CSE 351 - Section 7

Caches
Written Homework #3

• If you are having problems, come to office hours
Caches
What a cache looks like

A cache is just a **fixed-size hash table**!

- **key**: address
- **value**: data at that address

**Size of a data block is configurable**
What a cache looks like

A cache is just a *fixed-size* hash table!

- **key:** address
- **value:** data at that address

![Diagram of cache structure](image.png)
What a cache looks like

A cache is just a *fixed-size* hash table!

key: address  
value: data at that address

What hash function should we use?

[Diagram of a cache with hash function and data blocks]
What a cache looks like
(direct mapped cache)

What happens on a hash collision?

B bytes per data block
Pathological Case
(direct mapped cache)

A simple program:

```
int a[64];
int b[64];
for (i=0; i<64; ++i)
    b[i] = a[i];
```

What if:

- \(\&a = 0x000A\) 020 00
- \(\&b = 0x000B\) 020 00

There will be a cache miss on every access!

- \(\&a[0] = 0x000A\) 020 00
- \(\&b[0] = 0x000B\) 020 00
- \(\&a[1] = 0x000A\) 020 01
- \(\&b[1] = 0x000B\) 020 01

Note the alternating tags

Solution: associative sets
What a cache looks like
(set associative cache)

- select set
- find matching tag

B bytes per data block
What a cache looks like
(set associative cache)

\[ E = 2^e \text{ lines per set} \]

\[ S = 2^s \text{ sets} \]

Address of word:
- \( t \) bits: tag
- \( s \) bits: set index
- \( b \) bits: block offset

Data begins at this offset

\[ B = 2^b \text{ bytes data block per cache line (the data)} \]
Cache Example
### 2-way set associative cache

<table>
<thead>
<tr>
<th>Set index</th>
<th>Tag</th>
<th>Valid</th>
<th>Byte 0</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>09</td>
<td>1</td>
<td>86</td>
<td>30</td>
<td>3F</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>45</td>
<td>1</td>
<td>60</td>
<td>4F</td>
<td>E0</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>EB</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>06</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>C7</td>
<td>1</td>
<td>06</td>
<td>78</td>
<td>07</td>
<td>C5</td>
</tr>
<tr>
<td>5</td>
<td>71</td>
<td>1</td>
<td>0B</td>
<td>DE</td>
<td>18</td>
<td>4B</td>
</tr>
<tr>
<td>6</td>
<td>91</td>
<td>1</td>
<td>A0</td>
<td>B7</td>
<td>26</td>
<td>2D</td>
</tr>
<tr>
<td>7</td>
<td>46</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tag</th>
<th>Valid</th>
<th>Byte 0</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>00</td>
<td>BC</td>
<td>0B</td>
<td>37</td>
</tr>
<tr>
<td>0B</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>12</td>
<td>08</td>
<td>7B</td>
<td>AD</td>
</tr>
<tr>
<td>05</td>
<td>1</td>
<td>40</td>
<td>67</td>
<td>C2</td>
<td>3B</td>
</tr>
<tr>
<td>6E</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>F0</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>DE</td>
<td>1</td>
<td>12</td>
<td>C0</td>
<td>88</td>
<td>37</td>
</tr>
</tbody>
</table>

Suppose a program running on the machine with the above cache references the 1-byte word at address 0x0E34. Assume addresses are 13 bits.

Address in binary:

```
0b01110001110100
```

CT  CI  CO
2-way set associative cache

<table>
<thead>
<tr>
<th>Set index</th>
<th>Tag</th>
<th>Valid</th>
<th>Byte 0</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Tag</th>
<th>Valid</th>
<th>Byte 0</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>09</td>
<td>1</td>
<td>86</td>
<td>30</td>
<td>3F</td>
<td>10</td>
<td>00</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>1</td>
<td>45</td>
<td>1</td>
<td>60</td>
<td>4F</td>
<td>E0</td>
<td>23</td>
<td>38</td>
<td>1</td>
<td>00</td>
<td>BC</td>
<td>0B</td>
<td>37</td>
</tr>
<tr>
<td>2</td>
<td>EB</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0B</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>06</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>32</td>
<td>1</td>
<td>12</td>
<td>08</td>
<td>7B</td>
<td>AD</td>
</tr>
<tr>
<td>4</td>
<td>C7</td>
<td>1</td>
<td>06</td>
<td>78</td>
<td>07</td>
<td>C5</td>
<td>05</td>
<td>1</td>
<td>40</td>
<td>67</td>
<td>C2</td>
<td>3B</td>
</tr>
<tr>
<td>5</td>
<td>71</td>
<td>1</td>
<td>0B</td>
<td>DE</td>
<td>18</td>
<td>4B</td>
<td>6E</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td>91</td>
<td>1</td>
<td>A0</td>
<td>B7</td>
<td>26</td>
<td>2D</td>
<td>F0</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>46</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DE</td>
<td>1</td>
<td>12</td>
<td>C0</td>
<td>88</td>
<td>37</td>
</tr>
</tbody>
</table>

CT = Tag = 0x71
CI = Set = 0x5
CO = Offset = 0x0

Set 0x5 has one line entry, with tag 0x71. HIT
Return byte 0 from set 0x5. 0xB