NATIONAL RADIO ASTRONOMY OBSERVATORY
Green Bank, West Virginia

300-FOOT CONTROL COMPUTER MEMO NO. 18

CHARACTERISTICS OF THE SIGNAL AVERAGER FROM THE POINT OF VIEW OF THE TELESCOPE CONTROL COMPUTER

Mark H. Clark

Aug. 30, 1985

CHARACTERISTICS OF THE SIGNAL AVERAGER FROM THE POINT OF VIEW OF THE TELESCOPE CONTROL COMPUTER

DESCRIPTION

All communication with the Signal Averager (SA) is from the SA to the control computer. The computer is not able to query, set, or in any way affect the operation of the SA. The SA provides information to the control computer at two times--at the beginning and end of an integration. Both consist of an initial header word (see Figure 1). The headers consists of two pieces of information: 1) whether it is a start or end of integration header, and 2) whether the integration contains 4096 (full), 2048 (half), or 1024 (quarter) 16-bit words. A intra-integration and the inter-integration periods are indeterminate. The data dump begins immediately following the end-of-integration header. The size of the data dump is determined by the size bit (full, half, or quarter) in the preceding headers. Note that the last word sent in the data dump is the number of sweeps performed in the integration.

OPERATION

Figure 2 is a time line of the operation of the SA. It also contains the proposed operations of the DA/CP and UNIX systems for handling the SA data through the PI16 Parallel Digital I/O Port. The interface between the SA and the DA/CP has already been described.

The DA/CP will operate in two modes of array transfers since data sizes vary; one for receiving the start header and another for receiving the end header plus data. The start-AST signals the reception of two words: the header and the current LST counter. The end-AST signals the reception of either 4098, 2050, or 1026 words: the header, current LST counter, plus the data. The immediate passing of the start-AST allows the completion routine to set up the proper buffer size for the data transfer, but more importantly it allows the proper LST time and position to be calculated for the upcoming data dump. The completion routine for the end-AST reinitializes the DA/CP for two-word buffers and fills the tape buffer from the contents of the data buffer and header information (including the previously obtained time and position parameters). The pseudo code for the initialization, termination, and task completion routines follows.

```
Initialization:
        Fill generic tape tape header buffer (except for
                times, and positions)
        Open tape drive
        Open parallel port
        Set parallel port mode
        Set observe flag
        Start transfer for 2 words
        Define task completion routine on event


Termination:
        Clear observe flag
        Close parallel port
        Flush tape buffer
        Close tape drive


TCR:
        if not legal header
                Error
        else if start integration header and observe flag set
                Save LST counter
                Start transfer for 2 words plus data size
                Define task completion routine on event (?)
        else if end integration header and observe flag set
                Start transfer for 2 words
                Define task completion routine on event (?)
                Copy generic header to tape buffer
                Compute time and positions and copy to tape buffer
                Copy data buffer to tape buffer
                Start asynchronous tape write
        else
                Error
```

A possible problem with the above algorithm is the immediate reception of a start-AST after the end-AST. We must complete two or three steps (depending if the task completion routine must be re-called on every iteration) before the system is ready to start a new integration. An alternate approach is to receive one AST at the end of an integration which signals a buffer with the same contents* of both buffers in the above implementation. The draw back

---

* Minus the second buffer's LST counter, since the DA/CP only places one counter word in each buffer it fills.

is the user must inform the system the required data buffer size, since it is not able to pick it up from the first header.

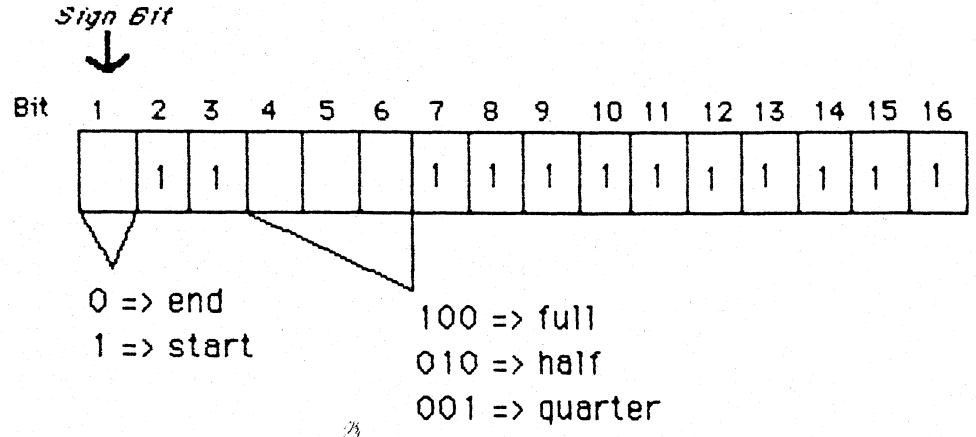

Figure 1

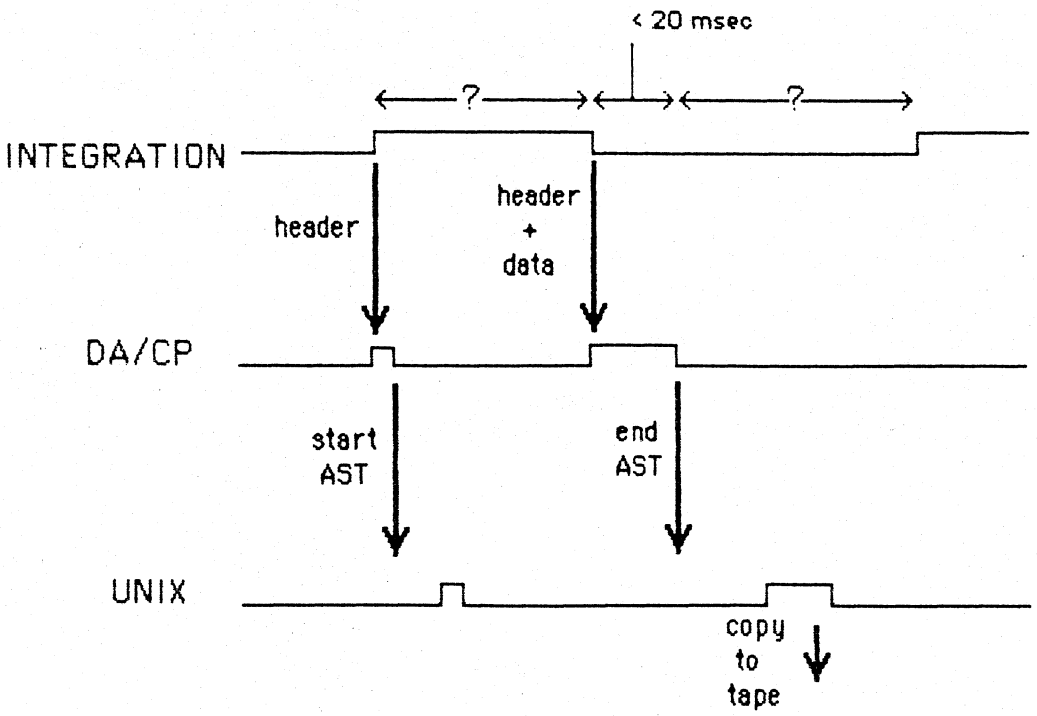

Figure 2