CacheHit

General Overview:

My CacheHit program is designed to help you test memory access speeds on your PowerMac. It does not require a second level(L2) cache.

The PowerPC 601 chip which is in most of the PowerMacs now shipping, has a L1 cache inside the CPU which is 32K in size. A second level cache, L2, is an option on some of these models, see below, and is designed to help speed up your machine by making data available faster to the CPU since it is held in a 10ns(or 12ns) L2 cache rather than in main memory, which is 80ns. Second level cache's are usually between 256K to 1024K in size. The original 6100/60 and 7100/66 PowerMacs came without a L2 cache, while the 8100/80 had a 256K second level cache. The newer 6100/66 and 7100/80 both come with a 256K second level cache.

Usage:

The main output window has the following columns:

Size: Size of data array being accessed(in K=1024 bytes)
Time:Time in ticks(1/60 of a second) to complete the memory reference.
IO Rate(MB/Sec): Calculated IO rate based on Size and Time.

The DataSize menu lets you set how much memory is going to be accessed in total. The relationship between size and DataSize is, DataSize indicates 512MB(by default) will be accessed and size indicates how big an array will be used. Thus, to access 512MB an 8K array must be traversed (512*1024)/8=65536 times. The operation done on the array is reading, one value then another(no writing is done). Why have a DataSize menu choice? On faster machines 1/60 of a second is not fine grain enough a timer, thus you may want to use a larger choice. It takes about 2 minutes for the program to run on my 7100 with the default options.

Under the Calculate menu, the 'Size of L2 cache' selection exists, but is grayed out. Once I have enough data, I will make this option work. Possibly in a refresh of this program in October 95.

Over View of the results:

Results for many machines are in two files, one an excel file, the other a text file which is best read by BBEdit since you don't want the lines to wrap. They contain the same information.

Accessing data in the L1(CPU internal) cache memory is about 5.5-6.5 times faster than reading data from L2 cache memory. Accessing data in L2 memory is about 4 times faster than going to main memory. L1 is about 23-26 times faster than going to main memory. Also, you will notice that clock chipping might make your 6100/60 go 80 Mhz but since the main memory bus is still operating slower than that of a machine like the 8100/80 or even the 7100/66, the memory access speeds on larger arrays will not be as fast as on those machines.

Note a definite drop off at the 28-30K data size range, which is what one would expect since the L1 cache is only 32K and the system plus test program are going to eat into that at least to some degree. You will also see a drop off just as the size of the L2 cache gets exhausted, which is just what you would also expect. The 6100's main memory access is by far the worst of the lot, I would guess that this is caused by the use of DRAM video such that the video and processor are both fighting for access to the memory bus. An L2 cache makes the largest difference in performance on such systems(one's using DRAM video). I would like to see some numbers from people with 7100/8100 using DRAM video to confirm this conclusion.

How much faster will your system go if you add a L2 cache, or get a larger one? I can't really say, and this program was just designed to give you an idea of the memory access speed. Real programs don't just stay in a tight loops reading from memory, my program does though.

Getting more detailed data reports:

If you try this program before September 95 please e-mail me your output along with: Machine(6100/7100/8100/etc), original clock speed, clock chipped speed, L2 cache size. In October I will update this file and make the larger test results known.

Your output is a lot worst than that in the results file, what's wrong?

Possibly nothing, you should try the test again after rebooting with extensions turned off. Non native control panels/extensions can cause a large performance degradation.

Revision history:

1.0	May 95	Initial release.

Glen Lalonde,
glalonde@vnet.ibm.com