International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 6, Issue 4 (April 2017), PP.38-42

Synthesises and Simulation of Binary AND Gate using Behavioral Level Modelling

Noor Ul Abedin,
Assistant Professor, Department of Electronics and Communication Engineering
Deccan College of Engineering and Technology

Abstract: The implementation of basic binary logic gates is the basic fundamental background in digital electronics. Thus it is our duty to make programming much simpler and applicable to all the possible forms of program modelling which leads to the different modelling techniques in VLSI designing i.e. Gate level modelling, Dataflow level modelling, Behavioral level modelling, Switch level modelling and Mixed level modelling techniques.
Keywords: AND gate, Verilog HDL Behavioral level modelling, Xilinx 14.7, Synthesis and Simulation.

I. INTRODUCTION

Familiarity with the binary logic gate AND which performs basic multiplication operation of two binary input signals, also familiar with its synthesis and simulation with 'Verilog HDL' using software aid such as Xilinx Software. In behavioral level the module is implemented in terms of desired design algorithm without concern of details of the hardware implementation. Here I am showcasing how a binary logic AND gate can be synthesised and simulated using behavioural modelling technique. Briefly, basically in synthesis we design the logic circuit layout of the AND gate and in the simulation we verify all of its logical combinations, i.e. verify its truth table. In simulation the logical verification is governed in the form of a timing diagram.

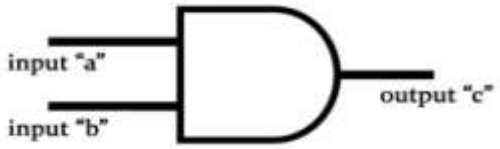


Fig.1. Circuit Symbol of AND Gate

Input a	Input b	Output c (c=a.b)
0	0	0
0	1	0
1	0	0
1	1	1

Table.1. Truth Table of AND Gate

In the implementation of AND gate using behavioural level modelling, module name is myandgate with inputs: a, b and output: c. The output here is taken a register continuing with assigning the inputs with always statement and using relative operator to perform logical AND operation.

II. VERILOG HDL DESCRIPTION

2.1. Module Program:

```
module myandgate(c,a,b); // module name: myandgate
input a,b;    // assigning inputs: a and b
output c;// assigning output: c
reg c; // assigning output as register
always @ (a,b) // assigning inputs: a and b using always statement
begin // beginning the initialization
c=a&b;  // c, a and b are operands and & is operator
end // end of initialization
endmodule // end of module/ program
```

The module program of Verilog HDL serves the purpose for providing the binary AND gate schematic layout.

2.2. Test bench/ Simulation Program:

```
module myandgatetest; // test bench module name: myandgatetest
reg a,b;  // considering inputs a and b as registers
wire c;   // considering the output c as a wire
myandgate uut (c,a,b); // considering our module program in unit under test (uut)
initial
begin
$monitor ($time,"a=%b,b=%b,c=%b",a,b,c);     // monitoring and display
#100 a=0;b=0;                                 // logic combination 1
#100 a=0;b=1;                                 // logic combination 2
#100 a=1;b=0;                                 // logic combination 3
#100 a=1;b=1;                                 // logic combination 4
end                                                  // end of initialization
endmodule                                     // end of test module/ program
```

The simulation/ test bench program associates the module program to verify the logical combinations of the gate.

III. WORKING ON XILINX 14.7

In this paper synthesis and simulation of a two input binary AND gate using Behavioral level modelling technique using by stating the output of the AND gate as register along with all the necessary statements. Synthesis and simulation is performed using Xilinx 14.7 software.

a. Synthesis performed on Xilinx 14.7

```
19  //
20  //////////////////////////////////
21  module myandgate(c,a,b);
22  input a,b;
23  output c;
24  reg c;
25  always @ (a,b)
26  begin
27  c=a&b;
28  end
29  endmodule
30
```

myandgate.v Design Sum

Fig.2. Module Program

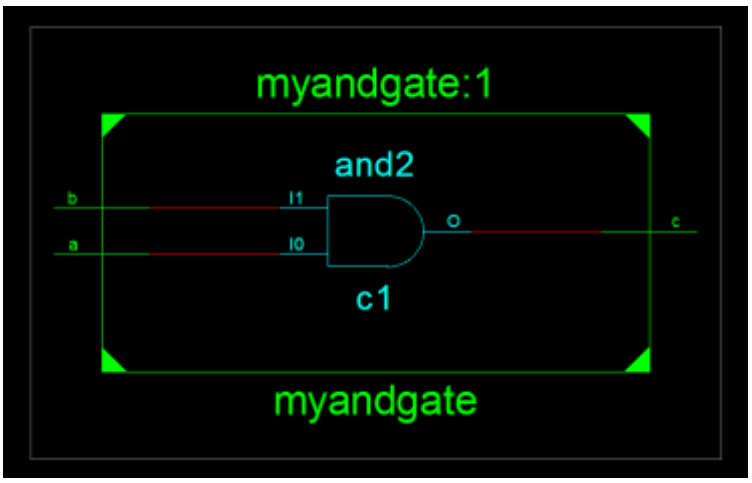
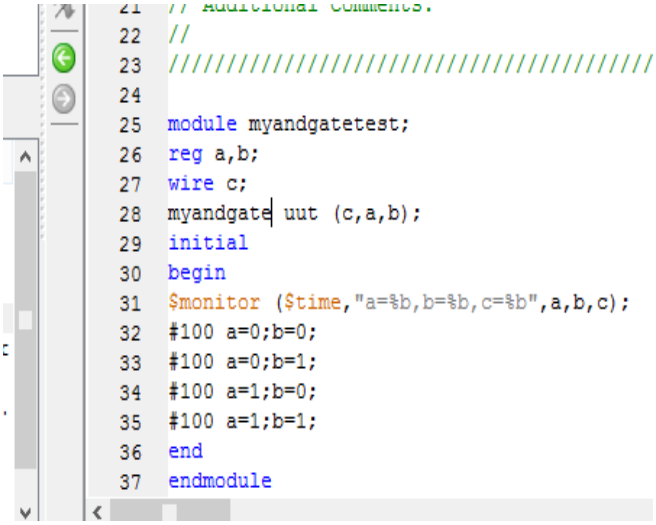


Fig.3. Synthesised Output

b. Performing of Simulation on Xilinx 14.7



```
21  // Additional Comments:
22  //
23  //////////////////////////////////////////
24
25  module myandgatetest;
26  reg a,b;
27  wire c;
28  myandgate uut (c,a,b);
29  initial
30  begin
31  $monitor ($time,"a=%b,b=%b,c=%b",a,b,c);
32  #100 a=0;b=0;
33  #100 a=0;b=1;
34  #100 a=1;b=0;
35  #100 a=1;b=1;
36  end
37  endmodule
```

Fig.4. Test Bench/ Simulation Program

Observed Simulated Outputs:

a. With inputs a=0, b=0 and output c=0

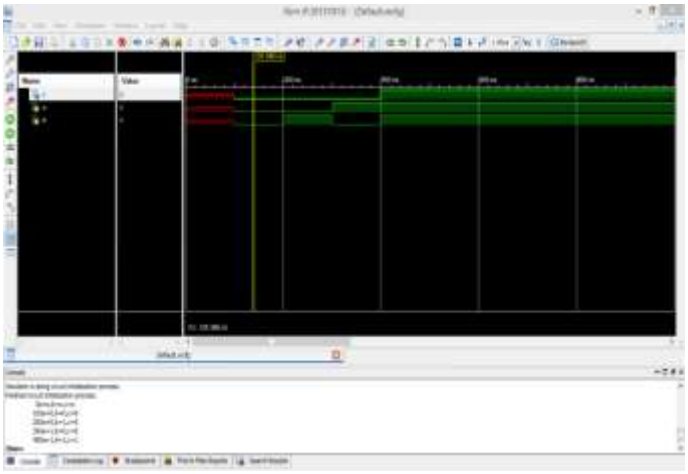


Fig.5.1.

b. With inputs a=0, b=1 and output c=0

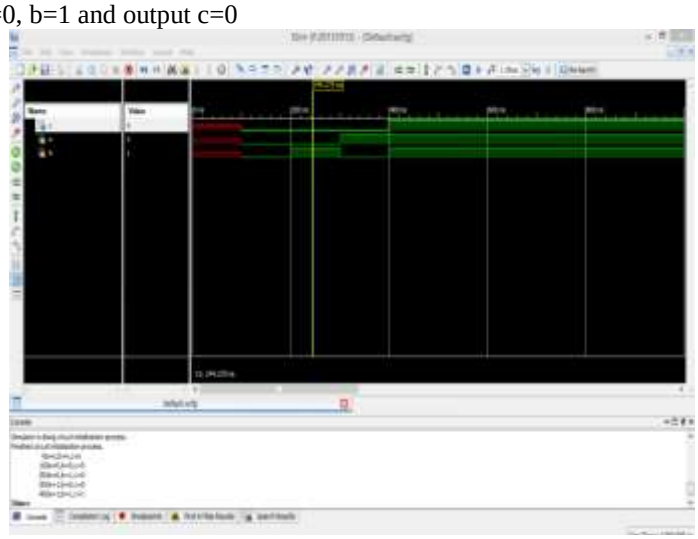


Fig.5.2.

c. With inputs a=1, b=0 and output c=0

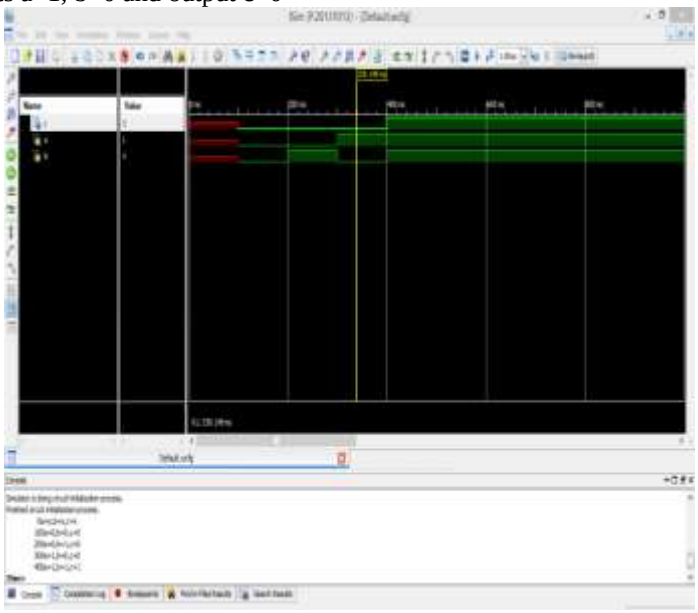


Fig.5.3

d. With inputs a=1, b=1 and output c=1

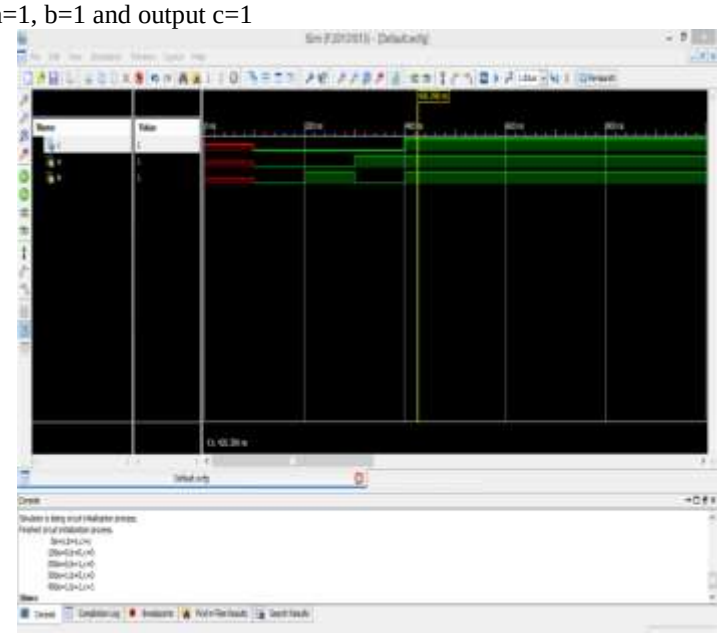


Fig.5.4.

IV. FUTURE SCOPE

Basic programming descriptions can be made easier which can be helpful in developing complex VLSI circuits.

V. CONCLUSION

Convincingly it has been observed that an binary AND gate can be easily simulated and synthesized by considering the output as a register in the module program, other than using conditional or casing statements in Behavioral level modelling.

VI. AUTHOR DETAILS

Mr. Noor Ul Abedin is working as an Assistant Professor in Dept. of Electronics and Communication, Deccan College of Engineering and Technology. He's field of interest are Analog and Digital Electronics.
He has lectured and thought various engineering subjects which include Anlaog Electronic Devices & Circuits, VLSI Design, Networks & Circuits, Antennas & Wave Propagation, Electronic Instrumentation, and

Automation in Process Control, Power Electronics, Digital Electronics & Logic Design and Linear Integrated Circuits.

REFERENCES

[1]. Verilog HDL by Samir Palnitkar
[2]. A Verilog HDL Primer by J. Bhasker
[3]. Introduction to VLSI Circuits and Systems by John P. Uyemura
[4]. The 8051 Microcontroller & Embedded Systems by Muhammad Ali Mazidi Janice Gillispie Mazidi and Rolin D. McKinlay
[5]. VLSI Design by P Sahu
[6]. VLSI Design by M. Michael Vai
[7]. Digital Systems by Tocci & Widmer
[8]. Digital Design by Morris Mano
[9]. Digital Hardware Design by John B. Peatman