TO: MSPM Distribution
FROM: R.M. Graham
SUBJ: BB.5.03
DATE: 06/07/67

The attached revision of BB.5.03 further amplifies the handling of software simulated faults and adds an additional fault assignment.

Published: 06/07/67
(Supersedes: BB.5.03, 05/09/67,
BB.5.03, 11/08/66)

Identification

Software Simulated Faults
R.M. Graham, R.L. Rappaport

Purpose

By convention the segment number (777777)8 is reserved for use in its pairs for software simulated faults. The segment is a dummy and will never exist. Any attempt to reference it will result in a hardware fault. When this fault occurs the fault interceptor will then signal the condition "fault_n", where n is the address portion of the its which caused the fault. "Fault_n" will be signaled in a ring determined by the value of n. This convention provides an additional 256K faults. The first 128K are reserved for system use, while the remainder are available for user programs.

Discussion

Reference to segment number (777777)8 will result in one of two hardware faults: the segment fault or bounds fault. Whenever either of these faults are generated, the fault interceptor can, by a simple test, determine if the fault is actually one of the software simulated faults. If this is the case, the fault interceptor will call a procedure to determine the simulated fault number and the ring in which the condition should be signaled. This procedure, known as sim_fault (described below), is provided with a pointer to the faulting word pair and returns the fault number and the ring number in which to signal this condition. The fault interceptor then switches to this ring and signals this condition.

Sim_fault

Sim_fault is called by the fault interceptor upon recognition of a software simulated fault. The calling sequence is:

```
call sim_fault (p, fault_no, ring_no)
```

where p is a pointer to the faulting word pair, fault_no is the number of the simulated fault and ring_no is the ring in which this condition should be signaled. If ring_no is minus one (-1), the ring is the ring in which the fault occurred.

Sim_fault relies upon a table of simulated fault assignments to determine what action should be taken. Any faults not listed in the table are automatically signaled in the faulting ring. Below is the list of simulated fault assignments.

Simulated Fault Assignments

1. Used as the null pointer for PL/I pointer type data. This fault is signaled in the faulting ring.

2. Used in an argument list to indicate a null argument. This fault is signaled in the faulting ring.

3. Terminate Process Fault: this fault should be caused to occur when a process exhausts all avenues for recovery. This can be accomplished by,

```
     tra   k,*

     ...

k:   (777777)8   its
          3
```

   This fault is signaled in the hardcore ring.

4. This fault is used in conjunction with calls out of the hardcore ring. The fault occurs once in a given protection ring on the first callout to this ring from the hardcore ring. This fault is roughly analogous to a linkage fault in the hardcore ring. Since the linker does not have access to the hardcore ring, ordinary linkage faults in the hardcore ring are prevented by pre-linking at system initialization time and by the use of the hardcore ring callout mechanism (see Section BD.9.07) which depends on this fault. This fault is signaled in the hardcore ring.