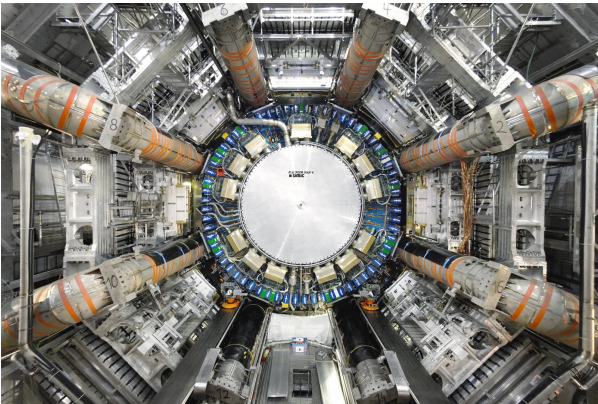


Future Development and Direction



RCE Training Workshop

Michael Huffer, mehsys@slac.stanford.edu
15 June, 2009



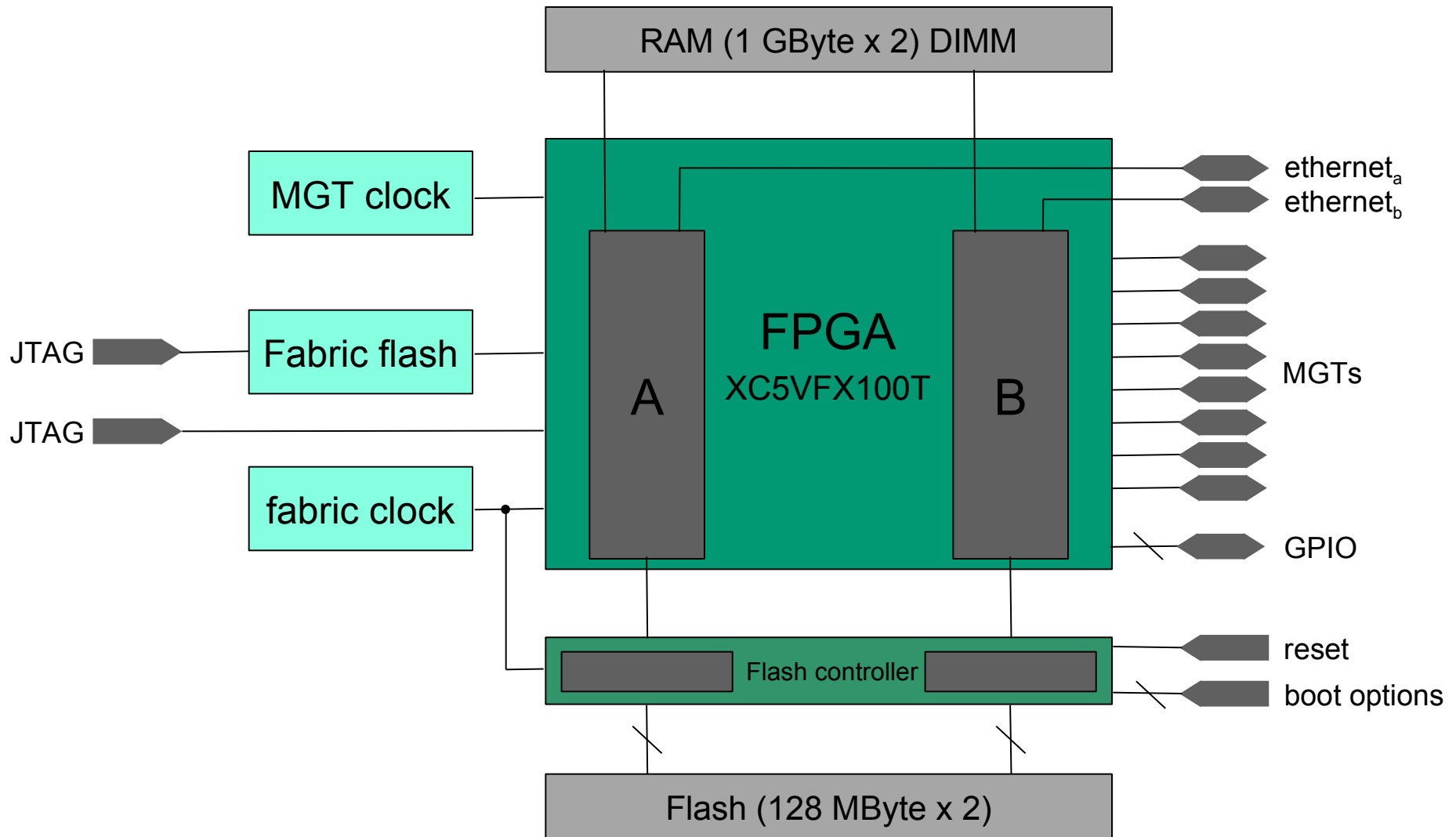
Outline

- Development underway for next generation (“G2”) building blocks
 - Addresses concerns uncovered in R & D phase
- The CIM will have evolutionary change
 - Will not affect its interfaces (therefore, not discussed)
- ATCA
 - Straightforward changes leveraged from gained experience
- RCE implementation changes radically & interfaces significantly
- Why?
 - Needs a radical diet
 - Lack of FPGA resources inhibits plug-in paradigm
 - Need to move to evolving silicon (Virtex-5 & 6)
 - Very different I/O & memory model
 - Plug-in Interface changes (both software & firmware)
 - Increased transfer latency & efficiency
 - New functionality to support more sophisticated plug-ins

Board & ATCA Enhancements

- **Timing interface**
 - Necessary for clock-synchronous systems
 - Necessary (but not sufficient) for “TTC like” interfaces
- **Rationalize JTAG connections & chains**
 - Move to front-panel (allows for better in-situ usage)
- **Provide IPMI board solution**
 - Commercial solution has been cost & licensing prohibitive
 - Hope is to use HEP based solution...
- **Rationalize P3 area**
 - Higher density connector
 - Better pin assignments
 - Separate power & signal
 - Better mechanical interface
 - Examine direct fiber interfaces
- **Research:**
 - Higher density fiber connections (“SNAP-12”) for RTMs
 - 10 Gbps backplanes

Physical Block Diagram of the RCE pair



Flash Configuration System

- Will increase in size & performance
 - Capacity will grow from 32 MByte to (at least) 128 MByte
 - Performance will (at least) double
- Controller moves off Virtex-5 FPGA
 - Reclaims a significant amount of FPGA gate real-estate
 - Addresses licensing issue associated with R-S encode/decode
 - Reset and boot options are now implemented on controller...
- User Interface will be a *Protocol Plugin*
 - Much more efficient interface in multi-use environment
- Increase boot-option space to 32 bits
 - 16 bits for boot images
 - 8 bits for image selection
 - 8 bits for parameterization
 - 16 bits for startup tasks
 - 8 bits for task/image selection
 - 8 bits for parameterization

Move to Virtex-5 & 6

- **Baseline will be XC5VFX100T**
 - Higher clock speed
 - More gates
 - More memory
 - Slices are denser
 - More DSP tiles (with increased functionality)
- **Processor moves from PPC-405 to PPC-440**
 - Approximately 1/3 faster
 - Modest improvements in cache
- **Hardened cross-bar for memory I/O**
 - Will allow the use of native Virtex memory controllers
 - Will free-up a significant amount of FPGA gate real-estate
- **Virtex 4 will be phased out**
 - I/O & memory models are so different, not worth supporting

RCE Memory Subsystem

- Based on native cross-bar
 - Only available starting with Virtex-5
 - Enables use of native memory system
 - Permits installation of industry standard/commodity memory
 - DIMM packaging (will allow replacement)
 - Benefits
 - Regains a significant amount of FPGA gate real-estate
 - Performance doubles: 4 - 8 GBytes
 - Size will go from (max) $\frac{1}{2}$ GBytes/s to 1-2 GBytes/s
- New feature:
 - All memory (at cacheline granularity) will be remappable
 - Allows for user memory placement semantics
 - Instrumental in allowing true "zero-copy" transfers

Protocol Plug-In Model

- Will be implemented through native cross-bar (128 bits wide)
 - No more PIC blocks
 - Will allow for up to eight (8) plug-ins
 - One is reserved for Configuration Flash System
 - One for 10-GE Ethernet MAC
 - Leaving up to six (6) user defined
- Memory management hidden from both software & firmware
 - Allocation policy is now variable sized
- Software Interface is considerably simpler
 - Memory management is mostly hidden
 - No more formal differentiation between header & payload
 - Remapping allows arbitrary definitions of packet boundaries
 - Remapping will allow transferred data to be user “placed”

New Plug-Ins

- Ethernet MAC
 - Layer-3 rather than Layer-2 interface
 - Backward compatible with RTEMs stack
 - Will implement:
 - packet fragmentation & reassembly
 - TCP windowing algorithm
 - How to integrate this functionality with BSD stack is an open question...
- S-link
 - Allows development with current infrastructure
- GBT
 - Not sure yet what this means...