1 Description

In recent years, increases in memory subsystem speed have not kept pace with the increase in processor speed, causing processor execution rates to become increasingly limited by the latency of accessing instructions and data. On-chip caches are a popular technique to combat this speed mismatch. As integrated circuits become denser, designers have more chip area that can be devoted to on-chip caches. Straight-forward scaling of cache sizes as the available area in- creases, however, may not be the best solution, since the larger a cache, the larger its access time. Using cache hierarchies (two or more levels) is a potential solution. This paper explores the tradeoffs in the design of on-chip microprocessor caches for a range of available on-chip cache areas.

There are a number of potential advantages of two-level on-chip caching with a mixed (in- struction and data) second-level cache over single-level on-chip caching. First, the primary cache (also referred to as the L1 cache) usually needs to be split into separate instruction and data caches to support the instruction and data fetch bandwidths of modern processors. By having a two-level hierarchy on-chip where the majority of the cache capacity is in a mixed second-level cache (L2 cache), cache lines are dynamically al- located to contain data or instructions depending on the program's requirements, as opposed to living with a static partition given by single-level on-chip cache sizes chosen at design time.

A second and more important potential advantage of two-level on-chip caching is an improve- ment in cache access time. As existing processors with single-level on-chip caching are shrunk to smaller lithographic feature sizes. If the additional area available due to a process shrink is used to simply extend the first-level cache sizes, the caches will get slower relative to the processor datapath. Instead, if the extra area is used to hold a second-level cache, the primary caches can scale in access time along with the datapath, while additional cache capacity is still added on-chip.

A third potential advantage of two-level cache structures is that the second-level cache can be made set-associative while keeping the primary caches direct-mapped. This keeps the fast primary access time of direct-mapped caches, but reduces the penalty of first-level conflict misses since many of these can be satisfied from an on-chip set-associative cache instead of requiring an off-chip access.

When primary cache sizes are less than or equal to the page size, address translation can easily occur in parallel with a cache access. However, most modern machines have minimum page sizes of between 4KB and 8KB. This is smaller than most on-chip caches. By using two-level on-chip caching, the primary caches can be made less than or equal to the page size, with the remaining on-chip memory capacity being devoted to the second-level cache. This allows the address translation and first-level cache access to occur in parallel. This is a fourth potential advantage of two-level cache structures.

A fifth advantage of two-level cache structures is that a chip with a two-level cache will usually use less power that one with a single-level organization (assuming the area devoted to the cache is the same). In a single-level configuration, wordlines and bitlines are longer, meaning there is a larger capacitance that needs to be charged or discharged with every cache access. In a two-level configuration, most accesses only require an access to a small first-level cache.

2 Why topic is interesting and important ?

Several trends in current technology have increased the vi- ability of a two-level cache, for example:

- Technology is yielding impressive reductions in processor speed, but memory speed has not kept pace. Therefore, the disparity between processor and memory speed is in- creasing.
- The appetite for memory is rapidly increasing and pri- mary memories of 100s to 1000s of megabytes will be common. Larger memories typically have larger access times because of packaging and physical constraints.
- As on-chip densities increase, it becomes possible to in- clude small on-chip instruction and/or data caches. These caches will need to be backed up by larger second-level caches. As one example, the MicroVAX 3500 has a 1K on-chip cache and a 64K second-level cache.
- Shared memory multiprocessors require local caches to reduce bus contention. A second-level global cache can help to further reduce access time on cache misses.

Therefore, from the above points it sums up that processors have more than one level cache in order to increase the capacity of the processor cache without also dramatically increasing the price of the processor. This careful mixture allows for processors that are faster and cheaper.

Cache memory is very useful because it saves the computer user a lot of time in opening common data and giving common commands. Cache memory is very convinient because opening data in cache is dramatically faster than opening data in the main hard disk. Because of this utter importance of cache it is really necessary to discuss on this topic and bring out some revolutionary improvement in this area.

3 Conclusion

We have studied several aspects of two-leve cache memories in uniprocessors.We have modeled the miss rate, cache area, and cache access time to achieve a solid basis to study on-chip memory system tradeoffs.

An extra level of cache memory can provide a worthwhile performance gain when used with proper combinations of small first-level caches and large main memory access times. Two-level on-chip cache hierarchies perform even better in low-cost systems without a board-level cache.

Combining this with set-associative second-level caches improves performance even further. This is because the increase in capacity provided by two-level exclusive caching increases as the second level of caching is made more associative.

Write-back strategies were studied for both cache levels. The best performance is provided by write-back at both levels The principal effect of write-back was the reduction of write bandwidth required at the next higher level of memory.

If only one of the caches were to use write-back,then write-back should be included in the L1 cache because

- The L1 cache typically has a faster access time, and writes could be processed more quickly without buffering, and
- The L1 cache is smaller than the L2 cache.

Lastely, multilevel caches provide a fruitful area for study, and much more needs to be done.

Certainly an important area for further evaluation is multi-level caches in multiprocessors. In this study, we concentrated on only 2-level Cache.