CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Electronics and Telecommunication Engineering

VLSI Technology Sessional

ETE 404

Experiment No:09

Introduction to Verilog HDL and Quartus II

Submitted by:

Shamanza Chowdhury

ID: 1908008

Submitted to:

Arif Istiaque Rupom

Lecturer

Dept. of ETE,CUET

October 20, 2024

1 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

2 Apparatus:

- **Software:** Quartus II

3 2x1 Multiplexer Design:

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

3.1 HDL Code:

Listing 1: 2x1 Multiplexer in Verilog HDL

```
module mux2x1 (A, B, S, Y);
input A, B, S;
output Y;
assign Y = (~S & A) | (S & B);
endmodule
```

3.2 Behavior Analysis:

Functional Simulation:

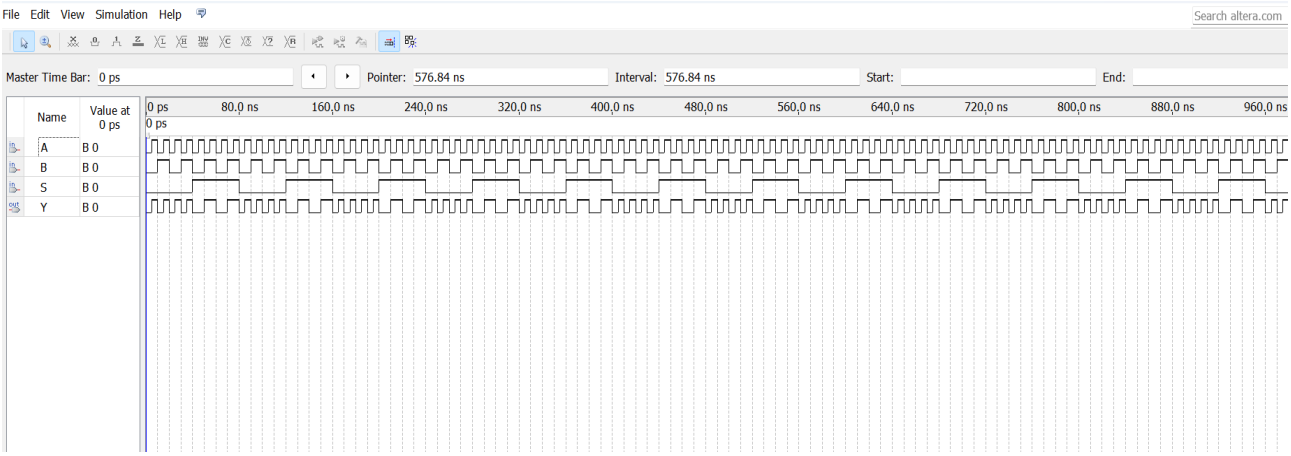

Figure 01:Waveform of Functional Simulation of Multiplexer.

Timing Simulation:

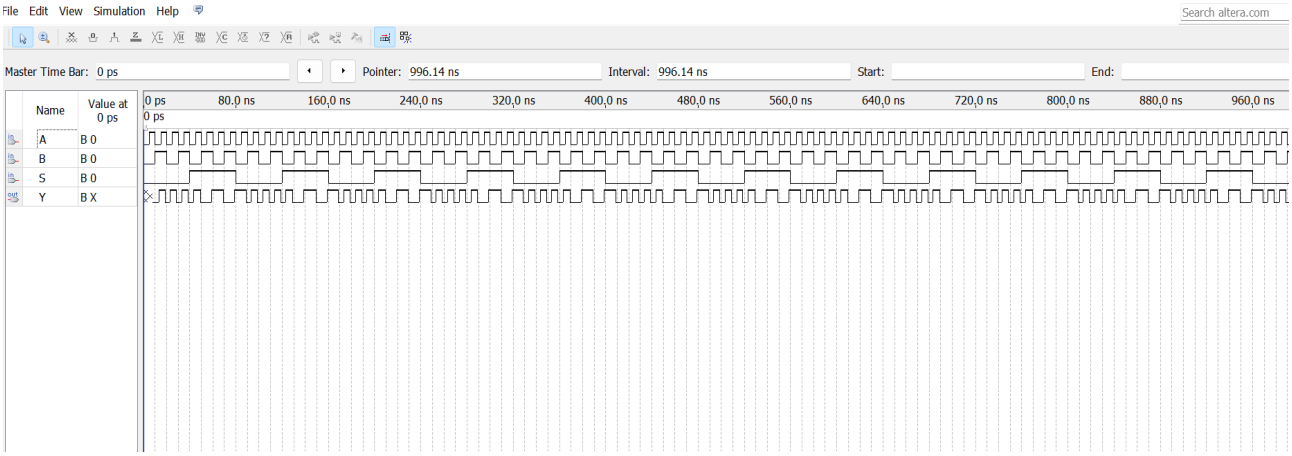

Figure 02:Waveform of Timing Simulation of Multiplexer.

3.3 RTL View:

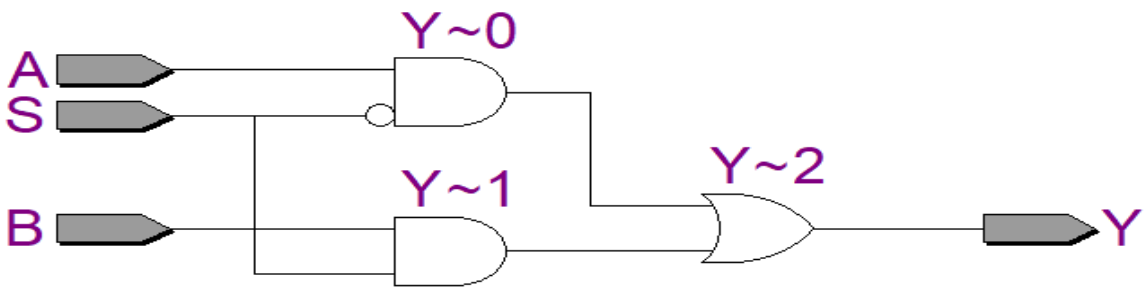

Figure 03: RTL View of Multiplexer.

4 Half-Adder Design:

The half-adder was implemented using the following Verilog code:

4.1 HDL Code:

Listing 2: Half-Adder in Verilog HDL

```
1 module half_adder(A,B,Cout,S);
2 input A,B;
3 output S, Cout;
4 assign S = A^B;
```

```
5 assign Cout = A&B;
6 endmodule
```

4.2 Behavior Analysis:

Functional Simulation:

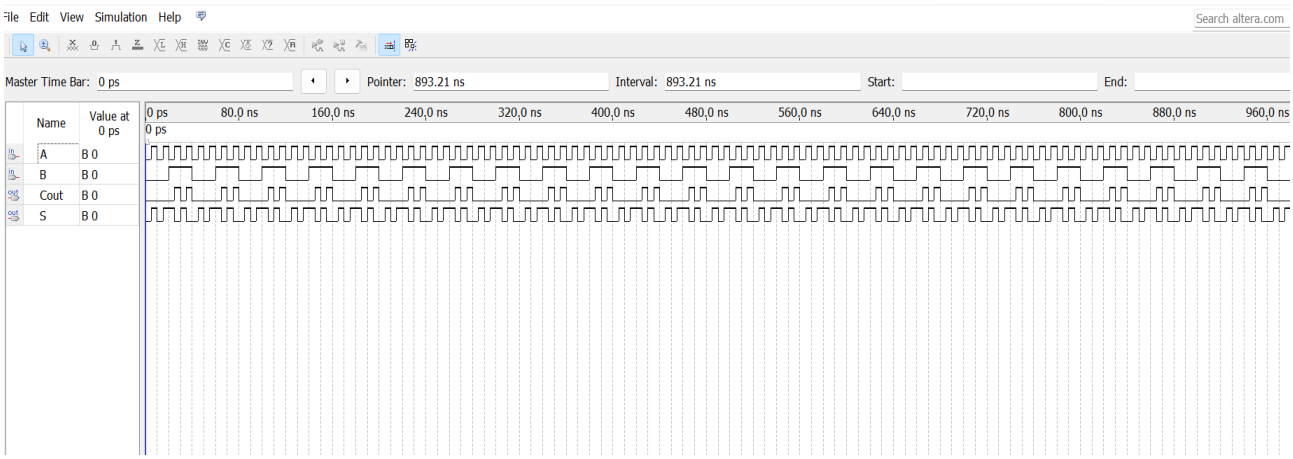

Figure 04:Waveform of Functional Simulation of Half-Adder.

Timing Simulation:

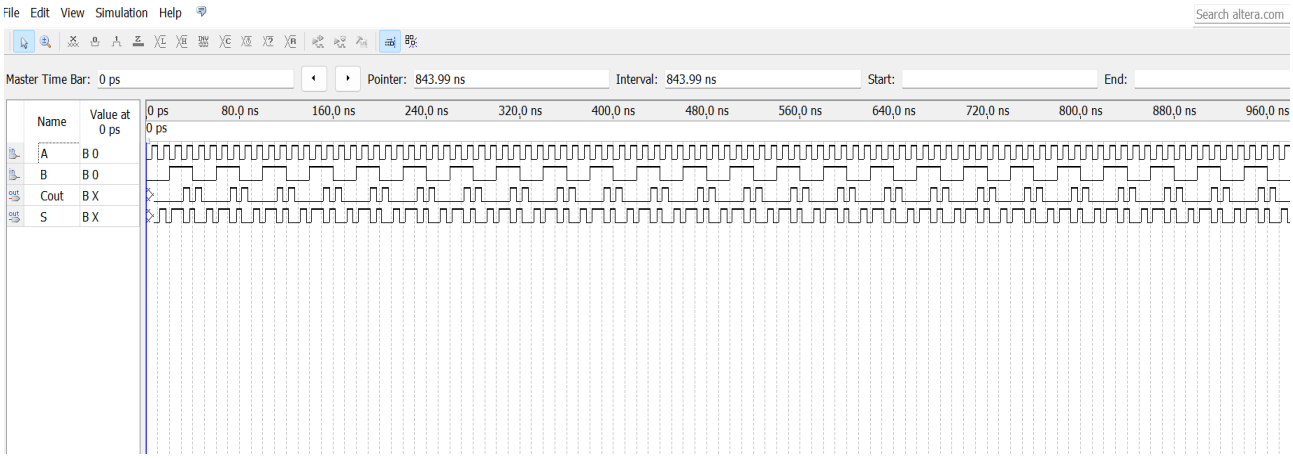

Figure 05:Waveform of Timing Simulation of Half-Adder.

4.3 RTL View:

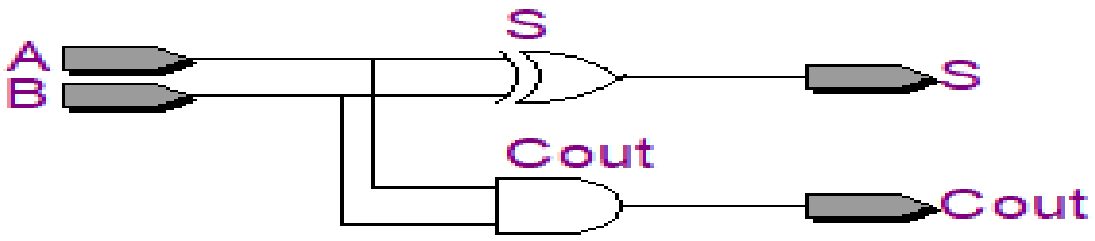

Figure 06: RTL View of Half-Adder.

5 Full-Adder Design:

The full-adder was implemented using the following Verilog code that consist of two half adders:

5.1 HDL Code:

Listing 3: Full-Adder using Half-Adder in Verilog HDL

```
module FA_001(A,B,C,Cout,S);
input A,B,C;
output Cout,S;
wire C1,C2,S1;
HA_001 f1(A, B, C1, S1);
HA_001 f2(S1, C, C2, S);
assign Cout = C1|C2;
endmodule
```

Listing 4: Half-Adder in Verilog HDL

```
module HA_001(a,b,c,s);
input a,b;
output c,s;
assign c = a&b;
assign s = a^b;
endmodule
```

5.2 Behavior Analysis:

Functional Simulation:

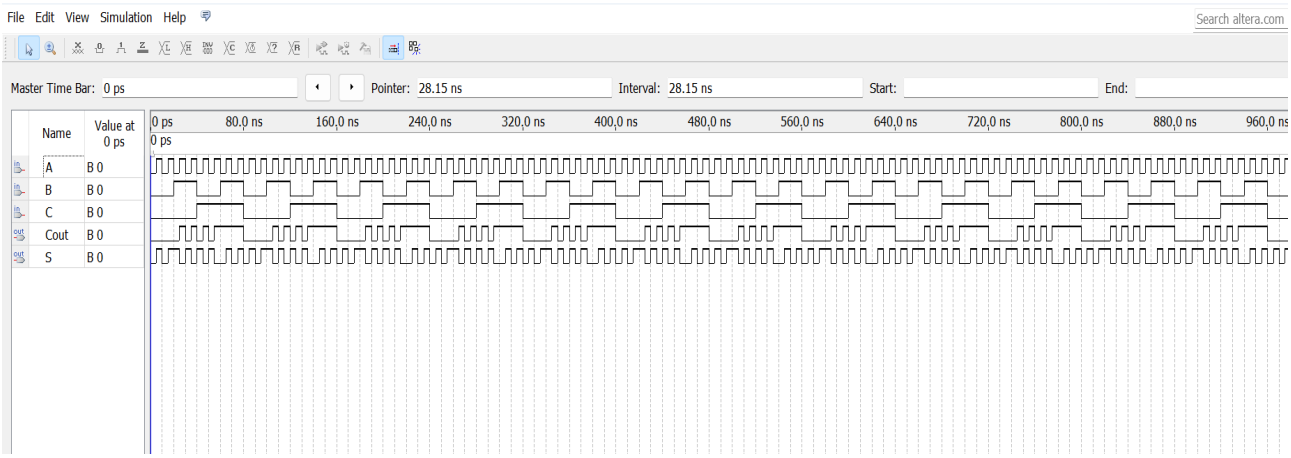

Figure 07:Waveform of Functional Simulation of Full-Adder.

Timing Simulation:

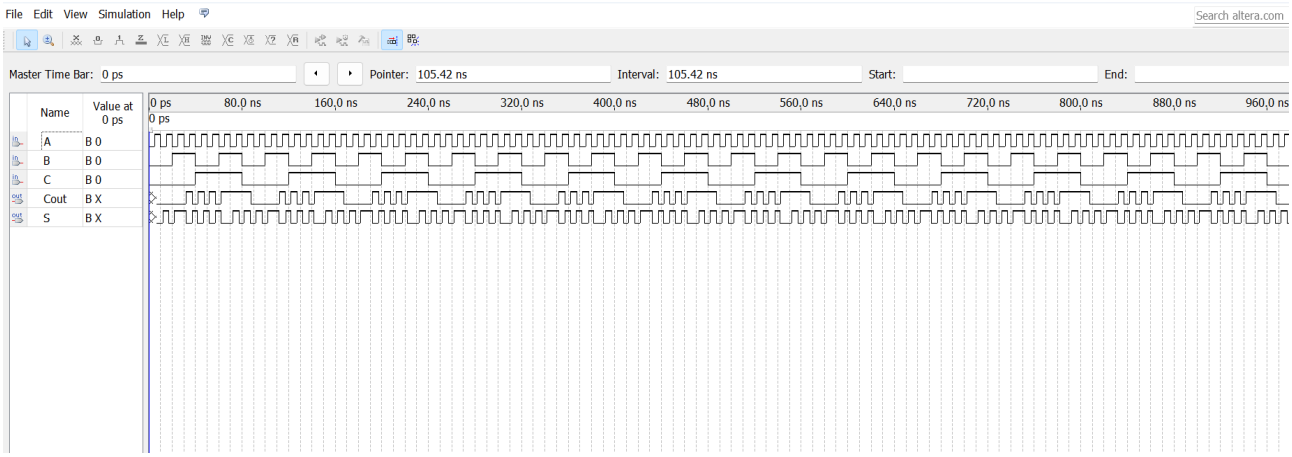

Figure 08:Waveform of Timing Simulation of Full-Adder.

5.3 RTL View:

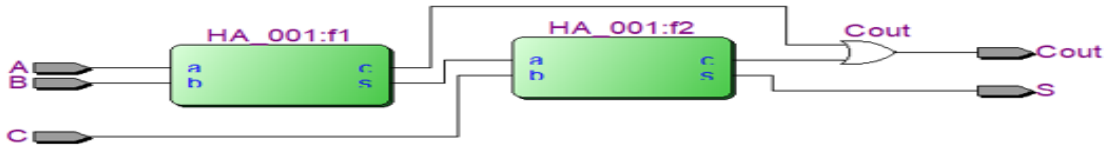

Figure 09: RTL View of Full-Adder.

6 2x4 Decoder Design:

6.1 HDL Code:

The 2x4 decoder was implemented using the following Verilog code:

Listing 5: 2x4 Decoder in Verilog HDL

```
module dec2to4(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
```

6.2 Behavior Analysis:

Functional Simulation:

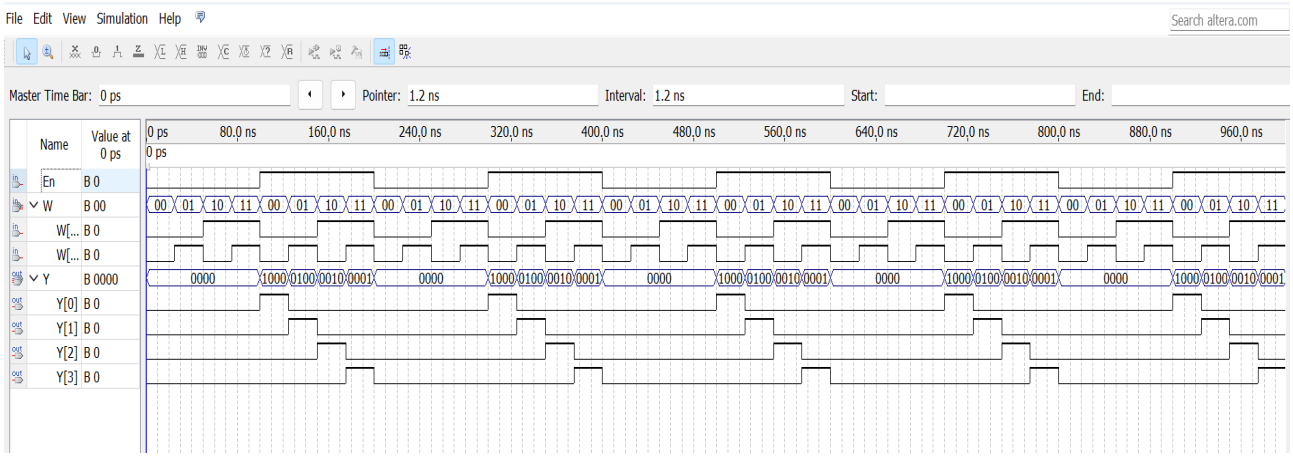

Figure 01:Waveform of Functional Simulation of Decoder.

Timing Simulation:

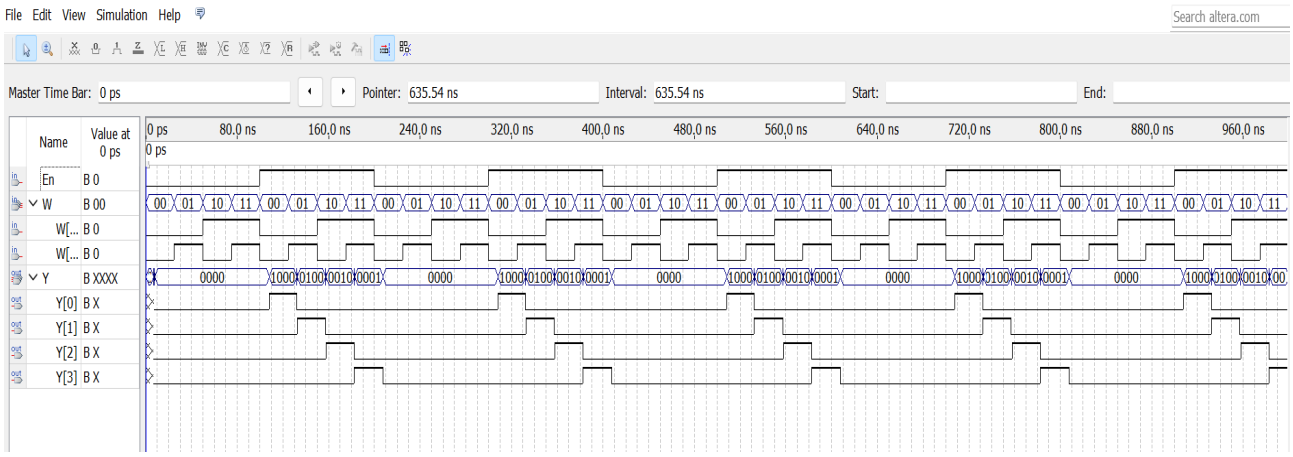

Figure 02:Waveform of Timing Simulation of Decoder.

6.3 RTL View:

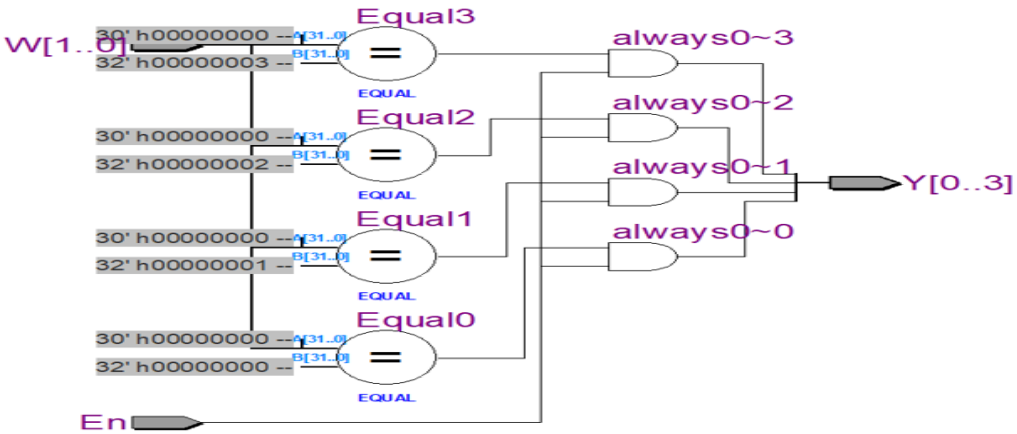

Figure 03: RTL View of Decoder.

7 Discussion:

- The experiment focused on implementing and simulating various digital designs in Verilog HDL using Intel Quartus II and ModelSim-Altera for behavioral and structural modeling.
- The RTL Viewer in Quartus II confirmed that the synthesized hardware matched the intended high-level descriptions for each design.
- The functional simulations of the 2x1 Multiplexer, Half-Adder, Full-Adder and 2x4 Decoder verified their correct functionality, with outputs responding as expected to various input combinations.

- Timing Analysis and functional analysis were observed. Timing simulations revealed propagation delays that were absent in functional simulations, offering a realistic perspective on circuit behavior under actual hardware conditions.
- Project file name and module name should be same.