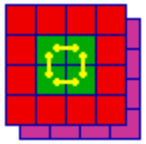
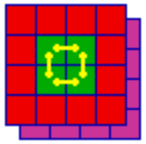


CMX: Update on status and planning

Yuri Ermoline, Wojciech Fedorko @CERN
Dan Edmunds, Philippe Laurens, Chip Brock @MSU
Michigan State University

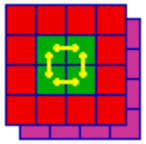
TDAQ Upgrade meeting, 7-Mar-2012

MICHIGAN STATE UNIVERSITY



CMX development work on 3 fronts

1. Engineering @MSU
   - Philippe, Dan, and Chip
2. CMX input module firmware @CERN
   - Wojtek and Yuri
3. VME/ACE/TTC (VAT) interface @CERN
   - Yuri

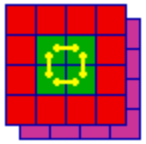


Initial CMX design studies phase

Addressed **two main questions:**

1) FPGA choice for implementing the Base-CMX functionality

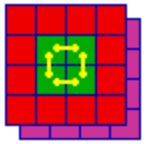
2) Strategies, costs and risks for adding Topological Processing capability to the CMX platform



Definition: **Base-CMX** functionality

1) **All of CMM functionality** (both Crate CMM and System CMM)
   - Receive and process 400 JEM/CPM input signals (@4x CMM rate)
   - All Crate CMXs send local summary to their System CMX through backplane connectors over LVDS cables (@higher rate than CMM)
   - System CMXs form and send triggering information to CTP over LVDS cables (same as CMM)
   - all CMXs send ROI and DAQ information over G-links (same as CMM)

2) **Send JEM/CPM info out to a TP** optically
   - using 12-fiber ribbons
   - with some level of duplication (at least 2x copies sent)
   - 6.4Gbps nominally sufficient for all raw data on **one** 12-fiber ribbon
   - also possible to send zero-suppressed data on less fibers

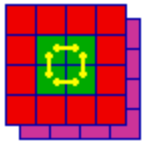


Definition: **TP-CMX** functionality

Topological Processing capability **on CMX platform**

1) Receive optical inputs from some/all 12x CMXs
2) Run multiple Topological Algorithms
3) Send Topological Triggering Information to CTP

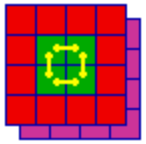
- TP-CMX functionality is needed if dedicated TP is not built, or availability delayed, or as additional safety/backup
- The CMX platform could then be asked to provide some generic TP functionality
- Or **added flexibility**: "TP-CMX" capability could also be used to support future upgrades e.g. new Base-CMX algorithms needing a larger volume of Crate->System communication



Initial CMX design studies phase

- Started with guidance from **Stockholm Review**
  - Use Xilinx Virtex 6
  - Design for **6.4Gb** optical outputs
  - Implement G-link encoding on FGPA
  - → Generally **be conservative** to deliver on time

- Conclusions from Feb 2012 **Technical Workshop at RAL**
  - Recommend **Two FPGAs**: one for Base-CMX + one for TP-CMX function
  - **Two CMX types**: e.g. an implementation might be 1 (plus spares) with 2 FPGAs, 11 (plus spares) with 1 FPGA
  - Plan for operating the Crate CMX to System CMX LVDS cables at up to 4x current speed, i.e. 160Mbps → may require new cables or RTMs
  - Base-CMX function should provide **2x** 12-fiber **output** ribbons
  - TP-CMX function should receive **3x** 12-fiber **input** ribbons



Dual FPGA: Base-CMX and TP-CMX

If TP-CMX functionality is desired on CMX platform → use two FPGAs:

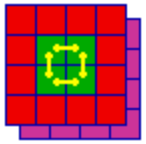
- Base-CMX and TP-CMX are totally separate functions:
  - logically don't share inputs or outputs
  - operationally in separate latency time segments
- TP-CMX needs access to CTP output → simply do not choose a System CMX to operate as a TP CMX

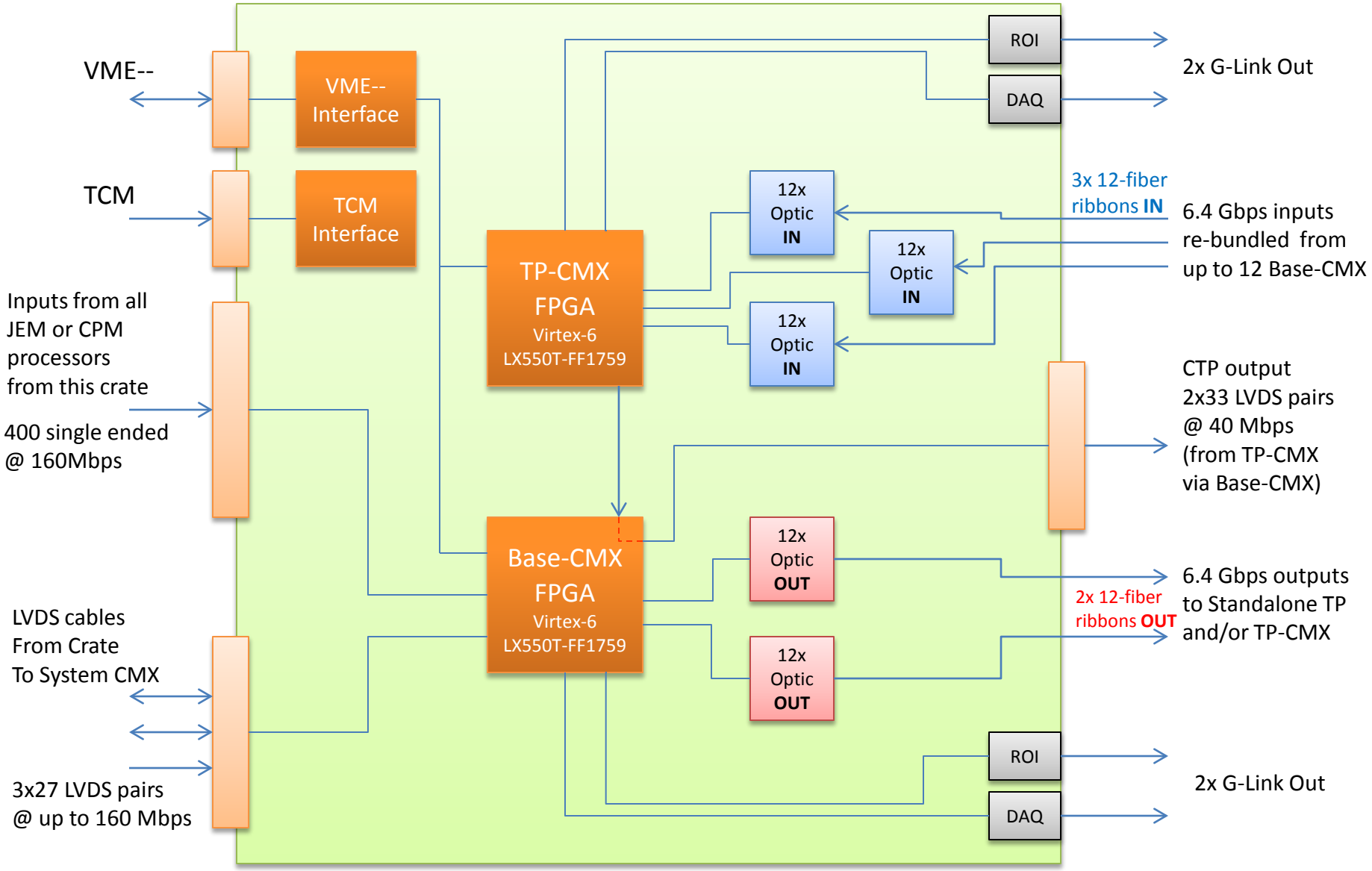
- **Separate** Base-CMX from TP-CMX **firmware** → **Easier commissioning**, easier **operation**, easier firmware **management** and easier updates

- Use smaller Virtex 6 package LX550T-FF1759 for both FPGAs
- Layout made simpler by separating routing challenges

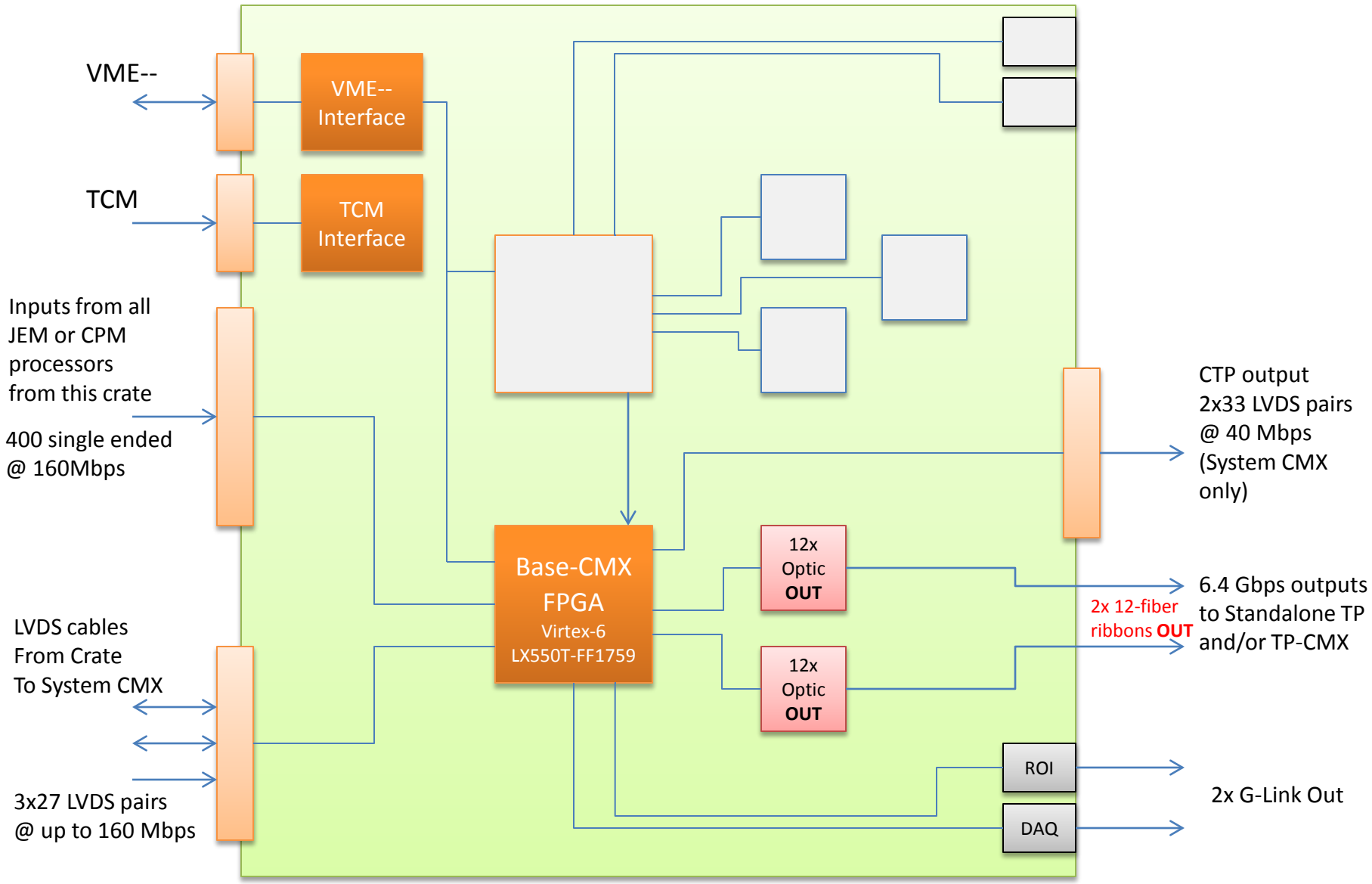
- A board TP CMX capability was originally only an option.
  - Accepting the RAL recommendations with Two CMX types will require only a modest increment in cost.

CMX Card with Base-CMX functionality and TP-CMX capability



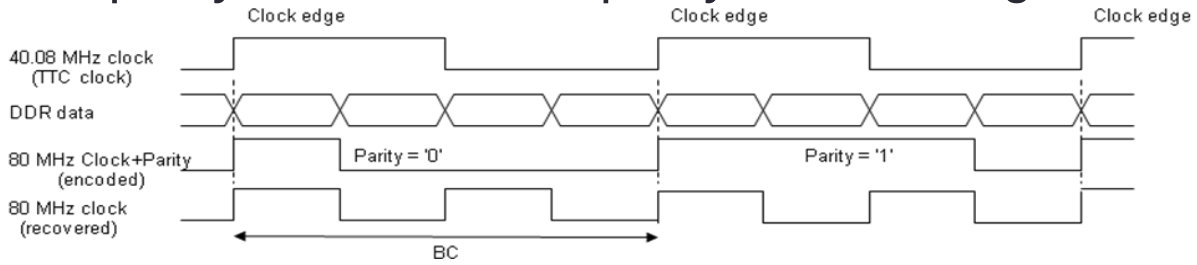


CMX Card with Base-CMX functionality only

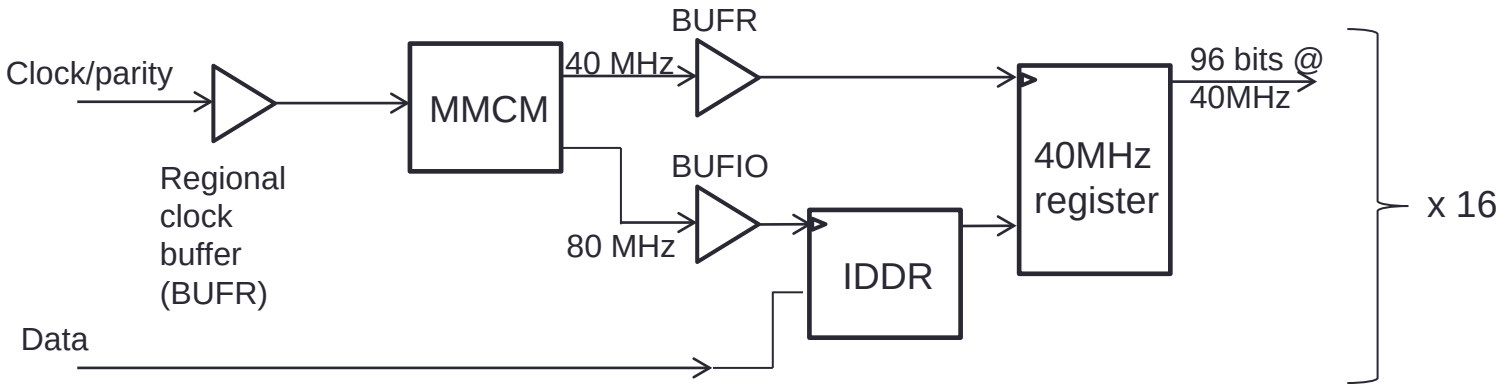


CMX input module firmware

- Purpose:
  - Capture 24 bits @ 160 MHz data and present 96 bits @40 MHz (x16 channels)
  - Compute parity of 96 bits
  - Decode parity from the clock/parity line and flag error



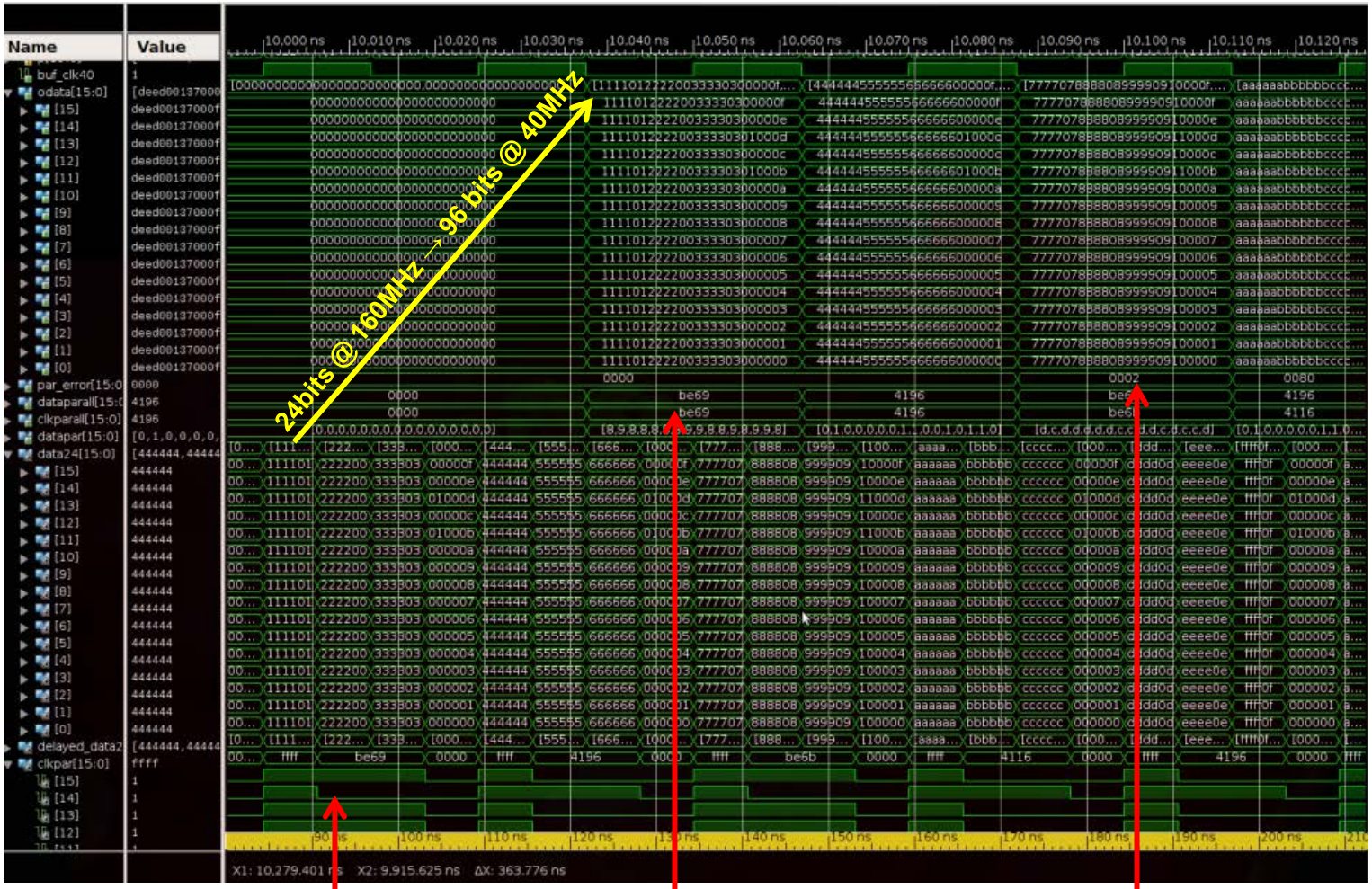
- Started from ULI's BLT firmware + Yuri's modifications
- Setup:



- Requires data pins and clock parity line (CC) to be connected in the same clock region

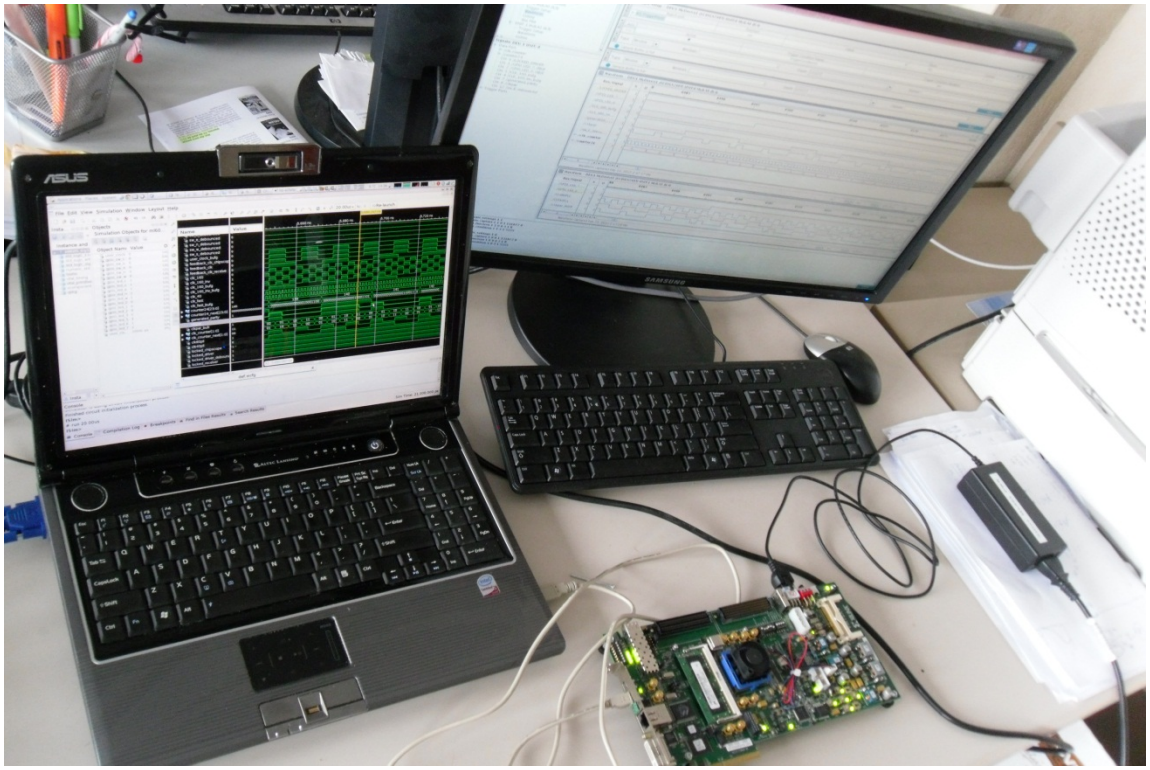
CMX input module firmware: Status

- Synthesis ✓
- Simulation ✓
- Implementation ✓
  - Toy pin assignment
  - ‘Original’ package



Test of MMCM lock test

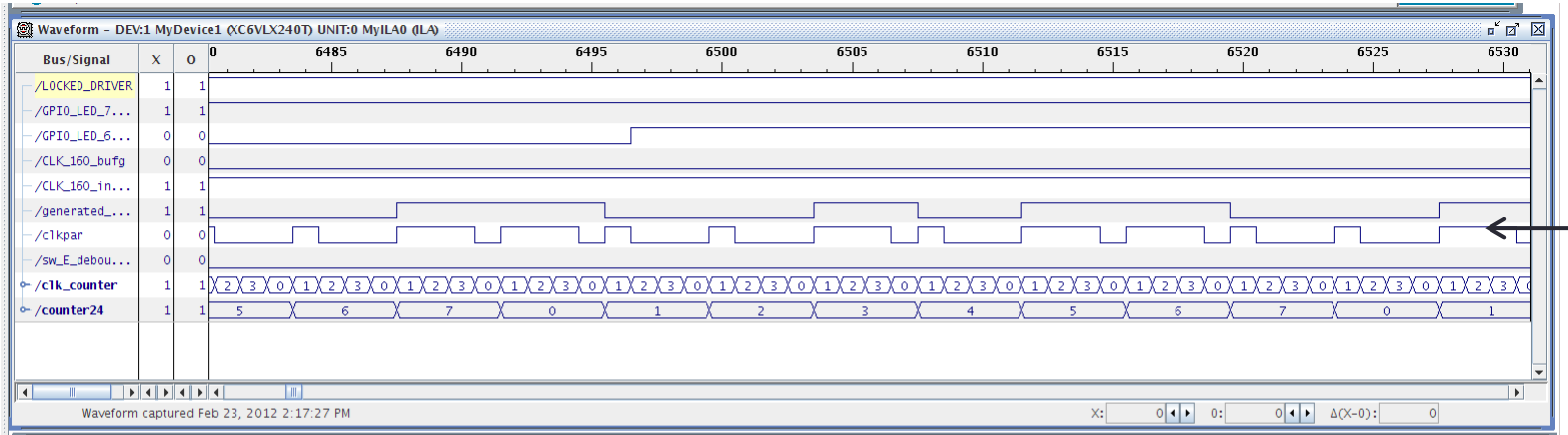
- ML605 setup:
  - Clock/Parity TX to RX routing internal in FPGA
  - Chipscope at 160 MHz with 2 ‘cores’ monitoring TX and RX



Lock test 1

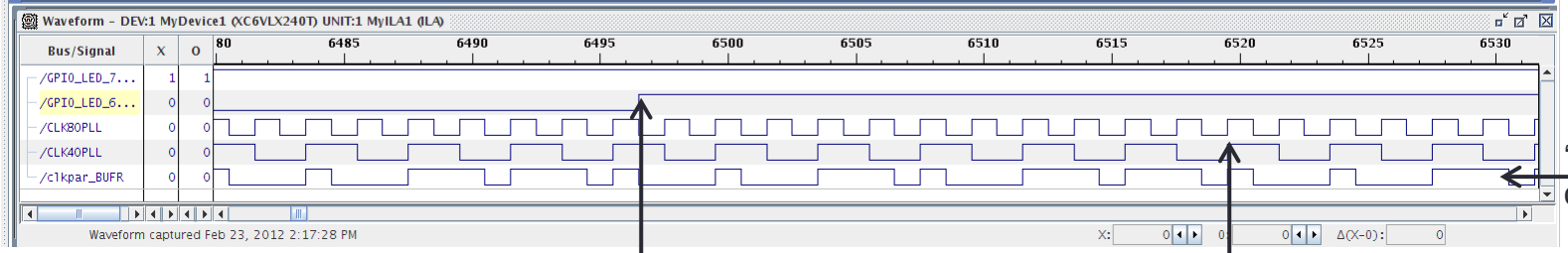
- Generate parity of a counter
  - Frequent 0-1 changes (max ‘run’ is length 2)

TX



‘sent’ clk/parity

RX



‘received’ clk/parity

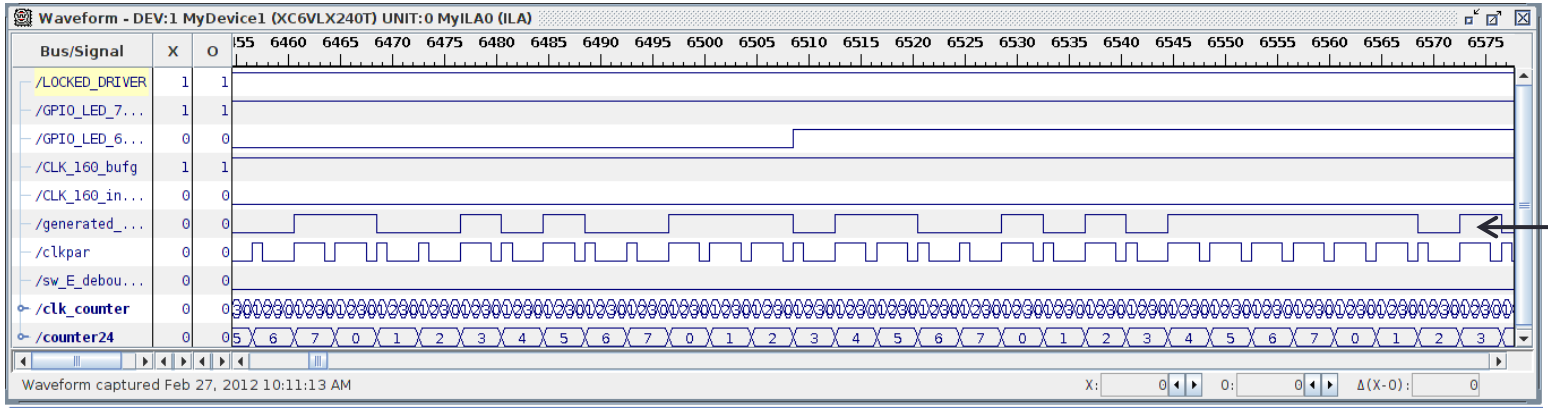
MMCM locked!

Generated 80 MHz and 40 MHz clocks in phase with rising edge of clk/parity line

Lock test 2

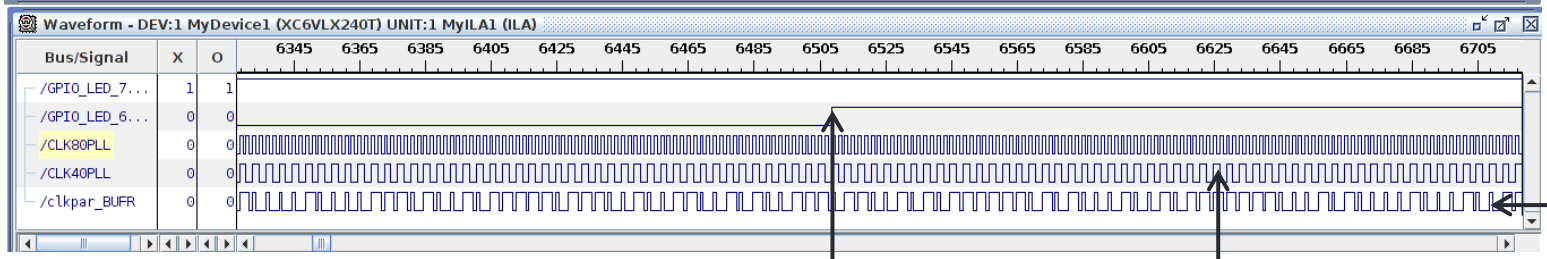
- Generate Pseudo Random Number, calculate parity
  - von Neumann PRN – long ‘runs’ occur

TX



‘sent’ clk/parity

RX



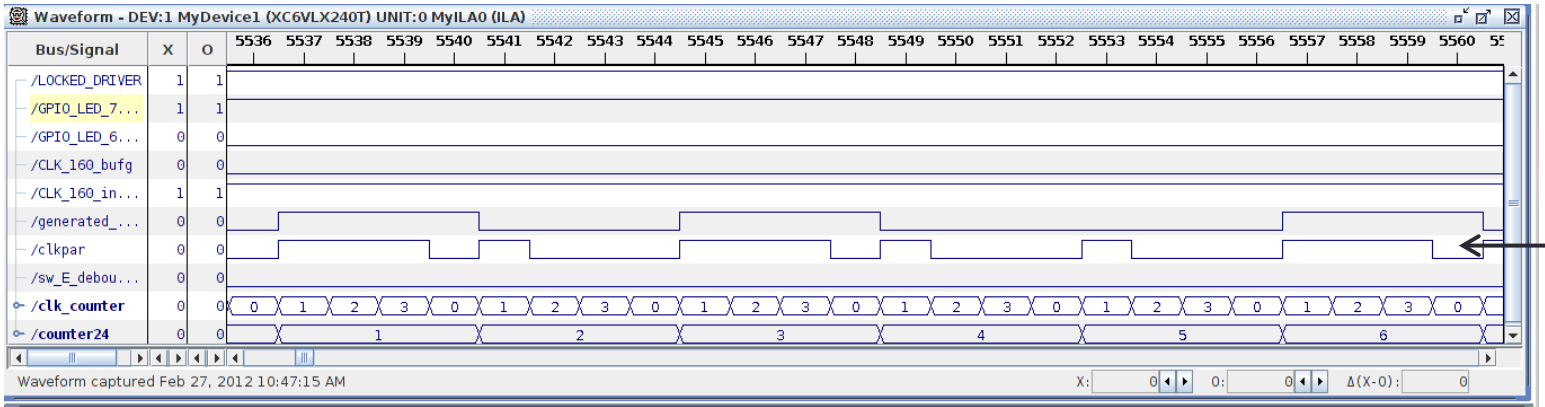
‘received’ clk/parity

MMCM locked!

Generated 80 MHz and 40 MHz clocks in phase with rising edge of clk/parity line

Lock test 3

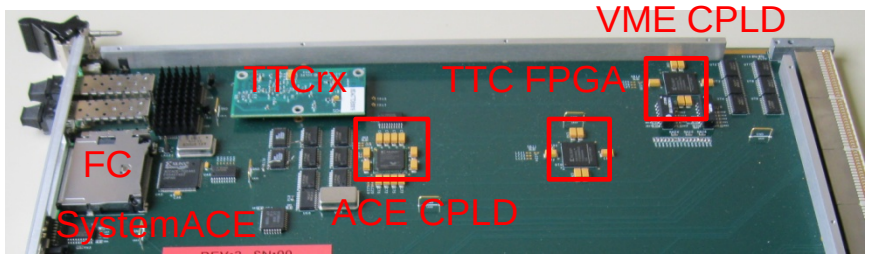
- Invert the PRN generated clk/parity
- No lock expected



CMX input module firmware: Next steps

- Move design to new proposed package
- Check implementation with latest constraints/layout proposal
- External pin loopback
- Characterize error rate
- Data formats discussion:
  - Decode data
  - Interface with CMM emulation from Pawel and Sam
  - Process ‘new’ data
- Code management:
  - Time to make it available in svn
  - Agree on svn structure, what to put where

VME/ACE/TTC (VAT) interface



VME/ACE/TTC (VAT) part of CMM

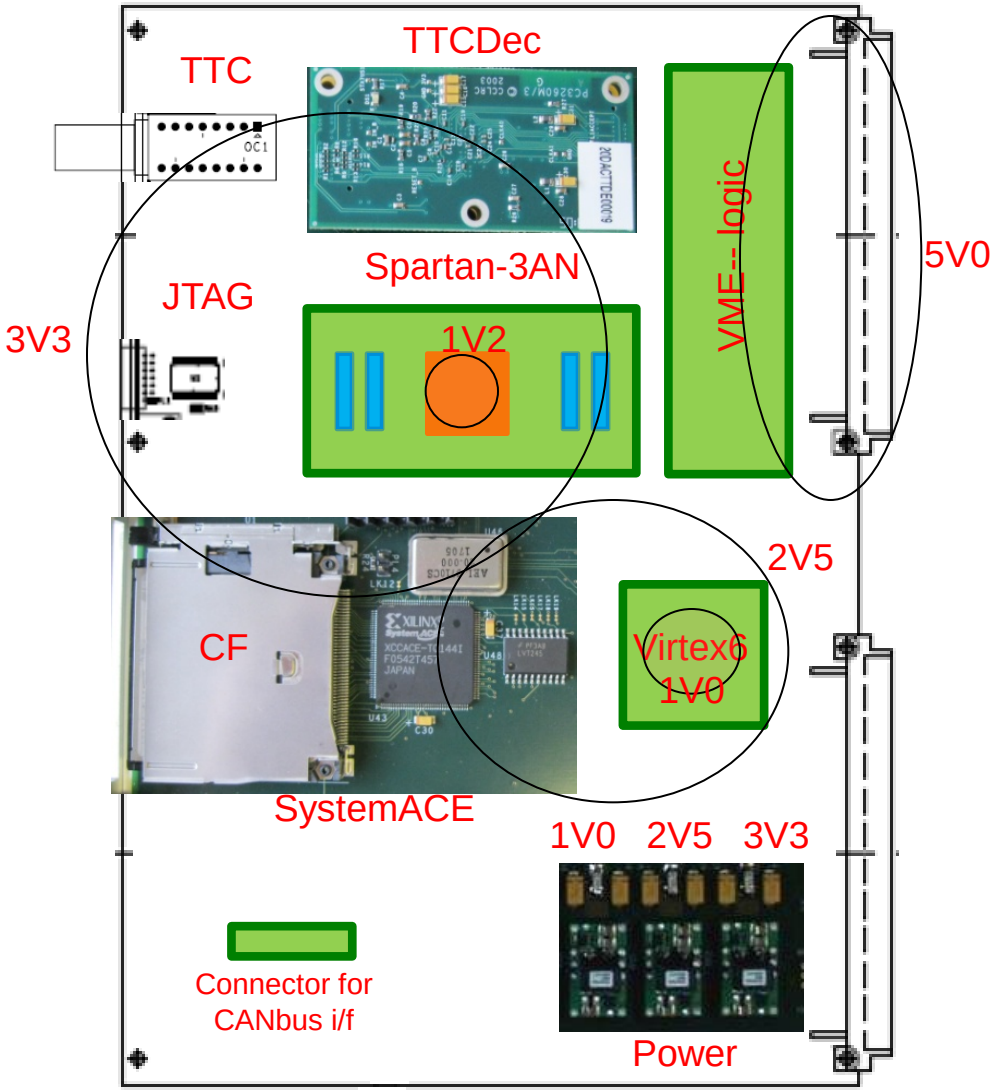
- Redesign HW with new components
  - Fit design into single device
- Implement on a daughter card
  - Different implementations
  - Use later on CMX
- 6U VME test card for
  - Hardware implementation
  - Main FPGA re-configuration
  - Software

Preliminary studies (VHDL)

- CMM source files from Ian Brawn
- Two CPLDs merged:
  - XILINX CoolRunner-II
- TTC FPGA in Spartan-3AN:
  - use internal memory
- 2 CPLDs and TTC FPGA:
  - Spartan-3AN XC3S200AN
  - integrated In-System Flash (ISF)
- Do we need to keep the CPLD?
- Main FPGA configuration
  - From Flash Card (System ACE)
  - Restart from VAT interface
    - ⇨ Restart VAT interface from CANbus interface

6U VME test card for VAT card

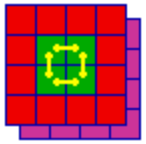
- Sources: CMM, VMM, TCM, BLT
- VMEbus logic
- TTC receiver + TTCDec card
- JTAG connector
- XILINX System ACE + CF card
- VAT daughter card
  - VME/ACE/TTC interfaces
- Virtex 6 FPGA XC6VLX75T
  - configuration
  - VME– interface
  - VAT interface
  - Spare connections?
- 3 I/O + 2 core power supplies
- CANbus daughter card ?
- Schematics ready



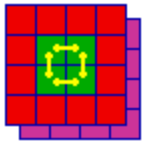
Next steps

- Hardware
  - PCBs (6U VME + VAT DC) layout
  - Conponents, PCBs production/assembly
- Firmware for VAT card
  - 2 CPLD + TTC FPGA → Spartan-3AN FPGA XC3S200AN-FTG256
  - Adapt VHDL code, create test bench
  - Specify VME register model
- Firmware for Virtex 6 FPGA
  - VME and VAT interfaces
- Test stand and software
  - SBC and L1Calo software
  - Test software

- CANbus ?



Backup slides

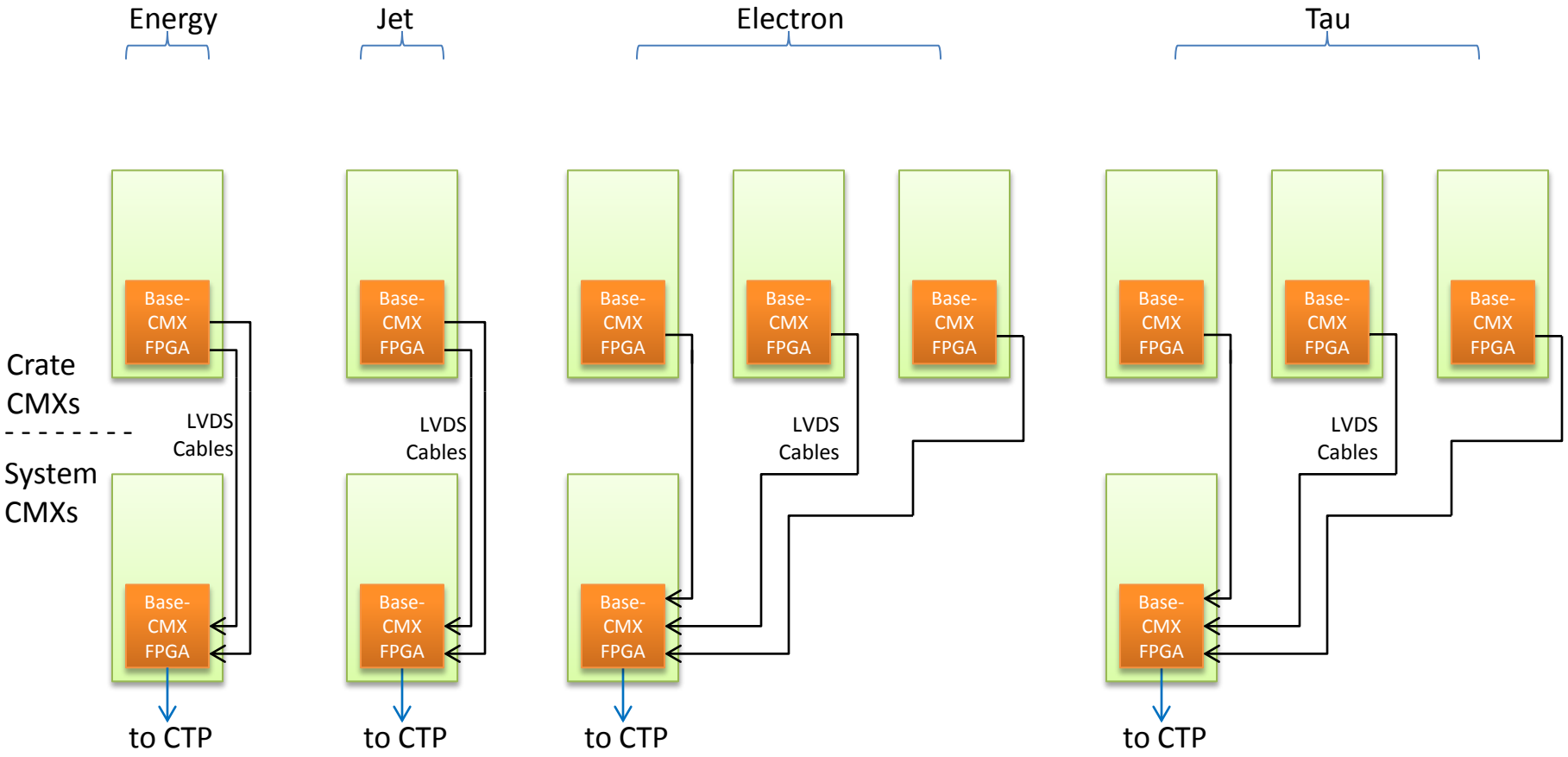


Overview of possible usage options

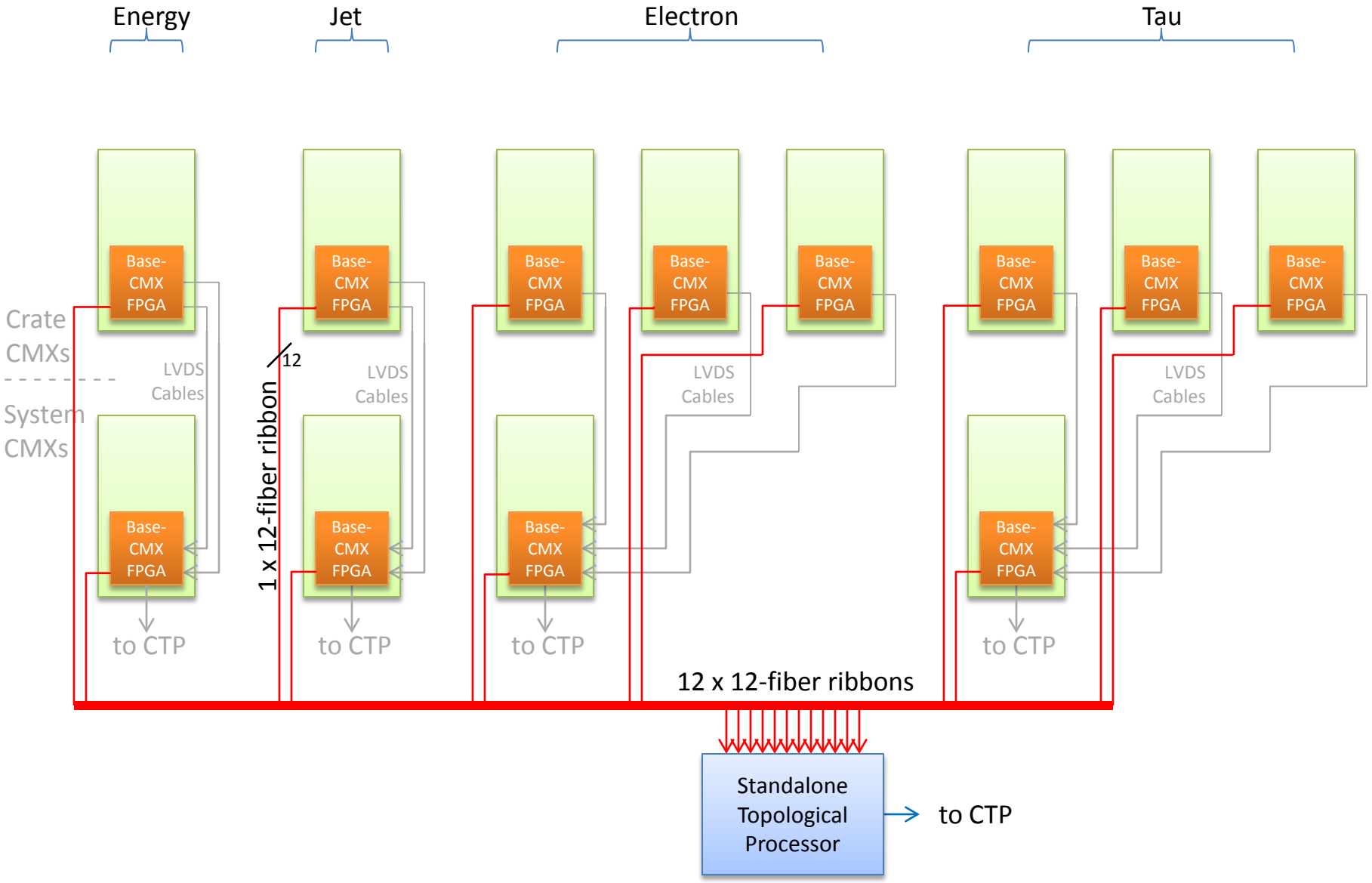
Flexibility is illustrated in following diagrams…

- CMM emulation only
- Base-CMX functionality only
  - Send data to Standalone TP (Raw or Zero-suppressed)
- TP-CMX functionality used
  - Only CMX-TP
  - Both CMX-TP and Standalone TP
  - In Crate to System CMX communication (not used as TP)

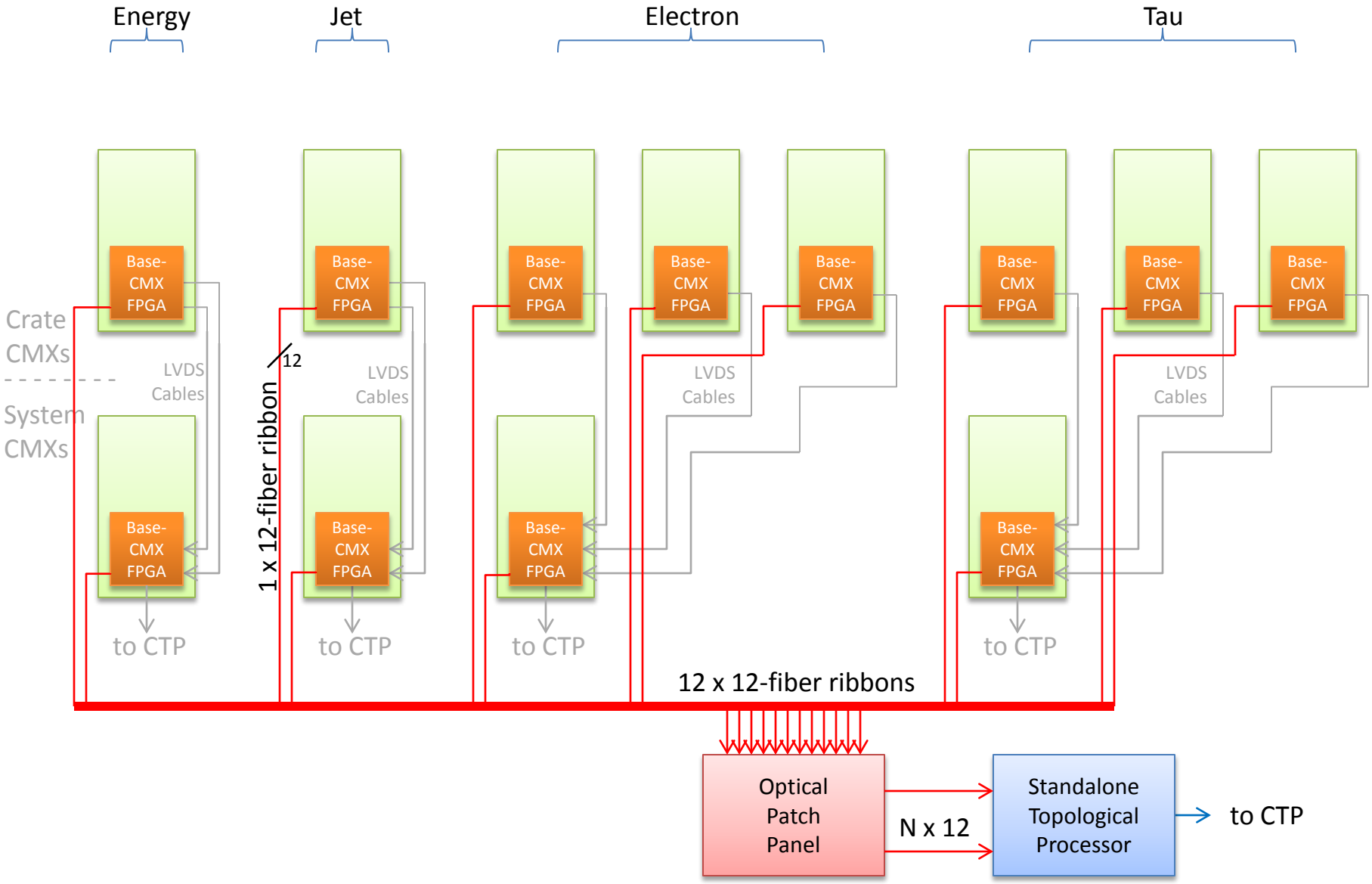
1. CMX emulation of CMM functionality (no TP involved)



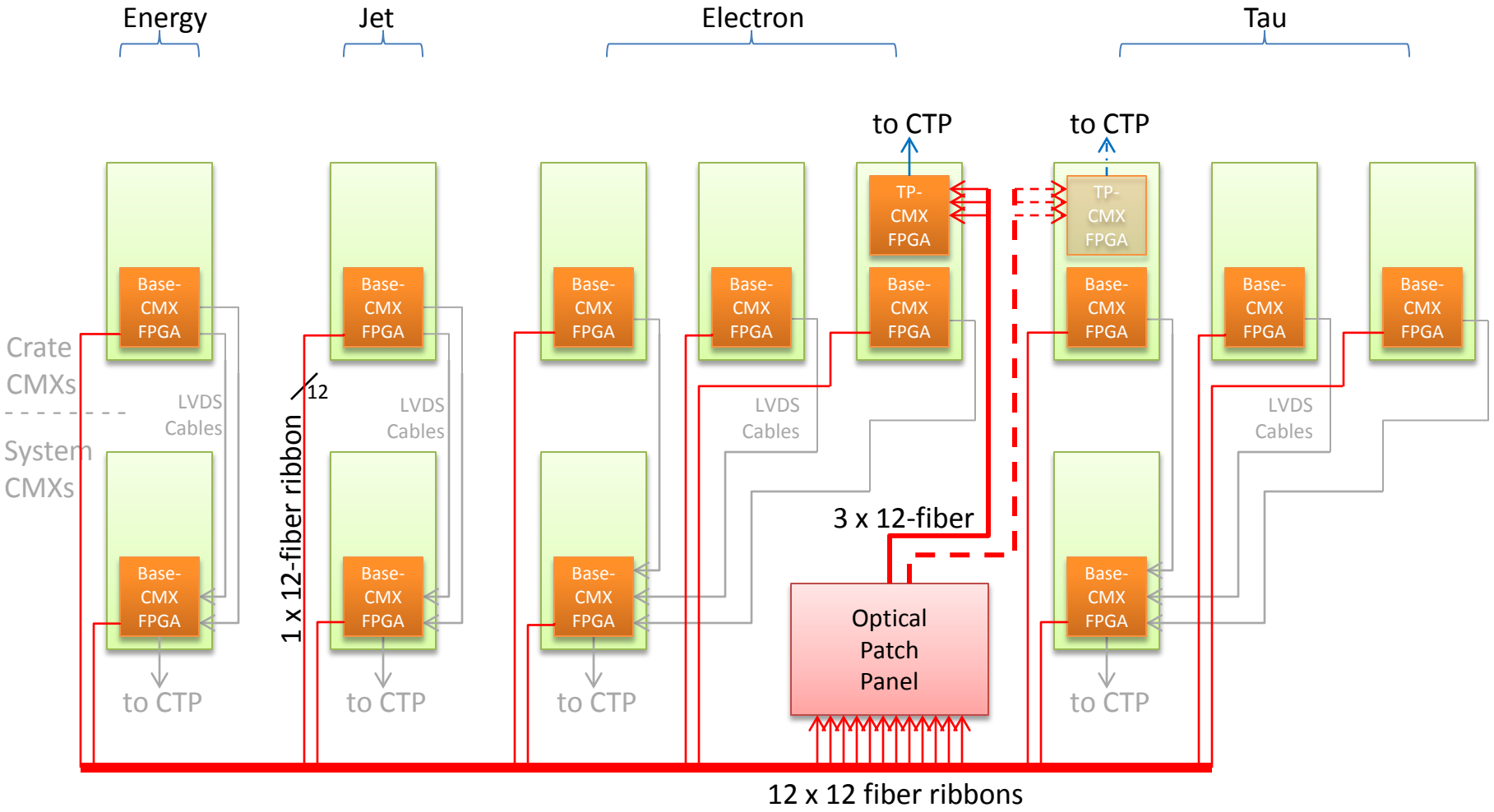
2. Standalone TP receiving Raw CMX Inputs



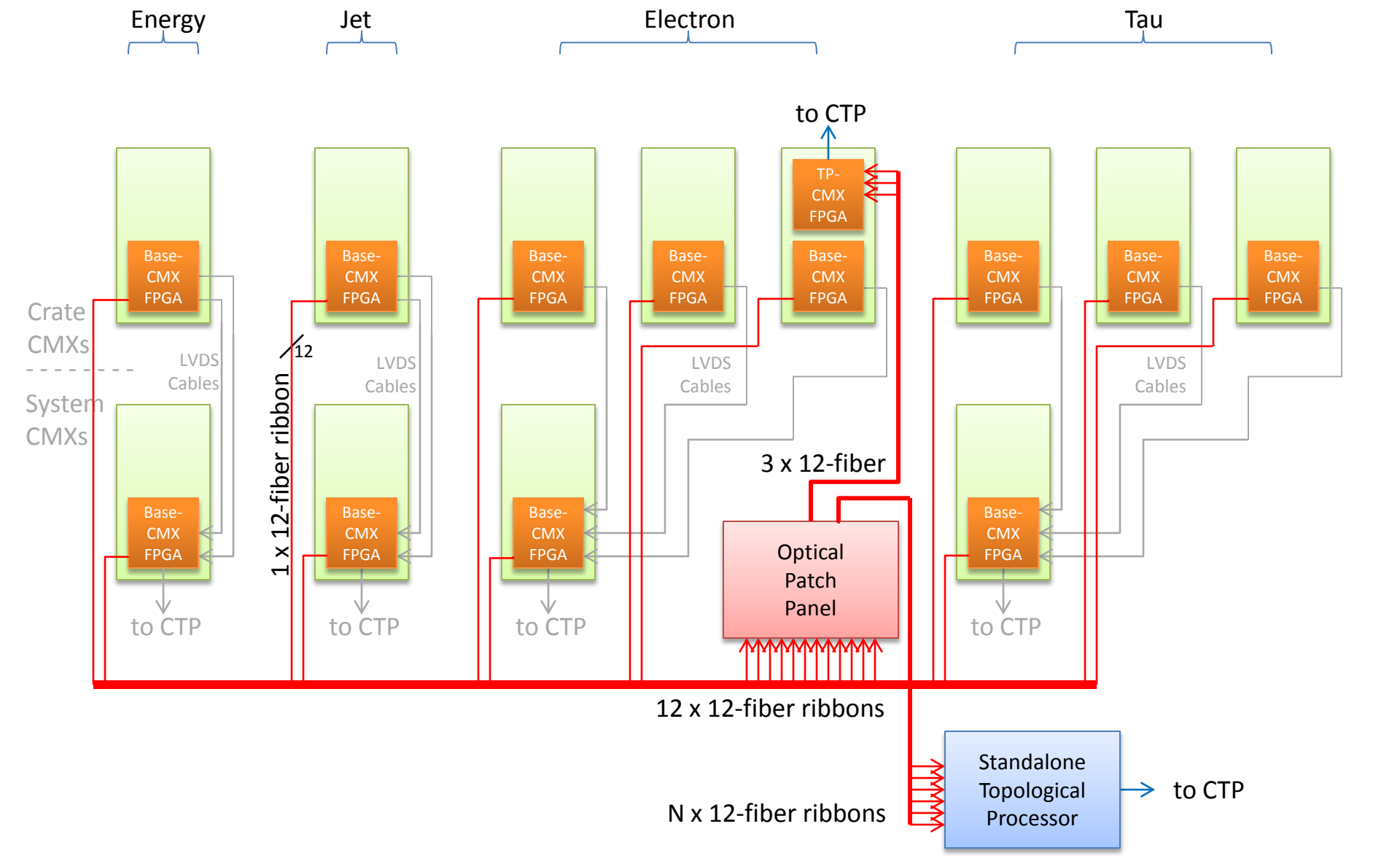
3. Standalone TP receiving Zero-Suppressed CMX Inputs



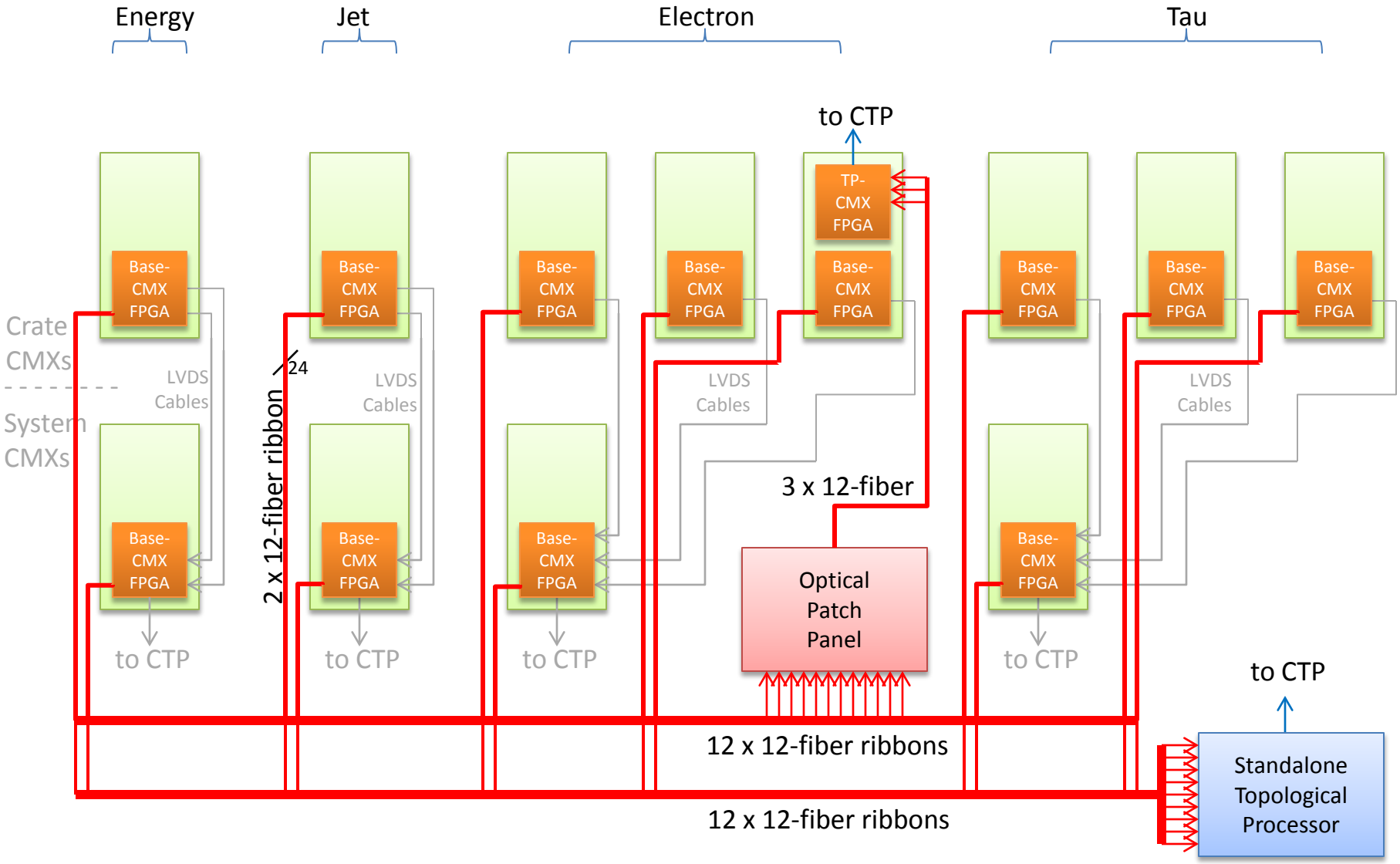
4. One (or more) CMX TP receiving Zero-Suppressed Inputs



5. CMX TP & Standalone TP receiving Zero-Suppressed Inputs



6. CMX TP receiving Zero-Suppressed & Standalone TP raw inputs



7. Example of TP-CMX FPGA in Crate to System communication