# Lab 2

**Shahed Ahmed**

3/19/2023

CDA3203 Computer Logic Design

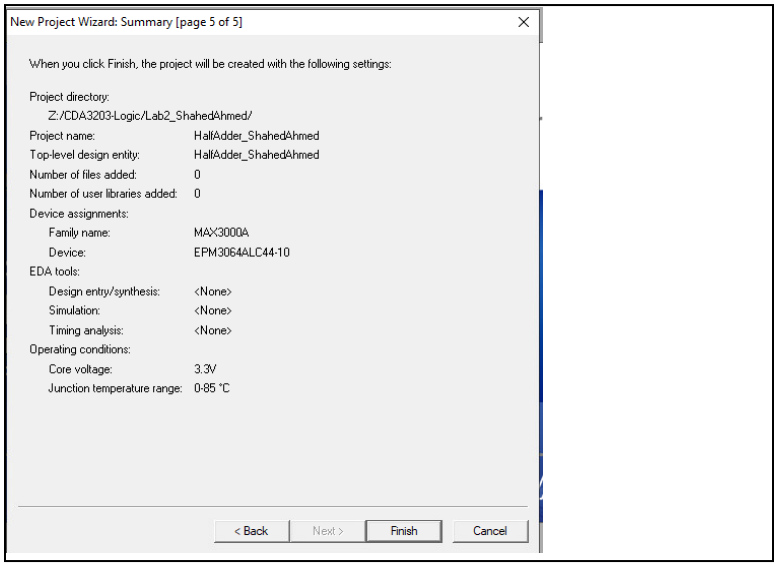
Spring 2023

Dr. Maria Petrie

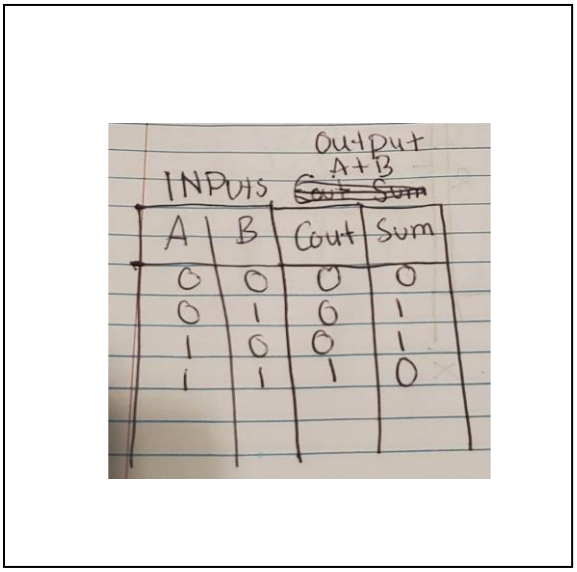
Florida Atlantic University

**Part 1:** A 1-bit Half Adder to add 2 binary bits (A, B) and results in a 1-bit Sum and 1-bit Carry-out (Cout) with only NAND gates.

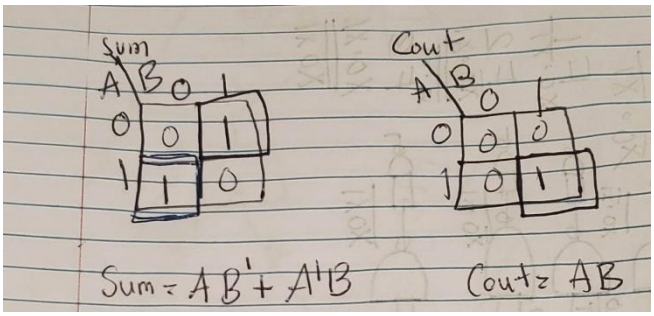
**Project Settings Snip**



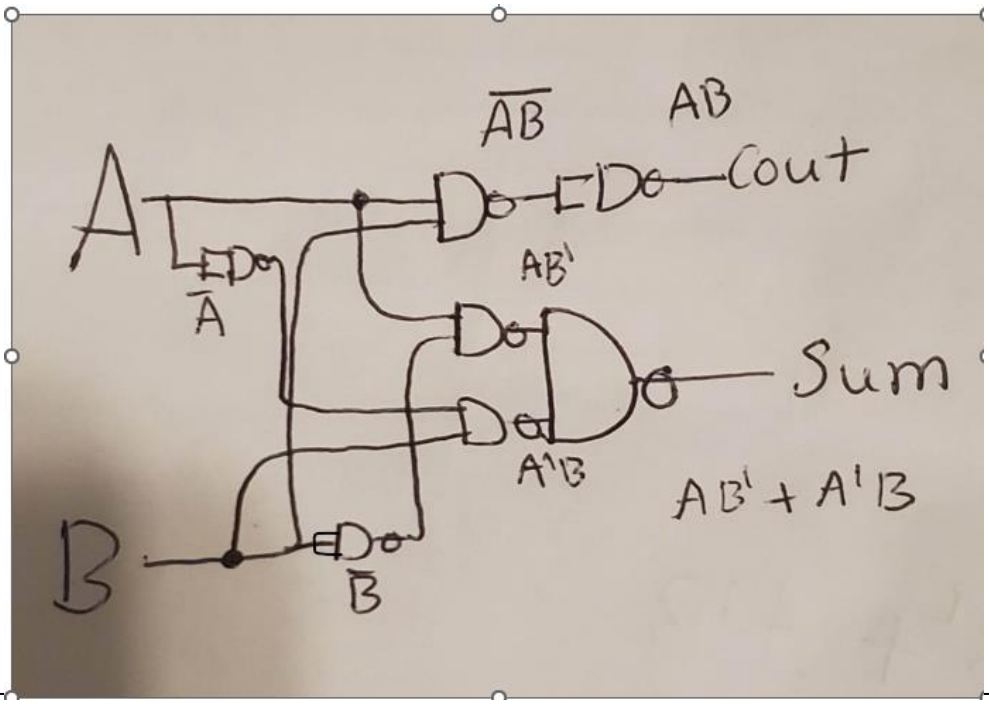

**Truth Table**




**K-maps and Simplest Sum of Products**




**Drawing Simplest NAND Circuit equivalent the Simplest Sum of Products**

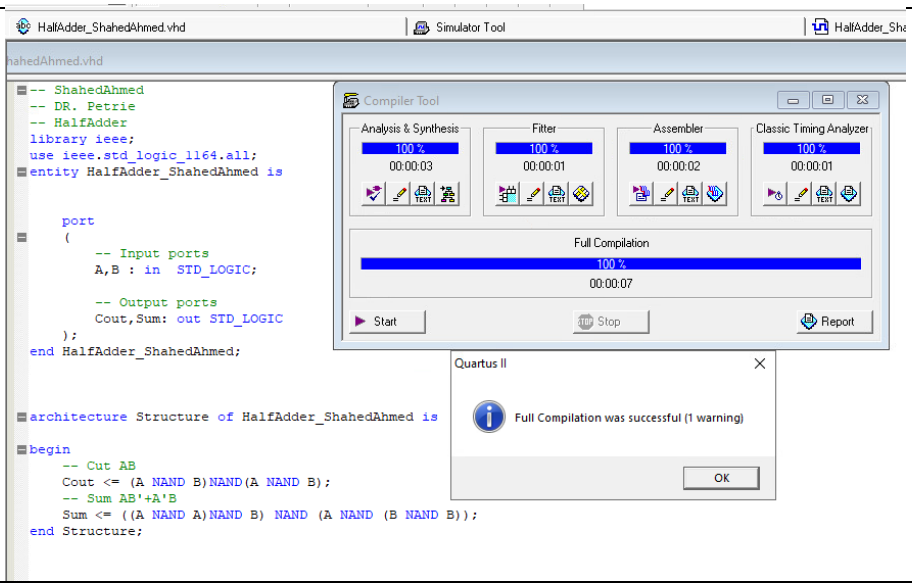


## VHDL Code

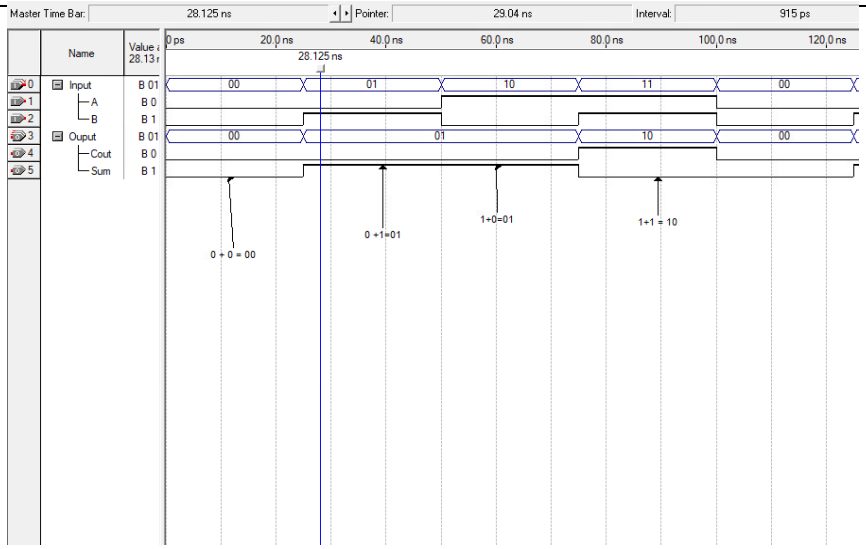
HalfAdder_ShahedAhmed.vhd | Simulator Tool

```
1  -- ShahedAhmed
2  -- DR. Petrie
3  -- HalfAdder
4  library ieee;
5  use ieee.std_logic_1164.all;
6  entity HalfAdder_ShahedAhmed is
7
8
9      port
10     (
11         -- Input ports
12         A,B : in  STD_LOGIC;
13
14         -- Output ports
15         Cout,Sum: out STD_LOGIC
16     );
17 end HalfAdder_ShahedAhmed;
18
19
20
21 architecture Structure of HalfAdder_ShahedAhmed is
22
23 begin
24     -- Cut AB
25     Cout <= (A NAND B)NAND(A NAND B);
26     -- Sum AB'+A'B
27     Sum <= ((A NAND A)NAND B) NAND (A NAND (B NAND B));
28 end Structure;
29
```

## Successful Compilation

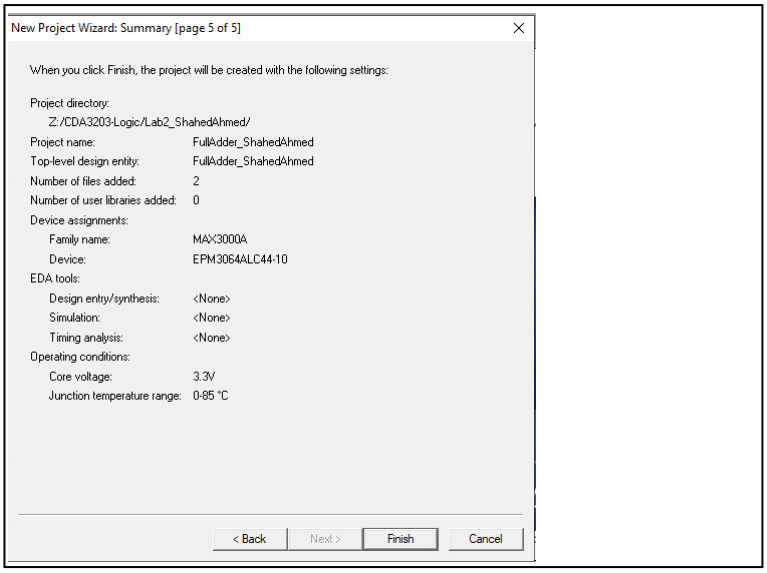



## Timing Diagram

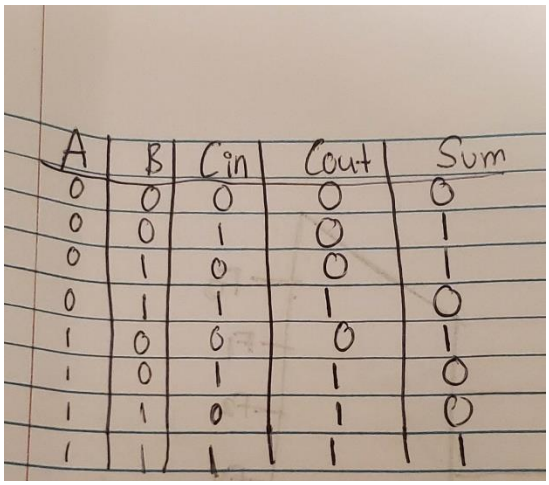


**Part 2:** A 1-bit Full Adder to add 2 binary bits (A, B) and a 1-bit Carry-in (Cin) and results in a 1-bit Sum and 1-bit Carry-out (Cout) with only NAND gates.

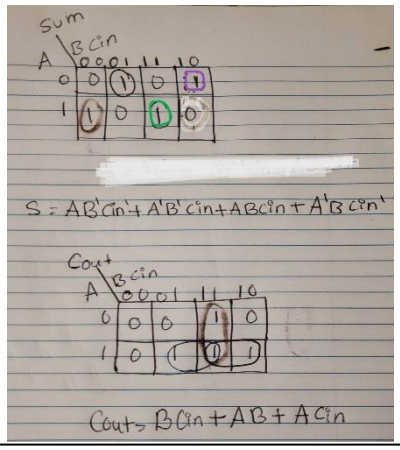
### Project Settings Snip



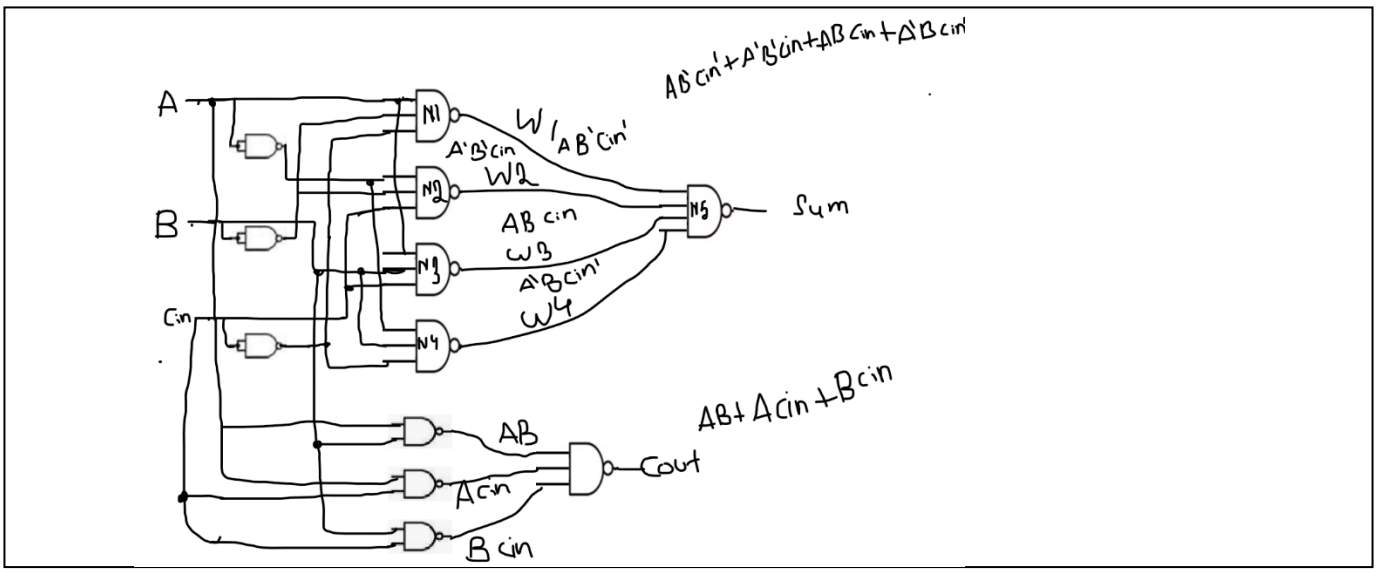

### Truth Table




### K-maps and Simplest Sum of Products




### Drawing Simplest NAND Circuit equivalent the Simplest Sum of Products



## VHDL Code

FullAdder_ShahedAhmed.vhd | Simulator Tool

```
1  -- ShahedAhmed -- DR.petrie -- FullAdder With NAND3 and NAAND 4
2  library ieee;
3  use ieee.std_logic_1164.all;
4  entity FullAdder_ShahedAhmed is
5
6      port
7      (   -- Input ports
8          A,B,Cin : in STD_LOGIC;
9          -- Output ports
10         Cout,Sum: out STD_LOGIC
11 );
12 end FullAdder_ShahedAhmed;
13
14 architecture Structure of FullAdder_ShahedAhmed is
15 SIGNAL W1,W2,W3,W4 : STD_LOGIC;
16
17 COMPONENT NAND3_ShahedAhmed is
18     port
19     (A,B,C  : in  STD_LOGIC;
20         x       : out STD_LOGIC);
21 end COMPONENT NAND3_ShahedAhmed ;
22 COMPONENT NAND4_ShahedAhmed is
23     port
24     (A,B,C,D    : in  STD_LOGIC;
25         x       : out STD_LOGIC);
26 end COMPONENT NAND4_ShahedAhmed;
27 begin
28 --- sum
29 N1: NAND3_ShahedAhmed PORT MAP (A,(B NAND B), (Cin NAND Cin),W1);
30 N2: NAND3_ShahedAhmed PORT MAP ((A NAND A),(B NAND B),Cin,W2);
31 N3: NAND3_ShahedAhmed PORT MAP (A,B,Cin,W3);
32 N4: NAND3_ShahedAhmed PORT MAP ((A NAND A),B, (Cin NAND Cin),W4);
33 N5: NAND4_ShahedAhmed PORT MAP (W1,W2,W3,W4,Sum);
34 -- cout
35 C1:NAND3_ShahedAhmed PORT MAP (A NAND B,A NAND Cin,B NAND Cin,Cout) ;
36 end Structure;
```

## Successful Compilation

FullAdder_ShahedAhmed.vhd | Compiler Tool

```
1  -- ShahedAhmed -- DR.petrie -- FullAdder With NAND3 and NAAND 4
2  library ieee;
3  use ieee.std_logic_1164.all;
4  entity FullAdder_ShahedAhmed is
5
6      port
7      (   -- Input ports
8          A,B,Cin : in STD_LOGIC;
9          -- Output ports
10         Cout,Sum: out STD_LOGIC
11 );
12 end FullAdder_ShahedAhmed;
13
14 architecture Structure of FullAdder_ShahedAhmed is
15 SIGNAL W1,W2,W3,W4 : STD_LOGIC;
16
17 COMPONENT NAND3_ShahedAhmed is
18     port
19     (A,B,C  : in  STD_LOGIC;
20         x       : out STD_LOGIC);
21 end COMPONENT NAND3_ShahedAhmed ;
22 COMPONENT NAND4_ShahedAhmed is
23     port
24     (A,B,C,D    : in  STD_LOGIC;
25         x       : out STD_LOGIC);
26 end COMPONENT NAND4_ShahedAhmed;
27 begin
28 --- sum
29 N1: NAND3_ShahedAhmed PORT MAP (A,(B NAND B), (Cin NAND Cin),W1);
30 N2: NAND3_ShahedAhmed PORT MAP ((A NAND A),(B NAND B),Cin,W2);
31 N3: NAND3_ShahedAhmed PORT MAP (A,B,Cin,W3);
32 N4: NAND3_ShahedAhmed PORT MAP ((A NAND A),B, (Cin NAND Cin),W4);
33 N5: NAND4_ShahedAhmed PORT MAP (W1,W2,W3,W4,Sum);
34 -- cout
35 C1:NAND3_ShahedAhmed PORT MAP (A NAND B,A NAND Cin,B NAND Cin,Cout) ;
36 end Structure;
```

Compiler Tool

Analysis & Synthesis 100 % 00:00:04 | Fitter 100 % 00:00:02 | Assembler 100 % 00:00:02 | Classic Timing Analyzer 100 % 00:00:01

Full Compilation 100 % 00:00:09

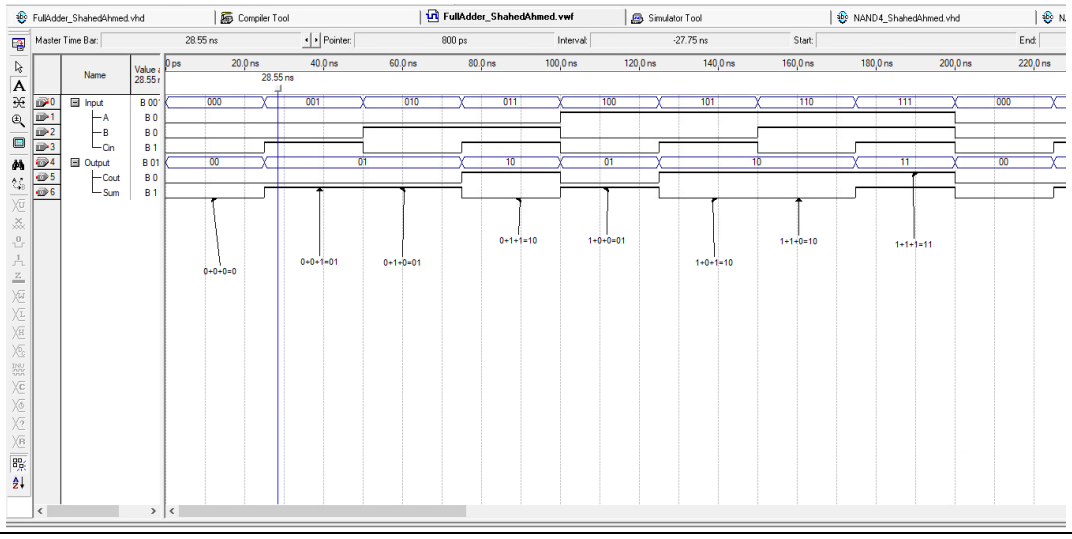
Start | Stop | Report

Quartus II

Full Compilation was successful (1 warning)

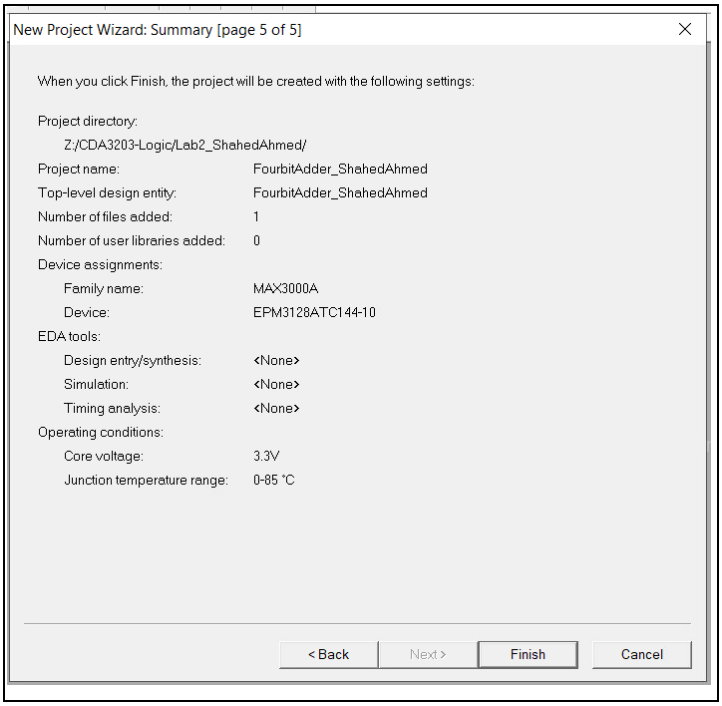
OK

## Timing Diagram

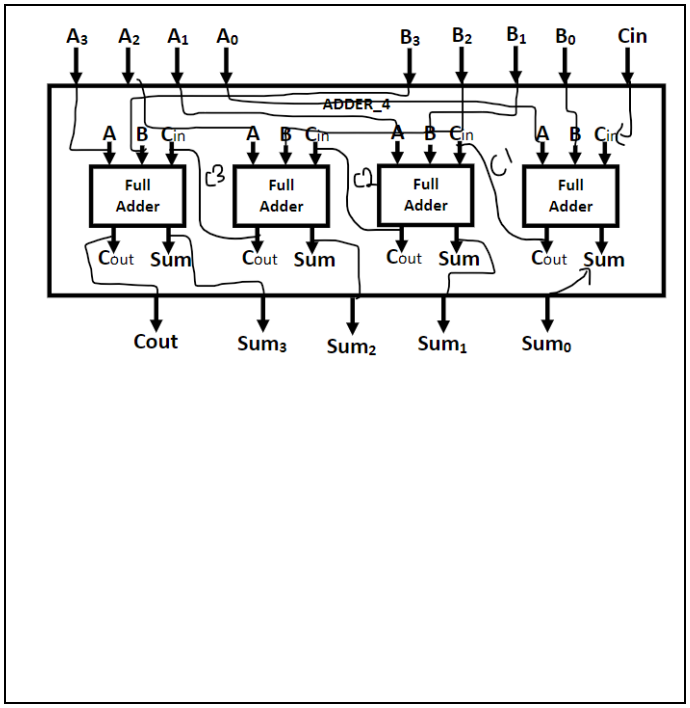


**Part 3:** A 4-bit Adder to add 2 4 bit binary numbers (A3,A2,A1,A0 and B3,B2,B1,B0 with A0 and B0 being least significant bits) and a 1-bit Carry-in (Cin), and results in a 4-bit Sum (S3,S2,S1,S0) and 1-bit Carry-out (Cout) with the 1-bit Full Adder component you built.

### Project Settings Snip




### Draw of your circuit

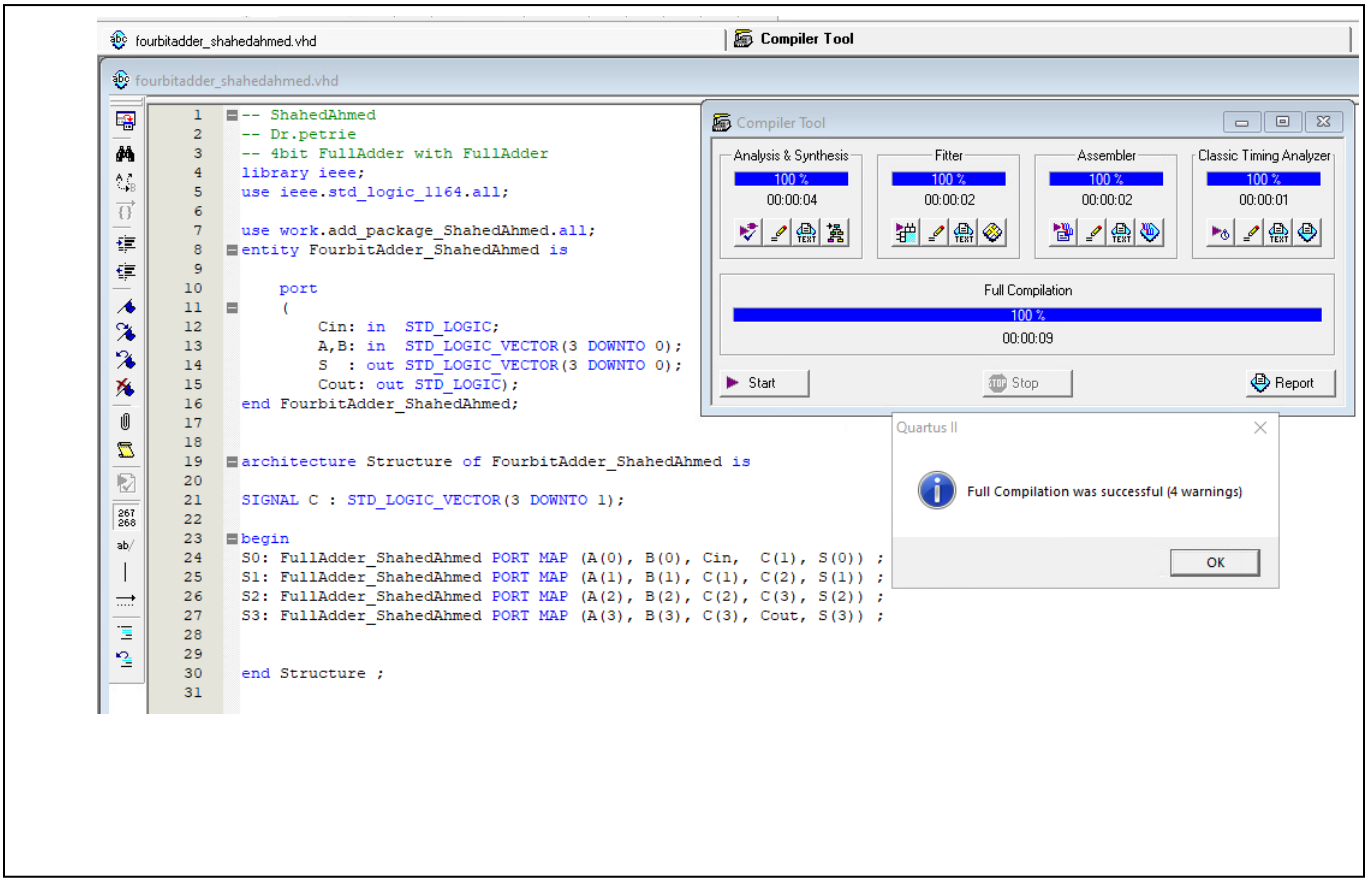



### VHDL Code

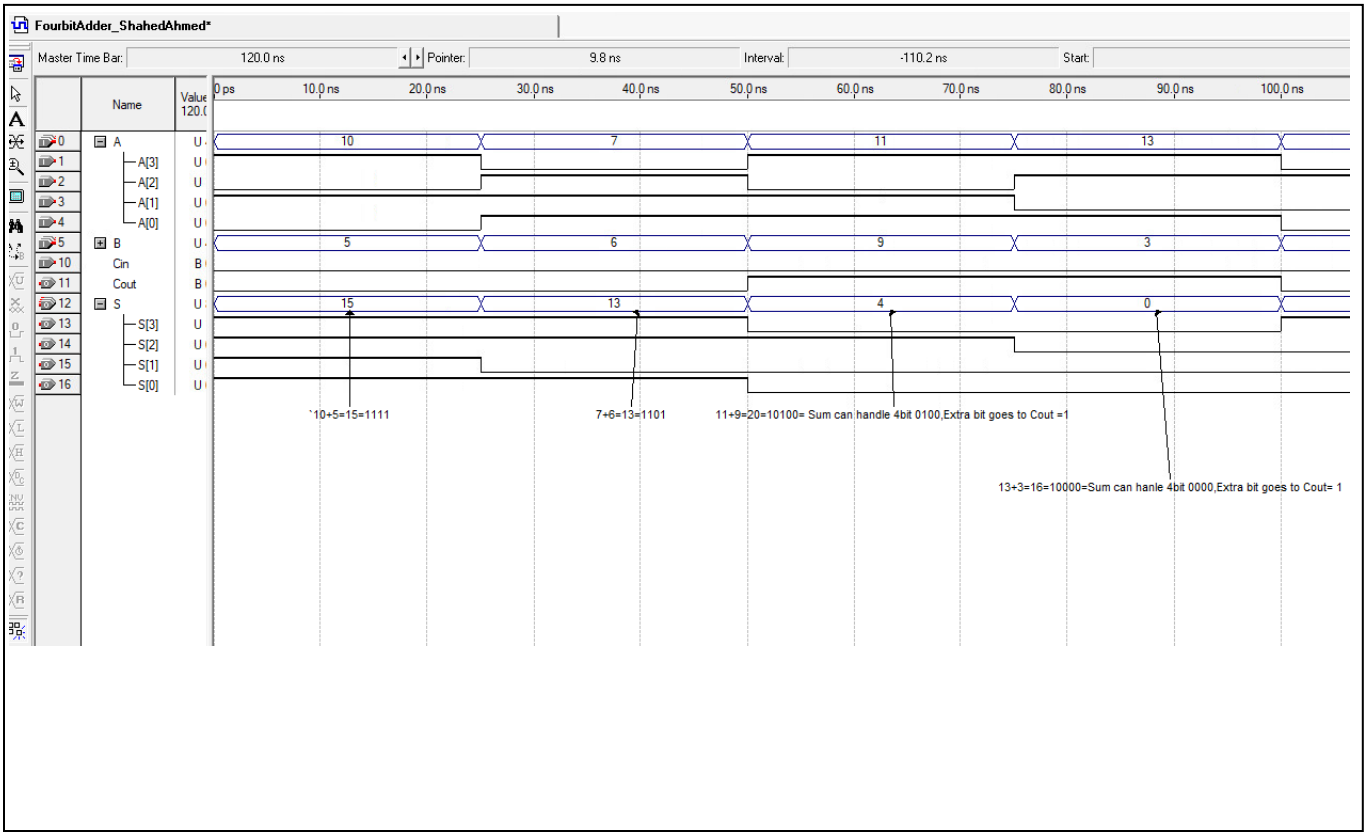
fourbitadder_shahedahmed.vhd

```
1   -- ShahedAhmed
2   -- Dr.petrie
3   -- 4bit FullAdder with FullAdder
4   library ieee;
5   use ieee.std_logic_1164.all;
6
7   use work.add_package_ShahedAhmed.all;
8   entity FourbitAdder_ShahedAhmed is
9
10      port
11      (
12          Cin: in  STD_LOGIC;
13          A,B: in  STD_LOGIC_VECTOR(3 DOWNTO 0);
14          S  : out STD_LOGIC_VECTOR(3 DOWNTO 0);
15          Cout: out STD_LOGIC);
16  end FourbitAdder_ShahedAhmed;
17
18
19  architecture Structure of FourbitAdder_ShahedAhmed is
20
21  SIGNAL C : STD_LOGIC_VECTOR(3 DOWNTO 1);
22
23  begin
24  S0: FullAdder_ShahedAhmed PORT MAP (A(0), B(0), Cin,  C(1), S(0)) ;
25  S1: FullAdder_ShahedAhmed PORT MAP (A(1), B(1), C(1), C(2), S(1)) ;
26  S2: FullAdder_ShahedAhmed PORT MAP (A(2), B(2), C(2), C(3), S(2)) ;
27  S3: FullAdder_ShahedAhmed PORT MAP (A(3), B(3), C(3), Cout, S(3)) ;
28
29
30  end Structure ;
```

## Successful Compilation

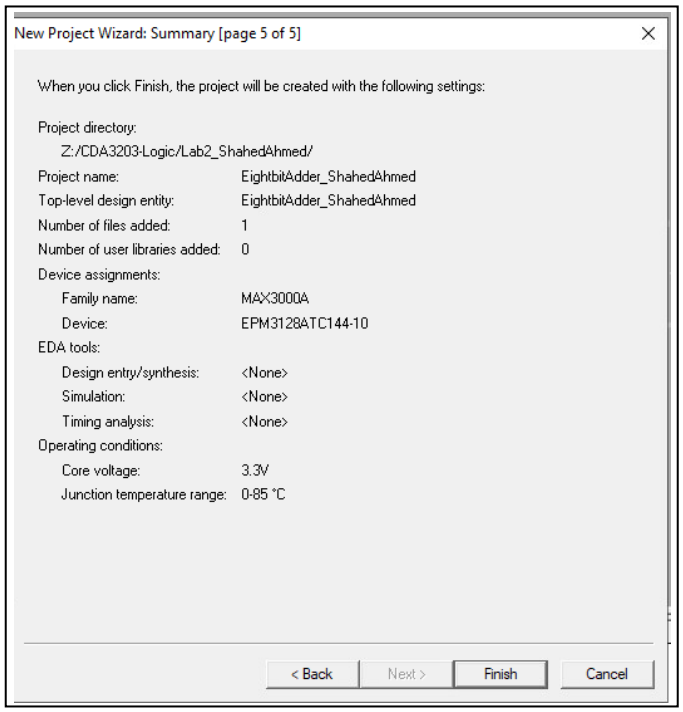



## Timing Diagram

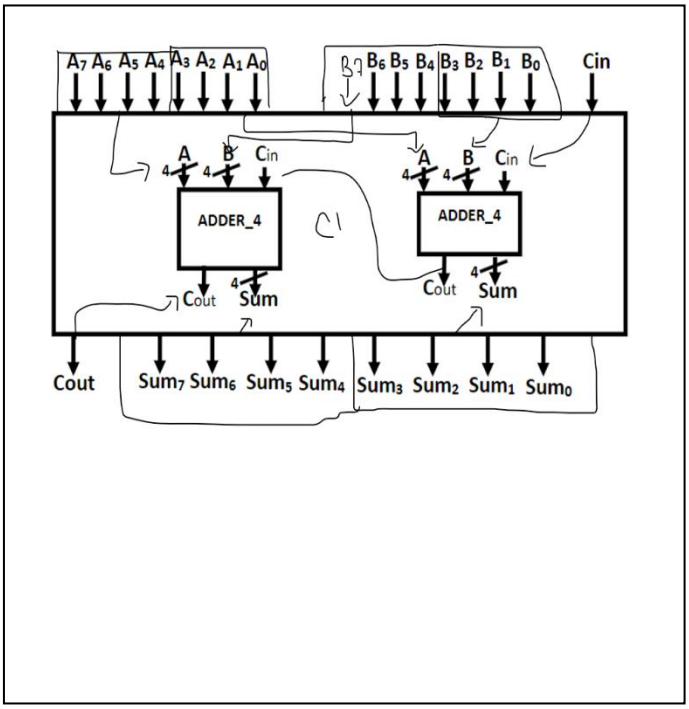


**Part 4:** A 8-bit Adder to add 2 8-bit binary numbers (A7,A6,A5,A4,A3,A2,A1,A0 and B7,B6,B5,B4,B3,B2,B1,B0 with A0 and B0 being least significant bits) and a 1-bit Carry-in (Cin), and results in a 4-bit Sum (S7,S6,S5,S4,S3,S2,S1,S0) and 1-bit Carry-out (Cout) with the 1-bit Full Adder component you built.

**Project Settings Snip**




**Draw your circuit HERE**

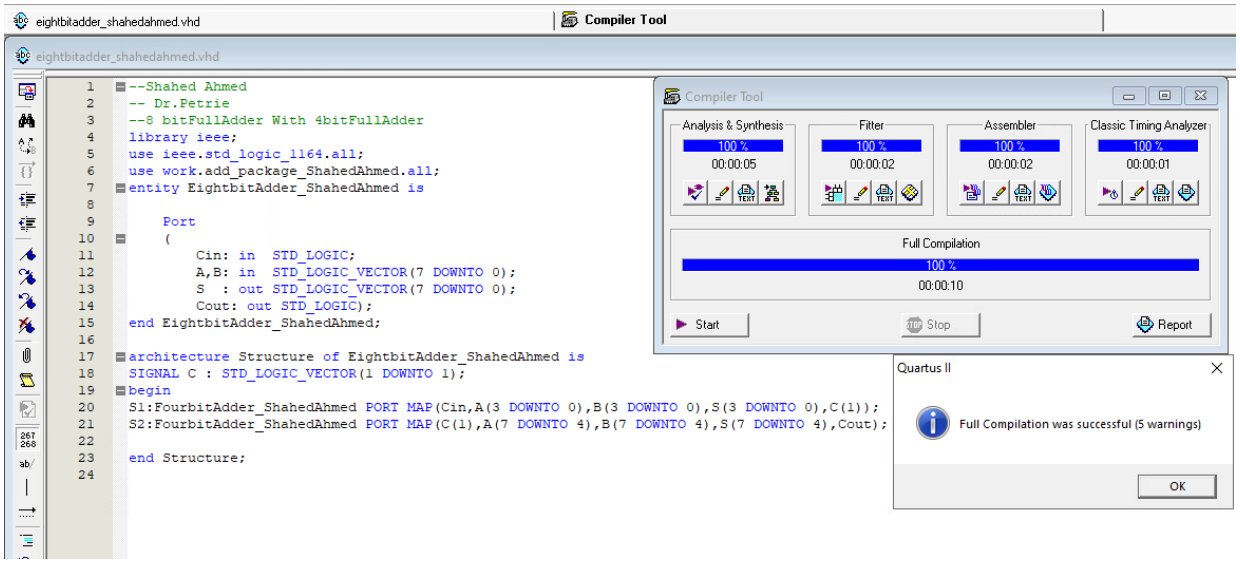



**VHDL Code**

eightbitadder_shahedahmed.vhd*

```
1  --Shahed Ahmed
2  -- Dr.Petrie
3  --8 bitFullAdder With 4bitFullAdder
4  library ieee;
5  use ieee.std_logic_1164.all;
6  use work.add_package_ShahedAhmed.all;
7  entity EightbitAdder_ShahedAhmed is
8
9      Port
10     (
11         Cin: in  STD_LOGIC;
12         A,B: in  STD_LOGIC_VECTOR(7 DOWNTO 0);
13         S  : out STD_LOGIC_VECTOR(7 DOWNTO 0);
14         Cout: out STD_LOGIC);
15 end EightbitAdder_ShahedAhmed;
16
17 architecture Structure of EightbitAdder_ShahedAhmed is
18 SIGNAL C : STD_LOGIC_VECTOR(1 DOWNTO 1);
19 begin
20 S1:FourbitAdder_ShahedAhmed PORT MAP(Cin,A(3 DOWNTO 0),B(3 DOWNTO 0),S(3 DOWNTO 0),C(1));
21 S2:FourbitAdder_ShahedAhmed PORT MAP(C(1),A(7 DOWNTO 4),B(7 DOWNTO 4),S(7 DOWNTO 4),Cout);
22
23 end Structure;
24
```

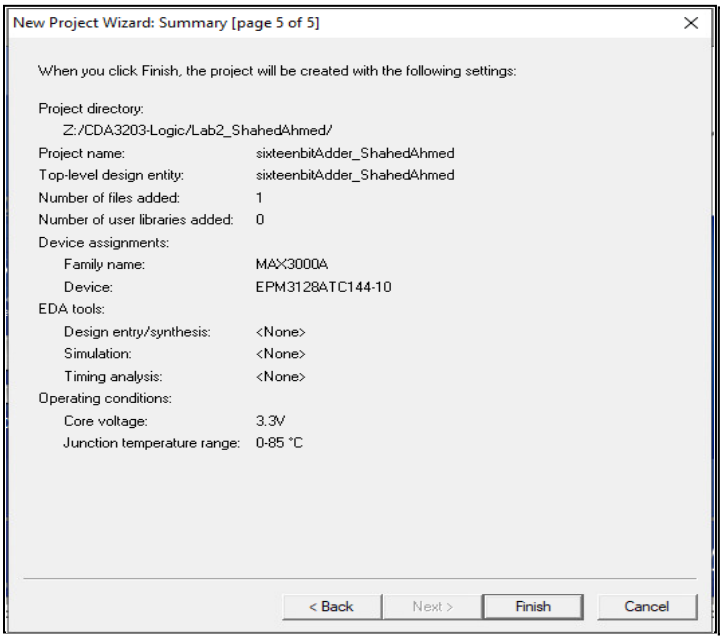
## Successful Compilation



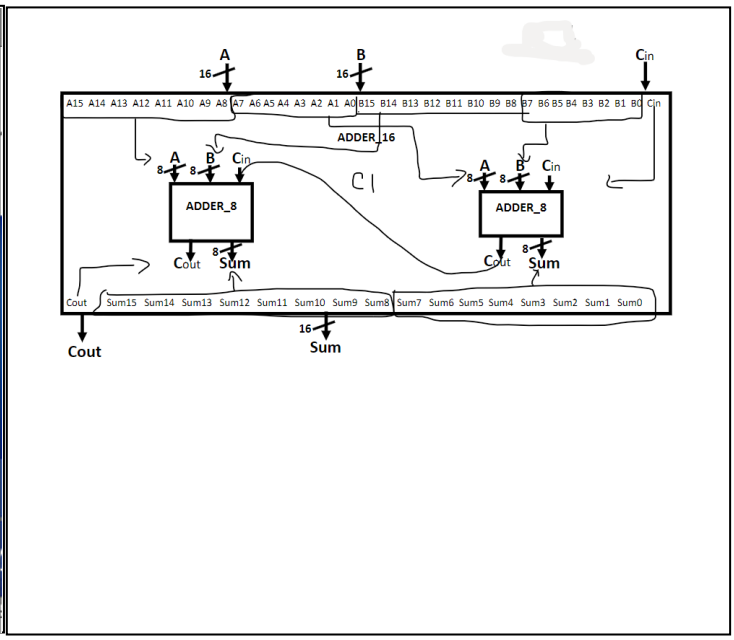

## Timing Diagram

**Part 5:** A 16-bit Adder to add 2 16-bit binary numbers(A15,A14,A13,A12,A11,A10,A9,A8,A7,A6,A5,A4,A3,A2,A1,A0 and B15,B14,B13,B12,B11,B10,B9,B8,B7,B6,B5,B4,B3,B2,B1,B0 with A0 and B0 being least significant bits) and a 1-bit Carry-in (Cin), and results in a 16-bit Sum (S15,S14,S13,S12,S11,S10,S9,S8,S7,S6,S5,S4,S3,S2,S1, S0) and 1-bit Carry-out (Cout) with component you built

### Project Settings Snip




### Draw your circuit Here



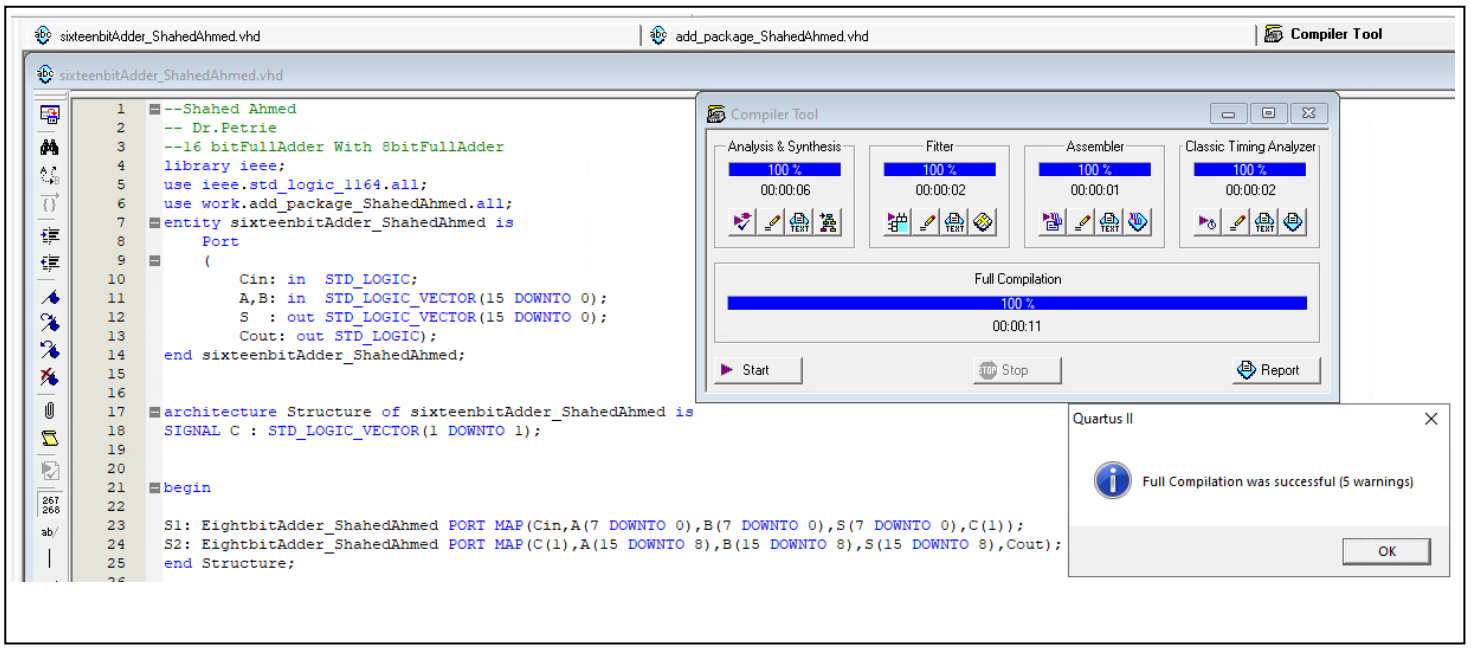

### VHDL Code

sixteenbitAdder_ShahedAhmed.vhd | add_package_ShahedAhmed.vhd

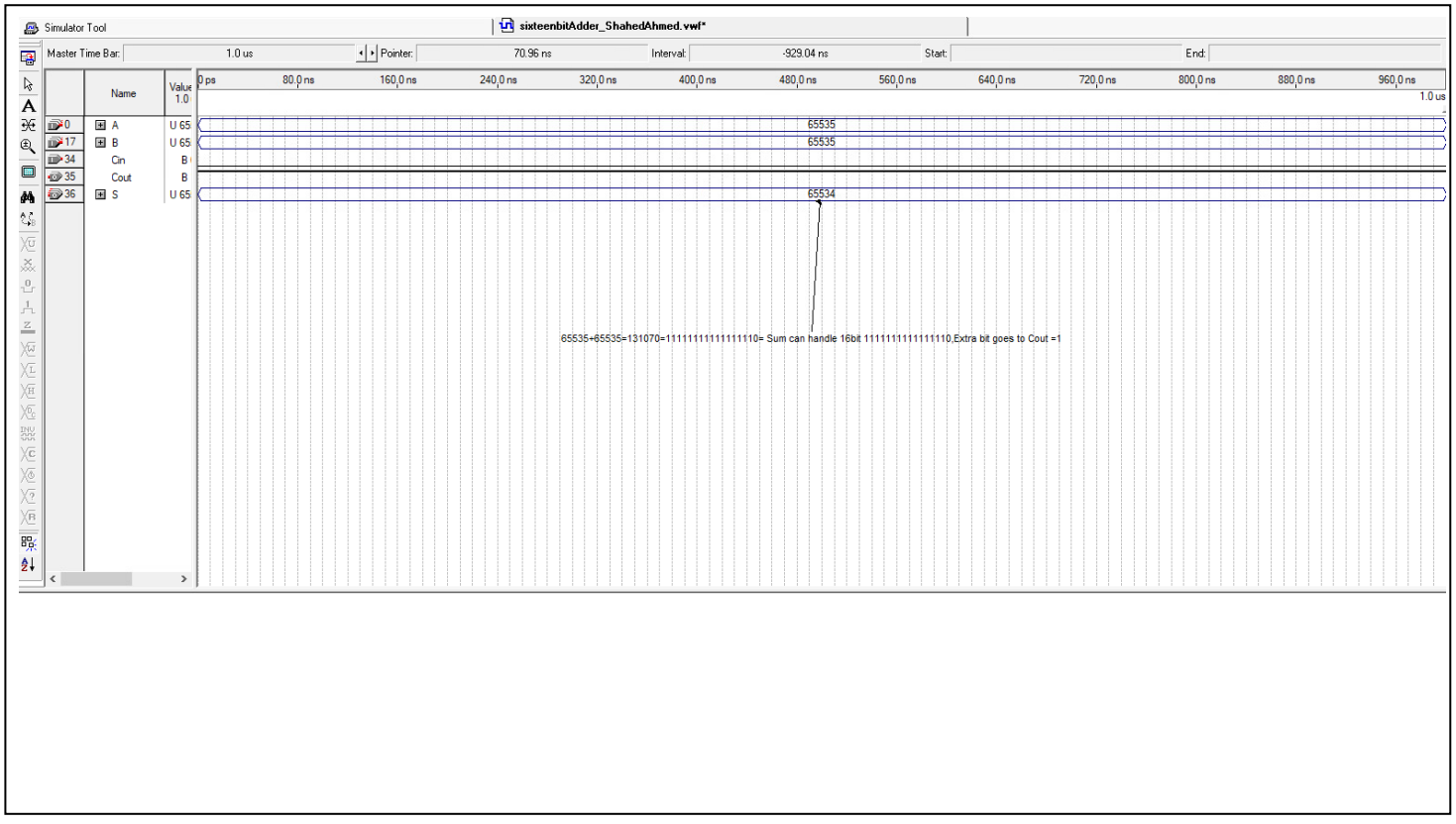
sixteenbitAdder_ShahedAhmed.vhd

```
1  --Shahed Ahmed
2  -- Dr.Petrie
3  --16 bitFullAdder With 8bitFullAdder
4  library ieee;
5  use ieee.std_logic_1164.all;
6  use work.add_package_ShahedAhmed.all;
7  entity sixteenbitAdder_ShahedAhmed is
8      Port
9      (
10         Cin: in  STD_LOGIC;
11         A,B: in  STD_LOGIC_VECTOR(15 DOWNTO 0);
12         S  : out STD_LOGIC_VECTOR(15 DOWNTO 0);
13         Cout: out STD_LOGIC);
14 end sixteenbitAdder_ShahedAhmed;
15
16
17 architecture Structure of sixteenbitAdder_ShahedAhmed is
18 SIGNAL C : STD_LOGIC_VECTOR(1 DOWNTO 1);
19
20
21 begin
22
23 S1: EightbitAdder_ShahedAhmed PORT MAP(Cin,A(7 DOWNTO 0),B(7 DOWNTO 0),S(7 DOWNTO 0),C(1));
24 S2: EightbitAdder_ShahedAhmed PORT MAP(C(1),A(15 DOWNTO 8),B(15 DOWNTO 8),S(15 DOWNTO 8),Cout);
25 end Structure;
26
```

## Successful Compilation




## Timing Diagram



# Extra Credit

Package:

sixteenbitAdder_ShahedAhmed.vhd | add_package_ShahedAhmed.vhd

```
1  LIBRARY ieee ;
2  USE ieee.std_logic_1164.all ;
3
4  PACKAGE add_package_ShahedAhmed IS
5  COMPONENT FullAdder_ShahedAhmed is
6  port
7      (
8          A,B,Cin : in STD_LOGIC;
9          Cout,Sum: out STD_LOGIC
10 );
11 end COMPONENT FullAdder_ShahedAhmed;
12 COMPONENT FourbitAdder_ShahedAhmed is
13
14     port
15     (   Cin: in  STD_LOGIC;
16         A,B: in  STD_LOGIC_VECTOR(3 DOWNTO 0);
17         S  : out STD_LOGIC_VECTOR(3 DOWNTO 0);
18         Cout: out STD_LOGIC);
19 end COMPONENT FourbitAdder_ShahedAhmed;
20     COMPONENT EightbitAdder_ShahedAhmed is
21
22     Port
23     (
24         Cin: in  STD_LOGIC;
25         A,B: in  STD_LOGIC_VECTOR(7 DOWNTO 0);
26         S  : out STD_LOGIC_VECTOR(7 DOWNTO 0);
27         Cout: out STD_LOGIC);
28 end COMPONENT EightbitAdder_ShahedAhmed;
29
30 END add_package_ShahedAhmed ;
```

NAND 3:

NAND3_ShahedAhmed.vhd

```
1  -- A library clause declares a name as a library.  It
2  -- does not create the library; it simply forward declares
3  -- it.
4  library ieee;
5  use ieee.std_logic_1164.all;
6
7  entity NAND3_ShahedAhmed is
8
9      port
10     (
11         -- Input ports
12         A,B,C   : in  STD_LOGIC;
13         -- Output ports
14         x       : out STD_LOGIC
15     );
16 end;
17
18 architecture Structure of NAND3_ShahedAhmed is
19
20
21 begin
22 -- fix
23 x <= (( A NAND B) NAND (A NAND B)) NAND C;
24 end;
```

NAND4:

NAND4_ShahedAhmed.vhd

```
1
2   library ieee;
3   use ieee.std_logic_1164.all;
4
5   entity NAND4_ShahedAhmed is
6
7       port
8       (
9           -- Input ports
10          A,B,C,D : in  STD_LOGIC;
11          -- Output ports
12          x       : out STD_LOGIC
13      );
14  end;
15
16  architecture Structure of NAND4_ShahedAhmed is
17
18
19  begin
20
21  x <= ((A NAND B) NAND (A NAND B)) NAND ((C NAND D) NAND (C NAND D));
22
23  end;
```

display all group values in decimal: check timing diagram 4bit  to 16bit

TD_LOGIC_VECTOR : Check 4 bit to 16 bit

Extra credit + if submit before Monday said by Dr.petrie