Interrupt Timing Table

Operations

Sun, Aug 25, 2002

Access to the Interrupt Timing Table (ITT) is currently available only via the Memory Dump page, or in general terms, by access to the 512-byte table at memory address 0x800—0x9FF. (This same address works for both MC68040-based systems and PowerPC-based systems, because of an automatic mapping scheme for low memory addresses. Each ITT entry is 16 bytes in size. The first 32 bytes of the table comprise the table header, which has the following layout:

Field	*Size*	*Meaning*
ITTQPtr	4	pointer to ITTLOG data stream queue
ITTAct	4	Pattern of active ITT entries this cycle
ITTCycT	4	Timer value at start of current 15 Hz cycle
ITTActL	4	Pattern of active ITT entries last cycle
ILEDCpy	2	Interrupt LEDs copy
TLEDCpy	2	Task LEDs copy
ILEDMsk	2	Interrupt LEDs selected mask
TLEDMsk	2	Task LEDs selected mask
ITTDate	8	Current time-of-day in usual BCD format

The format of each of the 30 ITT entries that follow the header is as follows:

Field	*Size*	*Meaning*
id	1	visual identifier of entry
flgBit	1	flags in upper 4 bits, LED bit number in lo 4 bits
count	2	count of such interrupts
elapsed	4	elapsed time in interrupt routine, or between interrupts
base	4	base time for measuring elapsed time
cycle	4	time since start of current cycle

By interacting with what is in these entries, one can use the ITT support to make a number of measurements. Note that nothing in this table, except for the pointer to the data stream queue, is especially critical. If something is overwritten by mistake, it may affect the measurements that can be made, but it should not place the operation of the underlying system in jeopardy. Not also that this kind of measurement is single user; there is no support for multiple simultaneous users to be able to make separate and independent measurements of interrupt timing.

When the system is initialized, several of the table entries are set up to provide timing and LED support for those interrupt routines. By default, they are set up to measure the elapsed time spent during interrupt routine execution, using an LED in most cases, and with logging inhibited.

For the most part, using the system to monitor performance involves changing the flags bits associated with an entry. In order to do that, it is necessary to identify which entry relates to which interrupt activity. Here are the id values used for the system interrupts:

id (hex)	*Interrupt*
A0	arcnet output
A1	arcnet input
AD	IRM 1KHz A/D
Bn	Timer interrupt

`C0`	little console output
`C1`	little console input
`CC`	`Cycle` routine in Update Task
`CD`	40 ms delay
`CE`	clock event
`CF`	15 Hz
`D0`	serial output
`D1`	serial input
`E0`	ethernet output
`E1`	ethernet input
`EC`	ethernet interrupt complete
`15`	MIL-STD-1553B
`8x`	dynamically-assigned

In the absence of a Tevatron clock to provide accelerator-synchronous 15 Hz interrupts, two 40 ms delays produce a backup rate of 12.5 Hz. The `Cycle` routine (`CC`) in the Update task is therefore invoked by either the 15 Hz interrupt (`CF`) or the 40 ms delay interrupt (`CD`). The 40 ms delay interrupt is always used to schedule server request support, as 40 ms after the start of the cycle represents a deadline for arrival of replies from nodes contributing to server requests, so that the composite replies can be delivered to the original requester. Thus, one will always see 40 ms interrupt activity, and the `Cycle` routine will always be invoked, but the 15 Hz interrupt may be missing.

The timer interrupts are especially useful with the PowerPC system digital PMC board, in which 4 timers are made available for node-specific uses.

The ethernet interrupts can be available for IRMs but not for the PowerPC, since vxWorks handles all that stuff.

The main result of using this diagnostic is in demonstrating what impact interrupt activity has in a system that purports to deliver reliable 15 Hz access to controls data. Detailed studies may be useful by setting the logging option for one or more `ITT` entries. A data stream request is the way to obtain the interrupt timing data logged in this way.

The flag bits are the upper 4 bits of the `flgBit` field in an `ITT` entry, having the following meanings:

flag#	*Meaning*
7	LED logic enabled
6	Measure time between interrupts, not time of interrupt routine
5	Measure maximum time of interrupt, or minimum time between interrupts
4	Logging enabled, with `ITT` data written to `ITTLOG` data stream.

For a typical example, if the `flgBit` byte contains `0x86`, it means that LED activity logic will be done using LED #6, and the elapsed time field will represent the last execution time of that interrupt. If it were changed to `0x96`, logging of such activity would be added, so that records are placed into the `ITTLOG` data stream queue. Since more than one interrupt can occur for a given type within one 15 Hz cycle, the only way to capture the timing data for multiple occurrences is via the logging option.