

## 1 Exceptions

1.1 Define precise exceptions

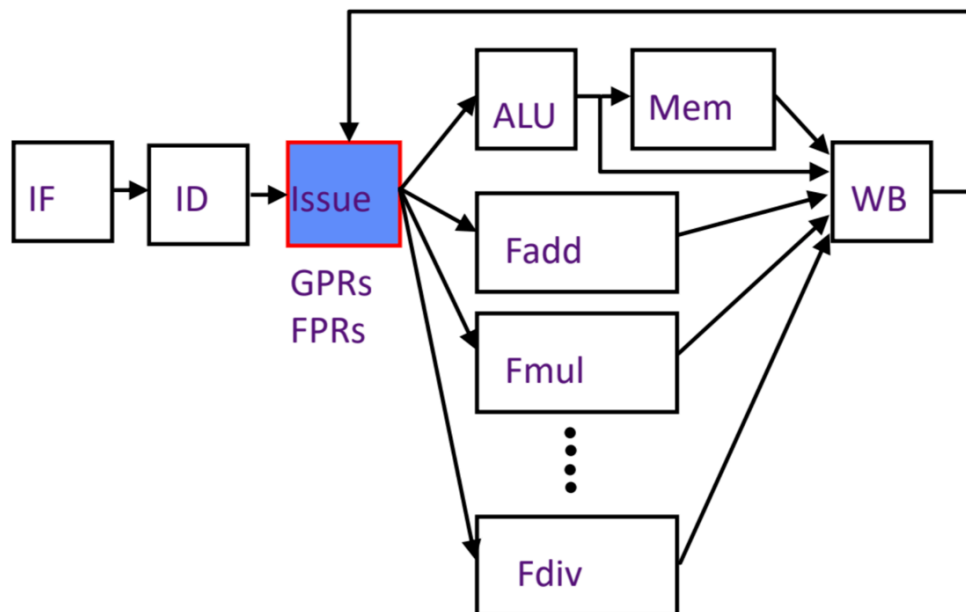
1.2 Why are precise exceptions useful?

1.3 Why might one not want to implement precise exceptions?

1.4 Describe how program execution time may be affected by adding support for precise exceptions.

1.5 Compare and contrast precise exception handling with branch misprediction

## 2 More Complex Pipelines



2.1 What additional hazards are introduced by this pipeline?

2.2 How are precise exceptions enforced in this pipeline?

## 3 Cache Organization

3.1 Why caches?

3.2 Define and give an example of a program that exhibits the following:

(a) Temporal locality:

(b) Spatial locality:

3.3 Consider a 1 KiB 4-way set-associative cache with 32-byte cache lines. The address is 12 bits wide. How are the address bits partitioned?

Tag:

Index:

Offset:

## 4 Replacement Policies

Suppose we see the following stream of accesses where A, B, C, D, and E represent unique addresses from different lines that all map to the same set:

A, B, C, D, B, A, E

Assume the cache has four ways and all lines in the set are initially invalid. When address E is accessed, one of the cache lines must be evicted. For each replacement policy, which line gets evicted?

### 4.1 First In, First Out (FIFO)

| Access | Way0 | Way1 | Way2 | Way3 | Policy State After Request |
|--------|------|------|------|------|----------------------------|
| A      |      |      |      |      |                            |
| B      |      |      |      |      |                            |
| C      |      |      |      |      |                            |
| D      |      |      |      |      |                            |
| B      |      |      |      |      |                            |
| A      |      |      |      |      |                            |
| E      |      |      |      |      |                            |

### 4.2 Not Most Recently Used (NMRU)

| Access | Way0 | Way1 | Way2 | Way3 | Policy State After Request |
|--------|------|------|------|------|----------------------------|
| A      |      |      |      |      |                            |
| B      |      |      |      |      |                            |
| C      |      |      |      |      |                            |
| D      |      |      |      |      |                            |
| B      |      |      |      |      |                            |
| A      |      |      |      |      |                            |
| E      |      |      |      |      |                            |

## 4.3 Least Recently Used (LRU)

| Access | Way0 | Way1 | Way2 | Way3 | Policy State After Request |
|--------|------|------|------|------|----------------------------|
| A      |      |      |      |      |                            |
| B      |      |      |      |      |                            |
| C      |      |      |      |      |                            |
| D      |      |      |      |      |                            |
| B      |      |      |      |      |                            |
| A      |      |      |      |      |                            |
| E      |      |      |      |      |                            |

## 4.4 Pseudo-LRU (PLRU)

| Access | Way0 | Way1 | Way2 | Way3 | Policy State After Request |
|--------|------|------|------|------|----------------------------|
| A      |      |      |      |      |                            |
| B      |      |      |      |      |                            |
| C      |      |      |      |      |                            |
| D      |      |      |      |      |                            |
| B      |      |      |      |      |                            |
| A      |      |      |      |      |                            |
| E      |      |      |      |      |                            |

## 5 Causes of Cache Misses

5.1 Define the following terms:

(a) Compulsory Misses

(b) Capacity Misses

(c) Conflict Misses

## 6 Cache Write Policies

6.1 Describe the cache write policies on:

(a) Cache Hit:

(b) Cache Miss:

6.2 What combinations of these policies make sense?

6.3 What issues arise when multiple CPUs access the same shared memory?

## 7 Cache Optimizations

For each technique, indicate whether implementing it will increase, decrease, or have no change on each aspect.

| Technique                            | Hit Time | Miss Penalty | Miss Rate | Hardware Complexity |
|--------------------------------------|----------|--------------|-----------|---------------------|
| Smaller, simpler caches              |          |              |           |                     |
| Multi-level caches                   |          |              |           |                     |
| Smart replacement policy             |          |              |           |                     |
| Pipelined writes                     |          |              |           |                     |
| Write buffer                         |          |              |           |                     |
| Sub-blocks (sector cache)            |          |              |           |                     |
| Code optimization                    |          |              |           |                     |
| Compiler prefetching                 |          |              |           |                     |
| Hardware prefetching (stream buffer) |          |              |           |                     |
| Victim cache                         |          |              |           |                     |