March 3, 2004

Stephen Campbell
Michael Brown

Pipeline Hazards

Data Hazards

- Arise when an instruction depends on the result of a previous instruction in a way that is exposed by the overlapping of instructions in the pipeline.

Control Hazards

- Arise from the pipelining of branches and other instructions that changes the PC.

Structural Hazards

- Arise from resource conflicts when the hardware cannot support all possible combinations of instructions simultaneously in overlapped execution.

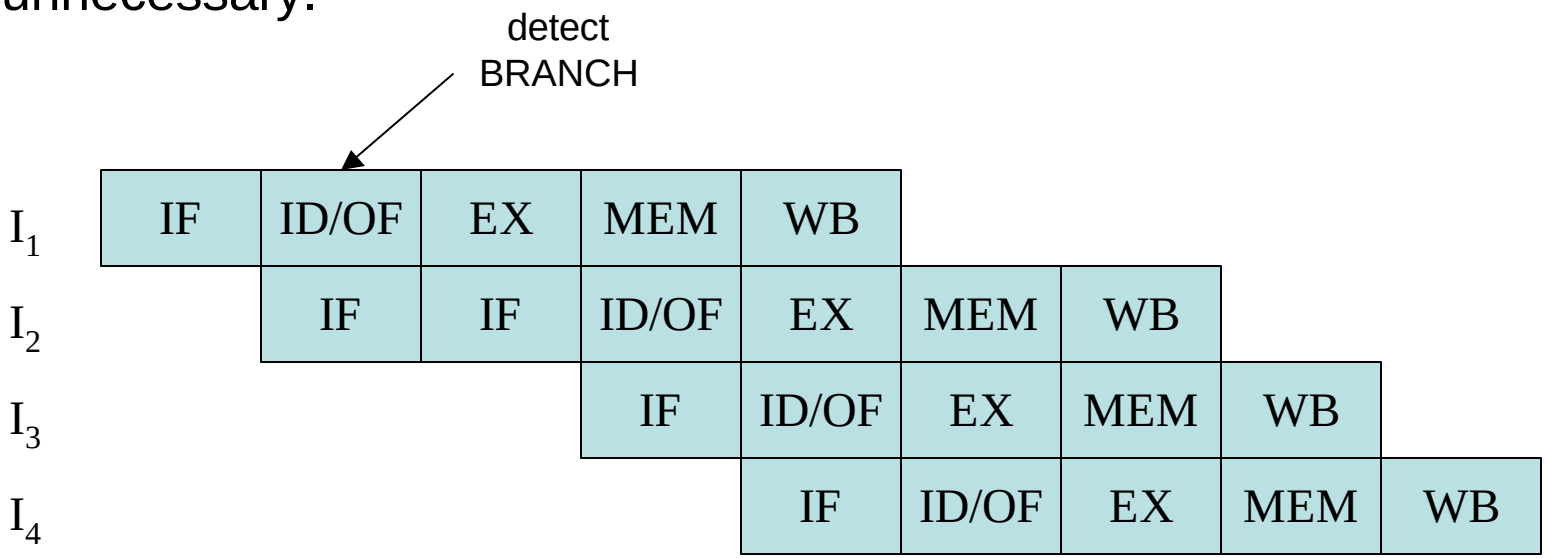
Data Hazards

Data hazards arise when an instruction depends on the result of a previous instruction in a way that is exposed by the overlapping of instructions in the pipeline.

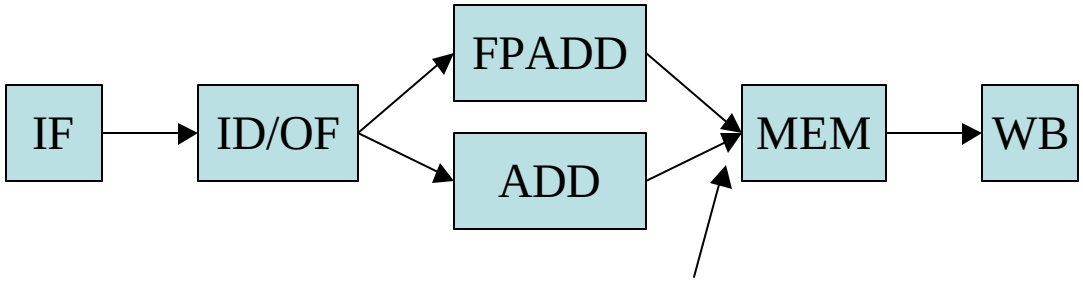
Control Hazards

Control hazards arise from the pipelining of branches, and other instructions that change the PC.

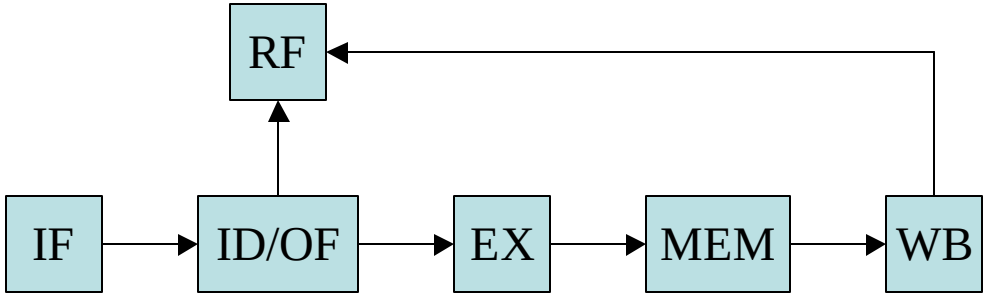
In this example, the branch occurs at the ID/OF in I_1. The easiest way to solve this would be to redo the IF of the instruction as soon as we detect the branch. The first IF basically works like a stall. If the branch is not taken, however, the repetition of IF becomes unnecessary.



Structural Hazards



There is a conflict, because a result from both FPADD and ADD are being written to Memory at the same time. This can be resolved by delaying one of the results.

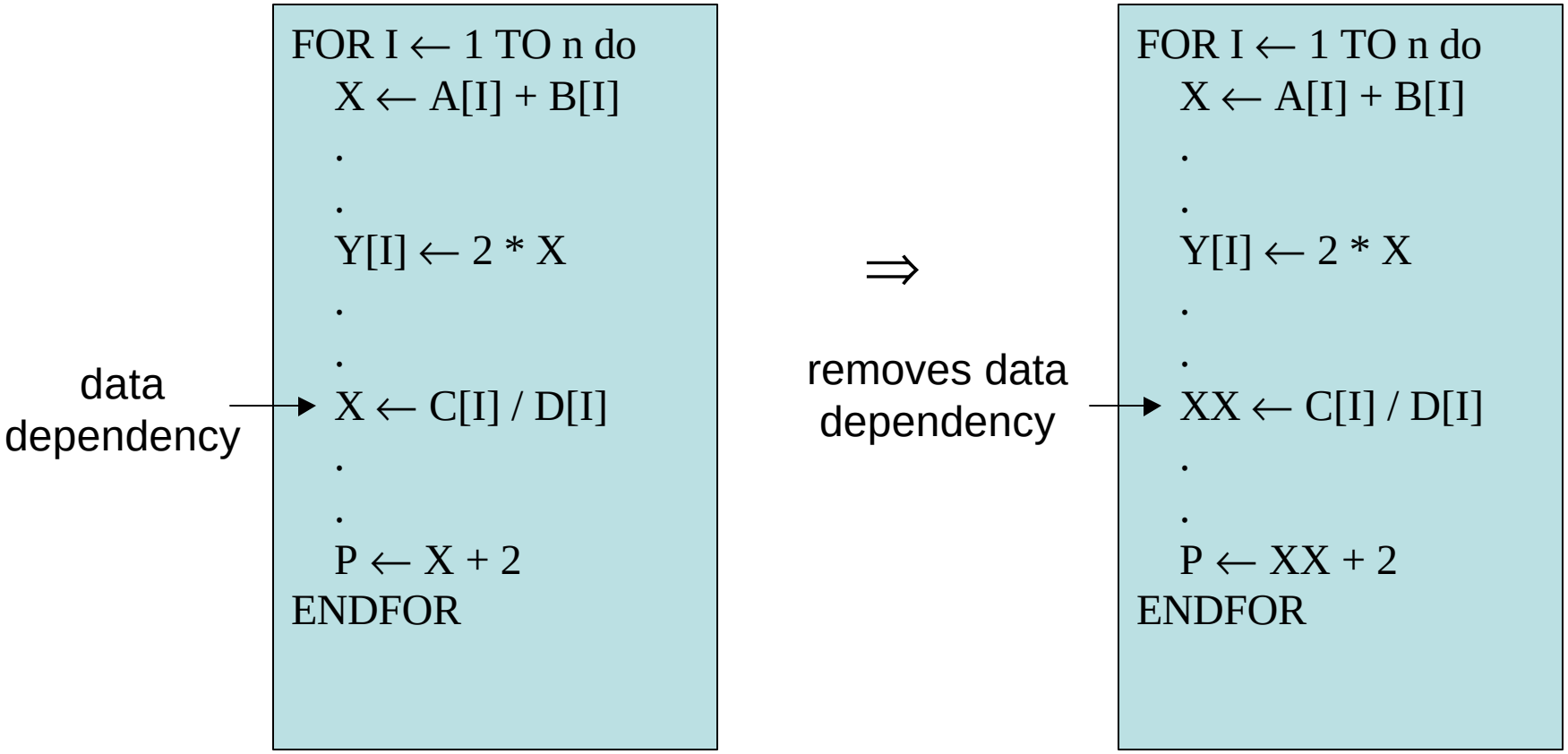


Structural Hazards

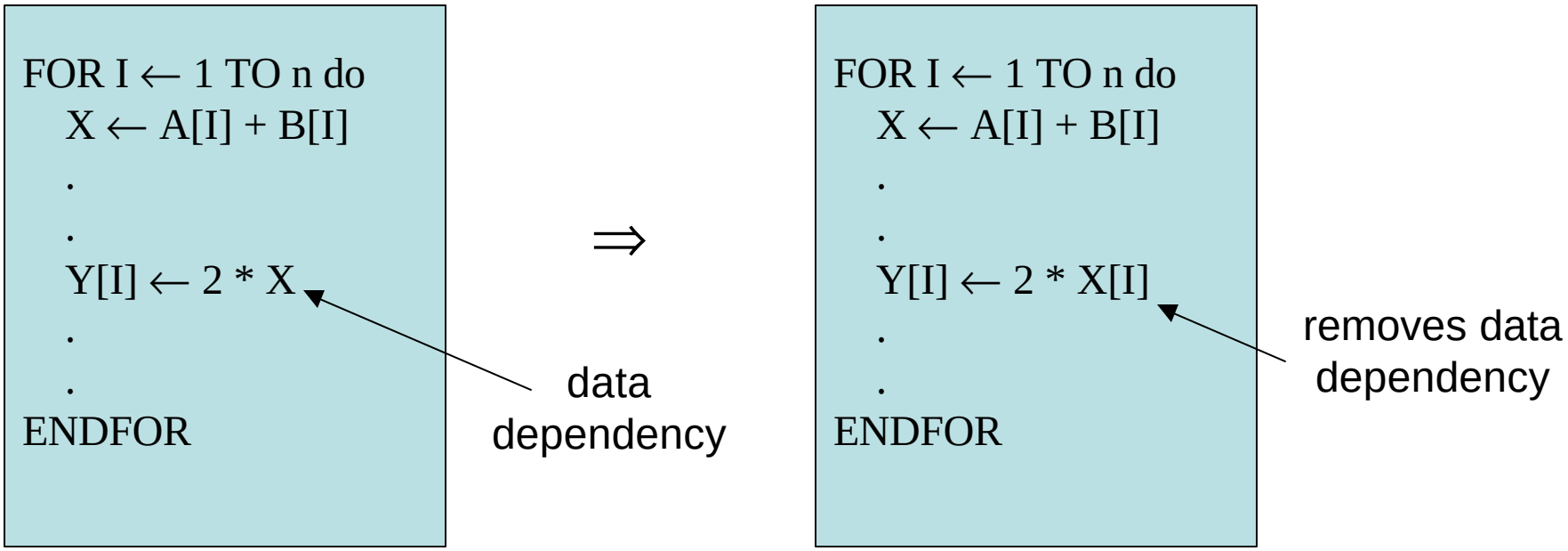
Structural hazards arise from resource conflicts when the hardware cannot support all possible combinations of instructions simultaneously in overlapped execution.

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9
LOAD	MEM	REG	EX	MEM	REG				
I_1		MEM	REG	EX	MEM	REG			
I_2			MEM	REG	EX	MEM	REG		
I_3				MEM	REG	EX	MEM	REG	
I_4					MEM	REG	EX	MEM	REG

Program Transformation



Scaler Expansion



Loop Unrolling

```
FOR I ← 1 TO n do
    X[I] ← A[I] * B[I]
ENDFOR
```

$\Rightarrow$

$$
\begin{aligned}
&X[1] \leftarrow A[1] * B[1] \\
&X[2] \leftarrow A[2] * B[2] \\
&\vdots \\
&X[n] \leftarrow A[n] * B[n]
\end{aligned}
$$

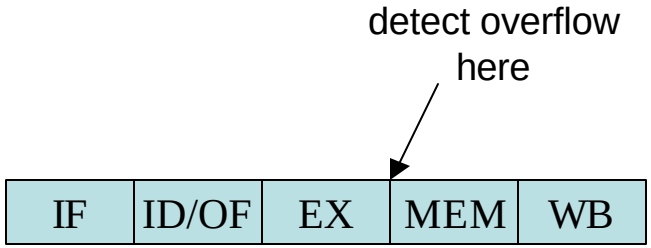
Loop Fusion or Jamming

```
FOR I ← 1 TO n do
    X[I] ← Y[I] * Z[I]
ENDFOR
FOR I ← 1 TO n do
    M[I] ← P[I] + X[I]
ENDFOR
```

⇒

a)
```
FOR I ← 1 TO n do
    X[I] ← Y[I] * Z[I]
    M[I] ← P[I] + X[I]
ENDFOR
```
b)
```
FOR I ← 1 TO n do
    M[I] ← P[I] + Y[I] * Z[I]
ENDFOR
```

Overflow

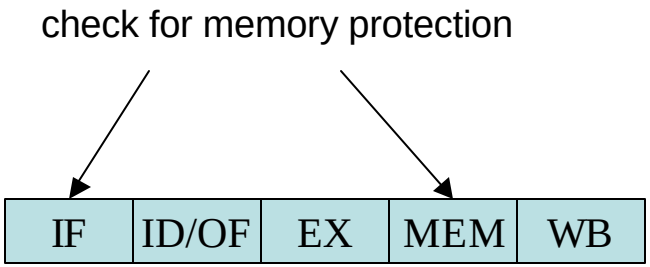


An overflow occurs when an arithmetic operation is performed, and the result is too large for the machine to hold.

If there is an overflow, the program is aborted and a new program is loaded by the IH.

The pipeline must be flushed, and then start again with IF of the first instruction of the IH.

Memory Protection

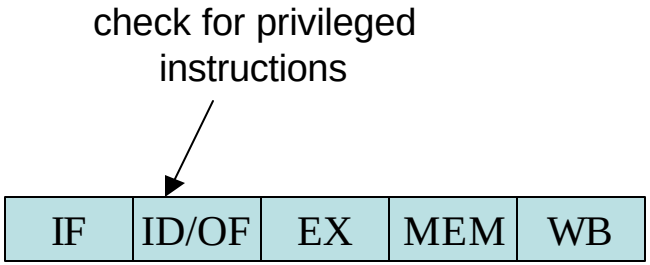


A memory protection violation occurs when Memory is trying to be accessed by more than one process at once.

Memory protection must be checked in both IF and MEM, before memory is accessed.

If a memory protection violation occurs, first the pipeline must be flushed, and then the PC must be changed.

Privileged Instructions



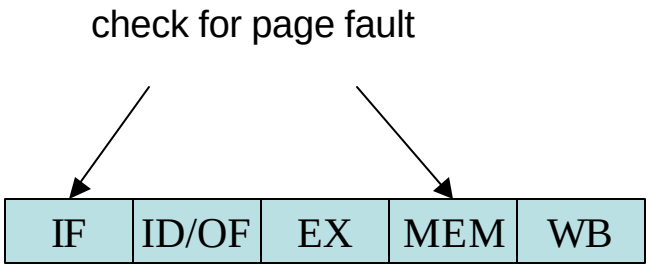
A privileged instruction violation occurs when the user tries to execute a privileged instruction.

Privileged instructions must be checked in ID

If a privileged instruction is trying to be executed by the user program, the program is aborted and a new program is loaded by the IH.

The pipeline must be flushed, and then start again with IF of the first instruction of the IH.

Page Fault

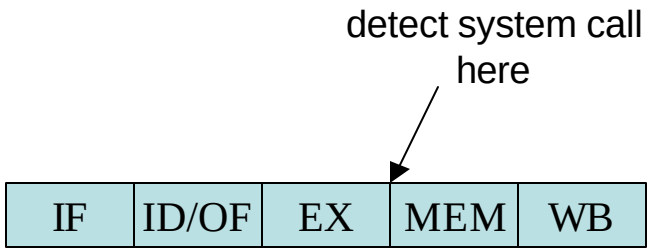


A page fault occurs when the program tries to access a page that is not currently in memory.

The program must check for page faults in both IF and MEM, before memory is accessed.

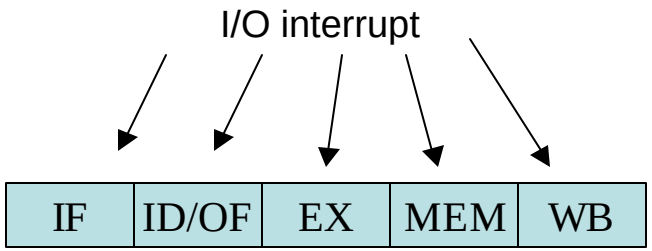
If a page fault occurs, the remainder of the pipeline is executed, and control is given to the OS.

System Call



If a system call is detected after the execute phase, MEM and WB are allowed to finish before control is passed to the RTL. Once in the RTL, the SVC flag is set to 1, and the program goes to the IH.

I/O Interrupt



An I/O Interrupt can occur at any time, but is detected after the EX phase.

If an I/O Interrupt occurs, execute the rest of the pipeline, and give control to the OS.