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	Assignment 3 = 20%
Semester	2
HE Level	6
Lab location	EEE Building PC labs 301, 304 as timetabled – Tuesday 1-4
Work	Individually
Timetabled time	21 hours (3 hours per week – Tuesday 1pm – 4pm)
Suggested private study	10 hours including report writing
How much time did it take you?	Let us know anonymously via https://bit.ly/EEECARES
Assessment method	Individual, formal word-processed reports (Block diagrams and ASMs can be hand drawn and scanned into the report)
Submission format	Online via Canvas
Submission deadline	Assignment 3: Sunday week 8 of Semester 2 24th March 2024
Late submission	Standard university penalty applies
Resit opportunity	Students failing the module and Assignment 3 will have an alternative assignment in the Summer
Marking policy	Marked and moderated independently
Anonymous marking	Yes
Feedback	Via comments on CANVAS submission on-line
Learning outcomes	LO1: Ability to design digital systems using the ASM design method LO2: Ability to implement digital systems using the Verilog Hardware Description Language 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. • Comprehensible language; punctuation, grammar and spelling accurate. • Equations legible, numbered and presented correctly. • Appropriately formatted reference list.	• Appropriate use of technical, mathematic and academic terminology and conventions. • Word processed with consistent formatting. • Pages numbered, figures and tables captioned. • All sections clearly signposted. • Correct cross-referencing (of figures, tables, equations) and citations.
Introduction, Method and Design	40%	• Problem background introduced clearly. • Evidence of a Top Down Design approach • Conceptual Design Choices introduced. • Design of each module follows a logical sequence. • ASMs correspond to designs for each block. • Software is clearly commented	• Appropriate range of references used. • Design decisions justified with alternatives given. • Calculations shown in full, justifying and explaining any decisions. • Correct ASM Syntax used. • Well-structured Verilog Code
Results	30%	• Simulation results present for each block and well annotated. • Results of full system in both simulation and experimentally presented. • Results for each task accompanied by a commentary. • Screen shots of results presented.	• Tests indicate that there are no problems caused by asynchronous inputs. • Clear explanation of how the instructions operate correctly
Discussion	10%	• Discussion on what worked and what didn't. • Critical assessment on the design – strength and weaknesses	• Discussion on how the system was fully tested.

ELEC373 Verilog Assignment 3 (2023-2024)
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 a 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 CANVAS. 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 the file 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 CANVAS, 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

Compile and download the design, you should see that it switches on some of the red LEDs.

SignalTap Logic Analyser

You should configure the SignalTap logic analyser 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. Show that your program functions correctly by taking a screen shot(s) of the SignalTap Logic analyser showing your program executing your modified programme. Make sure that the instruction and programme counter can be read.
3. In your report you should include your assembly language code and a screen dump of the SignalTap Logic analyser. Also include a photograph of the 7 segment displays showing your ID.

Assembling

You may find that hand-assembly is quite error prone and laborious. On CANVAS 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 programs to demonstrate that your hardware correctly implements the instructions. Your results should include print outs of the SignalTap logic analyser 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 CANVAS. 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: Sunday 24th March 2024 @ 11:59pm

Table 1 Instructions to implement

ID	Name	Instruction 1	Instruction 2	Instruction 3
201563687	Al Meraikhi, Turki	nor	andi	lb
201579371	Alzeyara, Saoud	xor	andi	lbu
201503123	Bounds, Dominic Gregor	nor	andi	lh
201676076	Cai, Xinghu	xor	xori	lhu
201556978	Cardwell, Stephen	nor	xori	lb
201676131	Chen, SHI	xor	xori	lbu
201341221	Desmond, Con Patrick	nor	xori	lh
201676280	Feng, Yiming	xor	andi	lhu
201676288	Fu, Yongchuan	nor	andi	lb
201556875	Gardiner, joshua	xor	andi	lbu
201511440	Gill, harvey	nor	xori	lh
200956435	Glover, Aaron Phillip	xor	xori	lhu
201676340	Guan, Jiale	nor	xori	lb
201676397	He, Zhengyang	xor	xori	lbu
201676470	Huang, Zijian	nor	andi	lh
201676486	Jia, Yuming	xor	andi	lhu
201676496	Jiang, Jiakun	nor	andi	lb
201676499	Jiang, Qingyuan	xor	xori	lbu
201676555	Lam Po Tang, Justin	nor	xori	lh
201676564	Li, Bohang	xor	xori	lhu
201676567	Li, Derun	nor	xori	lb
201600584	Li, Yiyuan	xor	andi	lbu
201676658	Liang, Chen	nor	andi	lh
201676684	Lin, Shaowei	xor	andi	lhu
201676686	Lin, Wenhao	nor	xori	lb
201676707	Liu, Ke	xor	xori	lbu
201676808	Ma, Xuheng	nor	xori	lh
201676815	Mao, Haolin	xor	andi	lhu
201551565	McCue, Francis	nor	andi	lb
201654980	Mohamad Zaid, Asyraaf Asyraaf	xor	andi	lbu
201536424	Pan, Jiachen	nor	xori	lh
201676865	Peng, Yukun	xor	xori	lhu
201676889	Qiu, Chufan	nor	xori	lb
201676892	Qiu, Minhao	xor	andi	lbu
201676902	Rabetokotany, Toavina	nor	andi	lh
201563996	Reade, Brandon Stuart	xor	andi	lhu
201676945	Shen, Yixiao	nor	xori	lb
201472463	Smith, Will	xor	xori	lbu
201676987	Su, Zihan	nor	xori	lh

201522294	Sun, Bin	xor	andi	lhu
201677013	Sun, Zhijia	nor	andi	lb
201677015	Tan, Lige	xor	andi	lbu
201677106	Wang, Siheng	nor	xori	lh
201677116	Wang, Xinyi	xor	xori	lhu
201677124	Wang, Xirui	nor	xori	lb
201677148	Wang, Yanchang	xor	andi	lbu
201600972	Wang, Yihang	nor	andi	lh
201384487	Wu, Zijia	xor	andi	lhu
201677291	Xu, Xiufa	nor	xori	lb
201677298	Xu, Yankai	xor	xori	lbu
201677359	Yang, Yang Yi	nor	xori	lh
201677430	Zeng, Zhijie	xor	andi	lhu
201677437	Zhang, Baicheng	nor	andi	lb
201677450	Zhang, Haoran	xor	andi	lbu
201677461	Zhang, Jinsong	nor	xori	lh
201677467	Zhang, Keying	xor	xori	lhu
201677471	Zhang, Mingyu	nor	andi	lb
201677575	Zheng, David	xor	andi	lbu
201677619	Zhu, Changwei	nor	andi	lh
201639455	Zhu, Zhiyuan	xor	xori	lhu
		nor	xori	lb