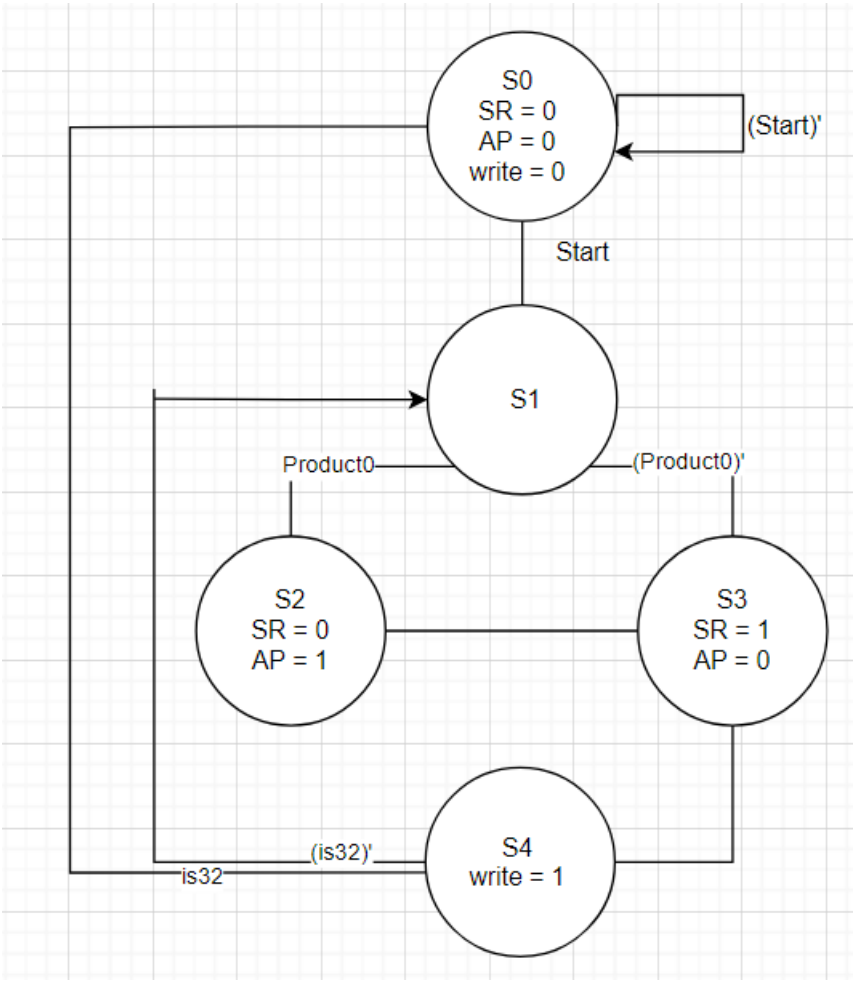
CSE 331/503

Computer Organization

Homework 3 – ALU with Multiplication Design

REPORT

1. STATE MACHINE FOR CONTROL



- ➔ AP = Add Multiplicand to the left half of the product & Place the result in the left half of Product Register
- ➔ Write = Writes
- ➔ SR = Sift the Product Register right 1 bit
- ➔ Is32 = If the number of repetitions is 32 then 1 , otherwise 0
- ➔ Product0 = Least significant bit of Product
- ➔ Start = To start FSM

There is 5 states so I can indicate the states by using 3 bit. Here is architecture of State Machine:

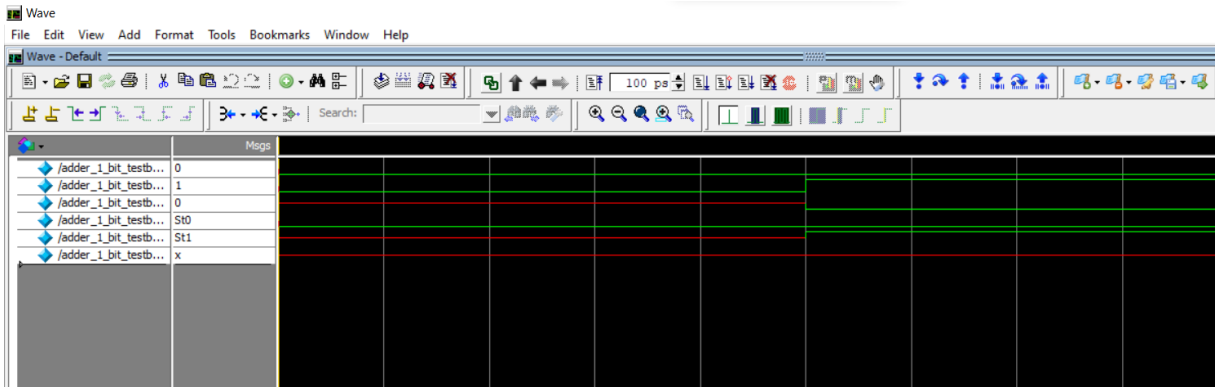
1. S0 -> 000 , S1 -> 001 , S2 -> 010 , S3 -> 011, S4 -> 100 (s2s1s0)
2. 3 bit for next states (n2n1n0)
3. Inputs: Product0 , is32, Start
4. Outputs: AP , SR, write

Present State			Inputs			Next State		
S2	S1	S0	Product0	Is32	Start	N2	N1	N0
0	0	0	-	-	0	0	0	0
0	0	0	-	-	1	0	0	1
0	0	1	0	-	-	0	1	1
0	0	1	1	-	-	0	1	0
0	1	0	-	-	-	0	1	1
0	1	1	-	-	-	1	0	0
1	0	0	-	0	-	0	0	1
1	0	0	-	1	-	0	0	0

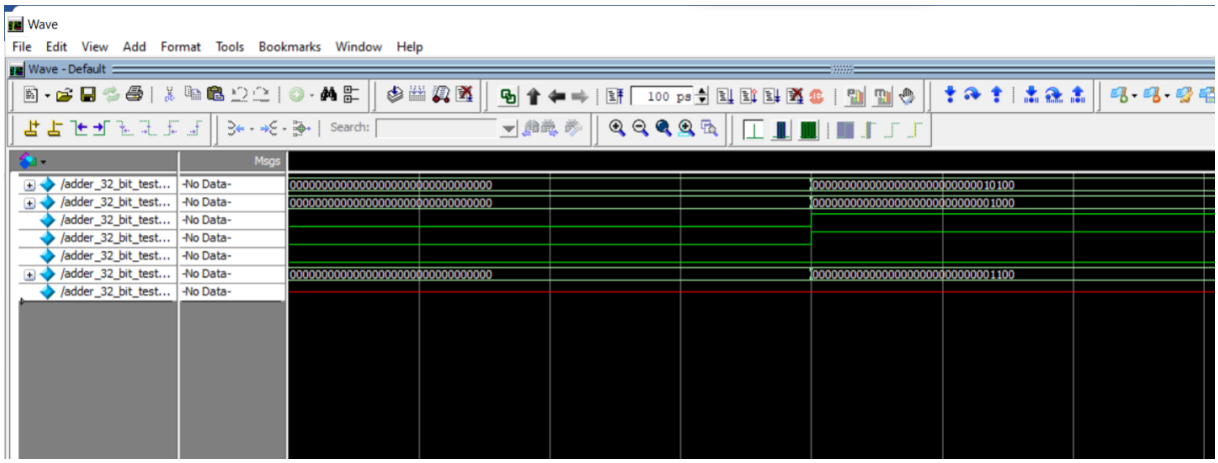
Boolean Expressions:

- N2 = S2’S1S0
- N1 = S2’S1’S0(Product0)’ + S2’S1’S0(Product0) + S2’S1S0’
- N0 = S2’S1’S0’Start + S2’S1’S0(Product0)’ + S2’S1S0’ + S2S1’S0’(Is32)’
- AP = S2
- SR = S3 + S4
- Write = S4

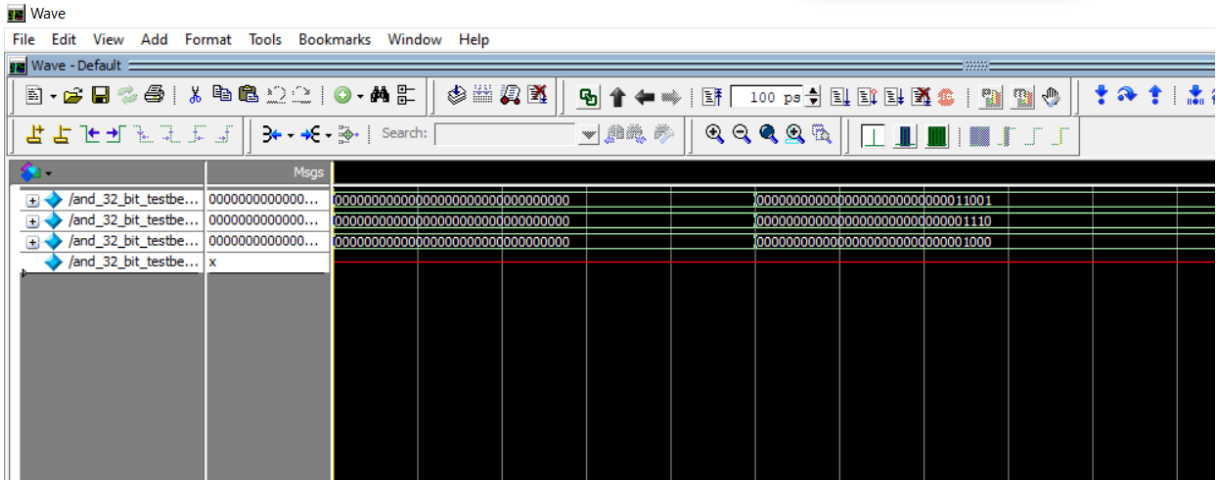
2. ALU32



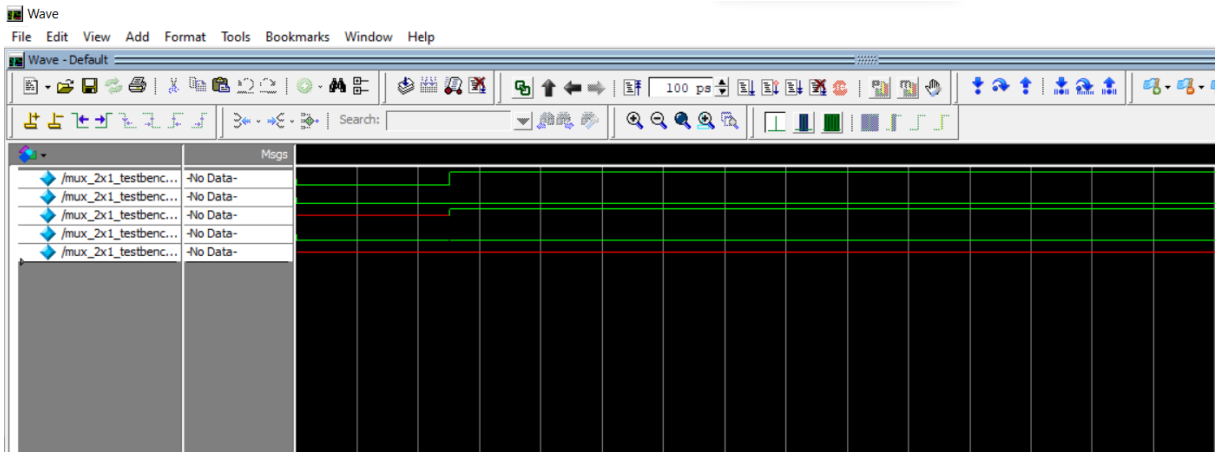
adder_1_bit.v



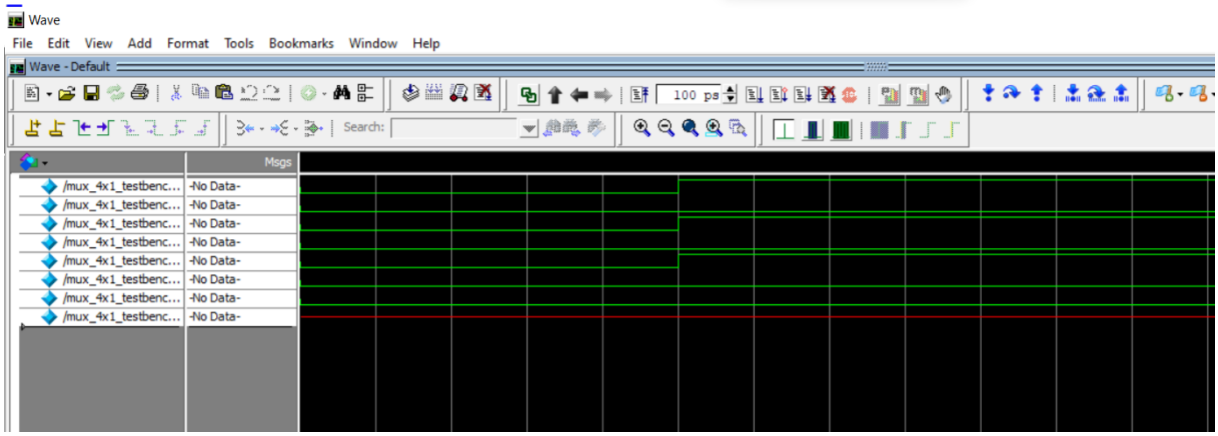
adder_32_bit.v



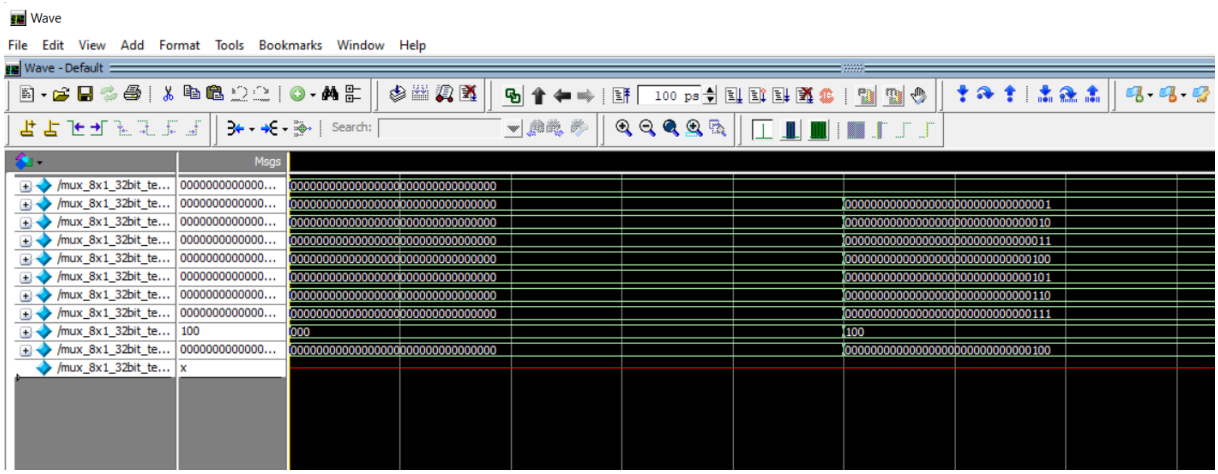
and_32_bit.v



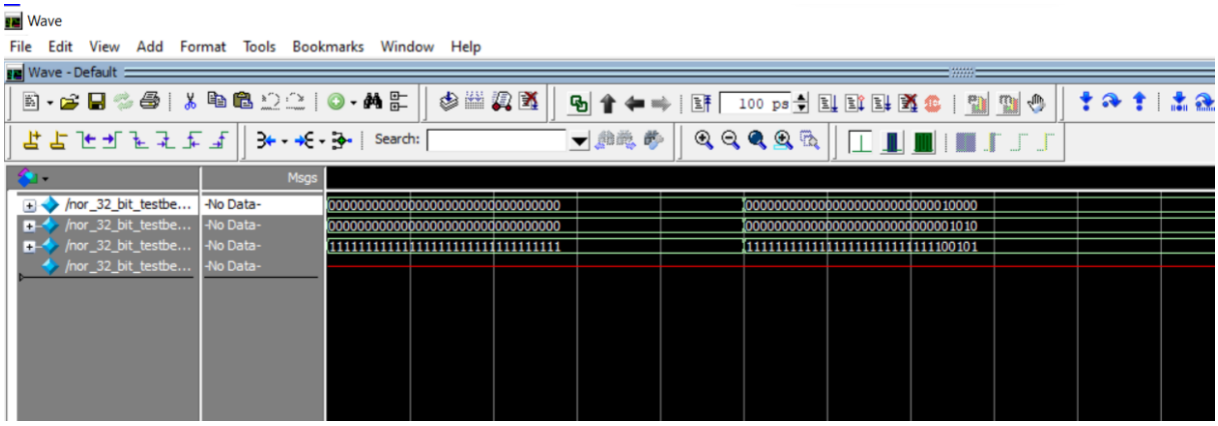
mux_2x1.v



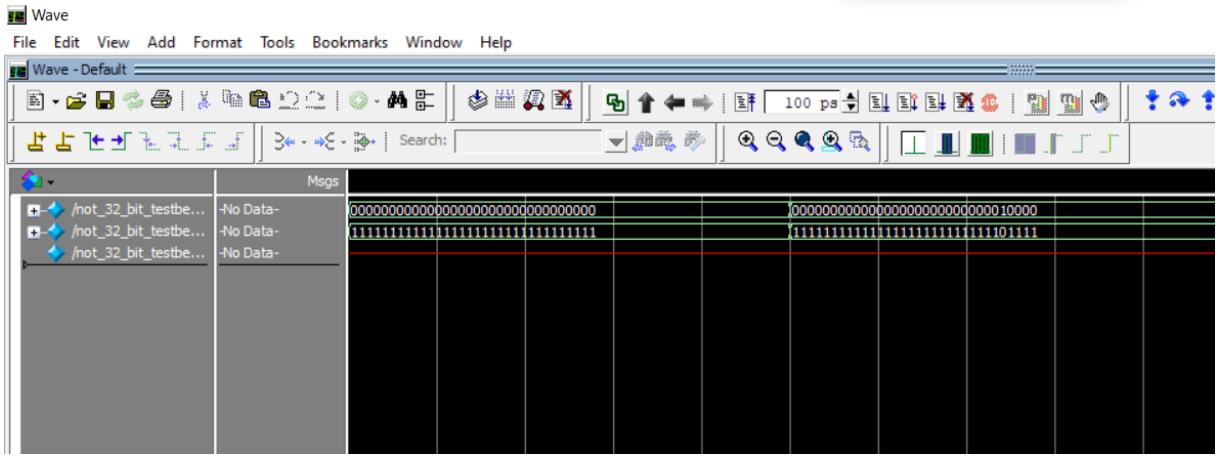
mux_4x1.v



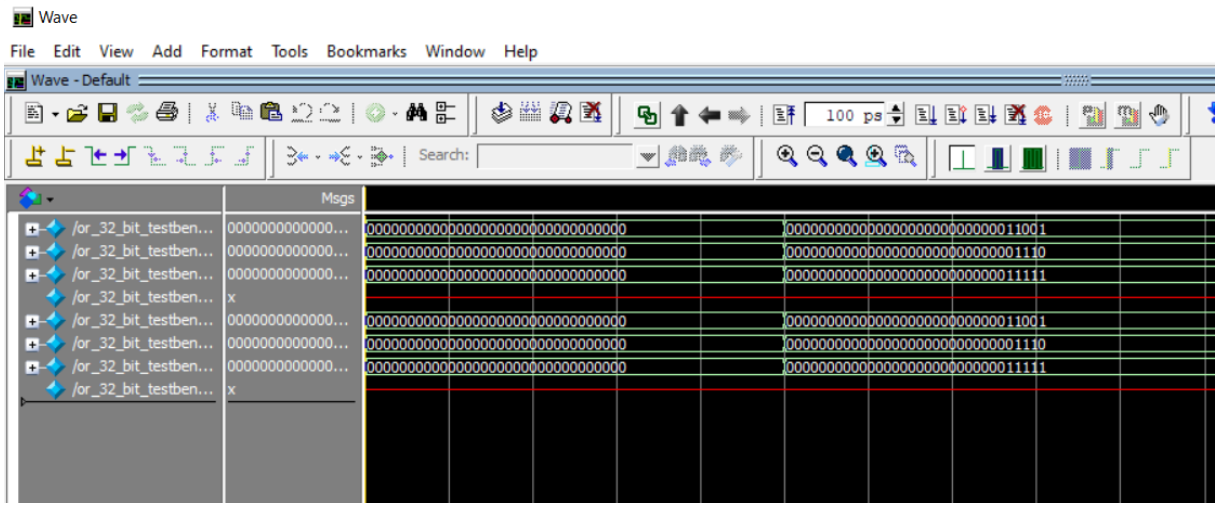
mux_8x1.v



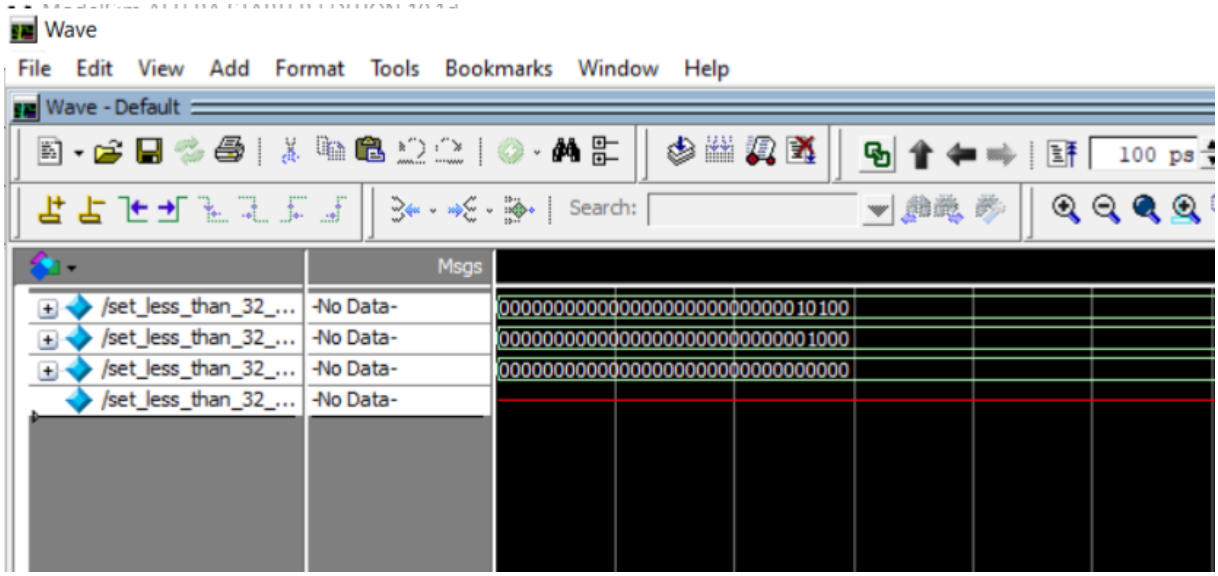
nor_32_bit.v



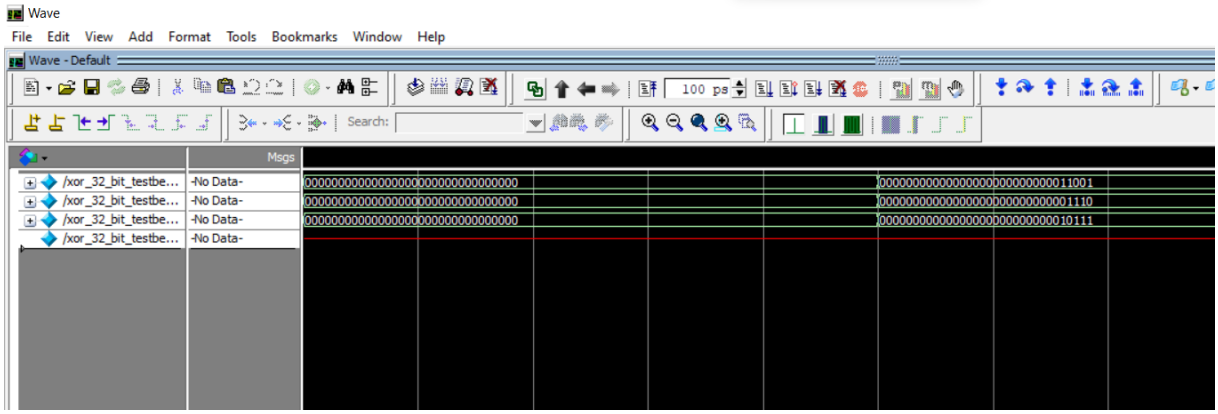
not_32_bit.v



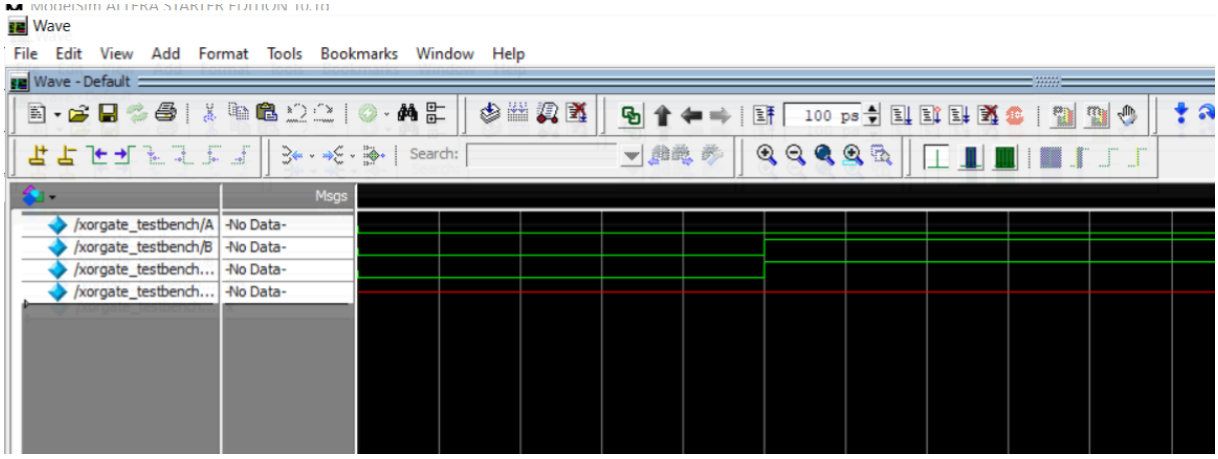
or_32_bit.v



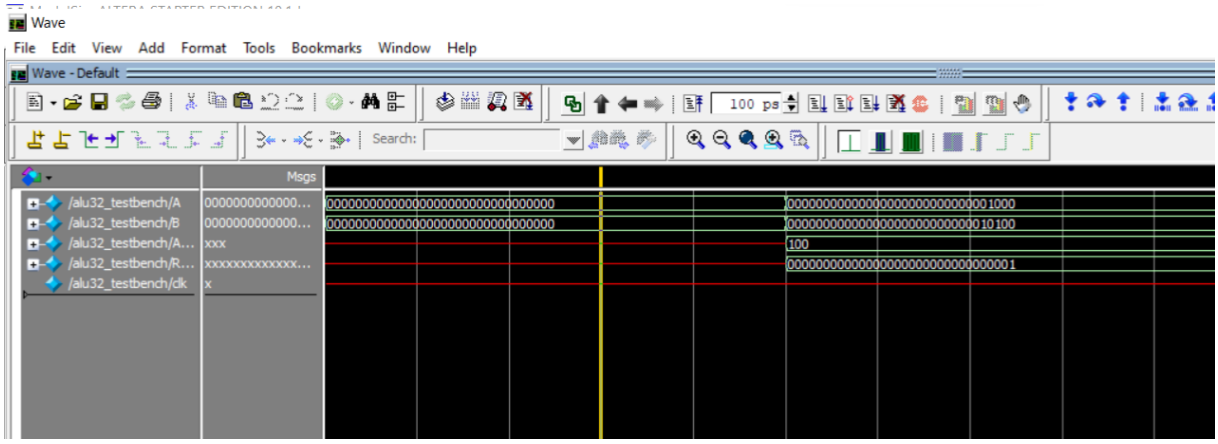
set_less_than_32.v



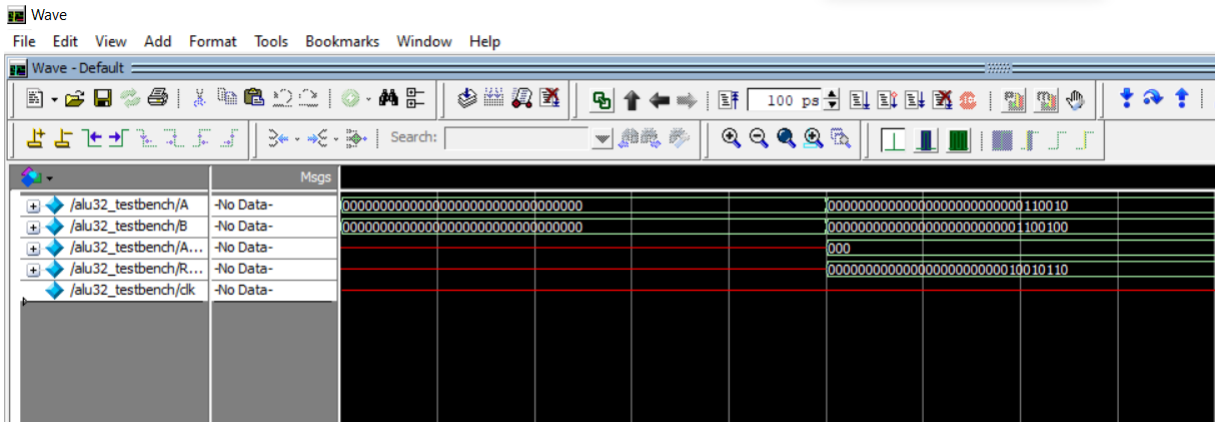
xor_32_bit.v



xorgate.v



alu32.v (A = 8 , B = 20 , SLT Operation)



alu32.v (50 + 100)