Topic V2C

Exceptions and Speculation in Pipelined Execution

Reading: (Section 4.9)

Exceptions in a Pipeline

Another form of control hazard

Example:

The following instruction:

lw x1, 0(zero)

Produces a segmentation fault

What should the processor do?

`lw x1, 0(zero)`

Prevent **`x1`** from being clobbered

Complete previous instructions

Flush **`lw`** and subsequent instructions

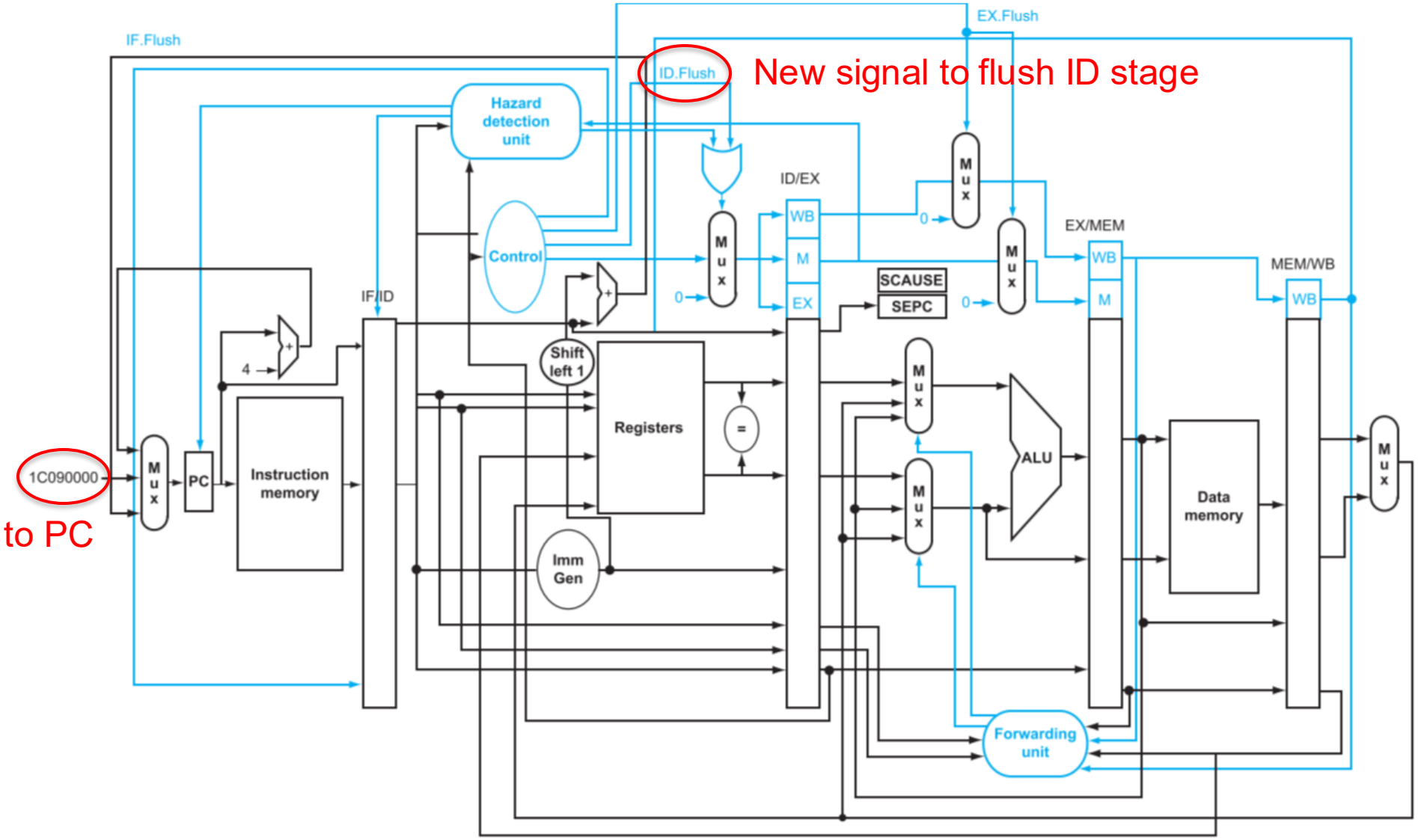
Set **`UCAUSE`** and **`UEPC`** register values

Transfer control to exception handler

Similar actions to misspredicted branch

- Use much of the same hardware

Pipeline with Exceptions



Re-Startable Exceptions

- Re-startable exceptions
  - Pipeline can flush the instruction
  - Handler executes, then returns to the instruction
    - Re-fetched and executed from scratch
- PC saved in UEPC register
  - Identifies causing instruction

Exception Example

```
    40      sub   x11, x2, x4
    44      and   x12, x2, x5
    48      or    x13, x2, x6
    4C      add   x1,  x2, x1
    50      slt   x15, x6, x7
    54      lw    x16, 50(x7)
    …
Handler:
    1C090000    sd    x26, 1000(x0)
    1C090004    sd    x27, 1008(x0)
    …
```

Assume that this add causes an exception

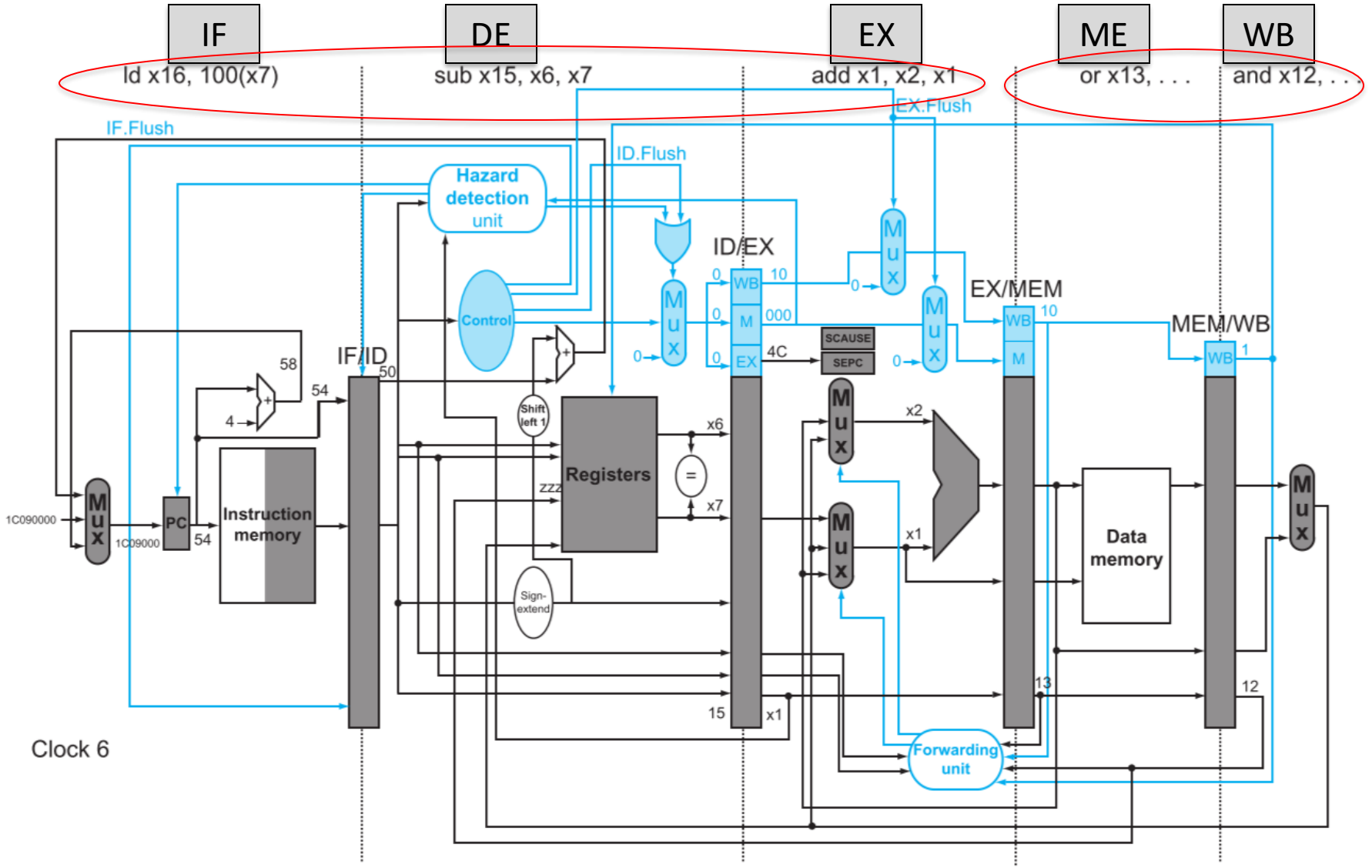
```
40      sub   x11, x2, x4
44      and   x12, x2, x5
48      or    x13, x2, x6
4C      add   x1,  x2, x1
50      slt   x15, x6, x7
54      lw    x16, 50(x7)
```

Exception Example

```
1C090000 sw   x26, 1000(x0)
1C090004 sw   x27, 1008(x0)
```

These three must become nop.

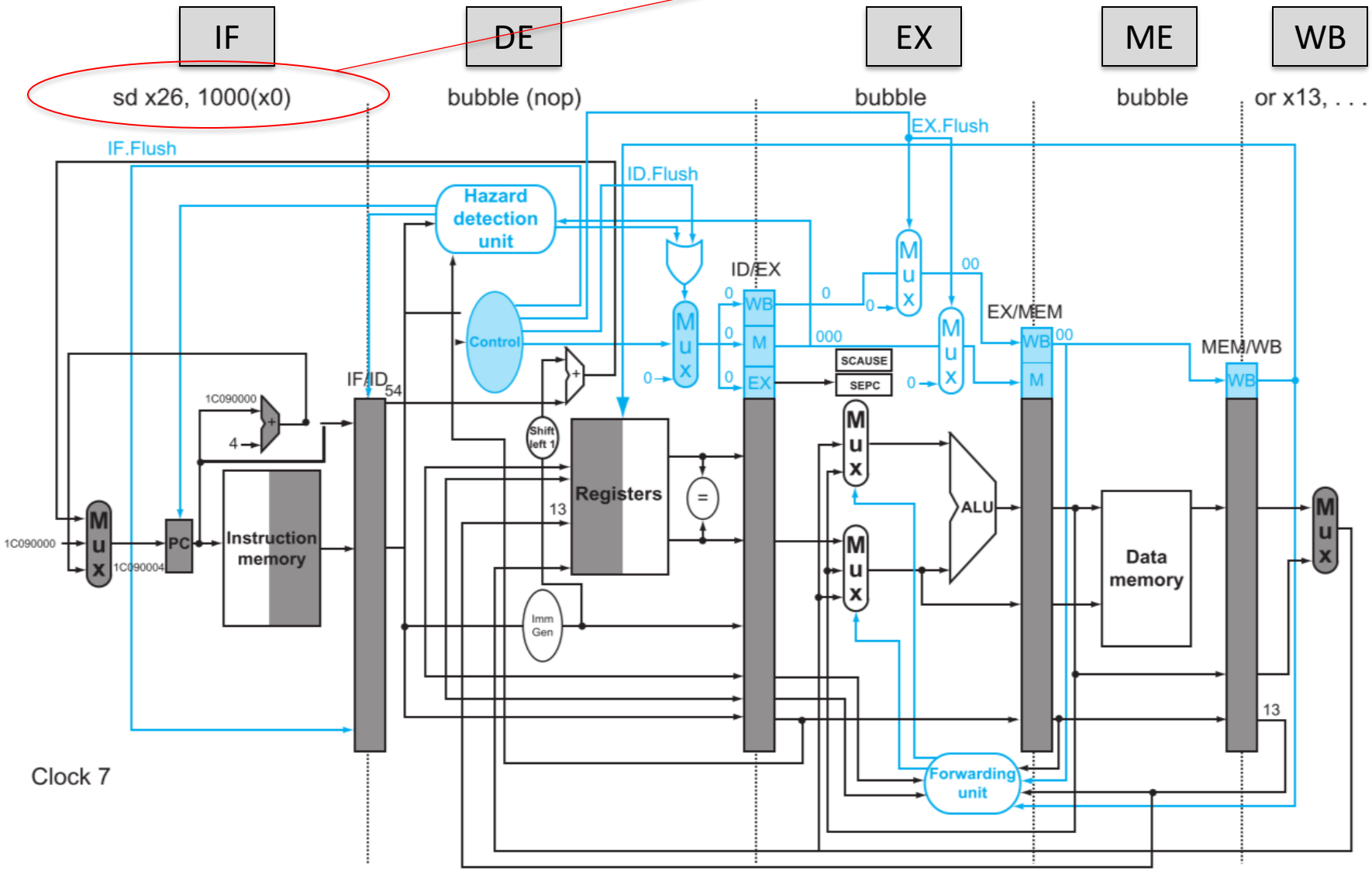
These instructions must finish execution



```
40        sub   x11, x2, x4
44        and   x12, x2, x5
48        or    x13, x2, x6
4C        add   x1,  x2, x1
50        slt   x15, x6, x7
54        lw    x16, 50(x7)
```

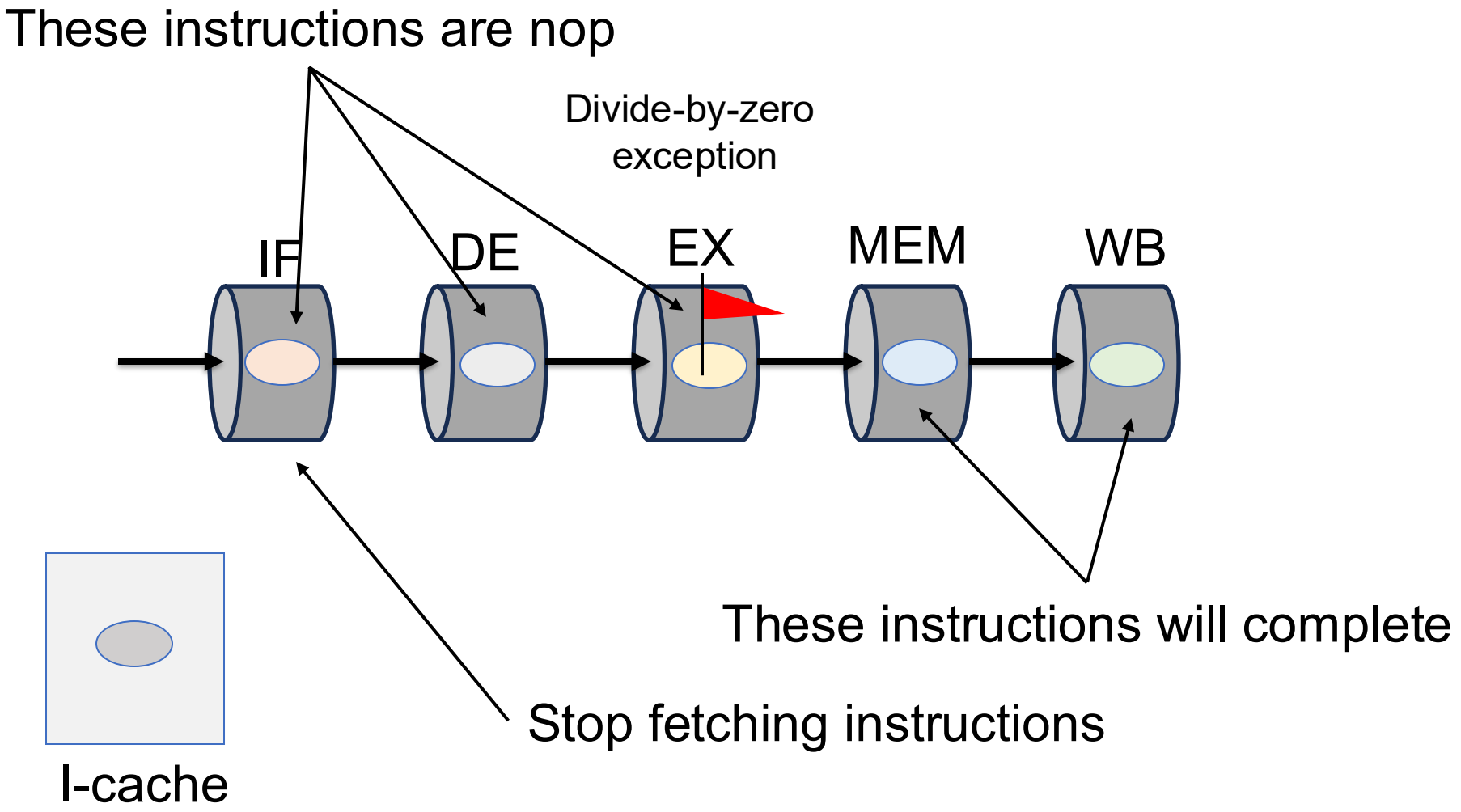
Exception Example

```
1C090000 sw    x26, 1000(x0)
1C090004 sw    x27, 1008(x0)
```

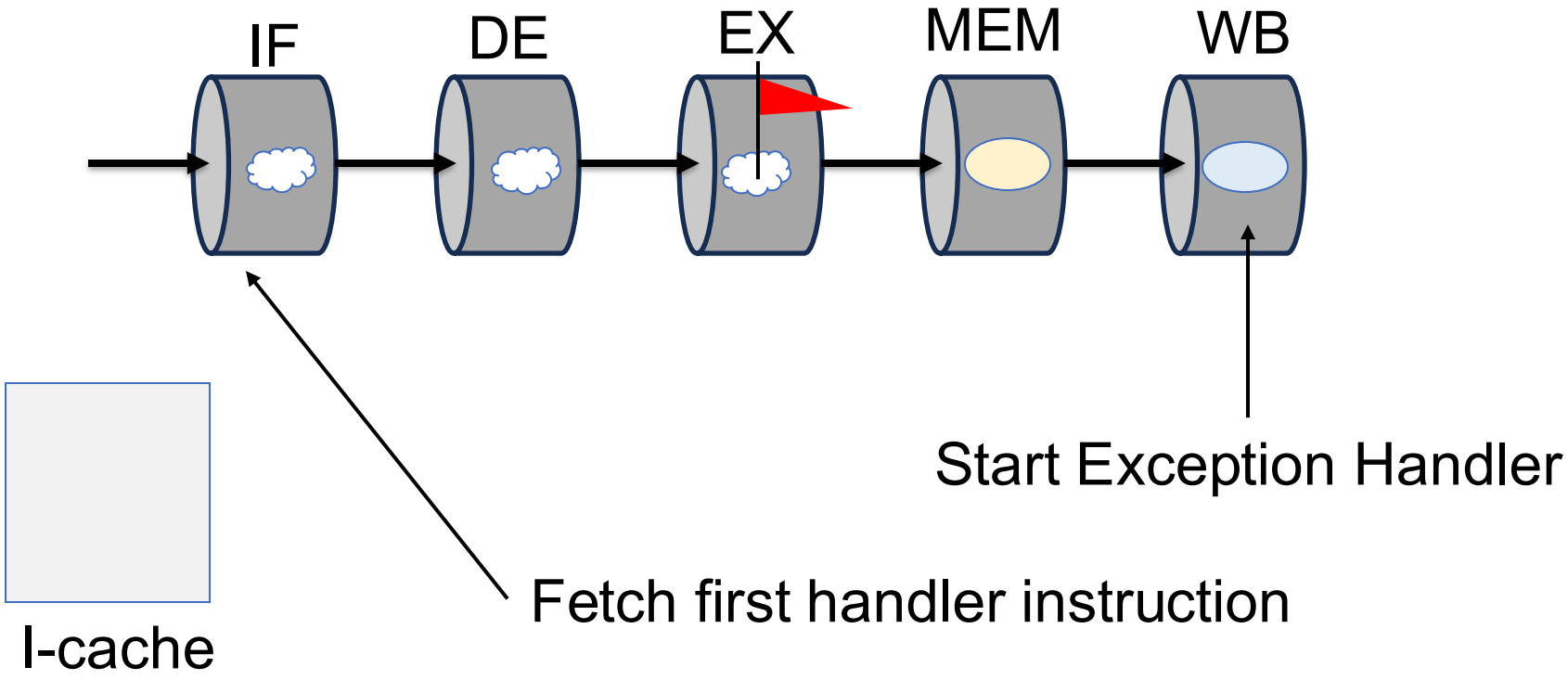


Precise Exceptions in the Pipeline

Bauer p. 41

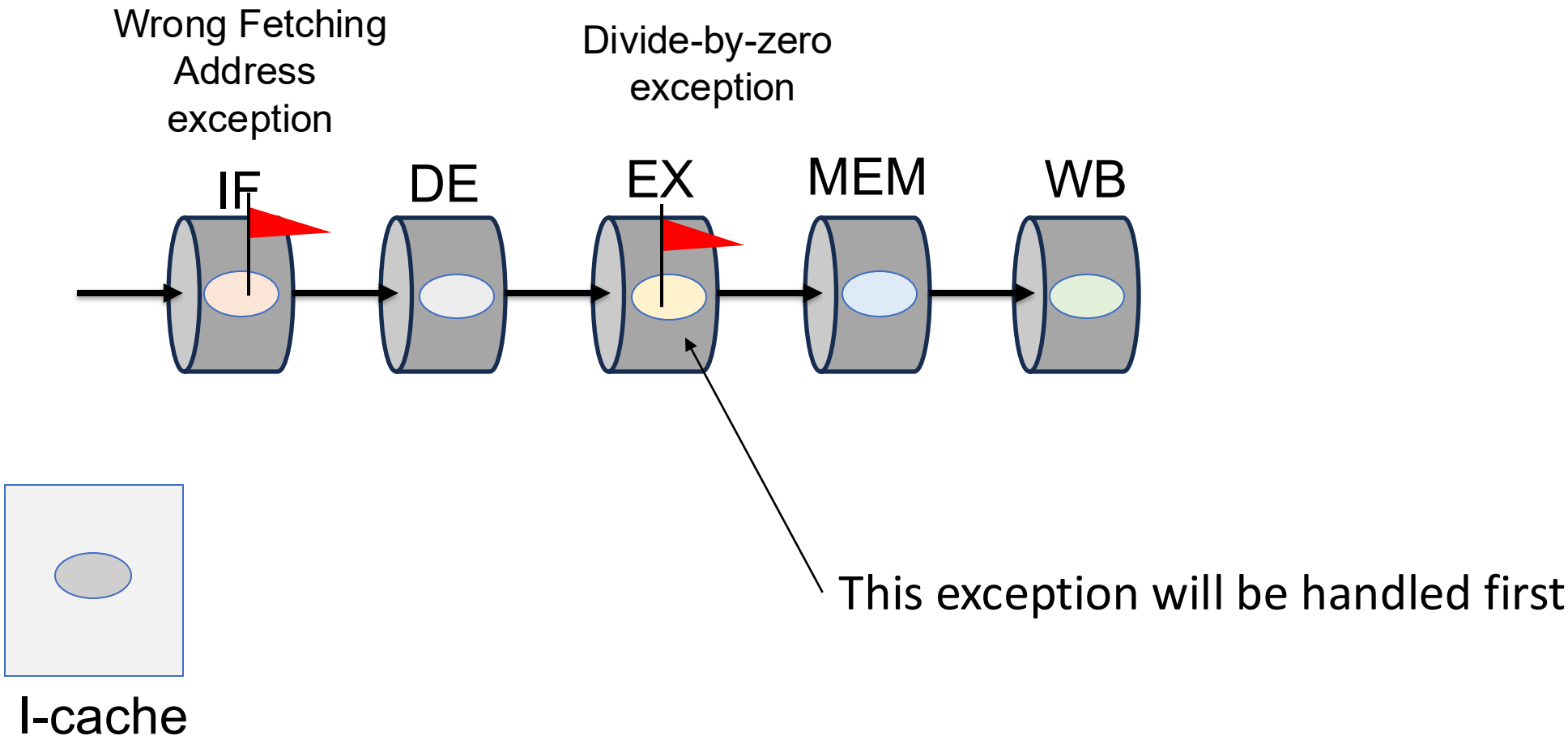

These instructions are nop
Divide-by-zero exception
IF
DE
EX
MEM
WB
These instructions will complete
Stop fetching instructions
I-cache

Bauer p. 41


IF
DE
EX
MEM
WB
Start Exception Handler
Fetch first handler instruction
I-cache

Bauer p. 41

Handling Exceptions in Program Order


Wrong Fetching Address exception
Divide-by-zero exception
IF
DE
EX
MEM
WB
This exception will be handled first
I-cache

Bauer p. 41

Multiple Exceptions

- Pipelining is executing multiple instructions
  - Could have multiple exceptions at once
- Simple approach: deal with exception from earliest instruction
  - Flush subsequent instructions
  - “Precise” exceptions
- In complex pipelines
  - Multiple instructions issued per cycle
  - Out-of-order completion
  - Maintaining precise exceptions is difficult!

Imprecise Exceptions

- Just stop pipeline and save state
  - Including exception cause(s)
- Let the handler work out
  - Which instruction(s) had exceptions
  - Which to complete or flush
    - May require “manual” completion
- Simplifies hardware, but more complex handler software
- Not feasible for complex multiple-issue out-of-order pipelines