A Scalable Processor with Embedded Software for Large-Scale Scientific Applications

Daniel Alex Finkelstein and Haldun Hadimioglu
Polytechnic University, Brooklyn, NY, USA

Outline

1. Motivation/Goals and Related Work
2. Peripheral Context
3. Experimental Platform
4. Associative Streaming Memory Processor
5. Conclusions

Motivation & Goals

1. Address the memory wall problem
   - More intelligence in memory and storage usage
   - Target the CPU, main memory, and peripherals
   - Exploit regular scientific large-scale applications
     - For high-speed processing, reconfigurable fabric is needed
     - Use embedded RISC cores for ‘slow’ tasks and algorithms too difficult to directly map onto FPGA fabric.

Motivation & Goals

2. RISC + FPGA fabric combination
    - Schedules large & fine grained data movements
    - Associates data streams, data within streams, and mixes data streams on-chip or using latency-controlled peripherals (cache)
3. Scalability to handle large problems: Efficient Multi-Chip Configurations
4. Different mixture of processor-memory-fabric-peripheral composition

Related Work

1. Molen & Garp
   - Uses both FPGA reconfigurable logic and processor cores, but for application acceleration. Garp also allows FPGA direct access to main memory.
2. RAW
   - Programmable insofar as the instructions and data can be rerouted through the tiles via switching.
3. RAMP
   - Though a simulation environment, some of the (many) proposed features include dataflow architectures for programming languages, high-bandwidth peripherals, and reusable logic cores for the FPGA fabric.
4. RSVP
   - Decoupled operand prefetch, vector stream units, and detailed vector stream descriptors for media-rich applications, but can be generalized.

Supercomputing Applications

Our criteria:

- Large data sets
- Floating point operations
- Regular data structures

- SPEC CPU 2000 FP suite
  - Stable, predictable memory profiles

Peripheral Bandwidth

Intel

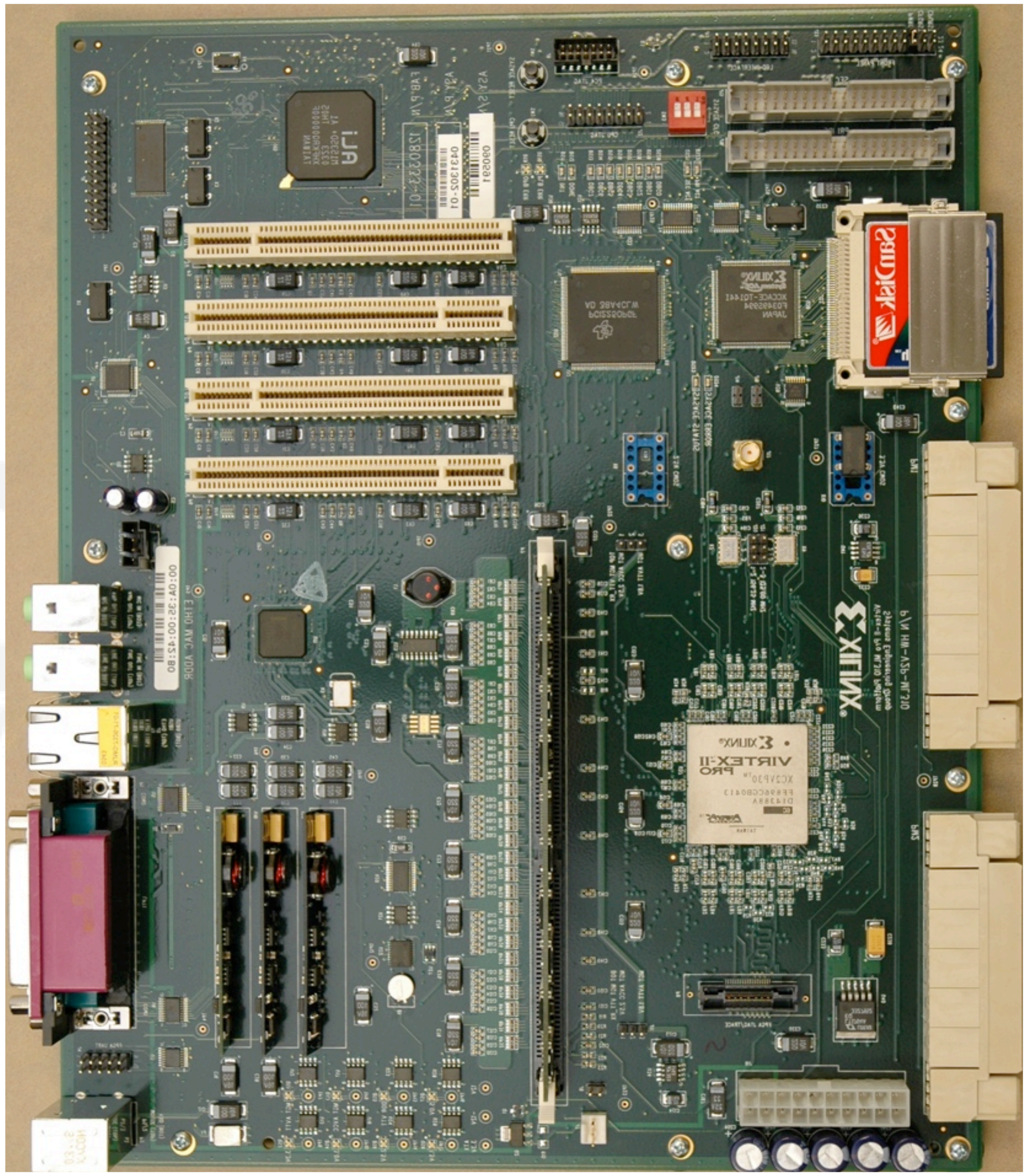
Intel 975X Express Chipset	3 Gbps	SATA
Intel 975X Express Chipset	85.6 Gbps	DDR2 DRAM
Intel IXP2855 Network Processor	57.6 Gbps	RDRAM
Intel IXP2855 Network Processor	10 Gbps	Ethernet

Xilinx

DDR DRAM	400 Mbps	Virtex 4
DDR2 DRAM	667 Mbps	Virtex 4
Rocket I/O	75 Gbps	Virtex-II Pro
Infiniband	10 Gbps	Virtex-II Pro X

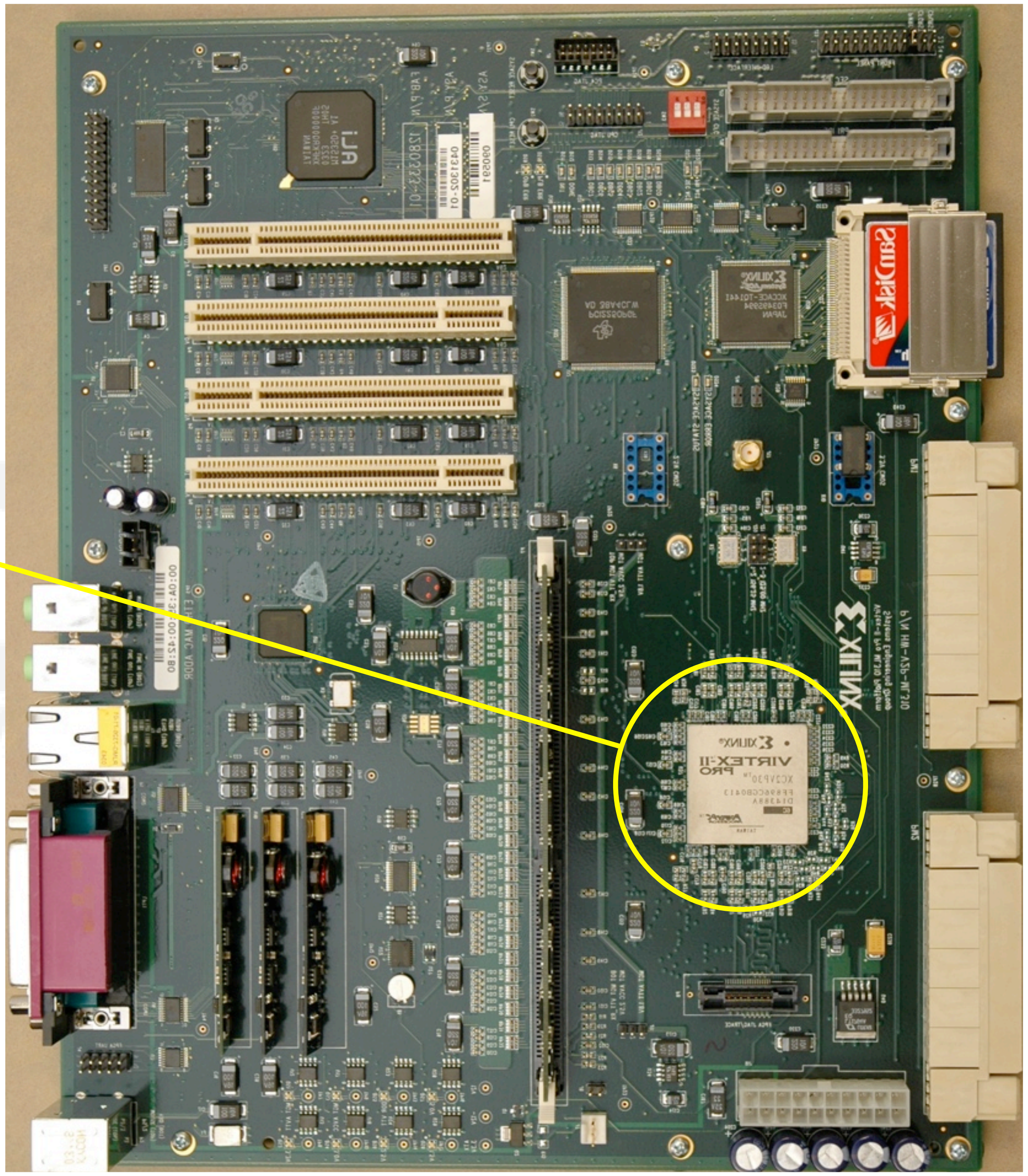
Peripherals are getting faster, but are usually separated from the processors by several levels of the memory hierarchy.

ML310 Development Board



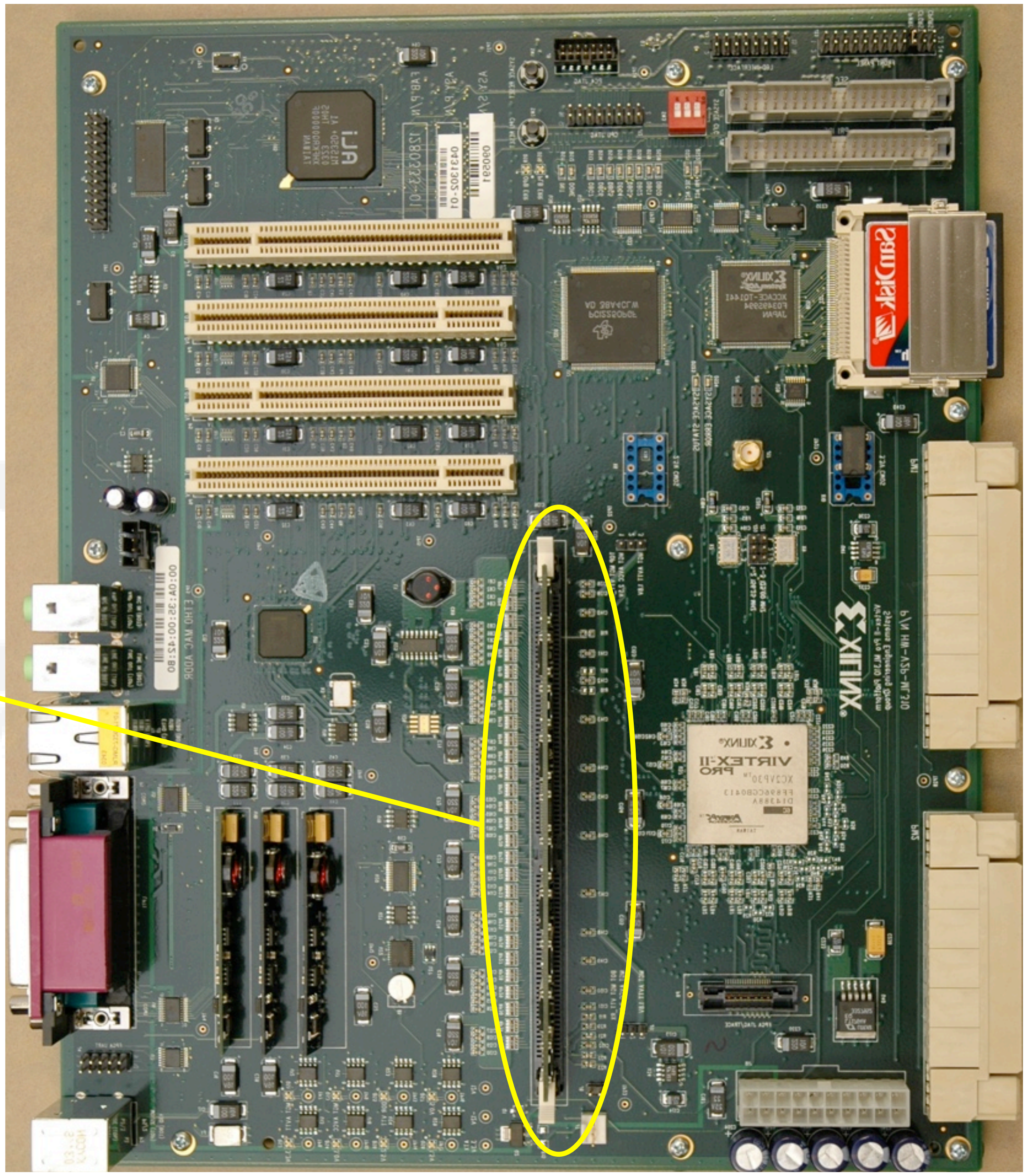
ML310 Development Board

- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores



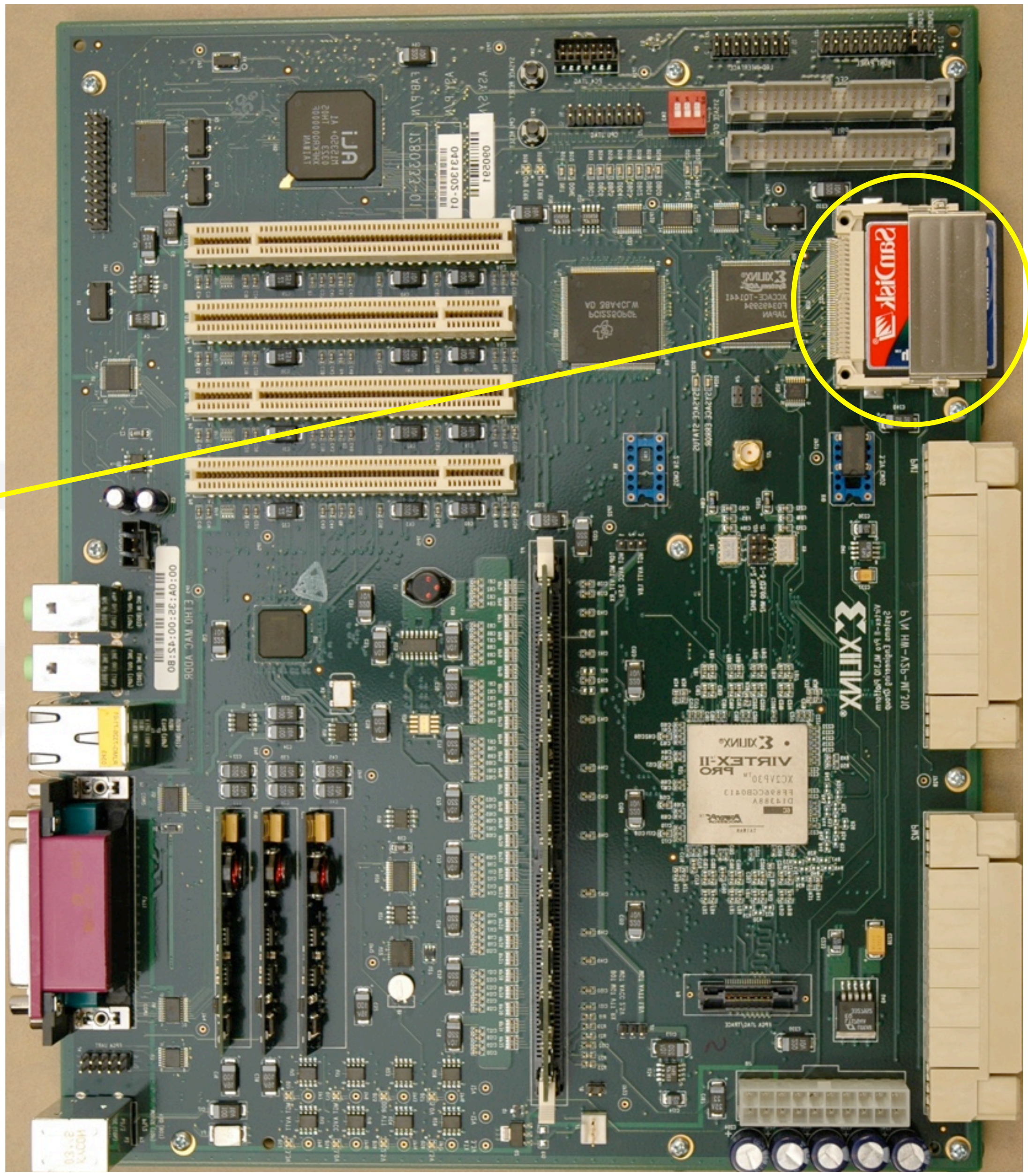
ML310 Development Board

- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
- 256 MB DDR DRAM



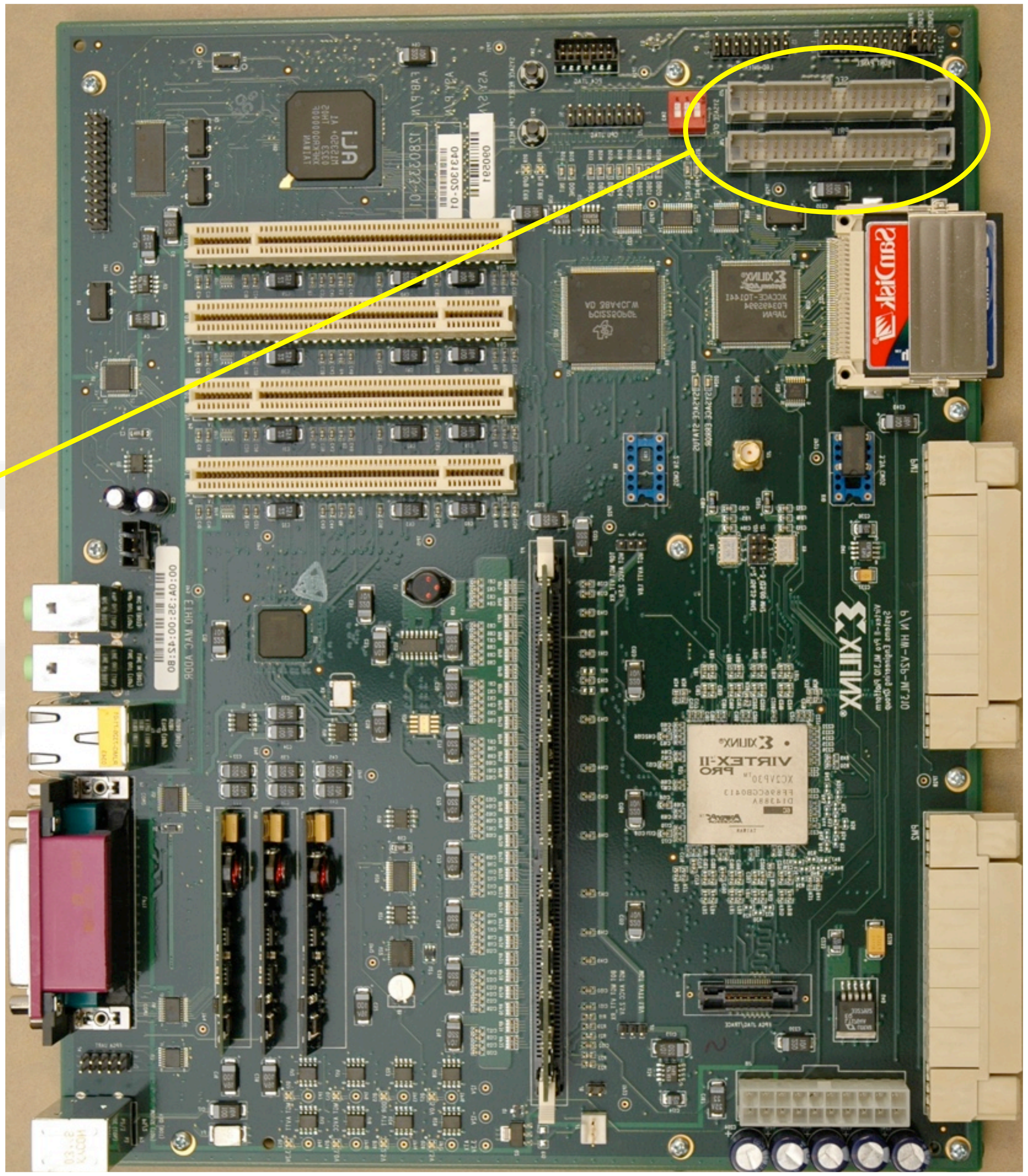
ML310 Development Board

- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
- 256 MB DDR DRAM
- SanDisk



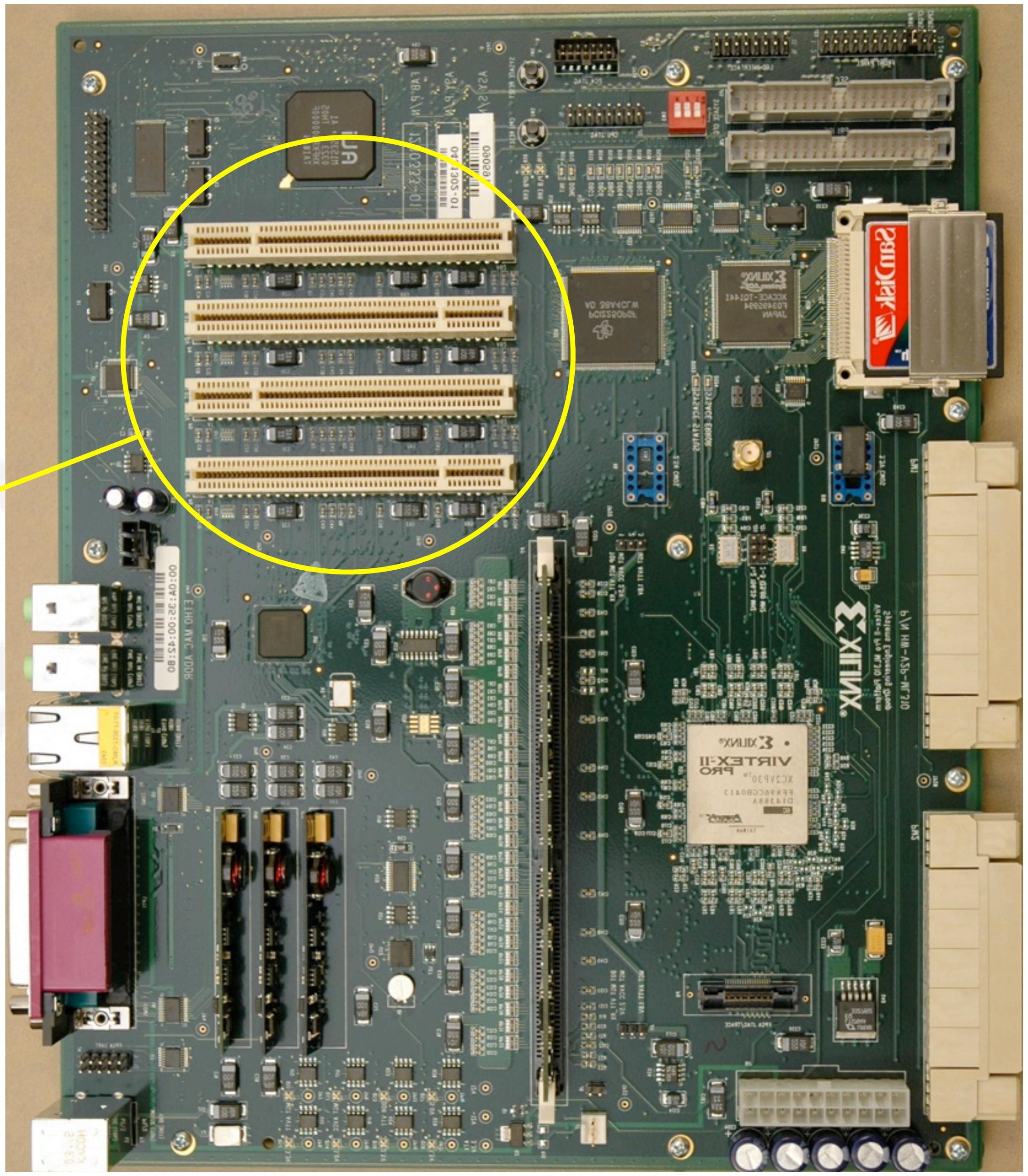
ML310 Development Board

- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
- 256 MB DDR DRAM
- SanDisk
- IDE interfaces



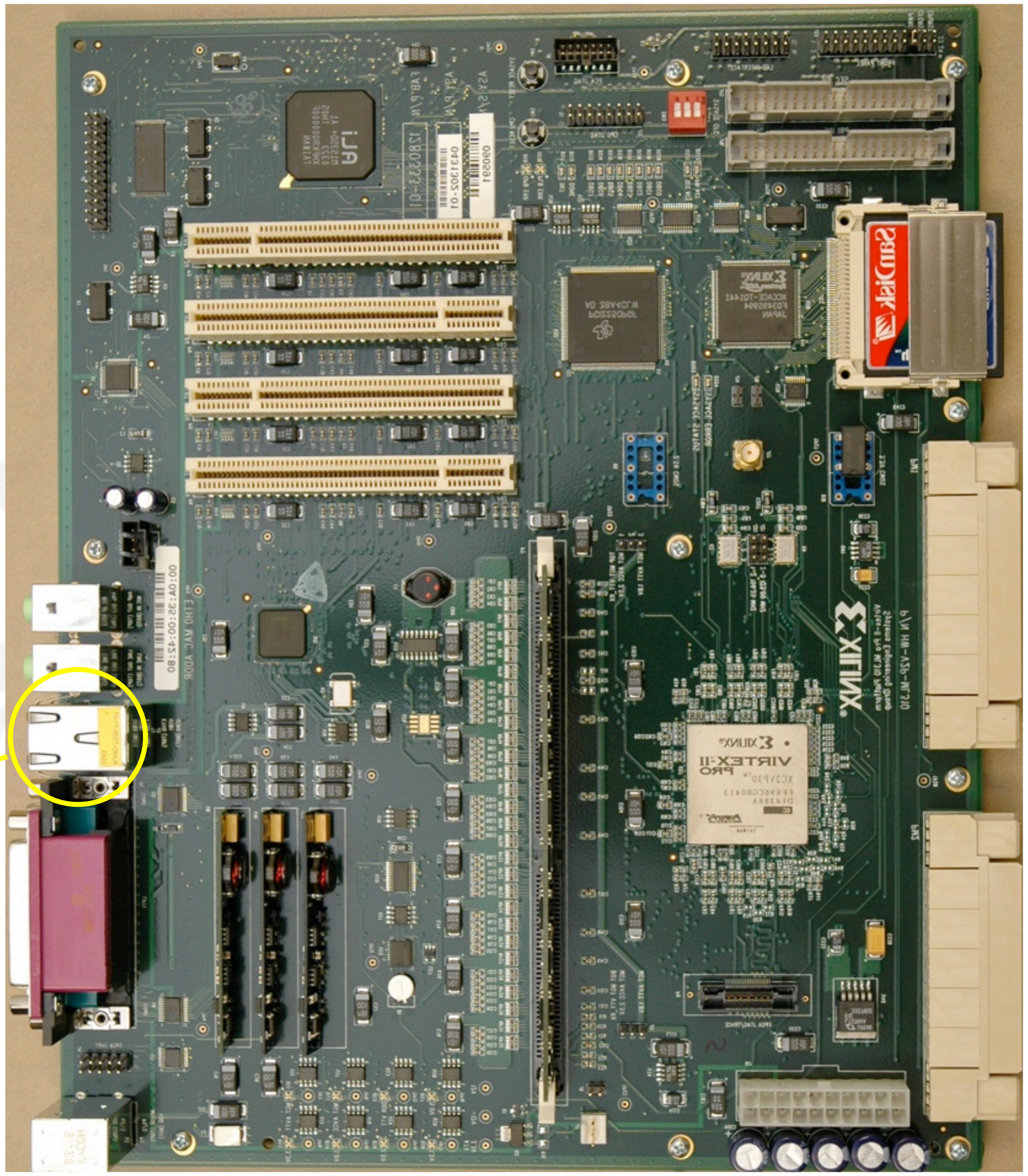
ML310 Development Board

- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
- 256 MB DDR DRAM
- SanDisk
- IDE interfaces
- PCI



ML310 Development Board

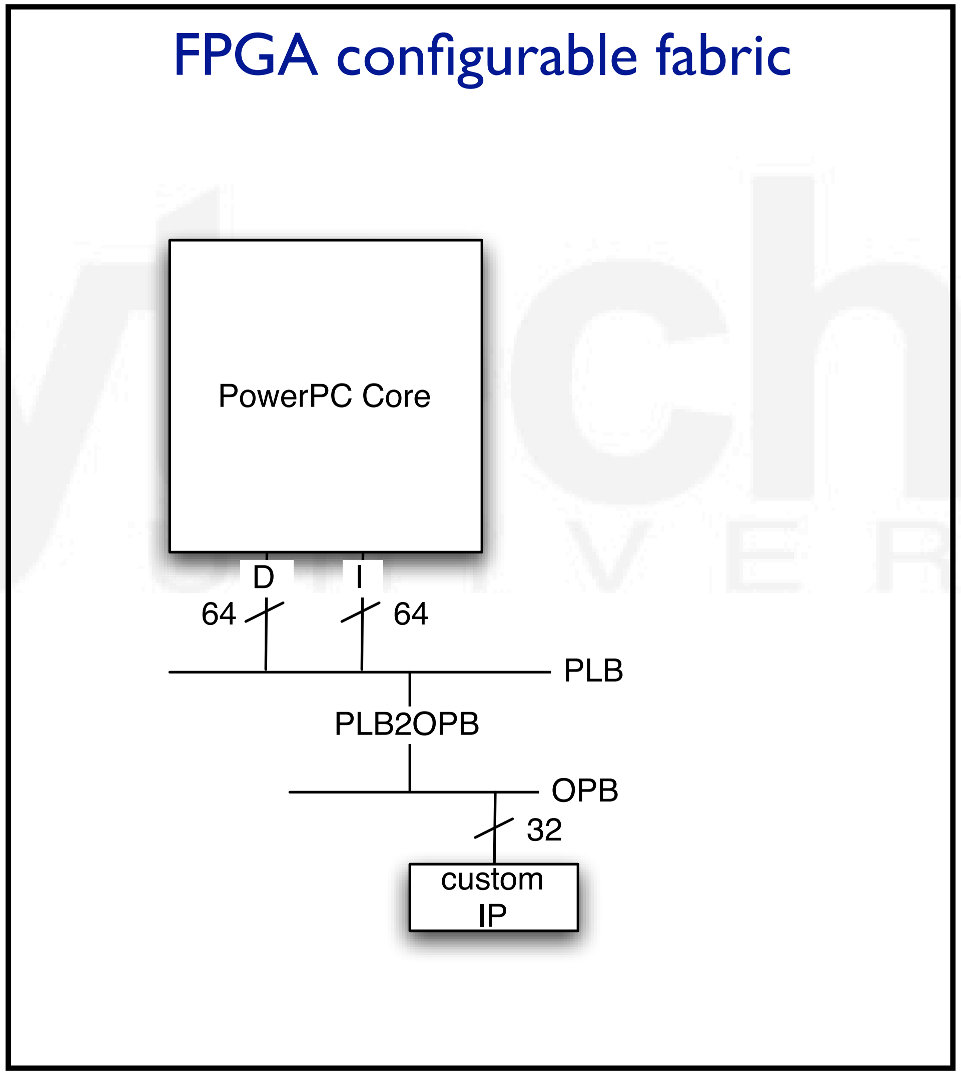
- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
- 256 MB DDR DRAM
- SanDisk
- IDE interfaces
- PCI
- Ethernet



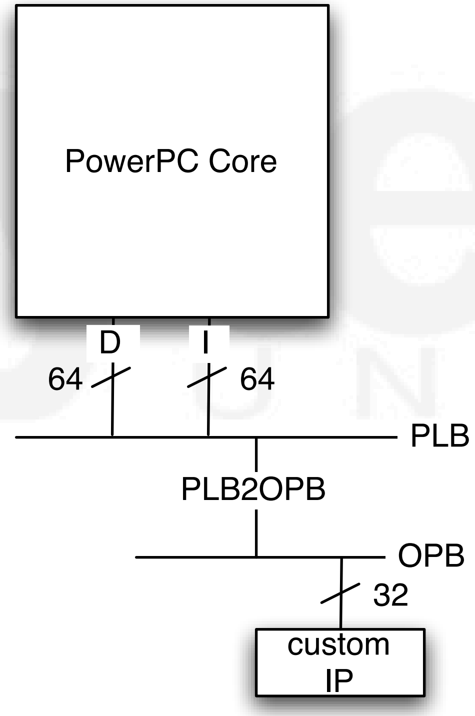
PowerPC - Custom IP Interface

PowerPC - Custom IP Interface

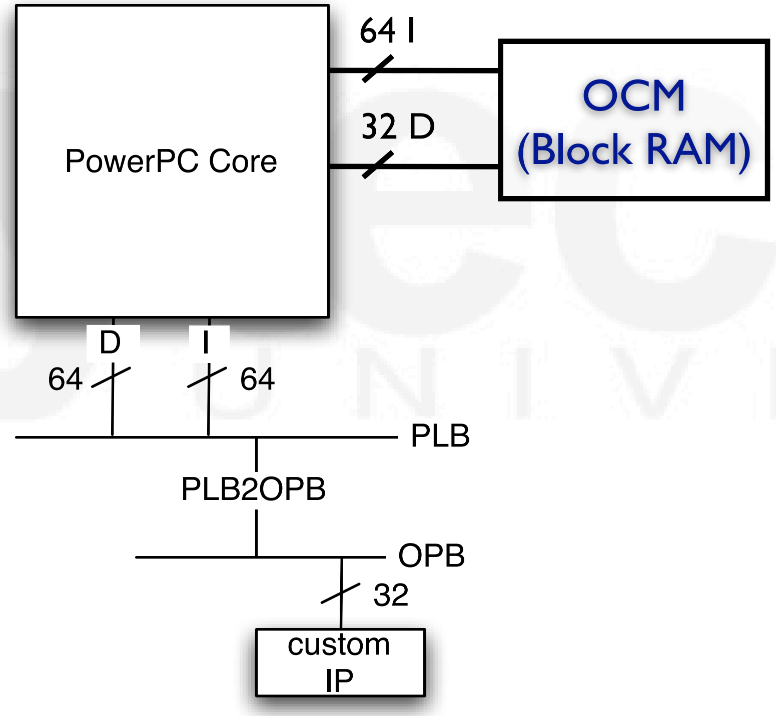
PowerPC - Custom IP Interface



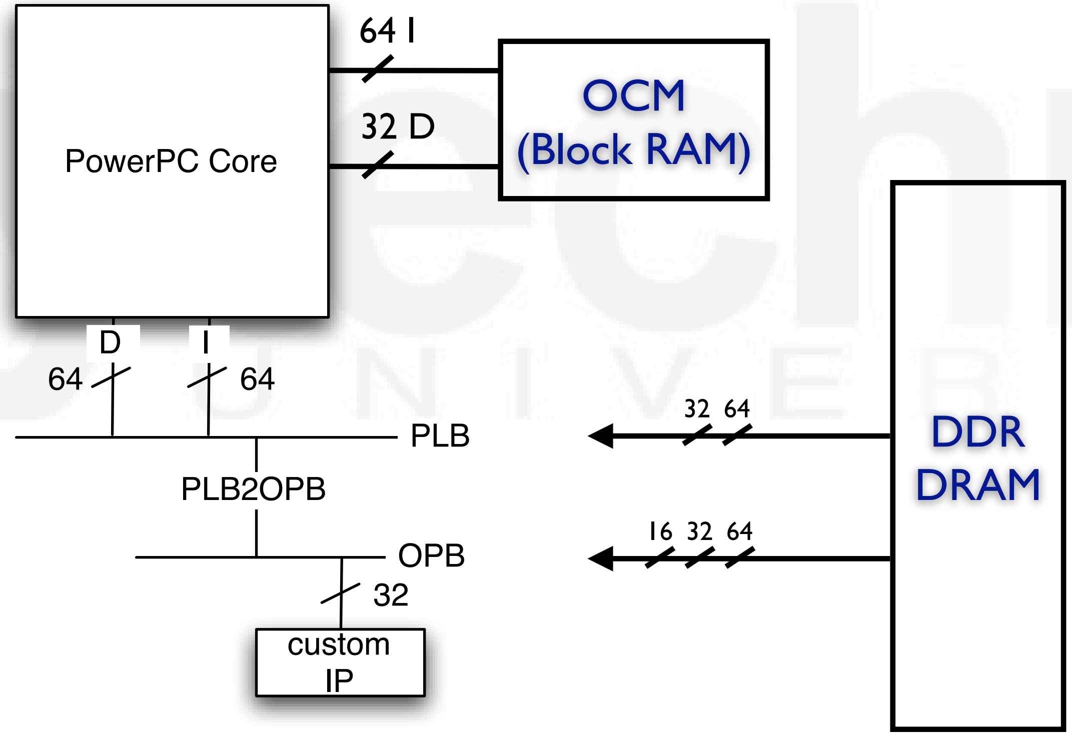
PowerPC - Custom IP Interface Memory Options



PowerPC - Custom IP Interface Memory Options



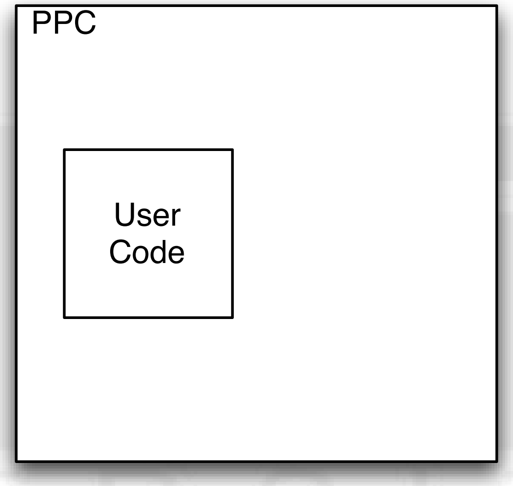
PowerPC - Custom IP Interface Memory Options



PowerPC: A High Level Controller

PowerPC: A High Level Controller

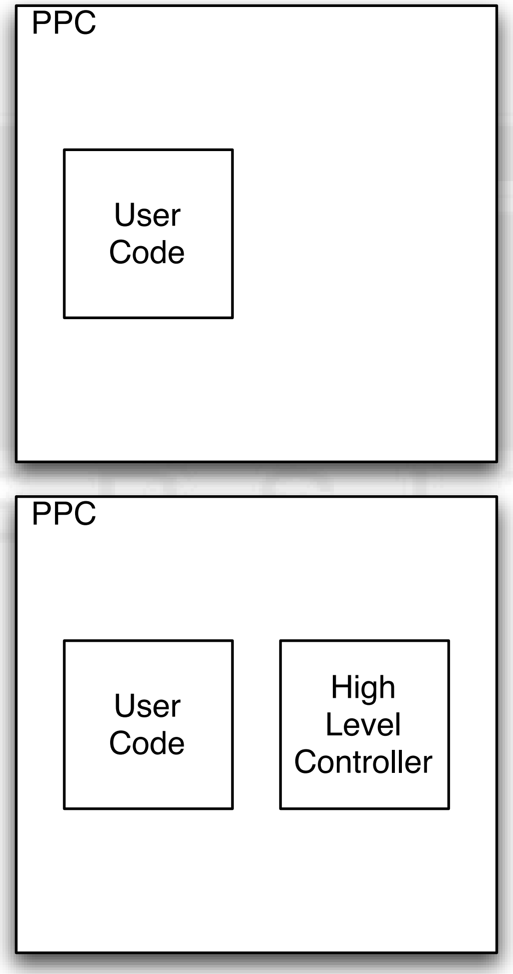
- PowerPC (PPC) core (with some hardware additions) can execute user programs



PowerPC: A High Level Controller

- PowerPC (PPC) core (with some hardware additions) can execute user programs

- PPC core may also perform control and monitoring functions for the processor system



High Level Controller Features

- Peripheral latency monitoring

Store delays between data request and data arrival for each peripheral

- Peripheral data prefetch control

Begin retrieving data streams and process immediately or buffer locally, based on latency values

- Peripheral resource scheduling

Control access to peripherals

- Coarse-grained data association, mixing, and buffering

Vectorized data streams allow the controller to associate streams with each other (dependencies), re-order data into memory buffers (useful in matrix-matrix multiplication and transposes), etc.

Low Level Controller Features

- The LLC is the Custom IP (shown earlier) connected to the OPB
- The LLC resides within the FPGA's configurable logic blocks (CLBs)
- The LLC performs both computations on the data streams and fine-grained data mixing

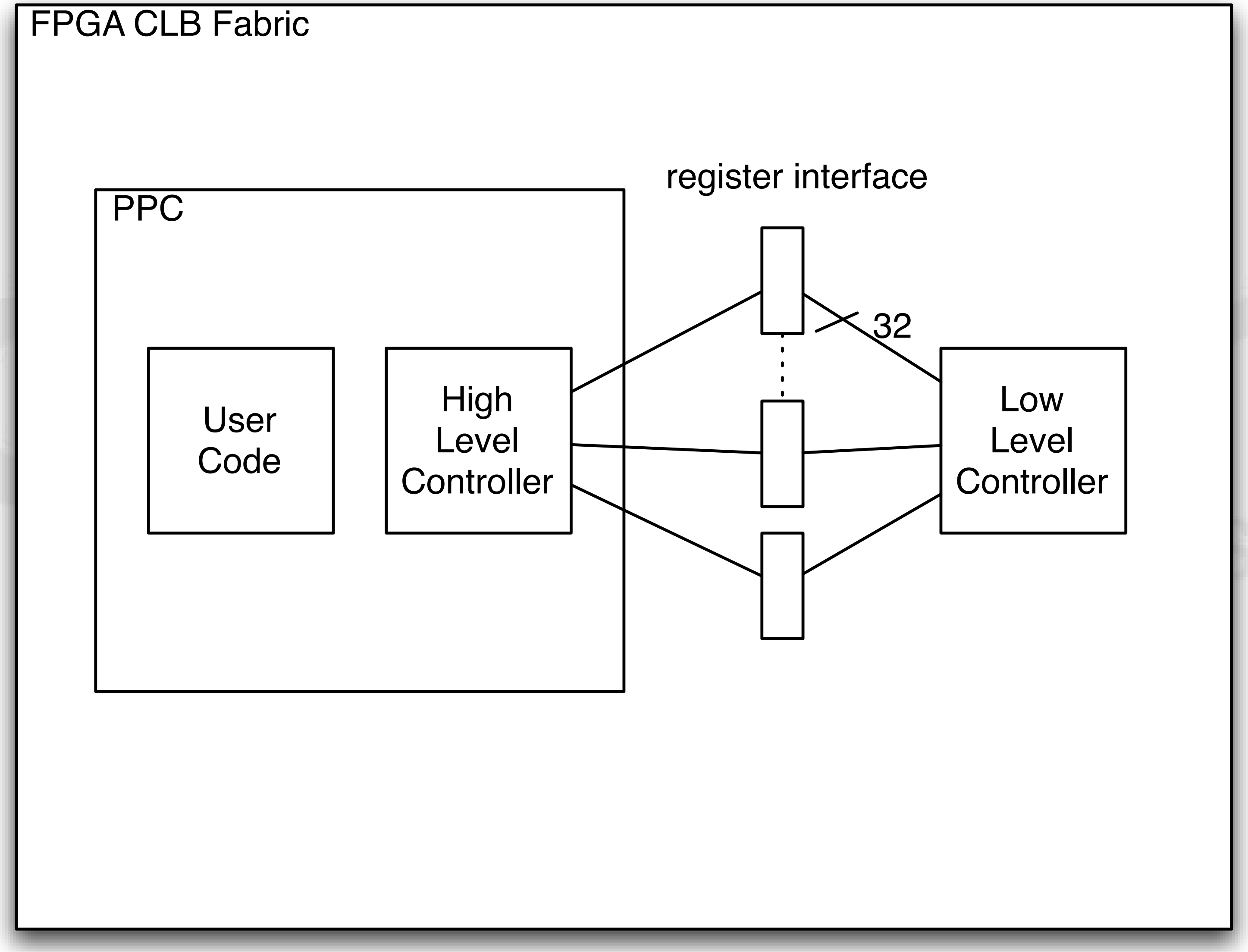
Low level controller is unique for each application.

Reusable functional logic can be dynamically mapped onto CLBs at compile-time or run-time.

Control signals exchanged with HLC to indicate status of buffers, dout_rdy, etc.

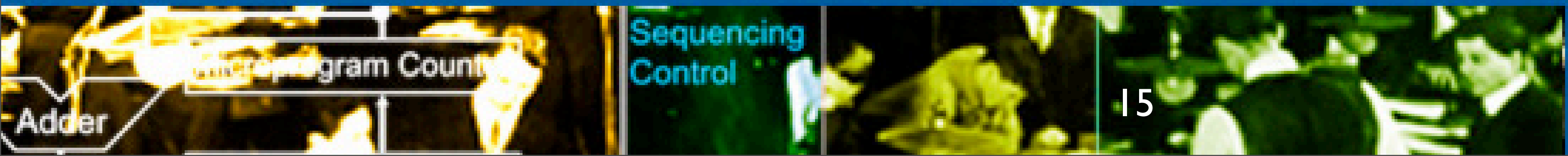
Data element arithmetic operations (int/FP) performed at this level. Elements in streams may be remixed to satisfy output constraints, dependent instructions, etc.

HLC - LLC Interaction



Conclusions

- Fast peripherals need not be limited by intermediate controllers, buses, and operating systems.
- Controller logic can be integrated alongside traditional processor components in a single package.
- Intelligent use of memory peripherals on local buses reduces processing latencies.
- Embedded software saves resources better devoted to time-sensitive computation.



Acknowledgments

We wish to acknowledge the support of Xilinx in providing us with ML310 development boards, design tools, and technical support.

This work was supported in part by a research fellowship from the U.S. Department of Education GAANN.

Thank you.

contact: dfinke01@cis.poly.edu