8 bit calculator

Final Report

January 29, 2020

André Oliveira – 83873

Markiyan Pyekh - 96959

Contents

List of Tables

1 Introduction

The main goal of our project is to make a functional calculator that operates with 8 bit signed integers, with a range of [-128:127]. The calculator is capable of doing sums, substractions, divisions and multiplications. In order to select the operation and the operands, we have to use the 8 switches of the board and three buttos to select the operation and the operands and a reset button.The 7 segment display is used to display the the operands,operation and results.

2 Software Diagram

The software diagram shows how the program works. First, the user uses the 8 switches to get the first operand and after that presses the button BTN3. Then the user uses 8 switches, again, to get the second operand and after that presses the button BTN2. Thirdly, the user uses again the switches to choose the operation (001 = sum), (010 = multiplication) and (100 = division). After this, all the operands are sent to the ALU and the result is displayed in the 7 segment display.

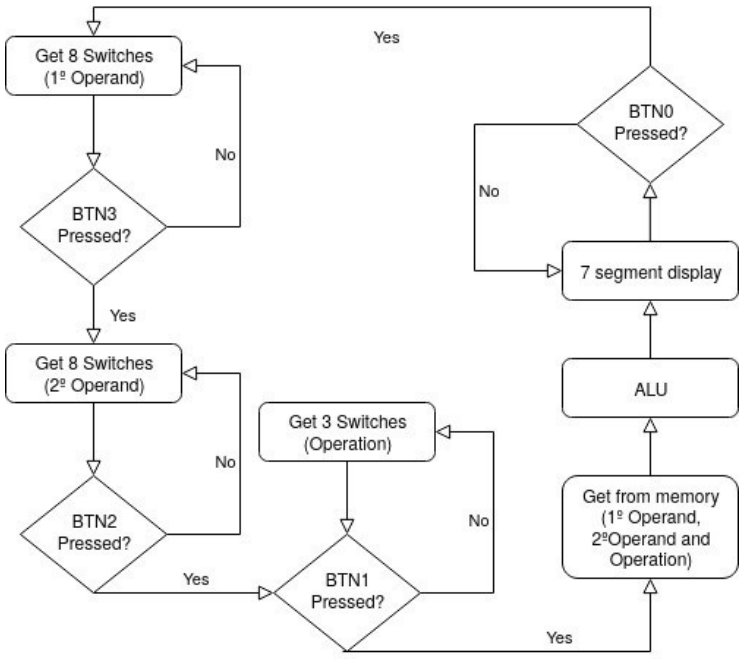

Figure 1:Block Diagram

3 Peripherals

All the peripherals that are used in the xtop module of the project.

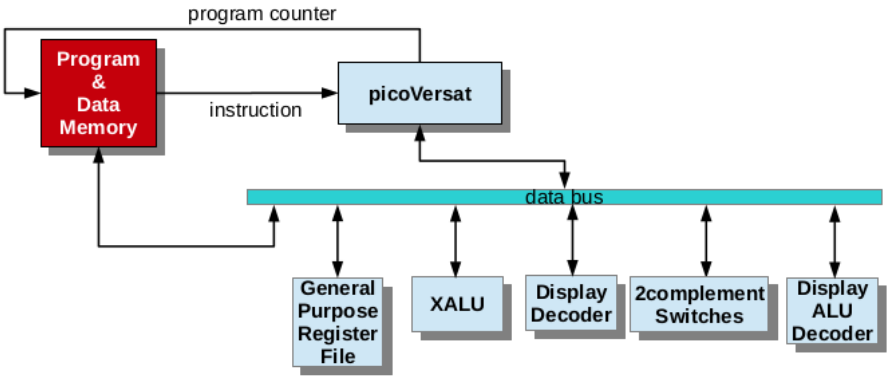

Figure 2:PicoVersat SoC with five peripherals

3.1 PicoVersat

In this program the picoversat is used to read the input from the buttons, switches and choose which verilog block is being used at the moment.

3.2 GeneralPurpose RegisterFile

This peripheral contains a 16x32bit register file that can be used by user programs.

3.3 2complement switches

If the result inputted by the user Is negative, this block makes his 2complement of the negative number inputted by the user.

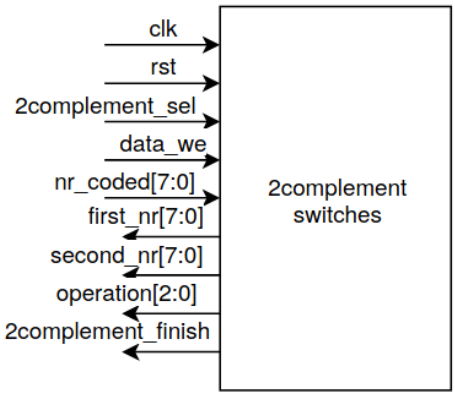

Figure 3: 2complement block

3.4 xAlu

This peripheral is used to decide which operation the user choosed and output the corresponded result.

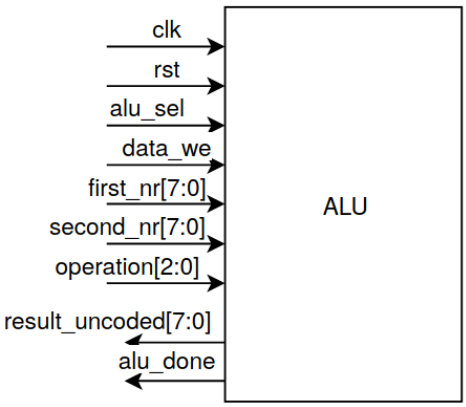

Figure 4: xAlu block

3.5 Display ALU decoder

Outputs the signal of the ALU result and converts negative numbers to positive numbers in order to display in 7 segment display.

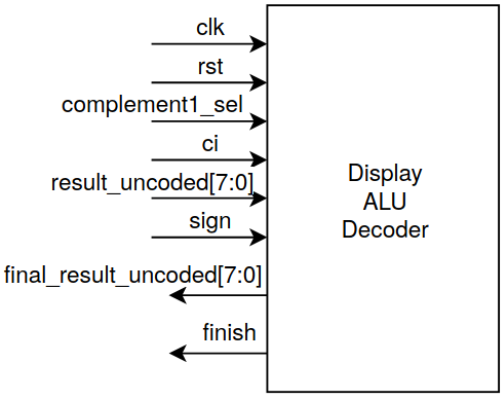

Figure 5: Display Alu decoder block

3.6 7-Segment Display Driver

This peripheral is used to display the result of the the arithmetical operation.

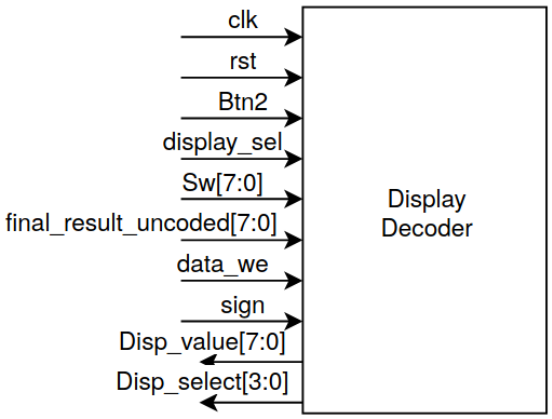

Figure 4: Display decoder block

4 MemoryMap

The memory map of the system, as seen by picoVersat programs, is given in Table1.

Mnemonic	Address	Read/Write	Read Latency	Description
REGF BASE_	512	Read+Write	0	Register file peripheral
CPRT BASE	600	Write only	NA	Debug printer periheral
SW_BASE	704	Read+Write	0	First Operand
BTN3_BASE	710	Read	0	Button BTN3 pressed
BTN2_BASE	714	Read	0	Button BTN2 pressed
SW2_BASE	718	Read+Write	0	Second operand
COMPLEMENT2_BASE	722	Read+Write	0	Sel convert negative nrs
ALU_BASE	726	Read+Write	0	Sel ALU
CONVERT_BASE	730	Read+Write	0	Convert final result
DISPLAY_BASE	734	Read+Write	0	Sel display block
BTN1_BASE	738	Read	0	Button BTN1 pressed
SW3_BASE	742	Read+Write	0	Operation

Table 1:Memory map base addresses

5 ImplementationResults

In this project the system occupied 64% of the LUTs, 25% of the block ram and it was used a 25 MHz clock. The system can represent numbers from [-128, 127] and can sum, subtract, multiplicate and divide. Shows the numbers being chosen and the operation in the 7 segment display.

6 Conclusions

Initially, we had a lot of problems to start understanding how picoversat works, but after this initial struggle, our project started to gain shape. We designed all the Verilog modules (adder and multiplication included) which were pretty good. Overall we made everything that we proposed to do in our initial proposal... As the first fully working Verilog project that we made, we think that we obtain very good results.