Table of Contents

[01. Introduction 3](#_Toc67911178)

[02. Assembly Program and its Machine Code 3](#_Toc67911179)

[Assembly program 3](#_Toc67911180)

[Machine Code 3](#_Toc67911181)

[03. 4 -bit Add/Subtract Unit 3](#_Toc67911182)

[Description 3](#_Toc67911183)

[VHDL Code 3](#_Toc67911184)

[Elaborated Design View (RTL Schematic view) 3](#_Toc67911185)

[Timing Diagram 3](#_Toc67911186)

[04. 3 – bit Adder 3](#_Toc67911187)

[Description 3](#_Toc67911188)

[VHDL Code 3](#_Toc67911189)

[Elaborated Design View (RTL Schematic view) 3](#_Toc67911190)

[Timing Diagram 3](#_Toc67911191)

[05. 3 – bit Program Counter (PC) 3](#_Toc67911192)

[Description 3](#_Toc67911193)

[VHDL Code 3](#_Toc67911194)

[Elaborated Design View (RTL Schematic view) 3](#_Toc67911195)

[Timing Diagram 3](#_Toc67911196)

[06. k – way b – bit Multiplexer 3](#_Toc67911197)

[Description 3](#_Toc67911198)

[VHDL Code 3](#_Toc67911199)

[Elaborated Design View (RTL Schematic view) 3](#_Toc67911200)

[Timing Diagram 3](#_Toc67911201)

[07. Register Bank 3](#_Toc67911202)

[Description 3](#_Toc67911203)

[VHDL Code 3](#_Toc67911204)

[Elaborated Design View (RTL Schematic view) 3](#_Toc67911205)

[Timing Diagram 3](#_Toc67911206)

[08. Program Rom 3](#_Toc67911207)

[Description 3](#_Toc67911208)

[VHDL Code 3](#_Toc67911209)

[Elaborated Design View (RTL Schematic view) 3](#_Toc67911210)

[Timing Diagram 3](#_Toc67911211)

[09. Instruction Decoder 3](#_Toc67911212)

[Description 3](#_Toc67911213)

[VHDL Code 3](#_Toc67911214)

[Elaborated Design View (RTL Schematic view) 3](#_Toc67911215)

[Timing Diagram 3](#_Toc67911216)

[10. Nano Processor 3](#_Toc67911217)

[Description 3](#_Toc67911218)

[VHDL Code 3](#_Toc67911219)

[Elaborated Design View (RTL Schematic view) 3](#_Toc67911220)

[Timing Diagram 3](#_Toc67911221)

[11. Conclusion 3](#_Toc67911222)

[12. Contribution of Members 3](#_Toc67911223)

# 01. Introduction

In this lab we designed a 4 – bit processor capable of executing 4 instructions (MOVI, ADD, NEG and JZR). In building the nano processor we built the following key components.

* 4 – bit add/sub unit.
* 3 – bit Adder
* 3 – bit Program Counter (PC).
* k – way b – bit Multiplexer.
* Register Bank.
* Program ROM.
* Instruction Decoder.

We developed the VHDL codes for the above key components, verified their functionality via simulation, obtained timing diagrams for each component and developed XDC VHDL Code.

In the XDC VHDL code Value of the register 7 was given as an output to 3 LEDs and to a 7-segment display. Also there are LED indicators LD14 and LD15 respectively for zero and overflow flags. When reset button is pushed all register values and program counter reset to zero.

# 02. Assembly Program and its Machine Code

## Assembly program

The assembly program is to calculate sum of all integers between 1 and 3 and store the final answer in the Register 7 (R7) using 4 instructions. In the code we implement ted a loop from 3 to 0. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Machine Code

# 03. 4 -bit Add/Subtract Unit

## Description

A and B are the inputs and we can toggle add/subtract status by Ctrl variable. If Ctrl variable is ‘1’ it gives the two’s complement value of the B input and addition between A value and 2’s compliment of B gives us A-B. When Ctrl is ‘0’ it gives A+B.

## VHDL Code

## Elaborated Design View (RTL Schematic view)

## Timing Diagram

# 04. 3 – bit Adder

## Description

This unit is used to increment the Program Counter. It add 12 to the 3 bit memory address coming out of the Program Counter.

## VHDL Code

## Elaborated Design View (RTL Schematic view)

## Timing Diagram

# 05. 3 – bit Program Counter (PC)

## Description

Since we should have the ability to reset the PC to 0 when required it is built using D Flip-Flop. Stores the memory location of the next instruction to be executed.

## VHDL Code

## Elaborated Design View (RTL Schematic view)

## Timing Diagram

# 06. Tri – State Buffer

## Description

## VHDL Code

## Elaborated Design View (RTL Schematic view)

## Timing Diagram

# 07. k – way b – bit Multiplexer

## Description

A k – way b – bit multiplexer takes k – inputs, each with b – bits and gives a single output of b – bits depending on log2 k selection lines. We designed 8 - way 4 – bit, 2 – way 4 – bit, 2 – way 3 – bit multiplexers using tri – state buffers with common bus implementation. The following code is for 8 - way 3 – bit multiplexer.

## VHDL Code

## Elaborated Design View (RTL Schematic view)

## Timing Diagram

# 08. Register Bank

## Description

Register bank contains 8 registers (4 - bit) named R0 to R7. The R0 register is detached from the common data bus and hard coded to 0.

Reset \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## VHDL Code

## Elaborated Design View (RTL Schematic view)

## Timing Diagram

# 09. Program Rom

## Description

This stores the Assembly instructions for the program. Program ROM is made of 8, 12 – bit slots. ROM based Look UP Table to map memory locations and instructions. The following VHDL code stures our assembly program in the program memory.

## VHDL Code

## Elaborated Design View (RTL Schematic view)

## Timing Diagram

# 10. Instruction Decoder

## Description

Instruction Decoder activates necessary components and data busses in - order to execute 4 kinds of instructions, ADD, MOVI, NEG and JZR. The 2 – to – 4 decoder inside the instruction decoder is used to switch in between instructions.

## VHDL Code

## Elaborated Design View (RTL Schematic view)

## Timing Diagram

# 11. Nano Processor

## Description

Nano Processor is the final result of all the above components. Has the ability to execute ADD, MOVI, NEG and JZR instructions. Since this is a very simple microprocessor it is called a nano processor.

## VHDL Code

## Elaborated Design View (RTL Schematic view)

## Timing Diagram

# 12. Conclusion

* As tri – state buffers greatly reduce unwanted connections instead of building our k – way b – bit MUX using basic logic gates we used tri – state buffers.
* We can add more components and also improve already used components in order to increase the capability of our nano processor (we can improve it to execute more complex instructions).
* Instruction decoder is the key component of the whole nano processor as it is the components responsible for activating and controlling necessary components and deactivate them once used.
* \*\*\*\*\*\*Clock signal for Register Bank and program counter should differ by phase angle of 180 degrees as fetching and executing are two stages and there is no instruction register in this Nano processor to hold the instruction. So, there should be a delay between fetching the instruction and executing the instruction.
* Considering the capabilities of current 4 – bit Add/Subtract unit, not only we implemented a negation instruction but also we can implement a instruction to subtract two numbers.

# 13. Contribution of Members