UNIVERSITY OF LIVERPOOL

Department of Electrical Engineering and Electronics

# ELEC373

# Digital Systems Design Assignment 3 MIPS Processor

| | |
|---|---|
| **Module** | ELEC373 |
| **Coursework name** | Assignment 3 |
| **Component weight** | 20% |
| **Semester** | 2 |
| **HE Level** | 6 |
| **Lab location** | PC labs 301, 304 as timetabled, at other times for private study |
| **Work** | Individually |
| **Timetabled time** | 12 hours (3 hours per week – Wednesday 2pm – 5pm) |
| **Suggested private study** | 10 hours including report writing |
| **Assessment method** | Individual, formal word-processed reports (Block diagrams and ASMs can be hand drawn and scanned into the report) |
| **Submission format** | Online via VITAL |
| **Submission deadline** | 23:59 on Wednesday 21st April 2021 |
| **Late submission** | Standard university penalty applies |
| **Resit opportunity** | August resit period (if total module failed) |
| **Marking policy** | Marked and moderated independently |
| **Anonymous marking** | No |
| **Feedback** | Via printed annotated copy showing corrections |
| **Learning outcomes** | LO1: Ability to design digital systems using the ASM design method<br>LO2: Ability to implement digital systems using the Verilog Hardware Description Language<br>LO3: Understanding the internal operation of a MIPS processor. |

**Marking Criteria**

| **Section** | **Marks available** | **Indicative characteristics** | |
|---|---|---|---|
| | | **Adequate / pass** (40%) | **Very good / Excellent** |
| Presentation and structure | 20% | • Contains cover page information, table of contents, sections with appropriate headings.<br>• Comprehensible language; punctuation, grammar and spelling accurate.<br>• Equations legible, numbered and presented correctly.<br>• Appropriately formatted reference list. | • Appropriate use of technical, mathematic and academic terminology and conventions.<br>• Word processed with consistent formatting.<br>• Pages numbered, figures and tables captioned.<br>• All sections clearly signposted.<br>• Correct cross-referencing (of figures, tables, equations) and citations. |
| Introduction, Method and Design | 40% | • Problem background introduced clearly.<br>• Evidence of a Top Down Design approach<br>• Conceptual Design Choices introduced.<br>• Design of each module follows a logical sequence.<br>• ASMs correspond to designs for each block.<br>• Software is clearly commented | • Appropriate range of references used.<br>• Design decisions justified with alternatives given.<br>• Calculations shown in full, justifying and explaining any decisions.<br>• Correct ASM Syntax used.<br>• Well-structured Verilog Code<br>• |
| Results | 30% | • Simulation results present for each block and well annotated.<br>• Results of full system in both simulation and experimentally presented.<br>• Results for each task accompanied by a commentary.<br>• Screen shots of results presented. | • Tests indicate that there are no problems caused by asynchronous inputs.<br>• Clear explanation of how the instructions operate correctly |
| Discussion | 10% | • Discussion on what worked and what didn’t.<br>• Critical assessment on the design – strength and weaknesses | • Discussion on how the system was fully tested. |

# ELEC373 Verilog Assignment 3 (2020-2021)
# Synthesising the MIPS Processor

## Assignment Outline

Assignment 3 is split into 2 parts, Part A and Part B. The objective of Part A is to get you familiar with the synthesised MIPS single cycle processor and to write some simple programs to control the processor. Part B requires you to extend the processor so that it will implement additional instructions.

## MIPS_System

The Verilog Code for the MIPS single cycle implementation are available on VITAL. Download the ZIP file called MIPS_System and extract it into a suitable location. The synthesised MIPS processor starts executing a program from location 0x0000000. The program is loaded into the FPGA via a Memory Initialisation File, when you program the FPGA. In this design it is called “insts_data.mif”. If you examine this file using the Quartus software you’ll find that the data it contains is:

0x3C020000, 0x24420055, 0x3C03FFFF, 0x24632008, 0xAC620000, 0x08000005

If you disassemble this you’ll find that the first instruction corresponds to: lui $2, 0x0000

Using the MIPS Instruction Coding available from Vital, disassemble the other instructions to understand what the program does.

## Memory Map

If you study the “Addr_Decoder.v” file you’ll find that the GPIO (General Purpose Input/Output) module is mapped from location 0xFFFF_2000. If you examine the “GPIO.v” file you’ll find the individual locations for the LEDs and switches on the DE2 board.

## Program Execution

If we had access to the DE2 Boards you would synthesise and download the design, and should see that it switches on some of the red LEDs, However as this year we don’t have direct access to the Boards you will simulate the design in ModelSim.You should use ModelSim so that you can see the appropriate signals changing in the synthesised MIPS core when the MIPS CPU is running.

## Assignment 3 Part A – 40%

1. Modify the MIPS assembly language program so that the program displays the lowest 8 digits of your ID on the DE2 board 7 segment display.
2. In your report you should include your assembly language code and a screen dump of the ModelSim waveforms.

## Assembling

You may find that hand-assembly is quite error prone and laborious. On Vital you’ll find a MIPS assembler (MARS 4.1) written in JAVA that will help you assemble your code. To get this to assemble code starting at location 0x0000000, select *“Settings->Memory Configuration->Compact, Text at Address 0”* that will ensure that any jumps have the correct memory location encoded.

## Assignment 3 Part B – 60%

The MIPS design presented in MIPS_System only implements a limited number of the MIPS instructions. For the R-Type instructions ADD, ADDU, SUB, SUBU, AND, OR and SLT are implemented. Your task is to modify the MIPS design so that it implements the additional instructions shown in Table 1 whilst still ensuring the existing instructions work correctly. Once you have modified your design you need to write a program to demonstrate that your hardware correctly implements the instructions. Your results should include print outs of ModelSim showing your program operating. Annotate the print out to explain what is happening. You should submit an electronic copy of your design and assembly language programs onto Vital. Your written report should explain what modifications you have made to the Verilog code and include the Verilog code you have developed. There is no need to include the Verilog code for the modules you haven’t modified. You should also include ASM/ASMD charts for your modified code. For your report on instruction 3 you should include a block diagram showing the extra data pathways you have added.

## Submission Deadline

Electronic copy: Wednesday 21st April 2021 @ 11:59pm

### Table 1 Instructions to implement

| ID | Name | Instruction 1 | Instruction 2 | Instruction 3 |
|---|---|---|---|---|
| 201360923 | Ayeni, Olawale Solomon | nor | andi | lb |
| 201447264 | Bai, Jie | xor | andi | lbu |
| 201315796 | Brown, Connor | nor | andi | lh |
| 201447455 | Ding, Qiyang | xor | xori | lhu |
| 201447513 | Fang, Zhaoyan | nor | xori | lb |
| 201357067 | Glowacki, Jakub | xor | xori | lbu |
| 201309197 | Hankinson, Thomas | nor | xori | lh |
| 201447621 | He, Jiayi | xor | andi | lhu |
| 201375874 | Huang, Yiming | nor | andi | lb |
| 201447715 | Huang, Zhongling | xor | andi | lbu |
| 201447717 | Huo, Kairun | nor | xori | lh |
| 201275074 | Ike, Nnadozie Jason | xor | xori | lhu |
| 201447758 | Jiang, Ziyi | nor | xori | lb |
| 201447759 | Jiao, Jian | xor | xori | lbu |
| 201447777 | Jing, Jiaxi | nor | andi | lh |
| 201347216 | Kovvuri, Sandeep Reddy | xor | andi | lhu |
| 201447803 | Lei, Jiaheng | nor | andi | lb |
| 201305681 | Lewis, Oliver Emlyn | xor | xori | lbu |
| 201447904 | Li, Yang | nor | xori | lh |
| 201447912 | Li, Yu | xor | xori | lhu |
| 201447973 | Lin, Yu-Cheng | nor | xori | lb |
| 201447999 | Liu, Jiacheng | xor | andi | lbu |
| 201298366 | Liu, Jie | nor | andi | lh |
| 201448055 | Liu, Yimian | xor | andi | lhu |
| 201448053 | Liu, Yinhang | nor | xori | lb |
| 201448076 | Liu, Zheyu | xor | xori | lbu |
| 201448082 | Long, Heyang | nor | xori | lh |
| 201448190 | Niu, Yusen | xor | andi | lhu |
| 201261946 | Parhar, Jai Singh | nor | andi | lb |
| 201448219 | Qi, Kexin | xor | andi | lbu |
| 201290370 | Schlaefli Morales, Fernando Rafael | nor | xori | lbu |
| 201448466 | Wang, Kefan | xor | xori | lh |
| 201219640 | Wang, Xuesong | nor | xori | lhu |
| 201448540 | Wang, Zhaoyang | xor | xori | lb |
| 201376946 | Xiang, Li | nor | andi | lbu |
| 201376978 | Xiong, Jiangao | xor | andi | lh |
| 201376988 | Xu, Congyi | nor | andi | lhu |
| 201448708 | Xue, Chunyan | xor | xori | lb |
| 201448725 | Yan, Yujie | nor | xori | lbu |
| 201448759 | Yang, Xintong | xor | xori | lh |
| 201448812 | Yin, Ruihong | nor | andi | lhu |
| 201448839 | Yu, Tiehan | xor | andi | lb |
| 201448855 | Yuan, Lixiang | nor | andi | lbu |
| 201448922 | Zhang, Jinhao | xor | xori | lh |
| 201449028 | Zhang, Zhanhua | nor | xori | lhu |
| 201449062 | Zhao, Tiange | xor | xori | lb |
| 201449111 | Zhou, Hang | nor | xori | lbu |