THE UNIVERSITY of WISCONSIN MADISON

Recent Advances in Architectures and Tools for Complex FPGA-based Systems

Michael Schulte

University of Wisconsin

TWEPP 2009

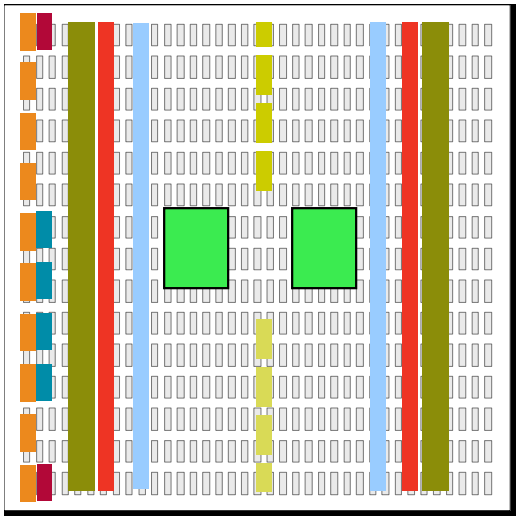
Slides provided by various sources

Outline

- Overview of FPGA Architectures
- Recent FPGA Trends
- Tools for FPGA-based Systems
- FPGAs in Particle Physics Experiments
- Conclusions

FPGA Architectures

FPGA Architecture



- Configurable Logic Blocks
- Embedded Memories
- Digital Signal Processing Slices
- High-performance Clocking
- Parallel I/O
- Serial I/O Transceivers
- PCIe Interfaces
- Ethernet Controllers
- Embedded Microprocessors

Modern FPGAs contain a wide variety of resources

These resources and their interconnections can be reconfigured for target applications

Configurable Logic Blocks

Configurable Logic Blocks

- Have reconfigurable hardware functionality
- Have reconfigurable interconnections

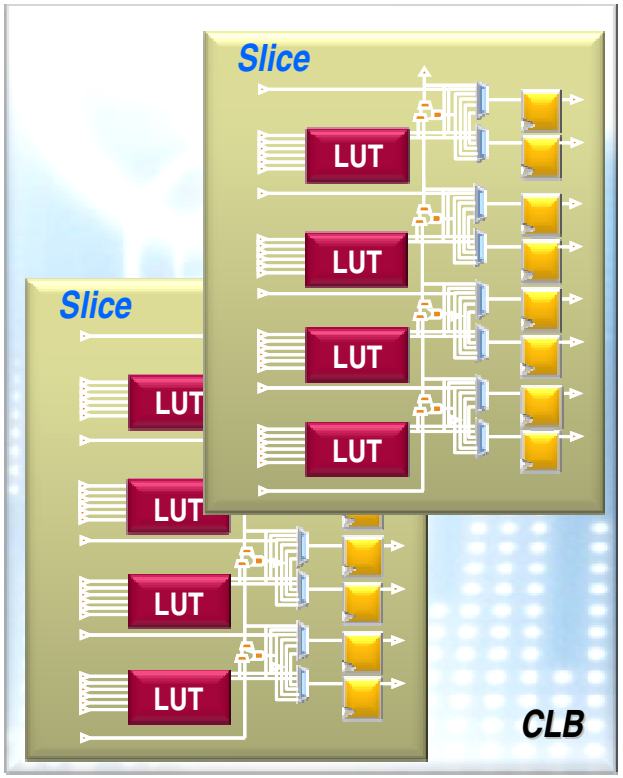
Virtex-6 Configurable Logic Block

- Each CLB contains two slices
- Each slice contains four 6-input Lookup Tables (LUT6)
- Slices also contain fast carry-logic, multiplexers, and registers

Each LUT6 can implement

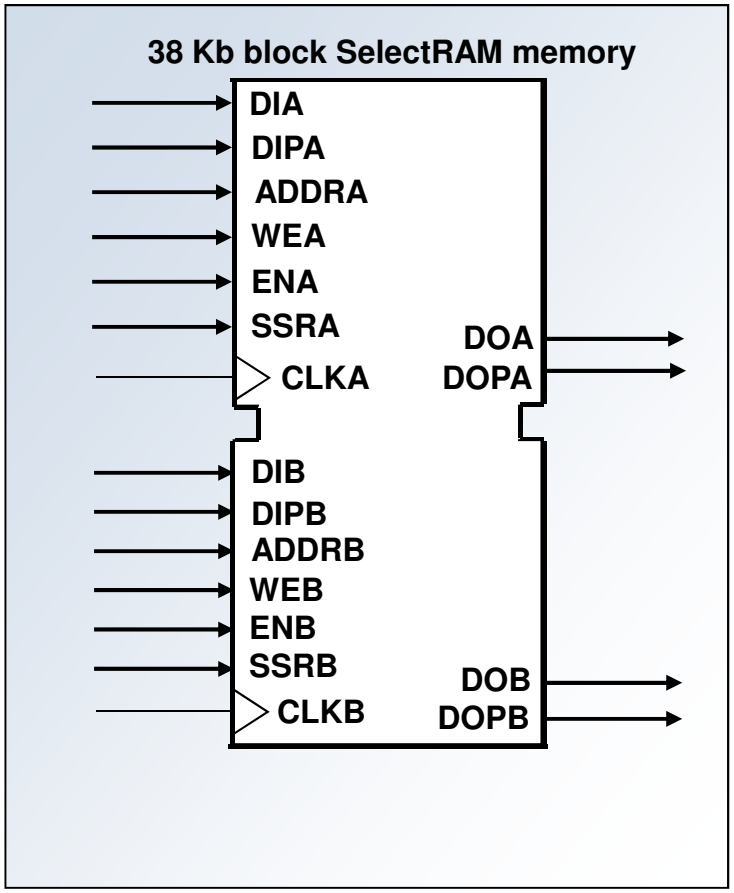
- Wide range of Boolean functions
- Memories of 64 bits
- Shift registers

Up to 118,560 slices per FPGA



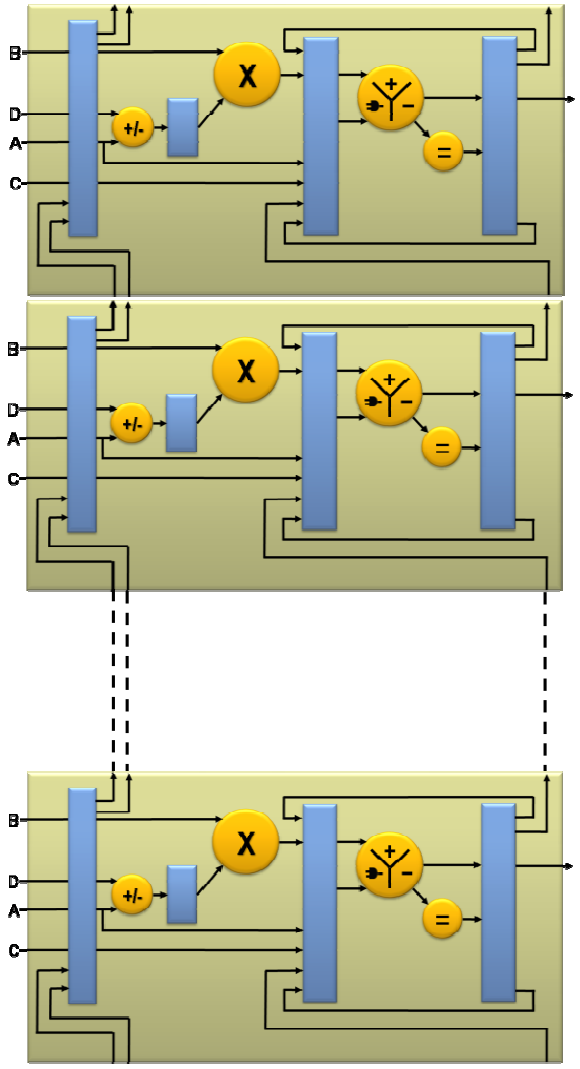
Block RAM

- Up to 38 Mb of RAM in 36-Kb blocks
  - Each 36-Kb block can be configured as two 18-Kb blocks
- Both the depth and the width of memory can be configured
- Dual-port memory
  - Each port has synchronous read and write capability
  - Different clocks for each port
- Several options for changing block RAM configuration

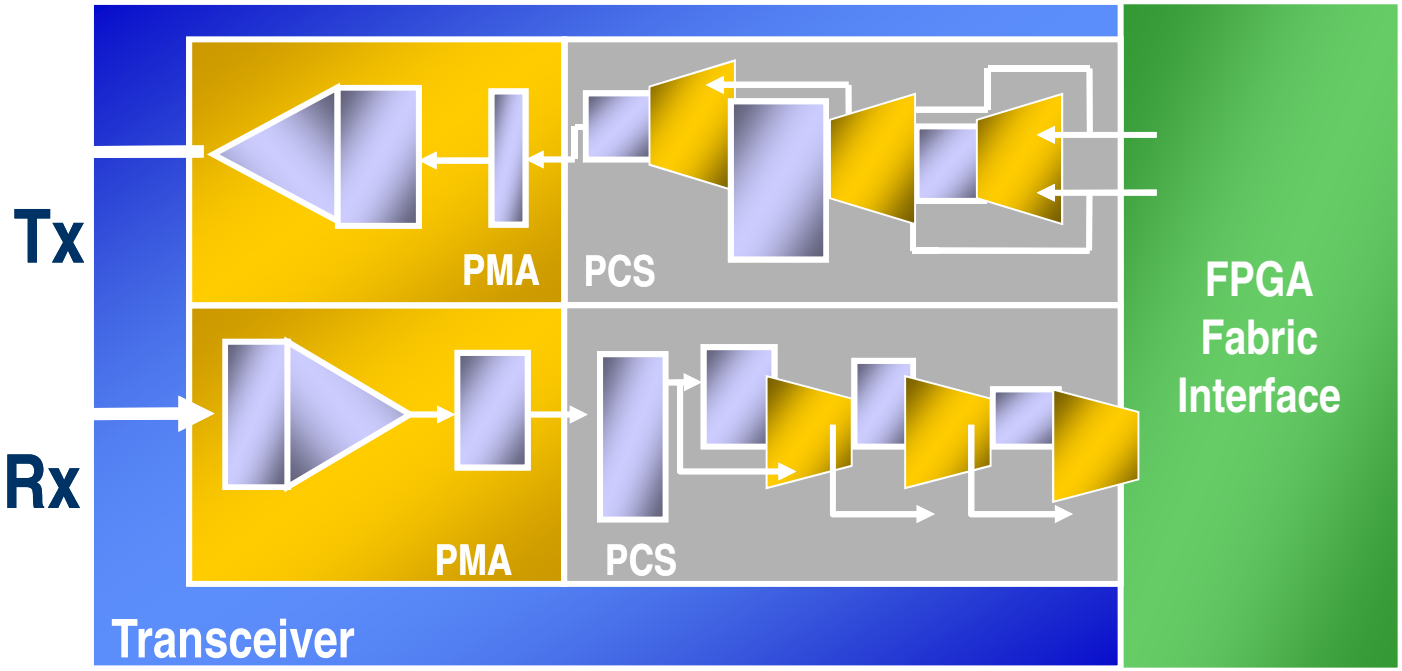


DSP Slices

- DSP slices provide fast arithmetic operations
- Virtex-6 DSP48E1 architecture
  - 30-bit add/subtract unit
  - 25x18-bit multiplier
  - 48-bit arithmetic and logic unit (ALU)
    - Add/substract/logic functions
    - SIMD operation
  - Final comparator
  - Chaining and feedback increase functionality
  - Up to 2K DSP Slices per FPGA

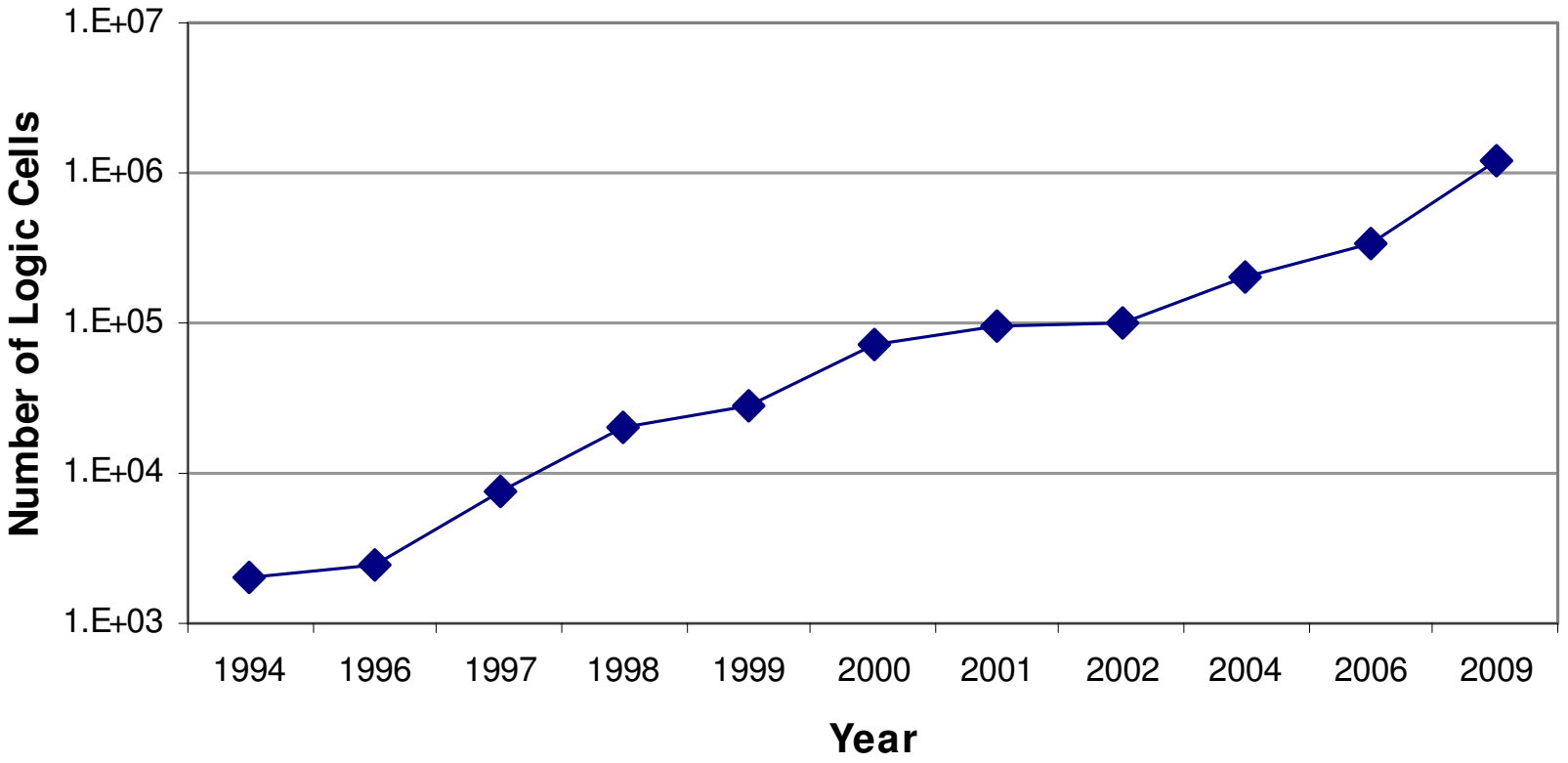


Multi-Gigabit Transceiver



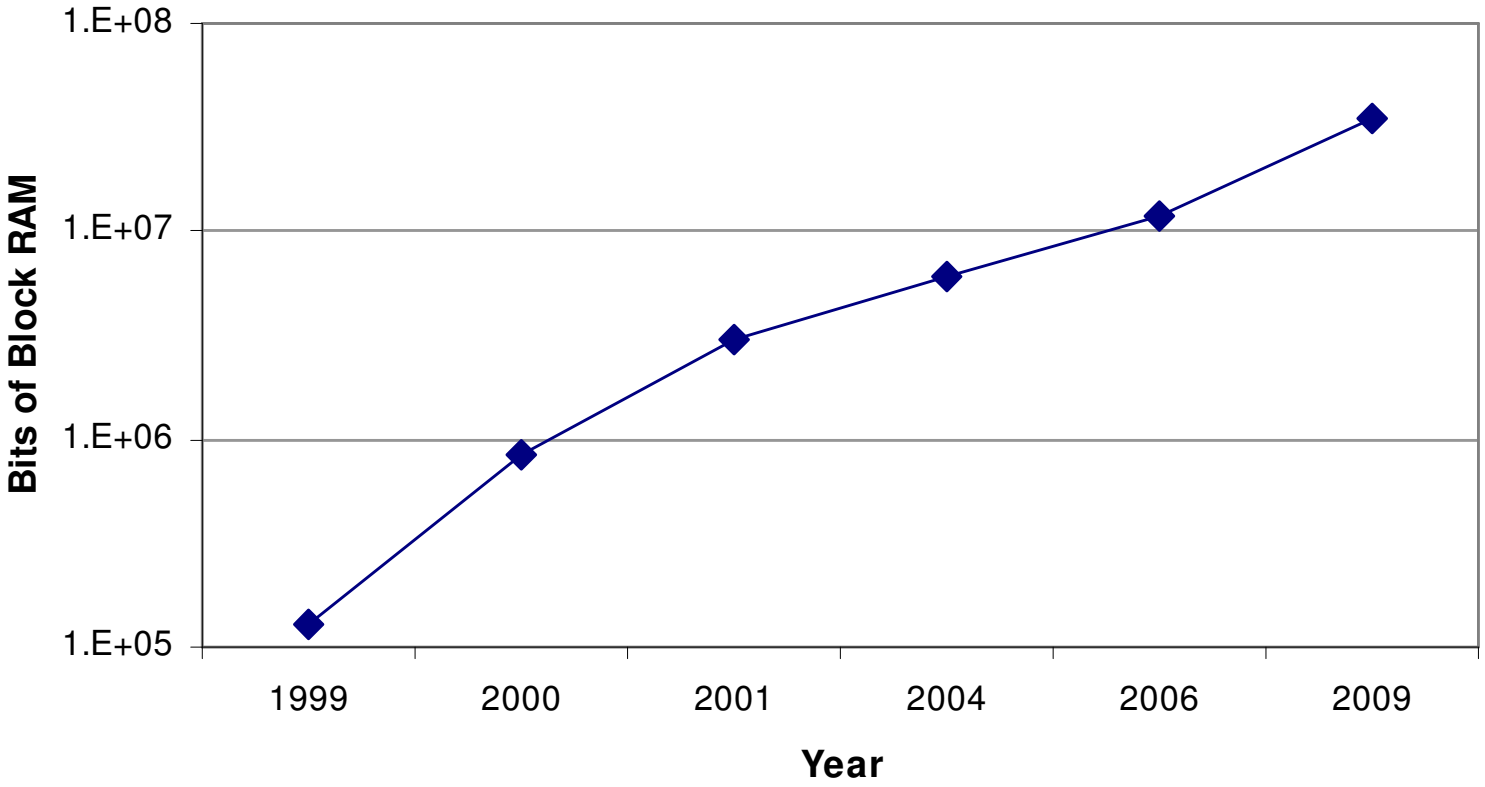
- Multi-Gigabit transceivers provide high-speed I/O
- GTX provides up to 6.5 Gbps per transceiver
- GTH provides up to 11 Gbps per transceiver
- XC6VHX565T has 24 GTH transceivers and 48 GTX transceivers
- Up to 576 Gbps of Serial I/O per FPGA

FPGA Trends – Logic Cells



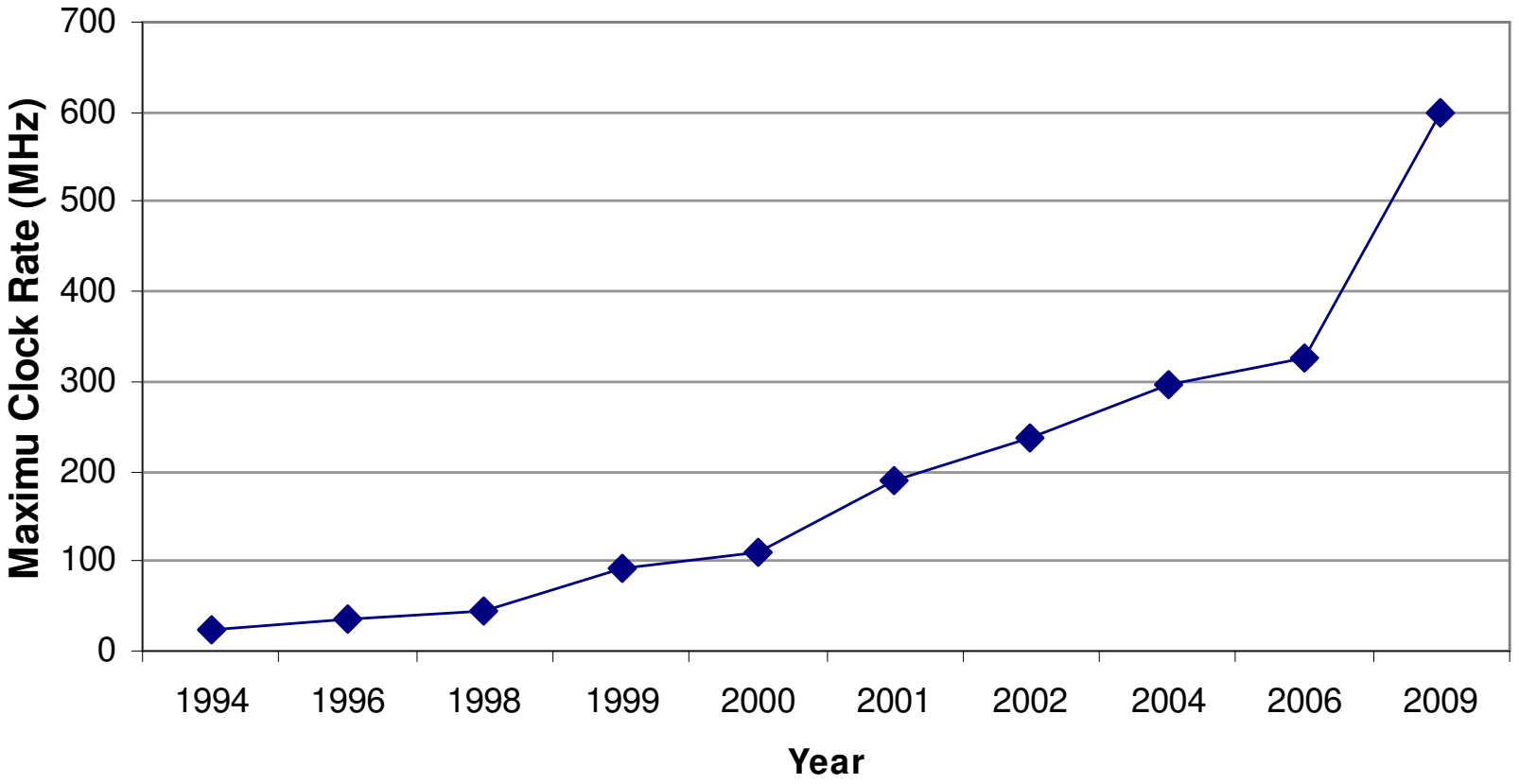
595x Increase in Logic Cells

FPGA Trends – Block RAMs



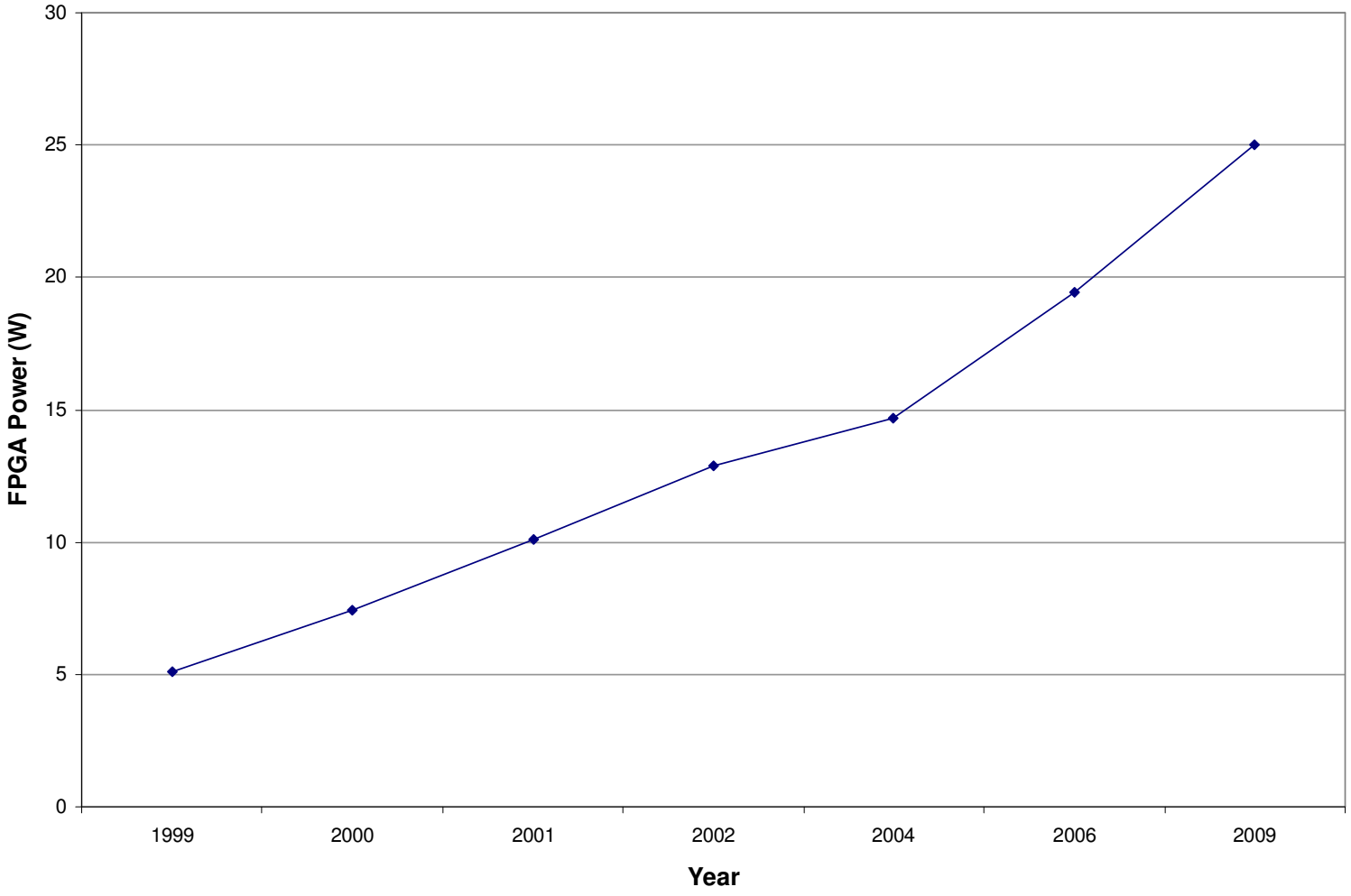
270x Increase in Block RAM Bits

FPGA Trends – Clock Rate



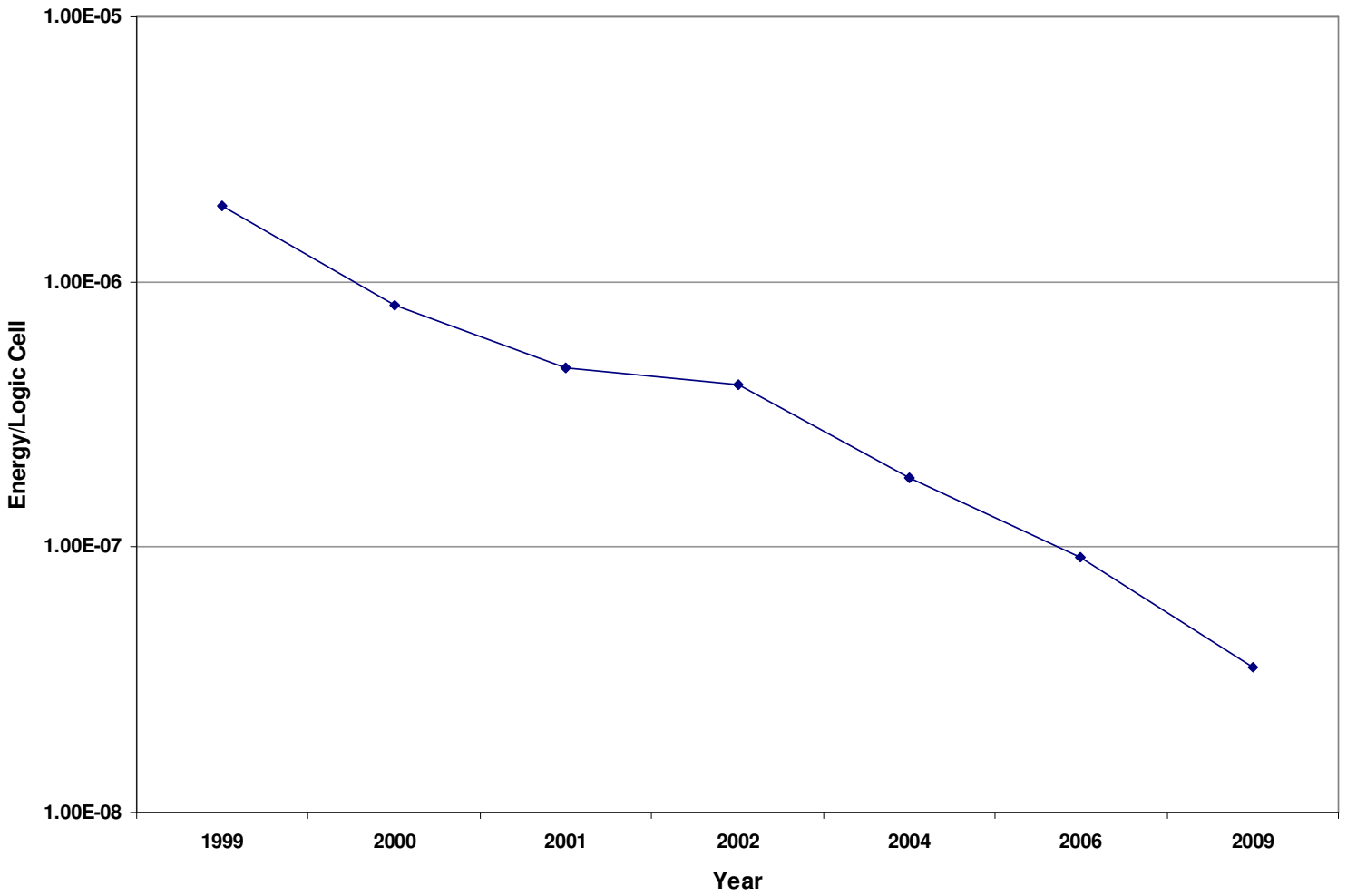
24x Increase in Clock Rate

FPGA Trends – Power



5x Increase in Power

FPGA Trends – Energy/LC

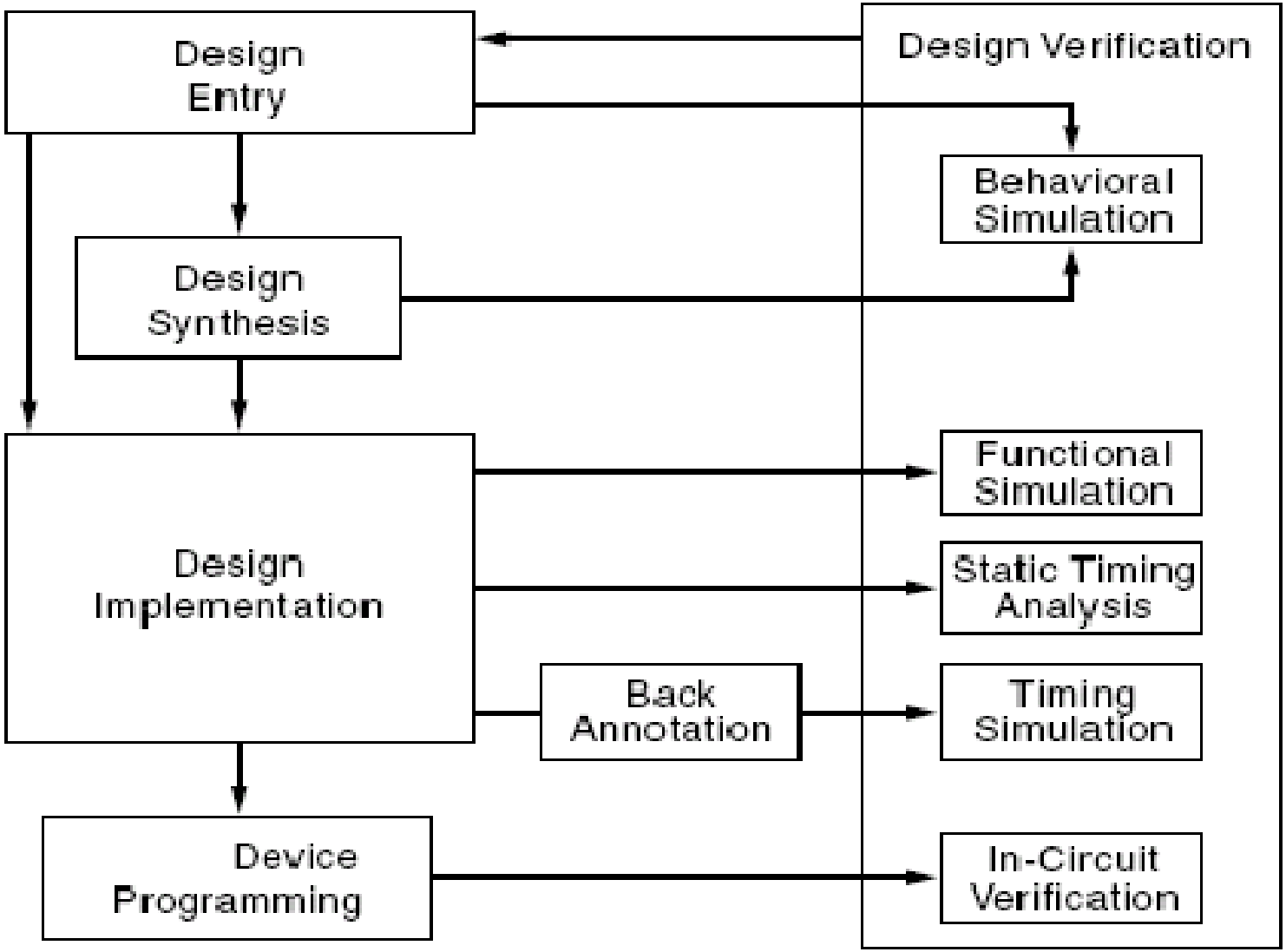


54x Decrease in Energy/Logic Cell

Virtex-6 FPGA Characteristics

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP48E1 Slices[2]	Block RAM Blocks			MMCMs[4]	Interface Blocks for PCI Express	Ethernet MACs[5]	Maximum Transceivers		Total I/O Banks[6]	Max User I/O[7]
		Slices[1]	Max Distributed RAM (Kb)		18 Kb[3]	36 Kb	Max (Kb)				GTX	GTH		
XC6VLX75T	74,496	11,640	1,045	288	312	156	5,616	6	1	4	12	0	9	360
XC6VLX130T	128,000	20,000	1,740	480	528	264	9,504	10	2	4	20	0	15	600
XC6VLX195T	199,680	31,200	3,040	640	688	344	12,384	10	2	4	20	0	15	600
XC6VLX240T	241,152	37,680	3,650	768	832	416	14,976	12	2	4	24	0	18	720
XC6VLX365T	364,032	56,880	4,130	576	832	416	14,976	12	2	4	24	0	18	720
XC6VLX550T	549,888	85,920	6,200	864	1,264	632	22,752	18	2	4	36	0	30	1200
XC6VLX760	758,784	118,560	8,280	864	1,440	720	25,920	18	0	0	0	0	30	1200
XC6VSX315T	314,880	49,200	5,090	1,344	1,408	704	25,344	12	2	4	24	0	18	720
XC6VSX475T	476,160	74,400	7,640	2,016	2,128	1,064	38,304	18	2	4	36	0	21	840
XC6VHX250T	251,904	39,360	3,040	576	1,008	504	18,144	12	4	4	48	0	8	320
XC6VHX255T	253,440	39,600	3,050	576	1,032	516	18,576	12	2	4	24	24	12	480
XC6VHX380T	382,464	59,760	4,570	864	1,536	768	27,648	18	4	4	48	24	18	720
XC6VHX565T	566,784	58,560	6,370	864	1,824	912	32,832	18	4	4	48	24	18	720

Conventional FPGA Design and Verification

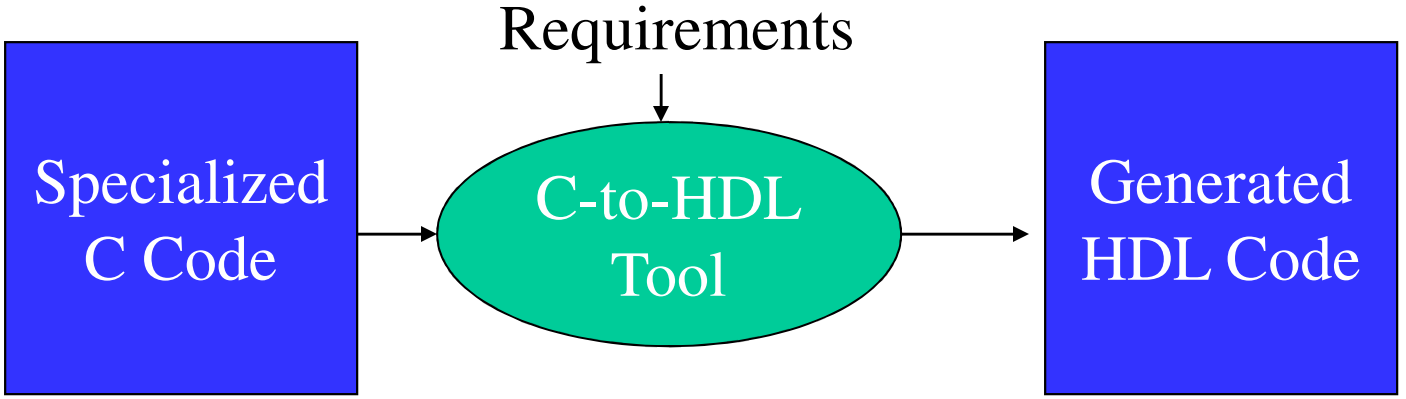


Tools for FPGA-based Systems

- FPGA-based systems are becoming increasingly complex
  - New tools can help manage this growing complexity
- These tools come from a variety of sources
  - FPGA companies (Xilinx and Altera)
  - EDA companies (Synopsys, Cadence, Mentor Graphics)
  - Open source projects
  - University research projects
- Several tools available to specify, design, test, and implement complex FPGA-based system
- This talk focuses on
  - Tools for C-to-HDL conversion and synthesis
  - Xilinx DSP tools

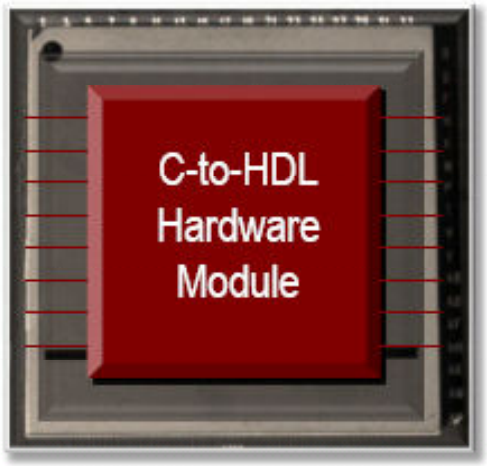
C-to-HDL Converters

- **Several tools have been develop that perform C-to-HDL conversion**
  - **Impulse-C (www.impulseaccelerated.com)**
  - **C-to-Verilog (www.c-to-verilog.com)**
  - **PICO Extreme FPGA (www.synfora.com)**
  - **C-to-Hardware Acceleration Compiler (www.altera.com)**
  - **NISC Technology (www.ics.uci.edu/~nisc)**
  - **SPARK Toolset (mesl.ucsd.edu/spark)**



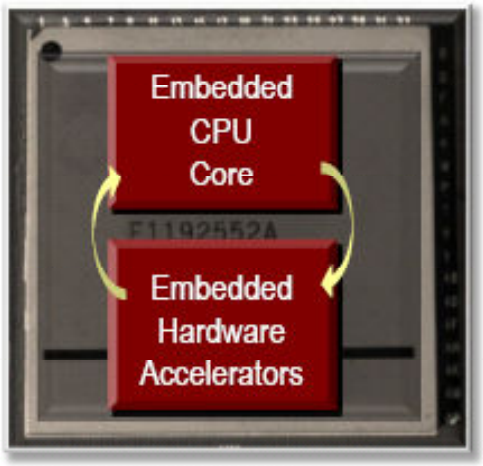
C-to-HDL Uses

1



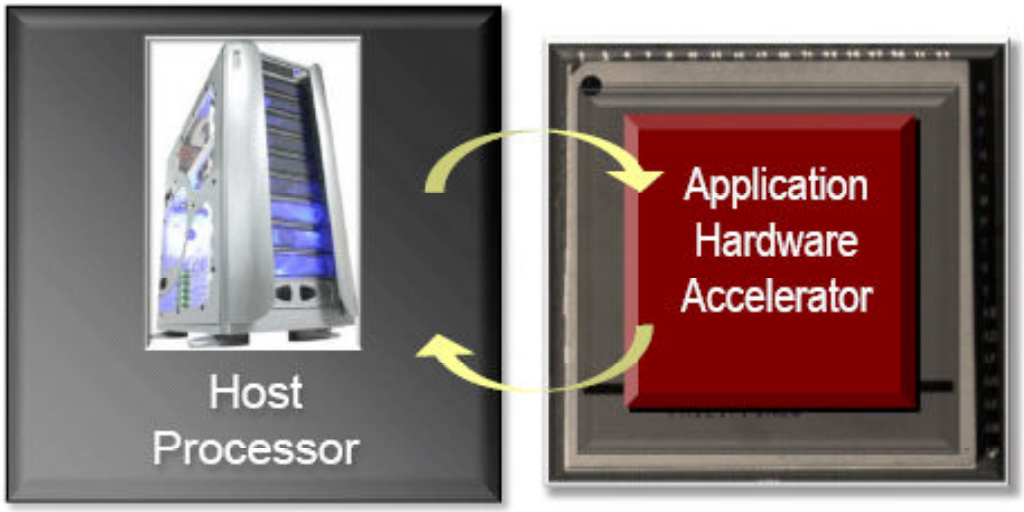
Create a hardware module

2



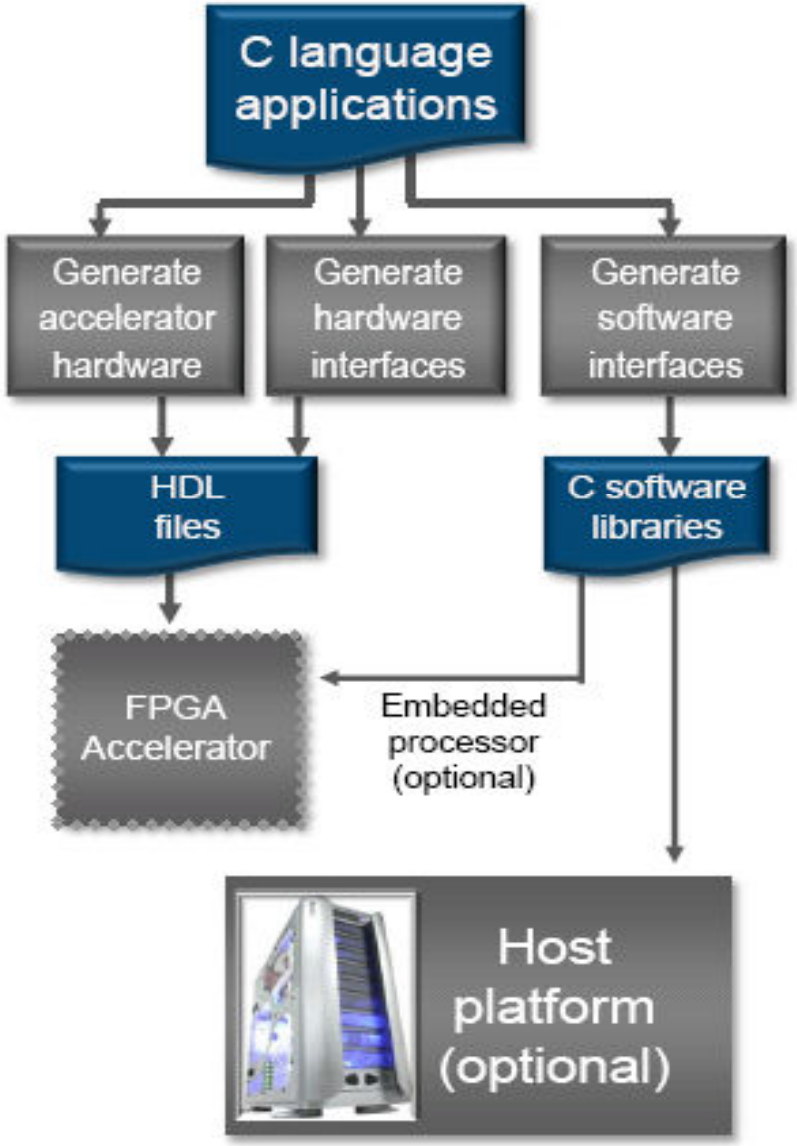
Accelerate an embedded CPU

3 *Accelerate an external/host CPU*



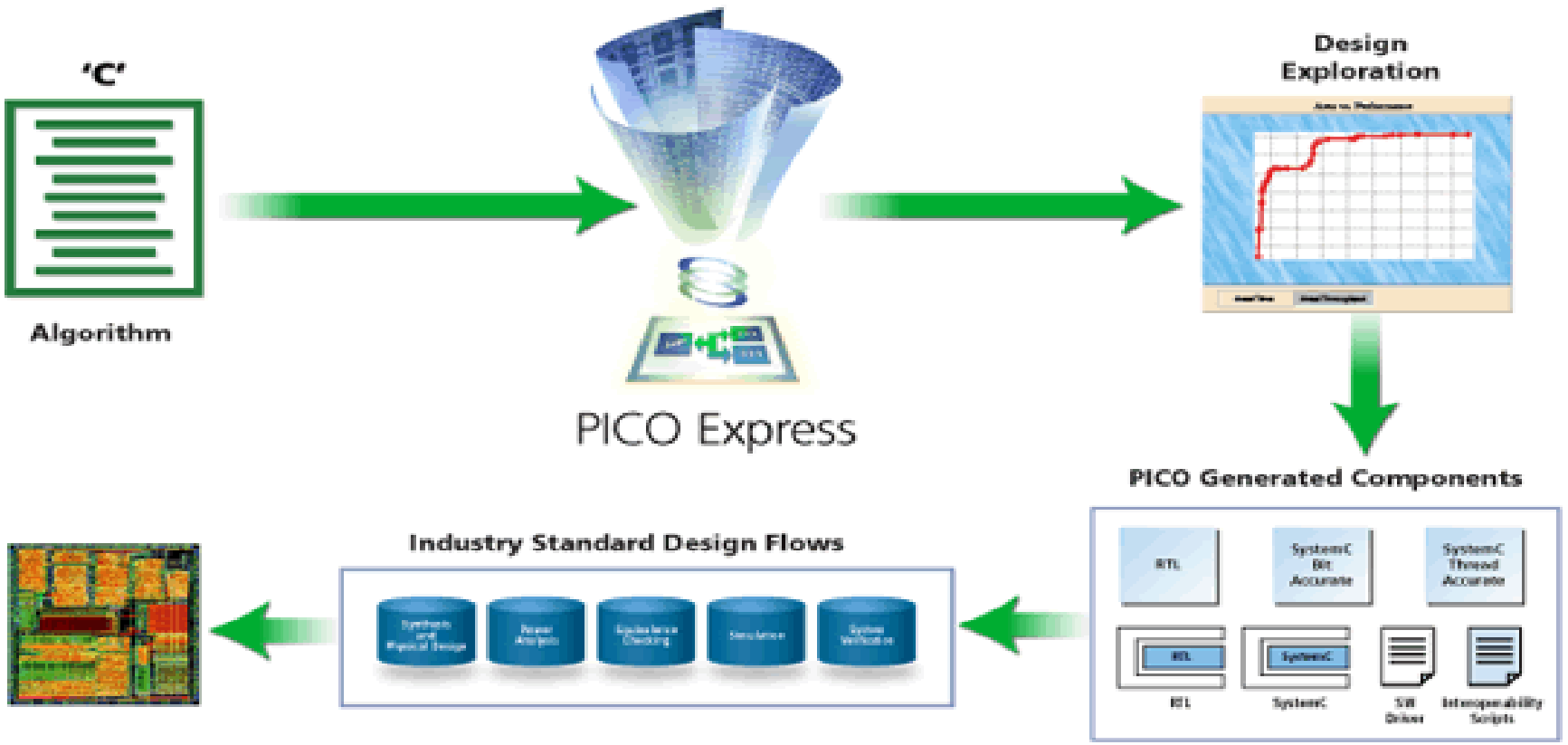
Impulse-C

- Intended for hardware/ software co-design
- Designs specified using C plus extensions
- Impulse-C
  - Optimizes C code for parallelism
  - Generates hardware/ software interfaces
  - Generates VHDL or Verilog
  - Provides interactive tools to improve results



PICO Express FPGA

- **Takes a C algorithm and a set of design requirements and automatically creates a series of implementation models**
- **Also creates testbenches, synthesis and verification scripts, and software drivers**

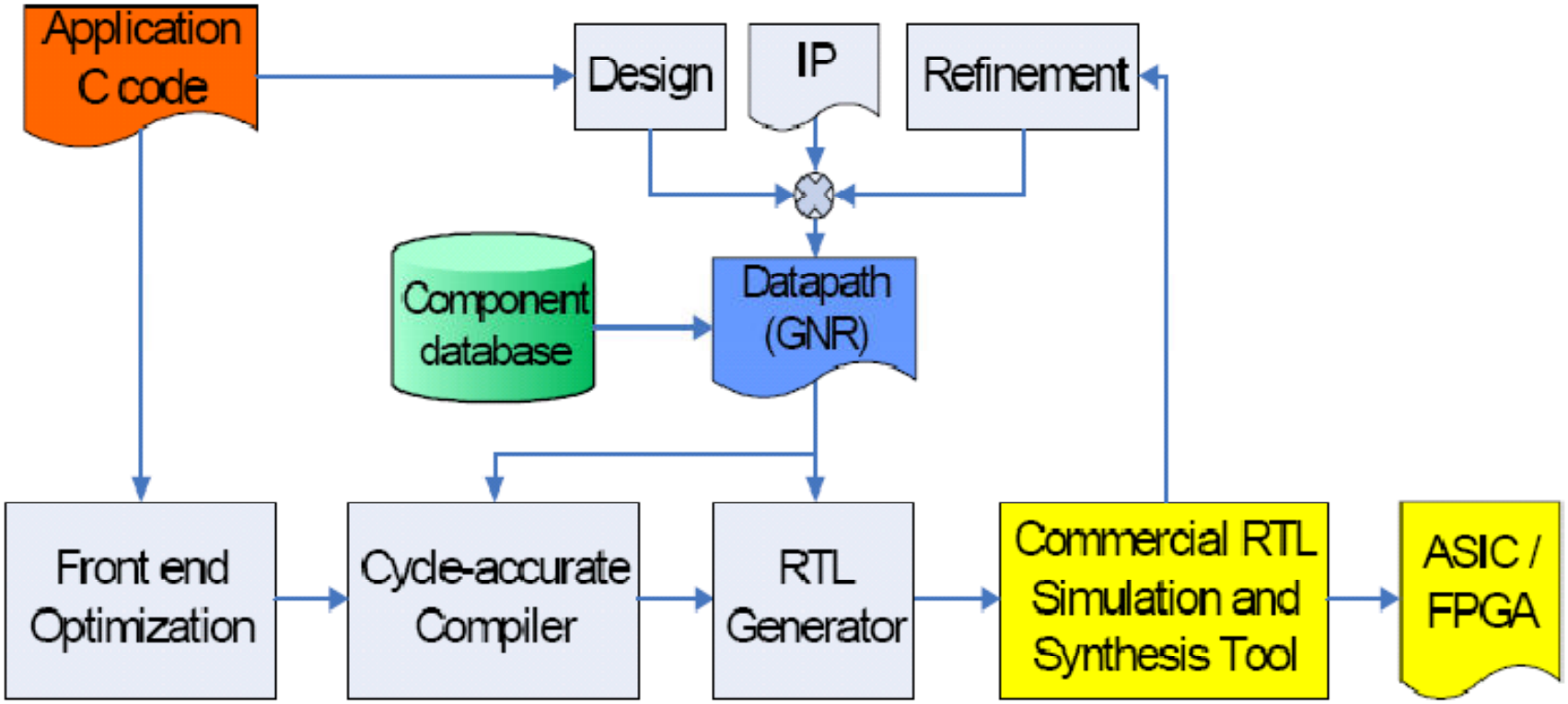


C-to-Verilog.Com

- Provides free online tool for C-to-Verilog conversion
- Converts standard C code with some restrictions
  - No recursive functions, structures, pointers to functions, or library function calls
- Separately specify implementation details
  - Word sizes, amount of loop unrolling, amount of hardware, FPGA family
- Automatically generates Verilog code and a Verilog testbench (with random inputs)
  - Performance of generated code can be quite good
  - Is difficult to read and debug
- Several design examples available online

NISC Technology

- **Provides an automated method for generating custom processors for a target application**
  - **Generates the processor datapath, control bits, and data memory**
  - **These are then mapped to an ASIC or FPGA using conventional tools**

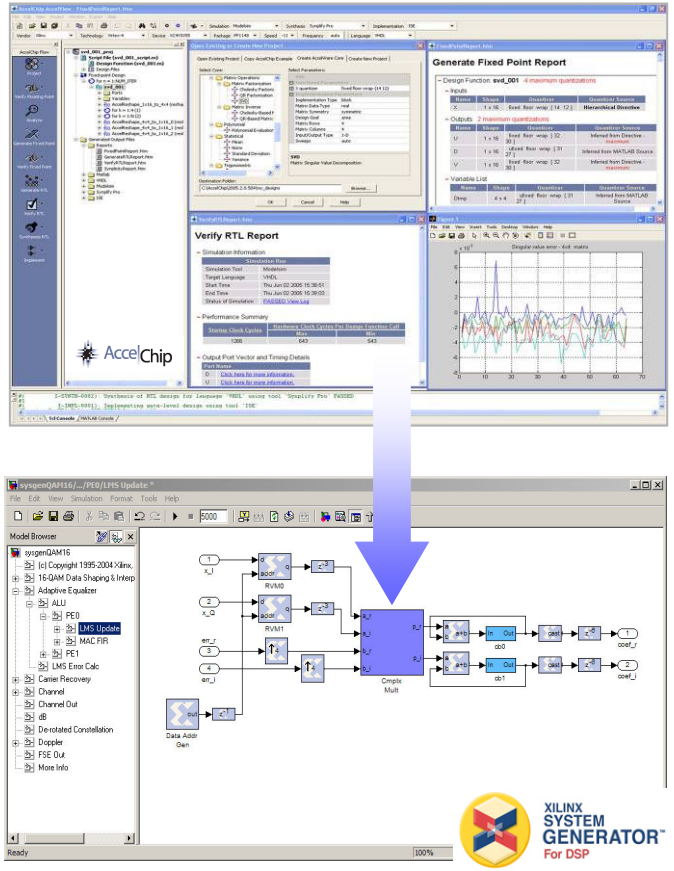


Summary of C-to-HDL Tools

- Simplify the design process
  - Especially for people not familiar with HDLs
  - Often generate useful resource – C support code, test benches, hardware interfaces, scripts, etc.
- Facilitate design-space exploration
- Can generate surprisingly good designs
  - As good or better than hand-optimized designs in many cases
- However,
  - C code has restrictions and/or needs extensions
  - Need to be careful with coding style
  - Often designed for hardware acceleration
  - Generated code is usually difficult to read and debug
- Tools should continue to improve

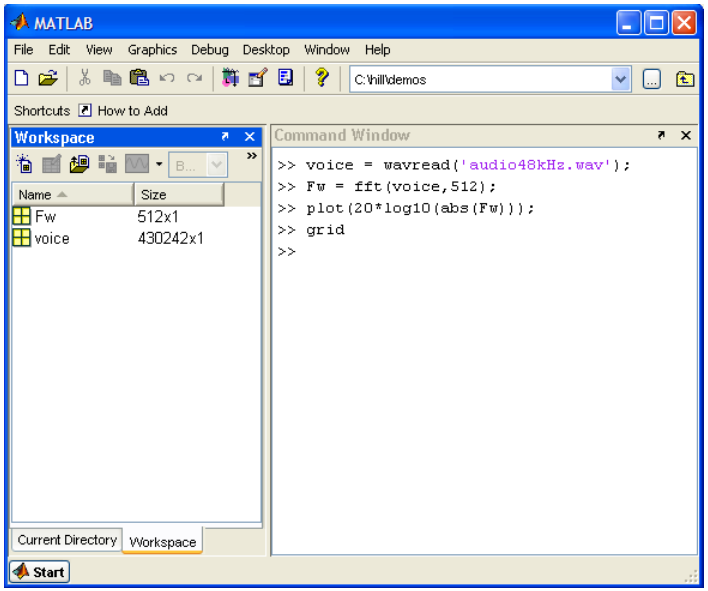
Xilinx DSP Tools

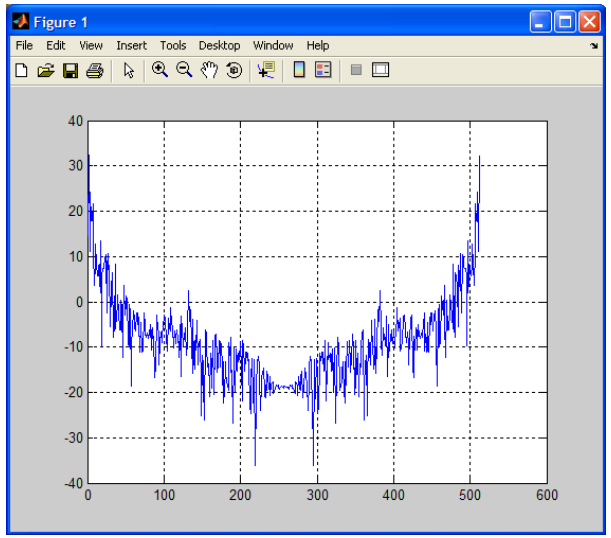
- Xilinx tools for digital signal processing include
  - System Generator for DSP
  - AccelDSP
- Integrated with other Xilinx Tools, MATLAB, and Simulink
- Facilitate exploration and implementation of DSP systems
- Useful in other data-intensive applications



MATLAB

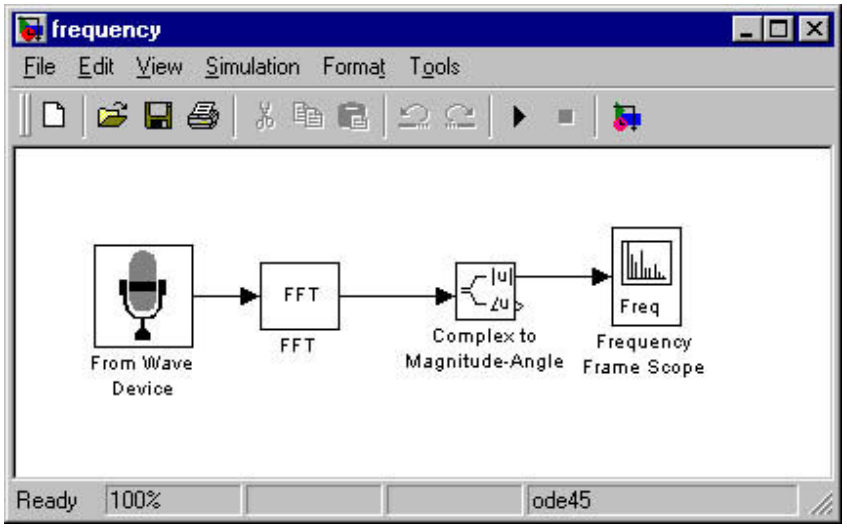
- **MATLAB provides an environment to perform mathematical modeling of complex systems**
  - **Extensive libraries for math functions, signal processing, scientific computing, etc.**
  - **Large variety of functions to plot and visualize data, systems, and designs**

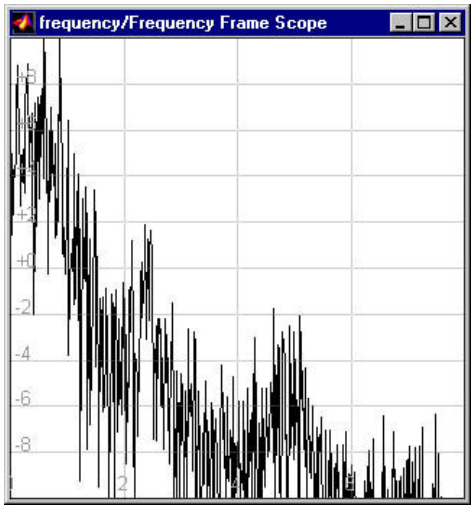




Simulink

- Simulink provides a model-based design environment
  - **Fully integrated with MATLAB**
  - **Graphical block editor**
  - **Event-driven simulator**
  - **Models parallelism**
  - **Extensive library of parameterizable functions**

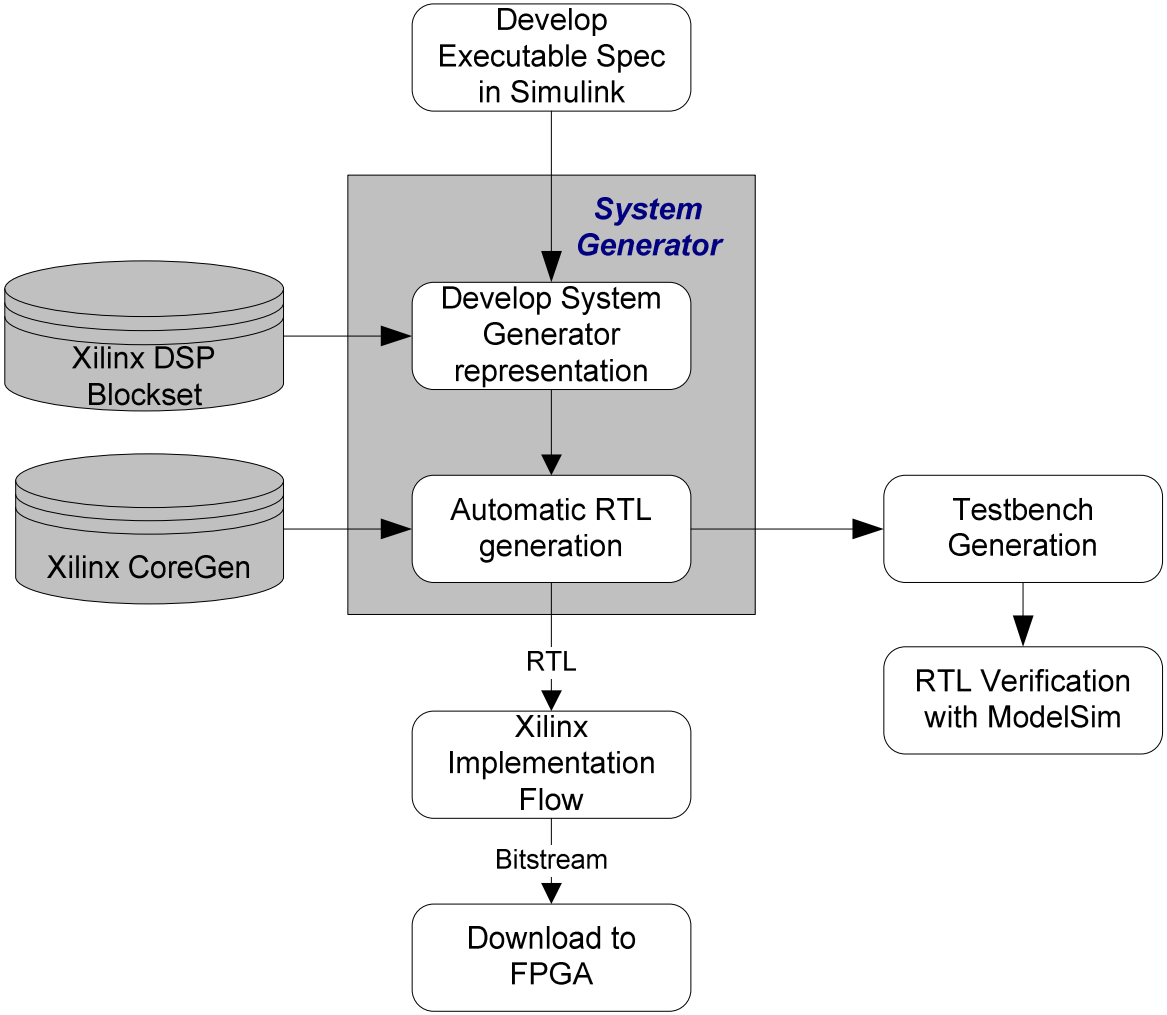




Xilinx System Generator

- **Provides a system-level design environment for Xilinx FPGAs**
  - **Enables a design flow from Simulink software to FPGA implementation. It leverages**
    - **MATLAB , Simulink**
    - **HDL synthesis, IP Core libraries**
    - **FPGA implementation tools**
- **Features include**
  - **Synthesizable VHDL/Verilog with hierarchy preservation**
  - **Automatic invocation of the CORE Generator™ software**
  - **Project generation to simplify the design flow**
  - **HDL testbench and test vector generation**
  - **Constraint file and simulation file generation**

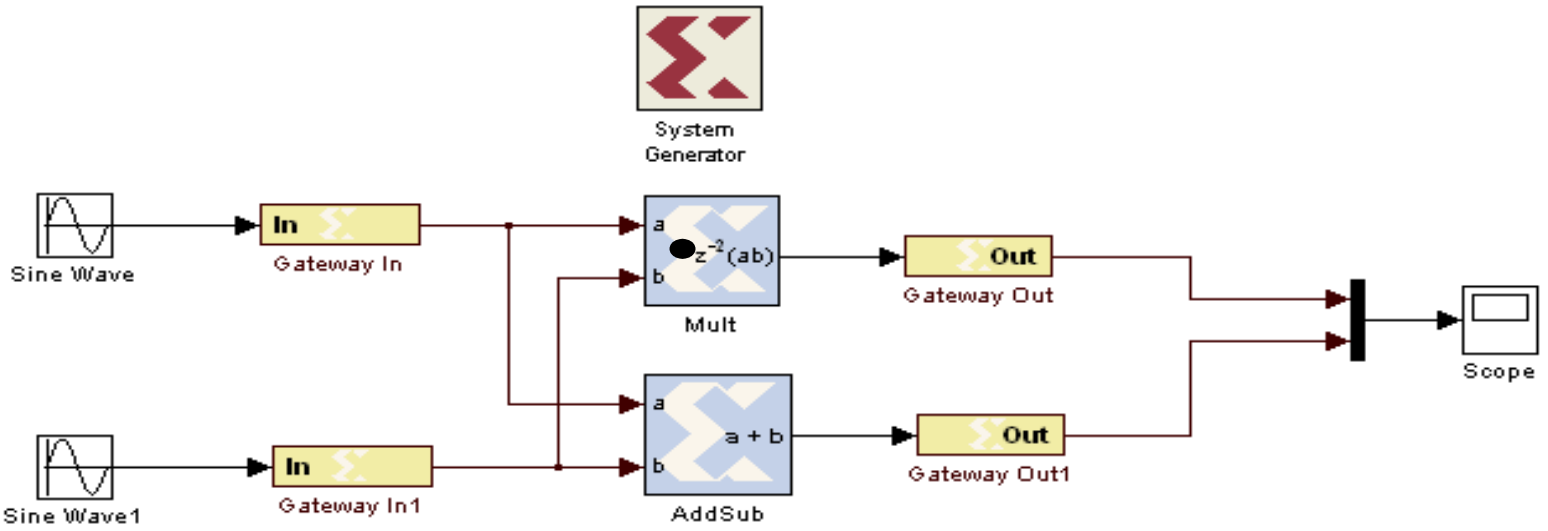
Basic SysGen Design Flow



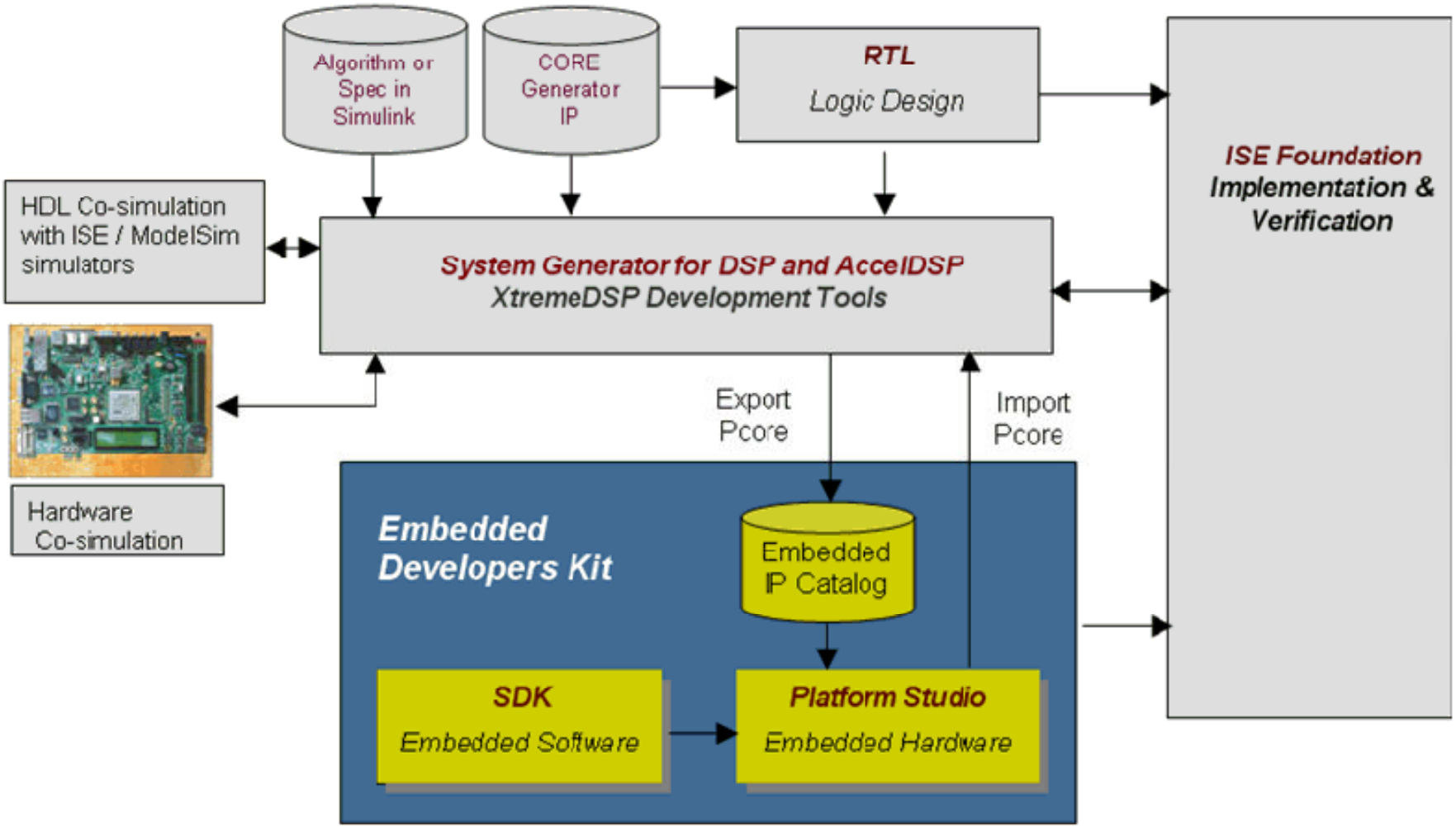
Enables designs specified using Simulink to be quickly implemented on FPGAs

System Generator DSP Blockset and the Simulink Library

- Provide a variety of arithmetic, logic operators, and DSP functions
  - Bit and cycle-accurate to FPGA implementation
  - Arbitrary precision fixed-point arithmetic including quantization and overflow
- Allow instantiation of controllers, memories, and IP blocks
- Facilitate input waveforms and output value sampling



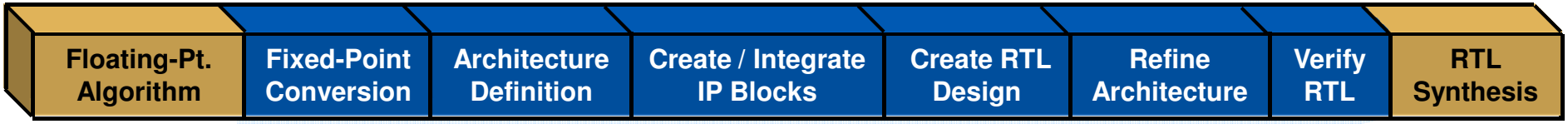
Design Integration with SysGen



- **System Generator is integrated with the rest of the Xilinx Tool Flow**

AccelDSP Design Flow

Typical Design Flow



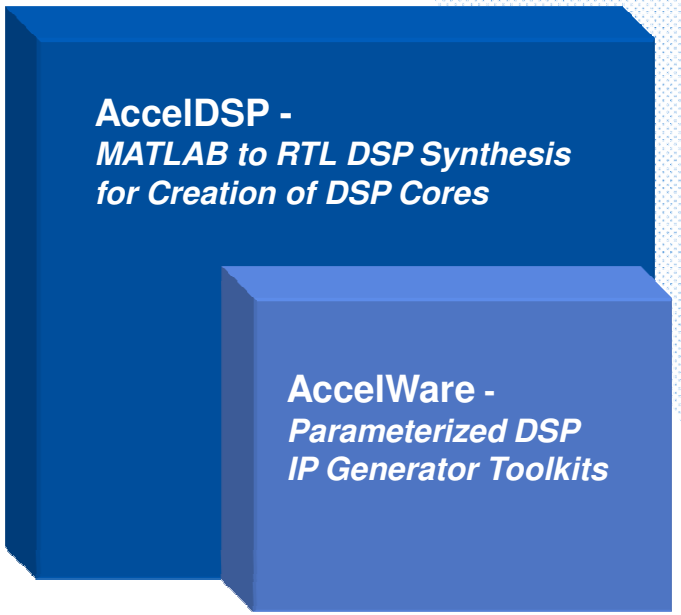
Steps performed by AccelDSP

AccelDSP Design Flow

Replaces manual steps

- Floating to fixed-point conversion
- RTL creation
- RTL and gate verification back to the original algorithm
- IP creation and integration

AccelDSP Design



AccelDSP Synthesis

MATLAB to RTL or Simulink

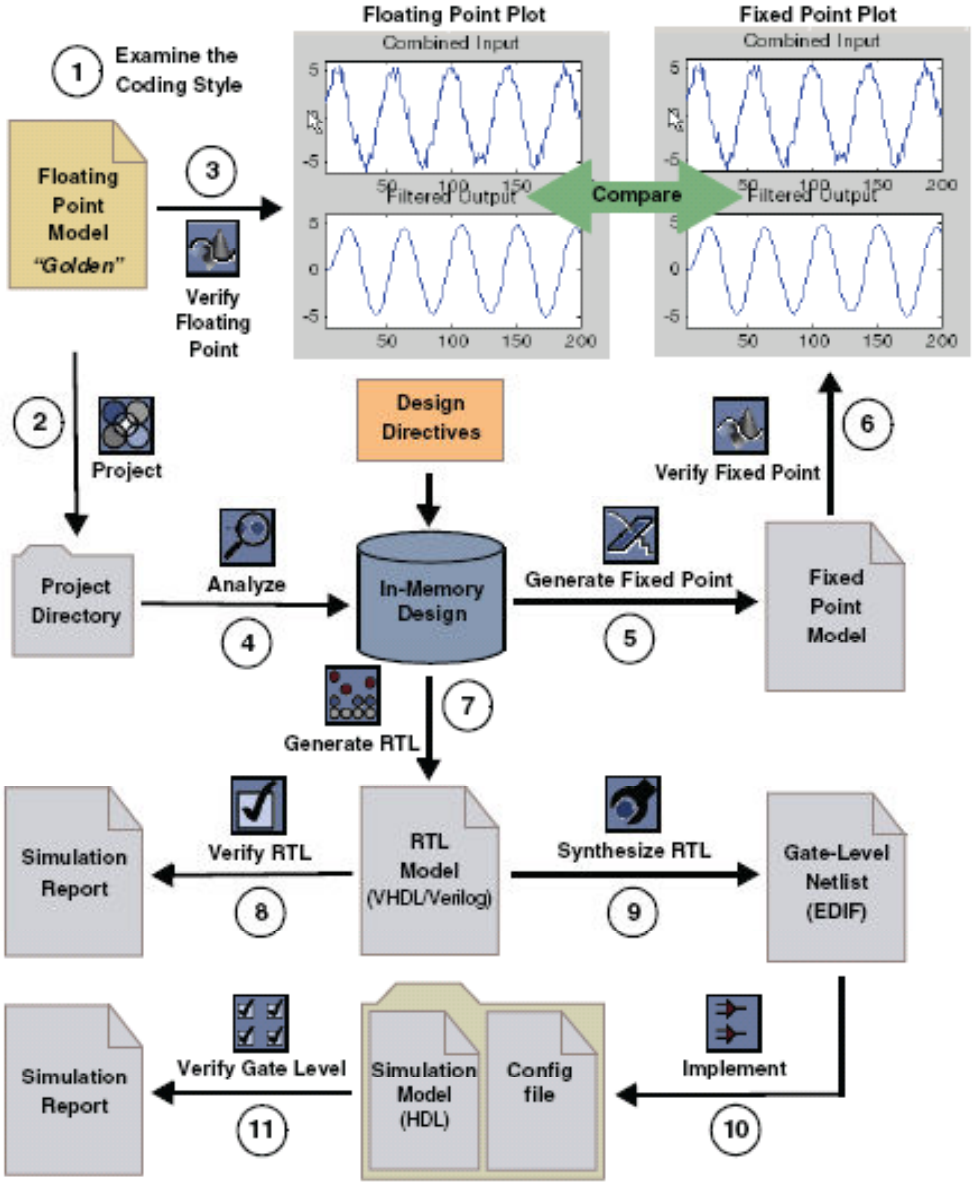
Design Inferred from MATLAB code

- **Quantization changes**
- **Loop rolling / unrolling**
- **Pipelining**
- **Device-specific memory mapping**

AccelWare IP Generators for DSP Cores

- **Signal processing**
- **Linear algebra**
- **Communications**

AccelDSP Design & Verification



- AccelDSP Synthesis automatically generates a fixed-point model from the floating-point source
  - Process is user interactive and controllable
- High level MATLAB functions and operators are automatically replaced by hardware accurate models
- Accurate estimates are made of resources and performance
  - IP-Explorer
- Fixed-point analysis features are provided to address reduced-precision arithmetic errors
  - Plots and histograms
  - Overflow and underflow reporting

Summary of Xilinx DSP Tools

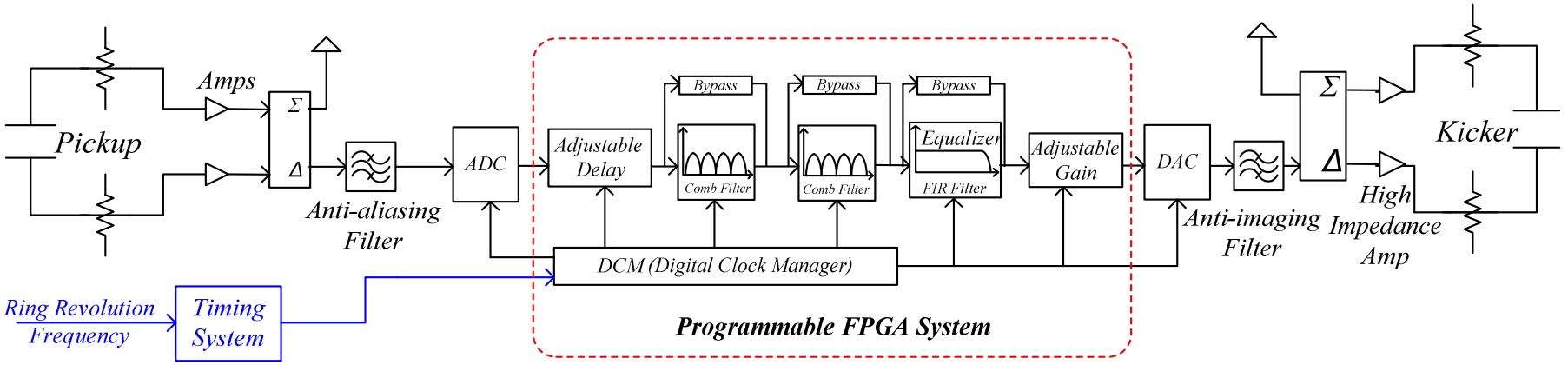
- Provide a high-level of abstraction for design
  - Developed using MATLAB and/or Simulink
- Facilitate design-space exploration
  - Can quickly examine different architectures and get resource and performance estimates
- Generate good designs in terms of both resource usage and performance
  - Especially for DSP-related tasks, but also for other data-intensive systems
- Well integrated with other Xilinx tools
- However,
  - MATLAB code has restriction and Simulink has limited capabilities
  - Need to be careful with coding style
  - Generated code can difficult to read
  - Portions of the generated code are Xilinx specific

FPGA-based Systems for Particle Physics

- **Recent FPGA architectures and tools have tremendous potential for particle physics systems**
  - **Over 10,000 FPGAs currently used in CMS electronics**
- **Can provide improved**
  - **System integration**
  - **Design space exploration**
  - **Design verification**
  - **Overall design quality**
- **Examples**
  - **Configurable feedback damper systems for the Spallation Neutron Source (SNS) at ORNL**
  - **Initial designs for upgrades to the Regional Calorimeter Trigger (RCT) for the CMS detector at CERN**

Feedback Damper Systems

- **A mixed-signal feedback damper system to control Electron-Proton (E-P) instabilities**
  - **Helps control ring oscillations due to E-P instabilities**
  - **Compensates for various source of error in the system**
  - **Includes a mixed of analog and FPGA-based hardware**
  - **Has very high-speed signal processing requirements**



Feedback Damper Systems

- **Preliminary FPGA design**
  - **Includes programmable delays, comb filters, parallel FIR filters, and gain control**
  - **Allows the system to be monitored and adapted at runtime**
  - **Implemented using VHDL and conventional Xilinx tools**
- **Later design using System Generator, AccelDSP, and ChipScope Pro had**
  - **Reduced resource utilization (roughly 25% reduction)**
  - **Improved performance (increase clock frequency by 20%)**
  - **Greater opportunities for design space exploration**
    - **Degree of parallelism, number of taps in FIR filter**
    - **Changes to arithmetic precision and rounding**
  - **Improved debug capabilities**

Regional Calorimeter Trigger

- **Investigating FPGA architectures, tools, and techniques to help upgrade the RCT of the CMS detector at CERN**
- **Original RCT implemented using ASICs**
- **Upgraded RCT to be implemented using FPGAs with new algorithms for increased beam luminosity**
- **Desired features**
  - **Greatly reduced component counts**
  - **Ability to quickly explore design alternatives**
  - **Rigorous testing and verification infrastructure**

Regional Calorimeter Trigger

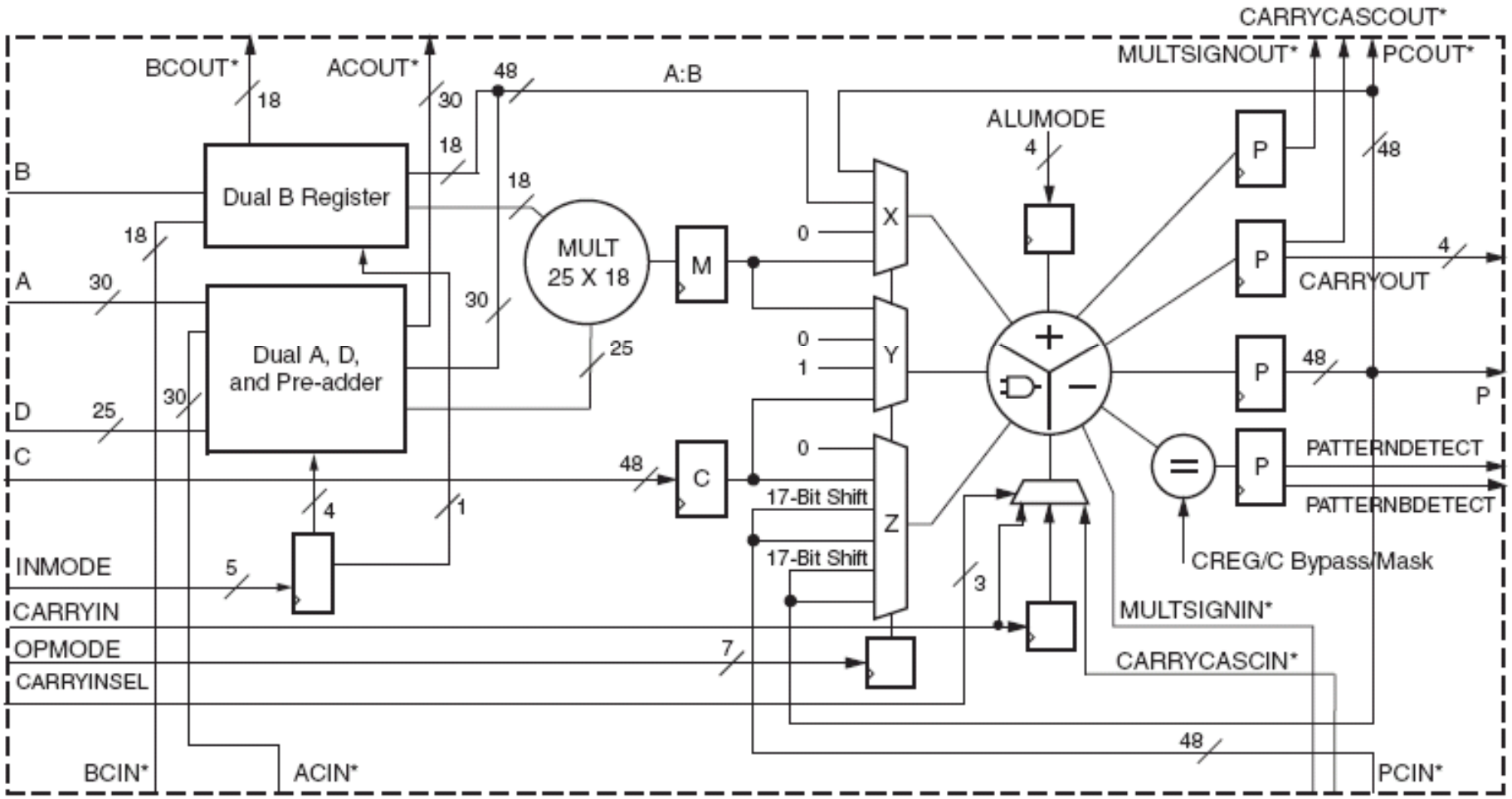
- **Investigating several tools and techniques including**
  - **The DICE environment for testing and verification**
  - **Dataflow languages (DIF and OpenDF) for design specification, validation, and implementation**
  - **Several Xilinx tools for design specification, implementation, debugging, and verification**
  - **Open source tools for version control (SVN) and firmware documentation (doxygen and doxverilog)**
  - **Modular and parameterized designs to facilitate design space exploration and algorithm changes**
- **Considerable work is still needed, but initial results are promising**

Conclusions

- **Recent FPGAs provide tremendous processing and communication resources**
  - **Future FPGAs will provide even more**
- **Recent tools help manage the design, test, and integration of complex FPGA-based systems**
  - **Additional tools are needed to cope with increasing design and verification complexity**
- **Recent FPGAs and tools can provide substantial benefits to experiments for high-energy physics**
  - **Further investigation and collaboration is needed to realize their full potential**

Backup Slides

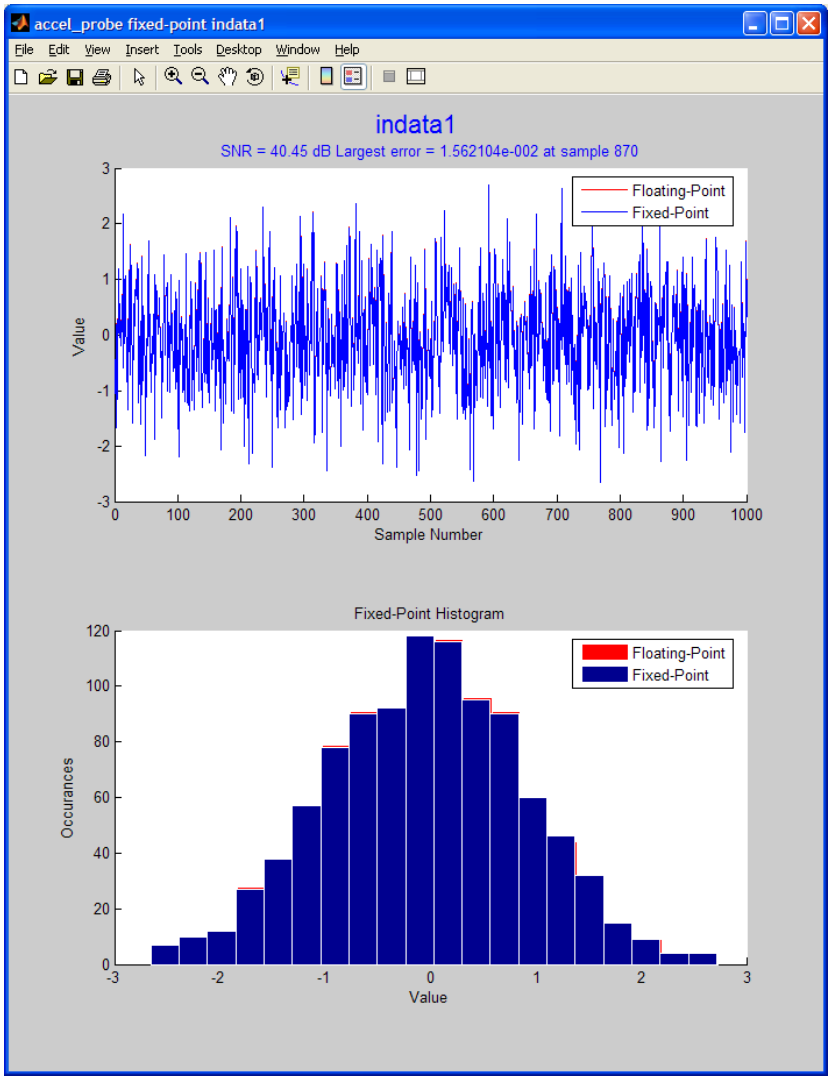
DSP48E1


CARRYCASCOUT*
MULTSIGNOUT*
PCOUT*
BCOUT*
ACOUT*
A:B
ALUMODE
B
Dual B Register
MULT
25 X 18
M
X
A
Dual A, D,
and Pre-adder
Y
D
C
C
17-Bit Shift
Z
P
CARRYOUT
P
PATTERNDETECT
PATTERNBDETECT
CREG/C Bypass/Mask
INMODE
CARRYIN
OPMODE
CARRYINSEL
MULTSIGNIN*
CARRYCASCIN*
BCIN*
ACIN*
PCIN*

Floating-Point to Fixed-Point Conversion and Verification

- AccelDSP Synthesis automatically generates a fixed-point model from the floating-point source
  - Process is user interactive and controllable
- High level MATLAB functions and operators are automatically replaced by hardware accurate models
- Accurate estimates are made of resources and performance
  - IP-Explorer
- Fixed-point analysis features are provided to address reduced-precision arithmetic errors
  - Plots and histograms
  - Overflow and underflow reporting

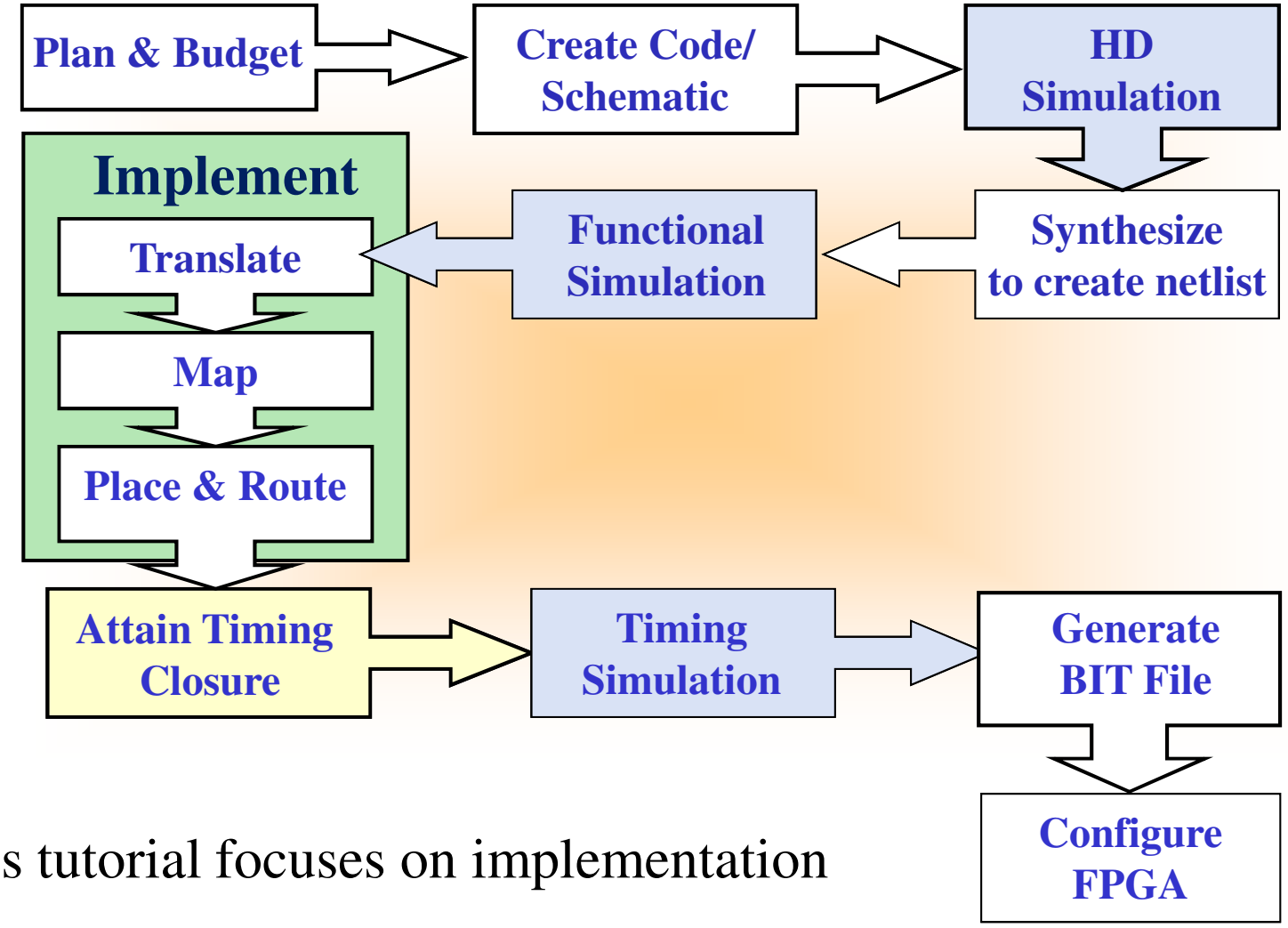
AccelProbe



Xilinx FPGA Tools

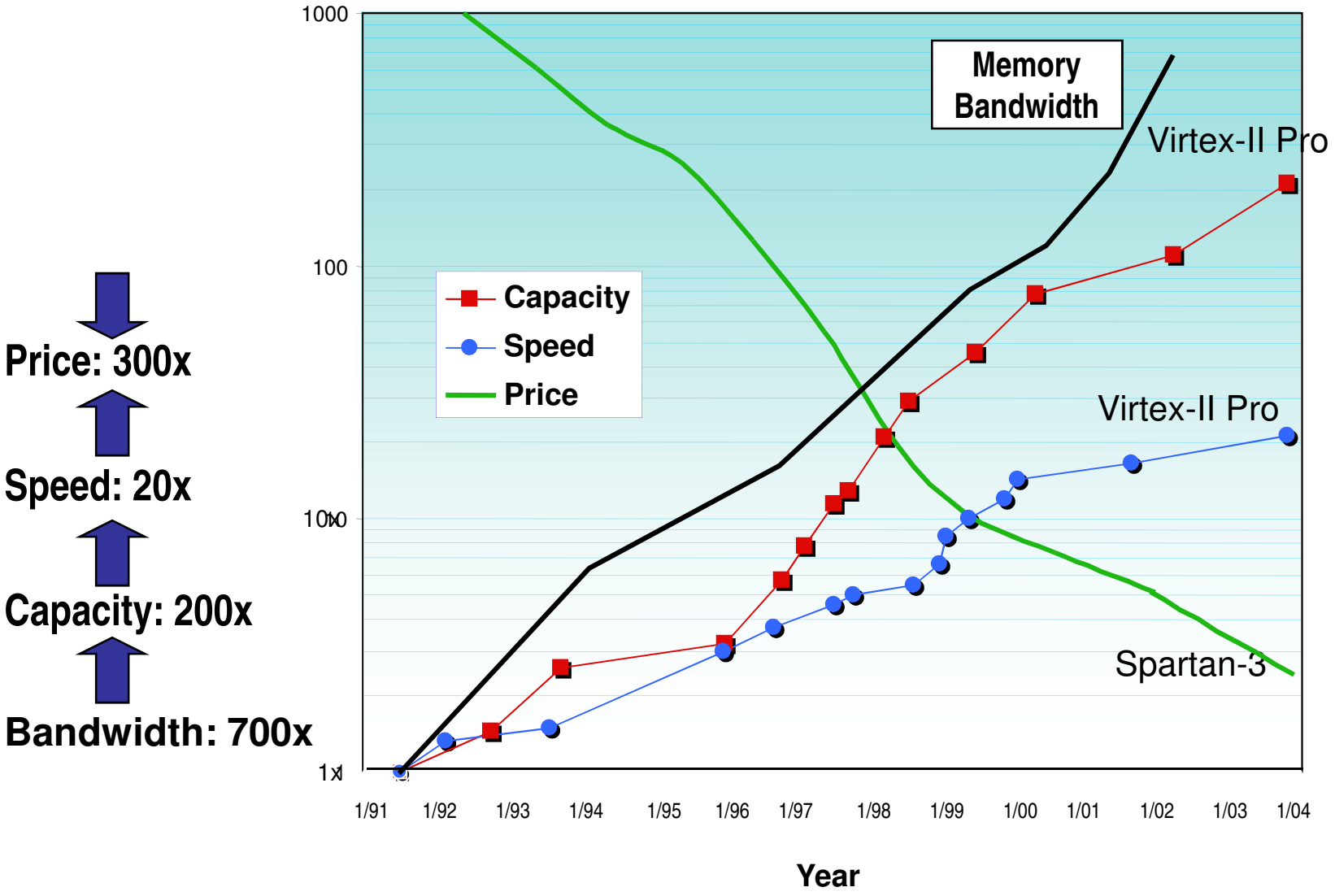
- ISE Foundation
  - Enter, simulate, synthesize and implement designs
- PlanAhead Design and Analysis Tool
  - Perform I/O pin planning, design analysis, floorplanning, and design-space exploration
- ChipScope Pro
  - Debug and validation post-implementation designs
- Embedded Development Kit (EDK)
  - Generate complete embedded processor systems
- System Generator for DSP
  - Design and analyze DSP systems
- See http://www.xilinx.com/tools/system.htm

FPGA Design Flow



This tutorial focuses on implementation

FPGA Trends


Price: 300x
Speed: 20x
Capacity: 200x
Bandwidth: 700x
1000
100
10
1
Memory Bandwidth
Virtex-II Pro
Capacity
Speed
Price
Virtex-II Pro
Spartan-3
1/91
1/92
1/93
1/94
1/95
1/96
1/97
1/98
1/99
1/00
1/01
1/02
1/03
1/04
Year