CMPEN 331 – Computer Organization and Design,
Lab 4

This lab introduces the idea of the pipelining technique for building a fast CPU. The students will obtain experience with the design implementation and testing of the first four stages (Instruction Fetch, Instruction Decode, Instruction Execute, Memory) of the five-stage pipelined CPU using the Xilinx design package for FPGAs. It is assumed that students are familiar with the operation of the Xilinx design package for Field Programmable Gate Arrays (FPGAs) through the Xilinix tutorial available in the class website.

1. **Pipelining**

   As described in lab 4

2. **Circuits of the Instruction Fetch Stage**

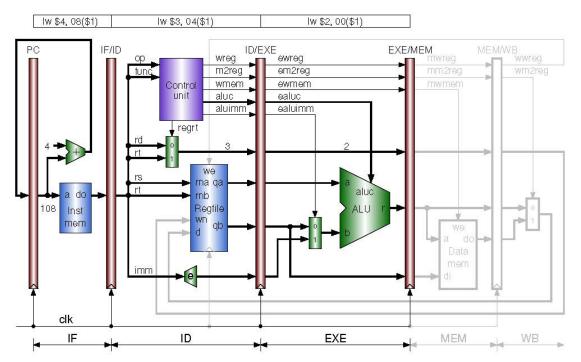   As described in lab 4

3. **Circuits of the Instruction Decode Stage**

   As described in lab 4

4. **Circuits of the Execu**

   Referring to Figure 1,   EXE stage. The ALU performs addition, and the multi   each control signal in order to distinguish it from that in the ID stage. The second instru   the ID stage and the third instruction is being fetched in the IF stage. All the four pip   at the end of the cycle.

5. **Circuits of the Memory Access Stage**

   Referring to Figure 2, (8.6) in the fourth cycle of the first instruction entered the MEM stage. The only task in this stage is to read data memory. All the control signals have a prefix "m". The second instruction entered the EXE stage; the third instruction is being decoded in the ID stage; and the fourth instruction is being fetched in the IF stage. All the five pipeline registers are updated at the end of the cycle.

Figure 1 Pipeline execution (EXE) stage

Figure 2 Pipeline memory access (MEM) stage

6.   Table 1 lists the names and usages of the 32 registers in the register file.

Table 1 MIPS general purpose register

Register Name	Register Number	Usage
$zero	0	Constant 0
$at	1	Reserved for assembler
$v0, $v1	2, 3	Function return values
$a0 - $a3	4 – 7	Function argument values
$t0 - $t7	8 – 15	Temporary (caller saved)
$s0 - $s7	16 – 23	Temporary (callee saved)
$t8, $t9	24, 25	Temporary (caller saved)
$k0, $k1	26, 27	Reserved for OS Kernel
$gp	28	Pointer to Global Area
$sp	29	Stack Pointer
$fp	30	Frame Pointer
$ra	31	Return Address

7.   Table 2 lists some MIPS instructions that will be implemented in our CPU

Table 2 MIPS integration instruction

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

8.   Initialize the first 10 words of the Data memory with the following HEX values:

A00000AA
10000011
20000022
30000033
40000044
50000055

```
60000066
70000077
80000088
90000099
```

9. Write a Verilog code that implement the following instructions using the design shown in Figure 2. Write a Verilog test bench to verify your code: (You have to show all the signals written into the MEM/WB register and output from EX/MEM register in your simulation outputs)

```
instruction                 comment
lw $2, 00($1)          # $2 ← memory[$1+00]; load x[0]
lw $3, 04($1)          # $3 ← memory[$1+04]; load x[1]
lw $4, 08($1)          # $4 ← memory[$1+08]; load x[2]
lw $5, 12($1)          # $5 ← memory[$1+12]; load x[3]
```

Assume that register $1 has the value of 0

10. Write a report that contains the following:
    a. Your Verilog design code. Use:
        i. Device: XC7Z010- CLG400 -1 or choose any other FPGA type. You can use Arria II if you are using Quartus II software.
    b. Your Verilog Test Bench design code. Add 'timescale 1ns/1ps as the first line of your test bench file.
    c. The waveforms resulting from the verification of your design with ModelSim showing all the signals written into th
    d. The design sch   ot use any area constraints.
    e. Snapshot of th
    f. Snapshot of th

11. REPORT FORMAT: Free form, but it must be:
    g. One report per student.
    h. Have a cover sheet with identification: Title, Clas
    i. Using Microsoft word and it should be uploaded in word format not PDF. If you know LaTex, you should upload the Tex file in addition to the PDF file.
    j. Double spaced

12. **You have to upload the whole project design file zipped with the word file.**