

Re: Caches in embedded systems

Source: <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2006-05/msg00171.html>

- *From:* Paul Keinanen <keinanen@xxxxxx>
 - *Date:* Thu, 04 May 2006 11:14:01 +0300
-

On 3 May 2006 18:04:17 -0700, "shrey" <shreyas76@xxxxxxxxxx> wrote:

I know caches are avoided in real time applications

Unless the cache is very badly implemented, the worst case timing occurs when the cache is disabled.

Thus, it is sufficient to verify that all high priority tasks with some definite deadlines are executed within the deadlines even when the cache is disabled.

A system usually also contains tasks that are not time critical (such as user interfaces or calculating weekly statistics), which are executed, when *_no_* high priority task is using the processor. Thus, the low priority tasks can progress quite slowly, if there is much high priority activity.

Enabling the cache reduces *_on_average_* the execution time of the high priority tasks, freeing up some additional time for the low priority tasks, which will progress more rapidly and ultimately allows the null task to run at times when there is no useful work to be done.

The net effect of enabling the cache is that the high priority tasks still complete within the deadlines, but the visible result is that the low priority background tasks executes faster.

Enabling the cache will also reduce the number of main memory accesses, thus reducing the memory and memory driver power consumption. This is usually true even if the low priority tasks can now perform more work, since the cache hit rate is usually quite high.

Paul

.