Lab 1

Sara Frazer

Oct 1, 2023

CDA3203 Computer Logic Design

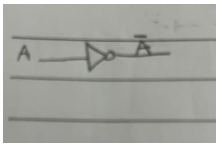
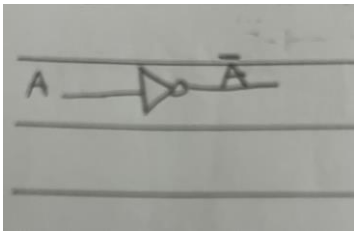
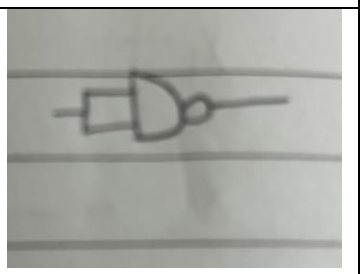
Fall 2023

Dr. Maria Petrie

Florida Atlantic University

Part 1: Design 9 circuits by completing below: Draw the symbol for the gate, its Truth Table, its Simplest Sum of Products Expression, draw its NOT-AND-OR Equivalent Circuit, its all-NAND Equivalent Circuit.

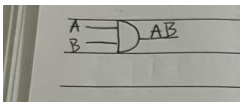
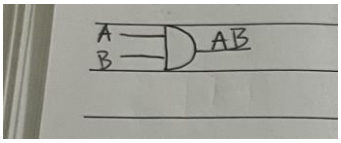
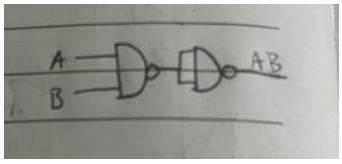
1.1 NOT gate.

Draw NOT gate	Truth Table and Simplest Sum of Products Equation	NOT-AND-OR Equivalent Circuit	all-NAND Equivalent Circuit
			

A	NOT(A)
0	1
1	0

Y1=A'

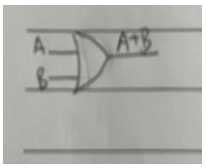
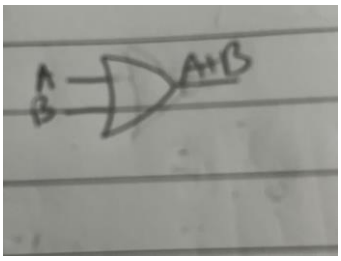
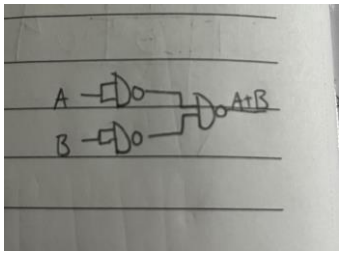
1.2- AND gate

Draw AND gate	Truth Table and Simplest Sum of Products Equation	NOT-AND-OR Equivalent Circuit	all-NAND Equivalent Circuit
			

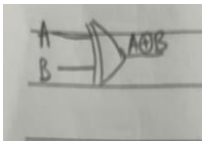
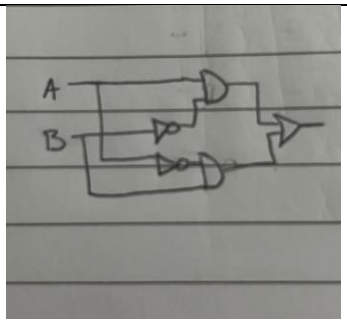
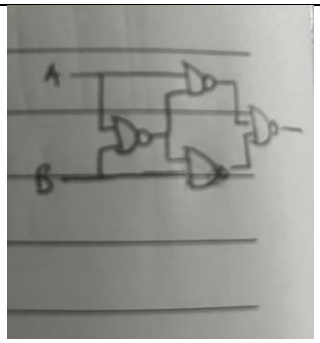
A	B	AND(A,B)
0	**0**	0
0	**1**	0
1	**0**	0
1	**1**	1

Y2=AB

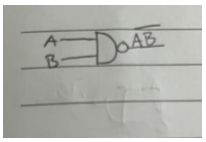
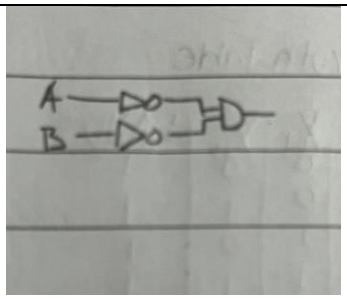
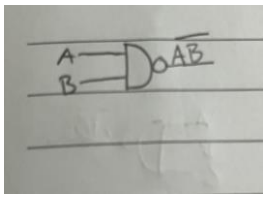
1.3- OR gate

Draw OR gate	Truth Table and Simplest Sum of Products Equation	NOT-AND-OR Equivalent Circuit	all-NAND Equivalent Circuit
	<table><tr><td>A</td><td>B</td><td>OR</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> Y=A+B		

1.4- XOR gate

Draw XOR gate	Truth Table and Simplest Sum of Products Equation	NOT-AND-OR Equivalent Circuit	all-NAND Equivalent Circuit
	<table><tr><th>A</th><th>B</th><th>XOR(A,B)</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table> Y4=AB'+A'B		

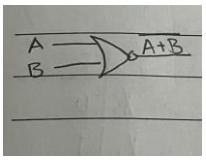
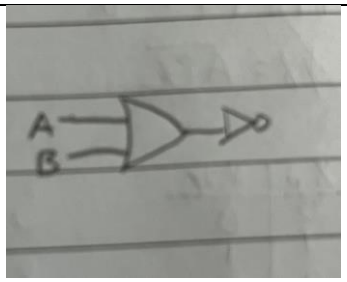
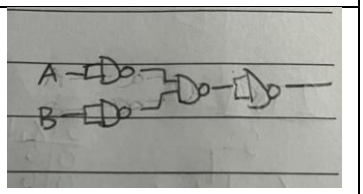
1.5- NAND gate

Draw NAND gate	Truth Table and Simplest Sum of Products Equation	NOT-AND-OR Equivalent Circuit	all-NAND Equivalent Circuit
			

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0

Y5=A'B'

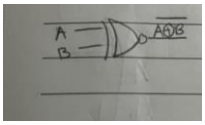
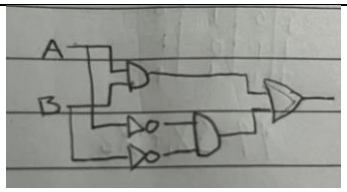
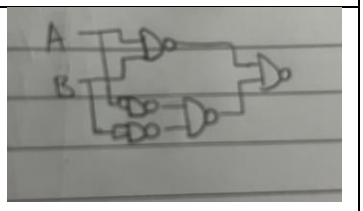
1.6- NOR gate

Draw NOR gate	Truth Table and Simplest Sum of Products Equation	NOT-AND-OR Equivalent Circuit	all-NAND Equivalent Circuit
			

A	B	NOR(A,B)
0	0	1
0	1	0
1	0	0
1	1	0

Y6=A'+B'

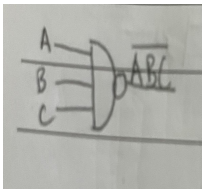
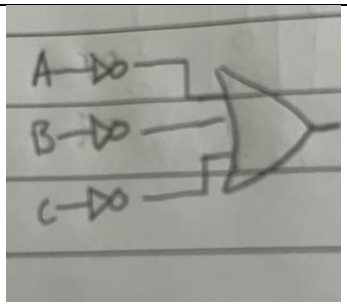
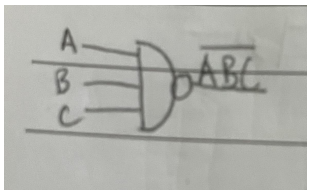
1.7- XNOR gate

Draw XNOR gate	Truth Table and Simplest Sum of Products Equation	NOT-AND-OR Equivalent Circuit	all-NAND Equivalent Circuit
			

A	B	XNOR(A,B)
0	0	1
0	1	0
1	0	0
1	1	1

Y7=(AB)+(A'B')

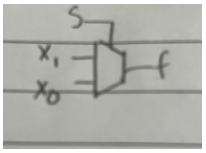
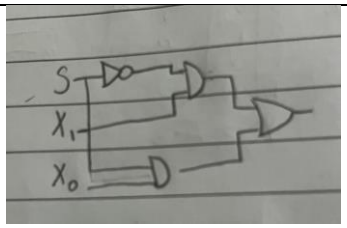
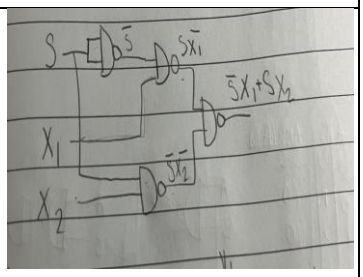
1.8 3-input NAND gate

Draw 3-input NAND gate	Truth Table and Simplest Sum of Products Equation	NOT-AND-OR Equivalent Circuit	all-NAND Equivalent Circuit
			

X2	X1	X0	NAND 3
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

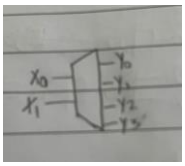
Y8=A'B'C'

1.9- 2 to 1 Encoder or Multiplexer (Mux)

Draw 2to1 Mux	Truth Table and Simplest Sum of Products Equation	NOT-AND-OR Equivalent Circuit	all-NAND Equivalent Circuit
	s x1 x0 Mux 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 1 1 0 0 0 1 0 1 1 1 1 0 0 1 1 1 1 Y8=S'X1+SX0		

1.10 - 2 to 4 decoder or Demultiplexer (DMux)

Draw 2to4 DMux	Truth Table and Simplest Sum of Products Equation	NOT-AND-OR Equivalent Circuit	all-NAND Equivalent Circuit



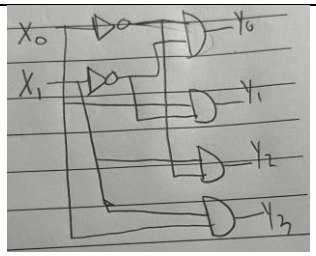
X_1	X_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Y3=X1X0

Y2=X1X0'

Y1=X1'X0

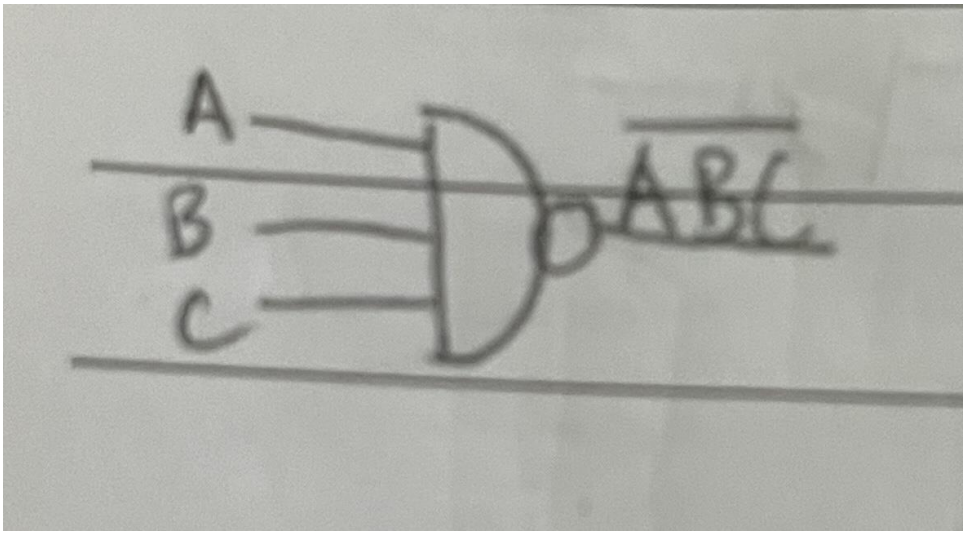
Y0=X1'X0'



3 Input NAND

Project wizard

Diagram

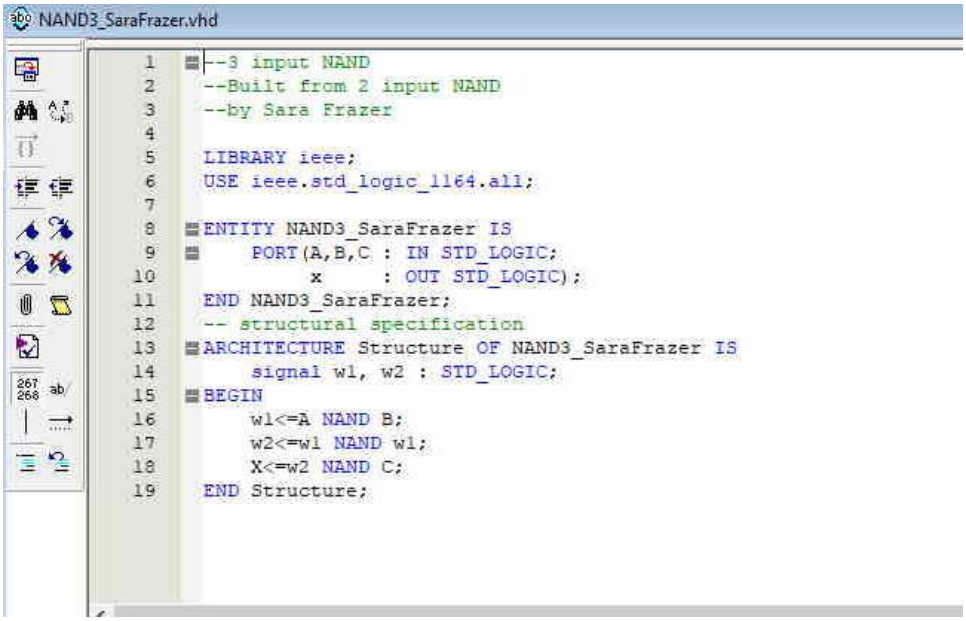


Truth Table

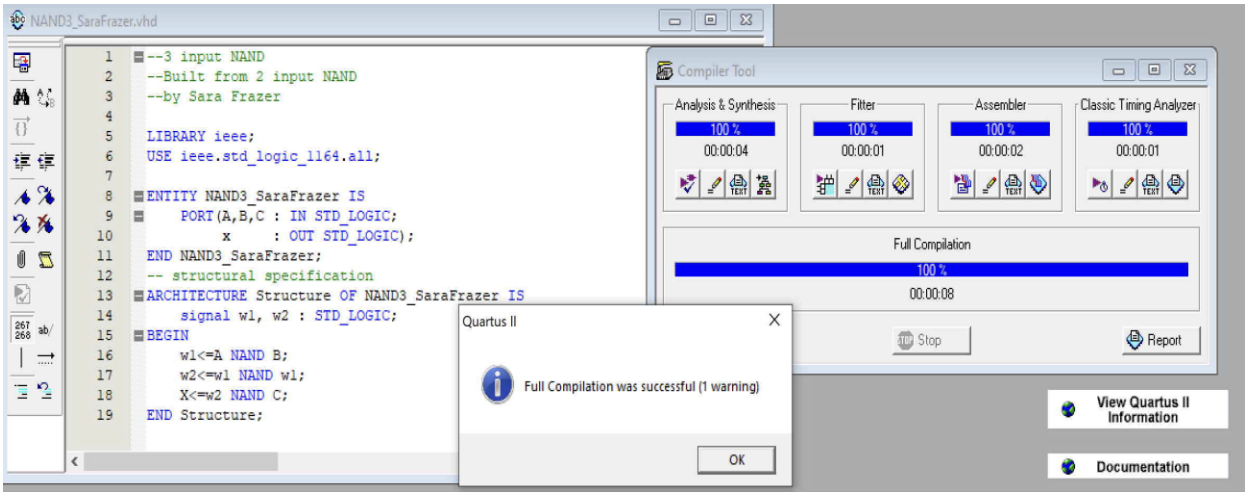
X2	X1	X0	NAND3
0	0	0	1
0	0	1	1
0	1	0	1

0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

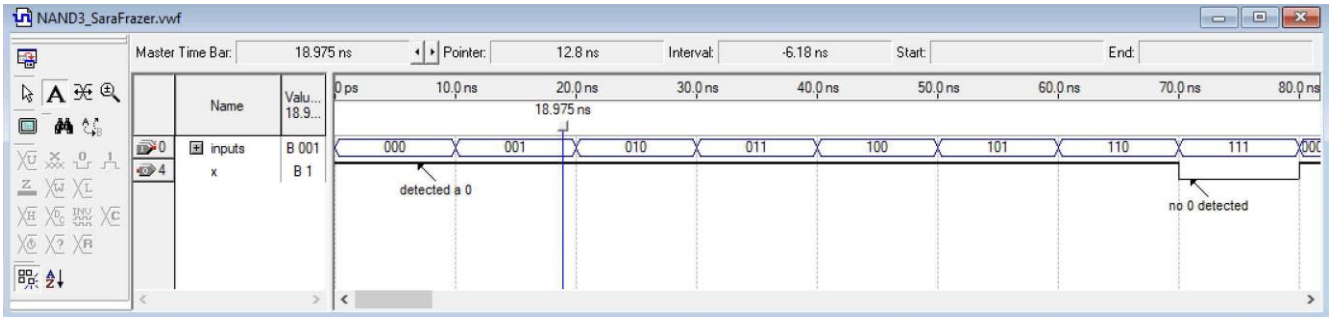
VHDL Code



Success Compilation

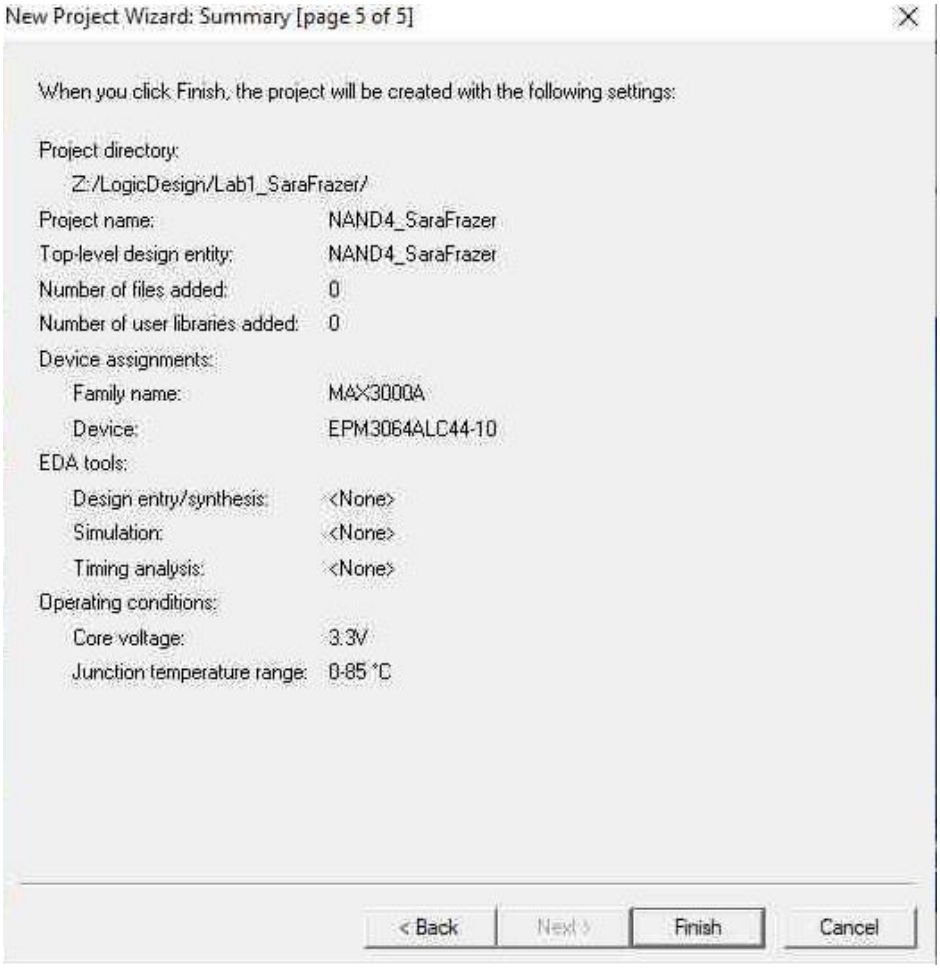


Timing Diagram

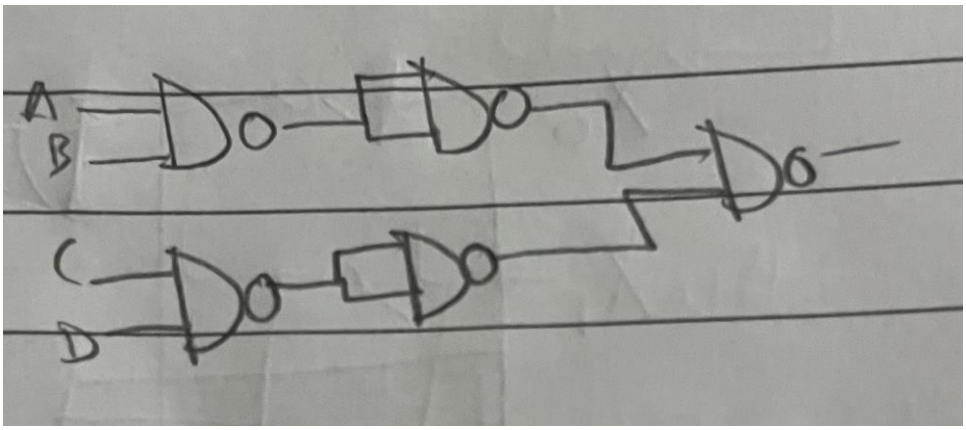


NAND 4

Project Wizard



Diagram



Truth Table

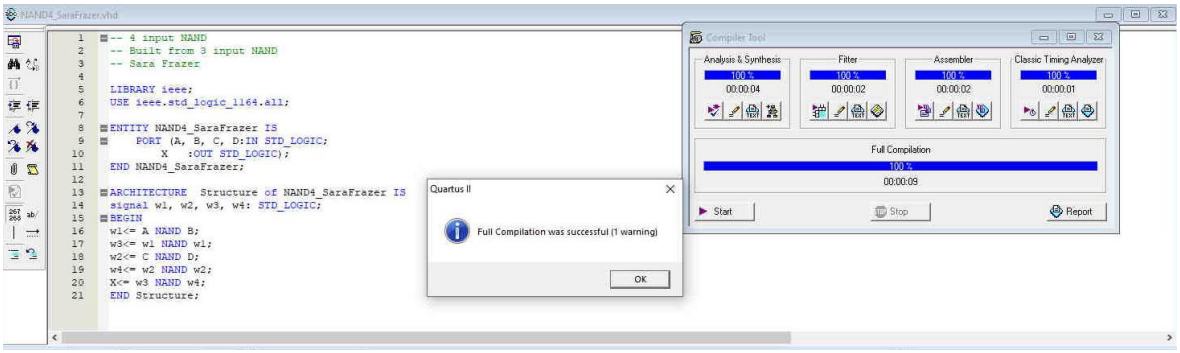
A	B	C	D	A’B’C’D’
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

VHDL Code

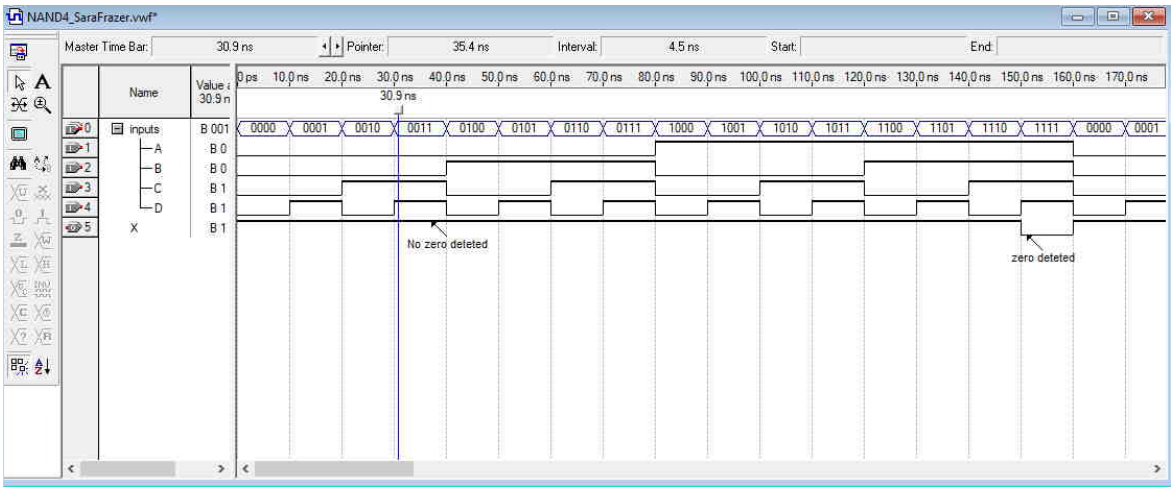
NAND4_SaraFrazer.vhd

```
1   -- 4 input NAND
2   -- Built from 3 input NAND
3   -- Sara Frazer
4
5   LIBRARY ieee;
6   USE ieee.std_logic_1164.all;
7
8   ENTITY NAND4_SaraFrazer IS
9       PORT (A, B, C, D:IN STD_LOGIC;
10            X   :OUT STD_LOGIC);
11  END NAND4_SaraFrazer;
12
13  ARCHITECTURE  Structure of NAND4_SaraFrazer IS
14  signal w1, w2, w3, w4: STD_LOGIC;
15  BEGIN
16  w1<= A NAND B;
17  w2<= w1 NAND w1;
18  w3<= C NAND D;
19  w4<= w2 NAND w2;
20  X<= w3 NAND w4;
21  END Structure;
```

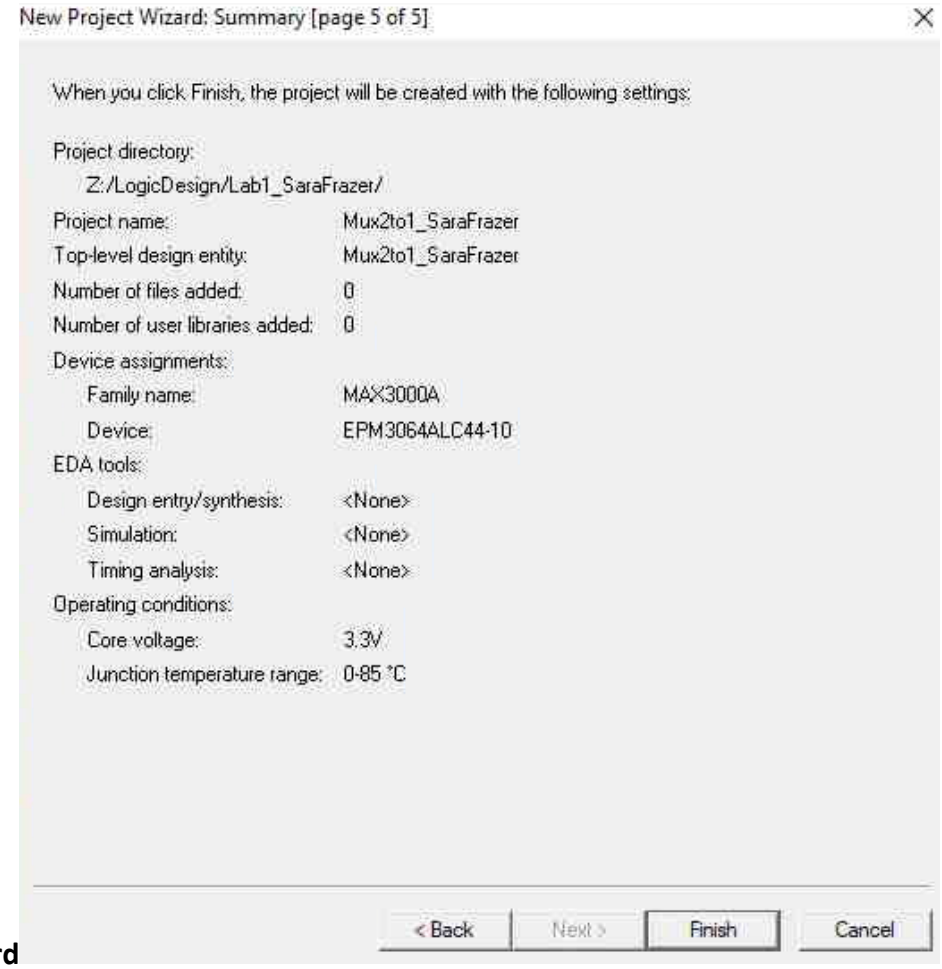
Success Compilation



Timing Diagram

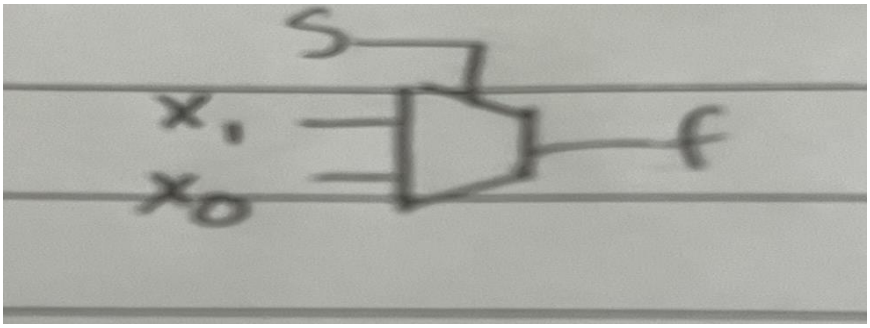

NAND4_SaraFrazer.vwf*
Master Time Bar: 30.9 ns
Pointer: 35.4 ns
Interval: 4.5 ns
Start:
End:
Name
inputs
A
B
C
D
X
30.9 ns
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
0000
0001
No zero deteted
zero deteted

2 to 1 Mux

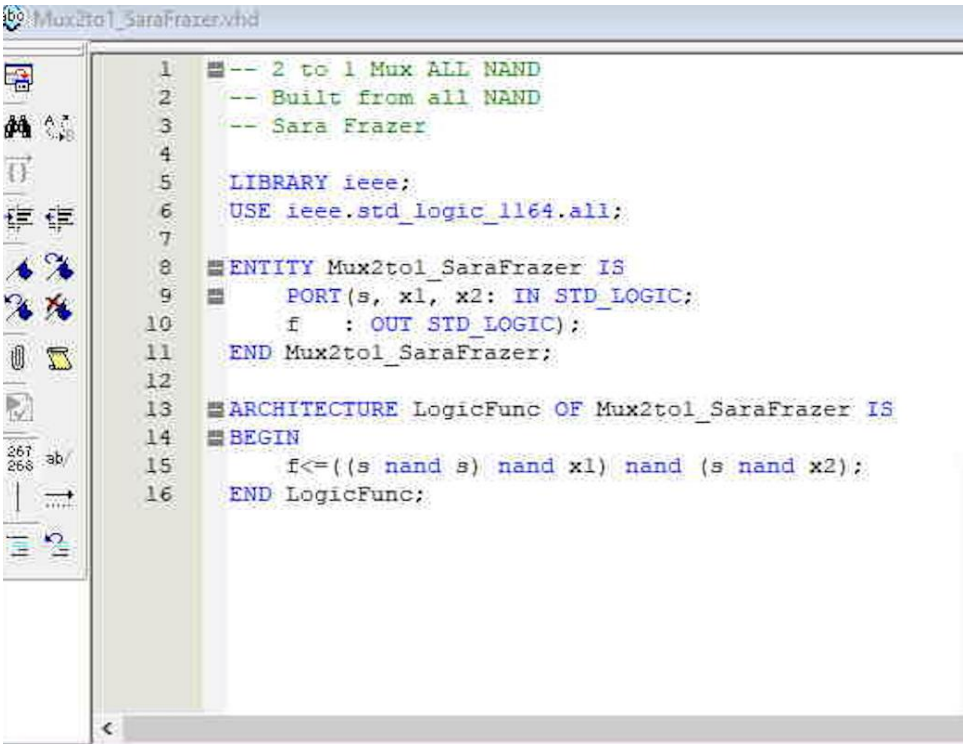


Project Wizard

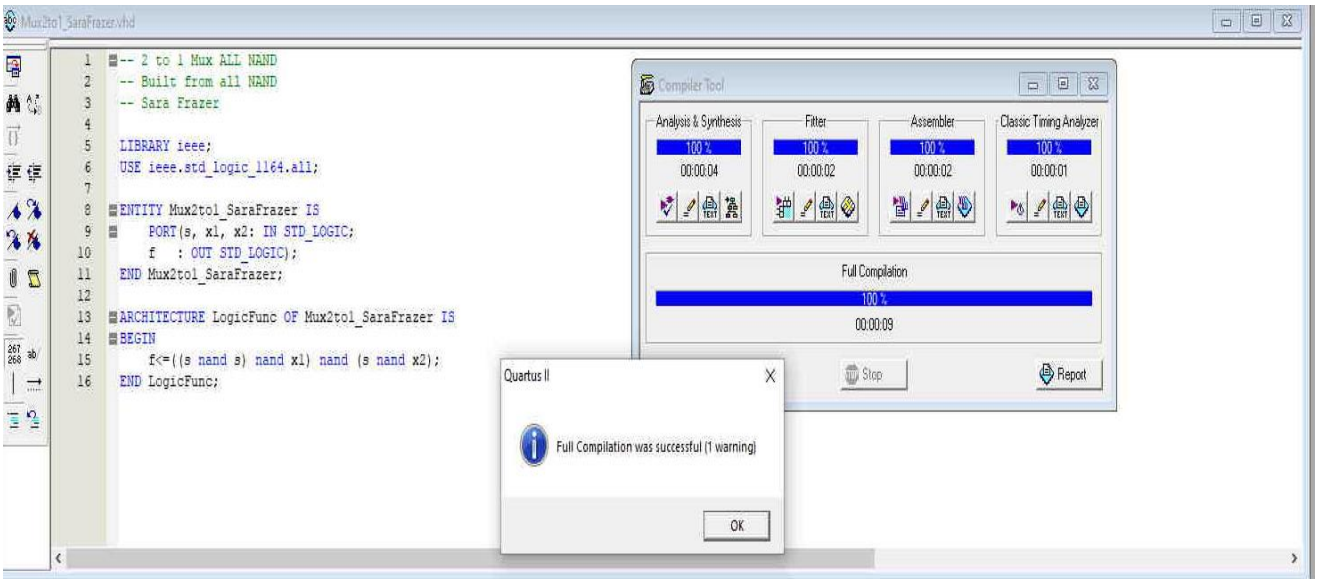
Diagram



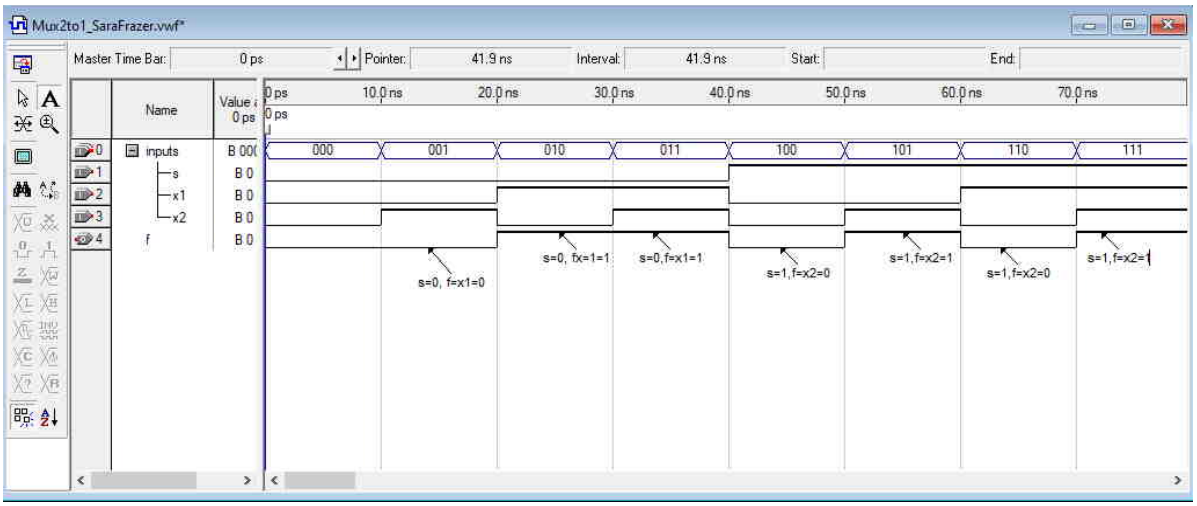
VHDL Code



Success Compilation

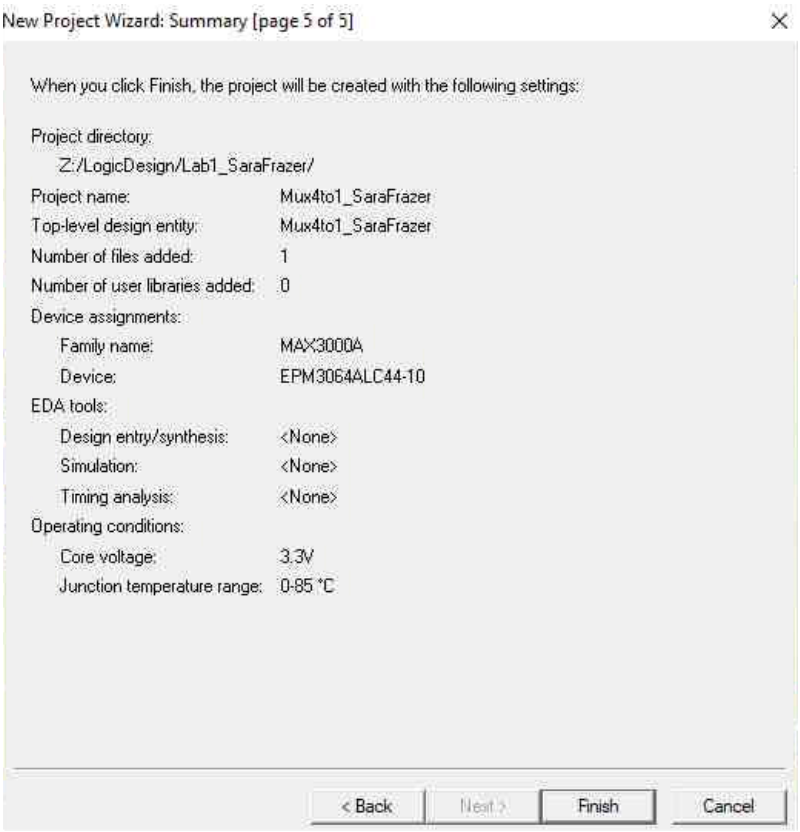


Timing Diagram

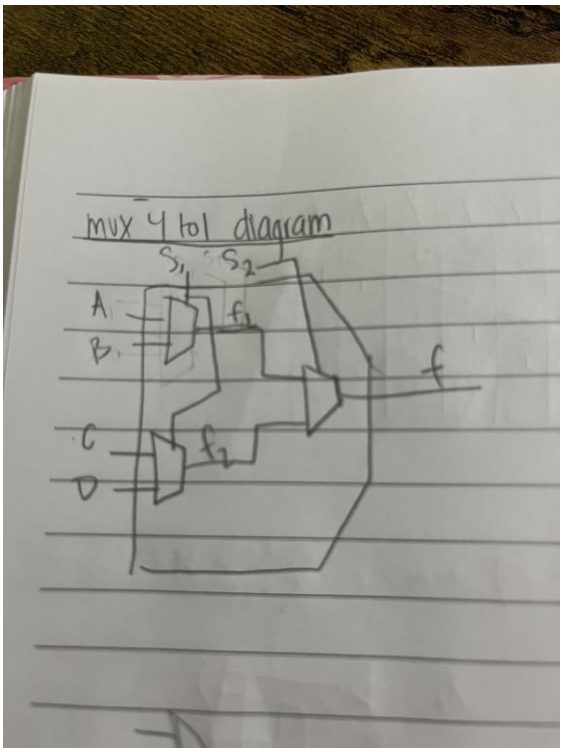


4 to 1 Mux

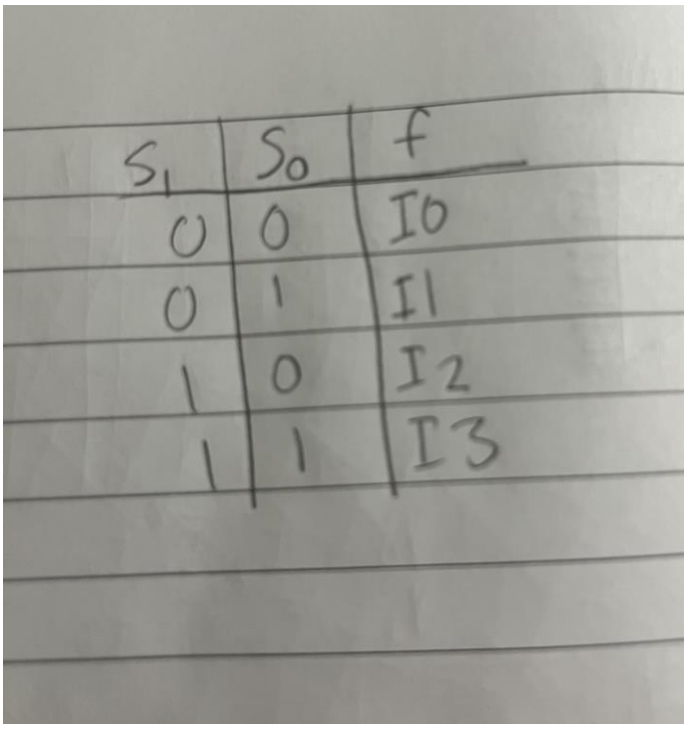
Project Wizard



Diagram



Truth Table



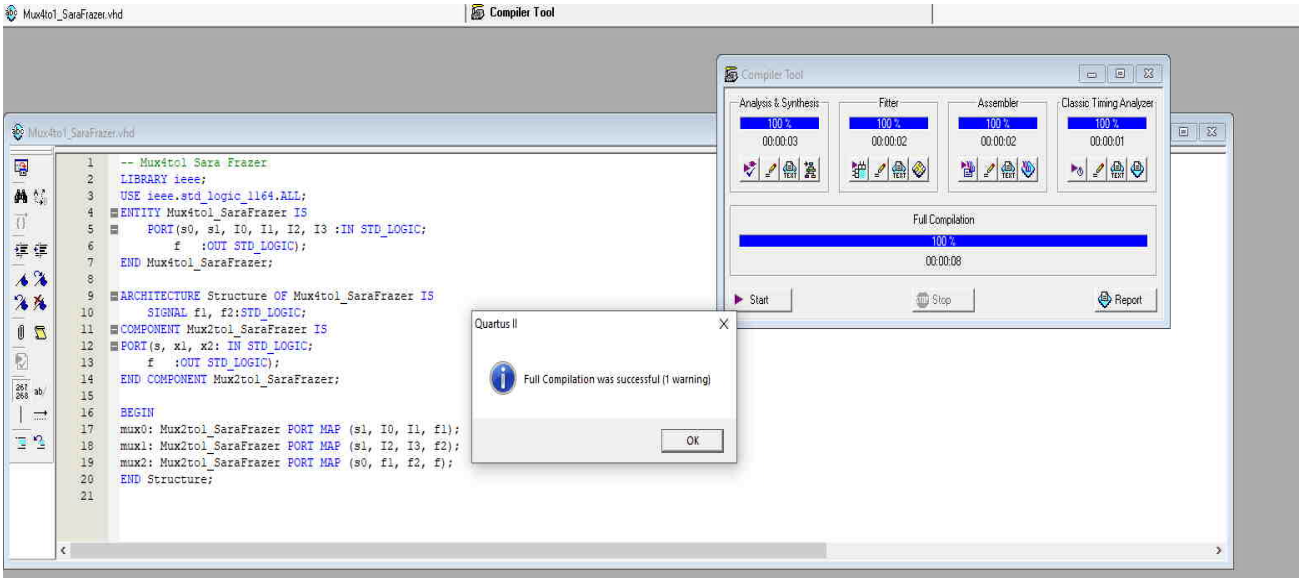
S_1	S_0	f
0	0	I0
0	1	I1
1	0	I2
1	1	I3

VHDL Code

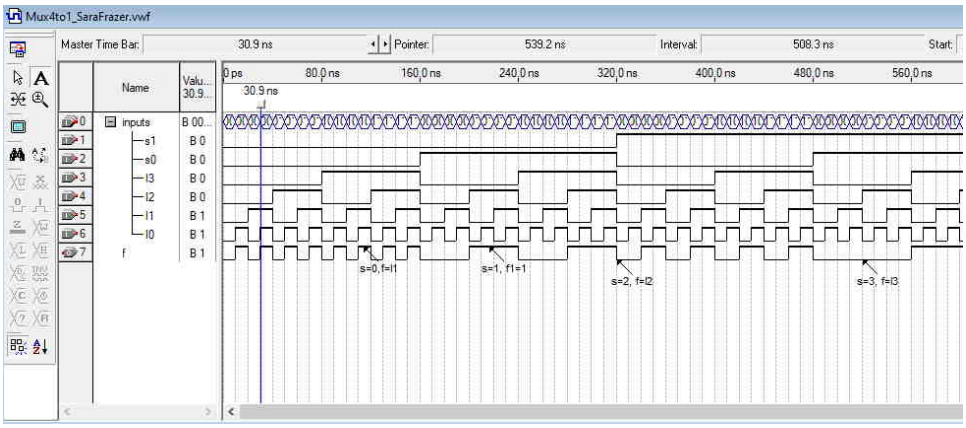
Mux4to1_SaraFrazer.vhd

```
1   -- Mux4tol Sara Frazer
2   LIBRARY ieee;
3   USE ieee.std_logic_1164.ALL;
4   ENTITY Mux4tol_SaraFrazer IS
5       PORT(s0, s1, I0, I1, I2, I3 :IN STD_LOGIC;
6            f   :OUT STD_LOGIC);
7   END Mux4tol_SaraFrazer;
8
9   ARCHITECTURE Structure OF Mux4tol_SaraFrazer IS
10      SIGNAL f1, f2:STD_LOGIC;
11  COMPONENT Mux2tol_SaraFrazer IS
12  PORT(s, x1, x2: IN STD_LOGIC;
13      f   :OUT STD_LOGIC);
14  END COMPONENT Mux2tol_SaraFrazer;
15
16  BEGIN
17  mux0: Mux2tol_SaraFrazer PORT MAP (s1, I0, I1, f1);
18  mux1: Mux2tol_SaraFrazer PORT MAP (s1, I2, I3, f2);
19  mux2: Mux2tol_SaraFrazer PORT MAP (s0, f1, f2, f);
20  END Structure;
21
```

Success Compilation

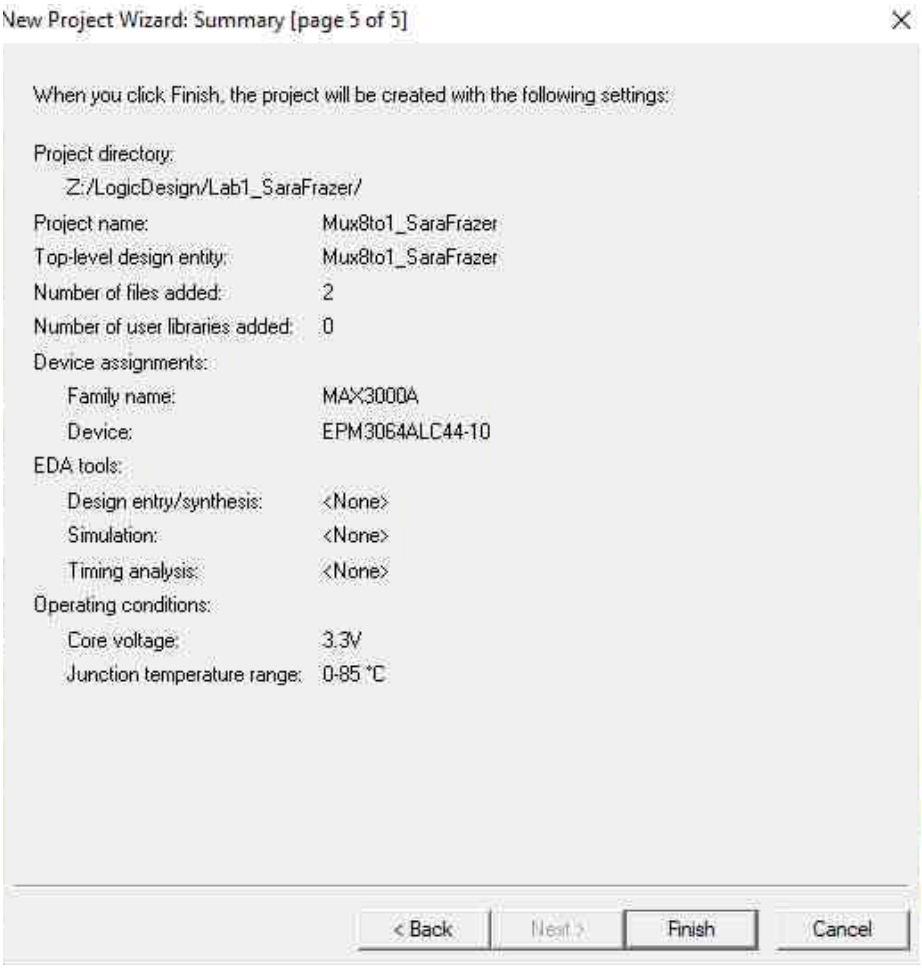


Timing Diagram



Mux 8to1

Project Wizard



I added both mux 2 to and mux 4 to 1 files. Does not change the diagram if I only add mux 4to1.

Drawing