Lab 2

Sophie Samuels

10/14/2023

CDA3203 Computer Logic Design

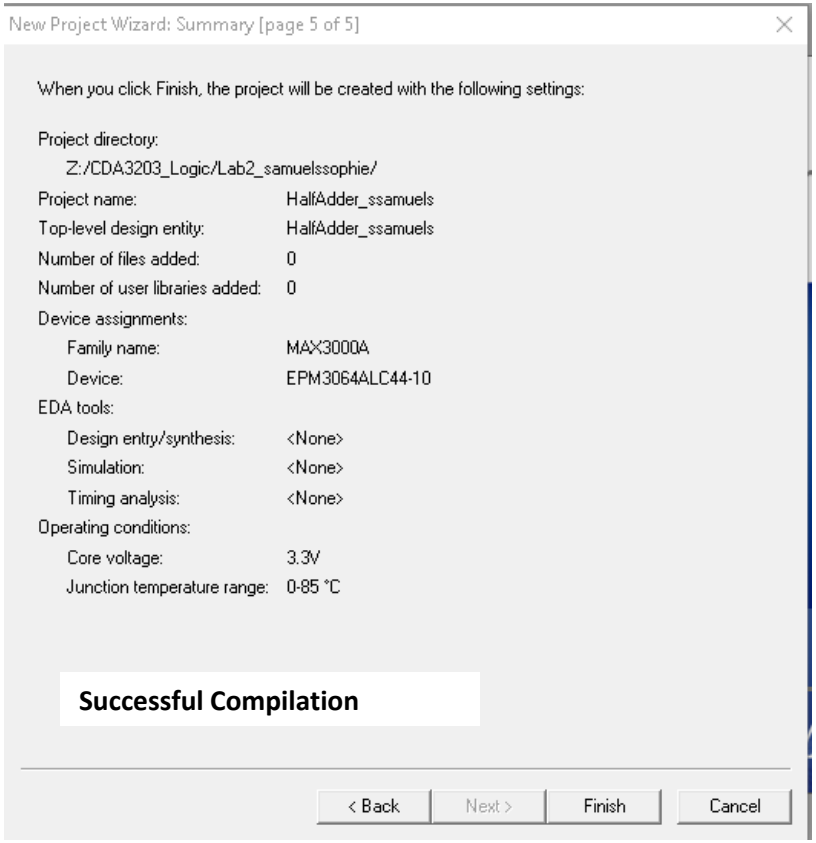
Fall 2023

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.

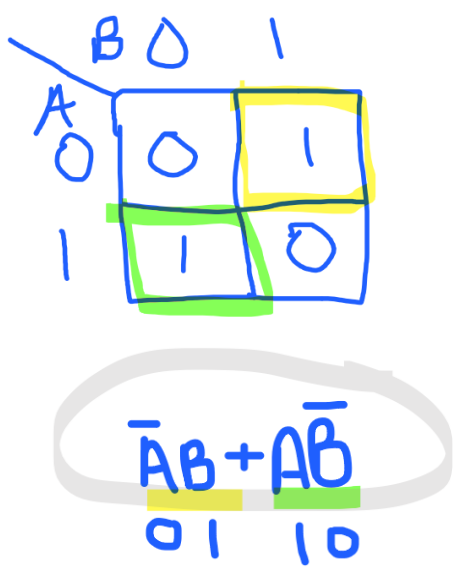
Project Settings Snip



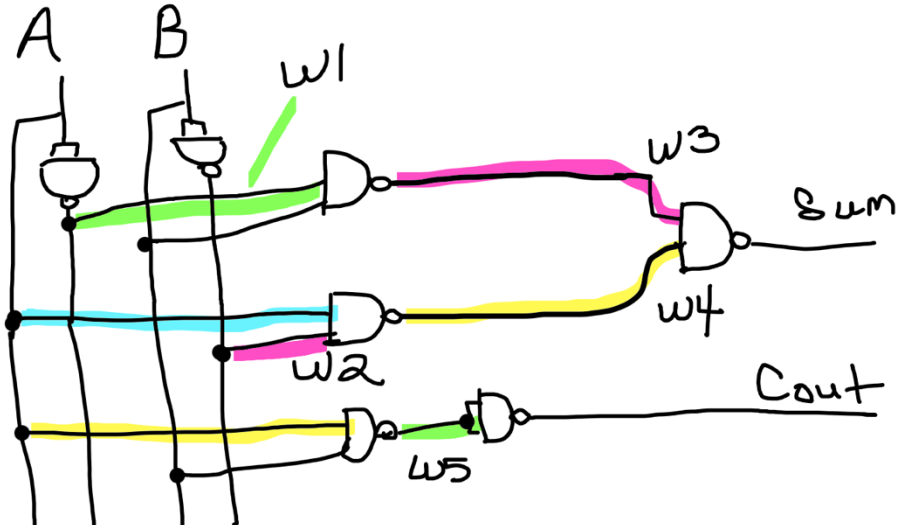
Truth Table

Inputs		Calculate A plus B in decimal	Outputs A plus B in binary	
A	B		C_{out}	Sum
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	2	1	0

K-maps and Simplest Sum of Products



Drawing Simplest NAND Circuit equivalent the Simplest Sum of Products

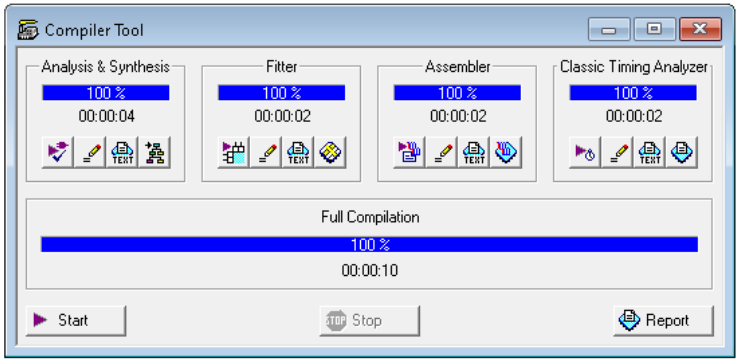


VHDL Code

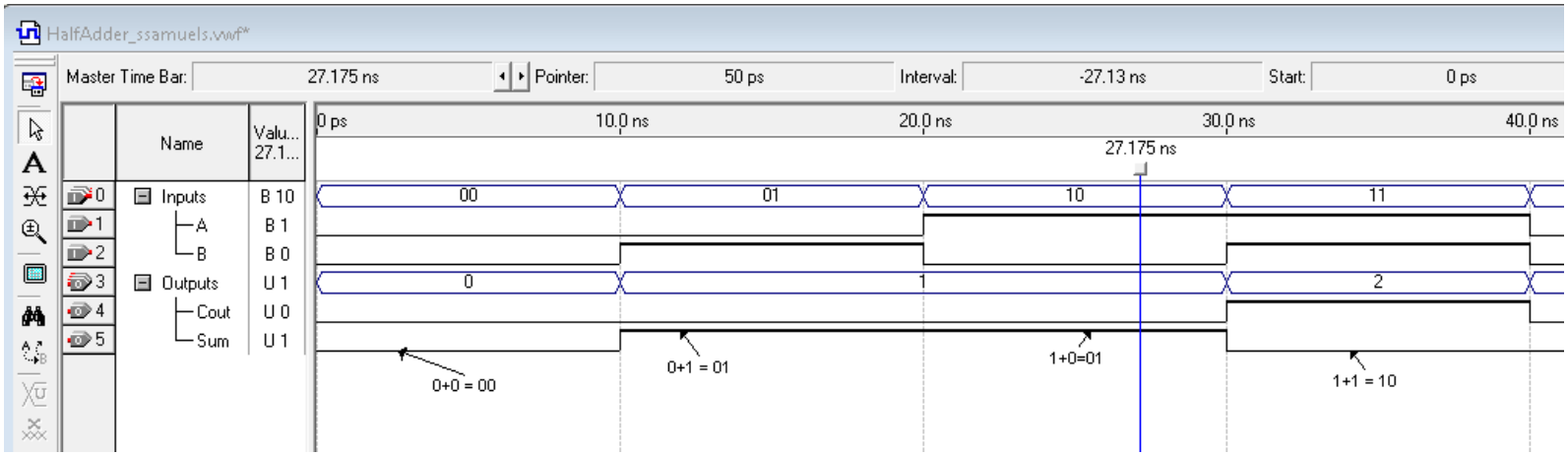
HalfAdder_ssamuels.vhd

```
1  -- Lab 2 Half Adder by Sophie Samuels
2
3  library IEEE;
4  use ieee.std_logic_1164.all;
5
6  entity HalfAdder_ssamuels is
7      port
8      (
9          --Input ports
10         A, B        : in  STD_LOGIC;
11
12         -- Output ports
13         Cout, Sum   : out STD_LOGIC
14     );
15 end HalfAdder_ssamuels;
16
17 architecture Structure of HalfAdder_ssamuels is
18
19     SIGNAL w1, w2, w3, w4, w5: STD_LOGIC;
20
21 begin
22
23     w1   <=  A NAND A  ; --A'
24     w2   <=  B NAND B  ; --B'
25     w3   <= w1 NAND B  ; --(A'B)'
26     w4   <=  A NAND w2 ; --(AB')'
27     w5   <=  A NAND B  ; --(AB)'
28     Sum  <= w3 NAND w4 ; --((A'B)'(AB')')'= AB + AB DeMorgan's Law
29     Cout <= w5 NAND w5 ; --(AB)'' = AB
30
31 end Structure;
```

Successful Compilation

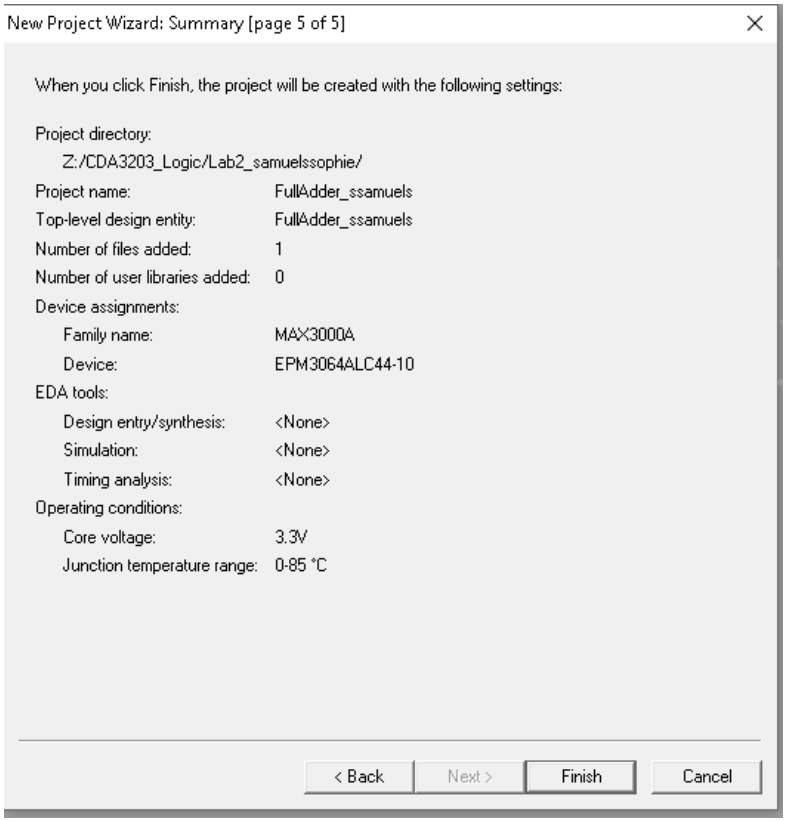


Timing Diagram



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.

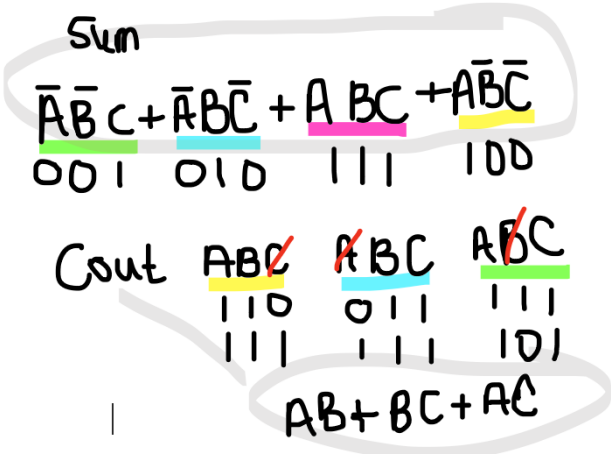
Project Settings Snip



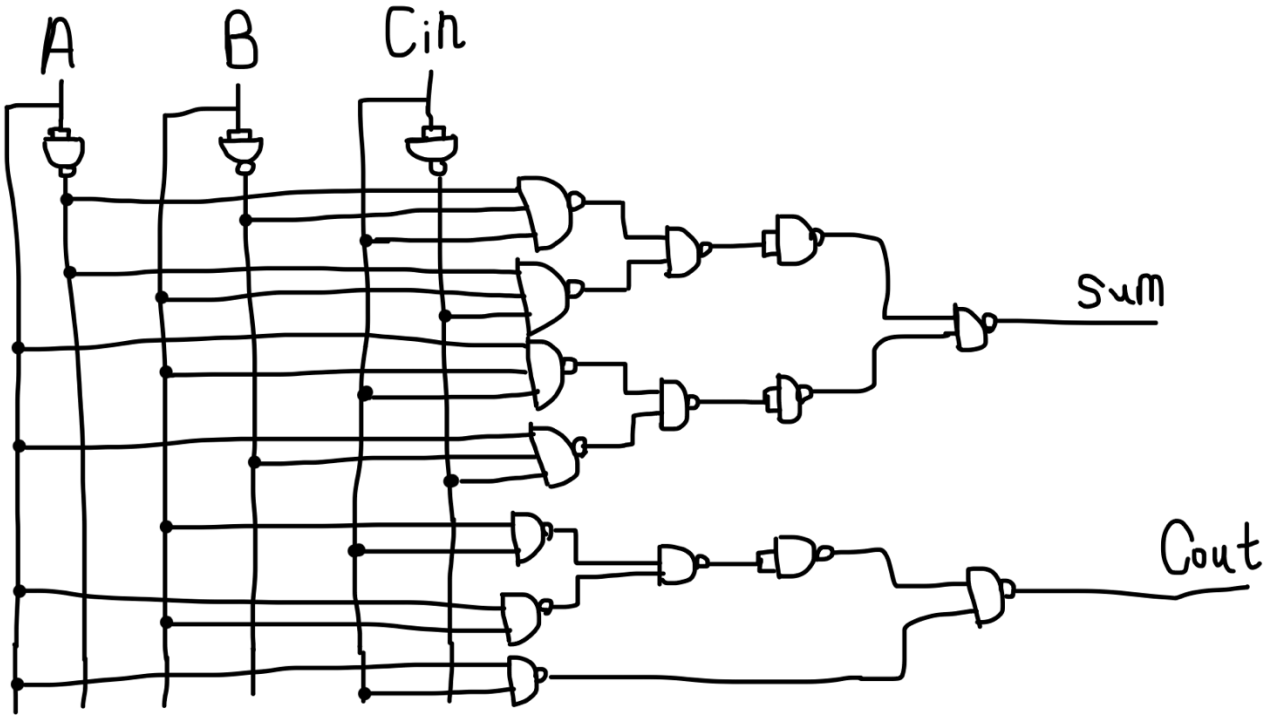
Truth Table

Inputs A	B	C_{in}	Calculate sum of A, B, C_{in} in decimal	Outputs in binary $C_{out}4$	Sum4
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	2	1	0
1	0	0	1	0	1
1	0	1	2	1	0
1	1	0	2	1	0
1	1	1	3	1	1

K-maps and Simplest Sum of Products



Drawing Simplest NAND Circuit equivalent the Simplest Sum of Products

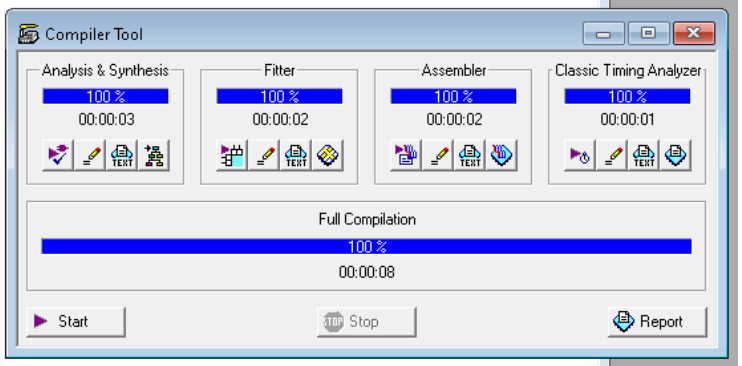


VHDL Code

FullAdder_ssamuels.vhd

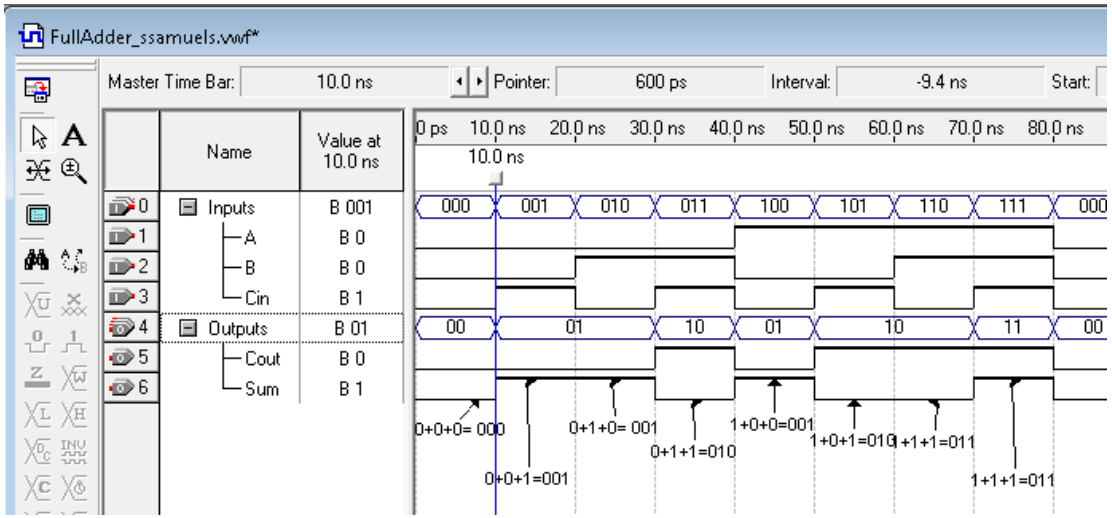
```
1  --Full Adder using NAND Gates By Sophie Samuels
2
3  LIBRARY ieee;
4  USE ieee.std_logic_1164.all;
5
6  ENTITY FullAdder_ssamuels IS
7      PORT (A, B, Cin : IN STD_LOGIC;
8            Cout, Sum : OUT STD_LOGIC);
9  END FullAdder_ssamuels;
10
11 ARCHITECTURE Structure OF FullAdder_ssamuels IS
12
13 SIGNAL temp1, temp2, temp3, temp4 : STD_LOGIC;
14 COMPONENT NAND3_ssamuels IS
15     PORT ( A, B, C : IN STD_LOGIC;
16            Y       : OUT STD_LOGIC);
17 END COMPONENT NAND3_ssamuels;
18
19 COMPONENT NAND4_ssamuels IS
20     PORT (A, B, C, D : IN STD_LOGIC;
21           Y          : OUT STD_LOGIC);
22 END COMPONENT NAND4_ssamuels;
23
24 BEGIN
25     --Cout = AB + ACin + Bin
26     n3 : NAND3_ssamuels PORT MAP (A NAND B, A NAND Cin, B NAND Cin, Cout);
27
28     --Sum = A'B'Cin + A'BCin' + ABCin + AB'Cin'
29     aNotbNotC : NAND3_ssamuels PORT MAP (A NAND A, B NAND B, Cin, temp1);
30     aNotBCNot : NAND3_ssamuels PORT MAP (A NAND A, B, Cin NAND Cin, temp2);
31     abc       : NAND3_ssamuels PORT MAP (A, B, Cin, temp3);
32     abNotcNot : NAND3_ssamuels PORT MAP (A, B NAND B, Cin NAND Cin, temp4);
33     Sum_temps : NAND4_ssamuels PORT MAP (temp1, temp2, temp3, temp4, Sum);
34 END Structure;
```

Successful Compilation



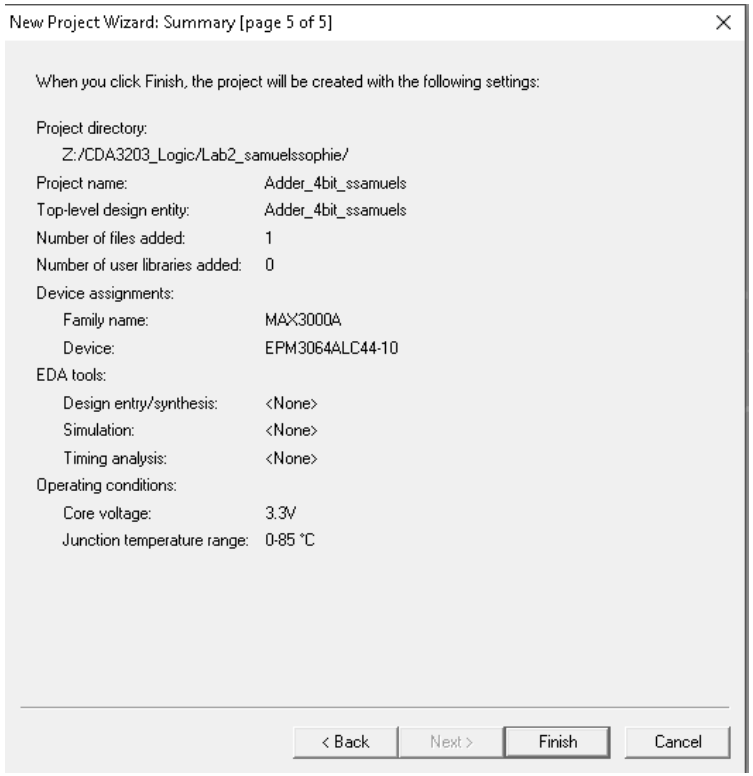
View Quartus II Information

Timing Diagram

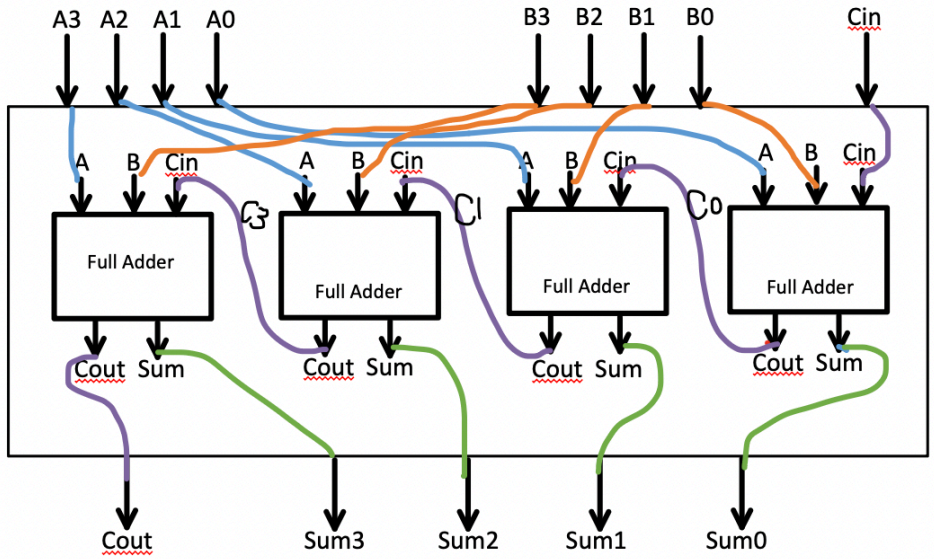


Part 3: A 4-bit Adder to add 2 4 bit binary numbers (A3,A2,A1,A0 and B3,B2,B1,B0 with A0 and B0 being least significant bits) and a 1-bit Carry-in (Cin), and results in a 4-bit Sum (S3,S2,S1,S0) and 1-bit Carry-out (Cout) with the 1-bit Full Adder component you built.

Project Settings Snip



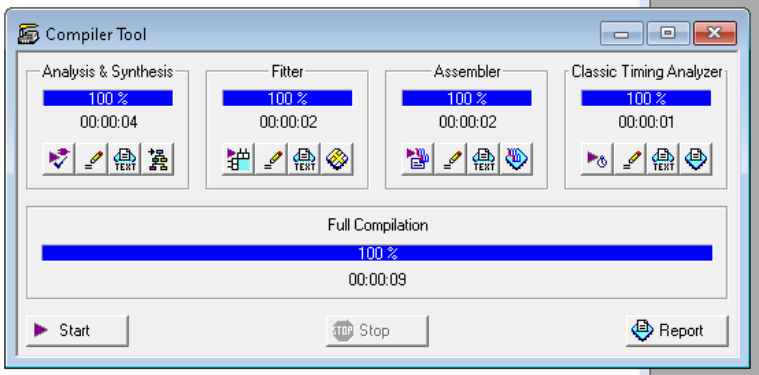
Draw of your circuit



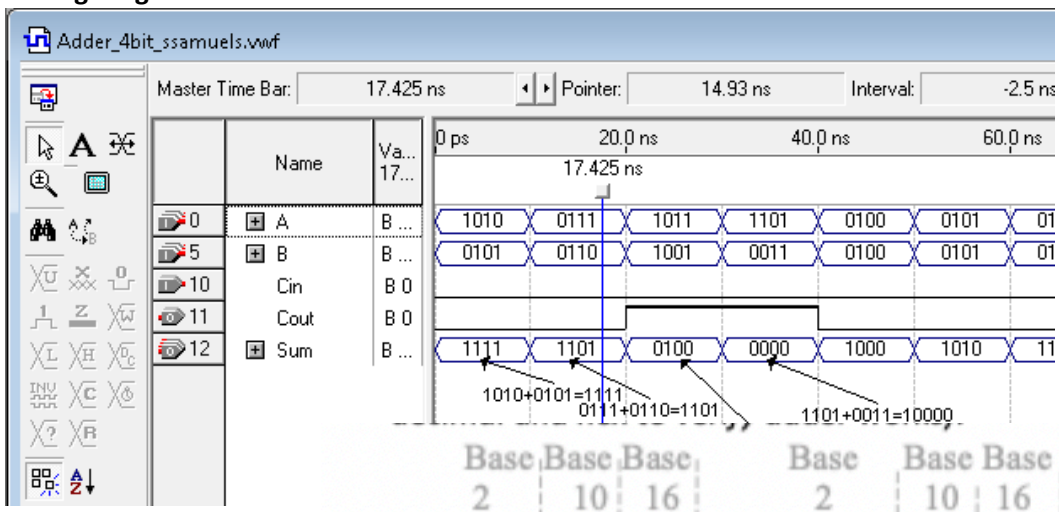
VHDL Code

```
--4-Bit Adder using Full Adder by Sophie Samuels

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY Adder_4bit_ssamuels IS
    PORT (A, B  : IN STD_LOGIC_VECTOR (3 downto 0);
          Cin   : IN STD_LOGIC;

          Cout  : OUT STD_LOGIC;
          Sum   : OUT STD_LOGIC_VECTOR (3 downto 0));
END Adder_4bit_ssamuels;

ARCHITECTURE Structure OF Adder_4bit_ssamuels IS

SIGNAL C0, C1, C2 : STD_LOGIC;

COMPONENT FullAdder_ssamuels IS
    PORT (A, B, Cin : IN STD_LOGIC;
          Cout, Sum : OUT STD_LOGIC);
END COMPONENT FullAdder_ssamuels;

BEGIN
fa0 : FullAdder_ssamuels PORT MAP (A(0), B(0), Cin, C0, Sum(0)) ;
fa1 : FullAdder_ssamuels PORT MAP (A(1), B(1), C0, C1, Sum(1))  ;
fa2 : FullAdder_ssamuels PORT MAP (A(2), B(2), C1, C2, Sum(2))  ;
fa3 : FullAdder_ssamuels PORT MAP (A(3), B(3), C2, Cout, Sum(3));
END Structure;
```

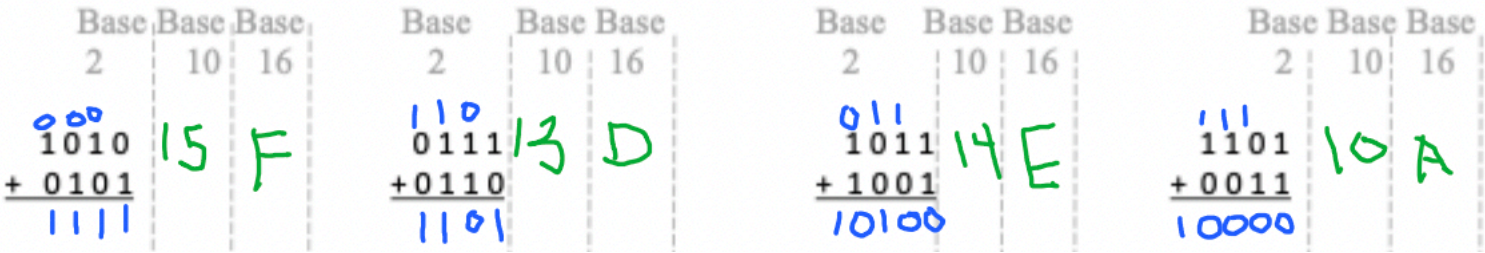
Successful Compilation



Timing Diagram

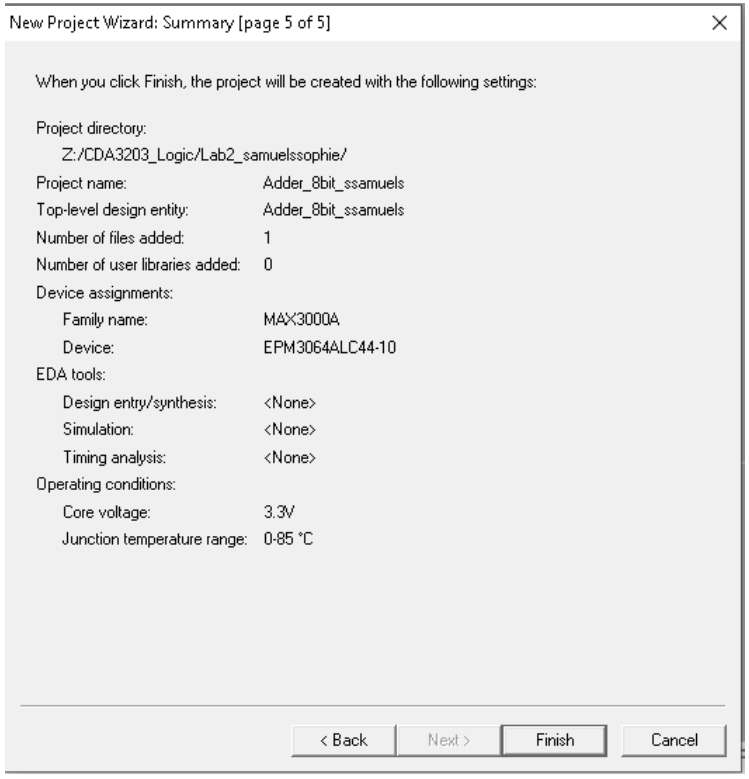


Use Cases

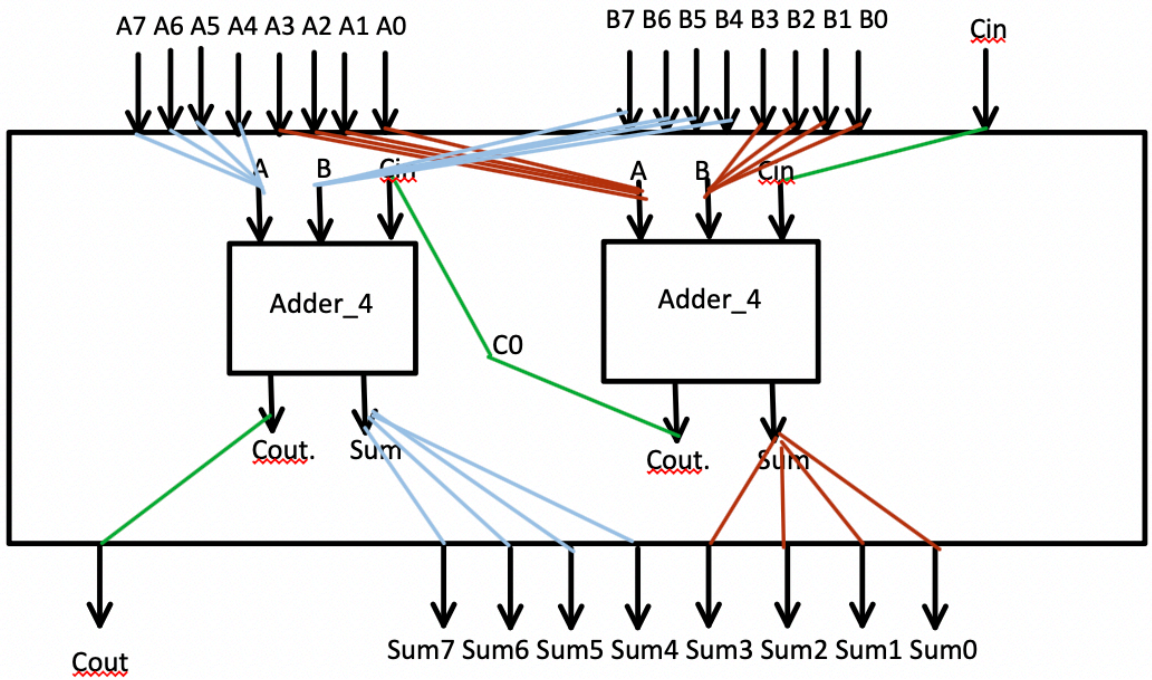


Part 4: A 8-bit Adder to add 2 8-bit binary numbers (A7,A6,A5,A4,A3,A2,A1,A0 and B7,B6,B5,B4,B3,B2,B1,B0 with A0 and B0 being least significant bits) and a 1-bit Carry-in (Cin), and results in a 4-bit Sum (S7,S6,S5,S4,S3,S2,S1,S0) and 1-bit Carry-out (Cout) with the 1-bit Full Adder component you built.

Project Settings Snip



Draw your circuit HERE



VHDL Code

Successful Compilation

Adder_8bit_ssamuels.vhd

```
1   --8-bit Adder using 4-bit Adder by Sophie Samuels
2
3   LIBRARY ieee;
4   USE ieee.std_logic_1164.all;
5
6   ENTITY Adder_8bit_ssamuels IS
7       PORT (A, B   : IN STD_LOGIC_VECTOR (7 downto 0);
8             Cin    : IN STD_LOGIC;
9
10            Cout   : OUT STD_LOGIC;
11            Sum    : OUT STD_LOGIC_VECTOR (7 downto 0));
12  END Adder_8bit_ssamuels;
13
14  ARCHITECTURE Structure OF Adder_8bit_ssamuels IS
15
16  SIGNAL C0 : STD_LOGIC;
17
18  COMPONENT Adder_4bit_ssamuels IS
19      PORT (A, B   : IN STD_LOGIC_VECTOR (3 downto 0);
20            Cin    : IN STD_LOGIC;
21
22            Cout   : OUT STD_LOGIC;
23            Sum    : OUT STD_LOGIC_VECTOR (3 downto 0));
24  END COMPONENT Adder_4bit_ssamuels;
25
26  BEGIN
27  fa0 : Adder_4bit_ssamuels PORT MAP (A(3 downto 0), B(3 downto 0), Cin, C0, Sum(3 downto 0));
28  fa1 : Adder_4bit_ssamuels PORT MAP (A(7 downto 4), B(7 downto 4), C0, Cout, Sum(7 downto 4));
29  END Structure;
```

Compiler Tool

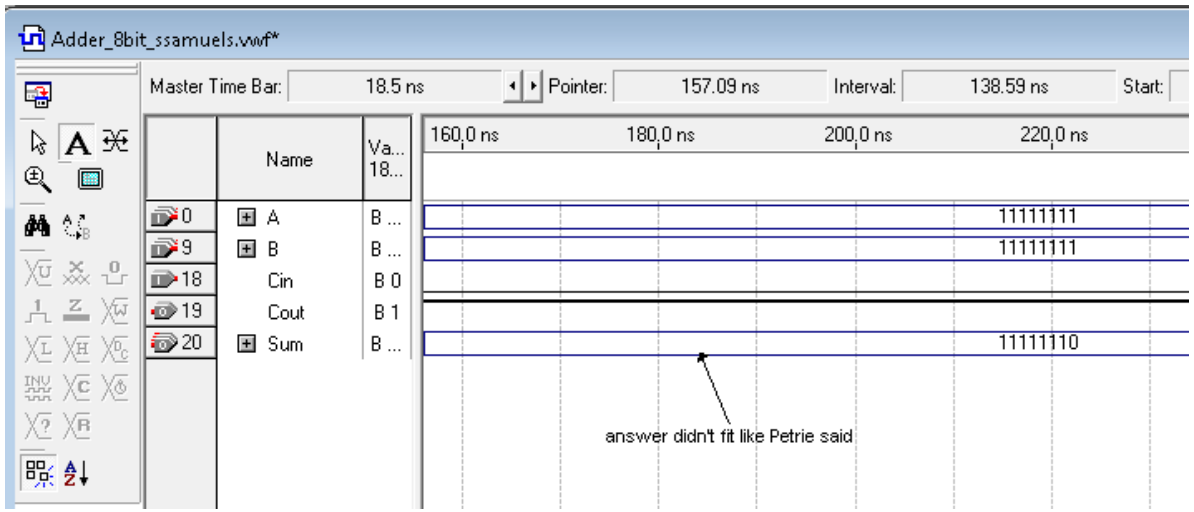
Analysis & Synthesis	Fitter	Assembler	Classic Timing Analyzer
100 %	100 %	100 %	100 %
00:00:05	00:00:02	00:00:01	00:00:02

Full Compilation: 100 %, 00:00:10

Start | Stop | Report

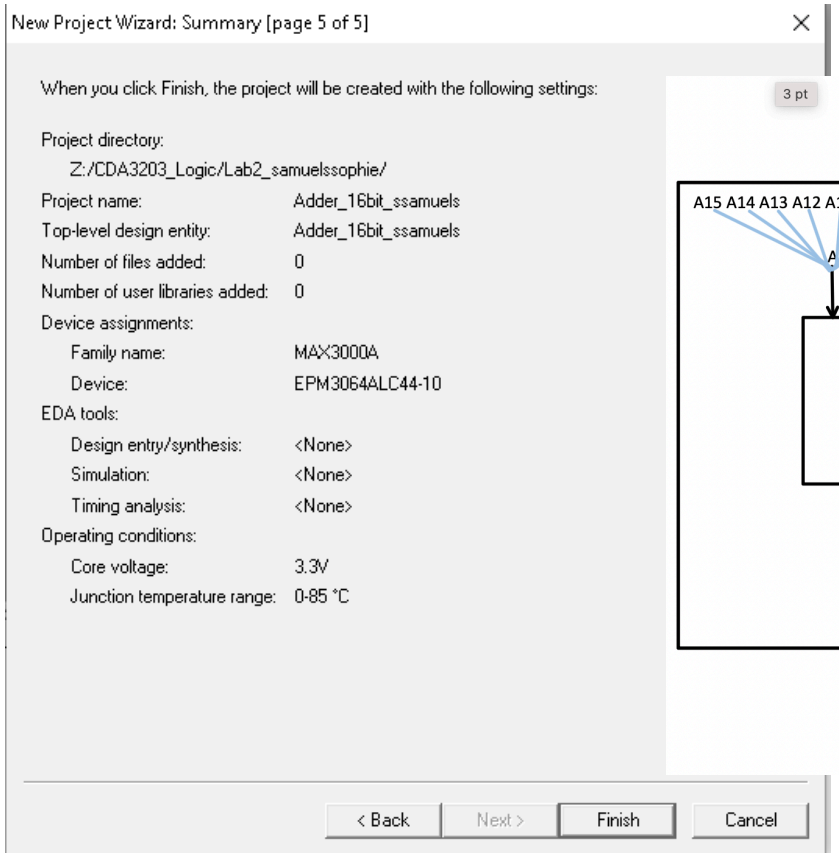
View Quartu Informatic

Timing Diagram

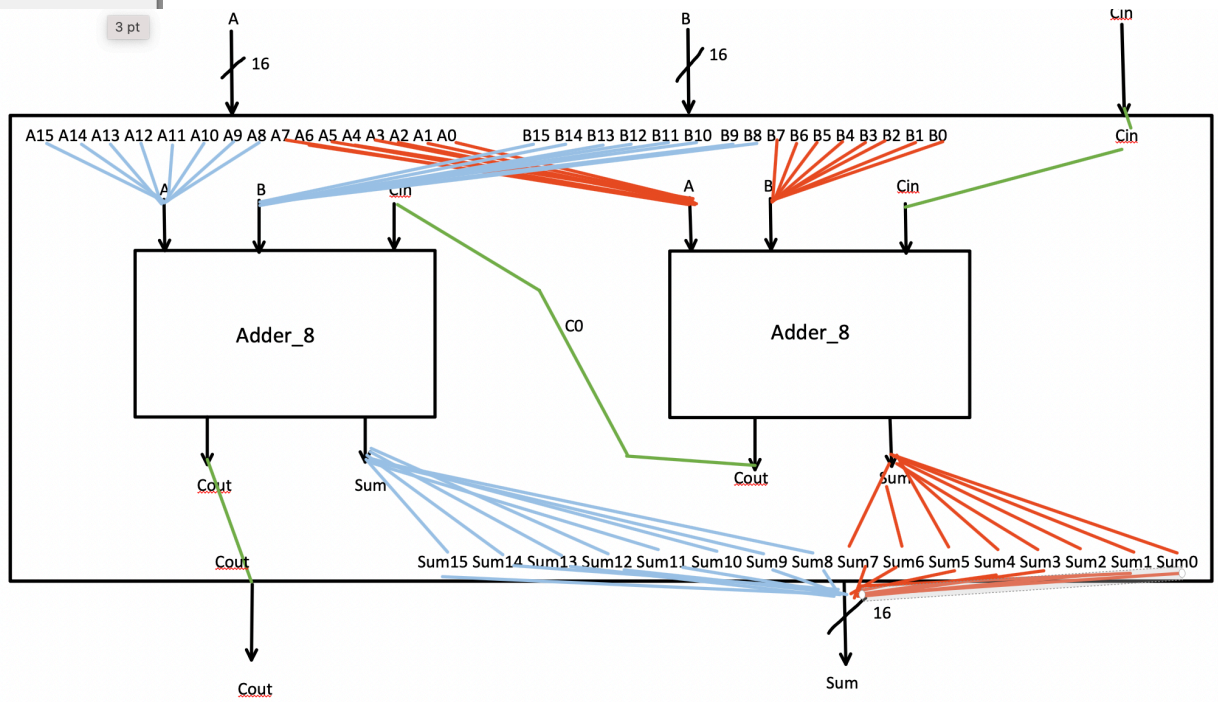


Part 5: A 16-bit Adder to add 2 16-bit binary numbers(A15,A14,A13,A12,A11,A10,A9,A8,A7,A6,A5,A4,A3,A2,A1,A0 and B15,B14,B13,B12,B11,B10,B9,B8,B7,B6, B5,B4,B3,B2,B1,B0 with A0 and B0 being least significant bits) and a 1-bit Carry-in (Cin), and results in a 16-bit Sum (S15,S14,S13,S12,S11,S10,S9,S8,S7,S6,S5,S4,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

Project Settings Snip



Draw your circuit Here



**Amended project settings after completing initial project settings to allow greater number of pins

VHDL Code

Adder_16bit_ssamuels.vhd

```
1   --16 bit Adder using 8 bit Adders by Sophie Samuels
2
3   LIBRARY ieee;
4   USE ieee.std_logic_1164.all;
5
6   ENTITY Adder_16bit_ssamuels IS
7       PORT (A, B : IN STD_LOGIC_VECTOR (15 downto 0);
8             Cin : IN STD_LOGIC;
9
10            Cout : OUT STD_LOGIC;
11            Sum : OUT STD_LOGIC_VECTOR (15 downto 0));
12  END Adder_16bit_ssamuels;
13
14  ARCHITECTURE Structure of Adder_16bit_ssamuels IS
15
16  SIGNAL CO : STD_LOGIC;
17
18  COMPONENT Adder_8bit_ssamuels IS
19      PORT (A, B  : IN STD_LOGIC_VECTOR (7 downto 0);
20            Cin   : IN STD_LOGIC;
21
22            Cout  : OUT STD_LOGIC;
23            Sum   : OUT STD_LOGIC_VECTOR (7 downto 0));
24  END COMPONENT Adder_8bit_ssamuels;
25
26  BEGIN
27  fa0  : Adder_8bit_ssamuels PORT MAP (A(7 downto 0), B(7 downto 0), Cin, CO, Sum(7 downto 0));
28  fa1 : Adder_8bit_ssamuels PORT MAP (A(15 downto 8), B(15 downto 8), CO, Cout, Sum(15 downto 8));
29  END Structure;
```

Compiler Tool

Analysis & Synthesis	Fitter	Assembler	Classic Timing Analyzer
100 %	100 %	100 %	100 %
00:00:07	00:00:02	00:00:01	00:00:01

Full Compilation: 100 % — 00:00:11

Start | Stop | Report

View Qu

Timing Diagram

Adder_16bit_ssamuels.vwf*

Master Time Bar: 18.425 ns | Pointer: 1.3 ns | Interval: -17.13 ns

	Name	Va... 18...	0 ps	10.0 ns	18.425 n
0	A	B ...	1111111111111111	0000000000000000	
17	B	B ...	1111111111111111	0000000000000000	
34	Cin	B 1			
35	Cout	B 0			
36	Sum	B ...	1111111111111110	0000000000000001	

answer does not fit like Petire said

Received Assistance from:
TA: Omair, Chelsea, Harry