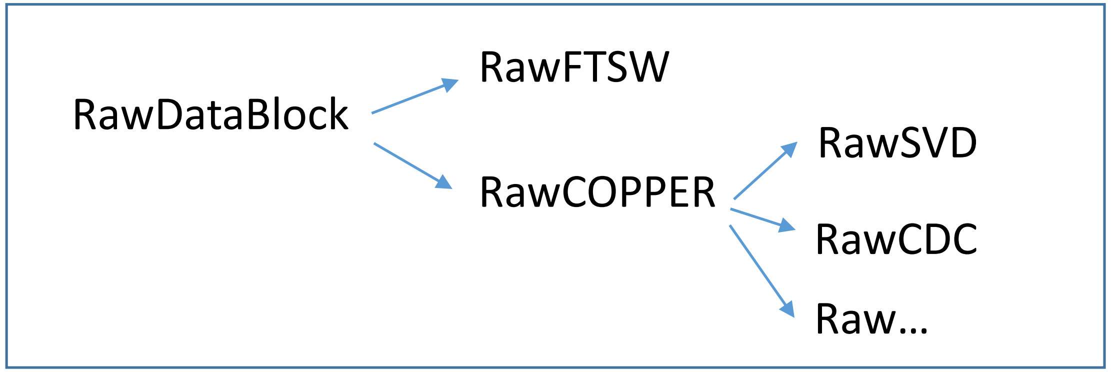
RawCOPPER data format

July 25, 2014 (svn rev. 11234)

Satoru Yamada

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



RawDataBlock

RawFTSW

RawCOPPER

RawSVD

RawCDC

Raw…

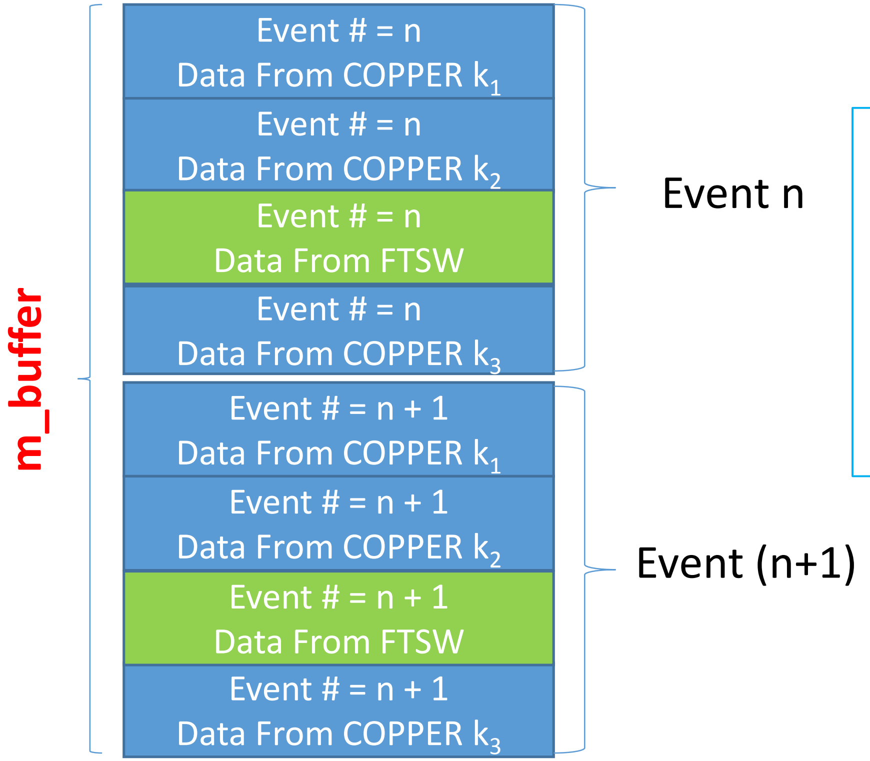
Source code :

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



m_buffer

Event # = n
Data From COPPER k_1

Event # = n
Data From COPPER k_2

Event # = n
Data From FTSW

Event # = n
Data From COPPER k_3

Event n

Event # = n + 1
Data From COPPER k_1

Event # = n + 1
Data From COPPER k_2

Event # = n + 1
Data From FTSW

Event # = n + 1
Data From COPPER k_3

Event (n+1)

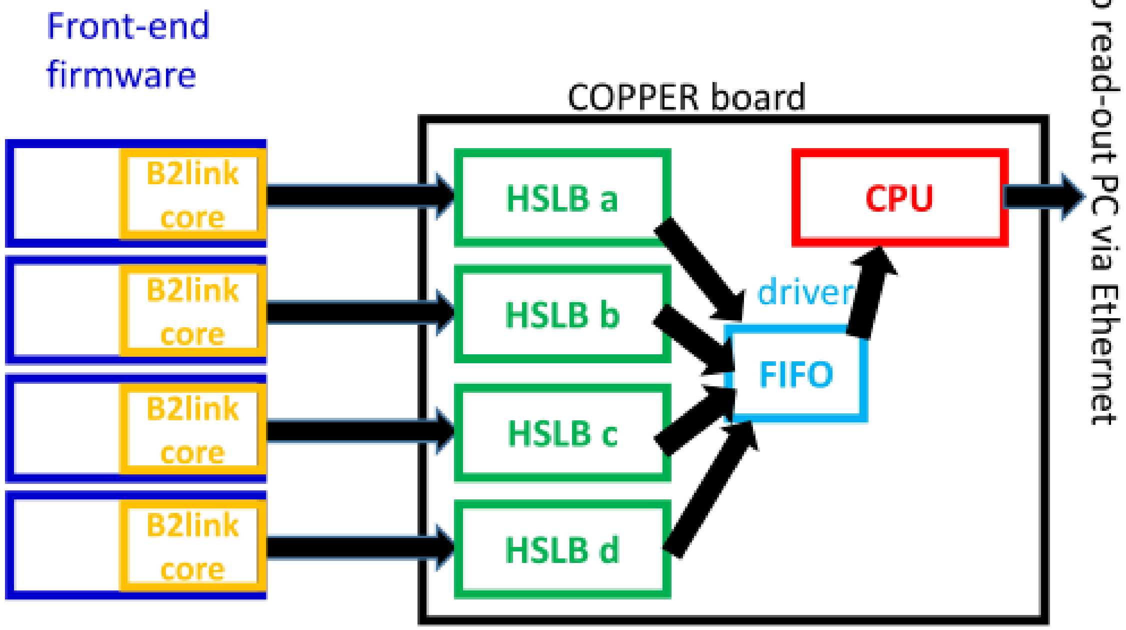
In this example,
M_num_nodes = 4
M_num_events = 2.

# of data blocks = 4 * 2 = 8

Overview of RawCOPPER format (one event block from one COPPER board)

(Reported at the last B2GM)

RawCOPPER header/trailer	-> See Sec. 2
COPPER header/trailer	-> See Sec.3
B2link(FEE+HSLB) header/trailer	-> See Sec.4
Detector buffer	-> Untouched by DAQ



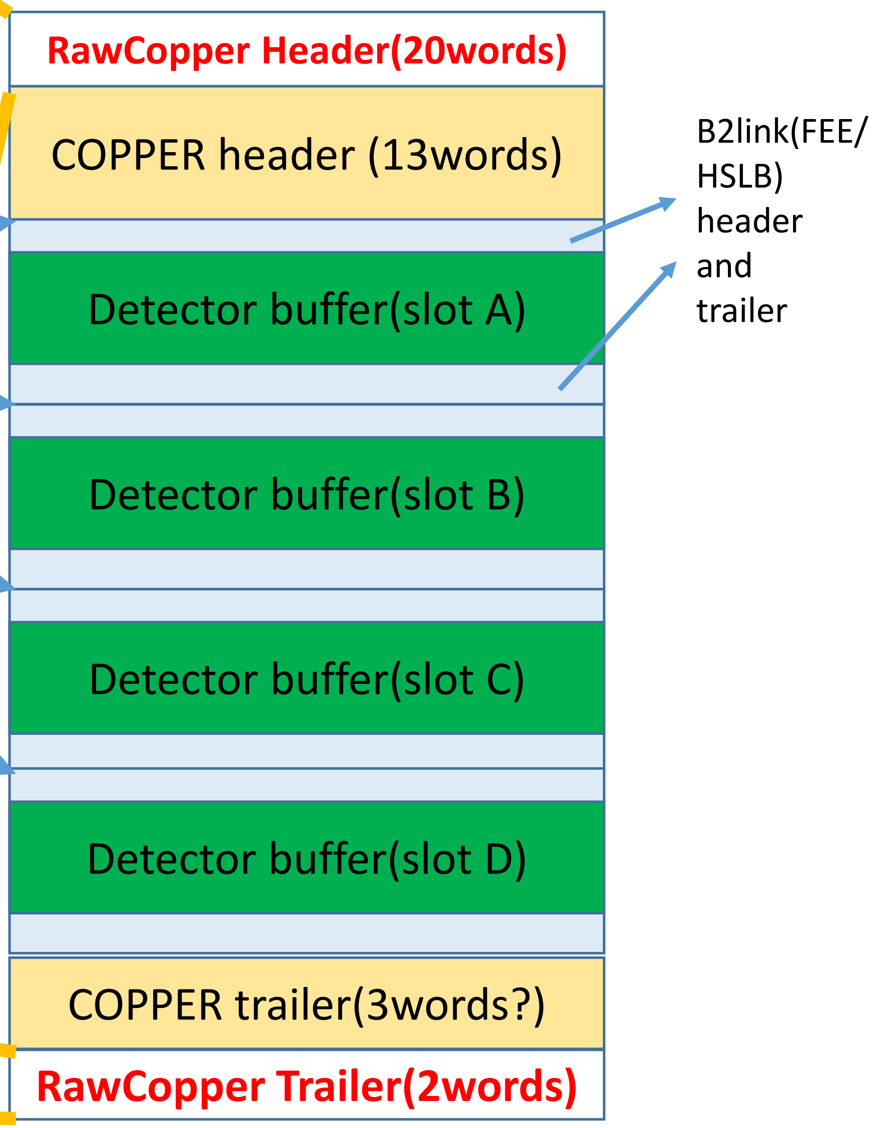
- RawCOPPER header
  - COPPER header
    - B2link HSLB header (slot A FINNESSE)
      - B2link FEE header(slot A FINNESSE)
        - Data contents(Detector buffer) (slot A FINNESSE)
      - B2link FEE trailer (slot A FINNESSE)
    - B2link HSLB trailer (slot A FINNESSE)
    - B2link HSLB header (slot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

HSLB : High speed link board

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.



RawCopper Header(20words)

COPPER header (13words)

Detector buffer(slot A)

Detector buffer(slot B)

Detector buffer(slot C)

Detector buffer(slot D)

COPPER trailer(3words?)

RawCopper Trailer(2words)

B2link(FEE/HSLB) header and trailer

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

2-1, 32bit Subsystem ID (A.K.A. node ID)

(31-24)	Detector ID	: 8bit=256	: detector & DAQ nodes
(23-17)	CRATE ID	: 7bit=128	:
(16-12)	SLOT ID	: 5bit=32	:
(11-0)	N.A.	: 12bit (4096) COPPER S/N?	

SubsystemID = “TTD ” = 0x54544420 and is reserved by FTSW ID now.

Detector ID (Defined in release/rawdata/dataobjects/include/RawCOPPER.h)

- #define SVD_ID 0x01 // tentative
- #define CDC_ID 0x02 // tentative
- #define BPID_ID 0x03 // tentative
- #define EPID_ID 0x04 // tentative
- #define ECL_ID 0x05 // tentative
- #define KLM_ID 0x06 // tentative
- #define PXD_ID 0x07 // tentative

Position of Subsystem(node) ID in headers

FTSW data format(2013/9/17)

1 Number of total words(=12)

2 Number of words in this header (=6)

3 Number of Events in this block(16bit) / # of boards(16bit)

4 exp no. (10bit=1024), run no.(22bit=4194304)

5 event number(32bit) of **the first data block**

6 Node ID(COPPER ID@COPPER, FTSW ID@VMIC / ROPC ID@ROPC)

7 Nakao-san word1

8 Nakao-san word2

9 Nakao-san word3 (0x00000000)

10 Nakao-san word4 (0x00000000)

11 SendTrailer1 (not used : maybe for checksum)

12 SendTrailer2 (magic word=0x7fff000?)

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

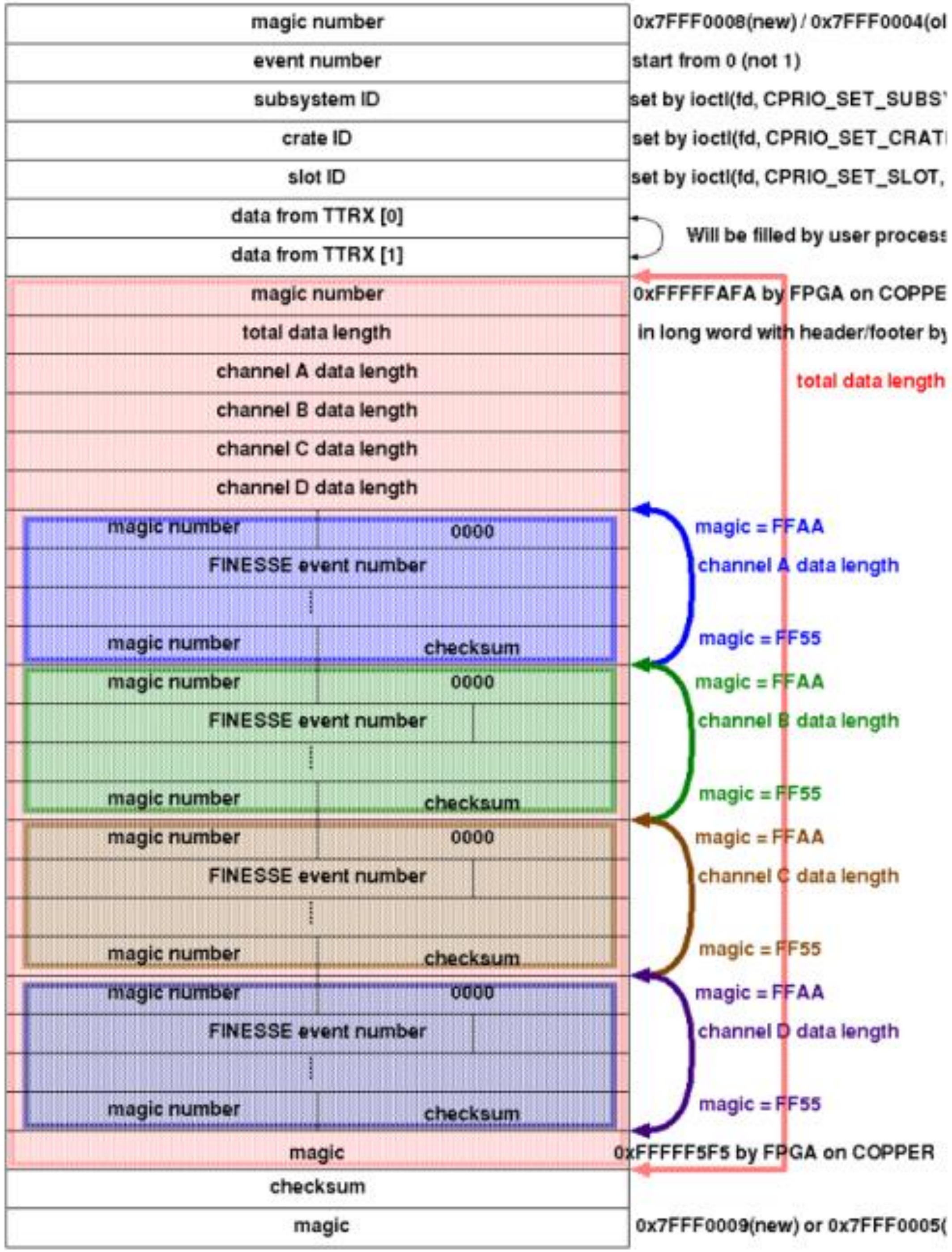
18 node ID 3

19 Node ID 4

20 Termination word of this block = 0x7fff0005

No one except Deserializer COPP ER fills this region now.

3, COPPER header and trailer from Belle document



magic number | 0x7FFF0008(new) / 0x7FFF0004(ol

event number | start from 0 (not 1)

subsystem ID | set by ioctl(fd, CPRIO_SET_SUBS

crate ID | set by ioctl(fd, CPRIO_SET_CRATI

slot ID | set by ioctl(fd, CPRIO_SET_SLOT,

data from TTRX [0]

data from TTRX [1]

Will be filled by user process

magic number | 0xFFFFFAFA by FPGA on COPPE

total data length | in long word with header/footer by

channel A data length

channel B data length

channel C data length

channel D data length

total data length

magic number | 0000

FINESSE event number

magic number | checksum

magic = FFAA

channel A data length

magic = FF55

magic number | 0000

FINESSE event number

magic number | checksum

magic = FFAA

channel B data length

magic = FF55

magic number | 0000

FINESSE event number

magic number | checksum

magic = FFAA

channel C data length

magic = FF55

magic number | 0000

FINESSE event number

magic number | checksum

magic = FFAA

channel D data length

magic = FF55

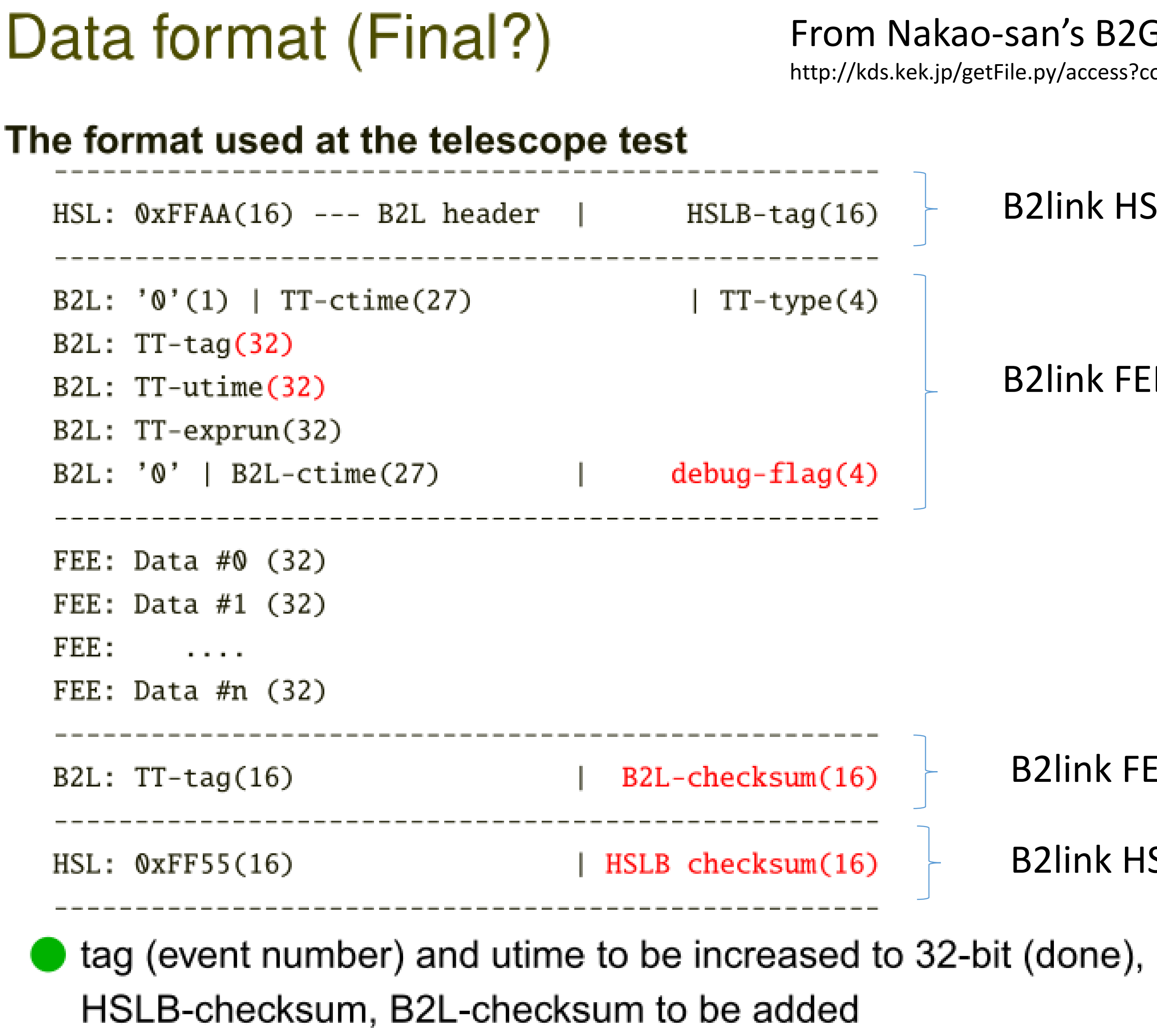
magic | 0xFFFFF5F5 by FPGA on COPPER

checksum

magic | 0x7FFF0009(new) or 0x7FFF0005(

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

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



Data format (Final?)

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)
--------------------------------------------------
```

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

B2link HSLB header

B2link FEE header

B2link FEE trailer

B2link HSLB trailer

5, Example : how to get information of RawCOPPER header

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

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

5-2, RawCOPPER class select proper “format class” depending on Pre/Post formats

Data taken at the DESY beam test(old format) can be read with the latest rawdata package

-> by checking data ver. In header.

New RawCOPPER class

- No change in style of the member functions -> No effect on derived class
- Does not have a format information in itself
  - Format class contains format information
    - RawCOPPERformat.cc -> the latest format
      - RawHeader.cc
    - RawCOPPERformat_v0.cc -> an old format
      - RawHeader_v0.cc
  - Assign format class to m_access in CheckVersionSetBuffer()
  - Use m_access to access buffer contents

```
inline int RawCOPPER::GetExpNo(int n)
{
CheckVersionSetBuffer();
return m_access->GetExpNo(n);
}
inline int RawCOPPER::GetRunNo(int n)
{
CheckVersionSetBuffer();
return m_access->GetRunNo(n);
}
```

RawData unpacker and packer

July 15, 2014 (SVN rev.11616)

Satoru Yamada

1, Unpacker

1-1, Example : how to get information of RawCOPPER header

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

```
StoreArray<RawCOPPER> raw_cprarray; // When StoreArray is used
 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("%.8x¥n", buf[ k ] );

   //     Get Detector Buffer (raw data from detector electronics)
   for( int finesse_num =0; finesse_num < 4; finesse_num++) {
        int* buf_slot = raw_cprarray[ i ]->GetDetectorBuffer( j, finesse_num );
        //        See contents of raw data from detector
        for( int k = 0; k < raw_cprarray[ i ]->GetDetectorNwords( j, finesse_num ); k++ ){
         printf("%.8x¥n", buf_slot[ k ] );          }
   }
  }
 }
```

1-2, 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 #.

2, Packer

2-1, Function to store data in RawCOPPER object

```
void RawCOPPER::PackDetectorBuf(  int* detector_buf_1st, int nwords_1st,  int* detector_buf_2nd, int nwords_2nd,  int* detector_buf_3rd, int nwords_3rd,  int* detector_buf_4th, int nwords_4th, RawCOPPERPackerInfo  rawcprpacker_info ){}
```

Input variables :

int* detector_buf_*** : pointer to the detector buffer that you want to store as ***th FINESSE data.

int nwords_*** : length of the detector_buf_*** (unit -> word = 4bytes)

RawCOPPERPackerInfo rawcprpacker_info : Information to

fill RawHeader

```
struct RawCOPPERPackerInfo {
   unsigned int exp_num; // 10bit
   unsigned int run_subrun_num; // 22bit
   unsigned int eve_num; // 32bit
   unsigned int node_id; // 32bit
   unsigned int tt_ctime; // 27bit clock ticks at trigger timing distributed by FTSW. For details, see Nakao-san's belle2link user guide
   unsigned int tt_utime; // 32bit unitx time at trigger timing distributed by FTSW. For details, see Nakao-san's belle2link user guide
   unsigned int b2l_ctime; // 27bit clock ticks at trigger timing measured by HSLB on COPPER. For details, see Nakao-san's belle2link user guide
   unsigned int hslb_crc16_error_bit; // 4bit errorflag for CRC errors in data transfer via b2link. ( bit0,1,2,3 -> finesse slot a,b,c,d)
   unsigned int truncation_mask; // Not defined yet
   unsigned int type_of_data; // Not defined yet
 };
```

(#include <rawdata/include/RawCOPPERPackerInfo.h>)

2-2, test program to store data in RawCOPPER object

- ➢ Module to fill dummy data in RawCOPPER

- rawdata/modules/src/DummyDataPacker.cc

- ➢ Script to run the above module

- $ rawdata/scripts/DummyDataPacker.py

Revision History of this document

- 2014. July 25 : remove an example section
- 2014. July 8 : subsystem ID/ node ID -> unified to node ID
- Jan.5, 2014 rev. 8376 : Add definition of tentative subsysID format
- Dec. 16, 2013 rev.7974 :
  - Add B2linkFEE header format
  - Add comments about handling StoreArray when unpacking Raw*** data.
- Oct.21, 2013 :rev.7133
  - Add instruction about Rawdata unpacking program
- Oct. 18, 2013 :rev. 7095
  - 1st draft