

Configurable Computing: Practical Use of FPGAs

Craig Ulmer

March 9, 1999

Computer Architecture and Systems Lab
Georgia Institute of Technology

Motivation: Practical FPGAs

- Programmable Logic Available for Long Time
 - Interesting Because of Reconfigurable Hardware
 - Dominant Use: Glue Logic
- Recent Technological Advances
 - Gate Density / Speed / Cost
- FPGAs for Computational Assistance?
 - Practical Use? Configurable Computing?

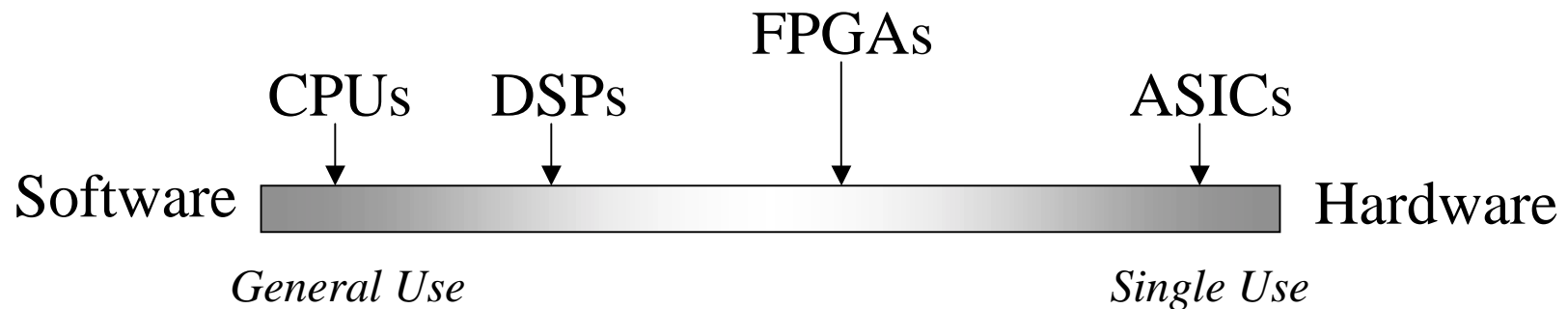


Outline

- Algorithms & Configurable Computing
 - Application of FPGAs
- Design Methodologies
- Performance Metrics
- Design Examples
- Obstacles & Future Enhancements

Algorithms: Software vs. Hardware

- Two Styles of Computing: Software and Hardware Based
- Software: General Purpose CPUs
- Hardware: Custom Circuits

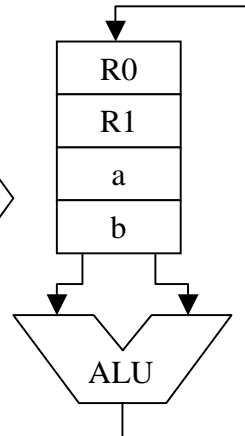


Software vs. Hardware Implementations

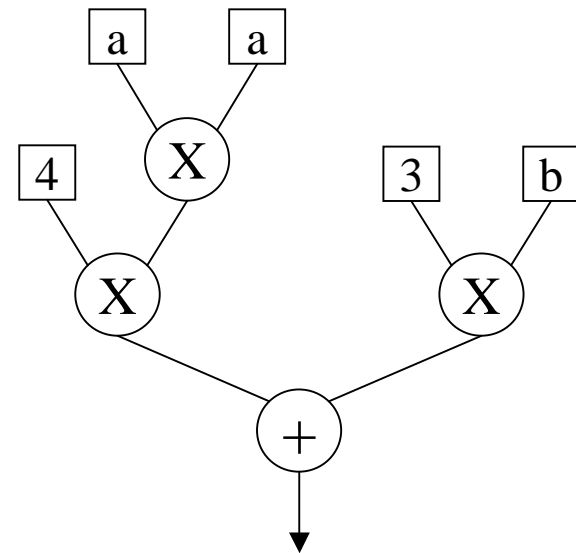
Example Logic Function = $4a^2 + 3b$

CPU Instructions

```
R0=a
R1=b
R0=R0*R0
R0=4*R0
R1=3*R1
R0=R0+R1
```



Software Implementation



Hardware Implementation

Configurable Computing

- Use FPGAs as ‘Soft’ ASICs
 - FPGA Configured as needed by Application
- Best of Software and Hardware
 - Reprogrammable
 - Hardware Acceleration

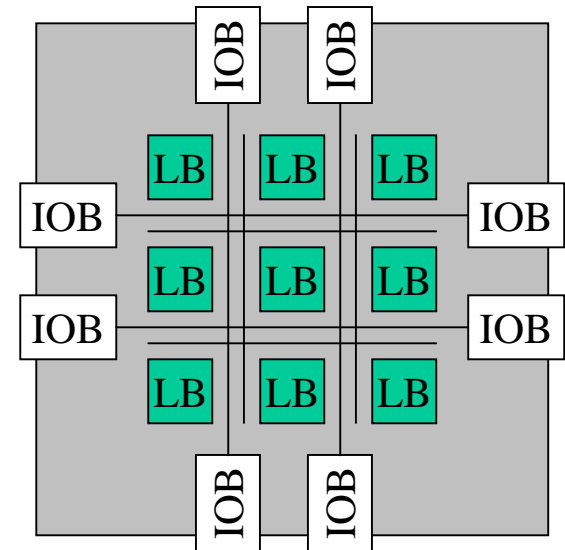
History of Programmable Logic

1960's	Estrin's 'Variable Logic' in CPUs
1970's	PLA / PLD
1985	Field Programmable Gate Arrays
1990's (Early)	Multi-FPGA Custom Computing Machines (CCMs)
1990's (Late)	Dense FPGAs, Few-Chip Cards

General FPGA Architectures

Three Components:

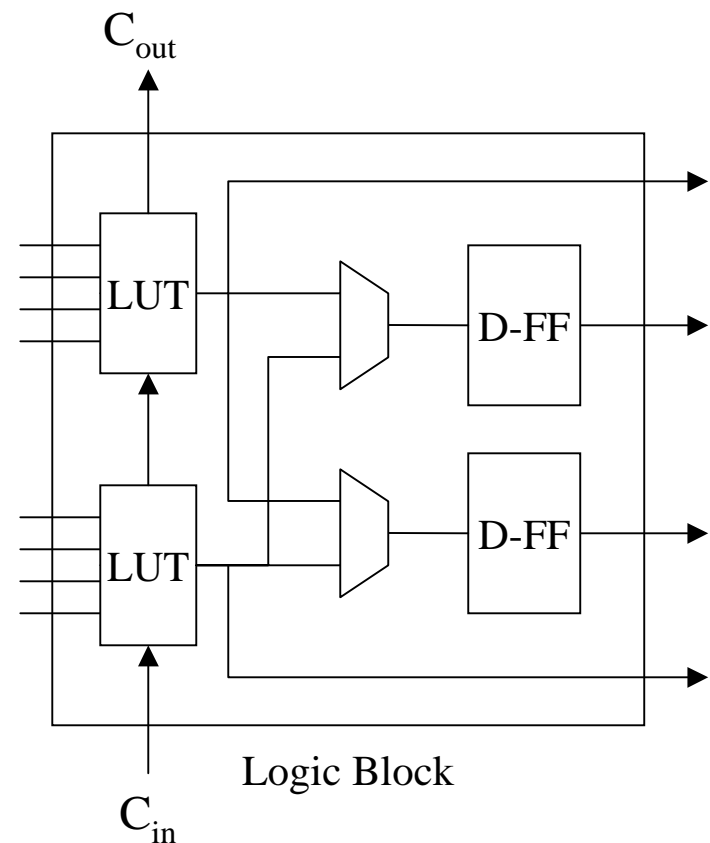
- Logic Blocks
- Interconnection Network
- I/O Blocks



Result: 5k-500k User Gates

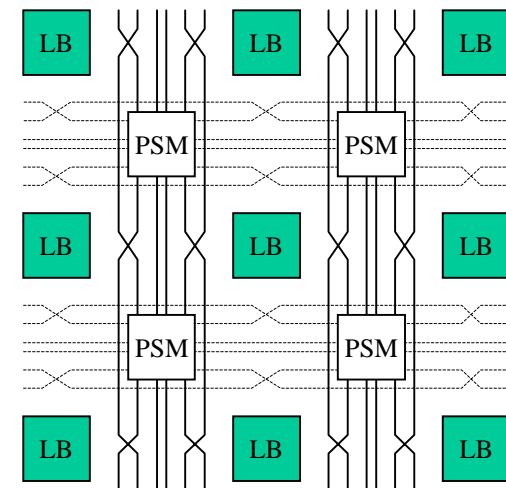
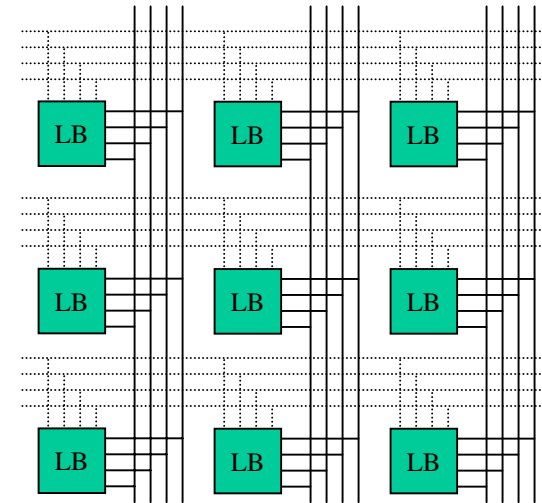
General Logic Block Architecture

- Three Components:
 - Lookup Table Function Generators
 - Internal Routing
 - Memory: D-Flip-Flops
- Operation Modes:
 - Logic
 - Arithmetic / Ripple
 - RAM / ROM



Interconnection Networks

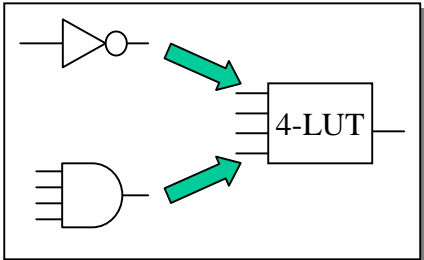
- Bus Based
 - Simpler Routing
 - Higher Parasitics
- Switch Based
 - Segmented Wires
 - Lowers Line Taps
 - Complex Routing



Consequences of Architecture

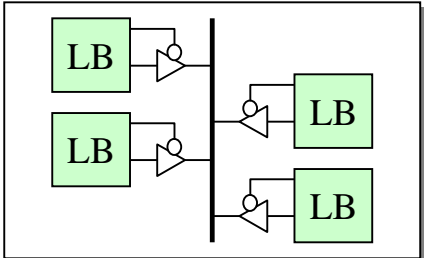
LUT Logic Forces Course Granularity

Delay for 1-bit NOT same as 4-bit AND



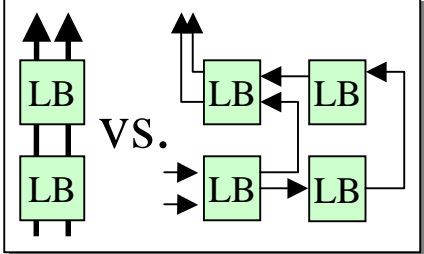
Interconnection Network Induces Delays

Routing not 'For Free'



Architecture Features like Fast Ripple

Nonlinear Performance



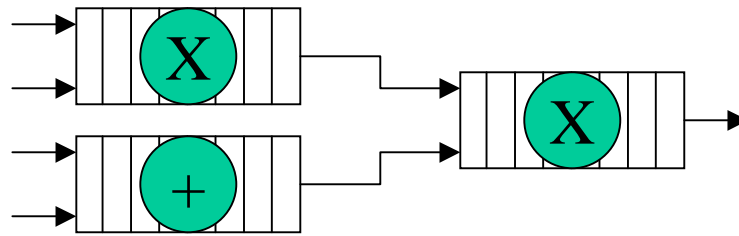
Result: VLSI Design Methods may not be Effective

FPGA Design Strengths: Pipelining

- Pipeline to Improve Throughput
 - Implies Stream Operations



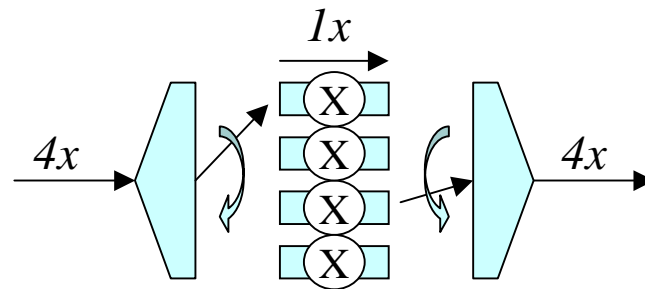
- Chaining for Complex Operations
 - Implies *Configurable Computing*



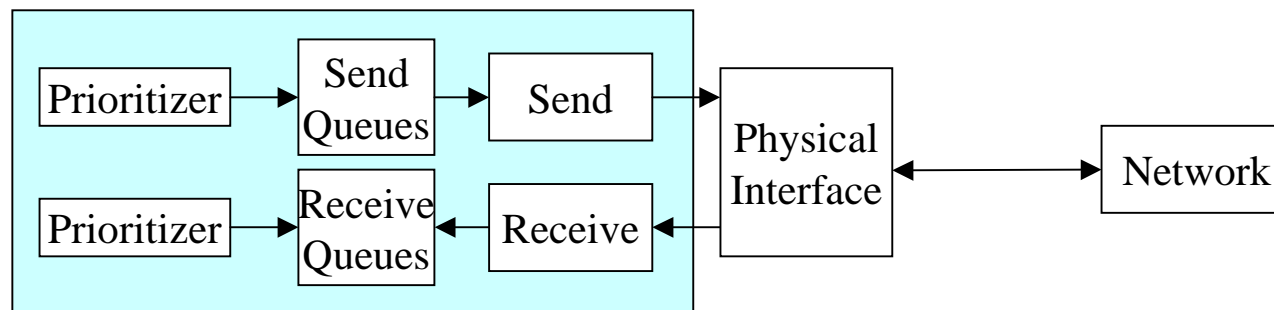
FPGA Design Strengths: Parallelism

FPGA Allows High Hardware Parallelism

- Data: Parallel Computation



- Control: Many State Machines

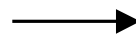


FPGA Design Strengths: Partial Evaluation

- Simplification of Hardware by Compile Time Knowledge
- Example: 4-bit Multiplication:

$$\begin{array}{r}
 a_3 a_2 a_1 a_0 \\
 \times b_3 b_2 b_1 b_0 \\
 \hline
 a_3 b_0 \\
 a_3 b_1 a_2 b_1 a_1 b_1 \\
 a_3 b_2 a_2 b_2 a_1 b_2 \\
 + a_3 b_3 a_2 b_3 a_0 b_3 \\
 \hline
 \end{array}$$

General Purpose Multiplication

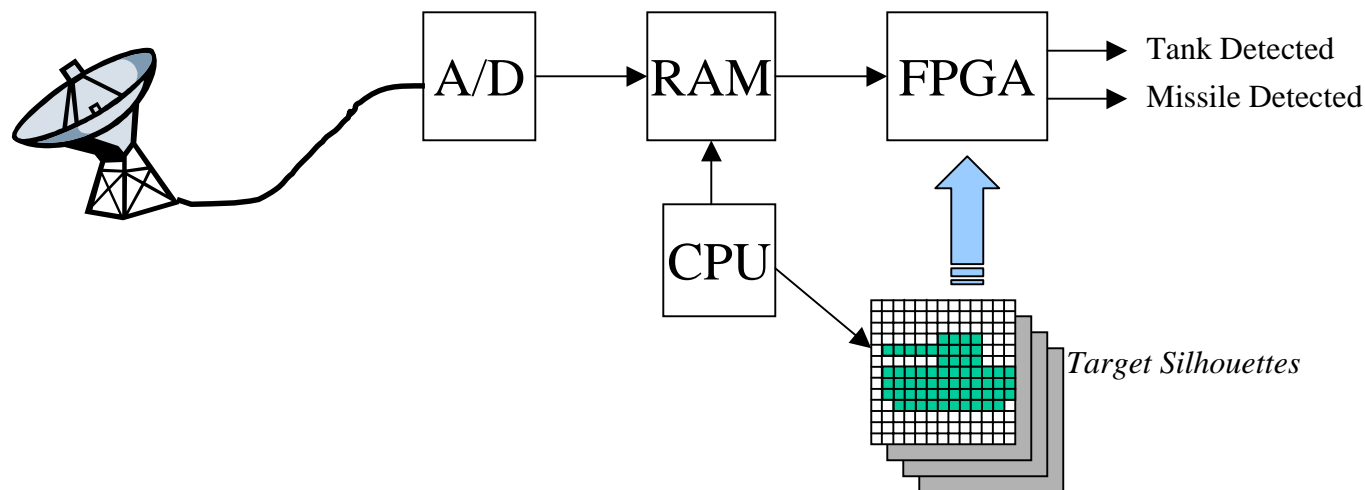


$$\begin{array}{r}
 a_3 a_2 a_1 a_0 \\
 \times 1010 \\
 \hline
 a_3 \\
 a_2 \\
 + a_1 \\
 \hline
 \end{array}$$

Partially Evaluated Multiplication

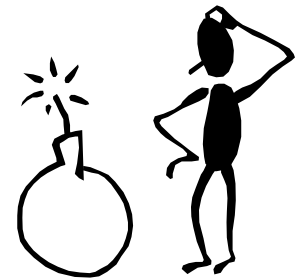
FPGA Design Strengths: Configurable Hardware

- Ability to Provide Hardware Support
- Reusable Hardware / Disposable Circuits
- Hardware Adapts to Specific Problem/Data Set
 - Example: Target Recognition



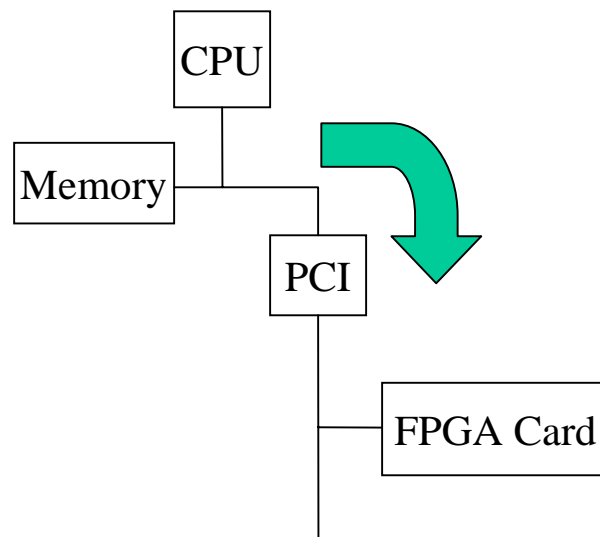
FPGA Weaknesses: Physical Limitations

- 3x Speed and 10x Density Degradation from ASICs
- Gate Density Limits Design Sizes
- Logic Blocks Can Never be Fully Utilized
- Poor Handling of Floating Point
- Pin Limitations



FPGA Weaknesses: System Integration

- Current Generation: Slow Reprogram
 - Reconfiguration time vs. Compute Time
- FPGA Location: Poor Data Proximity
 - Computation Moves Up & Down I/O Subsystem



Application Suitability

Suitable

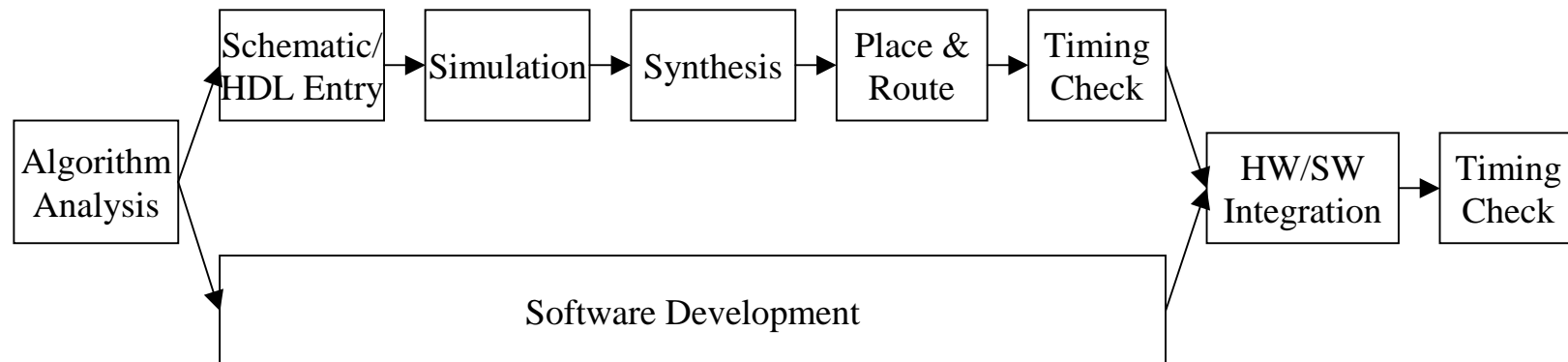
- Highly Parallel
- Streamlined Data
- Prior Knowledge
- Complex Custom Logic

Unsuitable

- Sequential Algorithms
- Floating Point
- Non-Localizable Data

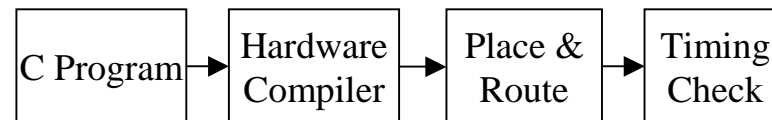
Design Methodologies: Manual Design

- Manual Design
 - Engineer Provides Analysis and Design
 - Best Results, Most Time Consuming



Design Methodologies: Compilation

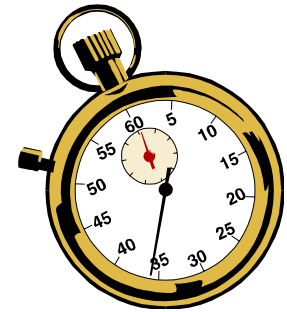
- Manual Design too Complicated
- Instead: Compile C Code to Hardware
 - Handel-C and RAW Projects
 - Automates Analysis Process



- Easy Design
- Low Performance (2x over Software)

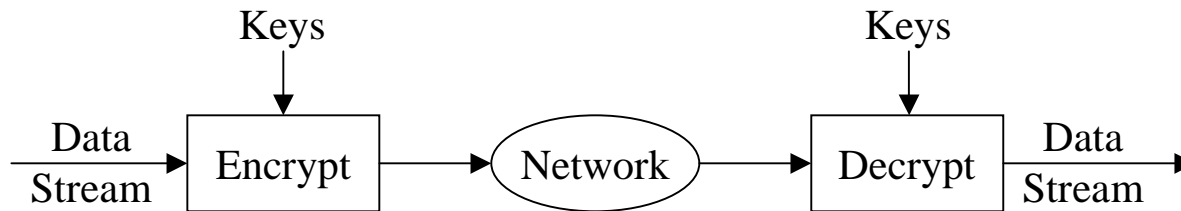
Performance Metrics

- Cross FPGA Comparison
 - RAW Benchmark Suite
 - Difficult for Fair FPGA Comparisons
- Design System Performance
 - Ultimate Speedup over Software-Only
 - Hardware Resource Costs
 - Side Effects on System Performance



Example: Cryptography

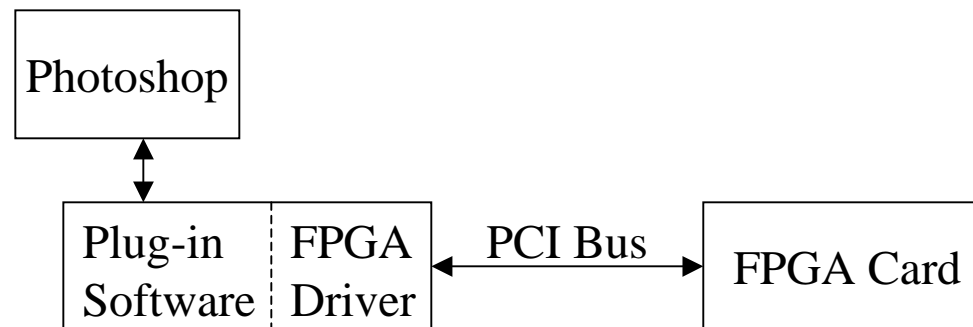
- Encryption Strength: Non-trivial Computation
- Result: Hardware Assistance for Data Streams



- FPGA Strengths:
 - Plug-in Encryption Algorithms
 - Partial Evaluation based on Keys : +35% Bandwidth, -45% Hardware
 - Key Breaking Applications: 1Mkeys/s (vs. 50Kkeys/s in software)

Example: DSP

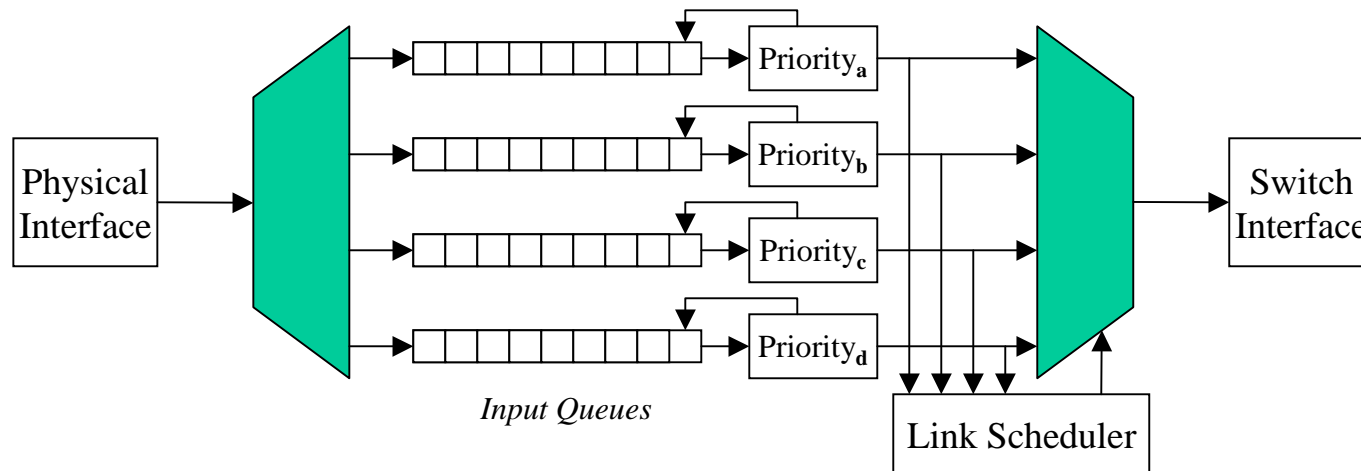
- Adobe Photoshop Filter Plug-ins
- FPGAs Provide Hardware Assistance in Image Processing



- Results:
 - On-FPGA Processing: 20Mpixels/s (10x Better than Quad-PowerPC)
 - System Wide Processing: 0.22Mpixels/s
 - Performance Lost in System Integration

Example: Networks

- High Data Rates: Gigabit ATM/Ethernet
- *More* Functionality: Quality of Service (QoS), Active Networks
- Illinois Pulsar-based Optical INTerconnect (iPOINT)
 - Gigabit ATM Switch
 - Complex QoS Queuing with FPGAs
 - Manage Packet Priorities at 622Mbps (OC-12)



Obstacles and Future Enhancements

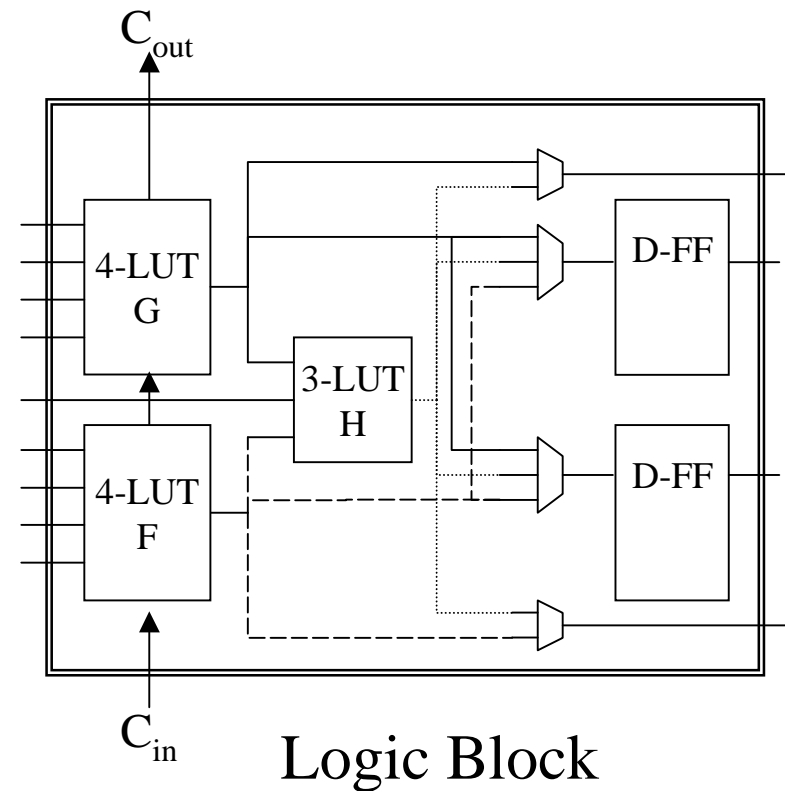
- System Integration
 - Embed FPGA in CPU (BRASS, Estrin)
 - Embed Communication Core in FPGA (RAMBUS)
 - Use Better Interfaces: AGP in PCs
- Design Environment
 - MATCH Matlab to VHDL Compiler
- Reconfiguration Time
 - Partial Reconfiguration, Context Switching FPGAs

Conclusions

- Configurable Computing: Infancy
- System Level Issues to Address
- Overall Value for a System?
- Next Generation of FPGAs

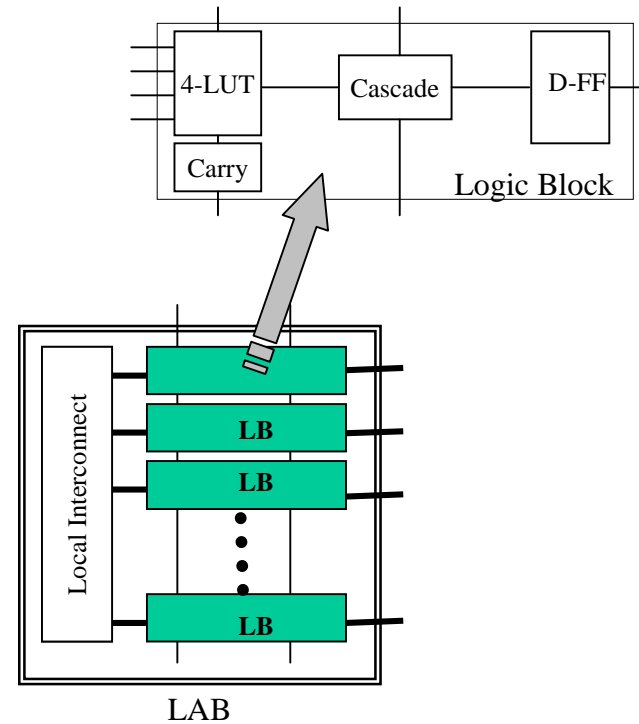
Commercial Example: Xilinx XC4000

- Complex LB
- Switch Interconnect
- 2k-500k User Gates
- .35-.25 μ m Process



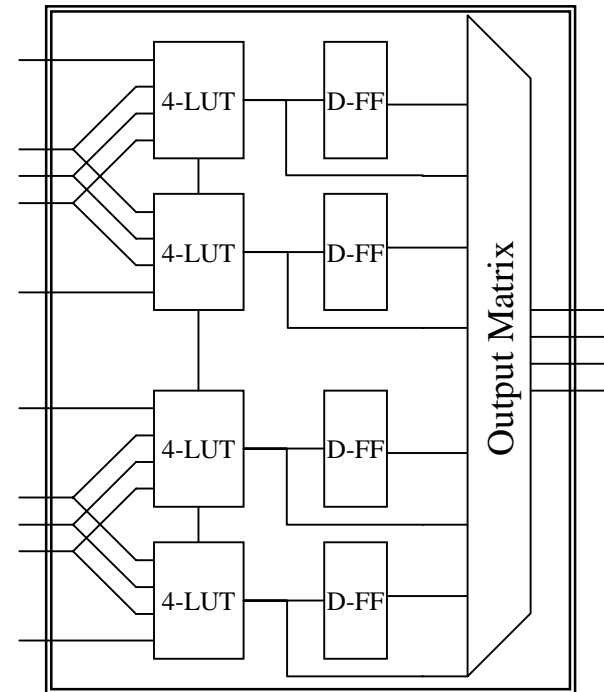
Commercial Example: Altera FLEX

- Simple LB
- Group LBs into LABs
- Bus Interconnect LABs
- 5k-250k User Gates
- .35-.25 μ m Process



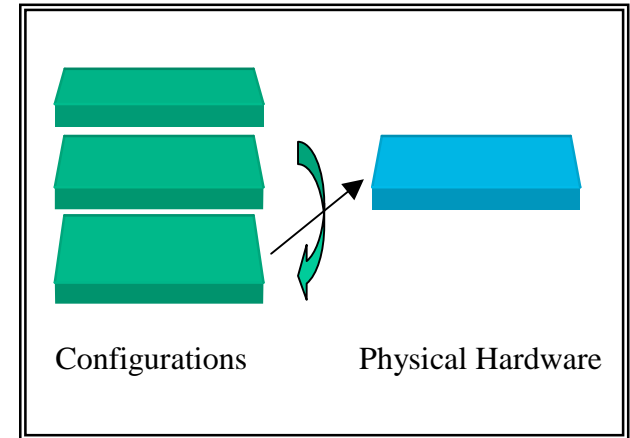
Commercial Example: Lucent ORCA

- Flexible LB:
 - Quad 4-LUT
 - Twin 5-LUT
 - Single 6-LUT
- Bus Interconnect
- 5k-100k User Gates
- .35-.3 μ m Process



Context Switching FPGAs

- Multiple Configurations
 - Share Hardware
 - Configurations at Each LB
 - “Dynamic”-PGAs
- Complex Design
- Future: Fine Grain Contexts



CS-DPGA

Embedding FPGAs in Processors

- Goal: Place FPGA in Processor
 - Tight Coupling
 - Estrin's Fixed & Variable Logic
- Berkeley BRASS Project
 - Garp: MIPS + FPGA
 - 24x Improvement in Basic Applications
- Requires System Thought
 - How to keep FPGA & Processor Active?
 - Can FPGA violate OS Protection?

