CHITTAGONG UNIVERSITY OF ENGINEERING & TECHNOLOGY

Department of Electronics And Telecommunication Engineering

Lab Report

Experiment Name: Introduction to Verilog HDL and Quartus II

Experiment No.: 09

Course Title: VLSI technology Sessional

Course No.: ETE 404

Date of Experiment: 06-10-2024

Date of Submission: 20-10-2024

Submitted By	Submitted To
Name: M.I. Yasir Arafat **ID:** 1908031 **Level:** 4 **Term:** I	Arif Istiaque Lecturer Dept. of ETE,CUET

Objectives

- To familiarize with Quartus II software environment
- To understand and implement Hardware Description Language (HDL)
- To learn behavioral and structural Verilog descriptions through practical implementation

Equipments

- Intel Quartus II software

Theory

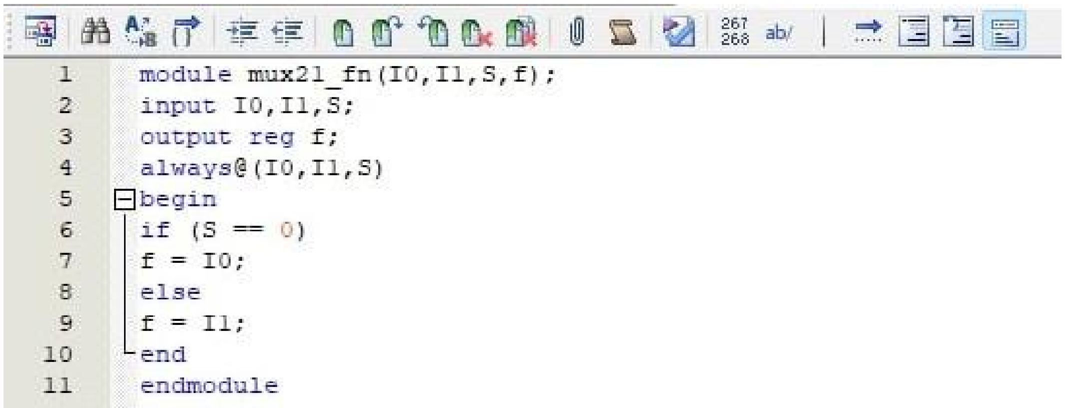
Hardware Description Language (HDL) is a specialized computer language used for describing electronic circuits' structure and behavior. Unlike traditional programming languages, HDL describes hard- ware implementations rather than software algorithms. Two IEEE standard HDLs were discussed:

- Verilog HDL
- VHDL (Very High Speed Integrated Circuit Hardware Description Language)

Experimental Procedure and Results

2x1 Multiplexer

The 2x1 multiplexer was implemented using the following Verilog code:



```
module mux21_fn(I0,I1,S,f);
input I0,I1,S;
output reg f;
always@(I0,I1,S)
begin
if (S == 0)
f = I0;
else
f = I1;
end
endmodule
```

Figure 1: Code of 2x1 Multiplexer

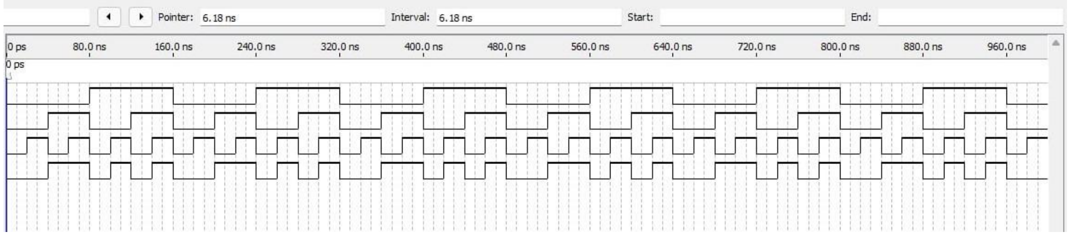


Figure 2: MUX simulation waveform

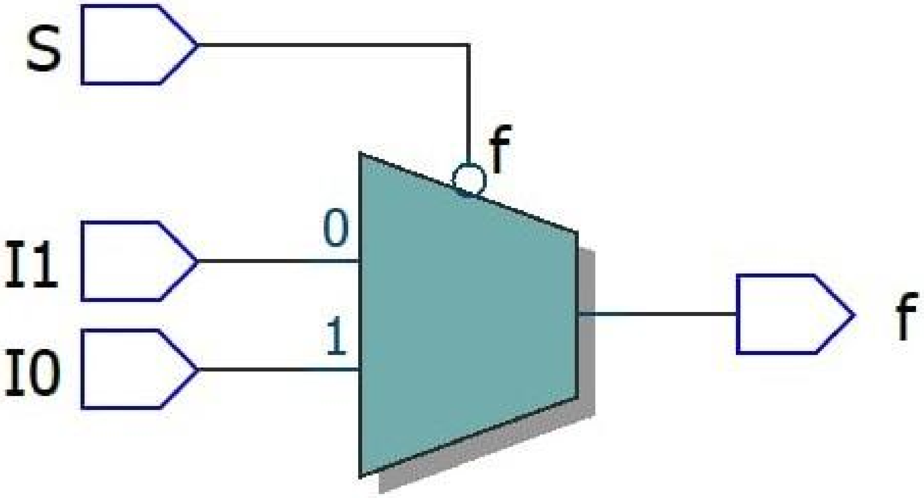
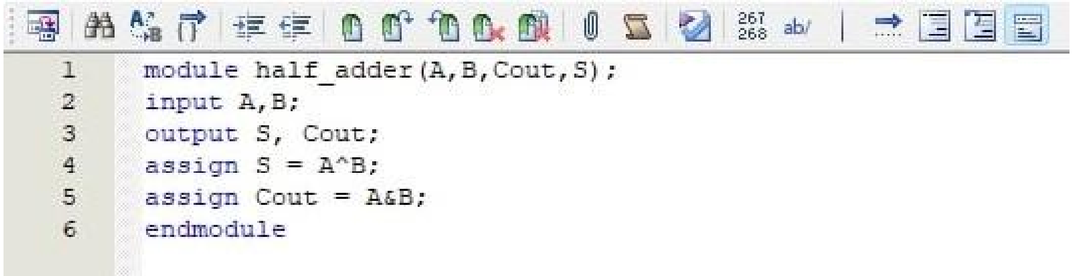


Figure 3: MUX RTL Viewer schematic

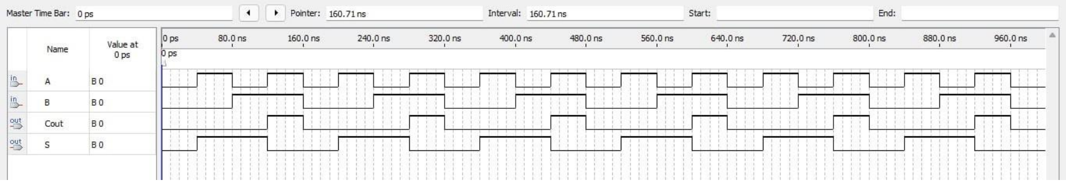
Half Adder

The half adder circuit was implemented and tested:



```
module half_adder(A,B,Cout,S);
input A,B;
output S, Cout;
assign S = A^B;
assign Cout = A&B;
endmodule
```

Figure 4: Code of Half Adder



Master Time Bar: 0 ps Pointer: 160.71 ns Interval: 160.71 ns Start: End:

Name	Value at 0 ps
A	B 0
B	B 0
Cout	B 0
S	B 0

Figure 5: Half Adder simulation waveform

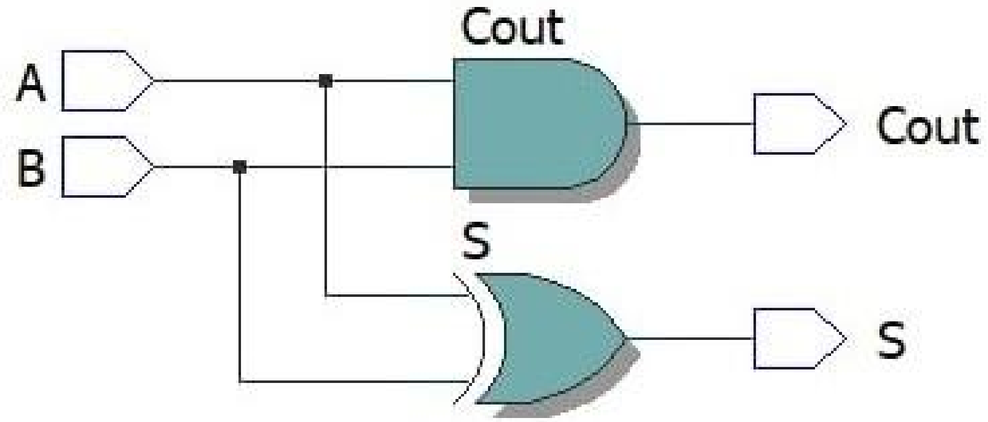
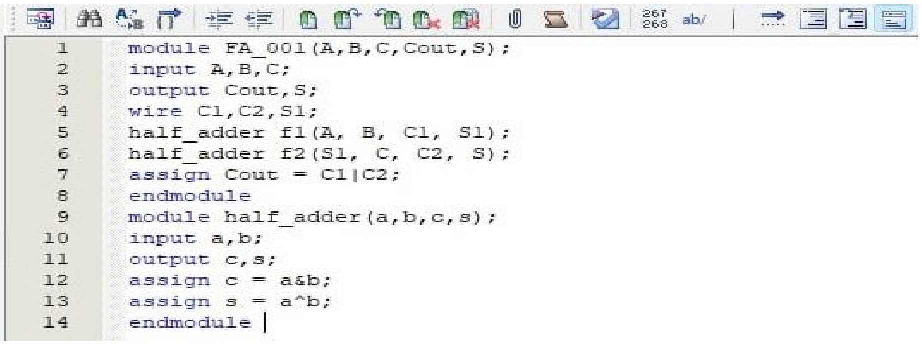


Figure 6: Half Adder RTL Viewer schematic

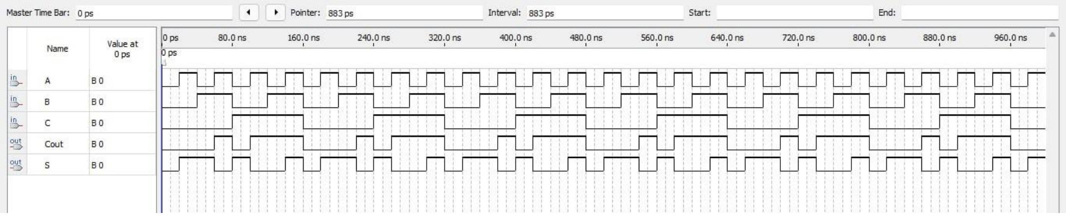
Full Adder

A full adder was constructed using two half adders



```
module FA_001(A,B,C,Cout,S);
input A,B,C;
output Cout,S;
wire C1,C2,S1;
half_adder f1(A, B, C1, S1);
half_adder f2(S1, C, C2, S);
assign Cout = C1|C2;
endmodule
module half_adder(a,b,c,s);
input a,b;
output c,s;
assign c = a&b;
assign s = a^b;
endmodule
```

Figure 7: Code of Full Adder



Master Time Bar: 0 ps

Pointer: 883 ps

Interval: 883 ps

Start:

End:

Name	Value at 0 ps
A	B 0
B	B 0
C	B 0
Cout	B 0
S	B 0

Figure 8: Full Adder simulation waveform

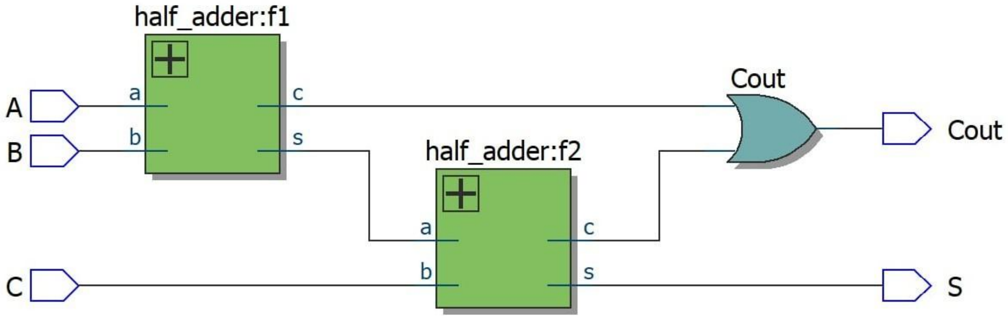
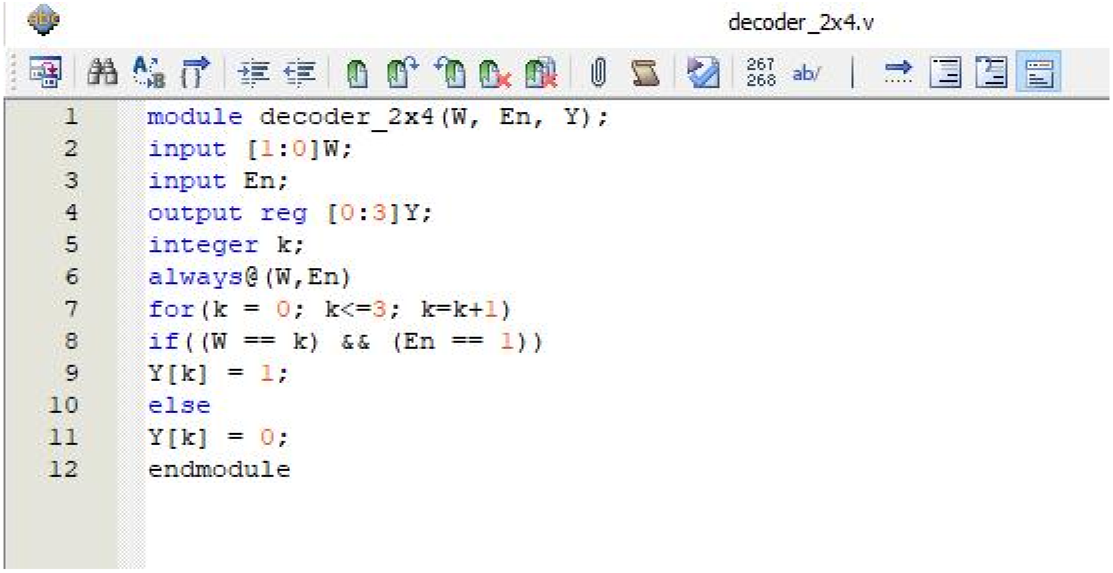


Figure 9: Full Adder RTL Viewer schematic

2x4 Decoder

The 2x4 decoder implementation demonstrated behavioral modeling:



```
module decoder_2x4(W, En, Y);
input [1:0]W;
input En;
output reg [0:3]Y;
integer k;
always@(W,En)
for(k = 0; k<=3; k=k+1)
if((W == k) && (En == 1))
Y[k] = 1;
else
Y[k] = 0;
endmodule
```

Figure 10: Code of Decoder

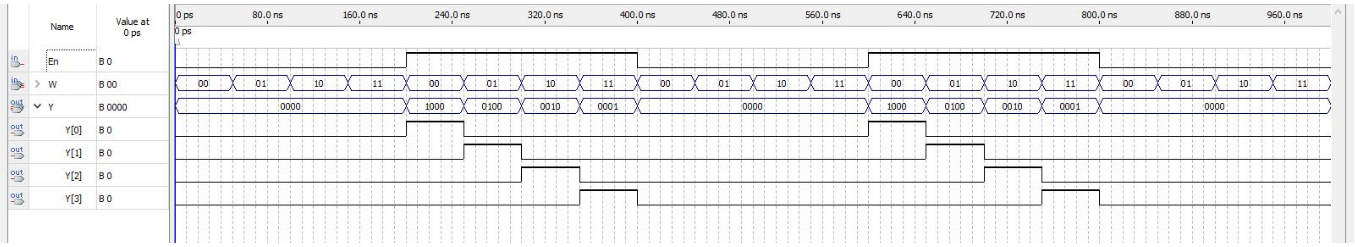


Figure 11: Decoder simulation waveform

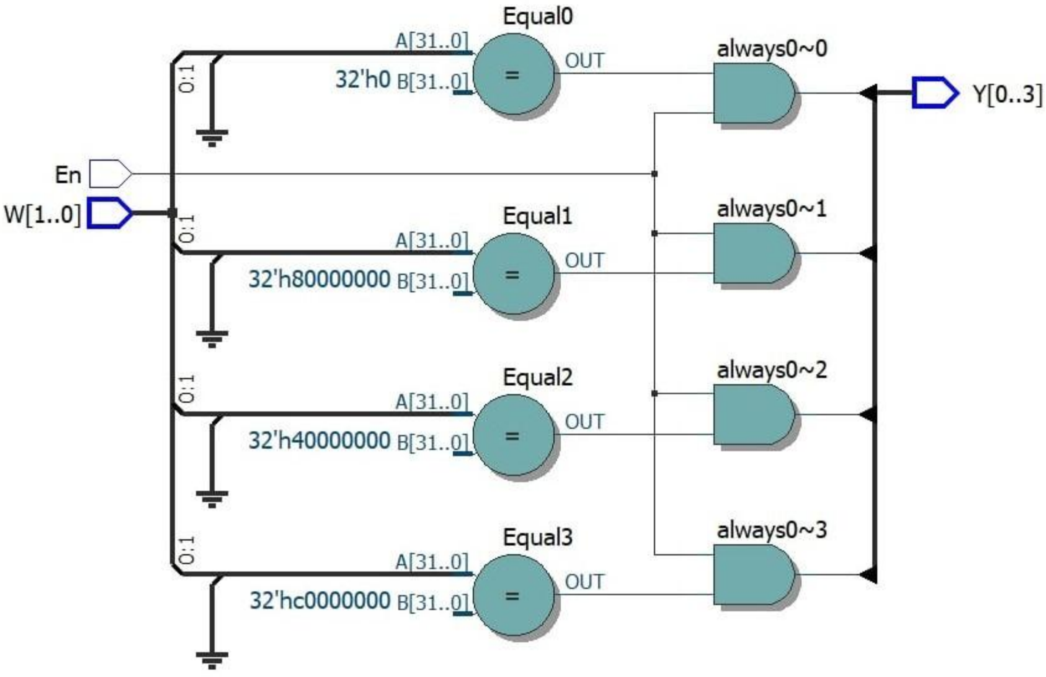


Figure 12: Decoder RTL Viewer schematic

Discussion

The implemented circuits were thoroughly tested and analyzed. The following observations were made:

- **RTL Viewer Schematics**: The synthesized hardware matched the expected design, ensuring that the high-level description of your circuit was correctly translated into hardware.
- **Half Adder Simulation**: The correct functionality of the sum and carry outputs for all input combinations confirms the proper implementation of XOR for sum and AND for carry.
- **Full Adder**: All possible input combinations were tested, showing the expected carry propagation across the cascaded half adders, validating the design.
- **2x4 Decoder**: The enable signal correctly gated the outputs, with the four output combinations being generated as expected, ensuring the correct operation of the decoder based on the input.
- **Timing Simulations**: The propagation delays observed only in timing simulations (not functional simulations) indicate realistic behavior under hardware constraints. These delays being within acceptable ranges means the design meets performance specifications for the device family.