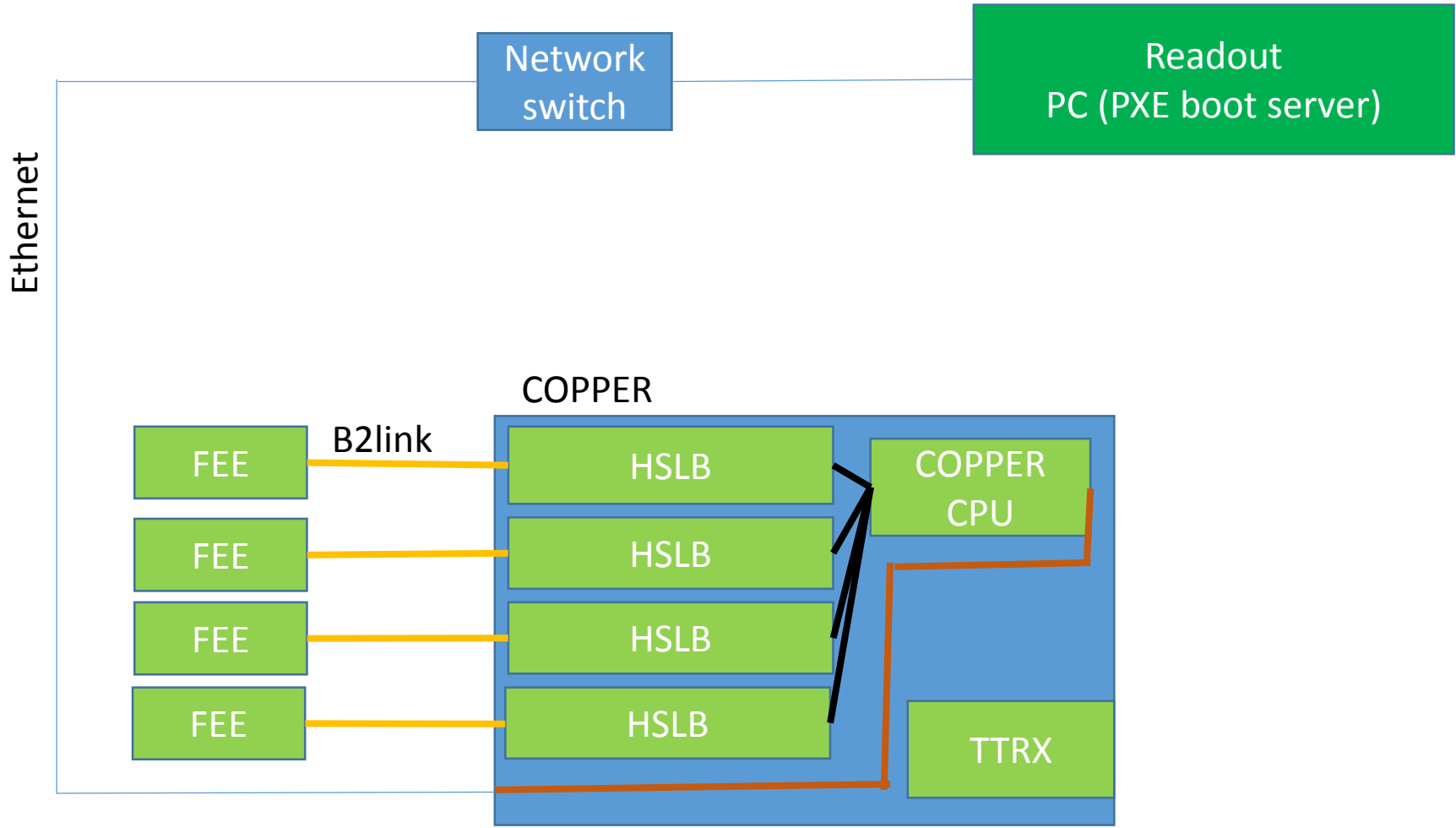


Pocket DAQ manual (2013.10.16)

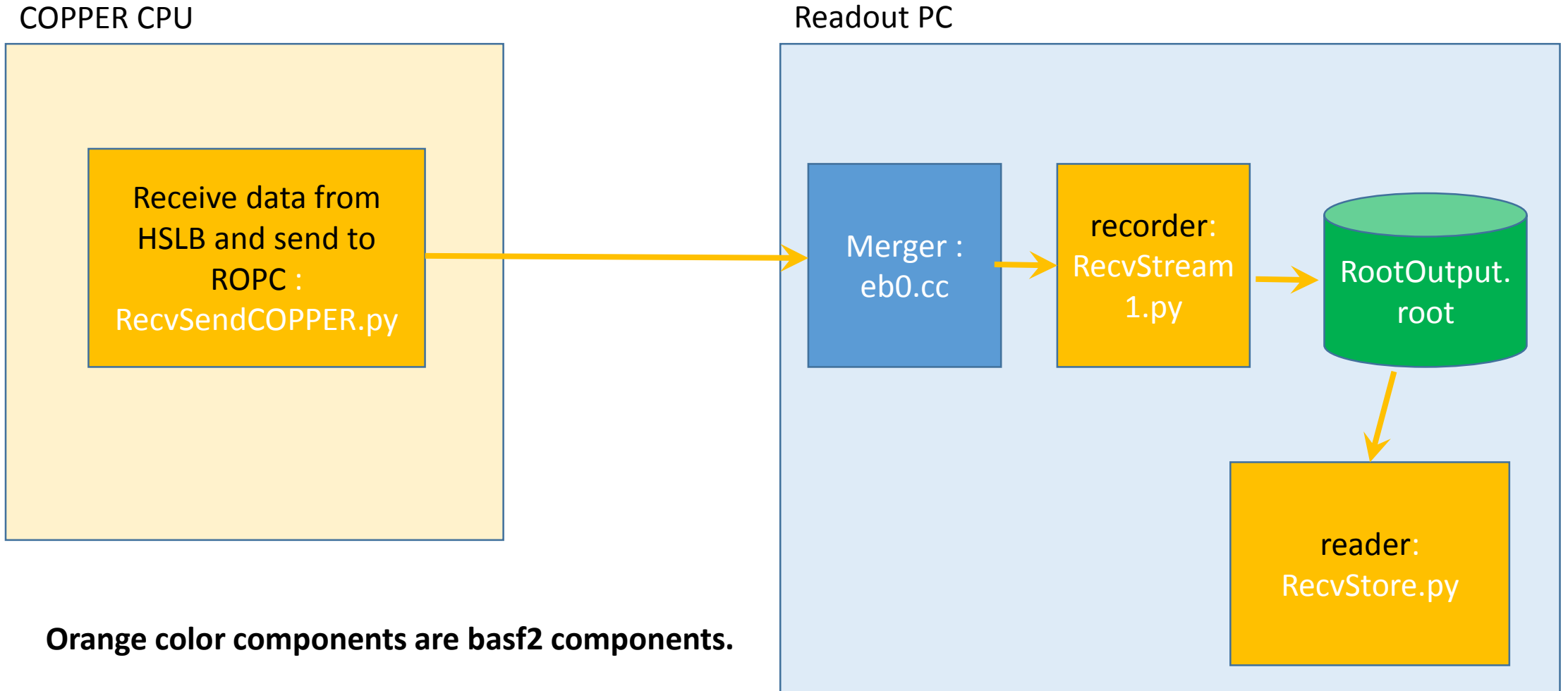
Satoru Yamada

Connection between components



Software components

These are stored under <https://belle2.cc.kek.jp/svn/trunk/software/daq>



Pocket DAQ w/o slow
controller and GUI

0. Before using pocket DAQ

- Setup a PXE boot server for COPPER CPU and install driver for COPPER etc
 - See <https://belle2.cc.kek.jp/~twiki/bin/view/Detector/DAQ/PocketDAQ>
- Install basf2 on both COPPER CPU and Readout PC
 - See <https://belle2.cc.kek.jp/~twiki/bin/view/Computing/SoftwareInstallation>
- For a release/daq directory, checkout the latest revision.
 - Check daq/Sconscripts and check if `env['CONTINUE'] = False` is commented out.
 - Compile with `scons`.
- Compile eventbuilder
 - `cd ${BELLE2_LOCAL_DIR}/daq/eventbuilder/evb0/ ; gmake eb0`

Install GLEW for running basf2 program

After external ver.0.5.0?, basf2 requires GLEW(The OpenGL Extension Wrangler Library).

The following procedure worked fine.

1, install the yum epel repository.

```
ropc01$ wget http://ftp-srv2.kddilabs.jp/Linux/distributions/fedora/epel/5/i386/epel-release-5-4.noarch.rpm
```

```
ropc01$ ssh cpr** -lroot
```

```
cpr**# rpm -ivh epel-release-5-4.noarch.rpm
```

```
cpr**# ls -l /etc/yum.repos.d/
```

```
cpr**# exit
```

2, install glew on COPPER

```
ropc01$ sudo yum install glew --installroot=/tftpboot/copper/root
```

0.5 Set parameters(1)

```
[ROPC] % cd ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts
```

```
[ROPC] % emacs run_start.sh
```

Set arguments of start_copper.sh

Usage : start_copper.sh **<HOSTNAME> <COPPER node ID> <FINNESSE bit flag: A=0x1, B=0x2, C=0x4, D=0x8> <Use network shared memory : Yes=1, No=0>**

e.g.)

```
/usr/bin/xterm -fn 7x14 -geometry 102x10+0+642 -e ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/start_copper.sh cpr006 1 1 0 &
```

```
/usr/bin/xterm -fn 7x14 -geometry 102x10+750+642 -e ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/start_copper.sh cpr007 2 3 0 &
```

<COPPER node ID> will be attached to RawCOPPER header.

```
[ROPC] % emacs start_eb0.sh
```

Set arguments of start_eb0

Usage : eb0 -n **<# of COPPERs> <COPPER HOSTNAME1> <COPPER HOSTNAME2> ... <COPPER hostname n>**
-b(send data to downstream basf2) -D(not send data to HLT)

```
/usr/bin/xterm -fn 7x14 -geometry 102x10+0+342 -e ${BELLE2_LOCAL_DIR}/daq/eventbuilder/evb0/eb0 -n 2 cpr006 cpr007 -b -D
```

0.5 Set parameters(2)

NOTICE : Environment variables for basf2

In `${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/copper.sh`, a line, “source ~/.bash_profile”, is for setting up basf2 environment. You need to add basf2 setting commands in your `.bash_profile` (or other script file).

Please see "Setup of Software Tools" at <https://belle2.cc.kek.jp/~twiki/bin/view/Computing/SoftwareInstallation> for details.

```
[ ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/copper.sh ]
```

```
#
```

```
# setup basf2 environment (See "Setup of Software Tools" at
```

```
https://belle2.cc.kek.jp/~twiki/bin/view/Computing/SoftwareInstallation
```

```
#
```

```
source ~/.bash_profile
```


0.5 Set parameters(3)

Specify whether communication with GUI via shared memory will be used or not :

1, Edit \${BELLE2_LOCAL_DIR}/daq/rawdata/examples/RecvStream1.py

If you use GUI, set 'UseShmFlag' = 1

`receiver.param('UseShmFlag', 1)`

If you don't use GUI set 'UseShmFlag' = 0

`receiver.param('UseShmFlag', 0)`

2, Edit \${BELLE2_LOCAL_DIR}/daq/rawdata/examples/RecvStream1.py

If you use GUI, set 'UseShmFlag' = 1

`receiver.param('UseShmFlag', 1)`

If you don't use GUI set 'UseShmFlag' = 0

`receiver.param('UseShmFlag', 0)`

1, How to start DAQ

```
[ROPC] % cd ${BELLE2_LOCAL_DIR}/release/daq/copper/daq_scripts
```

```
[ROPC] % ./run_start.sh
```

2, How to stop DAQ

- No stop button for now (If you use GUI, you have.)
- You need to specify max # of events or time to stop the run on a basf2 python file.

```
ROPC % cd ${BELLE2_LOCAL_DIR}/daq/rawdata/examples  
Edit RecvStream1.py
```

You can set following paramters to stop a run.

```
receiver.param('MaxTime', 300.)
```

```
receiver.param('MaxEventNum', 30.)
```

3. Read an output file and extract FEE buffer

- Output file name
 - `${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/root_output.root`
 - You can change the filename by editing `${BELLE2_LOCAL_DIR}/daq/rawdata/examples/RecvStream1.py`
- Read the root file
 - `[ROPC] % cd ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts`
 - `[ROPC]% ./ReadStore.sh root_output.root`
 - Modify `${BELLE2_LOCAL_DIR}/daq/rawdata/modules/src/PrintData.cc` as you like.

You can obtain a pointer to FEE data like this in event() function of PrintData.cc.

```
int* finnesse_buf_1st;
```

```
...
```

```
int* finnesse_buf_4th;
```

```
finnesse_buf_1st = rawcprarray[ j ]->Get1stFINNESSEBuffer(i);
```

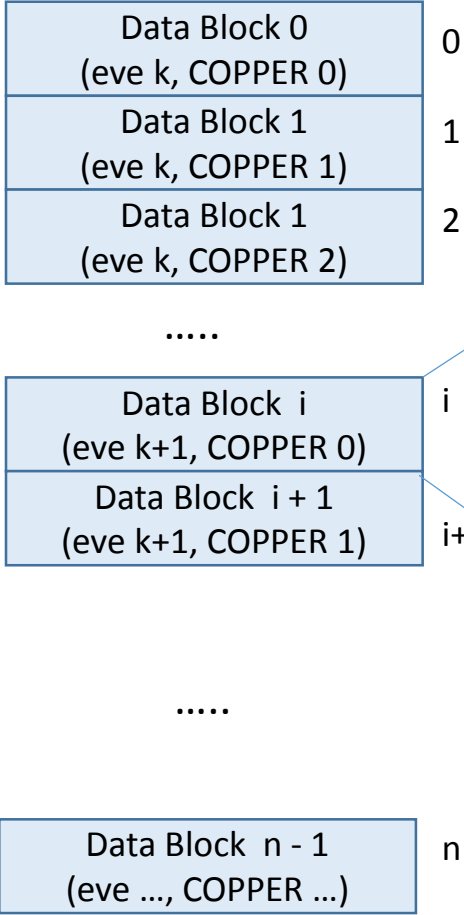
```
...
```

```
finnesse_buf_4th = rawcprarray[ j ]->Get4thFINNESSEBuffer(i);
```

See next page



Structure of RawCOPPER's buffer

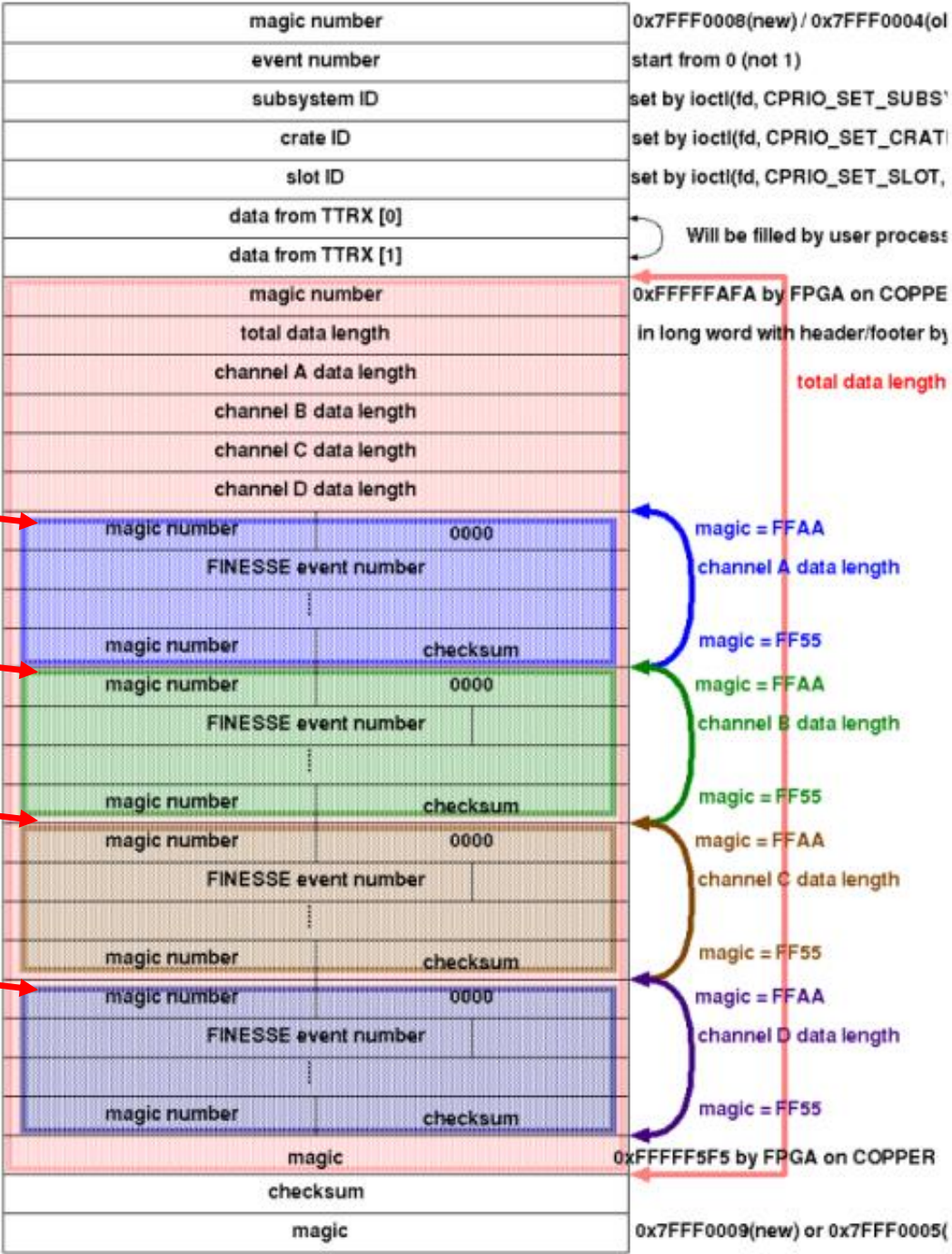


Get1stFINNESSEBuffer(i)

Get2ndFINNESSEBuffer(i)

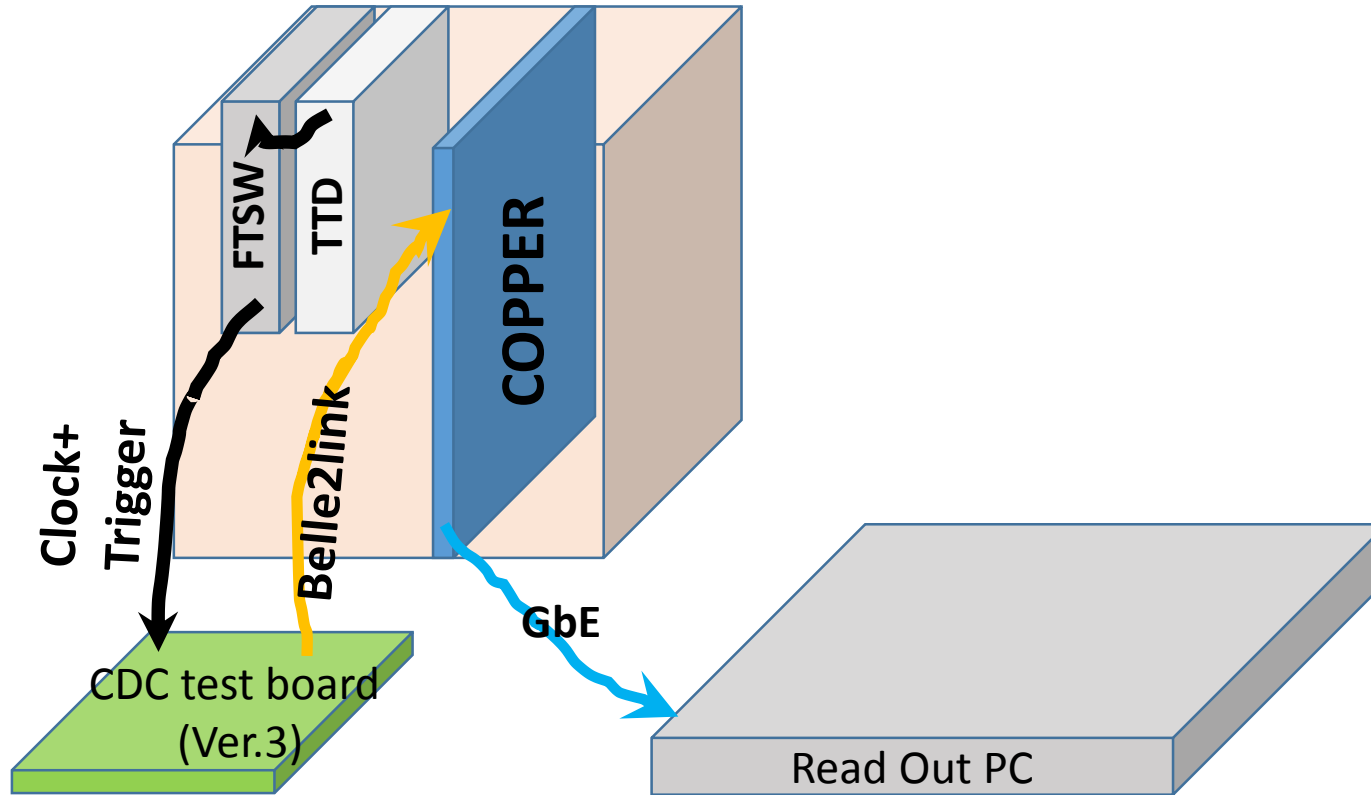
Get3rdFINNESSEBuffer(i)

Get4thFINNESSEBuffer(i)



End

Test bench at Tsukuba B3



Revision history

- 2013/10/16
 - Change explanation about option of run_start.sh script