Juan Vallejo

330 Homework

9 / 5 / 2014

4.8.1 Clock cycle time in non-pipelined processor is *1250ps*

Clock cycle time in pipelined processor *350ps*

4.8.2 Total latency for **LW** instruction is

Pipelined: 350ps x 5 = *1750*
Non-pipelined: *1250ps*

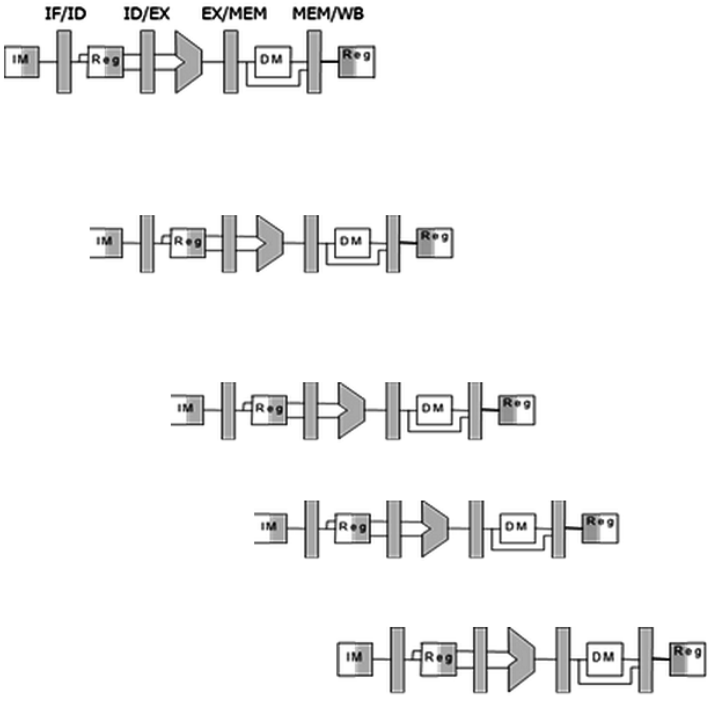
4.8.3 We would split **ID** instruction to have *175ps* latency.

New clock cycle time becomes *300ps*

2 Hazards in code are *register $2* is used before a value to it is assigned.

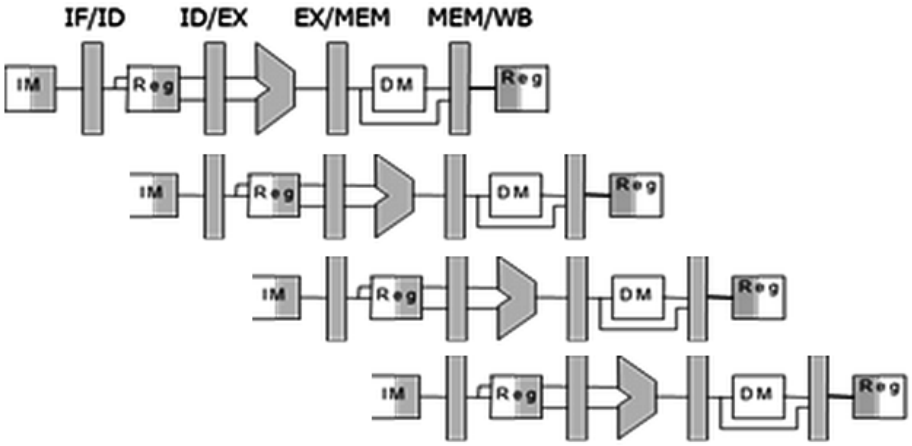
```
sub   $2,  $1, $3
nop
nop
and   $12, $2, $5
or    $13, $6, $2
add   $14, $2, $2
sw    $15, 100($2)
```

3 Path forwarding is used for the following instructions.



```
add     $3, $4, $6
sub     $5, $3, $2
lw      $7, 100($5)
add     $8, $7, $2
```

4 Path forwarding is used for the following instructions.




```
lw      $4, 100($2)
sub     $6, $4, $3
add     $2, $3, $5
```

5. Compare the performance for single-cycle, multi-cycle, and pipeline control using the SPECint2000 instruction mix

- 25% loads
- 10% stores
- 11% branches
- 2% jumps
- 52% ALU

The number of clock cycles for each instruction class:

- Loads: 5

- Stores: 4
- Branches: 3
- Jumps: 3
- ALU: 4

Start with performance of single-cycle machine:

- 200 ps for memory access
- 100 ps for ALU operation
- 50 ps for register file read or write

(a) What is the clock cycle time for single-cycle datapath?

(b) What is the average CPI for the multiple cycle design?

(c) What is the average CPI for the pipeline design?

(Loads, stores, and ALU take 1 clock cycle. Branches take 1 clock cycles when predicted correctly and 2 when not. Jump CPI = 2)

Note that the long cycle time of memory is a performance bottleneck for pipelined and multicycle design.

(d) Find the average instruction time for single-cycle, multicycle and pipelined designs.