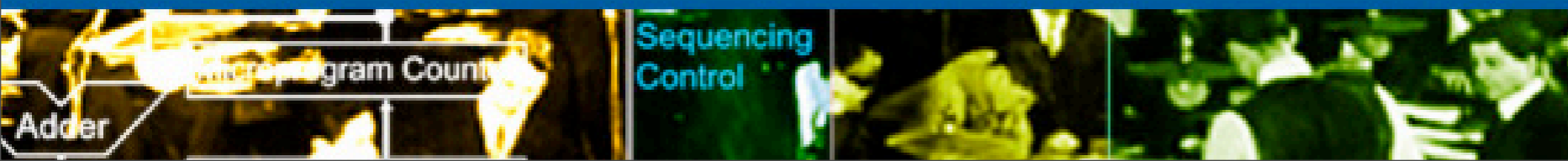


An Associative Streaming Memory Processor

IBM T.J. Watson Research Center
May 18, 2006

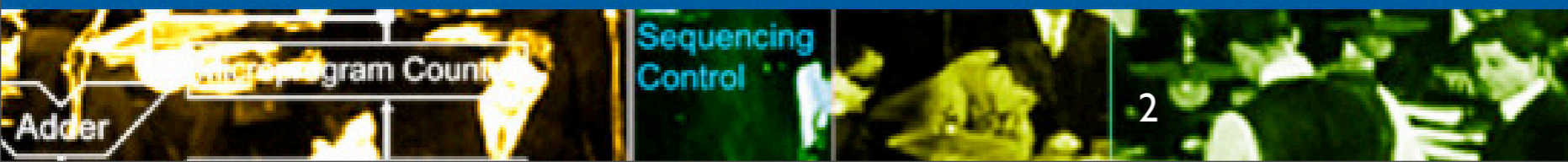
Daniel Alex Finkelstein and Haldun Hadimioglu
Polytechnic University, Brooklyn, NY, USA

© 2006 Polytechnic University



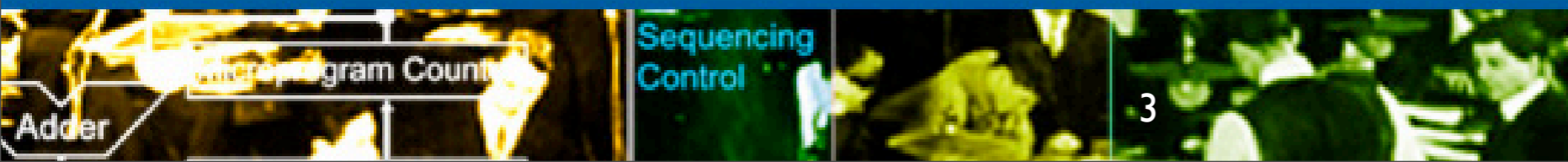
Outline

1. Motivation and Goals
2. Related Work
3. Design and Algorithms
4. Experimental Platform
5. Conclusions



Motivation & Goals

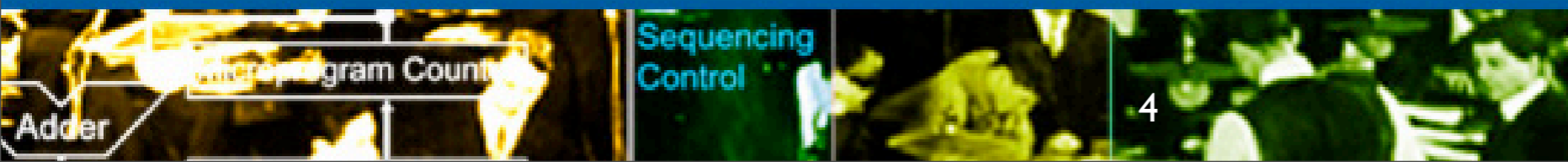
- I. Address the memory wall problem
 - More intelligence in memory and storage usage
 - System-wide design
 - Target the CPU, main memory, and I/O



Motivation & Goals

2. Application runs on RISC/reconfigurable fabric combination

- Schedules large & fine grained data movements
- Optimizes composition of processor-memory-fabric-I/O
- **Associates**
 - data streams
 - data within streams
- **Mixes**
 - data streams
 - on-chip
 - using latency-controlled I/O



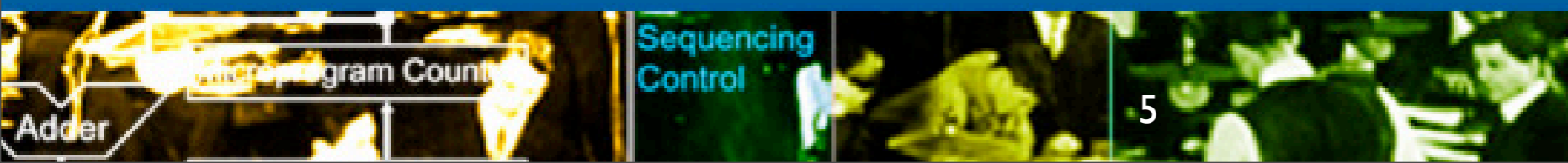
Motivation & Goals

3. Exploit regular large-scale scientific applications

- Use reconfigurable fabric for high-speed processing
- Use embedded RISC cores
 - ‘Slow’ tasks
 - Algorithms too inefficient to directly map onto FPGA fabric.

4. Scalability to handle large problems

- Multi-chip configurations



Related Work

1. Molen & Garp

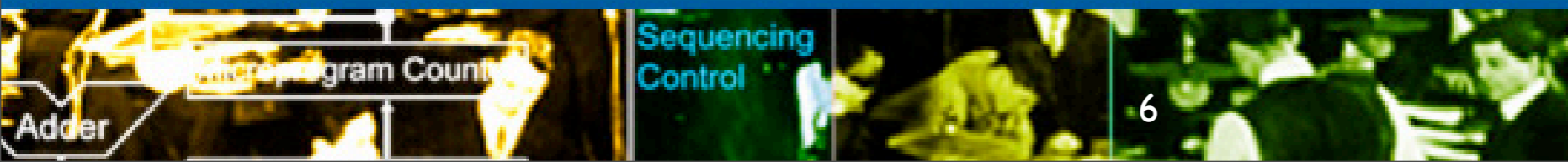
- Traditional application acceleration
 - Uses both FPGA reconfigurable logic and processor cores
- Garp adds FPGA direct access to main memory

2. RAMP simulation environment

- Dataflow architectures for programming languages
- High-bandwidth I/O
- Reusable cores for the FPGA fabric

3. RSVP™ for media (but can be generalized)

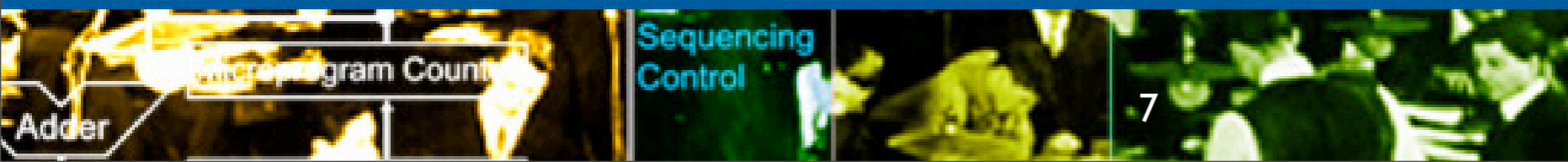
- Decoupled operand prefetch
- Vector stream units
- Detailed vector stream descriptors



Large-scale scientific applications

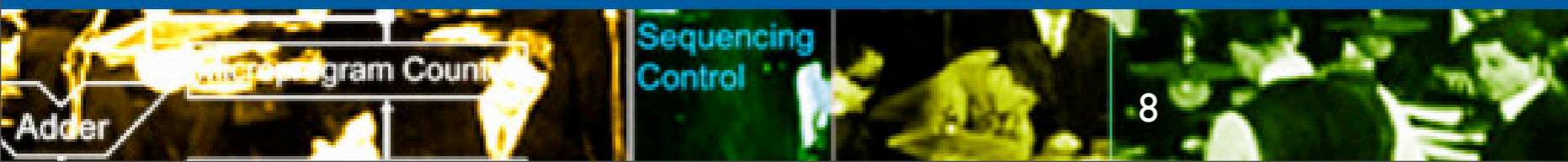
Our criteria:

- Large data sets
- Floating point operations
- Regular data structures
- SPEC CPU 2000 FP suite



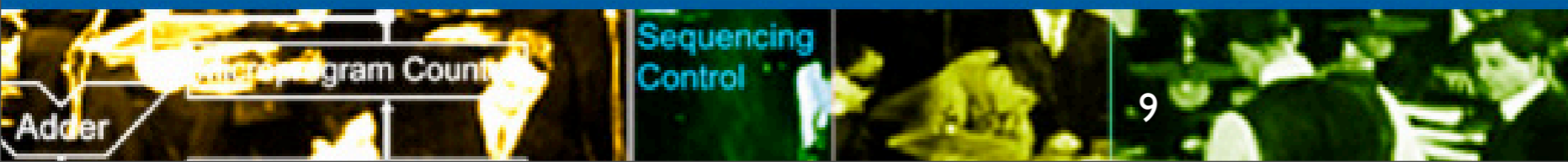
Contributions

- **Associate** data streams
- **Mix** data on-chip/off-chip
- Reduce stores and loads
- Monitor runtime resource usage
 - Reprogram/reconfigure to meet runtime performance parameters
- Decrease instruction density



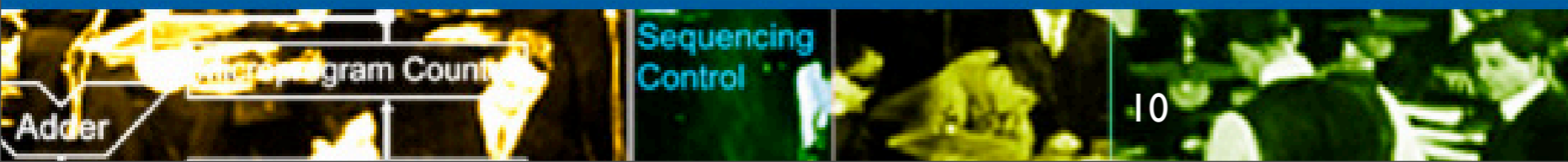
Data Stream Mixing

- Instruction prepended to input data stream
- Input and result streams recirculated on-chip if values are reused and small
 - Sent off-chip
 - To be recirculated when buffers are full
 - No further references are detected



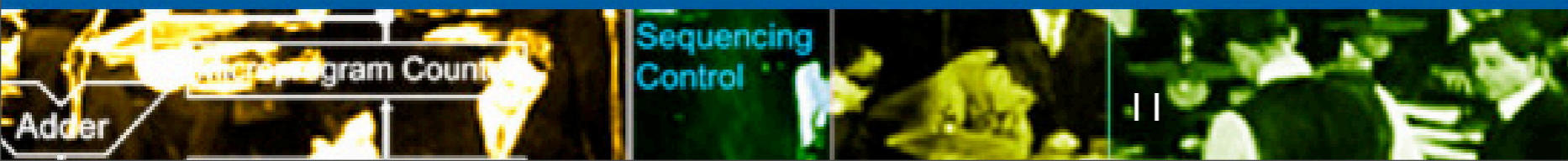
Stream Composition

- Header contains:
 - Opcode (instruction)
 - Data structure (if any)
 - ▶ Geometry (rows, columns, etc.)
 - Data element type
 - Length, or terminal symbol/signal
 - References of dependent streams, or memory locations



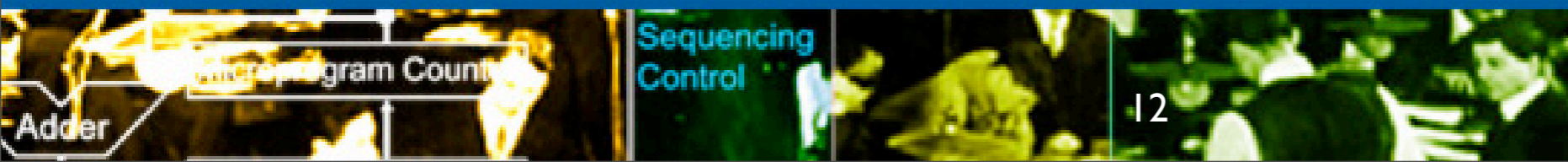
Design

- Our system (an “ASMP” chip) has two primary subsystems
 - High Level Controller
 - Low Level Controller



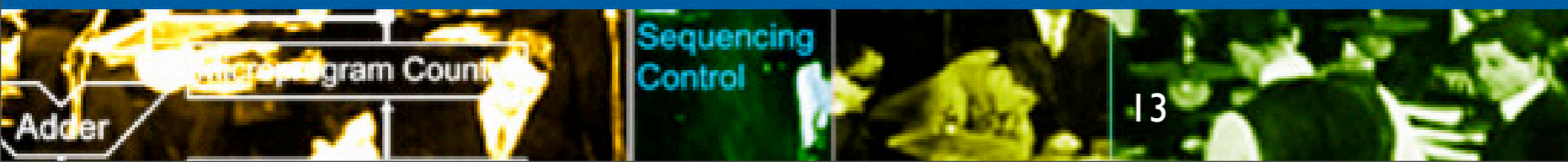
High Level Controller (HLC)

- Schedules large grained data movements
- Optimizes composition of processor-memory-fabric-I/O
- **Associates**
 - large data streams
- Helps Low Level Controller to **mix**
 - data streams
 - on-chip
 - using latency-controlled I/O



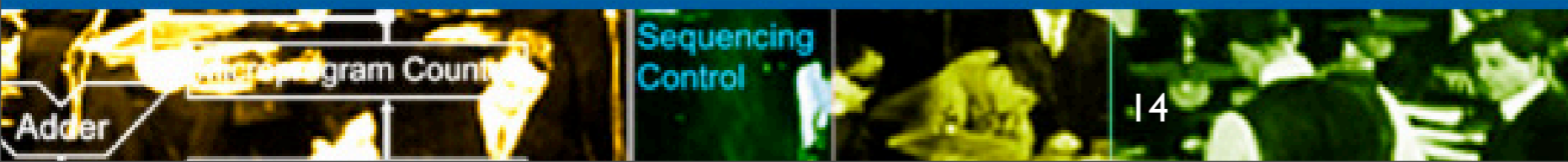
High Level Controller Features

- Off-chip latency monitoring
 - [Store delays between data request and data arrival
- Off-chip data prefetch control
 - [Begin retrieving data streams and process immediately or buffer locally, based on latency values
- Off-chip resource scheduling
 - [Control access to resources
- Coarse-grained data association and buffering
 - [Associate streams with each other (dependencies)
 - [Re-order data into memory buffers



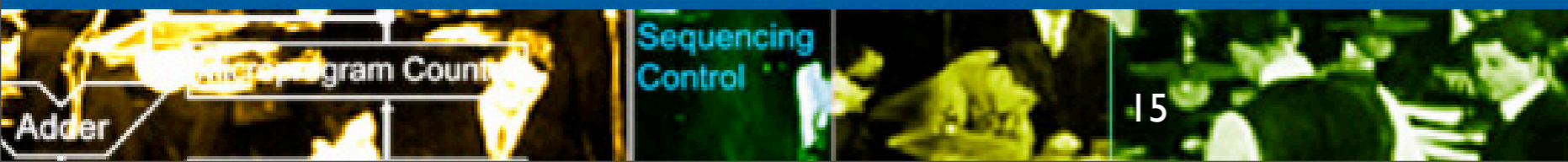
Low Level Controller (LLC)

- Schedules fine grained data movements
- **Mixes**
 - data streams
 - on-chip
- Performs computations on streams



Low Level Controller Features

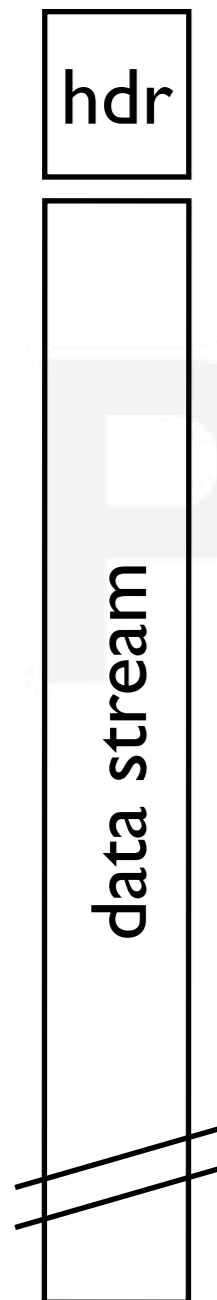
- The LLC is Custom IP
 - [Low level controller is unique for each application.
- The LLC resides within the FPGA's configurable logic blocks (CLBs)
 - [Reusable functional logic can be dynamically mapped onto CLBs at compile-time or run-time.
- The LLC performs both computations on the data streams and fine-grained data mixing
 - [Interaction with HLC
 - [Data element arithmetic operations (int/FP) performed at this level.
 - [Elements in streams may be remixed to satisfy output constraints, dependent instructions, etc.



Stream Operation Example

1. HLC processes header
 - a. Retrieves header from memory
 - b. Extracts opcode, data type, length
 - c. Performs dependence detection
2. HLC assigns stream reference/handle
3. LLC resources allocated
4. Stream handle passed to LLC
5. HLC awaits completion signal from LLC

vector stream



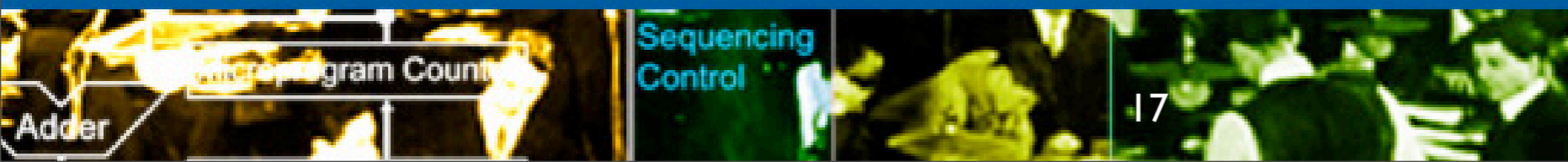
Data Stream Mixing Actions

Problem

- Data streams might not be in the optimal structure for a given operation
- Dependences between streams might benefit from speculation
- Multiple streams are used in a single operation

Action

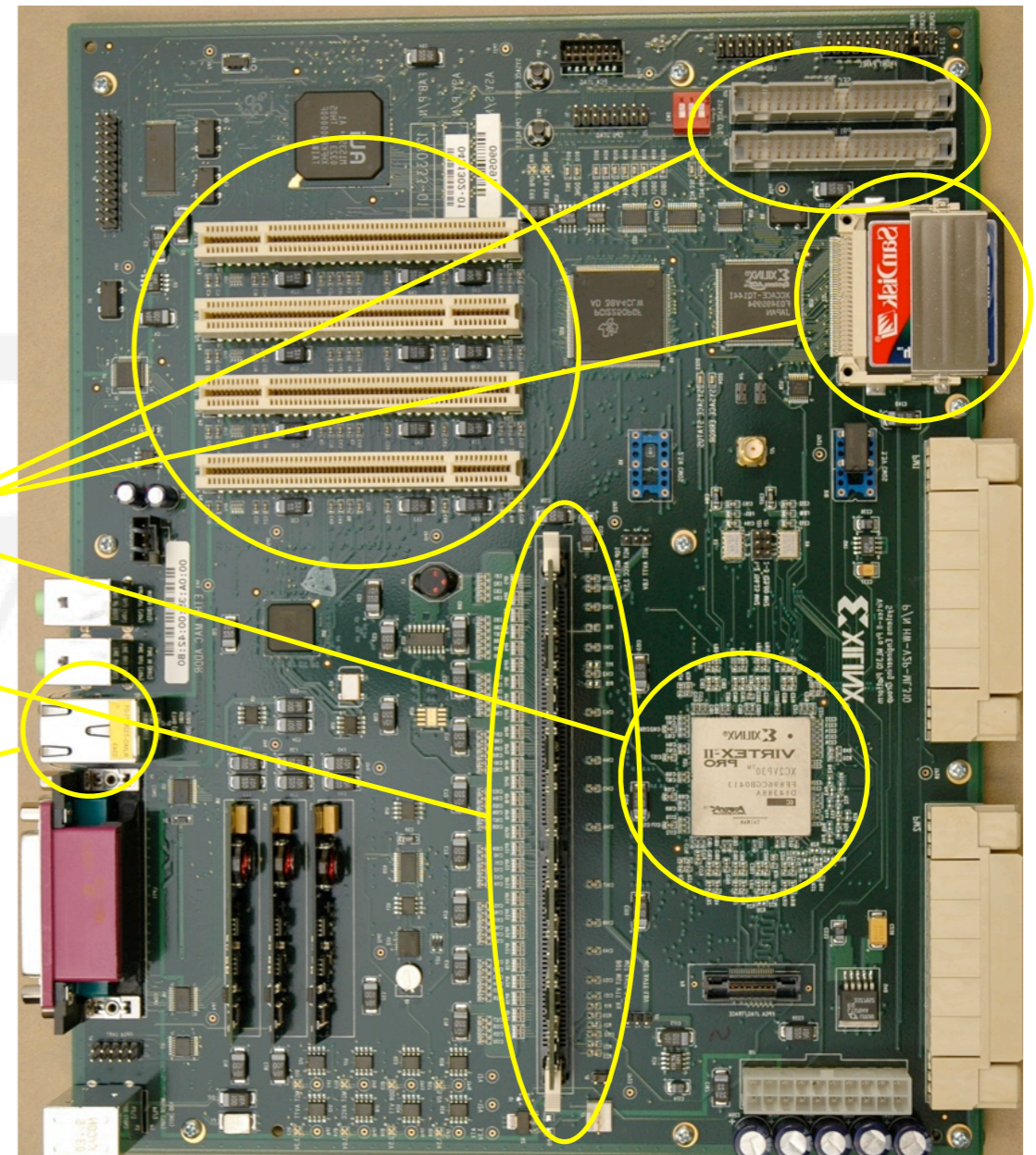
- Recirculate stream via on-chip RAM or off-chip DRAM, into optimal structure (mix)
- Send stream to multiple LLCs or HLCs for speculation
- If sufficient on-chip resources are available, send all streams.
- If data is segmented among streams, recombine into single stream (associate)



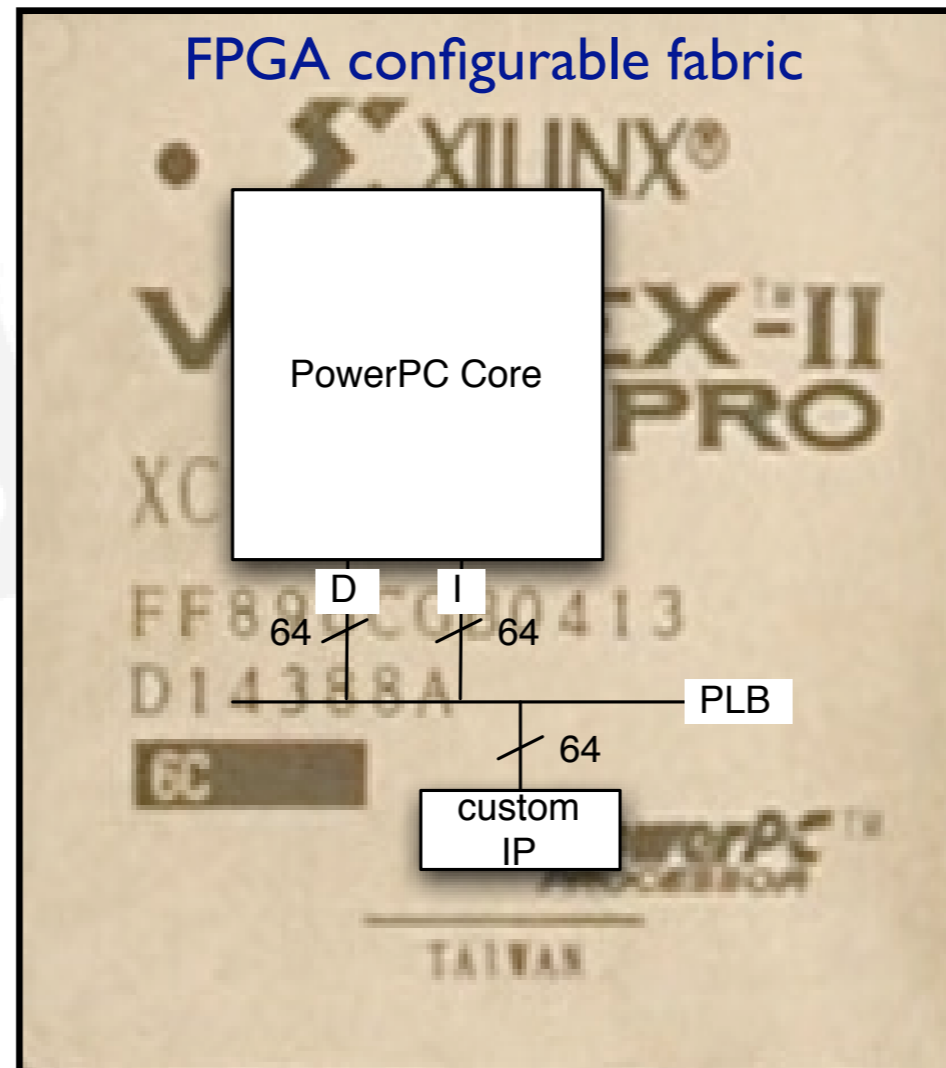
Experimental Platform

ML310 Development Board

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

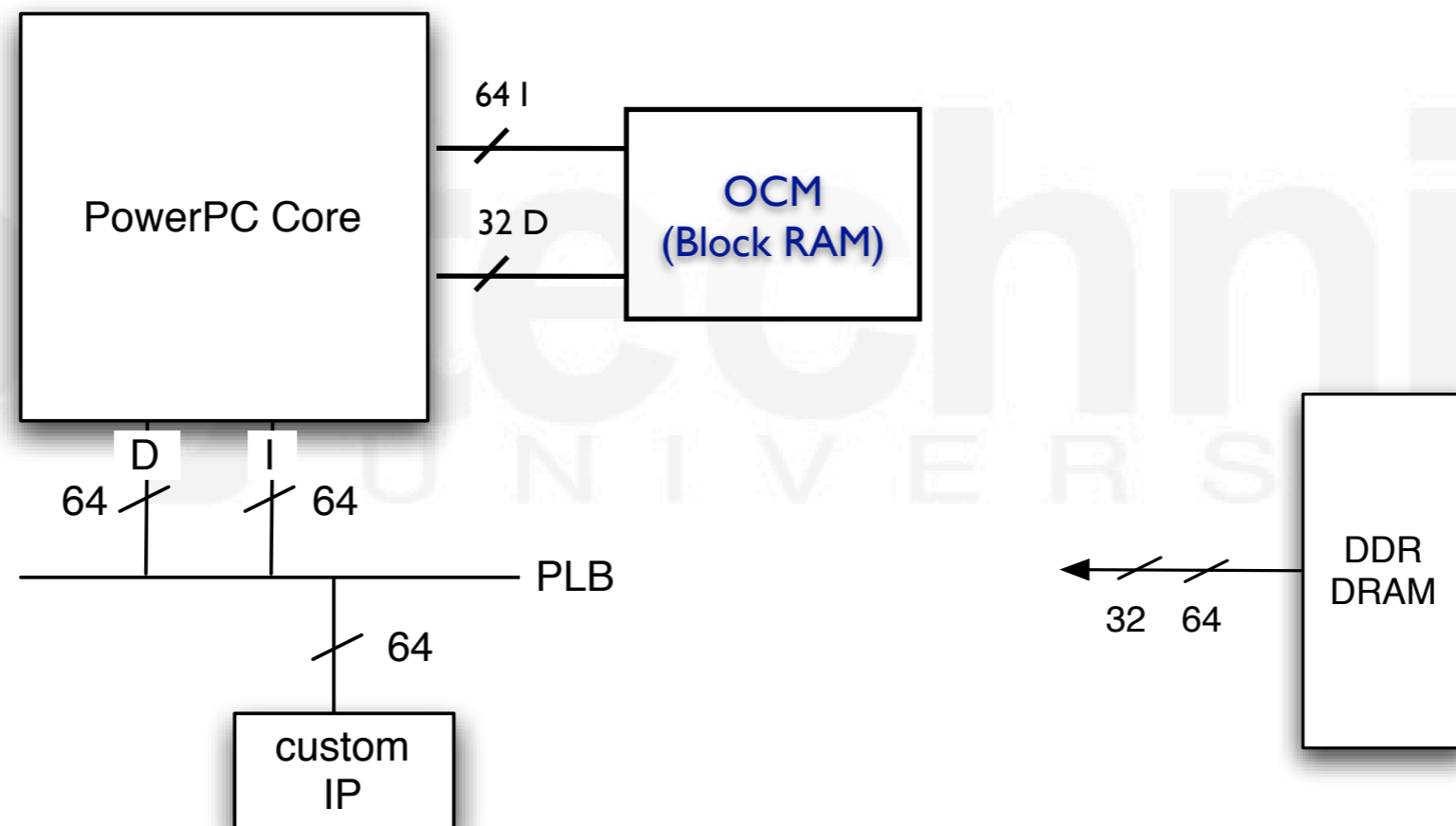


PowerPC - Custom IP Interface



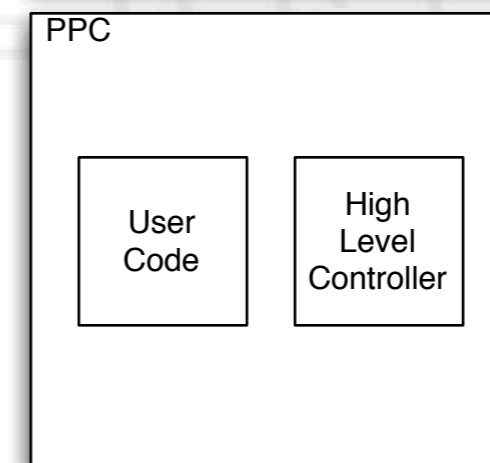
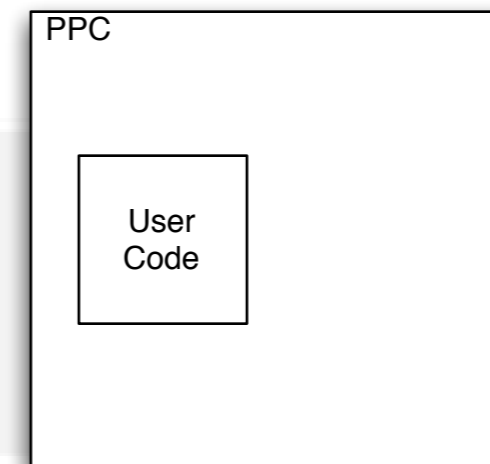
Sequencing
Control

PowerPC - Custom IP Interface Memory Options

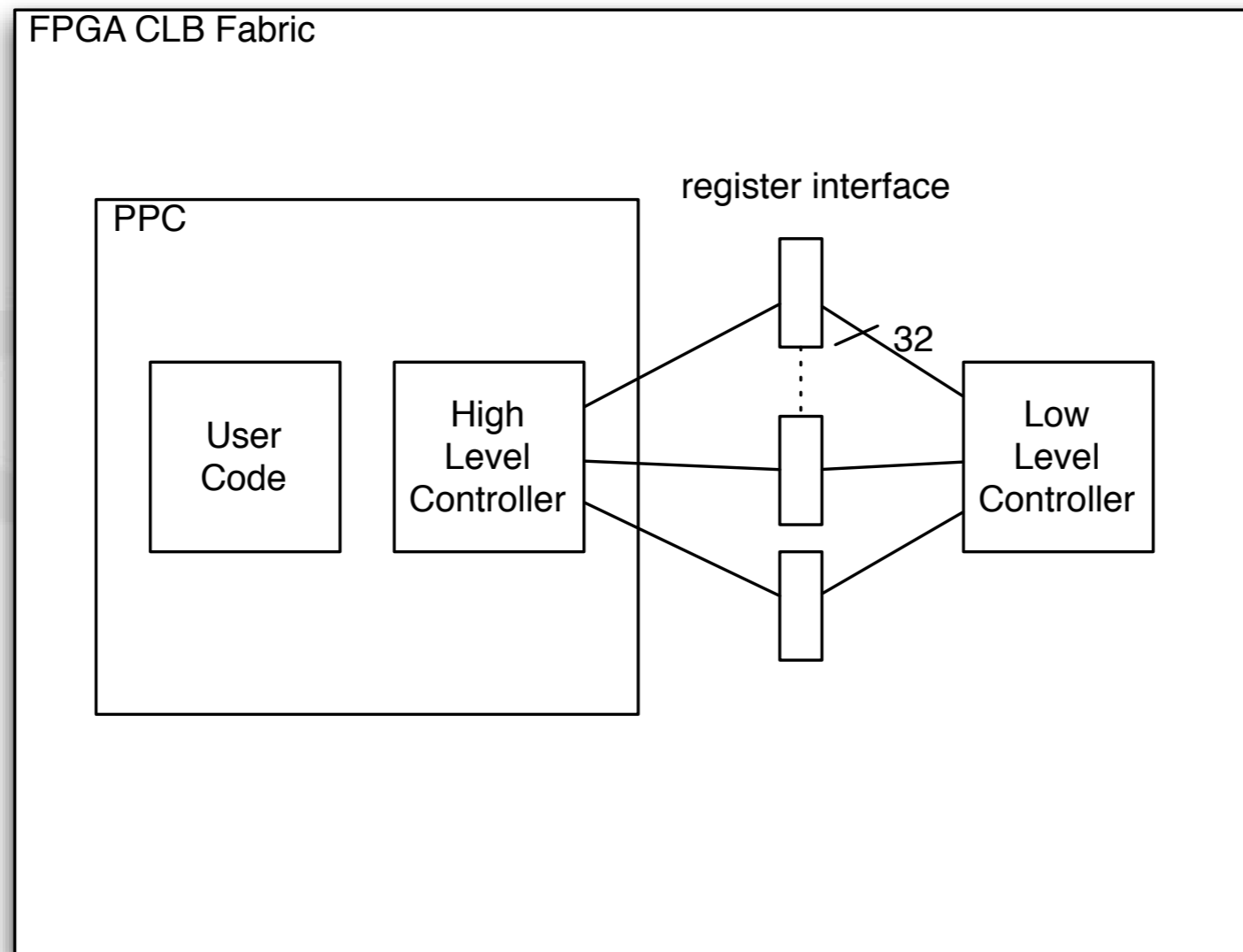


PowerPC: A High Level Controller

- PPC core (with some hardware additions) can execute traditional programs
- PPC core may also perform control and monitoring functions for the processor system



HLC - LLC Interaction

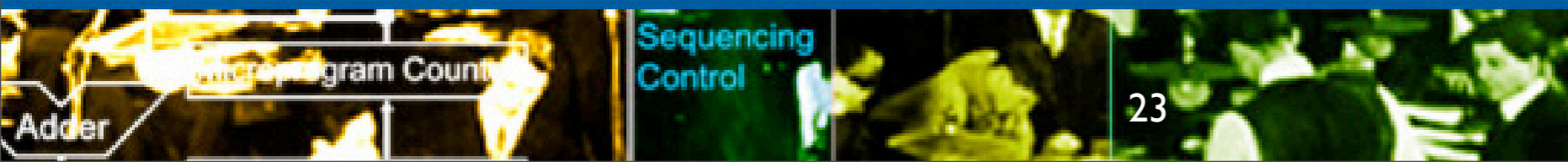


Sequencing
Control

Xilinx Virtex II-Pro FPGA

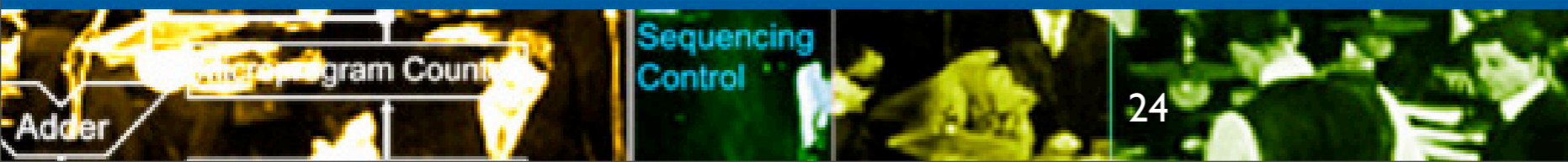
Limitations

- No hardware floating point support
 - FP IP must be on the PLB
 - Issues: Bus contention, Saturation
- Partial reconfiguration must use modular design flow
 - Interfaces between modules must be well-designed
- FPGA hardware generally obsoleted quickly



FPGA Improvements

- More hard cores for HLC instances
- Larger distributed BRAM
- More CLBs
- More embedded multipliers (and some FP units)
- Faster clocks
- More on-chip channels
- More external pins desired



I/O 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

I/O is getting faster, but are usually separated from the processors by several levels of the memory hierarchy.

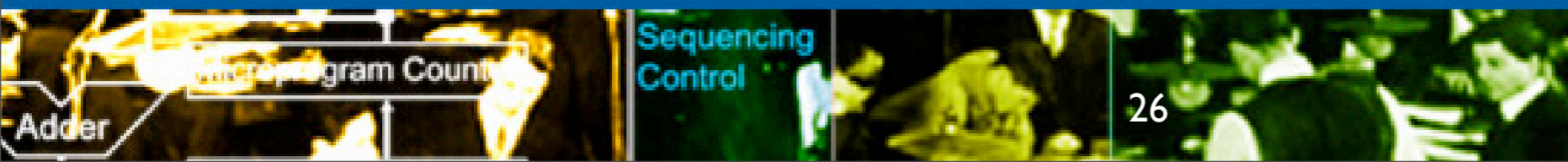


Sequencing Control



Conclusions

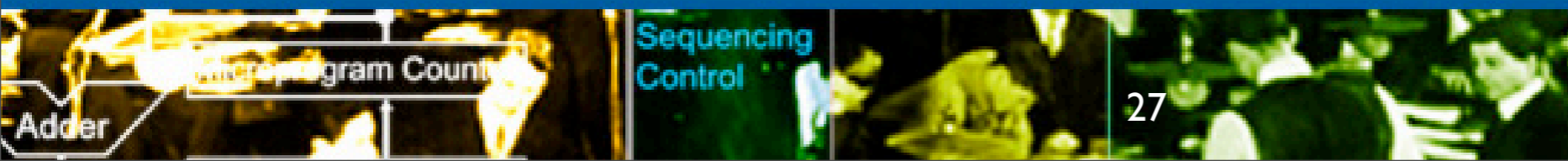
- A new execution model is proposed that also addresses the memory wall problem.
- Applications run on RISC cores and reconfigurable hardware that associate, mix, and compute streams.
- Embedded software saves resources better devoted to time-sensitive computation.
- Fast peripherals need not be limited by intermediate controllers, buses, and operating systems.
- We are gathering performance data for pure hardware, pure software, and combinations of hardware and software on the FPGA chip.



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.



Polytechnic
UNIVERSITY

Thank you.

contact: dfinke01@cis.poly.edu



Sequencing
Control

