Name: KEY **Id#**

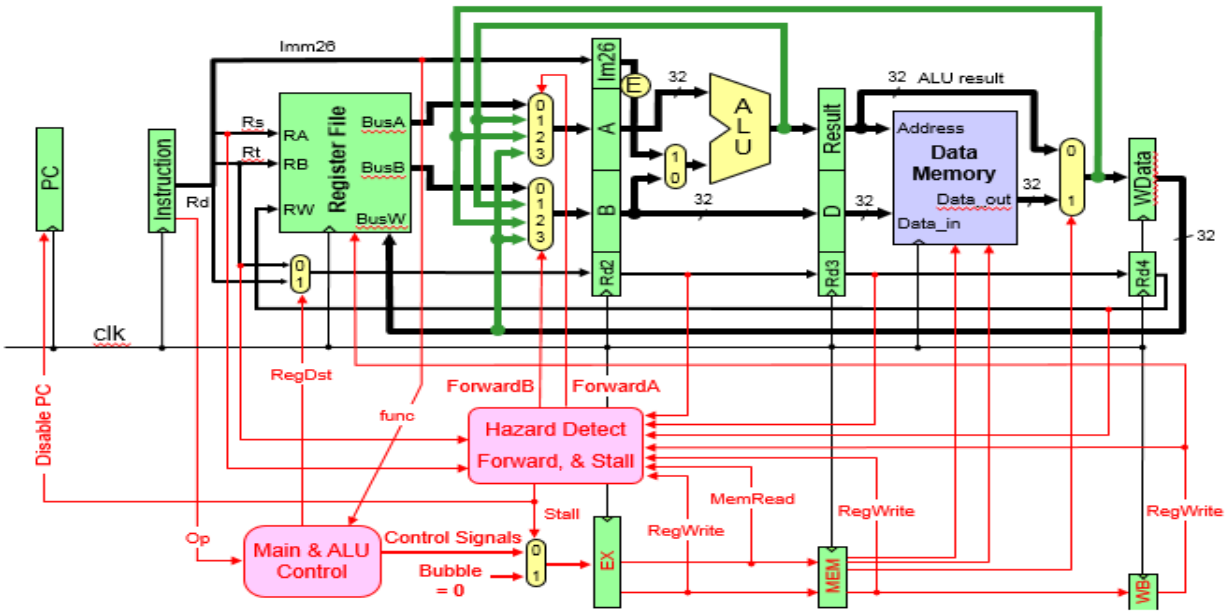
ICS 233, Term 142

Computer Architecture & Assembly Language

Quiz# 6

Date: Tuesday, May 12, 2015

Q1. Consider the pipelined MIPS processor design given below:



(i) Show the conditions that will be used for generating the ForwardA signals. In case both forwarding conditions from the ALU and Memory Mux are met for the ForwardA MUX, which one should be allowed to forward?

```
If       ((Rs != 0) and (Rs == Rd2) and (EX.RegWrite))      ForwardA ← 1
Else if  ((Rs != 0) and (Rs == Rd3) and (MEM.RegWrite))     ForwardA ← 2
Else if  ((Rs != 0) and (Rs == Rd4) and (WB.RegWrite))      ForwardA ← 3
Else                                                        ForwardA ← 0
```

In case both forwarding conditions from the ALU and Memory Mux are met, the value from the ALU is the one to be forwarded as it has the latest value in the destination register.

(ii) State the conditions that will be used for stalling the pipeline in the case of detecting RAW Hazard after Load.

```
if       ((EX.MemRead == 1)          // Detect Load in EX stage
and (ForwardA==1 or ForwardB==1))        Stall  // RAW Hazard
```

Q2. Consider the code given below:

```
add  $1, $1, $2
sub  $1, $1, $3
lw   $2, ($1)
addi $2, $2, 4
sw   $2, ($1)
```

(i) Identify all the **RAW** data dependencies in the above code. Which dependencies are data hazards that will be resolved by forwarding? Which dependencies are data hazards that will cause a stall?

RAW dependencies:

add $1, $1, $2	and	sub $1, $1, $3	(forwarding)
sub $1, $1, $3	and	lw $2, ($1)	(forwarding)
lw $2, ($1)	and	addi $2, $2, 4	(stall 1 cycle & forwarding)
addi $2, $2, 4	and	sw $2, ($1)	(forwarding)

(ii) Using a multiple-clock-cycle graphical representation, show the instruction execution across the pipeline including forwarding paths and stalled cycles if any. How many clock cycles will be needed to execute the instructions?