
Clk 7		N4			
Clk 8					
Clk 9					
Clk 10					
Clk 11					
Clk 12					
Clk 13					
Clk 14					

4. (20 pts)

a. Adding execution units will (circle one for each):

LATENCY:	Increase	<b>decrease</b>	has no effect
CLOCK PERIOD	Increase	decrease	<b>has no effect</b>

Latency *decreases* because adding more execution units will let more operations done in fewer clocks because more execution units are available to perform the required computations.

*No effect* on clock period because we are not doing anything to shorten the combinational paths in the design.

b. Pipelining execution units will (circle one for each):

LATENCY:	<b>Increase</b>	decrease	has no effect
CLOCK PERIOD	Increase	<b>decrease</b>	has no effect

Every pipeline stage added to an execution unit will *increase* the latency by one clock cycle. However, the clock period is *decreased* because the combinational delay paths in execution unit is reduced.

c. Using overlapped computations will (circle one for each):

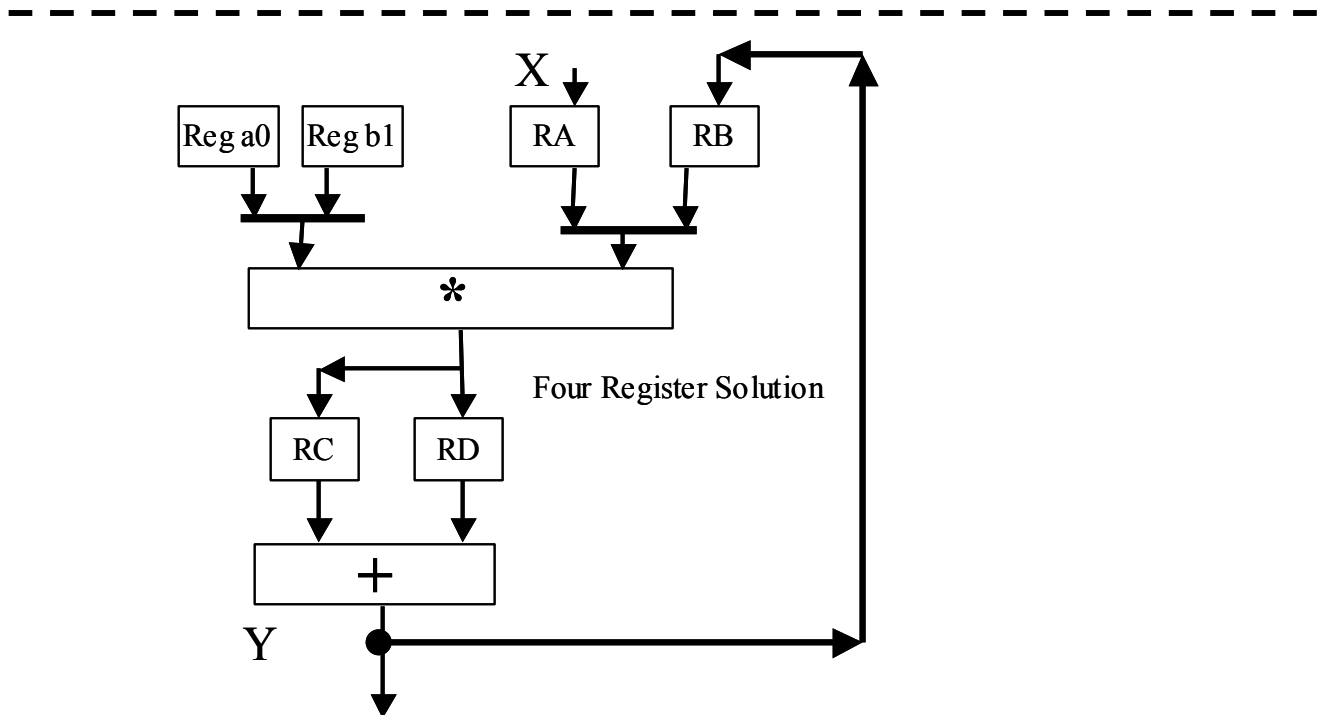
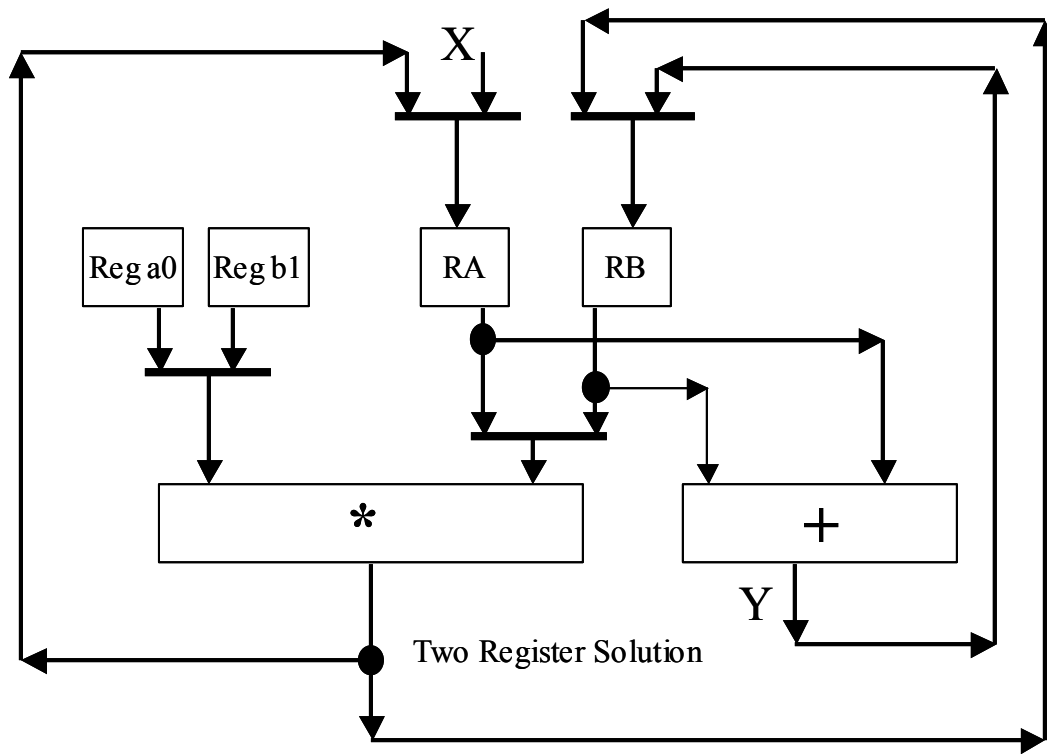
LATENCY:	Increase	decrease	<b>has no effect</b>
CLOCK PERIOD	Increase	decrease	<b>has no effect</b>
INITIATION RATE	Increase	<b>decrease</b>	has no effect

Working on more than one sample at a time (overlapped computations) *does not affect* the latency for a single sample.

*No effect* on clock period because we are not doing anything to shorten the combinational paths in the design.

Working on more than one sample at a time (overlapped computations) allows a new value to be input before the calculation for the current sample is finished, so the number of clocks between input values (initiation rate) will *decrease*.

5. (15 pts) Draw a datapath for your solution to problem #1 (two register solution matches schedule that is shown in #1 – the four register solution was another common approach).



Answer 3 of the next 5 problems: (15 pts)

6. What is an open drain output? Explain.

*Output that has two states of 0V (pull low) or high impedance.*

7. IO elements in FPGAs have a method for controlling output slew. Why is it important to control output slew rate ?

*Lower slew rates mean less  $di/dt$  when simultaneous outputs switch which mean less noise in the system.*

8. The IO element for most FPGAs has a DFF in it so that incoming signals can be latched if desired. When used to latch input signals, what timing parameter is minimized over simply using the DFF in one of the interior programmable cells?

*External set-up time.*

9. The IO element for most FPGAs has a DFF in it so that outgoing signals can be latched if desired. When used to latch output signals, what timing parameter is minimized over simply using the DFF in one of the interior programmable cells?

*Clock-to-out time.*

10. What is a very important delay source in an FPGA (actually, just about any complex programmable device) other than the delay of the basic programmable cells?

*Routing delays*