8-Bit Binary Calculator

Author: Ameya Chindarkar

Abstract—This project report presents a Term Long Project designed for Brock University students, in the Computer Science Course – Introduction to Computer Architecture (COSC 2P12). The project provides a practical introduction to Computer Hardware Simulation using digital building blocks and tools learned in the course – "8-bit Binary Calculator", which performs operations like addition, subtraction, multiplication and division. The calculator is coded using Verilog HDL, using the Vivado software, provided by Xilinx. The code is then programmed onto an FPGA board. This project was a great introduction to how hardware description languages are used to design and implement circuits, commonly used in our day to day lives.

Index Terms—

1) A field-programmable gate array (FPGA) device board, DIGILENT BASYS 3
2) Vivado Design Suite is a software suite produced by Xilinx for synthesis and analysis of hardware description language (HDL) designs

1 Introduction

Firstly, I would like to thank professor Carvalho for strategizing the Term Long project for this course, a great thought to give enormous opportunities to students to gain hands-on computer architecture experience. I thoroughly enjoyed the project life cycle, research and development on the digital circuit development platform. This has helped me a lot, understanding advanced concepts/features and application of the Hardware Descriptive language. In this project, I will be using verilog, a hardware description language, in the Vivado Design Software Suite, to program a 8-bit binary calculator. The calculator will take in two 8-bit inputs as operands. These operands will be either added, subtracted, multiplied or divided and produce an output. To implement this code into hardware, The DIGILENT Basys 3 FPGA board was used. The basys 3 consists of over 30,000 logic cells, which can be programmed using HDLs.

1.1 Motivation

I find Programming appeals to my creativity and this subject, especially TLP, has given me a wonderful opportunity to gain Hands On experience. The Class room sessions helped me to understand the concepts and build the foundations of Digital Gates – building blocks and HDL. I was always curious about how hardware simulations work in the machines/circuits we use in our daily lives. My motivation was to build something which I have learned and create something which I can relate to on a day to day basis.

1) 4-bit ALU: I started with this project, which I successfully implemented using VHDL and Quartus Prime Lite. (Refer to the Related Work section for details).
2) Digital Clock: I then decided to go for a more hands on approach and used a FPGA board to implement a circuit. For this I chose to implement a digital clock, which I was unable to implement onto the board.Failures are rewarded, during this I was able to learn a lot about FPGAs and Hardware Description Language, which led me to my next project.
3) 8-bit Calculator: The calculator is a tool used daily and also an application of digital building blocks. I was successfully able to code and implement onto the board.

Ameya Chindarkar, 7023609, E-mail:ac20qh@brocku.ca

1.2 Objective

1) The first objective for this project was to learn the Hardware Descriptive Language, Verilog . And using Verilog to implement the 8-bit calculator.
2) The next objective was to code the input buttons and their functionality.
3) Constraints to assign the inputs and outputs to their respective pins on the board.
4) Implementing and testing the fully functional calculator.

1.3 Contributions

I worked on this project individually.

- Analysis: I started with the analysis of various project ideas, choosing the appropriate FPGA board and language to implement the circuit.
- Design and Coding: I designed the circuit and coded the modules, the constraints and implemented the circuit onto the FPGA board.
- Testing: I tested the calculator and fixed any bugs and errors that were present.
- Documentation: I produced a lab report using overleaf and the given template.
- Demo: I created the video, demonstrating the step-by step project implementation and 8-bit calculator functions.

2 Related Work

An FPGA (Field Programmable Gate Array is an integrated circuit, which provides the user the ability to configure and reconfigure the hardware, to meet requirements. An FPGA contains logic blocks, along with a set of programmable interconnects which are configurable. This allows the design of simple logic gates as well as complex functions. [1] The board I chose to use is the Digilent Basys 3, this board includes 33,280 logic cells, with 1,800 kbits of fast block RAM. 16 user switches, a 4-digit 7-segment display, 16 user LEDs, 5 push buttons and much more. [2]

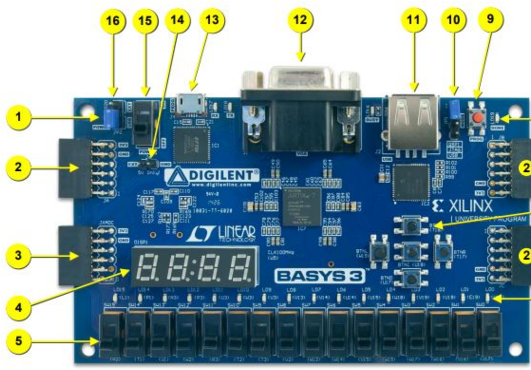


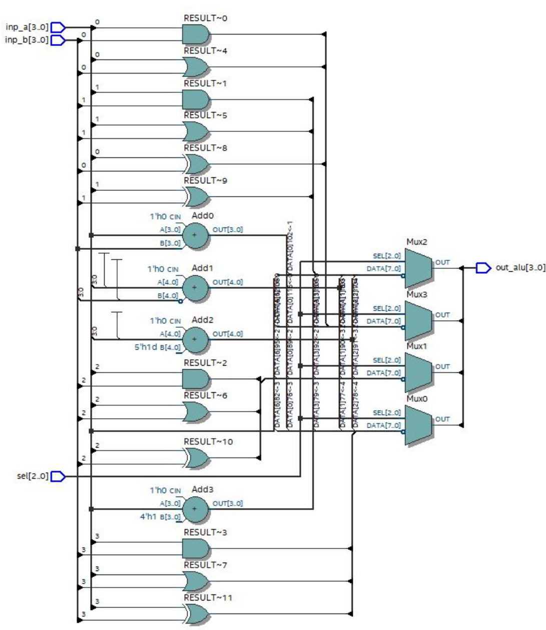
Figure 1. Basys 3 FPGA board with callouts.

Callout	Component Description	Callout	Component Description
1	Power good LED	9	FPGA configuration reset button
2	Pmod port(s)	10	Programming mode jumper
3	Analog signal Pmod port (XADC)	11	USB host connector
4	Four digit 7-segment display	12	VGA connector
5	Slide switches (16)	13	Shared UART/ JTAG USB port
6	LEDs (16)	14	External power connector
7	Pushbuttons (5)	15	Power Switch
8	FPGA programming done LED	16	Power Select Jumper

Table 1. Basys 3 Callouts and component descriptions.

Projects Attempted:

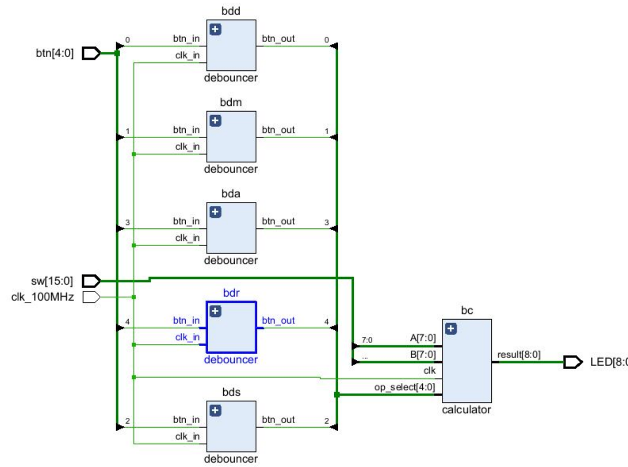
1) 4-bit ALU: Using VHDL and Quartus Prime Lite, I implemented the code for a 4-bit ALU. I learned about using Logism to build the ALU (Arithmetic Logic Unit). The AlU consister of 4-bit adder/subtractors, 4-bit OR, 4-Bit AND, and 4-bit XOR.



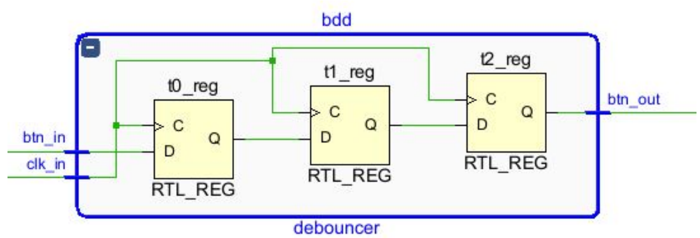
2) Digital Clock: I attempted to design and code a digital clock, which would display the time in HH:MM onto the Seven Segment display on the FPGA board. The time can be adjusted using the 5 buttons. [4]

3 System

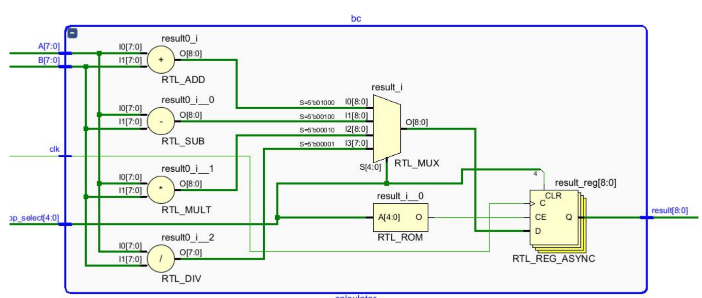
- The calculator takes in two 8-bit binary inputs as operands.
- The output of the calculator, depending on which button is pressed, is either the sum, difference, product, or quotient of the operands.
- The four functions were programmed using Verilog HDL.
- Submodules were used to code the functionality and the button assignments.



The figure above is the schematic of the circuit. Switch 15 down to 0, are the 16 user inputs, which determine the 2 operand numbers. Button 4 down to 0 are the 5 user inputs which will determine which operation is performed and displayed. The signals for the buttons are then connected to the debouncer sub-circuits, which take the button-input and output the respective operation assigned to the button. As shown below.



Next, the calculator module then takes the inputs from the switches: the operands and the button-output from the debouncer module and feeds it to the multiplexer and based on the operation, the result is then displayed in the form of the 9 LEDs. As shown in the figure below.



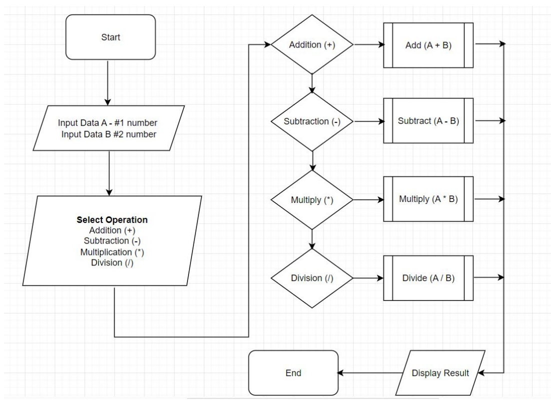
4 Methodology and Methods

- Hardware and Software used:

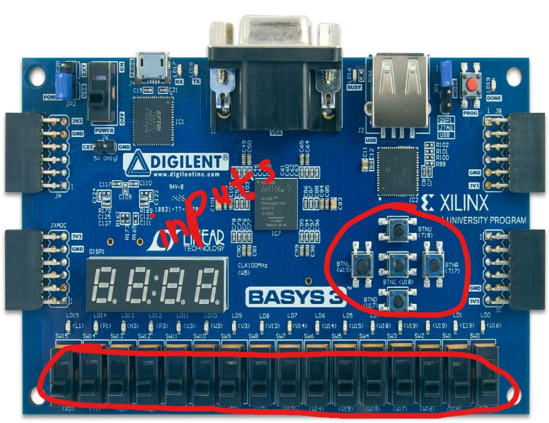
  1) A field-programmable gate array (FPGA) device board, DIGILENT BASYS 3, this board includes 33,280 logic cells, with 1,800 kbits of fast block RAM. 16 user switches, a 4-digit 7-segment display, 16 user LEDs, 5 push buttons

2) Vivado Design Suite is a software suite produced by Xilinx for synthesis and analysis of hardware description language (HDL) designs.
3) Verilog is a Hardware Description Language; a textual format for describing electronic circuits and systems. Applied to electronic design, Verilog is intended to be used for verification through simulation, for timing analysis, for test analysis (testability analysis and fault grading) and for logic synthesis. [3]

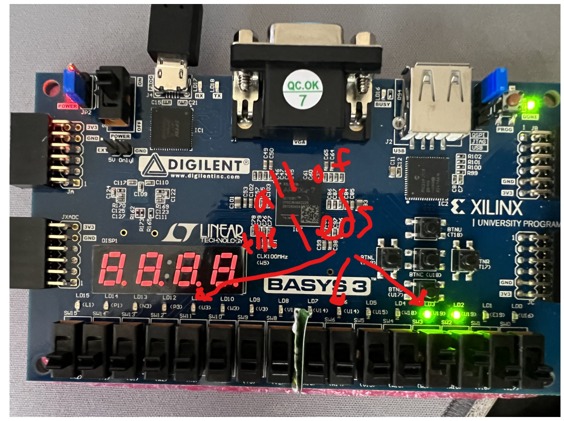
- Flowchart:



- In order to make a calculator, I needed to have a way for the user to be able to input numbers onto the calculator. To do this, I decided to use the 16 switches present. With 8 switches for each 8-bit number, their values would be stored in ‘A’ and ‘B’ This was programmed in the calculator module.



- Next for the functionality of the calculator, I chose addition, subtraction, multiplication and division. This is also meant to function as a user-input. In order for the user to pick the operations, I decided to use the push buttons. To assign the push buttons, A debouncer module was needed which would take the button being pushed as an input and output the respective operation it is assigned.
- The debouncer module along with the calculator module is called in the top-module, which outputs the final solution after the operations are performed on A and B. To display the output, the LEDs above the switches are used. As shown in the figure below. The output is also shown in 2’s complement.



- After finishing the code for the functionality, it was ready to be uploaded onto the FPGA, but before that, the constraints had to be provided, which would assign each input and outputs to the pins assigned to necessary components on the board (switches or leds or buttons).

5 Result

Youtube Demo: https://youtu.be/DmYlFk22v4A

Operation	Data A	Data B	Solution
Addition(+)	8	8	16
	12	2	14
	28	64	92
Subtraction(-)	8	8	(Blank)0
	16	3	13
	35	8	27
Multiplication(*)	8	8	64
	6	17	102
	15	15	225
Division(/)	8	8	1
	9	2	4
	32	16	2

5.1 Experiment 1: Addition

For this experiment, I tried adding 3 different values.:

- First I input the number 8 for both A and B, using the switches, and after pushing the button for addition, as the result the 5th led lit up, meaning the answer is 16.
- Next, I input 12 as A and 2 as B, this gave me 14 as the result.
- Lastly, I input 28 as A and 64 as B, this resulted in 92.

The addition aspect of the calculator works well, with all experiments resulting in the correct answer.

5.2 Experiment 2: Subtraction

For this experiment, I tried subtracting 3 different values using the calculator.

- First I input the number 8 for both A and B, using the switches, and after pushing the button for subtraction, got no LEDs lit up as a result, meaning 0.
- Next, I input 16 as A and 3 as B, this gave me 13 as the result.
- Lastly, I input 35 as A and 8 as B, this resulted in 27.

Subtraction aspect of the calculator works well without any issues.

5.3 Experiment 3: Multiplication

For this experiment, I tried multiplying 3 different values using the calculator.

- First I input the number 8 for both A and B, using the switches, and after pushing the button for multiplication, this resulted in the 7th led lighting up, meaning the answer is 64.
- Next, I input 6 as A and 17 as B, this gave me 102 as the result.
- Lastly, I input 15 as A and 15 as B, this resulted in 225.

Multiplication aspect of the calculator works well, however, results larger than 9-bits cannot be displayed.

5.4 Experiment 4: Division

For this experiment, I tried dividing 3 different values using the calculator.

- First I input the number 8 for both A and B, using the switches, and after pushing the button for multiplication, this resulted in the 1st led lighting up, meaning the answer is 1.
- Next, I input 9 as A and 2 as B, this gave me 4 as the result.
- Lastly, I input 32 as A and 16 as B, this resulted in 2.

Division aspect of the calculator works accurately only with whole numbers, any fractions are rounded up or down to the nearest whole number.

6 Conclusion

This project combines: digital design using building blocks and implementation using Verilog, Thorough testing was performed and met the objective of the calculator functions: addition, subtraction, multiplication and division. To summarize, although our project has met all the requirements and guidelines. I would recommend an improvement to make this calculator more user friendly by keyboard interface and output the results onto the seven segment display. To code the calculator, I used Verilog and implemented the design onto the basys 3, an FPGA board. All the operations worked without issues except for the division operation which only produces results in whole numbers. In the future I would like for my calculator to include decimals for the results. I would make sure the output can be atleast 16 bit so that I can multiply larger numbers. And I would also like to convert the numbers from binary and present them onto the seven segment display. This project was a great introduction to computer architecture and was a good way to acquire hands-on experience.

References

[1] AMD. (n.d.). What is an FPGA? field programmable gate array. Xilinx. Retrieved March 18, 2023, from https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html: :text=Field

[2] Basys 3™ FPGA Board Reference Manual Overview - Digilent. Basys 3™ FPGA Board Reference Manual. (2016, April 8). Retrieved March 18, 2023, from https://digilent.com/reference/-media/reference/programmable-logic/basys-3/basys3-rm.pdf

[3] DOULOS. (n.d.). What is Verilog? Doulos. Retrieved March 18, 2023, from https://www.doulos.com/knowhow/verilog/what-is-verilog/: :text=Verilog

[4] Ltd, A. A. (2022, July 27). Digital Clock Design with FPGA board. UKDiss.com. Retrieved March 18, 2023, from https://ukdiss.com/examples/digital-clock-implementation-with-fpga.php

[5] Mughal, P. (2019, September 13). 20 FPGA project digital clock — FPGA BASYS3 Board — verilog. YouTube. Retrieved March 18, 2023, from https://www.youtube.com/watch?v=8$_Um6Cx6bD8amp;ab_channel = ElectronicswithProf.Mughal$