

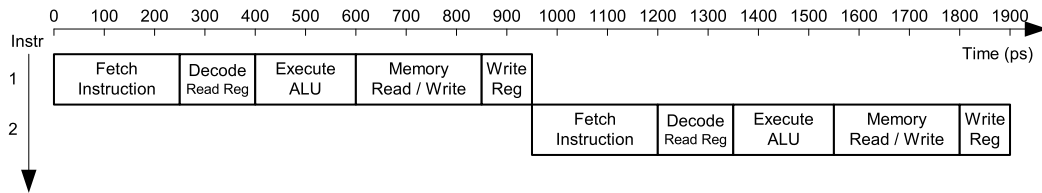
University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS150, Spring 2012

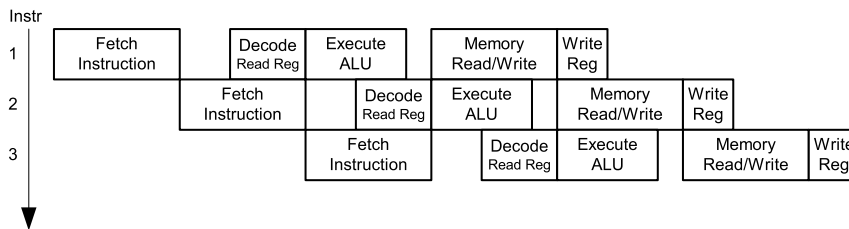
Homework Assignment 4: MIPS Architecture
Due February 16th, 2pm

1. In a single cycle MIPS processor, suppose the control signal *RegWrite* has a *stuck-at-1* fault, meaning that the signal is always 1 regardless of its intended value. Write an assembly program which would reveal this fault. Describe how it works by comparing the intended result and the observed result.
2. The diagram below is taken from the lecture notes. Assume *FetchInstruction* takes 250ps, *Decode/ReadReg* takes 150ps, *Execute/ALU* takes 200ps, *MemoryRead/Write* takes 250ps and *WriteReg* takes 100ps, and also assume no performance overhead is introduced when the pipelining registers are inserted.

Single-Cycle



Pipelined



- (a) We have a program of 2000 instructions running on the pipelined MIPS processor, and after the pipeline is filled, one instruction is completed each cycle. What is the total time required to run this program in the pipelined processor? How long would it take if you run the program on the single-cycle processor?
 - (b) A student redesigned the ALU, now *Execute/ALU* only takes 150ps. Repeat part 2a.
3. Assuming no data forwarding has been implemented in a pipelined MIPS processor, when would the processor be stalled if the code below are executed, assuming all state elements are posedge triggered?

```

sub    $t3, $s0, $s2
lw     $t4, 20($s2)
add    $t2, $t3, $s3
or     $t3, $t4, $t2

```

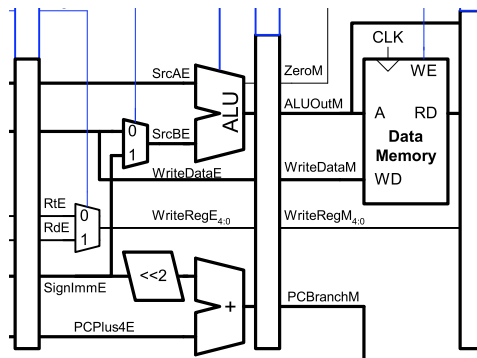
4. You are given the assembly code shown below, and it is run on a MIPS processor with 5 stage pipeline.

```

0x7C          add $t1, $t2, $t3
0x80          add $t5, $t4, $t1

```

- Explain what kind of hazard is present in the code above, and how many extra cycles are introduced if the MIPS processor has no data-forwarding. Assuming all state elements in the processor are posedge triggered.
- If the register file is written on the negedge of the clock, how many cycles would the two instructions take?
- To improve the performance, forwarding is implemented such that the inputs to ALU can be from the Memory stage. How would you modify the part of the datapath shown below for this to be possible? Name any new control signals you have added.



(d) In Verilog, design the control circuit coordinating the forwarding. Use the name of the signals from the lecture notes, and from part 4c.

5. A student is given a piece of C code, which he translates to an assembly program. However, the assembly code seems to give wrong result when it executes. Correct his error and also minimize the run time of the program.

C code

```
int sum=0;
int i;
for (i=0; i!=j; i=i+1)
{
    sum=sum+in1[i]-in2[i];
}
```

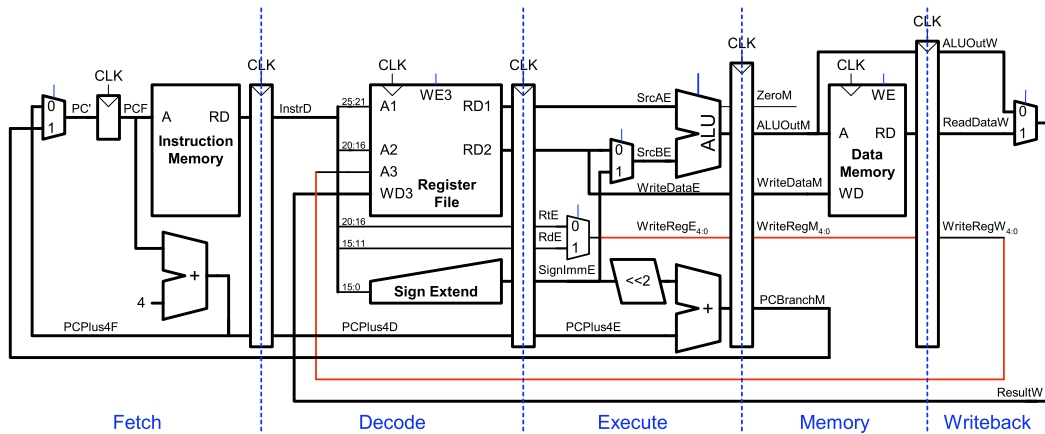
Assembly Code

```
# $s0 = i, $s1 = sum,
# $s2 holds j, $s3 holds pointer in1
# $s4 holds pointer in2
    addi $s1, $0, 0
    add  $s0, $0, $0
for:  add  $t1, $s3, $s0
      add  $t3, $s4, $s0
      lw   $t2, 0($t1)
      add  $s1, $t2, $s1
      lw   $t5, 0($t3)
      sub  $s1, $s1, $t5
      bne  $s0, $s2, for
      addi $s0, $s0, 1
```

6. You are to add a *store with postincrement* instruction to a MIPS processor with 5 pipeline stages. The instruction **swinc** updates the index register to point to the next memory word after completing the store. **swinc \$rt, imm(\$rs)** is equivalent to the following two instructions:

```
sw      $rt, imm($rs)
addi    $rs, $rs, 4
```

(a) How would you modify the following datapath to accommodate this instruction? Try to add as little hardware as possible.



(b) The following assembly program uses the new instruction, what hazard do you see? Modify the datapath you created in 6a to eliminate the stalls introduced.

```
swinc  $t3, 0x10($s2)
add    $t2, $s2, $s3
```