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 Pipelined Execution

June 21, 2021

In this assignment, you have to design an 8-bit processor that supports pipelined datapath for a subset of MIPS instruction set. In this design, each instruction is divided into *five* stages: instruction fetch (IF), instruction decode (ID), execution and address calculation (EX), data memory access (MEM), and write back (WB). The length of the clock cycle equals the maximum time to execute any single stage. Therefore, each instruction takes up to *five* clock cycles to be executed. The main components of the processor are as follows: instruction memory, data memory, register file, ALU, control unit, five pipeline registers, and a forwarding unit. Additional components such as comparators, adders, mux etc can be 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 temporary 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.
- **Pipeline registers:** There registers will be used between any two stages of the instruction. A *pipeline register* can be a single register or a collection of registers. There is no restriction on the size of the pipeline 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.
- The **forwarding unit** is used to detect data hazard and generate control bits to allow data forwarding.
- All clocks required in the circuit must be provided from a single clock source.

2 Instruction Set Description

In this assignment, we will consider only the four R-type instructions. You may choose any opcode value for the given instructions.

Instruction ID	Category	Type	Instruction
A	Arithmetic	R	add
B	Arithmetic	R	sub
C	Logic	R	and
D	Logic	R	or

The format of an R- type instruction is given below.

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

3 Memory Considerations

Two separate memory units should be used 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.

4 Handling Hazards

In this assignment, we will only consider data hazards can occur. More precisely, we consider data hazards that can be resolved by **forwarding**.
In particular you have to detect and resolve the following hazards by using *forwarding* techniques.

- **EX Hazard:** occurs when the dependent instruct is in the EX stage and the prior instruction is in MEM stage.
- **MEM Hazard:** when the dependent instruction is in the EX stage and the prior instruction is in WB stage.
- **Double Data Hazard:** occurs when the dependent instruction is in EX stage and it depends on two prior instructions, one of which, is in MEM stage, and the other is in WB stage.

- In the following example, inst 1 creates EX hazard for inst 2 and MEM hazard for inst 3. Moreover, inst 2 and inst 3 both create double data hazard for inst 4.

```
inst 1:   add   $t1, $t2, $t3
inst 2:   sub   $t4, $t1, $t2
inst 3:    or   $t4, $t1, $t3
inst 4:   add   $t4, $t4, $t3
```

- Note that no stall operation is required for your implementation.

5 Simulation and Evaluation

To simulate your design, an assembly code consisting only 3-5 instructions will be used. Progress of your implementation will be evaluated based on the handling of the following scenarios.

- **Pipelined datapath only (60% marks):** No data dependency exists among the instructions.
- **Pipelined datapath and Ex Hazard only (60% + 15% marks):** An instruction is dependent on its preceding instruction.
- **Pipelined datapath and MEM Hazard only (60% + 15% marks):** An instruction is dependent on its 2^{nd} preceding instruction.
- **Pipelined datapath, EX Hazard, and MEM Hazard (60% + 15% + 15% marks):** An instruction i is dependent on its preceding instruction. Another instruction $j \neq i$ is dependent on its 2^{nd} preceding instruction.
- **Pipelined datapath, EX Hazard, MEM Hazard, and Double Data Hazard (60% + 15% + 15% + 10% marks):** An instruction is dependent both on its preceding and 2^{nd} preceding instructions.

6 Report Content

Contents of the report are recommended as follows:

- Section 1: Introduction
- Section 2: Complete Block diagram of pipelined datapath.
- Section 3: Block diagrams and size of pipeline registers
- Section 4: Mechanism and block diagram of forwarding unit
- Section 5: Discussion

7 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).

Submission Deadline: Sunday, July 04, 2021 11:59 PM
For any kind of confusion, feel free to mail at `rezwana@teacher.cse.buet.ac.bd` with the subject “Pipelined Datapath".