CprE 381, Computer Organization and Assembly Level Programming

Lab 1 Report

Student Name __Thomas Gaul_________

Submit a typeset pdf version of this on Canvas by the due date. Refer to the highlighted language in the lab document for the context of the following questions.

[Part 1.c] Think of three more cases and record them in your lab report.

1. To get birthday date (19). Initialize weight value set iW to 5 and IldW to one before the positive edge and hold it for at least 1 clock cycle. S_W should be 5. Setting iX to 3 should set s_WxX to 15. After 3 clock cycles oX should be 3. Setting iY to 4 should get oY to be 19 after another clock cycle
2. Edge Case of 0 multiplication time a base number of 5 and base addition case of 30. Set iW to 5 and IldW to 1 before the positive edge and hold it for at least 1 clock cycle. S_W should be 5. Setting iX to 0 should set s_WxX to 0. After 3 clock cycles oX should be 0. Setting iY to 30 should get oY to be 30 after 3 clock cycles from setting.
3. Another case to try is multiplying by one and adding 0. Set iW to 6 and IldW to 1 before the positive edge and hold it for at least 1 clock cycle. S_W should be 6. Setting iX to 1 should set s_WxX to 6 After 3 clock cycles oX should be 1. Setting iY to 0 should get oY to be 6 after 3 clock cycles from setting.

[Part 1.e] For labels 1, 7, 22, and 28, specify where (VHDL file and line number) these values are located – some will be found in more than one place. Also attempt to explain the functionality of each label as it occurs in the code

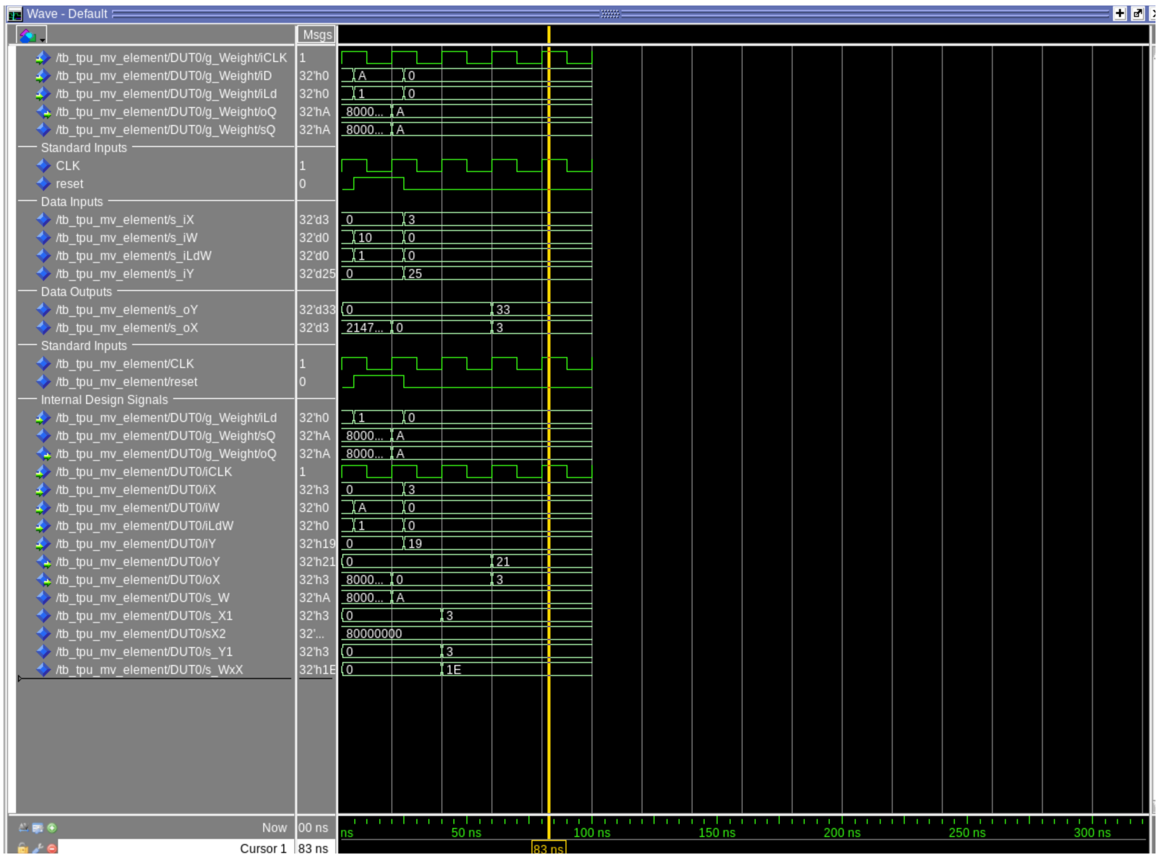
In the attached diagram area 1 is the entire TPU block and is defined in TPU_MV_Element.vhd line 23. Its functionality is the entirety of the element. Where it can multiply two elements iX and iW and add a 3rd iY to the product. I then outputs that number on oY and iX after 3 delays on oX

In the attached diagram area 7 is g_Add1 and is found in Adder.vhd line 26. It adds two integers iA and iB together and updates the output on the positive edge of the clock cycle on oC.

In the attached diagram area 22 is oQ and is found on line 86 of TPU_MV_Element.vhd. It can also be found of line 31 of RegLd.vhd. It becomes S_W going to g_Mult1 It is used as one of the two values to be multiplied together and is also ran as a input to the multiplexer to save the value.

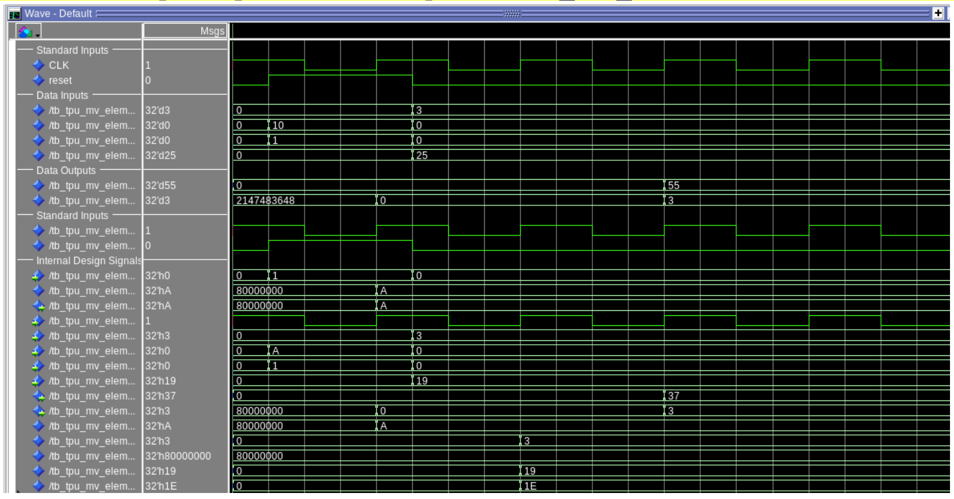
In the attached diagram area 28 is s_X1 the input g_Delay3 is in TPU_MV_Element.vhd on line 114. It comes from oQ on g_Delay1 and iD for g_Dleay3. It is one of the two multiplied input iX delayed by 3 clock cycles.

[Part 1.g.v] In your lab report, include a screenshot of the waveform. Describe, in plain English, any differences between what you expected and what the simulation showed.

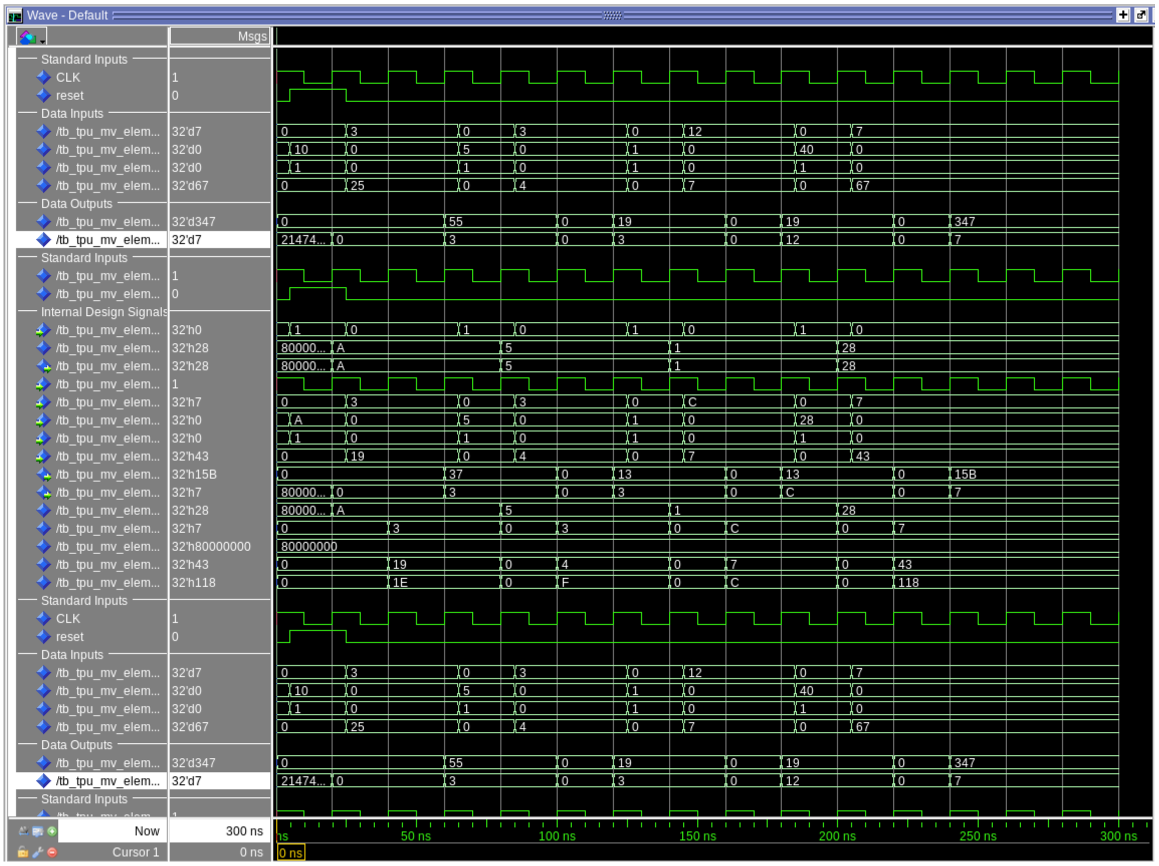

Wave - Default
Msgs
/tb_tpu_mv_element/DUT0/g_Weight/iCLK 1
/tb_tpu_mv_element/DUT0/g_Weight/iD 32'h0
/tb_tpu_mv_element/DUT0/g_Weight/iLd 32'h0
/tb_tpu_mv_element/DUT0/g_Weight/oQ 32'hA
/tb_tpu_mv_element/DUT0/g_Weight/sQ 32'hA
Standard Inputs
CLK 1
reset 0
Data Inputs
/tb_tpu_mv_element/s_iX 32'd3
/tb_tpu_mv_element/s_iW 32'd0
/tb_tpu_mv_element/s_iLdW 32'd0
/tb_tpu_mv_element/s_iY 32'd25
Data Outputs
/tb_tpu_mv_element/s_oY 32'd33
/tb_tpu_mv_element/s_oX 32'd3
Standard Inputs
/tb_tpu_mv_element/CLK 1
/tb_tpu_mv_element/reset 0
Internal Design Signals
/tb_tpu_mv_element/DUT0/g_Weight/iLd 32'h0
/tb_tpu_mv_element/DUT0/g_Weight/sQ 32'hA
/tb_tpu_mv_element/DUT0/g_Weight/oQ 32'hA
/tb_tpu_mv_element/DUT0/iCLK 1
/tb_tpu_mv_element/DUT0/iX 32'h3
/tb_tpu_mv_element/DUT0/iW 32'h0
/tb_tpu_mv_element/DUT0/iLdW 32'h0
/tb_tpu_mv_element/DUT0/iY 32'h19
/tb_tpu_mv_element/DUT0/oY 32'h21
/tb_tpu_mv_element/DUT0/oX 32'h3
/tb_tpu_mv_element/DUT0/s_W 32'hA
/tb_tpu_mv_element/DUT0/s_X1 32'h3
/tb_tpu_mv_element/DUT0/sX2 32'...
/tb_tpu_mv_element/DUT0/s_Y1 32'h3
/tb_tpu_mv_element/DUT0/s_WxX 32'h1E
Now 00 ns
Cursor 1 83 ns
50 ns
100 ns
150 ns
200 ns
250 ns
300 ns
83 ns

I see an issue with the output of oY it outputs 33 as opposed to the expected output of 55. This was caused by the g_Delay2 being hooked up to iX instead of the intended iY.

[Part 1.h] In your lab report, include a screenshot of the waveform. Describe, in plain English, how your waveform matches the expected result (e.g., reference the specific cycles and times). In your submission zip file, provide the completed *TPU_MV_Element.vhd* file in a folder called 'MAC'.



The waveform matches the expected results for the test case because it correctly multiples the two inputs to the multiplier iX and iW to get 30 seen in s_WxX and then correctly adds the iY input of 25 to it to get 55 on oY. Additionally oX is correct matching iX with a delay.



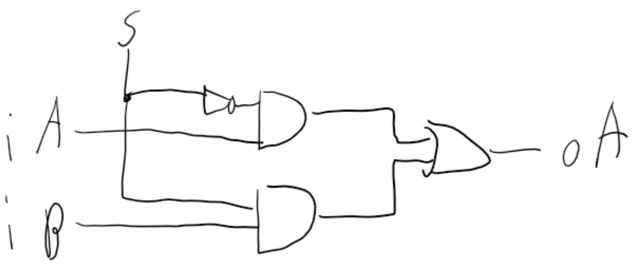
```
Wave - Default
                              Msgs
Standard Inputs
  CLK                         1
  reset                       0
Data Inputs
  /tb_tpu_mv_elem...          32'd7
  /tb_tpu_mv_elem...          32'd0
  /tb_tpu_mv_elem...          32'd0
  /tb_tpu_mv_elem...          32'd67
Data Outputs
  /tb_tpu_mv_elem...          32'd347
  /tb_tpu_mv_elem...          32'd7
Standard Inputs
  /tb_tpu_mv_elem...          1
  /tb_tpu_mv_elem...          0
Internal Design Signals
  /tb_tpu_mv_elem...          32'h0
  /tb_tpu_mv_elem...          32'h28
  /tb_tpu_mv_elem...          32'h28
  /tb_tpu_mv_elem...          1
  /tb_tpu_mv_elem...          32'h7
  /tb_tpu_mv_elem...          32'h0
  /tb_tpu_mv_elem...          32'h0
  /tb_tpu_mv_elem...          32'h43
  /tb_tpu_mv_elem...          32'h15B
  /tb_tpu_mv_elem...          32'h7
  /tb_tpu_mv_elem...          32'h28
  /tb_tpu_mv_elem...          32'h7
  /tb_tpu_mv_elem...          32'h80000000
  /tb_tpu_mv_elem...          32'h43
  /tb_tpu_mv_elem...          32'h118
Standard Inputs
  CLK                         1
  reset                       0
Data Inputs
  /tb_tpu_mv_elem...          32'd7
  /tb_tpu_mv_elem...          32'd0
  /tb_tpu_mv_elem...          32'd0
  /tb_tpu_mv_elem...          32'd67
Data Outputs
  /tb_tpu_mv_elem...          32'd347
  /tb_tpu_mv_elem...          32'd7
Standard Inputs
Now                           300 ns
Cursor 1                      0 ns
```

A more complete test is above that shows the supplied test as well as three others demonstrating it working with different inputs. The first input achieving the day of birthday of 19 by doing 5 * 3 + 4. I achieved the month and date of my birthday by doing 1*12+7 for 19 for oY and month being 12 of 0X. Finally I tried larger numbers of 347 = 40*7+67 and o_X output signal to be 7 after two positive edge of clock

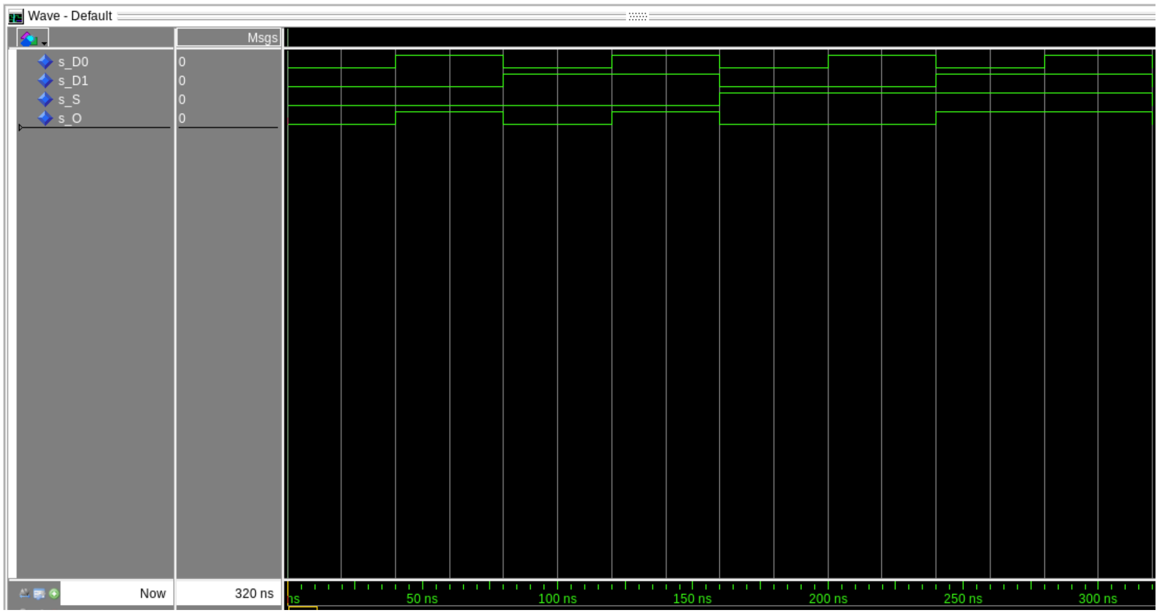
[Part 3.a] Draw the truth table, Boolean equation, and Boolean circuit equivalent (using only two-input gates) that implements a 2:1 mux. Include this in your lab report.

S	iB	iA	oA
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$\bar{S} \cdot iA + S \cdot iB = oA$

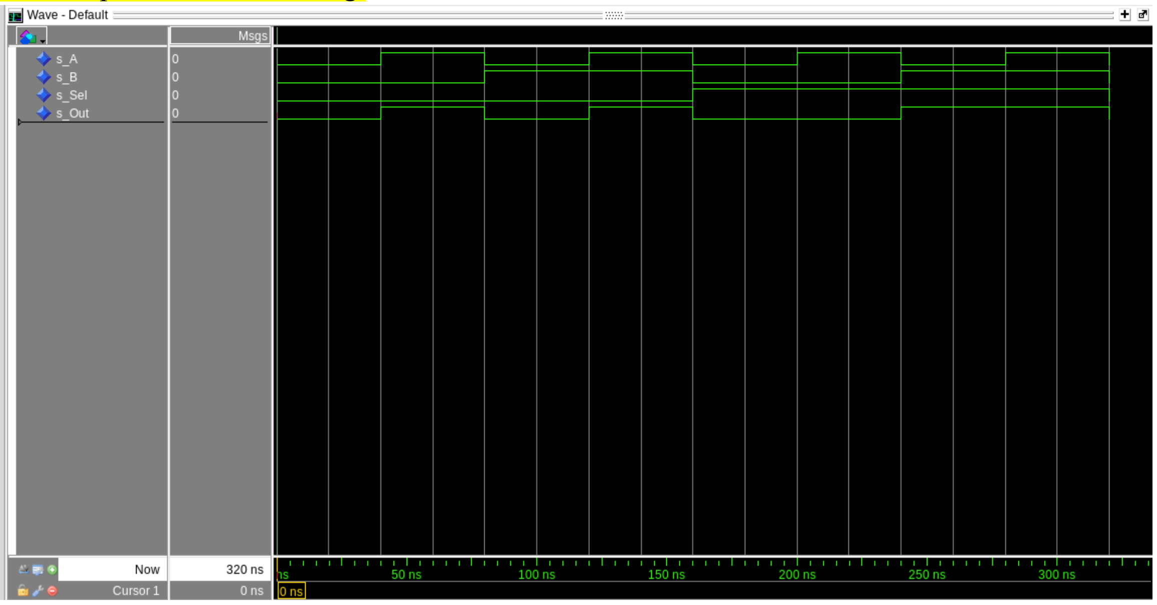


[Part 3.d] In your lab report, include a screenshot of the waveform. Make sure to label the screenshot with which module it is testing.



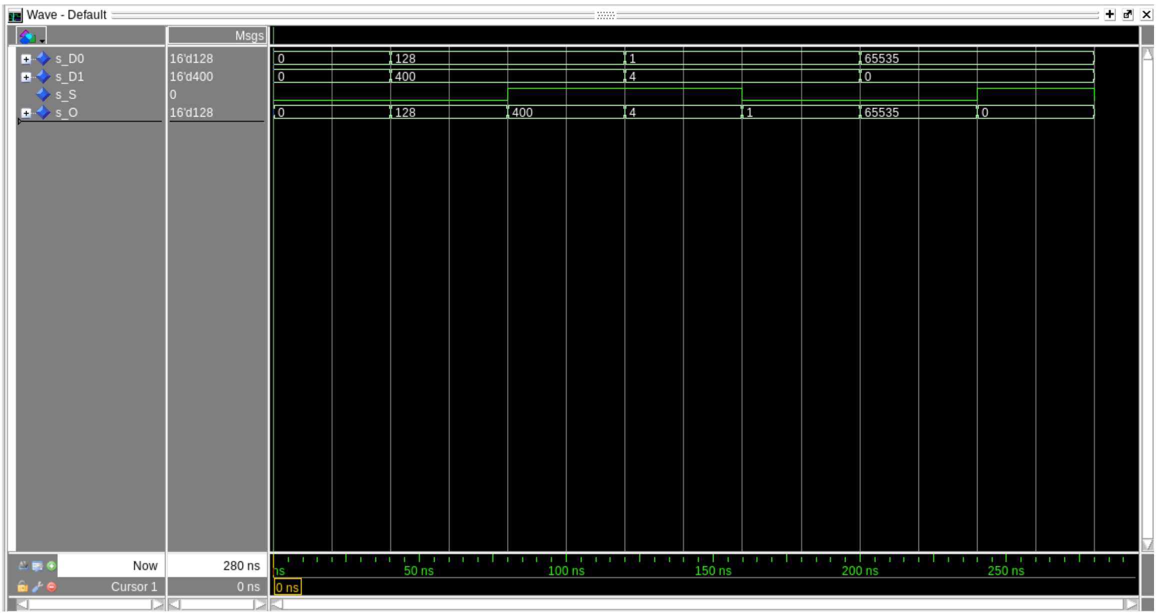
Structural Implementation of a 2 to 1 mux fully enumerated to test

[Part 3.e] Again, in your lab report, include a labeled screenshot of the waveform showing the dataflow mux implementation working.

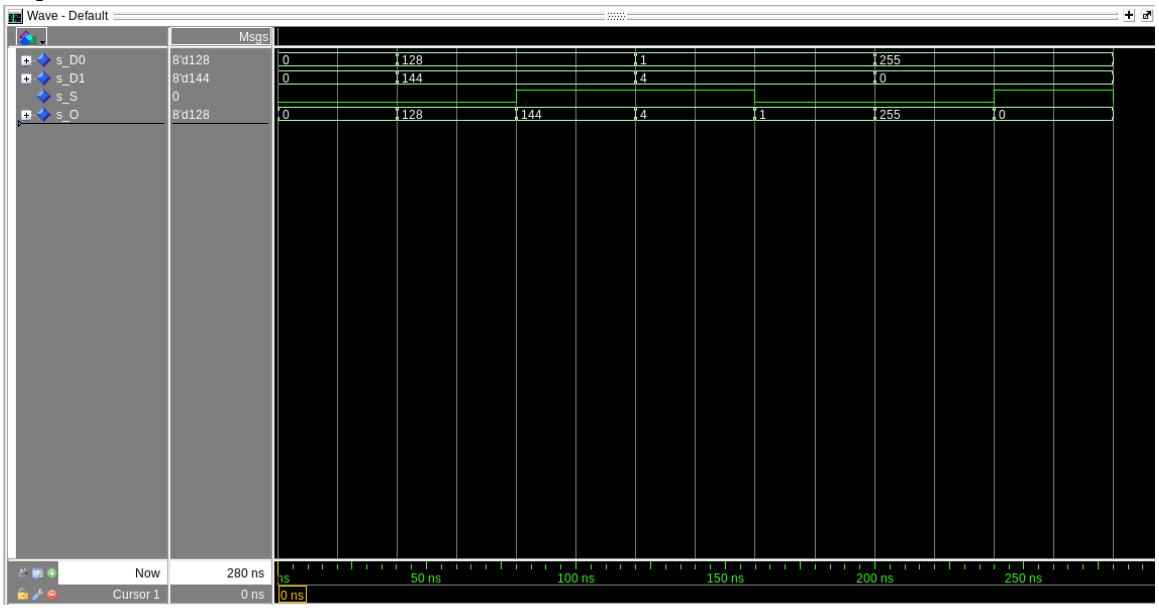


Dataflow implementation of a 2 to 1 mux fully enumerated to test matching the structural mux

[Part 4] Include a waveform screenshot and corresponding description demonstrating it is working correctly.

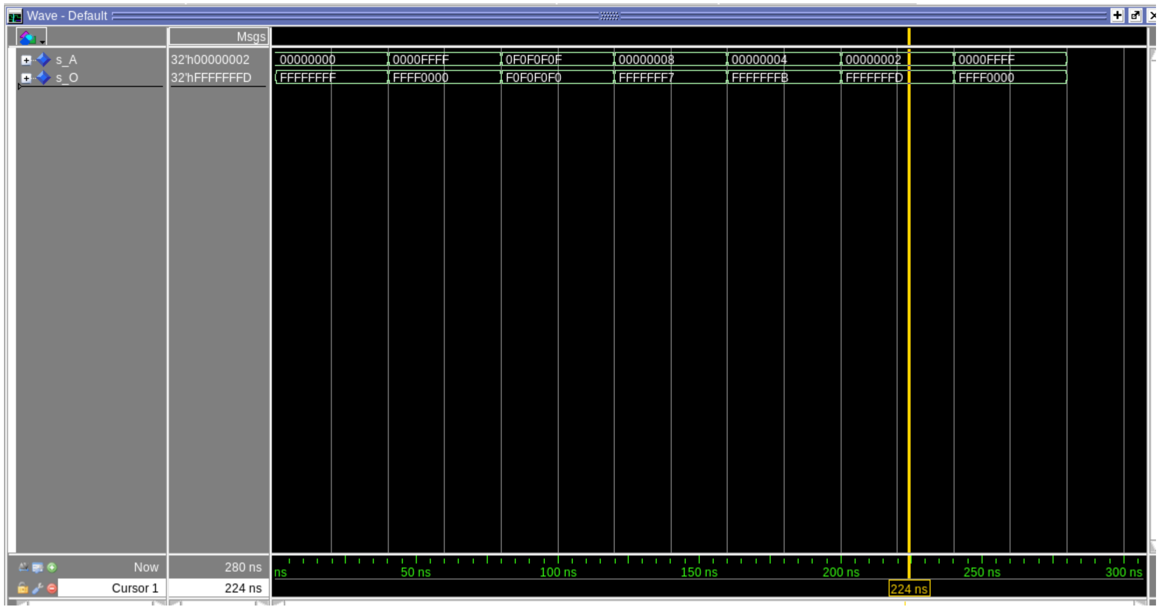


In the test simulations shown above in demonstrates the two to one bus multiplexer work with a 16 bit bus if the select bit is a one then the output bus is D1 and if the select bit is 0 then the output bus is D0. For instance it outputs 128(D0) when the select is 0 but once the select hits 1 the output switches to 400 the value of D1



Above is the exact same test ran but with the generic multiplexer set to 8 bits wide bus instead of 16 bits. This is demonstrated best by the D1 line when it goes to 144 and outputs 144 and D1 when it is at 255 and output matches it. This is due it only showing the first 8 bits cutting off the top 8 bits that were shown above.

[Part 5.b] Include a waveform screenshot and description in your lab report.



```
s_A   32'h00000002   00000000  0000FFFF  0F0F0F0F  00000008  00000004  00000002  0000FFFF
s_O   32'hFFFFFFFD   FFFFFFFF  FFFF0000  F0F0F0F0  FFFFFFF7  FFFFFFFB  FFFFFFFD  FFFF0000

Now        280 ns
Cursor 1   224 ns
```

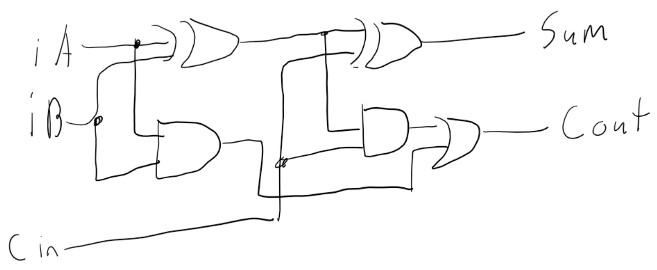
This is the inverter circuit. It inverts a input bus of Generic size. The particular example above is 32 bits but it can be change with one variable. I chose values to test that would be easy to see as inverses in hex. Whenever there is a F on an input it is 0 on output and vise versa. The final test has 8 with 7 being the inverse and 4 with B being the inverse and 2 with D being the inverse.

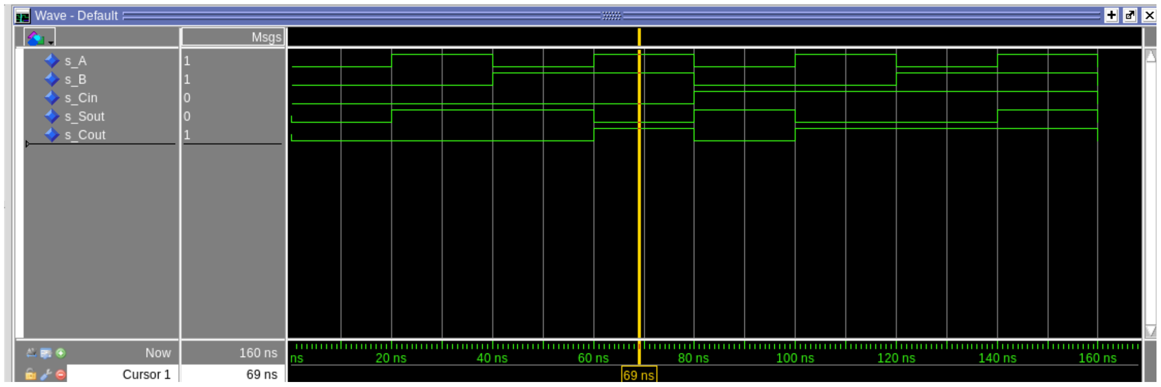
[Part 6.a] A full adder takes three single-bit inputs and produces two single-bit outputs – a sum and carry for the addition of the three input bits. Draw the truth table, Boolean equation, and Boolean circuit equivalent (using only two-input gates) that implements a 1-bit full adder. Include this in your report.

Cin	iB	iA	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$Cin \cdot iA + Cin \cdot iB + iA \cdot iB = Cout$$

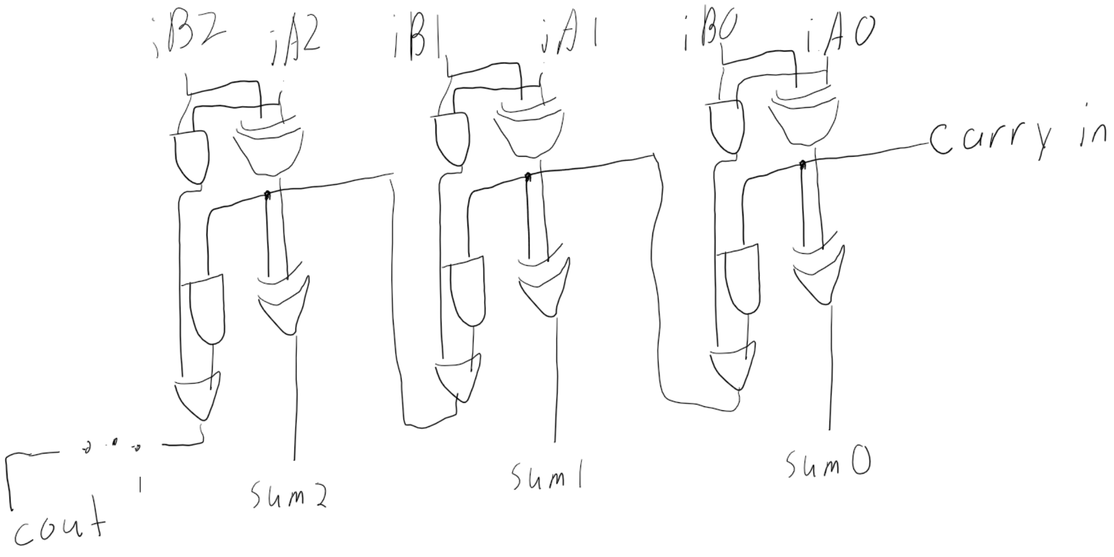
$$\overline{Cin} \cdot \overline{iB} \cdot iA + \overline{Cin} \cdot iB \cdot \overline{iA} + Cin \cdot \overline{iB} \cdot \overline{iA} + Cin \cdot iB \cdot iA = Sum$$





Full adder wave adder fully enumerated to make sure it's working properly.

[Part 6.c] Then draw a schematic of the intended design, including inputs and outputs and at least the 0, 1, N-2, and N-1 stages. Include this in your report.



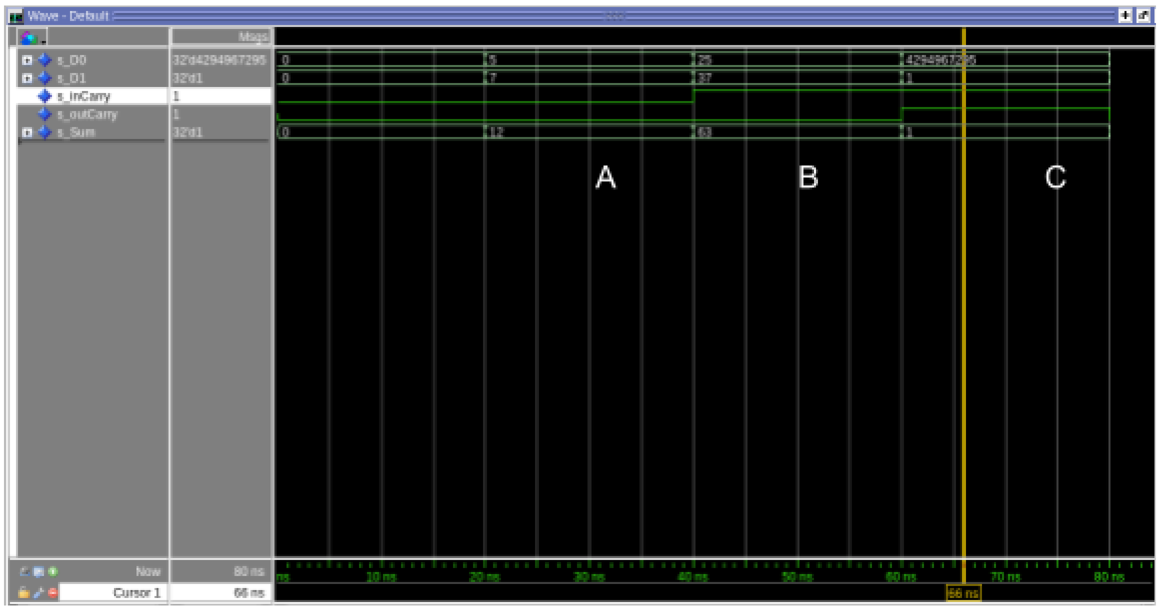
Test Cases:

A: General use test case: Input iA=5, iB= 7, Carry in = 0, Output: Sum = 12, Cout = 0

B: Test Carry In: Input iA=25, iB= 37, Carry in = 0, Output: Sum = 63, Cout = 0

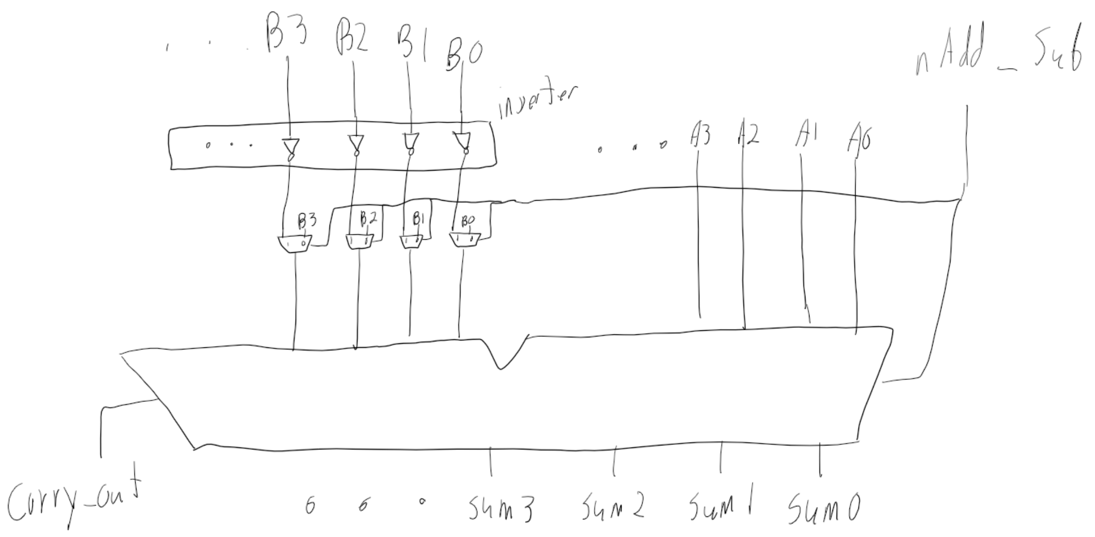
C: Test Carry Out: Input iA=FFFFFFFF, iB= 1, Carry in = 1, Output: Sum = 1, Cout = 1

[Part 6.d] Include an annotated waveform screenshot in your write-up.

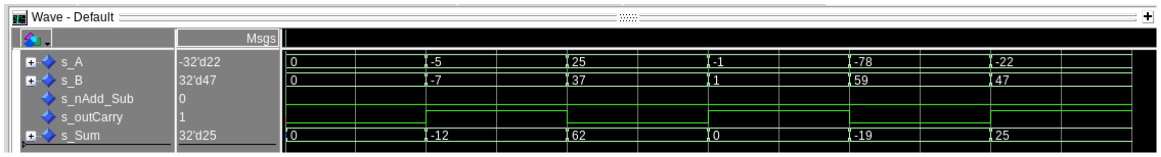


A: General use test case test basic addition: Input iA=5, iB= 7, Carry in = 0, Output: Sum = 12, Cout = 0
B: Test Carry In Test if the carry in bit adds one to the overall output: Input iA=25, iB= 37, Carry in = 0, Output: Sum = 63, Cout = 0
C: Test Carry Out Test if it maxes out the sum bits if it fills the carry out bit: Input iA=FFFFFFFF, iB= 1, Carry in = 1, Output: Sum = 1, Cout = 1

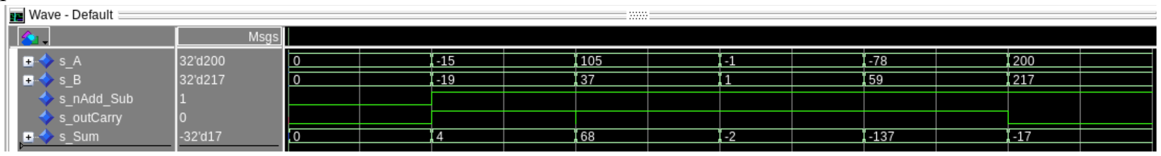
[Part 7.a] Draw a schematic (don't use a schematic capture tool) showing how an N-bit adder/subtractor with control can be implemented using only the three main components designed in earlier parts of this lab (i.e., the N-bit inverter, N-bit 2:1 mux, and N-bit adder). How is the 'nAdd_Sub' bit used? Include this in your report.



[Part 7.c] Provide multiple waveform screenshots in your write-up to confirm that this component is working correctly. What test-cases did you include and why?


Wave - Default
Msgs
s_A -32'd22 0 -5 25 -1 -78 -22
s_B 32'd47 0 -7 37 1 59 47
s_nAdd_Sub 0
s_outCarry 1
s_Sum 32'd25 0 -12 62 0 -19 25

Addition Tests:
I choose to test 0 + 0 = 0 as a base case, -5 + -7 = -12 to test adding two negative numbers, 25 + 37 = 62 for a general use two positive numbers, -1 + 1 = 0 to test the edge case of -1, -78 + 59 = -19 to test adding a positive and negative with a negative result and finally -22 + 47 = 25 to test a positive number with a positive result.


Wave - Default
Msgs
s_A 32'd200 0 -15 105 -1 -78 200
s_B 32'd217 0 -19 37 1 59 217
s_nAdd_Sub 1
s_outCarry 0
s_Sum -32'd17 0 4 68 -2 -137 -17

Subtraction Tests:
Once again I did 0 + 0 = 0 as a base case, -15 - -19 to subtract two negative numbers with a positive result, 105 - 37 = 68 to test two positive number with a positive result, -1-1 to subtract a positive number with a negative number result. -78-59 to do a negative with a positive with a negative result and finally 200-217 = -17 to test two positive numbers with a negative result.

With all my test I tried to get a combination of positive and negative numbers to ensure that the addition logic work properly as well as the inversion logic to handle subtraction.