Cache Replacement Algorithms

Replacement algorithms are only needed for associative and set associative techniques.

1. Least Recently Used (LRU) – replace the cache line that has been in the cache the longest with no references to it

2. First-in First-out (FIFO) – replace the cache line that has been in the cache the longest

3. Least Frequently Used (LFU) – replace the cache line that has experienced the fewest references

4. Random – pick a line at random from the candidate lines

Note1: LRU is probably the most effective

Note2: Simulations have shown that random is only slightly inferior to an algorithms based on usage

Cache Write Policies

If a cache line has not been modified, then it can be overwritten immediately; however, if one or more words have been written to a cache line, then main memory must be updated before replacing the cache line.

There are two main potential write problems:

- If an I/O module is able to read/write to memory directly, then if the cache has been modified a memory read cannot happen right away. If memory is written to, then the cache line becomes invalid.

- If multiple processors each have their own cache, if one processor modifies its cache, then the cache lines of the other processors could be invalid.

1. write through – this is the simplest technique where all write operations are made to memory as well as cache ensuring main memory is always valid. This generates a lot of main memory traffic and creates a potential bottleneck

2. write back – updates are made only to the cache and not to main memory until the line is replaced

cache coherency – keeps the same word in other caches up to date using some technique. This is an active field of reseach.

Cache Line Size

Cache lines sizes between 8 to 64 bytes seem to produce optimum results

Multilevel Caches

An on-chip cache reduces the processor's external bus activity. Further, an off-chip cache is usually desirable. This is the typical level 1 (L1) and level 2 (L2) cache design where the L2 cache is composed of static RAM.

As chip densities have increased, the L2 cache has been moved onto the on-chip area and an additional L3 cache has been added.

Unified vs Split Caches

Recent cache designs have gone from a unified cache to a split cache design (one for instructions and one for data).

Unified caches have the following advantages:

1. unified caches typically have a higher hit rate

2. only one cache is designed and implemented

Split caches have the following advantages:

1. parallel instruction execution and prefetching is better handled because of the elimination of contention between the instruction fetch/decode unit and execution unit.

Table 4.4 Intel Cache Evolution

Problem	Solution	Processor on which feature first appears
External memory slower than the system bus.	Add external cache using faster memory technology.	386
Increased processor speed results in external bus becoming a bottleneck for cache access.	Move external cache on-chip, operating at the same speed as the processor.	486
Internal cache is rather small, due to limited space on chip	Add external L2 cache using faster technology than main memory	486
Contention occurs when both the Instruction Prefetcher and the Execution Unit simultaneously require access to the cache. In that case, the Prefetcher is stalled while the Execution Unit's data access takes place.	Create separate data and instruction caches.	Pentium
Increased processor speed results in external bus becoming a bottleneck for L2 cache access.	Create separate back-side bus that runs at higher speed than the main (front-side) external bus. The BSB is dedicated to the L2 cache.	Pentium Pro
	Move L2 cache on to the processor chip.	Pentium II
Some applications deal with massive databases and must have rapid access to large amounts of data. The on-chip caches are too small.	Add external L3 cache.	Pentium III
	Move L3 cache on-chip.	Pentium 4

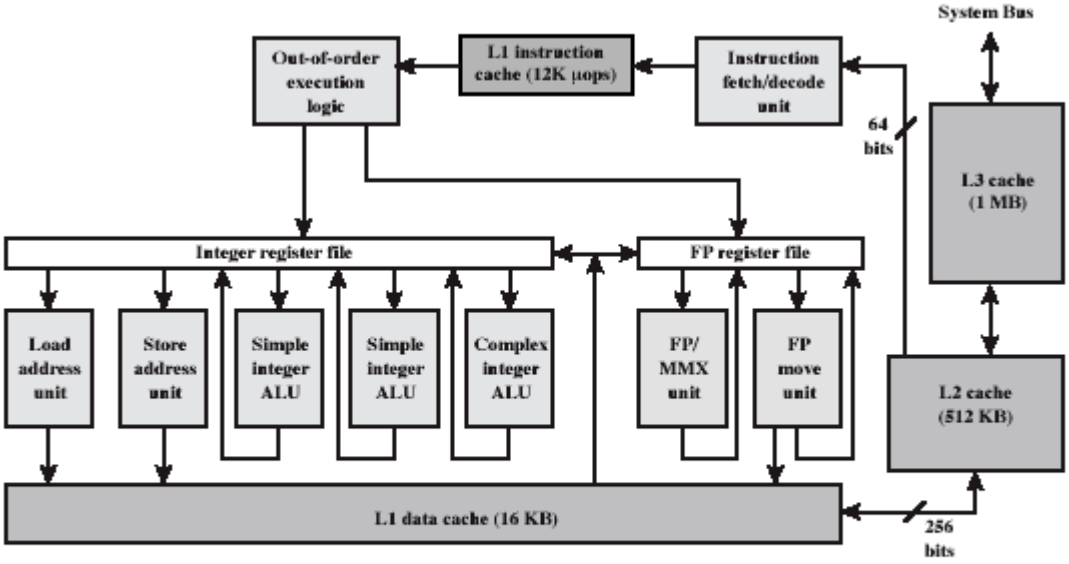

Figure 4.13 Pentium 4 Block Diagram

Table 4.6 PowerPC Internal L1 Caches

Model	Size	Bytes/Line	Organization
PowerPC 601	1 32-Kbyte	32	8-way set associative
PowerPC 603	2 8-Kbyte	32	2-way set associative
PowerPC 604	2 16-Kbyte	32	4-way set associative
PowerPC 620	2 32-Kbyte	64	8-way set associative
PowerPC G3	2 32-Kbyte	64	8-way set associative
PowerPC G4	2 32-Kbyte	32	8-way set associative
PowerPC G5	1 32-Kbyte, 1 64-Kbyte	32	8-way set associative

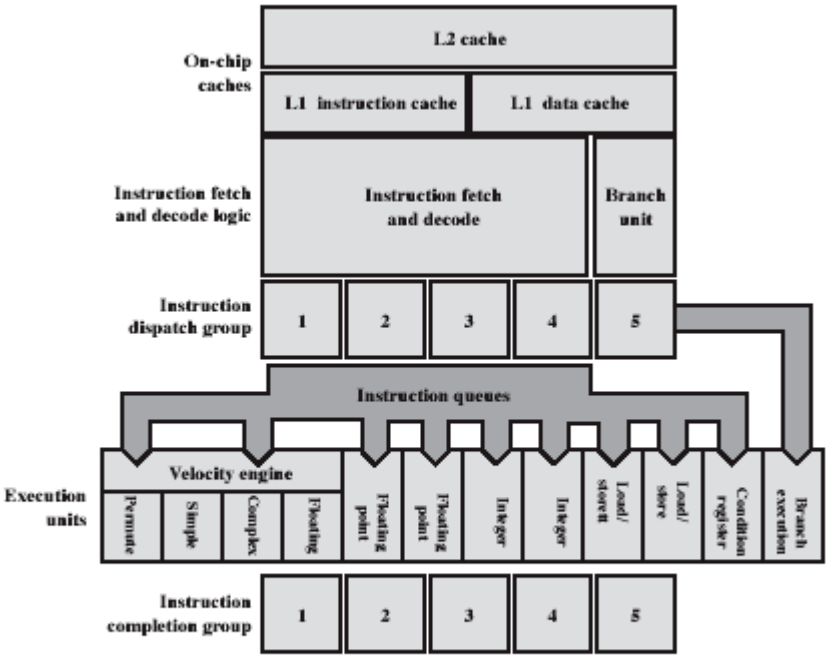

Figure 4.14 PowerPC G5 Block Diagram