CMX online software status

Duc Bao Ta **MSU**

L1 Calo Workshop 19th May 2014

Overview

- **CMX timing software**
  - Timing procedure
- **CMX online software**
  - CMX package
  - Bit manipulation class, data formats class
  - JEM jet simulation

CMX timing software

- **CMX timing procedure**
  - Finding the center of the delay setting window for each data line and channel using patterns or pseudo-random data
    - Delay scan over 31 taps for data lines
      - Can use the same delay setting for all data lines
    - Delay scan over 30 taps for clock lines
    - Check data error flag for each data line
    - Store “best” delay setting for each data line in database
  - Long term tests on delay settings or verification of delay setting(s) from database
  - Supply additional tests for backplane tests
    - Record parity errors at the same time with the data error flags

CMX timing software

- **Manual setup still needed**
- Firmware configurations
  - CMX firmware needs to be manually configured via JTAG
  - JEM can be configured via command line, but need two versions for pattern and random
  - CPM configured via dbHDMC GUI
  - *Maybe will be alleviated by final (single) firmware versions*
- Setup before scan for synchronization
  - Reset TTC module
  - Synchronization via BC reset manually via dbHDMC GUI in the TTC crate
  - Start address finding for JEMs, setting start addresses for pattern comparison
  - Start address procedure for CPM using “random” resets and analysis of the captured data
  - Verification of synchronous reception of the patterns

CMX timing software

- CMX timing procedure software
  - Implemented
    - Select pattern (single or all) or pseudo-random data from software
    - Upload of patterns into spy memories
    - Software scans the delays of data bit and source-synchronous clock
    - Histograming of data comparison errors and parity errors versus delay, storing counters
    - Manual merging of cycles, external code to analyse histograms
    - Running long-term tests with simple delay settings
  - Todo:
    - Finding of best setting and storing into database
    - Reading from database
    - Integration into the TDAQ software as a "calibration run"

CMX online software

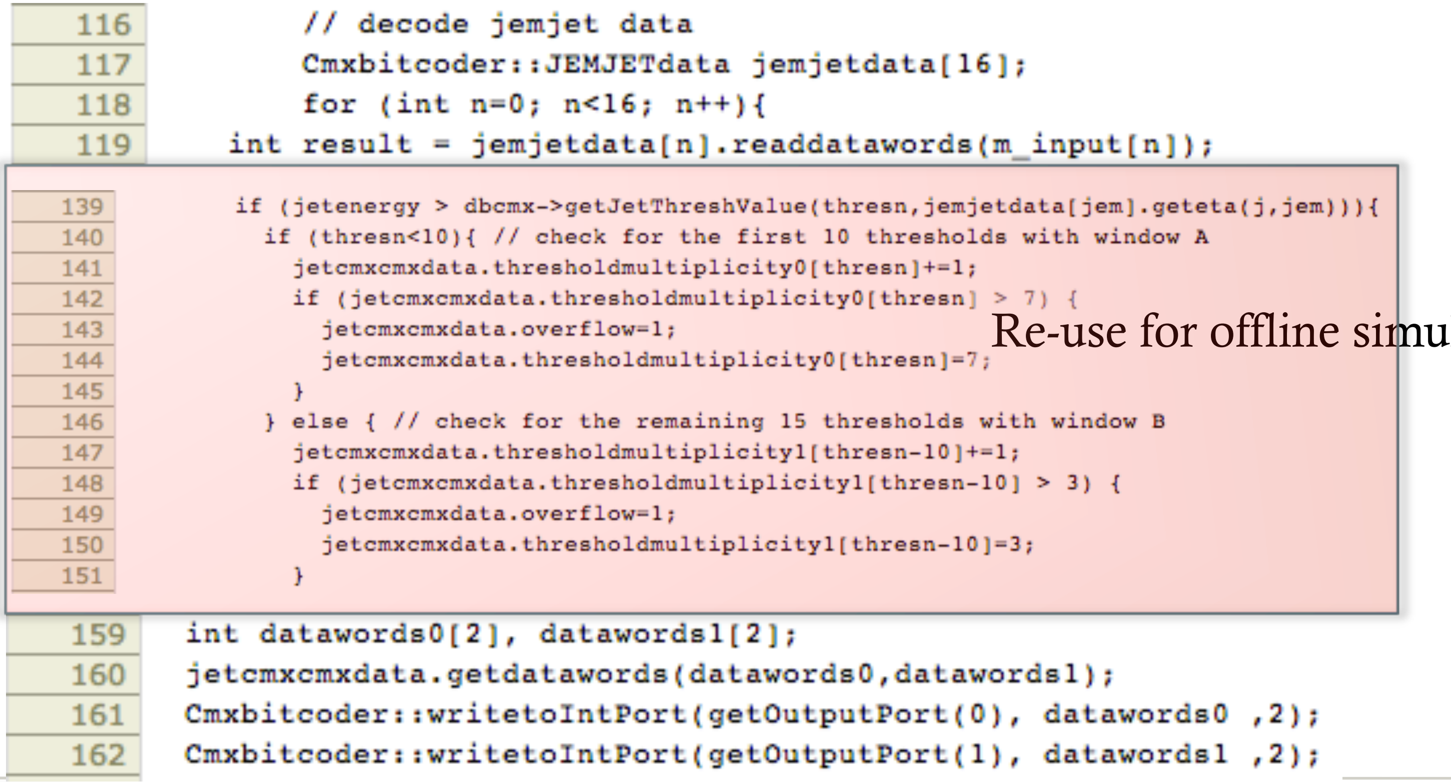
- **CMX online software package copy of CMM package**
  - Copy of CMM simulation
    - Basic structure available: cmxServices, cmxSim, cmxTests
    - Reading test vectors
    - Creation of test vectors, but no sensible patterns defined, yet
- **cmxServices = FPGA register map**
  - Update with FW development
  - Functions available for CMX tests:
    - setting delays (on 24+1 times 16 data/clock lines)
    - **reading/writing timing test patterns**

CMX online software

- **cmxSim**
  - Data formats implemented in classes
    - JEM (jet and energy sum), CP -> CMX
    - crate CMX -> system CMX
    - CMX -> ROS, CTP
    - Also partially support conversion into the other direction
      - Readability
      - Good for physics test vector generation
      - Data format coding from the other end

CMX online software

- **cmxSim**
  - Data formats implemented in classes



```
        // decode jemjet data
        Cmxbitcoder::JEMJETdata jemjetdata[16];
        for (int n=0; n<16; n++){
    int result = jemjetdata[n].readdatawords(m_input[n]);
```

```
          if (jetenergy > dbcmx->getJetThreshValue(thresn,jemjetdata[jem].geteta(j,jem))){
            if (thresn<10){ // check for the first 10 thresholds with window A
              jetcmxcmxdata.thresholdmultiplicity0[thresn]+=1;
              if (jetcmxcmxdata.thresholdmultiplicity0[thresn] > 7) {
                jetcmxcmxdata.overflow=1;
                jetcmxcmxdata.thresholdmultiplicity0[thresn]=7;
              }
            } else { // check for the remaining 15 thresholds with window B
              jetcmxcmxdata.thresholdmultiplicity1[thresn-10]+=1;
              if (jetcmxcmxdata.thresholdmultiplicity1[thresn-10] > 3) {
                jetcmxcmxdata.overflow=1;
                jetcmxcmxdata.thresholdmultiplicity1[thresn-10]=3;
              }
```

```
  int datawords0[2], datawords1[2];
  jetcmxcmxdata.getdatawords(datawords0,datawords1);
  Cmxbitcoder::writetoIntPort(getOutputPort(0), datawords0 ,2);
  Cmxbitcoder::writetoIntPort(getOutputPort(1), datawords1 ,2);
```

Re-use for offline simulation?

CMX online software

- **Online software development, finished tasks**
  - Thresholding of jets implemented
  - JEM system and crate merger implemented
  - **CMX timing software**

- **Online software development, todo (in order of priority)**
  - Update of CMM parts in the software
    - JET Daq merger
  - CMX test vector generation
  - CMX stand-alone test environment
    - Code is already available, need to adapt this
  - Update of CMM parts in the software
    - Energy sum merger, similar to CMM
    - CP thresholding and merging
  - CMX to Topo development

Same timeline as firmware development for M4

Summary

- **CMX timing software**
  - Implementation for test ready and successfully used
  - Need to complete connection to database
  - Need to make it more automatized
- **CMX online software**
  - Copy of CMM packages
  - Data mostly formats available, except Topo
  - Implemented functionality: thresholding of jets
  - Need to complete simulation chain (at least for jets) standalone and for firmware testing
  - Synchronize priorities with firmware development and deliveries for M runs