PostPDR list of modifications to the CMX specification

2 RELATED PROJECTS AND REFERENCES
List of related projects: should include CPM and JEM specifications

3.1 BACKWARD COMPATIBILITY
5.1 BACKWARD COMPATIBLE (CMM EMULATION, CMMe) MODE
It was also noted that while for the backwards-compatible mode the register should be as close as possible to that of the existing CMM, it is not an absolute requirement that the register map be identical. So small changes to accommodate the needs of the new hardware devices are acceptable.

4.1 PROCESSING FPGA
I would suggest we spend a few moments to reason about single FPGA vs. dual FPGA approach. For a single FPGA the parallel lines JUST fit. And that's true only if you are using single ended <--> differential converters external to the FPGA. Since you are, however, talking about possibly increasing the data rate on the legacy LVDS interfaces considerably, that might perhaps be a sub-optimal approach wrt signal integrity. I wouldn't mind seeing a separate system merger FPGA (possibly a cheap Spartan-6) that's in the data path for legacy mode only, and therefore doesn't impact TP mode latency.

4.1 ~~High speed transceivers~~ FPGA
The question was raised as to whether the optical links might be designed to run at up to 11 Gbit/s, rather than the nominal 6.4 specified in the document. The Committee felt that the time and effort such investigations would entail could delay the development schedule. It was stressed that such delays are only acceptable if a strong physics case is made that the higher speeds are necessary.

4.1 backplane receivers
For backplane transmission to the CMX, the document should be amended to specify that internal termination on the main CMX FPGA should be enabled for the clock/parity lines from all of the processor modules to ensure optimal data timing.
Just one comment / question. Figure 3 as drawn shows the 80MHz clock edges occurring in-time with data transitions, which reminds me ...
Will you add a delay in the CMX FPGA, or shall the CPM delay the phase of the outgoing backplane clock/parity signal (3.1 ns) to put the clock edges in the centre of the data bit. (This delay may be useful anyway for my firmware in the CPM's Readout controller as the clock functions in the Virtex-E devices are limited)

4.3 LVDS CABLES AND REAR TRANSITION MODULE
The current CMM rear transition modules (RTM) will be used for backward-compatible modes. Two will be needed for the MSU rig. If there are insufficient spares, more RTMs could be produced.

4.4 CTP INTERFACE
A request was made to clarify that after deployment of the CMX, the CTP will continue to receive the present jet and cluster threshold multiplicities, either from the CMX or the TP.

4.5 OPTICAL LINKS TO TP / FROM OTHER CMX
5.4 NUMBER OF OPTICAL LINKS IN DIFFERENT MODES OF OPERATION

For optical transmission to the TP, it is desirable to have as many transmitters as we can, within reason. Transmitter and receiver multiplicities should be in multiples of 12, for full occupancy of the parallel optical modules.
The optical links from the CMX to TP are crucial, and there should be close coordination between the CMX and TP developers to ensure their reliable operation.

4.10 POWER - new
Note: the CMX may only draw power from the 5V supply (not 3.3V).

6.2 HARDWARE DEVELOPMENT
Addendum: Care must be taken to include accessible test and ground points in order to allow timing and other critical signals to be studied.

6.3 FIRMWARE DEVELOPMENT
Two smaller programmable devices (I2C and VME interfaces) are based on obsolete chips, and will need to be ported to more modern devices. This implies some amount of design and coding effort.Under section 6.3 (firmware development), the additional, minor firmware mentioned above should be added.
The development of topo algorithm firmware for the Topo Processor has started, and would be going on in parallel with CMX development; since MSU has been involved in this area from the beginning, and is leading implementation of them in the simulation, I would hope that we could contribute to fpga implementation as well. But obviously our highest priority must be on the baseline CMX functionality in emulation mode and data source mode. I could imagine designing the algorithms to enhance their likely portability between CMX and TP (depending in e.g. on similarity organization of I/O on the CMX vs TP?), but that's better assessed by the experts. Yuri has made some efforts in the direction of an overall firmware environment including the interfaces between the various boards as part of a development environment for CMX and TP.

6.5 TEST AND INTEGRATION
To the test plan, it should be stated that a blind board with backplane connectors and mechanics will be produced to verify that the CMX will fit properly in the CMM positions in the CP and JEP. A front-panel mock-up is also requested.
Will the CMX have bracing-bars and extra strong handles as did the original CMM?
I just wanted the MSU guys to realize that the boards need non-standard handles to deal with the large forces on the module when inserting into the crate. I wasn't sure if you heard about our experience with standard plastic handles breaking, and boards bowing (esp on the CPM). Using custom connectors with a number of different length pins helped to stagger/spread the forces in time.

APPENDIX A: MSU TEST RIG
In the test rig description, it should be added that a CTP data sink can be implemented using either a DSS or merger module.

APPENDIX B: DATA FORMATS
The document should be amended to indicate that the backplane data format descriptions in Appendix B describe the data content sent to the CMX, but not the exact arrangement of the data, which is yet to be determined. It might be worth point out here that Heavy Ions may require separate treatment (and maybe therefore a different firmware load with different algorithms). The problem here is that the RoI reduction doesn't work, or requires an overflow condition, for highly central events. This requires thought at some point, though not highest priority.

PostPDR list of actions

1. One month after the review (29.07.11) - milestones
Milestones to reach agreement on interface specifications and data formats with connecting modules and systems to be defined and added to the specification (6.1).

2. Before FDR - CTP optical input
It was noted that the CTP upgrade plans include provisions for optical inputs. It was recommended that L1Calo look at the spare capacity of the existing CMM outputs to the CTP, and the impact of possible additional thresholds, and evaluate whether an optical output from the CMX would be justified. Such a decision would be arrived at by the time of the final design review (FDR).

3. Before FDR - Algorithms and Physics sign-off
One milestone must be physics sign-off by the ATLAS L1 community. This procedure is not currently understood, so the details of this must be worked out prior to the FDR.

4. Before FDR - engineering risk/benefit analysis
It was noted that the so-called Standalone Mode is challenging and adds significantly to the module complexity, and presents a potential risk that the core functionality of the CMX (operation with the TP) could be compromised. An engineering risk/benefit analysis needs to be performed to evaluate the number of input links that can be safely added. It is noted and understood that Standalone Mode is a backup solution for topological processing, and some compromise in capabilities compared with the full TP may be unavoidable.

5. Before FDR - timing and clock resource
It is crucial that timing and clock resource issues be understood completely, especially for high-speed data transmission. These may have a strong influence on the final design.

6. Before FDR - power consumption/dissipation
A power consumption/dissipation estimate needs to be made for the final design review (FDR), showing that the CMX power needs can be met in the CP and JEP crates. Note: the CMX may only draw power from the 5V supply (not 3.3V).

PostPDR (before FDR) technical/engineering study

- engineering risk/benefit analysis
It was noted that the so-called Standalone Mode is challenging and adds significantly to the module complexity, and presents a potential risk that the core functionality of the CMX (operation with the TP) could be compromised. An engineering risk/benefit analysis needs to be performed to evaluate the number of input links that can be safely added. It is noted and understood that Standalone Mode is a backup solution for topological processing, and some compromise in capabilities compared with the full TP may be unavoidable.

- timing and clock resource
It is crucial that timing and clock resource issues be understood completely, especially for high-speed data transmission. These may have a strong influence on the final design.
I would guess that this scheme requires a lot of global or regional resources for the incoming data. While generally regional buffers should do for latching parallel data in, I would suspect that the clock recovery scheme described here might actually need global buffers due to dedicated routing between MMCMs and global buffers.
On the other hand, plenty of global resources are required, if we want to run the MGT links in low latency mode. I suspect that with the currently existing devices we might run out of resources and we might have to give up on the idea of using phase alignment mode for latency reduction. We would rather have to use clock correction mode instead, and there someone would have to invest some effort on tuning the scheme towards low latencies. Clock correction mode, as it is available out of the box, uses deep buffers and yields horrible latency figures afaik.

- power consumption/dissipation
A power consumption/dissipation estimate needs to be made for the final design review (FDR), showing that the CMX power needs can be met in the CP and JEP crates. Note: the CMX may only draw power from the 5V supply (not 3.3V).
Just wanted to mention there is a brick wall on 5V power for CMX as there is a power pin rated at 16A: so 80W max.
On the CMM board there is also a 10A fuse (see attached, page 3) - therefore 50W, I guess...
Can be an argument to implement or not the standalone mode...

- CMM parts re-use
There is an FPGA on the current CMM that isn't mentioned in your text, or shown in your block diagrams: the I2C FPGA. This is a small device, an XCV100E that implements a VME interface to the various control and status pins of the TTCrx (I2C, Brcst, etc). For the CMM++ you'll need to keep this functionality. Incorporating it into your main FPGA would require ~30 extra pins, so I assume you'll also want to use a small, separate FPGA. (Porting the firmware for this FPGA should be straight forward, by the way.)

- FPGA selection
I would suggest we spend a few moments to reason about single FPGA vs. dual FPGA approach. For a single FPGA the parallel lines JUST fit. And that's true only if you are using single ended <--> differential converters external to the FPGA. Since you are, however, talking about possibly increasing the data rate on the legacy LVDS interfaces considerably, that might perhaps be a sub-optimal approach wrt signal

integrity. I wouldn't mind seeing a separate system merger FPGA (possibly a cheap Spartan-6) that's in the data path for legacy mode only, and therefore doesn't impact TP mode latency.
We probably need to reserve some FPGA pins for the "management" of the high speed optical link transmitters and receivers, i.e. the TXEN, TXDIS, RESET-, and FAULT- lines for the transmitters and the RXEN, ENSD, SD, SQEN lines for the receivers. This may be about another 10 to 15 signal pins on the FPGA. The SFP transceivers for the G-Link outputs also have some management lines. This may be about another 5 or so pin on the FPGA.
Regarding splitting the chip: I would worry about reducing the functionality on cost grounds without working though the implications. However, if the function of pumping data through to the TP can be separated from the "Insurance" function, then perhaps not ALL CMX need to be loaded with the more expensive device. Even in this case, there would need to be enough fully-equipped modules to provide working spares. You could perhaps regard this as contingency?

- Backplane hardware interface CPM/CMX and JEM/CMx
I rather use PLL to recover 80 MHz clock from encoded clock/parity line and FPGA input DDR register to fetch the data...Therefore, 160 MHz clock was neither transferred over the backplane nor used in the receiver part.
For the use of the latency minimized phase alignment (rather than elastic buffer) scheme on the GTX transceivers Xilinx has issued an erratum saying that this requires global clock resources. This is definitely true on the receiving end, but possibly also on the transmitter. Worth checking before going into detailed design phase. I do not know about latencies and global clocks wrt. GTH transceivers unfortunately.
Current plan is to do this on each (16) input bus. There are 18 MMCM in FPGA. The encoded clock is recovered in PLL and fetch incoming data on each clock edge using input DDR register.
Just one comment / question. Figure 3 as drawn shows the 80MHz clock edges occurring in-time with data transitions, which reminds me ...
Will you add a delay in the CMX FPGA, or shall the CPM delay the phase of the outgoing backplane clock/parity signal (3.1 ns) to put the clock edges in the centre of the data bit. (This delay may be useful anyway for my firmware in the CPM's Readout controller as the clock functions in the Virtex-E devices are limited).
I think we have to sit together and specify the CPM/CMX interface in details (as well as JEM/CMX) - how to do in convenient way for both of us. In Virtex 6 you can delay the input signal, but may be it will be more convenient to delay it in CPM...

- High-speed links
One other thought: the self-testing of CMX implies a number of receivers N equal to the number of transmitters N at least, if you are doing the test with just 2 boards. True, you could finesse this by configuring/soldering one board laid out as N transceivers as "all transmitters" and the other as "all receivers"; I leave to you engineers whether that more naturally implies traces for N transmitters and N receivers both attached to N transceivers on the FPGA, or 2N with dedicated layout for N out, N in, as a minimum complexity. My main point is defining a minimum sensible complexity for a self-testable CMX even if it is destined to never do standalone. I think what's described in the CMX document is a layout with N transceivers and traces to connect all of them as either N transmitters, or as N receivers, rather than 2N transceivers dedicated as N out, N in.

- optical connectors
I would guess this means the use of mid board transceivers with pigtails. However, space for the MTP/MPO feed through connectors needs to be provided anyway. I would like to point out that it is

important to provide plenty of spare connectivity on the CMX. If required, space must be made available on the front panel. The Fujitsu RS232 needn't necessarily be routed via Sub-D9. Maybe smaller connectors exist. And for the SystemACE one should make sure that the long-planned in situ update of the flash cards is developed into a reliable tool. If it works 100.00% then there is no need to mount the flash card on the front panel of the production version of the CMX! Technically simple approach: Two CF connector footprints, one near the front panel and one in a recessed position. Assembly where suitable. MPO feed through is mechanically connected to front panel only and isn't affected by an unused CF connector footprint on PCB. Also, one should seriously consider MTP/MPO for replacement of the existent SFPs.
Here you suggest that we do not need a lot of output bandwidth. In fact, further down (p.14) you go into some more detail and talk about data replication to more than one TP module. That's important, since paralleling the processing could possibly allow for a wider choice of algorithms *without* compromising latency. There is no need to wire up everything from the beginning, but opto sockets and front panel space should be made available.

YE 07.07.2011