Nancy Smith

THE ISEE 3 DATA PROCESSING SYSTEM

GAMMA RAY BURST EXPERIMENT

12/79

Contributors

The Gamma Ray Burst (GRB) experiment was launched August 12, 1978 from Cape Canaveral, Florida. The scientists involved with the GRB experiment are Dr. Thomas Cline, Dr. Bonnard TeeGarden, and Dr. Upendra Desai, all from NASA, Goddard Space Flight Center. The design engineer for the GRB experiment was Mr. Guido Porreca of Goddard.

The Data Processing System (DPS) was designed and developed by Code 664 of Goddard, headed by Mr. Gerald Muckel. The Lead Programmer was Ms. Jenny Jacques, Code 664 of Goddard, with significant contributions by Ms. Mary Esfandiari and Mr. Andrew Pelletier, both of Code 664,Goddard. Later in the document, each program is identified, with the particular programmer named.

Table of Contents

The ISEE-3 Gamma Ray Burst Experiment

SCIENTIFIC OBJECTIVES.

This paper describes the technical properties, operation and software system overview of an experiment launched on the ISEE-3 spacecraft which is intended to search for narrow lines in the spectra of gamma-ray bursts (see Figure 1). At the heart of the experiment is a radiatively-cooled Germanium solid-state photon detector. The instrumentation is capable of storing the entire spectrum of all but the largest bursts in the energy range 0.05-6.5 MeV. In addition, it analyzes the signals from two CsI detectors in two other experiments on the spacecraft and records event time histories from these to a few millisecond accuracy. A background mode permits spectral analysis during quiet times and will allow the determination of physically-interesting upper limits for narrow lines in the diffuse gamma-ray background radiation.

The ISEE (International Sun-Earth Explorer) Mission is a three-spacecraft series intended to study the spatial and temporal variations of the Earth's magnetosphere as influenced by the sun and associated interplanetary conditions. The third spacecraft, ISEE-3, which was launched on August 12, 1978, is in a halo orbit about the Lagrangian point along the Earth-sun line 230 Earth radii inward towards the sun. The experiment on board with which this paper deals is the Gamma Ray Burst Spectrometer (GRB). The scientists and engineers associated with it are B.J. Teegarden, G. Porreca, D. Stilwell, U.D. Desai and T.L. Cline from NASA/Goddard Space Flight Center, Laboratory for High Energy Astrophysics, Greenbelt, MD, and D. Hovestadt, Max-Planck-Institut für Physik und Astrophysik, Garching b. Munchen, W. Germany. The members of the GRB software team are G.A. Muckel, supervisor; J. Jacques, lead programmer; A. Pelletier, programmer; and M. Esfandiari, programmer.

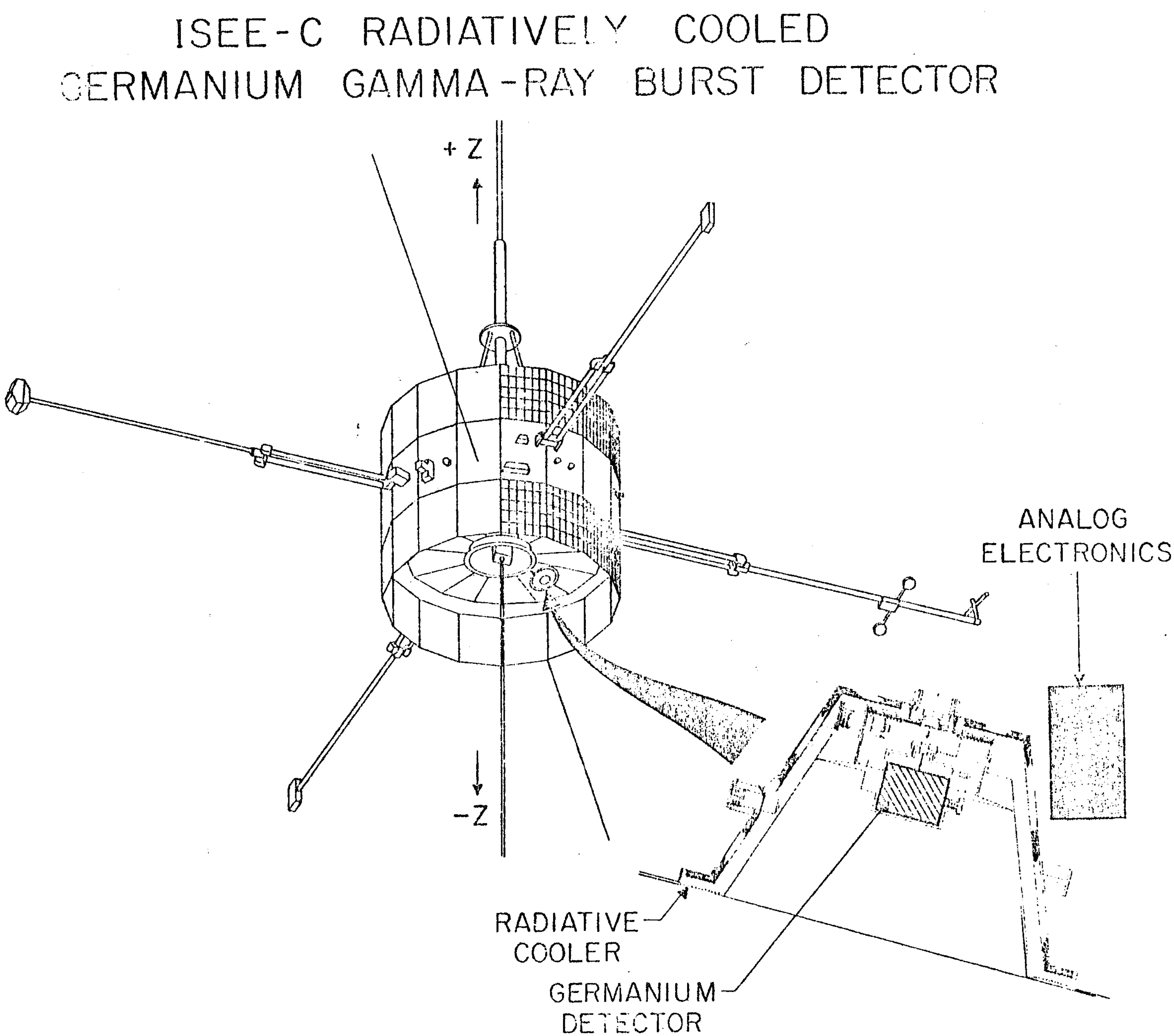

ISEE-C RADIATIVELY COOLED
GERMANIUM GAMMA-RAY BURST DETECTOR
+Z
-Z
ANALOG
ELECTRONICS
RADIATIVE
COOLER
GERMANIUM
DETECTOR

Figure 1

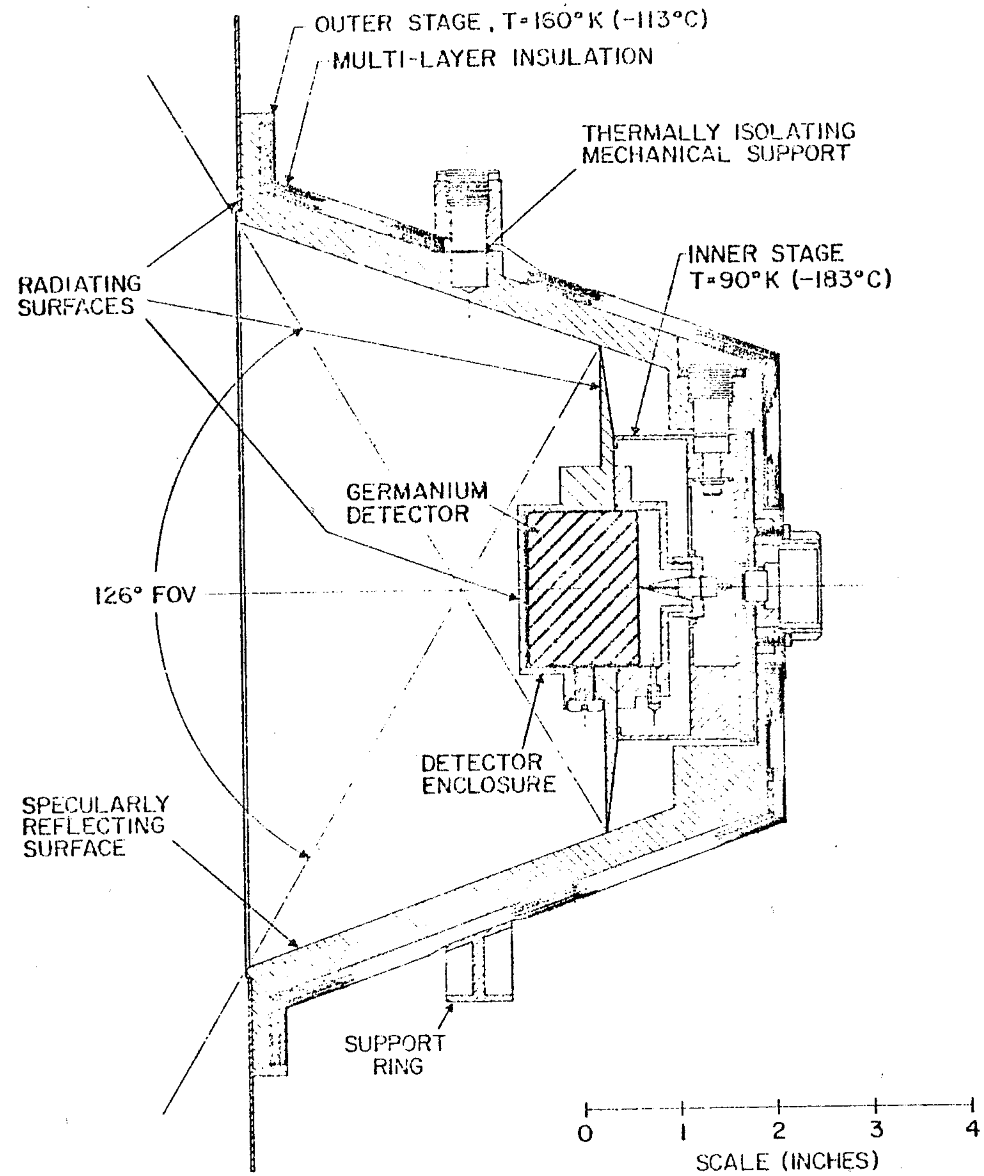
THE GAMMA RAY BURST EXPERIMENT (GRB).

The ISEE-3 instrument is an augmentation of gamma ray burst instrumentation developed for the ISEE-1 mission. This instrumentation is part of the Hovestadt low-energy charged particle experiment.

A cross-sectional view of the Germanium detector mounted in its cooler is presented in Figure 2. Its basic structure is divided into two parts--an inner and outer stage. The outer stage is designed to operate at an intermediate temperature ($\sim 160^{\circ}$K), and thereby provide a thermal buffer between the sensor (residing in the inner stage) and the spacecraft. The Germanium crystal is housed in the inner stage of the cooler in a hermetically-sealed Magnesium enclosure. The exposed surface is coated with a highly-emissive white paint to allow efficient radiation of heat. The inner stage is mechanically supported in a similar fashion to the outer stage so that a second level of thermal isolation is attained. The predicted equilibrium operating temperature of the inner stage is 100°K (-173°C) (see Figure 3).

In addition to the Germanium detector, there are two other detectors on board the spacecraft whose signals we analyze to give increased sensitivity to gamma ray bursts. These are CsI crystals located in the Hovestadt and Meyer experiments on the ISEE-3 spacecraft. The Hovestadt detector is a small cylindrical CsI crystal from which only time history (count rate) information is obtained. Both count rate and spectral analysis are available from the Meyer crystal which is much larger and has the shape of a truncated cone. This crystal is also significantly larger than the Germanium crystal; and although it has much poorer resolution, it will provide a more accurate record of the time history of the event.

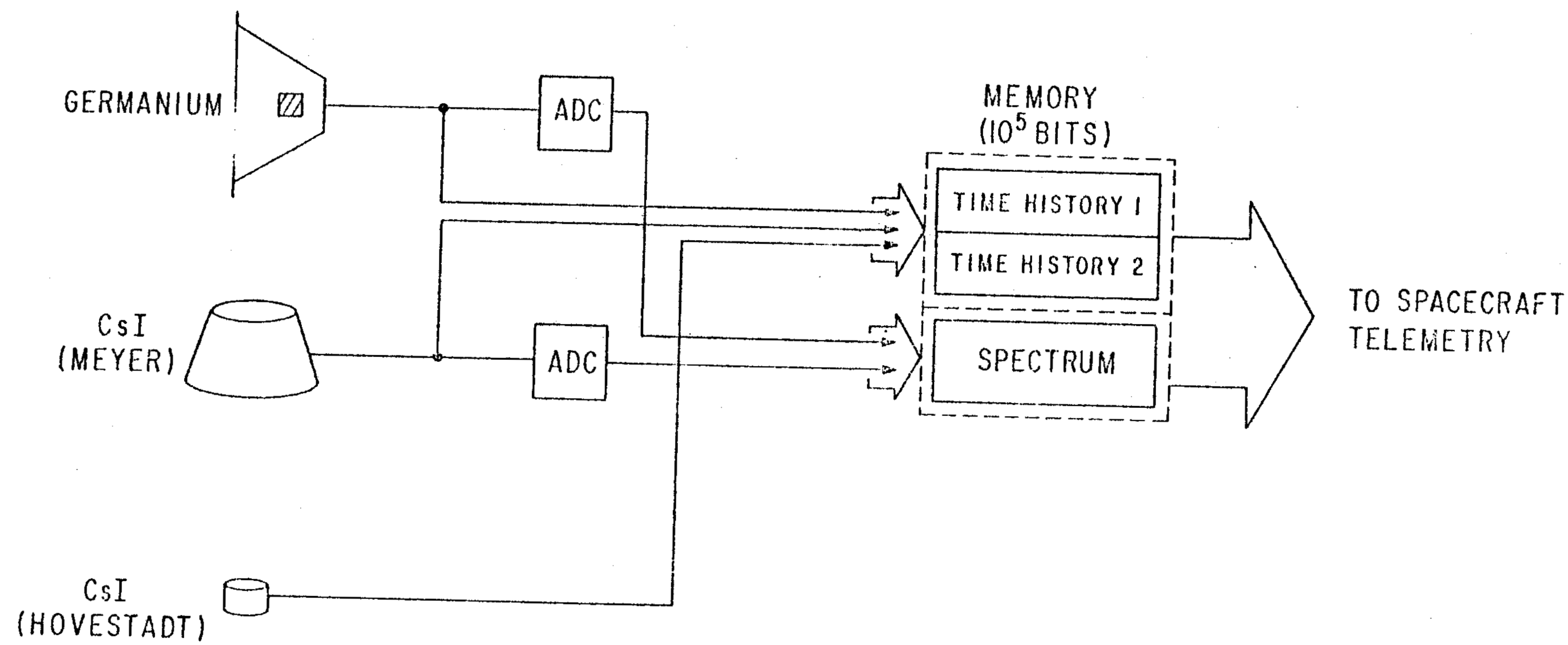
The digital portion of the electronics is physically housed in the Hovestadt experiment. The detection of a burst is accomplished by two independent trigger circuits. These circuits continuously sample the count rate and register a trigger if and when the rate exceeds a value selected by ground command. Any one of the three



ATD–RADIATIVELY COOLED
GERMANIUM DETECTOR FOR
ISEE–C

Figure 2

SIMPLIFIED BLOCK DIAGRAM OF ISEE-C GAMMA RAY BURST INSTRUMENTATION



TIME HISTORY

TIME HISTORIES ARE STORED VALUES OF ELAPSED TIME FOR THE OCCURRENCE OF A FIXED NUMBER N COUNTS.
MEMORY CAPACITY IS 2048 x N DETECTOR COUNTS.
ANY OF THE THREE SENSORS CAN BE CONNECTED BY GROUND COMMAND TO EITHER OF THE TWO TIME HISTORY MEMORIES.

SPECTRUM

INDIVIDUAL 12 BIT PULSE HEIGHTS + TIME TAGS ARE STORED IN SPECTRUM MEMORY. MEMORY CAPACITY IS 3000 EVENTS.
EITHER THE GERMANIUM OR THE MEYER CsI CAN BE CONNECTED BY GROUND COMMAND TO THE SPECTRUM.

Figure 3

sensors can be connected by ground command to each of the trigger circuits. More precisely, these circuits measure the time interval between the occurrence of N counts, where N is also selectable by ground command and can have any binary value between 1 and 128. Time is measured by counting a clock frequency whose value can also be selected from the ground.

Upon the occurrence of a trigger, or Trigger Mode, the burst data is stored in a 10^5-bit memory. This memory is partitioned into three sections, two of which are devoted to recording the time history of the event. The same technique used to define a burst--i.e., storage of a time interval rather than a count rate--is used to record the event in the memory. Each time history partition contains 2048 12-bit words so that 2048xN counts can be stored. It is very unlikely that any burst can exceed the storage capacity of the memory. The third memory partition is devoted to the storage of spectral information. Input from either the Germanium or the Meyer CsI crystal can be stored in this part of the memory as determined by ground control. This section of the memory is organized into 3072 16-bit words composed of 32 bits calendar time every 9 words of data. Each word contains 12 bits of pulse height and a 4-bit time vernier tag. Again, the time resolution is selectable by command and will most likely be $\sim$30 msec.

After the memory is filled it is slowly read out into the spacecraft telemetry stream, with housekeeping parameters and the experiment calendar clock included every 18 bytes of data (see Housekeeping Data Block Structure in the General Maintenance Guide). There are two readout modes: (1) automatic, where readout begins as soon as the memory is filled, and (2) manual, where readout is initiated by ground command.

During quiet times the instrument functions in what is called the Background Mode. In this mode a small fraction of the detected photons are analyzed and stored in a small (36 event) buffer memory. They are then read out in the spacecraft telemetry stream in the same manner as spectral data from a gamma-ray burst. This mode is intended to provide information on the detector calibration and background level. In addition, it can potentially allow the determination of meaningful upper limits for narrow lines in the diffuse gamma-ray background.

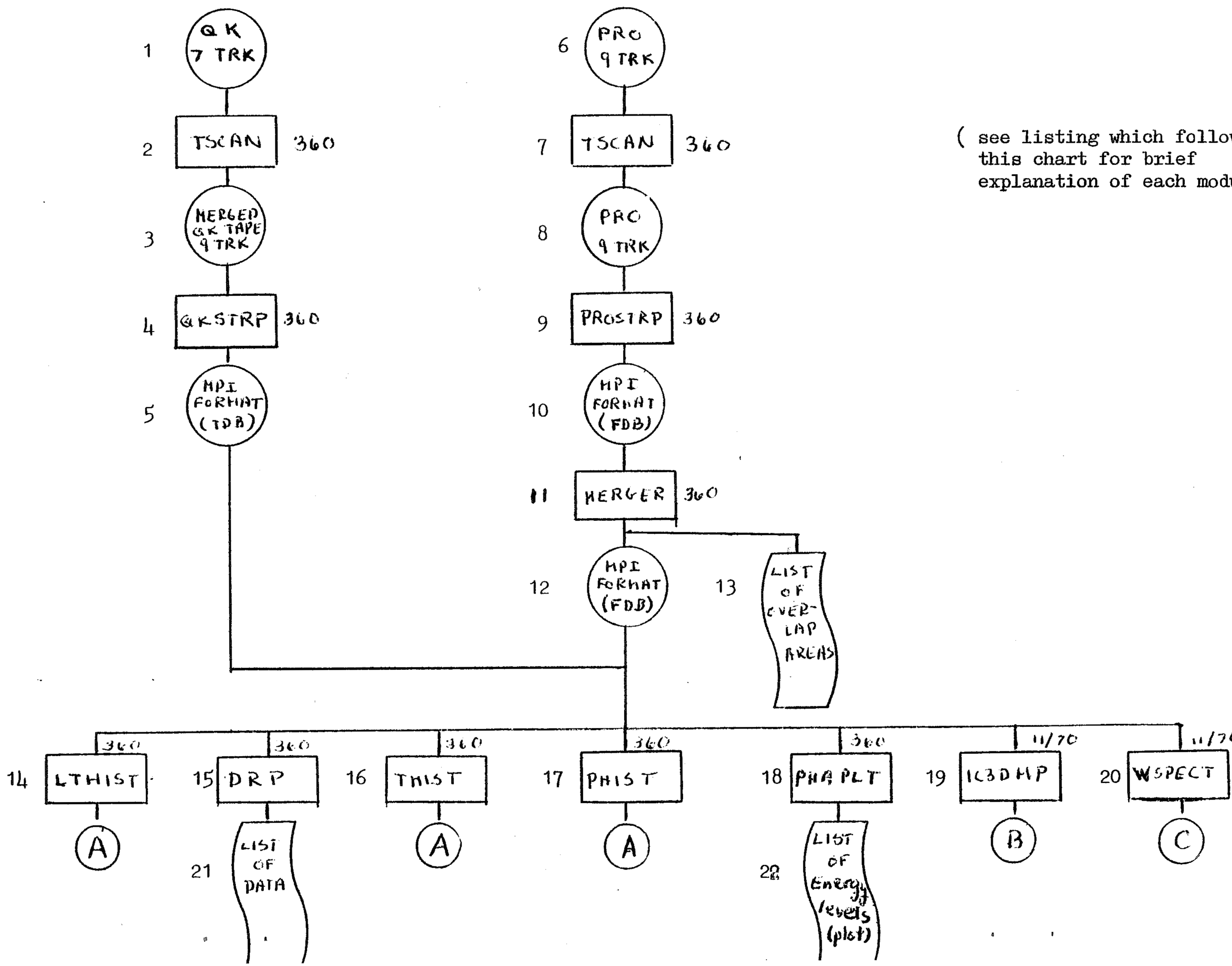
Software System

The software system is designed in two basic levels:

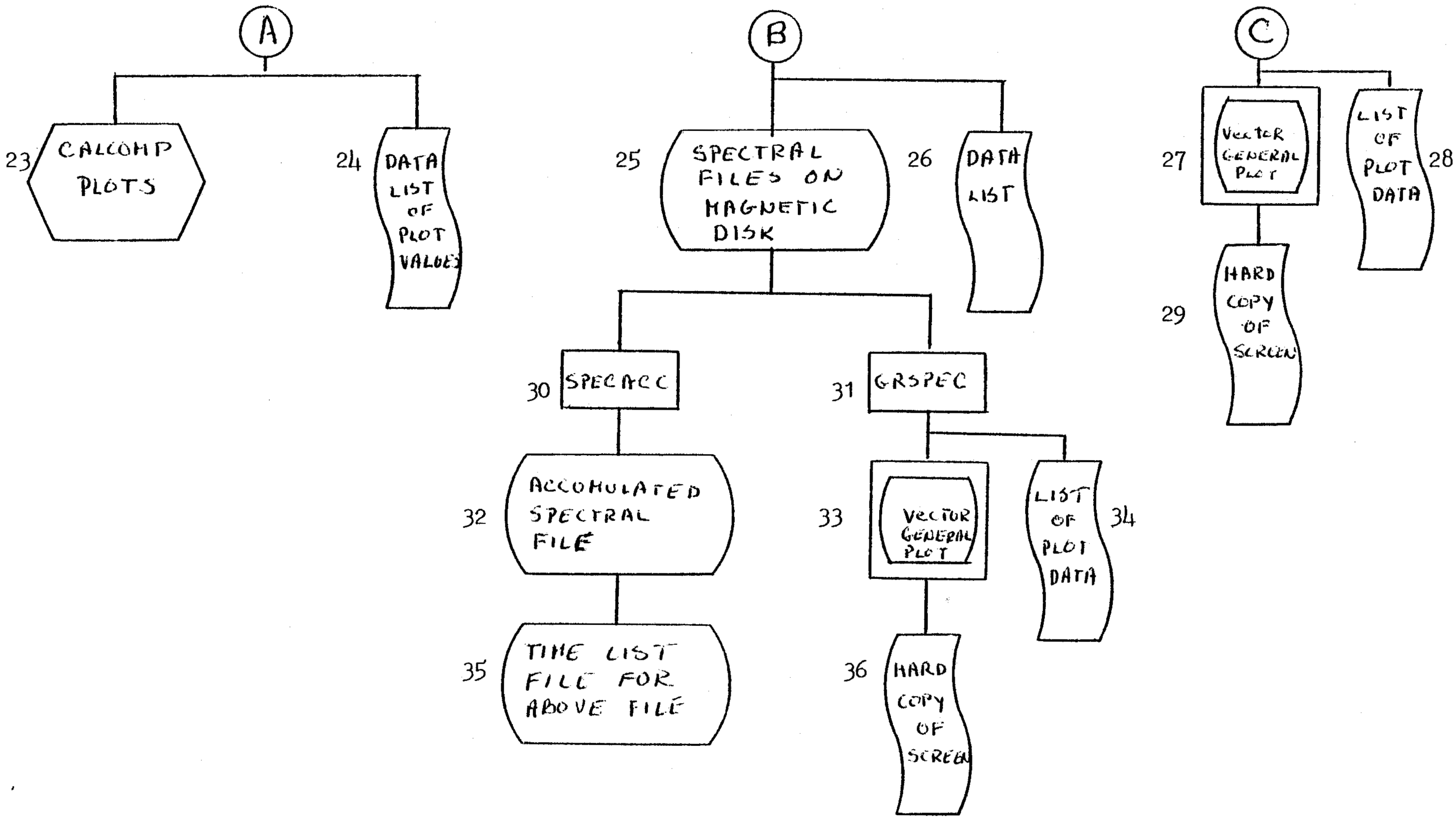
1. Conversion of both Quick Look and Production tapes into MPI format, specified by the Max-Planck-Institut of West Germany when they were to process our data. However, we now process the data before sending it overseas, thus reducing turn-around time. The MPI format tape from Quick Look data is in TDB, or temporary data base form and data quality is often poor with a lack of timing information. The MPI format tape produced from Production data is the final data base (FDB).
2. Scientific analysis programs of the MPI format tape.

ISEE 3 DATA PROCESSING SYSTEM (PAGE 1)

(see listing which follows this chart for brief explanation of each module)



ISEE 3 DATA PROCESSING SYSTEM (PAGE 2)



ISEE 3 Data Processing System

Module	Description
1	The seven track, 556 BPI quick look tape from MSOCC.
2	Copies the tape to a multi-filed 9 track, 1600 BPI tape.
3	The tape containing many quick look files.
4	This extracts the GRB data from the HOH quick look tape.
5	A mulit-filed 9 track, 1600 BPI tape containing only GRB data in MPI format.
6	The 9 track, 1600 BPI tape from U. of Md., known as a decom or production tape.
7	This creates our copy of the production tape.
8	The 9 track, 1600 BPI tape containing HOH data (decom or production tape)
9	This extracts the GRB data from the HOH data.
10	The resulting 9 track, 1600 BPI GRB tape in MPI format.
11	The program eliminates data overlaps, ensuring chronological data.
12	The resulting chronological MPI format tape.
13	The list of any overlap records found.
14	The long Term Time History plot program plots rates, temperatures, and voltages for background data.
15	This lists all GRB data with times, voltage, temperatures, and housekeeping.
16	The Time History plot program. It creates Calcomp plots of time history data of bursts.
17	Pulse Height Analysis plot program. Creates Calcomp plots of PHA calendar progression.
18	This creates an energy histogram of PHA burst data. It plots printer plots of average period vs. channel vs. counts.
19	Creates listings and spectral files of the GRB data on the PDP 11/70.
20	The window spectra plot program collects spectral information in certain energy windows over a time interval and plots rates vs. time
21	The DRP data listing, option for header list and GRB data list.

22 The energy histogram from the printer plotter.

23 The Calcomp plots from LTHIST, THIST, or PHIST.

24 The listings of data for plots for LTHIST, THIST, or PHIST.

25 The spectral files on the PDP 11/70 disk created from IC3DMP

26 The data listing of GRB housekeeping and PHA, time history from IC3DMP

27 The Vector General Interactive Graphics terminal. (VG)

28 The optional listing of data which is plotted.

29 The hard copy of the VG plot, if desired.

30 Spectral accumulation of one or more files

31 This plots the spectral files on the VG.

32 The resulting spectral accumulations file

33 Same as 27

34 Same as 28

35 The time list file catalog of all spectra included in the accumulation file.

36 Same as 29.

ISEE-3 Gamma Ray Burst Experiment

INTRODUCTION.

This guide consists of detailed explanations of the tape and data structure of the entire system. The following will be described:

(1) Quick-look Tape Structure: This is the 7-track tape from Multi-satellite Operation Control Center (MSOCC), Code 510, GSFC.

(2) FDB, TDB and MPI Tape Structure: The FDB (final data base), TDB (temporary data base), and MPI (Max-Planck-Institut) tapes all have the same structure. The TDB, however, lacks some of the header data included in MPI and FDB.

(3) Production Tape Structure: This is the 9-track IPD tape from the University of Maryland which is sent to Germany.

(4) GRB data formats.

QUICK-LOOK TAPE STRUCTURE.

The quick-look tape structure is used to create the GRB tape (see General Users' Guide). Specifications:

Label - no label.
Density - 556 BPI; after copy it is 1600 BPI.
Track - 7 track; after copy it is 9 track.
Logical Record Length - 600 bytes.
Blocksize - 600 bytes.

The quick-look tape contains one file with multiple records, the first record contains only file header data as follows:

BYTE #	MEANING
1- 3	Station ID (BCD)
4- 9	Year, Month, Day (BCD)
10-12	Day # in Binary
13-15	Hr, Mn, Sc in Binary
16-18	Orbit Number (BCD)
19-21	Spin Period (MS)
22-24	Bit Rate
25-27	Spacecraft Clock Day
28-30	Spacecraft Clock H-M-S

The remaining records are structured as follows:

(1) 1 major frame = 4 600-byte records.

(2) 1 600-byte record = 64 minor frames.

(3) 1 minor frame = 9 bytes, as shown below.

1	2	3	4	5	6	7	8	9

Byte #1 is the minor frame counter, 0-255 (4 records x 64 $\frac{\text{minor frames}}{\text{rec}}$ = 256 minor frames).

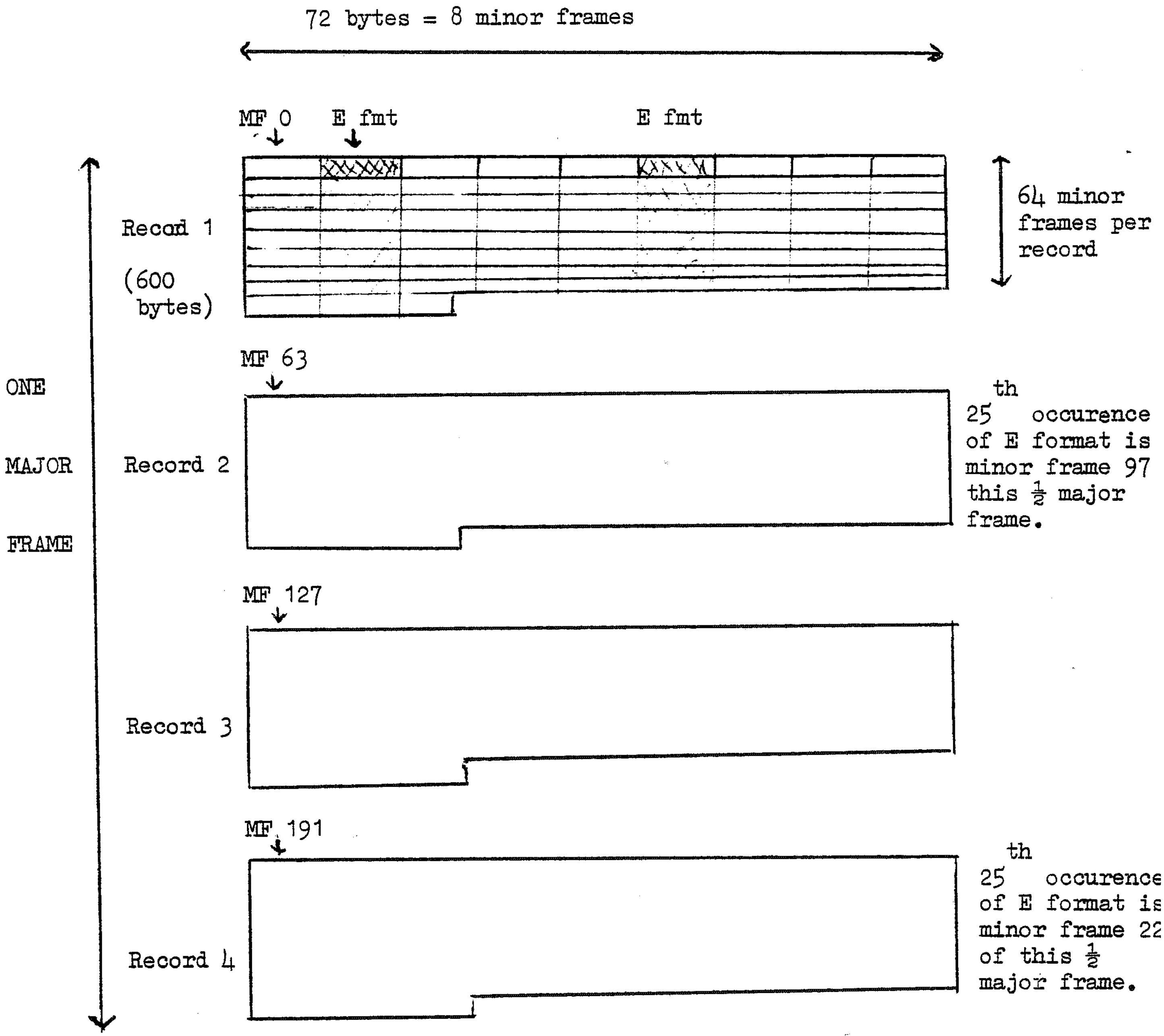
Byte #2 is the format indicator. The first 3 bits determine the telemetry format of the minor frame. Bit 4 is subcom sync, which is Ø for Records 1 and 2, 1 for Records 3 and 4 of every major frame.

Bytes 3-9 are data.

The GRB experiment's data is read out in E format, indicated by 011 in the first 3 bits of Byte #2 in the minor frame. However, GRB data is found only within the minor frames of the 25th to 32nd occurrances of E format in half a major frame (2 records). When a minor frame containing GRB data is located, the GRB data is 12 bits, 4 bits in Byte #5, and all 8 bits of Byte #6.

In a normal telemetry stream (See Figure 7), the E format minor frames occur once every 4 minor frames, beginning with the second minor frame. This would place GRB minor frames in minor frame numbers 97-125 by 4's (in the second record), and minor frame numbers 225-253 by 4's (in the fourth record). Thus there are 16 12-bit words, or 24 bytes, of GRB every major frame. These 24 bytes constitute one data

Telemetry Format



1 Major Frame yields $\left(2* \frac{8*12\text{bits}}{\frac{\text{bits}}{8\ \text{byte}}}\right)$ = 24 bytes of GRB data =

one housekeeping data block

FIGURE 7

block, as described in the Housekeeping Data Block Structure section in this Maintenance Guide.

PRODUCTION TAPE STRUCTURE.

The Production Tape Structure is used to create DB for analysis on the final data base system. Specifications:

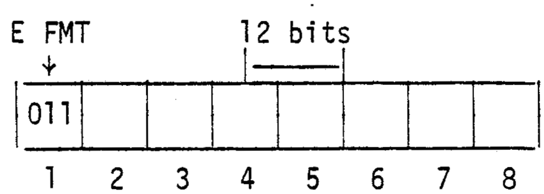
Label = no label.
Density = 1600 BPI.
Track = 9 track.
Logical Record Length = 752 bytes.
Blocksize = 3024 bytes.

The production tape is composed of many multi-record files, each record being 3024 bytes constituting 1 major frame. The first record of each file is a header record.

The record (major frame) is divided into four logical records, each with its own logical record header. Following the logical record header (48 bytes) are 64 minor frame of data, 8 bytes per minor frame. These contain, in certain bits, Gamma Ray Burst (GRB) data.

The data in each minor frame is distinguished by the first 3 bits, which indicates the telemetry format. GRB data is only in E format, which occurs every fourth minor frame starting with the second minor frame. But the GRB data are only in the last occurrences of E format per logical record. Thus there are GRB data in byte ranges 1065-1289 and 2793-2816 in the physical record.

The structure of the minor frame with GRB data is as follows:



From each record (major frame), 16 12-bit GRB words may be extracted. This constitutes 24 bytes, or 1 data block. Tables 1 and 2 describe specific format.

Table 1: FILE HEADER RECORD

Experiment #7, Hovestadt
Tape Type: 9 track, 1600 BPI, Odd Parity

DESCRIPTION	DATA OFFSET BYTES	DATA OFFSET BITS	STARTING BYTE #
GSFC Tag	180	1440	1
Satellite ID Number	4	32	181
Recording Station ID Number	4	32	185
Experiment ID Number	4	32	189
Start Time of Data			
Year	4	32	193
Day of Year	4	32	197
Milliseconds of Day	4	32	201
Stop Time of Data			
Year	4	32	205
Day of Year	4	32	209
Milliseconds of Day	4	32	213
Last Clock in this File	4	32	217
Next File Start Time			
Year	4	32	221
Day of Year	4	32	225
Milliseconds of Day	4	32	229
Next File Clock Start	4	32	233
Percentage of Data Recovered	4	32	237
Number of Minor Frames in File (actual)	4	32	241
Bit Rate (4096, 16384, etc.)	4	32	245
Shipping Group Number	4	32	249
Reel Sequence Number	4	32	253
File Number on this Tape	4	32	257
Orbital Position Data Flag (0,1,2)	4	32	261
0 = definitive 1 = predicted 2 = no orbit	↓ Fill Data to 3024 Bytes ↓		

Table 2: LOGICAL RECORD HEADER

Experiment #7, Hovestadt
Tape Type: 9 track, 1600 BPI, Odd Parity

DESCRIPTION	DATA OFFSET BYTES	BITS	BYTE #
Experimenter ID Number	2	16	1
Day of Year Jan. 1 is 1	2	16	3
Milliseconds of Day	4	32	5
Spacecraft Clock	4	32	9
Average Frame Rate (μsec)	4	32	13
Frame Counter	1	8	17
Fill Flag	1	8	18
Bit Rate Flag	1	8	19
Time Quality Flag	1	8	20
Data Quality Flags	4	32	21
Data Quality Flags	4	32	25
Data Quality Flags	4	32	29
Data Quality Flags	4	32	33
Orbital Position - GEI x	4	32	37
Orbital Position - GEI y	4	32	41
Orbital Position - GEI z	4	32	45
Minor Frame Data (MF0-MF63)	8	32	49
Analog Subcom #1 (0-63)	1 ea, 64 tot.	8 ea, 512 tot.	561
Analog Subcom #2 (0-63)	1 ea, 64 tot.	8 ea, 512 tot.	625
Digital Subcom (0-63)	1 ea, 64 tot.	8 ea, 512 tot	689

NEXT LOGICAL RECORD STARTS WITH BYTE #753; THEN 1054, 2257. FILL DATA UP TO 3024.

4 logical records per physical record of 3024 bytes. At the average bit rate, there are 15,120 major frames/week. This means a week's data will be on 2 tapes.

MPI Tape Structure

The MPI Tape Structure is the final data base structure. Specifications:

Label = no label
Density = 1600 BPI
Track = 9-track
Logical Record Length = 2564 bytes
Blocksize = 5128 bytes

The MPI tape is composed of many multi-record files, each file being from one Production tape, about five days of data.

The first logical record of each file is a header record. Then follows a variable number of logical records, blocked in 5128 byte blocks. See Tables 3 and 4 for byte descriptions.

TABLE 3

File Header Record

Length = 2564 bytes

Byte	Type	Item
1-2	I2	Record ID: Ø = file header record 1 = data record
3-4	--	Spare
5-6	I2	Day of year
7-8	I2	Year
9-12	I4	Milliseconds of day
13-16	I4	Bit rate for this file
17-2564	--	Spare

TABLE 4

Data Record

Length = 2564 bytes

Byte	Type	Item
1-2	I2	Record ID Ø = file header 1 = data record
3-4	--	Spare
5-8	I4	S/C Clock (1/2 seconds)
9-10	I2	Day of year
11-12	I2	Year
13-16	I4	Milliseconds of day
17-20	R4	GSE-X } Orbital Position (Earth Radii meters)
21-24	R4	GSE-Y } Orbital Position (Earth Radii meters)
25-28	R4	GSE-Z } Orbital Position (Earth Radii meters)
29-36	--	Spare
37-40	R4	Spin period (sec * 8195.68)
41-42	I2	Time Quality Flag: Ø = quick look 1 = Smoothed, delay corrected, but unverified by other stations 2 = Smoothed, delay corrected verified after ground time adjusted 3 = Same as 2 but no adjustment needed to ground time
43-44	I2	Orbit flag: Ø = definitive orbit 1 = predicted orbit 2 = unknown
45-48	16* 2 bits	Data quality flags: Ø = fill data 1 = unused 2 = good 3 = excellent (2 bit number per 12 bits data)

Byte	Type	Item
49-72	--	GRB Data Block
73-128	--	Spare
129-132	R4	PM-HK voltage
133-136	--	Spare
137	L1	4 words from temperature subcom, rotates in groupds of 32. (See GRB data formats in the General Maintenance Guide)
138	L1	
139	L1	
140	L1	
141-164	--	Spare
165-2564	*	Repeats groups of 160 bytes starting with byte 5 above

GRB Data Formats

The following will be described:

1. The GRB data character
2. Trigger mode data details
3. Housekeeping data block structure
4. Temperature formats

Data Character

The Gamma Ray Burst (GRB) experiment on ISEE-3 is a piggy-back experiment to Dr. Hovastadt of West Germany's Max-Planck-Institut. It has a very low bit rate, 1.5 bits per second telemetry readout. This data is strung out in the Hovastadt telemetry stream, and must be extracted in phase one of the data processing. The resulting GRB data is divided in packest of 24 bytes, found in one major frame of telemetry. It consists of 2 bytes of housekeeping, 4 bytes of timing, and 18 bytes of data.

When a sudden influx of gamma rays above a specified threshold are detected, a trigger, or burst occurs. The memory storage for bursts is filled rapidly with data of pulse heights and time histories of the event.

Because of the extremely slow bit rate, the memories for both background and trigger are rapidly filled, but slowly read outinto telemetry. For this reason, in background mode, 36 photon counts fill the memory in milliseconds, but are slowly readout over 128 seconds into one major frame. Thus, much data goes undetected. In trigger mode, there are three memories which are simultaneously filled rapidly: PHA (pulse height analysis), TH1 (time history 1), and TH2 (time history 2). They each are recording data prior to the burst in their small "pre-cursor" memory, and upon trigger, their large "post-cursor" memories are filled. This requires only a few minutes, but reading out into telemetry takes 9 hours for PHA and 4.5 hours each for TH1 and TH2.

In background and trigger PHA mode, the data is recorded as 1 channel number per photon detected. Each channel represents a narrow band of energy wavelengths, with 4096 channels in all detectable.

In time history mode, the time it takes to collect a specified number of counts of any wavelength above the threshold is stored.

The experiment has a clock which counts once per 1/8195.68 seconds. This calendar is stored in 32 bits in the data block of background and trigger PHA. In the time histories, the upper 8 bits are truncated and only 24 bits of the calendar are stored.

TRIGGER MODE DETAILS.

Pre-cursor, Post-cursor

The experiment must have a means of storing data in trigger memory during background mode so that in the event of a trigger, some knowledge of what occurred just prior to trigger would be available. Trigger PHA and both time histories each have this means: The pre-cursor memory. It may be thought of as a small area of memory in which a pointer continually writes data. When the end of the area is reached, data is written starting again at the top of the memory. When there is a trigger, the pointer stops immediately and begins writing in a large memory. Later, when the command is given to read the data into telemetry, each memory is read sequentially beginning with pre-cursor PHA. This means that somewhere in each pre-cursor is the newest piece of information written just prior to trigger, followed by the oldest piece of data which had not yet been written over. Thus, unlike post-cursor memory, the pre-cursors of the data in not sequential. The pre-cursor ends before PHA address 583 and time history address 33.

Another feature of trigger data is the anomaly of the PHA data. The housekeeping bytes, read from a different source is always first in the data block. The other 22 bytes of calendar and PHA data is always in a pattern of 4 bytes calendar followed by 18 bytes data, etc. However, when this pattern is read into telemetry, the beginning point may be any 2-word boundary. Thus, although the data block structure should be 2 bytes housekeeping, 4 bytes calendars, 17 bytes PHA data, it will be 2 bytes housekeeping, 22 bytes of calendar followed by PHA data, the PHA data being able to be split between data blocks. A method must be developed to shift this calendar - PHA pattern back to align it with the housekeeping data block. Figure 11 exemplifies this situation.

Futhermore, the pattern of mis-aligned data blocks changes two or three times in one trigger as pre-cursor memory and post-cursor memory are each read out.

It should be noted that the housekeeping information is inserted into the data stream independent of the mode. Thus, as trigger memory is readout, the current housekeeping is inserted, which may not contain the same values as at the time when the trigger occurred.

Trigger PHA Data Block

Two data blocks as they may appear in telemetry.

16-bits	16-bits	16-bits	
Housekeeping 1	Last half of Calendar #1	PHA-Ø	
PHA-1	PHA-2	PHA-3	24 bytes
PHA-4	PHA-5	PHA-6	
PHA-7	PHA-8	First half of Calendar #2	

Housekeeping 2	Last half of Calendar #2	PHA-Ø	
PHA-1	PHA-2	PHA-3	24 bytes
PHA-4	PHA-5	PHA-6	
PHA-7	PHA-8	First half of Calendar #3	

Figure 11

HOUSEKEEPING DATA BLOCK STRUCTURE.

The housekeeping data block is the basic unit of information. Figures 8 and 9 shows the 24-byte unit as divided into 2 bytes of housekeeping information, calendar data and GRB data.

The experiment data are derived from four sources: The first source is the background memory, which stores data during quiet times when no burst has occurred. Figure 8 illustrates the data block structure in background mode. The other three sources are utilized when a gamma ray burst has occurred; and the data are read out in trigger mode.

In the trigger mode, the first memory read out is called PHA, pulse-height analyzer, memory. It contains spectral data in sequence, with 2 words (16 bits/word) of calendar time every 11 words of data. This sequence is read out into a 24-byte data block, with the housekeeping word in Bytes 1 and 2 (Fig. 8). However, there is no synchronization such that the calendar is always in Bytes 3-6, as in background mode. Thus the data blocks must be thought of as one continuous stream of 2 words calendar, 9 words data, etc., where the data after the housekeeping word may begin with any of the 11 words. This continues until all 3072 words are read out.

The third and fourth data sources are time history memories 1 and 2. The format for these data blocks are represented in Figure 9. The first two bytes are the housekeeping word. The next byte is the location of the memory at the time of the beginning of the data block readout.

These 8 bits are Bits 9-11 of a 12-bit memory, the least significant three bits of the address being truncated.

The next 3 bytes are calendar bytes, and are actually 2 12-bit words. These 2 words are bitwise inverted.

Finally, there are 12 12-bit data words, each of which is the time interval between detection of N photons exceeding a given energy, N being set by ground command.

DATA BLOCK STRUCTURE

Housekeeping Word		M=2^{18}sec Calendar (32 bits) L=2^{-13}sec L			
ΔT0 ΔT3 ΔT6	PHA-0 3 6	ΔT1 ΔT4 ΔT7	PHA-1 4 7	ΔT2 ΔT5 ΔT8	PHA-2 5 8

Figure 8: Spectral Data Block--Background Mode

Housekeeping Word	Mem Addr.	M=2^{10}sec Calendar(24) L=2^{-13}L	
T0 T4 T8	T1 T5 T9	T2 T6 T10	T3 T7 T11

Figure 9: Time History Data Block--Trigger Mode, Time Histories 1 and 2

Each of these four modes may be determined by the A and B bits in housekeeping Word 6 (see Fig. 10) as follows:

A	B	MODE
0	0	Background Spectra
1	0	PHA Spectra
0	1	Time History 1
1	1	Time History 2

Each mode begins on a housekeeping update (HKØ).

Housekeeping Data.

The housekeeping word (16 bits) is composed of many parameters, as illustrated in Figure . The first three bits range from Ø to 7, and identify parameters for that data block. One cycle, from Ø to 7, or 8 data blocks, is called a housekeeping update. The meanings of the bits in the housekeeping word are explained as follows:

Housekeeping Parameters:

Bits 0-2: identifies the housekeeping word.

Bit 3: 0 = background mode, 1 = trigger mode.

	BIT	PARAMETER	MEANING
HKØ	4	Read Ready	0 = memory unfilled, 1 = memory is filled and ready to be readout.
	5- 7	T1	Non-resettable counter which increments with each trigger event in Time History 1.
	8- 9	FT1	Clock frequency flag for Time History 1. Defined as follows: 0 = 1 kHz, 1 = 2 kHz, 2 = 4 kHz, 3 = 8 kHz.
	10-12	N1	Flag for threshold of Time History 1, as follows: 2^{N1} = # counts.
	13-15	MT	Defines what fraction of both time history memories will be read out. 2^{MT+3} except MT=Ø means all memory will be read out.
HK1	4-15	TH1	Value (0-4095) which defines the threshold time lapse between photon events in Time History 1. If the time between photons is shorter, a trigger occurs.

	BIT	PARAMETER	MEANING
HK2:	4	ID	This identifies the meaning of Bits 6&7 of HK2, and Bits 4-15 of KH3 and 7. ID = 0: Bit 5 = 7ΔT; Bits 4-15 = Rates 1&2 ID = 1: Bit 6 = HV; Bit 7 = HT; Bits 4-15 = Rates 3&4 (See HK2, HK3, HK7 for definition of ΔT1, ΔTØ, HV, HT, Rates 1-4.)
	5- 7	ΔT, ID=0	The threshold time lapse between photons for both time histories. A shorter ΔT requires a greater rate of photons in order to trigger. $2^{\Delta T+4}$=lapse, except ΔT=0=4096.
	6	HV, ID=1	0 = GE High Voltage is off, 1 = on.
	7	HT, ID=1	0 = Heater Power off, 1 = on.
	8- 9	TS1	Selects which detector will provide signals to Time History 1. 0,1 = GE, 2 = MEH, 3 = HOH.
	10	Spare	Unused.
	11	Read Enable	0 = trigger memory is not being read out. 1 = trigger memory is being read out, if Read Ready (HKØ, HK4) is 1.
	12	GE, MEH	0 = GE, 1 = MEH, which detector is providing signals to time history generator.
	13-15	FP	Clock frequency flag for PHA and background data. 2^{FP+2}=frequency in kHz, except FP= 1024 kHz, F:=1=2046 kHz, FP=0,7.
HK3:	4-15	Rate 1 or 3	Compressed value which indicates the background counting rate. ID=0=Rate 1=HOH. ID=1=Rate 3=GEL. See IC3DMP for decompression algorithm.
HK4:	4-12	See HKØ	Same meanings as HKØ, except applied to Time History 2.
	13-15	MA	What fraction of PHA data will be readout. 2^{MA+4}=addresses, except MH=0=all addresses, MA=0,7.
HK5:	4-15	TH2	Same as TH1 in HK1, except applies to Time History 2.

	BIT	PARAMETER	MEANING
HK6:	4	HVG	One of the two bits necessary to GE high voltage.
	5- 6	B,A	Determine mode. See figure following this section.
	7	Spare	Unused.
	8- 9	TS2	Same as TS1 in HK2, only for Time History 2.
	10-12	HAT	Analog threshold control of HOH detector.
	13-15	GAT	Analog threshold control of GE detector.
HK7:	4-15	Rate 2&4	Same as Rate 1&3 of HK3 only ID=Ø=Rate 2=GEL, ID=1=Rate 4=GEH.

Calendar.

The ISEE-3 GRB experiment has a clock, or calendar, which operates at the same frequency as the spacecraft clock. The two clocks are linearly related by a factor which is based on the values of each clock when it was turned on. There may be some drift over a period of time, or the linear constant may change if either of the clocks is restarted. A program in the final data base system called IC3TIME compares these two clocks with GMT. Because in background and PHA mode there are only 32 bits, units of 10^{-13} sec, the calendar turns over every 6 days, 1 hour, 38 minutes and 8 seconds. In Time Histories 1 and 2, the high order byte is truncated, meaning the calendar turns over every 39 minutes and 8 seconds.

ISEE-C GE Housekeeping Data Format

HK / BIT	0	1 (Trigger 1 Threshold)	2	3 (Rate 1 (Compressed))	4	5 (Trigger 2 Threshold/PHA Address)	6	7 (Rate 2 (Compressed))
0	HK2	HK2	HK2	HK2	HK2	HK2	HK2	HK2
1	HK1	HK1	HK1	HK1	HK1	HK1	HK1	HK1
2	HK0	HK0	HK0	HK0	HK0	HK0	HK0	HK0
3	MEM.	MEM	MEM	MEM	MEM	MEM	MEM	MEM
4	Read Ready	TH1-0	ID=X	11	Read Ready	TH2-0	HVG	11
5	T1-2	1	ΔT-2	10	T2-2	1	B	10
6	T1-1	2	ΔT1=0 HV=1	9	T2-1	2	A	9
7	T1-0	3	ΔT0=0 HT=1	8	T2-0	3	Spare 1= ID=0 Spare 2= ID=1	8
8	FT1=1	4	TS1-1	7	FT2-1	4	TS2-1	7
9	FT1-0	5	TS1-0	6	FT2-0	5	TS2-0	6
10	N1-2	6	Spare	5	N2-2	6	HAT-2	5
11	N1-1	7	Read Enable	4	N2-1	7	HAT-1	4
12	N1-0	8	GE =0 MEH=1	3	N2-0	8	HAT-0	3
13	MT-2	9	FP-2	2	MA-2	9	GAT-2	2
14	MT-1	10	FP-1	1	MA-1	10	GAT-1	1
15	MT-0	Sign	FP-0	0	MA-0	Sign	GAT-0	0

Figure 10

DRP-Data Reduction Program

Purpose | PRGRMR: Jenny Jacques

OVERVIEW

DRP reads a tape in MPI format and performs the following functions.

1. Lists all GRB data blocks and housekeeping parameters.
2. Optionally list only those data blocks which have changed housekeeping values from previous values listed.
3. Lists the major frame header data, such as orbit coordinates and timing. It also marks data gaps, where there is a jump in the spacecraft clock.

INPUT REQUIRED

1 9-track 1600 BPI tape in MPI format (see General Maintenance Guide)

RESTRICTIONS

The IBM 360 /91 or /75. FTIØ package, FORTRANH with PARM = 'XL' option must be used.

PERFORMANCE

Output	CPU/IO per average file 360/75	360/91
Full listing and haeder listing	HO2HO3	HO1HO2
Change-only listing	HO1HO2	001HO2
Header listing only	HOOHO1	HOOOO1

ERROR HANDLING

ERROR	RECOVERY
Bad input tape type	List message and exit
Bad start record or file	List message and exit
Data gap	If header is listed, a message is printed, otherwise no message is printed.
I/O input tape error	Message is issued, record of the major frames skipped.

DATA FLOW

The tape is positioned to the file and record. As each record of 2564 2564 bytes is read in (a major frames), the mode is interpreted according to HK#6 and control is transferred from MAIN to either BKGND or TRGGER for processing of background or trigger mode data respectively. When the mode changes, control is returned to MAIN, where it is transferred to the appropriate routine again.

If the mode is trigger PHA, the calendar is put in sync by subroutine PHASNC. (See "General Maintenance Guide" for detail on PHA calendar position in a housekeeping data block.)

The spacecraft clock is used only as the record is read from tape. It is used to detect data gaps, and also to obtain the ID bit for one housekeeping update. (See General Maintenance Guide for detail on ID bit meaning in the housekeeping word.)

MATHEMATICAL FORMULA

The formula for decompressing housekeeping rates is used. See DRP maintenance guide for detailed equations.

ASSUMPTIONS

1. If housekeeping word #16 is corrupt or absent from telemetry, the previous update's made is assumed to continue.
2. The relation between S/C and ID bit doesn't change within the 16 major frame record. If this should occur, as many as 8 major frames may have mislabeled housekeeping numbers. This relation should change only once or twice a year at most, or whenever the S/C clock is turned off and on.

DRP

Internal Documentation

Overview

DRP, Data Reduction Program, interprets MPI format tapes and lists their contents according to user-input parameters.

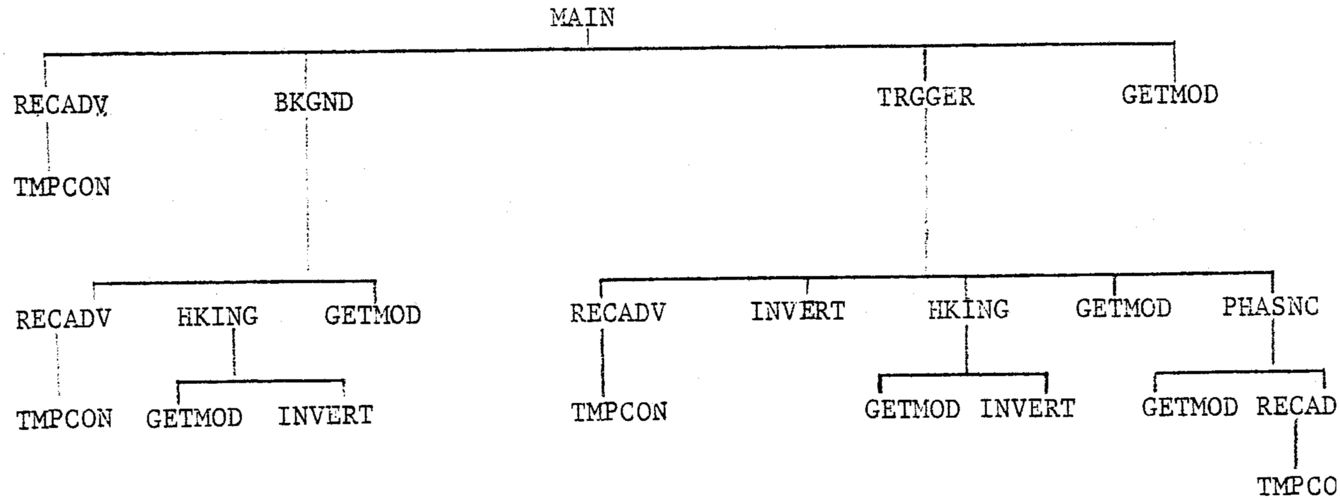
Structure

Routines

The following is a list & brief description of the routines used in DRP.

	Function
MAIN	Spaces to desired start file and record, controls flow of data.
BKGND	Called by MAIN to dump background data.
TRGGER	Called by MAIN to dump trigger PHA and time history data.
PHASNC	Called by TRGGER to align calendar words with the first byte after housekeeping.
GETMOD	Returns pointer to byte of desired data block in order to determine mode or ID bit.
INVERT	Inverts bitwise 24-bits in 12-bit words & returns them in a 4-byte word.
HKING	Interprets 2-bytes of housekeeping for each data block & prints each parameter. Also checks for change in parameters if brief type of listing was desired.
RECADV	Reads in new records listing header data if desired.
RDECOM	Formula to decompress the housekeeping rates into counts/sec.
TMPCON	Converts temperature bytes to degrees Kelvin

Block Diagram



Common Areas

The common areas are first defined and then their usage in DRP is outlined. Arrays hold 64 telemetry data blocks of data.

Common Names

1. /DATA/GMTSC, GMT, FILE9, FILNO, RECNO, BUFPLC, EFILE, EREC, BUFF, HEDLST, DATLST, DAQUAL, IDBIT, END

VARIABLE	TYPE	INITIAL VALUE	MEANING
GMTSC(64)	R*4	Ø	GMT seconds
GMT(4,64)	I*4	Ø	GMT year, day, hour, and minutes
FILE9	I*4	1	Current file number
FILNO(64)	I*2	1	File numbers of the data blocks
RECNO(64)	I*2	Ø	Record numbers of the data blocks
BUFPLC	I*2	Ø	Pointer to data block byte number in BUFF
EFILE	I*2	1	Last file to process

VARIABLE	TYPE	INITIAL VALUE	MEANING
EREC	I*2	999	Last record of last file to process
BUFF(1536)	L*1	Ø	GRB data
HEDLST	L*1	T	Header listing option
DATLST	L*1	T	Data listing option
DAQUAL(64)	L*1	Ø	Quality flags, Ø = bad
IDBIT(64)	L*1	Ø	Holds ID parameter value from housekeeping
END	L*1	F	Signals end of processing

2. /HKPARM/FP, LINCNT, MODE, BRIEF, CHNGE

VARIABLE	TYPE	INITIAL VALUE	MEANING
FP	R*4	Ø	The frequency housekeeping parameter
LINCNT	I*4	Ø	Counts lines for pagination -1 means print new mode lines.
MODE	I*4	Ø	Ø = background, 1 = PHA trigger, 2 = time history 1 3 = time history 2
BRIEF	L*1	F	Brief listing option
CHNGE	L*1	T	Signals whether housekeeping parameters changed

3. /PHACAL/ENDCAL, IFLG

VARIABLE	TYPE	INITIAL VALUE	MEANING
ENDCAL	I*4	---	Pointer to end of PHA data after PHASNC
IFLG	I*4	---	0 = background, time histori 1 = trigger PHA

Occurrence & Use of Commons

Under the USE heading is R, W, or RW for reads only, write only, or reads & writes into the common.

Common Name	Occurrence	Use
DATA	MAIN	RW
	BKGND	RW
	TRGGER	RW
	PHASNC	RW
	HKING	R
	RECADV	W
HKPARM	MAIN	RW
	BKGND	RW
	TRGGER	RW
PHACAL	BKGND	W
	TRGGER	W
	PHASNC	RW
	HKING	R

Other Variables

Each routine is listed with its' main variables defined. Loop increments, the ISTR4-LSTR1 technique (described below), common variables, & namelist variables are excluded.

The LSTR4-LSTR1 technique is used throughout DRP to convert 4-bytes of logical data into a 4-byte integer word. LSTR1 is a logical *1 array dimensioned 4 and equivalenced to ISTR4. Thus, by assigning the logical bytes to LSTR1, ISTR4 is created.

1. MAIN

VARIABLE	TYPE	MEANING
OEFILE	I*4	Holds EFILE when EFILE is set to -1 to signal RECADV to read records without analyzing them.

2. BKGND

(The variables are used on a data block basis.)

VARIABLE	TYPE	MEANING
DT(3,3)	L*1	Holds the delta time values
HK	I*4	Holds housekeeping number
PHA(3,3)		Holds the PHA words
TMP(2)	I*2	Equivalenced with ISTR2 to pick out INTEGER *2 words
ISTR2	I*2	Equivalenced with TMP
CAL4	R*8	Holds calendar of previous data block whose HK was Ø or 4
ISUM	I*4	Sum of delta time to compare with difference in calendar
CALDIF	R*8	Difference between current & previous calendar
CALNDR	R*8	Current calendar value
C4DIFF	R*8	Difference between current & previous calendars, if HK is Ø or 4
DBLOCK	I*4	Data block numbers, 1 to 32 corresponding to current record being analyzed
OLDCAL	R*8	Previous calendar value
SECODY	R*8	Seconds of day
MODE	I*4	-1 = not found, Ø = background, 1 = PHA trigger, 2 = time history 1, 3 = time history 2

3. TRGGER

(Many of the variables are as defined in BKGND and are not repeated here.)

VARIABLE	TYPE	MEANING
OLDK	I*4	Holds previous update's time history number, 1 or 2
THST(3,4)	I*2	The 12 delta time values in each time history data block

VARIABLE	TYPE	MEANING
MEMAD	I*4	Memory address for the data block
MODE2	I*4	Returns from GETMOD with -1 = not found, Ø = background, 1 = PHA trigger, 2 = time history 1, 3 = time history 2
CALTMP	I*4	3 byte time history calendar in proper bit order
OLDPLC	I*4	Keeps the pointer to BUFF so that BUFPLC may be set to Ø to analyze the PHA data. BUFPLC is restored to OLDPLC when time history analysis begins.

4. PHASNC

(Many variables have been defined in BKGND or TRGGER and are not repeated here.)

VARIABLE	TYPE	MEANING
CAL(2)	L*1	The two highest bytes of the PHA calendar
DIFF	I*4	Difference between current & preceeding calendars
		.TRUE. for first time in PHASNC, .FALSE. for the rest of the times
FIRST	L*1	Previous data block's calendar
CALKEP	I*4	Points to byte number in BUFF before the calendar
CALPNT	I*4	Counter to allow 3 consecutive matches of the calendar high bytes before those high bytes are accepted as the true calendar high bytes
NEWPLC	I*4	Pointer for filling BUFF with the data in sync with the housekeeping

5. GETMOD

VARIABLE	TYPE	MEANING
IHK	I*4	Either 2 or 6, depending whether housekeeping #2 or 6 is to be found.
ICNT	I*4	Counts the number of data blocks looked through to search within an update (8 data blocks).
HKNUM	I*4	Current housekeeping number.
MODE2	I*4	If IKH = 2, mode2 is the pointer to housekeeping data block 2 in the update, or -1 if it could not be found. If IHK = 6, mode2 indicates the MODE as described in BKGND variable definitions.
POINTR	I*4	The pointer in BUFF of the beginning of the current data block.

6. INVERT

VARIABLE	TYPE	MEANING
INT	I*4	Current 12-bit word to be inverted.
NUM	I*4	Resulting inverted 12-bit word.
LOG(3)	L*1	The 3-bytes to be inverted as 2 12-bits words.
HOLD	I*4	Keeps the first 12-bit word as the second one is processed.
ICNT	I*4	Ø = done with the first 12-bit word. 1 = done with the second 12-bit word.
RESULT	I*4	The 3-byte word created from the 2 inverted 12-bit words.

7. HKING

In defining many of these variables, the General Maintainance Guide must be referenced for the meaning of the variables used in the definition. The variables which are checked for change between data blocks are stored in a parallel array or variable with the prefix of 0, for "old".

VARIABLE	"OLD" NAME	TYPE	MEANING
A	---		A bit in HK6
B	---		B bit in HK6
N(4)	ON(4)	L*1	(1) = N1, HkØ
			(2) = not used
			(3) = N2, HK4
			(4) = HAT, HKG6
T(4)	OT(4)	L*1	(1) = T1, HKØ
			(2) = delta T, HK2
			(3) = T2,, HK4
			(4) = not used
FP	---	R*4	FP value in HK2, converted to KHz
FT(7)	OFT(4)	L*1	(1) = FT1, HKØ
			(2) = TS1, HK2
			(3) = FT2, HK4
			(4) = TS2, HK6
HZ(8)	---	R*4	The KHz values used to convert FP to KHz
ID	---	L*1	ID, HK2
MT(4)	OMT(4)	L*1	(1) = MT, HKØ
			(2) = FP, HK2
			(3) = MA, HK4
			(4) = GAT, HK6
TH(2)	OTH(2)	I*2	(1) = TH1, HK1
			(2) = TH2, HK5

VARIABLE	"OLD" NAME	TYPE	MEANING
HKG	---	I*4	Housekeeping bytes (16 bytes)
HKG4	OHKG4	I*4	Bit 4 of housekeeping
HKG6	OHKG6	I*4	Bit 6 " "
HKG7	OHKG7	I*4	Bit 7 " "
HKG11	OHKG11	I*4	Bit 11 " "
HKG12	OHKG12	I*4	Bit 12 " "
IPNT	---	I*4	Pointer to housekeeping block #2, returned from GETMOD
NCNT	---	I*4	Counts up to 4 times that the ID bit and IDBIT array don't agree in value.
RATE(4)	ORATE(4)	I*4	(1) = Rate 1 HOH, HKG3 (2) = Rate 2 MEH, HKG7 (3) = Rate 3 GEL, HKG3 (4) = Rate 4 GEH, HKG7
RDRY(4)	ORDRY(4)	L*1	(1) = Read ready, HKGØ (2) = ID, HKG2 (3) = Read ready, HKG4 (4) = HVG, HKG6
HKNUM	---	L*1	Housekeeping number
HKOUT	---	I*4	HKNUM + ID*8; this ranges from Ø to 15 and indicates a complete update. It is used for printing.
HKPLUS	---	I*4	HKNUM+1 used for array subscripts
IDFACT	---	I*4	"Polarity" of the S/C clock bit and the ID bit relationship. If IDFACT is 1, ID bit = S/C bit. If IDFACT = -1, the 2 bits oppose each other, one is Ø, the other is 1.

8. RECADV

VARIABLE	TYPE	MEANING
HK	I*4	Housekeeping number
ID	I*4	ID bit from housekeeping 2
T1,T2,T3	L*1	Holds temperature bytes
EOV	L*1	End of volume indicator
PLC	I*4	Pointer to INBUFF array where the GRB data is located, incremented by 160 each data block
GPLC	I*4	Pointer to BUFF array, incremented by 24 each data block
LREC	I*4	Logical record number
MASK	I*4	Used to create S/C clock excluding the high order byte
MODE	I*4	Ø = background, 1 = trigger
PAGE	I*4	Indicates pagination when new file is encountered
PLC1	I*4	Pointer to INBUFF, beginning 4 bytes bytes into record & incrementing by 160
PLC2	I*4	Pointer to INBUF2, beginning 2 words into record, incrementing by 80
PLC4	I*4	Pointer to INBUF4, beginning 1 word into record, incrementing by 40
STOP	L*1	Indicates end of data processing encountered so next call to RECADV uses second half of BUFF as input record
TEMP	I*4	Used to retrieve high bit in S/C clock
TIME	I*4	Milliseconds of day from GMT of major frame
COUNT	I*4	Used to count five consecutive 1's in temperature telemetry so as to find true temperatures

FIRST	L*1	Indicates first time through so the double buffering may be started.
INPLC	I*4	Pointer to INBUFF beginning 48 bytes into record and incrementing by 160.
OSYNC	I*4	Previous ID bit sync number (see IDSYNC)
POSIT	I*4	Indicates which quarter of BUFF is filled with the records' GRB data.
SPCLK	R*8	S/C clock for current major frame.
DBLOCK	I*4	Loop variable from 1-16 which is the data block number being processed in the record.
GRBPLC	I*4	Pointer to BUFF for filling in the data.
IDSYNC	I*4	This number is the half seconds difference between the change of the S/C clock bit used to get the ID bit, and the change of the housekeeping update. It is used to cause both to change synchronously.
INBUFF(2564)	L*1	Input logical record from MPI tape
INBUF2(1282)	I*2	Equivalenced with INBUFF to retrieve integer *2 words.
INBUF4(641)	I*4	Equivalenced with INBUFF to retrieve integer *4 words.
REBUF4(641)	R*4	Equivalenced with INBUFF to retrieve real *4 words.
LINCNT	I*4	Counts lines printed for pagination purposes.
MFMISS	I*4	The number of major frames missing in a data gap.
OSPCLK	R*8	S/C clock from last major frame.
SCDIFF	R*8	Difference in half seconds between current S/C clock and previous one.
TMPTUR(3)	R*8	Temperatures 1, 2, and 3 in K°.

9. TMPCON

VARIABLE	TYPE	MEANING
A	R*4	Experiment constants.
B	R*4	Experiment constants.
D	R*4	Voltage of the temperature byte.
R	R*8	First step in conversion, common to all 3 temperatures.
CHNL	L*1	Channel #, or temperature byte value before conversion.
COEFF(5,2)	R*8	Coefficient to the formula for final conversion to K°.

10. RDECOM

VARIABLE	TYPE	MEANING
C		Rate value from housekeeping which is returned as counts/second.

Namelist Variables

Namelist/INPUT/SFILE, EFILE, SREC, EREC, TAPENO, DATLST, HEDLST, BRIEF

VARIABLE	TYPE	DEFAULT	MEANING
SFILE	I*4	1	File with which processing is to begin
EFILE	I*2	1	File with which processing ends, inclusive.
SREC	I*4	1	Beginning record for processing.
EREC	I*2	999	Last record of end file to process.
TAPENO	I*4	1	Final Data Base tape number.

DATLST	L*1	T	T = data listing desired F = no data listing desired
HEDLST	L*1	T	T = header listing desired F = no header listing desired
BRIEF	L*1	F	T = print only those data blocks which have changing housekeeping parameters. F = print all data blocks

Algorithm of Data Flow

The following is an algorithm tracing data flow through DRP. The data is taken from an MPI format tape which contains header data (temperatures, GMT, etc.) and the GRB experiment data in the form of 8 data blocks per housekeeping update. The first time through, step 2 is done twice to double buffer the arrays.

ROUTINE	STEP	ACTION
MAIN	1	Options read in from namelist.
RECADV	2	Record of 16 data blocks is read from tape, or END flag set if done all required data.
RECADV	3	ID bit for each data block is found.
RECADV	4	Header data listed with marks of data gaps is HEDLST desired.
MAIN	5	Mode is found with GETMOD. If the update is background data, steps 6-9 are executed. If it's trigger, steps 10-20 are executed. When END flag is returned as true, the program ends.
BKGND	6	If new record is needed, RECADV is called.
	7	Difference in calendars and PHA-DT words are found.
	8	Housekeeping is listed by a call to HKING.
	9	PHA & DT words are listed. Then to step 6 for next data block.
TRGGER	10	PHASNC is called to synchronize the housekeeping and data words.
	11	Get the difference in calendars and separate PHA-DT words.
	12	Call HKING to list the housekeeping parameters.
	13	List PHA & DT words.
	14	Back to PHASNC step 10 for next record unless time history has started. If so, begin step 15.

15 Get next record if necessary by RECADV.

16 Get calendar by call to INVERT, & calculate difference in calendars.

17 Separate DT words.

18 List housekeeping by HKING.

19 List DT words.

20 Back to step 15 or, if done with trigger, return to MAIN.

DRP
Operator's Guide

Overview

DRP is the Data Reduction Program for ISEE-3 GRB. It creates listings of all the data relating to the GRB experiment which is on an MPI formatted tape. The operator must edit a small TSO file and change the input namelist before submitting the job.

Running DRP

DRP is in load module form in TSO file SEJSS.ISEE3.LOAD(DRP). It is linked with the namelist parameters in TSO file

SEJSS.ISEE3.CNTL(DRPJCL). This file must be edited using QED and the options changed as desired, then the file may be submitted as a job.

Step 1 - Know the namelist parameter values. The following is a list of the namelist parameters and their meaning:

Parameter	Default	Meaning
SFILE	1	File number of MPI tape with which to begin processing.
SREC	1	Beginning record of beginning file of MPI tape to process
EFILE	1	Last file number of MPI to process
EREC	999	Last record number of last file of MPI tape to process
TAPENO	1	MPI tape number
DATLST	T	T = GRB data listing desired F = suppress GRB data listing
HEDLST	T	T = header listing desired F = suppress header listing
BRIEF	F	T = brief listing, printing only changing housekeeping data F = list all GRB-data

Step 2 - Edit the TSO file containing namelist and JCL information.

(TSO session, * indicates computer prompt)

```
* READY
  QED 'SEJSS.ISEE3.CNTL(DRPJCL)'  NONUM
* QED
  TV
  L
```

The computer lists the data set. Using standard TSO change commands, change the following:

1. Jobcard name and CPU, IO time estimate. Each file takes about HOOHO1 on the /75 and HOOOO1 on the /91.
2. Title of run, if desired.
3. Input tape slot number as indicated by the VOL=SER=JSS##.
4. Namelist parameters. If any are left out, the defaults are used as specified in Step 1.

Step 3 - Submit the job

(TSO session, * indicates computer prompt)

```
* QED (of SEJSS.ISEE3.CNTL(DRPJCL))
  SUB  *
* ("JOB SUMBITTED" message)
   END  N
* READY
```

If the file was saved previously the command "SUB filename" is typed while in READY mode. Ex: SUB 'SEJSS.ISEE3.CNTL(DRPJCL)'

Error Handling

If DRP failed to run properly check that the correct tape was in the slot, and that the namelist parameters were correctly specified.

The only error, other than I/O errors on the tape, is if the start record did not exist in the start file as specified.

Additional Notes

1. Jobs with no top priority status should be run on STANDBY basis.
2. If the job times out, restart the run at the start file & record which was last printed. Don't re-run the entire job with greater time estimates!

GRSPEC

Purpose

PRGRMR: Jenny Jacques

Andrew Pelletier

Overview

GRSPEC displays a spectral histogram on the Vector General from a file created by IC3DMP. The horizontal axis is channels, (0 to 4095), and the verticle axis is counts/sec. The following are capabilities of the program:

1. Compression of channels by a binary number.
2. Variable axes scaling.
3. Subtraction of another spectral file.
4. Simple area calculation by choosing two points of the spectrum and calculating the area above and below the line between them.
5. Energy and channel display for individual points light penned.

150.107

GRSPEC

Overview

This program reads a data file of gamma ray spectrum and displays it on the Vector General graphics screen. The data is given in counts per channel where the channel number is a function of wavelength. There are 4096 channels numbered from 0 to 4095. This program also has the capability to sum the counts in adjacent channels, compressing the data. The number of channels summed to yield a data point is called the compression factor. For instance, if the total number of channels to be graphed is 2048 and the compression factor is 4, there will be 512 data points showing on the screen. If the start channel is 10, the first data point will be the sum of the counts for channels 10, 11, 12, and 13. The next data point would be the sum of the counts in channels 14, 15, 16, and 17.

Some additional features of this program are:

1. Data graphed semi-log as well as linear.
2. Data values maybe listed on the line printer.
3. Interactively finding the channel number of a light-penned point with an optional peak search.
4. Interactively finding the area beneath and above a line defined by the light pen.
5. Subtraction of another spectral file.

This program was designed for use with the ISEE-3 Gamma Ray Burst Experiment. The data files are created from the MPI tape with IC3DMP. The file has 4 header records followed by 512 64-byte records. Each record has 8 count values, each 8 bytes long in integer format.

Any changes to this format can be made by changing formats statements 600 and 982. It is assumed that the compression factor is an integral factor of the total number of channels. Except for this restriction, there is no need for the compression factor to be a binary number. The read routine assumes that the last channel to be read is not greater than 4095.

For this program, a data point is a summed group of adjacent channels. The compression factor is the number of channels per data point and is input selectable.

The time interval for the output is selected when the disk file is made. This program does not allow direct access to the catalog.

DAT = Data Array.
TEST = Array of all input to test for errors.

SCALE = Y for log scale.
LP2 = Y for light pen of energy.

PEAK = Max count in deviation.
XF = Ending channel number.
VAL - Count value maximum.
XPOS = X pos. of light pen hit.
YPOS = Y pos. of light pen hit.
NAS2 = Y for screen image of graph.
LP1 = Y for light pen of channel #.

Definition of Variables

A. Character arrays

Each item to be displayed on the VG must have a corresponding character array with a zero as last element to be displayed.

Char Array	Number	Purpose
FN	FILE	Store file name
CF	NCOMP	Compression factor
SC	XT	Starting channel
SS	MAX	Maximum of vertical axis
RCHNL	RCHANL	Reference channel
REFEN	RENGY	Ref. energy in keV
DV	DEV	Deviation for peak search
TT	TOT	Total channels
IPKAR	PKAR	Area under peak above curve
ITRPDR	TPDAR	Area under normal curve

B. Other arrays and variables

Variable	Type	Meaning
ANS2(2)	L*1	Answer to 'screen graph'. VG option.
ARRAY(20)	I*2	Used with RQATN to get L-pen position.
ARX1	I*4	L-pen X coordinate.
ARX2	I*4	L-pen X coord for second point in area computation section.
ARY1	I*4	See ARX1, Y coordinate
ARY2	I*4	See ARY2, Y coordinate
AXIS(11)	I*4	Vertical count values output.
CF(3)	L*1	Compression factor (ASCIZ format)
CH(5)	I*4	Horizontal channel numbers output.
DAT(1025)	I*4	The compresses counts. DAT(1) holds the count for the start channel.
DATTOT	I*4	Running total, while compressing.
DEV	I*2	Peak search deviation in data points.
DV(5)	L*1	" " " in CHAR format (ASCIZ).
ENERGY	R*8	Energy of a peak in keV.
FN(16)	L*1	File name.
I4	I*4	A do loop index, usually steps horizontal along graph.
I41	I*4	When used, I41 - I4+1

Variable	Type	Meaning
IC	I*2	RQATN returns function key(+1) here
ICODE	I*2	Same as IC
ICOMP	I*2	Number of channels compressed so far.
IDATP	I*2	Subscript of current data point.
IDIS	I*2	Height of 1 vertical letter (small size).
IFLAG	I*2	IFLAG = 1 IFF there are too many data points to draw horizontals.
INPUT(8)	I*4	One data record as read in, acts as an input buffer.
IPKAR(18)	L*1	Peak area in output format.
IR	I*2	Pointer to the buffer input(8).
IRR(20)	I*2	Same as array, but for area error.
IRRAY(20)	I*2	Same as array, but for second L-pen in computation section.
ITRPDR(20)	L*1	Area under line (shaped like a trapezoid). character format.
TXPOS	I*4	X-coord of peak.
JFLAG	I*2	= 0 if function key menu is on screen.
K	I*4	Number of horizontal lines in log grid.
KFLAG	I*2	= 1 in area mode, 0 in channel mode.
LARRAY(8)	I*2	First channel number of each column on LP:
LIN(8)	I*4	Vertical axis values in character form see AXIS( ).
LOG(8)	I*4	Same as LIN(8).
LP1(2)	L*1	Answer to 'L-pen for channels'
LP2(2)	L*1	Answer to 'L-pen for energy'
MAX	I*4	Maximum vertical axis value.
MAX1	I*4	A copy of AMX that is divided by 10 while computing vertical axis values.
MOD	I*4	Used in computing linear vertical axis values.
NCHART	I*2	Number of channels read in so far.
NCOMP	I*2	Compression factor.
NDATP	I*2	Number of data points. Always = TOT/NCOMP
NDATPR	I*2	# of channels per record of input file.
NDPOPL	I*2	# of data points on partial lines in LP: section.
NFL	I*2	# of full lines on LP:
P	R*4	Midpoint of each small trapezoid in area

Error Handling

All errors are detected by the program and displayed on the Vector General. The error recovery instructions are also displayed.

GRSPEC

Spectral Histogram of Count vs.
Channel on Vector General Graphics Screen

Operating Instructions

(1) Starting GRSPEC.

(a) Make sure the sign outside the door says UP. If it says DOWN, the computer is not available for use. If it says PRESERVE, the computer will be ready within an hour.

(b) At a free terminal, get MCR by typing C while holding the CNTRL key down. If this does not work, hold CONTROL and press Z, then try CNTRL and C again.

(c) If you plan to use the Vector General Graphics Screen, make sure no one else is using it in the near future. You must sign on to the Decwriter.

(d) Type HEL[uic], where uic is the user identification code given you. For example:

```
MCR>HEL [200,101]
```

To end the line, press the RETURN key. The computer will respond with MCR>.

(e) You are now signed on to the computer. To run a program, type RUN[uic]prog where uic is the user identification code that the program is listed under, and prog is the name of the program. Press the ESC key to end the line. If the uic of the program is the same as the uic you signed on with, it may be omitted. E.g.,

```
MCR>RUN [200,555]SPACER
```

or

```
MCR>RUN SPACER
```

The computer shows the ESC key as a dollar sign.

(f) Type RUN[200,101]GRSPEC. If you are signed on to the given uic, RUN GRSPEC will be sufficient. To end the line, press the ESC key on the left of the keyboard. The computer will respond with: PLEASE BE SEATED AT THE GRAPHICS SCREEN. On the screen is a menu of input data. Function Keys 18 and 21 are lit.

(2) <u>The Menu.</u>

A blinking cursor will appear on the FILENAME line (see Fig. 1). On the screen's keyboard, the FS key moves the cursor forward, BS moves it backward. To change lines, press → key. Note that only capital letters are allowed in the file name (see Fig. 2).

When all of the data values are the way you want them, press Function Key 18 on the function keyboard. Press Key 24 when you are done with the program.

If an error message appears, press Key 15, correct the errant data item and press Key 18 again.

When the computer is done graphing the histogram, function keys 19-23 will light up (see Fig. 3).

(3) <u>Interaction.</u>

You can now do any one of seven things:

(a) Light pen a point on the graph. The result of this action depends on the pen mode. Pen mode can either be channel mode or area mode.

- <u>Channel mode:</u> The channel number and/or energy are output above the channel after an optional peak search.

- <u>Area mode:</u> The point light-penned is the left point of a line sigment. The computer waits for the right point with the function key lights off. When the second point is light penned, the computer draws the line and calculates the area above and below the line. A log scale can make the areas look wrong, but that is a result of the fact that a line is not a line on a log scale. The answer is the same as if it were shown on a linear scale.

(b) Function Key 19 erases channel numbers, energy values and areas from the screen.

(c) Function Key 20 changes the pen mode back and forth between area and channel mode.

(d) Function Key 21 prints the (compressed) data on the line printer. The actual printing occurs when the program terminates (see Fig. 4).

(e) Function Key 22 returns to the menu of Section (2) when done with this histogram.

(f) Function Key 23 erases a summary of these keys from the screen so they don't appear on a hard copy.

SPTI.DAT

PIP

MCR PIP SPTI.DAT;*/LI

RUN GRSPEC

~~Go to Dec Pr~~

for Picture

SPEC BS —— Go to Dec Writer
On

CPY ESC

22
24

RUN[150,107]GRSPEC

1,1
4,0 > 3 files

2,1
3,0 > 1 file

Run [?,20]

SPT2 - 5
SPT3 - 6
SPT4 - 7
SPT5 - 8
SPT6 - 9
SPT7 - 10

Name	file
SPT8.DAT	11
SPT9.DAT	12
SPTØ.DAT	13

(g) Making a Hard copy: Press Function Key 23 to erase the key summary. Make sure you are signed on to the Decwriter. On the Vector General keyboard, hold the SPEC key and press BS to take a picture of the screen. Release. Wait for HARDCP .. STOP to appear on the Decwriter. For further hardcopies, simply press SPEC and BS and wait for HARDCP .. STOP. When you are finished taking pictures, type CPY and press ESC on the Decwriter. When the plotter finishes making the hardcopies, ...CPY—STOP will appear on the Decwriter.

(4) Finished with GEUPST.

When all analysis is finished, press Function Key 22 to return to the menu, then Function Key 24 to exit. Sign off.

```
FILE NAME . . . . . . . . . .  ␣
COMPRESSION FACTOR  . . . .   01
START CHANNEL . . . . . . .   0000
TOTAL CHANNELS  . . . . . .   0000
AXIS MAXIMUM  . . . . . . .   1000000
SCREEN GRAPH  . . . . . . .   Y
LOG SCALE . . . . . . . . .   Y
LP FOR CHANNELS . . . . . .   Y
LP FOR ENERGY . . . . . . .   Y
REFERENCE CHANNEL . . . . .   0001
REFERENCE ENERGY. . . . . .   00001.0000
PEAK SRCH DEVIATION . . . .   0005
TITLE FOR GRAPH . . . . . .    NUCLEAR DATA HISTOGRAM
 SUBTRACTED FILE . . . . . .   N
 SUBTRACTED FILE NAME. . . .


18...DRAW HISTOGRAM     24...END PROGRAM
```

Figure 1

FILE NAME	T01.001	5pTl.DAT
COMPRESSION FACTOR	02	01
START CHANNEL	0000	
TOTAL CHANNELS	0256	1024 – 511
AXIS MAXIMUM	0001000	200
SCREEN GRAPH	Y	
LOG SCALE	Y	N
LP FOR CHANNELS	Y	
LP FOR ENERGY	Y	N
REFERENCE CHANNEL	0001	
REFERENCE ENERGY.	00012.4500	
PEAK SRCH DEVIATION	0005	
TITLE FOR GRAPH	NUCLEAR DATA HISTOGRAM	
SUBTRACTED FILE	N	
SUBTRACTED FILE NAME. . . .		

18...DRAW HISTOGRAM   24...END PROGRAM

⋮

Figure 2

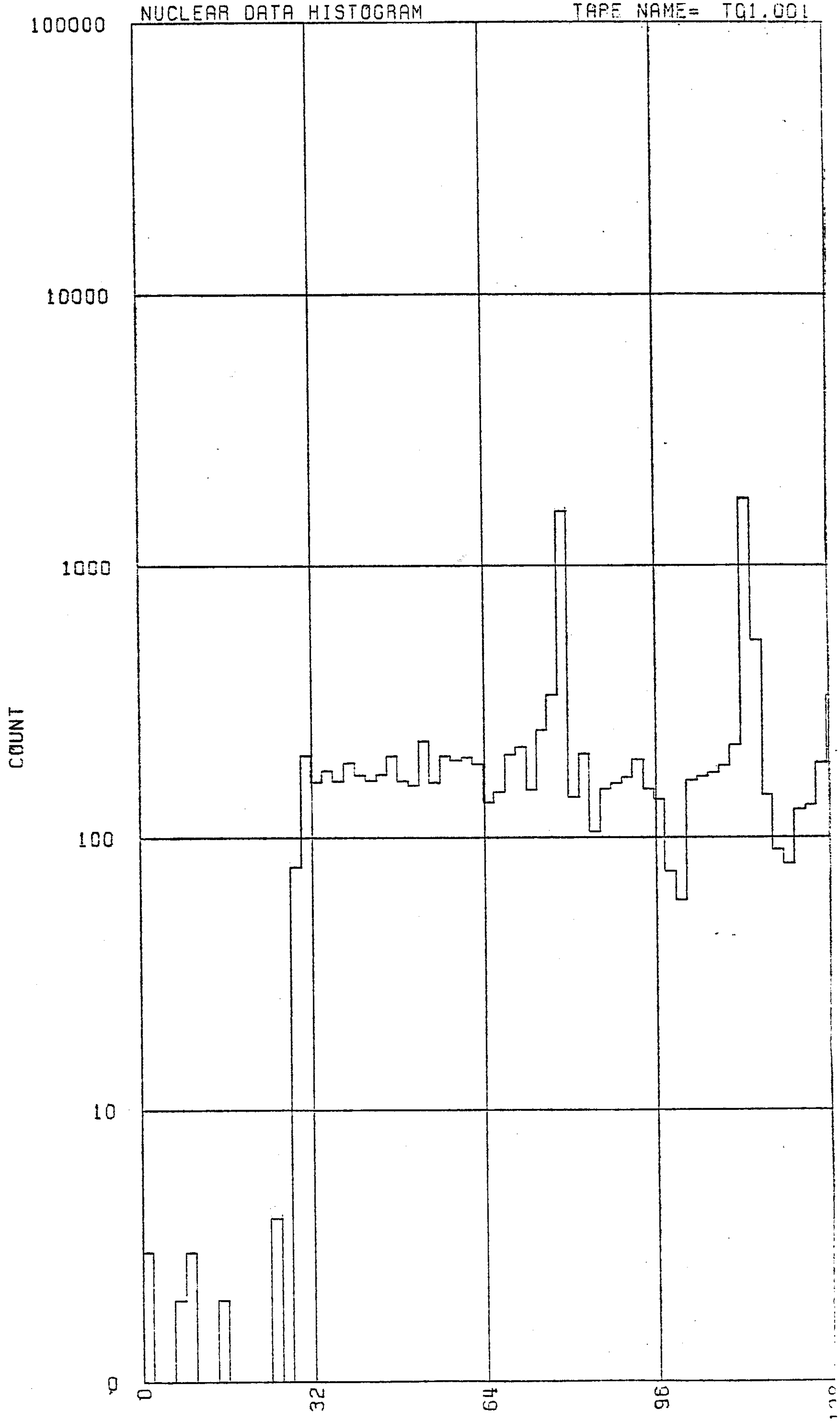

NUCLEAR DATA HISTOGRAM
TAPE NAME= TQ1.001
100000
10000
1000
100
10
0
COUNT
0
32
64
96

Figure 3

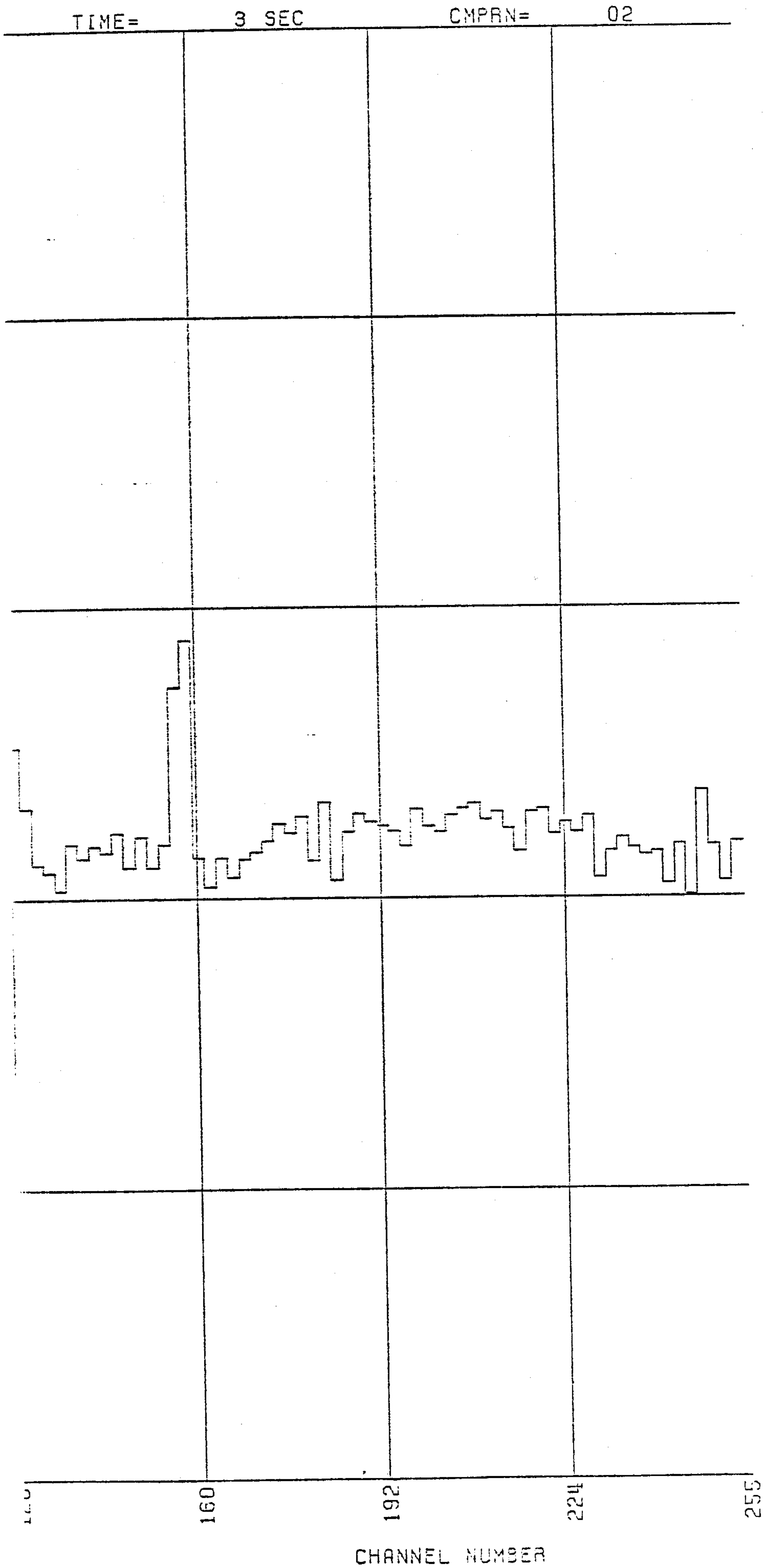


Figure 3.5

CHN #	COUNT	CHN #	COUNT	CHN #	COUNT	CHN #	COUNT	CHN #	COUNT	CHN #	COUNT	CHN #	COUNT	CHN #	COUNT
2 =	1	34 =	176	66 =	147	98 =	75	130 =	206	162 =	110	194 =	170	226 =	167
4 =	1	36 =	161	68 =	202	100 =	59	132 =	131	164 =	138	196 =	151	228 =	191
6 =	2	38 =	188	70 =	215	102 =	162	134 =	124	166 =	118	198 =	203	230 =	117
8 =	3	40 =	170	72 =	150	104 =	167	136 =	107	168 =	136	200 =	176	232 =	145
10 =	1	42 =	162	74 =	248	106 =	173	138 =	154	170 =	145	202 =	169	234 =	160
12 =	1	44 =	171	76 =	336	108 =	184	140 =	138	172 =	157	204 =	192	236 =	148
14 =	2	46 =	200	78 =	1592	110 =	219	142 =	152	174 =	180	206 =	204	238 =	140
16 =	1	48 =	161	80 =	141	112 =	1768	144 =	144	176 =	167	208 =	211	240 =	143
18 =	1	50 =	155	82 =	204	114 =	532	146 =	169	178 =	191	210 =	185	242 =	111
20 =	1	52 =	227	84 =	105	116 =	143	148 =	128	180 =	135	212 =	198	244 =	152
22 =	1	54 =	159	86 =	151	118 =	90	150 =	163	182 =	214	214 =	173	246 =	101
24 =	4	56 =	200	88 =	158	120 =	80	152 =	128	184 =	115	216 =	145	248 =	232
26 =	1	58 =	192	90 =	166	122 =	127	154 =	153	186 =	169	218 =	197	250 =	151
28 =	78	60 =	198	92 =	193	124 =	131	156 =	534	188 =	194	220 =	203	252 =	113
30 =	201	62 =	186	94 =	151	126 =	187	158 =	775	190 =	183	222 =	165	254 =	154
32 =	160	64 =	135	96 =	138	128 =	334	160 =	138	192 =	178	224 =	181	256 =	144

Figure 4

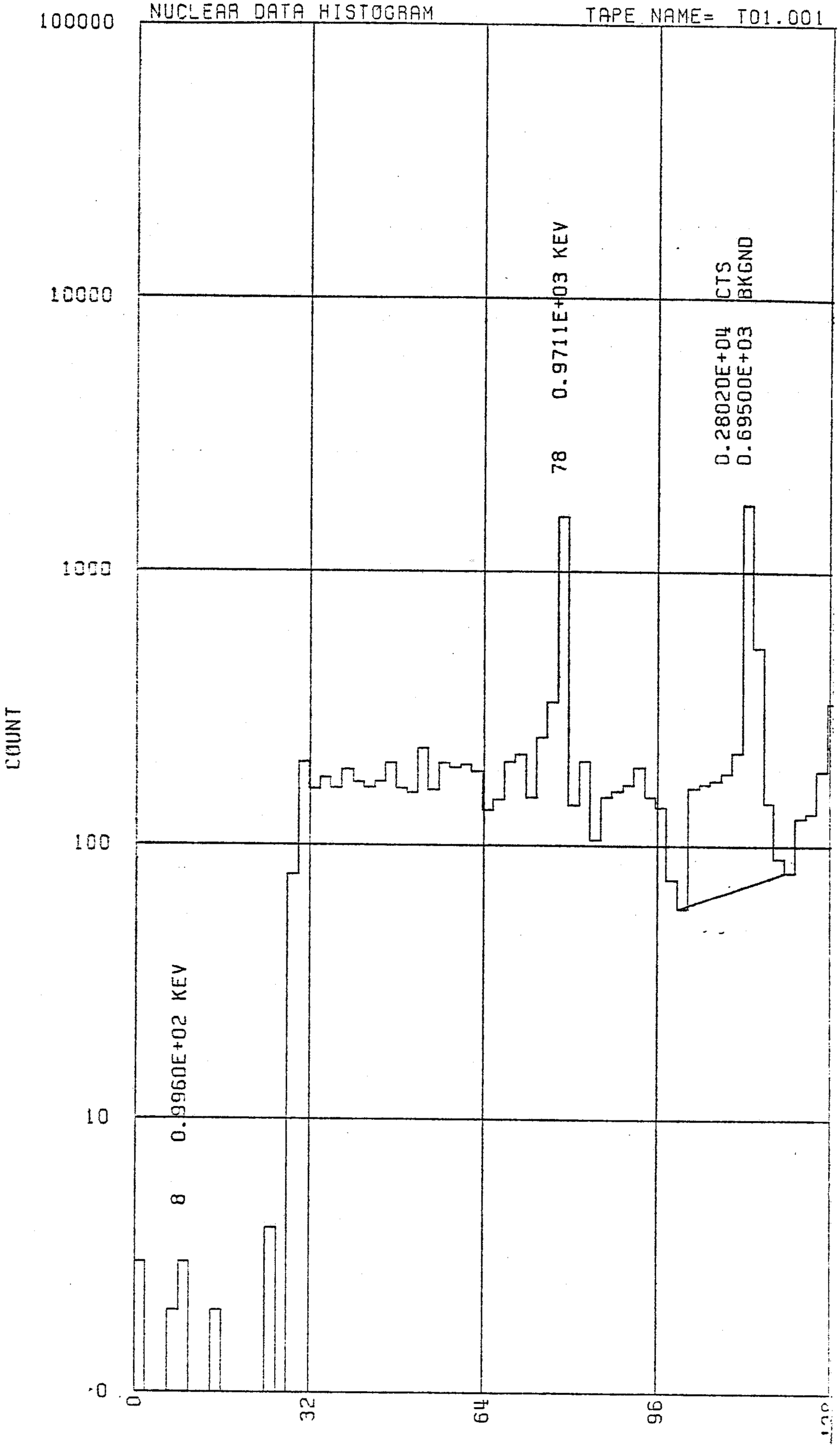
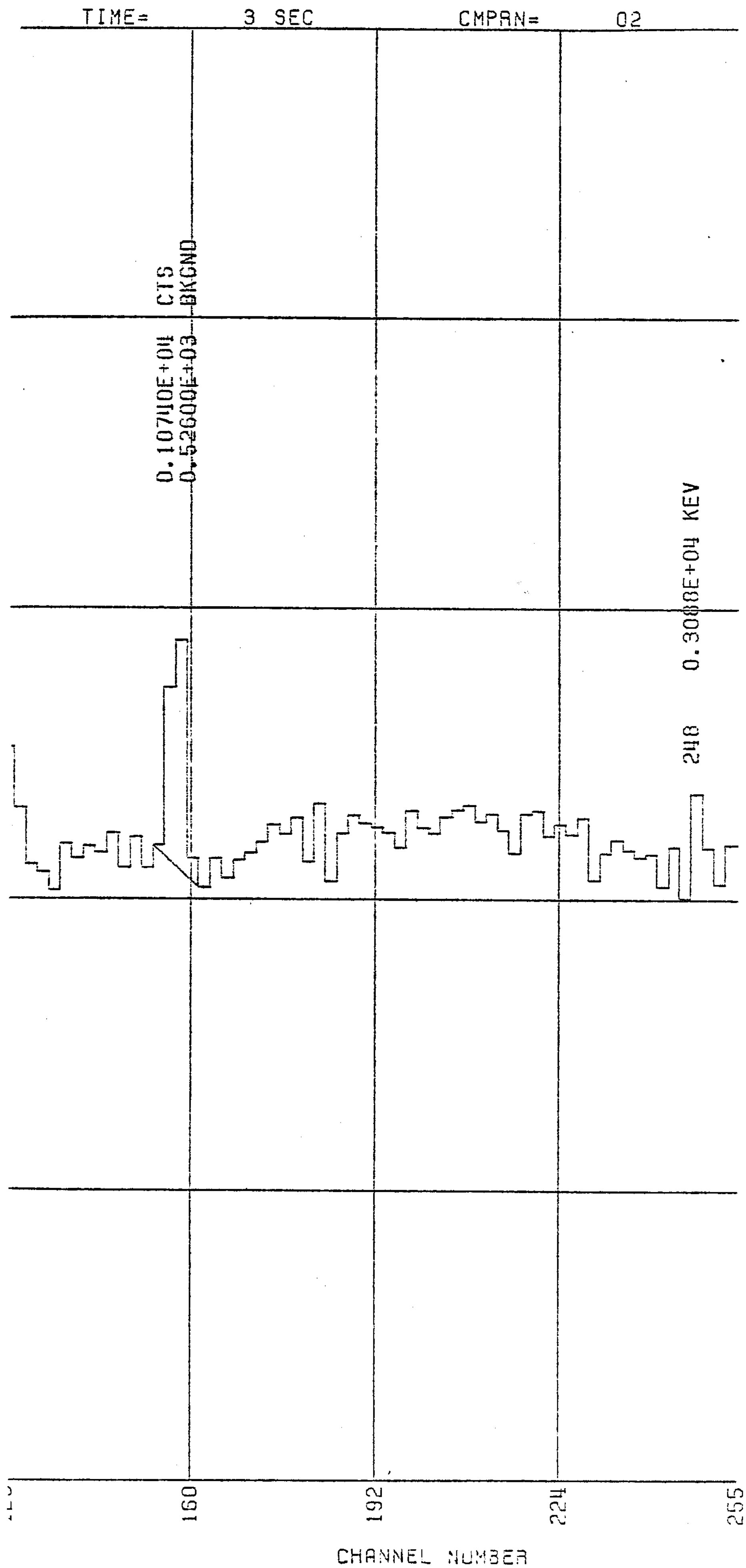

NUCLEAR DATA HISTOGRAM
TAPE NAME= T01.001
100000
10000
1000
100
10
0
COUNT
0
32
64
96
78 0.9711E+03 KEV
0.28020E+04 CTS
0.69500E+03 BKGND
8 0.9960E+02 KEV

Figure 5



TIME= 3 SEC
CMPRN= 02
0.10740E+04 CTS
0.52600E+03 BKGND
248 0.3088E+04 KEV
160
192
224
255
CHANNEL NUMBER

Figure 5.5

IC3DMP

Purpose PRGRMR: Jenny Jacques

Overview.

IC3DMP accepts a 9-track, 1600 BPI, no label GRB tape, MPI tape or TDB tape as input. It has an option to produce a listing of the GRB data on the line printer, with housekeeping parameters separated and converted to proper units. It has also the option to create spectral files for both trigger and background modes, which then are used with GRSPEC to create plots on the Vector General Interactive Graphics Unit. If the tape was TDB or MPI, daily spectra may be collected.

Restrictions.

This program uses PDP-11/70 FORTRAN IV+ and an FTIO package designed for the PDP-11/70 by Code 664. The input tape must be 9-track, 1600 BPI with each file containing 1 header record, followed by any number of data records, all 1536 bytes long (see IC3DMP's maintenance guide for format of these records).

Performance.

Each file takes approximately 2 minutes to process, and 5 to 15 minutes to finish printing--depending on the number of records.

Error Handling.

If there are any input tape errors, a message is issued and that record is skipped; processing continues with the next record. All output conversion errors--i.e., output format field is too small due to bad data--are intercepted by FORTRAN trap routines.

When a calendar in PHA mode (see Structure of a Data Block) cannot be depicted, that data block is zeroed.

All zero data blocks are not printed, nor are spectral data taken from them.

IC3DMP

Overview

IC3DMP interprets an MPI format tape, listing its contents and/or creating spectral files for use on the Vector General graphics terminal.

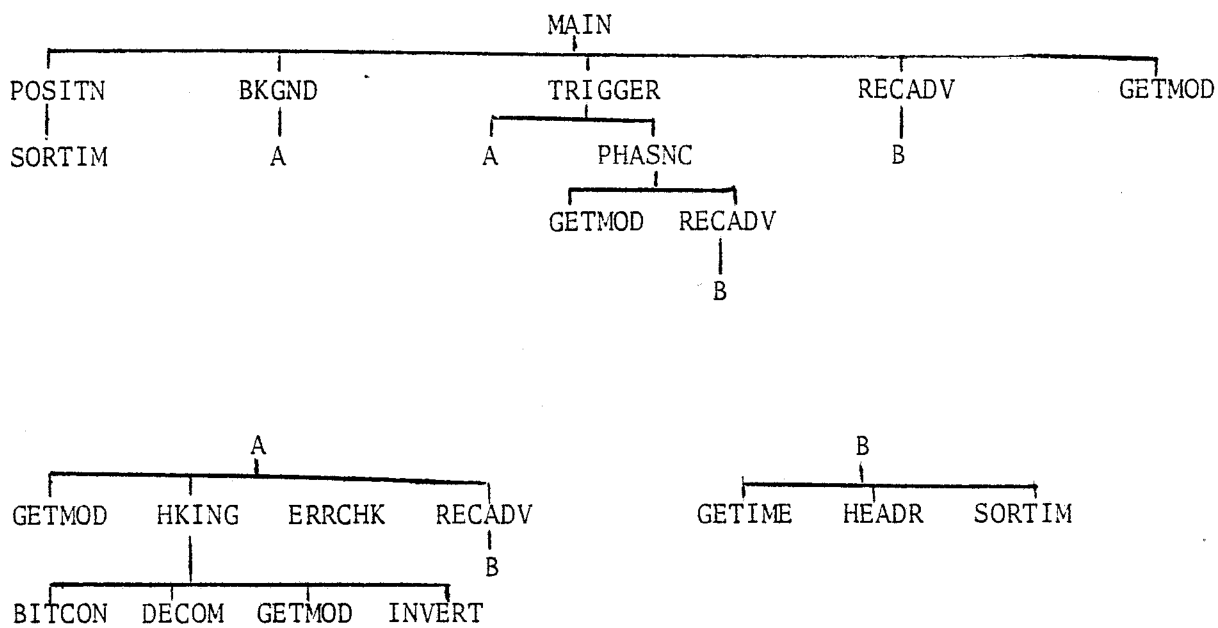
Structure

The following is a list and brief description of the routines used in DRP.

Routine	Function
MAIN	Spaces to desired start position, controls flow of data, and handles spectral file I/O
BKGND	Processes background data
TRGGER	Processes trigger data
PHASNC	Unwraps the trigger PHA data to synchronize with the housekeeping data block
HKING	Separates and lists the housekeeping data
GETHOD	Returns pointer to housekeeping data block 2 or 6 to retrieve the ID bit or mode bits
POSITN	Positions the tape according to user-input start time
SORTIM	Converts year, day, and ms of day bytes to one date which uses Jan. 1, 1978 as time Ø
DECOM	Function subprogram which decompresses the housekeeping background rates.
ERRCHK	Checks the current data block for being all zeros, or repeated housekeeping blocks
BITCON	Converts a 16-bit word into 16 bytes, 1 bit per byte
INVRT	Inverts bitwise one byte
RECADV	Advances the input tape one record, checking for end time and extracting GRB data from header data

Routine	Function
HEADR	Lists the file header
GETIME	Compares current major frame time with start and end times

Block Diagram



Common Areas

The common areas are first defined and then their usage in IC3DMP is outlined.

Common Names

1. /BUFF/QBUFF, MODE, IPLACE, LIST

Variable	Type	Initial Value	Meaning
QBUFF(1536)	L*1	Ø	GRB data, 64 major frames
MODE	I*2	-	Ø=background, 1=PHA trigger, 2=time history 1, 3=time history 2

Variable	Type	Initial Value	Meaning
1PLACE	I*2	Ø	Pointer to QBUFF
LIST	L*1	N	Y=list data, N=no list

2. /PHACAL/FP, ENDCAL, 1FLG

Variable	Type	Initial Value	Meaning
FP	I*2	1	The housekeeping FP value (see General Maintenance Guide)
ENDCAL	I*2	Ø	Pointer to last PHA trigger data block processed
IFLG	I*2	Ø	Ø=background or time history 1=PHA trigger

3. /TAPE/NF, LUN, IEND, EFILE, EREC, IREC

Variable	Type	Initial Value	Meaning
NF	I*2	1	Current file number
LUN	I*2	-	Logical unit number for the tape drive, set by FTIØ
IEND	I*2	Ø	Ø=more data, 1=end of data
EFILE	I*2	999	End file number
EREC	I*2	999	End record # of end file
IREC	I*2	-	Current record number

4. /TIMET/BYEART, BDAYT, BMST, EYEART, EDAYT, EMST, TFLAGT

Variable	Type	Meaning
BYEART	I*2	Beginning year, day, and milliseconds of day of trigger data processed
BDAYT	I*2	
BHST	I*4	

Variable	Type	Meaning
EYEART	I*2	End year, day, and milliseconds of day of trigger data processed
EDAYT	I*2	
EMST	I*4	
TFLAG	I*2	Ø=no trigger data so far 1=some trigger data processed

5. /TIMEB/BYEARB, BDAYB, BMSB, EYEARB, EDAYB, EMSB, TFLAGB

Variable	Type	Meaning
BYEARB	I*2	Beginning year, day, and milliseconds of day of background data processed
BDAYB	I*2	
BMSB	I*4	
EYEARB	I*2	End year, day, and milliseconds of day of background data processed
EDAYB	I*2	
EMSB	I*4	
TFLAGB	I*2	Ø=no background data so far 1=some background data processed

6. /CTIMES/CYERA, CDAY, CMSDAY, BTIME, ETIME

Variable	Type	Meaning
CYEAR(64)	I*2	The year, day, andmilliseconds of day for each major frame in QBUFF
CDAY(64)	I*2	
CMSDAY(64)	I*4	
BTIME	R*8	Beginning time to process, from user. Converted via TIMCON (described later)
ETIME	R*8	Ending time to process, from user. Converted via TIMCON (descrived later)

7. /SPCTRA/BKTR, IND1, IND2

Variable	Type	Meaning
BKTR(4096)	I*4	Holds spectral data. Background and trigger spectra are swapped in and out of this array from disk, as each mode is processed
IND1	L*1	'*'=don't collect spectra for background, anything else=collect spectra for background
IND2	L*1	'*'=don't collect trigger spectra, anything else=collect trigger spectra

8. /HEAD/INBUFF

Variable	Type	Meaning
INBUFF(5128)	L*1	Holds input MPI format tape record

Occurrence and Use of Commons

Under the USE heading is R, W, or RW for reads only, writes only, or reads and writes into the common. If -, then the common is not used.

Common Name	Occurrence	Use
BUFF	MAIN	RW
	HKING	R
	GETMOD	R
	BKGND	RW
	TRGGER	RW
	PHASNC	RW
	ERRCHK	R
	RECADV	RW
PHACAL	MAIN	-
	BKGND	W
	TRGGER	RW
	PHASNC	RW
TAPE	MAIN	RW
	POSITN	RW
	BKGND	R
	TRGGER	R
	PHASNC	R
	RECADV	RW

Common Name	Occurence	Use
TIMET	MAIN	RW
	PHASNC	RW
TIMEB	MAIN	RW
	BKGND	RW
CTIMES	MAIN	RW
	POSITN	R
	BKGND	R
	PHASNC	R
	RECADV	W
	GETIME	W
SPCTRA	MAIN	RW
	BKGND	RW
	TRGGER	RW
HEAD	MAIN	-
	POSITN	R
	RECADV	R
	HEADR	R

Other Variables

Each routine is listed with its main variables defined. Loop increments, common variables, and the ISTR2-L1 technique are excluded. The ISTR2-L1 technique is used to create an integer number from a logical byte variable. L1(logical*1) is equivalenced with ISTR2(integer*2), and when L1 is assigned to a logical byte variable, ISTR2 becomes an integer value equal to the locigal byte value. (Note that bytes are swapped for PDP 11/70 integers.)

1. MAIN

Variable	Type	Meaning
AGAIN	L*1	Y=do another spectral interval N=end program
BKEEP	R*8	Holds beginning time from previous spectral interval
FILKEP	I*2	Holds beginning file number of previous spectral interval
IBKTR	I*2	Ø=last mode processed was background 1=last mode was trigger

Variable	Type	Meaning
1DEN	I*2	Density of MPI tape
1POSN	I*2	1=position tape by file & record # 2=position tape by GMT times
LABEL	I*2	Equals Ø for non-label MPI tape
MODE2	I*2	Mode returned from GETMOD, same as MODE in common
QEND	L*1	T=end of tape or beginning time not found by POSITN F=tape positioned OD
SYR	I*2	User-input beginning year, day, and ms of day of data to process
SDY	I*2	
SMSOD	I*4	
EYR	I*2	User-input ending year, day, and ms of day of data to process
EDY	I*2	
EMSOD	I*4	
BTITLE(50)	L*1	Title for background spectral file
TTITLE(50)	L*1	Title for trigger spectral file
SFILE	I*2	Start file number
SREC	I*2	Start record number
IVSN(6)	L*1	Tape name for FTIØ
NAME(10)	L*1	Spectral file name

2. POSITN

Variable	Type	Meaning
CTIME	R*8	Current record time
LOST	I*2	Return code from FT10 1-successful -10=end of file mark

Variable	Type	Meaning
LENGTH	I*2	Length in bytes to read from record
LR	I*2	Length in bytes read from record

3. SORTIM

Variable	Type	Meaning
YEAR,YR(2)	I*2	Retrieves year, day, and milliseconds of day from byte data
DAY,DY(2)	I*2	
MILLI,MS(4)	I*4	

4. HKING

Variable	Type	Meaning
HK	I*2	Housekeeping number=Ø-7 for ID=Ø, 8-15 for ID=1
HTK		*Housekeeping number from data, bits 0-3
IC	I*4	Decompressed rate
ID	I*2	*The ID parameter
IPNT	I*2	Pointer to HTK=2 for ID bit
ITEMP	I*2	Used to invert
IV1	I*2	*Threshold housekeeping bits 5, 6, and 7
IV2	I*2	*Housekeeping bits 8 and 9
IV3	I*2	*Housekeeping bits 10, 11, and 12
IV4	I*2	*Housekeeping bits 13, 14, and 15
IV5,BUF(2)	I*2	*Housekeeping bits 4-15
N1	L*1	Used to invert threshold

*See General Maintenance Guide

Variable	Type	Meaning
HKG(16)	L*1	Contains the 16 housekeeping bits, packed 1 per byte
HZ(8)	I*2	*The frequencies used to decode FP

5. DECOM

Variable	Type	Meaning
IC	I*4	The decompressed rate
ICX	I*2	Bits 9-15 of IV5
ICY	I*2	Bits 4-7 of IV5
IV5	I*2	The rate before decompression

6. GETMOD

Variable	Type	Meaning
HKNUM	I*2	Housekeeping number of data block being checked
ICNT	I*2	Counts how many data blocks have been checked
POINTR	I*2	Pointer to data block being checked

7. BKGND

Variable	Type	Meaning
BKGND	R*8	Difference in DT units between current and previous data block
CALNDR	R*8	Current calendar value
CAL4DF	R*8	Difference in DT units between current and previous calendars which occurred on HKØ or HK4

*See General Maintenance Guide

Variable	Type	Meaning
CHECK	L*1	=blank if the sum of the DT's and CALDIF agree; '*' if they don't agree
EMPTY	L*1	T=Skip data block F=use this data block
FIRST	I*2	Ø=first background data block being processed; 1=not first data block
HK	I*2	Housekeeping number
INC	I*2	Data block number in buffer
ISUM	I*4	Sum of DT values
MODE2	I*2	Holds mode returned from GETMOD. Same as MODE in common
OCAL4	R*8	Previous calendar value which was HKØ or HK4
OCTCAL,OCTEMP(4)	L*4	Octal representation of calendar
OLDCAL	R*8	Previous calendar value
OLDOCT	L*4	Octal representation of previous calendar
DT(3,3)	L*1	Holds DT values
PHA(3,3)	I*2	Holds PHA values
TMP(18)	L*1	HOlds DT-PHA bytes

8. TRGGER

Note: Many of the variables are the same as in BKGND and are not repeated here.

Variable	Type	Meaning
J1,J2,J3	I*2	Integer representation of each byte of 3-byte time history calendar
MEMAD	I*2	Time history memory address
N1,N11	L*1	First time history calendar byte, and a dummy variable to keep it on even words

Variable	Type	Meaning
N2,N22	L*1	Same as above, only the second byte
N3	L*1	Third calendar byte
OLDPLC	I*2	Time history DT values

9. PHASNC

Note: See BKGND for definition of some variables.

Variable	Type	Meaning
CALKEP	I*2	Holds the first two bytes of the calendar as an integer
CALPUT	I*2	Pointer to the start of the calendar within the data block
COUNTR	I*2	Counts matches of calendar bytes to require 4 before the calendar is considered found
DIFF	I*2	The difference between the previous calendar's first two bytes and the 2 bytes which are candidate for the current calendar
NEWPLC	I*2	Pointer to the new buffer of synchronized data blocks
CAL(2)	L*1	The first two bytes of the suspected calendar word

10. RECADV

Variable	Type	Meaning
BUFPLC	I*2	Pointer to each major frame in INBUFF
CTIME	R*8	Time of major frame being checked
ENDFL1	L*1	T=end of data reached; F=data continues
ENFFL2	L*1	T=last call to RECADV set endfl1 to T, so end program; F=continue program

Variable	Type	Meaning
EOV	L*2	Ø=no end of file read; 1=end of file read
FIRST	L*1	T=first time RECADV is called; F=subsequent calls
FOUND	L*1	T=data to process was found in the record; F=no data found to process
INC	I*2	Increment for major frame number being processed
LOST	I*2	Status of read from FTIO
LENGTH	I*2	Length in bytes to read from tape
LR	I*2	Length in bytes actually read from tape
QPLC	I*2	Pointer to QBUFF array, incremented for each data block
STAT	I*2	Ø=use this data block; 1=end of data to process; 2=skip this data block

11. HEADR

Variable	Type	Meaning
BITRAT	I*4	Bit rate
YEAR	I*2	The year, day, and milliseconds of day of the header file
DAY	I*2	
MSDAY	I*4	
SCLOCK	R*8	Spacecraft clock

12. GETIME

Variable	Type	Meaning
CYR	I*2	The year, day, and milliseconds
CDY	I*2	
CMSC	I*4	

Variable	Type	Meaning
CTIME	R*8	The above times converted to one number

Algorithm of Data Flow

The following is an algorithm tracing data flow through IC3DMP. The data is read from an MPI format tape which contains header data and experiment data, with 32 data blocks per record read. Two records are read in the first time for double buffering, the second record's GRB data moved into the first half of the array and a new record read into the second half on subsequent reads.

Routine	Step	
MAIN	1	User options read in
POSITN	2	Positon tape to proper place
RECADV	3	Read in the record of data
GETIME	4	Times for each data block is stored
MAIN	5	Mode of first data block is found with GETMOD. If the mode is backgound Steps 6-10 are executed. If the mode is trigger Steps 11 are executed.
BKGND	6	If a new record is needed, RECADV is called
	7	Difference in calendars and PHA-DT words are found
	8	Housekeeping is listed by calling HKING
	9	If desired, PHA-DT words and calendar is listed
	10	If desired, spectra are collected
TRGGER	11	If mode is time history go to Step 20
	12	Synchronize data by calling PHASNC
PHASNC	13	Find the calendar
	14	Write synchronized data block over previously processed data
	15	If not PHA trigger, return
TRGGER	16	Find difference in calendars, and get PHA-DT words

Routine	Step	Action
	17	List data, HKING does the housekeeping
	18	If desired, collect spectra
	19	Go to Step 11
	20	Invert the calendar bytes to get calendar
	21	Get difference of calendars, and DT words
	22	List data, HKINC lists housekeeping part
	23	If not done with time history, go to Step 16
MAIN	24	Data is done, close spectral files and check to see if another interval is desired
	25	End if done, otherwise go to Step 1

Additional Notes

1. Because of space limits on the PDP 11/70, one array BKTR must be used for both background and trigger spectra. So, when one mode is done, the file is written to disk, and the new mode's file is read into the array.

2. In order to compare the sum of the DT's and the difference of the calendars, the listing of the data blocks are delayed until the processing of the next one has advanced to subtracting the calendars.

IC3DMP

OPERATOR'S GUIDE

Overview.

IC3DMP is used to create listings and spectral files of the Gamma-ray Burst (GRB) experiment on ISEE-3. It has the following options.

(1) Collect data by file and record specifications.

(2) Collect data by GMT times corresponding to dump time.

(3) List the data.

(4) Collect background and/or trigger spectra and store on user-specified disk file.

Input Required.

(1) User interaction to specify options.

(2) A 9-track, 1600 BPI tape in MPI format.

Running IC3DMP.

The following flow chart shows the options and how they affect the program. Input to at least one branch is required to complete the analysis. After the chart and corresponding explanation is a sample run. Each mode in the chart is a program/user interaction.Type ' RUN IC3DMP ' to initiate.

Options Description.

Letters correspond to Options Flow Chart (Page 2).

(A) ENTER 1 TO POSITION TAPE BY FILE, RECORD
ENTER 2 TO POSITION TAPE BY GMT TIMES

Entering 1 causes the tape to use file and record specifications in locating data to analyze. Entering a 2 will use GMT times. Generally, quick-look processing uses 1, and production uses 2.

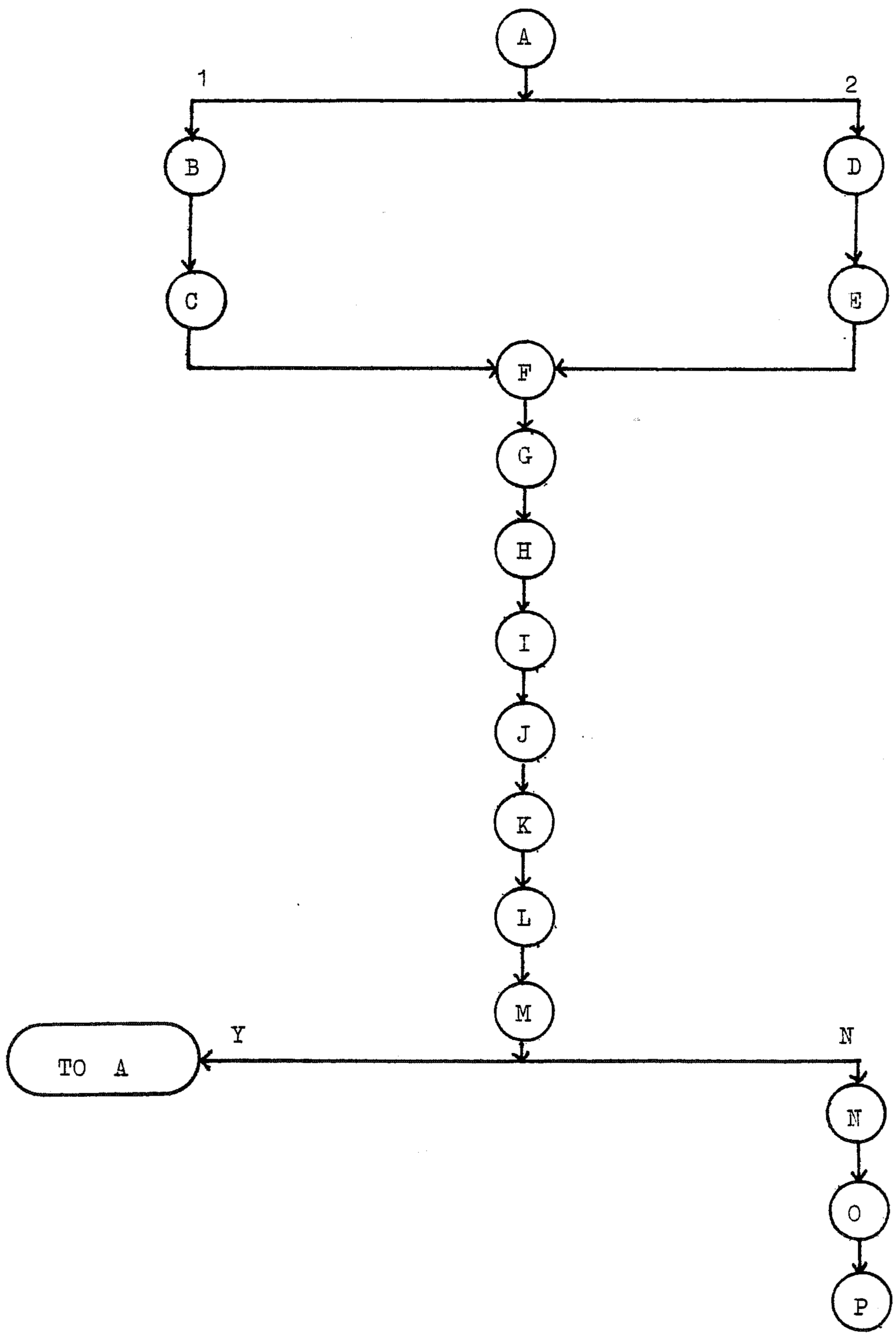
(B) ENTER START TILE, START RECORD

Enter file and record numbers at which processing is to begin.

(C) ENTER END FILE, END RECORD

Enter file and record numbers after which processing is to terminate.

OPTIONS FLOW CHART



(D) ENTER START YEAR, DAY, MILLISECONDS OF DAY

This is the time at which processing is to begin.

(E) ENTER END YEAR, DAY, MILLISECONDS OF DAY

This is time at which the program terminates

(F) IS A DATA LISTING DESIRED? (Y OR N)

Y means to list the data, N means to suppress the listing.

(G) INPUT THE NAME OF THE BACKGROUND SPECTRA FILE TO CREATE, OR TYPE * IF NO FILE IS TO BE MADE

This is the background disk file, name entered as A.B, where A and B are alphanumeric character strings, A being up to 9 characters, B being up to 3 characters. If an * is entered, no file is created.

(H) ENTER A TITLE FOR THIS FILE

Enter up to 70 characters, or press CR for a blank title.

(I) INPUT THE NAME OF THE TRIGGER SPECTRA FILE TO CREATE, OR TYPE * IF NO FILE IS TO BE MADE.

Same as (G), only this file will hold trigger PHA data.

(J) ENTER A TITLE FOR THIS FILE

Same as (H).

(K) The computer issues a mount-the-tape request.

(L) Press CNTL C to get MCR> and enter the following line:

MOU MMχ:/CHA=(FOR)

where χ is Ø or 1, depending on the drive upon which you have mounted the tape.

(M) END OF DATA PROCESSING
DO ANOTHER SPECTRAL INTERVAL? (Y OR N)

Enter Y to process more data, N to stop program.

(N) The computer issues a dismount-the-tape request.

(O) Press CNTL C to get MCR> and enter the following line:

DMO MMX:

where X is the same as in L.

The computer then issues a dismount completed message.

(P) Enter the following line:

CON IC3DMP

The program will stop.

Example.

The following example uses the branch A-D-E-F-G-H-I-J-K-L-M-N-O-P:

```
MCR>HEL [100,105]/NM
MCR>RUN IC3DMP$
 ENTER 1 TO POSITION TAPE BY FILE,RECORD
 ENTER 2 TO POSITION TAPE BY GMT TIMES
2
 ENTER START YEAR,DAY,MILLISECONDS OF DAY
78,268,0
 ENTER END YEAR,DAY,MILLISECONDS OF DAY
78,269,0
 IS A DATA LISTING DESIRED?(Y OR N)
N
INPUT THE NAME OF THE BACKGROUND SPECTRA FILE TO
CREATE, OR TYPE * IF NO FILE IS TO BE MADE.
FILE.BKG
ENTER A TITLE FOR THIS FILE
THIS IS A SAMPLE TITLE
INPUT THE NAME OF THE TRIGGER SPECTRA FILE TO
CREATE, OR TYPE * IF NO FILE IS TO BE MADE.
FILE.TRG
ENTER A TITLE FOR THIS FILE
THIS IS ANOTHER SAMPLE TITLE
**** IC3DMP - SUSPENDED
MOUNT IVSN   ON DRIVE 0 , ISSUE MOU COMMAND

MCR>MOU MMO:/CHA=[FOR]
MOUNT-**VOLUME INFORMATION**
        DEVICE  =MMO
        CLASS   =FOREIGN
        UIC     =[1,1]
        ACCESS  =[RWED,RWED,RWED,RWED]
        CHARAC  =[FOR,DCF]
MCR>CON IC3DMP$
 END OF DATA PROCESSING
 DO ANOTHER SPECTRAL INTERVAL?(Y OR N)
N
**** IC3DMP - SUSPENDED
REMOVE IVSN   FROM DRIVE 0 , ISSUE DMO COMMAND

MCR>DMO MMO:
DMO -- MMO: ** DISMOUNT COMPLETE **
MCR>CON IC3DMP$
IC3TST  --  STOP
```

Error Handling.

System and user errors may occur during the execution of IC3DMP. The fatal (non-recoverable) errors are listed in order of occurrance corresponding to the Options Flow Chart. All non-fatal errors which may happen due to bad user input are trapped by the program which requests correction before continuing.

Fatal Errors - require program to be re-run.

(1) Between (J) and (K), FTIO may abort as indicated by a message issued by the computer. This means both tape drives are in use. If a drive is available, the previous user failed to issue a dismount command. Thus, issue the DMO command for the proper drive (see O) and then type the following:

MCR>RUN [2,76] CLEF

This clears the flags. Now IC3DMP may be run again.

(2) Between (L) and (M) the message is issued:

ERROR--START RECORD NOT FOUND IN START FILE

The program continues with Step (N). Check to make sure the start file and record numbers are correct and existent.

(3) Between (L) and (M) the message is issued:

ERROR--BEGINNING TIME NOT FOUND ON TAPE VOLUME

The program continues with Step (N). Make sure the times are correct and existent.

LTHIST

Purpose: Purpose PRGRMR: Mary Esfandiari

The purpose of the LTHIST (Long Term Time History) program is to read the MPI format tape and extract, for a specified period of time, the background Rates, Temperatures, and/or Voltages. The extracted data can then be plotted as single values with an associated time or as averaged values over the midtime of the averaging interval. One has the option of choosing a log-linear or linear-linear plot for the rates. An optional listing is also created for all data points plotted.

Input:

Input to the LTHIST program is by Namelist.

Namelist/Input/ Variable Name	Meaning
IRATES	which rates to plot
ITEMPS	which temperatures to plot
AVGINT	averaging interval (in terms of hour(s))
ILIST	set if listing is desired
YMAXT	maximum terperature scale on plots
YMINT	minimum temperature scale on plots
YMAXR	maximum rate scale on plots
YMINR	minimum rate scale on plots
IBFIL	beginning file
IBREC	beginning record
IENDFL	ending file
IENDRC	ending record
ILOG	set if log-linear plots desired
IERBAR	set if error bars on rate plots are desired
IDV	set if voltage plot desired
ISUB	set if GEL-GEH is desired
QIVSN	input tape no. enclosed in quotes

LTHIST

Internal Documentation

Purpose:

The purpose of the LTHIST (Long Term Time History) program is to read the MPI format tape and extract, for a specified period of time, the background Rates (1-4), Temperatures (1-3), and/or Voltages. The extracted data can then be plotted as single values with an associated time or as averaged values over the midtime of the averaging interval. One has the option of choosing a log-linear or linear-linear plot for the rates. An optional listing is also created for all data points plotted.

Resources Required:

IBM 360/91 or 75
Wolf Plot Package
Fortran 'H' Compiler, Parameter 'XL'
7-Track Plot Tape
FT10 Package

Assumptions:

1. The data analysis period is within 1 magnetic tape volume.
2. The tape is in MPI format. (see General Users Guide)
3. TIME values of zero or those that do not conform to acceptable data quality are rejected as bad data.

Mathematical Formulations:

- Bit manipulation strings are used such as TBIT, SHFTR and LAND (logical and)
- Error bars are calculated by: $\sqrt{\frac{rate}{time}}$
- Rates are decompressed in the following manner:
  The rates are stored in housekeeping words 3 and 7.

bit positions of Housekeeping Word

IHK→	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(bits 3–15: Compressed Rate IS2)

extract compressed rate: IS2=LAND(IHK,4095)
ICX(bits 7-15)=LAND(IS2,127)
ICY(bits 4-8)=SHFTR(IS2,7)
NI=ICY-29
If ICY≤24 NI-ICY+3
If 0>ICY>31 ICY=25

decompressed rate=(1.+(FLOAT(ICX)+128.5)*2.**(FLOAT(NI-8)))/1024
If 25≤ICY≤28 Rate=-1
If ICY=24and ICX=127 Rate=0

Error Detection/Correction:

Numbers within the range of the Namelist parameters are required for the program to continue executing.

All FTIO FREADS are checked for status to determine if an end-file or an error in read has occurred. An error in read, generates a message and execution continues with the next record. Premature end-of-files (e.g. IENDFL to big) will generate a message that a premature end-of-file has occurred and execution will terminate.

A time of Ø from the MPI tape indicates bad data. A message is issued and execution continues with the next record.

Performance:

4.5 FILES take approxmiately .50 min CPU and 1.3 min I/O time to produce a listing with line printer plots of all selected rates.

Common Block Information:

The common block Values is used in passing information back to the Main program from the Subroutine STORE. Basically, it contains all of the arrays needed to store the rates and their errors.

Common Values

Variable/Array	Type	Function
TIMRT1	Real*4	time array for Rate 1'
TIMRT2	"	" " " Rate 2
TIMRT3	Real*4	" " " Rate 3
TIMRT4	"	" " " Rate 4
Rate1	Real*4	rate 1 array
Rate2	"	rate 2 "
Rate3	"	rate 3 "
Rate4	"	rate 4 "
ERATE1	Real*4	error array for Rate 1
ERATE2	"	" " " Rate 2
ERATE3	"	" " " Rate 3
ERATE4	"	" " " Rate 4
IHK	Integer*4	
I1-4	Integer*2	array counter - Rate 1-4
II1-4		array counter-time - Rate 1-4
M1-4	Integer*2	array counter-error Rate 1-4

The common block PLOTS is used in passing relevant information to the plotting routine (PLOTTR) so that plots can be made. It contains information such as the axis values on the rates, temperatures, & voltage plots.

Common Plots

Variable/Array	Type	Function
PLTEND	Real*4	end of a plot frame
TIMIN	"	minimum time (start time)
TIMAX	"	maximum time to plot
YMAXR	"	maximum Rate
YMINR	"	minimum Rate
YMINT	"	minimum Temperature
YMAXT	"	maximum Temperature
ILOG	Integer*4	log of linear plots
J5	Integer*4	# values of Rate 1
J7	"	" " " " 2
J9	"	" " " " 3
J11	"	" " " " 4
J1	Integer*4	# values of Temperature 1
J2	"	" " " " 2
J3	"	" " " " 3
J4	"	" " " time array for Temperatures
J6	Integer*2	" " " time arrav Rate 1
J8	"	" " " " " " 2
J10	"	" " " " " " 3
J12	"	" " " " " " 4
IRF1	Integer*2	select Rate 1 when set
IRF2	"	" " 2 " "
IRF3	"	" " 3 " "
IRF4	"	" " 4 " "
IPFLG	I*2	determines whether plot arrays are to be closed out
IDTEMP	"	which temperatures to store
IDRATE	"	which Rates to store
IDV	"	voltages

The common block ACCUM is used to pass information to the ACCUMS subroutine when averaging is needed.

Common ACCUM

Variable/Array	Type	Function
MS	Real*4	milliseconds of day
MSHR	Real*4	number of milliseconds in an hour
AVGINT	Real*4	indicates averaging interval (# hours)
IDAY	Integer*2	current day
IFLAG	Integer*2	indicates which rate to store (1-4)
IYR	I*2	current year
BTIME (seconds since epic year)	Real*8	beginning of averaging interval
ENTIME (seconds since epic year)	Real*8	end of averaging interval
ICX	Integer*4	used in decompressing rates
ICY	Integer*4	used in decompressing rates
NI	Integer*4	used in decompressing rates
ICNT1-4	I*4	records total # of rates/averaging interval

This common block is used in passing the extracted Temperatures from the EXTEMP subroutine to the PLOTTR subroutine for plotting.

Common Block TEMP

Variable/Array	Type	Function
TEMP1	Real*4 array	temperature 1 array
TEMP2	" "	" 2
TEMP3	" "	" 3
TIMTMP	" "	time array for temperatures
ITF1	I*2	when 1-select Temperature 1
ITF2	"	" " " " 2
ITF3	"	" " " " 3

Variable/Array	Type	Function
QREC	L*1 array	current record
IDBIT	L*1 array	ID Bit Values
IBFPTR	I*2	pointer into buffer QREC

The common block VOLTS is used to pass the voltage array and Gel-Geh array to the PLOTTR routine.

Common Block VOLTS

Variable/Array	Type	Function
VOL(240)	Real*4	voltage array
VOLTIM(240)	Real*4	time array for volts
SUB(240)	Real*4	Gel-Geh array
J13	I*4	# values in voltage array
J14	I*4	# values in Gel-Geh array
ISUB	I*2	flag indicates whether Gel-Geh is desired

Input Requirements:

Input is by the FORTRAN Namelist option.

Parameter	Type	Range	Default	Meaning of Values
IRATES	I*2	111≥IRATES≥Ø	0	1000 select Rate 1 0100 " " 2 0010 " " 3 0001 " " 4 0101 " Rates 2&4 1111 all select Rates
ITEMPS	I*2	111≥ITEMPS≥0	0	100 select Temperature 1 010 " " 2 001 " " 3 101 " Temperatures 1&3 111 select all Temperatures
AVGINT		cannot be >1.0 if Temperatures or Voltages are selected	0.0	in terms of hours
ILIST	I*2	32768≥ILIST≥0	0	0 no listing created >0 create listing
YMAXT	R*4	no restrictions	1000.0	for plotting; maximum Value of the Y axis (Temperatures)
YMINT	R*4	no restrictions	0.0	for plotting; minimum Value of the Y axis (Temperatures)
IERBAR	I*4	0, >0	0	if 0 specifies error bars for rate plots

Parameter	Type	Range	Default	Meaning of Values
YMAXR	R*4	no restrictions	150000.0	for plotting; maximum value of the Y-axis (Rates)
YMINR	R*4	" "	0.0	for plotting; minimum value of the Y-axis (Rates)
QREC	L*1	tape no.		
IBFIL	I*4	>0	0	start file
IBREC	I*2	>0	0	start record
IENDFL	I*4	>0	0	end file
IENDRC	I*2	>0	0	end record
ILOG	I*4	0,1	0	0-linear plots 1-log plots
IDV	I*2	0,>0	0	0-no voltages >0-voltages
ISUB	I*2	0, >0	0	0-Gel-Geh not required >0-request Gel-Geh
QIVSN	L*1	none	none	input tape no. enclosed in quotes

Output Generated:

The maximum output generated is a listing of plotted data points and 3 plots: Temperatures vs. Time, Rates vs. Time and Voltages vs. Time.

A plot tape is also generated which will produce the plots on the Calcomp plotter, as mentioned above.

Algorithms Used Throughout Main Program

IRATES and ITEMP are obtained through Namelist as 4 digit integers. Each digit, starting from the left, indicates whether rate 1 through 4 or temperatures 1 through 3 are to be selected. For example, IRATES=1101 would indicate that rates 1, 2 and 4 are to be selected.

In order to easily determine which rates are desired without testing each digit everytime a decision needs to be made, the IRATES and ITEMPS parameters are separated into flags IR1, IRF2, IRF3, IRF4 and ITF1, ITF2 and ITF3, respectively. When a flag is set (=1), that temperature or rate is desired.

So that scaling parameters for the plots can be setup, a maximum time or end time must be known. This is obtained by positioning to the last file requested and then calling successive FREADS to position to the last record desired. The time is then extracted from the last record and placed in the variable TIMAX. The tape is then positioned back to the beginning file and the first time extracted becomes TIMIN.

Logical recors are 2564 bytes and physical records are 5128 bytes. The first logical record of each file is a header reocrd and is skipped. Each FREAD reads a full physical record. IBFIL and IBREC are the first file and the first record to begin processing, respectively. If IBREC is 1, then the pointer into the FREAD butter (IBFPTR) is set to 2564 to skip the header.

If IBREC is greater than 1, then successive FREADS are performed to position to the beginning record where processing starts.

Additionally, the first 4 bytes of each logical record identify that record are not used in processing. Therefore, to begin data processing at each logical record, the IBFPTR is set to either 4 (for the first logical record in a physical block) or 2568 (for the last logical record in a physical block).

After all the above things are considered, each logical record has 2560 bytes of data to be processed. Each 2560 byte record consists of 16 160 byte blocks. Each 160 byte block goes through one cycle of processing before the counter into the record is updated by 160 and the cycle restarts.

From each data block the following things are extracted:

1) time - day & fraction of day
2) milliseconds of dav
3) day
4) year (stored the first time through and checked to see if it has changed successive times, if changed-store)

Also. each time is checked to see whether it is less than the previous time. If so, a year has changed and 365 days are added to the time. Zero times are rejected. Also, data quality flags are checked.

Temperature Extraction Algorithm:

The algorith for extracting the three temperatues follows a simple scheme. Four bytes/record are tested for the value 1. (bytes 137, 140) When five bytes in a row are "1" the previous three bytes contain the temperatures. These temperatures are then submitted to a decompression function TMPCON.

Rate Extraction Algorithm

The rates are found in bits 4-15 (first bit=4) of housekeeping words 3 and 7. Housekeeping word (IHK) 2 contains the ID value in bit position 4. When the ID=0, housekeeping words 3 and 7 contain rates 1a nd 2. When the ID=1, Housekeeping words 3 and 7 contain rates 3 and 4.

First the housekeeping word (IHK) is extracted (word 13). The first three bits of each housekeeping word contains the housekeeping word number (INUM) (0-7). This is also extracted by shifting the housekeeping word to the right. This value (INUM) is checked to see if it equals 2. If it does, then the vairable ID within the program is set according to bit 4 of IHK 2. If the first INUM encountered is not equal to 2, then a look-ahead is done to find the next ID bit. If the next ID bit is zero, then the current one is set to 1 and vice versa. These ID bits are set in subroutine GETID.

Once the ID bit is set, then INUM is checked to see whether it is 3 or 7. If not, then the pointeris incremented by 160 and the process restarts. Otherwise, rate flat 1 (IRF1) is checked. If IRF1 is set, then the ID must be zero and INUM must be 3 for a rate to be extracted.

For a rate extraction to occur:

Rate 1	Rate 2	Rate 3	Rate 4
ID=0	ID=0	ID=1	ID-1
INUM=3	INUM-7	INUM=3	INUM=7
IRF1=1	IRF2=1	IRF3=1	IRF4=1

When all of the conditions are met for a particular rate, the subroutine STORE is called with an argument of (1-4) indicating which rate to store.

The counter is then updated and processing continues with the next logical record.

Algorithm For Accumulating Rates Over An Averaging Interval.

The subroutine ACCUMS is called when AVGINT is greater than 0.0.

Subroutine ACCUMS determines whether the current time is within the averaging interval. If it is, the rate value is accumulated, the counter for that rate incremented and control is returned to the subroutine store.

When the current time falls outside the interval, a call is made to DIVERT for each Rate requested. Subroutine DIVERT calculates the averaged rate, the error associated with that rate and the midtime of the averaging interfal. Control is then returned to subroutine ACCUMS where the values are written out and the counters incremented.

Error Conditions and Messages

1) If INPUT parameters are out of range then the message:

   '**BAD INPUT DATA**'

   is generated and execution terminates.

2) When tape is being positioned to the end of the requested data if an error on the tape is detected, the following message:

   '**ERROR DETECTED ON TAPE AT RECORD NO . IN FILE ** RECORD SKIPPED**'

   is generated with the location of the error and the bad record is skipped.

3) If an error is detected while reading the last record the message:

   '**ERROR DETECTED ON LAST RECORD**'

   is issued and execution terminates.

4) When data is being processed and an error is detected in any record the message in number (2) is used and execution continues with the next record.

5) When a time of zero is encountered the message:

   '**BAD TIME DATA**'

   is issued and execution continues with the next record.

LTHIST

OPERATORS GUIDE

Purpose:

The purpose of the LTHIST (Long Term Time History) program is to read the MPI format tape and extract, for a specified period of time, the Rates (1-4), Temperatures (1-3), and/or Voltages. The extracted data can then be plotted as single values with an associated time or an averaged values over the mid-time of the averaging interval. One has the option of choosing a log-linear plot for the rates. An optional listing is also created for all data points plotted.

Running LTHIST:

LNGTRM is in load module form in a TSO file SBMAE.ISEE3.LOAD(LNGTRM). It is linked with the namelist parameters in another TSO file SBMAE.ISEE3. CNTL(LNGJCL). This is the file that must be edited with QED to change options. The file can then be submitted as a job.

Step 1

The following is a list of the namelist parameters and their meaning.

Parameter	Default	Meaning
IRATES	0	1000 select Rate 1
		0100 " " 2
		0010 " " 3
		0001 " " 4
		0101 " Rates 2&4
		1111 all Rates
ITEMPS	0	100 select Temperature 1
		010 " " 2
		101 select Temperatures 1&3
AVGINT	0.0	Averaging Interval (specified in terms of hours) [cannot be >1.0 if Temperatures or Voltages are selected]
ILIST	0	0-no listing >0-listing desired
YMAXT	1000.0	maximum values for Temperatures (Y-axis)

Parameter	Default	Meaning
YMINT	0.0	minimun values for Temperatures
IERBAR	0	0-no error bars >0-error bars
YMAXR	150000.0	maximum values for Rates plotes (Y-axis)
YMINR	0.0	minimum values for Rates plotes
IBFIL	0	file number of tape with which to begin processing
IBREC	0	record number of IBFIL with which to begin processing
IENDFL	0	last file number of tape to process
IENDRC	0	last record of last file to process
ILOG	0	0-linear plots } Rates only 1-log plots }
IDV	0	0-no Voltages >0-Voltages
ISUB	0	0-no Gel-Geh >0-Gel-Geh
QIVSN	blank	tape no.

Step 2

Edit the TSO file containing namelist and JCL information. (TSO session)

```
READY
QED'SBMAE.ISEE3.CNTL(LNGJCL)'  NONUM
QED
TV
L
```

The data set is then listed. Use standard TSO change commands to edit.

1. Job card name & CPU, ID time estimate.
2. Input tape number; indicated by GO.FT03...VOL=SER-
3. Plot tape number, indicated by GO.PLOTTAPE...VOL=SER=

Step 3

Submit the job.
(TSO session)

→QED
→SUB *
→END N
→READY

If the file was saved previously and has all the needed current information a submit can be done from READY mode.

SUB 'SBMAE.ISEE3.CNTL(LNGJLL)'

When the job has completed and a Return code of zero is obtained, submit the plot tape to the Calcomp plotter. Specify files 1-999 in their log book.

JCL Requirements:

The only JCL requirements needed are to specify the run time estimates on the JOBCARD and the INPUT TAPE and PLOT TAPE.

The plot tape is specified as follows:

```
//GO.PLOTTAPE  DD  DCB=(,DEN=1), LABEL=(,BLP,,OUT),UNIT=(7TRACK,,DEFER),
VOL=SER=
```

this line would have to be edited and have the plot tape no. inserted at the boxed location.

The input tape is specified as follows:

```
//GO.FT03F001  DD  LABEL-(,NL),UNIT=(2400-9,,DEFER),DISP=(OLD,KEEP),
DCB=(RELFM=F,BLKSIZE=5128,DEN=3),VOL=SER=
```

the same type of editing would have to be done for this line.

Note that the plot tape is a seven track tape.

MERGER

Purpose PRGRMR: Jenny Jacques

Overview

MERGER is used to eliminate any file overlaps in the MPI data base. It transfers the data onto a new tape, leaving out the second occurance of any areas which overlap in time. These areas are indicated on the printout generated by MERGER.

MERGER

Overview

MERGER is a simple program that transfers MPI format records to another tape. Upon encountering a record whose time is less than the previous record, MERGER skips to the next record. These time overlaps are eliminated.

Input Required

1. An MPI format tape
2. A blank tape, 9-track, 1600 BPI

Output Generated

1. An MPI format tape with no time overlaps
2. A listing of all time overlap areas eliminated

Data Flow Algorithm

Step	Action
1	Position both tapes to file 1.
2	Read a record from MPI tape; if end of volume go to Step 6
3	Check for time overlap with previous record
4	If no overlap, write to output tape
5	Go to Step 2
6	Write end of volume on output tape
7	Stop

Definition of Variables

Variable	Type	Description
INBUFF(5128)	L*1	Equivalenced arrays which hold the input MPI record.
INBUF2(2564)	I*2	
INBUF4(1282)	I*4	
YEAR	I*4	Year, day, and milliseconds of day of current MPI record.
DAY	I*4	
MILLI	I*4	
CTIME	R*8	Seconds since launch year of 1978 for the current record.
OLDTIM	R*8	The CTIME from the previous record.
TOTAL	I*4	The number of overlaps for the tape.
COUNTR	I*4	The number of overlaps for a particular file.

Variable	Type	Description
FILNO	I*4	File and record numbers of the input tape.
IREC	I*4	
LR	I*4	Length in bytes read from the tape.
END	I*4	Ø = no file mark on read 1 = one file mark read 2 = two file marks read, end of volume

Formulas Used

There is a formula used by means of a statement function to convert year, day, and milliseconds of day to seconds since January 1, 1978. The result is stored in a REAL*8 variable:

```
TIMCON(YEAR,DAY,MILLI)=(YEAR-78)*31536.D3+
   (DAY+(4-MOD(YEAR,4))/4)*864ØØ.DØ+MILLI/1ØØØ.DØ.
```

Error Handling

Upon encountering a tape read error, the record is skipped.

MERGER

Operator's Guide

Overview

This program eliminates any time overlaps on an MPI format tape.

Input Required

1. An MPI format tape
2. A blank 1600, 9-track tape
3. Tape slot names for 1 and 2 above

Output Generated

1. An MPI format tape with no time overlaps
2. A listing of all overlap areas encountered during the run

Running MERGER

Step	Action
1	Hang both tapes in their slots
2	QED the MERGER JCL in 'SEAMP.ISEE3.JCL (MERGER)'
3	Enter the tape slot names: MPI tape = FT09F001 Blank tape = FT10F001
4	Submit the program

Error Handling

If an input tape error occurs, the record is skipped and a message is issued to the printer.

PHAPLT

Purpose PRGRMR: Jenny Jacques

A. Overview.

PHAPLT generates printer plots of PHA trigger data collected over time intervals within a trigger. The plots are of channels versus the time interval over which they were collected, and uses printer characters according to the number of counts in the channel. The end effect is a three-dimensional histogram of counts/channel/time interval.

B. Horizontal Axis.

Each line contains 100 characters corresponding to channels. If the channel range is over 300 channels, for example, the characters printed represent the sum of three adjacent channels (i.e., compressed by three):

Character	Value
Blank	Ø
1 - 9	1 - 9
A - Z	10 - 35
*	$\geq$ 36

C. Vertical Axis.

Data is accumulated over fixed-length time intervals within one trigger event. Up to 100 of these intervals may be plotted on one graph, as determined by the start time, stop time and collection interval.

D. Equipment and Restrictions.

The IBM 360/75 or /91 with Fortran H and the Wolfplot package is used. Consequently, plots may be generated on the S4060 or Calcomp with minor changes to the Wolfplot routines.

E. Execution Time.

Each trigger requires about 001001 on the 360/75 and about H00001 on the 360/91. The core used is about 250K, so the program is quickly accepted and executed.

PHAPLT
Internal Documentation

Due to the failure of the PHA trigger data instrument, the maintenance documentation is limited to the internal (source) code and comments.

PHAPLT

Operator's Guide

A. Overview.

PHAPLT is a printer plot program which generates plots of trigger PHA data on the line printer from an MPI-formatted tape. The operator must edit a small TSO file and change the tape slot numbers and the FORTRAN namelist before submitting.

B. Running PHAPLT.

PHAPLT is in load module form in TSO file SEJSS.ISEE3.LOAD(PHAPLT). It is run using the namelist TSO file SEJSS.ISEE3.CNTL(PHAJCL). This file must be edited using QED and the options changed before the file is submitted as a job.

Step 1: Know both namelist's parameter values. There are two namelists, the HKEEP and the INPUT lists. The HKEEP namelist contains information from the trigger itself. The INPUT namelist concerns scaling and other variables which are plot dependent.

- HKEEP Namelist (all parameters required).

Parameter	Meaning
CYR CDAY CHR CMIN CSEC	This is the reference calendar year, day hour, minute, second, respectively (see section entitled "The Reference Calendar").
CALT	The reference calendar value (see section entitled "The Reference Calendar").
FILE	File number of tape in which the trigger occurs.
RECORD	Record number of beginning of trigger PHA data, as specified in DRP listing.
FPHA	Parameter specified by the experimenter for this particular trigger.

• INPUT Namelist.

Parameter	Default	Meaning
SYR SDAY SHR SMIN SSEC	Start of the PHA data.	The start times for data collection in corrected GMT units.
EYR EDAY EHR EMIN ESEC	End of the PHA data.	The end times for data collection in corrected GMT units.
SCHANL	0	Start channel number.
ECHANL	4095	End channel number.
AVGINT	.1	Time averaging or collecting interval.
PRINT	F	T = print data plotted. F = don't print data plotted.

Step 2: Edit the TSO file (TSO session, * indicates computer response).

```
* READY
  QED  'SEJSS.ISEE3.CNTL(PHAJCL)'  NONUM

* QED
  TV
  L
```

The computer lists the data set. Using standard TSO change commands, change the following:

(1) Jobcard name and CPU/IO time estimate. The plot takes about H00H01 on the /75 and H00001 on the /91.

(2) Input tape slot number as indicated by the VOL=SER=JSS##.

(3) Both HKEEP and INPUT namelist parameters. If any are left out, the defaults are used as specified in Step 1.

Step 3: Submit the job, and edit (TSO session, * indicates computer response).

```
* QED
  SUB *
* (job submitted message)
  END N
* READY
```

C. Error Handling.

The following is a list of possible errors and the recovery:

(1) A message stating a namelist variable was too high, too low or missing may be printed. If this happens, the job ends with a return code of 7.

(2) An I/O error on the tape may be found. The record is skipped and the next record is processed.

(3) The file and record entered as containing the PHA data do not contain the data. The program prints:

```
ERROR - NO PHA DATA FOUND ON TAPE.ABORT
```

Check the file and record numbers for correctness.

(4) The record number does not exist on the file indicated. The message

```
END REACHED BEFORE START RECORD FOUND.
  START FILE = _, START RECORD = _
```

is printed.

PHIST

Purpose

PRGRMR: Andrew Pelletier

Overview

PHIST is the histogram generator for burst PHA data. It is similar in nature to THIST which generates burst time history, using many of the same algorithm for some routines. The PHA data are analyzed by converting the difference in time between two calendars into a counts/sec rate, since the calendars are recorded after an integral number of counts have occurred.

The histogram can be "compressed" by using every N^{th} occurance of the calendar where N is a binary number. The usual scaling definitions, logarithmic or linear scale, and start, stop times are available.

PHIST

Overview

PHIST plots the burst PHA data as a histogram vs. time. It is similar in nature to the THIST program, but uses the calendars to create the histogram. The burst PHA part of the experiment is no longer working as of January 1979. Thus, this program will not undergo the ususal additions a viable program would. Only the Operator's Guide is furnished in order to re-run any of the bursts which occurred in the 5-month life span of the burst PHA experiments. The data flow is outlined below, and the source code is available and well documented.

Data Flow Algorithm

Step	Routine	Action
1	MAIN	Initialize Calcomp plotter
2	ASKHK	Read in housekeeping data namelist "HKEEP"
3	RESET	Initialize buffer pointers, and position the tape
4	PHASNC	Collect the burst PHA data into an array
5	STORE	Convert the PHA data into a time and and a rate array
6	ASK	Read in the user parameters from the "INPUT" namelist
		If done, go to Step 14
7	TAXIS	Plot the time axis
8	YAX	Plot the counts axis
9	GRID	Plot the grid if desired
10	HIST	Plot the histogram
11	HEADW	Draw the plot labels
12	PRINTR	Write the data to the line printer if desired
13	MAIN	End the plot and go to Step 6 for another run of this burst data
14	MAIN	Rewind the tapes and unload them
15	MAIN	Stop

PHIST

Operator's Guide

Overview

This program creates histograms of the burst PHA data. Different scaling may be done in one run for the same burst by stacking the "INPUT" namelists with the user specifications.

Input Required

1. Am MPI format tape with the burst data on it.
2. The "HKEEP" namelist as follows:

Variable	Type	Description
CYR	I*4	The reference year, day, hour, minute, and second of the burst. (no defaults)
CDAY	I*4	
CHR	I*4	
CMIN	I*4	
CSEC	R*4	
CALT	R*8	The reference calendar. (no default)
FILE	I*4	File and record of the burst PHA data tape (default = 1 for both)
RECORD	I*4	
NFILES	I*4	Number of files to read to find the burst. (Range 1-999, default = 999)
*FPHA	I*4	Clock frequency FP (Range 0-7, no default)
N	I*4	Compression factor, every N^{th} calendar is used. (Range 1-20, no default)
*T1	I*4	Time history 1 trigger counter (Range 0-7, no default)
*T2	I*4	Time history 2 trigger counter (Range 0-7, no default)
*GAT	I*4	Analog threshold of GE detector (Range 0-7, no default)
*PHASRC	I*4	Ø = Detector GE 1 = Detection MEH (Range 0-1, no default)
STRTMF	I*4	Start major frame from beginning of DRP listing, used if the beginning calendars are corrupt as shown on the DRP listing. (default = 1)

*Used for header labels only.

3. The "INPUT" namelist as follows:

Variable	Type	Description
YR	I*4	Year, day, hour, minute, and second of start of plot. These times are corrected for the GMT difference. Thus, if the plot is to start at the Reference GMT, subtract the GMT correction (about 8 minutes) from the Reference time in the HKEEP namelist. (default = start of burst)
DAY	I*4	
HR	I*4	
MIN	I*4	
SEC	R*4	
YMIN	R*4	The Y-axis (counts) minimum and maximum. (Range Ø.Ø - 1,000,000., default: YMIN = 10. YMAX = 100,000.)
YMAX	R*4	
NPLOTS	I*4	The number of plots to plot. (default = 1)
TTIME	R*4	Total time in seconds of one plot. This sets the scale for the time axis. The length is 20" so the scale (sec/inch) = TTIME/20", or, TTIME = 20" * Scale (sec/inch) (Range - .1-30., default = 10.)
LOG	L*1	T = logarithmic scale F = linear scale (default = T)
PRINT	L*1	T = List the data plotted F = Suppress data listing (default = F)
ERRBAR	L*1	T = Draw error bars F = Suppress error bars (default = F)
GRID	L*1	T = Draw a grid F = Suppress the grid (default = T)

Determining the Namelists

HKEEP: All information is recorded in the "Burst Housekeeping Parameters" listing.

INPUT: This is up to the person making the PHIST run request.

PROSTRP

Purpose PRGRMR: Jenny Jacques

Overview.

PROSTRP extracts GRB data from a production tape and writes it in MPI format to another tape. See General Maintenance Guide for details on both tape structures. Input file overlap is eliminated; however, intertape overlap, if any, must be removed with program MERGER.

Restrictions.

The IBM 360/91 or /75 FTIO, FORTRANH must be used. The PARM = XL option which allows in-line bit string functions must be specified.

Performance.

An average 5-day production tape uses:

360/75 = 001002
360/91 = HOOHO1

Error Handling.

When FTIO encounters an input tape read error, the message

INPUT TAPE ERROR ON RECORD #_ OF FILE # __.CONTINUE WITH NEXT RECORD.

is printed and the next record is processed.

Output Generated.

A summary of each input file is printed indicating end time of previous file, and beginning time of next one after overlap has been eliminated. A line is printed when enough data has been collected to form a record of GRB on the output tape.

PROSTRP

Internal Documentation

Overview

PROSTRP extracts GRB data from a production tape and creates an MPI tape with all desired header data. See General Maintainence Guide for detail on tape structures.

Subroutines Used

HEDFIL - fills the file header with timing etc. values.

HEDREC - fills each logical record with timing etc. values.

TIMCHK - Checks each record for advancing GMT. It indicates time overlaps, and corrupted records.

Common Blocks

COMMON/HEAD/HEDPLC,HEADR,BUFFER,YEAR,ORB

Variable	Type	Default or Initial Value	Meaning
OYR	I*4	0	Year, day and milliseconds
ODAY	I*4	0	of day of previous major
OMSDAY	I*4	0	frame of input tape.
YR	I*4	0	Year, day, and milliseconds
DAY	I*4	0	of day of current major
MSDAY	I*4	0	frame of input tape.
OFILE	I*4	0	Output file number

Namelist Variables

NAMELIST/INPUT/IFILE,SFILE,OFILE

Variable	Type	Default or Initial Value	Meaning
IFILE	I*4	---	Input tape start file
SFILE	I*4	---	Input tape stop file
OFILE	I*4	---	Output tape start file

Other Variables

The following is a list of the variables other than in common or the namelist, excluding loop variables.

A. MAIN

Variable	Type	Intital Value	Meaning
IEND	I*4	0	0=no EOF on last read 1=EOF on last read
RECNT	I*4	Ø	Record # of current input tape file.
FILNO	I*4	IFILE	Current file #
TWRITE	L*1	T	T=new file, write times F=not beginning of file, suppress times
BAD	L*1	---	T=skip major frame due to bad times, or overlap F=process the major frame
GRBPLC	I*4	---	Pointer to GRB array. It increments by 8, each half major frame. Each half major frame contains 8 12-bit words, or 12 bytes.
BUFPLC	I*4	24	Pointer to BUFF array. It increments by 80, because each major frame is 80 2-byte words.
GRB(16)	I*2		Stores one major frame of GRB data as 16 12-byte words (24 bytes).
BUFF(1282)	L*2		Equivalenced with output buffer. This array is used to stuff GRB data in 12-byte padded to 16-bit form, into the output buffer as 12-bit words unpadded.
OBUFF(2564)	L*1		The buffer which is written to the output tape. It is equivalened to HEADR, which plugs in header data, and to BUFF, which plugs in GRB data.

B. HEDREC

Variable	Type	Initial Value	Meaning
COUNT	I*4	Ø	Counts how many bad 12-bit words there are according to the data quality flags for each major frame.

C. HEDFIL

Variable	Type	Initial Value	Meaning
BITRAT	I*4	---	The bit rate from the file header.
OLDBIT	I*4	---	Stores last files bit rate to check for change.

D. TIMCHK

Variable	Type	Intital Value	Meaning
FIRST	L*1	1	T=first file processed F=other than first file
YR DAY MSDAY	I*4 I*4 I*4	--- --- ---	The current GMT of the major frame.
OYR ODAY OMSDAY	I*4 I*4 I*4	--- --- ---	GMT from last major frame to check for overlap.
DIFF	I*4	---	Gets the difference in days to check for the year turning over.

Algorithm of Data Flow

Step	Action
1	Read in user-input namelist paramters.
2	Position input and output tapes according to namelist.
3	Read in the file header record.
4	Call HEDFIL to extract file header data.
5	Read in a data record (one major frame). If EOF encountered, to to step 12.
6	(all TIMCHK to check for timing overlap. If BAD is .true., go to step 5. Else continue with step 7 to process this major frame.
7	Strip the GRB from the Hovastadt data and plug it into BUFF.
8	If BUFF has less than 16 major frames, to to step 5.
9	Write logical record (16 major frames, 2564 bytes) to output tape.
10	Zero all arrays and re-initialze pointers.
11	Continue with step 5.
12	If double EOF, to to step 13. Otherwise, position input tape to next file and go to step 3.
13	Write a double EOF on output tape and end program.

PROSTRP

Operator's Guide

Purpose.

To extract GRB data from a production tape and place it in MPI format on another tape.

Input Required.

(1) 9-track, 1600 BPI production tape.
(2) 9-track, 1600 BPI output tape.

Output Received.

(1) Listing of strip process for each file.
(2) TDB tape, 1 file per decom tape.

Running PROSTRP.

The disk file 'SEJSS.ISEE3.CNTL(PSTRPJCL)' must be edited and the following changed:

(1) Estimated CPU-IO run times on JOBCARD.
(2) Input and output tape slot numbers.
(3) Input tape start and stop file numbers.
(4) Output tape start file number.

The operator is assumed to be familiar with basic TSO edit commands.

In the following procedure, → indicates computer response.

(Log on TSO /75 or /91)

```
→ READY
  QED  'SEJSS.ISEE3.CNTL(PSTRPJCL)' NONUM
→ QED
  TV
  L
```

A list of the file follows. Use basic TSO change commands and change the necessary parameters as described above. Then proceed:

```
→ QED
  SUB *
→ (Job submitted line is issued)
  END N
  READY
```

Error Messages.

The following message may be printed when an error condition is encountered in reading the input tape:

> "INPUT TAPE ERROR ON RECORD # _ OF FILE # _. CONTINUE WITH NEXT RECORD."

QKSTRP

Purpose PRGRMR: Jenny Jacques

Overview

QKSTRP extracts GRB data from a quick look tape and writes it onto another tape in GRB format. See General Maintenance Guide for details on tape structure.

The header record from the quick look tape is copied directly to GRB tape.

Input Required

A 7-track quick look tape.

Output Generated

1. A 9-track GRB format tape.
2. A listing containing summary data for each major frame processed.

Restrictions

The IBM 360 /91 or /75 FI10 and Fortranh with PARM-XL option are used.

Performance

On the 360/75, the time is 001001/.500 records.
On the 360/91, the time is H00H00/.500 records.

Error Handling

If a tape read error occurs, the record in skipped, an error message issued, and processing continues with the next tape.

QKSTRP

Internal Documentation

Overview

The General Maintenance Guide describes the entire tape structure for both quick look and GRB tapes, therefore it will not be repeated here. It also describes the position of GRB within each record.

QKSTRP consists of only one routine, MAIN.

Variables Used

VARIABLE	TYPE	INITIAL COND. OR DEFAULT	MEANING (or contents)
GRB	I*4	Ø	12 bits of GRB from the major fram
OBC	I*4	1	POINTER to GRB in OBUFF
INREC	I*4	Ø	input record counter
OUTRC	I*4	Ø	output record counter
IEND	I*4	Ø	incremented when EOF mark is found
ISTART	I*4	1	input tape start file number
ISTOP	I*4	999	input file # to stop with
OSTART	I*4	-999	output file # to begin with
EQUART	I*4	---	quarter frame #
QBUFF(1024)	I*2	Ø	holds 12 bit GRB words in 16 bit field
OBUFF2(768)	I*2	Ø	holds 12 bit GRB words not padded to 16 bits
BUFF(600)	L*1	Ø	holds input record
HEADR(1536)	L*1	Ø	holds file header record to transfer to output tape
INDIC	I*4		indicates which half of major fra to process
YEAR	I*4	---	the beginning time of the dump as specified in the header
DAY	I*4	---	
MSDAY	I*4	---	

VARIABLE	TYPE	INITIAL COND. OR DEFAULT	MEANING(or contents)
IERR	I*4	Ø	the number of tepe read errors
OUT2(1282)	I*2	---	used to plug in header times
OUT4(641)	I*4	---	used to plug in s/c clock milliseconds of day, and spin period
SPIN	I*4	---	the spin period as read from header record
SCLOCK	I*4	---	the s/c clock, incremented each data block & filled in by program

LOGIC FLOWCHART

STEP	ACTION
1	Read the header record from the input tape.
2	Transfer S/C clock, spin period, bit rate, and header times to output array.
3	Write header record to output tape.
4	Read in one record, equal to 1/4 major frame.
5	If it is quarter frame 1 or 3, go step 4. Otherwise continue to step 6.
6	Extract the GRB from quarter frames 2 or 4. If it is 2 and quarter frame 4 of last major frame was skipped, fill in zero's and write last major frame to tape before extracting beginning of this one. If it is 4 and quarter frame 2 was skipped this major frame, then fill the first part of major frame with zeros.
7	If buffer is full, 16 major frames, then compress the 16-bit word array.
8	Plug in quality flag and repeat times, write S/C clock, & spin period.
9	Write record to output tape.

QKSTRP
Operators Guide

PURPOSE

To extract GRB data from a quick look tape and copy it to another tape.

INPUT REQUIRED

1. 7-track 556 BPI quick look tape.
2. 9-track 1600 BPI output tape.

OUTPUT RECEIVED

1. Listing of extraction process with resulting # files & records on both tapes.
2. Output GRB tape with another file of GRB data.

The file containing the spcifications for the program must be edited so slot numbers and file numbers may be specified.

EDITING THE FILE

Sign on TSO → indicates computer prompt. The operator is assumed to know basic TSO edit commands.

```
→ READY
  QED 'SEJSS.ISEE3.CNTL(QKSTRPJCL)' NONUM
→ QED
  TV
  L
```

The file is listed. Change the slot number and the namelist parameters as desired. The following is a list of the parameters & their meaning:

PARAMETER	MEANING
ISTART	Start file of input 7-track tape. It defaults to 1.
ISTOP	End file numberof the 7-track input tape. The default is 999.
OSTART	The output tape file number. This parameter must be specified.

SPECACC

Purpose PRGRMR: Jenny Jacques

It is useful to accumulate background radiation spectra over a period of time so prominent features will emerge from the overall noise background. SPECACC adds counts in each channel from two spectral files, creating a third spectral file. In this manner, a new set of background data may be added to an existing file of previous sums of data sets.

For background data, this resulting file represents a spectrum from data over a long period of time. SPECACC is not used for trigger data because trigger spectra are short time events, and accumulated spectra of trigger events have no scientific value.

For quick-look tapes, when the spectral file is added to the accumulations file, the start time is used as the stop time in the resultant file header.

For MPI tapes, the stop time from the input data set is used as the stop time of the resultant spectral file. This difference is due to the fact that the only timing information available from quick-look tapes is start time. However, all timing is present on the MPI tapes.

After each file has been added, a catalog associated with the file of accumulations is updated to indicate the new addition. An example of this catalog follows.

SPECACC

OPERATOR'S GUIDE

Purpose.

This program sums together any two spectral files on PDP-11/70 disk. The dates from each file are recorded in a corresponding time list file.

Input Required.

Two spectral disk files.
One time list file.

Output Generated.

One spectral file with corresponding time list file.

Running SPECACC.

Sign on [100,105] with: "MCR>HEL [100,105]".

(1) If a previous accumulation file is being added to, such that the timing list file is in existence, skip to step (2).

To create a new accumulation file from the first spectral file you are to sum:

```
MCR>PIP (accum name)=(spectral file name)
```

Example: MCR>PIP QACC.BKG=Q222.BKG

(2) Run the program ($ means esc key):

```
MCR>RUN SPECACC$
ENTER NAME OF FILE OF ACCUMULATIONS
EXAMPLE.BKG
  IS THIS A NEW ACCUMULATION FILE?  (Y OR N)
Y
ENTER NAME OF TIME LIST FILE
LIST.BKG
ENTER NAME OF FILE TO ADD TO ACCUMULATION
SPEC.BKG
 ADD ANOTHER FILE TO THIS ACCUM FILE?  (Y OR N)
Y
ENTER NAME OF FILE TO ADD TO ACCUMULATION
SPEC.BKG
 ADD ANOTHER FILE TO THIS ACCUM FILE?  (Y OR N)
N
END OF FILE ACCUMULATIONS.
SPECAC  --  STOP
```

NOTE: There is an unlimited number of files which may be summed in one run.

(3) Each file which has been added to the accumulation file has created a new version. If only the latest file is desired (this one has all the accumulated files), issue a purge command as follows:

```
MCR>PIP name/PU
MCR>PIP name;1=name;*/RE
```

Error Handling.

All user errors are trapped by the program and requests to correct the input issued.

THIST

Purpose

PRGRMR: Andrew Pelletier

THIST, or Time History program, is used to analyze burst time histories 1 and 2 for the GRB experiment on ISEE-3 from an MPI format tape. THIST produces Calcomp histograms of the rate (counts/sec) vs. time, along with a listing of data for this plot. In addition, the DT values for the time history (See General Maintenance Guide: Data Block Structure) may be summed in groups of 1, 2, 3, 4, 6, or 12 for a smoothing compressed time effect. THIST also allows the data to be collected according to a specified time interval in seconds. With this option, however, only a data listing is available.

Because the calendar is reset about every 34 minutes, the program must be supplied with a background calendar and it's Greenwich Mean Time (GMT) for the time history calendar may be computed. (GMT of the data is that of telemetry readout, not burst time.) This means that the times used as reference must be within 34 minutes (2048 seconds) of the burst time. THIST divides this time in half, so that the reference times must be $\pm$ 1024 seconds of the time history calendar, or 8 major frames.

Due to the slow bit rate, the data is often of only fair quality, containing data gaps and scrambled calendars. An attempt is made to weed these times out. However, since all timing is based on the first calendar, if it is bad then the program fails.

Quick-Look data may also be analyzed with THIST. However, there is no timing information available on the quick-look tape and thus only the features of the plot may be useful.

THIST

Internal Documentation

Overview

THIST plots the burst time histories 1 and 2 on the Calcomp plotter using the Cal780 plot package. An optional listing or data file may be created. A single time history plot requires 001H01 on the /91, and 001003 on the /75, plots and listings.

Input Required

1. User namelists

A. &HKEEP

&HKEEP	FT1 FT2	0-3	Frequency of the delta t clock (Hz)
	GAT	0-7	Ge ananlog threshold (no value necessary)
	HAT	0-7	HOH analog threshold (no value necessary)
	N1 N2	0-7	Number of photons per delt t
	T1 T2	0-7	Trigger counter
	TH1	-1,0-4095	Threshold, Time History 1 (no value necessary)
	TS1 TS2	0-3	Detector (no value necessary)
	CALT		Reference Calendar
	CYR CDAY CHR CMIN CSEC	Start Time	Reference GMT (if quick-look, this is the estimate of the trigger time, rounded to the hour)
	PHACAL	any integer	Sets a default start time, should be close to CALT
	NFILES		Number of files to read (-1 for quick-look)
	FILE		File of the tape where the time history data begins
	RECORD		Record of the tape where the time history data begins

B. &INPUT

LOG	.true.	Semi-log graph (default)
	.false.	Linear Graph
PRINT	.true.	Dump data on line k printer
	.false.	No dump (default)
ERRBAR	.true.	Add error bars to the hiso-gram
	.false.	No error bars (default)
GRID	.true.	Make a grid on the plot (default)
	.false.	No grid
NTHM	1	Plot Time History Memory 1 (default)
	2	Plot Time History Memory 2
TTIME	0<ttime≤1024.0	Total number of seconds over which to plot (240.=default)
YMIN	0≤ymin<ymax	Lowest y coordinate (1. is default)
YMAX	ymin<ymax ≤1,000,000	Highest coordinate (5. is default)
NCOMPR	0≤NCOMPR≤12	Compression factor in steps of 1,2,3,4,6, or 12
DISK	.true.	Create a disk file of the data.
	.false.	No file created (default)
YR	78,79,80, or 1978, ect.	Start time of plot. If not specified, PHACAL must be specified
DAY	1-366	
HR	0- 23	
MIN	0- 59	
SEC	59.999	
BKG	≥0.0	Background rate to be sub-stracted (default=0.0)
NPLOTS	0	Makes multiple plots changing only the start time
INTERV	≥0	Interval, in seconds, by which to collect the data. Default is not to use this method of data collection

2. An MPI structured Gamma-Ray Burst tape.

3. An empty disk file, if disk information is to be created.

Output Generated

1. Option for Calcomp plots.

2. Option for a data listing. There are two types of listings available:

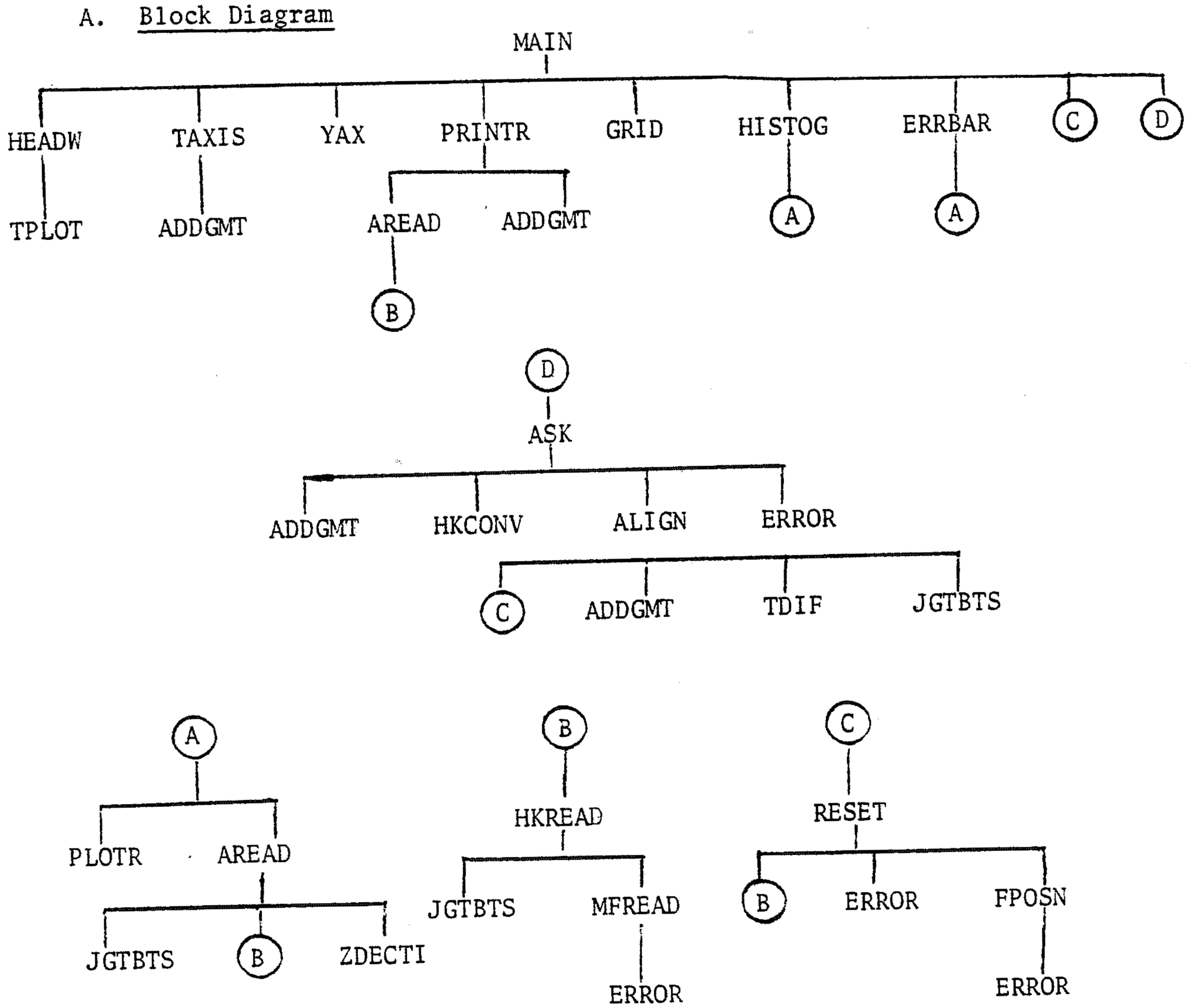
   a. Listing of rates compressed by 1,2,3,4,6, or 12.
   b. Listing of the data taken by a fixed interval of time.

3. Option for a disk file of the rate, DT, and time values.

Program Structure

Program Structure

A. Block Diagram



B. Module Definition

Module	Function
MAIN	Controls program flow
ADDGMT	Adds seconds to yr, dy, hr, min, sec.
ALIGN	Uses a pre-specified GRB calendar and GMT date to derive a GMT date for any calendar.
AREAD	Read in an update and weed out the 12 bit dt values. Concurrently, check for jumps in the memory of the time and signal both with a DT of -1, for they both signal breaks in the data. The end of data is signalled by a dt = 999.
ASK	Reads in and checks all user input options.
ERRBAR	Plots error bars on the histogram.
ERROR	Prints out desired error message.
FPOSN	Positions the MPI format tape at a particular file and record.
GRID	Draws a plot grid.
HEADW	Plots the plot header.
HISTOG	Plots the histogram
HKCONV	Separates housekeeping bits into their various parameter values.
HKREAD	Assembles up to 8 major frames to make a housekeeping update. It recognizes a HK update end by 1) housekeeping word 7, 2) a drop in the housekeeping number, and 3) an End of Tape (EOF). On EOF, the last HK update is not lost. As each major frame is read, the appropriate housekeeping is saved. The output is through the common block /PHKUCM/.

Module	Function
JGTBTS	Converts a string of N bits into a word.
MFREAD	Reads a logical record and outputs a Major Frame of GRB data. Logical records are read as needed, and the file header record is ignored. End of tape is signalled by QEND = .TRUE.
PLOTR	Moves the pen to a given position.
PRINTR	Prints the plot data and/or writes it to the disk file.
RESET	Reads in housekeeping updates until bits A and B indicate the appropriate time history. Sets up READCM.
TAXIS	Labels the time axis.
TDIF	Subtracts 2 times in the form year, day, milliseconds of day.
TPLOT	Plots time.
YAX	Draws and labels the Y axis.
ZDECTI	Returns the time elapsed between the beginning of the time history and the current calendar value.

C. Data Flow

The following is an explanation of the major processing flow.

Step	Routines	Action
1	MAIN,ASK	Read in user input
2	ASK,ALIGN	Set up timing reference
3	MAIN,RESET	Finds time history data on tape
4	MAIN,HISTOG	Draw one plot of histogram
	HISTOG,AREAD	Extract a DT value
	AREAD,HKREAD	Extract a housekeeping update
	HKREAD,MFREAD	Extract a data block
	AREAD,ZDECTI	Get proper timing from calendar

Step	Routines	Actions
5	MAIN,ERRBAR	Use Step 3 and loop A to draw error bars if desired
6	MAIN,PRINTR	Use Step 3 and loop A to print DT values or write to disk
7		Go to Step 3 for next plot until data is done
8		Go to Step 1 for next plot sequence until done

D. Error Handling

There are five kinds of errors:		May happen in:
1.	Incorrect input namelists	ASK
2.	Tape read errors	FPOSN,MFREAD
3.	Missing data blocks	Driver
4.	Zero data blocks	MFREAD
5.	Zero Δt values	HISTOG

Incorrect input namelists cause an error message to be printed on the line printer. The program may or may not abort depending on the severity of the error.

A tape read error generates an error message, but execution continues.

Missing data blocks are expected and allowed for, but no error message is output.

Zero data blocks are ignored.

Zero Δt values cause a star to appear on the time axis, but no further action is taken.

All error messages are printed by subroutine ERROR. They are accompanied by a list of the contents of all common blocks.

E. Common Blocks

1. /CALCOM/JICALT

Variable	Type	Description
JICALT	I*4	The first calendar in the time history

2. /FLGCOM/QPRINT,QPLOT,QGRID,QERBAR,QDISK

Variable	Type	Description
QPRINT	L*1	T = create a data listing
QPLOT	L*1	T = create a histogram plot
QGRID	L*1	T = draw a grid on the plot
QERBAR	L*1	T = draw error bars on the plot
QDISK	L*1	T = create a data disk file

3. /KHKCOM/KFT1,KFT2,KGAT,KHAT,KN1,KN2,KT1,KT2,KTH1,KTH2,KTS1,KTS2

These are the housekeeping parameters as defined in "General Maintenance Guide". They all are I*4, and the actual parameter names are preceded by a "K".

4. /LHKCOM/LFT1,LFT2,LGAT,LHAT,LMT,LN1,LN2,LT1,LT2,LTH1,LTH2,
LTS1,LTS2

These are the housekeeping parameters of KHKCOM converted to engineering units.

5. /PHKUCM/PHKU(24,9)

This array, L*1, contains 24 bytes (one housekeeping data block) of 8 data blocks (one update). The 9th position may hold the next update's first data block.

6. /READCM/IDPOIN,IHKU,MEMREC,SAVTIM,IERR,IPOINT,NMF,QFLAG,QEOF

Variable	Type	Description
IDPOIN	I*4	Points to next DT bit number to process; Increments by 12
IHKU	I*4	Housekeeping data block number to process
MEMREC	I*4	Current memory address

Variable	Type	Description
SAVTIM	R*4	Time from last DT value
IERR	I*4	Records which stage the memory is in: See comments at beginning of ZDECTI code for meaning of IERR
IPOINT	I*4	Pointer to major frame byte position in the tape record; Increments by 160
NMF	I*4	Pointer to major frame number as PHAU is filled
QFLAG	I*4	T = Major frame for this update was read into last updates 9th position
QEOF	I*4	T = End of file on tape was encountered

7. /SCALCM/XMIN,XMAX,YMIN,YMAX,QSCALE

Variable	Type	Description
XMIN	R*4	Plot time axis start in seconds
YMAX	R*4	Plot time axis end in seconds
YMIN	R*4	Plot Y axis minimum
YMAX	R*4	Plot Y axis maximum
QSCALE	L*1	T = logarithmic scaling

8. /TAPECM/LUN,FILE,RECORD,NLOGRC,NFILRI,NFILES,IRC

Variable	Type	Description
LUN	I*4	Logical unit number of tape
FILE	I*4	Start file number
RECORD	I*4	Start record number
NLOGRC	I*4	Number of logical records read
NFILRI	I*4	Number of files read
NFILES	I*4	Maximum number of files to read before ending in error because it's too long
IRC	I*4	Return code to indicate if there was any problem with the run

9. /THMCOM/GMTDIF,JCALT,NTHM,NCOMPR

Variable	Type	Description
GMTDIF	R*4	Difference in seconds between the GMT time on tape and the actual GMT value
JCALT	I*4	Calendar of data blocks being currently analyzed
NTHM	I*4	1 = Time History 1 analysis 2 = Time HIstory 2 analysis
NCOMPR	I*4	The number of DT value to seem as a single point; must be a factor of 12. (12 DT's/data block)

10. /TIMECM/IRY,IDAY,IHR,MIN,SEC,IYR2,IDY2,IHR2,MIN2,SEC2

Variable	Type	Description
IYR	I*4	Start time of the current plot
IDAY	I*4	
IHR	I*4	
MIN	I*4	
SEC	R*4	
IYR2	I*4	End time of the current plot
IDY2	I*4	
IHR2	I*4	
MIN2	I*4	
SEC2	R*4	

11. /LNFTCM/FRT,LN,LNCOMP,KBA,TNTRV,BKGRAT

Variable	Type	Description
LFT	I*4	Clock frequency, (calendar units/sec)
LN	I*4	Number of counts per DT readout
LNCOMP	I*4	Number of counts per averaging interval, NCOMPR* LN

Variable	Type	Desciption
KBA	I*4	Word created from housekeeping #6, B,A bits which indicate the mode: KBA = Ø = Background 1 = PHA 2 = Time History 1 3 = Time History 2
TINTRV	R*4	Collection interval in seconds if using time intervals to step through the data
BKGRAT	R*4	Background rate in counts/sec to be substracted off before plotting

F. Common Block Cross Reference

I = uses it as input
0 = writes into it, output

	Common	Occurrence	Input/Output	Function
1.	CALCOM	ASK	0	The first calendar in the time history readout
		ERROR	I	
		PRINTR	I	
		ZDECTI	I	
2.	FLGCOM	MAIN	I	Plot option flags
		ASK	0	
3.	KHKCOM	AREAD	I	Housekeeping in telemetry form
		ERROR	I	
		HKCONV	I	
		HKREAD	0	
		RESET	I	
4.	LHKCOM	AREAD	I	Housekeeping in engineering form
		ASK	0	
		ERRBAR	I	
		ERROR	I	
		HEADW	I	
		HKLONV	0	
		HKREAD	0	
		PRINTR	I	

	Common	Occurrence	Input/Output	Function
5.	PHKUCM	ALIGN	I	One housekeeping update
		AREAD	I	
		ERROR	I	
		HKREAD	O	
		RESET	I	
6.	READCM	AREAD	O	Pointers to data arrays, current memory address, and current time
		ERROR	I	
		HKREAD	O	
		HFREAD	O	
		PRINTR	I	
		RESET	O	
		ZDECTI	O	
7.	SCALCM	MAIN	I	Plot scale information
		ASK	O	
		ERRBAR	I	
		HISTOG	I	
		MFREAD	I	
		PRINTR	I	
		TAXIS	I	
		YAX	I	
8.	TAPECM	MAIN	O	Tape logical unit, fits, and record numbers
		ASK	O	
		ERROR	O	
		RESET	O	
9.	THMCOM	AREAD	I	GMT discrepancy, current calendar, time history number, and compression factor
		ASK	O	
		ERROR	I	
		HEADW	I	
		PRINTR	I	
		RESET	I	

	Common	Occurrence	Input/Output	Function
10.	TIMECM	ALIGN	I	Start and end times of the current plot
		ASK	O	
		HEADW	I	
		PRINTR	I	
		TAXIS	O	
11.	LNFTCM	ASK	O	Time history parameters from telemetry defining frequency, etc.
		ERRBAR	I	
		HISTOG	I	
		HKCONV	O	

Plot Description

The following list describes each piece of the plot and which routine drew it.

Plot Piece	
1. Axis time labels and tick marks	TAXIS
2. Grid	GRID
3. Histogram	HISTOG
4. Axis rate labels and tick marks	YAX
5. All labels at top of plot except actual times	HEADW
6. The start, stop times at top of plot	TPLOT
7. Error bars	ERRBAR

Miscellaneous Notes

1. The timing system in the program is stored in units of seconds from the first time in the time history memory. ZDECTI converts the calendars to this time frame, and ALIGN sets up the timing system for the plot. The delta time returned by AREAD for plotting is in this system as well.

2. In ADDGMT, the check on SECADD greater than 3000 was installed to prevent CPU timeouts from bad input values.

3. In ALIGN, JCALT is assumed to be within 1024 seconds of the first calendar in the time history memory due to the calendar being only 24 bits long.

4. For the logarithmic scale, the Y axis values are set to the power of 10 above the user-input maximum, or below the user-input minimum.

Restrictions

1. Uses FT10

2. Uses CAL780

3. Uses FORTRAN H

4. Uses LAND bit function

5. When HK#6 is missing from an update, bits A & B are assumed to have the same value as the previous update.

6. The information in each data block is assumed to be correct. Much work has been done allowing for missing data blocks, but no attempt has been made to detect or correct other errors.

Basic Algorithm

The MAIN program is responsible for overall control. It determines the order of the actions of the other routines.

Subroutine ERROR handles all errors, printing out appropriate error messages and setting return codes. It also prints all the common blocks to aid in determining the source of the error.

Subroutine ASK reads the input namelist (see Operators Guide), checks the validity of the input and converts to internal variable names and internal units. Many default values are set in ASK. The rest are set in the Block Data subprogram.

Interface Between Driver and Plotter

Before any information can be extracted, timing must be adjusted and all pointers must be positioned properly. The timing needs adjusting only when the namelist input values change. The timing is set by subroutine ALIGN, called from ASK. The pointers, however, must be reset to the start of the data buffer everytime a time history is read. This is accomplished by the calls to RESET in the MAIN program.

The data is extracted by repeated calls to AREAD. EAch call returns a Δt value and a time value, both in seconds. The time value is the time when the Δt begins.

There are 2 flagged conditions:

1. End of data; Signalled by DT = -999
   The time value is meaningless.

2. Skip in data; Signalled by DT = -1.
   The time is the time when the data restarts.

AREAD is responsible for setting these flags, the plot routines interpret them.

NOTE: After AREAD returns -999, it cannot be called again until RESET is called. Also not that all of the data is returned by AREAD. It is up to the plot routine to decide what data does not fit on the the plot. The plot routines is responsible for handling zero Δt values.

Subroutine PLOTR takes input in seconds, clips using XMIN,XMAX,YMIN and YMAX, and converts to plotter units. XMIN and XMAX are both in units of seconds from the first time in the time history memory.

The Y values are not quite as simple. On a linear scale, the input Y value, YMIN and YMAX are all in CTS/SEC. However, when the plot is semilog, YMIN and YMAX are the common logarithms of the max and min values.

For instance, if the graph was to go from 1Ø to 10,000 on a log scale, YMIN = 1, YMAX = 4, QSCALE = .TRUE. The input, however, is still in CTS/SEC.

The plotter routines are very independent of one another. Each one has its own function and they call each other only rarely. However, they must call each other in a certain order, for one routine may set a value that another routine needs. A list of these follows:

VARIABLE	SET OR CHANGED BY	USED BY	PASSED THROUGH
IYR2,IDY2,...SEC2 (stop time)	TAXIS	HEADW	TIMECM
NTICKS	YAX	GRID	MAIN
YMAX,YMIN	YAX	HISTOG,PLOTR,ERRBAR	SCALCM
LMT	HKREAD	PRINTR,HEADW	LHKCM

Return Codes

If the program returns with a return code of :

3: Recoverable error, execution continued
7: Non-recoverable error, abonormal termination

THIST
OPERATOR'S GUIDE

Purpose:

To plot ISEE-3 Gamma Ray Burst time history information on the IBM 360 CALCOMP plotter.

Input Required:

(1) MPI tape (see General Users and Maintenance Guide for detail on structure).

(2) Two Namelists, specifying input parameters.

Output Generated:

(1) 7-track Calcomp plot tape for plots.

(2) Optional data listings of time history(s).

Terminology:

To run this program, several changes must be made to the namelist data sets, or a new data set must be created. Before going into the explanation of how to make the changes, a few terms must be defined:

The "window" is the rectangle of edges on either side, above and below the plot. At the base of the window is the time axis with time of day increasing from left to right (see Figure 1). The most common change that has to be made is moving the window left and right, changing the start time (left edge) and the stop time (right edge). The start time is in units of year, day of year, hour, minute and second of day. The total time, which is one of the input parameters, is the number of seconds from the start time to the stop time. For example, if the desired start time is

78 300/16:34:20.000

and the desired stop time is

78 300/16:34:25.000,

the start time would be 78 200/16:34:20.00, and the total time would be 5.000 seconds.

There are two time histories that cover the same time span--Time History 1 and Time History 2. The housekeeping parameters (N, FT, T, TH, TS--see Housekeeping Data Block Structure in the General Maintenance Guide), are different for the two time histories. All values are input to make a single plot.

***There are 20 inches for one plot, so the scale is always (TTIIME/20) seconds/inch.

Structure of the Namelist Data Set

The input parameters must be stored in a data set referred to as the Namelist Data Set. There are four parts to the namelist data set (see Figure 2):

(1) Beginning JCL (slashed in Columns 1 and 2).
(2) Housekeeping namelist (&HKEEP ... &END).
(3) Input namelist (&INPUT ... &END).
(4) End JCL.

The End JCL never needs altering, and the beginning JCL only needs 2 changes, the input tape slot number and the output tape slot number.

The &HKEEP namelist defines housekeeping for the trigger. The housekeeping parameters must be given by the scientist requesting the plot, or found in the DRP data listing if available.

Each &INPUT namelist defines one time history plot set. The first &INPUT namelist draws on a time history plot set, the second &INPUT namelist draws another, etc. The number of &INPUT namelist must always be the same as the number of time history runs desired.

Input parameters may be omitted from the &INPUT namelists. Any parameter omitted from the first &INPUT namelist will assume a default value, as specified in the section "Namelists." Any parameter omitted from any other &INPUT namelists will assume the value it had in the preceeding namelist.

Special Instructions for Plotting a Burst for the First Time

Before a burst can be plotted, the associated housekeeping and timing parameters are needed, as well as, the file and record of the data on the tape. For MPI tapes, these parameters are all in the DRP data dump. For quick-look tapes, see "Special Instructions for Quick-Look Tapes," Page 3.

Housekeeping & Timing for &HKEEP Namelist

In the DRP data dump, find the background data that immediately preceeds the time history data to be plotted. Look for the RR column and not the first HK#Ø or #4 occurrence after RR changes from Ø to 1. Use the calendar and GMT times from this data block as the reference times. (CALT, YR, DAY, etc.) Record the other parameters needed for the run:

FT1	N1	
TH1		
TS1		
FT2	N2	
TH2		
TS2	HAT	GAT

All of these parameters have corresponding names in the &HKEEP namelist.

The parameters described in this section make up the &HKEEP namelist. Enter them in the Trigger Parameter Log and the run file.

Creating the Namelist Data Set

Create a new data set with TSØ (see Figure 6). Copy the beginning JCL from Figure 2, changing only the tape slot volume serial numbers. Enter the housekeeping namelist (&HKEEP) after the "//GO.DATA5 DD *". After this, enter the &INPUT namelist.

NTHM=1 &end
NTHM=2 &end

Copy the end JCL from Figure 2. Change slot numbers as in Figure 3. The namelist data set is now ready for use.

Special Instructions for Quick-Look Tapes

Before a quick-look tape can be plotted, it must be stripped and converted to MPI format. See Operator's Guide for QKSTRP.

From Upendra Desai or Bonnard Teegarden, get the follwong housekeeping parameters:

FT1	N1	T1
TH1		
TS1		
FT2	N2	T2
TH2		
TS2	HAT	GAT

Also, get an estimate of when the trigger occurred (an exact time is not needed). Round this estimate to the hour. This GMT is the input parameters CYR, CDAY, CHR, CMIN, CSEC.

From the data listings of IC3DMP, find the lowest calendar in the time history memories. Round this value down to the 10,000's. This value is entered as both CALT and PHACAL.

Create the data set as described under plotting a burst for the first time.

Namelist:

(1)	&HKEEP	FT1 FT2	0-3	Frequency of the delta t clock (Hz).
		GAT	0-7	Ge analog threshold (no value necessary).

	HAT	0-7	HOH analog threshold (no value necessary).
	N1 N2 }	0-7	Number of photons per delta t.
	T1 T2 }	0-7	Trigger Counter.
	TH1	-1,0-4095	Threshold, Time History 1 (no value necessary).
	TS1 TS2	0-3	Detector (no value necessary).
	CALT		Reference Calendar
	CYR CDAY CHR CMIN CSEC	Start Time	Reference GMT (if quick-look, this is the estimate of the trigger time, rounded to the hour).
	PHACAL	any integer	Sets a defulat start time, should be close to CALT.
	NFILES		No. of files to read (-1 for quick-look).
	FILE		File of the tape where the time history data begins.
	RECORD		Record of the tape where the time history data begins.
(2) &INPUT	LOG	.true. .false.	Semi-log graph (default). Linear Graph.
	PRINT	.true. .false.	Dump data on line k printer. No dump (default).
	ERRBAR	.true. .false.	Add error bars to the histogram. No error bars (default).
	GRID	.true. .false.	Make a grid on the plot (default). No grid.
	NTHM	1 2	Plot Time History Memory 1 (default). Plot Time History Memory 2.
	TTIME	$0<ttime\leq 1024.0$	Total number of seconds over which to plot (240.=default).

YMIN	$0 \le ymin < ymax$	Lowest y coordinate (1. is default).
YMAX	$ymin < ymax \le 1,000,000.$	Highest y coordinate (5. is default).
NCOMPR	$0 \le NCOMPR \le 12$	Compression factor in steps of 1, 2, 3, 4, 6, or 12.
DISK	.true. .false.	Create a disk file of the data. No file created (default).
YR	78,79,80 or 1978, etc.	Start time of plot. If not specified, PHACAL must be specified.
DAY	1-366	
HR	0- 23	
MIN	0- 59	
SEC	59.999	
BKG	≥ 0.0	Background rate to be substracted (default=0.0)
NPLOTS	0	Makes multiple plots changing only the start time.
INTERV	≥ 0	Interval, in seconds, by which to collect the data. Default is not to use this method of data collection.

The job takes approximately 1/2 minute CPU, 2 minutes I/0 (H00002) per time history. When it is finished, get the printout, and take the files list plot tape to the plotter room to be plotted. Specify to plot files 1-999 on 12" paper, fine black pen. Each plot takes 30 inches of paper and 7 minutes time.

Example:

```
//ZNCFC047 JOB (SR01218Z3A,T,ISEE03,H00002),033
//PLOT EXEC PGM=NOV1378,REGION=150K
//STEPLIB DD DSN=SEAMP.THIST.LOAD,DISP=SHR
//FT05F001 DD DDNAME=DATA5
//FT06F001 DD SYSOUT=A,SPACE=(CYL,(15,15))
//FT08F001 DD LABEL=(,NL),UNIT=(2400-9,,DEFER),
//      DCB=(RECFM=FB,LRECL=2564,BLKSIZE=5128,DEN=3),VOL=SER=JSS17
//PLOTTAPE DD UNIT=(7TRACK,,DEFER),LABEL=(,NL),DCB=DEN=1,
//      DISP=(NEW,KEEP),VOL=SER=JSS18
//DATA5 DD *
 &HKEEP T1=6,T2=6,TH1=190,TH2=4095,GAT=1,HAT=0,TS1=0,TS2=2,N1=5,N2=5,
             FT1=1,FT2=2,FILE=01,RECORD=1,NFILES=1,
         CYR=78,CDAY=309,CHR=18,CMIN=50,CSEC=0.,CALT=000,PHACAL=0,&END
 &INPUT    NTHM=2,TTIME=50.,YR=78,DAY=309,HR=19,MIN=01,SEC=30,&END
 &INPUT    NTHM=2,TTIME=10.,NPLOTS=2,SEC=50,&END
 &INPUT     NTHM=2,MIN=0,SEC=0,NPLOTS=1,TTIME=240.,&END
/*
// EXEC NOTIFYTS
/
```

This example will produce:

(1) Time history 2 plots with a time scale of 2.5 seconds/inch starting at 78/309 19:01:30.

(2) Two time history 2 plots with time scale of .5 seconds/inch starting at 78/309 19:01:50.

(3) One time history 2 plot with scale of 12 seconds/inch starting at 78/309 19:00:00

WSPECT

Purpose PRGRMR: Jenny Jacques

WSPECT, window spectra program, accumulates PHA background events within user-defined pulse-height channel windows for user-defined time intervals. It converts these window rates by applying appropriate normalization factors derived from the GEL,GEH count rates & plots the rates as a function of time.

Input to the system is an MPI format tape or a disk file previously created with WSPECT from an MPI tape. WSPECT has options to generate data listings, Vector General plots, and permanent disk files of the spectral data.

Specifications & Capabilities

The user may define up to 4 channel windows and any collection time interval between 0.0 and 24.0 hours. The data collection over this interval starts & stops according to user defined GMT's.

The Vector General plots may display up to all four windows, with or without error bars on each of the windows.

NOTE: Time averaging intervals should be at least 1 hour to ensure correct GEL, GEH averages because these values occur only 1 or 2 times per hour.

Let T_i to T_{i+1} = the averaging time interval

$\overline{GEL_i}$ = average value of GEL rate (cnts/sec) during the i^{th} interval

$\overline{GEH_i}$ = average value of GEH rate (cnts/sec) during the i^{th} interval

N_i = total number of PHA events over all channels during the i^{th} interval

W_i = total number of PHA events in channel window during the i^{th} interval

The window rate between T_i and T_{i+1} = $R_i = \frac{W_i}{N_i} (\overline{GEL_i} - \overline{GEH_i})$

and the error is approximately

$$E_i = \frac{R_i}{\sqrt{W_i}}$$

WSPECT

Internal Documentation

Overview

WSPECT, window spectrum plot program, creates files of the background spectra collected from an MPI tape over user-specified time & channel intervals (windows), and averaged over user-specified time intervals. Then, WSPECT allows options to save the data on disk and/or list it and/or use the Vector General Interactive Graphics System to plot the spectral rates versus time.

Routines

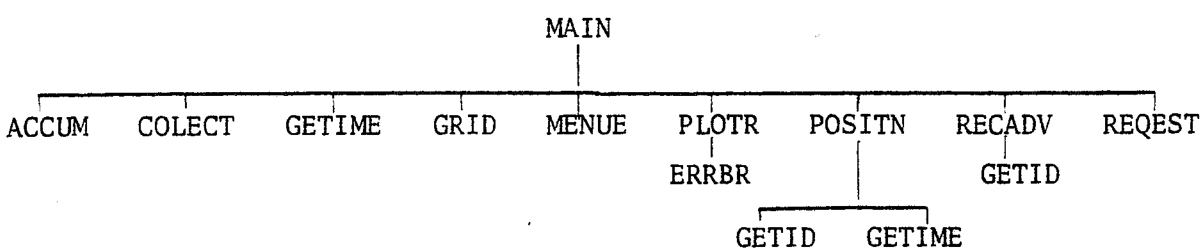
The following is a list of the routines in WSPECT and their function. A block diagram of WSPECT then follows.

NOTE: Initial values are not indicated because all input parameters must be specified by the users.

ROUTINES	FUNCTION(S)
MAIN	1. Obtain user-specified options and intervals.
	2. Space through each record to collect the data by subroutine COLECT.
	3. Act as main controller, coordinating other routines.
ACCUM	1. Average the rates values, convert to counts/sec and store on a disk file.
	2. List the averaged values and times.
RECADV	1. Read in next data record from tape.
POSITN	1. Position tape to correct record of MPI tape according to user-specified start GMT.
GETID	1. Fill an array with the identity bit of each data block in the current record. This bit distinguished Rate 1 from 3, and Rate 2 from 4.
COLECT	1. Sum the GEL, GEH background rates and the PHA spectra over collection time and channel intervals.
GETIME	1. Converts time bytes for each date block into seconds of year.

MENUE	1.	Displays options menue on the Vector General.
	2.	Reads and checks the validity of the options input.
GRID	1.	Draws axes grids, and labels for the Vector General plot.
PLOTR	1.	Plots the rates versus time on the Vector General.
ERRBR	1.	Plots error bars over rates on the Vector General.
REQEST	1.	Interactivity routine which allows error bars to be deleted or options list for a new plot to be displayed.

BLOCK DIAGRAM



Common Areas

A list of each common area and its variables in made. Following this list is a map of which routines use which common areas.

1. /COLCOM/CTIME, OLDTIM, LFREQ, UFREQ, INTRVL, MINMUM, NWINDO, TOTCNT

VARIABLE	TYPE	INITIAL VALUE	MEANING
CTIME	R*8	---	Current data block GMT in seconds of year.
OLDTIM	R*8	---	Previous data block GMT in seconds of year.
LFREQ(4)	I*2	---	Law value for 4 frequency windows.
UFREQ(4)	I*2	---	High value for 4 frequency windows.
INTRVL	R*4	---	Averaging interval for plot or listing.
MINMUM	R*4	---	Initial averaging interval with which the data was collected.
NWINDO	I*2	---	Number of frequency windows there are.
TOTCNT	I*2	---	

2. /GRCOM/START, STOP, VMAX, VMIN, VMING, LIST, LOGLIN, ERRBAR

VARIABLE	TYPE	INITIAL VALUE	MEANING
START	R*8	0.0	Start time in seconds of year of data.
STOP	R*8	0.0	Stop time in seconds of year of data.
VMAX	R*8	10.0	Vertical maximum of plot.
VMIN	R*8	0.0	Vertical minimum of plot.
VMING	R*8	---	Internal vertical minimum, is equal to VMIN if linear scale, or is equal to 10**VMIN if logarithmic scale.
LIST	L*1	N	Y=data listing desired N=no listing desired
LOGLIN	L*1	N	Y=logarithmic scale N=linear scale
ERRBAR	L*1	N	Y=draw error bars N=don't draw error bars

3. /POSCOM/BTIME

VARIABLE	TYPE	INITIAL VALUE	MEANING
BTIME	R*8	---	Beginning time in seconds of year to collect data.

4. /RECOM/LUN, BUFPLC, BUFF, IDBIT

VARIABLE	TYPE	INITIAL VALUE	MEANING
LUN	I*2	1	Logical unit number of tape drive.
BUFPLC	I*2	---	Pointer to BUFF array incremented by data blocks.
BUFF(2564)	L*1	---	Holds record form tape.
IDBIT(16)	L*1	---	The identification bit for each data block in the record. Ø=Rate 1 and 2 1=Rate 3 and 4

Map of Commons

R = Reads into common

W = Writes into common

- = Does not process at all

COMMON	ROUTINES	USE
COLCOM	MAIN	W
	ACCUM	R,W
	COLECT	R,W
	MENUE	R,W
	GRID	R
	PLOTR	R
GRCOM	MAIN	R,W
	ACCUM	R
	COLECT	R
	MENUE	W
	GRID	R
	PLOTR	R
	ERRBR	R
POSCOM	MAIN	W
	POSITN	R
RECOM	MAIN	R,W
	RECADV	R
	POSITN	R,W
	GETID	W
	COLECT	R

Other Variables

The following is a list of all variables not in common, with the exception of loop variables and other minor variables.

A. MAIN

VARIABLE	TYPE	MEANING
ANS	I*2	Used to read in yes & no user options.
BYEAR	I*2	Beginning year, day, & milliseconds of day of data to process.
BDAY	I*2	
BMS	I*4	
EYEAR	I*2	End year, day, & milliseconds of day of data to process.
EDAY	I*2	
EMS	I*4	
CYR	I*2	Current data block's year, day, & milliseconds of day.
CDAY	I*2	
CMS	I*4	
ETIME	R*8	End time in seconds of year
END	I*2	Ø = continue collecting data 1 = done collecting data
EXIST	L*1	T = process existent disk data file F = create new file from tape.
FACTR	R*8	Converted hours to seconds.
GEND	L*1	T = end graphics processing F = continue graphics
IFLG	L*1	T = not processing graphics yet F = graphics in use
NONE	L*1	T = no data processed yet F = at least one data block is processed
PERM	L*1	T = make data file permanent F = delete file when program ends
NAME(15)	L*1	Holds disk file name in ASCII

B. ACCUM

VARIABLE	TYPE	MEANING
CNTS	I*2	Numbers counts in a certain window and time interval.
FACT	R*8	Difference in average GEL and GEH
FIRST	L*1	Flag to indicate first time through the routine.

VARIABLE	TYPE	MEANING
GEH	R*8	The sum of the GEH values in a minimum interval.
GEHNUM	I*2	The number of GEH values in a minimum interval.
GEHSUM	R*8	Sum of GEH in averaging interval.
GEL	R*8	Same as GEH, only for GEL values.
GELNUM	I*2	
GELSUM	R*8	
GHNUM	I*2	The number of GEH values in GEHSUM.
GLNUM	I*2	The number of GEL values in GELNUM.
YEAR	I*2	Times for the current averaging interval.
DAY	I*2	
HR	I*2	
MN	I*2	
SC	R*4	
IFLG	L*1	T = Accum was called for non-graphics F = Accum was called for graphics
LAST	L*1	T = End of data work to accumulate F = More data to accumulate
NEWTIM	R*8	Time associated with record read from disk.
NUMBR	I*2	The number of counts in a window.
SPACE	I*2	Parameter for variable format statement to align listings independently of the number of windows.
ERROR(4)	R*8	Error associated with the average rate in the interval.
RATE(4)	R*8	The average rate in the interval.
SPECT(4)	I*2	The counts in each window as read from disk. It was collected over the minimum interval.

RECADV

VARIABLE	TYPE	MEANING
END	I*2	0 = no end of volume encountered 1 = end of volume encounted
LOST	I*2	Status of the FREAD command
LR	I*2	Length in bytes of the record read.

POSITN

VARIABLE	TYPE	MEANING
CFILE	I*2	Current file number.
CTIME	R*8	Time of first record in current file.
END	I*2	0 = no end of volume 1 = end of volume encountered.
LOST	I*2	Status of the FREAD command.
LR	I*2	Length in bytes of record read.
BUFFR(5128)	L*1	Buffer into which FTIØ reads the tape record.
IUSN(6)	L*1	FITØ array for tape name.

GETID

VARIABLE	TYPE	MEANING
HK	I*2	Housekeeping number of current data block.
ID	I*2	The ID bit found in HK 2.
IDSYNC	I*2	The difference between the S/C clock's position of HKØ and the experiment's housekeeping position of HKØ. This difference allows the bit of S/C which cycles with the HK numbers to be in sync as the HK numbers change.
OSYNC	I*2	IDSYNC of the last record.

C. COLECT

VARIABLE	TYPE	MEANING
GEL	R*8	Sum of GEL values in one interval.
PHA	I*2	Current 12-bit PHA word.
GELNUM	I*2	Number of GEL values in the GEL sum.
ID	I*2	The ID bit bit value derived from the IDBIT array and IDFACT.
PTIME	R*8	The time in seconds of year of the current data block.
HKNUM	I*2	The housekeeping number of the current data block.
IDFACT	I*2	The "polarity" of the ID bit - S/C clock relationship. If IDFACT=1, they are equal, if IDFACT=-1 then they oppose.
RATE	I*4	The GEL or GEH rate as decompressed from the housekeeping.
GEH	R*8	The sum of the GEH rates for an interval.
ID2	I*2	The ID bit in HK2.
GEHNUM	I*2	The number of GEH values in the GEH sum.
NCNT	I*2	Counts to 4 before the IDFACT changes polarity. This allows 4 mismatches between the ID bit and the S/C clock bit.

D. GETIME

DAY	I*2	Day of year of the data sent.
DY(2)	L*1	Equivalenced with DAY to retrieve day of year.
YEAR	I*2	Year of the data sent.
YR(2)	L*1	Equivalenced with YEAR to retrieve day of year.

VARIABLE	TYPE	MEANING
MILLI	I*4	Milliseconds of day of data sent.
MS(4)	L*1	Equivalenced with MILLI to retrieve millisecond of day from data sent.
DATA(8)	L*1	Bytes of timing data. Data (1-2) = day of year Data (3-4) = year Data (5-8) = milliseconds of day
TIME	R*8	Resulting seconds of year for data sent.

E. MENUE

VARIABLE	TYPE	MEANING
BYR BDY BMS	I*2 I*2 I*4	Beginning times of data to plot on the Vector General.
BARRAY(17)	L*1	Character array for above times, displayed on the Vector General.
EYR EDY EMS	I*2 I*2 I*4	Ending times of data to plot on the Vector General.
EARRAY(17)	L*1	Character array for above times, displayed on the Vector General.
NUM1-NUM8	I*2	Number of characters read from the screen for the particular array.
CODE	I*2	Returned from RQATN to indicate which function key terminated wait.
VMING	R*8	Minimum vertical axis value.
VMAXG	R*8	Maximum vertical axis value
GEND	L*1	T = End of graphics processing. F = Graph the data
ARRAY(20)	I*2	Returns various information from RQATN..
AVGY(8)	L*1	Character array for average interval.

VARIABLE	TYPE	MEANING
EBAR(2)	L*1	Y = plot error bars N = no error bars
LISTY(2)	L*1	Y = list data on line printer N = suppress listing
SCALE(2)	L*1	Y = log scale desired N = no log scale (linear)
VMAXY(10)	L*1	Character array for vertical axis maximum.
VMINY(10)	L*1	Character array for vertical axis minimum.

F. GRID

INC	I*2	The Y increment between grid lines for grid.
NUM	R*8	The Y axis label being plotted.
SYEAR	I*2	The start time year.
NUMBR	R*8	Decimal days for labeling time axis.
POS	I*2	X axis coordinate for grid lines.
FACTX	R*8	X axis increment between grid lines for labels.
FACTY	R*8	Y axis increment between grid lines for lavels.
POS1	I*2	Y axis coordinate for grid lines.
NUMDIV	I*2	Number of divisions in Y axis.
NUMG	R*8	Y axis label value.
LIMITS(13)	L*1	Upper and lower limits character array for display on screen.
NUMBY(10)	L*1	Character array for labeling X and Y axis values.
YR(5)	L*1	Character array for labeling the year.

G. PLOTR

VARIABLE	TYPE	MEANING
TIME	R*8	Time value of the point being plotted
ELEM	I*2	Graphics element number of point being plotted.
SPACE	I*2	Variable format parameter indicating the location of the TIME value on disk record.
VALUE	R*8	The rate value, adjusted to be in bounds.
IC	I*2	Attention code for RQATN call.
ARRAY(20)	I*2	Array returned from RQATN call.
ERROR(4)	R*8	The error associated with each window's rate.
NUM(2)	L*1	Character array to plot window number.
PNTS(5)	L*1	Character array specifying the number of points plotted.
RATE(4)	R*8	Each window's rate.

H. ERRBR

ELEM	I*2	The element number of the point.
TIME	R*8	The time value of the point.
LOW	R*8	The low value of the error bar.
ERROR	R*8	The error deviation value of the rate.
RATE	R*8	The rate value of the point.
HIGH	R*8	The high value of the error bar.

I. RQUEST

ARRAY(20)	I*2	Array returned from RQATN.
ID	I*2	The attention code returned from RQATN.
NWINDO	I*2	The number of windows for which points are plotted.

VARIABLE	TYPE	MEANING
DISPLY(2,4)	L*1	This array holds Ø, 1 values which indicate whether a certain window's rates and error bars are displayed on the screen. Ø = rate not displayed 1 = rate is displayed

Algorithm of Data Flow

Following is a step by step view of the flow of the program. Each action described shows how data is collected and analyzed.

STEP	ROUTINE	ACTION
1	MAIN	Read in all input parameters as prompted.
2	POSITN	Position tape to beginning times.
3	MAIN, COLECT	Space through each data block, collecting rates & times by calling COLECT.

(-- Set A follows one path of no graphics, and set B follows graphics path --)

4A	ACCUM	Accumulate the data according to user-input averaging intervals.
5A	MAIN	End, producing a listing if desired.
4B	MENUE	Display graphics menue and read in new parameters.
5B	ACCUM	Accumulate data from disk file according to the averaging interval generating a listing if desired.
6B	GRID	Display the grid on the graphics terminal.

STEP	ROUTINE	ACTION
7B	PLOTR	Plot rates values on the graphics terminal.
8B	ERRBR	Display error bars on the graphics terminal.
9B	RQEST	Option to delete error bars, go to Step 4B, or go to Step 10B, according to VG function keyboard input.
10B	MAIN	End program.

WSPECT

Window Spectra Plots

Operator's Guide

Overview

This program extracts data from an MPI format tape according to specified input parameters and plots these values on the Vector General.

Input Required

(1) MPI format tape *or* a window spectra disk file.
(2) Interaction with several questions requesting input information (see Running WSPECT).

Output Generated

(1) Graph on the Vector General.
(2) Listing of the data.
(3) Window spectrum file.

Running WSPECT

Before running, the operator must know information to complete at least one branch of the input flow chart. This chart shows all possible questions and options that WSPECT will ask for. The following letters correspond to those on the chart:

(A) WANT TO GRAPH OR LIST AN EXISTENT DISK FILE? (Y OR N)

Enter Y if data file is resident on disk.
Enter N if data is to come from tape.

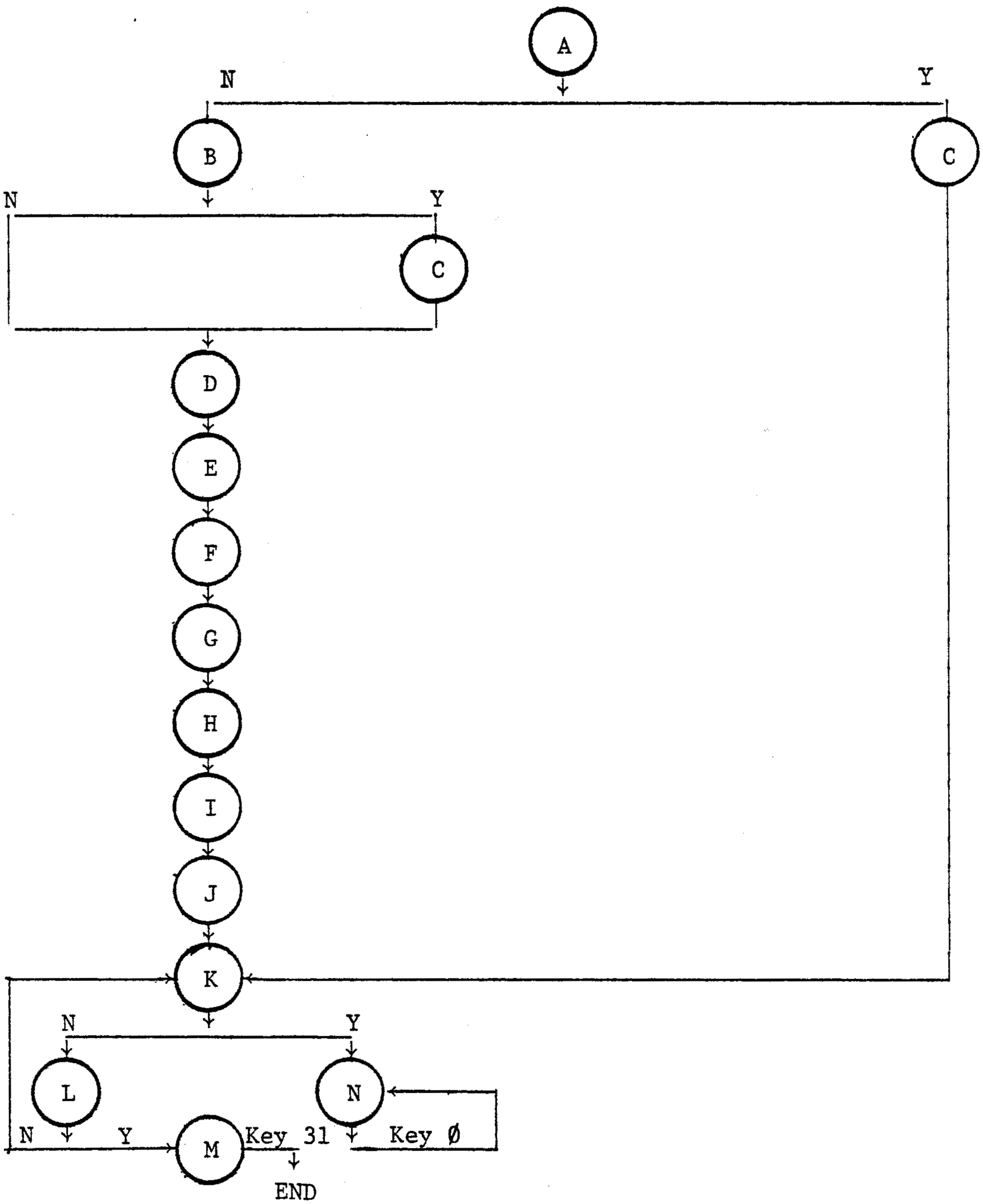
(B) WANT TO MAKE TAPE DATA PERMANENT DISK FILE? (Y OR N)

Enter Y if you want to keep this file for more plots later.
Enter N if the disk file is to be only temporary to this run.

(C) ENTER DATA SET NAME

Enter 'name'.DAT of spectra disk file, where 'name' is nine characters or less.

OPTIONS FLOW CHART



(D) ENTER START TIME. (YEAR, DAY, MS)

Type in year, day and milliseconds of day for which to begin the data collection. The year is 78 for 1978, 79 for 1979, etc. The day is between 1 and 365, or 366 if it is leap year. The ms is between Ø and 86400000.

(E) ENTER STOP TIME. (YEAR, DAY MS)

Same as D, only it specifies the end time of data collection.

(F) ENTER MINIMUM COLLECTION INTERVAL IN HOURS. (DEF = 1)

Enter a real number between Ø.0 and 24.0 for the smallest averaging interval that would be desired for this spectra. However, the smaller the interval, the fewer GEL, GEH values that may be averaged, so this number should not be less than 1.0 for accurate data.

(G) ENTER A FREQUENCY WINDOW RANGE OR -1,-1 IF DONE.
ENTER LOWER LIMIT, UPPER LIMIT (0,4095).

This is the spectral window specification. Up to 4 window ranges are allowed. If less than 4 are desired, enter them, then enter -1,-1 as the window. Each time 2 values are input, the prompt is repeated for 2 more.

(H) Computer requests user to issue mount command. User types CNTL C to get MCR> and then types the following line:

MOU MMχ:/CHA= FOR

where χ is the drive number, Ø or 1.

(I) After computer responds to the MOU command, type the following line:

CON WSPECT

(J) The computer issues a request to dismount the tape. The user types CNTLC to get MCR> and then types the following line:

DMO MMχ:

where χ is as in H.

(K) GRAPH DATA ON SCREEN NOW?

Enter Y to display the data on the Vector General. Make sure no one is using it.
Enter N is no display is desired.

(L) WANT ONLY A LISTING?

Enter Y to get a data listing and end the program. Enter N and Question H is asked again.

(M) ENTER AVERAGING INTERVAL

This is the interval in hours over which the data is to be averaged. It must be greater than or equal to the minimum specified in F or, if data set already existed, it must be greater than or equal to the minimum specified when that file was created.

(N) <u>Vector General Options List</u>

ENTER INPUT

1.	START TIME (Y,D,MS)	00 000 000000000
2.	END TIME (Y,D,MS)	00 000 000000000
3.	AVERAGING INTERVAL (HOURS)	0.000
4.	DATA LISTING? (Y OR N)	N
5.	VERTICAL MAX VALUE	000010.00
6.	VERTICAL MIN VALUE	000000.00
7.	LOG SCALE? (Y OR N)	N
8.	ERROR BARS? (Y OR N)	N

1. Same as D.
2. Same as E.
3. Same as J.
4. This allows a list of the data to be plotted.
5. Enter max value for rates scale.
6. Enter min value for rates scale.
7. This allows semi-log scale (Y) or linear scale (N).
8. Y=draw error bars.
   N=no error bars.

Pressing Key Ø executes options. Key 31 ends program.

Logging the Disk File

If a permanent disk file has been created, it is logged into WSPECT.LOG file on disk by WSPECT automatically.

If a disk file has been deleted, then delete the entry in the log using EDI.

Error Handling

All user input errors are trapped by the program and a request for correction is issued. However, there are several errors which terminate the program. These follow:

(1) Between G and H FTIO aborts, as indicated by a system message. This means both tape drives are in use. If there are free drives, then the previous user failed to properly dismount the drive. Issue the DMO command as in J. Then type:

RUN [2,76]CLEF

Now WSPECT may be run again.

(2) Between G and H the program just waits--no message is issued. The user must type CNTL C to get MCR, then types:

RUN [2,76]CLEF (esc key)

(3) NO DATA WITHIN TIME RANGE.
Check to make sure that the start times are correct and that the tape contains the desired time span.

STANDARD PRODUCTION GUIDE

The following three sections will describe the processing of a data tape in each of the three tape systems--quick-look, production and MPI. Any requests beyond those listed will be submitted on the request forms for ISEE and GRB.

QUICK-LOOK SYSTEM:

Input Tape.

7-track, 556 BPI, no-label tape we receive from MSOCC which must be returned after duplication. This tape has GRB data interspersed with other experiments data.

Preliminary Functions Performed.

1. Copy 7-track to 9-track tape.
2. Run 9-track tape through QKSTRP to extract the GRB data.

Data Analysis.

Listing of data values and housekeeping on the PDP-11/70.

Capabilities of System by Request Only.

1. Merge spectral file into accumulation file.
2. Display any spectral files.
3. Time history plots if there is such data.
4. Interactive graphics work.

Additional Functions.

A tape log will be kept containing times and tapes characteristics of each input tape. Also, another table is kept which indicates which functions have been performed on the tape, and lists spectral file names.

PRODUCTION TAPE SYSTEM:

Input Tape.

9-track, 1600 BPI tape which must be returned to University of Maryland. The GRB data is interspersed with data from other experiments. We will contact U. of Maryland once a week to obtain the latest tapes.

Preliminary Functions Performed

1. Duplicate tape.
2. Run PROSTRP to extract GRB data.

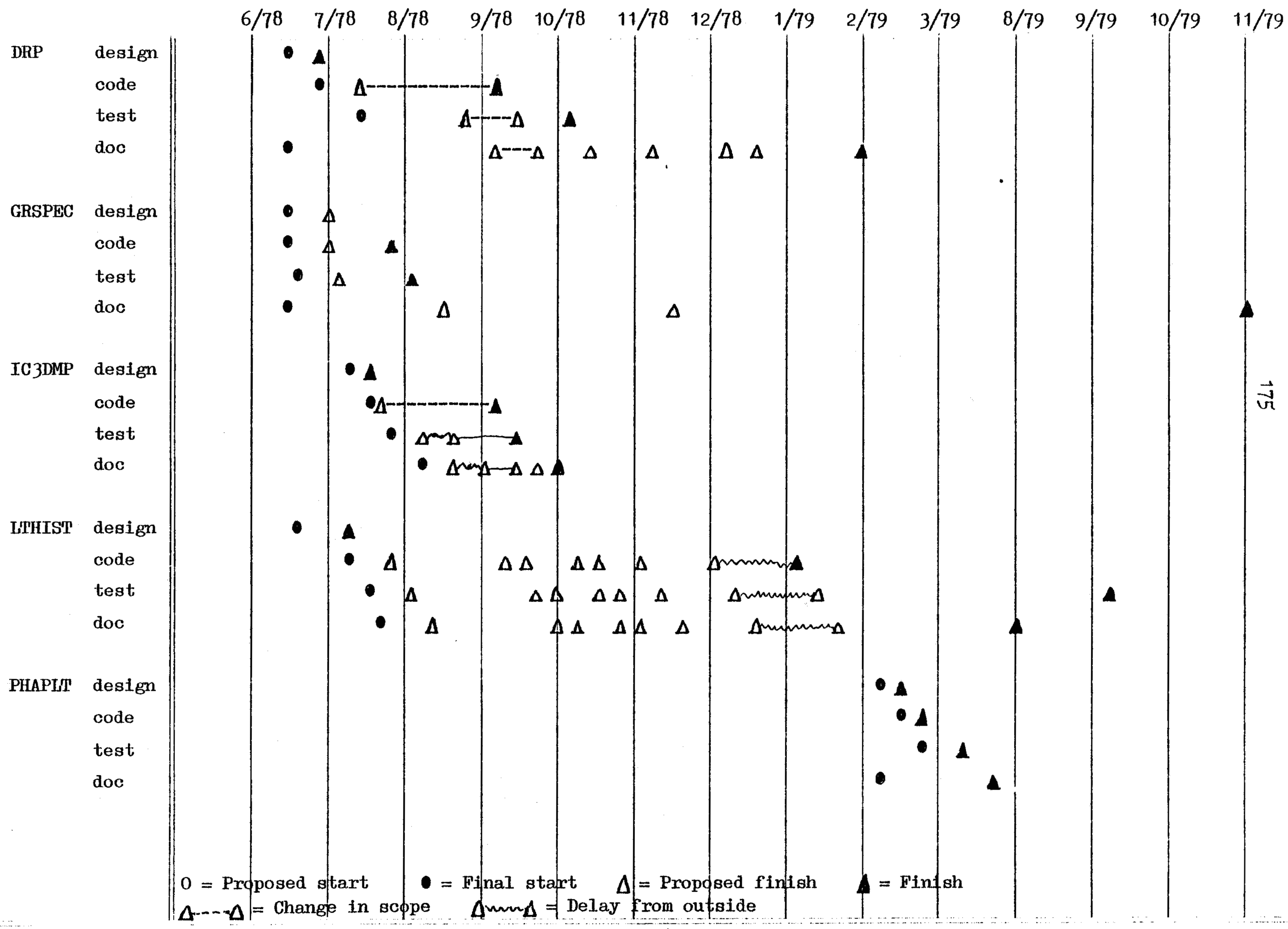
Data Analysis

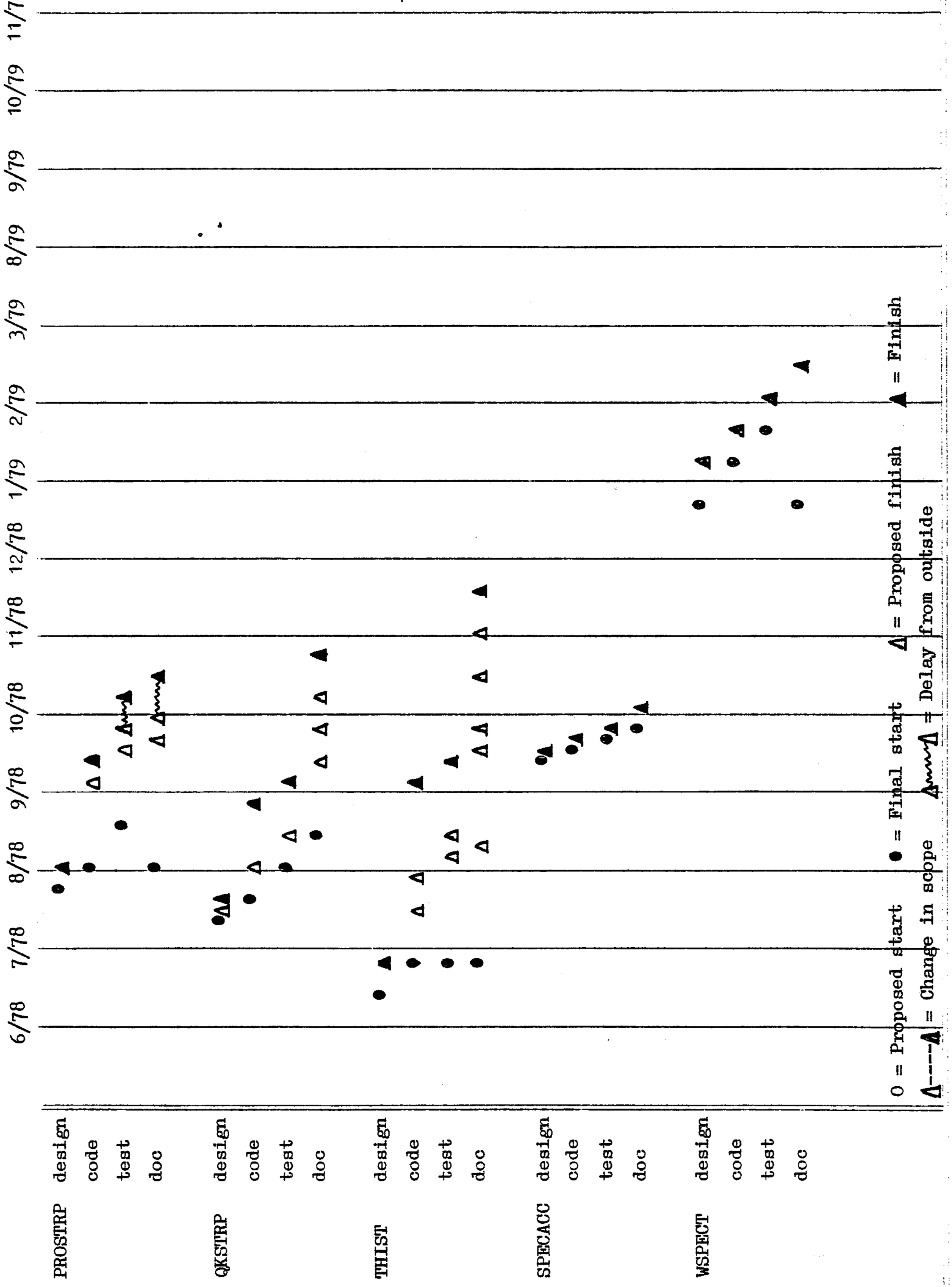
1. Full listing of data & housekeeping on IBM 360.
2. Produce daily spectral files on PDP 11/70.
3. Use Vector General GRSPEC with following parameters:
   a. Log scale
   b. Vertical Axis max such that max data value is $\approx$1 1/2" max graph value.
   c. Channels 0-1023, compression = 1.
   d. Channels 0-4096, compression = 4 and 16 (bkgnd only if comp=16).
   e. Listing of data values for above plots.
4. Data listing of data blocks which have a changed housekeeping parameter, from the previous data block. Print "brief" version.
5. Long term time history of rates, temps, voltages, with following parameters:
   a. Scale=log
   b. Vertical limits are 10→10,000
   c. Hourly averages plot
   d. Listing of data plotted.
6. Time history plots if there was burst data with following parameters:
   a. Log scale
   b. Scale=10→10,000
   c. No error bars
   d. 12 sec/inch, 240 sec/page
7. Pha histogram if burst data. (Same as Time History plots.)

Capabilities of System by Request Only

1. Any spectra collected, limited to units of records.
2. Listing from the timing program.
3. Any interactive graphics work.
4. Additional plots with other parameters.
5. Spectral accumulations & plots.

ISEE3 DATA PROCESSING SYSTEM





6/78
7/78
8/78
9/78
10/78
11/78
12/78
1/79
2/79
3/79
8/79
9/79
10/79
11/79
PROSTRP design
code
test
doc
QKSTRP design
code
test
doc
THIST design
code
test
doc
SPECACC design
code
test
doc
WSPECT design
code
test
doc
O = Proposed start
● = Final start
Δ = Proposed finish
▲ = Finish
Δ---Δ = Change in scope
Δ~~~Δ = Delay from outside

Location of Programs

The programs are stored on both the IBM 360 and the PDP 11/70 as follows:

IBM 360

The programs are under USERID 'SEAMP', and are named according to their format: Source code is in ISEE3.SRCE, JCL code is in ISEE3.JCL, and the load modules are in ISEE3.LOAD.

The partitions are as follows:

Program	Partition Name
DRP	(DRP)
LTHIST	(LTHIST)
MERGER	(MERGER)
PHAPLT	(PHAPLT)
PHIST	(PHIST)
PROSTRP	(PROSTRP)
QKSTRP	(QKSTRP)
THIST	(THIST)

For example, the source to DRP would be found in 'SEAMP.ISEE3.SRCE(DRP)'

PDP 11/70

The programs are under UIC of [150,102] and consist of source, command, and task files as follows:

.FTN = fortran source
.CMD = command file
.TSK = task file

Program	File Name
GRSPEC	GRSPEC.XXX
IC3DMP	IC3DMP.XXX
SPECACC	SPECACC.XXX
WSPECT	1.WSPECT.XXX 2.WSPECTGR.XXX - Source and object only (They are linked in the command file)

LTHIST Run

AVGINT = 24.0

ILIST = 1

ILOG = 1

IDV = 0

ISUB = 1

~~Plot~~ List from 1978 DAY 230

to 1979 DAY 100