

# Note W2 D2

Jingqi Xiao

## Review: 5 Steps of MIPS Datapath

- CC: input instructions, output signal
- Type of instructions
  - arith/logic: +, -, AND
  - data transfer: MOV, LD, ST
  - control: JMP, CALL, RET
- Signed extension: if number of 32-bit is added by 16-bit, then 16-bit should be extended into 32-bit and what is filled in the more bits depends on the original highest bit
- Data Stationary Control: divide pipelines in order to avoid next instruction affecting the last.

## New Topic: Hazard in Pipeline & Solution

Hazards prevent next instruction from executing during its designated clock cycle

### Structural hazards

HW cannot support this combination of instructions (single person to fold and put clothes away)

for example:

	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7	Cycle 8
Load	Ifetch	Reg	ALU	<b>DMem</b>	Reg			
Instr 1		Ifetch	Reg	ALU	DMem	Reg		
Instr 2			Ifetch	Reg	ALU	DMem	Reg	
Stall				<b>Ifetch</b>	Reg	ALU	DMem	Reg

In this table, DMem in Load and Ifetch in Stall may be executed at the same time.

### solution

use bubble

### Data hazards

Instruction depends on result of prior instruction still in the pipeline (missing sock)

- Read after Write (RAW)

the data should be read after written according to the order of instructions. But due to pipeline, the data may be read before written and makes result incorrect.

for example:

```
add r1, r2, r3
sub r4, r1, r3
```

- Write after Read (WAR)

The instruction writes before the instruction reads

for example:

```
sub r4, r1, r3
add r1, r2, r3
```

- Write after Write (WAW)

Instruction writes operand before instruction writes it.

for example:

```
sub r1, r4, r3
add r1, r2, r3
```

WAR and WAW can't happen in MIPS 5 stage pipeline because

- all instructions take 5 stages
- reads are always in stage 2
- writes are always in stage 5

However, if the instruction is executed out of order, this will cause a problem.

## solution

- renaming
- forwarding

## Control hazards

Caused by delay between the fetching of instructions and decisions about changes in control flow (branches and jumps)