

Computer Organization and Structure

Homework #5
Due: 2003/12/30

- How could we modify the following code to make use of a delayed branch slot?

```

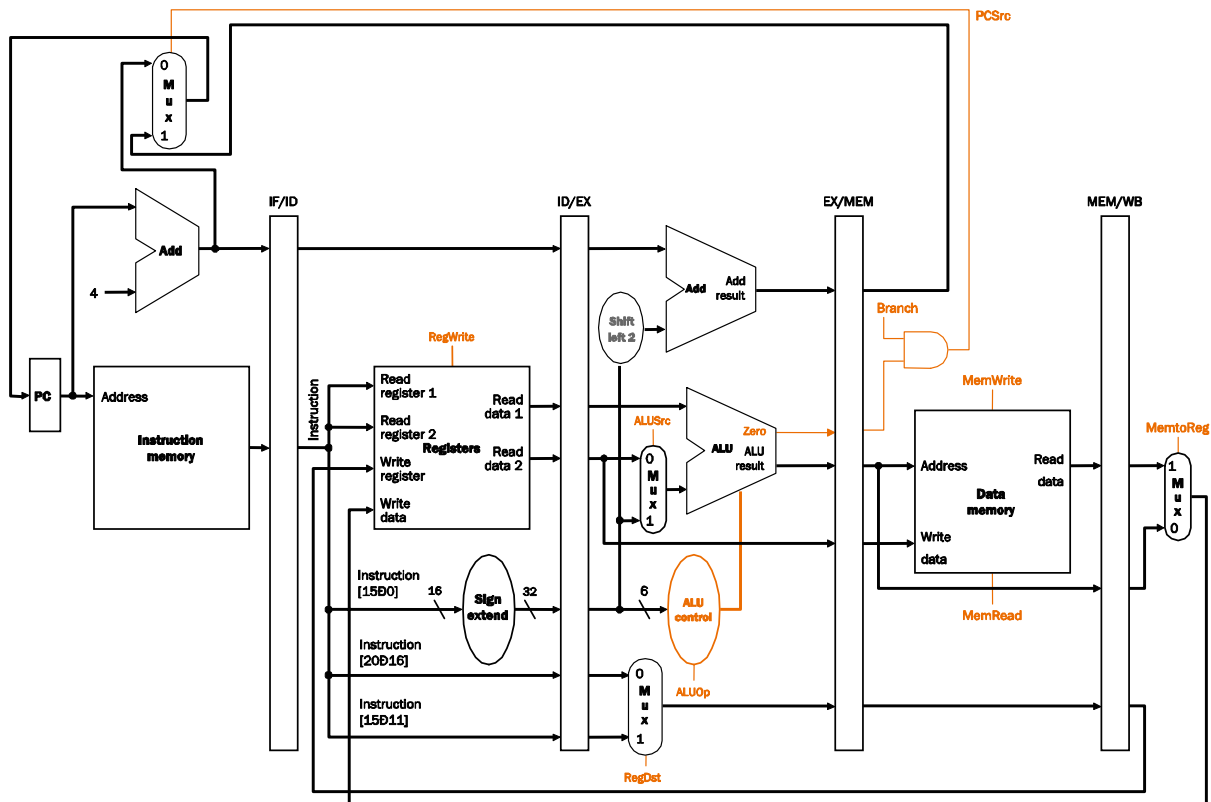
Loop:   lw    $2, 100($3)
        addi $3, $3, 4
        beq  $3, $4, Loop
    
```

- Identify all of the data dependencies in the following code. Which dependencies are data hazards that will be resolved via forwarding?

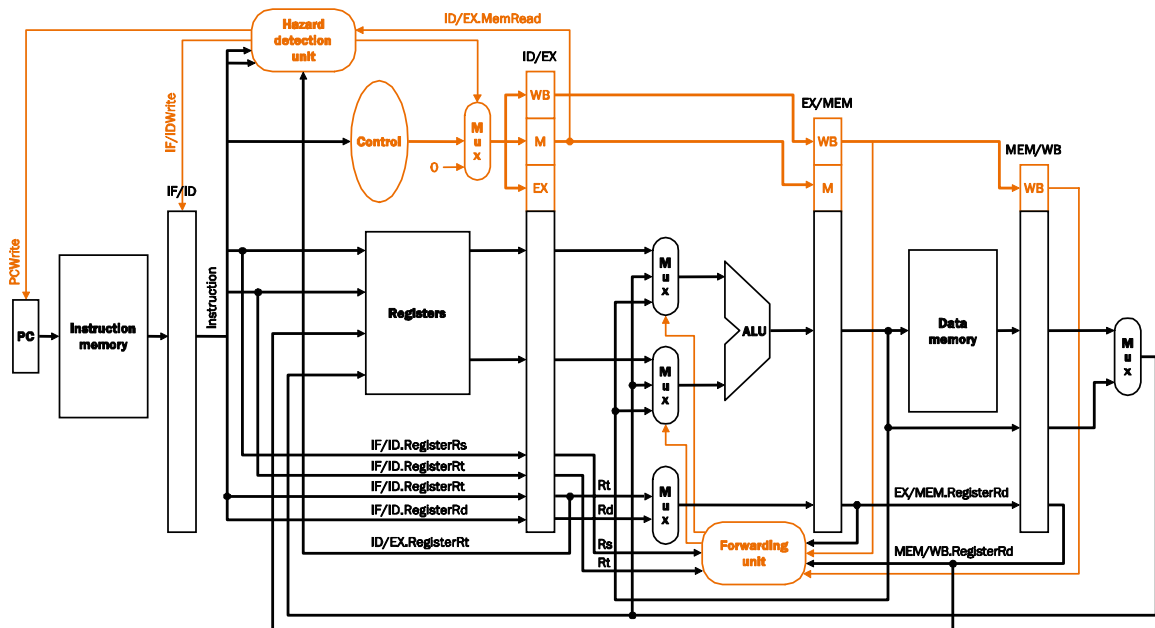
```

add    $2, $4, $5
add    $4, $2, $5
sw     $5, 100($2)
add    $3, $2, $4
    
```

- For each pipeline register in the following figure, label each portion of the pipeline register with the name of the value that is loaded into the register. Determine the length of each field in bits. For example, the IF/ID pipeline register contains two fields, one of which is an instruction field that is 32 bits wide.



4. Consider executing the following code on the following pipelined datapath:



```

add    $1, $2, $3
add    $4, $5, $6
add    $7, $8, $9
add    $10, $11, $12
add    $13, $14, $15
    
```

At the end of the fifth cycle of execution, which registers are being read and which register will be written? With regard to the above program, explain what the forwarding and the hazard detection units are doing during the fifth cycle. If any comparisons are being made, mention them.

5. The following code contains a “read after write” data hazard that is resolved by forwarding:

```

add    $2, $3, $4
add    $5, $2, $6
    
```

Consider the similar situation in which a memory read occurs after a memory write:

```

sw     $7, 100($2)
lw     $8, 100($2)
    
```

Write a paragraph describing how this situation differs from the one involving registers, and describe how the potential “read after write” problem is resolved.