**LAB NOTE**

**Subject: Digital Design Principles**

**Topic: Half Adder**

**Student: Minh Quan Tran**

**July 5th, 2024**

Table of Contents

[1. Objectives 4](#_Toc171084816)

[2. Theory and Design 6](#_Toc171084817)

[2.1 Theory 6](#_Toc171084818)

[2.2 Requirement 6](#_Toc171084819)

[2.3 Solution 6](#_Toc171084820)

[3. VHDL and Verilog 6](#_Toc171084821)

[3.1 VHDL code 6](#_Toc171084822)

[3.2 Verilog code 8](#_Toc171084823)

[4. Pin Planner 11](#_Toc171084824)

[4.1 Input and Output 11](#_Toc171084825)

[5. Result 13](#_Toc171084826)

[5.1 Half Adder 13](#_Toc171084827)

[5.2 Half Subtractor 13](#_Toc171084828)

**TABLE OF FIGURES**

[Figure 2‑1: Half Adder Truth Table 6](#_Toc171084858)

[Figure 2‑2: Half Subtractor's truth table 6](#_Toc171084859)

[Figure 4‑1: SCEMA5F31C6N board 11](#_Toc171084860)

[Figure 4‑2: SW0 and SW1 Pin No 12](#_Toc171084861)

[Figure 4‑3: LEDR0's Pin No 12](#_Toc171084862)

[Figure 4‑4: HEX's Pin No 12](#_Toc171084863)

[Figure 5‑1: Output for Half Adder. 13](#_Toc171084864)

[Figure 5‑2: Output for Half Subtractor 13](#_Toc171084865)

# Objectives

* Design an Half Adder using Quartus software.
* Write a VHDL and Verilog code from the design.
* Push the code and run to the SCEMA5F31C6N board.

# Theory and Design

## Theory

Half adder is used for adding a basic function of digital circuits.

* It is useful in many data processing applications.
* This circuit implements the following:
* 0 + 0 = 00
* 0 + 1 = 01
* 1 + 0 = 01
* 1 + 1 = 10

## Requirement

* Use VHDL and Verilog to implement half adder and display to LED [0:7].
* Introduction to the VHDL Case statement.
* **Above and beyond:** Adding 1 more modes: Half Subtractor

## Solution

To design this system, first need to identify input and output:

* Input A will be from SW0 to SW3.
* Input B will be from SW4 to SW7.
* Output will be the 7 LED.

Then identify what the output will be using the below truth table,

A table with numbers and symbols

Description automatically generated

Figure 2‑1: Half Adder Truth Table

A table with numbers and words

Description automatically generated

Figure 2‑2: Half Subtractor's truth table

# VHDL and Verilog

## VHDL code

**library** ieee;

**use** ieee.std\_logic\_1164.**ALL**;

**use** IEEE.NUMERIC\_STD.**ALL**;

**entity** MTran\_Lab6\_VHDL\_Adder **is**

**port**

(

mode : **in** std\_logic\_vector (9 **downto** 8);

A : **in** std\_logic\_vector (3 **downto** 0);

B : **in** std\_logic\_vector (7 **downto** 4);

LED : **out** std\_logic\_vector (7 **downto** 0);

HEX1 : **out** std\_logic\_vector (6 **downto** 0);

HEX0 : **out** std\_logic\_vector (6 **downto** 0)

);

**end** MTran\_Lab6\_VHDL\_Adder;

**architecture** Behavioral **of** MTran\_Lab6\_VHDL\_Adder **is**

-------------------------- Component Declaration---------------------------------

--Half Adder

**component** MTran\_Lab6\_VHDL\_HalfAdder **is**

**port**

(

A : **in** std\_logic\_vector (3 **downto** 0);

B : **in** std\_logic\_vector (7 **downto** 4);

Sum : **out** std\_logic\_vector (3 **downto** 0);

Carry : **out** std\_logic\_vector (3 **downto** 0)

);

**end** **component**;

--Half Subtractor

**component** MTran\_Lab6\_VHDL\_HalfSubtractor **is**

**port**

(

A : **in** std\_logic\_vector (3 **downto** 0);

B : **in** std\_logic\_vector (7 **downto** 4);

Difference : **out** std\_logic\_vector (3 **downto** 0);

Borrow : **out** std\_logic\_vector (3 **downto** 0)

);

**end** **component**;

--Full Adder

**component** MTran\_Lab6\_VHDL\_FullAdder **is**

**port**

(

A : **in** std\_logic\_vector (3 **downto** 0);

B : **in** std\_logic\_vector (3 **downto** 0);

Cin : **in** std\_logic;

Sum : **out** std\_logic\_vector (3 **downto** 0);

CarryOut : **out** std\_logic

);

**end** **component**;

-------------------------- End of Component Declaration--------------------------

-------------------------- Function Declaration----------------------------------

-- Function to calculate Sum of Half Adder

-- Seven Segment Display

**function** SevenSegmentDisplay(Number : integer) **return** std\_logic\_vector **is**

**begin**

**case** (Number) **is**

**when** 0 => **return** "1000000";

**when** 1 => **return** "1111001";

**when** 2 => **return** "0100100";

**when** 3 => **return** "0110000";

**when** 4 => **return** "0011001";

**when** 5 => **return** "0010010";

**when** 6 => **return** "0000010";

**when** 7 => **return** "1111000";

**when** 8 => **return** "0000000";

**when** 9 => **return** "0010000";

**when** **others** => **return** "1111111";

**end** **case**;

**end** SevenSegmentDisplay;

--------------------------End of Function Declaration--------------------------

--------------------------Variable and Signal Declaration----------------------

**signal** Sum\_HA : std\_logic\_vector(3 **downto** 0);

**signal** Carry\_HA : std\_logic\_vector(3 **downto** 0);

**signal** Difference\_HS: std\_logic\_vector(3 **downto** 0);

**signal** Borrow\_HS: std\_logic\_vector(3 **downto** 0);

**signal** Sum\_FA : std\_logic\_vector(3 **downto** 0);

**signal** Carry\_FA : std\_logic;

**signal** Cin\_FA: std\_logic;

**signal** TotalSum : integer;

**signal** tenth : integer;

**signal** unit : integer;

--------------------------End of Variable and Signal Declaration---------------

**begin**

-- Main code

-- Component

HalfAdder: MTran\_Lab6\_VHDL\_HalfAdder

**port** **map**(

A => A,

B => B,

Sum => Sum\_HA,

Carry => Carry\_HA

);

HalfSubtractor: MTran\_Lab6\_VHDL\_HalfSubtractor

**port** **map**(

A => A,

B => B,

Difference => Difference\_HS,

Borrow => Borrow\_HS

);

FullAdder: MTran\_Lab6\_VHDL\_FullAdder

**port** **map**(

A => A,

B => B,

Cin => Cin\_FA,

Sum => Sum\_FA,

CarryOut => Carry\_FA

);

**process**(mode, A, B)

**begin**

-- Reset Seven Segment Display and LED

HEX1 <= "1111111";

HEX0 <= "1111111";

LED <= "00000000";

-- Change mode

**if** mode = "00" **then** -- Half Adder mode

LED(3 **downto** 0) <= Sum\_HA;

LED(7 **downto** 4) <= Carry\_HA;

**elsif** mode = "01" **then** -- Half Subtractor mode

LED(3 **downto** 0) <= Difference\_HS;

LED(7 **downto** 4) <= Borrow\_HS;

**elsif** mode = "10" **then** -- Full Adder with Cin = 0 mode

Cin\_FA <= '0';

LED(3 **downto** 0) <= Sum\_FA;

LED(4) <= Carry\_FA;

TotalSum <= to\_integer(unsigned(Carry\_FA & Sum\_FA));

tenth <= TotalSum / 10;

unit <= TotalSum **mod** 10;

HEX1 <= SevenSegmentDisplay(tenth);

HEX0 <= SevenSegmentDisplay(unit);

**elsif** mode = "11" **then** -- Full Adder with Cin = 1 mode

Cin\_FA <= '1';

LED(3 **downto** 0) <= Sum\_FA;

LED(4) <= Carry\_FA;

TotalSum <= to\_integer(unsigned(Carry\_FA & Sum\_FA));

tenth <= TotalSum / 10;

unit <= TotalSum **mod** 10;

HEX1 <= SevenSegmentDisplay(tenth);

HEX0 <= SevenSegmentDisplay(unit);

**end** **if**;

**end** **process**;

**end** Behavioral;

## Verilog code

**module MTran\_Lab6\_Verilog\_Adder**(

**input** wire [9:8] mode,

input wire [3:0] A,

input wire [3:0] B,

output reg [7:0] LED,

output reg [6:0] HEX1,

output reg [6:0] HEX0

)**;**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Variable and Signal Declaration\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**wire** [3:0] Sum\_HA**;**

**wire** [3:0] Carry\_HA**;**

**wire** [3:0] Difference\_HS**;**

**wire** [3:0] Borrow\_HS**;**

**wire** [3:0] Sum\_FA**;**

**reg** Cin**;**

**wire** Carry\_FA**;**

**reg** [4:0] TotalSum**;**

**reg** [4:0] tenth**;**

**reg** [4:0] units**;**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End of Variable and Signal Declaration\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Module Instantiation\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

MTran\_Lab6\_Verilog\_HalfAdder HA (

.A(A),

.B(B),

.Sum(Sum\_HA),

.Carry(Carry\_HA)

);

MTran\_Lab6\_Verilog\_HalfSubtractor HS (

.A(A),

.B(B),

.Difference(Difference\_HS),

.Borrow(Borrow\_HS)

);

MTran\_Lab6\_Verilog\_FullAdder FA (

.A(A),

.B(B),

.Cin(Cin),

.Sum(Sum\_FA),

.CarryOut(Carry\_FA)

);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End of Module Instantiation\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Function Declaration\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

function [6:0] SevenSegmentDisplay;

**input** reg [4:0] Num**;**

**begin**

**case(**Num**)**

5'd0**:** SevenSegmentDisplay = 7'b1000000**;**

5'd1**:** SevenSegmentDisplay = 7'b1111001**;**

5'd2**:** SevenSegmentDisplay = 7'b0100100**;**

5'd3**:** SevenSegmentDisplay = 7'b0110000**;**

5'd4**:** SevenSegmentDisplay = 7'b0011001**;**

5'd5**:** SevenSegmentDisplay = 7'b0010010**;**

5'd6**:** SevenSegmentDisplay = 7'b0000010**;**

5'd7**:** SevenSegmentDisplay = 7'b1111000**;**

5'd8**:** SevenSegmentDisplay = 7'b0000000**;**

5'd9**:** SevenSegmentDisplay = 7'b0010000**;**

**default:** SevenSegmentDisplay = 7'b1111111**;**

**endcase**;

**end**

endfunction

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End of Function Declaration\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Main code

**always**@(mode,A,B)

**begin**

// Reset LED and 7 Segment

HEX1 = 7'b1111111;

HEX0 = 7'b1111111;

LED = 8'b000000000;

// Change mode

**if** (mode == 2'b00)

**begin**

LED[3:0] = Sum\_HA;

LED[7:4] = Carry\_HA;

**end**

**else** **if** (mode == 2'b01)

**begin**

LED[3:0] = Difference\_HS;

LED[7:4] = Borrow\_HS;

**end**

**else** **if** (mode == 2'b10)

**begin**

Cin = 1'b0;

LED[3:0] = Sum\_FA;

LED[7:4] = Carry\_FA;

TotalSum = {Carry\_FA, Sum\_FA};

tenth = TotalSum / 10;

units = TotalSum % 10;

HEX1 = SevenSegmentDisplay(tenth);

HEX0 = SevenSegmentDisplay(units);

**end**

**else** **if** (mode == 2'b11)

**begin**

Cin = 1'b1;

LED[3:0] = Sum\_FA;

LED[7:4] = Carry\_FA;

TotalSum = {Carry\_FA, Sum\_FA};

tenth = TotalSum / 10;

units = TotalSum % 10;

HEX1 = SevenSegmentDisplay(tenth);

HEX0 = SevenSegmentDisplay(units);

**end**

**end**

endmodule

# Pin Planner

## Input and Output

A close-up of a circuit board

Description automatically generated

Figure 4‑1: SCEMA5F31C6N board

Assigning:

SW[0]: SW0

SW[1]: SW1

SW[2]: SW2

Mode[0]: SW7

Mode[1]: SW8

EN: SW9

LED[0]: LEDR0

LED[1]: LEDR1

LED[2]: LEDR2

LED[3]: LEDR3

LED[4]: LEDR4

LED[5]: LEDR5

LED[6]: LEDR6

LED[7]: LEDR7

HEX0[0]: HEX0[0]

HEX0[1]: HEX0[1]

HEX0[2]: HEX0[2]

HEX0[3]: HEX0[3]

HEX0[4]: HEX0[4]

HEX0[5]: HEX0[5]

HEX0[6]: HEX0[6]

From the DE1\_SoC\_User\_Manual,

A screenshot of a slide switch

Description automatically generated

Figure 4‑2: SW0 and SW1 Pin No

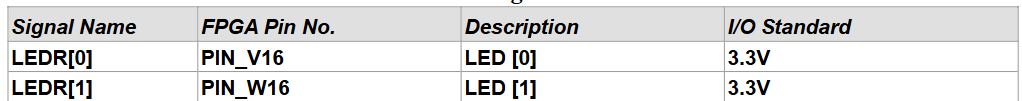


Figure 4‑3: LEDR0's Pin No

 A group of black text

Description automatically generated

Figure 4‑4: HEX's Pin No

# Result

## Half Adder

With inputA = “0001” and inputB = “1001”, Sum should be “1000” and Carry is “0001”

A close up of a circuit board

Description automatically generated

Figure 5‑1: Output for Half Adder.

## Half Subtractor

With inputA = “1000” and inputB = “1011”, Difference should be “0011” and Carry is “0011”

A close up of a circuit board

Description automatically generated

Figure 5‑2: Output for Half Subtractor

**REFERENCES**