16.1 GENERAL

A Real Time Clock (RTC) is available in PTS systems, control of which is by means of a key switch on the system operator's panel. Key positions are as follows:-

- * OFF - RTC stopped.
- * ON RTC - RTC running.
- * LOCK - RTC running.

Once running, the RTC generates a signal timed from the mains power-supply frequency and it cannot be stopped by a program. The generated signal may be connected to any of the 8 highest priority interrupt levels (0-7) and is it can raise an interrupt every 20ms for 50Hz supplies. The associated interrupt must be cleared using the Reset Internal Interrupt (RIT) instruction and the RTC routine may be used as required within the system.

16.2 CLOCK ROUTINES

The 20ms RTC interrupts are used to update two clocks in the system.

The first clock routine counts five 20ms intervals and schedules the clock task #M every 100ms for dispatching purposes (level 49). After queueing the #M task, the second clock routine, Monitor Clock, is updated to count seconds. This monitor clock routine maintains the time, which can be set by the 'set time' instruction and obtained by the 'get time' instruction. A 24-hour carry is also generated.

When the #M task is dispatched, it updates all timer values in the timer queue pointed to by the word TIMQUE (figure 16.1).

When a timer value becomes zero, a branch to the 'timeout address' is performed. The blocks in the queue are not released.
When the timer value becomes positive, the blocks are released from the timer queue and returned to the monitor blockpool.

Drivers which need timeout functions prepare blocks in the timer queue via a special request. These blocks are then updated by task #M.

The timer queue is pointed to by the word TIMQUE. Two blocks, each of 3 words, are reserved for each timer. The first word points to the next pair of blocks in the queue.

The second pointer points to the block containing the 'timeout address' and two parameters. The third word contains the timeout value (negative). When this value reaches zero, control is passed to the 'timeout address' in the second block. A positive timeout value results in the release of these two blocks.

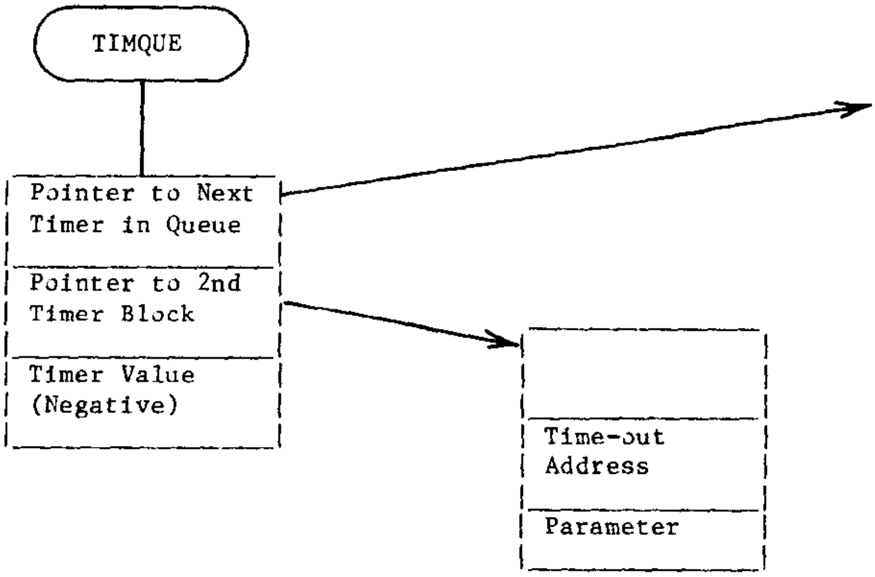

Figure 16.1. Timer Queue.