2021

ECTE 432: Computer Architecture

pROJECT 2: PIPELINE MIPS32 PROCESSOR

LAKSHAY RAMCHANDANI: 6273130

Github LINK: https://github.com/lnr681/ecte432\_project

Contents

[Loading of Instructions (First sixteen instructions): 2](#_Toc89990283)

[First branch on the waveform: 2](#_Toc89990284)

[Last high of memwrite, write data and address from data memory: 3](#_Toc89990285)

[Implementation of LUI: 3](#_Toc89990286)

[Total number of nops, reset, memwrites, memreads and clockcycles: 4](#_Toc89990287)

[Appendix: 5](#_Toc89990288)

[Simulink: 5](#_Toc89990289)

# Loading of Instructions (First sixteen instructions):

A screenshot of a computer

Description automatically generated with medium confidence

Figure :Waveform loaded by instructions 1 to 9

A screenshot of a computer

Description automatically generated with medium confidence

Figure :Instructions loaded by instructions 9 to 16

# First branch on the waveform:

A screenshot of a computer

Description automatically generated with medium confidence

Figure :First branch on the waveform (around clock cycle 36-40)

# Last high of memwrite, write data and address from data memory:

A screenshot of a computer

Description automatically generated with medium confidence

Figure :Waveform values at the last high of memwrite

As discussed above the last line (instruction) of the set of instructions is a conditional branch which is only true when value of register $t5 is set to 1. The value of register $t1 is initially set as 6 and it is reduced by 1 every time is traverses the loop. The value of register $t5 is set to 1 when value of register $t1 becomes less than 1 (this is checked at the instruction before this). Hence the loop instructions are traversed 6 times before the code stops. With every loop traversal the value in register $t2 (initially set at 5) added to itself. After 6 traversals the value that register $t2 has eventually becomes 5 added 5 times to itself (5\*6=30) and this value is represented as 1e in hexadecimal. This can be observed in Figure 4 above.

# Implementation of LUI:

To Implement the Lui the Datapath wasn’t required to be changed only a few additions were required to made to the VHDL’s of a few Black boxes. The code of Black boxes Control unit, Alu control and ALU had to be modified to accommodate Lui as a workable operation. The control unit had an extra case where if the opcode was equal to "001111" the control signals of Lui were being sent. Lui was given a unique code for alu\_op which was "11", this code was only meant for Lui (since lui is not an R type instruction and it had to do an operation different from addition, subtraction or any other basic function used by others). In the Alu control an extra statement was added to provide a operation code for the case of Lui if the alu\_op code matched. The operation code designated to lui was "011". The Alu had to be changed to add the functionality of lui where the lui operations functionality is actually described in case it is used.

# Total number of nops, reset, memwrites, memreads and clockcycles:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 5:Overall waveform output

The counts for the nops, reset, memwrites, memreads, branch and clockcycles are given below

|  |  |
| --- | --- |
| Signal | Count |
| Branch select | 6 |
| Memread | 6 |
| Memwrite | 7 |
| Nop | 6 |
| PC\_stall | 6 |
| IF\_ID\_flush | 6 |
| Clock(Last memwrite) | 154 |

# Appendix:

## Simulink:

Chart, bar chart

Description automatically generated

Figure :Simulink model