

Assumptions:

There were relatively few assumptions made in this paper. We assume that the modifications done to the cache simulator are correct and that the simulator performs as expected. We also assume that the multimedia and spec benchmarks yield an accurate representation of real world systems.

Summary:

Multimedia applications surround us in today's society. These applications come in everything from small handheld mp3 players the size of a pencil eraser to full-blown computers rendering frames for the newest PC games to cellular phones processing voice data. Central to all of these technologies is microprocessors that can process the data quickly and efficiently. Cache plays a big role in the throughput of instructions, and will be the central topic of this paper. Cache can be described by its total size, block size, and associativity. Current cache systems use set-associativity to reduce conflict misses, however this has the side effect of increasing the complexity of timing and the control logic. With this in mind, the objective of this research was to design a simple high performance cache system to support fast access time while reducing cache misses and power requirements.

The proposed cache design is called a dynamically aggressive spatial and adaptive temporal (DASAT) cache system. It is an extended combination of a dual direct-mapped cache module with a small block size and a fully associative cache module with a large block size at the same cache level. The fully associative cache with a large block size is used to exploit spatial locality. This cache is a traditional fully associative cache that has an added fetch mechanism that is adaptive and can fetch different data sizes based on the characteristics of the executing application. For each block in cache there is a tag field, a valid bit, a dirty bit and a hit bit. The hit bit is used to distinguish between referenced and unreferenced small blocks and the dirty bits are used to recognize data that has changed since the data was loaded. Two direct-mapped caches with small block sizes handle the temporal locality. There are  $n$  small block entries in these two caches and they cooperate logically as a main and shadow cache to reduce conflict misses. When a block is removed from the main cache, it is stored in the shadow cache at the same block index. If a hit occurs in the shadow cache, it is moved back into the main cache. The block size in this cache is kept small for fast access time and low power consumption.

When a memory reference is performed, both caches are searched in parallel within one cycle. When a hit occurs in the direct-mapped cache, it is handled like the common cache systems of today. When a miss occurs in the direct-mapped cache but hits in the spatial cache, the data is fetched and the hit bit is set. Simultaneously the hit bits for the cache are copied to the oldest fetch prediction buffer for future use in the dynamic fetch controller. When a global miss occurs, the dynamic fetch controller initiates the fetch of a new large block into the spatial cache from the next level of memory. This dynamic fetch can be 32, 64, or 96 bytes.

An interesting feature of dirty bits is that it reduces the write traffic to memory. Write-back in the proposed system only has to write dirty memory spaces saving on utilization of the memory bus.

Limitations:

In this paper, it was made very clear that the DASAT cache performs approximately the same as a traditional cache with 8 times the memory, and a victim cache with 2 times the memory. The paper does not examine the scalability of this design. A direct comparison with equal cache allowances is never made.

#### Experimental Results:

To get a good idea of how this cache performed in both traditional and multimedia applications a variety of benchmarks were used. These include six of the SPECint95 benchmarks, and ten multimedia benchmarks to represent multimedia and communication applications. Traces were generated by QPT2, and the DineroIV cache simulator was modified to simulate this cache system.

The first test was to determine how well utilized the dynamic fetch mechanism was. In the Spec95 tests, 32-byte fetch sizes were almost exclusively used. In the media benchmarks, the 96-byte fetch size was used more frequently, but the 32-byte fetch was still used frequently.

To determine the effectiveness of this caching system, the second test compared the performance of the proposed caching system to that of more traditional caches. For the simulations run, the DASAT cache was allotted two 4KB temporal caches and a 1KB spatial cache. The conventional caches used for the comparison had 32 and 64 KB cache sizes, as well as also 2 and 4 way set associative 16KB caches. After running the tests, it was discovered that the average miss ratio of the DASAT cache is approximately the same as that of the conventional cache's with approximately 8 times the space in regular applications. In multimedia applications, the DASAT shows better performance. The tests also reveal that the average memory access time of the DASAT cache is comparable in general application, but performs better in applications with a high degree of spatial locality. Media applications with large block fetch sizes show very high performance boosts when compared to traditional caching approaches. Comparisons were also made to the victim cache design implemented in many processors. The results show that the DASAT cache achieves better performance when allotted half or even a quarter of the same amount of space.

#### References:

- J.-H. Lee, S.-D. Kim, and C. Weems, "[Application Adaptive Intelligent Cache Memory System](#)," *submitted to ACM Transactions on Embedded Computing Systems, Special Issue on Memory Systems*, January 2002.