Published: 12/29/67

Identification

Processor Interval Timer Fault Inhibition
J. H. Saltzer

Discussion

The 645 central processor contains a 24-bit interval timer which counts the number of memory fetches performed by the processor. The timer register counts down; upon reaching the value zero it triggers a fault and continues counting. Recognition of the timer runout fault is automatically inhibited by the processor, if it is operating in master mode. On the other hand, connect faults and interrupts are not automatically inhibited in master mode; the inhibit bit must be used to achieve the equivalent effect.

The unwanted inhibition of the timer runout fault has two effects:

1. Statistics gathering procedures using the timer runout fault will find statistics distorted more than necessary, since no master mode procedure will even be interrupted to take statistics. (This affects measurements of where the system spent its time.)

2. If a master procedure (or series of master procedures) retain control long enough for the 24-bit clock to cycle through more than once, only one fault will occur, and there is no way to determine how many times the clock has cycled. Time accounting based on the interval timer will be correspondingly affected.

Note that the second problem is significant primarily because the 24-bit timer register is being incremented at a much higher rate than in the 635 processor, and cycling occurs every few seconds.

An appropriate hardware modification would be to permit timer runout faults to occur in master as well as slave mode procedures, and control the occurrence of this fault, as well as the connect fault and all interrupts, with explicit use of the inhibit bit. Note that the processor hardware ignores the inhibit bit when operating in a slave mode procedure.