MSUCL-34

The Data Acquisition Task
TOOTSIE*

Douglas L. Bayer

May 1971

CYCLOTRON LABORATORY
MICHIGAN STATE UNIVERSITY

*SUPPORTED BY THE U.S. NATIONAL SCIENCE FOUNDATION

TABLE OF CONTENTS

I. Introduction

The data acquisition task TOOTSIE, operating under the JANUS time-sharing supervisor, has been designed to process most of the two and three parameter data requiring some form of event identification taken by users of the Michigan State University Cyclotron. A variety of options provide for various calculations to be performed on the data before being stored. These calculations will be discussed in the interrupt routine section of this report.

The program operates in two modes, SETUP and RUN. In SETUP mode the data is stored in a two dimensional array of dimensions (64,64), (64,256), (128,128), or (256,64). The data are displayed on a Tektronix 611 storage scope, appearing as a series of curved bands across the screen. Polynomial fits to a set of points obtained by accepting the coordinates of a movable cross define the lower and upper bound for each band.

Up to eight independent detector systems may be used in a given experiment. Three routing bits associated with each event provide the necessary detector identification. In SETUP mode data from one of the detector systems is stored (all other being discarded), permitting the bands for each detector to be set.

When all bands have been set, the program is switched into RUN mode via a teletype command. Tables of 256 points per fit are generated and stored for each detector. Events are then checked against the tables corresponding to the detector indicated by the routing bits and the appropriate channel of the spectrum in which the match was found is incremented. The spectra range from 512 to 8192 channels.

II. Task Structure

2.1 General Structure

a) Overview of Core Layout

The task can be divided into three classes based on the state of the SLAVE and MAP bits in the program status double-work (PSD). These three classes are defined

1) Master Mode Unmapped: This corresponds to both the SLAVE and MAP bits off. Because code falling in this class is independent of the memory map, it is reserved for interrupt servicing routines. The system requires that this code be dedicated (locked into core). The routines in TOOTSIE which fall into this class are the data taking routines, scope switch reading routine, and the magnetic tape servicing routines.

2) Master Mode Mapped: This corresponds to the SLAVE bit off and the MAP bit on. Because this code is not subject to the access protection associated with the memory map, all pages referenced by this code must be "virtually dedicated", that is in core every time the task gets a time slice. The sections of the task which operate master mode mapped are those which communicate with JANUS and the trap servicing routines. This set of routines comprise what is called the "monitor".

3) Slave Mode Mapped: This corresponds to both the SLAVE and MAP bits on. Code and storage in this class are brought into core as they are needed and remain in core only as long as they are used. All calculations are done in this mode.

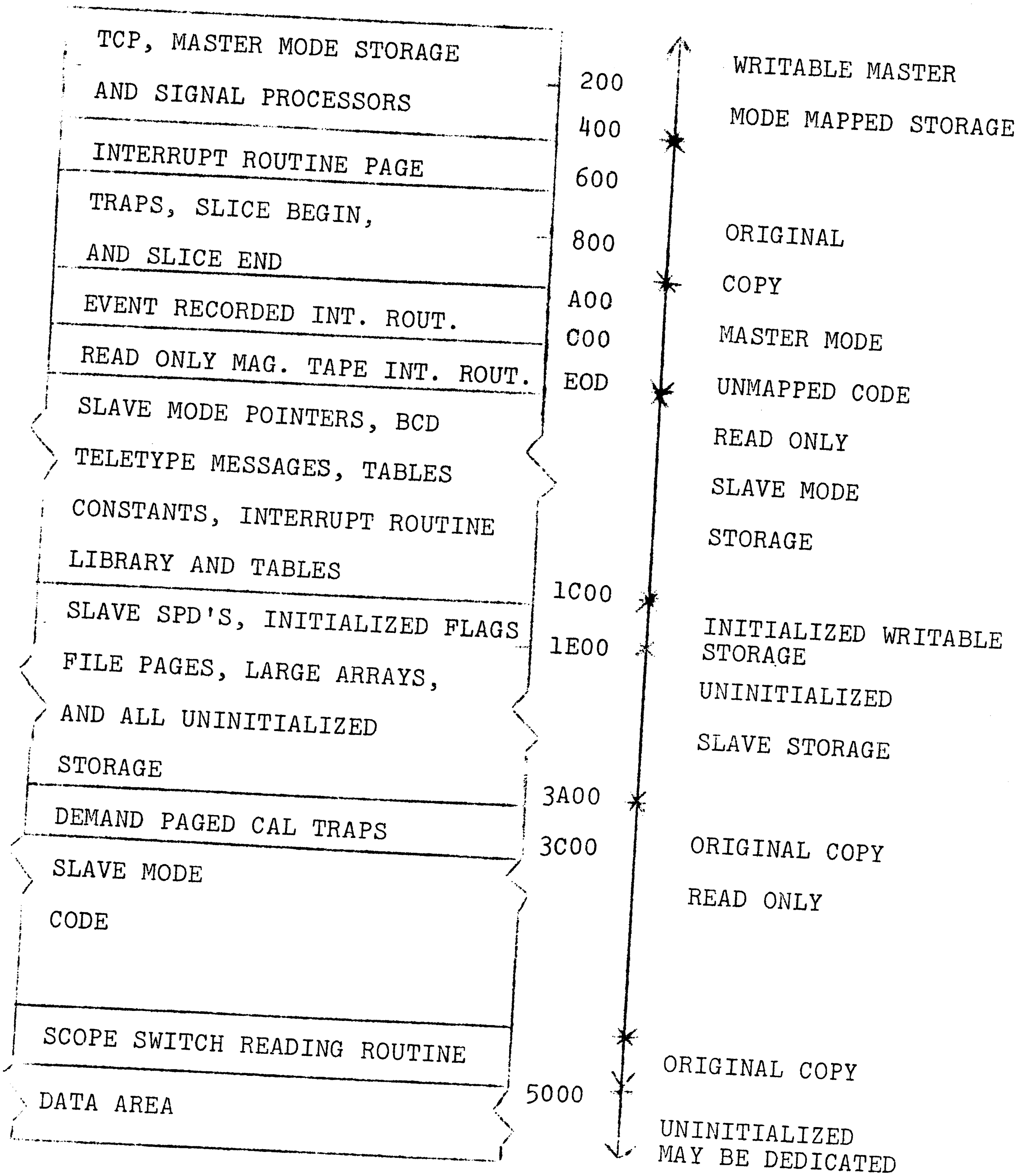
The basic unit of storage is a PAGE defined as 512 and 32-bit words of core storage. The organization of the task in page units is illustrated in Figure 1. The first two pages of the task contain all the writable and read only storage used by the monitor. The signal processors which fit in the remaining space are also included. The third page of the task is common to all copies which are currently running. When the task is operating in a data acquisition mode, this page is dedicated and provides two 150 word blocks for loading interrupt routines as well as sets of tables and flags required in interrupt routine operation. The remainder of this page and the next two pages contain all of the most often used monitor code. The remaining page of the monitor code in located in the slave mode section. This page is forced into core by the x'40' trap via an indirect reference to the page by the CAL instruction. Once in core the page is virtually dedicated until control is returned to slave mode.

The next two pages contain the event recorder tape servicing routine and write only tape servicing routine. These pages are dedicated when the tape drives are used by the task.

The next six pages contain all the read only storage for the slave mode section of the task.. The major part of this storage is the interrupt routine library. This library contains all the interrupt routines along with their initializing routines. The library structure will be discussed later.

FIGURE I

Block Diagram of Task



The next two pages contain all the slave mode writable storage which must be initialized i.e. flags, stack pointer doublewords, etc. The next fourteen pages from X'1E00' to X'3A00' contain the writable slave mode storage which need not be initialized. This storage includes large arrays required for the polynomial fitting routine, space for output file generation, etc. The next 11 pages from X'3A00' to X'5000' contain the demand paged part of the monitor and all the slave mode code. The last page of the task is the scope interrupt routine. This page is common to all copies of the task. It contains four sets of scope parameters and is thus capable of drivinga scope for each of four independent copies of the task. Each copy of the task places one level of dedication on the page when a scope is connected and removes one level when the scope is returned to the system.

b) Time-Slice Utilization

The time-slice allocated to the task has been subdivided into two intervals. The first part is two clock ticks long. This part of the time-slice is used to process all signals and start MOIRAE and NDPLOT subtasks.

The second part of the time-slice is used for all other functions of the task.

2.2 CAL TRAPS

All communication between the slave mode and the monitor is accomplished through the CAL traps. The conventions adopted for the register usage and communcation are:

1) Registers R6-R11 are volatile
2) Slave mode can transmit information to the monitor by three methods:
    1) General registers R6-R11
    2) By the effective work address of the CAL instruction. In this case the slave mode section must ensure the effective work being in core by referencing it indirectly.
    3) Predefined arrays in virtually dedicated storage.
3) CAL traps can return information to slave mode by two methods:
    1) Condition Codes:
        a) CC4=1 implies success
        b) CC4=0 implies failure
    2) Predefined arrays in virtually dedicated storage.

The task uses the CAL1 and CAL3 traps. All I/O operators are carried out through the CAL1 traps and all special functions, such as write directs, are carried out through the CAL3 traps.

2.2.a) CAL1 TRAPS

1) Disk File Generation: CAL1, 0

This trap provides the service of opening a disk file, closing and releasing a file to either a symbiont or subtask, and providing disk pages for file generation. The function of each CAL is transmitted through the effective work address of the CAL instruction. The defined codes are listed in TABLE I.

Disk page addresses are transmitted to the slave mode section of the task through the virtually dedicated push stack, FILESTK. This stack will contain from one to four disk page addresses when control is returned depending on the availability of core space. If no core space is available, four disk pages will be requested from the system and job change will be required before control is returned to slave mode. If core space is available, only the number of disk pages corresponding to the number of available core pages will be returned.

2) Teletype Input and Output: CAL1, 1

This trap provides the functions of freeing the input buffer and transmitting messages from the task to the teletype symbiont. The information is transmitted to the CAL as follows:

a) The effective work address constains the function. These are listed in TABLE II.

b) R6 contains the control word for outputing.

c) The virtually dedicated buffer OUTBUF contains the message to be output.

2.2.b) CAL3 TRAPS

1) CAL3,0: Write Directs for SLAVE MODE

The effective work address of the CAL instruction is picked up in R7 and then WD,R60,R7 is executed. Nothing is returned.

2) CAL3,1: Acquisition of Resources for Interrupt Routines

This routine interrogates the virtually dedicated flags WANT:0, WANT:1, WANT:2, WANT:3, WANTEV1, and WANTEV2 which have been setup by the slave mode START routine, to determine if the interrupts corresponding to the flags are needed. The interrupts, General Purpose Interface registers, etc. are acquired from JANUS, disarmed and connected.

If any of the resources out needed are not available the message "RESOURCES UNAVAILABLE I AM DYING" is typed out and the trap handle branches into the abort routine.

3) CAL3,2: Acquisition of Dedicated Storage

This routine expects the number of pages to be dedicated in R6 and the maximum number of dedicatable pages in R7. A block of pages is then acquired from JANUS and the specified portion of this block is then dedicated.

If core space is unavailable the trap handler branches into the abort routine.

4) CAL3,3: Inhibiting Count Zero Interrupts

This trap sets the counter interrupt group inhibit bit in the slave mode PSD.

TABLE I

EWA	FUNCTION
0	Enter from 1 to 4 more disk pages onto the file generation stack FILESTK
1	OPEN the file
X'20'	Close file and release it to line printer symbiont
X'30'	Close file and send it first to card punch symbiont and when it returns send it to the plotter
X'40'	Close file and send it only to card punch symbiont
X'50'	Close file and send it to card punch symbiont and to MOIRAE
X'60'	Close file and send it to plotter only
X'70'	Close file and send it to MOIRAE only
X'80'	Close file and send it to JBPM.

TABLE II

Function Codes for CAL1,1

EWA	FUNCTION
0	Free Input Buffer
1	Send message in OUTBUF to teletype symbiont
2	Free input buffer and send message in OUTBUF to teletype symbiont

5) CAL3,4: Uninhibit Count Zero Interrupts

This routine clears the counter interrupt group inhibit bit in the slave mode PSD.

6) CAL3,5: Acquisition of Storage Scope Resources

This routine acquires the necessary resources for a storage scope, dedicates the scope interrupt page, and connects one of the four available scope interrupt routines to BIG-BEN.

If no scopes are available or the scope interrupt routine page is not dedicable, the routine returns with CC4=0, if all went right the routine returns CC4=1.

7) CAL3,6: Returning the scope to the system

8) CAL3,7: Put the Task on Wait

9) CAL3,8: Trigger the Job Changer for Slave Mode

10) CAL3,9: Erase the Storage Scope for Slave Mode

This routine issues an erase order, to the scope, sets the scope to busy, and requests a signal in one second to reset the scope busy flag.

11) CAL3,10: Write to the Scope in Store Mode

The register containing the bit pattern to be written to the scope is specified by the address field of the CAL instruction. The contents of this register are written to the scope first in write through mode, then in store mode 23 microseconds later.

12) CAL3,11: Enters a Command on the Event Recorder Tape Command List

This routine expects the command to be added to the list in R6 and R7. Nothing is returned.

13) CAL3,12: Slave Mode Communication with JANUS Routines

This routine executes a BAL,R11 on the address field of the CAL instruction. Returns the condition codes as set by the JANUS routine.

14) CAL3,13: Initialization of Registers in a Register Page other than Zero

15) CAL3,14: Acquire MTA81 and Return all other Resources

2.3 INTERRUPT ROUTINE LIBRARY

The interrupt routine library consists of a set of routines assembled to run unmapped on real core page X'14'. Associated with each interrupt routine is an initialization routine which fills in the address field of some of the instructions of the interrupt routine after it has been copied over into dedicated storages.

Each interrupt routine is divided into two parts: a 2-D part which simply collects data in a two-dimensional analyzer and a 1-D part which compares the data against tables setup in dedicated storage and generates spectra of counts verses channel number. The START routine which identifies and copies the interrupt routine into the dedicated page also connects the interrupt routine to the rest of the task. In order to make this connection the following assumptions about interrupt routine are made:

1) The interrupt routines are constructed with the executable code first and all constants and temporary storage locations at the end (or else the interrupt initialization will fill the start address into the appropriate PSD).

2) The last location of each routine must contain the count of the number of channels which overflowed but were not put on the overflow stack.

3) Directly following the overflow counter is a table of four addresses:

a) Pointer to a routine which accepts teletype parameters used by the interrupt routine.

b) Pointer to the routine which initialized the 2-D part of the interrupt routine.

c) Pointer to the routine which initialized the 1-D part of the interrupt routine.

d) Pointer to the instruction in dedicated storage which branches around incrementing the channel zero live time counter. This instruction is always an unconditional branch. If the channel zero option is implimented this branch is changed to BCS,3.

2.5 TABLES USED IN DEFINING THE OPERATING CONFIGURATION

In this section the tables associated with the START routine will be discussed.

In order to add new interrupt routines entries must be made in a mimimum of four tables: ROUTINE, ANALS, ADC#TBL, and LIBRARY. If a new interrupt routine required ADC usage which is not described by the codes of the table ADC#TBL (see TABLE III) then entries must be made in the tables LEGALADC, and WHICHTBL.

The tables will be discussed in the order in which they are referenced by the START routine.

In the following discussion of the tables, the EDE interrupt routine will be used as an example in all cases. The attributes of this interrupt routine are:

1) It uses two ADCs.

2) It can operate in three input modes. Data can come in through ADCs W and X (referred to as WX configuration), ADCs Y and Z, or from magnetic tape.

3) The interrupt routine uses on 2-D analyzer in SETUP mode (unless routing is implemented).

1) ROUTINE

a) This table is used by the teletype input scanning routine to identify the interrupt routine name. Each entry is used as the odd register of the Compare Byte String instruction and as such is expected to be of the form:

BTCNT	TEST
0 7	31

BTCNT is the byte count of the interrupt routine name

TEXT is the byte address of the EBCDIC interrupt name

b) Example	ROUTINE	EQU	$
		GEN,8,24	3,BA(EDE)
	EDE	DATA,3	'EDE'

Each routine is expected to have two entries in the table:

one for the routine without event recording and one with event recording. The entries with and without event recording are symmetric about the middle of the table.

2) #ANALS

This is a byte table containing the number of 2-D analyzers used by the SETUP mode part of each interrupt routine. In the case of the EDE routine, this byte would contain a one.

3) ADC#TBL

This is a byte table containing codes specifying the number of parameters and the interrelation of these parameters for each interrupt routine. The codes defined at this time are listed in TABLE III. For the EDE routine the code is 0.

4) LEGALADC

This table contains a one word entry for each of the codes in ADC#TBL. Each entry is of the form

NUM	TABLE
0 7	31

TABLE III

Meaning of Entries in the Table called ADC#TBL

CODE	MEANING	EXAMPLE
0	2 ADCS with ADC2=F(ADC1)	EDELTAE
1	3 ADCS with ADC2=F(ADC1) ADC3=G(ADC1)	EDELTAE - EDELTAE
2	4 ADCS with ADC2=F(ADC1) ADC3=G(ADC4) ADC1 and ADC4 independent	XE&XE
3	3 ADCS with ADC2=F(ADC1) ADC3 independent and ignored in setup mode	LIGHT

F,G, are arbitrary functions of the specified ADC.

1) NUM is the number of legal input configurations.

2) TABLE is the word address of the table of legal ADC configurations.

Example:

```
LEDGALADC        GEN,8,24          3,TWOADC
                     :
TWOADC           GEN,8,24          2,BA(WX)
                 GEN,8,24          2,BA(YZ)
                 GEN,8,24          4,BA(TAPE)
                     :
WX               DATA,2            'WX'
YZ               DATA,2            'YZ'
TAPE             DATA              'TAPE'
```

5) WHICHTBL

This table contains a one word entry for each code in the table ADC#TBL. These entries are used to ascertain which ADCs must be acquired from the system for the code specified by ADC#TBL and for the ADC configuration specified by the user.

Example:

```
WHICHTBL         PZE        WANT:TBL-1
                  :
WANT:TBL         PZE        WANTWX      if EDE,WX
                 PZE        WANTYZ      if EDE,YZ
                  :
WANTWX          XPSD,8      WZPSDU
                XPSD,8      PLUGPSDU
WANTYZ           DATA        0,0
                XPSD,8      YZPSDU
                XPSD,8      PLUGPSDU
```

The START routine will use the code specified by ADC#TBL as an index for picking up an entry in WHICHTBL. This entry will be used as the address of a table of addresses ordered according to the ADC configurations available. The code corresponding to the ADC configuration specified by the user will be used as an index for picking the address of a list of four flags. These flags specify whether the W, X, Y, or Z ADCs are needed (0=not needed), and the instruction to be put at the interrupt location. These four flags will be copied over into virtually dedicated storage to be used by the CAL3,1 handler,

6) SIZE

This table is used by the teletype input scanning routine to determine the dimensions of the 2-D analyzer for SETUP mode. This table is independent of the interrupt routine being used.

7) XSCOPLIM, YSCOPLIM

These are half word tables ordered according to the entries in SIZE which specify the maximum channel number of the 2-D analyzer in the X and Y directions respectively.

8) LIBRARY

This table is used in locating the source address of the interrupt routine for the appropriate ADCs. This address is used to copy the interrupt routine into the appropriate section of dedicated storage.

Example:

```
LIBRARY        PZE          EDE-2
                .
                .
                .
 EDE           PZE          EDEWX
             GEN,8,24       EDEL,WXLOAD+EDEL
               PZE          EDEYZ
             GEN,8,24       EDEL,YZLOAD+EDEL
```

where EDEWX (EDEYZ) points to the end of the interrupt routine, EDEL is the length of the interrupt routine, and WXLOAD (YZLOAD) is the mapped address of the first location in the reserved area on the interrupt routine page.

9) SIZ1DTBL

This table is used by the teletype input scanning routine to identify the number of channels in each of the counts versus channel number spectra. The entries in this table are independent of the interrupt routine being used.

10) CHANNEL

This table is used by the teletype input scanning routine to identify the various options available to the user. These options include Channel Zero live time clock counter, Routing, size of event buffers and specification of ADC conversion gain other than the number of channels in each spectrum.

2.6 PARTICLE IDENTIFICATION

In this section the structure and core arrangement of the identification tables will be discussed. These tables contain all the information required for particle identification and determine the efficiency and selectivity of the identification system. The conventions adopted in scanning these tables will be discussed. It is hoped that this discussion will give the experimentalist insight into the power and short coming of the procedure.

In the following discussion some knowledge of the language used in SETUP mode is assumed, in particular the polynomial fits are discussed in terms of the "ANALYZER" on which they were generated. This classification is relavent when either redundant identification routing is used.

In the case of redundant identification where three ADSs are used, "ANALYZER ONE" refers to the two-dimentional data accumulated by storing ADC2 as a function of ADC1. The polynomial fits generated defining the bands of this display are used in RUN mode as the first identification criteria. "ANALYZER TWO" refers to the two-dimensional data accumulated by storing ADC3 as a function of ADC1. The polynomial fits generated to define bands on this display are used as the second identification criteria in RUN mode.

In the case where routing is used, eight "ANALYZERS" are defined. These analyzers correspond to the eight detectors which can be used. The bands for each detector are set independently by storing data in one "ANALYZER" at a time, i.e. only accepting data with the appropriate routing bits for the analyzer being used.

2.6a) ARRANGEMENT OF DEDICATED STORAGE IN RUN MODE

The arrangement of dedicated storage in RUN mode is:

1) The spectra occupy the first block of dedicated storage.

2) Floating on top of the spectra is the table containing the values of the polynomial fits corresponding to the lower boundary of each band.

3) Floating on top of the lower band table is the table corresponding to the upper boundary of each band. These tables are integer, half-word resolution with 256 entires for each fit.

2.6b) ARRANGEMENT OF FITS IN THE TABLES

The fits for each 2-D analyzer are arranged in the table in the order in which they were run. The different analyzers are arranged according to their number with analyzer one coming first. These arrangments are illustrated in Figure II.

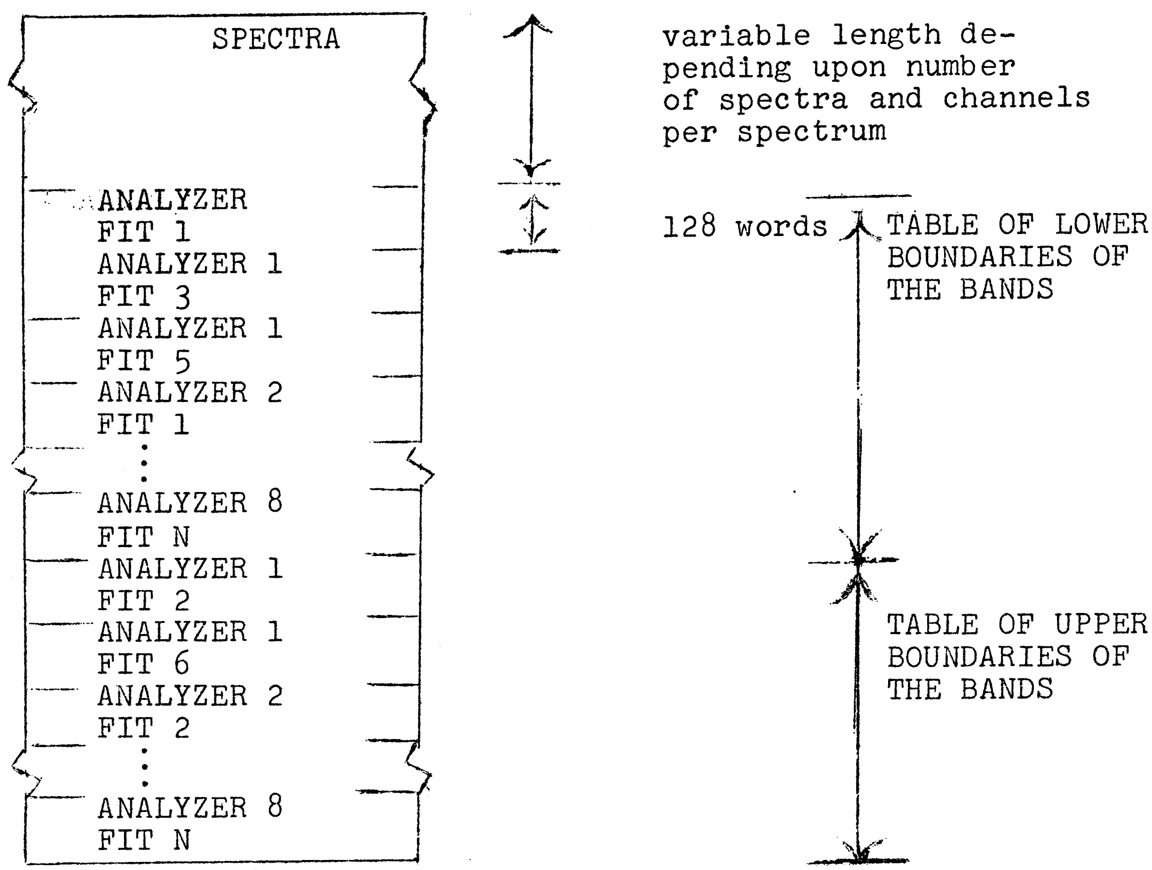
2.6c) THE INDIVIDUAL FITS

The polynomial fits are generated using single precision floating point arithmetic. It is assumed that the movable cross can be positioned to approximately one part in 30,000 i.e. the points used in generating the fits contain 15 bits of X and Y position information. The tables generated in RUN mode are truncated after being converted to integers to retain only the precision determined by the conversion gain of the ADCs. For example, of the ADC conversion gain is 1024, all entries in the tables will be in the range (0,1023) inclusive.

In addition to the coefficients and the order of the polynomial, the parameters XMAX, XMIN, YMAX, and YMIN corresponding to the maximum and minimum X and Y channel numbers on the 2-D display at the time the fit was generated are retained.

FIGURE II

Arrangement of Dedicated Core in RUN Mode



These channel numbers are used in the table generation process as follows:

1) The fits are only generated for the table positions between and including XMIN and XMAX. The remaining table entries are set to X'7FFF' which is greater than the maximum bit pattern the ADC'C can convert.

2) If the value of the fit at any point exceeds YMAX it is set to YMAX and if it is less than YMIN it is set to YMIN.

2.6d) SCANNING THE TABLES

1) Single Identification

In the case of routing (standard single identification will be considered one level routing) the calculations involving ADC1 and ADC2 are carried out, maintaining full precision when possible, to obtain the quantities corresponding the X and Y coordinates in the 2-D analyzer. The Y coordinate is then truncated to be consistent with the conversion gain on the ADCs and the X coordinate is truncated such that $0<X<255$. The appropriate multiple of 256 is then added to the X coordinate to point to the first table for the detector specified by the routing bits (if routing is not implemented this addition is skipped to save time). The Y coordinate is then checked against the lower boundary table. If the Y coordinate is greater than or equal to the lower boundary the upper bound table is checked. If the Y coordinate is less then the upper boundary of the band, the X resolution, consistent with the number of channels per spectrum, is restored and is used as an index to increment by one the channel to which it points. If either of these checks fails, X is incremented by 256 which sets everything up to check the next band (when a match is found and the resolution restored, the X coordinate always points to the correct channel of the correct spectrum) and repeats the tests.

2) Redundant Identification

In this case calculations involving ADC1 and ADC2 are carried out, leaving ADC3 alone, to obtain the X and Y coordinates for "ANALYZER ONE". The Y coordinate is then truncated to be consistent with the conversion gain on the ADC's and the X coordinate is truncated such that $0 \leq X \leq 255$. The tables corresponding to "ANALYZER ONE" for which double identification has been specified are scanned first. If a match is not found, the tables requiring only single identification are checked.

If a match is found in the portion of the tables requiring redundant identification, the number of the band in "ANALYZER ONE" is transmitted to the second identification loop. Now ADC1 and ADC3 are used to compute the X and Y coordinates corresponding to "ANALYZER TWO". A table of band numbers is consulted to determine which bands of "ANALYZER TWO" should be checked for the particular band in "ANALYZER ONE", in which the match was found. This second set of bands is then checked in the order in which the band numbers appear in the table. If a match is found the X resolution consistent with the number of channels per spectrum is restored and the appropriate channel of the appropriate spectrum is incremented. If the event does not lie in any of the bands specified by the band table, the event is discarded.

The table of band numbers for the second identification is input from the teletype via the "DEMAND" command. The bands of "ANALYZER TWO" are checked in the same order as the band numbers are typed in.

Note that routing is not available with redundant identification.

2.6.e COMMENTS ON BANDS AND IDENTIFICATION PROCEDURES

1) Since the order in which the user defines the bands determine the order in which the bands are checked, the user has limited control over dead time. If dead time is expected to be significant it is suggested that the band defining the most prolific reaction products be selected first.

2) If dead time is not a serious problem one can decrease background while insuring against band leakage by embedding one band within another. This is accomplished by first selecting a pair of fits quite close to the particle bands then selecting a second pair of fits below and above the first pair. Events which fall within the inner band will be classified in the first spectrum while those events falling between the inner and outer bands will fall in the second spectrum.

3) Since the fits are only defined over the region of the 2-D being displayed at the time the fits are run, one can define sharp cutoffs along the X and Y directions.

2.7 EVENT RECORDING

All the interrupt routines permit the experimentalist to collect all the raw data (with exception of channel zero events when the channel zero option is implimented) in buffers which when full are copied onto magnetic tape. This mode of operation provides the user with backup data which can be played back at his convenience.

Some caution must be exercised in using the event recorder. Since it provides no dead time meaurement, and since its dead time is somewhat greater than that of the routine which processes the data on line, it may prove to be unreliable for high count rate data (rates in excess of 5000 counts/sec.). Also, since every event is put on tape, one often finds the number of tapes required for a given experiment prohibitive (this factor becomes significant at count rates in excess of 500 counts/sec if the optional 240 byte records are used and at count rates of 250 count sec if the default 120 bytes/record are used).

2.7a EVENT RECORDING INTERRUPT ROUTINES

When event recording is implimented, the interrupt routine connected to the highest priority interrupt of the set available to the ADC's is altered by its initialization routine to permit the data to be stored in the event buffers via

	STW,RDATA	*BUFFER,X
or		
	STD,RDATA	*BUFFER,X

This first case being used with two parameter data and the second set being used with three and four parameter data.

The interrupt of the second ADC is used to increment the event buffer index and to switch buffers when one is full. This method relies heavily on the assumption that the interrupts will fire in the correct order. The experimentalist can easily violate this assumption by removing the ADCs from a coincidence group.

2.7b EVENT TAPES

1) Tape Records

All tape records are of the same length (including title records). The default case produces 120 byte records (30 two parameter of 15 three parameter events). The user can double the amount of data a tape can hold by requesting the task write 240 byte records. This is accomplished either as an option in the "DEFINE command or through the "RECORD" command.

2) The Form of the Tape

a) Title Records: A title record appears at the beginning of each run. It contains up to 79 characters (the rest of the 120 or 240 bytes are blank) typed in by the user at the beginning of each run.

b) Data Records: The data is written on the tape in binary form exactly as the data was read from the ADCs. In the case of two parameter data each event consists of one word. (4 bytes) of the form:

R1	ACC1	R2	ADC2

0 3 15 18 31

where R1 and R2 are the routing bits for ADCs 1 and 2, and ADC1 and ADC2 are the 13 bits read by the respective ADCs.

In the case of three parameter data each event consists of one doubleword (8 bytes) of the form:

R1	ADC1	R2	ADC2
R3	ADC3		0

c) Termination Records: These records are used to fill out the rest of a buffer before it is put on the tape. In order to indicate the end of the real data in the buffer, the remainder of the buffer is filled with X'FFFFFFFF' (-1). There may be as many as three records containing some (-1)'s. The last record is followed by a single end of file. The end of all data is signified by a double end of file.

2.8 OVERFLOW CONSIDERATIONS

Halfword storage is used in all the data arrays. This permits a given channel to accumulate up to 65535 counts before it overflows and is reset to zero. In many, experiments a significant number of channels in the strongly excited peaks will overflow many times in the course of a run. Since these strong peaks are often used for normalization and energy calibration, it is imperitive that the total number of counts in each channel be known.

TOOTSIE has a rather elaborate bookkeeping structure to insure that records of all overflows are maintained. The following procedures are used.

1) After a channel is incremented it is checked to see if it has overflowed. If it has overflowed its index is entered into a small array within the interrupt routine and the slave mode part of the task is signaled.

2) The Slave Mode section then empties the small interrupt routine array, checking each index against its own much larger list of channels (up to 512) which have previously overflowed. If the channel index is already on the slave mode list, the number of overflows is incremented by one (each entry can count up to 16383 overflows), otherwise the index is added to the list.

3) All printing, punching, plotting, analysis, and display routines scan this slave mode overflow list, correcting each channel found.

4) This bookkeeping procedure can lose track of overflows due to at least two factors: The interrupt array can fill before the slave mode can empty it, or the slave mode list can fill. In the event either of these happens, a backup counter is incremented. This counter keeps track of all overflows where the channel number information has been lost. This counter is punched in columns 70 and 71 of the title card of each spectrum.

III. Users Guide

TOOTSIE has been designed primarily as a data acquisition task. However, the capability to play back data tapes (e.g. tapes written either by TOOTSIE in event recording mode or by the task EVENT) has been provided. The tape input mode was included in the task for two reasons:

1) For low count rate experiments, on-line particle identification could prove impractical.

2) For some special case the two-dimensional analyzers of 256x256, 512x128, or 1024x64, available in addition to the standard 128x128, 64x256, 256x64, and 64x64 analyzers could prove more suitable for the identification procedure.

All the teletype commands and scope routines which are available to the user when the task is used for data acquisition are avilable when data comes in from tape.

In this chapter all the teletype commands will be discussed. The commands of greatest interest to the experimentalist are the DEFINE, SCOPE, GO, and RETURN Commands.

The DEFINE command is used to communicate to the task all the information needed to acquire or reserve the necessary resources (ADCs, Core space, Magnetic Tape Transports, etc.) for the experiment. Data accumulation in SETUP mode begins as soon as this command has been entered.

In order to see the data on one of the Tektronix 611 storage scopes, the user must use the SCOPE command. The polynomial fits for the various detectors (in the case of routing) or various parameters (in the case of redundant identification, can be generated. To facilitate this procedure the commands REPLACE, DEGREE,ANALYZER, SPECTRUM,FITS, and COPY may prove useful.

Once all the polynomial fits have been generated, the transition to RUN mode (collection of counts versus channel number spectra) is accomplished with the GO or RUN Commands.

Finally the experimentalist may wish to return to the SETUP mode to check the bands. This is accomplished with the RETURN command.

The remaining commands provide services ranging from the important I/O control functions to the novel alarm clock service.

3.1 SETUP MODE TELETYPE COMMANDS

Activate

ACTIVATE, OC:	Transfers all teletype control (both input and output) to the operators console (OC).
ACTIVATE,TY5:	Returns teletype control to the ASR teletype (TYA05)
NOTE.	One can always input to TOOTSIE on the operators console.
INTERNAL ROUTINE NAME:	ACTIVATE05

Alarm

ALARM ON,hhmm	The bell on the storage scope will ring at time hhmm, hh is the hour (on a 24 hour clock) and mm is the minute.
EXAMPLES:	ALARM,ON,1320 ALARM,ON,13:20 In both case the alarm will ring at 1:20 p.m.

ALARM,OFF,hhmm — The bell, which was set to ring at hhmm will not ring.

COMMENTS — The alarm routine will retain up to four alarm requests. The ALARM.OFF command does not illuminate an alarm request, but simply indicates that the request is to be ignored. The task will not exit if there are any alarm requests. It will release all its resources and wait for the request time to come.

INTERNAL ROUTINE NAME: ALARM

Analyzer

a) Redundant Identification

COMMENTS. — This command is used with redundant identification to switch from storing ADC2 versus ADC1 to storing ADC3 versus ADC1.

ANALYZER,2 — Store ADC3 versus ADC1

ANALYZER,1 — Store ADC2 versus ADC1

ANALYZER,? — Type current analyzer number

b) Routing

COMMENT: — This command is used to specify the detector to be stored. The routing bits for ADC1 are compared with the current analyzer number (actually the current analyzer number minus one), if a match is found the event is stored, otherwise the event is ignored.

Examples:

ANALYZER,5 — Store events from detector number 5 and ignore all others.

ANALYZER,? — Type out current analyzer number.

INTERNAL ROUTINE NAME: ANALCHEK"

Channel 0

COMMENT:	If the channel zero live time clock option is implemented, this command causes the contents of the live time scaler to be typed out.

Continue, aa

aa	The ADC configuration as it was typed in the DEFINE command.
CONTINUE,WX	If preceded by a STOP or WAIT command, this command causes the interrupts to be armed and enabled. If no STOP or WAIT command has been previously typed in, this command does nothing.
INTERNAL ROUTINE NAME:	TSTCONTU

Copy

COPY,A1(F1,F2)A2, A1(F1,F2)A2,...

A1 Source analyzer number
A2 Destination analyzer number
F1 First Fit to copy
F2 Last fit to copy

Example: COPY,1(1,4)2,1(5,8)3

This means copy the first four polynomial fits (i.e. the first two bands) which have been run on ANALYZER 1 to ANALYZER 2. Also copy fits 5 to 8 of ANALYZER 1 to ANALYZER 3. Upon completion of the copy operation ANALYZER 2 will have four fits defined (numbered 1 to 4) which will be identical to the first four fits of ANALYZER 1. Any fits which had previously been accepted in ANALYZER 2 will be overwritten and lost. Similarly ANALYZER 3 will have four fits defined (labeled one to four) which will be identical to fits five to eight of ANALYZER 1.

COMMENT: This routine is intended to be used with routing. If one should be fortunate enough to have identical detection systems, the same identification bands may suffice for all detectors.

DEFINE

(INTERNAL NAME: STARTI)

```
DEFINE,rr,aa,dd,cc,nn,00,00,00,...
DEFINE,rr,aa,dd,dd,dd,...,cc,nn,00,00,...
```

rr is the name of the interrupt routine

aa is the input specification (either the appropriate ADC configuration or TAPE to indicate reading in old data from magnetic tape)

dd is the dimensions of the 2-D analyzer

cc is the number of channels per spectrum

nn is the number of spectra

oo is one of the options available

NOTE: For redundant identification one can specify different dd's for the two analyzers as long as the total number of channels in the two analyzers is the same. In the case of routing each of the 8 analyzers can have different dd's.

1) Examples:

a) Single Identification

DEFINE, EDE,WX,64x256,1024,3,2048,C

rr=EDE. This stands for E versus delta E.

aa=WX: Implies ADC1=W and ADC2=X

dd=64x256, Means use a 2-D analyzer with 64 channels along the Y axis and 256 channels along the X axis.

cc=1024, Means accumulate spectra of 1024 channels each in the RUN mode

nn=3, Means accumulate 3 such spectra

oo=2048 Expect the ADC conversion gain to be 2048. The code will divide all the energy signals by 2 before incrementing a channel of a spectrum.

oo=C; Implement the channel zero option.

b) Routing:

```
       DEFINE,EDE,YZ,128x128,512,6,1024,R,C
or     DEFINE,EDE,YZ,128x128,64x256,512,6,1024,R,C
```

rr=EDE, aa=YZ, dd=128x128, cc=512, nn=6, oo=1024, oo=c

oo=R (This means routing.)

In the second example the dimensions of ANALYZER 2 will be 64x256 while the dimensions of the other seven analyzers will be 128x128.

c) Redundant Identification

```
DEFINE,EDELTAE&EDELTAE,WX&WY,128x128,2048,4,C
DEFINE,E*T**2&EDELTAE,XY&XZ,64x256,128x128,1024,3,C
```

```
rr=EDELTAE&EDELTAE       A redundant identification routine
aa=WX&WY                 use ADCs W.X, and Y
dd=128x128               both 2-D analyzers will be 128x128
cc=2048                  2048 channels/spectrum
nn=4                     4 spectra
00=c                     implement channel zero
```

d) Tape Input.

DEFINE, E*T**2,TAPE,256x256,2048,5,4096

```
rr=E*T**2                single identification
aa=TAPE                  read the data in from magnetic
                         tape rather than the ADCs
dd=256x256               note a 64K analyzer
cc=2048
nn=5                     five spectra
oo=4096                  conversion gain of 4096
```

2) Options

The DEFINE Command is used to specify all the parameters necessary for data acquisition. The type of calculation to be performed on the data and the ADC configurations available for each calculation will be presented later. The procedure for implementing each of the options and a definition (of sorts) will also be given.

a) Channel Zero

This option is implemented by typing C in the option field

DEFINE,EDE,WX,WX,128x128,1024,3,C

This option consists of an internal scalar which is incremented when an interrupt occurs for which all the ADCs in the coincidence group have converted zeros.

The user puts the channel zero pulses (standard logic pulses) into the ADCs through the BNCs on the ADC logic box labeled LIVE TIME CLOCK INPUTS .

b) Conversion Gain

This option is implemented by typing in '512', '1024', '2048', '4096', or '8192' in the option field.

DEFINE,XE/E,YZ,128x128,1024,3,4096

This option permits the use of more precision in the calculation and identification than is permitted by the number of channels in the spectra being accumulated. The data taking routine is initialized to accept numbers in the range specified by the CONVERSION GAIN (normally numbers larger than the highest channel number are discarded) and will divide these numbers by the appropriate factor of two before a spectrum is incremented.

c) Event Recording

This option is implemented by adding '&EVENT' to the name of any of the interrupt routine names:

DEFINE,EDELTAE&EVENT,WX,128x128,2048,3

This will cause the data to be entered into buffers which are then written to magnetic tape. The data is put in the buffers before any overflow checking, calculations, or identification.

d) Routing

This option is implemented by typing "R" in the option field.

DEFINE,XE/E,WX,128x128,512,4,R

The data from each ADC comes to the computer as a 16 bit quantity, of which 13 bits correspond to the output of the ADC and 3 bits (8 possible combinations) correspond to the routing bits.

```
[ R |        ADC        ]
0 3                   15
```

TOOTSIE interrogates the routing bits associated with the highest ADC being used (W is higher than X is higher than Y is higher than Z) and checks only the bands corresponding to those routing bits. The routing signals are put into the ADCs through the eight BNCs on the panel labeled ADC ROUTING.

e) Record Size

This option is applicable only when event recording. It permits the user to control the amount of data written out in each physical tape record (either 120 or 240 byres/record).

The option is implemented by typing either 120 or 240 in the option field.

DEFINE,EDELTAE&EVENT,WX,128x128 1024,3,240

This will force 240 bytes/record.

3) Interrupt Routines

<u>EDELTAE or EDE</u>

INPUT: W=E X=DELTAE, Routing Bits: 8 BNCs for WADC
or
Y=E, Z=DELTAE, Routing Bits: 8 BNCS for YADC
or
TAPE

TIMING: 43.8μs/count for a match on first spectrum

7.0μs/count increment per mismatch

OPERATION:

SETUP or 2-D mode:

Allows the user to display DELTA E versus E on a two dimensional analyzer. The display of the 2-D analyzer is a projection of a section of the Z axis, andwiched between two planes, onto the upper plane. Channels with counts below the lower plan or greater than the upper plane are not displayed. Up to 14 bands can be defined (28 fits ranging in degree from 1 to 9 inclusive). The display of the two dimensional analyzer can be expanded to facilitate fitting.

ONE-D or RUN mode

The bands defined in SETUP are used to classify events into spectra. The code assumes that the fits were run in pairs with the lower fit run first. To minimize the time necessary to scan, fits should be run in order of decreasing count rate.

SEE THE COMAND: GO

EXAMPLE: %DEFINE EDE,YZ,64x64,4096,3

<u>E*DELTAE or E*DE</u>

INPUT: W=E, X=DELTAE, Routing Bits: 8 BNCs for WADC
or
Y=E, Z=DELATE, Routing Bits: 8 BNCs for YADC
or
TAPE

TIMING: 51µs/count for a match on first spectrum

7µs/count increment per mismatch

OPERATION: Same as EDELTAE except the product of E and DELTA E is used in place of DELTA E.

E*T**2 or E*T**

INPUT: W=E, X=TIME, Routing Bits: 8 BNCs for WADC
or
Y=E, Z=TIME, Routing Bits: 8 BNCs for YADC
or
TAPE

TIMING: 83.4µs/count for match on first spectrum

7µs/count increment per mismatch

OPERATION:

SETUP:

Allows the user to plot the function A+E*(-TZERO)**2/NORM versus E on a two-dimensional analyzer. The parameters A,TZERO, and NORM are typed in on the teletype (see the command PARAMETER). These parameters can be changed at any time. Up to 14 bands can be defined.

ONE-D or RUN mode:

Same as EDELTAE except the above function is used for table look up.

LIGHT

INPUT: W=L, $X=\frac{dL}{dt}$, Y=TOF, Routing Bits: 8 BNCs for WADC
or
X=L, $Y=\frac{dL}{dt}$, Z=TOF, Routing Bits: 8 BNCs for XADC

TIMING: 54µs/count for match of first spectrum

7µs/count increment per match

OPERATION:

SETUP:

Plots $\frac{dL}{dt}$ versus L in the 2-D analyzer. The time-of-flight (TOF) information is ignored.

RUN

Particle identification is accomplished using the L and $\frac{dL}{dt}$ signals. Spectra of counts versus TOF are generated.

XE/E

INPUT: W=E, X=XE, routing Bits: 8 BNCs for WADC
or
Y=E, Z=XE, Routing Bits: 8 BNCs for YADC

TIMING: 60µs/count for match on first spectrum
7µs/count increment per match

OPERATION:

SETUP:

Plots E versus the quotent XE/E in the 2-D analyzer.

RUN

Particle identification is acoomplished using the E and XE/E information. Spectra of counts versus XE/E are accummulated.

It should be pointed out that wild channel to channel oscillations due to trunciation errors introduced by the division can result of the conversion gain (see CONVERSION GAIN option) is the same as the number od channels per spectrum. To avoid these problems the conversion gain should exceed the number of channels per spectrum by a factor of four. (i.e. if one chooses 512 channels per spectrum, one should specify a conversion gain of 2048).

E*DELTAE&*DELTAE

INPUT: W=E, X=E_1, Y=E_2
or
X=E, Y=E_1, Z=E_2
or
TAPE

TIMING: Acquisition time: 39.8 μs/count with 15 level derandomization

Identification Time: 97.4 μs/count to double identify best case

7.0 μs/count increment for each mismatch on first table set

12.6 μs/count increment for each mismatch on second table set

OPERATION:

SETUP Mode

Allows the user to put the product of E and E_1 versus E on the first analyzer and the product of E and E_2 versus E on the second analyzer. Up to ten bands (20 fits with lower gate run first) can be defined on each analyzer. It is assumed that the first fits on the first analyzer corresponding to spectra which require double identification are run first followed by any spectra which require only single identification. See the commands: GO, DEMAND, and ANALYZER

ONE-D or RUN Mode

In this mode the program uses the gates set up in 2-D mode to classify incoming events into spectra. The first N bands (see the command GO) of analyzer one are assumed to require double identification hence a match in these spectra initiates a second scan through the bands of analyzer two which have been specified as legal (see the command DEMAND).

EXAMPLE:

%DEFINE,E*DELTAE&E*DELTAE,WX&WY,256x64,128x128,2048,4,C

Means connect routine E*DELTAE&EDELTAE plot the first product on a 256x64 2-D analyzer and plot the second product on a 128x128 2-D analyzer, allow for 4, 2048 channel spectra and implement channel zero.

EDELTAE&EDELTAE

INPUT: W=E, $X=E_1$, $Y=E_2$
or
X=E, $Y=E_1$, $Z=E_2$
or
TAPE

TIMING:

Acquisition time: 39.8μs/count with 15 level derandomization

Identification time: 91.3μs/count to double identify best case i.e. match first spectrum of analyzer one and a match in first spectrum of analyzer two

12.6μs/count increment for each mismatch on second table set

7.0μs/count increment for each mismatch on first table set

OPERATION: Same as E*DELTAE&E*DELTAE

E*T**2&EDELTAE

INPUT: W=E, Z=TIME, Y=E
or
X=E, Y=TIME, Z=E
or
TAPE

TIMING: Acquisition time: 39.8 μs/count with 15 level derandomization

Identification time. 120 μs/count for double identification of the first spectra on analyzer one and the first spectrum on analyzer two

7 μs/count increment for each mismatch on analyzer two

OPERATION:

SETUP Mode

Plots A+E*(T-ZERO)**2/NORM versus E on the first 2-D analyzer and plots E versus E on the second analyzer. Operation is the same as E*DELTAE&DELTAE. See E*T**2, PARAMETER, GO, and DEMAND.

ONE D or RUN Mode

Same as E*DELTAE&E*DETLAE

E*T**2&E*DELTAE

INPUT: W=E, X=TIME, Y=E
or
X=E, Y=Time, Z=E
or
TAPE

TIMING: Acquisition time: $39.8\mu s$/count with 15 level derandomization

Identification time: $127\ \mu s$/count best case

$7\ \mu s$/count increment per mismatch on first table set

$12.6\ \mu s$/count increment per mismatch on second table set

XE&X E

INPUT: $W=E_1$, $X=XE_1$, $Y=E_2$, $Z=XE_2$
or
TAPE

TIMING: Acquisition time:

Identification time:

OPERATION:

SETUP Mode

Plots E_1 versus XE_1/E_1 on ANALYZER ONE, and plots E_2 versus XE_2/E_2 on ANALYZER TWO.

RUN MODE

Uses E_1 and XE_1/E_1 to make the initial identification. If a match is found the bands selected by the user with the DEMAND command are scanned using E_2 and XE_2/E_2. Spectra of counts versus XE_1/E_1 are generated.

DEGREE,N

(INTERNAL NAME C:DEGREE)	is the degree (or order) of the polynomial fit to be used in drawing the bands. Legal values of n are (1,9) inclusive.

Example:

Degree,3	The code will use a third order polynomial.
Degree,?	Returns the order of the polynomial currently being used.

DEMAND,(A_1=N_1,N2,N3), (A2=N3)

(INTERNAL NAME DEMAND)

A1,A2	the band numbers (numbered from zero) on ANALYZER 1
N1,N2,N3	the band numbers (numbered from zero to check in ANALYZER 2.

SUMMARY OF ROUTINES AND OPTIONS

The following table summaries the legal input for each of the fields of the define statement. Having chosen a routine, one can choose any of the messages from the columns Input, Anal. One, Anal. Two (if applicable), and Chan. Per Spect. The entries in the No. Spect. Column indicate the maximum number of spectra permitted for the number of channels in each spectrum (i.e. a maximum of 14, 512 channel spectra for EDE). Any or all of the entries in the Option Column may be implimented.

Routine	Input	Anal. One	Anal. Two	Chan. Per Spect.	No. Spect.	Options
EDELTAE	WX	128x128	---	512	14	Channel Zero
EDE	YZ	256x64		1024	12	Routing
	TAPE	64x256		2048	10	Conversion Gain
		64x64		4096	5	
		*256x256		8192		
		*512x128				
		*1024x64				
E*DELTAE	WX	128x128	---	512	14	Channel Zero
E*DE	YZ	256x64		1024	12	Routing
	TAPE	64x256		2048	10	Conversion Gain
		64x64		4096	5	
		*256x256		8192	3	
		*512x124				
		*1024x64				
E*T**2	WX	128x128	---	512	14	Cnannel Zero
	YZ	256x64		1024	12	Routing
	TAPE	64x256		2048	10	Conversion Gain
		64x64		4096	5	
		*256x256		8192	3	
		*512x128				
		*1024x64				
EDELTAE&EVENT	WX	128x128	---	512	14	Channel Zero
	YZ	256x64		1024	12	Conversion Gain
		64x256		2048	10	
		64x64		4096	5	Tape Record Size
				8192	3	

Routine	Input	Anal. One	Anal. Two	Chan. Per Spect.	No. Spect.	Options
E*DELTAE&EVENT	WX	256x64	---	512	14	Channel Zero
	YZ	128x128		1024	12	Conversion Gain
		64x256		2048	10	Tape Record Size
		64x64		4096	5	Routing
				8192	3	
E*T**2&EVENT	WX	256x64	---	512	14	Channel Zero
	YZ	128x128		1024	12	Conversion Gain
		64x256		2048	10	Routing
		64x64		4096	5	Tape Record Size
				8192	3	
EDELTAE&EDELTAE	WX&WY	128x128	128x128	512	10	Channel Zero
	XY&XZ	256x64	256x64	1024	10	Conversion Gain
	TAPE	64x256	64x256	2048	8	
		64x64	64x64	4096	4	
		*256x256	*256x256	8192	2	
		*512x128	*512x128			
		*1024x64	*1024x64			
E*DELTAE&E*DELTAE	WX&WY	128x128	128x128	512	10	Channel Zero
	XY&XZ	256x64	256x64	1024	10	Conversion Gain
	TAPE	64x256	64x]56	2048	8	
		64x64	64x64	4096	4	
		*256x256	*256x256	8192	2	
		*512x128	*512x128			
		*1024x64	*1024x64			
E*T**2&EDELTAE	WZ&WY	128x128	128x128	512	10	Channel Zero
	XY&XZ	256x64	256x64	1024	10	Conversion Gain
	Tape	64x256	64x256	2048	8	
		64x64	64x64	4096	4	
		*256x256	*256x256	8192	2	
		*512x128	*512x128			
		*1024x64	*1024x64			

Routine	Input	Anal. One	Anal. Two	Chan. Per Spect.	No. Spect.	Options
E*T**2&E*DELTAE	WX&WY	128x128	128x128	512	10	Channel Zero Conversion Gain
	XY&XZ	256x64	256x64	1024	10	
	TAPE	64x256 64x64 *256x256 *512x128 *1024x64	64x256 64x64 *256x256 *512x128 *1024x64	2048 4096 8192	8 4 2	
LIGHT	WXY	128x128	---	512	14	Channel Zero Conversion Gain Routing
	XYZ	256x64		1042	12	
		64x256 64x64		2048 4096 8192	10 5 3	
LIGHT&EVENT	WXY	128x128		512	14	Channel Zero Conversion Gain Routing Tape Record Size
	XYZ	256x64		1024	12	
		64x256 64x64		2048 4096 8192	10 5 3	
XE/E	WX	128x128	---	512	14	Channel Zero Conversion Gain Routing
	YZ	256x64		1024	12	
		64x256 64x64		2048 4096 8192	10 5 3	
XE/E&EVENT	WX	128x128	---	512	14	Channel Zero Conversion Gain Routing Tape Record Size
	YZ	256x64		1024	12	
		64x256 64x64		2048 4096 8192	10 5 3	
XE&XE	WXYZ	128x128	---	512	14	Channel Zero Conversion Gain
	TAPE	64x256		1024	12	
		256x64 64x64		2048 4096 8192	10 5 3	

Routine	Input	Anal. One	Anal. Two	Chan. Per Spect.	No. Spect.	Options
XE&XE&EVENT	WXYZ	128x128	----	512	14	Channel Zero
		64x256		1024	12	Conversion Gain
		256x64		2048	10	Tape Record Size
		64x64		4096	5	
				8192	3	

EDELTAE&EDELTAE&EVENT   Same as EDELTAE&EDLTAE

E*DELTAE&E*DELTAE&EVENT   Same as EDELTAE&EDELTAE

E*T**2&EDELTAE&EVENT   Same as E*T**2&EDELTAE

E*T**2&DELTAE&EVENT   Same as E*T**2&E*DELTAE

*applies only for input from tape

Examples.
%DEFINE,EDELTAE,WX, 64x256,2048,5
%DEFINE,EDELTAE&EDELTAE&EVENT,WX&WY 128x128 64x256 1024,8 CHANNEL ZERO

COMMENT: This routine is meaningful only for redundant identification. See 2.6e

EXAMPLE: DEMAND,(0=1,2),(1=0)

This means if an event falls in band zero (that is between fits numbered one and two) of ANALYZER 1, bands 1 and 2 of ANALYZER should be checked in making the second identification. Also, if an event falls in band one (between fits 3 and 4) of ANALYZER 1, band zero of ANALYZER 2 should be checked.

ERASE,aa,ff

(Internal Name ERASETI)

aa — ADC Configuration as it was typed in the DEFINE command

ff — Function to be performed. Three are available:

1) Blank - Erase the 2-D analyzer
2) Enable - Erase the 2-D analyzer if the "ERASE SWITCH" on the scope is toggled.
3) Disable - Do not erase the 2-D analyzer if the scope switch is toggled.

Example:

ERASE,WX — Will erase the 2-D

ERASE,WX,DISABLE — Will inhibit the scope switch

ERASE,WX,ENABLE — Will activate the scope switch

COMMENT: One need not worry about accidentally erasing their data if the DISABLE or ENABLE are misspelled. The data will only be erased if the next non-blank character following the specification of the ADCs is a new line" character.

EVENT CONTINUE,aa

aa — ADC configuration as it was typed in the DEFINE command.

EVENT CONTINUE,WX — If the event recorder is waiting (i.e. EVENT WAIT or WAIT had previously been typed in), the event recorder will be started. If the event recorder is running this command does nothing.

EVENT START,aa

See EVENT CONTINUE

EVENT STOP,aa

aa — ADC Configuration as it was typed in the DEFINE command.

EVENT STOP,WX — This command stops the event recorder from taking data, fills the rest of the buffer currently being filled with (-1) minus ones, writes the buffer on the tape, and then writes an end of file on the tape.

EVENT TITLE,ttt

ttt — Up to 79 EBCDIC characters.

COMMENT: — This command must be typed in order to start the event recorder after either the "DEFINE", "STOP", or "EVENT STOP" commands.

EVENT WAIT,aa

aa — ADC configuration as it was typed in the DEFINE command.

EVENT WAIT,WX — This command stops the event recorder from taking data. Neither the event recorder buffers nor the magnetic tape are affected.

EXIT,aa

aa — ADC configuration as it was typed in the DEFINE command.

EXIT,YZ — If event recording, this routine goes first to EVENT stop, then writes a second end file on the tape.

NOTE: The EXIT command is the only routine which writes a double end file on the tape. To assure that a tape is properly terminated, it should not be dismounted until the task has exited.

FILE, RUN NUMBER, TITLE

FILE,125,$^{16}O(p,d)^{15}O$ — This command will cause a CPUNCH deck of the 2-D analyzer with the title card $^{16}O(p,d)^{15}O$ to be entered onto the permanent file disk. The name by which the file can be recovered is typed out on the teletype. This name consists of a concatination of the user's name and the run number. If the same run number appears on two files, the new copy of the file over writes the previous copy. Therefor ONE MUST SPECIFY A DIFFERENT RUN NUMBER FOR EACH FILE TO BE SAVED.

The FILE command causes TOOTSIE to create a file on the swap disk which contains as its first three cards:

```
!JOB    ACCOUNT,NAME
!ASSIGN M:BO(FILE,NAMERUN0125),(SAVE,31DEC99)
!COPY   SI,BO
```

This file is then released to the J B SCHEDULER task which starts up a JBPM. It is JBPM which creates the file on the permanent disk. The important point is that TOOTSIE has no way of knowing whether the file was actually put on the permanet disk. The message typed out by TOOTSIE indicated only that the temporary file has been generated and released to the monitor. The only way one can be sure that the file has been put on the permanent disk is to look at the output of the JBPM monitor started up by TOOTSIE.

FITS,?

COMMENT: Causes the number of fits run and the number of fits remaining to be typed out. The number typed out in the "REMAINING FITS" field assumes that each analyzer can generate two fits for each spectrum. While this is true in redundant identification, it is not true for routing. When one is using routing one must simply add up the number of fits generated and subtract it from twice the number of spectra specified in the define statement to obtain the number of fits remaining.

GO,aa,nn

aa ADC configuration as typed in the DEFINE statement.

nn — The number of bands in ANALYZER 1 requiring double identification.

EXAMPLES:

GO,WX — Will cause the code to go into RUN mode, no double identification.

GO,WX&WY,3 — Will cause the code to go into RUN mode. If an event falls in one of the first three bands of ANALYZER 1, the bands (specified by the "DEMAND" command) of ANALYZER 2 will be checked.

NAME,nn

nn — Up to 8 BCD characters. This name will be used on all headers, bill-board cards, and subtask control cards. No dirty words please.

PARAMETER, A=nn, TZERO=nn, NORM=nn, GAIN=nn

nn — is a signed, fixed point number

COMMENT: This routine accepts any or all of the parameters listed above. These parameters are entered into the interrupt routine tables (if applicable) corresponding to the analyzer currently in use (if routing is used one can specify up to 8 different sets of parameters). For the routines reading data in from tape the parameters are used on the Y data.

Y=GAIN*Y/NORM+A.

PRINT, RUN NUMBER, TITLE

PRINT,250,2-D PRINT OUT OF DATA

This routine prints out the contents of current 2-D analyzer in a form that can easily be cut and pasted together to present the data in the form of a large matrix.

PUNCH, RUN NUMBER, TITLE

PUNCH,250,2-D PRINT OUT OF DATA

This routine causes a title card followed by a CPUNCH deck of the current 2-D analyzer to be punched out. The CPUNCH routine assumes that the 2-D analyzer is just a spectrum of either 4096, 16384, or 66536 channels. When read into a FORTRAN program using CREAD, the data can be reference as a double dimensioned array IDATA(IX,IY) where IDATA is dimensioned the same as the 2-D analyzer, IX is the X direction (i.e. the absissa) and IY is the ordinate.

Example: If one is trying to read in an analyzer which was dimensioned 64x 256 one would have

```
DIMENSION IDATA(256,64)
        :
Call CREAD(IDATA,NCHAN,NRUN,NERR)
     DATA(IX,IY)
        etc
```

RECORD

RECORD,120 — Causes each magnetic tape record written by the event recorder interrupt routine to contain 120 bytes.

RECORD,240 — Causes each magnetic tape record written by the event recorder interrupt routine to contain 240 bytes.

RECORD,? — Types out the current record length

COMMENT: This routine simply sets a parameter which is interrogated by the "DEFINE" routine. It does not change the record length in midstream. In order for this routine to be effective it must be used BEFORE THE OPERATING CONFIGURATION IS DEFINED.

REPLACE,nn

nn	The number of the fit to be replaced.
COMMENT:	This routine simply sets a flag to indicate that the next fit accepted is to replace the nnth of the current analyzer. Example: If one decides that fit number 2 of the current analyzer is unacceptable, one types in "REPLACE, 2", then accepts a new fit. This new fit will replace the previous fit.

RUN,aa,nn

See GO

SCOPE

SCOPE,ON	Causes the task to attempt to acquire one of the storage scopes from the system.
SCOPE,OFF	Causes the task to return a scope to the system.

START,aa

aa	The ADC configuration as typed in the DEFINE command.
START,WX	If a STOP or WAIT command has previously been typed in, this routine restarts the ADCs (i.e. arms and enables the appropriate interrupts), then jumps into EVENT START routine. If neither a STOP or WAIT command has been typed in, this command simply becomes EVENT START.

NOTE: If for some reason you think the computer is not responding to the ADCs, you must first type in WAIT, then START to get them going again.

STOP,aa

aa — The ADC configuration as typed in the DEFINE command.

STOP,WX — Causes the interrupts corresponding to the appropriate ADCs to be turned off (disarmed and disabled), then jumps into the EVENT STOP routine.

TAPE

COMMAND: This command is only meaningful when the input specification field of DEFINE command (the "aa" field) contains the word TAPE. The purpose of this command is to provide the user with control over the tape drive so that multiple scans of the data tape can be performed quickly and easily. It is assumed that the runs on the tape are separated by a single end file (tape mark) and that the first record of each run is a title record.

TAPE,CONTINUE — Starts reading tape after a TAPE,WAIT command.

TAPE, RESTART	Causes the tape to space back one tape mark to the beginning of the run, skip over the title record, and start reading the data.
TAPE,RESTART&WAIT	Causes the tape to space back one tape mark to the beginning of the run, skip over the title record, and stop.
TAPE,REWIND	Causes the tape to return to LOAD POINT
TAPE,SPACE,n	Causes the tape to space n files (forward in n is positive) or n -1 files backwards (if n is negative).
TAPE,WAIT	Causes the code to stop reading the tape.
NOTE:	The commands TAPE, RESTART and TAPE, RESTART&WAIT assume that the end of the current run (RUN 1) has not been reached (i e the message "END OF FILE ENCOUNTERED ON TAPE" has not been typed out since the last title record was typed). If the end of a run has been reached, the tape will be positioned at the beginning of the next run (RUN 2). To force the tape to return to the beginning of the RUN 1 the command TAPE, SPACE, -2 must be used. The commands TAPE, RESTART and TAPE, RESTART&WAIT will simply cause the tape to be positioned beyond the title record on RUN 2.

TITLE,n,t

n — This is either the spectrum number (spectra are numbered from zero) or the word "ALL".

t — Is the title. Its placement in the internal title list is dependent upon n.

1) n is a spectrum number. The title t can be 53 characters in length and starts in column 17 of the title card corresponding to the spectrum number.

2) n is the word ALL. The title t can be 19 characters in length and appears in columns 51 to 69 of all the title cards. Note that this will overwrite whatever was previously in these columns.

COMMENT: At the time a title is output, columns 31 to 49 are checked. If they are blank the date and time at which the run was started along with the time at which the spectrum is being output are filled into the title card.. If any column between 31 and 38 are non-blank, the date is suppressed. If any column between 39 and 49 are non-blank the date and time is suppressed.

NOTE: Columns 70-80 are reserved for the lost overflow count (columns 70,71 and the channel zero live time clock scalar (columns 72-80)

WAIT,aa

aa — The ADC configuration as typed in DEFINE command.

WAIT,WX — This causes the interrupts corresponding to the appropriate ADCs to be turned off (disarmed and disabled).

3.2) SCOPE SWITCHES FOR SETUP MODE
3.2) SCOPE SWITCHES FOR SETUP MODE

a) Switches Affecting The Z-axis.

To provide the user with some feel for the number of counts in channels of the 2-D analyzer, two levels are provided.

a) Upper level (or plane)

Channels contai ng more counts than the upper plane are not displayed.

b) Lower level (or plane)

Channels containing less counts than the lower plane are not displayed.

Seven switches are provided to change the upper and lower planes (i.e. the display levels).

a) INC LOWER	Continuously increments the lower plane at a rate of about 20 (1300 with ZIP switch on) counts/sec.
b) DECREMENT LOWER	Continuously decrements the lower plane at a rate of 20 (1300) with ZIP switch on) counts/sec.
c) DECREMENT UPPER	Continuously decrements upper plane at 20 (1300 with ZIP switch on) counts/sec.
d) INCREMENT UPPER	Continuously increments the upper plane at 20 (1300 with the ZIP switch on) counts/sec.

e) ±	If this switch is on (off) the next two switche will decrement (increment) the lower and upper planes.
f) SINGLE STEP LOWER	Increments or decrements the lower plane and forces an update of the display.
g) SINGLE STEP UPPER	Increments or decrements the upper plane and forces an update of the display.

b) Switches Affecting Polynomicl Fitting

1. DEFINE BAND	This routine expects the user to define three points: 1) A point defining the left end of the band (this point should be to the left of the data). 2) A point anywhere within the data comprising the band of interest. 3) A point defining the right end of the band (this point should be to the right of the data). This routine starts at the left most end, fitting a straight line between this point and the next point on the right. Using the line as a

guide, the routine scans each column of 2-D to find the upper and lower most channels of the data being displayed. The outer boundaries of the data are determined by adding or subtracting YMAX/16 to the upper (lower) most channel number of the data being displayed. Polynomial fits are then generated to these points. The "tightness" of the fits is adjusted by raising or lowering the plane to include or exclude the outer regions of the band.

2) GO FIT

Uses the points accepted with the cross to generate a polynomial fits. There must be N+1 points to generate a fit of degree N. The fit coefficients are put in the "just run" station to be accepted or rejected. If no points have been accepted the previously accepted fit of the current analyzer is moved back into the "just run" station.

3) ACCEPTED FIT/BAND

Adds the fit or band in the "just fit" station to the list of accepted fits (adds two fits when the DEFINE BAND switch was used).

4) REJECT FIT/BAND — Rejects the fit or band in the "just fit" station. To reject the last fit which has been accepted simply hit the GO FIT switch (with no points accepted) then the REJECT FIT switch.

5) D PLAY FITS — Toggling this switch once will cause all the fits for the current analyzer to be displayed.
Toggling this switch a second time will delete the display of the fits.

c) Special Display Switches

1) ANALYZE 2-D — Simulates the RUN mode by generating and displaying spectra which are formed by summing columns of the 2-D analyzer between bands.

2) 3-D DISPLAY — Creates a isometric display of the 64x64 section of the 2-D analyzer. The starting channel for the 3-D are the coordinate of the movable arrows.

d) Switches Affecting Data

1) ERASE — Causes the 2-D analyzer to be zeroed. This switch can be disabled with ERASE teletype command.

e) Switches for Expanding the 2-D Display

The 2-D display can be expanded to display any subsection of the analyzer. The subsection to be displayed is defined with the movable arrows along the bottom and left side of the display. One accepts the new limits with the ACCEPT LEFT, ACCEPT RIGHT, ACCEPT UPPER and/or ACCEPT LOWER switches, then toggles the UPDATE switch. The update routine will use the new limits to generate the new display. To return to the full display, one toggles the RESET EXPAND switch.

f) Switches For The Cross And Arrows

1) DISPLAY PLANE AND ARROWS When this switch is on the upper and lower planes are displayed in the upper left hand corner, the movable arrows and the cross are also displayed. The display of the quantities keeps the scope flood gun at full intensity which decreases the life span of the scope phospher. The life of scope will be greatly increased if this switch if turned off when the planes, cross, and/or arrows are not being used.

2) MOVE CROSS/ARROWS When this switch is on (off) the control switches move the cross (arrows).

g) Switches For Accepting And Rejecting Points

k) REJECTS ALL POINTS	Sets the number of accepted points to zero.
2) REJECT POINT	Deletes the last accepted point.
3) ACCEPT POINT	Adds the current X and Y coordinates (15 bits worth) to the array of accepted points. One can accept up to 256 points.

h) Motion Switches

1) FAST	Increases the speed of the cross or arrows by a factor of five.
2) ZIP	Increases the speed of the cross or arrows for a factor of 16.
3) LEFT/RIGHT	Moves the cross or arrows left and right.
4) UP/DOWN	Moves the cross or arrows up and down.

3.3 TELETYPE COMMANDS FOR ANALYZER 2-D MODE

PLOT, rr

rr — 0-4 digits speciying the run number. If left blank the previous run number is used.

COMMENT: A file containing title cards and CPUNCH decks for the mass spectrum and the spectra obtained by summing each column of the two dimensional analyzer between each of the bands is created. This file is then used as input for the task NDPLOT which plots the spectra in histogram form on the Calcomp plotter. The titles output for the spectra are those which have been typed in using the "TITLE" command.

PRINT, rr

rr — 0-4 digits specifying the run number. If left blank the previous run number is used.

COMMENT: The mass spectrum and the spectra obtained by summing each column of the two-dimensional analyzer between each of the bands are printed out in FORMAT (10I8). A running sum is output at the end of each line. The titles output with these spectra are those typed in using the "TITLE" command.

PUNCH, rr

rr — 0-4 digits specifying the run number. If left blank the previous run number is used.

COMMENT: Title cards and CPUNCH decks of the mass spectrum and spectra obtained by summing each column of the two-dimensional analyzer between each of the bands are output on the card punch.

FILE, rr

rr — 0-4 numeric characters specifying the run number. If left blank the previous run number is used.

COMMENT: A file containing the title cards and CPUNCH decks for the mass spectrum and the spectra obtained by summing each column of the two-dimensional analyzer between each of the bands is created. The control cards:

```
!JOB ACCOUNT,NAME
!ASSIGN M:BO (FILE,NAME RUN rrrr),(SAVE,31DEC99)
!COPY SI,BO
```

are appended to the file. This file is then turned over to the JOB SCHEDULER, which in turn starts a JBPM monitor which creates the file on the permanent file disk. NOTE: See the explanation given with the FILE command for SETUP mode, in particular the last paragraph.

<u>MOIRAE</u>, rr

rr	0-4 digits specifying the run number. If left blank, the previous run number is used.
COMMENT:	A file containing the title cards and CPUNCH decks for the mass spectrum and the spectra obtained by summing each column of the two-dimensional analyzer between each

of the bands is created. Thie file is then used as input for the task MOIRAE providing analysis of the spectra. Note that these spectra are "snap shots" in that they are copies of the data at the time the command is typed in.

PUNCH&PLOT, rr

rr	0-4 numeric characters specifying the run number. If left blank, the previous run number is used.

See the PUNCH and the PLOT commands.

PUNCH&MOIRAE, rr

See the PUNCH and the MOIRAE commands.

3.4 TELETYPE COMMANDS FOR RUN MODE

ERASE,aa,s,s,s

aa	ADC configuration is typed in the DEFINE command.
s	The spectrum number to be erased (if no spectrum numbers are specified all spectra will be erased).

COMMENT:	This routine disarms and disables the interrupts for the ADCs, zeros out the appropriate spectra, records the time, and arms and enables the interrupts for the ADCs.

FILE, rr,s(C1,C2),s(C1),s,...

rr	0-4 numeric characters specifying the run number to be punched on each of CPUNCH decks.
s	The spectrum number (spectra are numbered from zero). If no spectrum numbers are input, all the spectra are output.
C1	First channel to be output.
C2	Last channel to be output.
	If C1 and C2 are not input, the whole spectrum will be output.
	If just C1 is input the channels from C1 to the end of the spectrum are output.
COMMENT:	A file, containing title cards and a CPUNCH decks for each of the spectra specified in the FILE command, will be generated.
	The control cards

```
!JOB ACCOUNT NAME
!ASSIGN M:BO (FILE,NAME RUN rrrr),(SAVE,31DEC99)
!COPY SI,BO
```

are added to the beginning of the file. This file is turned over to the JOB SCHEDULER, which in turn starts a JBPM monitor which creates the file on the permanent file disk.

NOTE: See the explanation given with the FILE command for the SETUP mode, in particular the last paragraph.

PLOT,rr,s(C1,C2),s(C1),s,s

See FILE above for definition of terms.

COMMENT: A file, containing title cards and CPUNCH decks for the spectra indicated in the PLOT command is created. This file is then used as input for the task NDPLOT which plots the spectra in histogram form on the Calcomp plotter. The titles output for these spectra are those which have been typed in using the "TITLE' command.

PRINT,rr,s(C1,C2),s(C1),s,...

See the FILE command for a definition of terms.

The spectra specified are printed on the line printer.

PUNCH,rr,s(C1,C2),s(C2),s,...

See the FILE command for a definition of terms.

COMMENT: Title cards and CPUNCH decks for the spectra indicated are punched on the card punch.

MOIRAE,rr,s(C1,C2),s(C1),s,...

See FILE command for a definition of terms.

COMMENT: A file, containing title cards and CPUNCH decks for the spectra indicated in the MOIRAE command, is created. This file is then used as input for the task MOIRAE which provides analysis of the spectra. Note that these spectra are "snap shots" in that they are copies of the data at the time the command is typed in.

Since only one copy of MOIRAE can be used at a time, TOOTSIE will save up to 12 MOIRAE requests, starting a new copy each time the previous copy exits.

PUNCH&PLOT,rr,s,(C1,C2),s(C1),s,...

See FILE command for definition of terms.

COMMENT: See the PUNCH and the PLOT commands for an explanation of what is done.

PUNCH&MOIRAE,rr,s(C1,C2),s(C1),s,...

See FILE command for definition of terms.

COMMENT: See the PUNCH and the MOIRAE commands for an explanation of what is done.

MAXIMUM,OO,s,a,C1,C2

OO	Indicates whether the checking is to be started (ON) or stopped (OFF).
s	The spectrum number to check (spectra are numbered from zero).
A	The total number of counts in the peak.
C1	The left most (lowest channel) channel of the peak.
C2	The number of channels in the peak.

MAXIMUM,ON,3,2000,900,30	This will cause the task to come in every 30 seconds, add up the counts of spectrum 3 between channels 900 and 930, check if this sum is greater than or equal to 2000. If the sum exceeds 2000 a message is typed out and the alarm is sounded.
COMMENT:	This command, together with the ALARM command inform the user when either a specific amount of time or a specific statistical accuracy is achieved.
MAXIMUM,OFF	Turns of the peak checking routine.

<u>RETURN</u>,aa

aa	ADC configuration as typed in the DEFINE command.
	This command returns the task to the SETUP mode of operation.
COMMENT:	All sepctra are destroyed when this command is typed in.

<u>SPECTRUM,n,?</u>

n	The number of bands in ANALYZER 1 used for double identification. This field is deleted if double identification is not used.

SPECTRUM,?

This routine types out the spectrum number associated with each pair of fits of eac h analyzer. If one were using routing and one had 6 fits on ANALYZER 1, four fits on ANALYZER 2 and 2 fits on ANALYZER 4, this routine would be typed out:

```
ANALYZER=1, FITS=1,2, SPECTRUM=0
ANALYZER=1, FITS=3,4, SPECTRUM=1
ANALYZER=1, FITS=5,6, SPECTRUM=2
ANALYZER=2, FITS=1,2, SPECTRUM=3
ANALYZER=2, FITS=3,4, SPECTRUM=4
ANALYZER=4, FITS=1,2, SPECTRUM=5
```

3.5 SCOPE SWITCHES FOR RUN MODE

3.5.a) Description of RUN mode scope display

The scope can display either one spectrum using the entire height of the scope or can split the screen in half displaying one spectrum one each half. Several numbers will appear along the left side of the screen (assuming the DISPLAY PLANES and ARROWS switch is on). Starting from the upper lefthand corner these numbers are:

1) Adjustable upper spectrum counts limit. When the UPPER SCALE switch is on this number is used as the number of counts full scale in computing the scope coordinates of each channel.

2) Adjustable lower spectrum counts limit. This number serves the same purpose for the lower display.

3) Counts full scale for upper spectrum. This number will be the highest number of counts in the protion of the spectrum currently being displayed (if the SELF SCALE switch is on, this number will be the same as the upper spectrum counts limit).

4) Spectrum number: The convention followed is that a spectrum number of minus one (-1) is used to inhibt the display of this spectrum. When the upper display is inhibited the lower display will expand to fill the entire screen.

5) Counts full scale for lower spectrum. This has the same meaning as the counts full scale for the upper spectrum.

6) Spectrum number: Again (-1) minus one suppresses the display of this spectrum and the upper display will expand to fill the entire screen.

A movable arrow will appear below each spectrum. These arrows indicate the channel number to which they point and are used to expand the spectra.

3.5.b) Description of RUN mode scope switches

In this section the switches which have different meaning in RUN mode which have a second label (of a different color) appearing below the label used in SETUP mode will be described.

1) SINGLE STEP LOWER If the $\mp$ switch is off (on), the <u>upper</u> spectrum number is incremented (decremented by one. If

the new spectrum number exceeds the number of spectra defined, the spectrum number is reset to zero.

2) SINGLE STEP UPPER If the $\mp$ switch is off (on), the lower spectrum number is incremented (decremented) by one.

3) SELF SCALE If this switch is on the adjustable counts limits are used to determine the display scales. If this switch is off the highest number of counts is used.

4) SCALE Divides the upper counts limit of the upper display by two.

5) SCALE Divides the upper counts limits of the lower display by two.

6) MOIRAE Sends the current display over to the analysis task MOIRAE. Hitting this switch many times will start many copies of MOIRAE.

7) FWHM This routine expects the left and right sides of up to 50 peaks to have been defined by accepting a point to the left and right of each peak. The centroid, standard deviation from the centroid

$$SD = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n(n-1)} ,$$

the area of the peak obtained by summing all the channels between the peak limits (no background subtraction), and the full width at half maximum (FWHM) are typed out. If the channel zero option is implimented, the ratio of the peak area to the number of counts in the live time clock is also output.

3.6) HARDWARE PROBLEMS

The failure of certain devices can put the experimenter in the frustrating position of having an operational cyclotron and computer, but an inoperable card punch, teletype, magnetic tape drive, and/or ADC. Provisions allowing the computer to be used when some of these devices fail have been included. In this section these problems and their solutions are discussed.

a) CARD PUNCH FAILURE If the card punch becomes unuseable two options are available:

1) The PUNCH command can continue to be used. Decks will be stacked on the card punch symboint queue and (assuming the system does not crash) will be output when the punch becomes operational.

2) The FILE command can be used to store the data on the permanent file disk (see note of caution discussed in SETUP mode FILE command). The data can be used in a FORTRAN program with the appropriate assign cards and/or can be copied to card punch when it is operational.

b) TELETYPE FAILURES In case the users teletype (TY5) fails control can be transferred to the operators console (OC) with the ACTIVATE command (typed in on the OC or TY5).

The present operating system requires that the OC always be operational. If the OC becomes inoperable, JANUS will quickly grind to halt. TOOTSIE, therefore assumes that the OC is always available. Under normal conditions messages can be typed into TOOTSIE on either teletype, but all output appears on TY5. If these messages cannot be output, TOOTSIE's internal buffers will soon fill and communications will be lost. (The only way out of this is to type !X ID# on the OC.) It is advisable that the ACTIVATE command be used to transfer output over to the OC as soon as problems with TY5 are recognized.

c) MAGNETIC TAPE PROBLEMS The tables on the following pages list all the known ways in which the read only and write only tape handlers can fail along with the available corrective procedures.

It should be pointed out that TOOTSIE can get itself into a state where it will not accept teletype input (i.e. all attempts to communicate using the assigned prefix will be answered with the response "MESSAGE LOST TRY AGAIN"). When the task is in this state, it can only be aborted by typing in MTO ABORT (all corrective measures should be tried before the task is aborted since most errors are recoverable). This rather awkward situation can only occur when a tape drive fails while data is being event recorded. TOOTSIE will wait for all tape operations associated with a given command to be completed before another command is accepted. If the tape drive becomes inoperable, the task will wait forever, necessitating the ABORT command.

d) ADC PROBLEMS ADC problems tend to fall into two categories: Real hardware failures (<E%), and ADC logic problems induced by experimentors arbitrarily flicking switches on the ADC logic box while data is being processed by the ADC's (>95%). The first class of problems require technical assistance and will not be discussed. The second class of problems, once recognized are easily solved. The problems are:

1) The most common symtom is one or more of the ADCs in a group indicate 100% dead time and no data is being accumulated. Check:

1) All ANALYZE switched are on.
2) Only the switches for the ADCs specified in the DEFINE command are in the group, and that all the specified ADC's are in the group.

If these conditions are satisfied, the ADC hangup can usually be cleared by removing the input cables, simultaneously moving all the switches on the ADC logic box (which were in the group) to the center position and then simultaneously moving them back into the group. If this doesn't work, one can try placing all the switches in the center position and toggling the ANALYZE switch for each ADC, then returning them to a group. If these procedures fail, call for help.

2) A second commom symtom is that the computer is suddenly hung by the data taking program. This can occur with TOOTSIE when event recording and/or redundant identification are used. It is caused by the experimentor not putting all ADCs being used in a group.

3) Counts appearing along the axes of the 2-D analyzers and in channel zero of 2-D analyzers. This is the result of timing problems between the input signals. The ADCs require that the maximum of all pulses lie within 1.2 μsec.

4) Data being read by the ADCs but not appearing on the 2-D. This results when data presented to the computer exceeds the conversion gain specified in the DEFINE command (if no conversion gain is specified, the maximum accepted is equal to the number of channels per spectrum). This problem can also arise in the XE/X (XE&XE) routine if XE exceeds E or in the E*T**2 routine if the normalizations are improperly chosen.

TAPE COMMANDS

MTX GO	Causes the tape to retry the previous operation.
MTX ABORT	Causes the current tape operation to be aborted and the task to exit leaving the tape right where it is.
MTX DOWN	Causes the tape to abort the current operation and the task to exit. JANUS is also informed the tape is not operating properly, thus removing it from the system.

MESSAGE	MEANING	ACTION
MTX HUNGRY	Tape Manual.	Press "START" button on tape unit.
MTX INTRLK	Tape is non-operational because of 1) Power off, 2) door down, 3) tape is rewinding.	Correct condition and type in MTX GO.
MTX RATE ERROR	IOP is overloaded.	Type in MTX GO once and if error repeats contact computer personnel.
MTX WRITE PROTECTED	There is no write ring in the tape reel.	Insert write ring and type in MTX GO.
MTX FULL	Tape has passed end-of-tape indicator.	Mount new tape and type in MTX GO. Caution: Do not exit the task until a new tape has been mounted or the data at the beginning of the tape may be destroyed.
MTX BAD TAPE	The previous operation was retried 10 times without success.	Type in MTX GO once; if error still exits contact computer personnel.
MTX MISSING	Neither tape unit is displaying the number being used by EVENT.	Set the dial to the correct number and type in MTX GO.