Class:	**CPE300L 1001**	Semester:	**Fall 2024**

Points		Document author:	**Alireza Bolourian**
		Author's email:	**bolouria@unlv.nevada.edu**
		Document topic:	**Postlab 08**

Instructor's comments:

1. Theory of Operation

In a MIPS single-cycle processor, each instruction is completed in a single clock cycle. This makes the single clock cycle longer so it can accommodate the more complex instructions. It also utilizes more resources for the hardware components. In MIPS multi-cycle processor, instructions are separated into multiple stages with each stage being completed in one clock cycle. This clock cycle needs to only accommodate the longest stage not the entire instruction. This makes the clock shorter, and it is possible to reuse hardware components such as ALU. It is, however, harder to implement due to additional control logic and registers to store intermediate results.

2. Prelab:

I-format

addi rt, rs, immediate Requires 4 cycles

opcode	rs	rt	immediate
6 bits	5 bits	5 bits	16 bits

Cycle 1

PCwrite	1
IorD	0
memRead	1
memWrite	0
IRWrite	1
PCSource	00
ALUOp	00
ALUSrcB	01
ALUSrcA	0
Regwrite	0

Cycle 2

ALUOp	00
ALUSrcB	00
ALUSrcA	0

Cycle3

ALUOp	00
ALUSrcB	10
ALUsrA	1
PCwrite	0

Cycle 4

memtoReg	0
RegWrite	1
RegDst	0

J-format

J target Requires 2 cycles

opcode	offset
6 bits	26 bits

Cycle 1

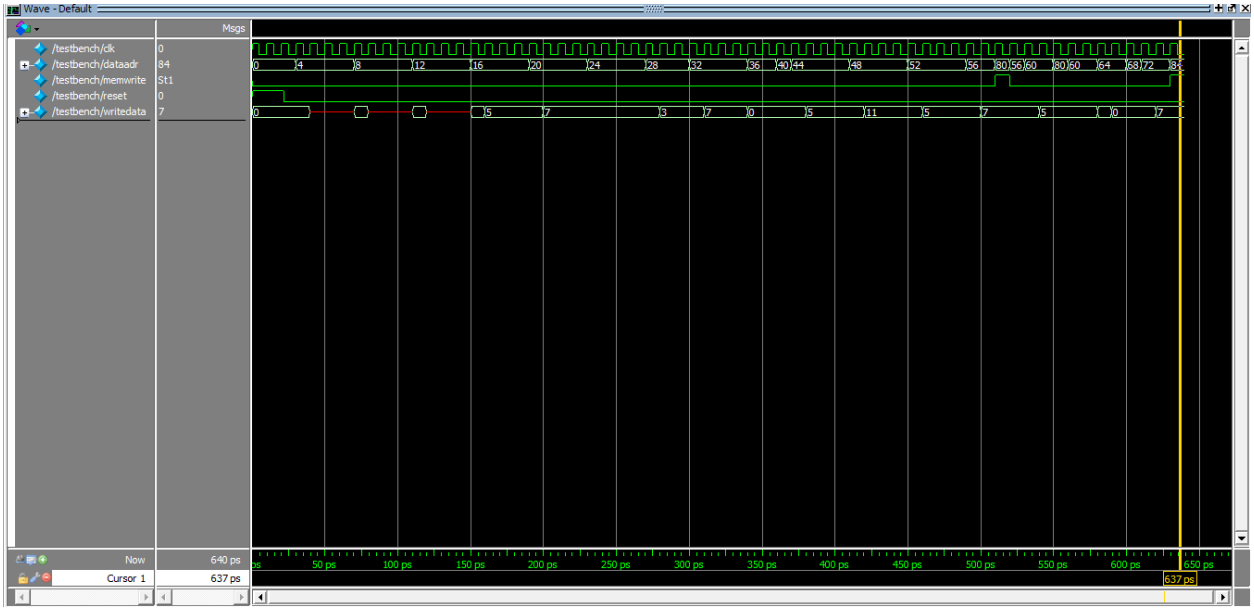
PCwrite	1
IorD	0
MemRead	1
memWrite	0
IRWrite	1
PCSource	00
ALUop	00
ALUSrcB	01
ALUSrcA	0
RegWrite	0

Cycle 2

PCSource	10
PCwrite	1

3. Results of the experiments

The MIPS Code is uploaded separately.




```
# Loading work.regfile
# Loading work.signext
# Loading work.sl2
# Loading work.mux4
# Loading work.mux3
# Loading work.mem
VSIM 38> run 100
VSIM 39> run 1000
# Simulation succeeded
# Break in Module testbench at D:/lab08/testbench.v line 28
```

The instruction code performs the same operations as in prelab 07. The waveform confirms that value 7 is in address 84 as expected.

4. Questions

1. What are the typical stages in MIPS multicycle implementation. List them all.

1. Instruction Fetch: The instruction is fetched from memory and the Program Counter (PC) is incremented.
2. Instruction Decode: The instruction is decoded, and the necessary registers are read.
3. Execution: The operation specified by the instruction is performed. This could involve arithmetic operations, address calculations, or determining branch targets.
4. Memory Access: Data is read from or written to memory if needed.
5. Write Back: The result of the instruction is written back to the register file

5. Conclusions

In this lab, I learned about the tradeoffs between single-cycle and multi-cycle MIPS. Even though more complex, the cycle time is reduced by dividing the tasks into separate stages. However, the improvement in cycle time is not as noticeable as would be expected due to the register delays added in each stage. In addition while one instruction is taking place in one stage, every other stage is waiting for the whole instruction to end. This multi-cycle processor can be further improved by the pipelining architecture.