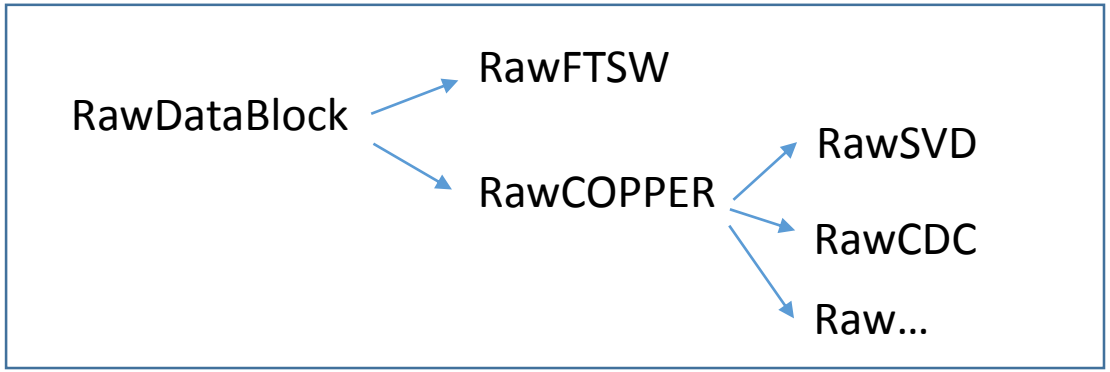
RawCOPPER data format

Dec. 16, 2013 (svn rev.7974)

Satoru Yamada

1, RawDataBlock object (to handle Raw data from COPPER board)



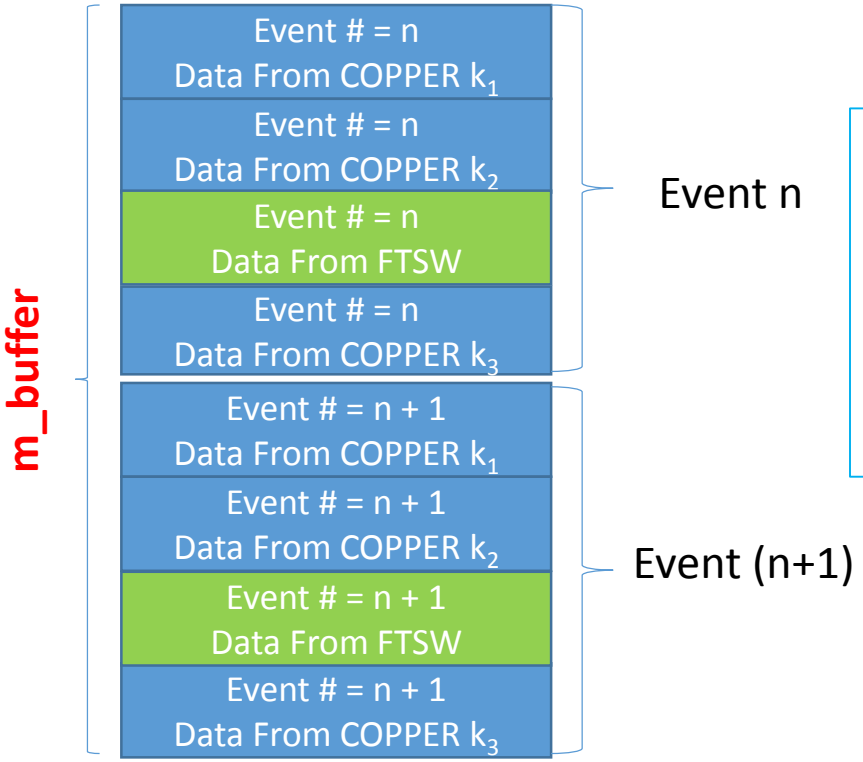
Source code :

https://belle2.cc.kek.jp/svn/trunk/software/rawdata/dataobjects/

```
RawDataBlock{
        methods to access data;
        int m_num_nodes; // # of nodes
        int m_num_events;// # of events

        int* m_buffer;  -> buffer for data
}
```

Example of data structure



In this example,
M_num_nodes = 4
M_num_events = 2.

# of data blocks = 4 * 2 = 8

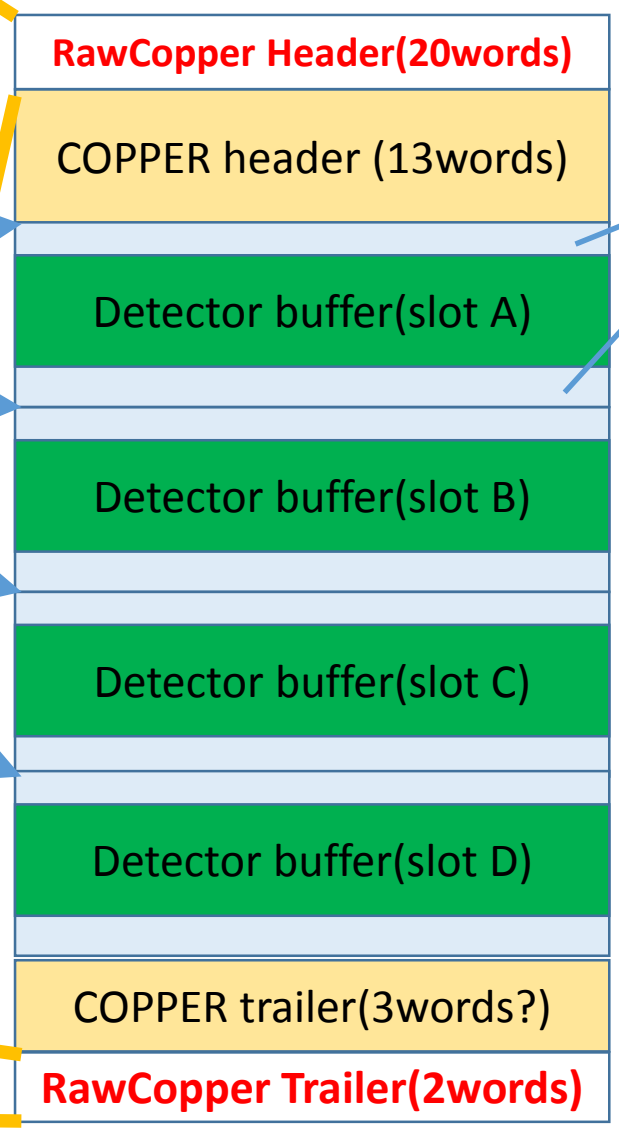
Overview of RawCOPPER format (one data block from a COPPER board)

- RawCOPPER header
  - COPPER header
    - B2link HSLB header (slot A FINNESSE)
      - B2link FEE header(slot A FINNESSE)
        - Data contents(Detector buffer) (slot A FINNESSE)
      - B2link FEE trailer (slot A FINNESSE)
    - B2link HSLB trailer (slot A FINNESSE)
    - B2link HSLB header (slot B FINNESSE)
      - B2link FEE header(slot B FINNESSE)
        - Data contents(Detector buffer) (slot B FINNESSE)
      - B2link FEE trailer (slot B FINNESSE)
    - B2link HSLB trailer (slot B FINNESSE)
    - B2link HSLB header (slot C FINNESSE)
      - B2link FEE header(slot C FINNESSE)
        - Data contents(Detector buffer) (slot C FINNESSE)
      - B2link FEE trailer (slot C FINNESSE)
    - B2link HSLB trailer (slot C FINNESSE)
    - B2link HSLB header (slot D FINNESSE)
      - B2link FEE header(slot D FINNESSE)
        - Data contents(Detector buffer) (slot D FINNESSE)
      - B2link FEE trailer (slot D FINNESSE)
    - B2link HSLB trailer (slot D FINNESSE)
  - COPPER trailer
- RawCOPPER trailer

2, "RawCOPPER header" and trailer format : 2013/8/26 (Not yet confirmed)

1	Number of total words
2	Number of words in this block (= 20)
3	exp no. (10bit=1024), run no.(22bit=4194304 including subrun)
4	event number(32bit)
5	From B2link FEE header 1
6	From B2link FEE header 2
7	subsystem ID (ID, crate no, slot no)
8	truncation mask (truncated or not) / type of data (compressed, calibration, ...)
9	offset to 1st block of user's data(FEE data)
10	offset to 2nd block of user's data(FEE data)
11	offset to 3rd block of user's data(FEE data)
12	offset to 4th block of user's data(FEE data)
13	reserved
14	reserved
15	# of nodes
16	node ID 1
17	node ID 2
18	node ID 3
19	Node ID 4
20	Termination word of this block = 0x7fff0005

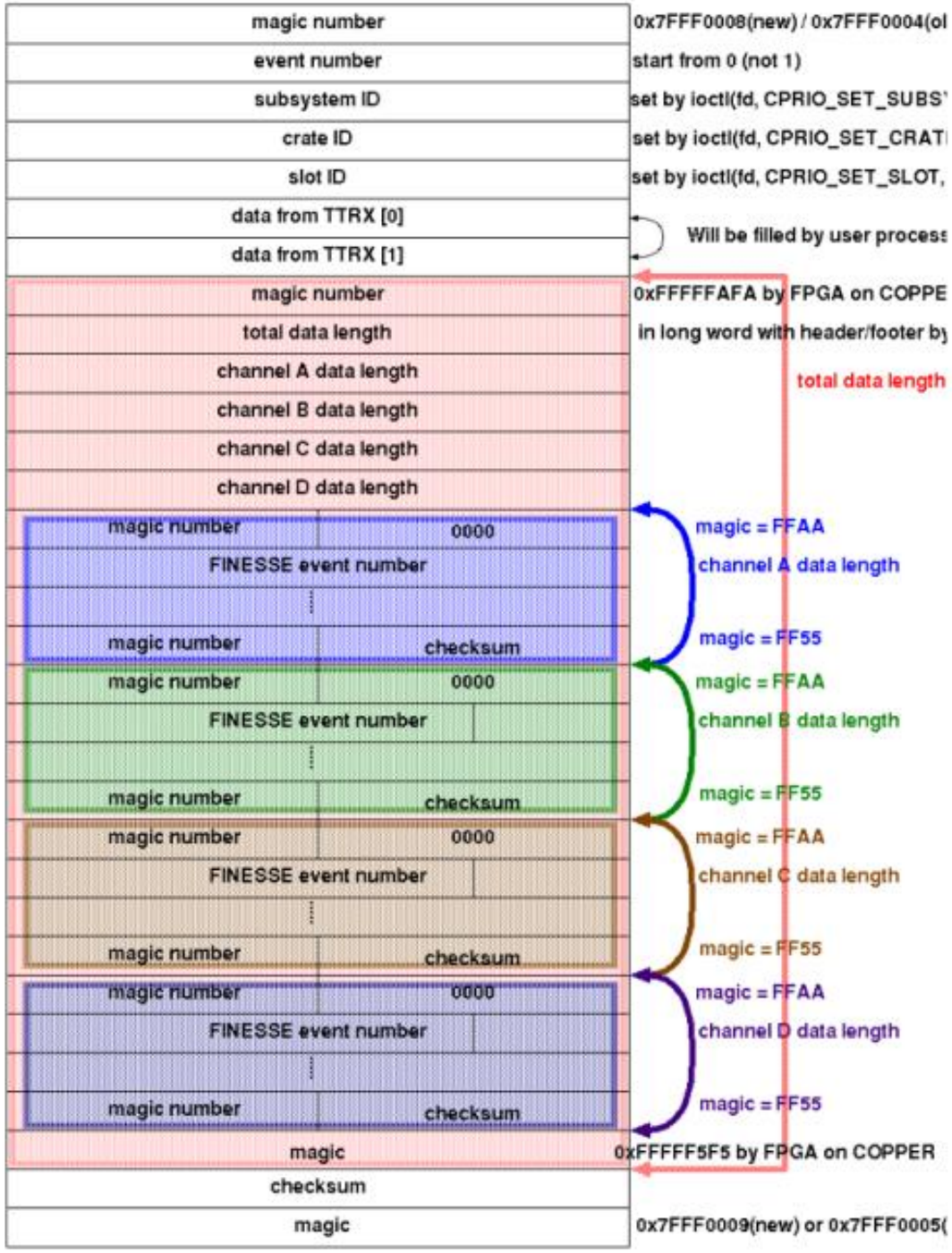
Information of nodes which this packet has gone through.



B2link(FEE/ HSLB) header and trailer

1	reserved (maybe used for checksum)
2	termination word of this block = 0x7fff0006

3, COPPER header and trailer from Belle document



4, B2link FEE header/Trailer, B2link HSLB header/Trailer

Data format (Final?)

From Nakao-san's B2GM slides:
http://kds.kek.jp/getFile.py/access?contribId=143&sessionId=38&resId=0&materialId=slides&confId=13911

The format used at the telescope test

```
-----------------------------------------------
HSL: 0xFFAA(16) --- B2L header  |      HSLB-tag(16)
-----------------------------------------------
B2L: '0'(1) | TT-ctime(27)              | TT-type(4)
B2L: TT-tag(32)
B2L: TT-utime(32)
B2L: TT-exprun(32)
B2L: '0' | B2L-ctime(27)         |     debug-flag(4)
-----------------------------------------------
FEE: Data #0 (32)
FEE: Data #1 (32)
FEE:    ....
FEE: Data #n (32)
-----------------------------------------------
B2L: TT-tag(16)                  |  B2L-checksum(16)
-----------------------------------------------
HSL: 0xFF55(16)                  | HSLB checksum(16)
-----------------------------------------------
```

B2link HSLB header

B2link FEE header

B2link FEE trailer

B2link HSLB trailer

- tag (event number) and utime to be increased to 32-bit (done), HSLB-checksum, B2L-checksum to be added

5, Example : how to get information of RawCOPPER header

You can get event # info from RawCOPPER object like this;

```
StoreArray<RawCOPPER> raw_cprarray;
 for (int i = 0; i < raw_cprarray.getEntries(); i++) {
```
When StoreArray is used

```
                for ( int j = 0; j < raw_cprarray[ i ].GetNumEntries(); j++) {
//              Get Event number
                        unsigned int event_no = raw_cprarray[ i ].GetEveNo( j );
//              Get RawCOPPER data block
                        int* buf = raw_cprarray[ i ].GetBuffer( j );
//              See contents of a data block (from RawCOPPER header to RawCOPPER trailer)
                        for( int k = 0; k < raw_cprarray[ i ] .GetBlockNwords( j ); k++ ){
                                printf("%d¥n", buf[ k ] );              }
//              Get Detector Buffer (raw data from detector electronics)
                        int* buf_slot_a = raw_cprarray[ i ].Get1stDetectorBuffer( j );
                        int* buf_slot_b = raw_cprarray[ i ].Get2ndDetectorBuffer( j );
                        int* buf_slot_c = raw_cprarray[ i ].Get3rdDetectorBuffer( j );
                        int* buf_slot_d = raw_cprarray[ i ].Get4thDetectorBuffer( j );
                        int* buf_slot[4]; for( int k = 0; k < 4;k++){ buf_slot[ k ] = raw_cprarray[ i ].GetDetectorBuffer(j,k) }
//              See contents of raw data from detector
                        for( int k = 0; j < raw_cprarray[ i ].Get1stDetectorNwords( j ); k++ ){
                                printf("%d¥n", buf_slot_a[ k ] );       }
                        for( int k = 0; j < raw_cprarray [ i ].Get2ndDetectorNwords( j ); k++ ){
                                printf("%d¥n", buf_slot_b[ k ] );       }
                        …..
                }
}
```

Test program to read RawCOPPER(RawCDC) data

1, Get dummy data file (data from two CDC FEE boards connected to FINESSE A and C.)
login.cc.kek.jp : ~yamadas/rawdata/root_output_RawCDC_rev7133.root

2, See contents of the data
% cd ${BELLE2_LOCAL_DIR}/daq/; svn update
% cd ${BELLE2_LOCAL_DIR}/daq/rawdata/examples/
% basf2 **ReadStoreTemplate.py** -i ./root_output_RawCDC_rev7133.root | less

```
[INFO] Steering file: ReadStoreTemplate.py
>>> basf2 Python environment set
>>> Framework object created: fw

===== DataBlock(RawCDC) : Block # 0 : Event # 0 : node ID 0x00000000 : block size 224 bytes
== Detector Buffer(FINESSE A)
0x0094c13a 0x91000001
== Detector Buffer(FINESSE C)
0x0094c13a 0x91000001

===== DataBlock(RawCDC) : Block # 1 : Event # 1 : node ID 0x00000000 : block size 224 bytes
== Detector Buffer(FINESSE A)
0x0094c23f 0xf1000001
== Detector Buffer(FINESSE C)
0x0094c23f 0xf1000001

===== DataBlock(RawCDC) : Block # 2 : Event # 2 : node ID 0x00000000 : block size 224 bytes
== Detector Buffer(FINESSE A)
0x0094c30d 0x69000001
== Detector Buffer(FINESSE C)
0x0094c30d 0x69000001
....
```

In this data,
Detector buffer contains only 2words(=8bytes) per/FINESSE/event.

Note that block # is a number used by DAQ software for handling data and not related with Event #.