LAB 04

FULL ADDER

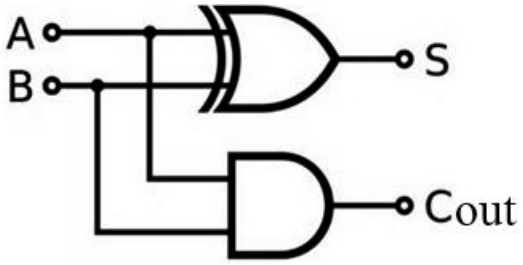
CECS 225 – DIGITAL LOGIC AND ASSEMBLY PROGRAMMING

Professor: Xiaolong Wu
Student: Thanh Nguyen _ ID: 026843815

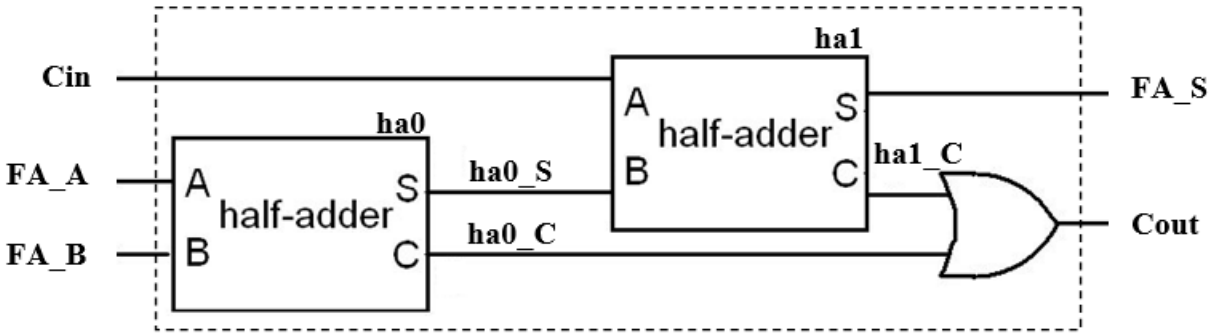
1. Introduction:

- *Definition:* Full Adder is the center of most digital circuits that perform addition or subtraction. It adds together two binary digits, plus a carry-in digit to produce a sum and a carry-out digit.
- *Structure:*
    - It requires 3 inputs and produces 2 outputs. Three outputs consist 2 binary digits A and B, and a carry-in digit called Cin. The outputs are the sum of 2 binary digits and carry-in known as Cout, and a carry-out digit output.

Half Adder



FullAdder



    - It is implemented from 2 Half Adders and one OR logic gate. With this logic circuit, 2 bits can be added together, taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude. As we see in the circuit above, the inputs of Half Adder 1 are a carry-in **Cin** is input **A** and a carry-out from Half Adder 0 **ha0_S** is input **B**. For the outputs, **FA_S** is the final carry-out digit which is **S** and **ha1_C** is the final sum **C**. Then we got:

HalfAdder ha1(.A(Cin), .B(ha0_S), .Cout(ha1_C), .S(FA_S))

- *Truth table:*

C_{in}	A	B	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

2. Verilog Codes

- ***Design Code***

```
// Project 04: Full Adder
// Name      : Thanh Nguyen - ID: 026843815

module HalfAdder(A, B, Cout, S);
  input A, B;
  output Cout, S;

  assign Cout = A & B;
  assign S = A ^ B;

endmodule

module FullAdder(Cin, FA_A, FA_B, FA_S, FA_Cout);
  input FA_A, FA_B, Cin;
  output FA_S, FA_Cout;

  wire ha0_S, ha0_C, ha1_C;

  HalfAdder ha0 (   .A   (FA_A) ,
                    .B   (FA_B) ,
                    .Cout(ha0_C),
                    .S   (ha0_S)
                );

  HalfAdder ha1 (   .A   (Cin)  ,
                    .B   (ha0_S),
                    .Cout(ha1_C),
                    .S   (FA_S)
                );

  // This is the carry out for the Full Adder
  assign FA_Cout = ha0_C | ha1_C;

endmodule
```

- ***Testbench***

SV/Verilog Testbench

```
// Project 04: Full Adder
// Name      : Thanh Nguyen - ID: 026843815

`timescale 1ns/1ps

module testbench();
  reg FA_A1, FA_B1, Cin1;
  wire FA_S1, FA_Cout1;

  FullAdder FA1(Cin1, FA_A1, FA_B1, FA_S1, FA_Cout1);

  initial
    begin
      //Dump waves
      $dumpfile("dump.vcd");
      $dumpvars(1, testbench);

      // Cin = 0, FA_A = 0, FA_B = 0
      $display("Test Case 0");
      Cin1 = 1'b0; FA_A1 = 1'b0; FA_B1 = 1'b0;
      $display("Cin     = %b", Cin1, "     FA_A = %b", FA_A1, "
     FA_B = %b", FA_B1);
      #1
      $display("FA_Cout = %b", FA_Cout1, "     FA_S = %b",
FA_S1);

       // Cin = 0, FA_A = 0, FA_B = 1
      $display("Test Case 1");
      Cin1 = 1'b0; FA_A1 = 1'b0; FA_B1 = 1'b1;
      $display("Cin     = %b", Cin1, "     FA_A = %b", FA_A1, "
     FA_B = %b", FA_B1);
      #1
      $display("FA_Cout = %b", FA_Cout1, "     FA_S = %b",
FA_S1);

      // Cin = 0, FA_A = 1, FA_B = 0
      $display("Test Case 2");
      Cin1 = 1'b0; FA_A1 = 1'b1; FA_B1 = 1'b0;
      $display("Cin     = %b", Cin1, "     FA_A = %b", FA_A1, "
     FA_B = %b", FA_B1);
      #1
      $display("FA_Cout = %b", FA_Cout1, "     FA_S = %b",
FA_S1);
```

```
38
39        // Cin = 0, FA_A = 1, FA_B = 1
40       $display("Test Case 3");
41       Cin1 = 1'b0; FA_A1 = 1'b1; FA_B1 = 1'b1;
42       $display("Cin     = %b", Cin1, "     FA_A = %b", FA_A1, "
      FA_B = %b", FA_B1);
43        #1
44       $display("FA_Cout = %b", FA_Cout1, "     FA_S = %b",
FA_S1);
45
46       // Cin = 1, FA_A = 0, FA_B = 0
47       $display("Test Case 4");
48       Cin1 = 1'b1; FA_A1 = 1'b0; FA_B1 = 1'b0;
49       $display("Cin     = %b", Cin1, "     FA_A = %b", FA_A1, "
      FA_B = %b", FA_B1);
50        #1
51       $display("FA_Cout = %b", FA_Cout1, "     FA_S = %b",
FA_S1);
52
53        // Cin = 1, FA_A = 0, FA_B = 1
54       $display("Test Case 5");
55       Cin1 = 1'b1; FA_A1 = 1'b0; FA_B1 = 1'b1;
56       $display("Cin     = %b", Cin1, "     FA_A = %b", FA_A1, "
      FA_B = %b", FA_B1);
57        #1
58       $display("FA_Cout = %b", FA_Cout1, "     FA_S = %b",
FA_S1);
59
60       // Cin = 1, FA_A = 1, FA_B = 0
61       $display("Test Case 6");
62       Cin1 = 1'b1; FA_A1 = 1'b1; FA_B1 = 1'b0;
63       $display("Cin     = %b", Cin1, "     FA_A = %b", FA_A1, "
      FA_B = %b", FA_B1);
64        #1
65       $display("FA_Cout = %b", FA_Cout1, "     FA_S = %b",
FA_S1);
```

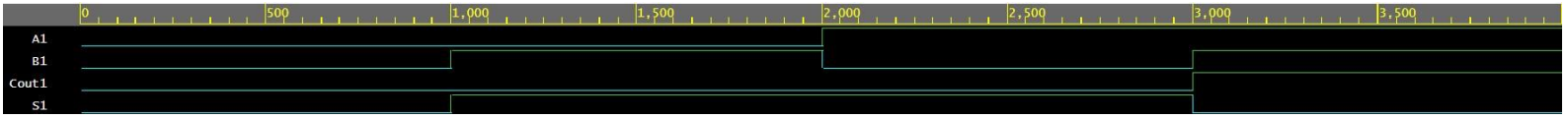
```
66
67        // Cin = 1, FA_A = 1, FA_B = 1
68       $display("Test Case 7");
69       Cin1 = 1'b1; FA_A1 = 1'b1; FA_B1 = 1'b1;
70       $display("Cin     = %b", Cin1, "     FA_A = %b", FA_A1, "
      FA_B = %b", FA_B1);
71        #1
72       $display("FA_Cout = %b", FA_Cout1, "     FA_S = %b",
FA_S1);
73     end
74
75 endmodule
```

3. Simulator Waveform

- ***Half Adder:***
  - Inputs and Outputs:

```
Test Case 0
A    = 0     B = 0
Cout = 0     F = 0
Test Case 1
A    = 0     B = 1
Cout = 0     F = 1
Test Case 2
A    = 1     B = 0
Cout = 0     F = 1
Test Case 3
A    = 1     B = 1
Cout = 1     F = 0
```

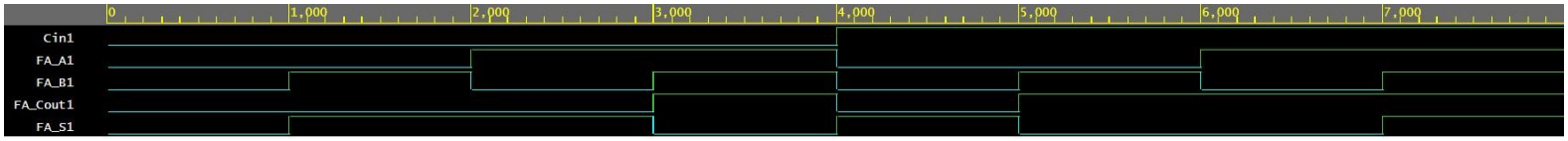
- Waveform:



- ***Full Adder:***
  - Inputs and Outputs:

```
Test Case 0
Cin     = 0     FA_A = 0     FA_B = 0
FA_Cout = 0     FA_S = 0
Test Case 1
Cin     = 0     FA_A = 0     FA_B = 1
FA_Cout = 0     FA_S = 1
Test Case 2
Cin     = 0     FA_A = 1     FA_B = 0
FA_Cout = 0     FA_S = 1
Test Case 3
Cin     = 0     FA_A = 1     FA_B = 1
FA_Cout = 1     FA_S = 0
Test Case 4
Cin     = 1     FA_A = 0     FA_B = 0
FA_Cout = 0     FA_S = 1
Test Case 5
Cin     = 1     FA_A = 0     FA_B = 1
FA_Cout = 1     FA_S = 0
Test Case 6
Cin     = 1     FA_A = 1     FA_B = 0
FA_Cout = 1     FA_S = 0
Test Case 7
Cin     = 1     FA_A = 1     FA_B = 1
FA_Cout = 1     FA_S = 1
```

  - Waveform:


0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
Cin1
FA_A1
FA_B1
FA_Cout1
FA_S1