CIS 429/529

Cache Organization II

Prof. Michel A. Kinsky
Cache Algorithm (Read)

• Look at Processor Address, search cache tags to find match. Then either

  Found in cache
    a.k.a. HIT
    Return copy of data from cache

  Not in cache
    a.k.a. MISS
    Read block of data from Main Memory
    Wait …
    Return data to processor and update cache

Q: Which line do we replace?
### Placement Policy

**Cache**
- Fully Associative: anywhere
- (2-way) Set Associative: anywhere in set 0, \((12 \模 4)\)
- Direct Mapped: only into block 4, \((12 \模 8)\)

**Memory**
- Block Number

**Set Number**
- 0: 1111111
- 1: 1111111
- 2: 1111111
- 3: 1111111
- 4: 1111111
- 5: 1111111
- 6: 1111111
- 7: 1111111

- 0: 01234567
- 1: 01234567
- 2: 01234567
- 3: 01234567

Block 12 can be placed anywhere.
Direct-Mapped Cache

- Tag
- Index
- Block Offset

- $V$
- $t$
- $k$
- $b$

- 2^k lines

- HIT

- Data Word or Byte
Direct Map Address Selection

- **Index**
- **Tag**
- **Block Offset**

**Diagram:**
- **Block Offset**:
  - $v$
  - $t$

- **Data Block**: $2^k$ lines

- **HIT**: Data Word or Byte

**Equation:**

$$V = t$$
Hashed Address Selection

Address

Block Offset

t

V
Tag

Data Block

Hash

2^k lines

Data Word or Byte

HIT

\( V \)

\( t \)

\( b \)
2-Way Set-Associative Cache

Tag | Index | Block Offset
---|---|---

V Tag Data Block V Tag Data Block

\[ t \]

\[ k \]

HIT

Data Word or Byte

Computer Architecture and Embedded Systems Laboratory (CAES Lab)
Fully Associative Cache

Diagram showing the components of a fully associative cache, including tags, data blocks, and block offsets. The diagram illustrates how data is accessed and validated for a hit.