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.