Lab 1

Sophie Samuels
10/1/23

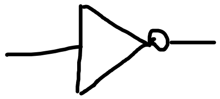
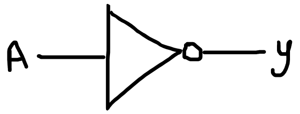
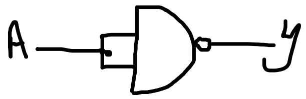
CDA3203 Computer Logic Design

Fall 2023

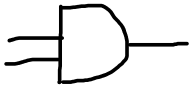
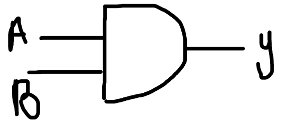
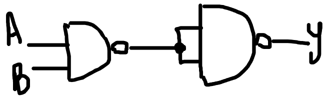
Dr. Maria Petrie
Florida Atlantic University

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.

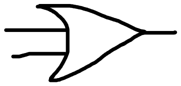
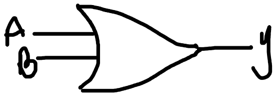
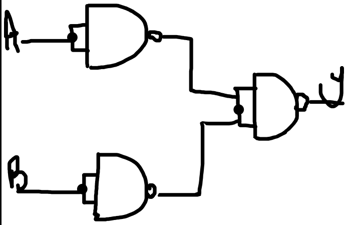
1.1 NOT gate.

<table>
<tr><th>Draw NOT gate</th><th>Truth Table and Simplest Sum of Products Equation</th><th>NOT-AND-OR Equivalent Circuit</th><th>all-NAND Equivalent Circuit</th></tr>
<tr>
<td></td>
<td><table><tr><th>A</th><th>NOT(A)</th></tr><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>
Y1=A’</td>
<td> </td>
<td> </td>
</tr>
</table>

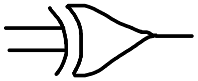
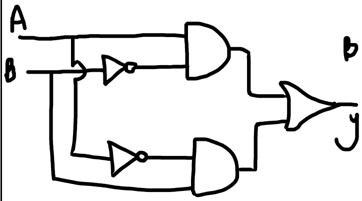
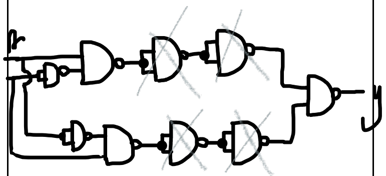
1.2- AND gate

<table>
<tr><th>Draw AND gate</th><th>Truth Table and Simplest Sum of Products Equation</th><th>NOT-AND-OR Equivalent Circuit</th><th>all-NAND Equivalent Circuit</th></tr>
<tr>
<td></td>
<td><table><tr><th>A</th><th>B</th><th>AND(A,B)</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>
Y2=AB</td>
<td> </td>
<td> </td>
</tr>
</table>

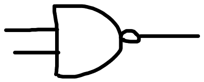
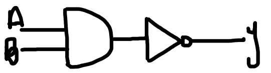
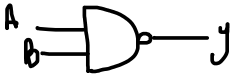
1.3- OR gate

<table>
<tr><th>Draw OR gate</th><th>Truth Table and Simplest Sum of Products Equation</th><th>NOT-AND-OR Equivalent Circuit</th><th>all-NAND Equivalent Circuit</th></tr>
<tr><td></td><td><table><tr><th>A</th><th>B</th><th>OR(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>1</td></tr></table>
Y3=A+B</td><td> </td><td> </td></tr>
</table>

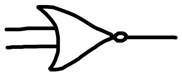
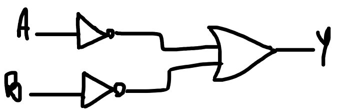
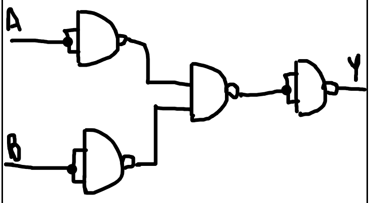
1.4- XOR gate

<table>
<tr><th>Draw XOR gate</th><th>Truth Table and Simplest Sum of Products Equation</th><th>NOT-AND-OR Equivalent Circuit</th><th>all-NAND Equivalent Circuit</th></tr>
<tr><td></td><td><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=A⊕B</td><td> </td><td> </td></tr>
</table>

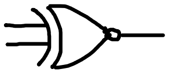
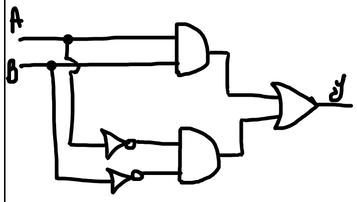
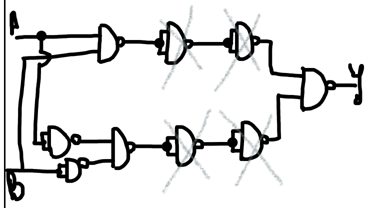
1.5- NAND gate

<table>
<tr><th>Draw NAND gate</th><th>Truth Table and Simplest Sum of Products Equation</th><th>NOT-AND-OR Equivalent Circuit</th><th>all-NAND Equivalent Circuit</th></tr>
<tr>
<td></td>
<td>
<table>
<tr><th>A</th><th>B</th><th>NAND(A,B)</th></tr>
<tr><td><b>0</b></td><td>0</td><td>1</td></tr>
<tr><td><b>0</b></td><td>1</td><td>1</td></tr>
<tr><td><b>1</b></td><td>0</td><td>1</td></tr>
<tr><td><b>1</b></td><td>1</td><td>0</td></tr>
</table>
Y5=(AB)’
</td>
<td>
</td>
<td>
</td>
</tr>
</table>

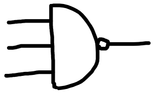
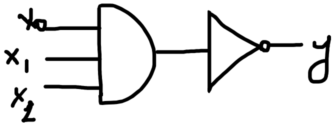
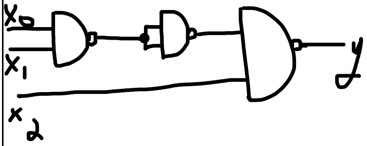
1.6- NOR gate

<table>
<tr><th>Draw NOR gate</th><th>Truth Table and Simplest Sum of Products Equation</th><th>NOT-AND-OR Equivalent Circuit</th><th>all-NAND Equivalent Circuit</th></tr>
<tr>
<td></td>
<td>
<table>
<tr><th>A</th><th>B</th><th>NOR(A,B)</th></tr>
<tr><td><b>0</b></td><td>0</td><td>1</td></tr>
<tr><td><b>0</b></td><td>1</td><td>0</td></tr>
<tr><td><b>1</b></td><td>0</td><td>0</td></tr>
<tr><td><b>1</b></td><td>1</td><td>0</td></tr>
</table>
Y6=(A+B)’
</td>
<td>
</td>
<td>
</td>
</tr>
</table>

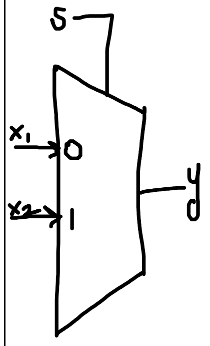
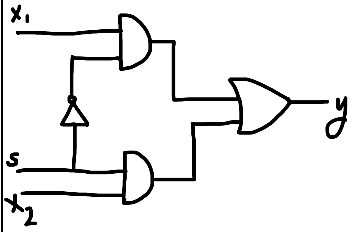
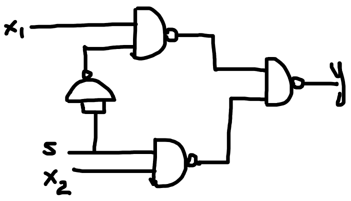
1.7- XNOR gate

<table>
<tr><th>Draw XNOR gate</th><th>Truth Table and Simplest Sum of Products Equation</th><th>NOT-AND-OR Equivalent Circuit</th><th>all-NAND Equivalent Circuit</th></tr>
<tr>
<td></td>
<td>
<table>
<tr><th>A</th><th>B</th><th>XNOR(A,B)</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
Y7=A⊙B
</td>
<td>
</td>
<td>
</td>
</tr>
</table>

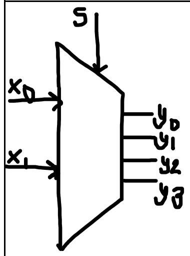
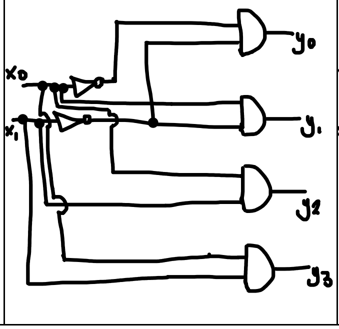
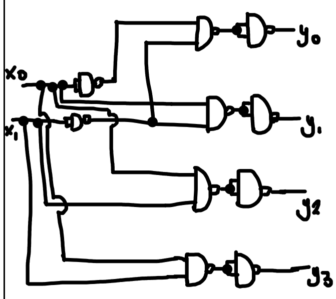
1.8 3-input NAND gate

<table>
<tr><th>Draw 3-input NAND gate</th><th>Truth Table and Simplest Sum of Products Equation</th><th>NOT-AND-OR Equivalent Circuit</th><th>all-NAND Equivalent Circuit</th></tr>
<tr>
<td></td>
<td>
<table>
<tr><th>X_2</th><th>X_1</th><th>X_0</th><th>NAND3</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table>
Y8=(ABC)’
</td>
<td>
</td>
<td>
</td>
</tr>
</table>

1.9- 2 to 1 Encoder or Multiplexer (Mux)

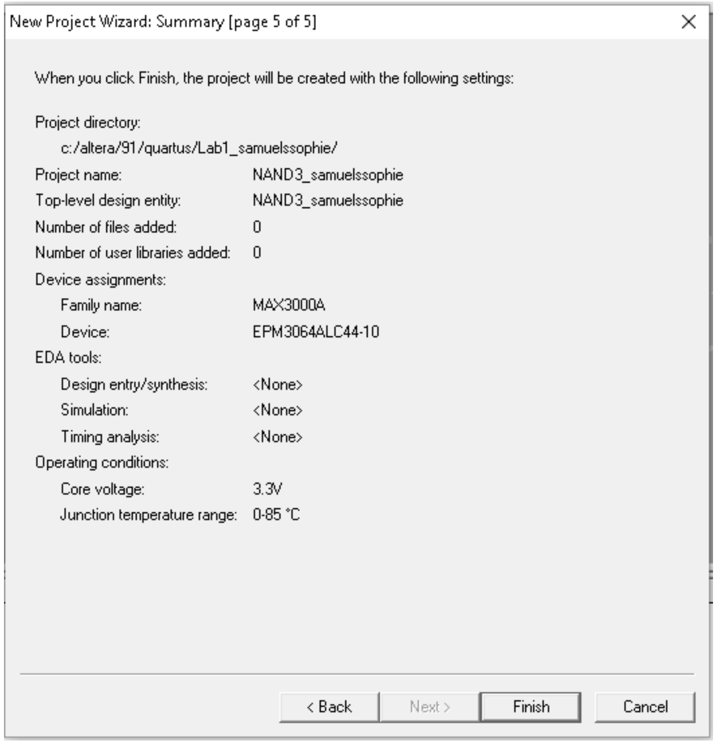
<table>
<tr><th>Draw 2to1 Mux</th><th>Truth Table and Simplest Sum of Products Equation</th><th>NOT-AND-OR Equivalent Circuit</th><th>all-NAND Equivalent Circuit</th></tr>
<tr>
<td>
</td>
<td>
<table>
<tr><th>s</th><th>X_1</th><th>X_2</th><th>Mux</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
$Y9 = s'x_1 + sx_2$</td>
<td>
</td>
<td>
</td>
</tr>
</table>

1.10 - 2 to 4 decoder or Demultiplexer (DMux)

<table>
<tr><th>Draw 2to4 DMux</th><th>Truth Table and Simplest Sum of Products Equation</th><th>NOT-AND-OR Equivalent Circuit</th><th>all-NAND Equivalent Circuit</th></tr>
<tr>
<td>
</td>
<td>
<table>
<tr><th>X_1</th><th>X_0</th><th>Y_3</th><th>Y_2</th><th>Y_1</th><th>Y_0</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table>
$Y3=X_0X_1$
$Y2=X_0'X_1$
$Y1=X_0X_1'$
$Y0=X_0'X_1'$</td>
<td>
</td>
<td>
</td>
</tr>
</table>

2.1 Create a new project in Altera Quartus using VHDL of a **3-input NAND** circuit call it NAND3_YourName, using all-NAND gates

Project Wizard



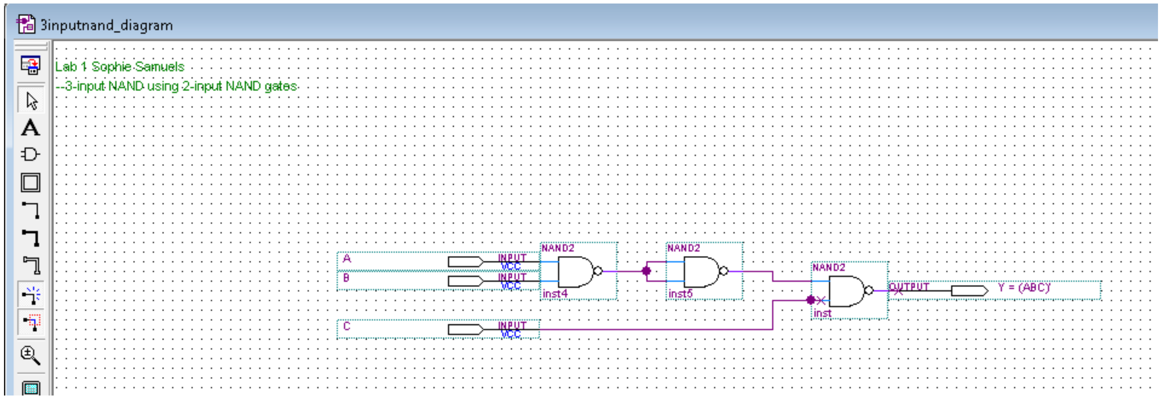
New Project Wizard: Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

Project directory:
c:/altera/91/quartus/Lab1_samuelssophie/

Project name:	NAND3_samuelssophie
Top-level design entity:	NAND3_samuelssophie
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Family name:	MAX3000A
Device:	EPM3064ALC44-10
EDA tools:	
Design entry/synthesis:	<None>
Simulation:	<None>
Timing analysis:	<None>
Operating conditions:	
Core voltage:	3.3V
Junction temperature range:	0-85 °C

3-input NAND Diagram



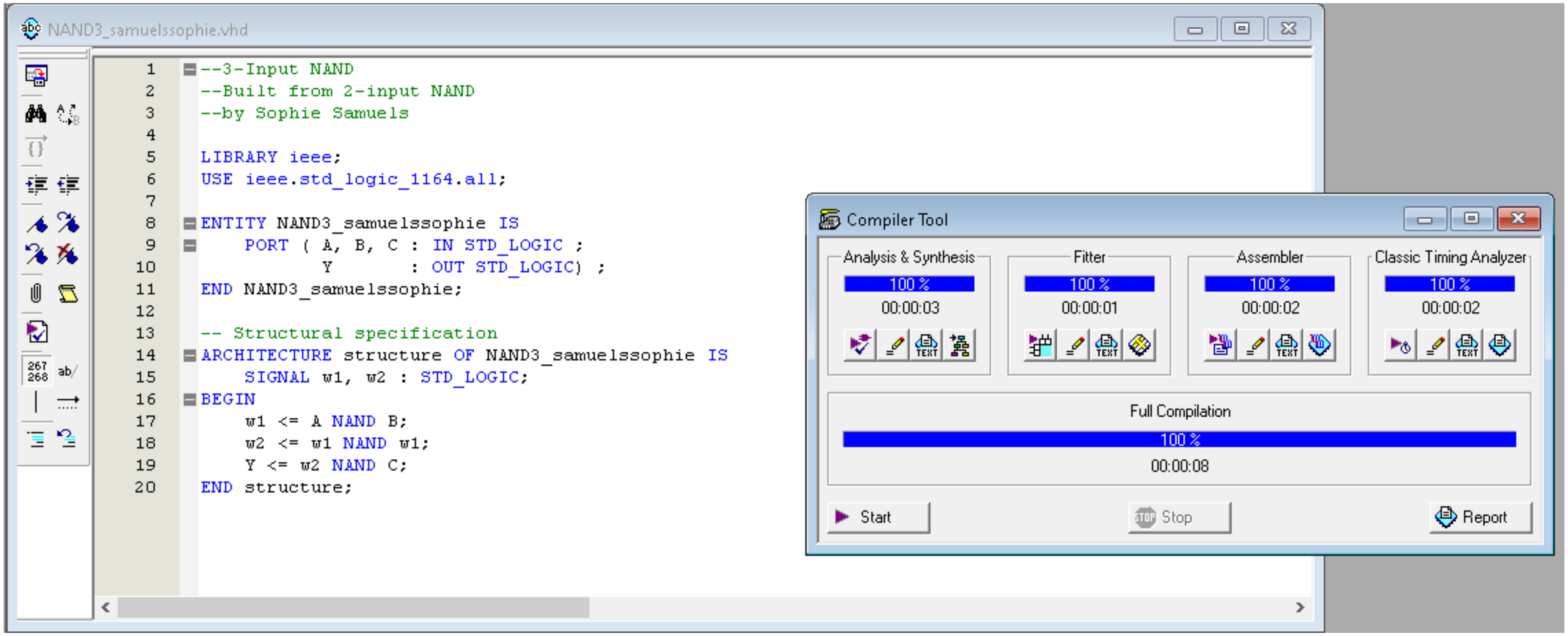
3inputnand_diagram

Lab 1 Sophie Samuels
--3-input NAND using 2-input NAND gates

Truth Table

X_2	X_1	X_0	**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

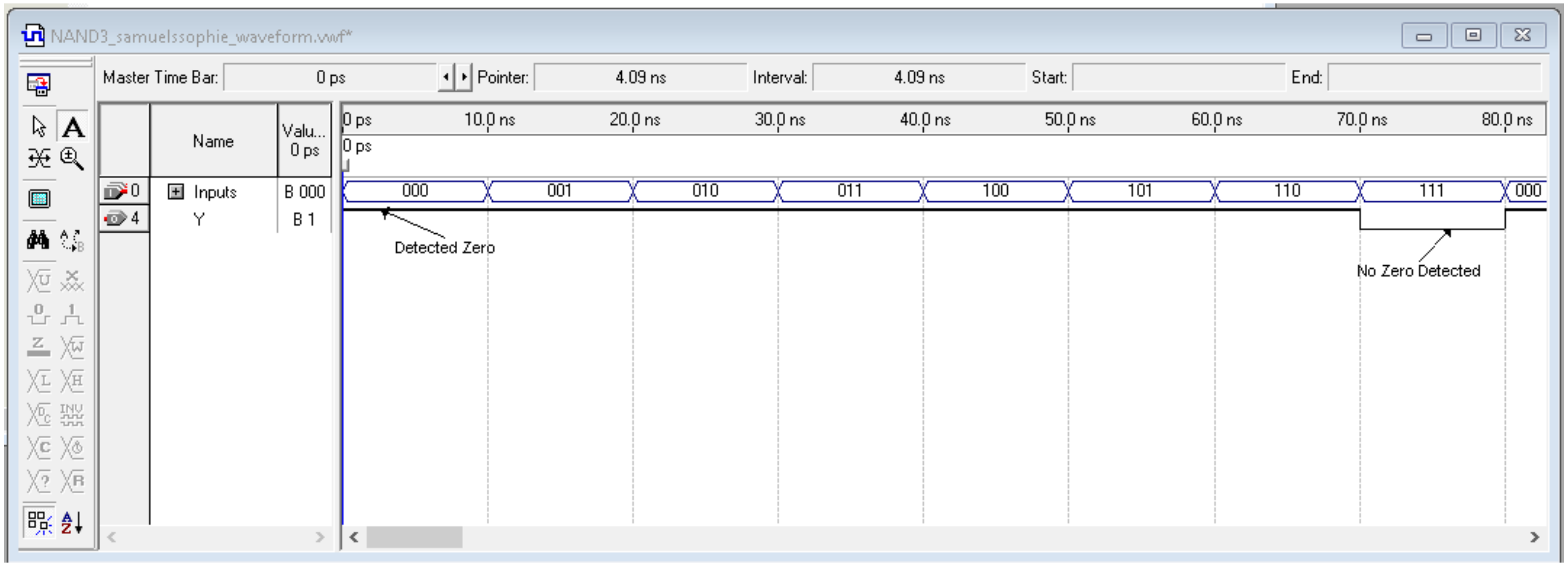
Successful Compilation



```
--3-Input NAND
--Built from 2-input NAND
--by Sophie Samuels

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY NAND3_samuelssophie IS
    PORT ( A, B, C : IN STD_LOGIC ;
           Y       : OUT STD_LOGIC) ;
END NAND3_samuelssophie;

-- Structural specification
ARCHITECTURE structure OF NAND3_samuelssophie IS
    SIGNAL w1, w2 : STD_LOGIC;
BEGIN
    w1 <= A NAND B;
    w2 <= w1 NAND w1;
    Y <= w2 NAND C;
END structure;
```

Analysis & Synthesis: 100 % 00:00:03
Fitter: 100 % 00:00:01
Assembler: 100 % 00:00:02
Classic Timing Analyzer: 100 % 00:00:02
Full Compilation: 100 % 00:00:08

Timing Diagram



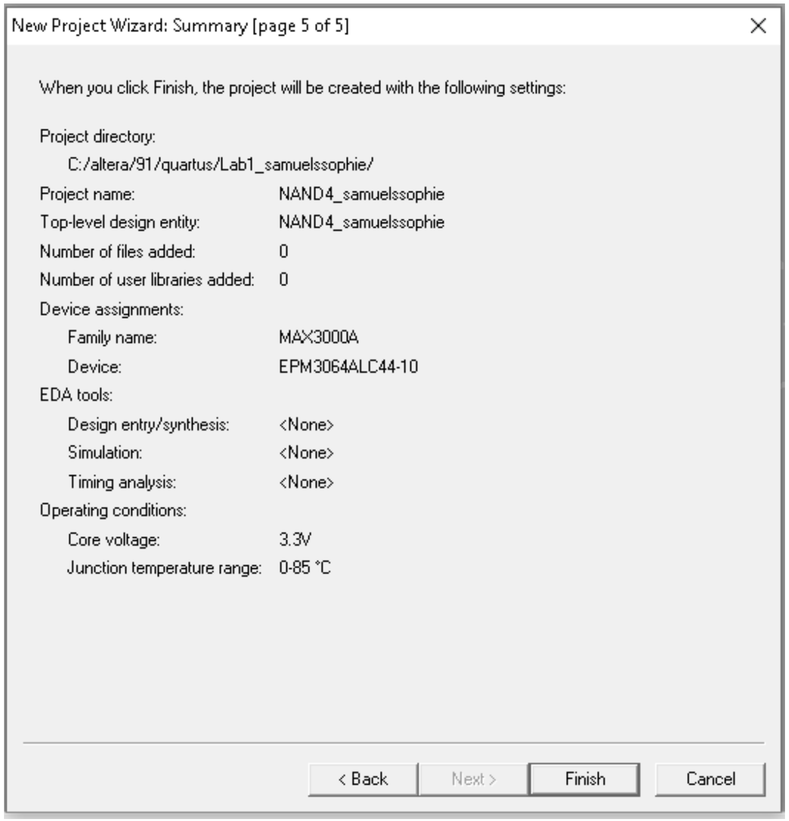
Master Time Bar: 0 ps Pointer: 4.09 ns Interval: 4.09 ns Start: End:

Detected Zero

No Zero Detected

2.2 Create a new project in Altera Quartus using VHDL of a **4-input NAND** circuit, call it NAND4_YourName, **using all-NAND gates**

Project Wizard

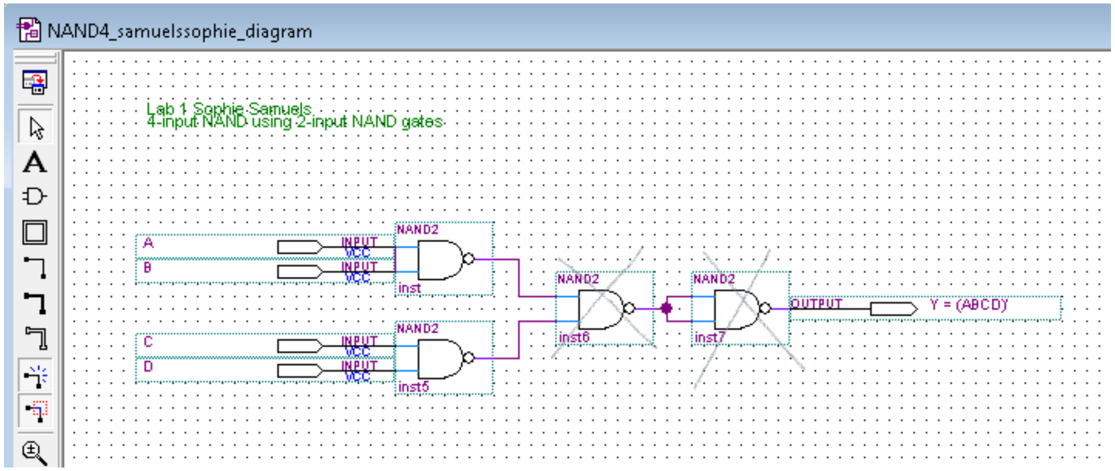


New Project Wizard: Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

Project directory:	
C:/altera/91/quartus/Lab1_samuelssophie/	
Project name:	NAND4_samuelssophie
Top-level design entity:	NAND4_samuelssophie
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Family name:	MAX3000A
Device:	EPM3064ALC44-10
EDA tools:	
Design entry/synthesis:	<None>
Simulation:	<None>
Timing analysis:	<None>
Operating conditions:	
Core voltage:	3.3V
Junction temperature range:	0-85 °C

4-input NAND Diagram



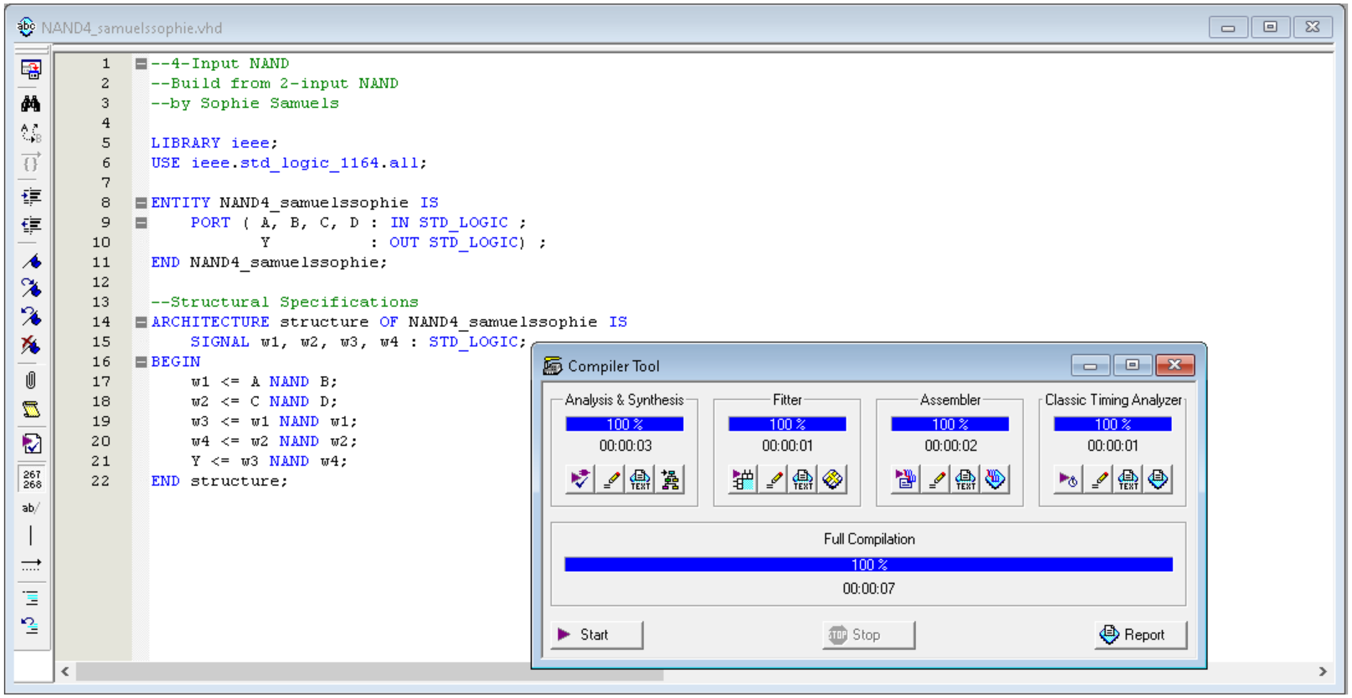
NAND4_samuelssophie_diagram

Lab 1 Sophie Samuels
4-input NAND using 2-input NAND gates

Truth Table

A	B	C	D	**NAND4**
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

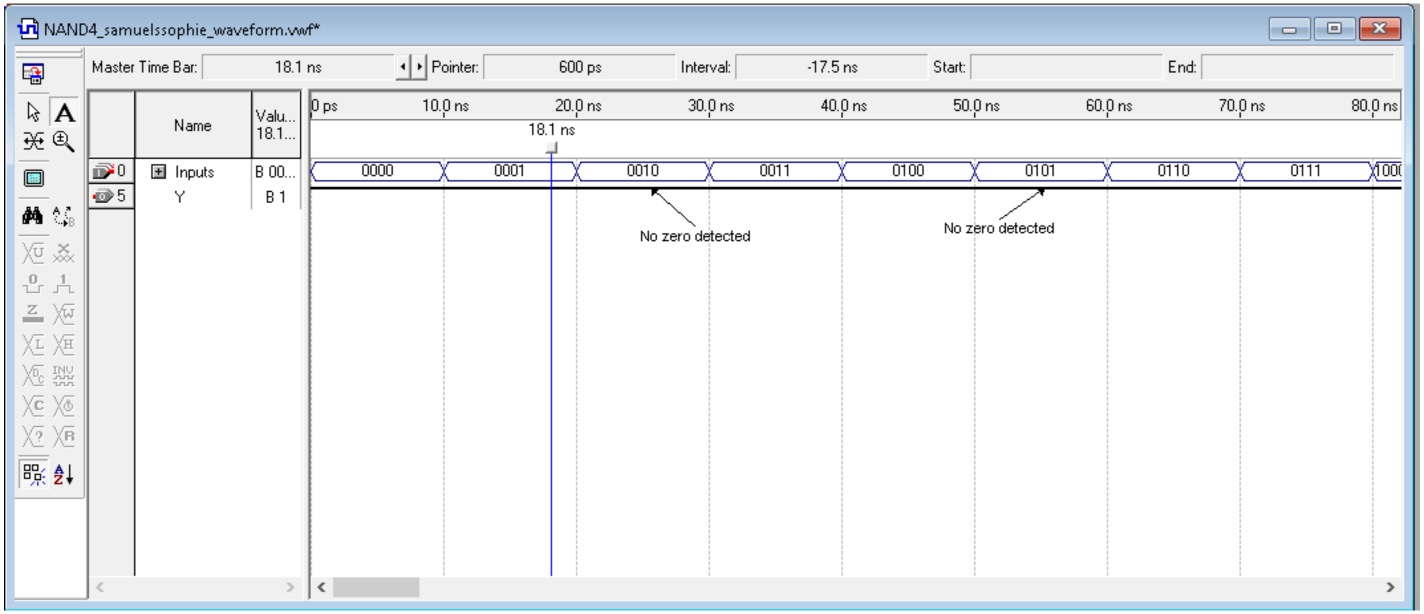
Successful Compilation



```
--4-Input NAND
--Build from 2-input NAND
--by Sophie Samuels

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY NAND4_samuelssophie IS
    PORT ( A, B, C, D : IN STD_LOGIC ;
           Y          : OUT STD_LOGIC) ;
END NAND4_samuelssophie;

--Structural Specifications
ARCHITECTURE structure OF NAND4_samuelssophie IS
    SIGNAL w1, w2, w3, w4 : STD_LOGIC;
BEGIN
    w1 <= A NAND B;
    w2 <= C NAND D;
    w3 <= w1 NAND w1;
    w4 <= w2 NAND w2;
    Y <= w3 NAND w4;
END structure;
```

```
Analysis & Synthesis   Fitter     Assembler   Classic Timing Analyzer
100 %                  100 %      100 %       100 %
00:00:03               00:00:01   00:00:02    00:00:01

Full Compilation
100 %
00:00:07
```

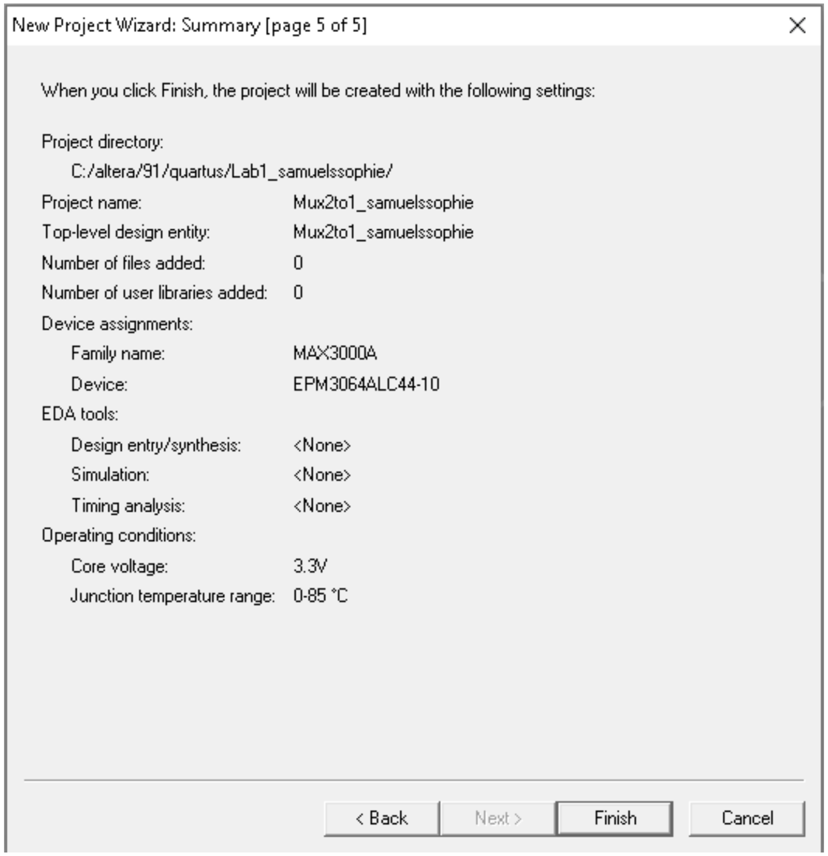
Timing Diagram



Master Time Bar: 18.1 ns Pointer: 600 ps Interval: -17.5 ns Start: End:

2.3 Create a new project in Altera Quartus using VHDL of a **2 to 1 Multiplexer/Encoder (Mux)** circuit, call it Mux2to1_YourName, **using all-NAND gates**

Project Wizard



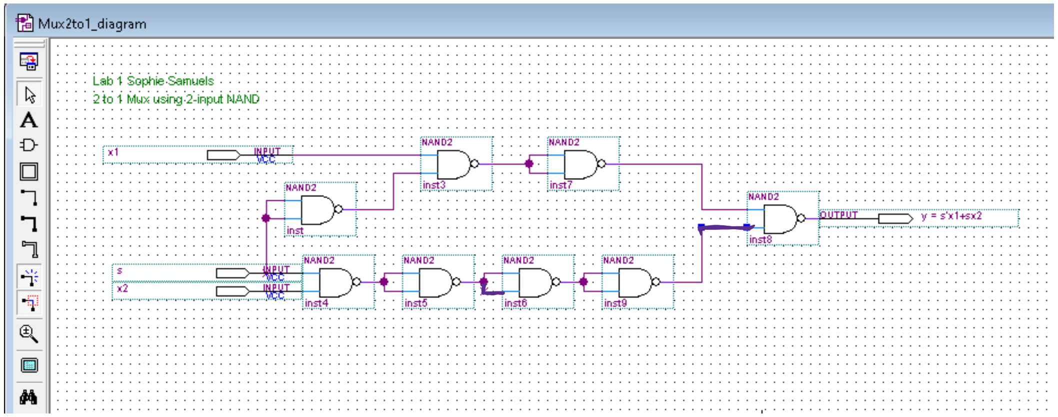
New Project Wizard: Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

Project directory:
C:/altera/91/quartus/Lab1_samuelssophie/

Project name:	Mux2to1_samuelssophie
Top-level design entity:	Mux2to1_samuelssophie
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Family name:	MAX3000A
Device:	EPM3064ALC44-10
EDA tools:	
Design entry/synthesis:	<None>
Simulation:	<None>
Timing analysis:	<None>
Operating conditions:	
Core voltage:	3.3V
Junction temperature range:	0-85 °C

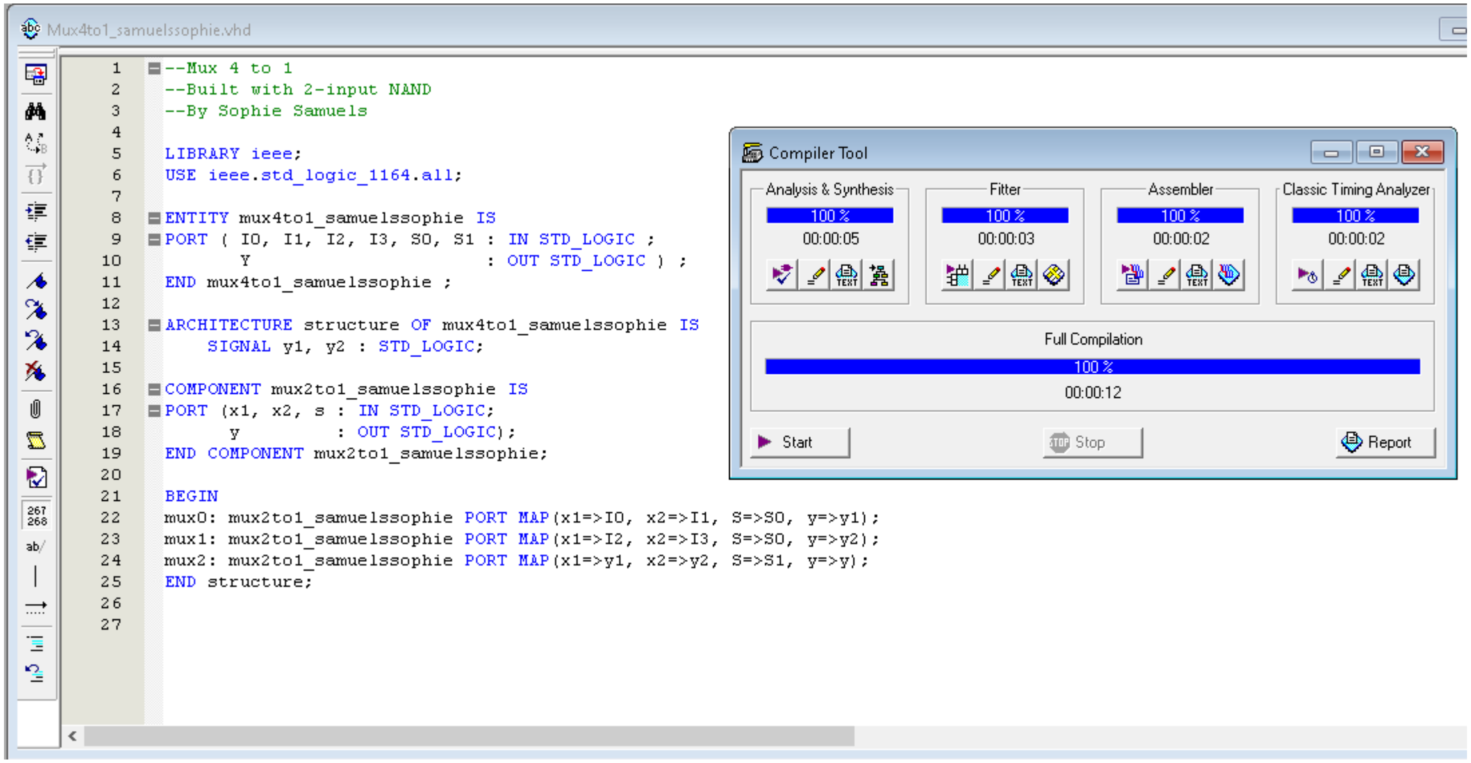
Mux2to1 Diagram



Truth Table

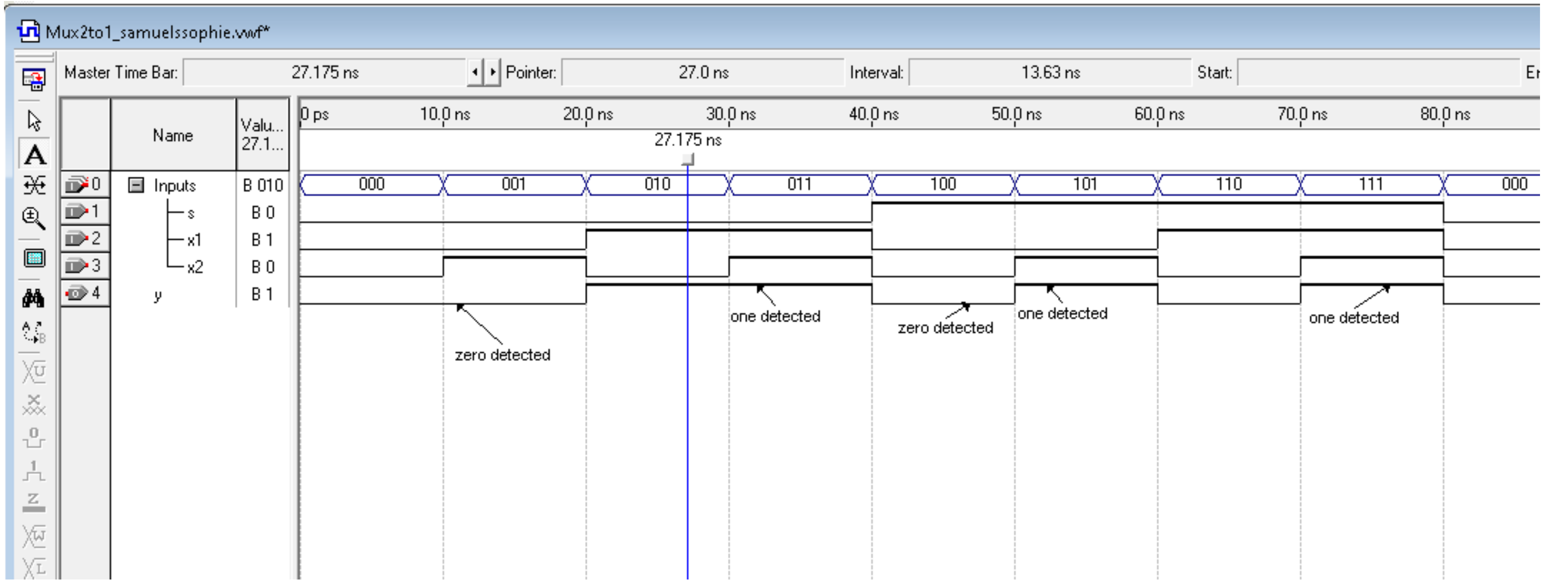
s	X_0	X_1	**Mux2:1**
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Successful Compilation



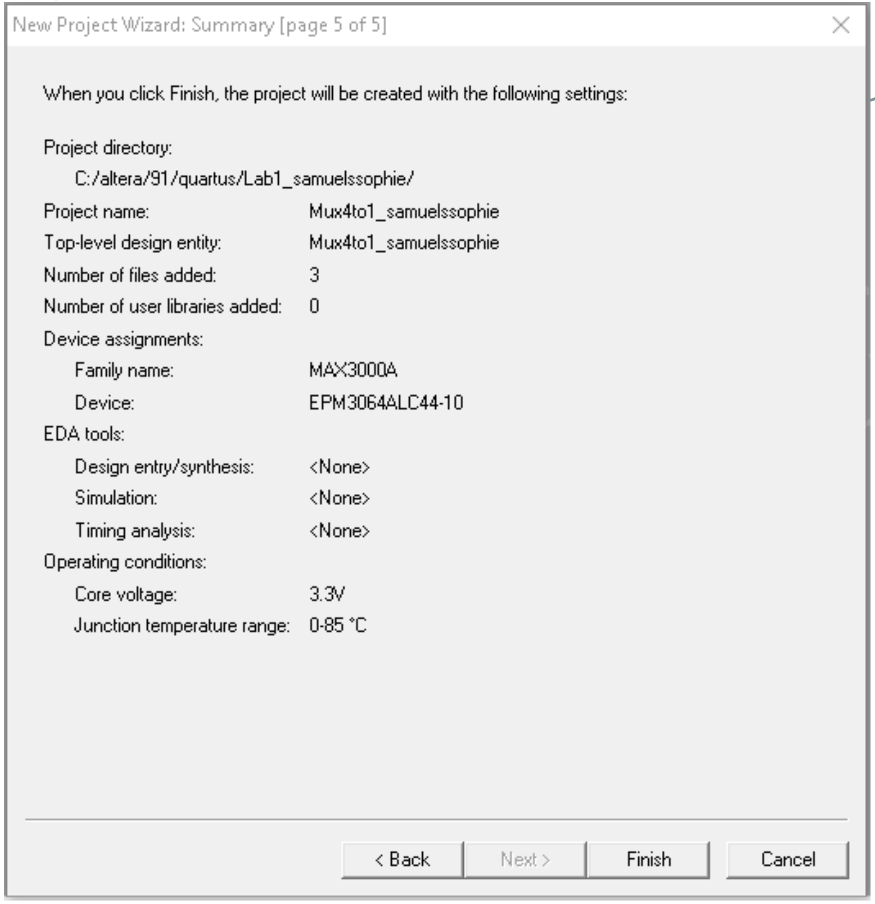
```vhdl
--Mux 4 to 1
--Built with 2-input NAND
--By Sophie Samuels

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY mux4to1_samuelssophie IS
PORT ( I0, I1, I2, I3, S0, S1 : IN STD_LOGIC ;
       Y                      : OUT STD_LOGIC ) ;
END mux4to1_samuelssophie ;

ARCHITECTURE structure OF mux4to1_samuelssophie IS
    SIGNAL y1, y2 : STD_LOGIC;

COMPONENT mux2to1_samuelssophie IS
PORT (x1, x2, s : IN STD_LOGIC;
      y         : OUT STD_LOGIC);
END COMPONENT mux2to1_samuelssophie;

BEGIN
mux0: mux2to1_samuelssophie PORT MAP(x1=>I0, x2=>I1, S=>S0, y=>y1);
mux1: mux2to1_samuelssophie PORT MAP(x1=>I2, x2=>I3, S=>S0, y=>y2);
mux2: mux2to1_samuelssophie PORT MAP(x1=>y1, x2=>y2, S=>S1, y=>y);
END structure;
```

Timing Diagram



2.4. Create a new project in Altera Quartus using VHDL of a **4-to-1 Mux** circuit, call it Mux4to1_YourName, **using only Mux2to1 components**.

Project Wizard



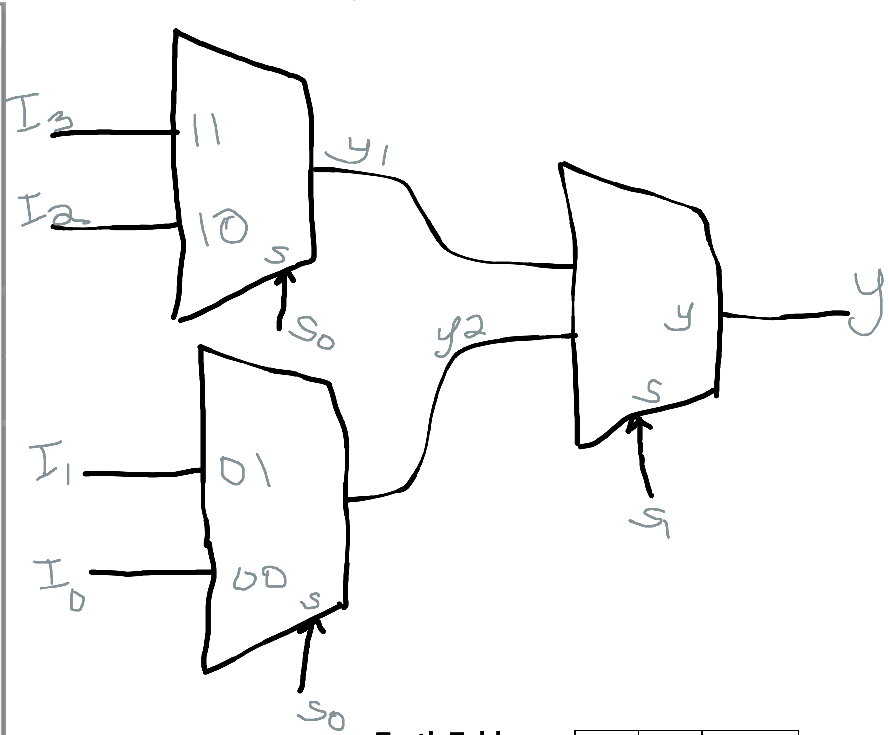
New Project Wizard: Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

Project directory:
C:/altera/91/quartus/Lab1_samuelssophie/

Project name:	Mux4to1_samuelssophie
Top-level design entity:	Mux4to1_samuelssophie
Number of files added:	3
Number of user libraries added:	0
Device assignments:	
Family name:	MAX3000A
Device:	EPM3064ALC44-10
EDA tools:	
Design entry/synthesis:	<None>
Simulation:	<None>
Timing analysis:	<None>
Operating conditions:	
Core voltage:	3.3V
Junction temperature range:	0-85 °C

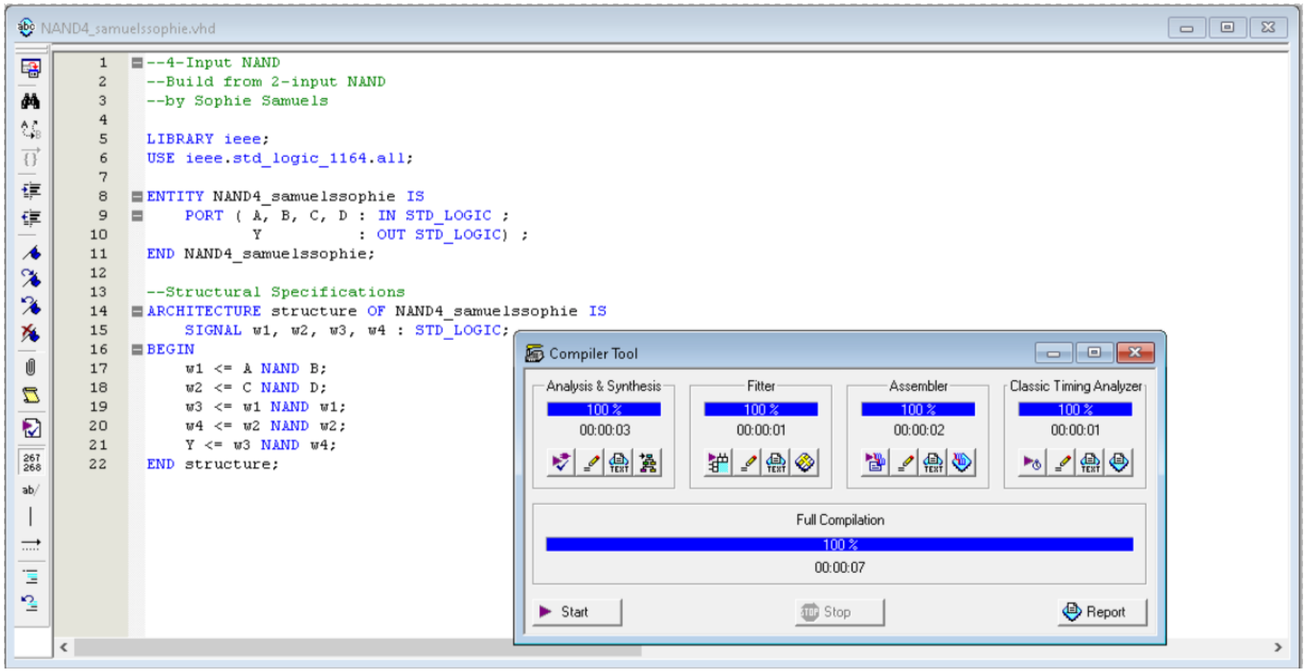
Mux4to1 Diagram



Truth Table

S_0	S_1	Mux4:1
0	0	I0
0	1	I1
1	0	I2
1	1	I3

Successful Compilation


NAND4_samuelssophie.vhd

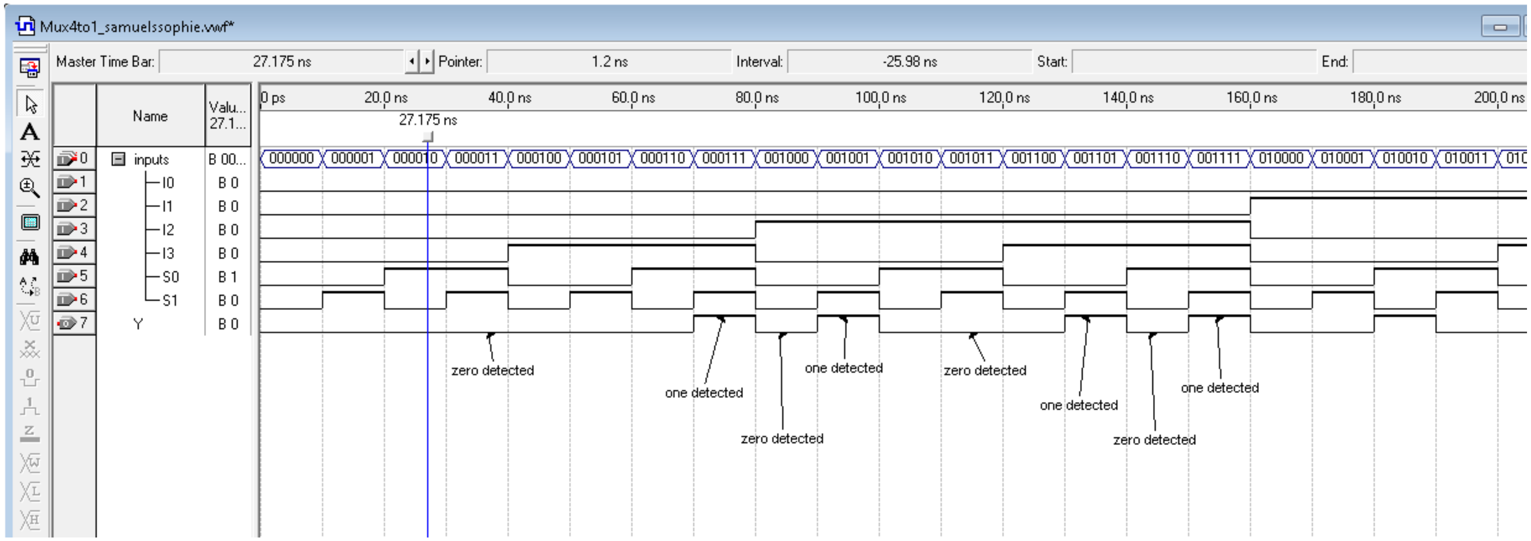
```
--4-Input NAND
--Build from 2-input NAND
--by Sophie Samuels

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY NAND4_samuelssophie IS
    PORT ( A, B, C, D : IN STD_LOGIC ;
           Y          : OUT STD_LOGIC) ;
END NAND4_samuelssophie;

--Structural Specifications
ARCHITECTURE structure OF NAND4_samuelssophie IS
    SIGNAL w1, w2, w3, w4 : STD_LOGIC;
BEGIN
    w1 <= A NAND B;
    w2 <= C NAND D;
    w3 <= w1 NAND w1;
    w4 <= w2 NAND w2;
    Y <= w3 NAND w4;
END structure;
```

Compiler Tool

Analysis & Synthesis	Fitter	Assembler	Classic Timing Analyzer
100 %	100 %	100 %	100 %
00:00:03	00:00:01	00:00:02	00:00:01

Full Compilation: 100 % — 00:00:07

Timing Diagram

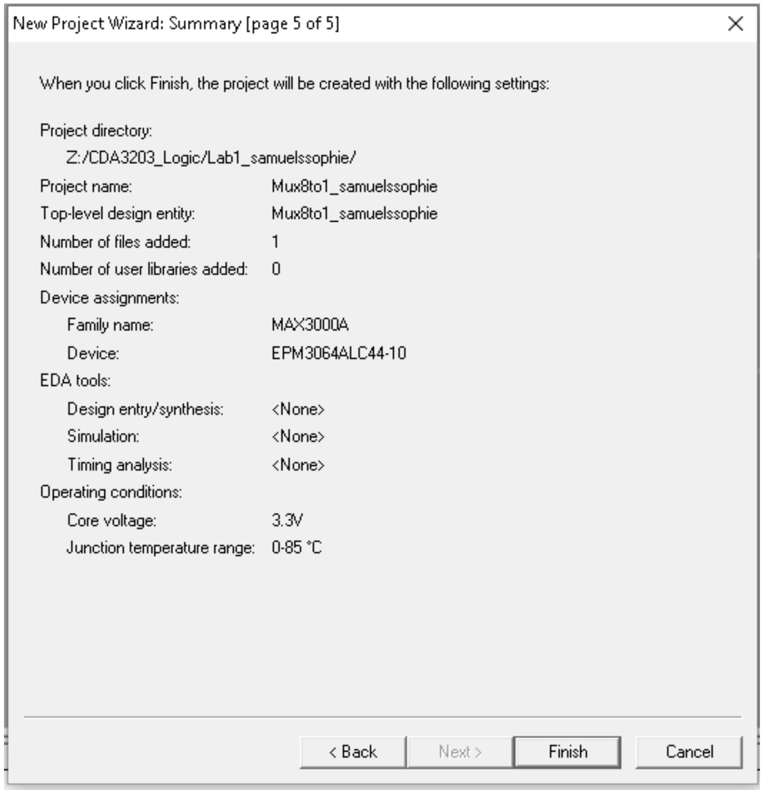

Mux4to1_samuelssophie.vwf*

Master Time Bar: 27.175 ns   Pointer: 1.2 ns   Interval: -25.98 ns   Start: End:

	Name	Valu... 27.1...
0	inputs	B 00...
1	I0	B 0
2	I1	B 0
3	I2	B 0
4	I3	B 0
5	S0	B 1
6	S1	B 0
7	Y	B 0

2.5 Create a new project in Altera Quartus using VHDL of a **8-to-1 Mux**, call it Mux8to1_YourName, using **only Mux4to1 and/or Mux2to1components**

Project Wizard

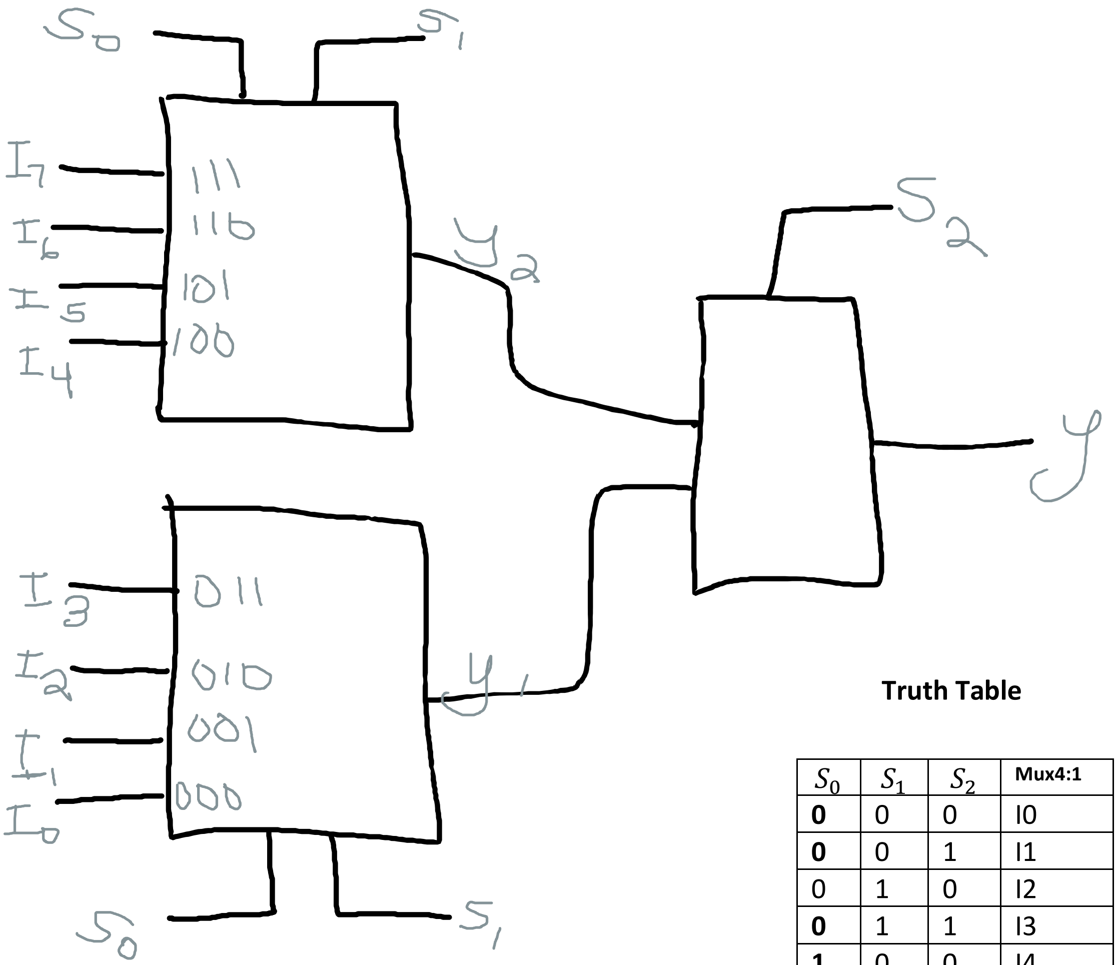


New Project Wizard: Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

Project directory:	
Z:/CDA3203_Logic/Lab1_samuelssophie/	
Project name:	Mux8to1_samuelssophie
Top-level design entity:	Mux8to1_samuelssophie
Number of files added:	1
Number of user libraries added:	0
Device assignments:	
Family name:	MAX3000A
Device:	EPM3064ALC44-10
EDA tools:	
Design entry/synthesis:	<None>
Simulation:	<None>
Timing analysis:	<None>
Operating conditions:	
Core voltage:	3.3V
Junction temperature range:	0-85 °C

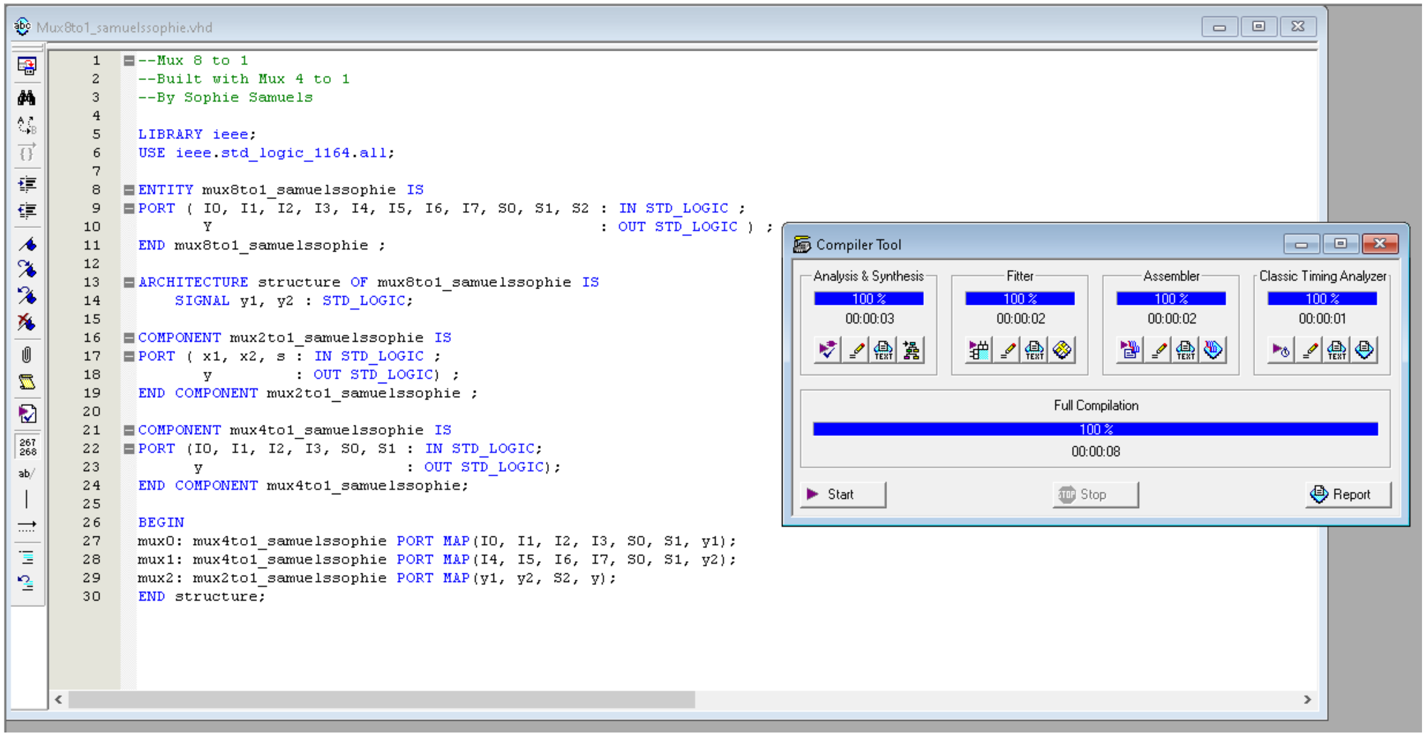
Mux8to1 Diagram



Truth Table

S_0	S_1	S_2	**Mux4:1**
0	0	0	I0
0	0	1	I1
0	1	0	I2
0	1	1	I3
1	0	0	I4
1	0	1	I5
1	1	0	I6
1	1	1	I7

Successful Compilation



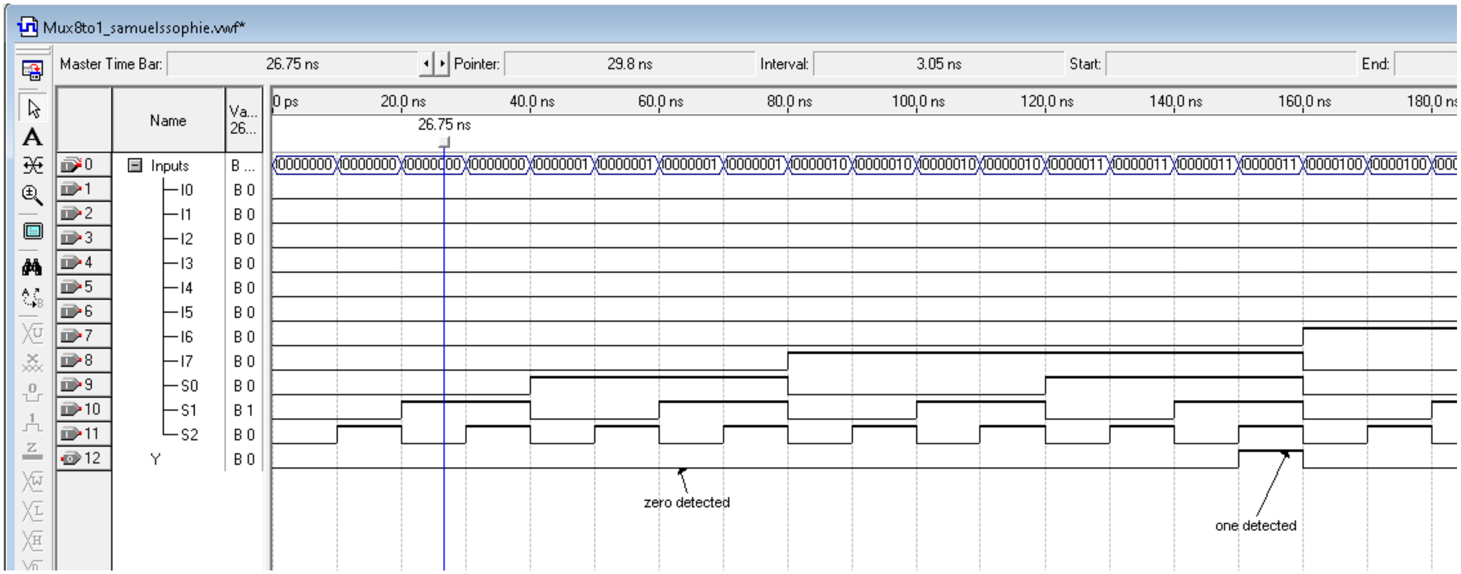
Mux8to1_samuelssophie.vhd

```vhdl
--Mux 8 to 1
--Built with Mux 4 to 1
--By Sophie Samuels

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY mux8to1_samuelssophie IS
PORT ( I0, I1, I2, I3, I4, I5, I6, I7, S0, S1, S2 : IN STD_LOGIC ;
       Y                                           : OUT STD_LOGIC ) ;
END mux8to1_samuelssophie ;

ARCHITECTURE structure OF mux8to1_samuelssophie IS
    SIGNAL y1, y2 : STD_LOGIC;

COMPONENT mux2to1_samuelssophie IS
PORT ( x1, x2, s : IN STD_LOGIC ;
       y         : OUT STD_LOGIC) ;
END COMPONENT mux2to1_samuelssophie ;

COMPONENT mux4to1_samuelssophie IS
PORT (I0, I1, I2, I3, S0, S1 : IN STD_LOGIC;
      y                      : OUT STD_LOGIC);
END COMPONENT mux4to1_samuelssophie;

BEGIN
mux0: mux4to1_samuelssophie PORT MAP(I0, I1, I2, I3, S0, S1, y1);
mux1: mux4to1_samuelssophie PORT MAP(I4, I5, I6, I7, S0, S1, y2);
mux2: mux2to1_samuelssophie PORT MAP(y1, y2, S2, y);
END structure;
```

Compiler Tool

```
Analysis & Synthesis   Fitter     Assembler   Classic Timing Analyzer
100 %                  100 %      100 %       100 %
00:00:03               00:00:02   00:00:02    00:00:01

Full Compilation
100 %
00:00:08
```

Timing Diagram

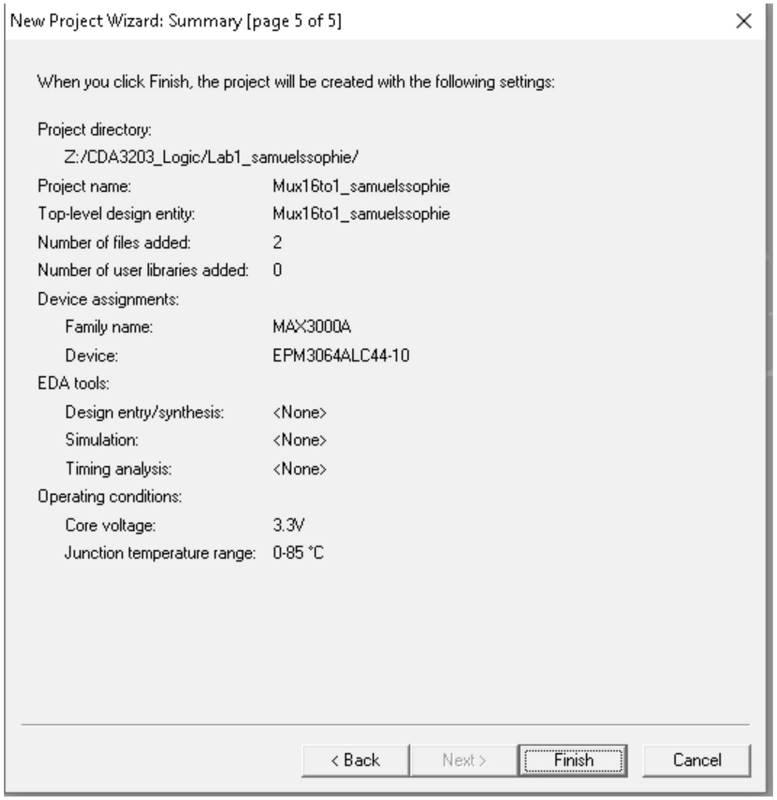


Mux8to1_samuelssophie.vwf*

Master Time Bar: 26.75 ns Pointer: 29.8 ns Interval: 3.05 ns Start: End:

2.6 Create a new project in Altera Quartus using VHDL of a **16-to-1 Mux**, call it Mux8to1_YourName, **using only Mux8to1, Mux4to1 and/or Mux2to1 components.**

Project Wizard

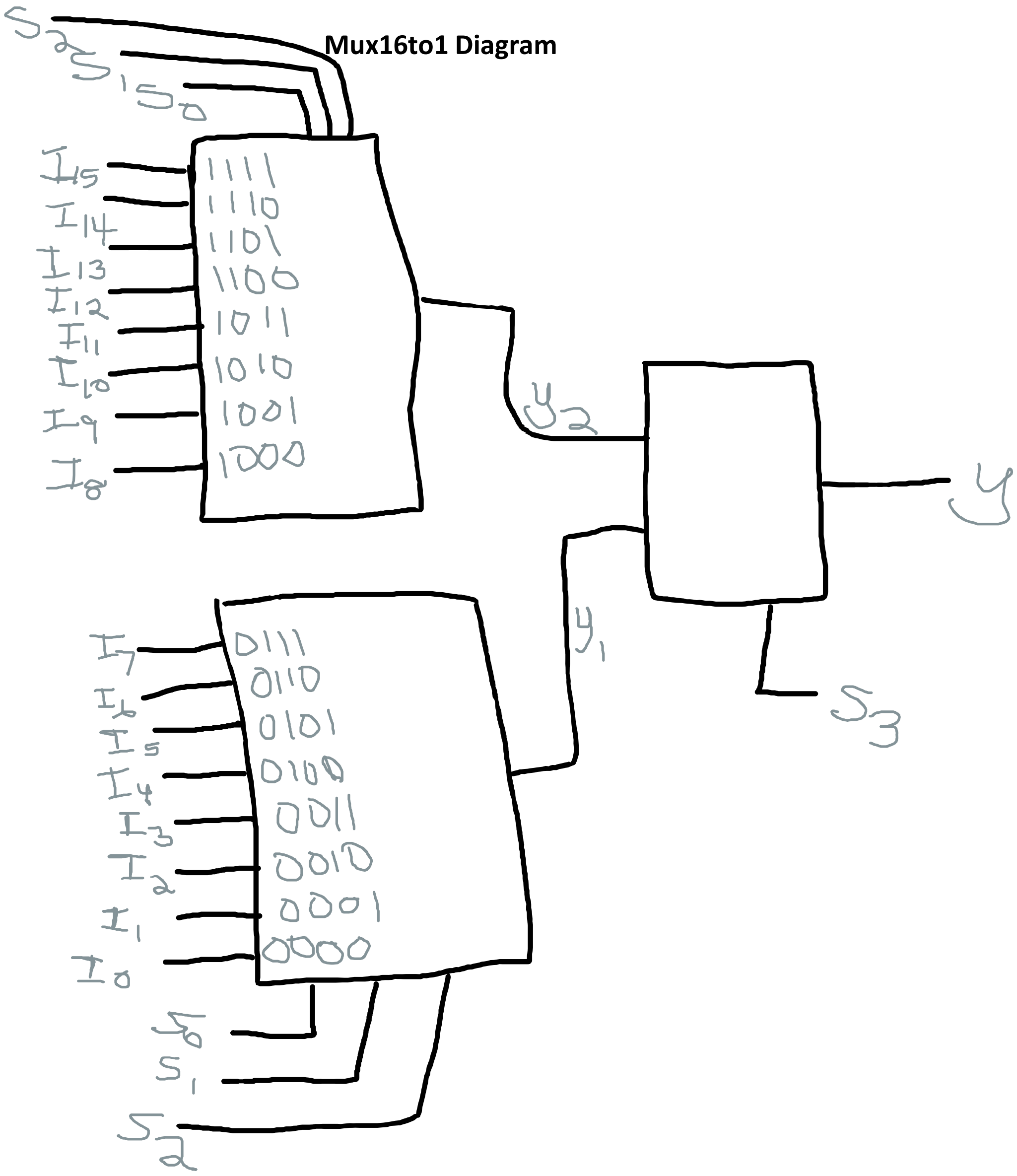


New Project Wizard: Summary [page 5 of 5]

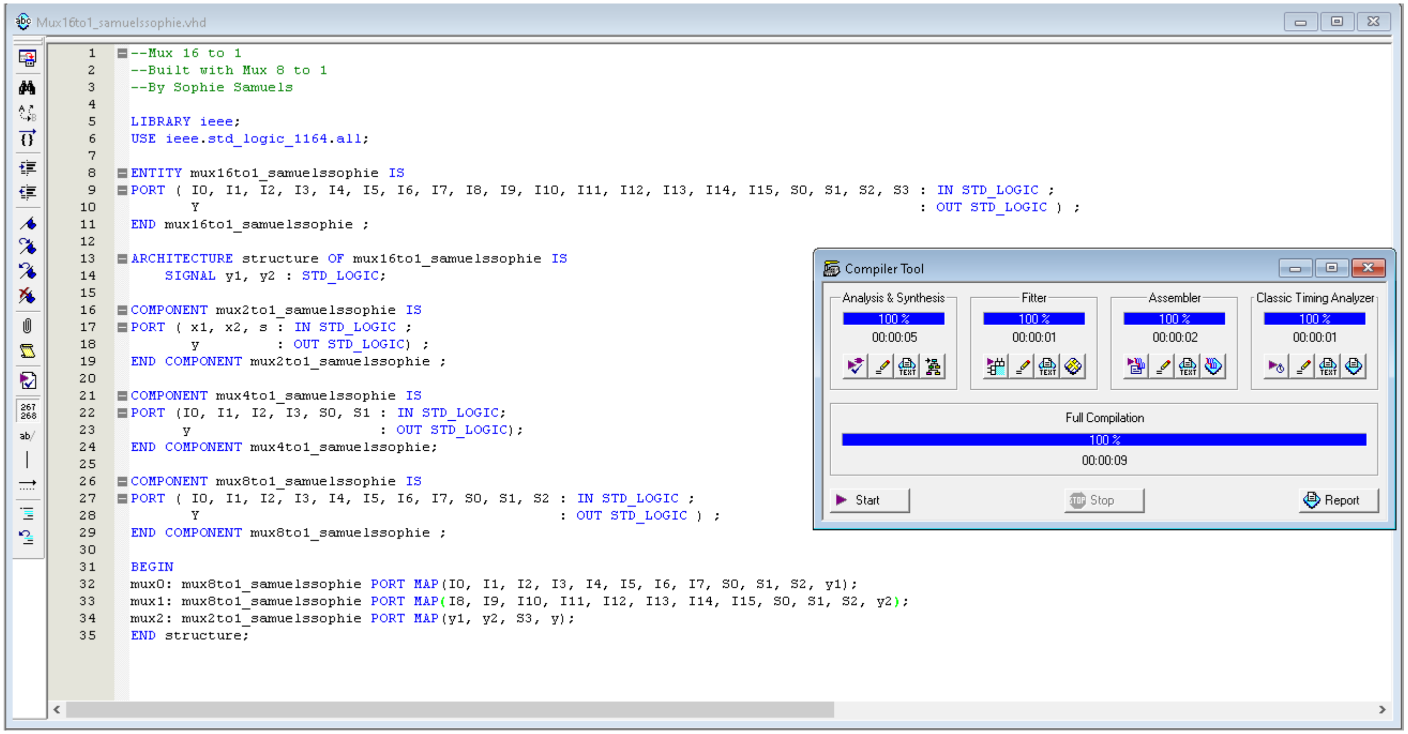
When you click Finish, the project will be created with the following settings:

```
Project directory:
    Z:/CDA3203_Logic/Lab1_samuelssophie/
Project name:                   Mux16to1_samuelssophie
Top-level design entity:        Mux16to1_samuelssophie
Number of files added:          2
Number of user libraries added: 0
Device assignments:
    Family name:                MAX3000A
    Device:                     EPM3064ALC44-10
EDA tools:
    Design entry/synthesis:     <None>
    Simulation:                 <None>
    Timing analysis:            <None>
Operating conditions:
    Core voltage:               3.3V
    Junction temperature range: 0-85 °C
```

Mux16to1 Diagram



Successful Compilation

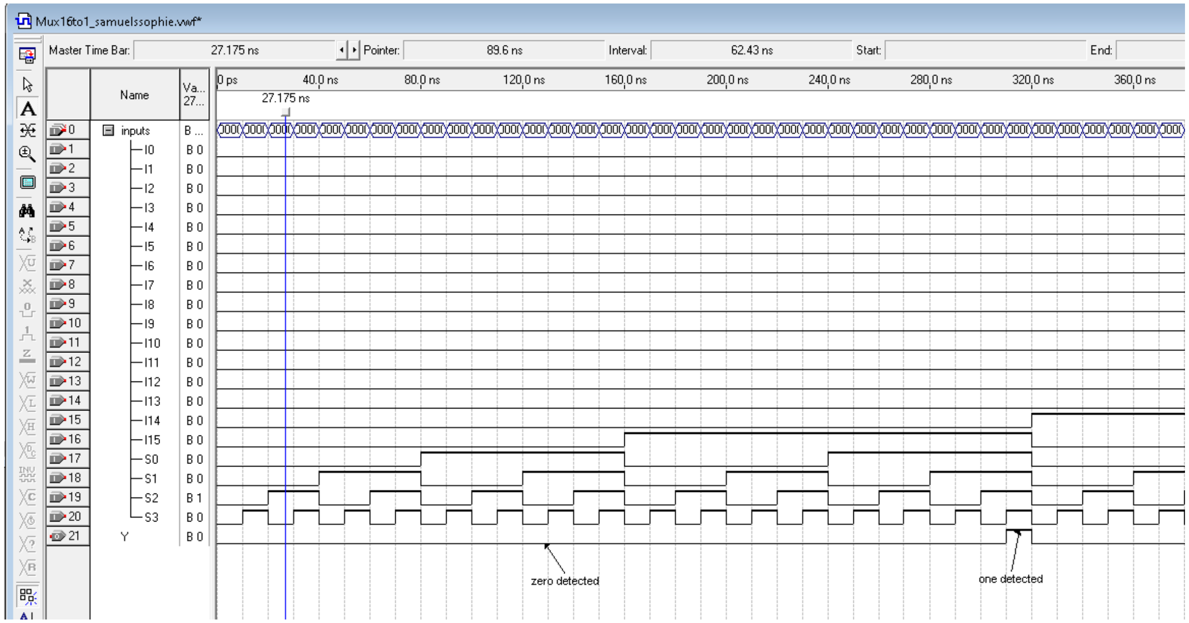


```
--Mux 16 to 1
--Built with Mux 8 to 1
--By Sophie Samuels

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY mux16to1_samuelssophie IS
PORT ( I0, I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14, I15, S0, S1, S2, S3 : IN STD_LOGIC ;
        Y                                                                                    : OUT STD_LOGIC ) ;
END mux16to1_samuelssophie ;

ARCHITECTURE structure OF mux16to1_samuelssophie IS
    SIGNAL y1, y2 : STD_LOGIC;

COMPONENT mux2to1_samuelssophie IS
PORT ( x1, x2, s : IN STD_LOGIC ;
        y        : OUT STD_LOGIC) ;
END COMPONENT mux2to1_samuelssophie ;

COMPONENT mux4to1_samuelssophie IS
PORT (I0, I1, I2, I3, S0, S1 : IN STD_LOGIC;
       y                     : OUT STD_LOGIC);
END COMPONENT mux4to1_samuelssophie;

COMPONENT mux8to1_samuelssophie IS
PORT ( I0, I1, I2, I3, I4, I5, I6, I7, S0, S1, S2 : IN STD_LOGIC ;
        Y                                         : OUT STD_LOGIC ) ;
END COMPONENT mux8to1_samuelssophie ;

BEGIN
mux0: mux8to1_samuelssophie PORT MAP(I0, I1, I2, I3, I4, I5, I6, I7, S0, S1, S2, y1);
mux1: mux8to1_samuelssophie PORT MAP(I8, I9, I10, I11, I12, I13, I14, I15, S0, S1, S2, y2);
mux2: mux2to1_samuelssophie PORT MAP(y1, y2, S3, y);
END structure;
```

Truth Table

S_0	S_1	S_2	S_3	Mux4:1
0	0	0	0	I0
0	0	0	1	I1
0	0	1	0	I2
0	0	1	1	I3
0	1	0	0	I4
0	1	0	1	I5
0	1	1	0	I6
0	1	1	1	I7
1	0	0	0	I8
1	0	0	1	I9
1	0	1	0	I10
1	0	1	1	I11
1	1	0	0	I12
1	1	0	1	I13
1	1	1	0	I14
1	1	1	1	I15

Timing Diagram



This project was done by: Sophie Samuels

following the tutorial videos created by Dr. Petrie and material in the class textbook,
with no other outside resources or help with the following exceptions:

_✓_ Received help from Teaching Assistant: Chelsea, Harry

_✓_ Found the following material on the internet: https://www.electronics-tutorials.ws/logic/logic_5.html,

___ Other: