

CSE 352 - HW 4

Figure 1 – 5 stage MIPS pipeline

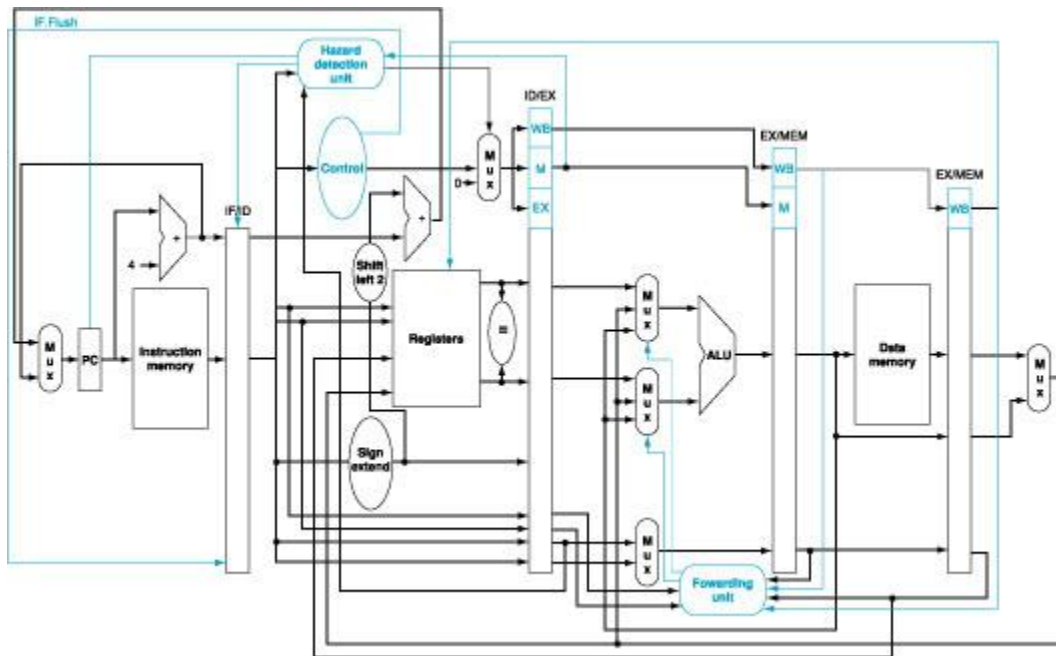
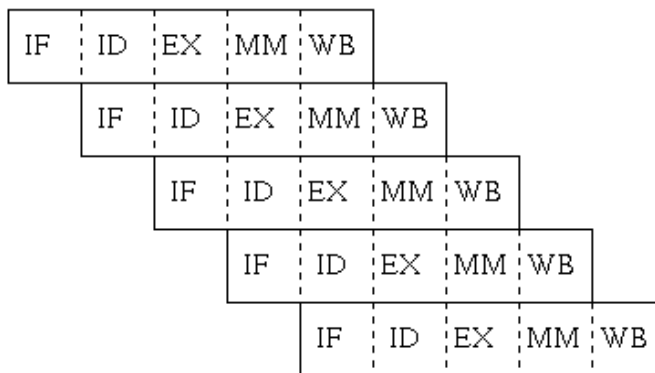


Figure 2 – The pipeline progresses over time

A 5-segment instruction pipeline



- IF: instruction fetching
- ID: instruction decoding
- EX: instruction execution
- MM: memory accessing
- WB: write back to registers

Q1:

Consider the following C code:

```
a = b + e;
```

```
c = b + f;
```

Here is the generated MIPS code for this segment, assuming all variables are in memory and are addressable as offsets from \$t0:

```
lw $t1, 0($t0)          # b is stored at 0 offset from $t0
```

```
lw $t2, 4($t0)          # Word size is 4 bytes in MIPS
```

```
add $t3, $t1, $t2
```

```
sw $t3, 12($t0)
```

```
lw $t4, 8($t0)
```

```
add $t5, $t1, $t4
```

```
sw $t5, 16($t0)
```

- Find the data hazards in the code segment.
- Draw the busses in use for forwarding in each hazard using copies of the attached diagram "[MIPS.jpg](#)". Label what is being forwarded.
- Reorder the instructions to avoid any pipeline stalls.
- Consider the new reordered sequence, draw a table to show where the instructions are in which stages in time for N cycles (pick N such that all the instructions complete).
- Consider the new reordered sequence, let $b = 1$, $e = 2$, $f = 3$. What is the value of ALU result when `add $t5, $t1, $t4` is fetched to the pipeline.
- Consider the new reordered sequence, which Select value is used (0 for top or 1 for bottom) for the mux in the Write Back stage when `sw $t3, 12($t0)` is fetched to the pipeline.

Q2:

Consider the following MIPS instruction

SBZ r1, r2, Z

SBZ - Subtract r1 and r2 and branch if the result is zero. The program counter is advanced by the immediate value Z if the branch is taken.

Is any more hardware needed to implement this? How many delay slots are required for this instruction for a pipelined CPU? Is there any way to decrease the number of delay slots for this instruction?

Now, consider the following instructions. How many clock cycles are needed to complete this program? Illustrate in the pipeline diagram what value is present in the forwarding buses during every cycle of execution. Use a new copy of the "[MIPS.jpg](#)" diagram for every cycle.

```
#MEM[20] = 10  
addi $t0, $0, 10  
lw $t1, 10($t0)  
sw $t0, 14($t0)  
sbz $t0, $t1, FOO  
sub $t2, $t1, $t0
```