Alamein University
Faculty of Computer Science & Engineering

Course Name: Computer Architecture and Organization
Course Code: CSE132

Final-Project

The manager of the *ALameinBank* bank branch, located in a *Alamein* is proposing to install an embedded system to monitor the client queue in front of the tellers. The proposed system, called *ABQM™*, is to display various information about the status of the queue. The detailed specification of *ABQM™* is as follows:

1. Both queue ends are equipped with a photocell. Each photocell generates a logic '1' signal if nobody interrupts a light beam generated at the corresponding queue end. When the light beam is interrupted, the photocell output changes to logic '0' and stays at that value until it is no longer interrupted.
2. Clients are supposed to enter the queue only from the back end and leave only from the front end.
3. The number of people standing in the queue (waiting to be served by a teller), *Pcount*, and the expected waiting time in the queue before being served, *Wtime*, are to be displayed on **Seven Segment Display**.
4. *Pcount* is to be incremented by only one when a client enters the queue and is to be decremented by only one when a client leaves, even if a client stands in front of the light beam for a long time period.
5. *Wtime*, in seconds, could approximately be given by the formulas:

   $Wtime\ (Pcount = 0) = 0,$

   $Wtime\ (Pcount \neq 0, Tcount) = 3*(Pcount+Tcount-1)/Tcount$

   where *Tcount* is the number of tellers currently in service ($Tcount \in \{1,2,3\}$) and *Wtime* is rounded by ignoring the fraction part.
6. *BBqM™* maintains binary *empty* and *full* flags that reflect the status of the queue.
7. A responsible person should have the capability of resetting the system. Resetting the system clears the *full* flag and *Pcount*, and the sets *empty* flag.

The design team has met few times and decided on the following for the *BBqM™* implementation:

- The *BBqM™* will be implemented using the softcore of the single-cycle MIPS processor on FPGA.
- The maximum *Pcount* value will be (**2^n - 1**), with a default value of 7, where *n* is a generic value, with a default value of 3.
- **<u>There will be a system clock and reset for the MIPS processor.</u>**

In this project, you are going to model the operation of *BBqM™* **in C Language. Then translate some parts to MIPS assembly language and verify it via simulation utilizing Single-Cycle MIPS processor. Here is the list of deliverables:**

a) In a table, identify *BBqM™* inputs and outputs and briefly describe their meaning and possible values.
b) Draw a flow chart showing the *BBqM™* workflow.
c) Model the up-down counter as a separate function using C language.
d) Model the required functions in C language.
e) Translate the C function *Wtime* into MIPS assembly language.
f) Convert the *Wtime* assembly instructions, from (g) into machine code.
g) Upload the machine code of the *Wtime* into MIPS memory.
h) Verify your code by simulating the MIPS processor in ModelSim.
i) Identify two testcases and show that your code for *Wtime* is working correctly.

Although you can download the single-cycle MIPS on FPGA and verify the overall system, **it is not required in the current phase**.

Here are some hints:

- In order to translate *Wtime* into MIPS assembly language, you may save all probabilities for Wtime in an array. Or you may use shift right and shift left operations to apply multiplication and division. Or You may modify the ALU of MIPS to support multiplication and division operations!
- In order to get the flow, refer to section 7.6 in “Digital Design and Computer Architecture” By David and Sarah Harris.
- Figure 7.59 in chapter 7 illustrates a block diagram for the *single-cycle MIPS*.
- Download the single-cycle MIPS implementation from Here.

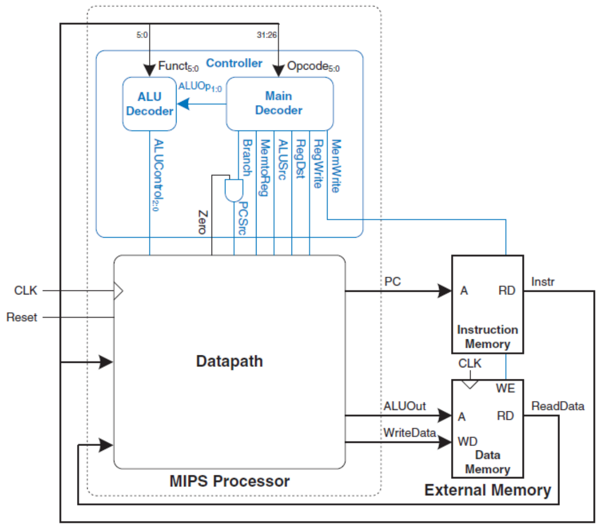

Figure 7.59 MIPS single-cycle processor interfaced to external memory

GOOD LUCK,
Dr. Ahmed Shalaby