DESCRIPTION OF TEC ANALYSIS UTITILITY ROUTINES

-- J. Paradiso [16-Dec.-82]
[Revised 20-April-83]

The library "HSTUFF.FOR" contains three types of routines; "Analysis" routines (called by the user to analyze TEC data), "Utitity" routines (called by the analysis routines, but not usually by the user), and "Histogram" routines (which process HBOOK histograms, and can be used when needed).

These routines were developed to handle the HP data tapes taken during the August, 1982 CERN tests, but can easily be adapted to other data formats. Most of the system dependent software resides in the event reading routines (see READEV.DOC), which strip data off of tape and unpack it. This package requires these event reading routines, and the histogram programs require HBOOK.

The routines were revised during March/April of 1982 to make them more capable and flexible; this document has been updated to incorporate the changes.

I) Organization and Common Blocks

NOTE: The second index in the 2-dim. arrays below refers to the "wire configuration number" of anode/pwd pairs. For the CERN tests it had dimension 2 (since we used only two configurations), but in tests with larger numbers of wires it must be increased in the COMMON statements in the subroutines.

There are five primary common blocks used in these routines. They are discussed below.....

```
COMMON /TEC/ IRN,IEVT,IFADC(256,4),ITDC(4,3),IPU
```

This common block contains the event data. It is filled by the "GETEVT" sequence in the event reading software. The contents are:

IRN: Run Number
IEVT: Event Number
IFADC(256,4): Flash ADC data. In these tests we had a 256-word deep memory, and 4 FADC channels, thus the dimensions.
ITDC(4,3): TDC data. This is not used with HSTUFF. See READEV.DOC.
IPU: Pattern unit word. Not used here.

This common block is effictively the major "hardwired" portion of the program; for more ADC's of differing length, the dimensions of IFADC (and/or RFADC) must be changed in these routines. The subroutine package as it now exists uses this common block for the CERN data.

```
COMMON /WIRES/ MAP(4,2)
```

This common block groups the FADC array into wire configurations. The entire HSTUFF package uses it to access FADC data. The array contains the following:

MAP(1,I) = Index of FADC channel containing anode signal for wire configuration #I.
MAP(2,I) = Index of FADC channel containing pickup-difference signal for wire configuration #I.

MAP(3,I) = Assigned number (1-23) for wire configuration #I. This corresponds to the relative position of the wire configuration in centimeters.
MAP(4,I) = Depth of FADC memory (ie. 256 for the CERN tests, 1024 for the Siegen FADC's, etc.).

Since there were only two wire configurations used in the CERN tests, the index "I" ranges only 1-2. The mainline program written by the user sets the MAP array by calling the routine "MAKMAP" (see below) at the beginning of the program. The program then refers to the data by the configuration number "I". To get the anode signal for configuration I, one does: IFADC(J,MAP(1,I)) (looking at FADC cell #J); correspondingly the pickup-difference can be located: IFADC(J,MAP(2,I)) (if MAP(2,I) is zero, there is no PWD signal associated with configuration#I). The wire position in centimeters is MAP(3,I). The total number of cells which one expects to find is given by MAP(4,I).

This mapping scheme was introduced to avoid major program changes when processing runs where different signals from different wires were plugged into different FADC channels. By only changing the MAPADC file, the indices are re-specified, and the program automatically takes care of locating the proper data.

COMMON /DRVEL/ VDR1,VDR2,ADCPRM(2,2)

This block contains a few vital run parameters, and is set by MAKMAP.

VDR1: Drift velocity in the drift region (cm/usec.).
VDR2: Drift velocity in the detection region (cm/usec).
ADCPRM(1,I): FADC clock period for configuration# I (nsec.).
ADCPRM(2,I): FADC Pretrigger value for configuration# I (nsec).

COMMON /BIASES/ BIAS(2,2),SDEVS(2,2),INIT,NT,NDO

This block contains the average DC parameters of the FADC data calculated over several events. It is set by the routines "GTBIAS" and "PRMBLE".

BIAS(1,I): DC bias level of anode in configuration# I
BIAS(2,I): DC bias level of PWD in configuration# I
SDEVS(1,I): Noise of anode in configuration# I
SDEVS(2,I): Noise of PWD in configuration# I
INIT: Initialization flag. If valid "global" bias levels were calculated by GTBIAS, INIT=1.
NT: Total # of events scanned by GTBIAS.
NDO: Maximum number of FADC cells to scan in the CTROID routine before and after the beginning and end parameters found by PLUCK. This parameter is set in routine "PRMBLE".

COMMON /ASTRT/ NO,NCH

NO: Starting index of ADC histograms. Set in BDADC.
NCH: Total number of wire configurations used. This is the range of parameter "I" in the 2-dim. arrays above, thus for the CERN data NCH is 2. Set in MAKMAP.

NOTE: These common blocks are "hardwired" in the subroutine package for the CERN data; if other data formats are used, the dimensioning of the arrays must be changed accordingly.

II) Description of Analysis routines:

CALL CRSHFT

This routine corrects the 10-cell shift in the first FADC channel. It is called after GETEVT. This is a "fix" routine used to compensate for a hardware defect in the LUND/BNL FADCs used in the Aug. '82 tests, and hopefully won't be necessary in the future.

CALL PRMBLE

Asks at the TTY for tape unit and density; these parameters are used to open an I/O channel via "STRTEC" (See event reading desc.). Also asks for the # of files to skip on a tape. This routine is quite useful; when called at the beginning of the mainline, all tape manipulation is completed internally. Needless to say, it's quite system dependent. PRMBLE also sets a few initialization parameters.
--- Calls FNDFIL, STRTEC, PTBIAS, PTROID

CALL MAKMAP(IRN)

This routine sets the "MAP" array in COMMON /WIRES/ and COMMON /DRVEL/, along with a few other initialization parameters. IRN is the assigned number of the run to be analyzed, and is used as a key into a sequential ASCII data file (the user is prompted at the TTY to enter the file name at runtime). The file currently has the format:

```
  [Previous runs]
       .
       .
IRN,NCH,VDR1,VDR2
MAP(1,1),MAP(2,1),MAP(3,1),MAP(4,1),ADCPRM(1,1),ADCPRM(2,1)
MAP(1,2),MAP(2,2),MAP(3,2),MAP(4,2),ADCPRM(1,2),ADCPRM(2,2)
       .
       .
 [NCH total subentries]
       .
 [Further runs]
       .
       .
     [EOF]
```

IRN is the assigned run number, NCH it the total number of wire configurations in this run, and see COMMON /WIRES/ and COMMON /DRVEL/ for the others. The file is read via "*" format.

If a configuration is without a PWD channel, set its PWD parameters to zero in the data file.

This routine is called only once at the beginning of the mainline.
--- Calls PRTPLK

CALL GTBIAS(NEV,ISCH)

This routine scans over "NEV" events to calculate "global" DC biases for the four FADC channels. The parameters are written into the block: COMMON /BIASES/. The anode signal is scanned for peaks; the anode and pickup cells are summed starting at cell# "ISCH", and ending 5 cells before the beginning of the first peak. This is done independently for both configurations.

GTBIAS is generally called at the beginning of the mainline (after the routines discussed above are involked). See "DOBIAS" for more info. A "GOBACK" call is performed at the conclusion of GTBIAS, thus the input dataset is re-positioned at the beginning-of-file.

---- Calls CRSHFT,ADYDX,GETEVT,GOBACK ----

CALL PLUCK(L1,L2,IWR,NBEG,IGMX,IBSTR,IESTR,NPKSTR,NPEAKS,IBPEK,IEPEK,
. NPLPEK,CTRPEK,ISHI,IRNOFF)

This routine finds all significant peaks in an anode signal (plucks peaks out of an event like plucking feathers from a chicken, thus its name). The anode signal is differentiated, and the signal "jumps" are traced to seperate the peaks. A conglameration of peaks occuring within a maximum spacing of "IGMX" cells are grouped into a so-called "structure". If the routine runs off of the edge of the FADC data looking for the end of a structure, a "RUNOFF" condition is flagged. This indicates either peaks grouped close to the end of the FADC data, or an error in PLUCK (where it found a jump starting a peak, but couldn't find a reverse jump to end it). The parameters describing structures have the suffix "STR" above, while those describing peaks bear "PEK". In more detail......

Input parameters:

L1: Dimension of peak arrays.
L2: Dimension of structure arrays.
IWR: Wire configuration # to pluck. The anode is found via MAP.
NBEG: The FADC cell in which to start the peak scan.
IGMX: The minimum spacing between peaks required to define "structures'

Output parameters:

NSTR: Number of structures found.
IBSTR(I): Beginning of structure #I (in FADC cells).
IESTR(I): End of structure #I.
NPKSTR(I): Number of peaks contained in structure #I.
----- I runs 1 thru NSTR (If NSTR=0; no data) -----
NPEAKS: Total number of peaks found
IBPEK(J): Beginning of peak #J (in FADC cells).
IEPEK(J): End of peak #J
NPLPEK(J): Number of "plateaus" (or flat areas) in peak #J.
CTRPEK(J): Rough maximum of peak (where slope goes through zero; not the centroid).
------ J runs 1 thru NPEAKS. ------
ISHI: "Start-high" flag. If the ADC data started over the bias level ISHI=10*N+1, otherwise ISHI=10*N, where N=# of negative peaks found. If N>1, a problem is assumed, and IRNOFF is set to 1.

IRNOFF: "Runoff" flag. Set if PLUCK ran off the end-of-data looking for the conclusion to the current structure (see above). It is also set to 1 via the negative peak condition. IRNOFF=2 means there were too many peaks found for the input arrays (use larger dimensions) and IRNOFF=4 if there were too many structures found for the structure arrays (" "). If IRNOFF=6, both peak and structure arrays were too small. The program "RECVER" tries several theraputic tricks to recover from IRNOFF=1.

The structure and peak arrays are arranged in order of increasing FADC cell#; earlier structures and peaks are first, later ones last. This enables one to index the peak data for a particular structure. The peak parameters for the first structure have the indices 1 thru NPKSTR(1). The peak parameters for the second structure have the indices NPKSTR(1)+1 thru NPKSTR(1)+NPKSTR(2). The peak parameters for following structures can be extracted in like fashion.

This routine is called after GETEVT. It is called once for each configuration.

---- Calls ADYDX ----

```
CALL SETPLK(ITH1,ITH2,IAMX,IBC,IMXGAP,NEGPOS)
```

Allows the user to set the internal parameters used in PLUCK in order to avoid program modification for different data behavior. ITH2 is the threshold used (on the differentiated signal!!) while scanning for a peak, and ITH1 is the threshold used (" ") during a peak to look for its conclusion. The default values in PLUCK are ITH1=1, ITH2=3 for the CERN data (6-bit ADC's). For 8-bit ADC's, these can be scaled up by approx. 4 (assuming similar noise conditions). In general a ratio of ITH2/ITH1 = 3 (roughly) is found to be sufficent.

IAMAX is the minimum value of the FADC signal over the global bias level required at the start of the ADC data to flag the "STRT-HI" condition. It is set to 7 by default for the CERN data.

If a peak has a "plateau" region longer than IMXGAP cells, the peak is cancelled provided the original jump is smaller than BCUT. IMXGAP=25 and BCUT=10 by default for the CERN data.

NEGPOS is a flag which dictates the polarity of signals which PLUCK is to scan for. NEGPOS=1 implies positive-going peaks, while NEGPOS=-1 implies negative-going peaks. NEGPOS=1 by default.

```
CALL GETPLK(ITH1,ITH2,IAMX,IBC,IMXGAP,NEGPOS)
```

Fetches internal PLUCK parameters. See SETPLK for details. By first calling GETPLK, then SETPLK, one may modify only certain parameters and leave the other defaults intact.

```
CALL PRTPLK
```

Prints out current PLUCK parameters. Called by MAKMAP.

```
CALL RECVER(L1,L2,IWR,NBEG,IGMX,NSTR,IBSTR,IESTR,NPKSTR,NPEAKS,IBPEK,
.   IEPEK,NPLPEK,CTRPEK,ISHI,IRNOFF,ICD1,ICD2)
```

Recovers from PLUCK IRNOFF=1 condition. Arguments are the same as in the original call to PLUCK, except for the last 2, which hold status codes.

This routine first calls ENDMEM to see if the runoff was due to actual activity at the end of the FADC array. If so, IRNOFF is set to zero, and ICD1 returns zero.

If ENDMEM didn't solve the problem, REVPLK is called, which reverses the FADC array and calls PLUCK backwards. If this worked, IRNOFF is set to zero and ICD1 returns 10.

If REVPLK didn't work, FLTPLK is called. This routine filters the FADC array with succesively larger time constants before calling PLUCK (up to 10 tries are attempted). It returns ICD1=20+TC-1, where TC is the last time constant used (ranges 1-10). If it worked, IRNOFF is set to zero (it always works.... but one can loose peaks).

ENDMEM is called after every step in case something is actually happening at the end of FADC data. ICD2 is the resulting ENDMEM code. If ICD2=0, ENDMEM was not needed. If ICD2=1, ENDMEM found activity at the end-of-data, but no peaks were cut-off. If ICD2=2, ENDMEM found a peak which was cut-off at the end-of-data; in this case the peak is erased from the peak and structure arrays (if it started a structure, it is also erased).

--- Calls ENDMEM, REVPLK, FLTPLK

CALL DOBIAS(NC,IBEG,NSTR,IBSTR,IESTR,BIAS,RNOISE,NT,NG)

This routine manages the bias determination proceedure for the event. The "structures" of activity found by PLUCK are cut out of the signal (50 cells are ignored after a structure to lower tail sensitivity and 5 cells are ignored before the start of a structure).
The remaining cells are summed to find the event bias. The biases are calculated for both the anode and pickup-difference of a configuration.
A minimum of 20 ADC cells are required to be included in the bias calculation for validity. If this condition is not met, the routine uses the "global" bias calculated over several events by "GTBIAS".
The resulting biases are subtracted from the FADC arrays; following the subtraction, the FADC arrays (formerly integers) become floating-point.
If one is using a 16-bit computer, one must beware of this feature, since flotaing constants are double-words, whereas integers are single-words.
The NT argument controls the mode of bias subtraction (see below). Default parameters reside in data statements in routine ABIAS.
In detail.....

Input parameters:

NC: Configuration number. Data is located via MAP.
IBEG: ADC cell# in which to start bias scan.
NSTR: No. of structures (output from PLUCK).
IBSTR: Array containing beg. of structures (" ").
IESTR: Array containing end of structures (" ").
NT: See below....

Output parameters:

BIAS(I): Signal biases
RNOISE(I): Signal noise
---- I=1 Anode, I=2 Pickup. Biases are automatically subtracted. ----
NT: No. of ADC cells used in calculation (on output).
If the NT argument is negative on input, the program only subtracts the baseline under the defined structures (+/- NDO cells), and does not calculate the noise value [used for quick operation]. If NT is positive on input, the baseline is subtracted throughout the ADC data. One must remember that NT is written into on output, thus must be redefined each time the routine is called. DO NOT put a fixed constant for NT directly into the call-line.

NG: Action flag. NG=0 means good event bias found and used.
NG=1 means event bias not calculated, global bias used.
NG=-1 means event bias not calculated and global bias was not specified by calling GTBIAS. Nothing done.

DOBIAS is called after calling PLUCK.
---- Calls ABIAS ----

CALL PTBIAS

Prints out current bias parameters. (Called by PRMBLE).

CALL CTROID(L1,NC,NPW,NPEAKS,IBPEK,IEPEK,CTRDS,SUMS,SIGS,RMAXS)

Calculates centroids, integrals, sigmas, and maxima of peaks or structures. This routine first calculates the peak parameters for the boundries as they are input from PLUCK. It then takes the limits +/- 3*SIGMA from the resulting centroid and calculates again. If the argument NC>0, the routine repeats the calculation in this way until the old and new sigmas agree within pre-set limits, the original peak limits are extended over more than NDO cells (see COMMON /BIASES/), or the current peak limits approach within 3 cells of an adjacent peak (if these limit conditions occur at any time, the program stops calculating and returns). If NPW=2, the PWD parameters are calculated using the limits employed by the last anode calculation. The calculations are done in routine GTROID.

Input Parameters:

L1: Primary dimension of output arrays; ie. the program assumes the form "CTRDS(L1,NPW)".
NC: IABS(NC)=configuration#. Data is located via MAP. If NC<0, the program does not perform iterations on sigma (see above).
NPW: =1 means calculate only for anode, =2 means calculate for both anode and PWD.
NPEAKS: Number of peaks to calculate (output from PLUCK)
IBPEK: Array containing beginning of peaks (" ").
IEPEK: Array containing end of peaks (" ").
---- NSTR, IBSTR, and IESTR can also be used above to look at structures ----

Output Parameters:

CTRDS: Array (dimensioned L1,NPW) containing peak centroids.
SUMS: Array (" ") containing peak integrals.
SIGS: Array (" ") containing peak sigmas.
RMAXS: Array (" ") containing peak maxima.
NOTE: The anode parameters are in CTRDS(N,1), and the PWD parameters (if calculated) are in CTRDS(N,2).

CTROID is called after DOBIAS. It must not be called before, because it assumes that the FADC data is in floating point format (see above).
--- Calls GTROID

CALL PTROID

Prints out CTROID parameters which reside in data statements. Called by PRMBLE.

CALL DMPLK(IWR,NSEPST,IPFG)

This routine performs the entire PLUCK, DOBIAS, CTROID, etc. scenerio for wire configuration# IWR (assumes the data is in COMMON /TEC/, the map is already defined via MAKMAP, etc.). The detailed information output from each stage of the process is printed (#structures + #peaks, begin/end, centroids, sigmas, status and recover flags, etc..). NSEPST is the structure seperation constant for PLUCK ("IGMX"), and IPFG is a flag which (when set to 'Y') books and fills 4 histograms with the ADC signal, then prints 'em with HISTDO (if IPFG is not 'Y', this is inhibited).

With a routine such as this (omitting the PRINT statements and options), one can easily build a dedicated "spacepoint" subroutine to handle all of the PLUCK, DOBIAS, etc. and give only centroids in x and y as results. I haven't yet gone quite so far, since the hardware is still in a test phase, and I want ultimate flexibility from the software. However routine DMPLK shows one the route to go for mainline simplicity..........

III) The Standard Mainline Structure..........

Below I outline the structure of a typical mainline program to process an event using these routines. (We assume an analysis of the Aug. '82 CERN data).

```
        [Dimension arrays (see above desc.)]
        DIMENSION IBSTR(20),IESTR(20),NPKSTR(20),IBPEK(50),IEPEK(50),NPLPEK(50),
      . CTRPEK(50),BIAS(2),RNOISE(2),CTRAS(50,2),SUMA(50,2),SIGA(50,2),
      . RMAXA(50,2)

        COMMON /TEC/ IRN,IEV,IFADC(256,4),ITDC(4,3),IPU

        [Basic initialization]

        CALL PRMBLE        [Open input unit, and skip files]

        CALL GETEVT(IE)         [Get the run#]
        CALL MAKMAP(IRN)        [set MAP array. ]

        CALL GTBIAS(300,7)    [Get global bias over 300 events; start @ cell# 7]

        [Event Loop.....]

        CALL GETEVT(IE) [Get the event in COMMON /TEC/]

        [Check IE for EOF]

        CALL CRSHFT     [Correct for shift in FADC #1]

        DO 100 IC = 1,2   [Loop over wire configurations; I= config. no.]

        CALL PLUCK(50,20,IC,5,3,NSTR,IBSTR,IESTR,NPKSTR,NPEAKS,IBPEK,IEPEK,
      . NPLPEK,CTRPEK,ISHI,IRNOFF)
                               [Find peaks in config. #IC. Start at ADC cell# 5
                                  and require 3-cell inter-structure spacing.]
              [Check on start-hi, array overflows, etc.]


        IF (IRNOFF.EQ.1) CALL RECVER(50,20,IC,5,3,NSTR,IBSTR,IESTR,NPKSTR,
      . NPEAKS,IBPEK,IEPEK,NPLPEK,CTRPEK,ISHI,IRNOFF,ICD1,ICD2)
                            [Recover from IRNOFF=1]

        [Skip if NSTR=0, Check on Runoff, start-hi conditions]

        NT=-1               [Specify fast operation]
        CALL DOBIAS(IC,5,NSTR,IBSTR,IESTR,BIAS,RNOISE,NT,NG)
        [Does bias calculation and subtraction for configuration #IC. Scan
         starts at cell #5.  Check NG >=0. ]

        CALL CTROID(50,IC,2,NPEAKS,IBPEK,IEPEK,CTRAS,SUMA,SIGA,RMAXA)
          [Calculate centroids and other params. for all peaks in anode and
            pickup difference.]


        [Process information.......]

100     CONTINUE

        [More Processing.........]

        [Get next event......]
```

IV) Description of Utility Routines

These routines are used by the analysis package. They are not necessarily employed by the user.

CALL FNDFIL

Asks at the TTY the no. of files to skip, then goes and does it. Used in PRMBLE.
---- Calls SKFTAP

CALL ADYDX(IN,IOUT,N)

Differentiates input array (IN) and places result in IOUT. IN and IOUT are assumed to be dimensioned N. Used by PLUCK and GTBIAS.

CALL AFLTR(IN,IOUT,TC,N)

Filters input array IN using time constant TC (runs 0. to 1.), and places results in array IOUT. Filters twice to avoid phase shifts. IN and IOUT are assumed to be dimensioned by N.

CALL ABIAS(IFNO,NTOT,IBEG,NSTR,IBSTR,IESTR,BIAS,RNOISE,NT,NG)

Calculates the event bias for FADC channel #IFNO (MAP array is used for this). NTOT is the total # of cells in the FADC channel IFNO. The parameters IBEG, NSTR, IBSTR, IESTR are all input from DOBIAS, and the output parameters BIAS, RNOISE (here not arrays), NT, and NG are output through DOBIAS, so see the DOBIAS description above for more details.

CALL GTROID(I1,I2,IFNO,CTR,SUM,SIG,RMAX)

Calculates centroid, sum, sigma, and maximum swing between cells I1 and I2 for FADC channel# IFNO. Used by CTROID.

CALL ENDMEM(NCELLS,NSTR,IESTR,NPKSTR,NPEAKS,IBPEK,IEPEK,IFLG)

Checks for FADC activity at the end-of-data. NCELLS is the total# of cells in the FADC channel used, IFLG is the status output, and all other parameters are output from PLUCK. See RECVER for more detail.....

CALL REVPLK(L1,L2,IWR,NBEG,IGMX,NSTR,IBSTR,IESTR,NPKSTR,NPEAKS,IBPEK,
. IEPEK,NPLPEK,CTRPEK,ISHI,IRNOFF,ICDE)

Reverses the FADC array and calls PLUCK backwards. Parameters are the same as in PLUCK; if ICDE is nonzero, it holds ENDMEM codes. The original FADC data (in the common block) is left untouched, and the order of the peaks, structures, etc. is increasing with FADC cell#, as normally output from PLUCK.
--- Calls INVRTA, ENDMEM, PLUCK

CALL FLTPLK(L1,L2,IWR,NBEG,IGMX,NSTR,IBSTR,IESTR,NPKSTR,NPEAKS,IBPEK,
. IEPEK,NPLPEK,CTRPEK,ISHI,IRNOFF,ICD1,ICD2)

Filters the FADC array with succesively larger time-constants before calling PLUCK. ICDE holds the last time-constant used (1-10) and if IDC2 is non-zero, it holds ENDMEM codes. The FADC data itself is left intact.
--- Calls AFLTR, ENDMEM, REVPLK, PLUCK

CALL INVRTA(INOUT,N)

Inverts the order of array INOUT. INOUT is assumed to be dimensioned by N (ie. the first N elements are inverted). Used in REVPLK.

V) Description of Histogram Routines

These routines operate on HBOOK histograms.

CALL HFLTR(I1,I2,ALPHA)

Filters the histogram I1 and stores the result in I2. ALPHA is the time constant (goes 0. - 1.; small ALPHA means light filtering, large ALPHA gives heavy filtering). Can be used for any 1-dim. histogram. Filters both ways to remove transients.

CALL HDYDX(I1,I2)

Puts the derivative (vs. bin coords.) of histogram I1 into histogram I2. The output histogram (I2) has dimension N(I1)-1, and the bin edges are displaced by 1/2 of a bin-width (ie. the slopes are calculated only at the bin-edges of the input histogram). Can be used for any 1-dim. histogram. Analogous to ADYDX (see above).

CALL HDMPIT(I)

Dumps histogram #I nicely on the line printer (no error bars). Can be used for any 1-dim. histogram.

CALL RUNTIT(I)

If I=1, creates a global histogram title with the current run and event numbers. If I=0, creates a global histogram title with the current run number only.

CALL DMPSTA

Prints the # of entries, mean, and sigma of all booked histograms. (Does the same for 5 slices of all 2-dim. histograms).

---- The following routines deal with a set of histograms into which the FADC signals are placed. -----

CALL BKADC(I)

Books histograms for all FADC channels starting at histo. index #I.

CALL PCKADC(I)

Packs FADC arrays into the histograms booked above (I is the same).

CALL PKADCS(I)

Same as above, but corrects FADC channel #1 for the 10-cell shift before packing the data into the histos. Does not affect the original FADC arrays. Dedicated to the CERN Aug. '82 run. See CRSHFT (above).

CALL SETSCL(I,RMIA,RMAA,RMIP,RMAP)

Fixes the scale on all FADC histograms (I is the index offset, as earlier).
RMIA and RMIP are the scale minima for anode and pickup respectively, and RMAA and RMAP are the scale maxima for anode and pickup respectively.
Uses the MAP array to find anode and pickup histos, so MAKMAP should be called first!!

CALL SCRAP(I)

Deletes the histograms booked by BKADC (starting at #I).

EVENT READING ROUTINES -- J.PARADISO, 29-SEPT.-82

THESE ROUTINES ARE STORED IN "TAPLIB". DUMMY AND SUBSTITUTE ROUTINES ARE IN "DSKLIB". THE TAPLIB ROUTINES ACTUALLY READ FROM TAPE, AND THE DSKLIB ROUTINES READ DISK FILES CREATED BY THE "CPYRUN" PROGRAM. THE CALLS TO BOTH PACKAGES ARE IDENTICAL; THE PROGRAM GOES TO DISK FOR DATA WITH DSKLIB (YOU MUST ENTER THE INPUT FILE SPEC AT THE PROMPT; THE DEFAULT UID IS [1161,1], AND THE PROGRAM GOES TO MAG. TAPE WITH TAPLIB.

THE "CPYRUN" PROGRAM IS EXTREMELY EASY TO USE; ALL REQUIRED INFORMATION IS REQUESTED PLAINLY, AND THE OUTPUT FILESPEC IS ENTERED AS WITH THE DSKLIB PACKAGE. CPYRUN MUST BE LOADED WITH TAPLIB.

ALL THAT IS NEEDED TO READ DATA INTO A PROGRAM ARE STRTEC, GETEVT, GOBACK, AND SKFTAP. THESE ARE DESCRIBED BELOW....

CALL STRTEC(IDRIVE,IDENS)

THIS ROUTINE INITIALIZES THE READ SOFTWARE AND ATTEMPTS TO OPEN THE TAPE DRIVE. THE ARGUMENTS ARE:

IDRIVE = TAPE DRIVE SELECTED (0 = MTA100, 1 = MTA101)
IDENS = READ DENSITY (3 = 800 BPI, 4 = 1600 BPI, 5 = 6250 BPI)

THESE ARGUMENTS ARE IGNORED IN "DSKLIB".

CALL GETEVT(IEFG)

THIS ROUTINE PUTS THE NEXT SEQUENTIAL EVENT INTO THE COMMON BLOCK:

COMMON /TEC/ IRUN, IEVT, IFADC(256,4), ITDC(4,3), IPU

THE VARIABLE ASSIGNMENTS ARE OBVIOUS FOR THE ENLIGHTENED.
THE ARGUMENT "IEFG" = 0 IF A READ WAS SUCESSFUL, = 1 IF AN EOF WAS ENCOUNTERED. THIS ROUTINE (WHEN GOING TO TAPE) PRINTS OUT DETAILS OF ITS INTERNAL WORKINGS (ERROR MESSAGES, ETC.) AND CAN READ PAST AN EOF.

CALL GOBACK(N)

THIS ROUTINE EITHER REWINDS OR BACKFILES THE INPUT DATASET. WHEN GOING TO DISK, IT ALWAYS REWINDS. WHEN USING TAPE, THE OPERATION DEPENDS UPON THE VALUE OF THE ARGUMENT "N". IF N=0, THE ROUTINE SKIPS BACK TO THE BEGINNING OF THE CURRENT FILE (HANDY FOR PRERUNS). IF N=1, THE ROUTINE REWINDS THE TAPE. TO GO BACK SEVERAL FILES, GOBACK CAN BE CALLED SEVERAL TIMES IN SUCESSION.

CALL SKFTAP

THIS ROUTINE SKIPS ONE FILE FORWARD ON TAPE. TO SKIP MULTIPLE FILES, CALL IT SEVERAL TIMES IN SUCESSION (IN A DO LOOP, ETC.). IT IS USED TO INDEX INTO A TAPE TO SELECT A REQUIRED RUN. THIS ROUTINE IS A DUMMY WHEN GOING TO DISK.

THE FOLLOWING ROUTINES HANDLE MORE DETAILED DATA TRANSFER AND PROCESSING. THEY ARE NOT NEEDED FOR SIMPLE EVENT READING, BUT QUICK DESCRIPTIONS ARE INCLUDED HERE FOR REFERENCE.

CALL NEWEV(IE)

HANDLES THE READ AND BUFFER UNPACKING OF AN EVENT. IE = 0 FOR ERROR DIAGNOSTICS, IE = 1 TO IGNORE ERRORS (BUT DOESN'T DO MUCH ANYWAY, NEWEV(0) IS CUSTOMARILY USED). THE MOST IMPORTANT COMMON BLOCK IS COMMON /EVENT/IEV(1300),IEOF. IEV HOLDS THE CURRENT EVENT, AND IEOF IS A LOGICAL VARIABLE SET "TRUE" UPON ENCOUNTERING AN EOF, "FALSE" OTHERWISE. THE ARRAY IREC(1000) IN COMMON/DATA/ CONTAINS THE CONTENTS OF THE CURRENT PHYSICAL TAPE RECORD (UNPACKED INTO 16-BIT WORDS).
THIS ROUTINE PRINTS OUT VARIOUS THINGS OF DIAGNOSTIC IMPORTANCE.

CALL XPAND(IN,IOUT,N)

THIS ROUTINE UNPACKS ARRAY "IN" (DIMENSIONED BY "N") INTO 16-BIT WORDS PLACED INTO "IOUT" (DIMENSIONED BY 2*N). THIS ROUTINE IS USED BY NEWEV.

CALL LUNDIN

THIS ROUTINE INITIALIZES BUFFER POINTERS, ETC. USED IN THE LUND SOFTWARE.

CALL UNFADC

UNPACKS THE FLASH ADC INFORMATION, AND PUTS THE FIRST FOUR CHANNELS INTO THE /TEC/ COMMON BLOCK.

CALL CRS2(IC,IV)

THIS ROUTINE BREAKS THE EVENT BUFFER INTO 6-BIT WORDS USED TO UNPACK THE FLASH ADC'S. CALLED BY UNFADC.

IRAM(I,NB,IN) [FUNCTION]

SHIFTS "I" RIGHT BY "NB" BITS, ANDS THE RESULT WITH "IN". USED BY CRS2.

CALL SHIFTL(IN,IOUT,ISH)

ASSEMBLER ROUTINE TO SHIFT WORD "IN" BY "ISH" BITS, AND PUT RESULT IN "IOUT". ISH POSITIVE IS LEFT SHIFT, ISH NEGATIVE IS RIGHT SHIFT.

CALL OPNTAP(IDRIVE,IDENS,MTIND,IBKSZ,ISTAT)

OPENS A TAPE DRIVE. ARGUMENTS:

```
IDRIVE = DRIVE USED (0 = MTA100, 1 = MTA101)
IDENS  = READ DENSITY (3 = 800 BPI, 4 = 1600 BPI, 5 = 6250 BPI)
MTIND  = "INDUSTRY" FLAG. (1 = INDUSTRY, 0 = DEC COMPATIBLE)
IBKSZ  = BLOCK SIZE IN WORDS
ISTAT  = STATUS OUTPUT. (0 = SUCESS, -1 = PROBLEM ALLOCATING IO CHANNEL,
                -2 = PROBLEM IN OPEN, >0 MEANS PROBLEM SETTING DENSITY,
                >100 MEANS PROBLEM SETTING BLOCK SIZE).
BEWARE OF MULTIPLE CALLS TO OPEN!! (SEE "CLOSE" DESCRIPTION)
```

CALL TPREAD(BUFFER,NWRDS,ISTAT)

READ A RECORD FROM THE TAPE INTO "BUFFER". "NWRDS" = BUFFER SIZE, ISTAT = 0 IF SUCESSFUL, =-1 IF DIFFICULTY.

CALL TAPSTA(ISTAT,ICNT)

PUTS THE DEVICE STATUS WORD INTO "ISTAT", AND PUTS THE CHARACTER COUNT OF THE LAST READ OPERATION INTO "ICNT".

CALL UNSTAT(ISTAT,ICDE)

INTERPRETS THE STATUS WORD "ISTAT" OUTPUT FROM ROUTINE "TAPSTA". DIAGNOSTIC MESSAGES ARE PRINTED (CALL ONLY IF THE STATUS OUTPUT FROM "TPREAD" IN NONZERO), AND ICDE IS SET TO 1 (END-OF-FILE), 2 (END-OF-TAPE), OR 3 (ERROR ENCOUNTERED).
"ISTAT" IS INPUT (FROM A PREVIOUS CALL TO "TAPSTA"). THIS ROUTINE CALLS "CLRSTA" TO CLEAR SET EOF AND ERROR BITS.

CALL CLRSTA(IGO)

CLEARS THE EOF AND ERROR BITS IN THE DEVICE STATUS REGISTER TO ENABLE CONTINUED IO OPERATIONS.
IGO = 0 (SUCESS), -1 (GOOF).

CALL SKFTAP

SKIPS FORWARD ONE FILE ON TAPE.

CALL BKFTAP

SKIPS BACK ONE FILE ON TAPE. DOES NOTHING AT BOT.

CALL SKRTAP

SKIPS FORWARD ONE PHYSICAL RECORD ON TAPE.

CALL SKBTAP

SKIPS BACKWARD ONE PHYSICAL RECORD ON TAPE.

CALL REWTAP

REWINDS THE TAPE.

CALL CLSTAP

CLOSES THE TAPE CHANNEL. (BEWARE OF CALLING OPEN MORE THAN ONCE IN A PROGRAM, SINCE IT MODIFIES INSTRUCTIONS. DO NOT CHANGE TAPE DRIVES IF MULTIPLE CALLS TO OPEN ARE ATTEMPTED!!!).