PUBLISHED: 6/02/66

Identification.

The Processor Communication Table
Chester Jones

Purpose.

The Processor Communication Table, a system-wide data base that is shared by all processes running under the same version of multics, is used to generate inter-processor communication signals. When a multics system is initialized, the information a process must have in order to direct a connect or an interrupt signal to any processor in the system is collected and stored in the Processor Communication Table. (See Section BL for a discussion of System Initialization and Section BC.cj for a discussion of the hardware requirements for inter-processor communication signals.) The information in the Processor Communication Table is arranged so that by knowing only the number (0-7) of the "target" processor, a process may direct a connect or an interrupt signal to any processor in the system. The Processor Communication Table is "wired" into core memory and cannot be paged onto secondary storage.

Contents of the Processor Communication Table.

The information contained in the Processor Communication Table is divided into three parts as follows:

1. SMIC-Patterns. The Processor Communication Table contains three arrays of eight 36-bit patterns each which are used with the "set memory controller interrupt cells" instruction to produce the internal interrupts required by the Traffic Controller (Section BJ). It is important to note that within a given array, all eight entries may or may not contain the same pattern.

   a. Time-out pattern array. This array contains eight patterns for producing time-out interrupts. The nth entry contains the pattern for producing the time-out interrupt for processor number n.

   b. Pre-emption pattern array. This array contains eight patterns for producing pre-emption interrupts. The nth entry contains the pattern for producing the pre-emption interrupt for processor number n.

   c. Quit pattern array. This array contains eight patterns for producing quit interrupts. The nth entry contains the pattern for producing the quit interrupt for processor number n.

2. SMIC-Pointers. The Processor Communication Table contains three arrays of eight 72-bit ITS pointers each

which are used with the "set memory controller interrupt cells" instruction to produce the internal interrupts required by the Traffic Controller.

a. Time-out pointer array. The n<u>th</u> entry contains an ITS-pointer which points (indirectly, through a segment descriptor word) to the memory controller through which the time-out interrupt signal is sent to processor number n.

b. Pre-emption pointer array. The n<u>th</u> entry contains an ITS-pointer which points (indirectly, through a segment descriptor word) to the memory controller through which the pre-emption interrupt signal is sent to processor number n.

c. Quit pointer array. The n<u>th</u> entry contains an ITS-pointer which points (indirectly, through a segment descriptor word) to the memory controller through which the quit interrupt signal is sent to processor number n.

3. Connect data. The Processor Communication Table contains two arrays of eight 36-bit entries each, which are used with the "connect" instruction to produce inter-processor connect signals.

a. Connect operand word array. The n<u>th</u> entry contains a connect operand word (cow) for processor number n. A cow for processor n contains in its three low-order bits the number (0-7) of the module port through which processor number n interfaces with memory.

b. Connect flag array. The n<u>th</u> entry serves as a signal between processor number n and another processor. (See Section BK.3.08 for a discussion of how the connect flag array is used.)

<u>Examples of Processor Communication Table Usage</u>.

The following two examples are provided to indicate how the Processor Communication Table is used. In the two examples, pct stands for the segment number of the Processor Communication Table, quit_pattern stands for the base of the quit pattern array, quit_ptr stands for the base of the quit pointer array, flag stands for the base location of the connect flag array, and cow stands for the base location of the connect operand word array. The machine code is written in BSA source language. (See Section BE.7.02, BSA.)

In order to send a quit interrupt signal from processor number m to processor number n (for all m and n, 0 to 7, inclusive), the process in execution on processor number m performs the following steps:

```
          ldq      2*n,dl             prepare index into the Processor
          ldx1     n,du               Communication Table arrays
          lda      =its (pct,quit_pattern,1),*   get quit pattern
          smic     =its (pct,quit_ptr,ql*),*   generate interrupt
```

In order to send a connect signal from processor number m to processor number n (for all m and n, 0-7, inclusive), the process in execution on processor number m performs the following steps:

```
          lda      n,dl               prepare index into connect tables
          stc1     =its (pct,flag,al),*      set flag to non-zero
          cioc     =its (pct,cow,al),*       issue connect
          szn      =its (pct,flag,al),*      wait for response
          tnz      *-1                        from processor n
```