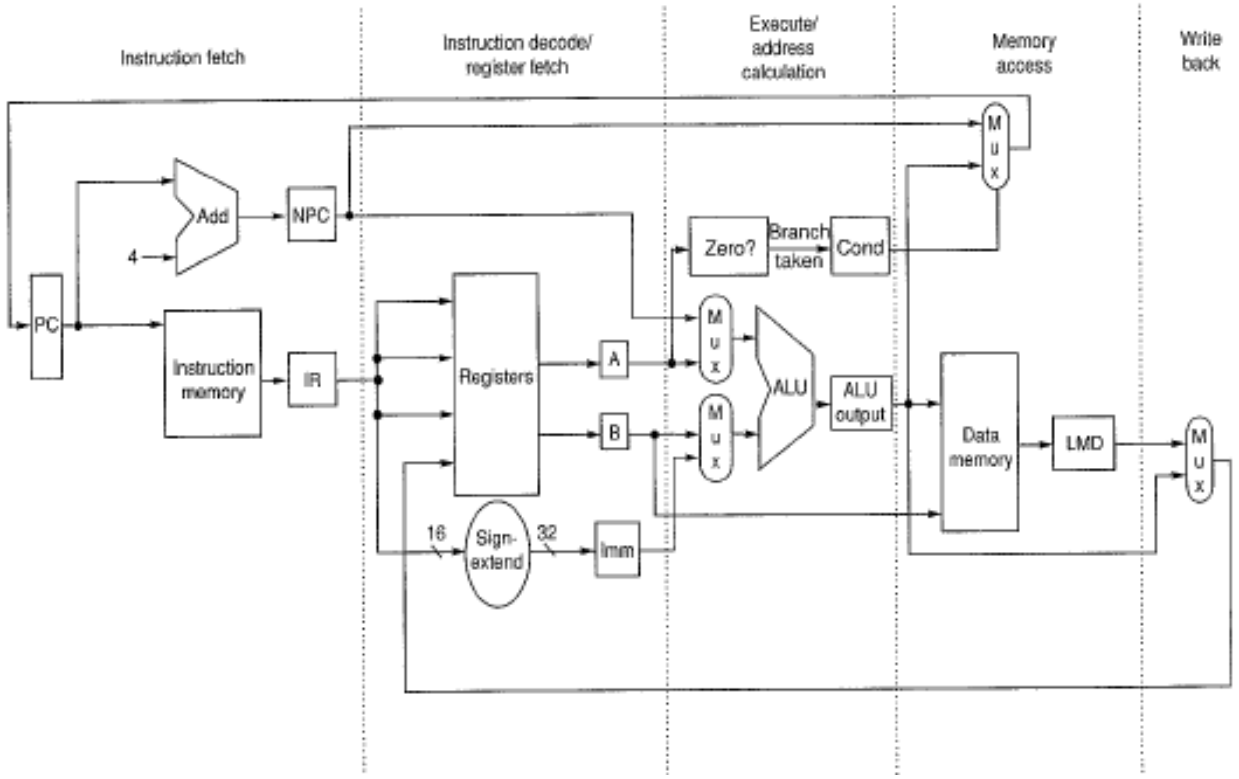
25. MIPS Pipeline

MIPS Non-pipelined

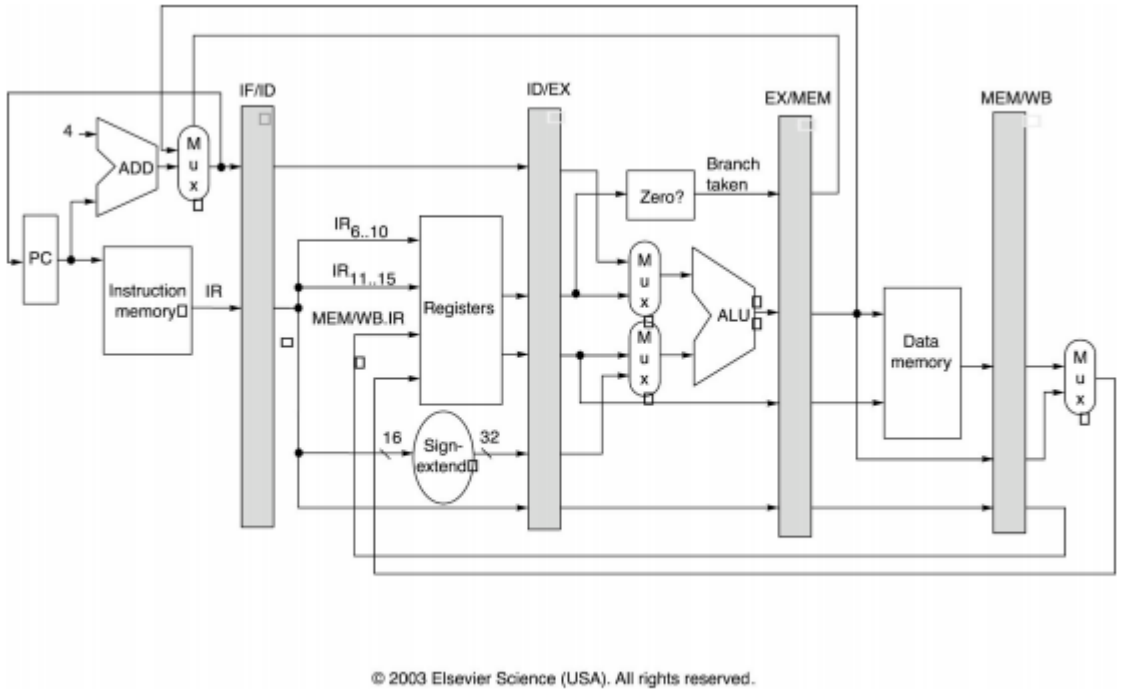


MIPS Pipelining

- Let's consider the previous non-pipelined processor.

- Why is this processor non-pipelined? That is, if each stage IF, ID/RF, EX, MEM, WB becomes a pipe stage, what cannot be done in parallel?

MIPS Pipelining



© 2003 Elsevier Science (USA). All rights reserved.

Figure 1: Generic MIPS 5-stage pipeline

MIPS Pipelining

- A set of registers (IF/ID, ID/EX, EX/MEM, MEM/WB) is placed between each pipe stage
  1. used to save instruction state as it propagates through the pipe
  2. instructions are only active in one pipe stage at a time
  3. inter-stage registers are master-slave D flip-flops
     a) master receives new data from previous stage
     b) slave provides data to the next stage
- We can also think of the PC as a register, so each pipeline stage has a pipeline register leading to the stage
- The register file is written in the first half of the clock cycle and read in the second half of the clock cycle

Practice

- Given the following MIPS program, show the pipeline using IF,ID,EX,MEM,WB and S for stall. Assume no forwarding.

```
.text
main:
  dadd r1,r2,r3
  dsub r4,r1,r5
  and r6,r1,r7
  or r8,r1,r9
  xor r10,r1,r11
  halt
```

Instr	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14
1														
2														
3														
4														
5														
6														

Practice Answer

- Let’s execute on the WinMIPS64

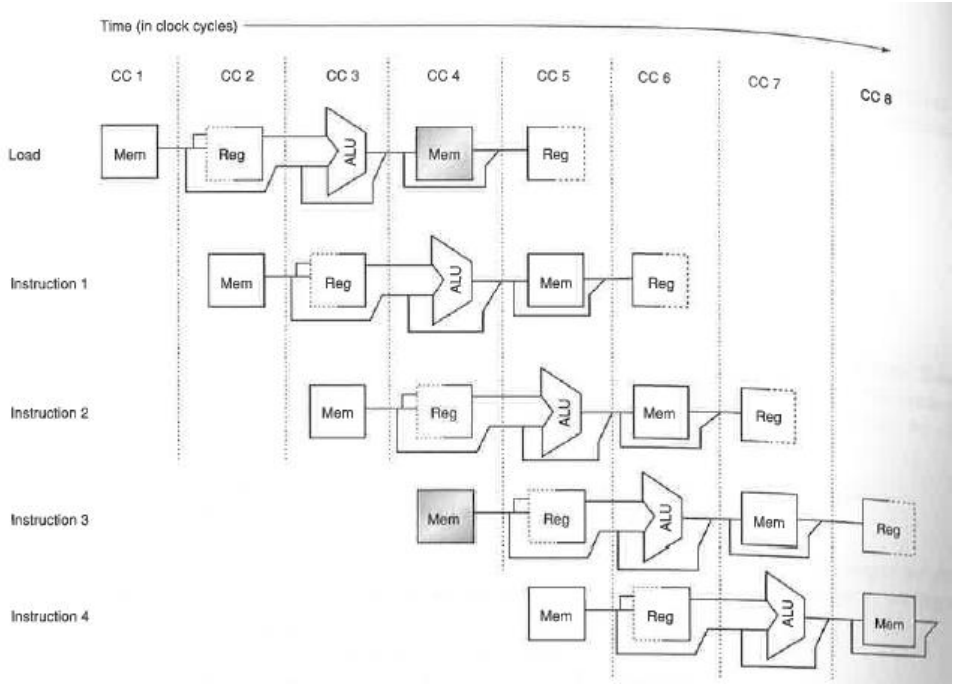
- What’s different from your prediction?

Structural Hazards

- In pipelining, the overlapped instruction execution of instructions requires:
  - pipelining of functional units
  - duplication of resources
- A structural hazard is a combination of instructions that cannot be accommodated because of resource conflicts
- A stall (or pipeline bubble or bubble) is required to resolve the hazard

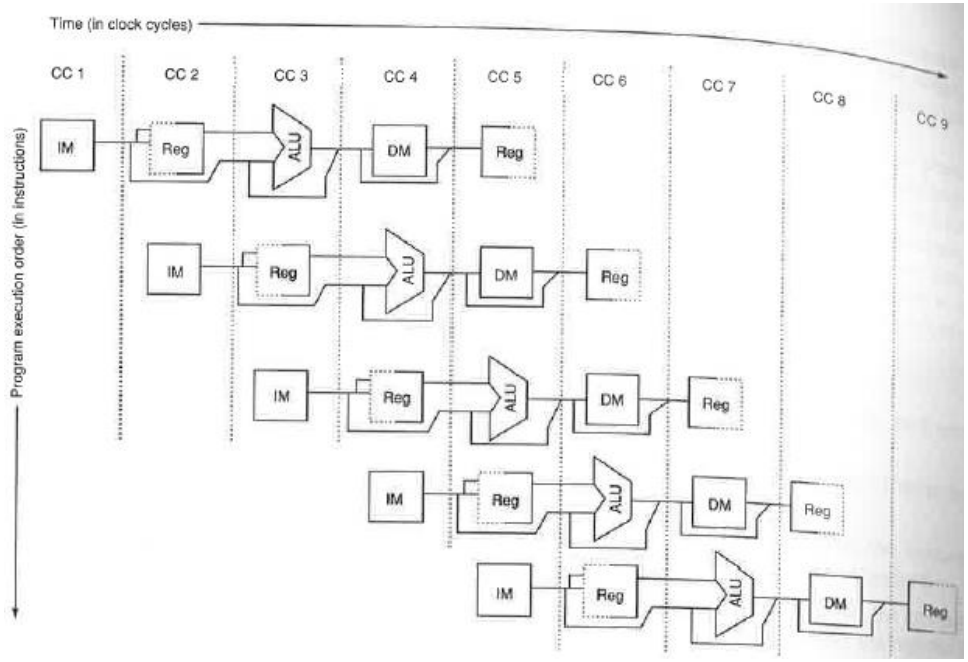
Structural Hazards

- Processor with one memory port



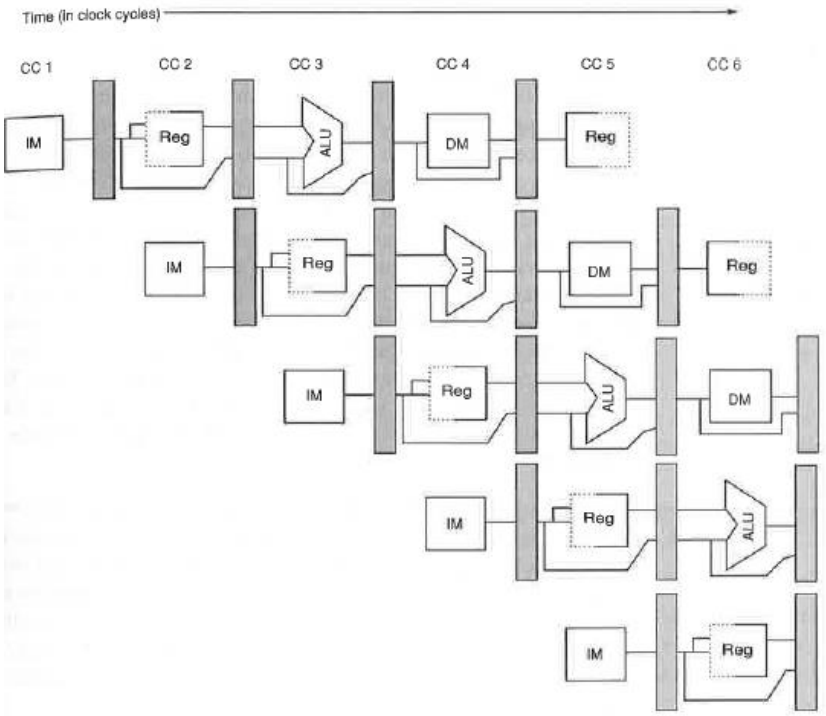
Structural Hazards

- Possible Solution: Separate Instruction & Data Memory



Pipeline Registers

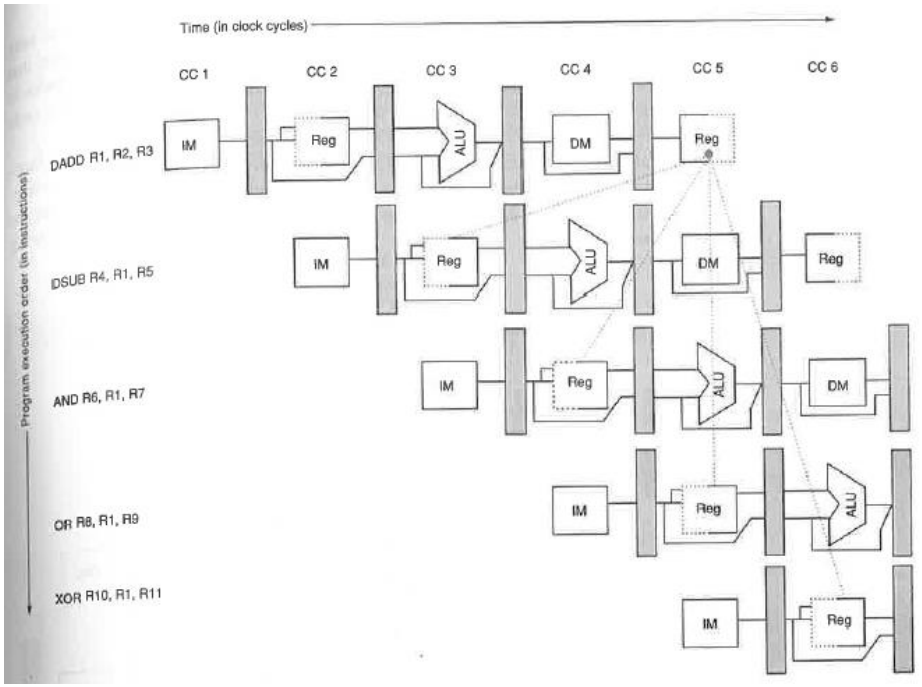
- Remember, we need separate pipeline registers between successive stages



Data Hazards

- Notice that r1 is needed by each subsequent instruction after the dadd

```
.text
main:
  dadd r1,r2,r3
  dsub r4,r1,r5
  and r6,r1,r7
  or r8,r1,r9
  xor r10,r1,r11
  halt
```



Data Hazards

- Minimizing Data Hazard Stalls by Forwarding

- Question: Going back to the previous slide, when is the result from DADD really needed in DSUB?

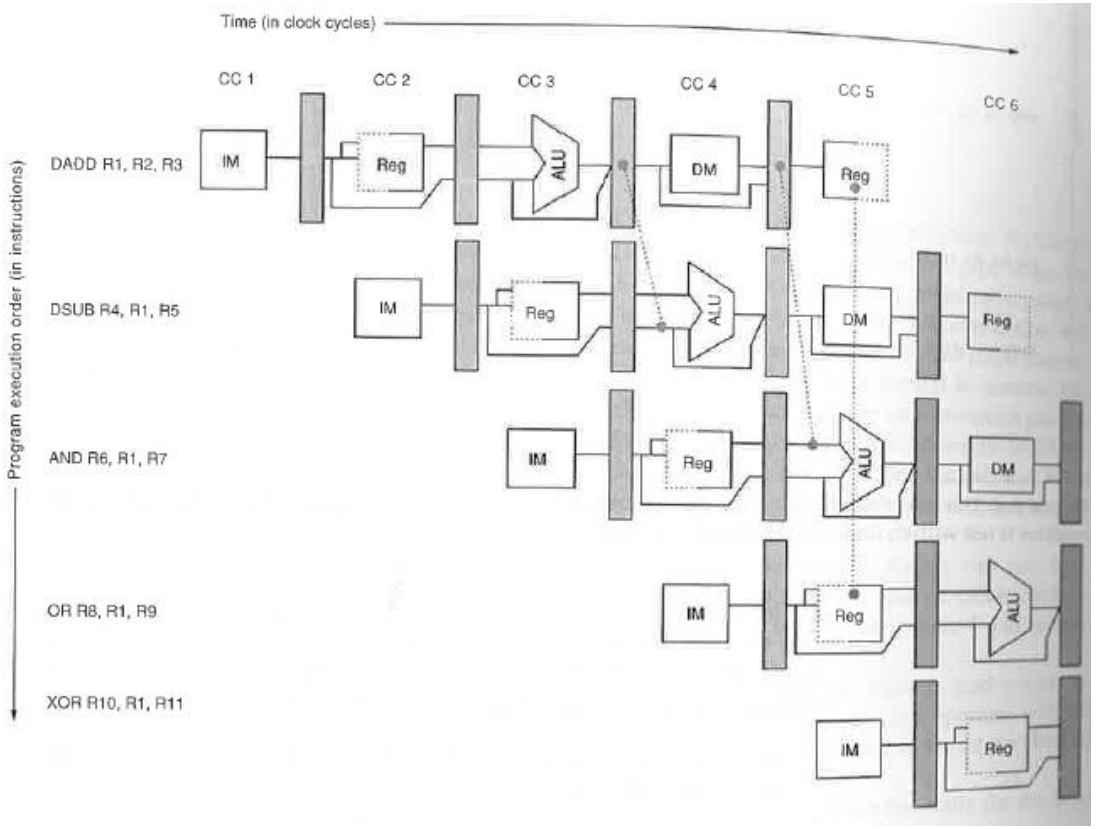
Data Hazards

- Possible Solution

1. The ALU result from EX/MEM and MEM/WB pipeline registers is always fed back to the ALU inputs
2. The forwarding hardware detects that ALU operation has written the register, thus control logic selects forwarded result

Data Hazards

- Possible Solution



Practice

- Given the following MIPS program, show the pipeline using IF,ID,EX,MEM,WB and S for stall. Forward the data as much as possible.

```
.text
main:
  dadd r1,r2,r3
  dsub r4,r1,r5
  and r6,r1,r7
  or r8,r1,r9
  xor r10,r1,r11
  halt
```

Instr	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14
1														
2														
3														
4														
5														
6														