CMX:
FIRMWARE STATUS

Wojtek Fedorko

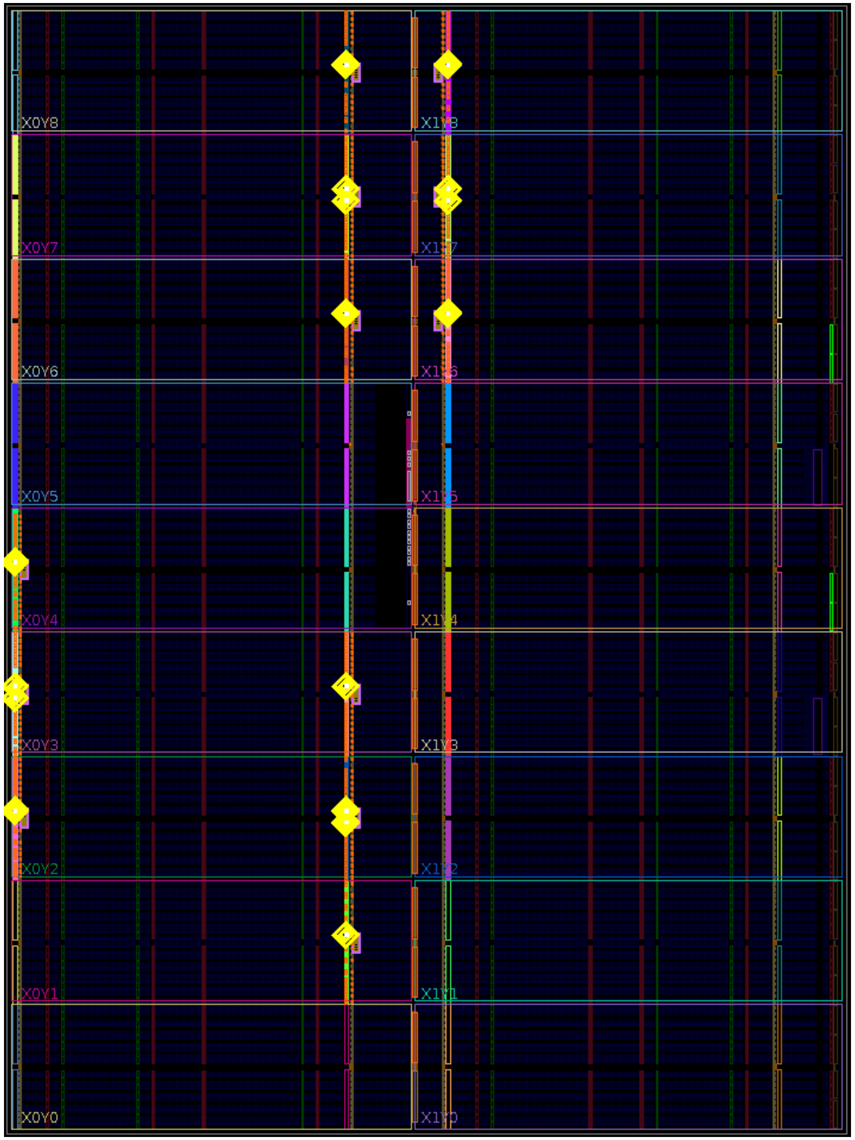
Firmware activities

- Past and present focus: Hardware design feedback
  - Backplane data capture
  - Power estimation
  - CMM re-implementation in Virtex 6 (Pawel)
  - VME/TTC/ACE test card
- Current and Future effort:
  - Defining firmware modules and functionality
  - Interfaces
  - Coordination and Implementation

Backplane data capture

- Challenge:
  - 16 processor inputs
  - 400 signals, 384 data
  - 160 Mbps rate
  - Unequal line delays
- Feedback to the PCB layout
  - Provide high signal integrity
  - Limit PCB complexity
  - Fit within FPGA resources (components and routing)

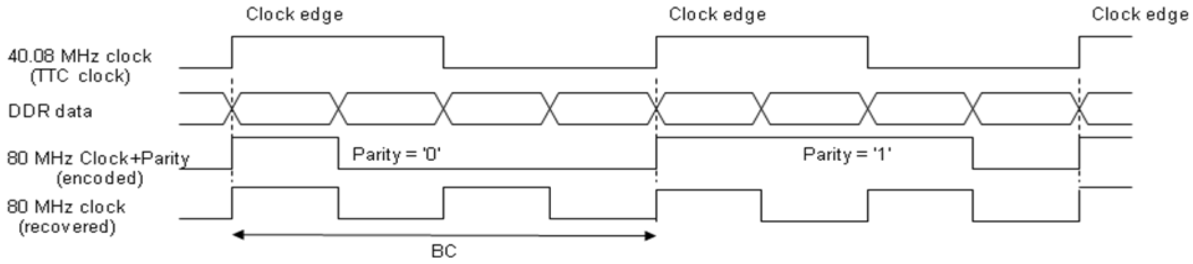
Virtex 6 FPGA: input



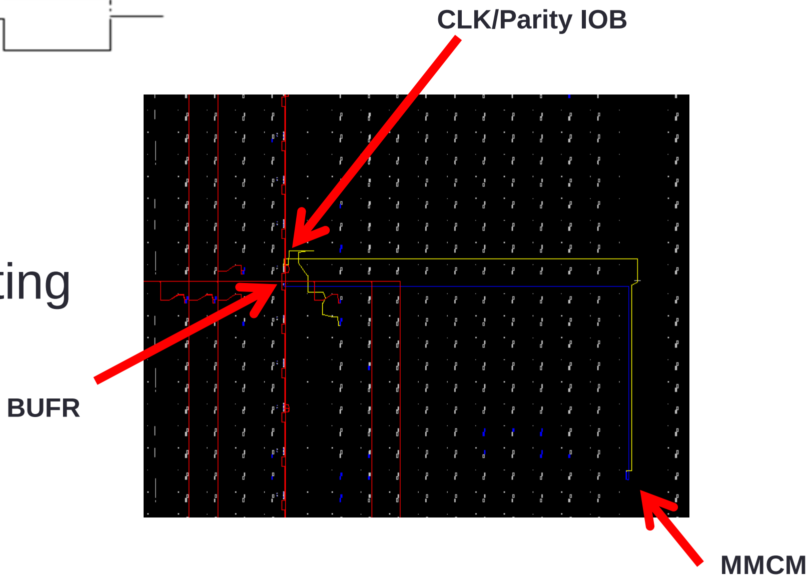
- 18 ‘clock regions’ in 9 rows and 2 columns
- 3 input columns:
  - ‘outer left’, ‘inner left’, ‘inner right’
- 2 ‘Mixed Mode Clock Managers’ (MMCMs) per row
- IODELAY circuits
  - Programmable delays (31 78ps ‘taps’)
- Dedicated DDR input registers
- Global, regional, and local clock networks
  - Restrictions on clock propagation
  - ‘2nd order’ resource use

Source synchroneous clocking schemes

- Clock+parity bit encoded on bit 24

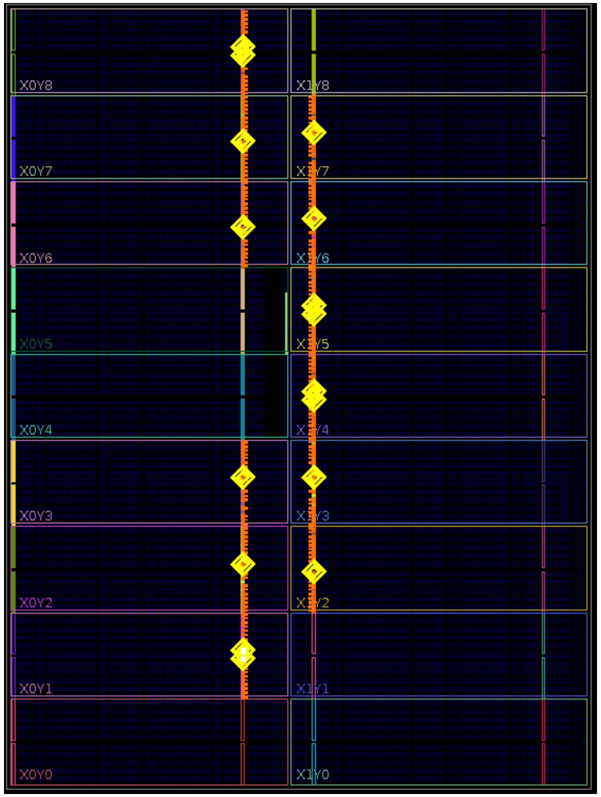


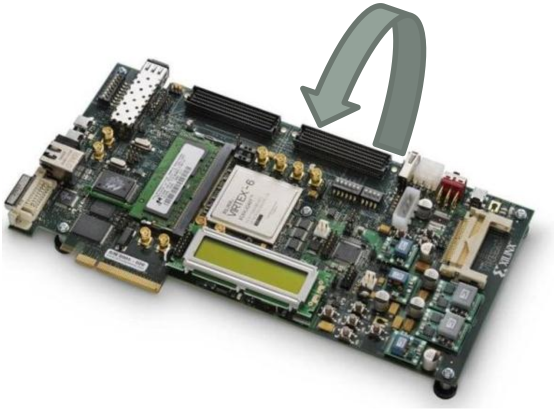
  - Non-standard
  - May pose problems for PLL
  - Requires a lot of low level work and arm twisting
- 80 MHz DDR scheme
  - Standard practice
  - Adopted, data formats modified
  - Less data
  - Requires special framing pattern to be sent
- Similar clocking network usage
  - Clock+parity uses additional resources (e.g. MMCM/processor input)



PCB Layout and kit tests

- Based on input from firmware side
  - FPGA input locations adjusted
  - Supports clock network distribution
  - Enables MMCM use if needed
- Data capture firmware tested in ML605 kit
  - 3 ‘channels’, 4 bits+clock each
  - Similar clock arrangement as on target FPGA
  - Both clocking schemes tested





Power Consumption estimate: Why and How

- Crucial for hardware design choices being made now
  - Power supply, heat dissipation, mechanical
- Estimated for CMM V6 implementation (Sam and Pawel)
- And the data capture part:
- Used ISE 14.2 XPower Analyser
- 2 methods:
  - ‘By Hand’ specification of clock frequencies and input switching activity
    - ‘Medium’ confidence level reported – internal nodes activity ‘guessed’ by the tool
  - Simulation based:
    - Generate of post Place-and-Route simulation model
    - Perform ISIM simulation and generate a switching activity file
      - Test vectors follow the expected time structure (160 Mbps data centered around R and F edges of 80Mhz clock)
      - ‘startup’ pattern followed by independent random data
    - ‘High’ confidence level reported

Results

- SAIF activity specification:

Source	Voltage (V)	Total Current (A)	Dynamic (A)	Quiescent (A)
Vccint	1	3.4	0.2	3.2
Vccaux	2.5	0.6	0.0	0.6
Vcco25	2.5	0.0	0.0	0.0
MGTAVcc	1	1.4	0.0	1.4
MGTAVtt	1.2	1.0	0.0	1.0

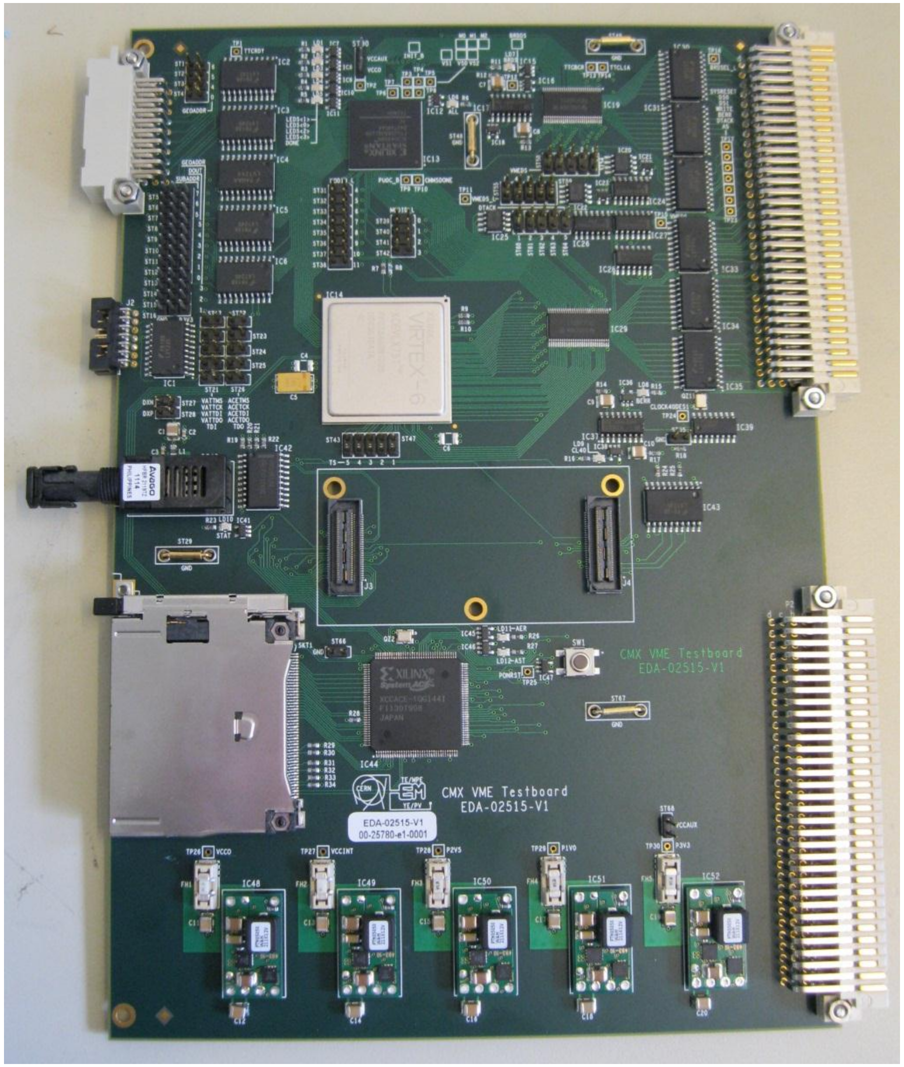
Power (W)
Total:
7.5
Dynamic:
0.3
Quiescent:
7.2

+0.4 W dynamic power consumption from the CMM emulation logic (thanks to Pawel and Sam)

- Very small dynamic power estimated
- Large portions of the design not implemented (GTX!)
- Further studies necessary:
  - Implement ‘dummy’ GTX TXs inside the design
  - Test reliability of the estimate using ML605 system monitor

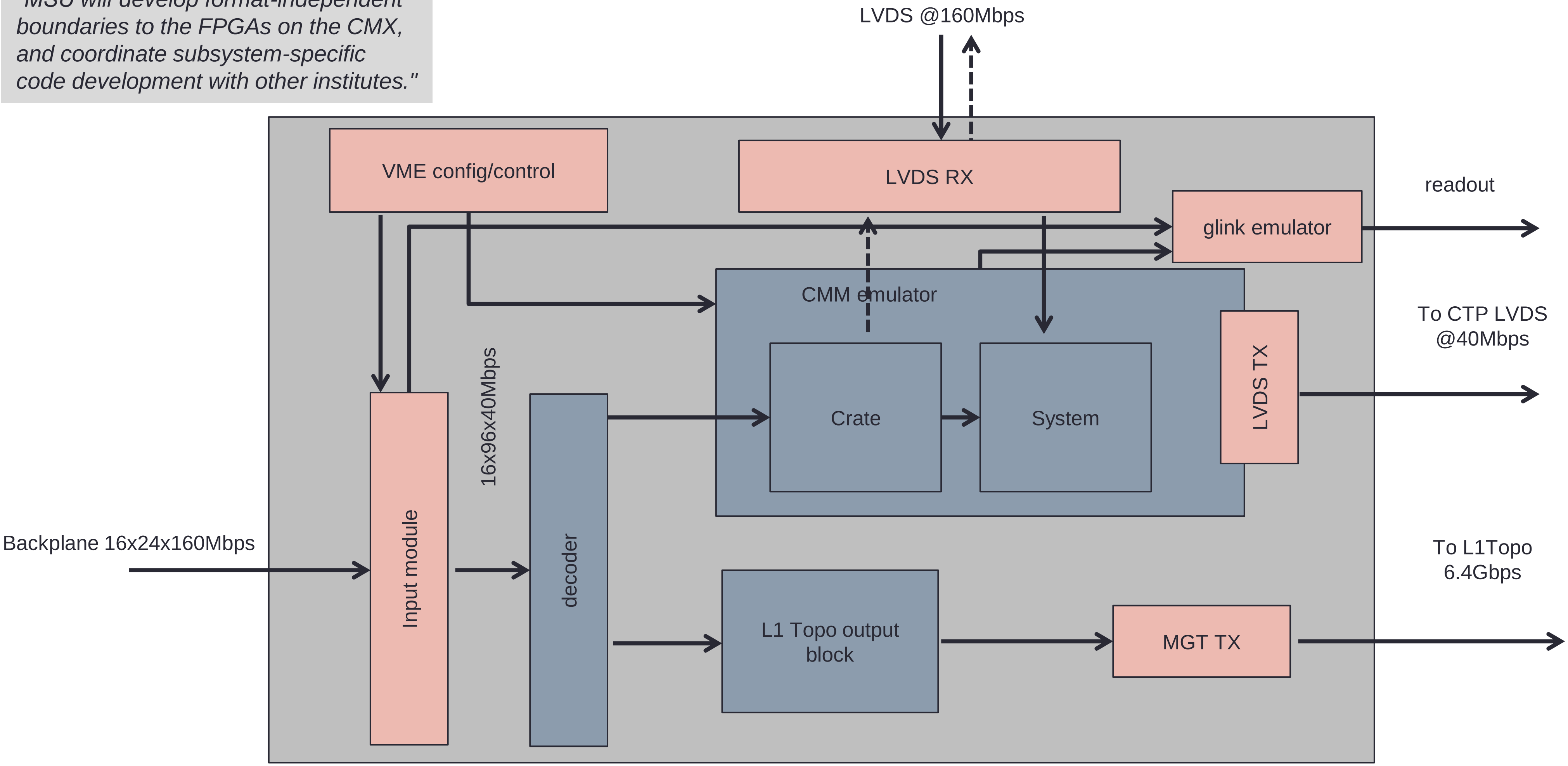
VME/ACE/TTC test board

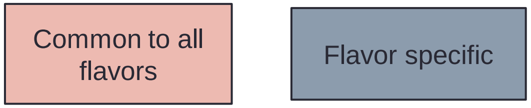
- Design will be merged with the CMX design
- Currently debugging CF main FPGA configuration
- VME communication part not started yet



Planning ahead CMX (base) FPGA functional blocks

“MSU will develop format-independent boundaries to the FPGAs on the CMX, and coordinate subsystem-specific code development with other institutes."



Legend: 

Firmware modules

- Need to define functionality and interfaces between firmware ‘modules’
- Resource sharing (e.g. MMCM – clock generation and buffering)
- Probably ‘Take out’ some functionality from the CMM emulator module
- Start work from the ‘Jet’ version of CMX
  - Expect that common module interfaces will work for other versions
    - Use Pawel’s expertise with the CMM firmware
  - Will be able to use this or stripped down version of this firmware for board tests
  - Need complete firmware before prototype arrives early next year. Hope ready sooner.
  - Part of firmware testable in Yuri’s TTC/VME/ACE test card.
  - Other firmware versions will follow after prototype tests.

Points of discussion

- Data Formats:
  - Decided, need documentation.
  - Need light ‘protocols’
  - Input framing. Once? Once per orbit?
  - Output synchronization?
  - Event header?
  - K?
  - ‘unused’ bits?
- Synchronization of processor input inside the FPGA logic.
  - Input module: 16 short FIFOs? Better options (same or ½ frequency unknown phase)?
- Behavior on parity error?
  - Include information in the real time data?
  - Special L1A to force readout?
- …

CMX Topo FPGA

- Plans for firmware development need to be discussed
- Hope for common firmware architecture with L1Topo
- Test version needs to be ready for prototype testing.
    - Link tests

Conclusions

- Bulk of necessary work guiding hardware design complete
  - Bulk of input module:
    - Data capture
    - Data framing
    - 160Mbps → 40 Mbps
    - No synchronization between processor inputs
  - Need still some effort (GTX power estimate), but no ‘wasted’ work
  - VME/TTC/ACE card work ongoing
- Need concentrated planning soon, then a divide and conquer campaign to implement firmware modules