Name: Deepak Maruti Phadtare
ID:113786897

Part 1:

Max Frequency : 909.09 MHz

Critical Path:
Startpoint: f/data_out_reg[1]
(rising edge-triggered flip-flop clocked by clk)
Endpoint: m2/y_data_reg[21]
(rising edge-triggered flip-flop clocked by clk)

Power:

	Internal Power		Switching Power		Leakage Power		
Total =	(989.1874 uW	+	134.1521 uW	+	4.6461e+04 nW)	=	1.1698e+03 uW

Total Dynamic Power = 1.1233 mW

Area:
Total cell area: 2276.693972 um^2

a)Arithmetic operations required to convolve a vector x of length M=12 with a vector f of length N=5 :

L=(M-N)+1=8
Multiplication=L*N=8*5=40
Additions=L*(N-1)=8*4=32
Total Arithmetic Operations =40+32=72
General Formula= L*N + L*(N-1)

b) Control Module Working:

You can see the state diagram below.

There are total of 9 states:
init = will assign addr_x and addr_f to counter x and counter f variables respectively.
And make x_ready = 1, f_ready = 0, y_valid = 0 , en_acc = 0 , clear = 1;

upload = This state will make x_ready = 1 f_ready = 1 so that system can simultaneously upload their values to the memory by going to next_sate, upload_x and upload_f

upload_x = This will start uploading x values to memory (x_ready = 1) and i

upload_f = This will start uploading f values to memory (f_ready = 1)

*upload_x and upload_f happens simultaneously

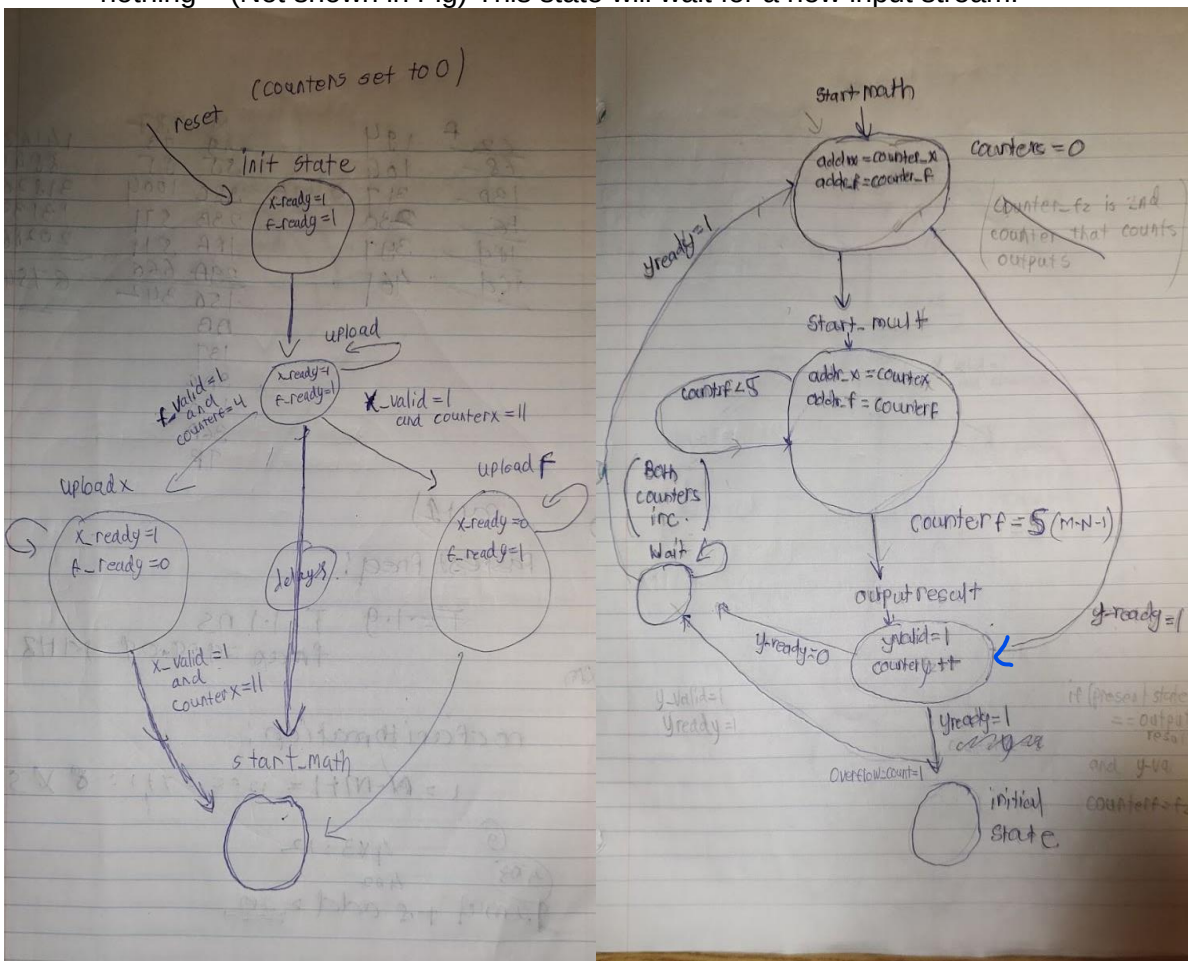
start_math = resets all the counters to zero.

start_mult = This states handles MAC en signal makes (en_acc=1) and increment counter x and f (in fig counter f <5 that 5=M-2(because address stars from zero))

output_result = This will display the output and increment counter y

wait_s = This state will incur if y_ready=0 and wait till it becomes 1.

nothing = (Not shown in Fig) This state will wait for a new input stream.



c) This system will not overflow, because

max value a 9-bit reg(x_data and y_data) =(+ or -) 511

number of additions for each value of y=N-1=4

max accumulated value (for single y value) =511*511*4= (+ or -)1044484

(+ or -)1044484 < (+ or -) 4194303 (max value 23 bit accumulator can hold).

And we are clearing the accumulator after we calculated new value of y.

d) I verified my system by testing against both simple and random testbench you provided and there were no errors while testing with both of them.I faced one problem while testing random testbench ,that I forgot to take into the consideration that both input f and x upload their last values at the same time. I fixed it with adding the condition in upload state :

```
if(present_state == upload && x_valid == 1 && counterx == M-1 && counterf == N-1)
        next_state = start_math
```

e)

Max Frequency : 909.09 MHz

Critical Path:
Startpoint: f/data_out_reg[1]
Endpoint: m2/y_data_reg[21]

The critical shown is correct in my design the critical path will from x memory output or f memory output to output of mac i.e, y_data because that path has a multiplier and adder.

Power:

		Internal Power		Switching Power		Leakage Power		
Total	=	(989.1874 uW	+	134.1521 uW	+	4.6461e+04 nW)	=	1.1698e+03 uW

Total Dynamic Power = 1.1233 mW

Area:
Total cell area: 2276.693972 μm^2

f)
Clock cycles the system takes to load one set of inputs: 12 cycles(Depends on size of bigger vector).
Clock cycles the system takes to compute y(i.e. multiply and accumulate): 40 cycles.
Clock cycles to clear the accumulator: 7 clock cycles(That's L-1)

Total Clock Cycles From the first cycle of loading the input until the last cycle of Outputting the result= 59 clock cycles
PERIOD=1.1 ns
Minimum delay of the system = 59*1.1 = 64.9 ns

g)
Area-delay product = 2276.69*64.9 =147757.181 × 10^{-21} m^2s

h)
Energy consumed by system while computing one convolution:

E = Total Dynamic Power*Delay= 1123.3*64.9= 72.902pJ (p=10^{-12})

i)

operation count = 72

Energy consumed per arithmetic operation=E/operation count

=72.902/72=1.012 pJ (p=10^{-12})

Part2:

1) While designing the system for part 1, I parameterized my code so I can use that for part2.So,I used M and N values for state transition in my FSM for part 1,that made it easy for me to implement it for part 2.I just needed to change the parameter values.
parameter M = 112 // This is size of x
parameter N = 49 // This is size of f

2)To design a system for larger M and N ,I needed to parameterize the FSM and Memory sizes. which I already did in my design. So ,my design can be easily modified for larger values of M and N. As values of M and N increases the size of register that holds the values of addr_x ,addr_f, counterx,counterf, countery will increase that will result in more cell area.The larger M and N will increase the system delay and dynamic power.
The practical limits would be the amount of cells in FPGA ,delay,power and area.
1 convolution will take very long time because of size of vector. we hav to calculate N multiplication for each value in y that would add large amount of delay.

3)
e)
Max Frequency : 909.09 MHz
Critical Path:
Startpoint: f/data_out_reg[3]
(rising edge-triggered flip-flop clocked by clk)
Endpoint: m2/y_data_reg[16]
(rising edge-triggered flip-flop clocked by clk)

Power:

	Internal Power		Switching Power		Leakage Power
Total =	(9.8321e+03 uW	+	111.9281 uW	+	2.3124e+05 nW)=1.0175e+04uW

Total Dynamic Power = 9.9440 mW
Area: Total cell area= 13229.24 μm^2

The critical shown is correct in my design the critical path will from x memory output(data_x) or f memory output (data_f) to output of mac i.e y_data because that path has a multiplier and an adder.

f)
Clock cycles the system takes to load one set of inputs: 112 cycles(Depends on size of bigger vector)

Clock cycles the system takes to compute y(i.e. multiply and accumulate): 3136 cycles .

Clock cycles to clear the accumulator: 63 clock cycles(That's L-1),

Total Clock Cycles From the first cycle of loading the input until the last cycle of

Outputting the result= 3311 clock cycles
PERIOD=1.1 ns
Minimum delay of the system = 3311*1.1 = 3642.1 ns

g)
Area-delay product =13229.24 *3642.1 =48182215 × 10^{-21} m^2s

h)
Energy consumed by system while computing one convolution:

E = Total Dynamic Power*Delay= 9944 *3642.1 =36217.042 pJ (p=10^{-12})

i) **L=112-49+1=64**
N=49
operation count = L*N + L*(N-1) =6208
Energy consumed per arithmetic operation=E/operation count
=36217.042 /6208
=5.834 pJ (p=10^{-12})

4)We changed the accumulator from 23 bits to 26 bits because vectors sizes are changed.
Is this sufficient to guarantee the system cannot overflow?
=> Yes, because
max value a 9-bit reg(x_data and y_data) =(+ or -) 511
number of additions for each value of y=N-1=48
max accumulated value (for single y value) =511*511*48= (+ or -)12553808
(+ or -)12553808 < (+ or -) 33554431 (max value 26 bit accumulator can hold)
And we are clearing the accumulator after we calculated new value of y.

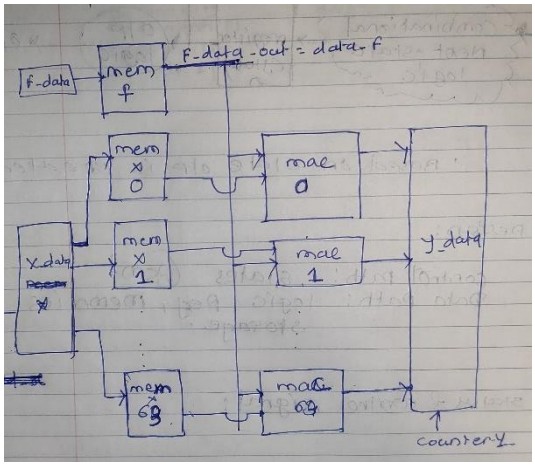
Part3:

a)
To boost the speed I tried to make my system more parallel by adding multiple MAC and X Memory units.
I used the multiple memory units to store values of vector x that needed for calculating y values.

For example:
y[0] = x[0]*f[0] + x[1]*f[1] + x[2]*f[2] = 10*5–20*4+30*3 = 60
y[1] = x[1]*f[0] + x[2]*f[1] + x[3]*f[2] = –20*5+30*4+40*3 = 140
y[2] = x[2]*f[0] + x[3]*f[1] + x[4]*f[2] = 30*5+40*4-50*3 = 160
y[3] = x[3]*f[0] + x[4]*f[1] + x[5]*f[2] = 40*5–50*4+60*3 = 180

I used memory module x_0 to store = x[0], x[1] , x[2] .
And MAC module m2_0 to calculate y[0]
So number of memory module x and MAC module m2 = M-N+1

Rough block diagram of my design:



This way I managed to reduce number of clock cycles required to get an output significantly. So even though the period is somewhat same as part 2 but this system has way better throughput.

b)

i)

Max Frequency : 813 MHz

Period:1.23 ns

Critical Path:

Startpoint: f/data_out_reg[9]

Endpoint: genblk1[35].m2/y_data_reg[25]

The critical shown is correct in my design the critical path will from x memory output(data_x) or f memory output (data_f) to output of one of the mac(m2) i.e y_data because that path has a multiplier and an adder.

Power:

		Internal Power	Switching Power	Leakage Power
Total	=	(1.8456e+05 uW	+3.6074e+03 uW	+ 5.2355e+06 nW)= 1.9340e+05 uW

Total Dynamic Power = 188.1771 mW

Area:

Total cell area= 294938.4 μm^2

ii)

Clock cycles the system takes to load one set of inputs: 112 cycles(Depends on size of bigger vector)

Clock cycles the system takes to compute y(i.e. multiply and accumulate): 49 cycles .

Clock cycles to cycle through all outputs : 63 clock cycles(That's L-1).

Total Clock Cycles From the first cycle of loading the input until the last cycle of

Outputting the result= 224 clock cycles

Minimum delay of the system = 224*1.23 = 225.23 ns

ii)

Area-delay product == 294938.4 * 225.23 =66428975.8 × 10^{-21} m^2s

iv)

Energy consumed by system while computing one convolution:

E = Total Dynamic Power*Delay= 188177.1 *225.23 =42383128.2 pJ (p=10^{-12})

v) L=112-49+1=64
N=49
operation count = L*N + L*(N-1) =6208

Energy consumed per arithmetic operation=E/operation count
=42383128.2 /6208
=6827.17 pJ (p=10^{-12})

c)

Comparison between part 2 and part3:

	PART 2	PART3	RESULT
MIN. DELAY	**3642.1 ns**	**225.23 ns**	**Faster**
AREA DELAY PRODUCT	**48182215 × 10^{-21} m^2s**	**66428975.8 × 10^{-21} m^2s**	**Larger**
Energy consumed per arithmetic operation	**5.834 pJ**	**6827.17 pJ**	**Worse**

So in part3 design I optimized the delay of the part2 system.