Software-Managed Caches: Architectural Support for Real-Time Embedded Systems

Bruce Jacob

Electrical (and Computer) Engineering
University of Maryland, College Park

OUTLINE:
- Motivation: the problem with caches
- One Solution: software-managed

Why Traditional Caches Suck

NON-DETERMINISM
Solutions to the Problem

USE DSP-STYLE DATA CACHES
- Software explicitly manages movement
- What about instructions?

WIRE DOWN REGIONS OF MEMORY
- Usually at a page granularity (in TLB)
- Requires operating system assistance

PARTITION THE CACHES
- Solves part of the problem
- Inter-partition consistency an issue

DISABLE CACHES

Software-Managed Caches

- Guaranteed slow access-time
- Statistically fast access-time
- Guaranteed fast access-time
Software-Managed Caches

Top bits determine memory-access behavior

OS ensures that virtual-physical translation is the same for both subspaces

Other possibilities:
- Physical/virtual
- Faulting/non-faulting
- Which cache or memory structure

Application Behavior

```c
int *array = malloc (N * sizeof int);
int *stream = malloc (N * sizeof int);
int *mix = malloc (N * sizeof int);

for (i=0; i<N; i++)
    x = array[i];

stream |= MIN_NEG_INT; /* 0x80000000 */
for (i=0; i<N; i++)
    x = stream[i];

for (i=0; i<N; i++)
    x = (cache_it (i)) ? mix[i]
        : (mix | MIN_NEG_INT)[i];
```
Compiler Behavior

Group Data & Instructions to MINIMIZE CACHE CONFLICTS

Issue: Discontinuities

DATA SPACE
- Relatively easy to rearrange items

CODE SPACE
- Can move FUNCTIONS around easily
- PORTIONS of code is another matter ...

FINE-GRAINED PLACEMENT:
- Virtual addresses vs. physical addresses
- Segmented addresses potentially better

PROBLEM: fine-grained relocation at granularity of TLB page
Summary

Hardware support has been explored:


Hardware/software issues being explored as part of F-ZONE project

http://www.ee.umd.edu/~blj/