Updates on Caribou

WG vertex and tracking detector technology meeting

27 March 2020

Tomas Vanat
tomas.vanat@cern.ch

Outline

- Firmware modules
  - Pattern generator
  - Test pulse generator
  - Timestamp generator
  - Data buffer/merger
- New CLICpix2 FW
- WiP: SW upgrage
  - Peary modifications
  - Upgrade to latest Poky
- WiP: HW upgrade

New Firmware modules

- Whole firmware is chip-specific.
  - Different speeds, protocols, control/status signals, …
- Similar tasks/control sequences are performed
- Two basic approaches:
  - Frame-based / triggered *(CLICpix2, CLICTD)*
  - Continuous (data-driven) readout (ATLASPix)
- Firmware can be built from configurable and reusable blocks + a little of a custom logic
  - Timestamp generator
  - Data buffer
  - Output pattern generator
  - …

Pattern generator

- Inspired by original CLICpix2 wave generator by Adrian Fiergolski
  - With a few new features added
- Can generate an output sequence (waveform) on multiple channels
- Transitions between output states can be executed
  - After a defined time
  - After a logic condition is met
  - Combination of both
- Outputs and trigger inputs are fixed in FW
- Sequence and transition conditions are defined in SW
- Can run in an infinite loop or for a defined number of cycles (SW config.)

Pattern generator configuration file

- One pattern per line, whitespace-separated columns
- Time in clock cycles
- Inputs: fixed order defined in FW: `High, Low, Rising, Falling, any Edge, Always, Never`
- Outputs are defined in a header file

```
# IN(TRG,RD) GLOBAL_COND    TIMEOUT  OUTPUTS   DURATION
  L,A        AND                 0  PWR            100
  N,N        TRUE                0  PWR,SH           1
  N,R        OR          300000000  PWR,SH         160
  N,N        TRUE                0  PWR           1000
  N,N        TRUE                0  PWR,RO           1
```

Test pulse generator

- Can control an on-CaR-board analog pulser or generate a digital output only, e.g. for CLIC(pix2|TD) with on-chip pulser
- Defined in SW:
  - Time of pulse active
  - Time of pulse idle
  - Number of repetitions
  - Idle state high or low (output invert)
- Can be started e.g. from SW or by a pattern generator

Timestamp generator

- Counts clock cycles
- Has programmable number of inputs (defined in FW)
- SW can set active edge on each input that generates a trigger
- Counter value and input states are stored to FIFO when trigger occurs

Data buffer/merger

- For frame-based chips
- Prepares a data frame package and stores it to a FIFO so the DAQ can keep going and does not need to wait for SW to read the data.
- After a trigger (e.g. readout end), it stores to a FIFO:
  1. A header with total number of data bytes in data buffer
  2. A header with number of timestamp bytes in timestamp buffer
  3. Timestamp data
  4. Readout data
- SW can read frames asynchronously from the FIFO:
  - First header tells the total number of bytes to be read for a frame
  - Reads the number of bytes
  - Next frame follows

New CLICpix2 FW

- Replaced pattern generator with the new one
  - We needed a trigger for closing the shutter at DESY II testbeam
- New CP2 slow control interface
  - Before: a std. complex SPI core with Linux drivers for a non-std. CP2 SPI-like iface
    - Hacks (patches) were needed in both FW and drivers to fit the non-standard behavior
    - Patches needed updates with each new version of SPI core (Vivado) and Linux kernel (Poky)
  - Now: a simple SPI core controlled by memory-mapped AXI registers:
    - SPI address + data to be send (R/W)
    - Received SPI data (R)
    - Control register: start-read, start-write, reset, start-readout (W)
  - New SPI core can generate a readout command in FW
- Pattern generator can trigger a readout without SW intervention
  - Data buffer for asynchronous readout / continuous DAQ is implemented
- Not beam-tested yet

Work in progress: Peary modifications

- New commercial user/contributer: Fastree3d (A. Fiergolski)
- Separation of CaR board specific features in HAL
  - Easier adjusting of Peary for different hardware
  - by Simon Spannagel + Adrian Fiergolski
- Memory-mapped (FPGA FW) registers should be accessed in a same way as any other interface
  - Currently we use a different approach for FW registers
- Multiple interfaces in HAL
  - Current implementation allows only one interface to a device (e.g. I^2C)
  - CaribouDevice may need multiple interfaces
    - e.g. firmware registers and internal registers

Work in progress: Upgrade to latest Poky

- Latest Poky version is Zeus
  - meta-openembedded and meta-xilinx are available for Zeus
- Changes in libraries/headers and recipes structure
  - i2c-dev-user.h no longer available
  - Changes in device tree building (meta-xilinx)
- Changes in output files structure
  - Some scripts (e.g. prepare_sd) do not work anymore
- New I^2C kernel drivers
  - We may not need the patches anymore (T.B.C.)
  - Kernel panic error due to a race condition may be fixed (T.B.C.)

Work in progress: New CaR board (v 1.2)

- PCB design by BNL (Hongbin Liu, Elena Zhivun, Hucheng Chen)
- Voltage levels of TXB0304 will be set by a DAC from software
  - No more (solder) jumpers needed
  - CaR board will be more versatile
- A new power management circuit
  - Will allow to set an electronic fuse on the power supplies
  - Protects chips prone to latch-ups
- Will require software modifications – breaks backward compatibility
  - Automated identification of the board version via on-board EEPROM
- Mechanical dimensions stay the same