Class Announcements

1. Programming Assignment 2 was due by 9 AM today. You can submit it upto 48 hours after the deadline with penalties.

2. Email me if you will have conflicts with the CS354 Midterm Exam 2:
   Nov 10th Tues 5:30 PM to 7:00 PM at Van Vleck Room B130(Section 2)
   (Come to the Location about 15 mins earlier)

Lecture Overview

1. Types of Cache misses

2. Looking up the cache contents in Set Associative Caches

3. Tracing through an example Set Associative Cache
Types of Misses

- **Compulsory or cold misses**: Cache is empty to start with and will miss.

- **Conflict misses**: Cache has space but because objects map to the same cache block they keep missing.

- **Capacity misses**: Cache does not have space because size of the working set exceeds the size of the cache.

Conflict misses are common

- **Consider**:

```c
float dotprod(float x[8], float y[8])
{
    float sum = 0.0; register int i;
    for(i=0;i<8;i++)
        sum += x[i] * y[i];
    return sum;
}
```

Analyze for \((S,E,B,m) = (2,1,16,6)\)
Conflict misses are common

- **It causes thrashing:** repeatedly loading and evicting same cache blocks

- **Thrashing is easy to avoid once you know it is going on:** Use padded arrays so that the accessed elements are mapped to different cache sets

---

**Set Associative Cache Organization**

<table>
<thead>
<tr>
<th>Set 0: Valid</th>
<th>Tag</th>
<th>Cache block</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set 1: Valid</th>
<th>Tag</th>
<th>Cache block</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set S-1: Valid</th>
<th>Tag</th>
<th>Cache block</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**To reduce conflict misses**

- Increase set associativity

**Example**

- **4-way set associative**
- 2 blocks per set (line)

---

**Larger set size**

- Tends to lead to higher hit ratio (due to fewer conflict misses)
- Amount of circuitry goes up, leading to increase in $T_c$
Looking up a memory address in Set Associative Cache

1. The valid bit must be set.
2. The tag bits in one of the cache lines must match the tag bits in the address.
3. If (1) and (2), then cache hit, and block offset selects starting byte.

Tracing through a sample Set Associative Cache from CSAPP textbook practice problem 6.13

Set Associative Cache Practice Problem 6.13-6.16
- Consider a cache with: \( (S,E,B,m) = (8,2,4,13) \)
- Analyze memory references to:
  - 0x0E34
  - 0x0DD5
  - 0x1FE4
  The memory layout is shown in the next slide.
Cache Replacement Policies

- Which block to replace or evict to make space for new blocks?
  - Random Replacement Policy: chooses a random victim block.
  - Least Recently Used (LRU) Policy: chooses the block that was last accessed furthest in the past.
  - Least Frequently Used (LFU) Policy: chooses the block that was least frequently accessed in the past.