Memory

Memory Technologies

SRAM static random access memory
DRAM dynamic ""
SMDRAM sync. DRAM
Rom read only memory
PRom programmable Rom
EPROM erasable PRom
EE PROM  electrically-erasable PROM
Flash  is EE PROM
Disk  magnetic media

Principle of locality
Spatial locality
int sumvec ( ) {
    int sum, i;

    sum = 0;
    for (i = 0; i < N; i++) {
        sum = sum + vec[i];
    }

    return (sum);
}
Temporal locality

hierarchicy

- A - register file
  - B - L1 cache
  - C - L2 cache
  - D - L3 cache
  - E - main memory
  - F - disk

- on-chip SRAM
- off-chip DRAM
movement of data between levels in fixed size units called blocks

A $\leftrightarrow$ B  Word 32-bit or 64 bits

B $\leftrightarrow$ C  4 to 8 words

E $\leftrightarrow$ F  4 Kbyte or more

on miss, move block from level i+1 (when it is) to level i. This might displace something from level i.
Need replacement policy

(e.g., random, recently used (LRU))

Consider two-level cache system

\[ B \leftrightarrow E \]

cache  main memory
Cache is org. as collection of sets

If $S = 2^s$ is # of sets, $s$ is # bits to ID set

Each set contains 1 or more cache lines

Each stores 1 block

$E$ is # of lines per set

If $B = 2^b$ is block size (in bits)

in bytes, $b$ is # bits to ID byte within block
Each line stores:

- Block (Data)
- Valid bit
- Tag bits (where in memory?)
- Dirty bit (is memory up to date?)

- If \( M \) is size of address memory, \( M = 2^m \)

\( m \) is # bits of address (logical addr)
characterize cache org. by $(S, E, B, m)$

divide $m$ addr. bits into 3 components

tag  set  index  block offset
1) direct-mapped cache \((E = 1)\)

Each set is 1 line

\(S\)

1 row

\(1 \leftrightarrow t \rightarrow < \rightarrow 8B \rightarrow > \rightarrow 1\)

Length: \(S \times B\) in bytes
on write, can either write through
or write back

on write miss, write allocate or
write no allocate

Classic pair

1) write through - no allocate
2) write back - allocate