

EECE 321 - MICROPROCESSOR SYTEMS

May 18, 2004

NAME: _____ ID: _____ Section: 9 AM or 10 AM

CLOSED BOOK/CLOSED NOTES (2 Hours)

WRITE YOUR NAME AND ID NUMBER IN THE SPACE PROVIDED ABOVE.

PROVIDE YOUR ANSWERS IN THE SPACE PROVIDED ON THE QUESTION SHEET.

THE SCRATCH BOOKLET WILL NOT BE CONSIDERED IN GRADING.

BE AS CLEAR AND AS NEAT AS POSSIBLE.

CIRCLE YOUR SECTION TIME.

Problem	Points	Over
1		20
2		20
3		16
4		55
5		52
6		17
Total		180

Problem 1: General questions (20 points)

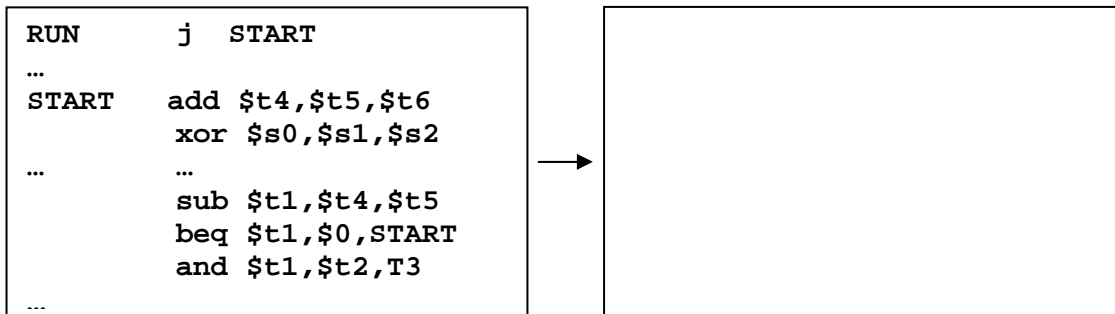
- 1) What is the speedup from pipelining on a k -stage pipelined processor that stalls on average m clock cycles per instruction over an unpipelined processor? (2 points)

Speedup =

- 2) **True/False:** On a k -stage pipelined processor, any instruction takes at most k clock cycles to execute. (1 point)
- 3) **True/False:** Pipelining decreases the average execution time of an instruction. (1 point)
- 4) The situation where an instruction in the pipeline may need a resource being used by another instruction in the pipeline is called a _____. (1 point)
- 5) Assume the following MIPS code is to be executed on a 5-stage MIPS pipelined processor with support for forwarding from WB to MEM and EX stages. If the compiler reschedules the code to maximize performance, how many times will the pipeline stall? Answer = _____. (2 points)

```
add $t2, $t3, $t4
sw  $t2, 5($t2)
lw  $t1, 7($t3)
add $t5, $t1, $t3
```

- 6) Reschedule the following code to reduce the penalty of a control hazard using the *branch delay slot*: (3points)



- 7) **True/False:** In dynamic branch prediction, the compiler predicts the behavior of a branch and reschedules it to minimize the penalty of a control hazard. (1 point)
- 8) **True/False:** The control unit on a pipelined MIPS processor handles overflow exceptions by flushing all pipeline registers. (1 point)
- 9) In virtual memory with 34-bit virtual addresses and 32KB pages, the size of a 128-entry fully-associative TLB including dirty bits is at least _____ bits. (3 points)
- 10) Complete the following entries with “Decreases”, “Increases”, or “No Effect”. (5 points)

Cache Parameter	Miss Penalty	Hit Time	Capacity misses	Conflict Misses	Compulsory Misses
Larger Cache Size					
Larger Block Size					
Higher Associativity					

Problem 3: Multi-cycle MIPS Datapath (16 points)

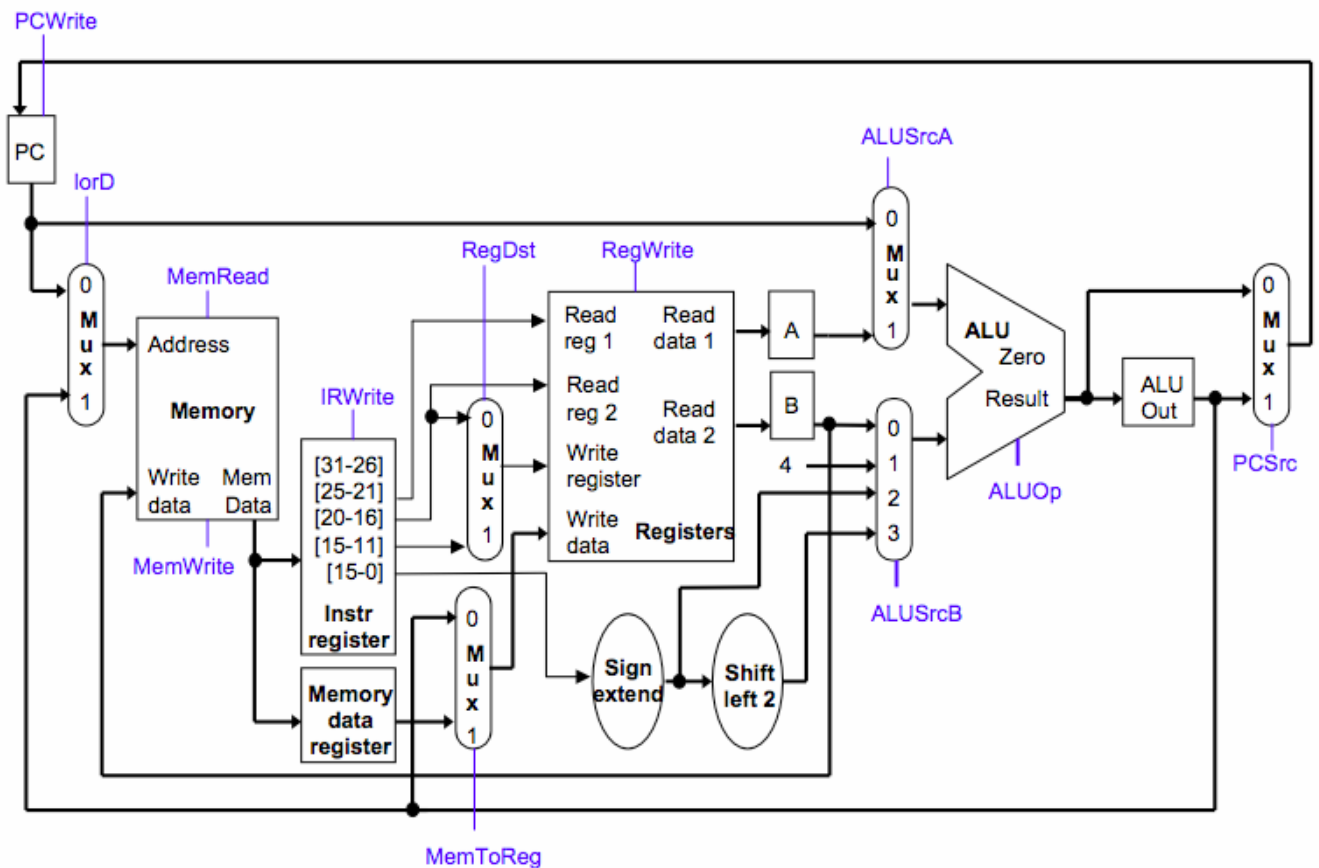
Consider extending the MIPS architecture with the instruction below, which loads two consecutive words of data from memory and stores them into two destination registers.

```
ld    rt, rd, rs # rt = Mem[rs]; rd = Mem[rs+ 4]
```

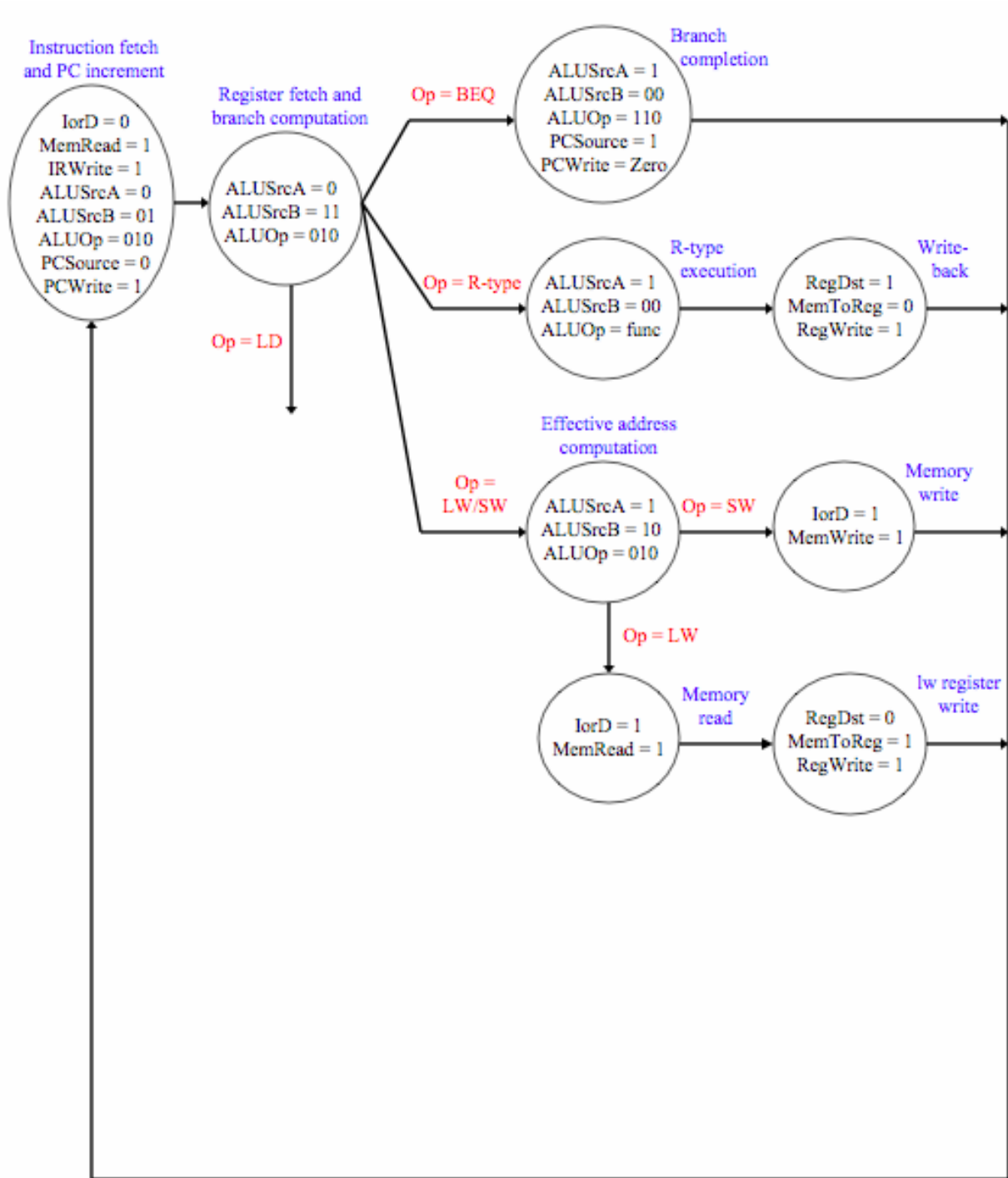
This will use the same format as R-type instructions, shown here for reference (**shamt** and **func** are not used).

Field	op	rs	rt	rd	shamt	func
Bits	31-26	25-21	20-16	15-11	10-6	5-0

a) A multicycle datapath similar to the one discussed in class appears below and the corresponding controller FSM appears on the next page. You are asked to make only a single modification to one of the functional units in the datapath (memory, register file, ALU, multiplexers, or intermediate registers) to support the instruction **ld**. Explain. (10 points)



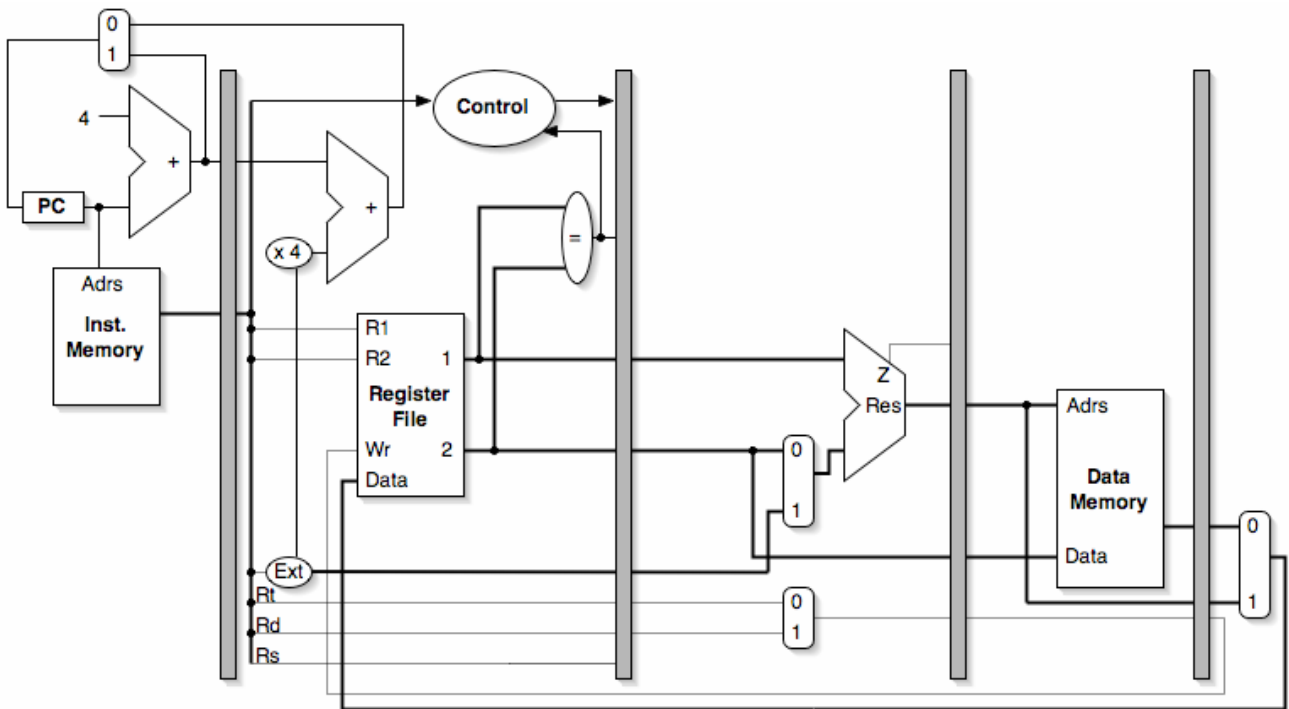
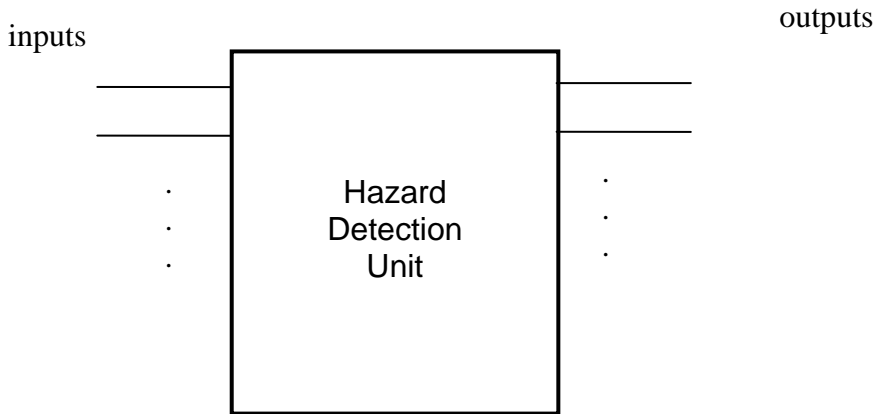
- b) Complete the FSM below according to the modification you have done to the datapath to provide appropriate control signal(s) for **ld**. You are allowed to add at most three states to the FSM. Control values not shown in each stage are assumed to be 0. For each state you add, show the values of the necessary control signals. Remember to account for any control signal(s) that you added or modified in the datapath. (6 points)



Problem 4: Pipelining (55 points, 4 parts)

This problem refers to the pipelined datapath shown below. All parts of this problem are independent. The datapath figure is repeated where necessary for convenience.

- a) The pipelined datapath below does not support forwarding. Hence, stalls must be inserted on **data hazards**. Ignore all other types of hazards. Modify the datapath, and add a hazard detection unit and show the necessary connections to that unit. Assume you can read and write from the register file in one clock cycle. To simplify things, label the signals in the datapath that must be connected to the hazard detection unit, and write them next to the box below. (10 points)



- b) Assume we execute the following MIPS code on the datapath after adding the hazard detection unit. Indicate all types of dependencies between the instructions below using arrows. (10 points)

```

LOOP: add $3, $3, 4
      lw  $5, -4($3)
      add $4, $4, 4
      lw  $6, -4($4)
      mul $7, $5, $6
      add $8, $8, $7
      bne $3, $9, LOOP

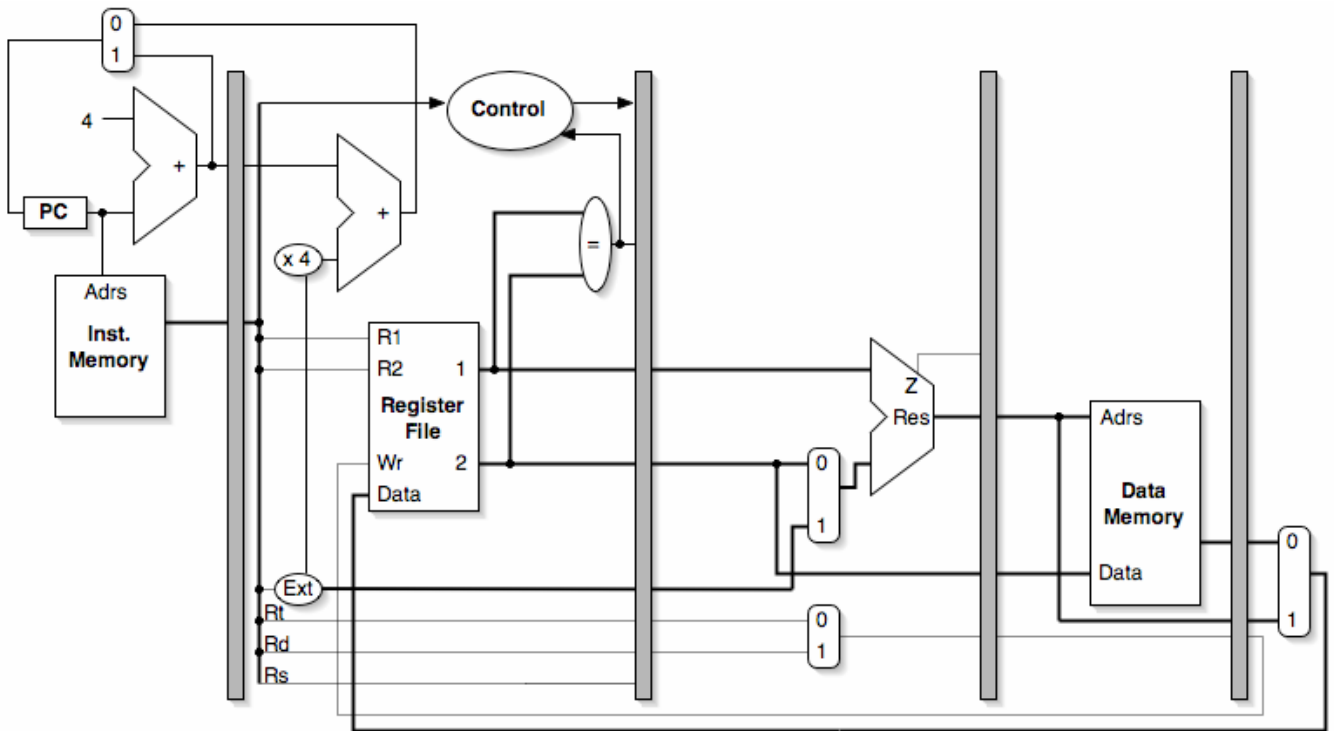
```

If the loop executes many times, we are mostly concerned with the performance of the “steady state” execution where the branch is taken. Assume the pipeline has a branch prediction unit that predicts “not-taken”. Using the grids below, indicate the pipeline stage each instruction is in for each cycle (stalls can be marked with an ‘S’). Then compute the steady state CPI of the loop by computing the time between the start of one iteration and the next. State any assumptions you make.

Inst	iter	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
Add	N	F	D	E	M	W																	
Lw	N		F																				
Add	N																						
Lw	n																						
Mul	n																						
Add	n																						
Bne	N																						
Add	N+1																						

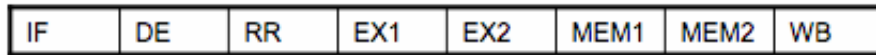
Steady state CPI = _____ (you can leave your answer as a fraction)

c) In this part we concentrate only on the possible pipeline hazards generated **when the producer is an R-format instruction and the consumer is any instruction**. Ignore all other types of hazards in this part. You are asked to modify the datapath to eliminate all such types of pipeline hazards. Write down forwarding equations where necessary using the convention discussed in class. Assume that the ALU in the EX stage and the register file in the ID stage have equal delay, while all muxes have negligible delays. Minimize the number of control signals you to the datapath. You need to show all control signals that are relevant to forwarding. To simplify the equations, give short labels to your forwarding signals. (15 points)



Forwarding equations:

d) Assume now the pipeline is elongated to **eight** stages: (20 points)

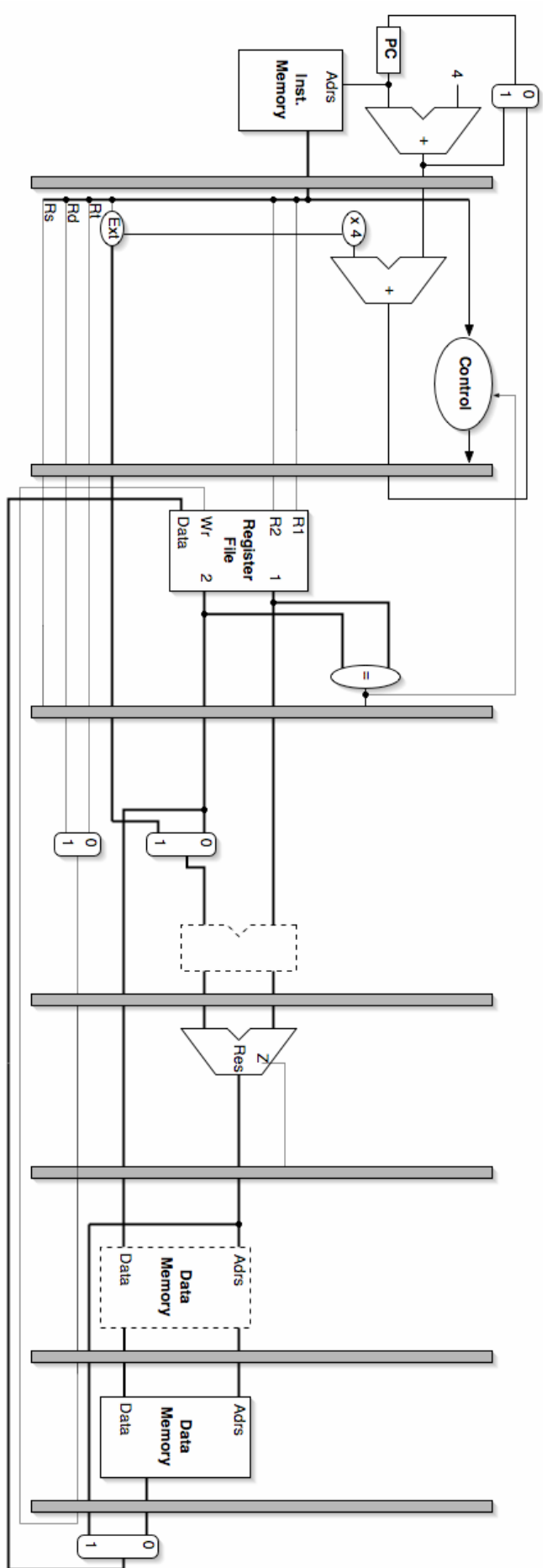


where decode is split into two stages Decode (DE) and Register Read (RR), and both the Execute and Memory Access stages each take **two stages to generate their result**. The modified datapath is shown on the next page.

i) Add forwarding logic to the new datapath to minimize the number of stalls on data hazards when the producer instruction is an R-format instruction. Again concentrate only on the possible pipeline hazards generated when the producer is an R-format instruction and the consumer is any instruction. Ignore all other types of hazards in this part. Show your modifications on the next page. (14 points)

ii) How many cycles would the machine need to stall (if any) in the following cases assuming the forwarding logic in (i) above has been added to the datapath? Explain. (6 points)

Case	# of stall cycles
add \$r1, \$r2, \$r3 add \$r4, \$r1, \$r2	
add \$r1, \$r2, \$r3 lw \$r1, 0(\$r2)	
lw \$r1, 0(\$r2) add \$r2, \$r1, \$r3	
lw \$r1, 0(\$r2) sw \$r1, 0(\$r3)	
add \$r1, \$r2, \$r3 beq \$r1, \$r2, LABEL	
lw \$r1, \$r2, \$r3 beq \$r1, \$r2, LABEL	

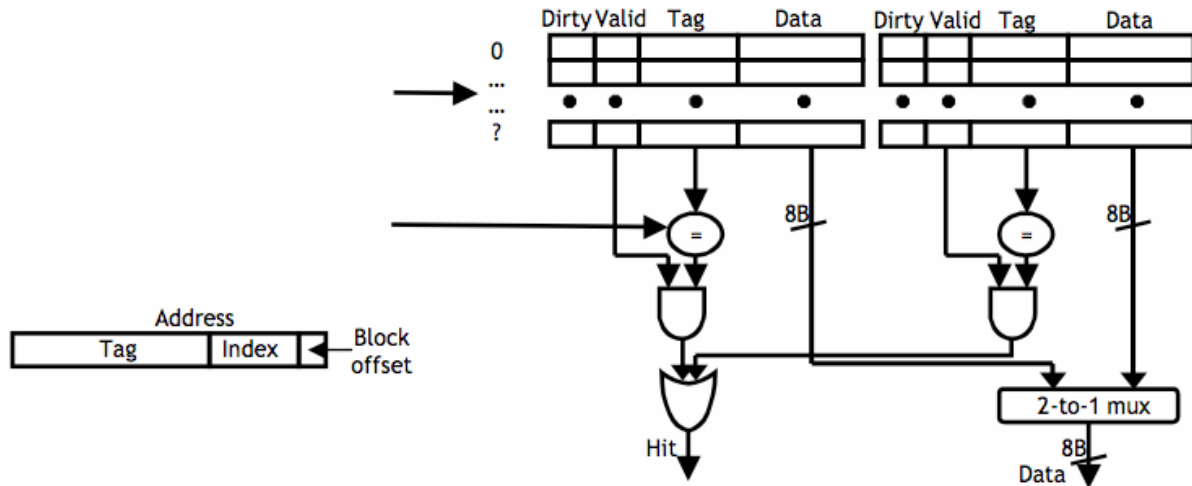


[End of problem 4]

Problem 5: Caches (52 points, 4 parts)

The parts of this problem are independent.

- a) The organization of a cache for a **byte-addressable** memory system is shown below. The cache holds 256 bytes of data in 8 byte blocks (recall that 8B = 8 bytes, where 8b = 8 bits). Implementing the cache requires 320 bytes of storage. (16 points)



Answer the following questions: (write “unknown” if the answer cannot be discerned from the information provided)

Associativity	
Number of Sets	
Write back or write-through	
Number of block offset bits	
Number of index bits	
Number of tag bits	
Number of address bits total	
Block replacement policy	

b) For this question, you are given a 16-byte cache (initially empty) and the sequence of memory accesses (byte loads) shown in the table. For each of the following cache configurations explain whether it can produce this sequence. If so, what is the result of the last access (shown as ???) for each of the matching sequences? **(16 points)**

1. 4-byte blocks, 2-way set-associative (e.g., 2 sets of 2 blocks each)

<u>addr</u>	<u>hit/miss</u>
0	(miss)
12	(miss)
1	(hit)
17	(miss)
15	(hit)
3	(miss)
7	(???)

2. 4-byte blocks, direct-mapped (4 blocks)

3. 8-byte blocks, direct-mapped (2 blocks)

4. 4-byte blocks, fully associative (4 blocks)

c) Assume a byte-addressable memory system of size 2^h bytes. Determine the size of the tag field (in bits) for a 2^m -way set associative cache with block size 2^b bytes, if the data size of cache is 2^σ bytes. **(8 points)**

Problem 6: URPH vs. AUB (17 points)

A top student from the UNIVERSITY OF REMOVING PIPELINE HAZARDS proposes the following pipelined MIPS datapath and claims that his/her design outperforms the best pipelined MIPS datapath discussed in our class. The student was so excited that he decides to start a company to implement the new design.

What was the student trying to improve? (7 points) Do you agree that this design is better than our design? (5 points) If so, explain why. If not, explain why not through an example, and send the student an application to transfer to AUB! (5 points)

(The pipeline control logic signals and the program-counter logic are not shown in the figure for simplicity.)

