

Reconfigurable Computing - (RC)

Yogindra S Abhyankar

Hardware Technology Development Group,

C-DAC yogindra@cdac.in



Outline

- **Motivation**
- **Architecture**
- **Applications**
- **Performance**
- **Summary**

HPC Fastest Growing Sector

HPC, the massive horsepower of IT is one of the fastest growing sector in the Industry

- Rapidly growing data generated in the enterprise...
- Every industrial sector

Every one Wants fast Results

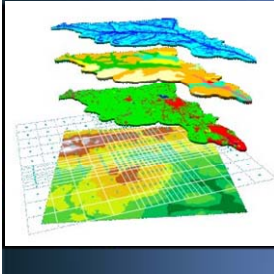
HPC Segment Application Areas

Financial



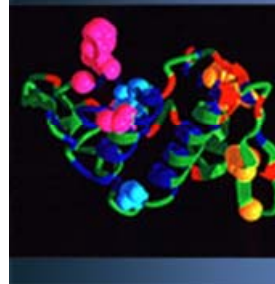
- Fin. Modeling
- Data Mining

Oil & Gas



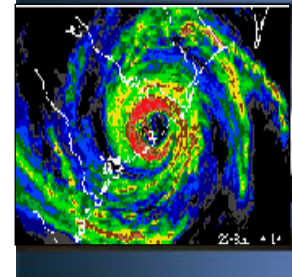
- Seismic
- Reservoir Mod

Materials & Life Sciences



- Molecular Dyn
- Med. Imaging
- DNA / iRNA
- Material Sci

Government



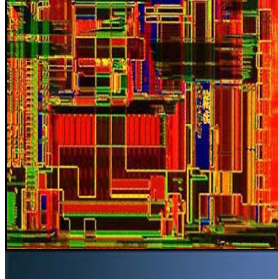
- Weather
- Crypto
- Data Mining

Film / Video



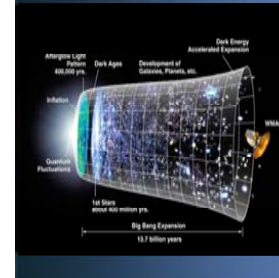
- Rendering
- Compositing

EDA



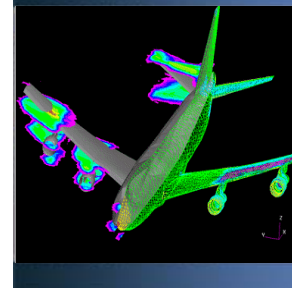
- Verification
- Layout

Scientific Research



- Cosmology
- Physics
- Math

CAE/CAD/CAM



- Structures
- Fluids
- Impact

Emerging Application Areas

..sometime not evident

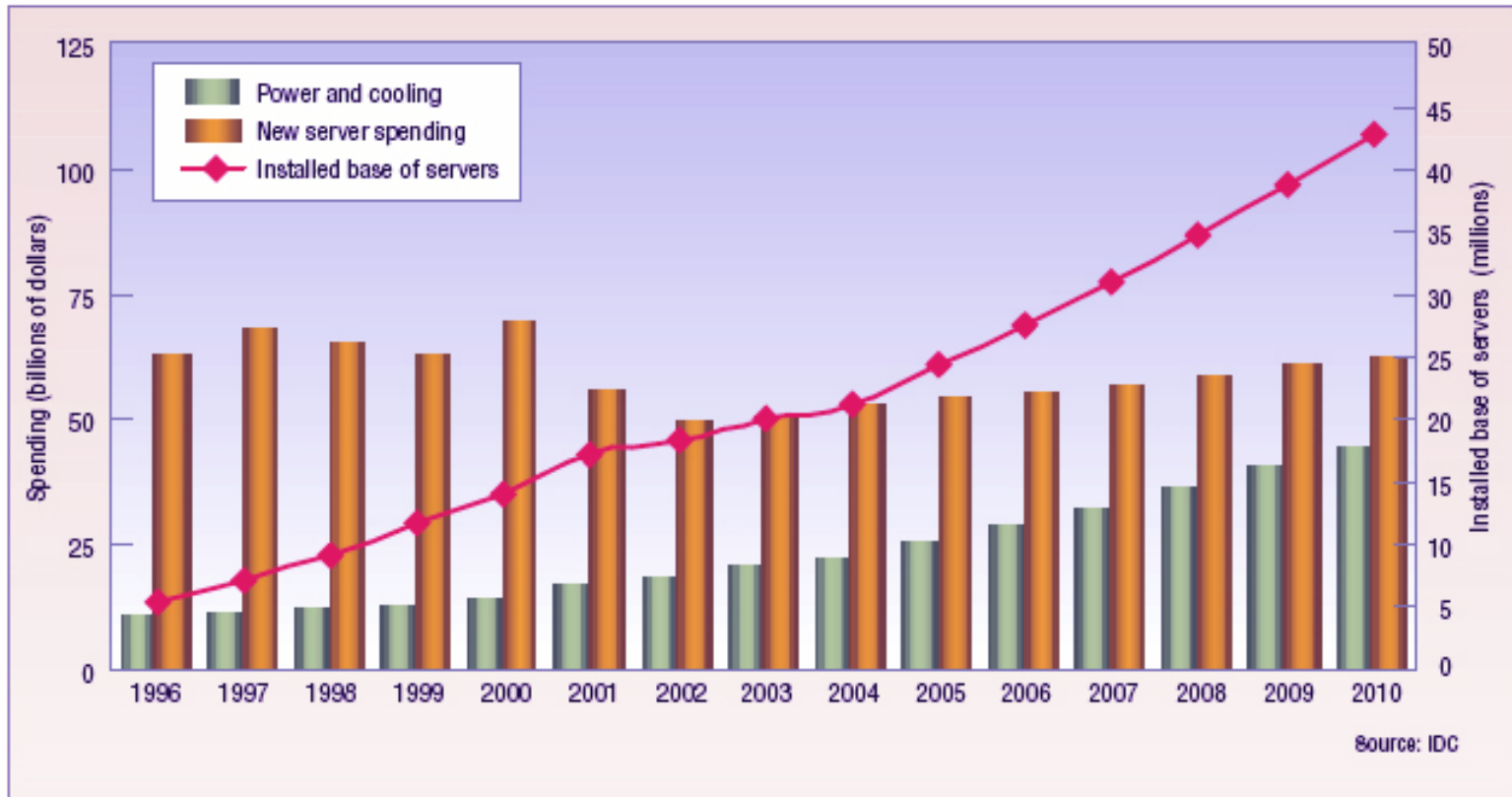
- ◆ *Froth formation in the washing machine*
- ◆ *Model production/Packaging of Pringles (potato chips)*
- ◆ *Rotting of wood*

Ref: SC'07

Faster Solutions

- ◆ Large clusters
- ◆ Application tuning
- ◆ Latest processors – Frequencies/Cores/Architectures..
- ◆ Change Algorithm to suite the FIXED hardware

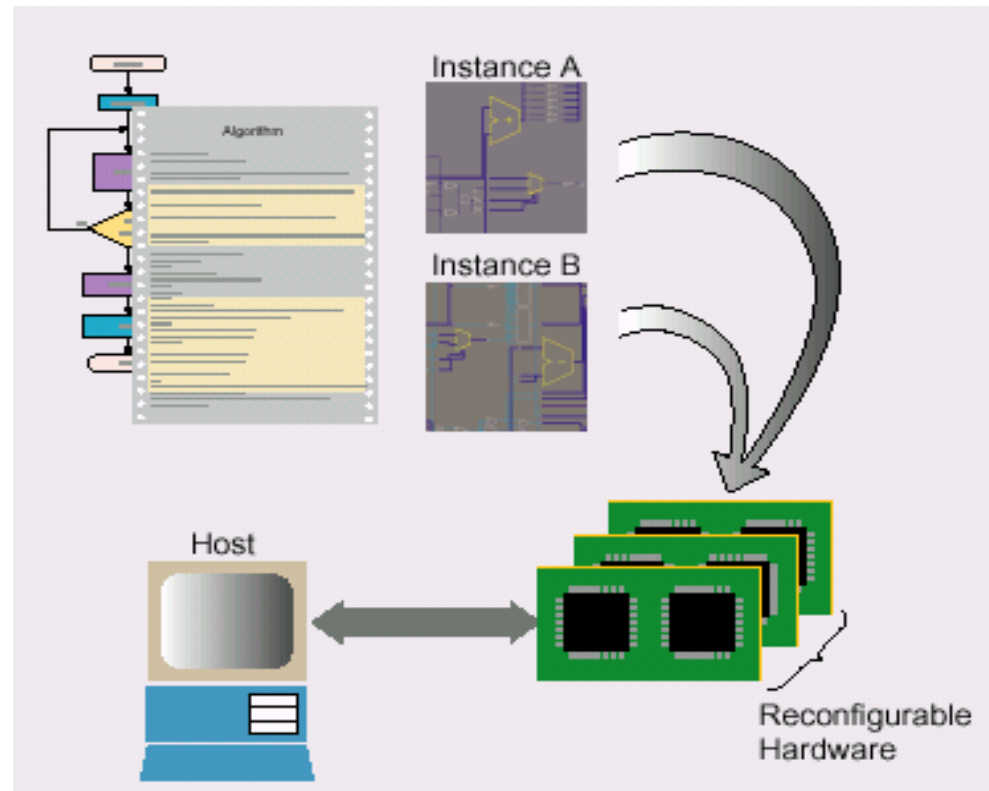
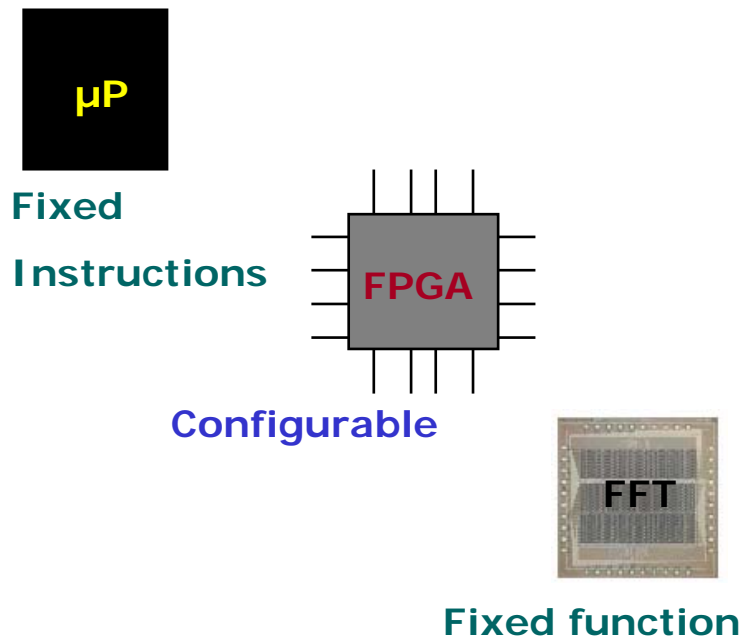
Power Requirement Trends



Power and cooling costs increasing faster than equipment cost
Power density in urban centers breaks power grids

Reconfigurable Computing System

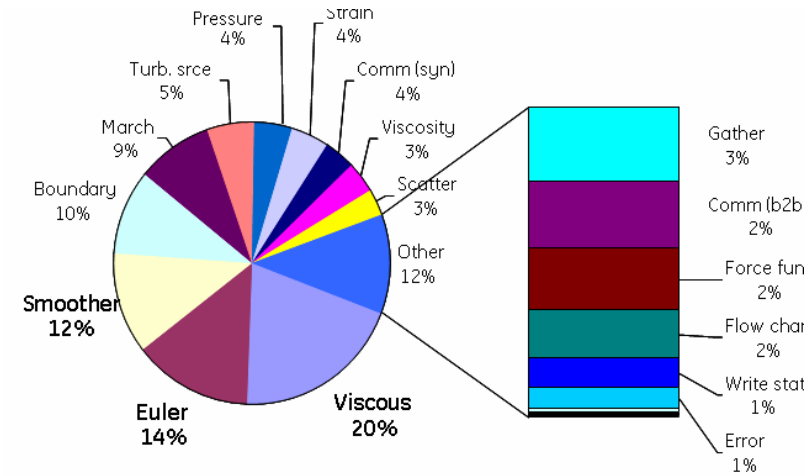
- ◆ **Systems that dynamically change their hardware for accelerating applications - Mainly based on **FPGAs****



Why one gets Application Acceleration

- **Selecting suitable application**

- Application/ Kernel profiling
 - **Compute intensive functions**
 - **Inner loop run many times**
- Analyze required precision
- SW/HW Partitioning



- **Efficient HW logic**

- Pipelined & Parallel
- High degree of Instruction efficiency
- Analyze required HW resources

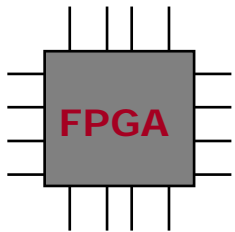
Outline

- Motivation
- Architecture
- Applications
- Performance
- Summary

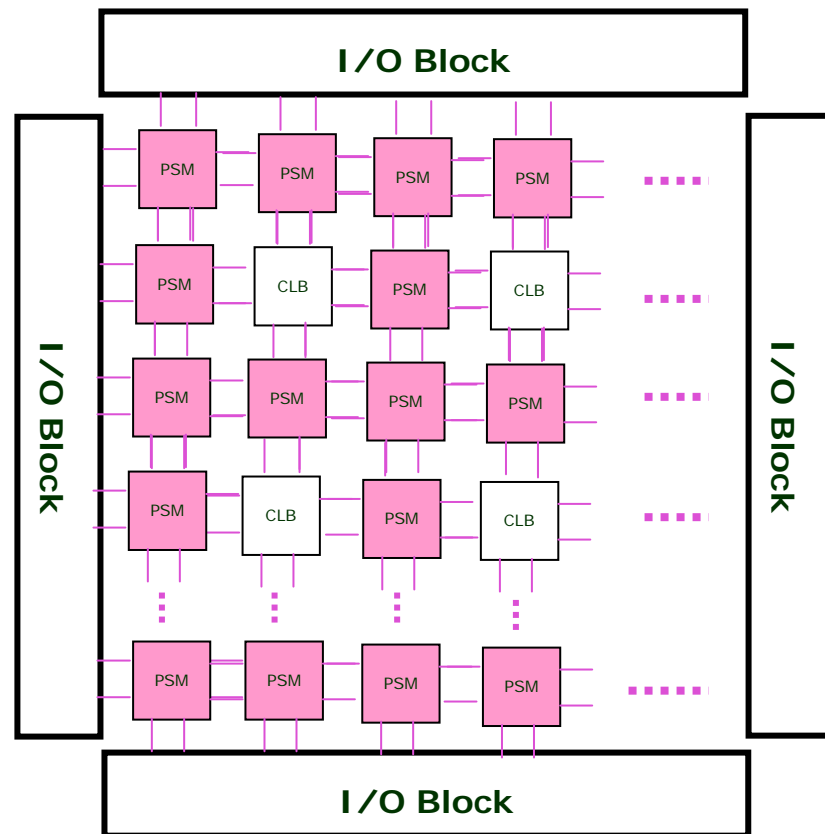
Field Programmable Gate Array (FPGA)

- ◆ *A chip that can be configured by a user to implement different digital logic circuits*
- ◆ *Configurable Logic Blocks and interconnects*

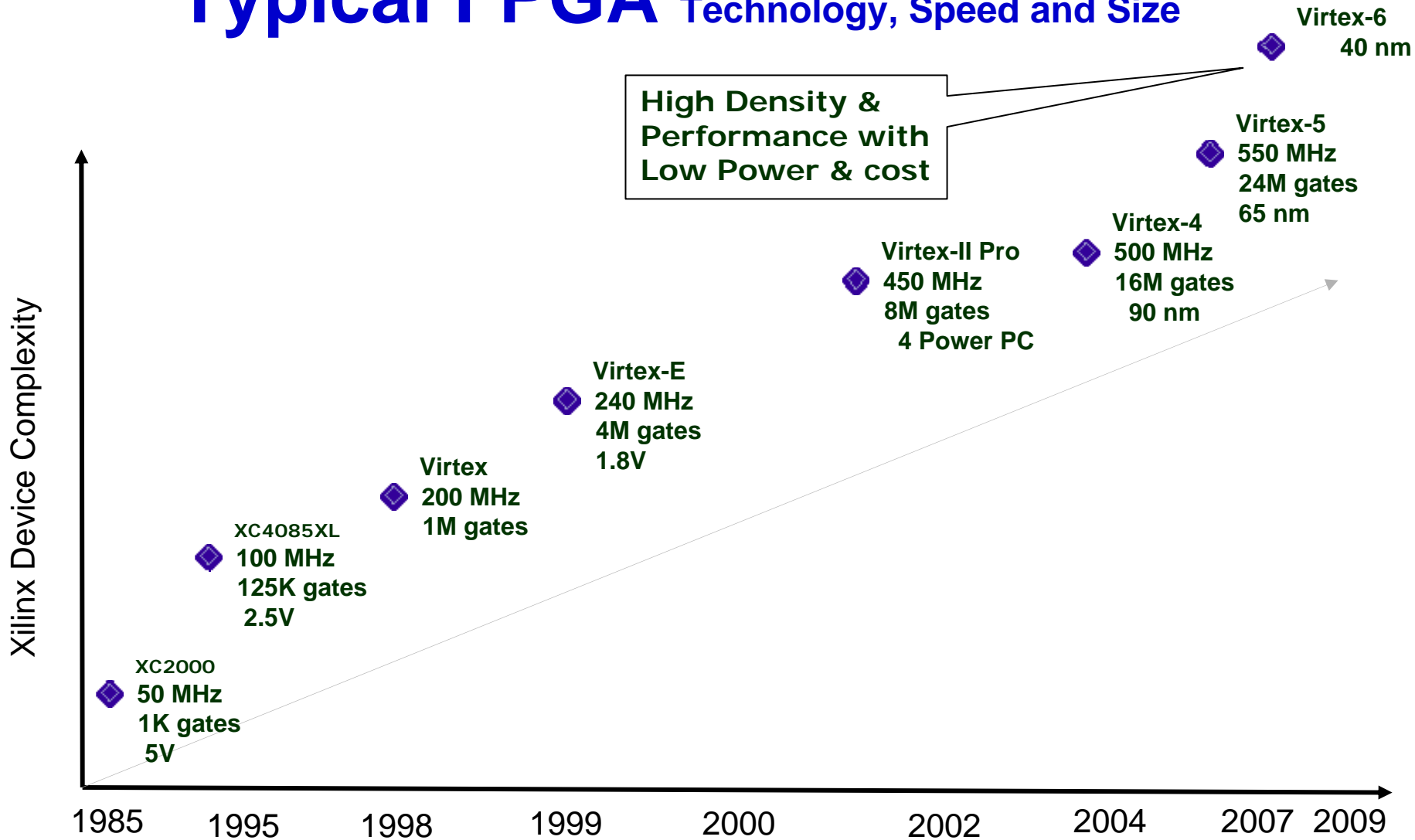
Invented
in 1984



Configurable



Typical FPGA Technology, Speed and Size



Major FPGA Vendors

Xilinx

Altera

Atmel

Lattice Semiconductor

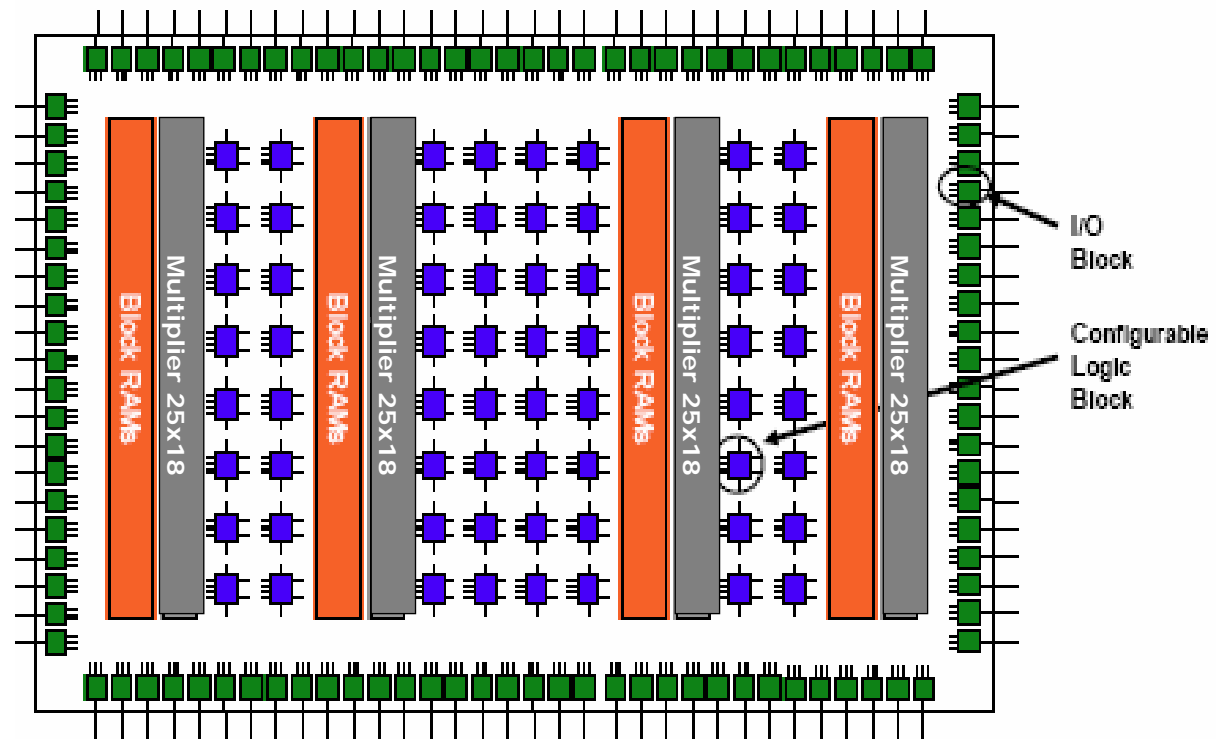
Actel Corp.

Quick Logic Corp.

} > 60% market

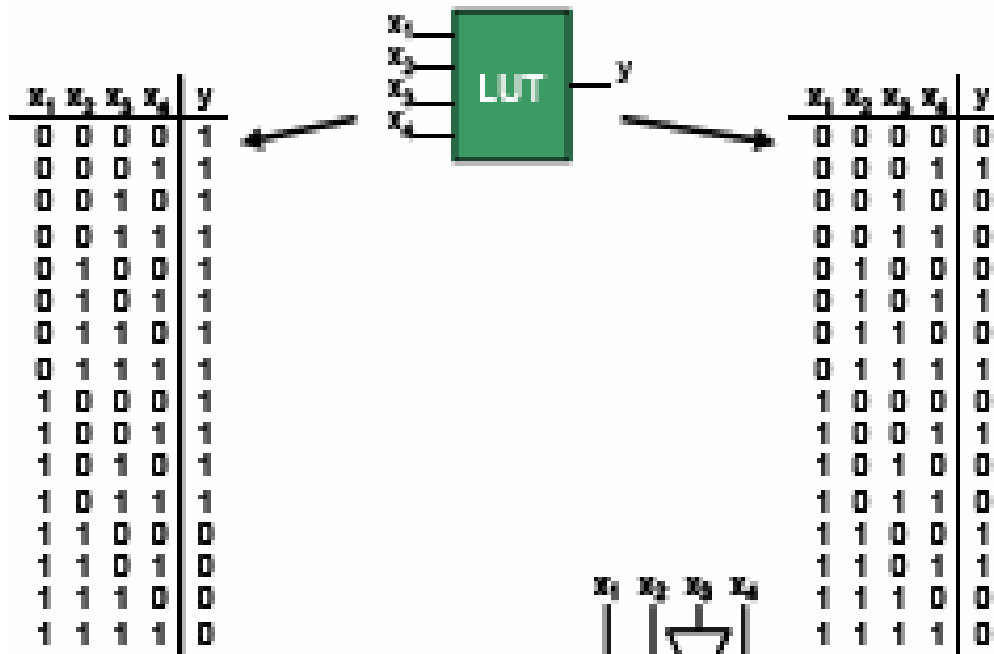
High End FPGA Architecture

- ◆ Large CLB array with wide input
- ◆ Better routing (diagonal)
- ◆ Block RAM
- ◆ Wide multiplier
- ◆ Low Voltage
- ◆ Low Power

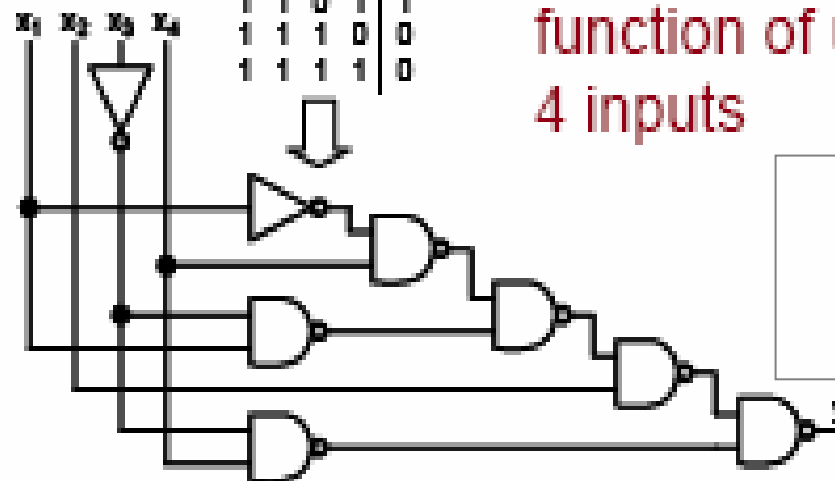
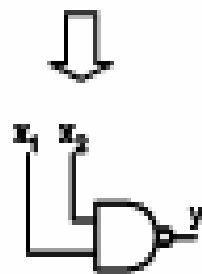


Source: [Xilinx, Inc.]

CLB: Look-up Table (LUT) + Memory



- Look-Up tables are primary elements for logic implementation
- Each LUT can implement any function of up to 4 inputs



Virtex-5
6 input LUT

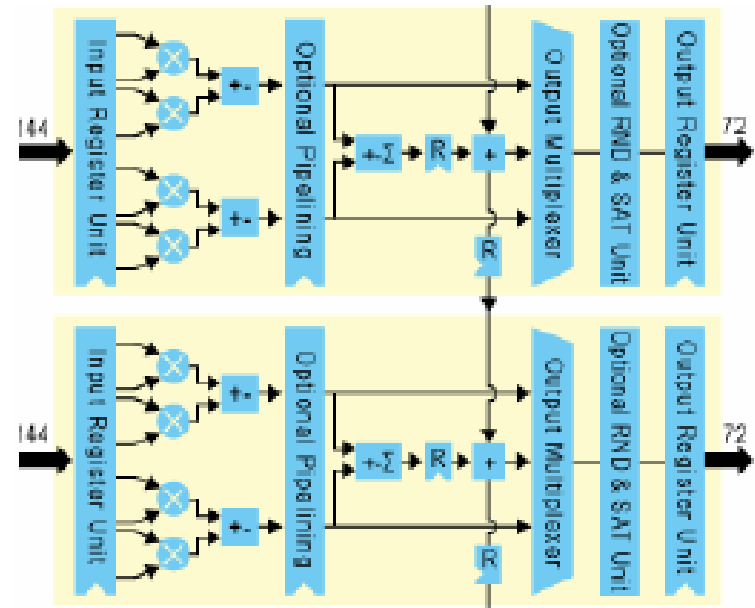
Prefabricated Functionality

- ◆ Some functionality maps bad on LUTs (multiplication)
- ◆ Some functionality is required in most designs (addition)
 - ⇒ Hardwired macros for frequently used functions
 - ◆ Adder
 - ◆ Multiplier
 - ◆ MAC
 - ◆ ALUs
 - ◆ PCI Express

Runs much faster ~ 500 MHz

Costs no routing resources

Low power Vs Soft IP

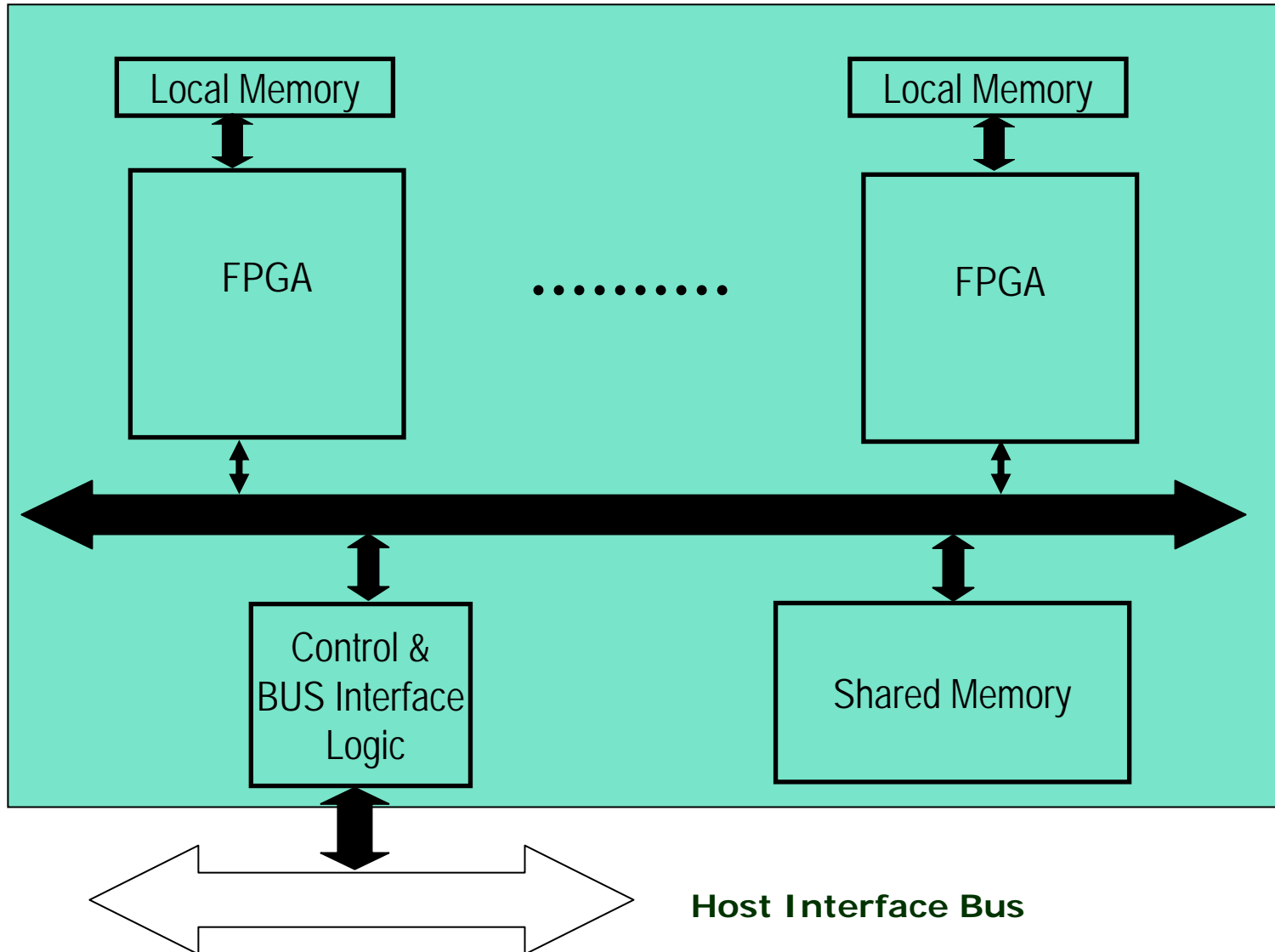


Acknowledgement: Altera

Reconfigurable Computing Boards (Accelerator)

- ◆ **Single or many interconnected FPGAs chips**
- ◆ **Support different Bus standards (host interface)**
 - PCI, PCI-X, PCIe, VME, FSB
- ◆ **Local on-board memory – SRAM, DDR,DDR2**
- ◆ **Driver API functions includes functionalities such as Reset, Open, Close, Set clock, configure, DMA, Read, Write**
- ◆ **Board-to-Board interface**

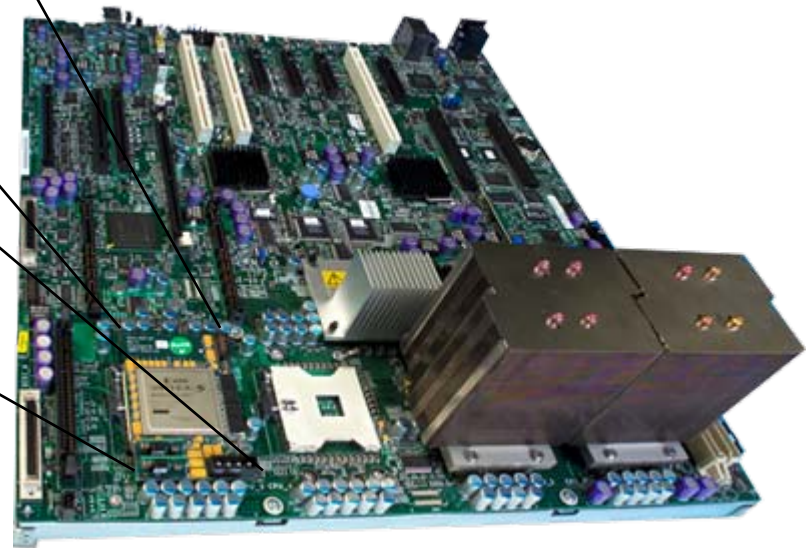
General RC Board Architecture



Virtex-5 on Intel Xeon Server



M1 ACP Module



- In socket accelerator module
- Xeon CPU pinout
- Intel FSB bus interface (soft logic) inside FPGA

Acknowledgement: Xilinx

DRC RPU for AMD Opteron HT

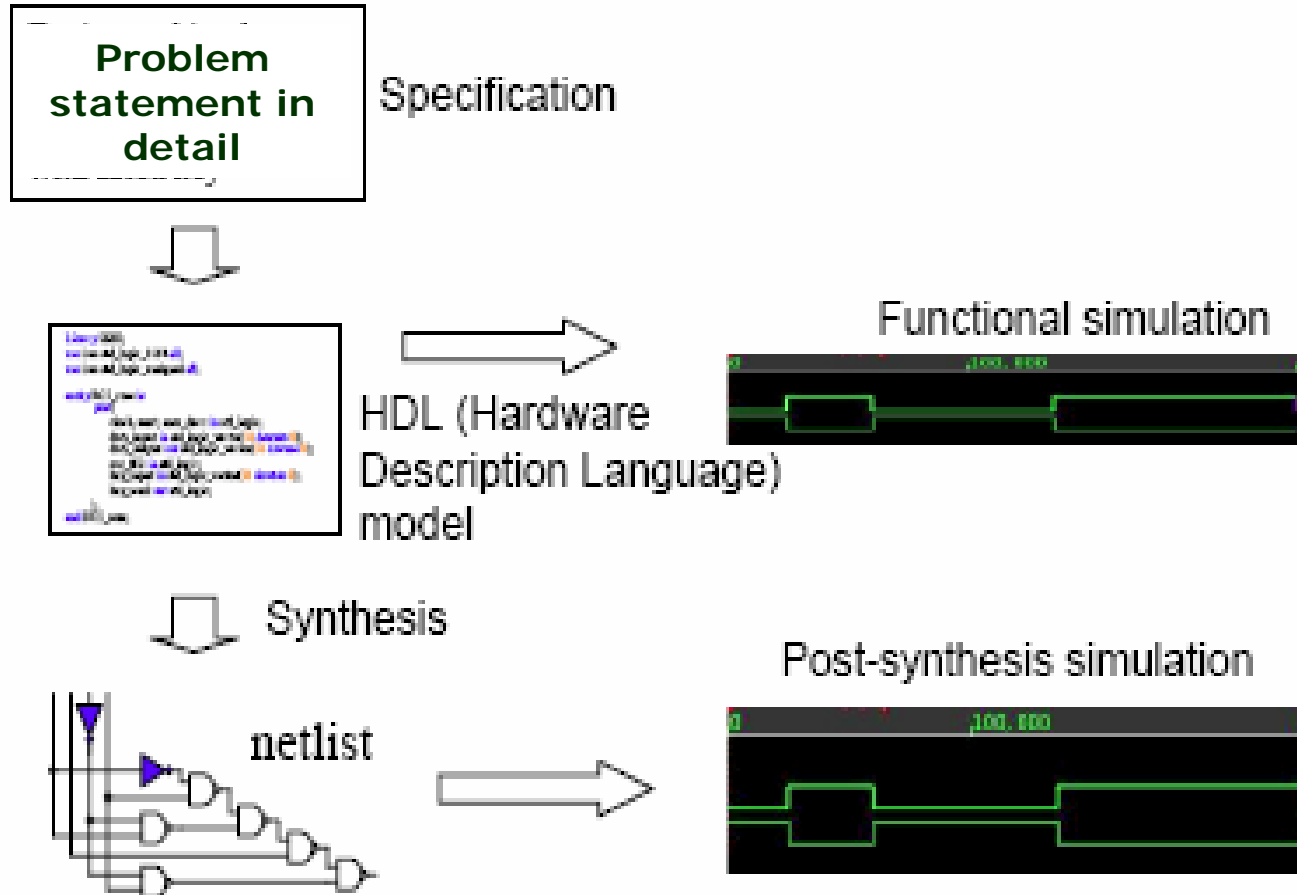


- In socket FPGA based accelerator modules
- HyperTransport (HT) Socket interface
- Fits in AMD processor sockets

Acknowledgement: DRC

Design Cycle

◆ Specification to Synthesis

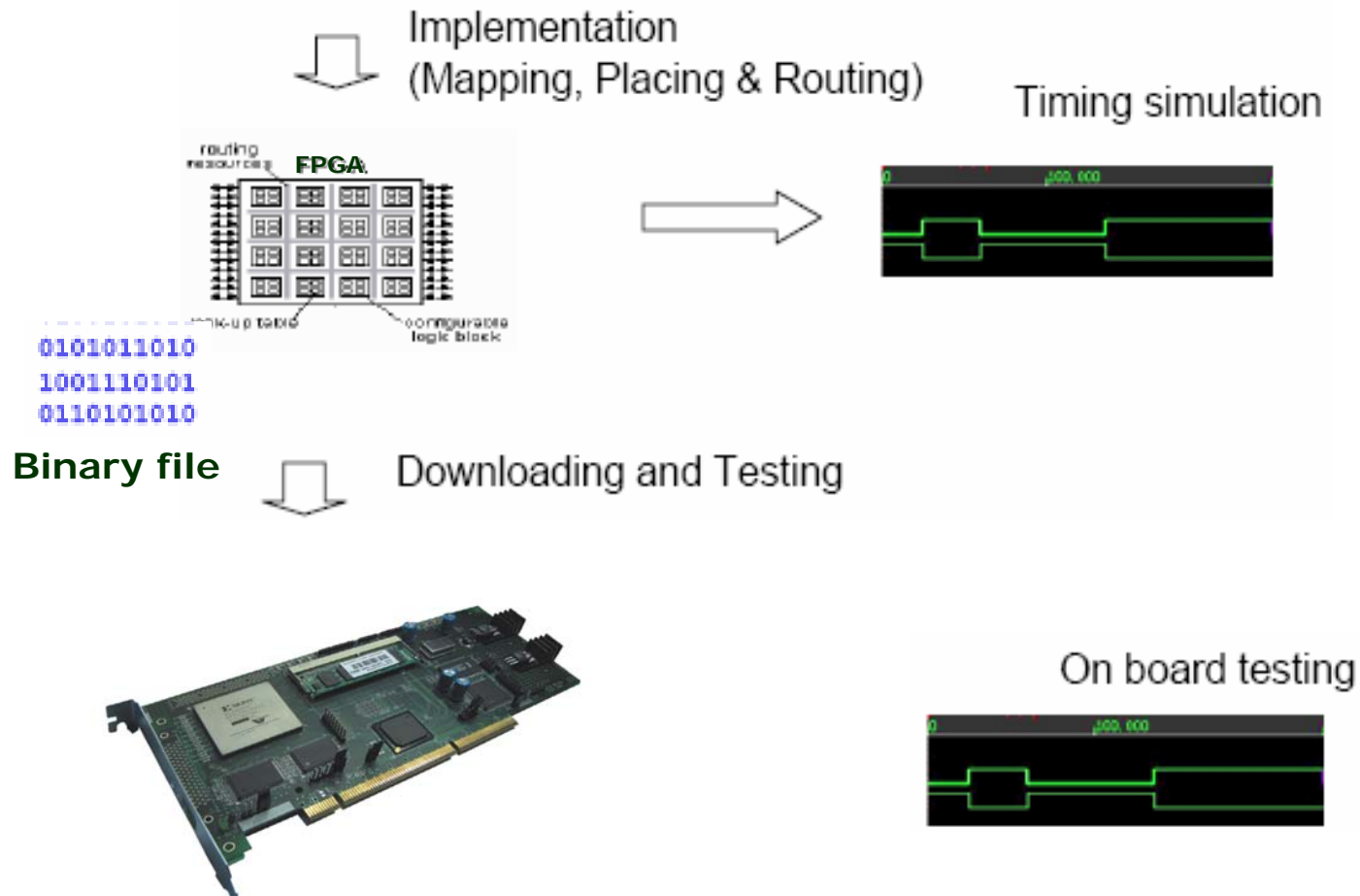


ISE Alliance and Foundation
Series Design Software

Xilinx
or other tool

Design Cycle

◆ Implementation to Downloading & testing

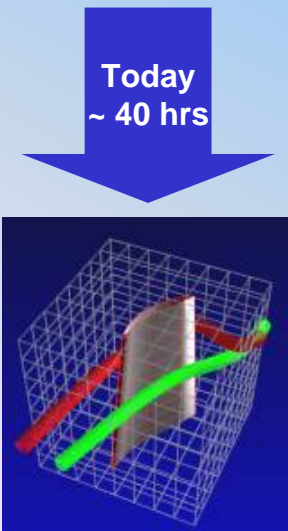


Outline

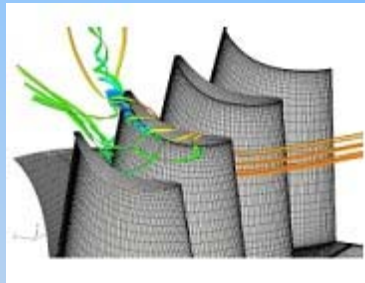
- Motivation
- Architecture
- Applications
- Performance
- Summary

One Example: CFD Motivation

Computational Fluid Dynamics: Used to simulate the movement of air & fuel through a jet engine

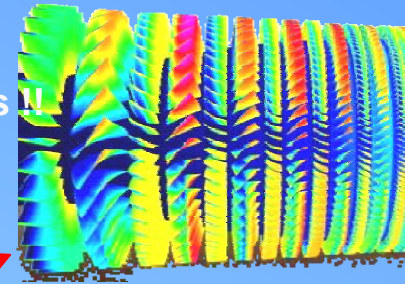


Single
Blade



Blade
Row

3.1×10^9
GFLOPs



Full
Compressor

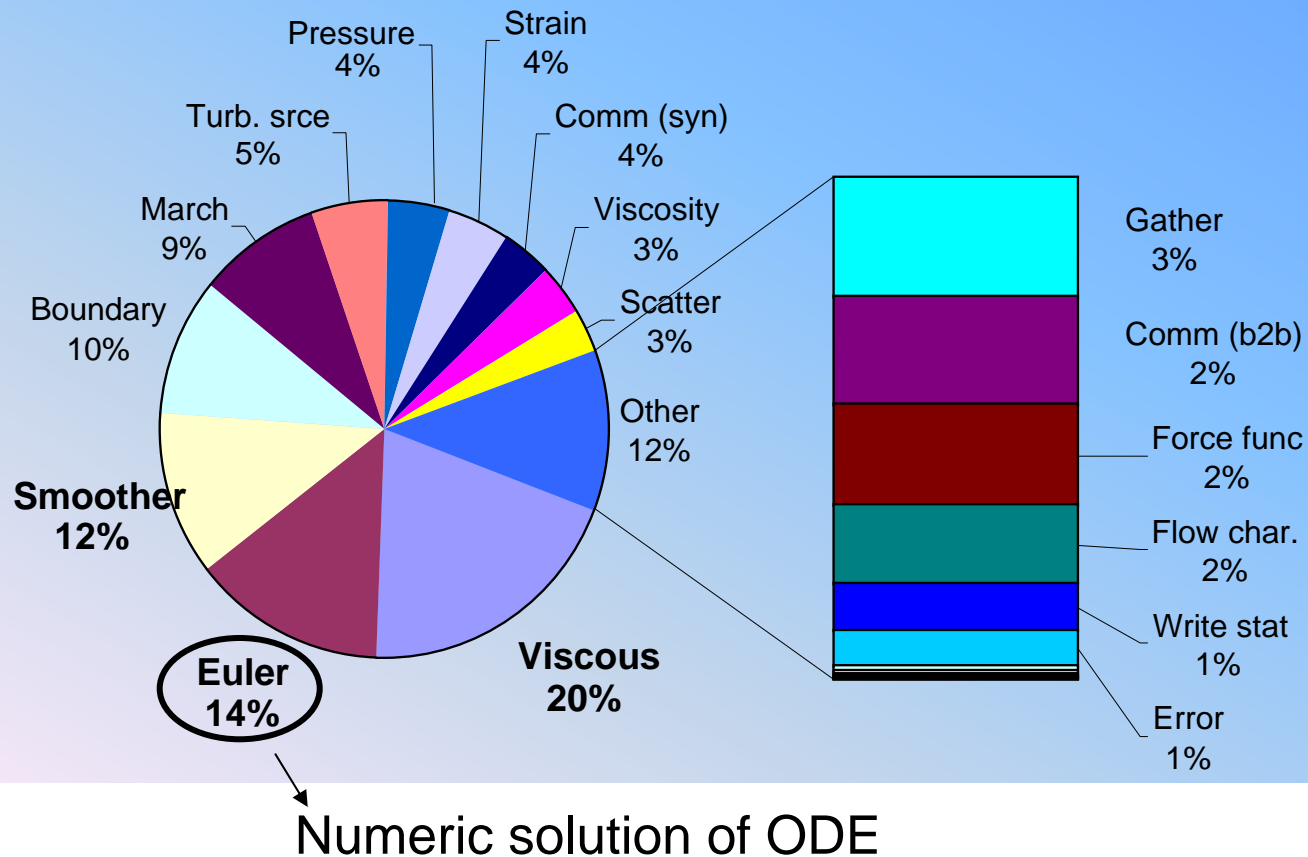
Benefits:

- Higher efficiency
- Greater part life
- Lower noise
- Fewer prototype builds

Source:OpenFPGA

Breakdown of CFD Processing

GE Production Multi-Grid Analysis on 4 Processors in GE GR Beowulf Cluster 129 x 73 x 49 grid size, Total Run Time ~40 hours [wall clock], ~150 hours [CPU time]



Source: OpenFPGA

Programming Languages

◆ HDLs (Hardware Description Languages)

- **VHDL** (Very high speed integrated circuit Hardware Description Language)
- **Verilog**

◆ Text Based HLLs (High Level Languages)

- **Handel-C** (DK design suite from Agility; earlier from Celoxica)
- **Impulse-C**
- **Mitrion-C**

◆ Graphical Tools

- **DSPLogic**
- **SysGen**

Hardware Description Language

VHDL

```
entity test is
port (
  a, b, c : in std_logic_vector(5 downto 0);
  op : out std_logic_vector(5 downto 0)
);
end entity test;
```

```
architecture test_a of test is
begin
```

```
...
```

```
if(b < "001010")then
```

```
  op <= a + b;
```

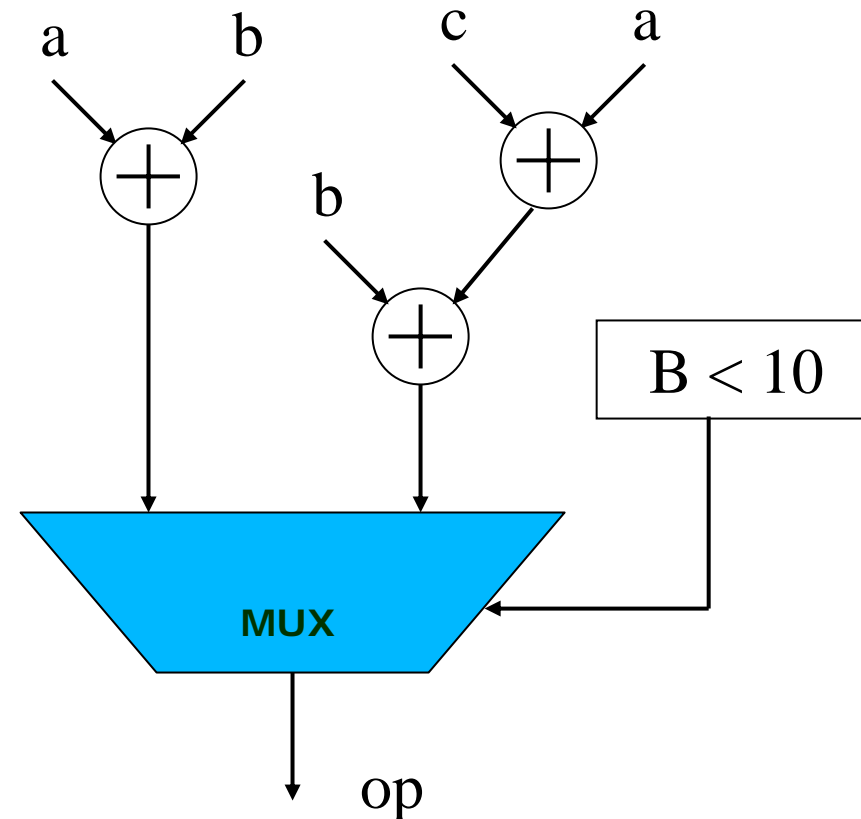
```
else
```

```
  op <= c + a + b;
```

```
end if;
```

```
...
```

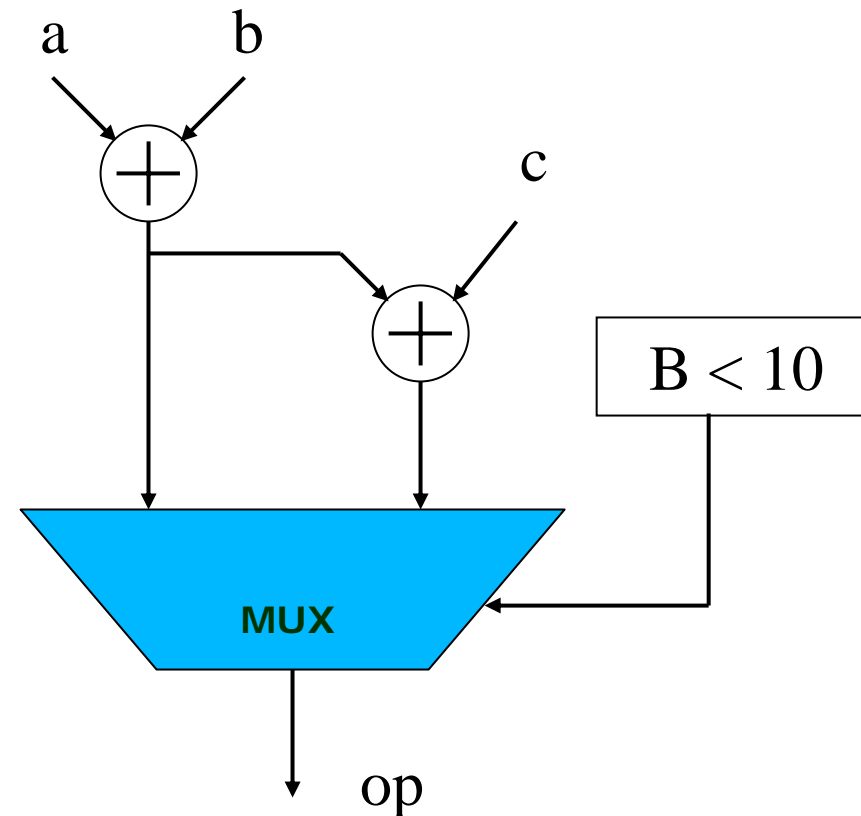
```
end architecture test_a;
```



Hardware Inference: coding style

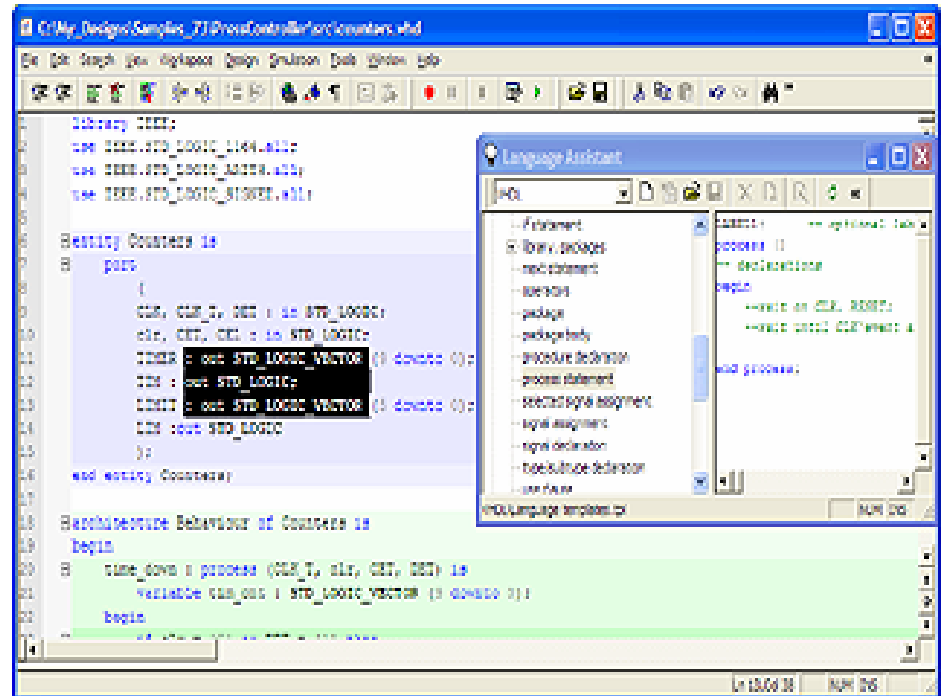
```
entity test is
port (
  a, b, c : in std_logic_vector(5 downto 0);
  op : out std_logic_vector(5 downto 0)
);
end entity test;
```

```
architecture test_a of test is
begin
  ...
  if(b < "001010")then
    op <= a + b;
  else
    op <= c + (a + b);
  end if;
  ...
end architecture test_a;
```



High Level Languages

- C-to-hardware compiler to generate synthesizable hardware.
 - Trident (open source compiler)
- Load the resulting bitmap file to the FPGA device



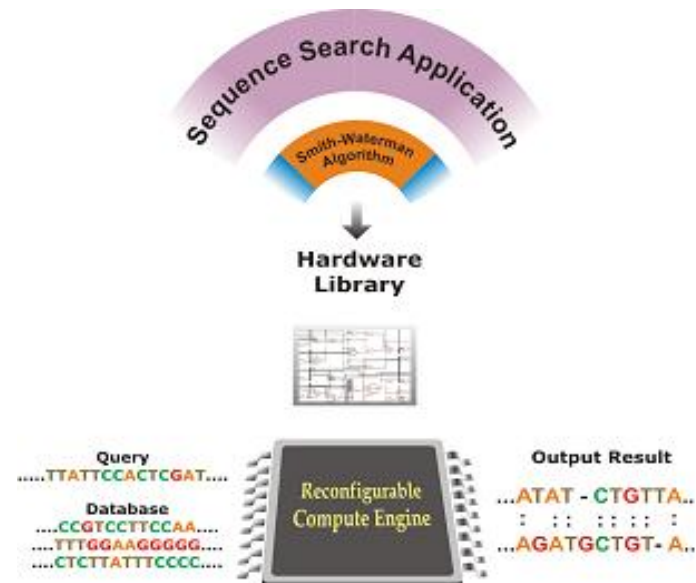
Outline

- Motivation
- Architecture
- Applications
- Performance
- Summary

Do FPGAs provide a significant performance increase in real applications to justify a paradigm shift?

Performance on C-DAC's RC

Bioinformatics Sequence search taking **528 days** using pure software solution was completed in just **12 days!**



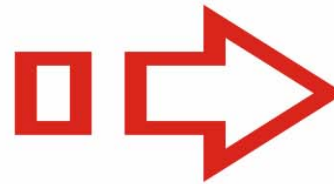
...Up to 90x acceleration

Multi card support for additional performance gain (2x,3x..)

Comparable to a Computing Cluster



comparable to a
Computing Cluster !



Computing cluster

RC Performance

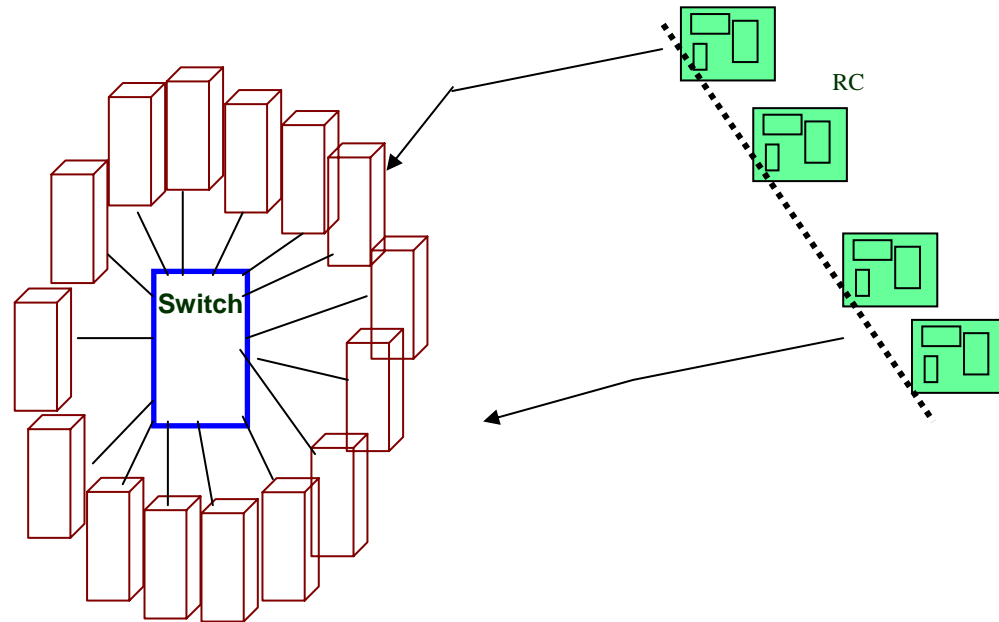
- ◆ **Host to RC bandwidth**
 - **Interface**
 - **Amount of Data**
 - **Mode of Data Transfer**
- ◆ **Data Locality**
- ◆ **Overlapping Computation with Communication**
 - **Double buffering**
- ◆ **Optimized library (logic) for speed**

Reconfigurable Supercomputers

HPC community has now started looking towards this technology for supercomputing needs.

*“Efficient high performance computing using parallel and distributed systems of both **reconfigurable hardware resources and conventional microprocessors**”*

RC cluster



- RC HW with state of the art FPGAs in each node.
- Enabling application for cluster
- MPI

Major Benefits of RCS

- **Results in less time**
- **Cost effective solution than a cluster**
- **Performance comparable to a high-end cluster.**
- **Low power consumption compared to a cluster.**
- **Less space required compared to a cluster.**

Power Metrics

CPU

- AMD Opteron: 55W .. 120W (frequency scaling)
- Intel Xeon: 65W .. 150W (frequency scaling)

FPGA

- Xilinx Virtex-5: 15W .. 30W

Complex thermal design for > 30 W power envelope

- Space requirements + fan power increase exponentially

FPGA:

- Much lower power for the same compute performance
- Significantly simpler thermal solution design

Source: Xilinx

Lessons Learned

- ◆ **Most significant acceleration can be achieved when developing the code from scratch; developer has the freedom to:**
 - Structure algorithm to take advantage of RC platform resources and structure
 - Select effective SW/HW partitioning scheme
 - Select data format and flow that maps best on RC
- ◆ **Effective use of Automated tool requires limited hardware knowledge**
 - RC architecture specific techniques: Pipelining, streams...
 - Memory organization
- ◆ **High level tools have usually performance penalty**
 - May be helpful for initial development

Path Ahead

- ◆ **Programming standards & Portability among platforms required**
 - Many (different!) SDKs are available
 - Tools still hard to use
- ◆ **API Standardization and Availability of Large libraries is must**
- ◆ **Lessons learnt will be useful to widely use FPGAs in HPC**
- ◆ **New FPGAs and other Programmable Devices with rich resources including multiple ALUs..in pipeline**

The Performance Gain, Energy savings and small footprints are compelling to Look further

FPGA Communities

...Taking Care of some of the issues

◆ **OpenFPGA** [www.openfpga.org] consortium to foster RC;

Established in 2004

General API Specification July 2008

- Industry standard API for high-level language
- Portability across a wide range of available RC platforms.
- Minimal essential functionality to support general application acceleration
- Simple to use, understand and implement.

◆ **FPGA High Performance Computing Alliance (FHPCA)**

[www.fhpca.org]

Established in 2004

◆ **Center for High Performance Reconfigurable Computing (CHREC)**

[www.chrec.org] Since 2007

Thank You.
yogindra@cdac.in