CONESTOGA
Connect Life and Learning

DIGITAL DESIGN PRINCIPLES

LAB 6 – HALF ADDER & FULL ADDER

Name & Surname : Sinan KARACA **Date :** 15 / 07 / 2021

Student ID : 8743013

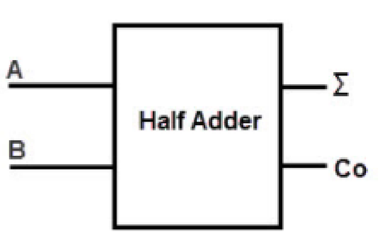
INTRODUCTION

The purpose of the experiment is to implement half adder and full adder, each design consist of VHDL and verilog files. Each verilog or vhdl file consist of 1 bit adder components. The logic of the 1 adder components has been implement from truth table by applying K-Map method.

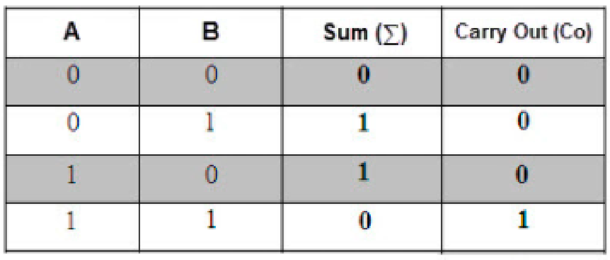
WORKING PRINCIPLE

HALF ADDER

Each bit of half adder design are independent from each other, the logic of the design is shown down below,

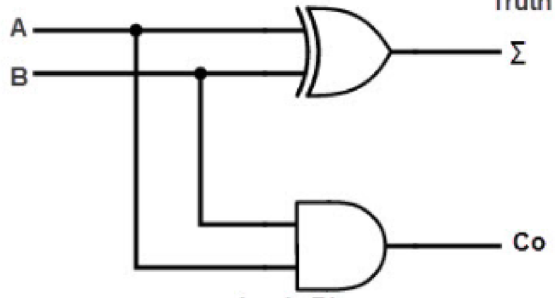


Block Diagram



A	B	Sum (∑)	Carry Out (Co)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth Table

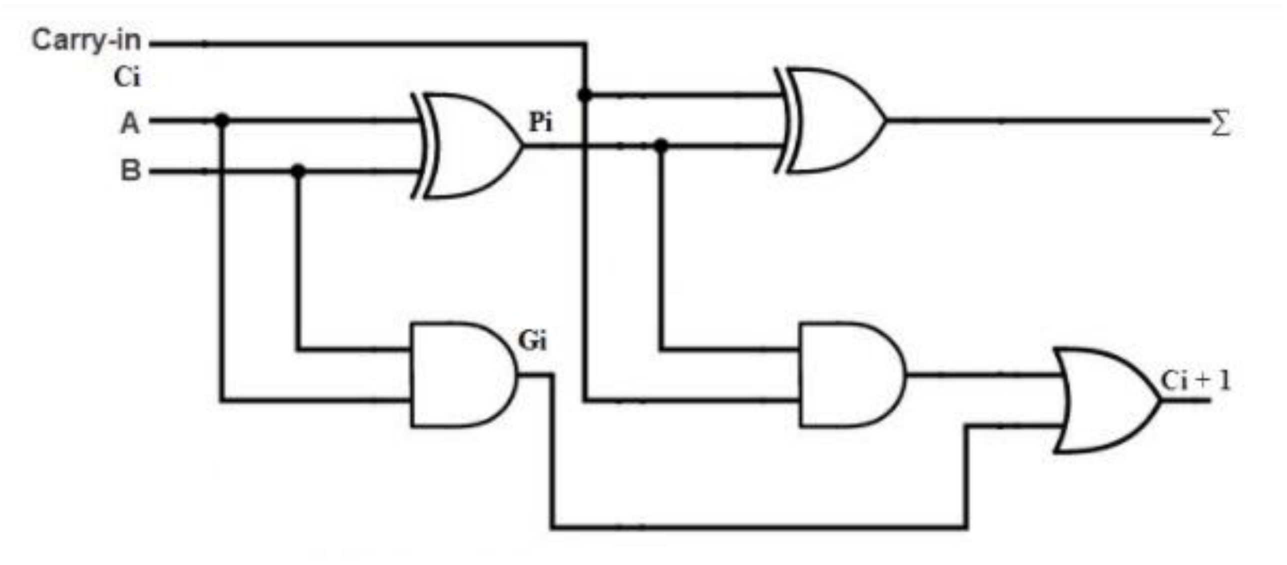


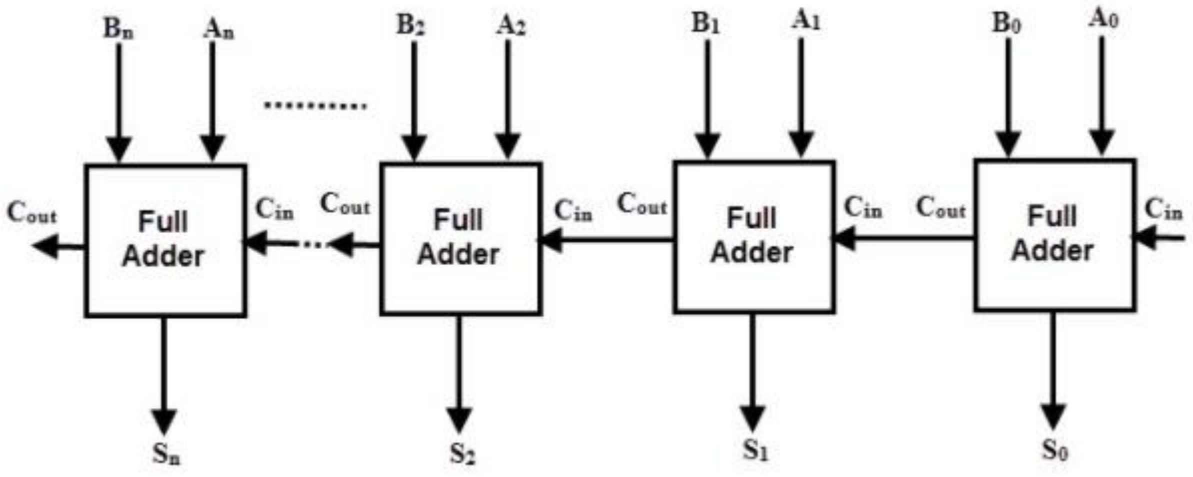
Logic Diagram

The design consist of 4 oneBitComparator components, the logic circuit above has been designed with truth table of 4 bit magnitude comparator.

FULL ADDER

Each bit of full adder logic is shown down below, and also each bit of 4 bit full adder are dependent to each other. They are connected through Cout(0) to Cin(1).





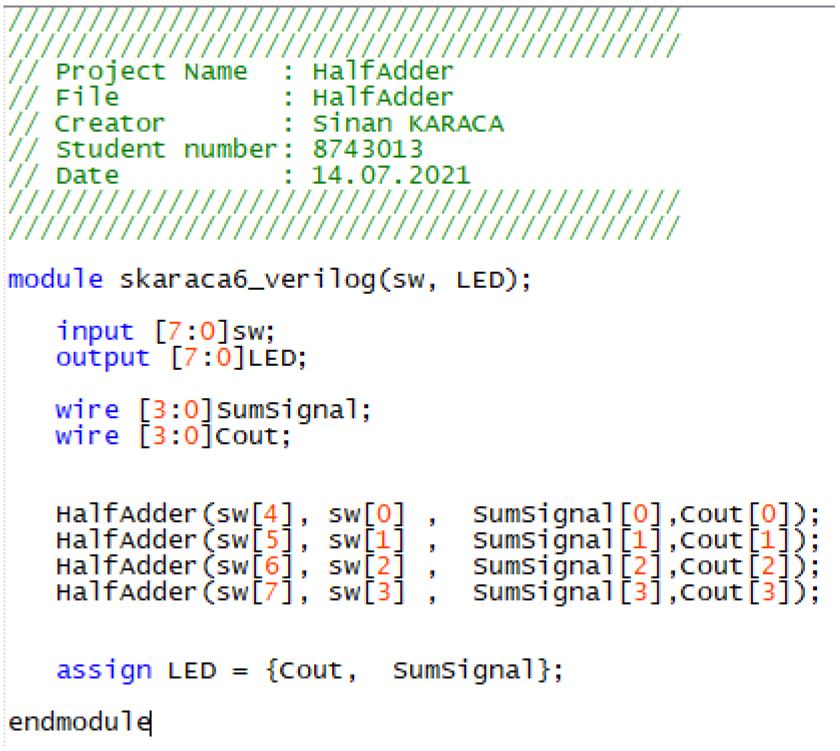
VHDL CODE OF HALF ADDER

```
----------------------------------------------------
----------------------------------------------------
-- Project Name   : Half Adder
-- File           : skaraca_lab4
-- Creator        : Sinan KARACA
-- Student number: 8743013
-- Date           : 14.07.2021
----------------------------------------------------
----------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity halfAdder is
   port(
      InputA : in STD_LOGIC;
      InputB : in STD_LOGIC;
      Sum    : out STD_LOGIC;
      Cout   : out STD_LOGIC);
   end halfAdder;


architecture Behavioral of halfAdder is

begin

   Sum  <= InputA xor InputB;

   Cout <= InputA and InputB;


end Behavioral;
```

VHDL COMPONENT CODE OF HALF ADDER

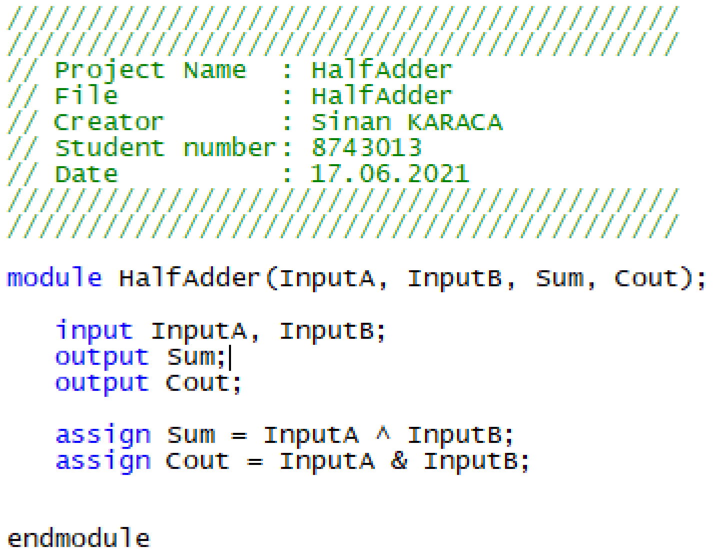
```vhdl
-----------------------------------------
-----------------------------------------
-- Project Name  : 4 Bit Half Adder
-- File          : skaraca_lab4
-- Creator       : Sinan KARACA
-- Student number: 8743013
-- Date          : 14.07.2021
-----------------------------------------
-----------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity skaraca_Lab6 is
   port(
      sw    : in STD_LOGIC_VECTOR(7 downto 0);
      LED   : out STD_LOGIC_VECTOR(7 downto 0));
   end skaraca_Lab6;


architecture Behavioral of skaraca_Lab6 is


--Component initialization
component halfAdder is
   port(
      InputA : in    STD_LOGIC;
      InputB : in    STD_LOGIC;
      Sum    : out   STD_LOGIC;
      Cout   : out   STD_LOGIC);
   end component halfAdder;


-- Signals for outputs of halfadders
signal SumSignal  : STD_LOGIC_VECTOR(3 downto 0);
signal CoutSignal : STD_LOGIC_VECTOR(3 downto 0);

begin

--Component declerations
-- LSB
   H1: halfAdder port map (sw(4), sw(0), SumSignal(0), CoutSignal(0));
   H2: halfAdder port map (sw(5), sw(1), SumSignal(1), CoutSignal(1));
   H3: halfAdder port map (sw(6), sw(2), SumSignal(2), CoutSignal(2));
---MSB
   H4: halfAdder port map (sw(7), sw(3), SumSignal(3), CoutSignal(3));


-- 4 bit LED output
   LED <=  CoutSignal & SumSignal;

end Behavioral;
```

VERILOG CODE OF HALF ADDER



```verilog
//////////////////////////////////////////
//////////////////////////////////////////
// Project Name  : HalfAdder
// File          : HalfAdder
// Creator       : Sinan KARACA
// Student number: 8743013
// Date          : 14.07.2021
//////////////////////////////////////////
//////////////////////////////////////////

module skaraca6_verilog(sw, LED);

   input  [7:0]sw;
   output  [7:0]LED;

   wire [3:0]SumSignal;
   wire [3:0]Cout;


   HalfAdder(sw[4], sw[0] ,  SumSignal[0],Cout[0]);
   HalfAdder(sw[5], sw[1] ,  SumSignal[1],Cout[1]);
   HalfAdder(sw[6], sw[2] ,  SumSignal[2],Cout[2]);
   HalfAdder(sw[7], sw[3] ,  SumSignal[3],Cout[3]);


   assign LED = {Cout,  SumSignal};

endmodule
```

VERILOG COMPONENT CODE OF HALF ADDER



```verilog
//////////////////////////////////////////
//////////////////////////////////////////
// Project Name  : HalfAdder
// File          : HalfAdder
// Creator       : Sinan KARACA
// Student number: 8743013
// Date          : 17.06.2021
//////////////////////////////////////////
//////////////////////////////////////////

module HalfAdder(InputA, InputB, Sum, Cout);

   input InputA, InputB;
   output Sum;
   output Cout;

   assign Sum = InputA ^ InputB;
   assign Cout = InputA & InputB;


endmodule
```

VHDL CODE OF FULL ADDER

```
-- File          : skaraca_lab4
-- Creator       : Sinan KARACA
-- Student number: 8743013
-- Date          : 14.07.2021
-----------------------------------------
-----------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity skaraca_Lab6 is
   port(
      sw    : in STD_LOGIC_VECTOR(7 downto 0);
      LED   : out STD_LOGIC_VECTOR(4 downto 0));
   end skaraca_Lab6;

architecture Behavioral of skaraca_Lab6 is


--Component initialization
component fullAdder is
   port(
      InputA : in STD_LOGIC;
      InputB : in STD_LOGIC;
      Cin    : in STD_LOGIC;
      Sum    : out STD_LOGIC;
      Cout   : out STD_LOGIC);
   end component fullAdder;


-- Signals for outputs of halfadders
signal SumSignal  : STD_LOGIC_VECTOR(3 downto 0);
signal CoutSignal : STD_LOGIC_VECTOR(3 downto 0);

begin

--Component declerations
   H1: fullAdder port map (sw(4), sw(0), '0'          , SumSignal(0), CoutSignal(0));
   H2: fullAdder port map (sw(5), sw(1), CoutSignal(0), SumSignal(1), CoutSignal(1));
   H3: fullAdder port map (sw(6), sw(2), CoutSignal(1), SumSignal(2), CoutSignal(2));
   H4: fullAdder port map (sw(7), sw(3), CoutSignal(2), SumSignal(3), CoutSignal(3));

-- 4 bit LED output
   LED <=  CoutSignal(3) & SumSignal ;
```

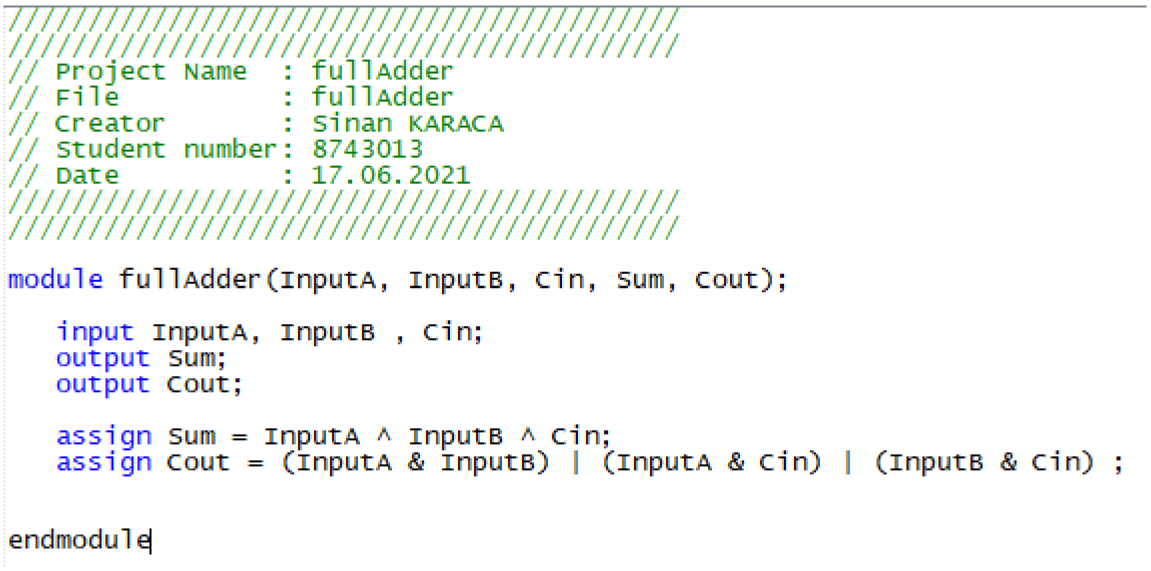
VHDL COMPONENT CODE OF FULL ADDER

```vhdl
-----------------------------------------
-----------------------------------------
-- Project Name   : Full Adder
-- File           : skaraca_lab4
-- Creator        : Sinan KARACA
-- Student number: 8743013
-- Date           : 14.07.2021
-----------------------------------------
-----------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity fullAdder is
   port(
      InputA : in STD_LOGIC;
      InputB : in STD_LOGIC;
      Cin    : in STD_LOGIC;
      Sum    : out STD_LOGIC;
      Cout   : out STD_LOGIC);
   end fullAdder;

architecture Behavioral of fullAdder is

begin

   Sum  <= InputA xor InputB xor Cin;

   Cout <= (InputA and InputB) or (InputA and Cin) or (InputB and Cin) ;


end Behavioral;
```

VERILOG CODE OF FULL ADDER

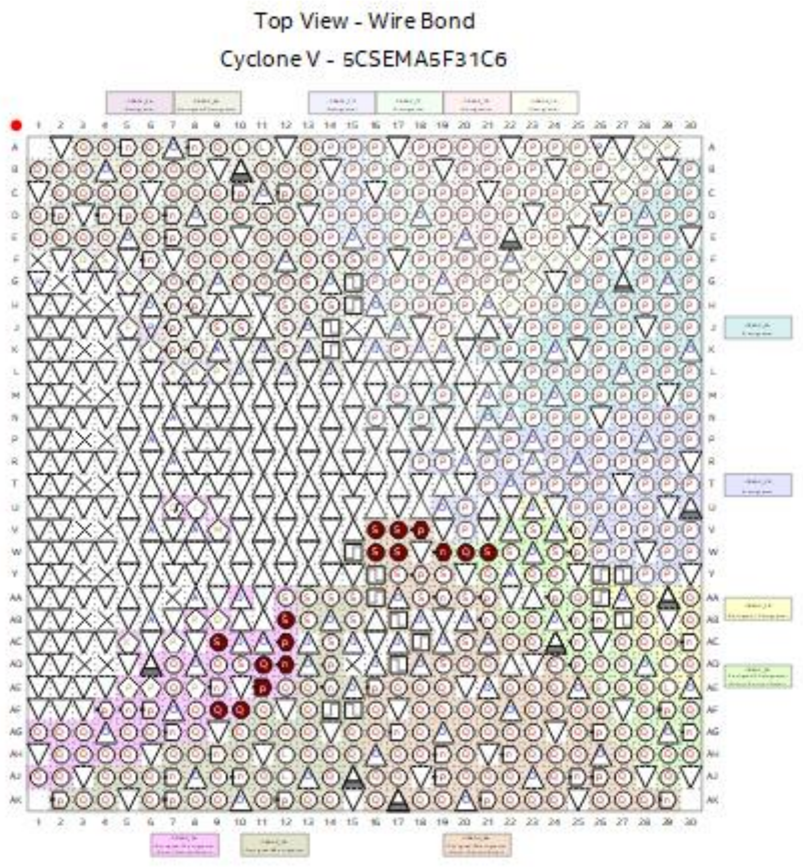
```verilog
//////////////////////////////////////////
//////////////////////////////////////////
// Project Name   : Full Adder
// File           : Full Adder
// Creator        : Sinan KARACA
// Student number: 8743013
// Date           : 14.07.2021
//////////////////////////////////////////
//////////////////////////////////////////

module skaraca6_verilog(sw, LED);

   input [7:0]sw;
   output [4:0]LED;

   wire [3:0]SumSignal;
   wire [3:0]Cout;


   fullAdder(sw[4], sw[0] , 1'b0   , SumSignal[0],Cout[0]);
   fullAdder(sw[5], sw[1] , Cout[0], SumSignal[1],Cout[1]);
   fullAdder(sw[6], sw[2] , Cout[1], SumSignal[2],Cout[2]);
   fullAdder(sw[7], sw[3] , Cout[2], SumSignal[3],Cout[3]);


   assign LED = {Cout[3],  SumSignal};

endmodule
```

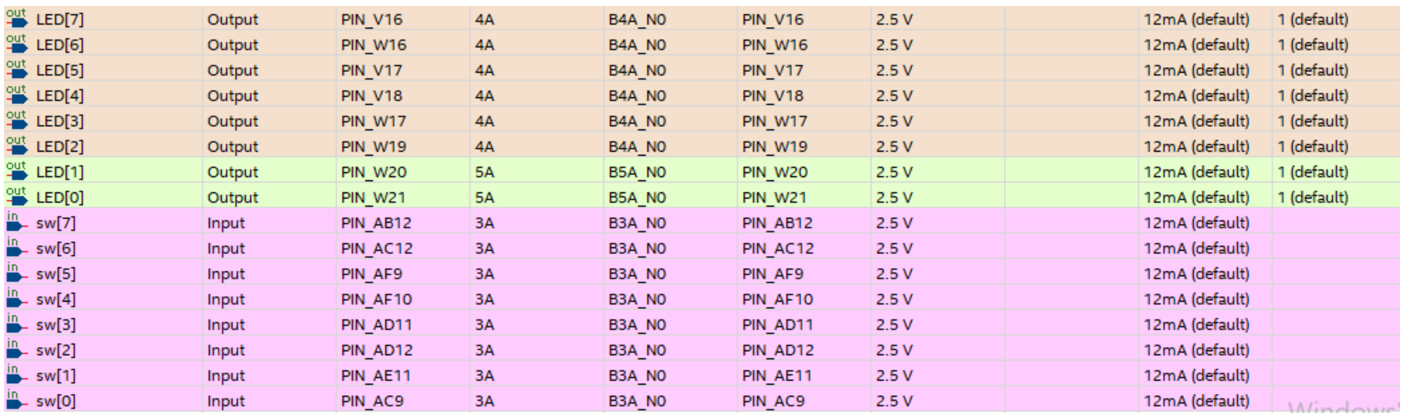
VERILOG COMPONENT CODE OF FULL ADDER



```
///////////////////////////////////////////
///////////////////////////////////////////
// Project Name   : fullAdder
// File           : fullAdder
// Creator        : Sinan KARACA
// Student number : 8743013
// Date           : 17.06.2021
///////////////////////////////////////////
///////////////////////////////////////////

module fullAdder(InputA, InputB, Cin, Sum, Cout);

   input InputA, InputB , Cin;
   output Sum;
   output Cout;

   assign Sum = InputA ^ InputB ^ Cin;
   assign Cout = (InputA & InputB) | (InputA & Cin) | (InputB & Cin) ;

endmodule
```

PIN ASSIGNMENT

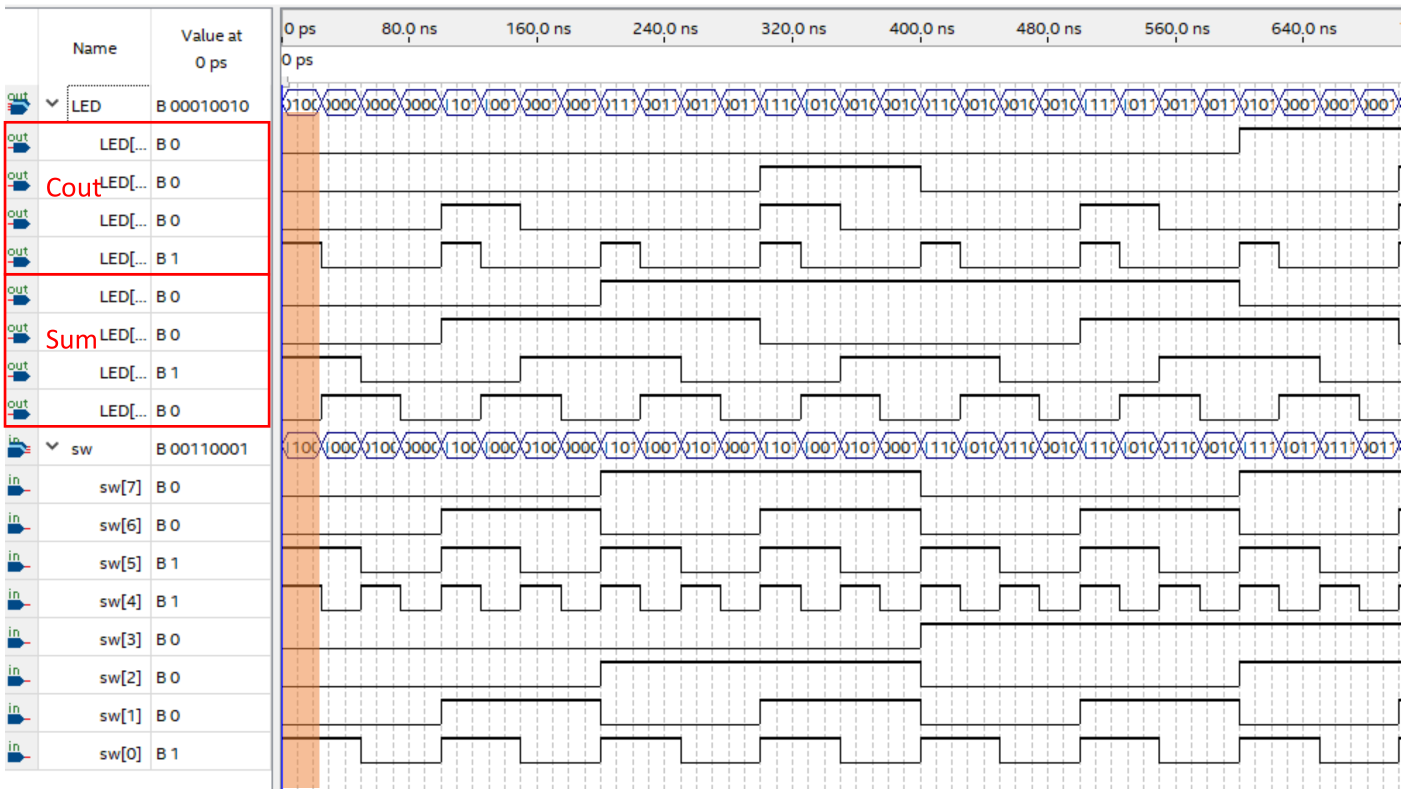




LED[7]	Output	PIN_V16	4A	B4A_N0	PIN_V16	2.5 V		12mA (default)	1 (default)
LED[6]	Output	PIN_W16	4A	B4A_N0	PIN_W16	2.5 V		12mA (default)	1 (default)
LED[5]	Output	PIN_V17	4A	B4A_N0	PIN_V17	2.5 V		12mA (default)	1 (default)
LED[4]	Output	PIN_V18	4A	B4A_N0	PIN_V18	2.5 V		12mA (default)	1 (default)
LED[3]	Output	PIN_W17	4A	B4A_N0	PIN_W17	2.5 V		12mA (default)	1 (default)
LED[2]	Output	PIN_W19	4A	B4A_N0	PIN_W19	2.5 V		12mA (default)	1 (default)
LED[1]	Output	PIN_W20	5A	B5A_N0	PIN_W20	2.5 V		12mA (default)	1 (default)
LED[0]	Output	PIN_W21	5A	B5A_N0	PIN_W21	2.5 V		12mA (default)	1 (default)
sw[7]	Input	PIN_AB12	3A	B3A_N0	PIN_AB12	2.5 V		12mA (default)	
sw[6]	Input	PIN_AC12	3A	B3A_N0	PIN_AC12	2.5 V		12mA (default)	
sw[5]	Input	PIN_AF9	3A	B3A_N0	PIN_AF9	2.5 V		12mA (default)	
sw[4]	Input	PIN_AF10	3A	B3A_N0	PIN_AF10	2.5 V		12mA (default)	
sw[3]	Input	PIN_AD11	3A	B3A_N0	PIN_AD11	2.5 V		12mA (default)	
sw[2]	Input	PIN_AD12	3A	B3A_N0	PIN_AD12	2.5 V		12mA (default)	
sw[1]	Input	PIN_AE11	3A	B3A_N0	PIN_AE11	2.5 V		12mA (default)	
sw[0]	Input	PIN_AC9	3A	B3A_N0	PIN_AC9	2.5 V		12mA (default)	

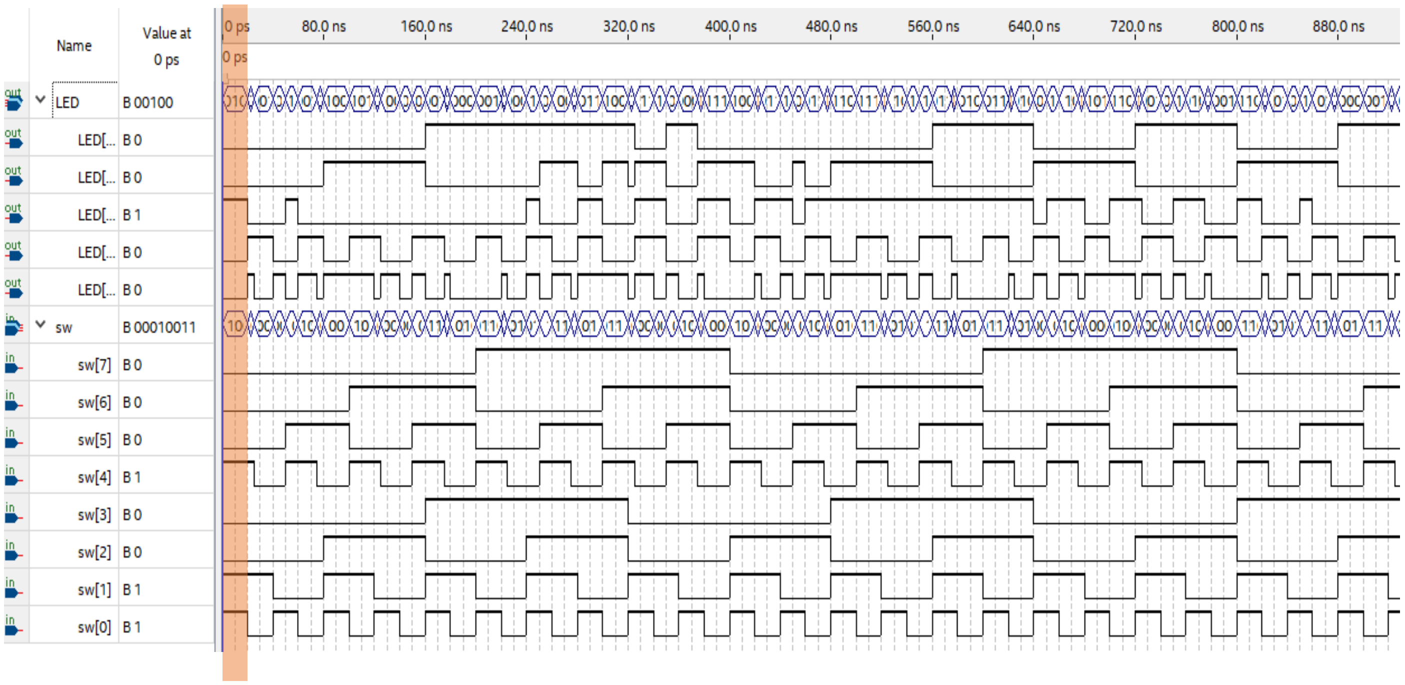
SIMALATION WAVEFORM

Half Adder



Last digits of my student id is “1” , “3” , it is shown above. The result is Sum = 0010 and Cout = 0001.

Full Adder



Last digits of my student id is “1” , “3” , it is shown above. The result of full adder is Sum = 0100 and Cout = 0.

CONCLUSION

Design of full adder and half adder has been experienced in the this experiment, 4 different codes has been written and run, each has its own components. Components consist of logical operations, as they are implemented from K-Map method.