Name Key

Rose-Hulman Institute of Technology
Electrical and Computer Engineering

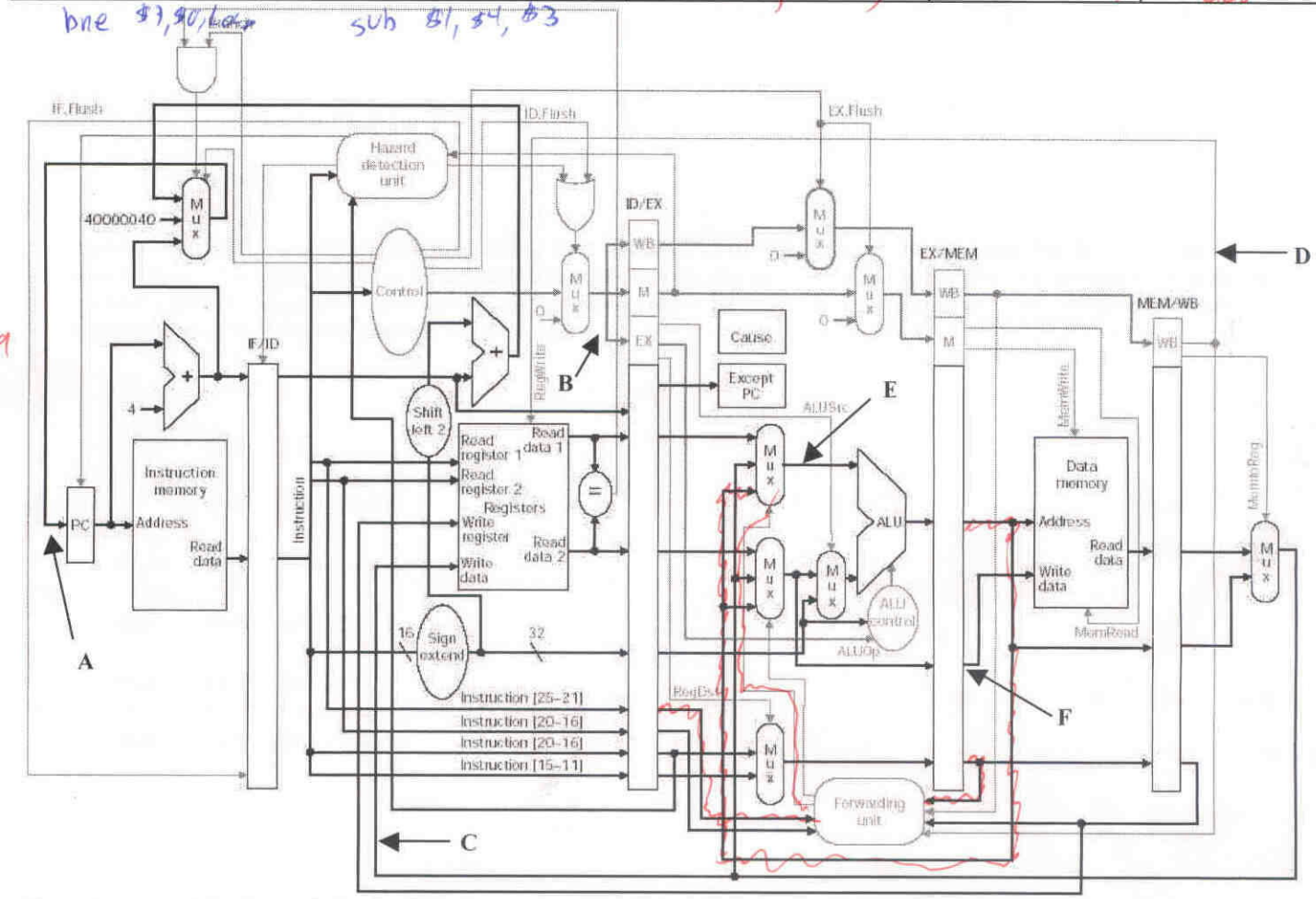
EC 332 - Exam 1 — Thursday, December 19th, 2002

CLOSED BOOK. Work each problem in the space provided on its sheet. Be sure the work you present is clear so I can understand what you have done. You may use one 3" by 5" card of notes. No other aids, animate or inanimate, are permitted. Please do your own work.

Problem 1 [25 points] - The code below is to be run on the pipelined datapath shown below. Assume the datapath **has** hardware support for forwarding and the branch operation is performed in the Decode stage. Also, a register can be written and read during the same clock cycle.

IF: sub $1,$4,$3	ID:	EX: lw $4, 10($2)	MEM: sw $3,10($2)	WB: add $2,$3,$

bne $3,$0,Loop sub $1, $4, $3



Show the state of the datapath for the following code sequence when the first **add** instruction is in the **WB** stage by filling the slots below with the values for the indicated lines. If you do not know the value of the signal, use the notation **$X** to indicate the contents of register **X** and **M($X)** to indicate the contents of memory location **X**. Assume the "zero" input to all MUXs is the upper input. **Be sure to write the instruction above each stage.**

PC

```
A00h          li    $1,8
A004h         li    $2,5
A008h         li    $3,1
A00Ch  Loop:  add   $2, $3, $1
A010h         sw    $3, 10($2)
A014h         lw    $4, 10($2)
A018h         sub   $1, $4, $3
A01Ch         Bne   $3, $0, Loop
```

3 pts. -1 if value not shown

A. A018h (C) A018h (20)
B. 0 sub
C. $2 = 9
D. RegWrite = 1
E. $2 = 9
F. $3 = 1

10($2)

Problem 2 [25 points] - Suppose each stage of a pipeline takes the following times:

IF	ID	EX	MEM	WB
1ns	1ns	1ns	3ns	1ns

a. How often can a new instruction be fetched in this pipeline?

3ns

A design team decides to split the MEM stage into two stages. The times for the new pipeline are:

IF	ID	EX	MEM1	MEM2	WB
1ns	1ns	1ns	1.5ns	1.5ns	1ns

b. How often can an instruction be fetched in this new pipeline?

1.5ns

new

1

c. Assume the pipeline above has data forwarding from both **MEM** and **WB** stages to the **EX** stage, but does not stall for data hazards. Therefore, you must insert nop instructions explicitly into your code so that the loop executes correctly on the new pipeline. Also assume that there is no forwarding to the **ID** stage. (To save space, you may designate multiple nops with "nop N" notation, where N is an integer)

```
        li    $t0, 0
        li    $t1, 20
Loop:   lw    $t2, 100($t0)
        nop 2
        addi  $t2, $t2, 10
        nop
        sw    $t2, 100($t0)
        addi  $t0, $t0, 4
        nop 3
        bne   $t0, $t1, Loop
```

IF ID EX M1 M2 WB

IF ID EX M1 M2 WB

IF ID EX M1 M2 WB

IF ID EX M1 M2 WB

IF ID EX M1 M2 WB

IF ID EX M1 M2 WB

Branch done in ex stage

IF ID EX

IF ID EX

IF ID EX

d. For what is the 40000040 input to the MUX to the left of the IF/ID pipeline register used on the datapath on page 1?

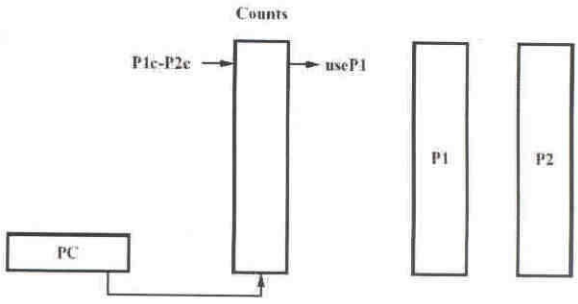
exceptions

4 pts each

Problem 3 [25 points] - Short Answer.

a) Describe how a Combined Predictor works.

It used 2 predictors and notes which one is works best.
It uses the best predictor.



b) Why does a Combined Predictor get better performance than non-combined predictors?

It can pick uses the best of the non-combined predictors.

c) Given the pipelined MIPS implementation, a friend suggest adding dynamic branch prediction to speed things up. Is this a good idea? Explain.

No. There is only 1 delay slot, so the extra hardware won't help.

d) Acording to Keith Diefendorff in *PC Processor Microarchitecture*, what is the biggest performance obstacle? Why?

Memory Speed.

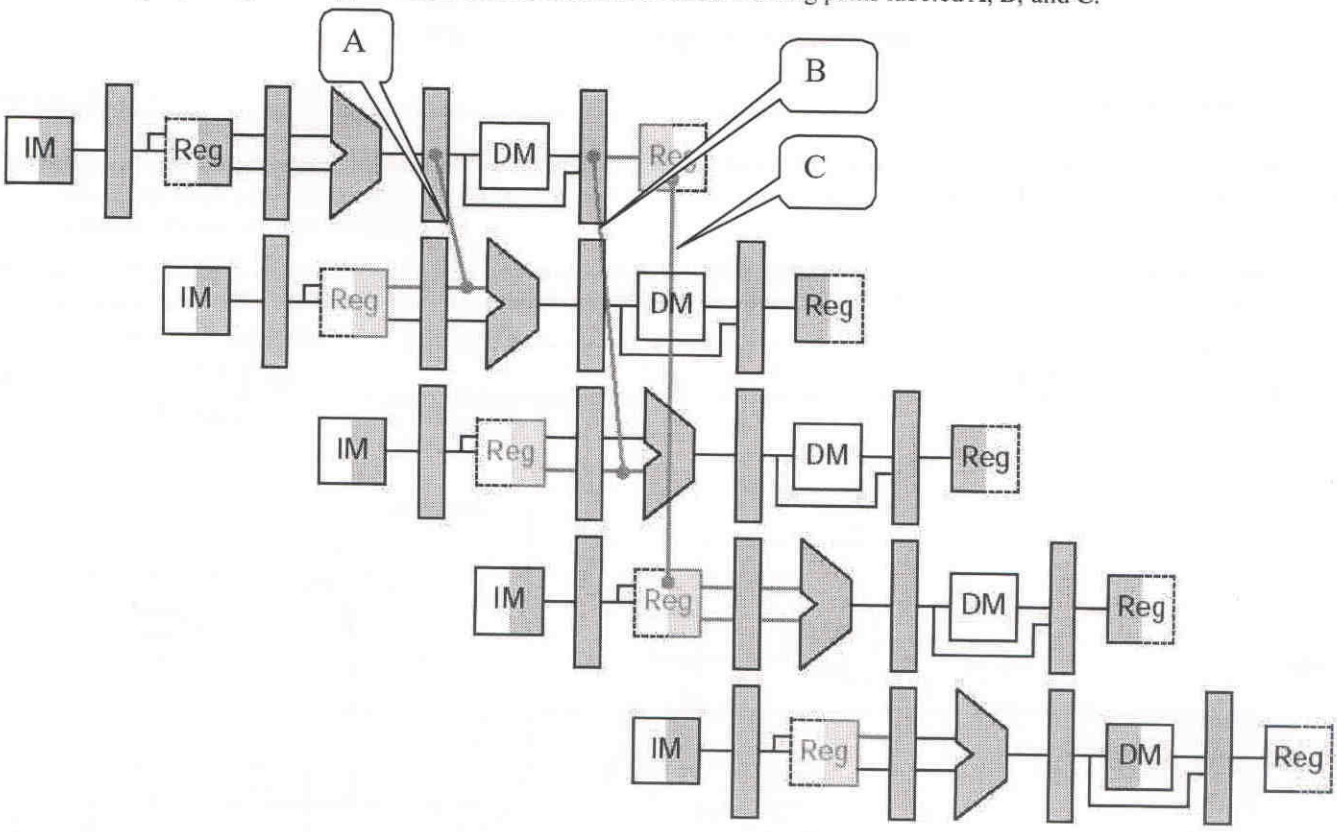
e) For gshare, which of these stores only the global history?

a. GR
b. PC
c. GR⊕PC
d. Counts table

f) For gshare, which of these stores the local history, but not the global?

a. GR
b. PC
c. GR⊕PC
d. Counts table

Problem 4 [25 points] - The figure below shows three different forwarding paths labeled **A**, **B**, and **C**.



a. On the datapath on page 1, highlight the hardware and the signal lines that are used for the forwarding path **A**.

b. What is necessary for forwarding path **C** to work?

Write Reg first 1/2 half of $ cycle
Read Reg 2nd second half of cycle

c. Below is some control logic for handling the forwarding. Give a code sequence that will satisfy the forward logic below and cause the data to be forwarded to the first input of the ALU.

```
If (MEM/WB.RegWrite
    and (MEM/WB.RegisterRd ≠ 0)
    and (EX/MEM.RegisterRd ≠ ID/EX.RegisterRs)
    and (MEM/WB.RegisterRd = ID/EX.RegisterRs)) forwardA = 01
```

add $1, $1, $1
sub $2, $1, $1
add $1, $1, $1

add $1, x, x
sub $1, x, x
add x, $1, x

d. In the forwarding control logic above, why is the line "`and (EX/MEM.RegisterRd ≠ ID/EX.RegisterRs)`" needed?

So this → add $1, x, x
sub $1, y, y
add z, $1, w

will work right. i.e. sub is forwarded, not first add.