ClockMaster

User Guide

TÉCNICO LISBOA

November 6, 2019

Group 4

Pedro Santos, nº93892

Sara Soares, nº87737

Professor José Teixeira de Sousa

Contents

1 Introduction 7

2 Block Diagram 8

3 Interface Signals 9

4 Peripherals 10

4.1 Peripherals Used 10

4.2 Timer Counter 10

4.3 7-Segments Display 10

4.4 Push Buttons 10

5 Instructions 10

5.1 Clock 10

5.2 Counter 11

5.3 Timer 11

6 Implementation Results 11

7 Conclusions 13

List of Tables

1 Peripheral interface signals. 9

2 Push Buttons. 10

3 Push Buttons. 10

List of Figures

1 Basys 2 board 7
2 Block diagram 8
3 Display Diagram 11
4 Timer Diagram 12
5 Timer Demonstration 12

1 Introduction

The Basys2 Figure 1, board is a circuit design and implementation platform that anyone can use to gain experience building real digital circuits. Built around a Xilinx Spartan-3E Field Programmable Gate Array and a Atmel AT90USB2 USB controller, the Basys2 board provides complete, ready-to-use hardware suitable for hosting circuits ranging from basic logic devices to complex controllers.

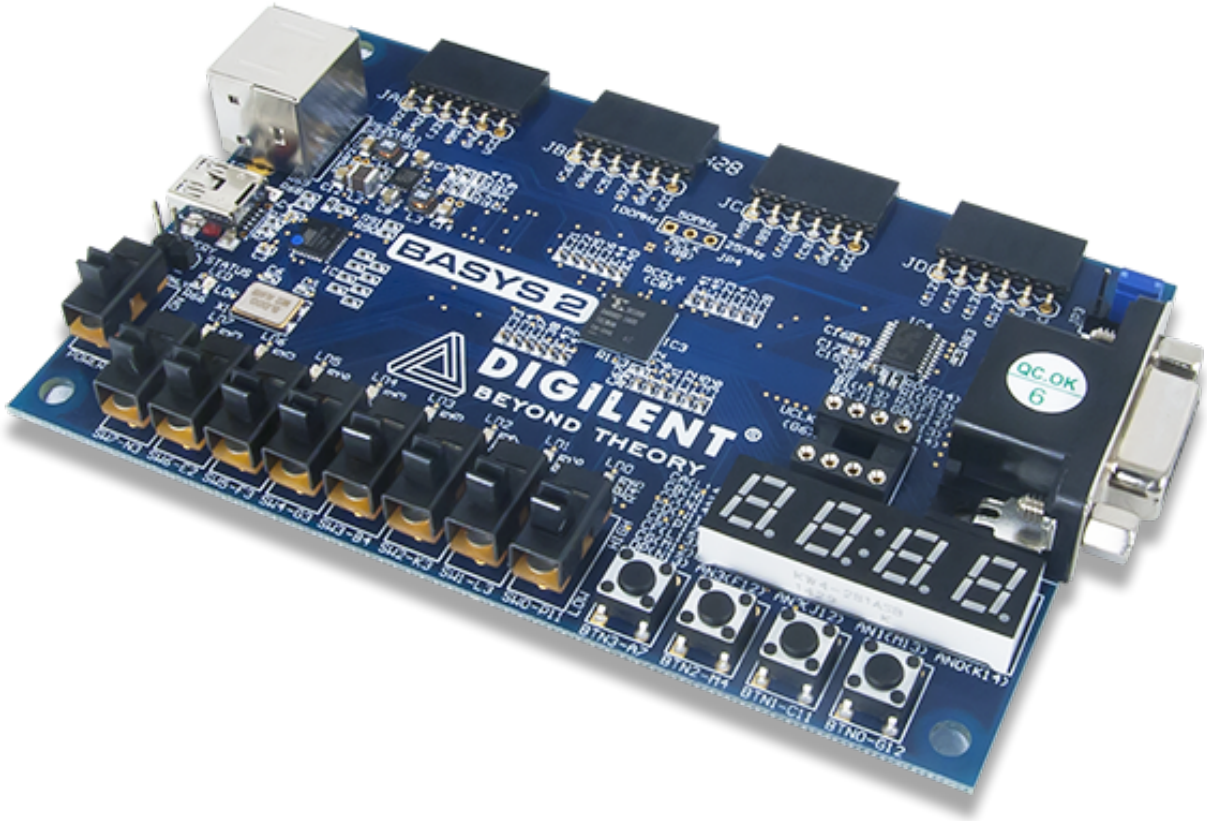

Figure 1: Basys 2 board

Our project uses the Basys2 to make a clock with some features. We propose to use the 7 segment display as an output and the push buttons as input, to set the time and navigate through the menus.

2 Block Diagram

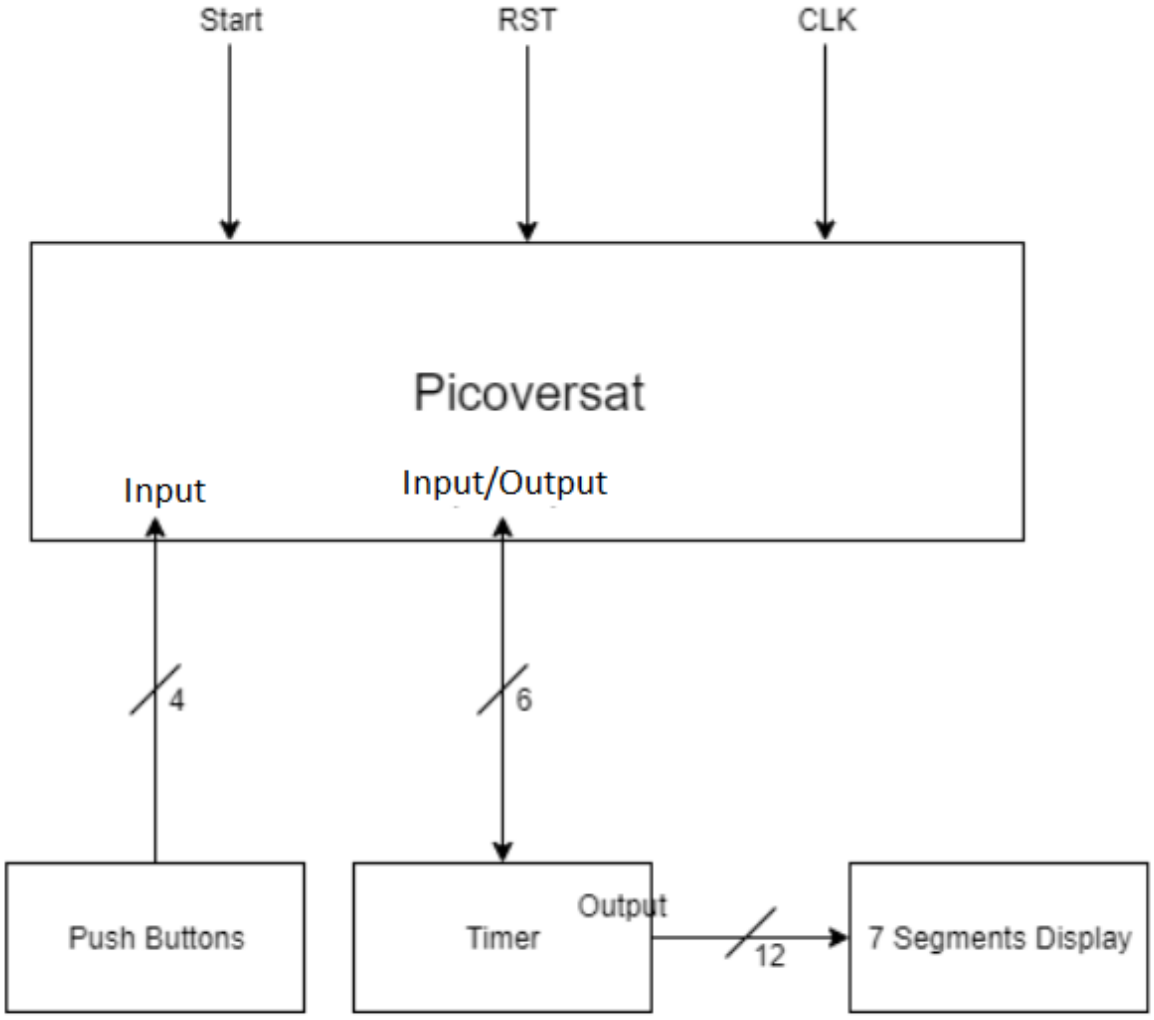

Figure 2: Block diagram

3 Interface Signals

Table 1: Peripheral interface signals.

Name	Direction	Description
clk	IN	Clock signal
rst	IN	Reset signal
mode	IN	Changes the mode of operation (clock, counter or timer)
seth	IN	Sets the hours
setm	IN	Sets the minutes
display	OUT	Shows the clock, timer or counter (depends on mode)

4 Peripherals

The Basys2 has manny peripherals options such as: -7-segments display -on/off switch -push buttons -VGA port -PS/2 port -6-pin headers

4.1 Peripherals Used

The peripherals that we will de using are the 7-segments display and the push buttons.

4.2 Timer Counter

In the Timer we use this addresses.

Table 2: Push Buttons.

	Timer	min0	min1	hour0	hour1
BASE	1600	1600	1601	1602	1603
Addr_W	2	-	-	-	-

4.3 7-Segments Display

The 7-segment displays will not need an memory address since it will receive the values to display directly from the timer module. For this to work we rely on the following diagram in the figure 3.

4.4 Push Buttons

Table 3: Push Buttons.

Name	Address	Bits	Description
BUTTON_BASE	1500	3:0	Reset, mode, set hours and set minutes buttons

5 Instructions

5.1 Clock

The ClockMaster will have the function to show the time in the 7 segment dislay. With the help of the push buttons we will be able to set the time correctly.

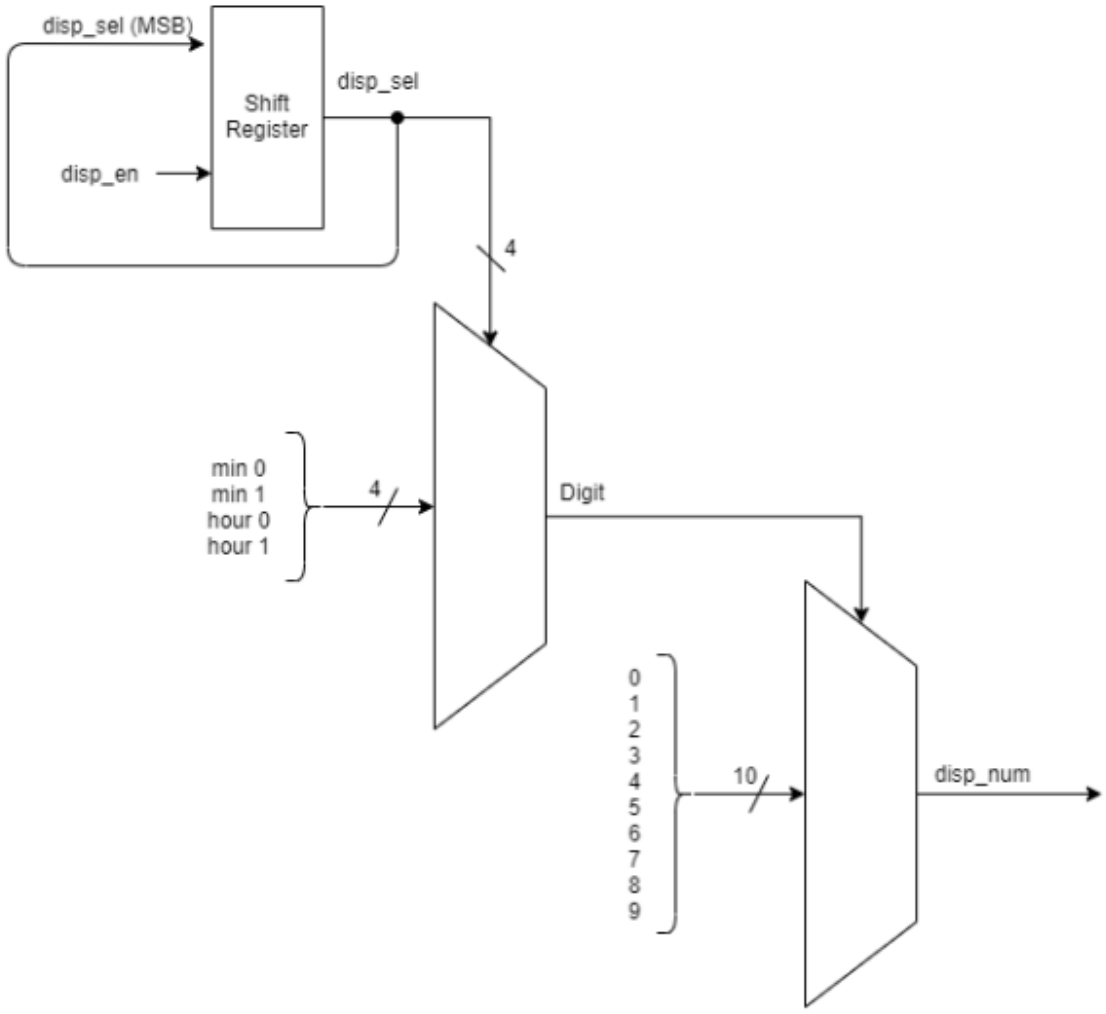

Figure 3: Display Diagram

5.2 Counter

The Counter is a feature that we can set a desired value and counts until reach that value.

5.3 Timer

This feature use a previously entered value and stop counting when it reaches 0.

6 Implementation Results

For the implementation, it was agreed that we would create 2 separate verilog modules for the project: a display module and a timer module.

The display module is a module designed to receive a certain value and display it in the 4 7-segment displays the used FPGA offers. It has a refresh rate in order to be able to display it, as well as in order to pass by all 4 displays in order to give a simultaneous output in all 4 displays, when in reality, only one is turned on at each moment in time. This module also transforms each number in a output that draws the correct number in the 7-segment display peripheral.

The timer module is designed to count the time up to 24 hours. It has a counter variable to transform the time an instruction takes in a second, and from then on, it simply counts the seconds to make a minute and so on until it reaches the desired 24 hours and then it returns to zero. In the diagram of the Figure 4 you can see what was explained before.

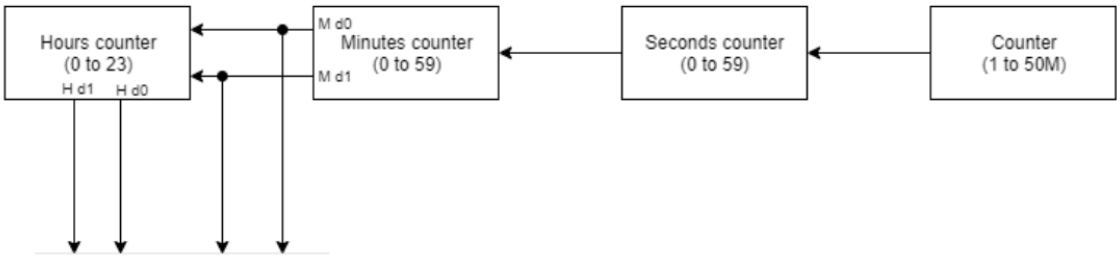

Figure 4: Timer Diagram

For this project, we concluded that it was not necessary to create a separate module for the push button inputs.

In order to use all these peripherals and modules, they need to be added to the xtop and xaddr_decoder verilog modules present in picoversat.

The programming part of the project was made using the C to assembly compiler provided by the teacher. However, the resulting code exceeded the memory available in picoversat. With this, adjustments to the memory were needed, and it was decided to doouble the available memory space in picoversat. This was done by changing the ADDR_W number, on the xdefs.vh file, as well as the addresses dependant on this value.

At the end of the project deadline, the clock was able to sucessfully count up to 24 hours and display it in the FPGA, with a single continous dot in the middle to separate the hours from the minutes. However the push buttons were not able to be implemented. To finish, we put a picture of our timer running before completing another 24 hour cycle, Figure 5.

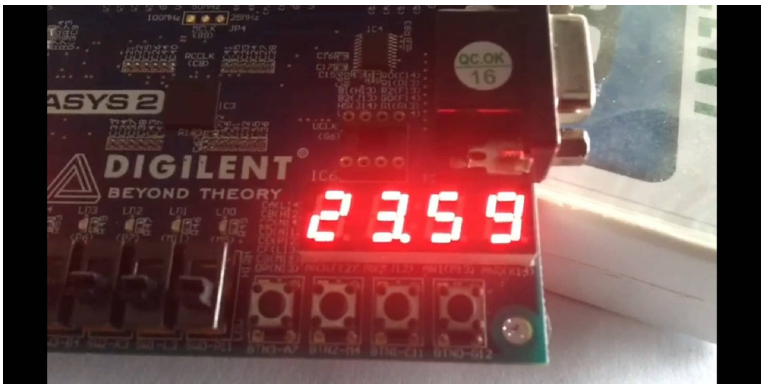

Figure 5: Timer Demonstration

7 Conclusions

By the end of this project, there was a better understanding of all the components used in the making of this clock, as well as an understanding on to how better use FPGAs in a project, using the provided datasheet to analyze what could or could not be done and how to use the provided peripherals.

There were some dificulties, some of them due to the lack of knowledge of the verilog programming language, others due to lack of debugging, simulating and testing. In the end, the proposed project was not completed in its entirety, due to said difficulties and pinpointing the correct cause of the observed problems we faced.