BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CSE 306 JANUARY 2021 SEMESTER
COMPUTER ARCHITECTURE SESSIONAL

Assignment on 8-bit MIPS Design and Simulation

June 3, 2021

In this assignment, you have to design an 8-bit processor that implements the MIPS instruction set. Each instruction will take 1 clock cycle to be executed. The length of the clock cycle will be long enough to execute the longest instruction in the MIPS instruction set. The main components of the processor are as follows: instruction memory, data memory, register file, ALU, and a control unit. Additional components such as multiplexors, adders etc can added as required by your design.

1 DESIGN SPECIFICATION

- Address bus and data bus are multiplexed.
- Each of data and address has a size of 8-bits.
- An 8-bit *ALU* will be required, hence the name 8-bit MIPS.
- The *register file* must include the following temporary registers:

  $zero, $t0, $t1, $t2, $t3, $t4

  Each register has a size of 8-bits. The assembly code that will be provided to simulate your design will use only the above mentioned registers.
- The *control unit* should be micro-programmed. The control signals associated with the operations should be stored in a special memory (you can use a separate ROM for this purpose) units as Control Words.
- All clocks required in the circuit must be provided from a single clock source. Each instruction should be fetched and executed in a single clock cycle.

- Your design will be evaluated based on accuracy (correct flow of execution for each implemented instruction), completeness (all assigned instructions have been implemented), and efficiency (minimizing the number of ICs used, automated assembler).

2 Instruction Set Description

Instruction ID	Category	Type	Instruction
A	Arithmetic	R	add
B	Arithmetic	I	addi
C	Arithmetic	R	sub
D	Arithmetic	I	subi
E	Logic	R	and
F	Logic	I	andi
G	Logic	R	or
H	Logic	I	ori
I	Logic	R	sll
J	Logic	R	srl
K	Logic	R	nor
L	Memory	I	sw
M	Memory	I	lw
N	Control-conditional	I	beq
O	Control-conditional	I	bneq
P	Control-unconditional	J	j

3 MIPS Instruction Format

Our MIPS Instructions will be 20-bits long with the following three formats.

- R-type

Opcode	Src Reg 1	Src Reg 2	Dst Reg	Shft Amnt
4-bits	4-bits	4-bits	4-bits	4-bits

- I-type

Opcode	Src Reg	Dst Reg	Address / Immediate
4-bits	4-bits	4-bits	8-bits

- J-type

Opcode	Target Jump Address	0	0
4-bits	8-bits	4-bits	4-bits

4 Memory Considerations

For a single cycle implementation of MIPS instruction set, you need to have two separate memory units for instruction and data.

- Instruction Memory is accessed through an 8-bit address which is stored in an 8-bit Program Counter (PC) register. Each access to the instruction memory provides 20-bit (instruction) data.
- Data Memory is also accessed through an 8-bit address.

5 Instruction Set Assignment

The opcodes of the instruction will be between 0 to 15 based on the sequence of instruction id given below. Sequence ABCDEFGHIJKLMNOP means add instruction's opcode will be 0, addi instruction's opcode will be 1, sub instruction's opcode will be 2, and so on.

Group ID	Section A1	Section A2	Section B1	Section B2
1	FBKOINHDACFGELMP	GHCIAFEBFKDPLNOM	IEMFPDFNKLCHGBAO	PLIADGFBKFNCEMOH
2	HCAGNMIFDBOFEPKL	GAPOLNDMFKHCFEIB	KAGBLIPFNFMDCHEO	AKEHPDFOBCNLGMFI
3	HOCIKFEBAFNLDGMP	FDNKLIMFEOPBGCHA	IONHAKEBDLMFPCFG	GBFOLNAIEDFMKHCP
4	IOECDBPNKMGHFFLA	HFABDNKMPOLIECFG	AHLOGBFNMFDPECKI	HBDMNEKCPLAIGFFO
5	LNKIFHPBECOGFMAD	KILHFDPNBMFGEACO	EFNPOCDBIGAHMFKL	PAGIFOEHFNMDCLKB
6	POFMBNDECAGIHKLF	FLHGNIKAFBCMPEDO	MICHBDFLOAGFKPNE	CLIFMHOEBKFGDNAP

6 Extra Features

You have to implement push and pop operations in your MIPS design using a *Stack*. To achieve this task, the stack memory will be shared with the data memory in the following way. The data memory should start from the minimum address ($0x00$) and grow to the increasing memory addresses. On the contrary, the stack memory should start from the maximum address ($0xFF$) and grow to the decreasing memory addresses. The top of your stack will be held by a stack pointer (**$sp**). Initially **$sp** will hold an address of **0xFF** (highest address of the stack memory). The push and pop operations will be used in the provided assembly code according to the following table and you have to implement them using your MIPS instruction set.

Instruction	Description
push $t0	mem[**$sp**] = **$t0**
push 3($t0)	mem[**$sp**] = mem[**$t0**+3]
pop $t0	**$t0** = mem[**$sp**]

7 Simulation

To simulate your design, an assembly code will be used. Before starting the simulation, you have to convert the given assembly code into MIPS machine code and load the machine code into the instruction memory. The conversion process must be automatic. For example, you can write code in your preferred programming language for this conversion.

8 Report Content

Contents of the report are recommended as follows:

- Section 1: Introduction
- Section 2: Instruction Set
- Section 3: Complete Block diagram of an 8-bit MIPS processor. The block diagram must follows necessary descriptions.
- Section 4: Block diagrams of the main components.

  - Instruction memory with PC
  - Register file
  - Data memory with the Stack
  - Control unit
- Section 5: Approach to implement the push and pop instructions
- Section 6: ICs used with their count
- Section 7: Discussion

9 Submission Guideline

- Create a folder named “<Lab Group>_<Group ID>_Simulation" and put all the necessary simulation files in this folder.
- Create a second folder named “<Lab Group>_<Group ID>_Necessary_Content" and put all your codes to convert assembly codes into MIPS machine codes (or any others extra contents that will be necessary to simulate your design).
- Finally create a third folder named “<Lab Group>_<Group ID>_Submission" and put your report along with the previously prepared two folders. Now zip this final folder and upload the zip file to the Moodle submission link (single submission from each group).
- Remember that proper submission will carry 10% of total assignment marks.

Submission Deadline: Sunday, June 20, 2021 11:59 PM
For any kind of confusion, feel free to mail at `rezwana@teacher.cse.buet.ac.bd@gmail.com`, `tmadnan10@gmail.com` with the subject “MIPS Simulation".