

PowerPC Bus Errors

Diagnostic aid

Mon, May 19, 2003

During operation of the system code, the `vxMemProbe()` function is invoked from time to time to test whether a memory address would produce a bus error if accessed. This avoids having the task suspended when a bus error occurs. This note describes a log facility that captures bus errors detected by `vxMemProbe()`.

The scheme replaces the calls to `vxMemProbe()` by a call to a wrapper function that provides the new logging facility. The wrapper function is the following:

```
STATUS MemProbe(char *adrs, int mode, int length, char *pVal);
```

The calling sequence is identical to that for `vxMemProbe()`. The new function calls `vxMemProbe()`. If the return value is `ERROR`, it writes to a data stream a record that describes the detected bus error. If the return value is not `ERROR`, it simply returns `OK`.

The format of the 16-byte data stream record is as follows:

<i>Field</i>	<i>Size</i>	<i>Meaning</i>
<code>taskNum</code>	1	current task number in range 0–15
<code>rwLength</code>	1	flag 0x10 if write; length 1,2,4 in ls 3 bits
<code>eTime</code>	2	elapsed time for the call to <code>vxMemProbe()</code> , in TB units
<code>adrs</code>	4	address whose access produced bus error
<code>time</code>	8	time of occurrence in usual BCD format plus ms

The data stream name is `BUSError`, and its queue is assigned to a `0xE000` in low memory.

Under normal operating conditions, one expects that no bus errors occur. This diagnostic tool can help to verify that.

There is a similar facility in the IRM code. In that case, the queue is very short, allowing for only 8 entries, and the only information logged consists of the address accessed and the program counter. One can easily see whether bus errors are occurring, because there is a counter that is incremented every time a record is written. This new facility should be better.