PUBLISHED: 9/13/66

Identification.

Timer Runout Fault Handler
Chester Jones

Purpose.

This section describes the actions of the Fault Interceptor Module in response to the timer runout fault.

Discussion.

Each GE-645 processor contains an interval timer, a 24-bit register which counts down whenever the processor makes a memory access. When the interval timer counts to zero, the processor generates a timer runout fault and the interval timer continues counting into the negative numbers. The design of the Traffic Controller (Section BJ) requires that the interval timer produce an interrupt signal, rather than a fault, when it counts to zero. This interrupt signal, the time-out interrupt, competes with other interrupt signals on a priority basis for recognition by the processor containing the interval timer. (See Section BJ.9, Restart, for a discussion of how the time-out interrupt is handled.)

The timer runout fault handler is the procedure in the Fault Interceptor Module which transforms the timer runout fault into the time-out interrupt signal. The timer runout fault handler is shared by all processes running under the same version of Multics and is executed entirely in master mode with interrupts inhibited.

Actions of the Timer Runout Fault Handler.

When a Multics processor generates a timer runout fault, control automatically enters the timer runout fault handler which executes on behalf of the process that is running at the instant the fault occurs. The actions of the timer runout fault handler are as follows:

1. Temporarily stores the processor state in the Process Concealed Stack. (See Section BJ.1.05 for a description of the Process Concealed Stack.)

2. Obtains the processor index number (0-7) from the Processor Data Block. (See Section BK.1.02 for a description of the Processor Data Block.)

3. Uses the processor index number to obtain the appropriate pattern for setting the time-out interrupt cell for the processor on which it is executing. This pattern is found by using the processor index number as an index into the time-out pattern array of the Processor

Communication Table. (See Section BK.1.04 for a description of the Processor Communication Table.)

4. Sets up and executes a "set memory controller interrupt cell" instruction whose address points (indirectly, through the time-out pointer array) to the memory controller through which the time-out interrupt signal is sent to that processor.

5. Restores the processor state and returns control to the point at which the timer runout fault occurred.

Implementation Example.

The following machine code illustrates the initial implementation of the timer runout fault handler. In the example, pds stands for the segment number of the Process Data Segment (Section BJ.1.03), fault_int represents the segment number of the fault interceptor, pseg represents the segment number of the Processor Data Segment (Section BK.1.01), proc_index is a location in the Processor Data Block (Section BK.1.02) that contains the 3-bit processor index number, pct represents the segment number of the Processor Communication Table (Section BK.1.04), time_pattern stands for the base location of the time-out pattern array, and time_ptr stands for the base location of the time-out pointer array.

```
        rem     The following two instructions appear
        rem     in the processor fault vector.

        inhib   on
        scu     =its(pds,4,*),* control unit into Concealed Stack
        tra     =its(fault_int,timer),*   go to timer runout
                                          fault handler



timer:  staq    =its(pds,2,*),*   store A and Q temporarily
        ldq     =its(pseg,proc_index),*   get index number
        rem
        rem   The assumed format of proc_index is    zero   2*n,n
        rem
        lda     =its(pct,time_pattern,ql),*   get time-out pattern
        smic    =its(pct,time_ptr,qu*),*   generate interrupt signal
        ldaq    =its(pds,2,*),*     reload A and Q
        rcu     =its(pds,4,*),*   restore control unit
        inhib   off
```