

Seminar Caches and APCS

Mladen Nikitovic
mladen@imit.kth.se



Caches - locality

- Exploits *locality*
 - *temporal* - probability that data is used again soon
- ```
for(...) {
 a = data[0];
 b = c + a;
 data[0] = b;
}
```

September 21, 2004

2B1447 Embedded Systems

2

## Caches - locality



- Exploits *locality*
    - *spatial* - probability that surrounding (or following) data elements are used again
- ```
for(...) {  
    a = data[0];  
    b = data[1];  
    c = data[2];  
}
```

September 21, 2004

2B1447 Embedded Systems

3

Caches - locality



- Programs with regular access pattern - good locality
 - sequential calculation of data
 - matrix calculation
- Programs with irregular access pattern - poor locality
 - linked lists
 - database lookup

September 21, 2004

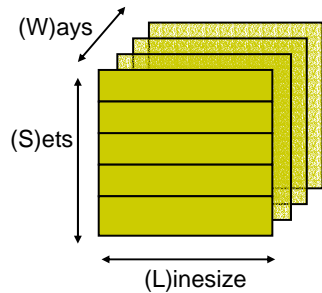
2B1447 Embedded Systems

4



Caches - terminology

- Set? Way? Block? Line? Index? Tag? Offset?
- Get the terminology right when discussing caches - or be prepared to be confused!



$$\text{Total size} = S \cdot L \cdot W$$

Typical question:
- total size & linesize given

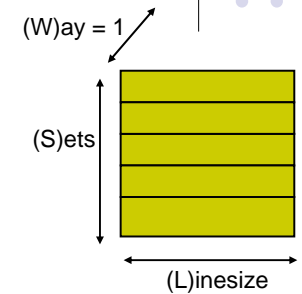
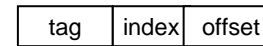
$$256 = S \cdot 16 \cdot W?$$

S or W determined by
determining associativity

Caches - terminology

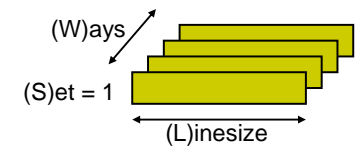
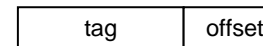
- Direct mapped cache ($W=1$)

$$\text{Total size} = S \cdot L \cdot 1 \quad S = 256/16$$



- Fully associative cache ($S=1$)

$$\text{Total size} = 1 \cdot L \cdot W \quad W = 256/16$$



Caches - Lab 3 checklist

- Make sure that cache module is running
 - console window should show that HarvardCache is loaded with correct parameters
- Read the cache parameters in the lab PM carefully
- Be sure to configure the cache correctly, otherwise the result is useless
- Don't forget to close debugger to generate statistics!!



APCS - Lab 4 checklist

- When using multiple source files (C,S), put them in the same project in the APM.
- Only one starting point of a program (main) is allowed, means either:
 - C - *main* function
 - S - *main* label followed by ENTRY
- Make the remote subroutine visible by doing a *extern* declaration inside its own source file
- Compile & run!!