

PowerPC Optimizations

Efficiency improvements

Thu, Aug 29, 2002

Although the PowerPC MPC750 processor, running at 233 MHz, is many times faster than the MC68040, running at 25 MHz, the slow access time to nonvolatile memory reduces the actual measured performance improvements of the PowerPC system. A number of improvements to the system code have been made to reduce the number of accesses made to nonvolatile memory. This note describes many of those changes in brief. More complete documents are available for each change.

One of the first changes made to respect the slow access time to nonvolatile memory was in the alarm scan logic. The original logic scanned through every possible allocated analog channel, even those entries in the `ADATA` table that are not marked to be scanned for alarms, or those that are even unused. For each channel marked for alarm scanning, it performed the check, then advanced to the next channel number. A node with 1024 allocated channels, which is typical, needed 1024 accesses even if no channel was marked for alarm scanning. The logic was rewritten to maintain a list of all those channels that are marked for alarm scanning, updating the list as various channels are marked or unmarked for alarm scanning to reflect operational needs. Also, when performing an alarm check, care was taken to minimize the accesses required to obtain the reading, the nominal and tolerance values, and the alarm flags and trip count. As a result, the actual alarm scanning time in Linac nodes, even for those with 2048 allocated channels, rarely exceeds 1 ms. (Actually, the alarm scan also allows for scanning digital status bits, too, and there may be as many as 2048 bits allocated.) The time required for a complete alarm scan is about $(40 + 4*N)$ μ s, where N is the number of devices (channels or bits) checked for alarm conditions. An alarm scan time of 1.056 ms was measured for a heavy case of 244 devices (146 channels and 98 bits) checked.

Another modification improved the ability to search the `CINFO` table, which contains parameters about channels that have fast digitizer interfaces. When an `FTPMAN` request is being processed, this table is searched to see whether the requested channel is found as being one that has fast digitizer capability. Not only was the search done for each requested channel, it was also done multiple times, each time looking for different entry (digitizer) types. In planning for the future, we had allocated 512 entries in this table; hence, a search would require 512 accesses even when the table is empty. A request for 4 devices, none of which was found in the table, required 16 ms to satisfy! The solution for this one was to maintain a copy of the table in fast memory and recognize when the actual used length of the table is short, so that even the fast search is short.

Recently a scheme was implemented to avoid unnecessary searches of network-related tables, those that relate node numbers and IP addresses. Table lookups are used instead, so that there is no longer a dependence on N, the length of the table.

The `LATBL` of local applications is reviewed every 15 Hz cycle, so new logic was added to keep a fast memory copy. The fast copy is updated every time an `LATBL` entry is modified.

The `RDATA`, or Data Access Table, is interpreted every 15 Hz cycle, too, so a scheme was implemented to maintain a fast memory copy of this table, checking frequently to see whether the nonvolatile table has been modified, updating the copy if it has.

Another source of slow accesses is in the Digital PMC board. Such accesses also require about 1 μ s. One kind of access that is common during system operation is to turn LEDs on and off to signal system activity to any nearby human. A new scheme was designed to provide timing of interrupt routines, since many of the LEDs are used for that purpose. (One does not always have ready access to a scope to view the LED signals.) Part of the scheme supports access to LEDs. Multiplexing hardware only brings 4 interrupt LED signals out externally. If a LED that is to be set is not exposed externally via the multiplexer setting, the logic for setting and clearing it can be ignored, which can save at least two accesses per interrupt.