

MUON STRING SYSTEM SOFTWARE STATUS

Although the Terak - IS11 system software is by no means complete, enough of the pieces are working so that we can lay out the organization of the applications program in the IS11 computer. In this report, I will describe where we are in terms of software presently and propose a software structure for the IS11 applications program. Some useful information for people who want to use the system is given in Appendices A-C.

I.) Present Status

First, let me summarize the present status:

A.) Early last Summer (see DUMAND note 81-13, June 81), we were able to download programs from the Terak to the IS11 for stand alone running on the IS11.

B.) At the beginning of the Summer, Bob March brought RTMULTI from Fermilab. This system is composed of two parts: the data acquisition part (DA) and the data analysis and control part (MULTI). The first part (DA) is specialized to the Fermilab hardware (magtape, Jorway branch driver, Bison Box, etc.) and is not directly of use to us. The second part (MULTI) is quite general and was brought up by Bob last Summer.

C.) During the Spring and Summer, I worked on the communications package (and protocol) for the communications between the IS11 and the Terak via the Computrol modems. Presently, this package allows the sending of commands (bidirectional), alarm messages (downstairs to upstairs), and buffered data (down to up). In the future, I hope to allow exchange of data in common blocks and to allow access to the Terak disk by the IS11. More will be given below about the subroutine calls to use these features.

D.) This Fall, I have greatly modified (with much advice from Dave Ritchie) many of the DA routines. Routines rewritten for our application are: BHRUN, COUSER, ENRUN, IUSER, MULPOL, POPROC, READEV, RSRUN, STUSER, SURUN. Many of these routines had been dummed out. Now we have essentially a complete system (DA plus MULTI). We may begin a data taking run, writing a begin run record on disk (if LOG=YES) and notifying the IS11 that it may begin sending data over the Computrol. The IS11 packs data into buffers (double buffered), and, when there is not enough room for the next event, sends the buffer to the Terak. The data is then written to disk (if LOG=YES), and the data is handed to MULTI, if MULTI isn't busy. Once the data is handed to MULTI, the full facilities of MULTI are available. MULTI unpacks events from the buffer and can be used to print the data or make plots using the data. For more information, see the MULTI manual.

Alarm messages are printed by MULTI when received. The modified routines also allow us to end runs (writing an end run record to disk), suspend runs, resume runs, stop MULTI, etc. It is also possible to set DATA-QBI and read back data off the disk and analyze it. The RTMULTI system on the Terak is essentially complete.

E.) Finally, I have tested the Fortran callable CAMAC library supplied by Interface Standards for the IS11. It is possible to generate LAM's using the Visual Display module and have control pass to an interrupt service routine. This is necessary to be able to read out muon coincidences. For a listing of the routines available, see the IS11/Fortran Users Manual from Interface Standards. Note that the routine CCLM gives an undefined global diagnostic mes-

sage when linking.

Users should note that CAMAC functions are carried out in this system using subroutine CFSA and not ICAM.

II.) Proposed Applications Program Structure

Logic block diagrams for the downstairs program, generalized from its present form, are shown in Figs. 1-3. Access to CAMAC occurs during a "run". When a run begins, the necessary CAMAC initialization is done for the type of data to be taken. CAMAC is then turned on, and "events" are packed into data buffers and sent to the Terak. Events may be interrupt driven (serviced by MULAM, WFDLAM, etc.) or non-interrupt driven (serviced by DORUN). Linkage to the interrupt service routines is handled during CAMAC initialization at the beginning of the run. Data sent to the Terak may be raw or digested. The type of data taken is controlled by the runtime, which is set by MULTI. CAMAC is turned off when a run is suspended or ended, stopping the flow of events.

If this structure is adequate, then I will set it up with dummy subroutines.

III.) Proposed CAMAC Common Structure

One of the problems that usually occurs is that the CAMAC configuration is changed after many routines have been written, and it is necessary to recompile and relink everything. To make this easy, I propose that we use the following common block structure to define the CAMAC modules:

```
COMMON/CAMCOM/MAXCRT,CRTNM(2),MAXMOD,CRATE(40),STAT(40)
INTEGER MAXCRT,CRTNM,MAXMOD,CRATE,STAT
DATA MAXCRT/2/,CRTNM/"164000","166000"/,MAXMOD/40/
DATA CRATE/10*"16400",...
DATA STAT/2,6,2,.....
COMMON/CANDES/ADC1,WFDIG,....
INTEGER ADC1,WFDIG,...
DATA WFDIG/1/,ADC1/2/,...
```

In order to refer to a particular module, you would use the index corresponding to it:

```
ICRT=CRATE(WFDIG)
```

If necessary, the common block assignments can be changed, and the subroutines can be recompiled and relinked without too much trouble. MULTI has a facility to dump common blocks into fortran routines (like Patchy) so we don't all have to update the block each time.

APPENDIX A

Subroutine Calls for Message Transmission from IS11.

The following call is used to transmit alarm messages:

CALL WRTALM(ALMCOD,ARRAY,NWORD) , NWORD<11

For ALMCOD=1, a formatted message is transmitted in ARRAY, and MULTI just prints it. For other codes, MULTI formats the message using the numerical data passed to it in ARRAY (MULTI must know the format).

In order to buffer data, the following call is used:

CALL WRTDAT(ARRAY,NWORD,EVTYP)

This routine takes data out of ARRAY and packs it into the data buffer. The user must have already read the data into ARRAY. The MULTI header is written into the buffer along with the data. Events are packed into the buffer until there is not enough room available for the next complete event. Then the buffer is written to the Terak. The maximum amount of data that can be written is 255-3=252 words. Be careful, since no provision presently exists for sending partial events. For instance, if you write 129 words each time, the buffers will be only half full!

The interrupt service routine calls a slightly lower level routine:

ADDRES=BUFPTR(NWORDS,EVTYP)

This routine returns the starting address where data is to be written in the buffer, so that only one transfer has to be done. The same rules apply concerning the amount of data.

APPENDIX B

Diskette Usage

The present system uses:

A.)MULTI diskettes (set of 5). Contains MULTI system disk, all of MULTI system, DUMAND routines, etc. Used to create MULTI.SAV.

B.)IS11 program diskettes (set of 2). System disk and user disk (FAH). In the future, we can probably make the MULTI system disk and the IS11 system disk identical. These are used to create the program to be downloaded into the IS11, RWDOWN.

C.)Runtime disk - contains monitor, MULTI.SAV, RWDOWN.SAV, and various MULTI runtime files. To run, this disk must be on QB0:. If you want to write data to disk, place a formatted and initied disk on QB1:.

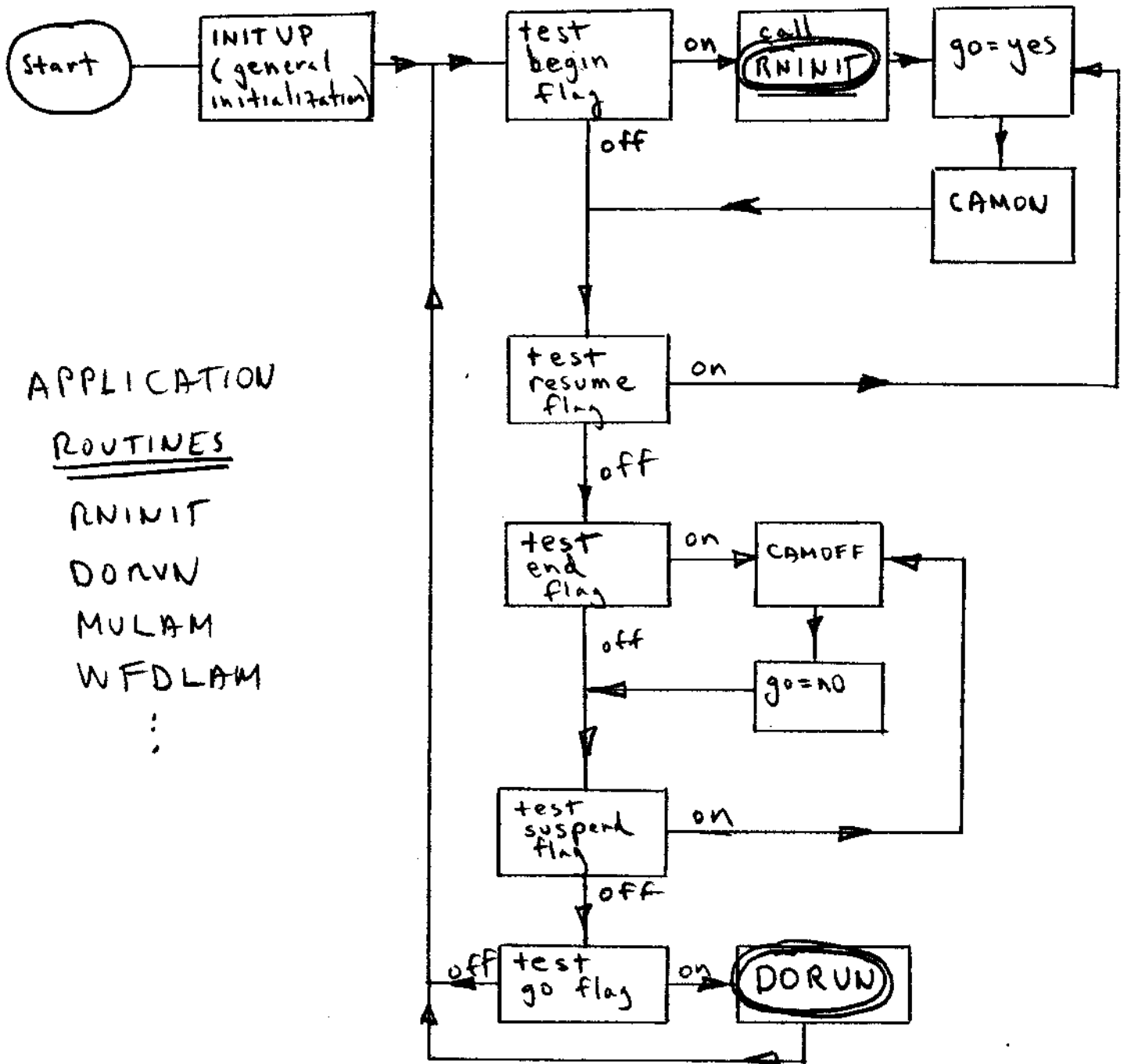
APPENDIX C

IS11

Harry has installed a ROM which loads the bootstrap
When powered up, the IS11 is ready to be downloaded. Normal
switch positions (now):

- A.)clock on
- B.)CAMAC on.

DNMAIN



APPLICATION

ROUTINES

RNINIT

DORUN

MULAM

WFDLAM

:

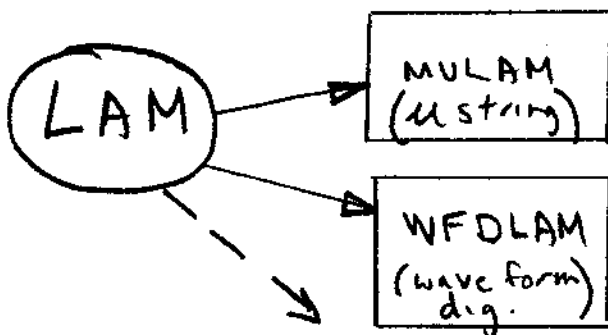


FIG. 1

RNINIT

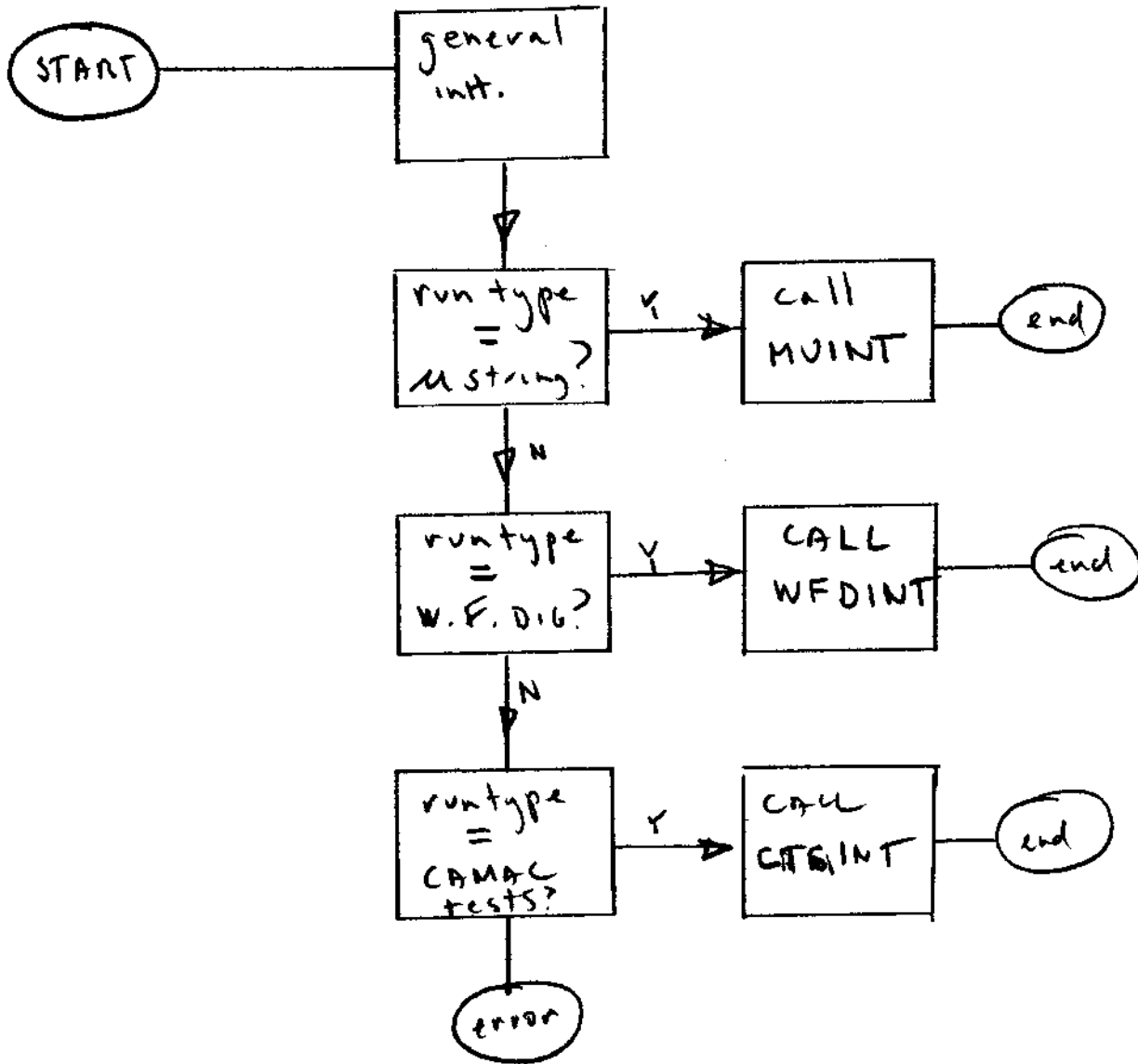


Fig 2

DORUN

