Belle2link as a user

Mikihiko Nakao (KEK-IPNS)

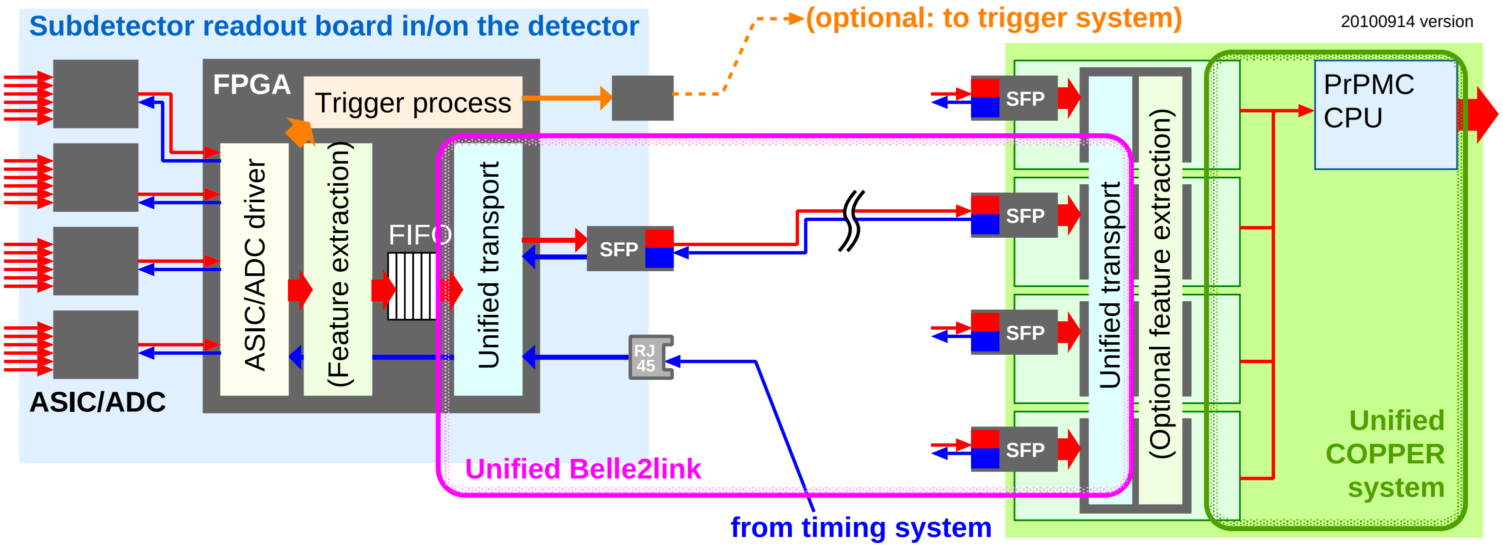
`mikihiko.nakao@kek.jp`

January 17, 2012

Outline

1. **Introduction**
2. **Status of Hardware, Firmware & Software**
3. **How to run the setup at KEK**
4. **How to proceed with a new board**
5. **Wishlist**

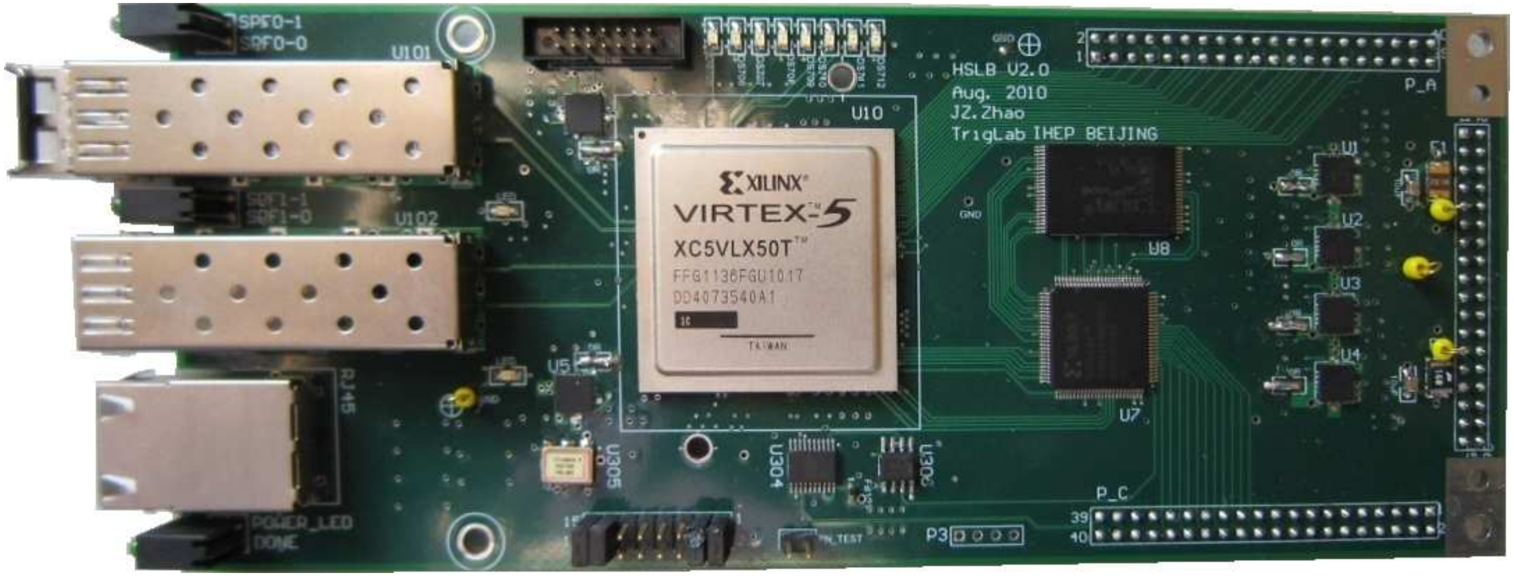
What is Belle2link



20100914 version

- (part of) **Firmware on front-end board**
  (to be integrated with detector dependent firmware)
- **Receiver FINESSE (HSLB)**
- **Firmware on HSLB**
- (part of) **Software to control HSLB**
  (part of) **Software to read data**
  (to be integrated with a DAQ program on COPPER)

The HSLB



- **10 boards produced by IHEP and brought to KEK**
- **10 more to be produced soon** (mini production)
- **700 of them needed** (everybody except PXD/B-PID)
- **Virtex5-LX50T is a bit overspec and costly, but plan is to go with it**

Firmware on HSLB

- **Driven by an external system clock** (or by an on-board Xtal)
- **Two SFP ports for RocketIO** (only one will be used)
  - **Bi-directional link, can write/read register data of the frontend board**
  - **FINESSE register address space is very limited** (A7D8)**, not large enough for direct mapping** (even for CDC)
- (Probably) **not to be touched by detector groups unless data manipulation is needed**

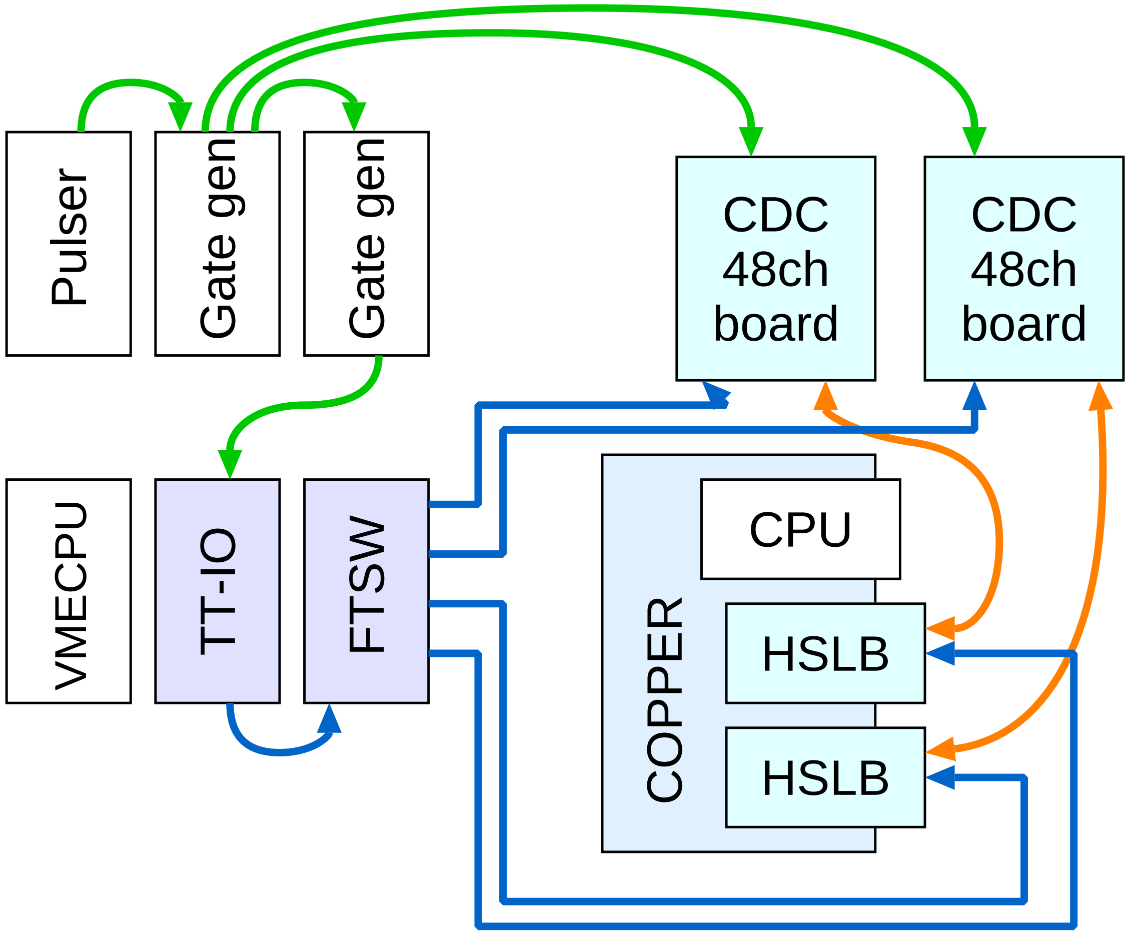
Software for HSLB

- **Two device drivers**
  - **fngeneric:** (T.Higuchi)
    to control HSLB registers and frontend registers through HSLB
  - **copper:** (S.Y.Suzuki)
    to readout data transmitted to HSLB through Belle2link
- **To simple programs for control and readout CDC board**
  - **record: simply open and read copper device and write data in a text format**
  - **B2LDo: HSLB control**

Firmware on frontend board

- **CDC firmware**
  - Based on CDC standalone firmware which reads out data using SiTCP
  - ADC and TDC data handling, (part of) ADC and ASD control
- **dummy data firmware**
  - For the CDC board, but nothing taken from the CDC firmware
  - Dummy data (1024 byte, 0..255 incremented 4 times)
  - Should work on any Virtex 5LXT based board by tailoring I/O ports (but not for Spartan 6 based boards)
- **In both cases, prototype belle2tt firmware is integrated**

Setup



- **Pulser + gate generator for a test pulse and a delayed trigger**
- **TT-IO as NIM to LVDS/RJ-45 converter and trigger gate**
- **FTSW to distribute clock and trigger**
- **Two boards read out together**

How to use (1)

1. **Connect all the cables**
2. **Power on COPPER, power on CDC board**
3. **Download HSLB firmware, download CDC firmware**
   HSLB firmware should be stored on the board
4. **Establish the “link” of Belle2link**

```
cpr%  ./B2LDo -d link
```

- ⚠ not plug‘n’play, and if fails, need to download CDC firmware again
- ⚠ has to check with LED light, should be checked with software

5. **Check the connection by reading out frontend registers**

```
cpr%  ./B2LDo -d checkfee
```

- ⚠ it returns some invalid cached data even if link goes down

6. **Close the gate of the trigger**

```
tt%  regio ext 0
```

How to use (2)

7. **Set up all CDC parameters** (window, delay, threshold, etc)

```
cpr%  ./B2LDo -d -p window 40
cpr%  ./B2LDo -d -p delay 5
cpr%  ./B2LDo -d -p tdcth 3700
cpr%  ./B2LDo -d verbose
cpr%  ./B2LDo -d trigger
```

- ⚠ can’t read back what was set to the frontend
- ⚠ more (difficult) parameters are still missing, those which need SPI based ADC control

8. **Start readout program**

```
cpr%  ./record1
```

- ⚠ This program needs to be tailed for your data format

9. **Turn on trigger**

```
tt%  regio ext 4
```

How to initialize the system

- **Currently we do not have foolproof initialization scheme**
- **Here’s my wishlist**
  - **Once RocketiIO link is lost, it should automatically try to reset to reestablish the link**
  - **Link reset should be also performed from the timing link**
  - **Link and other status should be monitored from timing link**
  - **Event number has to be reset upon run start**
  - **Belle2link should also monitor the timing link status**

How to quantify the link quality?

- **While we had the power supply voltage problem, it was not clear for us about the** (rather unstable) **link quality**
- **Current ECL may have a similar problem**
- **Link speed — 2.54 Gbps / 3.175 Gbps?**
- **Need a standardized IBERT** (i bit error rate test) **set up and/or integrated PRBS** (pseudo random bit sequence) **code into the firmware to quantify the link quality**
  - PRBS-xx with xx>7 generates a bit-stream which has a more tighter condition than 8b10b encoded bit-stream, and bit-error rate with PRBS (with several xx) will be a good measure
  - It's not only for when setting up a new board, link quality should be quantified during the installation, commissioning and regular running

How to proceed with new board

- **Start from a dummy data firmware**
  - Should work on any Virtex5 based board (but why not for ECL?)
- **Key components**
  - **RIO_trans, RIO_receiver** (no source files!)
  - **RIO_frame_gen** (dummy data generator)
  - **Para_decode** (for slow control, need to be tailored)
  - **CDC_RIO** (RocketIO I/F, shouldn't have CDC in the name)
  - **belle2clk, belle2trg** (from FTSW)
- **Data interface**
  - **SiTCP_TX_WE** (write enable) **& SiTCP_TX_WD[7:0]** (write enable)
  - ⚠ **Is 8-bit data out OK? Shouldn't it be better with 16-bit for some?**
- **Slow control interface**
  - **RW, EN, DATAi[7:0], DATAo[7:0], ADDR[6:0]**

HDL choice, ISE choice

- **VHDL vs Verilog**
  - (I think) **VHDL is more popular than Verilog in Belle II community, but CDC firmware is written in Verilog**
  - **Therefore(?), current CDC firmware is written in mixed VHDL & Verilog, with the top level written in Verilog**
  - ⚠ (I think) **it's better to provide the top level example in both**
- **ISE version**
  - **Initially, Beelle2link was developed with ISE11.3** (outdated)
  - **Moving around versions is not always trivial**
  - ⚠ (I think) **Belle2link should be supported on stable (12.4) and latest (13.3 for now) versions**
- **Coding standard** (suggestion by Igor)
  - http://www.ohwr.org/projects/hdl-core-lib/documents
  - **I wouldn't take it for the moment**

Need more address space!

- **CDC firmware handles A32D8 address/data for control**
  - A7D8 of Belle2link is not enough, despite we though CDC board does not have many parameters to fill it
  - Current CDC firmware cannot set the data output phase of the ADC chips, and cannot fully readout data
- ⚠ **Suggestion: general A32D8 register I/O on top of A7D8**
  - 4 A7D8 registers for address: **A31_24, A23_16, A16_8, A7_0**
  - 1 A7D8 registers to write: **DATAo**
  - 1 A7D8 registers to read: **DATAi**
  - 1 A7D8 registers to stream write: **DATAso**
  - 1 A7D8 registers to stream read: **DATAsi**
  - 2(or 4?) A7D8 registers to stream length: **DATAlen**
- **Could be implemented for CDC, and helpful for others**
  - ⚠ better to interface LOC_REG instead of Para_decode setting registers

On Spartan6

- SVD and ECL (and BPID) use Spartan6 for the frontend board
- SVD board (FTB) is ready and we'd like to start testing
- SVD group has man-power, and has offered to take the task to port it to Spartan6
- I received an email from IHEP that they'd like to keep responsibility for Spartan6, too
- ⚠ But then what is the schedule?

Software on COPPER

- **Software on COPPER has to be integrated with a run-control and data transmission program**
- **Probably taken care by KEK?**

Summary of wishlist

(in the order of urgency)

- **Spartan6 support of Belle2link**
- **Source file for Belle2link**
- **Link-quality evaluation / monitoring scheme**
- **More proper start-up scheme**
- **A generic slow control** (remote register I/O) **scheme**
- **Package consists of a library** (instead of/in addition to an example)**, software and document**