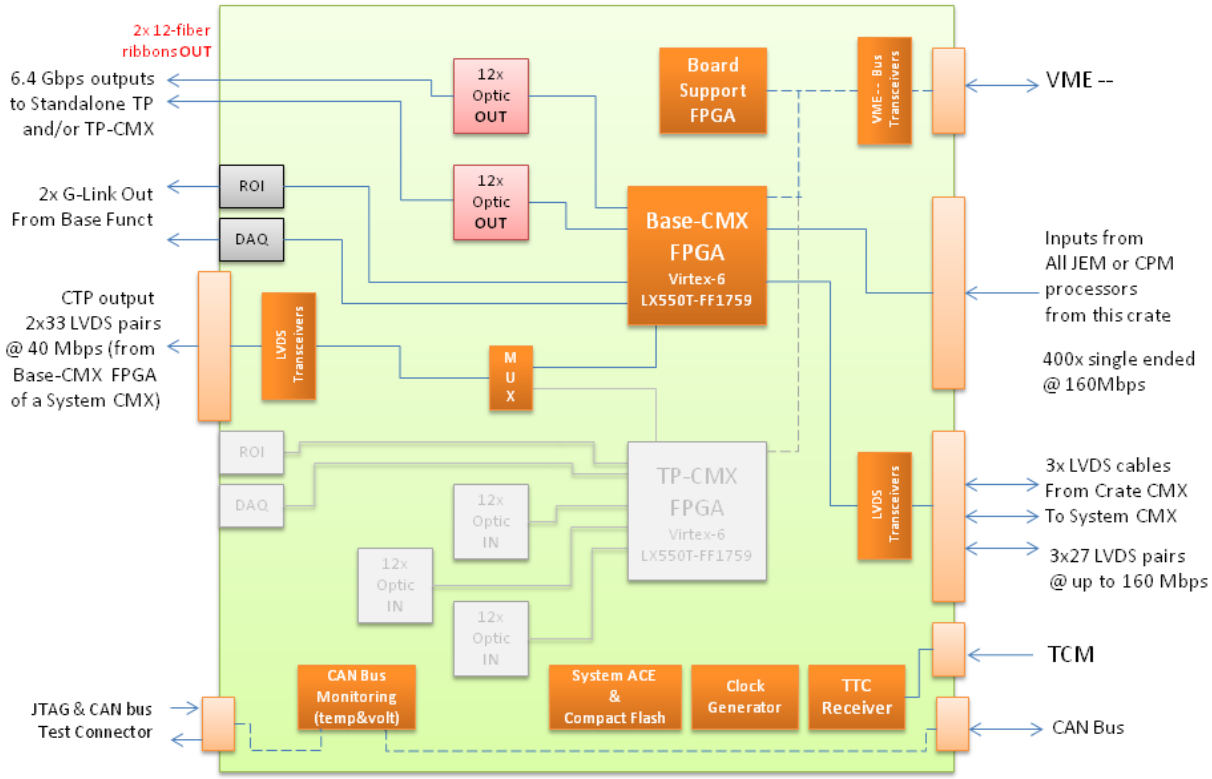
CMX status: Hardware, Firmware, Software

W. Fedorko

P. Laurens, D. Edmunds, Y. Ermoline, S. Caughron

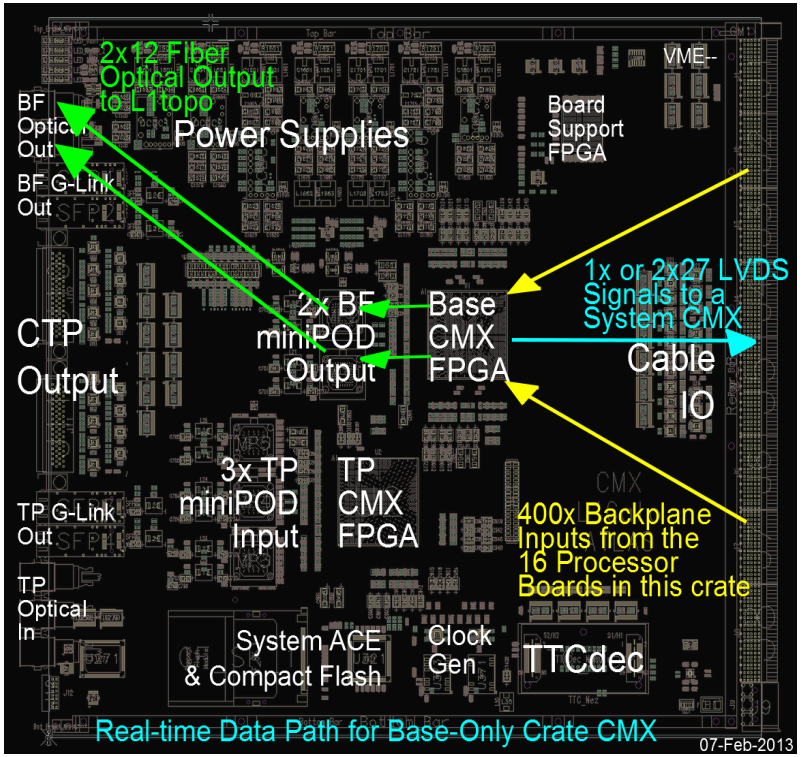
R. Brock, J. Linnemann,

CMX overview



CMX hardware status

- Prototype readiness review last Thursday, in progress
  - http://www.pa.msu.edu/hep/atlas/l1calo/cmx/specification/3_prototype_review/
- Design in good shape
  - Support for required interfaces
    - Backplane
    - Optical to L1Topo
    - LVDS (system <-> crate, CTP)
    - glink (readout)
    - TTC, VME, CAN, JTAG
  - Component placement and PCB layer organization
  - Mechanical design
    - Front panel
    - Stiffening bars/airflow



CMX hardware status

- Major work still needed
  - Pre-review design not routed
  - CF card placement on FP requires front panel re-work and re-placing components (3x12 VS 36 fiber connectors)
  - Topo ROD implementation on Topo FPGA
    - New design requirement!!
    - Need to study what is needed exactly
    - Clocking (100 and 40 MHz to BF and Topo MGTs and logic) (which ones?)
      - 40 MHz likely needed for glink
    - TP-BF traces
    - Two 2 Gbps (vs 1Gbps) traces from FP mounted SFP cages to BF and TOPO FPGA's
  - Mechanical test:
    - MiniPod height exceeds VME spec (0.7 mm into between-board space)
    - Test with Mechanical only model needed.
  - Add analog multiplexer to the CAN-Bus microprocessor for current readout
  - 2 or 4 LEMO connectors (ROD function+clock monitoring)

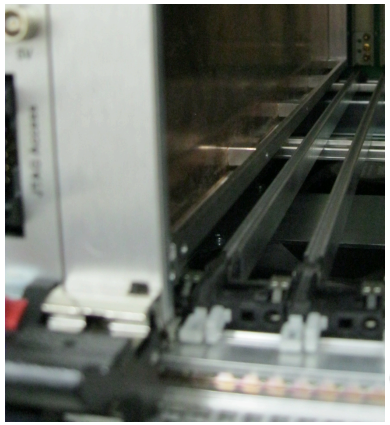
VAT Card

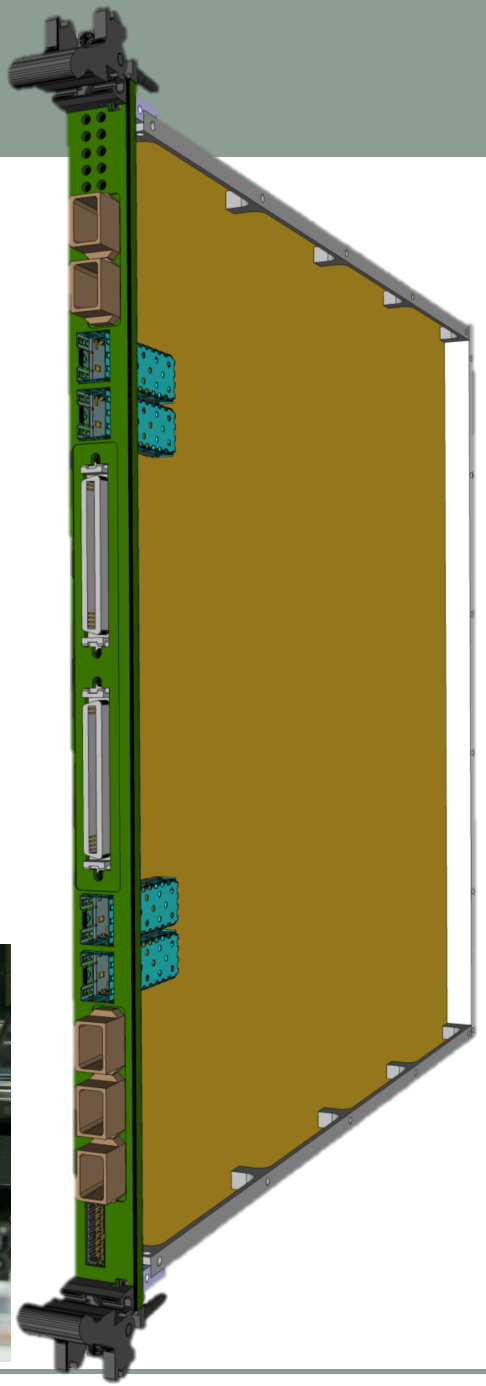
- The current status:
  - Spartan and Virtex configuration via JTAG tested
  - Virtex-6 configuration from CF card tested
  - VAT card operated in the MSU test rig – access from VME and configuration via IMPACT
  - VME interface tested in the VME crate for Spartan and Virtex (board support and BF)
  - VAT card operated from TTC clock
- Plans:
  - VME address map and registers/bits,
  - TTCrx chip access via I2C interface in Spartan
  - CF card access from VME via XILINX System ACE chip

Mechanical-only card

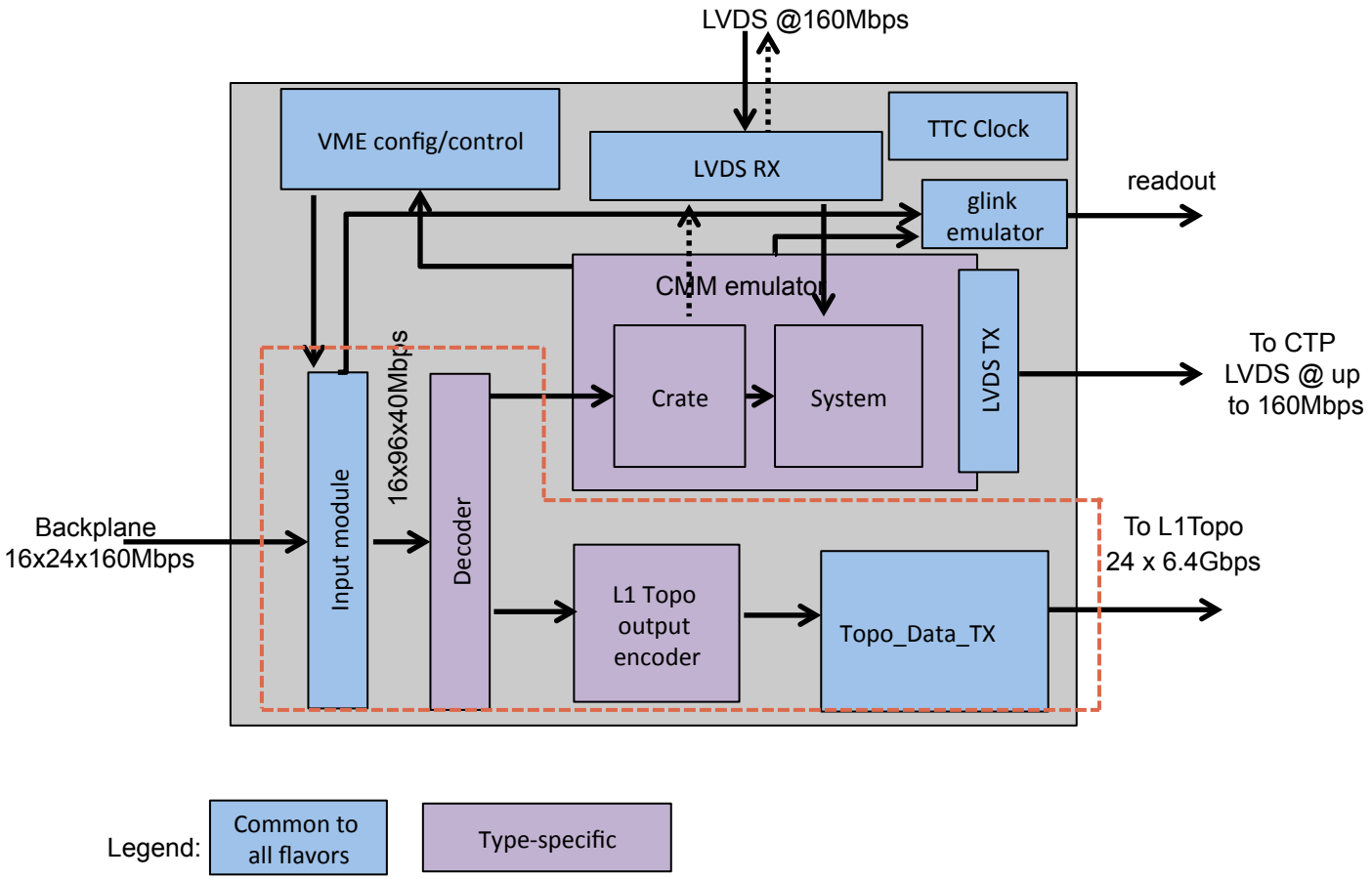
- Verified backplane connector placement
- Stiffener bar design provide insertion load transfer
  - Minimal airflow obstruction
- Minipod TXs will be glued to the model and testing for interference done in CERN rig
  - heatsink height over VME spec

    (0.7 mm into the in-between space)



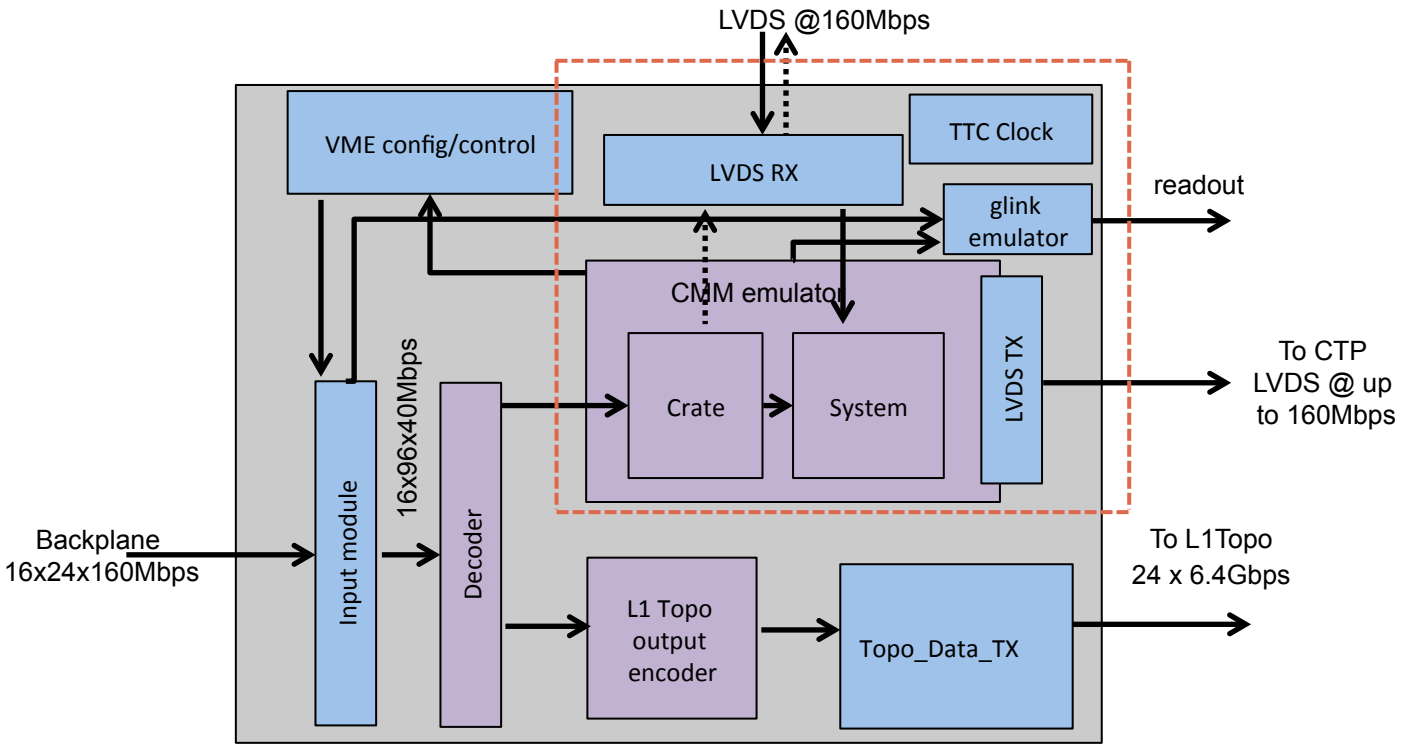


Firmware overview: Base Function FPGA



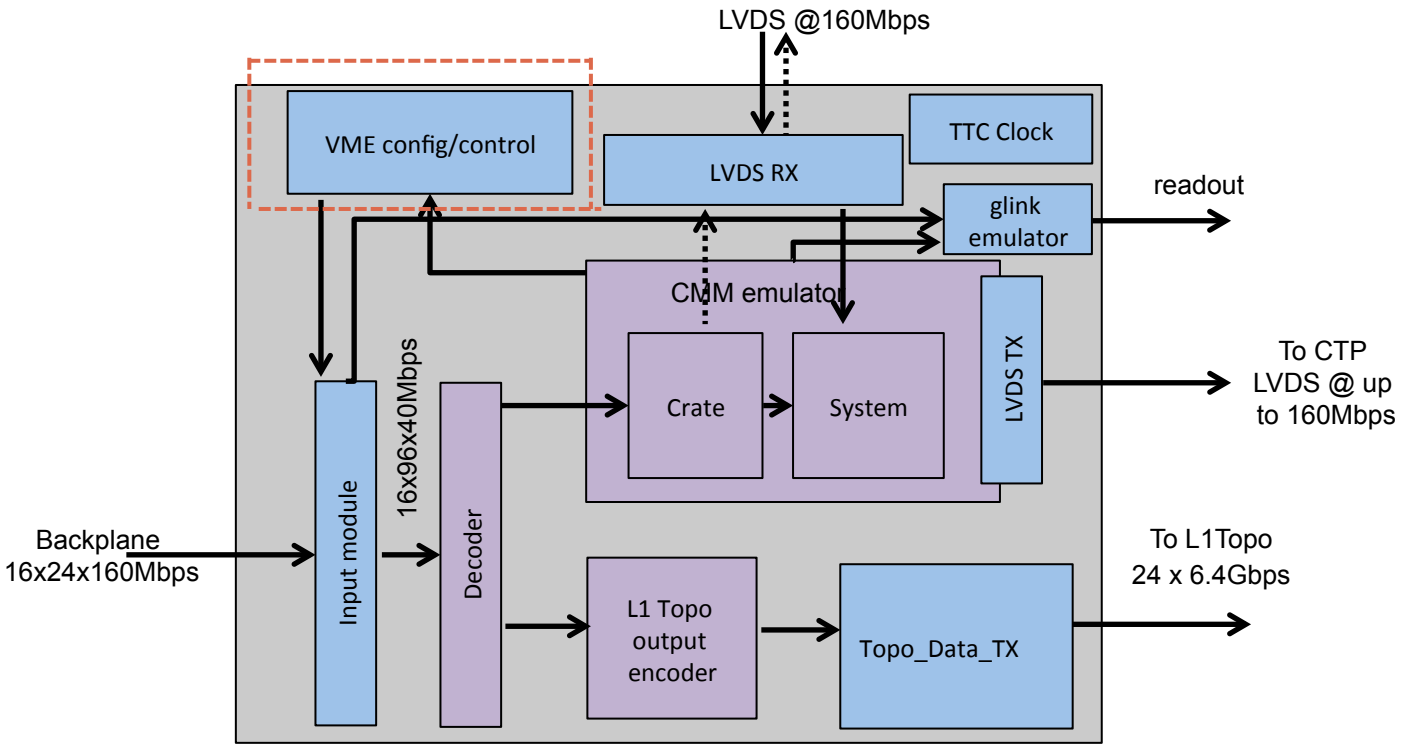
- Approach: Develop the common modules for the TOPO RT path and Jet CMX specific components first
- Wojtek: developing physical interfaces and encoding, Pawel: Decoder (0-suppression)

Firmware development: CMM emulator



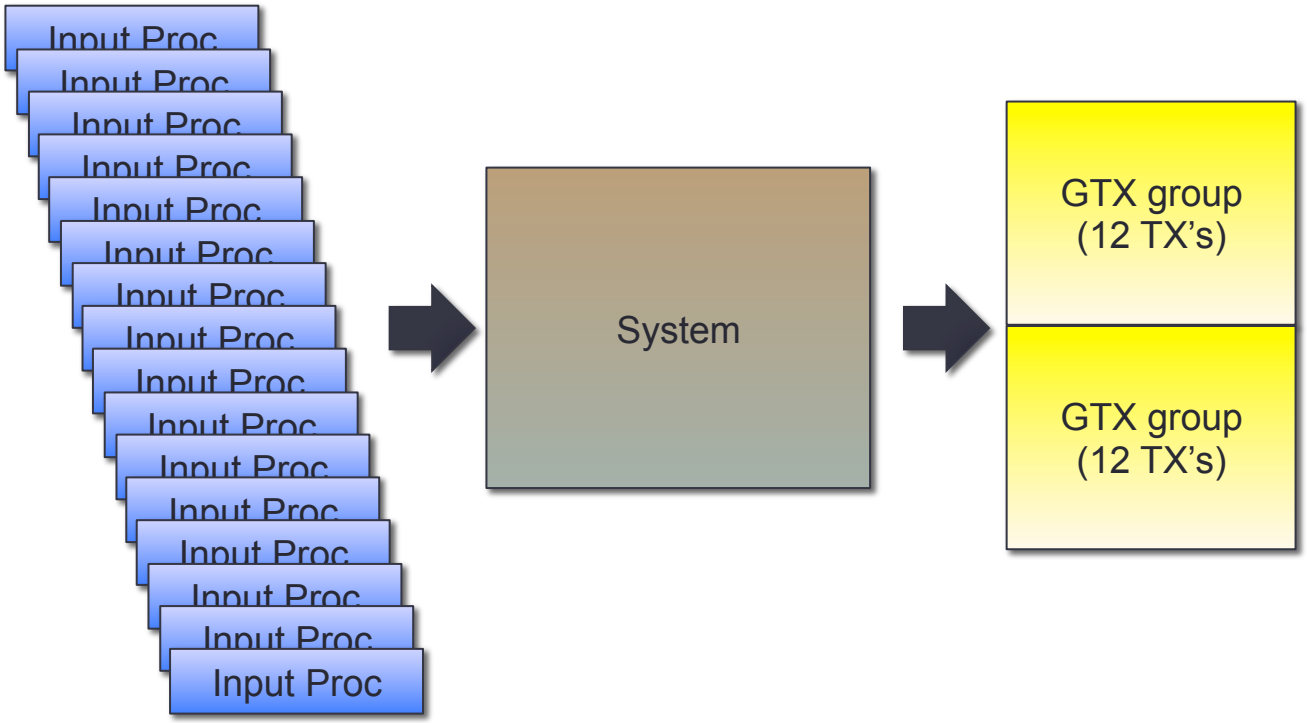
- Jet version developed by Pawel
- Needs an 'upgrade' ➔more thresholds
- Splitting out common sub-modules
- Other types will follow

Firmware development: VME config and control



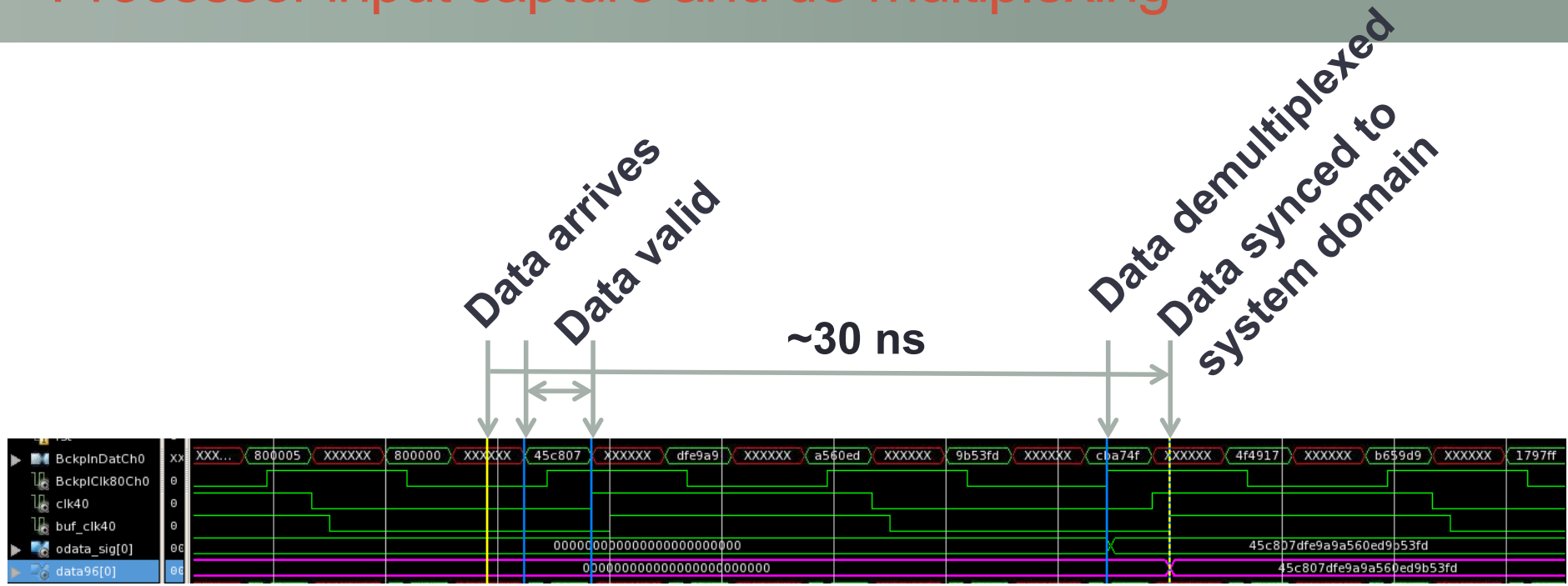
- Yuri is developing firmware in context of VAT demonstrator
- Identified implemented new registers (thresholds, IODELAY values)
  - More may be needed

CMX Base: RT Data flow, Clock domains



- 16 input processor domains
- 1 system domain (several clocks)
- 2 output domains (2 groups of 12 GTXs synchronised)
  - Reference clocks shared within these 2 groups
- 3 MMCMs (PLL circuits) (1 system, 2 GTX domains)

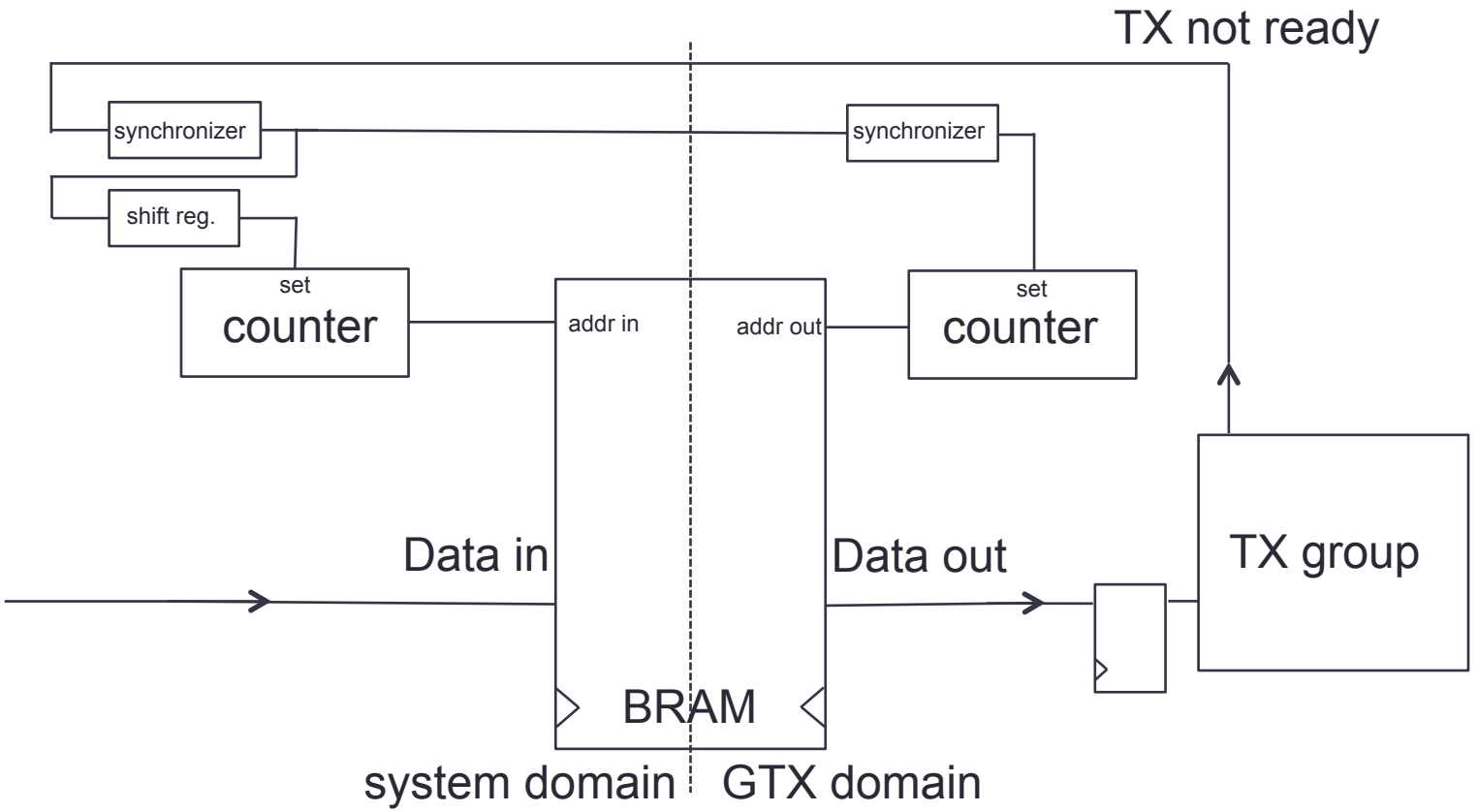
Processor input capture and de-multiplexing



- Data from processor inputs edge-aligned to DDR clock
  - Requires IODELAY delay of the clock (max shift 2.4 ns) to center
- Data captured and de-multiplexed to 40 Mbps in the source-sync domain.
- Timing analysis indicates robustness of data capture against clock jitter (~1ns) and small (50%) data validity window
- Requires a ‘training pattern’ from the processors at a start (sent only once).
  - Need to come to an agreement of when and how this is done.
- Domain crossing to system with a simple register
  - Phase relation encoded in the constraints
  - Need to implement programmable phase setting for the system MMCM

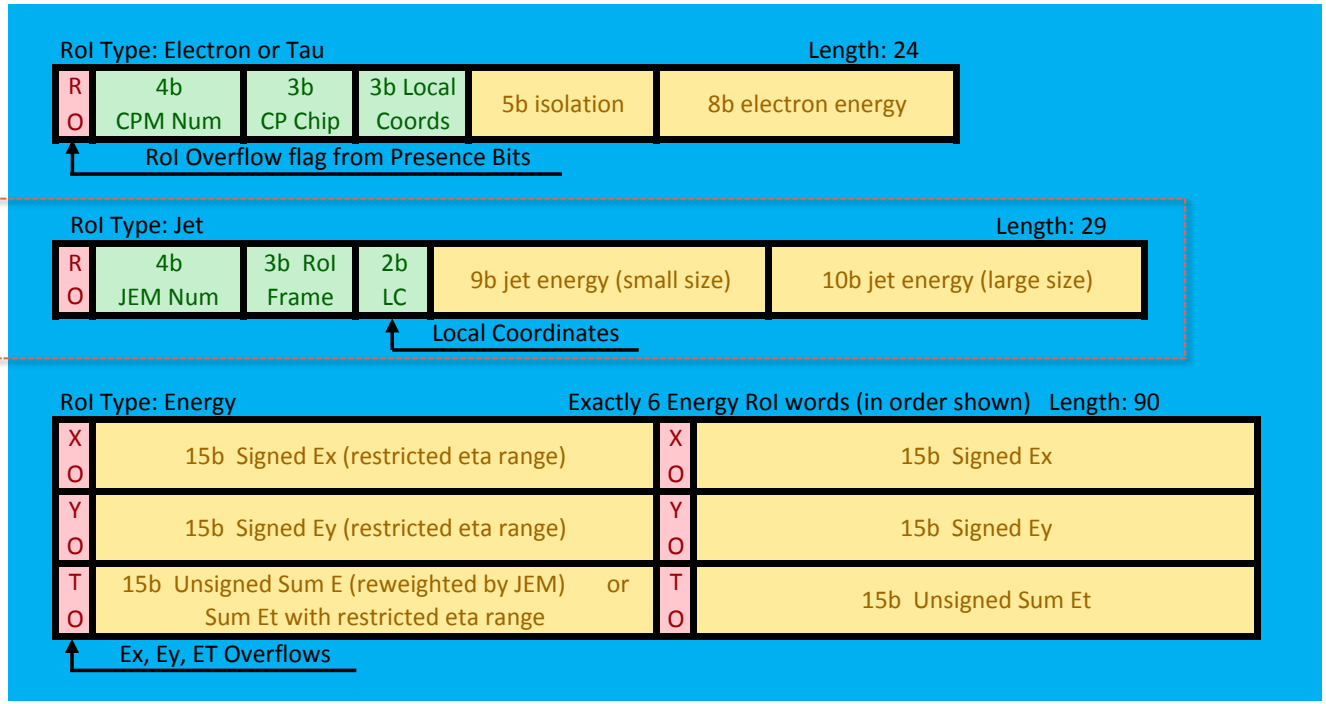
Topo TX, data serialization and domain crossing

- Data partially serialized to 320 Mbps in the system domain
- Domain crossing using BRAMs (two port config)
- XILINX FIFO routines too high latency ➔custom design



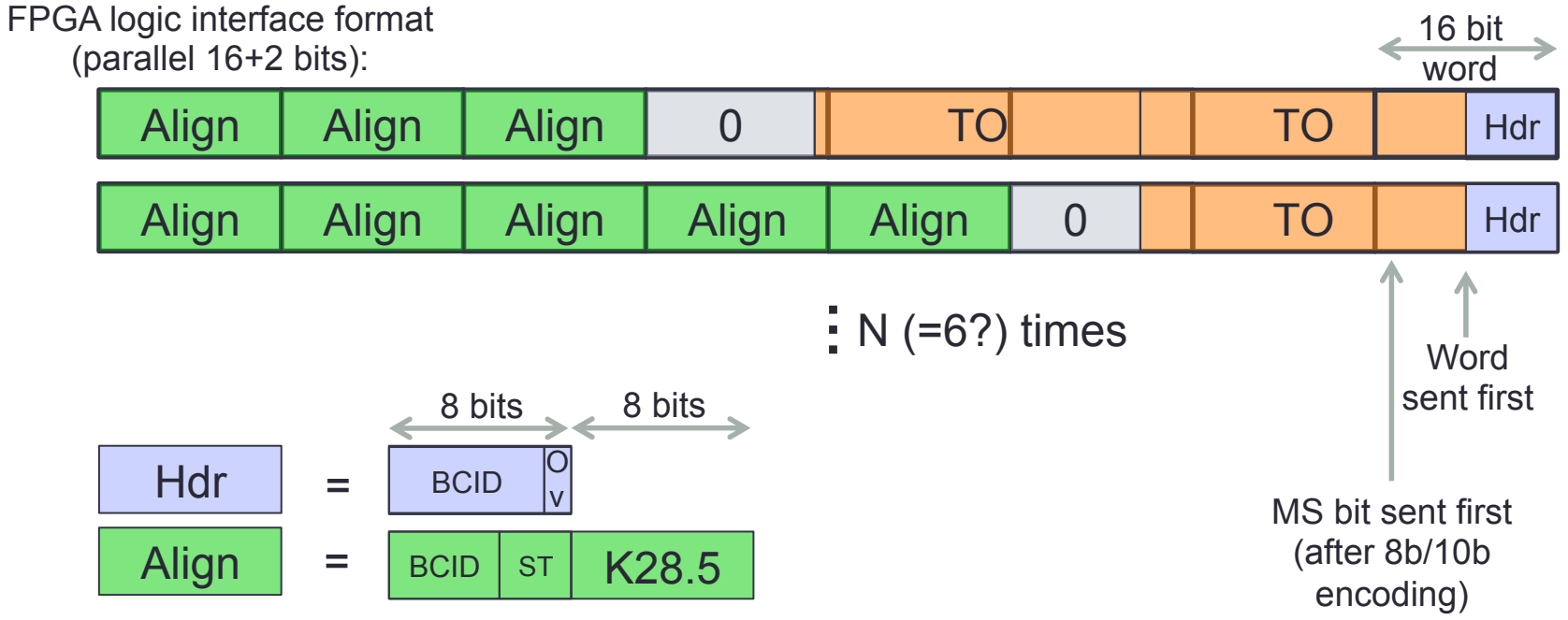
Comment on the Data formats

Steve's proposal



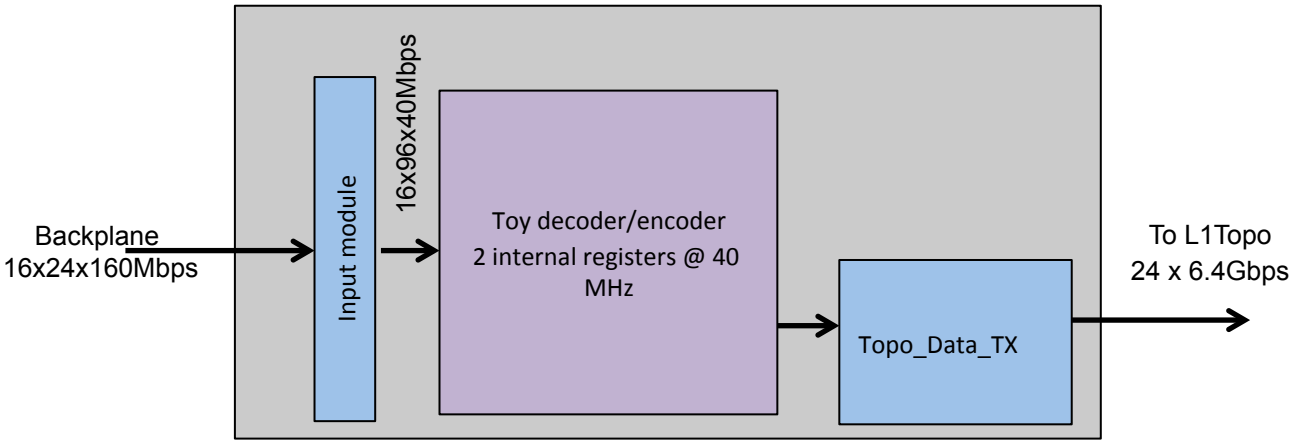
- Jet eta, phi decoded in CMX, 1 more bit needed
- Overflow needs not to be attached to every TO
- More information needed:
  - Always sending 128 bits/fiber/event. Parallel interface 16 bits x 8 @320 MHz
  - Need byte, word, subtick and event alignment information embedded in the data stream
  - Want to be sending this alignment information if we can for monitoring/sanity

Proposal for CMX ➔L1 Topo fiber format



- Header word:
  - One Overflow (Backplane or 0-suppression) (Ov in diagram)
  - BCID e.g. 7 LSBs
- TOs ‘front loaded’ onto TXs
- Opportunistic alignment on even bytes when word == ’0000’
  - Change ‘00’ to alignment word (BCID (5 bits), subtick(3 bits); K28.5)
- 0-padding to nearest word boundary after payload

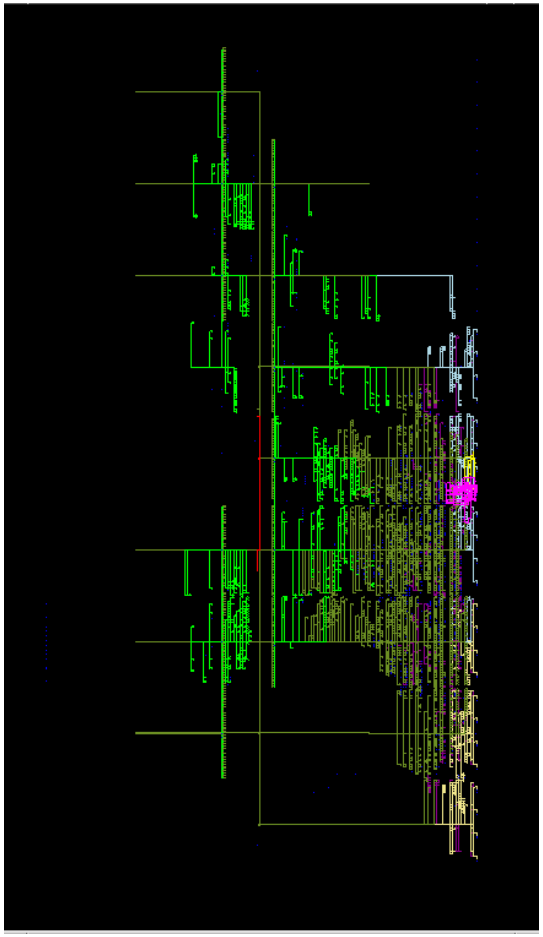
Topo path interfaces: putting it together



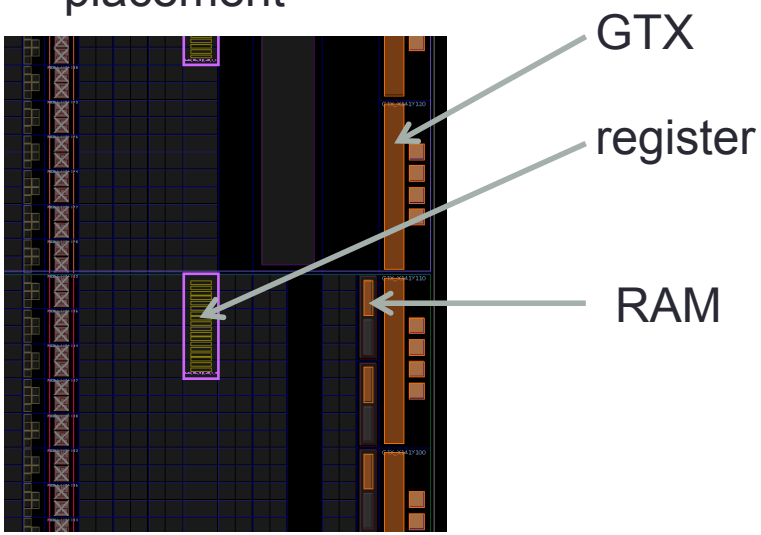
- Physical interfaces for the Topo path and domain crossing implemented
- Low configurable logic resource use (occupied slices 5%)
- 24/1264 BRAM18 used
- 16 BUFRs 5/32 BUFGs
- 3/18 MMCMs

Topo RT path timing

Clock networks in FPGA



RAM, register and GTX placement



- Some by-hand component placement required
- Pipeline registers needed in GTX logic (320 MHz)
- Duplicate registers removal off, register duplication on
- Timing satisfied but slack low <0.1 ns

Topo path latency

Stage	Latency
Input Capture and synchronization	~30 ns *
Decoding and ‘0 suppression’	~50 ns
Serialization to 320 Mbps and domain crossing to GTX	~12 ns **
GTX Serialization	~30 ns ***
Total	**~120 ns**

* Counted from the start of **first** word on the backplane

** delay for the first word to be ready on GTX TX input

*** delay for round trip FPGA parallel interface ➔ FPGA interface (for the first word) / 2;
based on simulation (no TX buffer, RX elastic buffer)

CMX firmware plans

- Ready for integration of of the Jet type Topo RT path
  - Work planned on sidelines of this meeting
- Re-organize/upgrade CMM emulation module and integration
  - ‘Unplugging’ common pieces into modules
  - Threshold support
- Testing firmware for prototypes
  - Based on the same modules as the Jet type
  - MGT RX implemented but not ‘activated’ in the TopoTX module
- Development of ‘logic’ modules for other CMX types (Base Function)
- Topo function FPGA
  - Testing firmware and data I/O will re-utilize same modules as BF
  - Currently no concrete plans for ‘logic’

CMX software status

- CMM software model cloned to CMX (Seth thanks to Murrough)
- ‘MSU crate’ up (at CERN)
- Partition with CPM and CMM in the crate.
- VME read/writes from registers
- Basic tests of loading data and reading data into/from playback memories.
- Need:
  - Establish playback of test data CPM to CMM
    - Static and changing data patters
  - Learn in detail how such tests are controlled/coded in the infrastructure
  - Extend framework to model CMX registers
  - Provide test infrastructure for CMX
    - essential for board/system comissioning
- Seth is back on the task

Conclusions

- Hardware well advanced
  - Layout
  - Mechanical
  - Significant work remains
    - Prototype readiness review held
    - Evaluating impact of new requirements
    - Cost/benefit analysis of proposed alternative solutions
- Firmware BF ready for 1st stage of integration (Jet type)
  - RT interfaces developed (L1Topo path)
  - Jet decoder/0 suppression
  - Integration starting here
  - Needs organization for commissioning support
- Software
  - Lot of work previously done
  - Effort re-started
  - Evaluating the needs for commissioning support

Backup

CMX schedule from joint meeting Oct '12