VLSI Technical Skilling Major Project on

2-BIT DECIMAL COMPARATOR

COURSE CODE :17TS403

A Major Project based lab report submitted to

Submitted in partial fulfilment of the award of degree

BACHELOR OF TECHNOLOGY

(ECE)

KLEF

Submitted By

Mushtaq Hussain Shaik (170040582)

Under the esteemed guidance of

NAME OF THE FACULTY:MR.EDIGA RAGHUVEER

(DEEMED TO BE UNIVERSITY)

KONERU LAKSHMAIAH
EDUCATION FOUNDATION
(Deemed to be University, Estd. u/s. 3 of UGC Act 1956)

Koneru Lakshmaiah Educational Foundation
Vaddeswaram-522502
Guntur (Dist)

CERTIFICATION

This is to certify that the project based laboratory report entitled “**2-BIT DECIMAL COMPARATOR**” submitted by **Mushtaq Hussain Shaik (170040582),** bearing to the **Department of Electronics and Communication Engineering, K L University** in partial fulfillment of the requirements for the completion of a project based Laboratory in “SKILLING AND DEVELOPMENT” course in III B Tech I Semester, is a bonafide record of the work carried out by him/her under my supervision during the academic year 2019-2020.

PROJECT SUPERVISOR	**HEAD OF THE DEPARTMENT**
Mr.Ediga Raghuveer	**Dr.Suman Maloji**

ACKNOWLEDGEMENT

It is great pleasure for us to express our gratitude to our honorable President **Sri.Koneru Satyanarayana** for giving the opportunity and platform with facilities in accomplishing the project based laboratory report

We express our sincere gratitude to our principal **Dr. K Subba Rao** for his administration towards our academic growth.

We express our sincere gratitude to our Coordinator Mr. John Philip for his leadership and constant motivation provided in successful completion of our academic semester.

We record it as our privilege to deeply thank our pioneer **Dr.Suman Maloji** HOD-ECE for providing us the efficient faculty and facilities to make our ideas into reality.

We express our sincere thanks to our project supervisor **Mr.G.Rakesh** for his novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

Mushtaq Hussain Shaik(170040582)

ABSTRACT

In this project, a simple **7**-bit **comparator** is designed and implemented in Verilog HDL. Truth table, K-Map and minimized equations for the comparator are presented. The Verilog code of the comparator is simulated by ModelSim and the simulation waveform is presented. The main aim of this project is to compare the greatest number in the given two digit integer number. In this project the given two digit decimal number will be converted to binary by dividing with 2 and it compares the two decimal number which is greater and displays the greatest one on LCD display.

The specification of the 7-bit comparator is as follows:
Input: 7-bit A and B for comparison

Output:

A_greater_B: high if A > B else low

A_equal_B: high if A = B else low

A_less_B: high if A<B else low

Example : 2 3

Let A=2 , B=3, A<B.

So, in LCD it displays A<B.

CONTENTS

INTRODUCTION

Verilog is a HARDWARE DESCRIPTION LANGUAGE (HDL). It is a language used for describing a digital system like a network switch or a microprocessor or a memory or a flip−flop. It means, by using a HDL we can describe any digital hardware at any level. Designs, which are described in HDL are independent of technology, very easy for designing and debugging, and are normally more useful than schematics, particularly for large circuits.

Verilog supports a design at many levels of abstraction. The major three are −

- Behavioral level
- Register-transfer level
- Gate level

Behavioral level

This level describes a system by concurrent algorithms (Behavioural). Every algorithm is sequential, which means it consists of a set of instructions that are executed one by one. Functions, tasks and blocks are the main elements. There is no regard to the structural realization of the design.

Register−Transfer Level

Designs using the Register−Transfer Level specify the characteristics of a circuit using operations and the transfer of data between the registers. Modern definition of an RTL code is "Any code that is synthesizable is called RTL code".

Gate Level

Within the logical level, the characteristics of a system are described by logical links and their timing properties. All signals are discrete signals. They can only have definite logical values (`0', `1', `X', `Z`). The usable operations are predefined logic primitives (basic gates). Gate level modelling may not be a right idea for logic design. Gate level code is generated using tools like synthesis tools and his netlist is used for gate level simulation and for backend.

Lexical Tokens

Verilog language source text files are a stream of lexical tokens. A token consists of one or more characters, and each single character is in exactly one token.

The basic lexical tokens used by the Verilog HDL are similar to those in C Programming Language. Verilog is case sensitive. All the key words are in lower case.

White Space

White spaces can contain characters for spaces, tabs, new-lines and form feeds. These characters are ignored except when they serve to separate tokens.

White space characters are Blank space, Tabs, Carriage returns, New line, and Form feeds.

Numbers

You can specify a number in binary, octal, decimal or hexadecimal format. Negative numbers are represented in 2’s compliment numbers. Verilog allows integers, real numbers and signed & unsigned numbers.

The syntax is given by − <size> <radix> <value>

Size or unsized number can be defined in <Size> and <radix> defines whether it is binary, octal, hexadecimal or decimal.

Identifiers

Identifier is the name used to define the object, such as a function, module or register. Identifiers should begin with an alphabetical characters or underscore characters. Ex. A_Z, a_z,_

Identifiers are a combination of alphabetic, numeric, underscore and $ characters. They can be up to 1024 characters long.

Operators

Operators are special characters used to put conditions or to operate the variables. There are one, two and sometimes three characters used to perform operations on variables.

Data Types

Value Set

Verilog consists of, mainly, four basic values. All Verilog data types, which are used in Verilog store these values −

0 (logic zero, or false condition)

1 (logic one, or true condition)

x (unknown logic value)

z (high impedance state)

use of x and z is very limited for synthesis.

Wire

A wire is used to represent a physical wire in a circuit and it is used for connection of gates or modules. The value of a wire can only be read and not assigned in a function or block. A wire cannot store value but is always driven by a continuous assignment statement or by connecting wire to output of a gate/module. Other specific types of wires are −

Wand (wired-AND) − here value of Wand is dependent on logical AND of all the device drivers connected to it.

Wor (wired-OR) − here value of a Wor is dependent on logical OR of all the device drivers connected to it.

Tri (three-state) − here all drivers connected to a tri must be z, except only one (which determines value of tri).

FPGA BASYS 3 BOARD

The Basys 3 is an entry-level FPGA development board designed exclusively for Vivado Design Suite, featuring Xilinx Artix-7 FPGA architecture. Basys 3 is the newest addition to the popular Basys line of FPGA development boards, and is perfectly suited for students or beginners just getting started with FPGA technology. The Basys 3 includes the standard features found on all Basys boards: complete ready-to-use hardware, a large collection of on-board I/O devices, all required FPGA support circuits, a free version of development tools, and at a student-level price point.

More I/O:

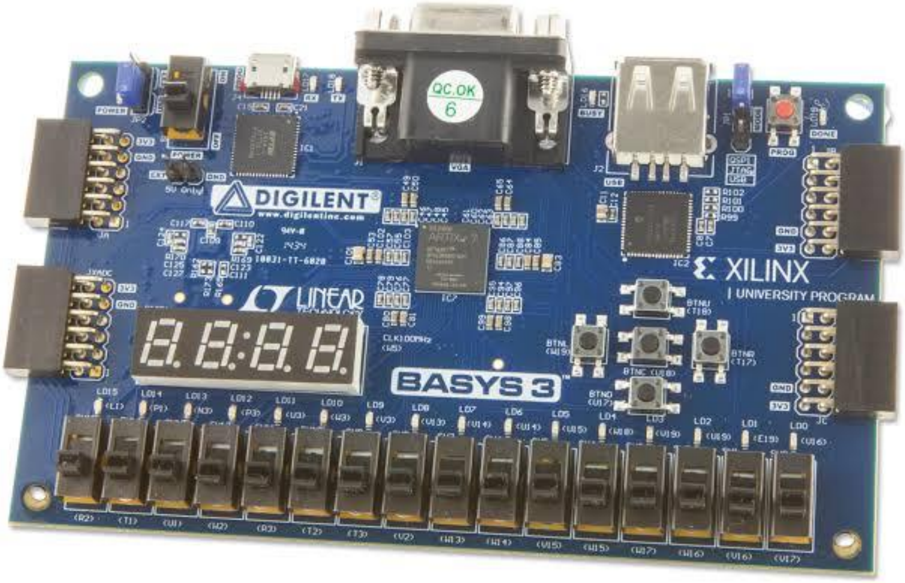
Double the user interface switches, double the number of onboard outputs, upgraded the number of external ports (moving from 6-pin single-row Pmods to 12-pin double-row Pmods) and included for the first time on a Basys class device a USB-UART bridge.

Modern Architecture and Modern Programming Challenges:

Due to the migration from the Spartan-3E family to the Artix-7 class of device, the Basys 3 offers a substantial increase in hardware capabilities. With the new Artix FPGA comes 15X the logic cells (from 2,160 to 33,280) and the upgrade from multipliers to true DSP slices. It also adds over 26X the amount of RAM.

Industry's First SoC Strength Design Suite:

The most significant change to the Basys 3 is the upgrade to Xilinx Vivado Design Suite (WebPACK edition available for free download from Xilinx), the most modern design tool chain used by professional engineers worldwide. Compared to ISE®, Vivado offers an improved user experience and expanded capabilities. These capabilities include block-based IP integration (which can reduce development time up to 10x) and the Vivado Logic/Serial I/O analyzer*****.

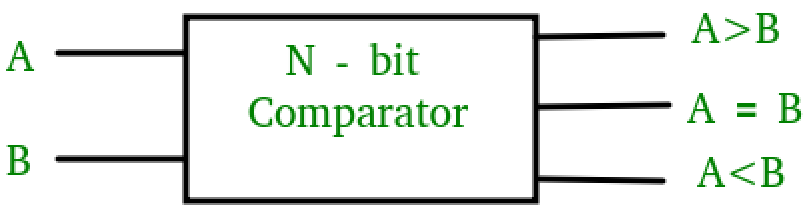


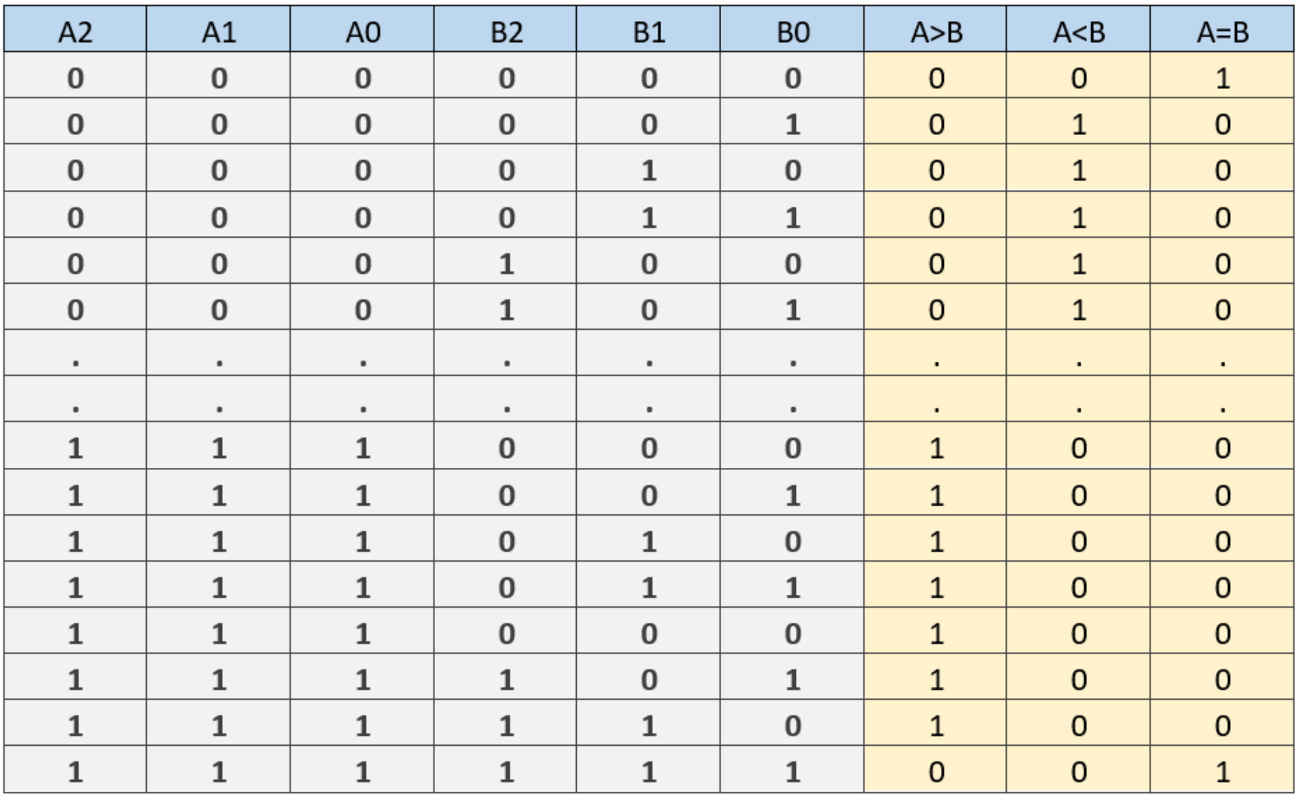
Features:

- Features the Xilinx Artix-7 FPGA: XC7A35T-1CPG236C
- 33,280 logic cells in 5200 slices (each slice contains four 6-input LUTs and 8 flip-flops)
- 1,800 Kbits of fast block RAM
- Five clock management tiles, each with a phase-locked loop (PLL)
- 90 DSP slices
- Internal clock speeds exceeding 450 MHz
- On-chip analog-to-digital converter (XADC)
- Digilent USB-JTAG port for FPGA programming and communication
- Designed Exclusively for Vivado Design Suite (Vivado Design Suite WebPACK edition can be downloaded for free from Xilinx). Expanded features are available through purchase of the Design Edition.
- Free WebPACK™ download for standard use.
- Micro-B USB cable not included.
- Serial Flash
- USB-UART Bridge
- 12-bit VGA output
- USB HID Host for mice, keyboards and memory sticks
- 16 user switches
- 16 user LEDs
- 5 user pushbuttons
- 4-digit 7-segment display
- 4 Pmod ports: 3 Standard 12-pin Pmod ports, 1 dual purpose XADC signal / standard Pmod port

DESCRIPTION

A magnitude digital Comparator is a combinational circuit that compares two digital or binary numbers in order to find out whether one binary number is equal, less than or greater than the other binary number. We logically design a circuit for which we will have two inputs one for A and other for B and have three output terminals, one for A > B condition, one for A = B condition and one for A < B condition.





A2	A1	A0	B2	B1	B0	A>B	A<B	A=B
0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	1	0
0	0	0	0	1	0	0	1	0
0	0	0	0	1	1	0	1	0
0	0	0	1	0	0	0	1	0
0	0	0	1	0	1	0	1	0
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
1	1	1	0	0	0	1	0	0
1	1	1	0	0	1	1	0	0
1	1	1	0	1	0	1	0	0
1	1	1	0	1	1	1	0	0
1	1	1	0	0	0	1	0	0
1	1	1	1	0	1	1	0	0
1	1	1	1	1	0	1	0	0
1	1	1	1	1	1	0	0	1

The half adder adds two one-bit binary numbers (AB). The output is the sum of the two bits (S) andthecarry(C).Notehowthesametwoinputsaredirectedtotwodifferentgates.Theinputsto the XOR gate are also the inputs to the ANDgate.

Verilog Code

```
module comparator_7(input [6:0]a,b,output e,g,l);
assign e=a==b;
assign g=a>b;
assign l=a<b;
endmodule
```

Test Bench

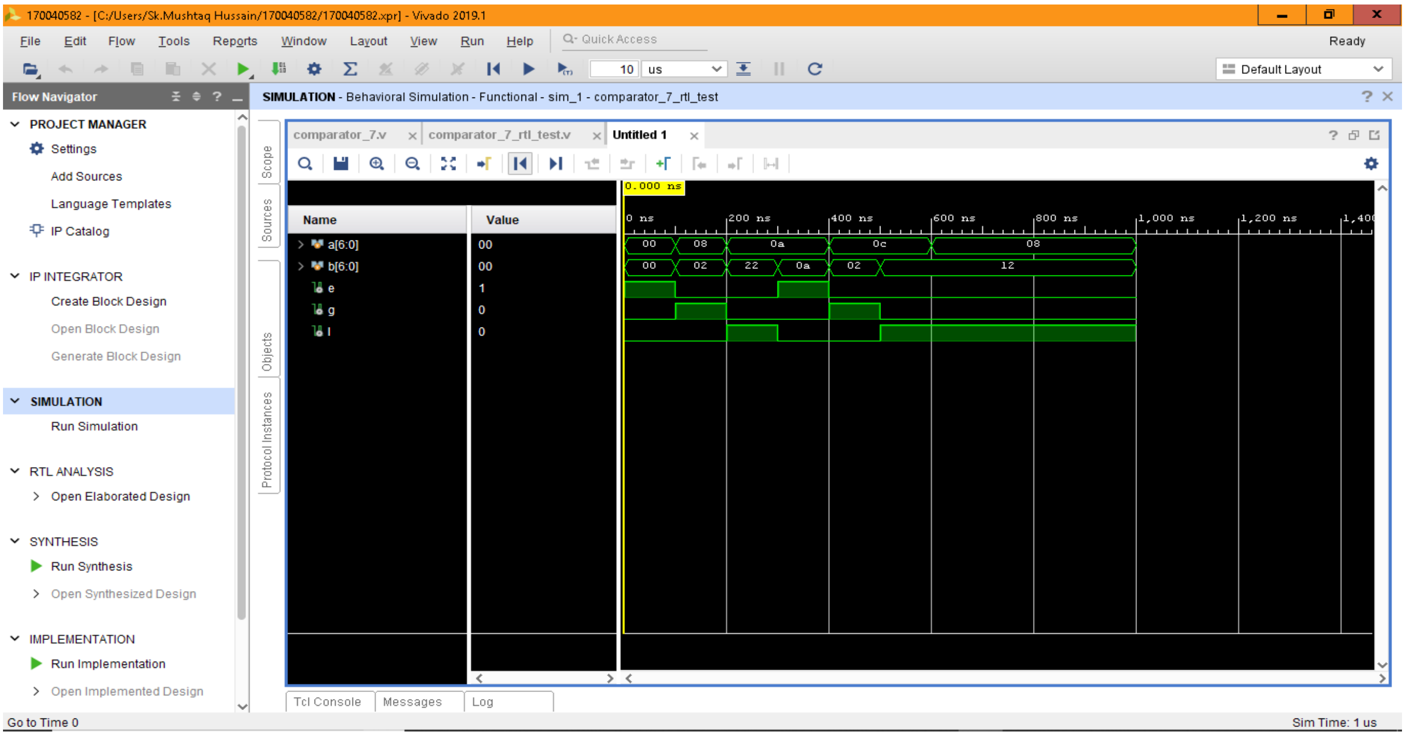
```
module comparator_7_rtl_test( );
reg [6:0]a,b;
wire e,g,l;
comparator_7 dut(.a(a),.b(b),.e(e),.g(g),.l(l));
initial
begin
a=7'b000000;b=7'b0000000;#100
a=7'b0010000;b=7'b0000010;#100
a=7'b0010010;b=7'b1000010;#100
a=7'b0001010;b=7'b0010010;#100
a=7'b0011000;b=7'b0000100;#100
a=7'b0011000;b=7'b0100100;#100
a=7'b0010010;b=7'b0100100;#100
a=7'b0000000;b=7'b0100000;#100
a=7'b0001100;b=7'b0010100;#100
a=7'b0010100;b=7'b1100100;#100
a=7'b0010010;b=7'b0011100;#100
a=7'b0010000;b=7'b0010010;#100
a=7'b0000100;b=7'b0100110;#100
a=7'b0011000;b=7'b1100010;#100

end
endmodule
```

Verilog Restrictions for Synthesis

- Not all HDL constructs are synthesizable.
- Simulatable designs are not necessarily synthesizable.
- Synthesizable constructs are tool dependent
- Use only few HDL commands
- case
- if else
- concurrent and sequential statements
- Keep the intended circuit architecture in mind during design description.
- Using C-like programming style increases the silicon area dramatically.
- Type conversions and test stimuli definitions cannot be synthesized.
- Make extensive use of comments.
- Use headers for all modules, functions
- Explain the operating modes of the modules
- Explain all input and output signals
- Compiler directives reside within comments
- Smallest HDL code does not imply smallest silicon.
- Describe the architecture clearly.
- Cover all possible states within a if-else or case statement.
- Do not use nested loops for circuit description
- Do not define functions when instantiating parts within one entity.

RESULTS



170040582 - [C:/Users/Sk.Mushtaq Hussain/170040582/170040582.xpr] - Vivado 2019.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - comparator_7_rtl_test

Name	Value
a[6:0]	00
b[6:0]	00
e	1
g	0
l	0

Go to Time 0

Sim Time: 1 us

Behavioural Simulation

CONCLUSION& FUTURE SCOPE

This project is useful in many industries and public transports like Airports , Railway stations ,bus stations ,banks , offices etc. The major application of this project is Ascending the token numbers of people who are in queue , sorting the numbers of trains in railway stations and displaying on the display board, Displaying the sorted numbers of buses in bus stations ,flight numbers in airports ,sorting the passengers list in all the categories mentioned above.

This project deals with four digit numbers and four inputs only, it can be developed with more complexity and more variety using verilog code .This code can be modified and dumped into baysis for displaying the output on the 7 segment led display.

REFERENCES/BIBLIOGRAPHY

https://embdev.net/topic/359078

https://embdev.net/topic/359078795612nik567end452ku