CB-Note 92 Rev.12

LEAR Crystal Barrel Experiment, PS 197
Crystal Data Reconstruction Software Ver. 2.02/01

F.-H. Heinsius
T. Kiel
P. Schmidt
I. Institut für Experimentalphysik
Universität Hamburg

January 23, 1995

Contents

List of Tables

List of Figures

1 Introduction

This document describes the *crystal data reconstruction software*. This software analyses one event only on the basis of the crystal data. The main results are the data of the Particle Energy Depositions (short PED[1]), that is energy (in MeV), direction cosines, etc. These are stored in the **TBTK** linear structure which is described in chapter 2.2.4. Note that the global identification character for the *crystal data reconstruction software* is 'BC' (exceptions: BGTNBR, BITOPT).

Energies are quoted in MeV, angles in rad. Except routine BCANGL, which returns angles in degree. Two numbering schemes are used for the crystals: (φ,ϑ) with $\varphi = 1, 60$ and $\vartheta = 1, 26$ and as a shorter form[2]: $index = \varphi + (\vartheta - 1) \cdot 60$. Transformation from index to (φ,ϑ) is done with subroutine BITOPT.

Warning:

ECLUBC and EPEDBC set to 20 MeV by default, old value was 10 MeV. This should result in less electromagnetic split offs. A new energy correction method is introduced, see routine BCEGAM. This is used for new calibrations. A new variable (MOKFBC) allows a higher threshold for FERAs if the corresponding 2282 is ok. (Version 1.43)

Due to an error in the covariance error matrix it is needed to call routine BCTTKS for all data produced before version 1.41/01. Energy resolution default changed to 2.8% (Version 1.41). This can be changed in already produced data by calling BCTTKS (see the description of it). Energy dependent errors for direction cosines. Lower cutoff energy for the FERA (20 ADC counts) (Version 1.4). Lower cutoff energy for the 2282 ADCs introduced (in Version 1.1). The cuts ECLUBC and EPEDBC are done before the energy correction! So the minimum photon energy is slightly above 20 MeV.

Note:

For version 2.00/03 a **new algorithm** for the calculation of PED directions has been *optionally* included (subroutines BCNPED). It determines the four-momentum by the energy of the sum of nine crystals instead of the sum of all crystals in the cluster. Furthermore all crystals involved are weighted due to their topology and PED energy. Additionally in case of n PEDs in one cluster the energy of the overlapping crystals (i.e. crystals neighbouring two or more local maxima) is divided according to the energies of the n local maxima (in BCNPED). Include this algorithm with the additional '**PDRG**' parameter. Alternatively N.Hesseys PED-smoothing algorithm can be incorporated by using the additional '**PDSM**' parameter. PED-smoothing includes the following subroutines BCPOLY,BCPCOR, BCSINI, XYZRTP, RTPXYZ. Please note, that at this moment people are not advised to use both codes simultaneously. So, when using one of these algorithms, your **XTAL** card should read like this:

XTAL 'DECF' 'DECL' 'TRAK' 'ALCE' 'CLST' 'PEDS' **'PDSM' (or 'PDRG')**

[1]It was decided to use this term on July 13, 1988 at the software meeting in Karlsruhe.

[2]The short form is only used in the **TBCL** and **TBEN** banks, which are rarely used for physics analysis. It gives certain advantages in storage usage and coding.

Further note: to distinguish different production methods some bits are set in IQ(LTBTK):
Bit 0: Event has PED with crystal type 13
Bit 1: New energy correction used
Bit 2: PED smoothing enabled
Bit 3: New algorithm for one and n PED/cluster enabled 'PDRG'

Above all the direction of each crystal, determined in BCOSXT is calculated due to the centre of area instead of its geometrical centre. BCPEDS has been duly updated, another new subroutine has been added BCOSX9 replacing BCOSCL, which has been kept in in the software paket in case of need.

New subroutines for the DOLBY split-off recognition by N.Hessey are included in the code (for detailed description see CB - Note 199): DBCCOA, DBCINI, DBCNOS, DBCSPL and the SMART algorithm by J.Salk (see CB-Note 182, performance studies of DOLBY-C, SMART CB-Note 216). The SMART code involves the following routines: BSMART, BCNXTP, BCOSC2.

New user routine BCVROC to do the inverse of the vertex correction routine BCVCOR. Subroutine BCALCE changed to allow for gain correction from lightpulser. (Version 1.43) New routines to handle covariance error matrices added for use with PI0FND (Version 1.42). In addition to the vertex for neutral particles a z-disalignment between the two barrel halfs is allowed. See common /BCVRTX/. Function BCVRSN is now obsolete; the version number is taken directly from CMZ (Version 1.41). (If you still use PATCHY you might get an old version number and a missing sequence, but that does not affect the code.) The energy calculation in BCALCE was modified to allow to correct for electronic problems e.g. nonlinearity (see BCLINE). Subroutine to recalculate errors with given energy resolution in global track bank (BCTTKS). User callables subroutines MCKINE and MCVERT to print the **VERT** and **KINE** banks. (Version 1.4)

This production release includes an energy correction function for the PEDs to give correct energies. (Its tested for GCB 4.02 Monte Carlo single photons.) To analyze Monte Carlo data produced *before* GCB version 4.02 you have to apply the patchy flag `MC`. (This feature will be deleted in a future release, as the old MC data is wrong). The usercallable subroutine BCMCOR tries to correlate reconstructed PEDs with the particles created in the Monte Carlo.

Please report any errors found in the software immediately to:

Michael Doser (DOSER@CERNVM)

Fritz-Herbert Heinsius (F31HEI@DHHDESY3)

Torsten Kiel (I04KIE@DHHDESY3)

The structure of this document is copied from the *chamber reconstruction software* description written by Curtis A. Meyer.

This software is part of the Crystal Barrel offline software. It is available on the general group accounts on the CB-DECstations, VSXTAL Vax and on CERNVM (Ask the Softwaremanager Michael Doser for more information). Send comments to the authors of this document.

2 Data Banks for the Barrel Calorimeter

This section describes all Zebra banks used by the crystal data reconstruction software. Figure 1 shows the main connections of the banks *created* by this software. These banks contain only the data of one event. (Not shown are the calibration banks.) The first word of the **HTBC** bank is the integer version number of the software. It has 5 down links: -1 points to the **TBEN** bank, -2 to the **TBCL**, -3 to the **TBTK**, -4 to the **TBEF** and -5 to the **TBEL** bank.

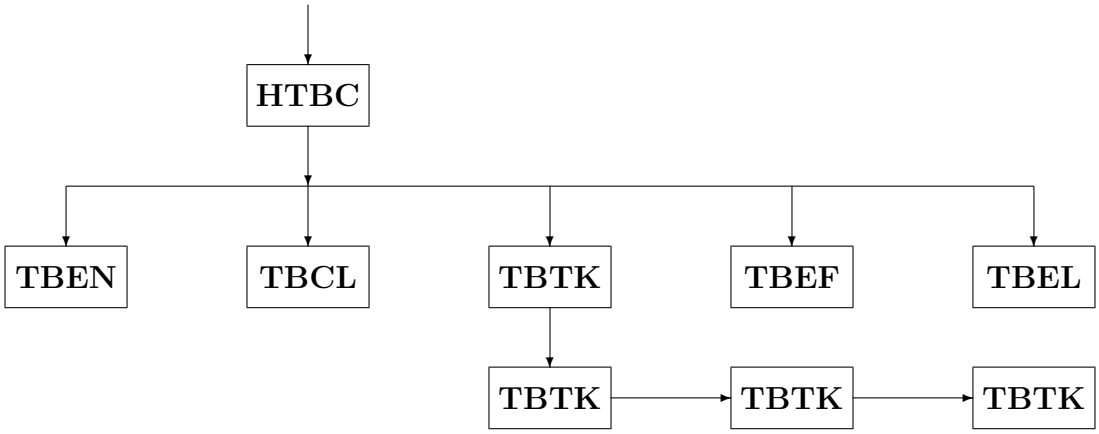

Figure 1: Structure of the banks

2.1 Description of the Barrel Calorimeter Calibration Banks

The calibration banks are loaded from the database into the ZEBRA store at the beginning of each run.

2.1.1 CBPF and CBPL

In the **CBPF** and **CBPL** banks are the pedestals for the FERA and for the LeCroy 2282 ADC systems stored, see table 1.

Offset	TYPE	*Quantity*
$+\varphi + (\vartheta - 1) \cdot 60$	REAL	*Pedestal* [ADC-Channel]
$\vdots$	$\vdots$	$\vdots$
+1561	REAL	*Threshold* [ADC-Channel]

Table 1: Data stored in the **CBPL** and **CBPF** bank (Pedestals LeCroy, FERA)

2.1.2 CBTF

The **CBTF** bank is temporary over one run and holds the thresholds above the pedestal for the FERA ADC System, see table 2. If the rounded value of the pedestal plus threshold is less than 255 then (-1) $\times$ threshold is stored in the bank otherwise the pedestal is stored (because then the FERA data is not pedestal subtracted, but this should be only in case of bad hardware).

Offset	TYPE	*Quantity*
$+\varphi + (\vartheta - 1) \cdot 60$	INTEGER	*negative offset to pedestal* [ADC-Channel]
$\vdots$	$\vdots$	$\vdots$

Table 2: Data stored in the **CBTF** bank (Thresholds for FERA)

2.1.3 CBEF and CBEL

In the **CBEF** and **CBEL** banks are the energy constants for the FERA and for the LeCroy 2282 ADC systems stored, see table 3.

Offset	TYPE	*Quantity*
$+\varphi + (\vartheta - 1) \cdot 60$	REAL	*Slope (Energy/ADC-Channel)* [MeV]
$\vdots$	$\vdots$	$\vdots$

Table 3: Data stored in the **CBEL** and **CBEF** bank (Calibration constants LeCroy, FERA)

2.1.4 CBXF and CBXL

The **CBXF** and **CBXL** banks contain a list of the the FERA and LeCroy 2282 ADCs which are not used for energy calculation, see table 4.

Offset	TYPE	*Quantity*
+1	INTEGER	*Number of bad channels N*
+2	INTEGER	*(φ,ϑ) of first bad channel*
⋮	⋮	⋮
+N+1	INTEGER	*(φ,ϑ) of last bad channel*

Table 4: Data stored in the **CBXL** and **CBXF** bank (Bad unused channels LeCroy, FERA)

2.1.5 CBCF and CBCL

In the **CBCF** and **CBCL** banks are correction functions for nonlinear channels of the FERA and LeCroy 2282 ADC channels stored (table 5). For a description of the correction function see subroutine BCLINE.

Offset	TYPE	*Quantity*
-K	REFERENCE LINK	*Pointer to type of correction function* N
⋮	⋮	⋮
-1	REFERENCE LINK	*Pointer to type of correction function* 1
LQ(↑)		
IQ(↓)		
+1	INTEGER	*Number of correction functions N*
+2	INTEGER	*(φ,ϑ) for first correction function*
⋮	⋮	⋮
+N+1	INTEGER	*(φ,ϑ) for Nth correction function*
+N+2 +0	INTEGER	*type of first correction function*
+N+2 +1	INTEGER	*number of real parameters K*
+N+2 +2	REAL	*parameter 1*
+N+2 +K+2	REAL	*parameter K*
⋮	⋮	next correction functions

Table 5: Data stored in the **CBCL** and **CBCF** bank (Nonlinear channels LeCroy, FERA)

2.2 Description of the Barrel Calorimeter Data Banks

2.2.1 TBEL and TBEF

In the **TBEL** bank are pedestal subtracted ADC counts from the 2282 ADC system stored, in the **TBEF** bank from the FERA ADC system. They are calculated from the **RBCL** and **RBCF** banks with the subroutines BCDECL and BCDECF. Note that in the **TBEL** bank is no entry if the 2282 ADC is in overflow (i.e. greater than MAXLBC of common /BCCUTS/) or less than the threshold MINLBC of common /BCCUTS/.

- *Number of Crystals* for which are entries in this bank.
- *(φ,ϑ)* coordinates of crystals as index = $\varphi + (\vartheta - 1) \cdot 60$
- *ADC counts pedestal subtracted* is the pedestal subtracted ADC count (in integer) of crystal (φ, ϑ).

Offset	TYPE	*Quantity*
+1	INTEGER	*Number of crystals* N
The following block is repeated N times for every crystal (I=1,N)		
$+2 \cdot I$	INTEGER	*(φ,ϑ)*
$+2 \cdot I + 1$	INTEGER	*ADC counts pedestal subtracted*

Table 6: Data stored in the **TBEL** and **TBEF** bank (ADC counts, pedestal subtracted)

2.2.2 TBEN

In the **TBEN** bank are the calibrated crystal energies stored, as calculated from the **TBEL (RBCL)** and **TBEF (RBCF)** bank with subroutine BCALCE. It will be created also if no crystal energy data exists, but then the first data word (number of crystals) will be 0. The first bit of the status bits (i.e. word IQ(LTBEN)) is set to 1 for new calibrations which requires the usage of the BCEGAM energy correction.

Offset	TYPE	*Quantity*
+1	INTEGER	*Number of crystals* N
The following block is repeated N times for every crystal (I=1,N)		
$+2 \cdot I$	INTEGER	*(φ,ϑ)*
$+2 \cdot I + 1$	REAL	*Energy of crystal* I [MeV]

Table 7: Data stored in the **TBEN** bank (crystal energies)

- *Number of Crystals* for which are entries in this bank.
- *(φ,ϑ)* coordinates of crystals as index $= \varphi + (\vartheta - 1) \cdot 60$
- *Energy of crystal* is the deposited energy in crystal (φ, ϑ) in MeV.

2.2.3 TBCL

The **TBCL** bank contains for every cluster a list of the crystals in that cluster. See table 8, an example can be found in table 9.

The reference link J of the **TBCL** bank points to the number of crystals in cluster J. A cluster is defined as a group of crystals where each crystal has the same edge or same corner as the next crystal. There is a minimum energy for a single crystal and for one whole cluster defined in common /BCCUTS/. The **TBCL** bank is filled in subroutine BCLUST. If there are no clusters the first data word of the bank will be 0.

- *Number of clusters* is the total number of clusters found.
- *Number of crystals* is the number of crystals in one special cluster. The *Offset* `JTBCL` for the cluster J can be calculated using:

```
JTBCL  = LQ(LTBCL-J)
```

- *(φ,ϑ)* coordinates of crystals as index $= \varphi + (\vartheta - 1) \cdot 60$

Offset	TYPE	*Quantity*
-K	REFERENCE LINK	*Pointer to Number of crystals in cluster* K
⋮	⋮	⋮
-1	REFERENCE LINK	*Pointer to Number of crystals in cluster* 1
LQ(↑)		
IQ(↓)		
+1	INTEGER	*Number of clusters* K
The following block is repeated *K* times for every cluster (J=1,K)		
LQ(LTBCL-J)+0	INTEGER	*Number of crystals* N
+1	INTEGER	*(φ,ϑ)*
⋮	⋮	⋮
+N	INTEGER	*(φ,ϑ)*

Table 8: Data stored in the **TBCL** bank (list of clusters)

Offset	VALUE	*Quantity*
-2	LTBCL+6	*Pointer to cluster 2*
-1	LTBCL+2	*Pointer to cluster 1*
LQ(↑)		
IQ(↓)		
+1	2	*Number of clusters*
+2	3	*Number of crystals*
+3	610	*(φ = 10,ϑ = 11)*
+4	670	*(φ = 10,ϑ = 12)*
+5	611	*(φ = 11,ϑ = 11)*
+6	1	*Number of crystals*
+7	410	*(φ = 50,ϑ = 7)*

Table 9: Example data for the **TBCL** bank

2.2.4 TBTK

The top **TBTK** bank contains only one data word: number of PEDs. The status bit number 1 (rightmost) is set to 1 if the events has a PED with the center crystal in ϑ 1 or 26. The reference links (number equal to number of PEDs) points to the linear structure of the **TBTK** banks. They contain the results from one analysed Particle Energy Deposition. At least one PED is created for every cluster.

The format is described in table 10.

You can loop through all PEDs using the following code:

```
    JTBTK = LTBTK
    IF (JTBTK .NE. 0) JTBTK = LQ(LTBTK-1)
 10 IF (JTBTK .NE. 0) THEN
       .
       . (this code is executed for every PED)
```

Offset	TYPE	*Quantity*	*filled*[a]
+1	INTEGER	φ *of central crystal*	•
+2	INTEGER	ϑ *of central crystal*	•
+3	INTEGER	*Number of cluster in which PED is in*	•
+4	INTEGER	*Number of PEDs in the cluster*	•
+5	REAL	u *direction cosine* $\cos\phi\sin\theta$	•
+6	REAL	v *direction cosine* $\sin\phi\sin\theta$	•
+7	REAL	w *direction cosine* $\cos\theta$	•
+8	REAL	*Corrected energy of PED [MeV]*	•
+9	REAL	*Error* Δu *of direction cosine*	•
+10	REAL	*Error* Δv *of direction cosine*	•
+11	REAL	*Error* Δw *of direction cosine*	•
+12	REAL	*Error in energy of PED [MeV]*	•
+13	REAL	*Energy of Cluster [MeV]*	•
+14	REAL	*Energy of central crystal [MeV]*	•
+15	REAL	*Energy sum of 9 [MeV]*	•
+16	REAL	*Invariant showermass [MeV]*	•
+17	REAL	*Second moment of shower*	•
+18	REAL	*Angle* Φ *[rad 0..2*π*]*	•
+19	REAL	*Angle* Θ *[rad]*	•
+20	REAL	$\sqrt{E[MeV]}$	•
+21	REAL	$(\Delta\Phi)^2$	•
+22	REAL	$\Phi - \Theta$ *correlation*	○
+23	REAL	$(\Delta\Theta)^2$	•
+24	REAL	$\Phi - \sqrt{E}$ *correlation*	○
+25	REAL	$\Theta - \sqrt{E}$ *correlation*	○
+26	REAL	$(\Delta\sqrt{E})^2$	•

[a] Meaning of *filled*:
• Entry correct and final
○ Entry not final, only approximately correct

Table 10: Data stored in the **TBTK** banks (ParticleEnergyDeposits)

```
   . (PED number is IQ(JTBTK-5) )
   .
   JTBTK = LQ(JTBTK)
   GOTO 10
END IF
```

You can reach the PED with number `N (N` $\leq$ `IQ(LTBTK+1))`:

```
JTBTK = LQ(LTBTK-IQ(LTBTK-2) - N)
```

3 Common Blocks used for PED Analysis

3.1 Description of the Barrel Calorimeter Common Blocks

All common blocks can be included with the patchy `+SEQ,`*name*`.` command. Only the BCENER and BCVRTX common blocks should be accessed from user routines.

3.1.1 BCBADX

In the /BCBADX/ common block are lists of crystals which should not be used or which are nonlinear stored:

```
 COMMON /BCBADX/ QBXLCB(1560), QBXFCB(1560)
&                IBCLCB(1560), IBCFCB(1560)
 LOGICAL QBXLCB, QBXFCB
 INTEGER IBCLCB, IBCFCB
```

- QBXLCB is true for all LeCroy ADC which should not be used.
- QBXFCB is true for all FERA ADC which should not be used.
- QBCLCB(I) is the number of the correction function used for LeCroy 2282 ADC I. 0 if no correction.
- QBCFCB(I) is the number of the correction function used for FERA ADC I. 0 if no correction.

3.1.2 BCCUTS

In the /BCCUTS/ common block are some constants stored:

```
 INTEGER MINLBC,MAXLBC,MINFBC,MOKFBC
 REAL EMINBC,EXTLBC,ECLUBC,EPEDBC,ECLSBC
 LOGICAL QSUBBC
 COMMON /BCCUTS/ MINLBC,MAXLBC,MINFBC,MOKFBC
&                EMINBC,EXTLBC,ECLUBC,EPEDBC,ECLSBC,QSUBBC
```

- MINLBC is the minimum ADC count in the LeCroy ADC to be stored in the TBEL bank (pedestal subtracted, zero supression)
- MAXLBC is equal the maximum counts in the LeCroy ADC (pedestal unsubtracted)
- MINFBC is the minimum ADC count of the FERA to be used for energy calculation (pedestal subtracted)
- MOKFBC is the minimum counts in FERA
- EMINBC is the minimum energy of one crystal used for analysis (**TBEN** bank)
- EXTLBC is the minimum energy of one crystal in one cluster (**TBCL** bank)
- ECLUBC is the minimum energy of one cluster

- EPEDBC is the minimum energy of the central xtal of one PED (not for first PED in a cluster)
- ECLSBC is minimum energy needed in a crystal to start a cluster
- QSUBBC is a logical which defines that the subroutines BCDECL and BCDECF do pedestal subtraction if the value is true (default).

3.1.3 BCENER

The /BCENER/ common block contains the array of the calibrated crystal energies for one event. It is filled in routine BCALCE or BCTRAK independent of the existence of the **TBEN** bank.

```
REAL ENERBC
COMMON /BCENER/ ENERBC(60,26)
```

- ENERBC(φ, ϑ) contains the energy in MeV for crystal (φ, ϑ).

3.1.4 BCFLAG

The /BCFLAG/ common holds the flags needed for steering the reconstruction of the crystal data. The common is filled in subroutine CBFFGO according to the values of the data card 'XTAL'.

```
 LOGICAL         TRAKBC, DECFBC, DECLBC, ALCEBC, CLSTBC, PEDSBC,
&                PDSMBC, PDRGBC
 COMMON/BCFLAG/ TRAKBC, DECFBC, DECLBC, ALCEBC, CLSTBC, PEDSBC,
&                PDSMBC, PDRGBC
```

- TRAKBC if true analyze crystal data.
- DECFBC if true decode FERA data.
- DECLBC if true decode LeCroy 2282 data.
- ALCEBC if true calculate energies from ADC data.
- CLSTBC if true find clusters.
- PEDSBC if true search for PEDs.
- PDSMBC flag for PED-smoothing algorithm
- PDRGBC flag for polynomial PED position algorithm

3.1.5 BCFORM

In the /BCFORM/ common are the characteristics for the bank formats stored. These are filled by MZFORM and used by MZBOOK.

```
INTEGER MTBEBC,MTBTBC
COMMON /BCFORM/ MTBEBC,MTBTBC
```

- MTBEBC contains the format for the **TBEN** bank.
- MTBTBC contains the format for the **TBTK** bank.

3.1.6 BCLOOK

The data in the /BCLOOK/ common connects the hardware numbers of the ADCs to the software crystal coordinates.

```
INTEGER MVSN,MAX22
INTEGER LOOKBC,LK22BC
PARAMETER (MVSN=90,MAX22=30*48)
COMMON /BCLOOK/ LOOKBC(3,0:15,0:MVSN-1), LK22BC(3,0:MAX22-1)
```

- MVSN are the maximum virtual station numbers of the FERA ADCs.
- MAX22 is the maximum number of 2282 ADCs.
- LOOKBC(I,SUBADDRESS,VSN) i=1: phi,i=2: theta, i=3: compound crystal index of FERA ADC with virtual station number VSN and SUBADDRESS.
- LK22BC(I,CHANNEL) i=1: phi,i=2: theta, i=3: compound crystal index of 2282 ADC with number CHANNEL (numbered according to the readout sequence!).

3.1.7 BCNBRS

The /BCNBRS/ common block contains arrays to calculate the neighbours of one crystal. It is used by the routine BGTNBR.

```
INTEGER IPNEBC,NEIGBC
COMMON /BCNBRS/ IPNEBC(60,26), NEIGBC(2,280)
```

- IPNEBC is a pointer to NEIGBC for every crystal
- NEIGBC(IPNEBC()) contains the number of neighbours for a given crystal and in the following storage positions the relative positions of the neighbour crystals as seen from the central crystal.

3.1.8 BCSTAT

In the /BCSTAT/ common block is the statistics of the analysis stored.

```
INTEGER ICALBC,IEVTBC,IPEDBC,MPEDBC,IERFBC,IERLBC,IVERBC
COMMON /BCSTAT/ ICALBC(6),IEVTBC(4),IPEDBC,MPEDBC,
&   IERFBC(5),IERLBC(5),IVERBC
```

- ICALBC stores the number of calls to the different reconstruction routines.
- IEVTBC counts number of events passed to different analysis steps.
- IPEDBC counts the total number of PEDs.
- MPEDBC gives the maximum number of PEDs in one event.
- IERFBC counts the number of FERA errors.
- IERLBC counts the number of 2282 errors.
- IVERBC is the BCTRAK version number.

3.1.9 BCVRTX

The /BCVRTX/ common block holds the vertex, which will be used to reconstruct the direction of the PEDs.

```
REAL             VRTXBC,VRZOBC
COMMON /BCVRTX/ VRTXBC(3),VRZOBC(2)
```

- VRTXBC(3) stores (x,y,z) of the vertex to be used for PED recontruction (in cm).
- VRZOBC(1) z-offset of the downstream half of the barrel.
- VRZOBC(2) z-offset of the upstream half of the barrel.

3.1.10 BCBCPC

This is the common block used by the PED-smoothing algorithm. It stores the PED position smearing coefficients.

```
COMMON /BCBCPC/ BCPCNP, BCPCTH, BCPCPH
INTEGER BCPCNT, BCPCMP
PARAMETER (BCPCNT = 52, BCPCMP = 21)
INTEGER BCPCNP(2, BCPCNT)
REAL BCPCTH(BCPCMP, 2 BCPCNT), BCPCPH(BCPCNT, 2, BCPCNT)
```

- BCPCNT is the number of theta divisions the corrections are split into
- BCPCMP is the max. number of coefficients per correction
- BCPCNP is the number of coefficients used for each theta
- BCPCTH, BCPCPH are arrays of correction coefficients

3.1.11 BCSPLI

This common block stores the spline coefficients used by the polynomial PED position algorithm (PDRG)

```
 INTEGER MENEBC,MSPLBC,MTYPBC
 PARAMETER (MENEBC=10,MSPLBC=4,MTYPBC=13)
 REAL RFUNBC, RFUMBC, RFUOBC, RENEBC
 COMMON /BCSPLI/ RFUNBC(MSPLBC,0:MENEBC-1,1:3,MTYPBC),
&                RFUMBC(MSPLBC,0:MENEBC-1,7:9,MTYPBC),
&                RFUOBC(MSPLBC,0:MENEBC-1,13:15,MTYPBC),
&                RENEBC(0:MENEBC)
```

- MSPLBC is the number of spline coefficients (cubic)
- MENEBC is the number of energy intervalls used for spline fitting
- MTYPBC is the number of xtal - types

- RFUNBC, RFUMBC, RFUOBC are the spline parameter for calculation of weight with the following meaning: 0 = **1, M = **2, N = **3, dim 1 = spline coefficient, dim 2 = energy interval, dim 3 = coefficient for neighbouring xtal - type (+/- theta), dim 4 = xtal - type
- RENEBC is the array storing the energy intervals

3.1.12 DBC

This is the common block used by the DOLBY - C method (see CB-Note 199 for further details).

```
REAL CTE, CTASYM, CTCOPA, CTEPI, CTNONG
COMMON /DBC/ CTE, CTASYM, CTCOPA, CTEPI, CTNONG
INTEGER GOODG, CHARGD, NONG
PARAMETER (GOODG = 0, CHARGD = -1, NONG = -2)
```

4 Particle Energy Deposition Reconstruction Software

This section gives a brief overview of each subroutine used in the *crystal data reconstruction software*. The software flow is shown in figure 2.

An asterisk (*) in front of a call argument indicates that this variable is changed by the subroutine. Unless specifically mentioned all variable types correspond to the FORTRAN defaults.

The subroutines described in section 4.2 can be called from the user routines.

4.1 Description of the Subroutines

4.1.1 SUBROUTINE BCTRAK

Author: F.-H. Heinsius
Creation Date: 20 July, 1988
Call Arguments: (*IERR)
Common Blocks Used: CBBANK, CBLINK, CBXDIV, CBCNTL, BCSTAT
Subroutines Referenced: BCDECF, BCDECL, BCALCE, BCLUST, BCPEDS, USEVNT

This subroutine is the main entry into the crystal data reconstruction software. ¿From here are the other subroutines called (BCDECF, BCDECL, BCALCE, BCLUST, BCPEDS). If the energy calculation is not yet done first BCDECF is called to decode the FERA data (enabled with option 'DECF' on card XTAL), then BCDECL decodes the LeCroy 2282 data (enabled with option 'DECL' on card XTAL), and BCALCE is called to calculate the energies (enabled with option 'ALCE' on card XTAL).

If the energy calculation is done the `ENERBC` array is filled with the energies from the **TBEN** bank. BCLUST is called only when the calculations of the clusters is not already done (enabled with option 'CLST' on card XTAL). BCPEDS called when the PED calculation is not yet done (enabled with option 'PEDS' on card XTAL); if there are no clusters only an empty **TBTK** bank is created.

An error code IERR not equal to zero is returned when a problem occurs in the decoding of either the FERA or the 2282 ADC information.

USEVNT is called at entry to the routine with parameter 10 and then with the following parameters: 11 after BCDECF, 12 after BCDECL, 13 after BCALCE, 14 after BCLUST, 15 after BCPEDS.

Reminder: If data is already calculated it will not be recalculated. To do a reanalysis you have to drop the banks with the old data or call the analysis routines (e.g. BCPEDS) from the USEVNT routine (e.g. `IF (I.EQ.15) CALL BCPEDS`, with `I` the parameter of USEVNT).

4.1.2 SUBROUTINE BCALCE

Author: F.-H. Heinsius
Creation Date: 18 July, 1988
Call Arguments: (*IERR)
Common Blocks Used: CBBANK, CBLINK, CBXDIV, BCBADX, BCCUTS, BCENER, BCFORM, BCSTAT
Subroutines Referenced: (ZEBRA) MZBOOK, MZPUSH
Functions Referenced: BCLINE

This routine transforms the ADC-counts of the LeCroy 2282 and FERA ADCs to energy. Input is taken from the Zebra banks **TBEL** and **TBEF**. Output is stored in the Zebra bank **TBEN**, if the energy is greater EMINBC.

Calibration constants are taken from the Zebra banks **CBEF**, **CBEL**. These constants are multiplied by the gain corrections taken from the lightpulser banks **CLGE**.

Nonlinearity is corrected by function BCLINE (in case of electronic failures). Only ADCs which are not marked as bad are used QBXLCB, QBXFCB.

If the ADC count of the LeCroy 2282 is greater than MINLBC above pedestal and less than MAXLBC (not pedestal subtracted) the LeCroy data is used; the FERA data is used if the 2282 data is not used and the ADC count of the FERA is greater MINFBC above the pedestal. If the corresponding 2282 is ok (array QBXLCB) the ADC count is the FERA must be greater MOKFBC above the pedestal. This routine fills also the /BCENER/ common.

4.1.3 SUBROUTINE BCDECF

Author: G. Folger
Creation Date: December, 1988
Call Arguments: (*IERR)
Common Blocks Used: CBBANK, CBLINK, CBXDIV, BCCUTSS, BCLOOK
Subroutines Referenced: CBDATE, ERRMES (ZEBRA) MZBOOK, MZPUSH, (KERNLIB) JBIT, JBYT

This subroutine decodes the FERA raw data from **RBCF** bank into the **TBEF** bank. Packed FERA data with and without pedestal subtraction can be decoded ('H0' and 'H1'). If the variable QSUBBC from common /BCCUTS/ is not set to false the resulting data will be pedestal subtracted ADC counts.

Error return codes:
IERR = 1: no **RBCF** bank
IERR = 2: expected FERA header, did not get one.
IERR = 3: found FERA header,expected data.
IERR = 4: reading data beyond limit.
IERR = 5: Attempt to write beyond end of bank.

4.1.4 SUBROUTINE BCDECL

Author: G. Folger
Creation Date: December, 1988
Call Arguments: (*IERR)
Common Blocks Used: CBBANK, CBLINK, CBXDIV, BCCUTS, BCLOOK
Subroutines Referenced: CBDATE, ERRMES (ZEBRA) MZBOOK, MZPUSH, (KERNLIB) JBYT

This subroutine decodes the LeCroy 2282 raw data from **RBCL** bank into the **TBEL** bank. Up to now only unpacked data can be decoded. If the variable QSUBBC from common /BCCUTS/ is not set to false the resulting data will be pedestal subtracted ADC counts. Note that no entry is put in the **TBEL** bank if the 2282 ADC is in overflow (i.e. greater than MAXLBC of common /BCCUTS/) or less than the threshold value (MINLBC of common /BCCUTS/).

Error return codes:
IERR = 1: no **RBCL** bank
IERR = 4: reading data beyond limit.

4.1.5 SUBROUTINE BCDONE

Author: F.-H. Heinsius
Creation Date: 25 January, 1990
Call Arguments: (ICODE)
Common Blocks Used: BCSTAT
Subroutines Referenced:

This routine is called once at the end of a job to write out information to the log file on diagnostics/summaries of the barrel recontruction code. The value of ICODE is a severity code which identifies if the job terminated normally or not.

4.1.6 FUNCTION BCECOR

Author: F.-H. Heinsius
Creation Date: 23 February, 1990
Call Arguments: (ENER, ITHE)
Common Blocks Used: none.
Subroutines Referenced:

This function returns a corrected energy for a given PED energy ENER and the theta ITHE of the center crystal of the PED. It corrects the most propable gamma energy deposit to the right gamma energy for Monte Carlo data.

$$E_\gamma = \text{ENER} \times C_m C_t(\text{ITHE}) C_r(\text{ENER})$$

- $C_m = 1.038$
- $C_t = 1.05$ (ITHE=1,26), 1.005 (ITHE=13,14), 1.004 (ITHE=2,3,24,25), 1. (ITHE=4–12,15–23)
- $C_r = 1 + \frac{0.6}{\text{ENER [MeV]}}$

4.1.7 FUNCTION BCEGAM

Author: F.-H. Heinsius
Creation Date: 20 June, 1991
Call Arguments: (ENER, ITHE)
Common Blocks Used: none.
Subroutines Referenced:

This function returns a corrected energy for a given PED energy ENER and the theta ITHE of the center crystal of the PED. It corrects the *mean* gamma energy deposit to the right gamma energy for Monte Carlo data. See Nigel Hesseys talk on june 1991 meeting. The usage of this function rquires the calibration to be performed with it. See the source for the used function.

4.1.8 SUBROUTINE BCERRS

Author: F.-H. Heinsius
Creation Date: 15 August, 1990
Call Arguments: (JTBTK)
Common Blocks Used: CBLINK
Subroutines Referenced:

This subroutine calculates the errors of the direction cosines, direction angles and energy and fills these values in the **TBTK** bank pointed to by the link JTBTK.

4.1.9 SUBROUTINE BCHCEN

Author: F.-H. Heinsius
Creation Date: 11 January, 1989
Call Arguments: (ENER, CNAME, *IERR, *QWRN, *QERR, *CTEXT)
Common Blocks Used: none.
Subroutines Referenced:

This Subroutine is used by BCHECK. Checks the energy ENER for being in a correct range. Warning if energy equal zero. QWRN (warning), QERR (error) are LOGICAL and CNAME (name of bank), CTEXT (description of error) are CHARACTER*(*).

4.1.10 SUBROUTINE BCHCPT

Author: F.-H. Heinsius
Creation Date: 9 January, 1989
Call Arguments: (*QFIRST, IPHI, ITHE, CNAME, *IERR, *QERR, *CTEXT)
Common Blocks Used:
Subroutines Referenced:

This Subroutine is used by BCHECK. Checks the numbers IPHI, ITHE for representing correct and existing crystal numbers. Also checks for double defined crystals. (Must be called before first test with QFIRST = .TRUE.). QFIRST, QERR (error) are LOGICAL and CNAME (name of bank), CTEXT (description of error) are CHARACTER*(*).

4.1.11 SUBROUTINE BCINIT

Author: F.-H. Heinsius
Creation Date: 20 July, 1988

Call Arguments: None.
Common Blocks Used: BCBADX, BCCUTS, BCFORM, BCNBRS, BCSTAT, BCVRTX, BCENER, BCLOOK, BCFLAG, CBBANK, CBLINK, CBXDIV, CBUNIT
Subroutines Referenced: (ZEBRA) MZFORM

This subroutine is called once at the start of the program. It initializes the BCBADX, BCCUTS, BCFORM, BCSTAT and BCVRTX common. Also the banks for the calibration constants (**CBEF**, **CBEL**, **CBPF**, **CBPL**, **CBTF**) are created and filled with constants for usage with the GCB-Monte-Carlo data (This is only done if the banks are not yet defined).

Also prints an identification of the software version and the date of the last CMZ run.

4.1.12 FUNCTION BCLINE

Author: F.-H. Heinsius
Creation Date: 22 October, 1990
Call Arguments: JCBC, IADC
Common Blocks Used: CBBANK, CBLINK, CBXDIV
Subroutines Referenced: None.

This function calculates a corrected value of the ADC counts given by IADC. The correction function is stored in the ZEBRA array IQ at position IQ(JCBC). It can be a maximum number of ADC counts or a polyline function. For details see the inline documentation.

4.1.13 SUBROUTINE BCLUST

Author: F.-H. Heinsius
Creation Date: 15 April, 1988
References: Crystal Ball CONREG routine
Call Arguments: None.
Common Blocks Used: CBBANK, CBLINK, CBXDIV, BCCUTS, BCENER, BCSTAT
Subroutines Referenced: BGTNBR, (ZEBRA) MZBOOK, MZPUSH

The BCLUST routine finds the clusters of crystals with energy. All crystals in one *cluster* must share at least one edge or corner with another crystal of the same cluster. The minimum energy of one crystal in a cluster is EXTLBC, the minimum energy of one cluster is ECLUBC (Both variables are taken from common /BCCUTS/). The result is stored in the **TBCL** bank.

4.1.14 SUBROUTINE BCOSCL

Author: T. Kiel
Creation Date: 13 May, 1988
Call Arguments: (ICL,*RX,*RY,*RZ,*RMASS)
Common Blocks Used: CBBANK, CBLINK, CBXDIV, BCENER
Subroutines Referenced: BCOSXT

For the energy cluster number ICL is the geometrical centre calculated. Returned are the direction cosines (RX, RY, RZ) of the center of gravity using the following formula (for RX):

$$R_x = \frac{\sum_{Crystals} E_i \cdot x_i}{\sqrt{(\sum_{Crystals} E_i \cdot x_i)^2 + (\sum_{Crystals} E_i \cdot y_i)^2 + (\sum_{Crystals} E_i \cdot z_i)^2}}$$

Also the invariant mass of the cluster will be calculated.

$$R_{\text{mass}} = \sqrt{\left(\sum_{Crystals} E_i\right)^2 - \left(\sum_{Crystals} \vec{p}_i\right)^2}$$

4.1.15 SUBROUTINE BCOSX9

Author: R.Glantz
Creation Date: 10 August, 1992
Call Arguments: (ICL,*RX,*RY,*RZ,*RMASS)
Common Blocks Used: CBBANK, CBLINK, CBXDIV, BCENER
Subroutines Referenced: BCOSXT, BC1PED

As opposed to BCOSCL for the energy cluster number ICL the centre of area is calculated. Returned are the direction cosines (RX,RY,RZ) after they have been corrected due to their topological and energy dependency in BCNPED. The invariant mass is calculated as in BCOSCL. Note: this routine is called only for one PED in the cluster.

4.1.16 SUBROUTINE BCPEDS

Author: F.-H. Heinsius
Creation Date: 25 July, 1988
References: T.Skwarnicki, DESY F31-86-02, App. A
Call Arguments: None.
Common Blocks Used: CBBANK, CBLINK, CBXDIV, BCCUTS, BCFORM, BCENER, BCSTAT, PI2PI
Subroutines Referenced: BCERRS, BCOSX9, BCOSXT, BCVCOR, BGTNBR, BCECOR, BCEGAM, BCNPED, (ZEBRA) MZBOOK

This subroutine searches for local maxima in all clusters and defines every local maxima as one PED. For every cluster there will be at least one PED defined. Note: The second, third etc. PEDs in one cluster must have at least EPEDBC Energy (common /BCCUTS/) in the central crystal. The **TBTK** bank is created, it supports the linear structure of the PEDs. This is also created and filled. See the description of the **TBTK** banks for the calculated data. This routine has been updated according to a new method for calculating PED directions.

Specification of algorithm:

1. For all clusters do
   (a) Calculate the energy of the cluster
   (b) Flag all crystals in the cluster
   (c) Repeat until no more flagged crystals:
      i. Find the crystal with the highest energy of all flagged crystals (for secondary PEDs energy greater than EPEDBC)
      ii. Unflag this crystal
      iii. Loop over all neighbour crystals
         A. If neighbour crystal has more energy go to 1.(c) (no local maximum)
         B. Unflag neighbour crystal
         C. Calculate energy sum over neighbours (sum of 9)

iv. Create the **TBTK** bank for this PED and fill with the following items:
- φ, ϑ of maximum crystal
- cluster number
- energy of maximum crystal, sum of 9, cluster
- invariant showermass, second moment

v. Increment counter for number of PEDs in this cluster

vi. In case Logical PDRGBC is set true (i.e. 'PDRG' has been chosen):
- check which crystals are neighbours to more than one local maximum (overlap)
- store sum of local maxima in overlap crystal in array ENRDIV

2. For all PEDs in the **TBTK** banks do

(a) Fill number of PEDs in the cluster to **TBTK** bank

(b) Calculate the energy and error, correct the energy using BCECOR or, if bit 1 of IQ(LTBEN) is 1 correct using BCEGAM.

i. One PED in a cluster: $E = E_{Cluster}$

ii. For n PEDs in a cluster (with PDRGBC set .TRUE.): recalculate E_9 in case of overlapping neighbour crystals. The energy of the overlap crystal is split in n parts depending on the number of local maxima it is a neighbour to.

$$E^i_{overlap/split} = \frac{E^i_{loc.max}}{\sum E_{loc.max}} E_{overlap/total}$$

iii. For n PEDs in a cluster (with PDRGBC = .TRUE. take new determined sum of nine mentioned above)

$$E = \frac{E_9}{\sum_{PEDs} E_9} E_{cluster}$$

Energy is stored in array RENPED needed in BCNPED.

(c) For calculating the direction of PED; for one PED per cluster and/or for n PED per cluster BCNPED if PDRGBC is .TRUE.:

i. Flag neighbouring crystals of local maxima

ii. Calculation of PED energy follows the fomulas mentioned above. Crystals being neighbours to n local maxima in case of more than one PED/cluster, their energy $E_{overlap}$ is divided due to the energies of these n maxima according to the formula:

$$REN = \frac{E_{overlap} \cdot E_{localmax}}{\sum_{loc.max} E}$$

With $\sum_{loc.max} E$ being the sum of the energies of n local maxima crystals.

(d) Calculate the direction cosine and errors

(e) if flag PDSMBC (using the 'PDSM' parameter) is set .TRUE. the PED-smoothing algorithm will be invoked, and thus the direction cosine and errors calculated accordingly

For merged π^0's use PI0FND with QMRGPI=.true..

4.1.17 SUBROUTINE BCNPED

Author: R.Glantz
Creation Date: 10 August, 1992
Call Arguments: (*INRBS,*IMXTL,*IPHI,*ITHE,*RENPED,*REN,RX,RY,
RZ)
Common Blocks Used: /CBLINK/BCENER/
Subroutines Referenced: BCOSXT

The output parameters RX,RY,RZ returned by this subroutine are the PED directions. To calculate these values BCNPED needs both central crystals with local maxima IMXTL, their neighbours INBRS with their crystal indices IPHI,ITHE. The energy of the PEDs, stored in array RENPED, being another input argument, is calculated according to the formula given in BCPEDS. The energy of each crystal neighbouring the n local maxima is stored in array REN (see BCPEDS). The detailed formula for the determination of the direction cosines is:

$$\vec{r}_{PEDx,y,z} = \frac{\sum_{i\in\Sigma_9} f(E_i/E_{max})E_i\vec{x_i}}{|\sum_{i\in\Sigma_9} f(E_i/E_{max})E_i\vec{x_i}|} \tag{1}$$

The weight function $f(E_i/E_{max})$ takes care of the energy and topology dependency of the spatial resolution and follows the algoritm:

$$f(E_i/E_{max}) = 1 + p_l\left(1 - \frac{E_i}{E_{max}}\right) + p_q\left(1 - \frac{E_i}{E_{max}}\right)^2 + p_k\left(1 - \frac{E_i}{E_{max}}\right)^3 \tag{2}$$

x_i being the directions of the crystals of the sum of nine determined in BCOSXT. The coefficients p_l, p_q, p_k are given by a spline-function which is determined for an energy range of 1 GeV divided into twelve nonlinear intervals.

4.1.18 SUBROUTINE BCVCOR

Author: F.-H. Heinsius
Creation Date: 11 May, 1990
Call Arguments: (*RX, *RY, *RZ, ITHE)
Common Blocks Used: /BCVRTX/
Subroutines Referenced: BCXTAL

This subroutines calculates from the given direction cosines RX, RY, RZ assuming a vertex at (VRTXBC(1), VRTXBC(2), VRTXBC(3)). The vertex can be set in common /BCVRTX/. For the inverse operation see description of subroutine BCVROC.

The showermaximum is taken at 4 cm from the face of the crystals (average for a 250 MeV photon). (See review of paricle porperties for formula or proposal for picture.) Note that it may vary between 2 and 6 cm for different energies, but the error in the correction is less than 10% due to this uncertainty (< 2 mrad).

4.1.19 FUNCTION BCVRSN

Author: F.-H. Heinsius
Creation Date: 13 June, 1990
Call Arguments: (CTIT, LENG)
Common Blocks Used: SLATE (KERNLIB)
Subroutines Referenced: (KERNLIB) CSQMBL, ICNEXT, ICFIND

Obsolete since version 1.41/01.

This integer function returns 10000 times the major version id plus the minor version id as given in the PAM title in the character variable CTIT of length LENG.

4.2 Description of the User Callable Subroutines

4.2.1 SUBROUTINE BCANGL

Author: T. Kiel
Creation Date: 26 July, 1988
Call Arguments: (RX,RY,RZ,*RPHI,*RTHE,*IPHI,*ITHE)
Common Blocks Used: BCVRTX
Subroutines Referenced:

This routine returns for a given directions cosine (RX, RY, RZ) (or momentum vector) the corresponding angles (RPHI, RTHE) in degree and the crystal numbers (IPHI, ITHE). If the direction does not hit any crystal, (0,0) will be returned for (IPHI, ITHE). Results are corrected for barrel disalignment in z-direction according to values given in common /BCVRTX/

4.2.2 SUBROUTINE BCCCTP

Author: T. Kiel
Creation Date: 20 February, 1991
Call Arguments: (VXYZ,CXYZ,*VRTP,*CRTP)
Common Blocks Used:
Subroutines Referenced:MTXMLT, MTXMBT

This routine transforms a momentum vector VXYZ(3) and the corresponding covariance matrix CXYZ(6) of a **C**harged particle from **C**artesian **T**o **P**olar coordinates VRTP(3), CRTP(6). The covariance matrix must be in lower triangular form.

4.2.3 SUBROUTINE BCCPTC

Author: T. Kiel
Creation Date: 20 February, 1991
Call Arguments: (VRTP,CRTP,*VXYZ,*CXYZ)
Common Blocks Used:
Subroutines Referenced:MTXMLT, MTXMBT

This routine transforms a vector VRTP(3) and the corresponding covariance matrix CRTP(6) of a **C**harged particle from **P**olar **T**o **C**artesian coordinates VXYZ(3), CXYZ(6). The covariance matrix must be in lower triangular form.

4.2.4 SUBROUTINE BCDUMP

Author: F.-H. Heinsius
Creation Date: 3 January, 1989
Call Arguments: (IUNIT, LINK, *QBAD)
Common Blocks Used: CBBANK, CBLINK, CBXDIV
Subroutines Referenced: BITOPT, (KERNLIB) UHTOC

This subroutine can be used by the user to dump a bank or linear structure pointed to by LINK. Output is written to FORTRAN unit IUNIT in a human readable format. This

routine can dump the following banks: **TBEF, TBEL, TBEN, TBCL, TBTK**. In case of the **TBTK** bank, all corresponding **TBTK** banks are also dumped. When it is called with LINK equal zero or with a link pointing to a bank, whose name is not mentioned above it will return the LOGICAL QBAD equal true. QBAD equal false is the normal return code.

4.2.5 SUBROUTINE BCHECK

Author: F.-H. Heinsius
Creation Date: 3 January, 1989
Call Arguments: (LINK, CNAME, *IERR, *QWRN, *QERR, *CTEXT)
Common Blocks Used: CBBANK, CBLINK, CBXDIV
Subroutines Referenced: BCHCEN, BCHCPT, BITOPT, (KERNLIB) UHTOC
Functions Referenced: none.

QWRN, QERR are LOGICAL and CNAME, CTEXT are CHARACTER*(*).

This subroutine checks the data of the bank pointed to by LINK and with the name CNAME for being consistent. (Tests can be performed for the following banks: **TBEF, TBEL, TBEN, TBCL, TBTK**.) If everything (reference links, numeric bank identifier, hollerith bank identifier, total number of links, number of data words and the data words) seems ok IERR returns 0, QWRN and QERR false.

For warnings (unusual data, wrong bank name, LINK equal 0) IERR returns a positive number, QWRN is true and QERR returns false. For errors (wrong data, links etc.) IERR returns a negative number, QWRN is false and QERR returns true. In any case a descriptive text is returned in CTEXT.

Note that the structural links are not tested. You can use the ZEBRA routine DZVERI to check the chaining of banks and the validity of the links (structural, next, up, origin links) in a whole division (Options 'CLU' and 'S' for store parameter).

4.2.6 SUBROUTINE BCISPI

Author: F.-H. Heinsius
Creation Date: 15 August, 1990
Call Arguments: (JTBTK)
Common Blocks Used: CBLINK
Subroutines Referenced: BCERRS, BCVCOR

This subroutine is useful for π^0's with two merged photons.

It calculates new direction and energy of one PED based on the whole cluster and not only one PED. Also the errors of the direction cosines, direction angles and energy are calculated. Results are filled in the **TBTK** bank pointed to by the link JTBTK. JTBTK should point to the PED with maximum energy. The other **TBTK** banks of this cluster might be dropped.

4.2.7 SUBROUTINE BCMCOR

Author: F.-H.Heinsius
Creation Date: 11 January, 1989
Call Arguments: (*IPA,*ITR,*EP,NMAX,*NCO,*NPART,*NPEDS)
Common Blocks Used: CBBANK, CBLINK, CBXDIV, BCENER, BCCUTS
Subroutines Referenced: BITOPT

This subroutine is still under developement. Please send any suggestions to the author. See also subroutines MCKINE and MCVERT.

The subroutine tries to correlate the reconstructed PEDs with the tracks (particles) given by the Monte Carlo bank `MBEN`. In the array IPA(NMAX) is for every PED the number of the correlated particle stored. In the array ITR(NMAX) is for every MC-particle the resulting PED number stored, or 0 if there is no correlation, -1 if the particle has no energy deposited or -2 if the particle has less than 1.05 times the minimum cluster energy deposited. NCO gives the total number of correlated PEDs with particles. NPART is equal the number of generated MC-particles which have more than 1.05 times the minimum cluster energy deposited. NPEDs gives the total number of PEDs.

One PED is correlated to a particle if the crystal with the maximum deposited energy is for both the same.

4.2.8 SUBROUTINE BCNCTP

Author: T. Kiel
Creation Date: 20 February, 1991
Call Arguments: (MSQRD,VXYZ,CXYZ,*VRTP,*CRTP)
Common Blocks Used:
Subroutines Referenced:MTXMLT, MTXMBT

This routine transforms a momentum vector VXYZ(3) and the corresponding covariance matrix CXYZ(6) of a **N**eutral particle from **C**artesian **T**o **P**olar coordinates VRTP(3), CRTP(6). The covariance matrix must be in lower triangular form. MSQRD (REAL) is the mass of the neutral particle.

4.2.9 SUBROUTINE BCNPTC

Author: T. Kiel
Creation Date: 20 February, 1991
Call Arguments: (MSQRD,VRTP,CRTP,*VXYZ,*CXYZ)
Common Blocks Used:
Subroutines Referenced:MTXMLT, MTXMBT

This routine transforms a vector VRTP(3) and the corresponding covariance matrix CRTP(6) of a **N**eutral particle from **P**olar **T**o **C**artesian coordinates VXYZ(3), CXYZ(6). The covariance matrix must be in lower triangular form. MSQRD (REAL) is the mass of the neutral particle.

4.2.10 SUBROUTINE BCOSXT

Author: T. Kiel
Creation Date: 10 May, 1988
Call Arguments: (IPHI,ITHE,*DCX,*DCY,*DCZ)
Common Blocks Used: BCBADX
Subroutines Referenced:

Returns the direction cosines (DCX, DCY, DCZ) of the centre of the crystal (IPHI, ITHE). The centre of the crystal being the centre of area. If the values of (IPHI, ITHE) are uncorrect it will return in (DCX, DCY, DCZ) (0,0,0). Corrects also for barrel z-disalignment.

4.2.11 SUBROUTINE BCTBTK

Author: F.-H.Heinsius
Creation Date: 30 January, 1990
Call Arguments: (MXTRK,*NTRK,*P4(4,MXTRK))
Common Blocks Used: CBLINK
Subroutines Referenced:

Returns the four vectors $(p_x/p, p_y/p, p_z/p, E)$ of all found PEDs as obtained on the `TBTK` banks in array P4. Number of PEDs is given in NTRK, P4 is filled up to PED MXTRK if NTRK>MXTRK.

4.2.12 SUBROUTINE BCTTKS

Author: F.-H. Heinsius
Creation Date: 13 September, 1990
Call Arguments: (JTTKS, DE)
Common Blocks Used: CBLINK
Subroutines Referenced:

This subroutine is needed, if you use DST-data, which was not produced with the correct error definitions. As the parameter DE you have to supply the energy resolution, e.g. 0.03 (0.025) for data and 0.02 for Monte Carlo. This subroutine calculates then errors of the direction cosines, direction angles and energy and fills these values in the **TTKS** bank pointed to by the link JTTKS. It also fills the covariance matrix needed for kinematic fitting.

4.2.13 SUBROUTINE BCVROC

Author: T. Kiel
Creation Date: 21 January, 1992
Call Arguments: (*RX, *RY, *RZ, ITHE, RVRTX(3))
Common Blocks Used:
Subroutines Referenced: BCXTAL

This subroutines calculates from the given direction cosines RX, RY, RZ assuming vertex at (RVRTX(1), RVRTX(2), RVRTX(3)) new direction cosines (output to RX, RY, RZ) assuming a vertex at (0,0,0). This is the inverse operation to the call of subroutine BCVCOR.

The showermaximum is taken at 4 cm from the face of the crystals (average for a 250 MeV photon). (See review of paricle porperties for formula or proposal for picture.) Note that it may vary between 2 and 6 cm for different energies, but the error in the correction is less than 10% due to this uncertainty (< 2 mrad).

4.2.14 FUNCTION BCXTAL

Author: T. Kiel
Creation Date: 25 May, 1988
Call Arguments: (IPHI, ITHE, ISEL)
Common Blocks Used: BCVRTX
Subroutines Referenced:

This function return various individual crystal parameters of the crystal (IPHI, ITHE) according to the value of ISEL:
ISEL = 1: crystal volume in cm^3

ISEL = 2: radius distance of crystal face to the center of the detector
ISEL = 3: front face area in cm^2
ISEL = 4: rear face area in cm^2
ISEL = 5: radius distance of crystal face to the center of the detector corrected for barrel z-disalignment
ISEL = -1: dimension H (see proposal)
ISEL = -4: dimension D (see proposal)
ISEL = -5: dimension B (see proposal)
ISEL = -6: dimension A (see proposal)
ISEL = -8: dimension D' (see proposal)
ISEL = -9: dimension B' (see proposal)
ISEL = -10: dimension A' (see proposal)

4.2.15 SUBROUTINE BGTNBR

Author: G. Folger
Creation Date: 11 August, 1987
Call Arguments: (IPHI, ITH, *NNBR, *IPHNB, *ITHNB)
Common Blocks Used: BCNBRS
Subroutines Referenced: None.

Finds the adjacent crystals to the given crystal (IPHI,ITH). The number of neighbours (5-9) are returned in NNBR. NNBR will be set to 0, if the crystal (IPHI,ITH) is a none existing crystal of ITH=1,2,3,24,25,26. The list of the neighbours are given in IPHNB(1..NNBR) and ITHNB(1..NNBR). Calculation is done using a list in common /BCNBRS/.

4.2.16 SUBROUTINE BITOPT

Author: F.-H. Heinsius
Creation Date: 1 August, 1988
Call Arguments: (IXTL,*IPHI,*ITHE)
Common Blocks Used: None.
Subroutines Referenced: None.

This subroutine transforms the **B**arrel **I**ndex **TO P**hi and **T**heta (used in the **TBEN** and **TBCL** bank). The crystal coordinates (IPHI, ITHE) are calculated from the crystal index IXTL using the inverse of the formula

$$\text{IXTL} = \text{IPHI} + (\text{ITHE} - 1) \cdot 60.$$

4.2.17 SUBROUTINE MCKINE

Author: R. Brun/F.-H. Heinsius
Creation Date: 9 January, 1990
References: GEANT subroutine GPKINE
Call Arguments: (LUN,IT)
Common Blocks Used: CBLINK
Subroutines Referenced:

Prints kinematics bank **KINE** for track number IT. If IT equal 0 prints all kinematics banks. Output on unit LUN.

4.2.18 SUBROUTINE MCVERT

Author: R. Brun/F.-H. Heinsius
Creation Date: 9 January, 1990
References: GEANT subroutine GPVERT
Call Arguments: (LUN,IV)
Common Blocks Used: CBLINK
Subroutines Referenced:

Prints vertex bank **VERT** for vertex number IV. If IV equal 0 prints all vertex banks. Output on unit LUN.

4.2.19 SUBROUTINE MTXMLT

Author: T. Kiel
Creation Date: 9 April, 1985
References: S. Brandt, Datenanalyse,BI-Verlag, 2. Aufl., Anhang B ISBN 3-411-01591-8
Call Arguments: (A,B,R,M,L,N)
Common Blocks Used:
Subroutines Referenced:

4.2.20 SUBROUTINE MTXMBT

Author: T. Kiel
Creation Date: 9 April, 1985
References: S. Brandt, Datenanalyse,BI-Verlag, 2. Aufl., Anhang B ISBN 3-411-01591-8
Call Arguments: (A,B,R,M,L,N)
Common Blocks Used:
Subroutines Referenced:

4.3 Description of the Block Data Routines

4.3.1 BLOCK DATA BCBLCK

Author: G. Folger
Creation Date: 10 August, 1987
Common Blocks filled: BCNBRS

In this block data is a list of neighbouring crystals for each crystal stored. It should only be referenced using the subroutine BGTNBR. For more information see the header of the block data.

4.4 Flow Chart of the Software

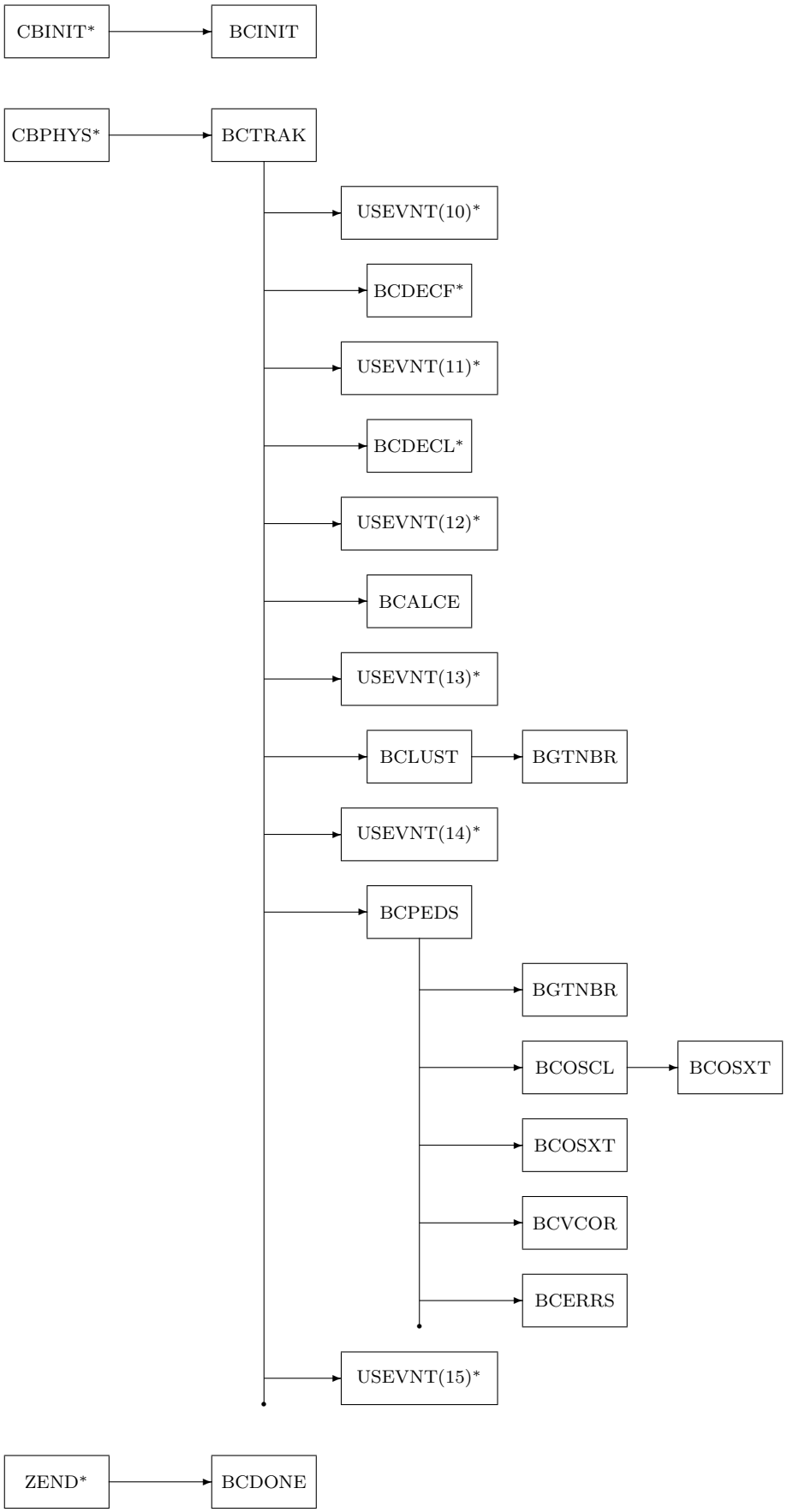

Figure 2: Crystal data reconstruction software flow chart. Initialization and calculation routines. The routines marked with * are not described in this document. They are part of the main offline package.

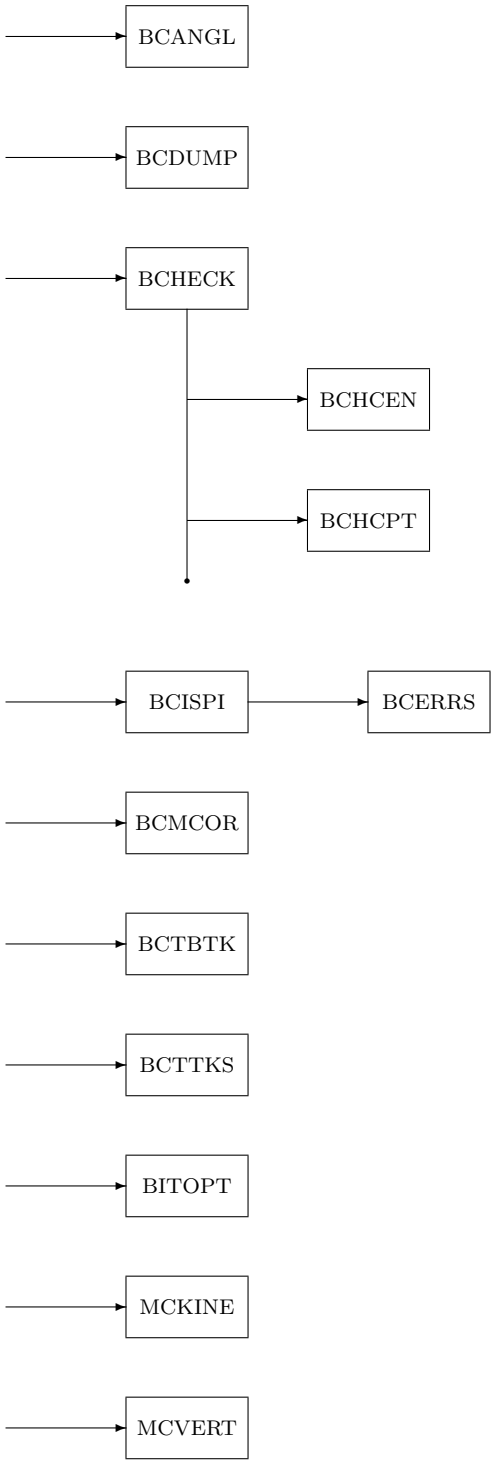

Figure 3: Crystal data reconstruction software flow chart. Only user callable routines.