The entire system operates in a configuration described as the “Fixed-Plus-Variable” Structure Computer such that the same elements used for the special computer may be reorganized for other problem applications.

– Gerald Estrin (UCLA) 1962
Course Objectives

› Prerequisites
  - Computer programming in C
  - Basic digital systems (combinatorial circuits, sequential circuits, finite state machines, data paths, microprocessor architecture)
  - Experience using a hardware description language (Verilog or VHDL)

› Objectives
  - An introduction to the field of reconfigurable computing
  - Advance digital design skills by developing a reconfigurable computing application
  - An introduction to research methodology
Course Website

Topics

› Introduction to Reconfigurable Computing
  - What is reconfigurable computing, applications

› Computer Architecture
  - Arithmetic, pipelining, data paths, microcode

› Programming Abstractions
  - Microcode, Case studies

› Examples from research

Lecture Schedule
1. Introduction
2. FPGA architecture
3. Control Logic
4. Arithmetic
5. Single Cycle RISC Processor
6. Multi Cycle RISC Processor
7. Pipelined RISC Processor
8. Elliptic Curve Processor
A major part of this course are the tutorials

- Tut1 – VTR exercise on computer architecture (lecture 2)
- Tut2 – Finite state machine to generate Walsh sequence (lecture 3)
- Tut33 – Discrete Walsh Transform (DWT) processor (lecture 7)

Report

- Write a paper on the DWT processor suitable for submission to a conference
Introduction to Reconfigurable Computing

- FPGAs
- Reconfigurable computing
- Applications
Most electronics rely on application-specific ICs (ASICs) for perf, cost and P

Source: Arvind MIT
A generalised ASIC
- Logic blocks for digital operations
- Programmable interconnect for routing

Arbitrary digital circuits can be implemented

Functionality downloaded to FPGA memory (in seconds)

Source: Steve Wilton (UBC)
FPGA Embedded Blocks

Source: Xilinx

Hard IP blocks for widely-used functions: faster, more efficient, lower power
Careful choice: every user must pay for these functions, whether used or not

Source: Xilinx
Zynq (ARM+ Reconfigurable Fabric)

Source: Xilinx
**FPGA Families**

Xilinx 7-series FPGAs, 28nm

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP Slices</td>
<td>740</td>
<td>1,920</td>
<td>3,600</td>
</tr>
<tr>
<td>DSP Performance (symmetric FIR)</td>
<td>930GMACs</td>
<td>2,845GMACs</td>
<td>5,335GMACs</td>
</tr>
<tr>
<td>Transceiver Count</td>
<td>16</td>
<td>32</td>
<td>96</td>
</tr>
<tr>
<td>Transceiver Speed</td>
<td>6.6Gb/s</td>
<td>12.5Gb/s</td>
<td>28.05Gb/s</td>
</tr>
<tr>
<td>Total Transceiver Bandwidth (full duplex)</td>
<td>211Gb/s</td>
<td>800Gb/s</td>
<td>2,784Gb/s</td>
</tr>
<tr>
<td>Memory Interface (DDR3)</td>
<td>1,066Mb/s</td>
<td>1,866Mb/s</td>
<td>1,866Mb/s</td>
</tr>
<tr>
<td>PCI Express® Interface</td>
<td>x4 Gen2</td>
<td>x8 Gen2</td>
<td>x8 Gen3</td>
</tr>
<tr>
<td>Analog Mixed Signal (AMS)/XADC</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Configuration AES</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>I/O Pins</td>
<td>500</td>
<td>500</td>
<td>1,200</td>
</tr>
</tbody>
</table>

Source: Xilinx
ASIC vs FPGA Cost

Crossover volume increases with decreasing feature size.
ASIC Development Costs Today

Drives Fewer ASIC Design Starts

ASIC Design Starts Lag Moore’s Law;

Development Cost ($M)

Process Node

Source: Altera
### Return on Investment Analysis

**New Product Business Plan Version 1.0**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ASIC / ASSP (65nm)</strong></td>
<td>$55M</td>
</tr>
<tr>
<td><strong>Development Cost</strong></td>
<td></td>
</tr>
<tr>
<td><strong>20% of Revenue on R&amp;D</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Revenue Target</strong></td>
<td>$275M</td>
</tr>
<tr>
<td><strong>$10 to $50 Device ASP</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Unit Volume Req’d</strong></td>
<td>5Mu to 27Mu</td>
</tr>
</tbody>
</table>

**Very Few High Volume Applications Justify ASIC / ASSP Development**

*Source: Altera*
Application Domains

Relative Growth Potential (3 to 5 Year Horizon)

End Market Weight (% of Revenue in 2009)

Larger

Source: Altera
Typical High Performance Commercial Applications

**Application**
- Optical Transport OTU Transponder
- 40GbE/100GbE Switch
- Radar

**Requirements**
- >350 MHz performance
- 28 Gbps transceivers
- 10GBASE-KR backplane support
- High-performance on-chip memory
- High-performance and flexible memory controller
- Hard system-level IP for bandwidth
- High precision DSP

**Solution**

**Process: 28HP**
- >350 MHz performance
- Lowest power in its class
- Up to 1.1M LEs on a monolithic die

**Transceiver: 14.1 Gbps/28 Gbps**

**Product Architecture:**
- Soft memory controller supports 800MHz DDR3 DIMM
- 2,560 M20K memory blocks
- 54x54 variable precision DSP

**System IP:**
- PCIe Gen3 x8, 40 GbE/100 GbE, Interlaken

**Source:** Altera
Dynamic Reconfiguration

Semiconductor devices that can optimally transform themselves in software
(Reconfigurable technology)

- Conceptual examples -

Achieves Low power as well as Multifunctionality and flexibility

Sony Virtual Mobile Engine 1
Sony Virtual Mobile Engine 2

Although a general-purpose CPU can do anything, its power consumption is poor.

A general-purpose CPU handles light control tasks.

The VME handles heavy work that would consume power efficiently.

Transforms into optimal special-purpose circuits while running

Therefore, it is flexible and low power!
Flexibility vs Performance

The Age Requires Programmability
Product diversification and multifunctionality
Multiformat support

Stamina
Power efficiency
Performance

Low
High

Programmability (flexibility)

Good
Bad

Optimal special-purpose HW

Flexible hardware
The hardware configuration can be modified by software

VME

CPU+VME

Sony provides stamina
Architectural Choices

General-purpose processor

Application-specific processor

Dedicated accelerators

Reconfigurable processor

Source: Rabaey UCB
Flexibility vs Energy

- **Embedded Processors**
  - SA110: 0.4 MIPS/mW

- **ASIPs**
- **DSPs**
  - DSP: 3 MOPS/mW

- **Reconfigurable Processor/Logic**
- **Pleiades**
  - 10-80 MOPS/mW

- **Dedicated HW**

---

Benchmark numbers @ 1999

- SA110: 0.4 MIPS/mW
- Pleiades: 10-80 MOPS/mW

Approximately three orders of magnitude in inefficiency from general-purpose to dedicated!

Source: Rabaey UCB
FPGA vs DSP and CPU Cost Comparison

Berkeley BEE2 cost comparison (FPGA, DSP1, DSP2, uP)
› Traditionally designed using ASIC development tools
  - VHDL/Verilog very low level
› Recent advances
  - Vivado HLS
  - OpenCL
› Extensive module generators and libraries e.g. filters, fft, floating-point, maths coprocessors, soft processors, network controllers, memory controllers, I/O controllers …
› Still an active research topic
Comparison of FPGAs with uP and ASIC

- Compared with uP and DSP
  - higher speed, lower power, smaller variance in execution time
  - Longer development times, higher cost per unit

- Compared with ASICs
  - Lower initial cost

- Rides Moore’s Law, development costs amortised over users
  - Faster time to market, lower risk
  - Can be customised to problem in ways not possible with ASICs
Overview

› FPGAs
› Reconfigurable computing
› Applications
Application of FPGA devices to computing problems
FPGAs allow computational problems to be accelerated through
- Parallelism
- Customisation
- Integration
Parallelism

- Do what would take many cycles on uP in fewer cycles (instruction level parallelism)
- Do many independent tasks/threads/processes in parallel (multiprocessor)
- Tradeoff latency with throughput by doing things in stages (pipelining)

Parallelism Example

- Microprocessor: data passed sequentially to computing unit
- FPGA & ASIC: spatial composition of parallel computing units (multiple muls, pipelining)
- E.g. 4-tap FIR filter, FPGA 1 output per cycle, uP takes multiple cycles
- Lower power and higher speed

Source: DeHon “The Density Advantage of Configurable Computing”
- More specific functions can be implemented more efficiently
- Too expensive to design ASIC to perform very specialised function
- FPGAs can be heavily customised due to their programmability i.e. only do one thing efficiently
  - Tradeoffs between speed and accuracy can be exploited, on uP, only get single or double; char, short or long
  - General operators can be replaced with specific ones
- E.g. Chip which only encrypts for a specific password
Integration

- Networking, chip IO and computation on same device
  - Reduction of buffering can help latency
  - Single chip operation massive interconnect within chip exploited
  - Multiple (small) memories within FPGA offer enormous memory bandwidth
Overview

› FPGAs
› Reconfigurable computing
› Applications
Vehicle Control Module uses Virtex-II devices
- gearbox, differential, traction control, launch control and telemetry
High speed real-time control and DSP application

Source: BMW Williams
Compact Muon Solenoid

- $10^{15}$ collisions per second
- Few interesting events ~ 100 Higgs events per year
- 1.5Tb/s real-time DSP problem
- More than 500 Virtex and Spartan FPGAs used in real-time trigger

Source: Geoff Hall, Imperial College
Square Kilometre Array (SKA) will be one of the largest and most ambitious international science projects ever devised (€1.5 billion).

CSIRO Developing Australian SKA Pathfinder (ASKAP), a $150M next-generation radio telescope using FPGA technology for the data collection & processing.

Source: CSIRO
Other RC Applications

› Applications suited to acceleration
  - seismic processing astrophysics FFT
  - adaptive optics (transforming to frequency domain and removing telescope image noise)
  - biotech applications such as BLAST, Smith Waterman and HMM
  - computational finance

› Functions well suited to FPGA acceleration
  - searching & sorting
  - signal processing (audio/video/image manipulation)
  - encryption
  - error correction
  - coding/decoding
  - packet processing
  - random-number generation for Monte Carlo simulations

Source: cray.com
uPs are the most flexible technology but performance (speed and power) is relatively low.

FPGAs provide:
- Easy interfacing with hardware (tighter coupling than GPUs)
- Parallelism
- Have become large enough to implement DSP and ML algorithms
- Very interesting research area: architectures, tools, applications

ASICs becoming only be suitable for highest volume, highest performance applications, FPGAs will do the rest.

Many of the highest performance accelerators, particularly for real-time problems, are FPGA-based.