APRIL 1979 INT 93/79

DATA TRANSFER AND STORAGE ON T C A

R. Schreiber, J.B. Lister

Introduction

The part of the data acquisition system discussed in this report is the transfer of raw data from the TCA Tokamak experiment into the dedicated PDP 11/60 computer and its permanent storage for subsequent analysis.

The main philosophy chosen for TCA is to perform all data capture in local intelligent modules with their own memory. These modules are all controlled by microprocessors. This philosophy greatly simplifies the subsequent transfer of data to the PDP since there is no real-time control or interaction with the peripherals which acquire the data autonomously.

Initially it was hoped to use the GALE software - Generalized Acquisition for Laboratory Experiments - from Garching (Ref 1) which would have more than met our needs. However, this would not have taken advantage of our main features of a) a standardized communication interface (known as RS232C) and b) no real time work. As a result of these considerations it was agreed to write our own acquisition software which would normally be an extremely time-consuming process. The factors mentioned above have, however, allowed us to write a sufficiently flexible package, reasonably quickly. It should not be forgotten, however, that a long study of the GALE system has greatly simplified the organisation of the software. In

the following we discuss the basic layout and then the components of the software in detail. A complete description of the individual routines is not presented as these are adequately documented already (Ref. 2).

Description

A simplified layout of the data transfer system is shown in Figure 1. The software operates as with three distinct components; the TCADLG file which decides what the acquisition program should do; the TCAACQ task which performs the actual acquisition and storage; the TCADLG task which controls the TCADLG file.

The TCAACQ task is normally permanently running in the PDP, waiting for a Tokamak shot to take place. On receipt of a trigger it acquires the data from the microprocessors.

The TCADLG file controls this acquisition procedure, telling the program which microprocessors to read in, where to read them and where to store the data. The advantage of this is that the acquisition can be changed immediately by running the TCADLG task at a computer terminal. In this way the actual acquisition task is modified very rarely, which requires time and knowledge of its precise functioning.

The data acquired by the TCAACQ task are stored on disk in the SHT file for subsequent analysis. At the end of a shot, TCAACQ runs other tasks which can perform some automatic analysis on the data.

Finally, there are utility programs for the transfer of data from disk to magnetic tape, from tape back to disk, and for cleaning up the disks. We discuss now in turn the TCADLG file, the SHT data file, the TCAACQ task the TCADLG program. Finally retrieval of the data is mentioned, and its mass storage.

The TCADLG File

The TCADLG file is a small disk file in direct-access format, composed of several 64-Byte logical records. The first of these blocks is the Header Block and each subsequent block is called a Microprocessor Bookkeeping Block (MBB).

a) The Header Block

The structure of this block is shown in Table I.

BYTE	LENGTH	NAME	SET BY	CONTENT
1	1	YIVERS	ACQ	VERSION #
3	2	YITIM	ACQ	TIME
5	3	YIDAT	ACQ	DATE
9	2	ISHACQ	ACQ	ACQ SHOT #
11	2	ISHMC	ACQ	M/C SHOT #
13	1	YNBLKF	ACQ	NB OF BLKS IN TCA____.SHT
14	1	YNPROC	ACQ	NB OF ACTIVE MPS
15	16	YMPROC	ACQ/DLG	POINTERS TO MBB
41	24	MCPRM	DLG	M/C PARAMETERS

- The version number is the version number of the TCAACQ software in use.
- Time and date are the values at the last shot on which TCAACQ was run.
- Acquisition shot is the last transferred shot number.
- Machine shot is the number of the machine cycle at the last transferred shot.
- The number of blocks is the total number of 512-Byte blocks in the SHT data file for the last transferred shot.
- The number of active Microprocessors is the number declared to be online for the previous shot.

- The MBB pointers are the "mini-block" numbers of the Microprocessor Bookkeeping Blocks in the last SHT file (see description of SHT file). If it is negative, that microprocessor is to be read out before the next shot; if positive, then after the shot. If zero, that microprocessor is declared to be offline.

- Machine Parameters is available space to be filled by data transferred from the control system (e.g. fields, voltages, pressures, etc.). Each microprocessor known to the system (at present, up to 16) has a Microprocessor Bookkeeping Block in the TCADLG file. Its structure is shown in Table II where

BYTE	LENGTH	NAME	SET BY	CONTENT
1	1	YID	SOURCE	MICROPROCESSOR #
2	1	YITYPE	DLG	TYPE
3	8	NAME	DLG	NAME
11	1	YITT	DLG	TT CHANNEL
12	1	YMDBLK	ACQ	MDB BLOCK POINTER
13	2	MDBYT	ACQ	MDB LENGTH (BYTES)
15	1	YMDLEN	ACQ	MDB LENGTH (BLOCKS)
23	1	YNHBYT	DLG	HEADER BYTES IN MDB
24	1			
25	40	IPARM	DLG	PARAMETER BLOCK

- The MP number corresponds to the number of a physical box which communicates with the computer.

- The type is a code for the box-function,
  e.g. Type = 0 is a simple data block with no extra information
  Type = 1 is an ADC box microprocessor
  Type = 2 is a CAMAC crate microprocessor

- The name is for ease of reference.

- TT Channel is the physical socket in the PDP multiplexer module (DZ11), from 1 to 20 (OCTAL) into which this microprocessor is plugged.

- MDB (Microprocessor Data Block) Pointer is the pointer to the first block in the SHT file in which the data from this microprocessor were stored in the last shot.
- MDB length (bytes) is the total data-block length in bytes for this microprocessor.
- MDB length (blocks) is the number of 512-byte blocks used for this microprocessor.
- Header bytes is the number of bytes in the data block before beginning the raw data. This allows the microprocessor to transfer some extra information.
- The parameter block contents vary with the microprocessor type and are detailed in the reference manual.

The SHT data file

The data from a Tokamak shot (e.g.11064) are stored in a file named by its shot number (e.g. TCA11064.SHT; 1) which is only opened for WRITE operations by the TCAACQ task. The file is a direct access file of blocks of 512 bytes (such that a physical disk block is equal to a logical Fortran block). The first one or two blocks contain a compressed version of the TCADLG file in which the Microprocessor Bookkeeping Blocks of non-active microprocessors are removed. For 12 active microprocessors, block No. 1 contains the Header Block plus 7 MBB's, and block No. 2 contains the Header Block plus 5 MBB's (as an example).

There is one further distinction between the various microprocessors. Some are read out just before the shot after receipt of TRIGGER-1 and are stored first in the SHT file. The others are read out after TRIGGER-2 is received.

Any block which is not full, that is, if the number of bytes transferred is not a multiple of 512, is completed with zeros.

The TCAACQ Program

A flow chart of the task TCAACQ is shown in Fig. 2. On activation of the task the program is initialised during which the user is asked whether to run AUTO or MANUAL. The latter asks whether or not to continue, following each shot. The free disk-space is checked to ensure that there is sufficient for the following shot. Messages during the running of TCAACQ are sent to a device specified by the user, either the user terminal, the line-printer or a file. The program then waits for the TRIGGER-1 which announces that a shot is on the way.

On receipt of this trigger, the program inhibits all changes to the TCADLG file so that while the acquisition is underway, none of the parameters may be changed.

It then disables the checkpointing (rollout) of itself, assuring top speed of response. It opens the TCADLG file and updates the header (e.g. with the Time, Shot number, etc.). The SHT data file is created ready for data, and the active microprocessors are assigned a logical unit number in the task.

The "before shot ready" microprocessors are then read out sequentially and the data stored in the SHT file. The actual data transfer from each microprocessor is extremely simple. It consists of a one word read transfer which contains the number of subsequent byte transfers, followed by a read transfer of all the following bytes, in blocks of 128 bytes. It is thus completely independent of the nature of the data transferred and the MP type.

In between the block transfers control words are exchanged to request more data or repeat the data. After each 128 byte transfer, a checksum is transmitted to the PDP and the block is accepted or rejected. It is hoped that this transfer can operate at 9600 Baud (~600 words/sec). {Now tested!}

The program now waits for TRIGGER-2 which is normally sent at the end of the Tokamak pulse. All remaining active microprocessors can then be read out. This read operation is performed exactly as the previously described transfer, except that the microprocessors are read out simultaneously to save total time to transfer all the data. In the RSX 11-M operating system which we have in the PDP, this can be done using FORTRAN - a considerable advantage.

Once all microprocessors have successfully completed their data transfer, the TCADLG file is updated with all the new block pointers and word counts. the contents of the TCADLG file are transferred, where applicable, to the SHT file which is then closed, as is the TCADLG file. The TCADLG file is then enabled for other access and checkpointing of the TCAACQ task is enabled. Other tasks can now run in the PDP.

A task called ACQRUN is then started automatically by the TCAACQ task. This is a tiny few-line task which only activates other tasks for automatic post-shot analysis. ACQRUN is only a way of doing this without having to change the TCAACQ task itself.

Finally we check that the amount of contiguous free disk-space is sufficient for another shot and either wait for TRIGGER-1 (in AUTO) or ask whether or not to continue (in MANUAL).

The TCADLG task

This task controls the TCADLG file which is responsible for the running of the acquisition task. It is an interactive task which will normally be run at a hard-copy terminal for a permanent record. On entering the task the Header information is presented, and the user is asked what he wishes to change. When no more changes are to be made, all the changes are checked by the task for self-consistency and if they fail the procedure is restarted.

Each Microprocessor Bookkeeping Block which passed the checks is written to a second file called TCADLG.SAV so that the risk of losing the TCADLG.DLG file is greatly reduced. Finally the user is asked if he waits a new print-out of the TCADLG file to his terminal.

Each time a new microprocessor type is introduced to our system, we must modify the TCADLG task which is admittedly a lot clumsier than the GALE functioning. However, we only need to modify slightly the routines MPBB which contains the types of MP, MPPARM which contains the charge of parameter dialogue, and PRPARM which types out the parameter names. Since this will be performed so rarely, we consider it to be no great loss.

Saving of Data

Since only a limited amount of data can be stored on disk at present (more in 1980 we hope) the back-up storage on tape must be quite flexible. A single command can store on tape all shots not previously stored. This can be run automatically by ACQRUN, giving us shot-by-shot back-up, or it can be done when the disk is full. When full, the disk can be selectively cleared by an interactive program. This allows us to keep some shots and delete others of no interest, but only if they have been saved. Shots stored on tape can be recovered from tape and put back on disk by the same interactive task.

Retrievel of Data

The retrieval of data can only be performed from disk-files since the SHT file is a direct-access file (disk only). The structure of the SHT file and its header is transparent to the user who simply needs to make one Fortran call to get a shot into his program, and all the bookkeeping is automatically prepared. In order to access a block of microprocessor data a simple call is available. The most usual request will be for an ADC trace, for which a call specifying simply the signal cable number is

available. These and other Data Retrieval And Treatment facilities will be described elsewhere (Ref. 3).

Summary

The complete TCA Data Transfer system has been written as our idea of the compromise between flexibility and simplicity. The whole process can be run by a physicist who understands the hardware but knows no programming. The total use of microprocessor controlled acquisition equipment allows the transfer to be very simple, and a standard communication between microprocessor and computer allows all the transfer to be carried out by available Fortran calls.

References

1. GALE Terminal Users Guide - R. Lathe et al. IPP R/23.

2. The TCA Acquisition Reference Manual - R. Schreiber.

3. DRAT routines reference manual - J.B. Lister

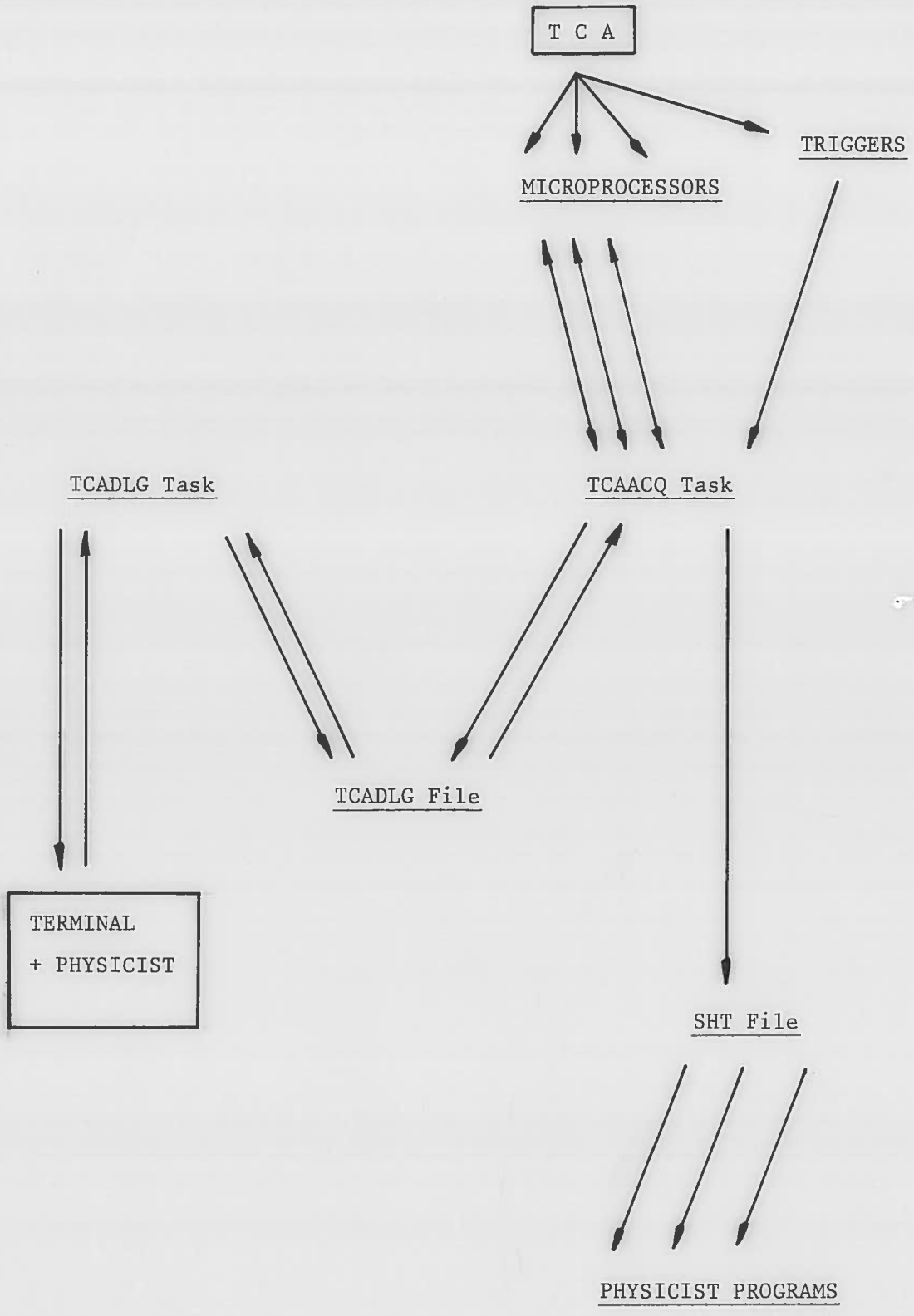

FIG. 1 General Layout

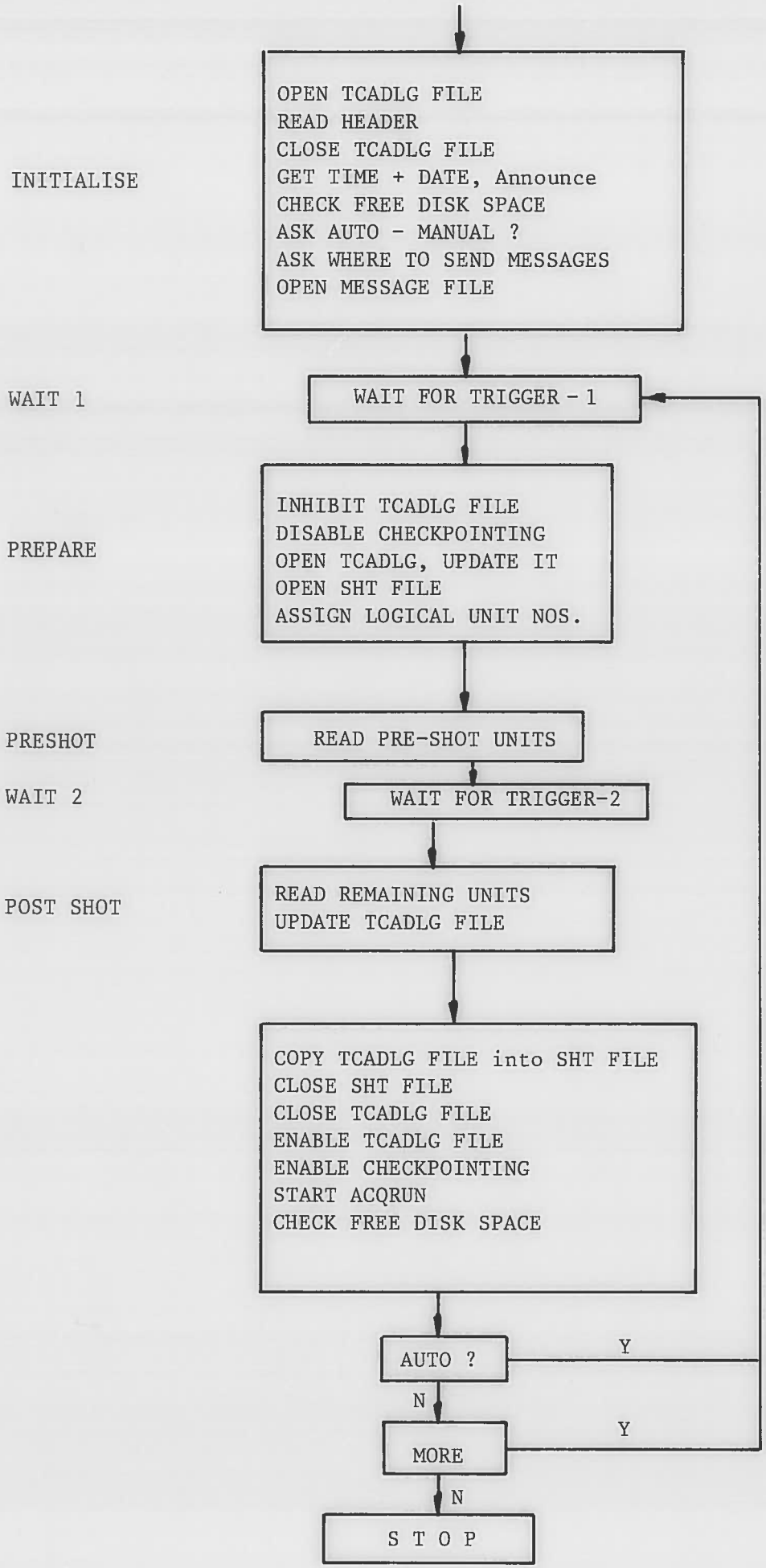

FIG. 2 TCAACQ FLOW CHART