Cache Organization

March 18, 2016
CACHE
S Sets

Set 0

Set 1

CACHE

Set S-1
In this example, \( E = 3 \).
Each Block is $B$ bytes
This week (Chapter 6)

- Memory Hierarchy
- Locality
- Cache (Idea)

Today
- Cache Organization
Cache \rightarrow \text{Lines} \ (\# \ of \ lines = E)

Set 0

\vdots

Set s-1

In total \Rightarrow S \ sets.
Each set has \ E \ lines.
Set 0

Block

Size is B bytes

What are these blocks?

Blocks

$k+1$
Set 0

Tag bits  Block

Valid bit
0 → Not Valid
1 → Valid

B bytes

It is like an identifier
Uniquely identifies each line

Basic Structure //
6-bit address space (Example)

Partition this address

\[ \begin{align*}
\text{tag bit(s)} & \quad t = 2 \\
\text{set bit(s)} & \quad s = 1 \\
S & = 2^s = 2 \text{ sets}
\end{align*} \]

Block offset \( b = 3 \)
Cache has 2 sets

Example:

Value is at address $1001110$

Set 0

\[ 001100 \]

\[ \rightarrow \text{set 1} \]

If we did $s = 2$

Then $S = 2^s = 4$

00 \rightarrow \text{set 0}

01 \rightarrow \text{set 1}

10 \rightarrow \text{set 2}

11 \rightarrow \text{set 3}$

Scanned by CamScanner
Tag bits:

Eq. 010101

Set 0

The data item d at this address is somewhere in here.

Main Memory:

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>010110</td>
<td>0x78</td>
</tr>
<tr>
<td>010101</td>
<td>0x56</td>
</tr>
<tr>
<td>010100</td>
<td>0x34</td>
</tr>
<tr>
<td>010011</td>
<td></td>
</tr>
<tr>
<td>010010</td>
<td></td>
</tr>
<tr>
<td>010001</td>
<td></td>
</tr>
<tr>
<td>010000</td>
<td></td>
</tr>
<tr>
<td>001111</td>
<td></td>
</tr>
</tbody>
</table>
Block offset \( (b = 3) \)

\[ B = 2^b = 8 \text{ bytes} \]

Each block can store 8 bytes!

\[ d \text{ is at } 010100 \text{ offset} \]

Set 0

[Diagram of memory layout with hexadecimal values and addresses]
Suppose we want to access an integer \( x \) at 010000.

Read 4 bytes starting at offset 0000.

Suppose there is no line with the tag bit we were looking for!

\[ \downarrow \]

Cache miss.

Read the relevant data block from \((k+1)\) and we evict/replace an existing line.

\[ \rightarrow \] If the valid bit is not set (=0) then evict that line. (Simple)
If all lines have valid bit set (= 1), use a replacement policy.

Typically we do not read in a single byte!

We usually read/access 4 bytes at a time. (In a 32-bit system).

4 bytes \rightarrow 1 word.
Distinction

IA 32 (Instruction set)

1 byte = 1 byte
1 word = 2 bytes (16 bits)
1 double word = 4 bytes (32 bits)

32-bit processor

1 word = 32 bits (4 bytes)

Size of the cache

= \( S \times E \times B \) bytes
(excluding the tag bits and the valid bit).

= \( 2 \times 3 \times 8 = 48 \) bytes \( \parallel \).
How to describe a cache?

\[(S, E, B, m)\]

\[(2, 3, 8, 6)\]

How many bits for the set?

\[s = \log_2(S)\]
\[b = \log_2(B)\]

\[t = m - \left(\frac{s + b}{1 + 3}\right) = 2\]
Different classes of cache

\[
\text{based on the number of cache lines: } (E)
\]

If \( E = 1 \) (Direct Mapped Cache)
\[(S, E, B, m)\]  

\[m = 6\]

\[t s b\]

\[E = 1\]

\[E = 1000 \Rightarrow \text{Bigger the } E \text{ value, the fewer the misses you'll get.}\]

But it makes no sense to have \(E > 2^5\). Because you'll only be able to reference 2^5 lines at max.

\[C = S \times 10 \times B\]