Arcnet Interrupt Debug

Diagnostic in PowerPC

Mon, Apr 12, 2004

Arcnet is used for communication with SRMs in the PowerPC version of the system. (It also exists for IRMs, but nearly all SRMs connect to Linac PowerPC systems today.) A diagnostic was added to the arcnet interrupt code to guard against being stuck inside the interrupt because of unrelenting interrupts being presented by the arcnet controller. The logic of the interrupt routine checks each possible interrupt cause and processes each one. As long as a status bit indicates an interrupt is active, the code will loop to process it. If too much time is ever spent in the interrupt routine, a diagnostic is written about this occurrence and the interrupt mask register is cleared, hoping that will cause the controller to stop the interrupt.

An area assigned for use with this arcnet diagnostic in nonvolatile memory is outside any system table. Its address is currently `0x4800FE00`, where the 24-byte structure is as follows:

Field	*Size*	*Meaning*
`key`	4	`0xDEAD` if used, else 0
`status`	1	copy of arcnet controller register status
`arcmask`	1	copy of arcnet interrupt mask
`counter`	2	count of #times used
`IRQtime`	4	elapsed time in interrupt routine, in microsec
`maxIRQTime`	4	max elapsed time ever
`date`	8	BCD date and time last used

Two fields, `IRQtime` and `maxIRQTime`, are updated during every execution of an arcnet interrupt. (The field `maxIRQtime` is only updated if `IRQtime` exceeds it.) The rest of the structure is only updated when `IRQtime` is seen to exceed 100000, or 100 ms. Many nodes never see a `DEAD` record. In principle, none ever should.

Reviewing these records across 21 nodes in Linac using arcnet, it was noticed that some values of `maxIRQtime` are negative, which would be interpreted as an exceedingly large elapsed time. The implication is that the elapsed time measurement is occasionally faulty. It is based upon the 1MHz counter that is part of the Digital PMC board. The logic to do this reads the upper 16 bits, the lower 16 bits, and the upper 16 bits again to be sure it has not changed. In this way, we can get a proper 32-bit reading. But the evidence shows that this can (rarely) fail. We may want to change the software to use the PowerPC timebase register instead, which is faster to access anyway.

The hardware that may not be quite perfect is possibly due to the hardware version actually installed. When this problem was uncovered in `node061C`, which had a `counter` field value of 8, we replaced the PMC board with one that has the latest version of the (hardware) code. Even without changing the software, this may fix this particular node. It may take a week to be sure we are ok. After installing the replacement PMC board in `node061C`, it was seen to have values for `IRQtime` and `maxIRQtime` of 126 and 178 μs, respectively.

As for other nodes, `node0626` exhibited a `counter` value of 5, and `node0621` had a `counter` value of 2. Eight other nodes had values of 1. The remaining 10 nodes had no `DEAD` records.

After writing the `DEAD` record and clearing the interrupt mask registers, we should be sure that the copy of the mask register does not have bit #0 set; else, the `ARCXPRI` routine will not allow any further transmits at all, so that arcnet communications will cease.