Xilinx Introduction

2008

Introducing Xilinx

- **Leader in one of the fastest growing semiconductor segments**
  - Invented programmable chip in 1984
  - 7,500+ customers; 50,000 design starts/year
- **Leader in semiconductor process technologies**
  - First to 180nm, 150nm, 130nm, 90nm
  - First to 65nm with Virtex™-5 in 2006
- **Enable hardware to "change its spots"**
  - Fastest time-to-market
  - Field upgradeability
  - Real-time reconfiguration
- **Pioneer of fabless semiconductor model**
  - Focus on design, marketing, support
  - Partner for everything else
- **A well-managed company and a great place to work**

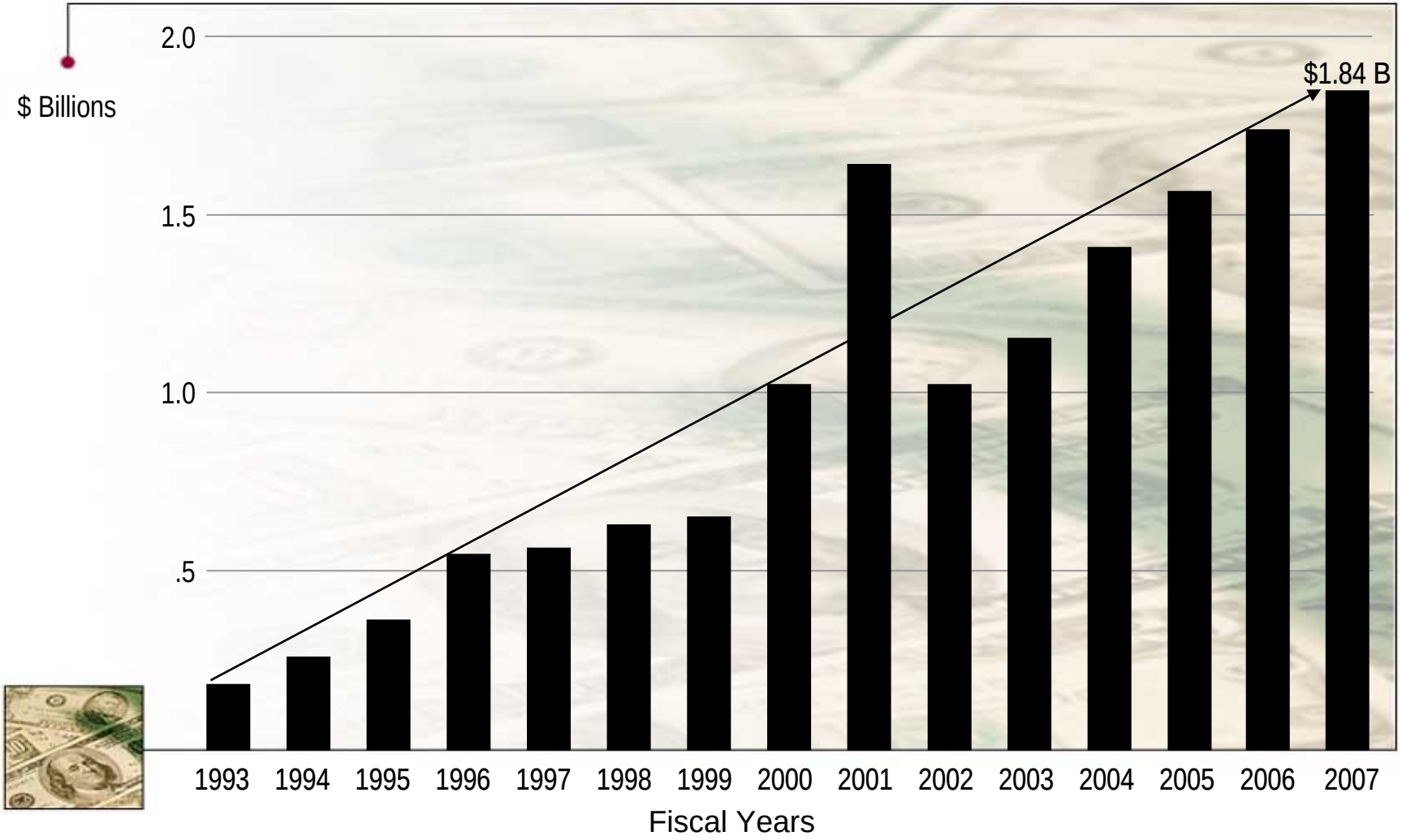
Programmability Is Mainstream

ASIC/PLD Vendor Rankings

	2006	2005	2004	2003	2002	2001	2000	1999
1.	XILINX	IBM	IBM	IBM	IBM	IBM	IBM	IBM
2.	IBM	XILINX	XILINX	NEC	NEC	Agere	Lucent	Lucent
3.	NEC	Fujitsu	NEC	XILINX	Agere	LSI Logic	LSI Logic	NEC
4.	Altera	NEC	Fujitsu	Fujitsu	XILINX	XILINX	NEC	LSI Logic
5.	Fujitsu	Altera	Altera	Agere	Fujitsu	NEC	XILINX	Fujitsu
6.	Toshiba	Toshiba	Renesas	Toshiba	LSI Logic	Altera	Fujitsu	XILINX

Source: Gartner Dataquest (1998-2001 rankings), iSuppli (2002-2006 rankings)
Note: Lucent spun-off their semiconductor division in 2001 creating Agere Systems

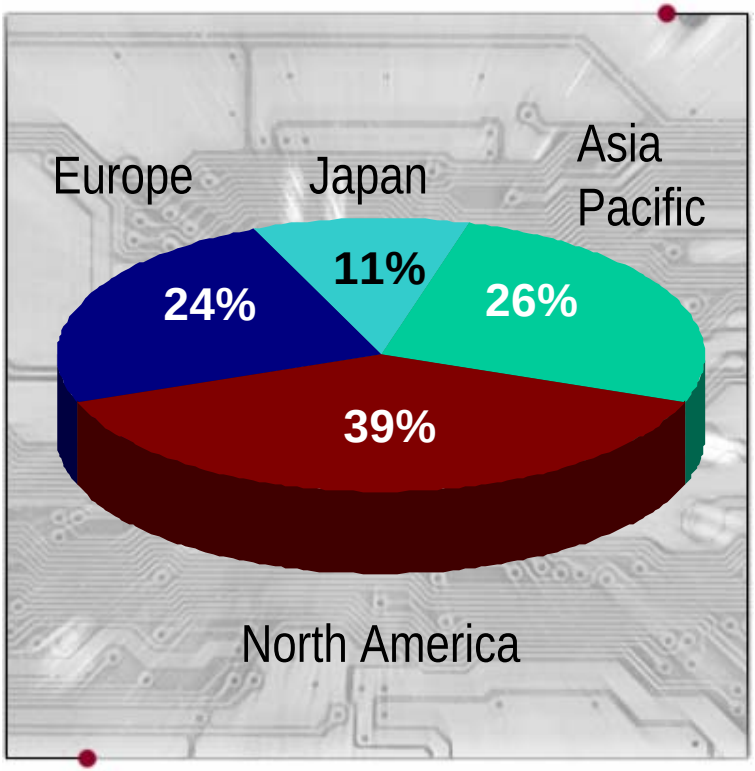
Xilinx Revenue



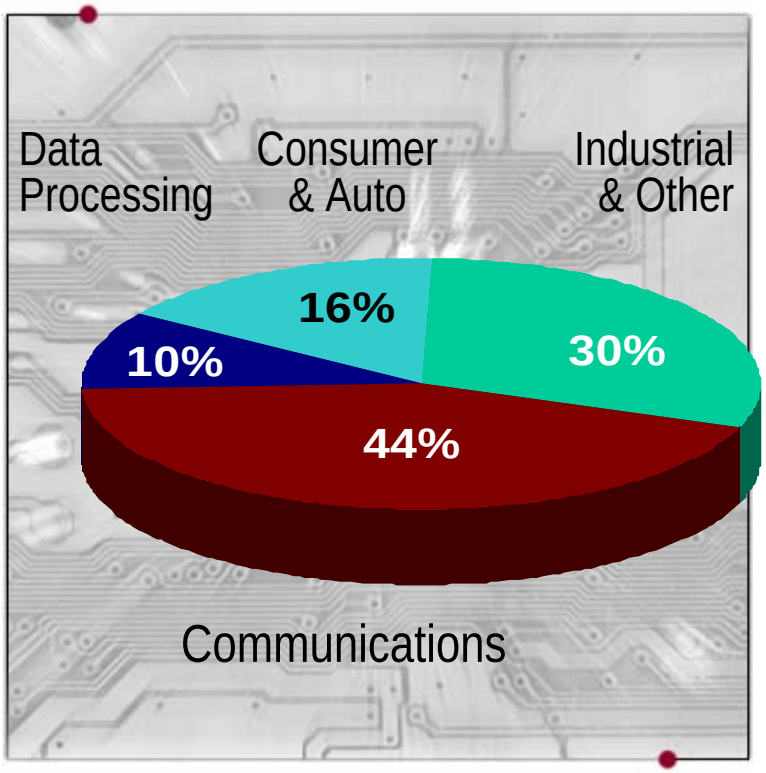
Xilinx Revenue Breakdown

Q1 Calendar Year 2007

Revenue by Geography



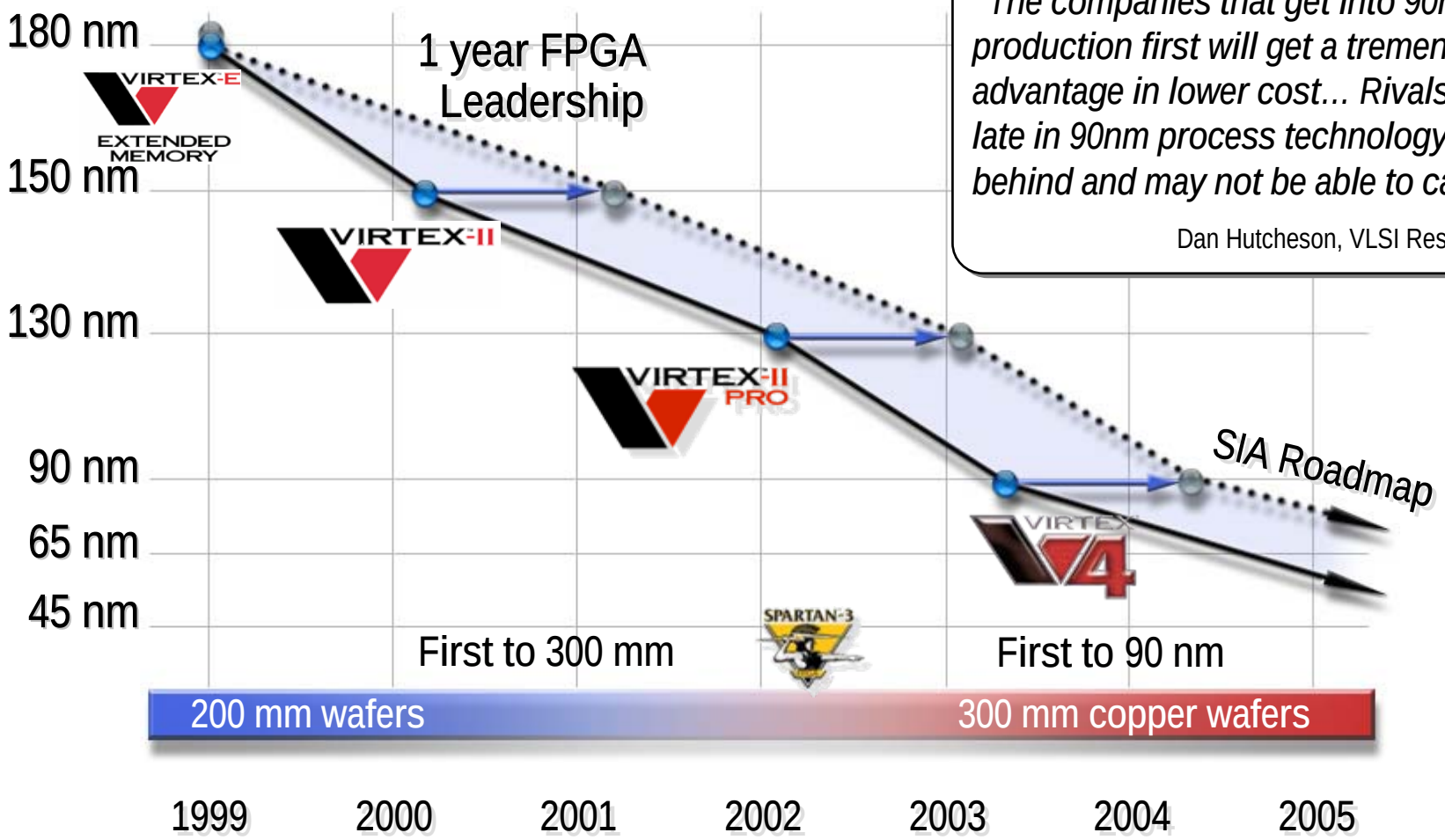
Revenue by End Market



Source: Xilinx, Inc.

World Class Process Technology

Enabled through IBM and UMC Partnerships

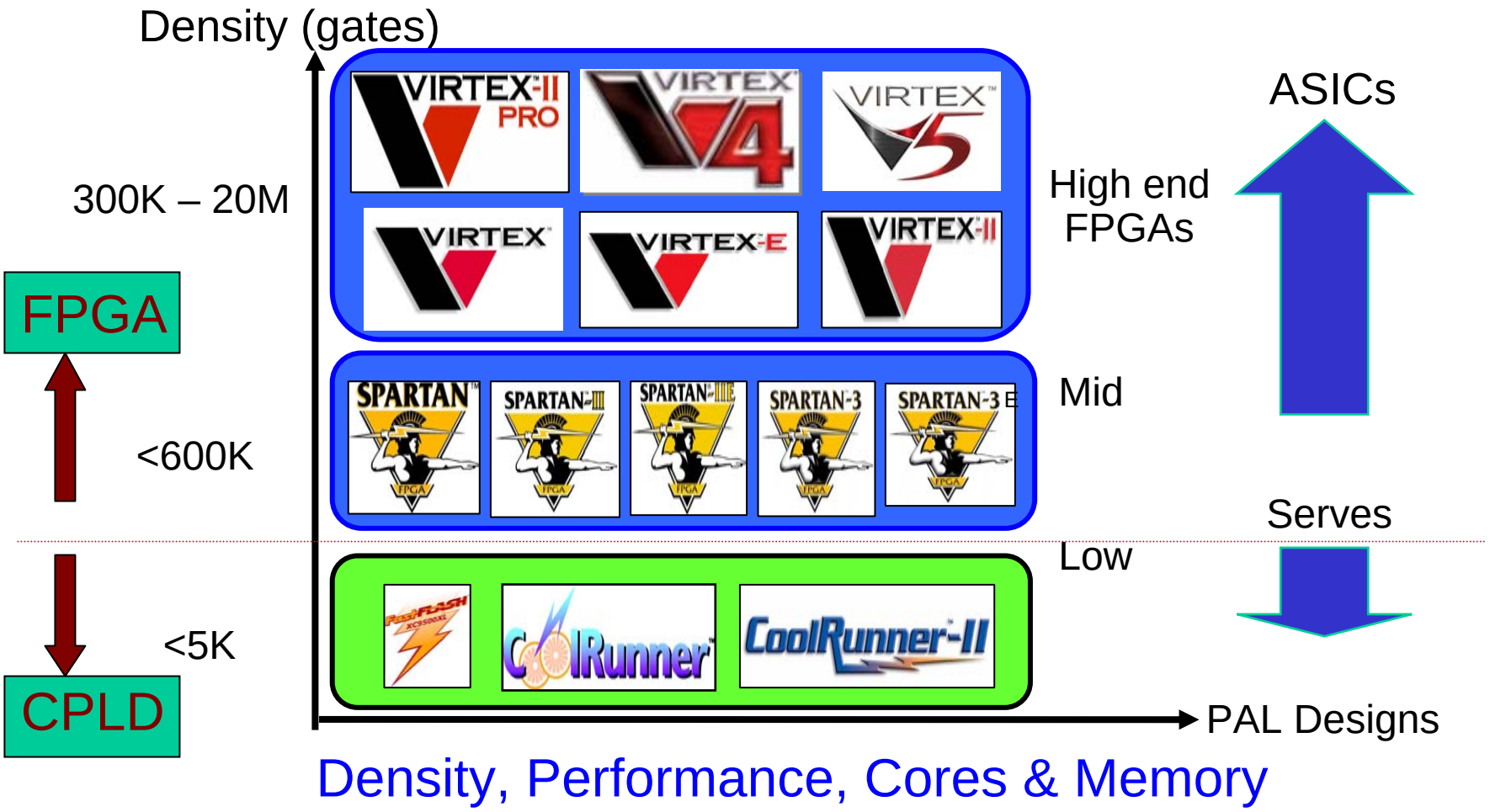


"The companies that get into 90nm production first will get a tremendous advantage in lower cost… Rivals who are late in 90nm process technology will fall behind and may not be able to catch up."

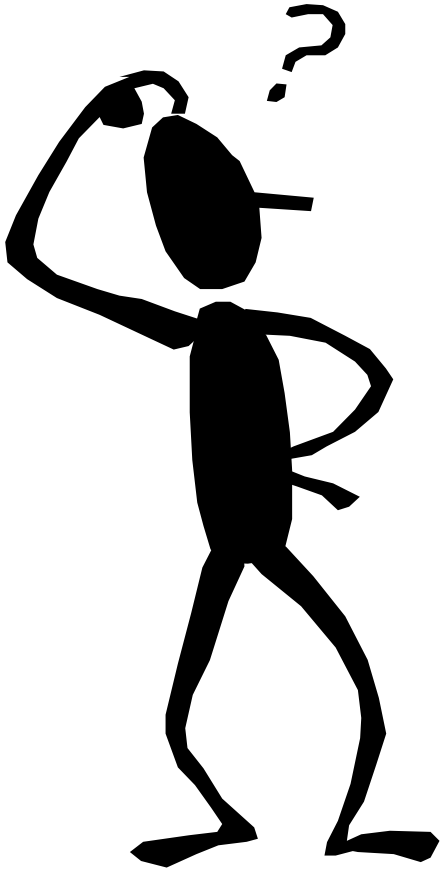
Dan Hutcheson, VLSI Research, Inc.

E-ELEMENTS

Full xilinx Design



How to select a device?



- What's your need?
  - Density ?
  - Style ?
  - I/O ?
  - Performance ? x
  - Power ?
  - Configuration ?
  - Cost ?
- Device's architecture & features?

CPLD or FPGA?

- CPLD
  - Non-volatile
  - Consistent pin-to-pin timing
  - Simple timing model
  - Very low power consumption
  - Lowest cost point
  - Fast internal performance
  - Small packages (QF, CP)
  - Applications: Logic decode and integration, state machines, or standard bus interfaces (SPI, I2C, or SMBus, for example)

- FPGA
  - Volatile
  - Needs a memory device to load design at power up
  - Complex timing model
  - Larger, more complex designs
  - Memory resources
  - Applications: PCI, high-speed serial communication, or embedded processors

Basic FPGA Architecture

Slice and I/O Resources

Memory and Clocking Resources

E-ELEMENTS

Outline

- Overview
- Slice Resources
- I/O Resources
- Block RAM and FIFO
- Clocking
- Summary

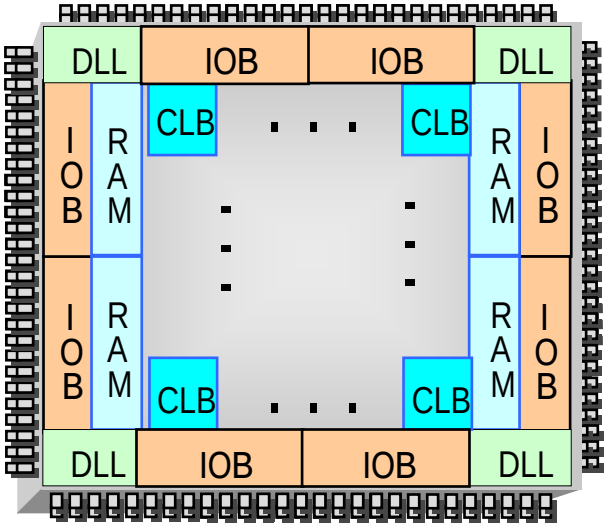
Spartan-II/IIE , Virtex/VirtexE

(1) Logic & Routing

Flexible logic implementation
Vector Based Routing
Internal 3-State bussing

(2) System Interface

SelectI/O+™ Technology
Support major I/O standards



(3)System Memory

Block Memory(4k bit)
Distributed Memory
External Memory

(4) System Clock Management

Digital Delay Lock Loops (DLLs)

ISE ALL THE SPEED YOU NEED

LogiCORE Alliance CORE

IP Solutions

Xilinx Global Services

E-ELEMENTS

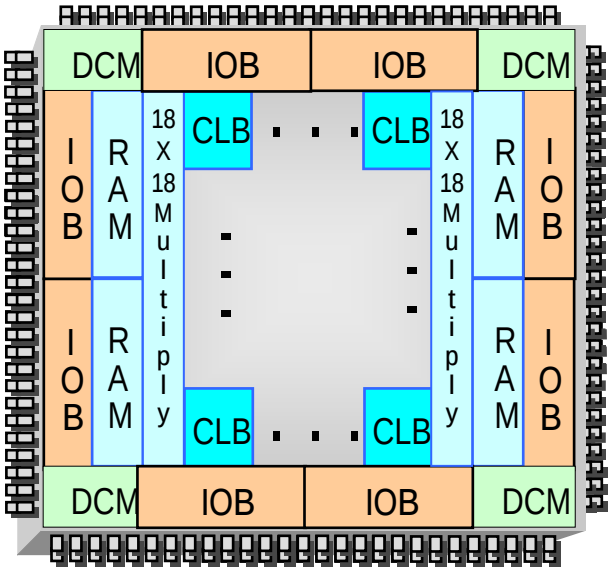
Spartan-3 ,VirtexII FPGA

(1) Logic & Routing

Low cost CLB
Wider Input Functionality
Vector Based Routing

(2) System Interface

Higher I/O at lower cost
Digitally Controlled Impedance
Built in DDR Registers



(4) System Clock Management

Digital Clock Management (DCMs)

(3) System Memory

Block Memory (18Kbit)
Distributed Memory
External Memory

(5) Embedded Multiplier

Software

IP Solutions

Xilinx Global Services

Global Support & Services

E-ELEMENTS

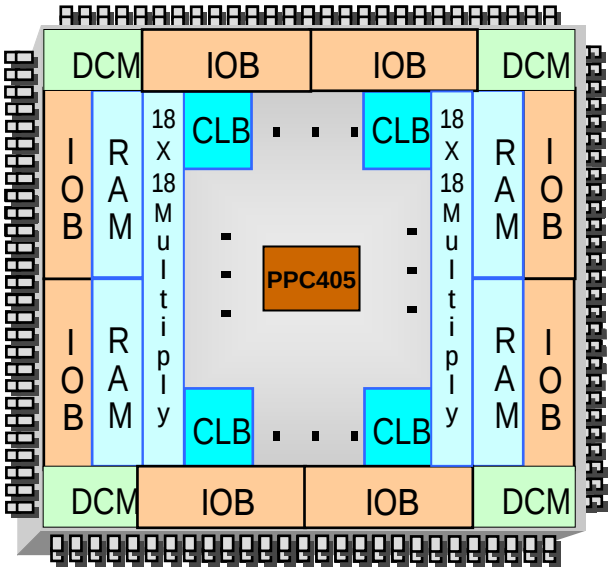
VirtexII-Pro FPGA

(1) Logic & Routing

Low cost CLB
Wider Input Functionality
Vector Based Routing

(2) System Interface

Higher I/O at lower cost
Digitally Controlled Impedance
Built in DDR Registers



(3) System Memory

Block Memory (18Kbit)
Distributed Memory
External Memory

(4) System Clock Management

Digital Clock Management (DCMs)

(5) Embedded Multiplier

(7) Rocket IO (3.125 G)

(6) Embedded Processor

ISE ALL THE SPEED YOU NEED

Software

LogiCORE Alliance CORE

IP Solutions

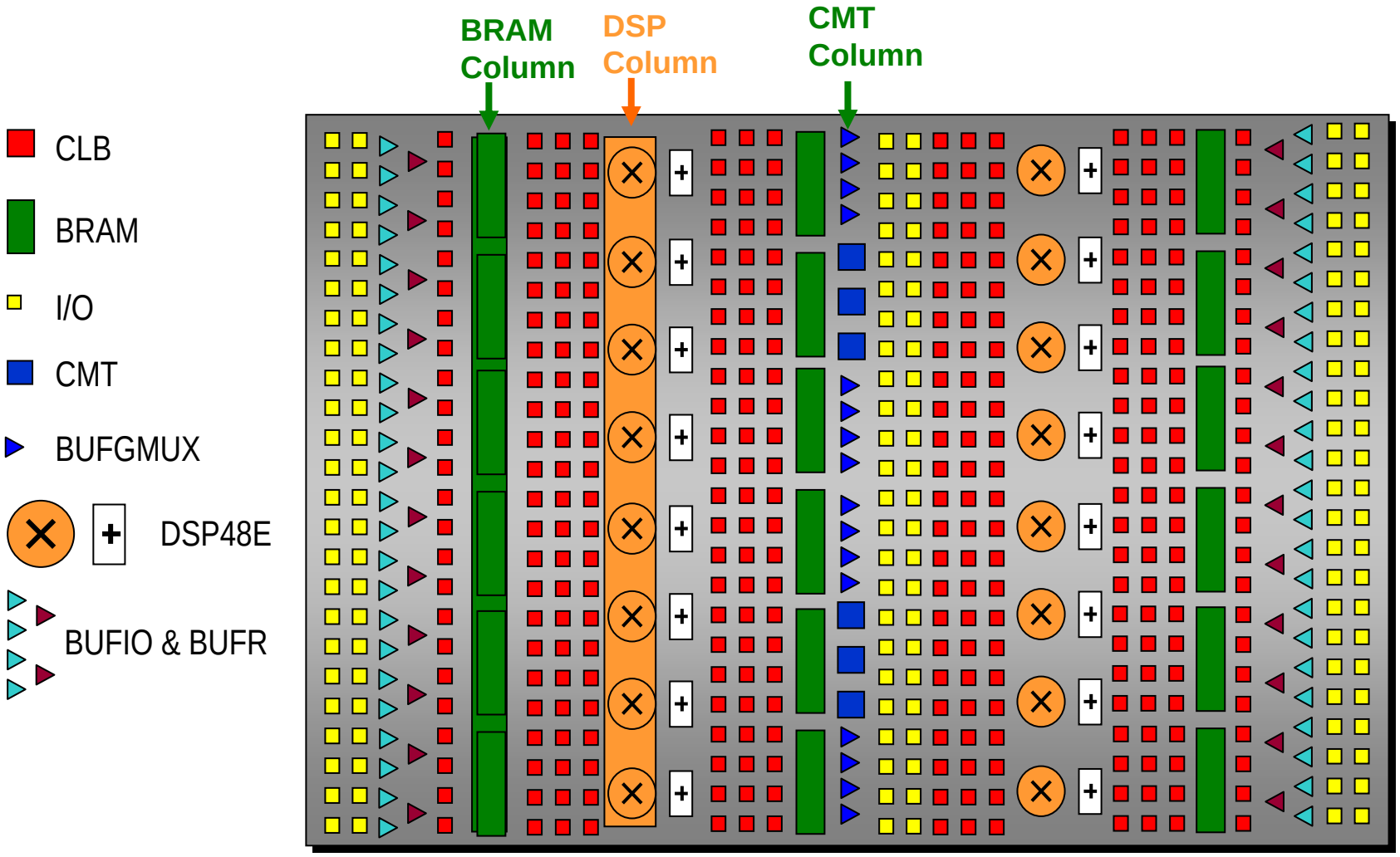
Xilinx Global Services

Global Support & Services

E-ELEMENTS

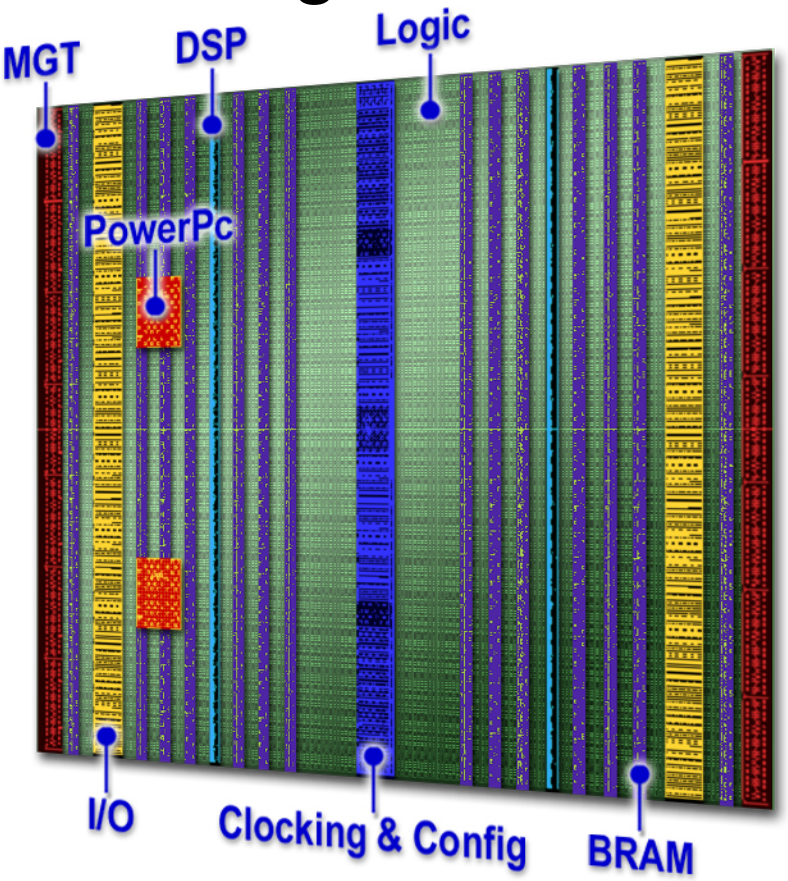
Virtex-5 FPGA Platform

Feature Overview



Virtex-4/5 is based on Columnar Architecture Domain Optimized Logic

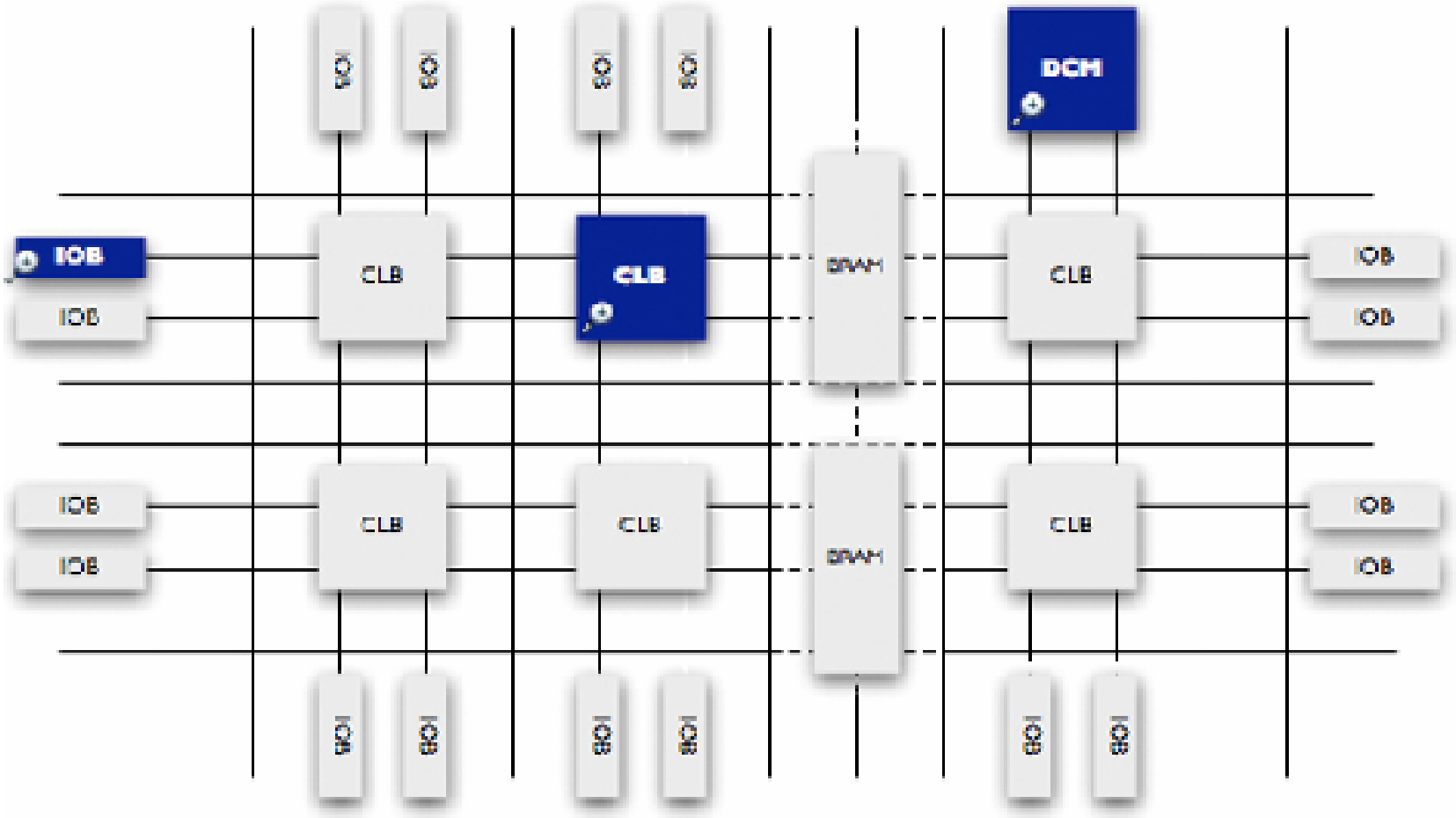
- Revolutionary Advance in FPGA Architecture
- Enables “Dial-In” Resource Allocation Mix
  - Logic
  - DSP
  - RAM
  - I/O
  - MGT
  - DCM
  - PowerPC®



- Enabled by Flip-Chip Packaging Technology
  - I/O Columns Distributed Throughout the Device
  - Distributed IO improves signal integrity and PWR / GND distribution

http://www.xilinx.com/virtex4

Xilinx FPGA Fabric



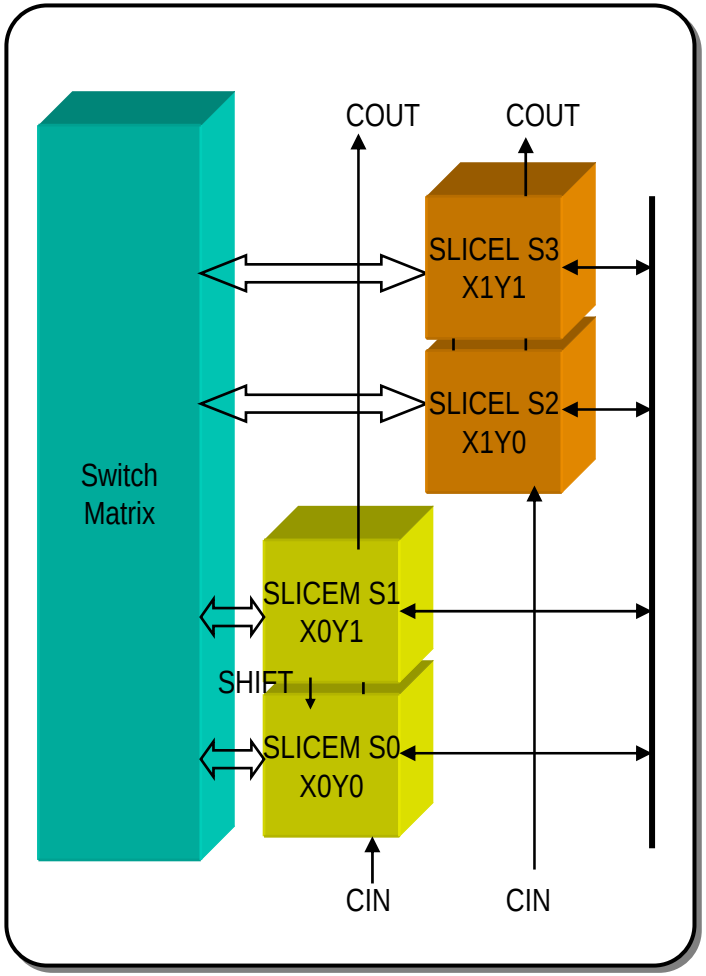
E-ELEMENTS

Outline

- Overview
- Slice Resources
- clocking
- I/O Resources
- Block RAM and FIFO
- Clocking
- Summary

Configurable Logic Block (CLB)

Flexible Logic Building Blocks

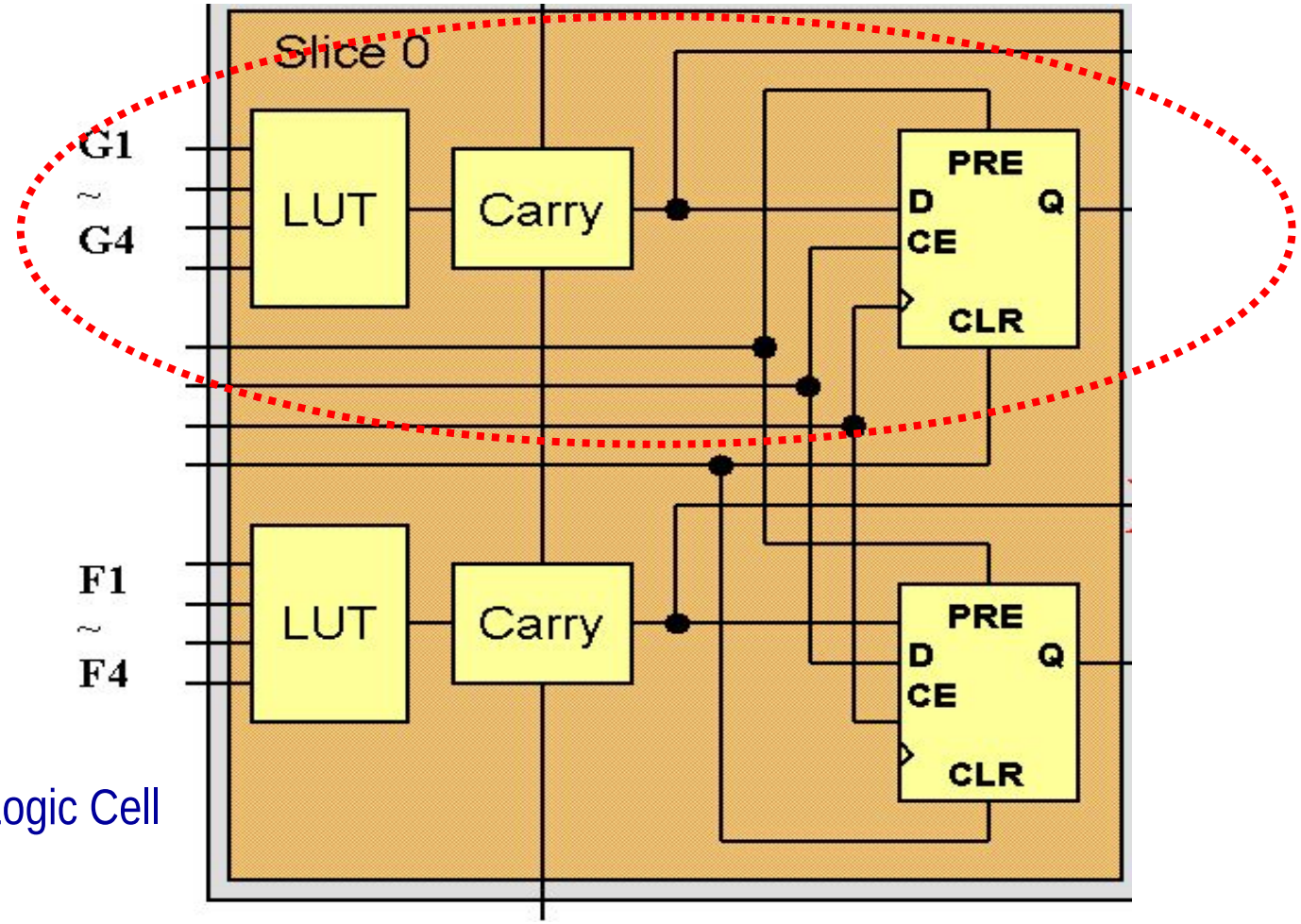


- 4 slices per CLB
  - 2 SLICEM contain Memory
  - 2 SLICEL are Logic only
- Wide-input functions
  - 16:1 multiplexer in 1 CLB
  - 32:1 multiplexer in 2 CLBs, 1 level
- Fast arithmetic functions
  - 2 look-ahead carry chains per CLB column
- Four 16-bit addressable shift registers
  - Cascadable
- General routing via switch matrix
  - Plus fast direct routing

Simplified Slice Structure

- 4 slices in each CLB

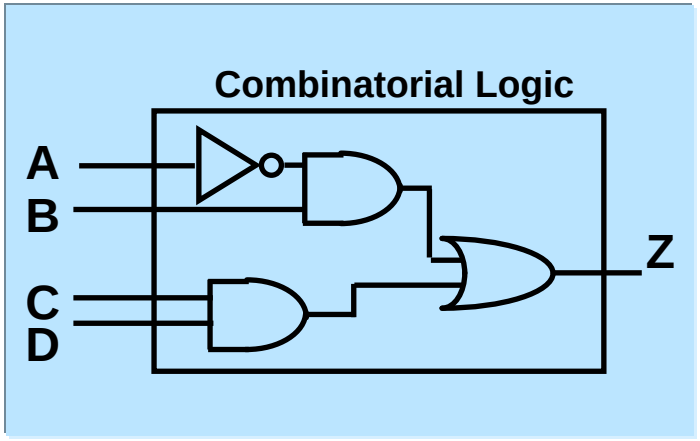
Logic Cell



1 Slice = 2.3 Logic Cell

Look-Up Tables

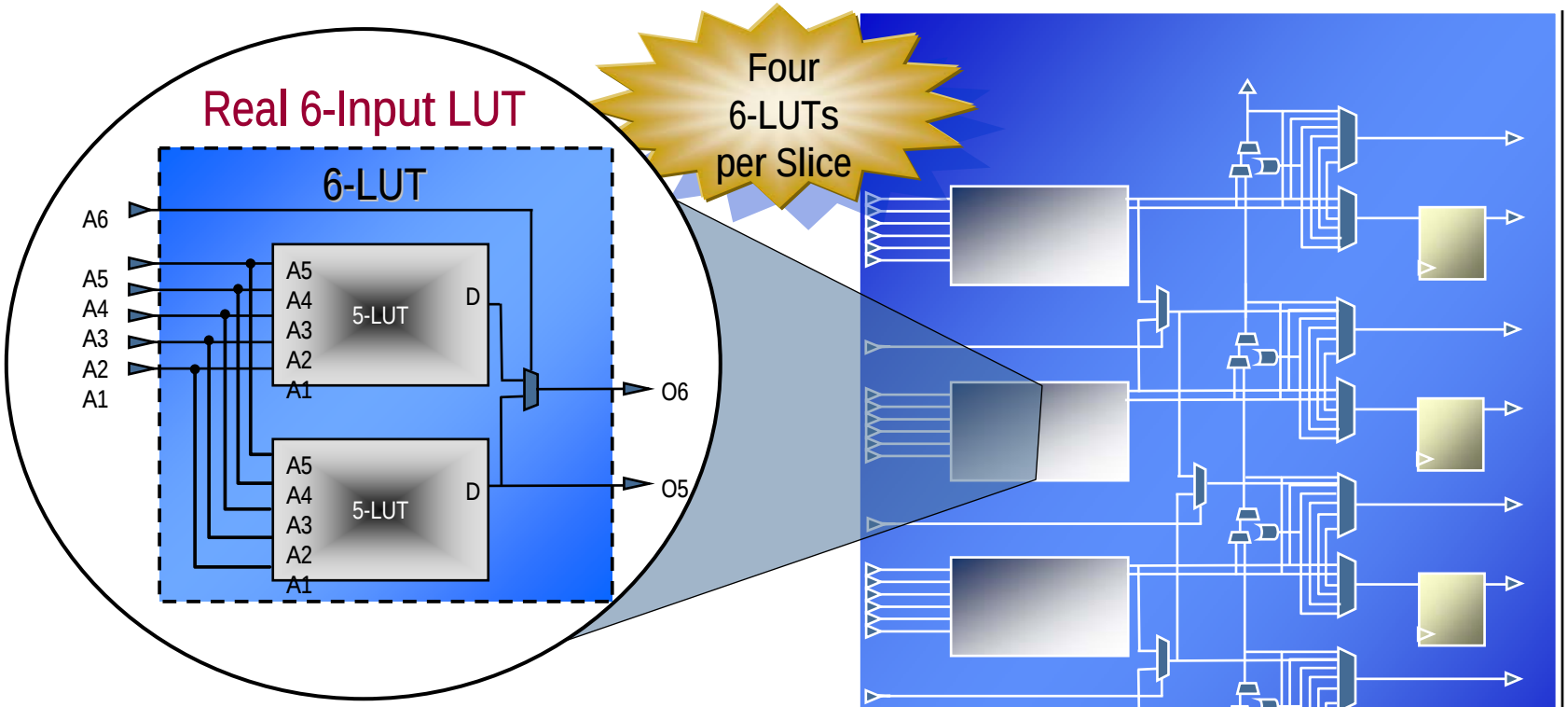
- Combinatorial logic is stored in Look-Up Tables (LUTs)
  - Also called Function Generators (FGs)
  - Capacity is limited by number of inputs, not complexity
- Delay through the LUT is constant



A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
	.	.	.	
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

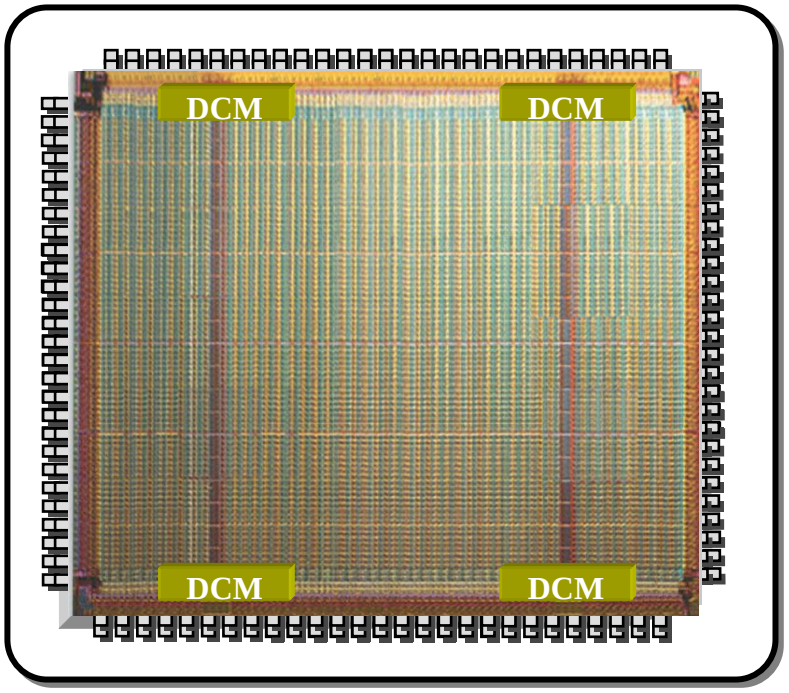
Real 6-Input LUT

Reduce Logic Levels and Improve Performance



- Approximately 10 percent bigger CLB (than if the Virtex-4 FPGA were in 65 nm)
- But implements approximately 35 percent more logic

Digital Clock Manager (DCM)



- 4 DCMs located at top and bottom of the Block RAM columns

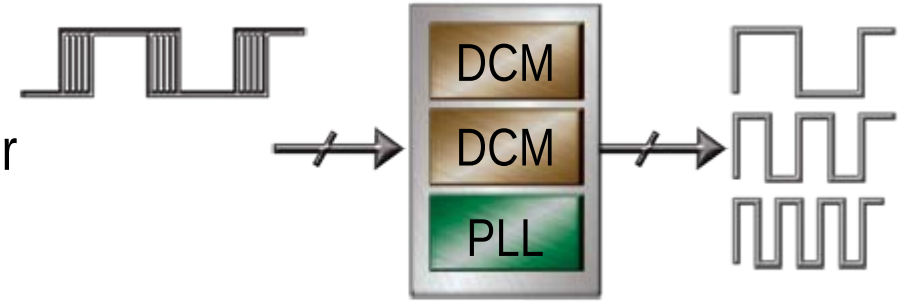
- Clock phase de-skew
- 50% Duty cycle correction
- DLL performance
  - 25Mhz to 325Mhz
  - 100ps Jitter
- Phase Shift
  - 0, 90, 180, 270
  - CLK Period/256
  - m/n clock multiply & divide
  - M= 2 to 32, D= 1 to 32
- Temperature compensation

Higher Precision 550MHz Clock Management

Design Challenge

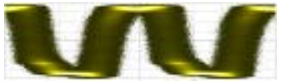
- Meeting high frequency clocking requirements with lower jitter

- Differential Global Clocks and I/O Clocks ensure low skew and jitter
- DCMs provide precise phase control for better design margins
- PLLs provide greater than 2x jitter filtering



Clock Management Tile (CMT)

PLL Input with > 400 ps jitter



PLL

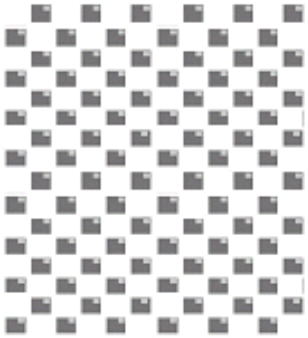
PLL Output with < 100 ps jitter

Example measurement with a 400MHz clock in a quiet XC5VLX30 device

Embedded Memory

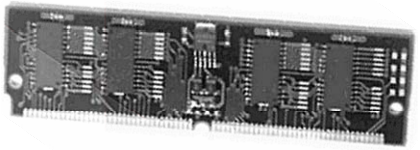
Maximizing Memory Bandwidth and Flexibility

DSP coefficients
ROM
Small FIFOs
Wide/Shallow



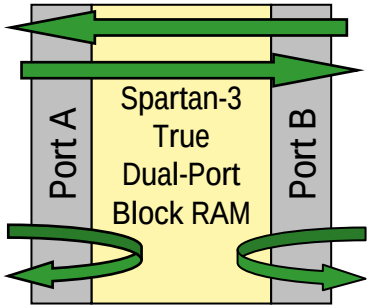
Distributed RAM
Bytes

External



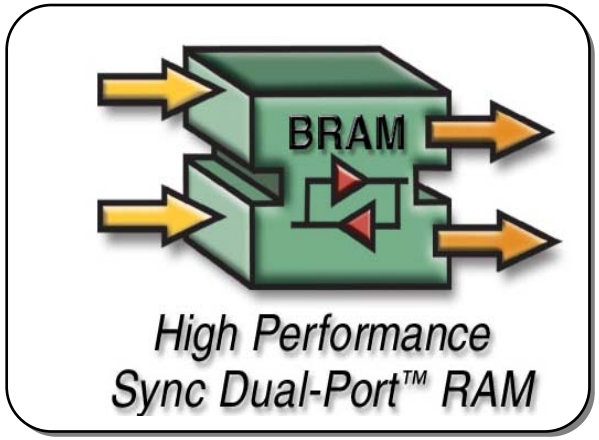
Megabytes
High-speed memory interface
DDR,QDR, FCRAM, ZBT

Cache tag memory
Large FIFOs
Large ROM
Video line buffers
Deep/Wide

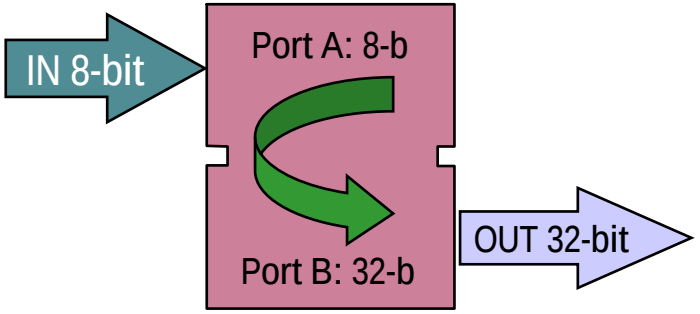


Block RAM
Kilobytes

Embedded Block RAM



- Synchronous Operation
- Independent port A and B configuration
- Support for data width conversion including parity bits
- 2 ns Read Access

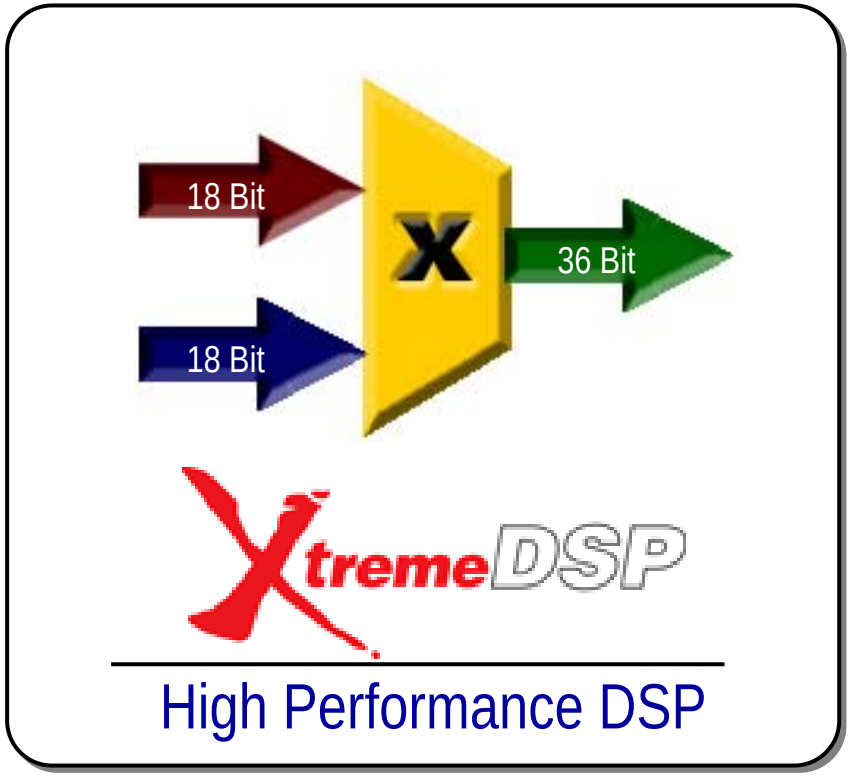


Configuration	Depth	Data bits	Parity bits
16K x 1	16Kb	1	0
8K x 2	8Kb	2	0
4K x 4	4Kb	4	0
2K x 9	2Kb	8	1
1K x 18	1Kb	16	2
512 x 36	512	32	4

Configurations available on each port

Embedded Multipliers

High Performance 18-bit x 18-bit Multipliers



- Flexibility
  - From 12 to 104 ***embedded*** multipliers
  - 18 x 18 bit Signed or 17 x 17 bit Unsigned operation
  - 2's complement signed operation
  - 4 to 18 bit operands
  - Combinational & pipelined options
- Enabling Real-time DSP Processing
  - Build up to 104 18 x 18 MACs
  - Up to 26 complex 18 x 18 multiplies
- 1:1 association with block RAM

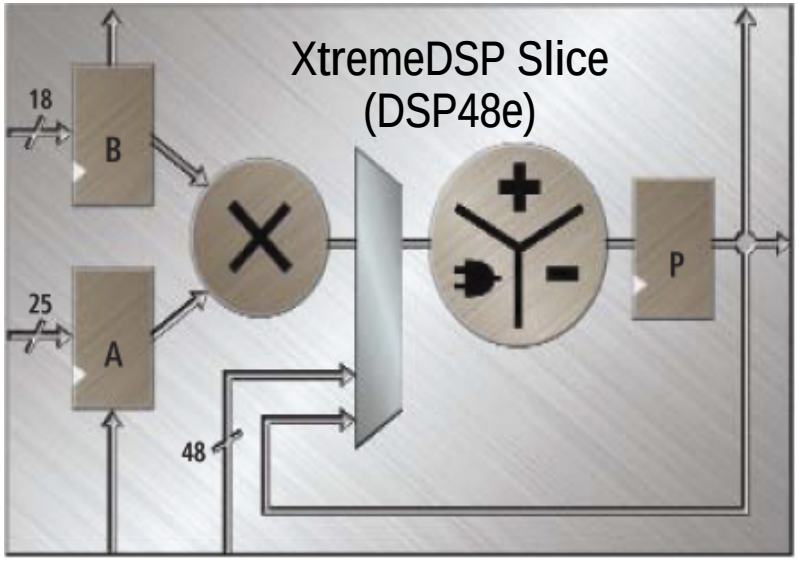
- Click here to learn more on the Xilinx DSP Solution -

High Precision 550 MHz DSP48E Slice

Design Challenge

- Need wider multipliers for higher precision floating point applications
- Lower power consumption for high performance designs

- Up to 175% bandwidth increase over Virtex-4
  - Multiplier width increased to 25x18 bits
  - Fully cascadable for adder chain architectures
  - Build wider filters with fewer slices, lower cost
- 40% power reduction over Virtex-4
  - 1.38mW/100MHz at a 38% toggle rate
  - 1.46W for 192 slices @ 550MHz
    - ➔ 105 GMACCs/s bandwidth

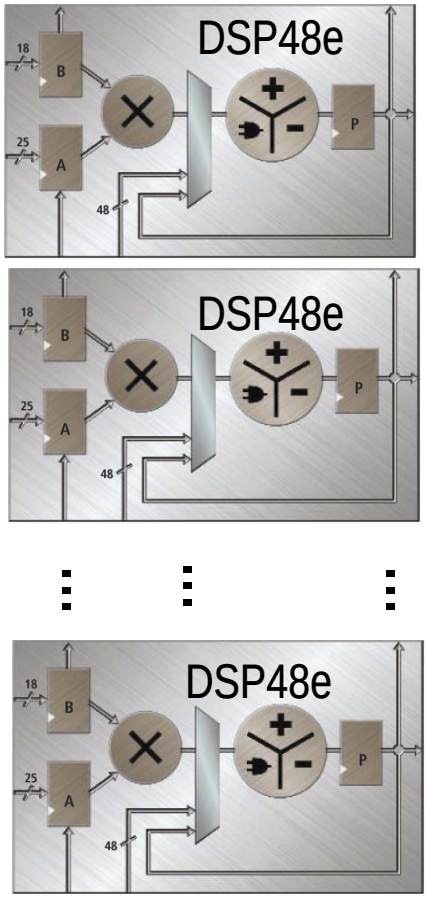


Cascadable DSP48E Slices

Design Challenge
- Need to cascade DSP functions without performance loss
- Minimize resources for complex DSP functions

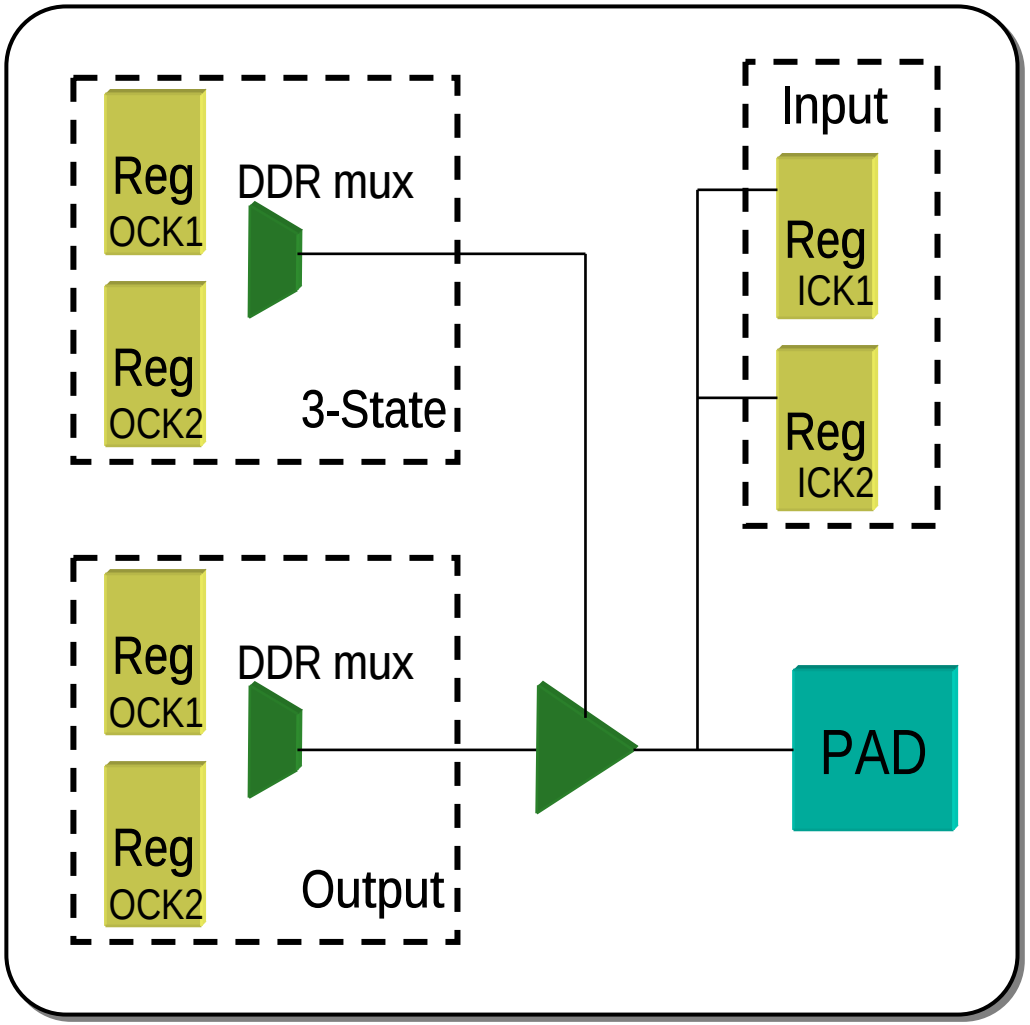
- Adder Chain Implementation
  - 550MHz performance
  - Implements high precision filters with no additional logic
- Complex functions implemented with fewer slices

Functions	Virtex-4 FPGA	Virtex-5 FPGA
35x25 Bit Multiply	4 Slices	2 Slices
35x25 Complex Multiply	8 Slices	4 Slices
24x24 S. P. Floating Point	4 Slices	2 Slices



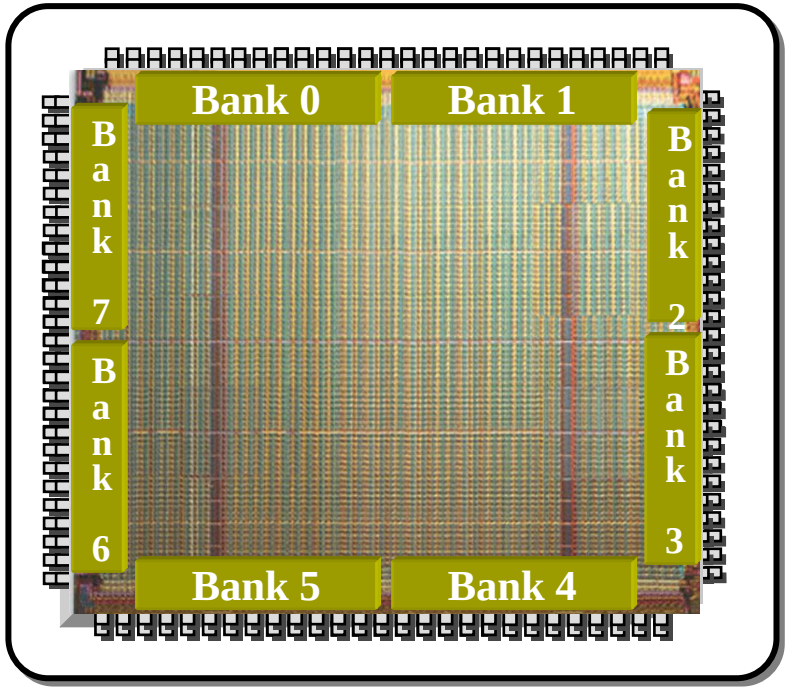
Parallel Adder Chain Implementation

IOB Element



- Input path
  - Two DDR registers
- Output path
  - Two DDR registers
  - Two 3-state DDR registers
- Separate clocks for In & Out
- Set and reset signals are shared
  - Separated sync/async
  - Separated Set/Reset attribute per register

Comprehensive I/O Connectivity



- Single ended and differential
  - 784 single-ended, 344 differential pairs
  - 622 Mb/sec LVDS
  - 23 I/O standards, 8 flexible I/O banks
  - PCI 32/33 and 64/33 support
  - Eliminate costly bus transceivers
  - Multiple package options
- Voltages: 3.3V, 2.5V, 1.8V, 1.5V, 1.2V
- On Chip Digitally Controlled Impedance

8 I/O banks enable multiple simultaneous standards

Chip-to-Chip Interfacing: LVDS, LVCMOS, LVTTL

Backplane Interfacing: GTL, GTL+, PCI, BLVDS

High-speed Memory Interfacing: HSTL, SSTL

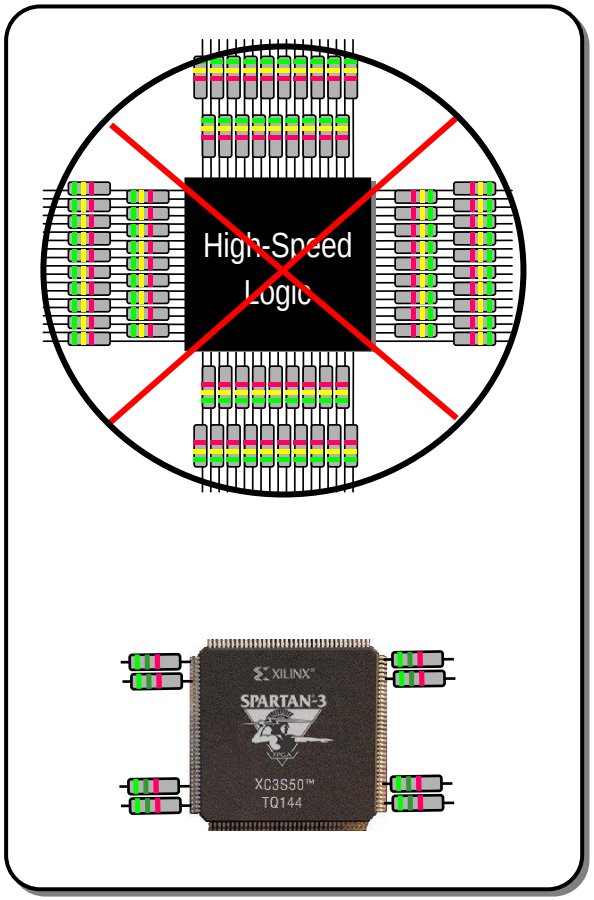
Select I/O-Ultra

	Standard	Output V_{CCO}	Input V_{REF}
Single ended	LVTTL	3.3V	--
	LVCMOS33	3.3V	--
	LVCMOS25	2.5V	--
	LVCMOS18	1.8V	--
	LVCMOS15	1.5V	--
	LVCMOS12	1.2V	--
	PCI 32/64 bit 33MHz	3.3V	--
	SSTL2 Class I	2.5V	1.25V
	SSTL2 Class II	2.5V	1.25V
	SSTL18 Class I	1.8V	0.9V
	HSTL Class I	1.5V	0.75V
	HSTL Class III	1.5V	0.9V
	HSTL18 Class I	1.8V	0.9V
	HSTL18 Class II	1.8V	0.9V
	HSTL18 Class III	1.8V	1.1V
	GTL	--	0.8V
	GTL+	--	1.0V
Differential	LVDS2.5	2.5V	--
	Bus LVDS2.5	2.5V	--
	Ultra LVDS2.5	2.5V	--
	LVDS_ext2.5	2.5V	--
	RSDS	2.5V	--
	LDT2.5	2.5V	--

- 3.3v to 1.2v standards
- More standards for system integration
  - Chip-to-chip, chip-to-memory, chip-to-backplane
- Differential standards
  - Higher I/O performance
  - Lower power and EMI
  - Lower cost

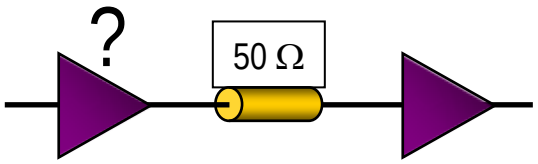
DCI Technology

Digitally Controlled Impedance

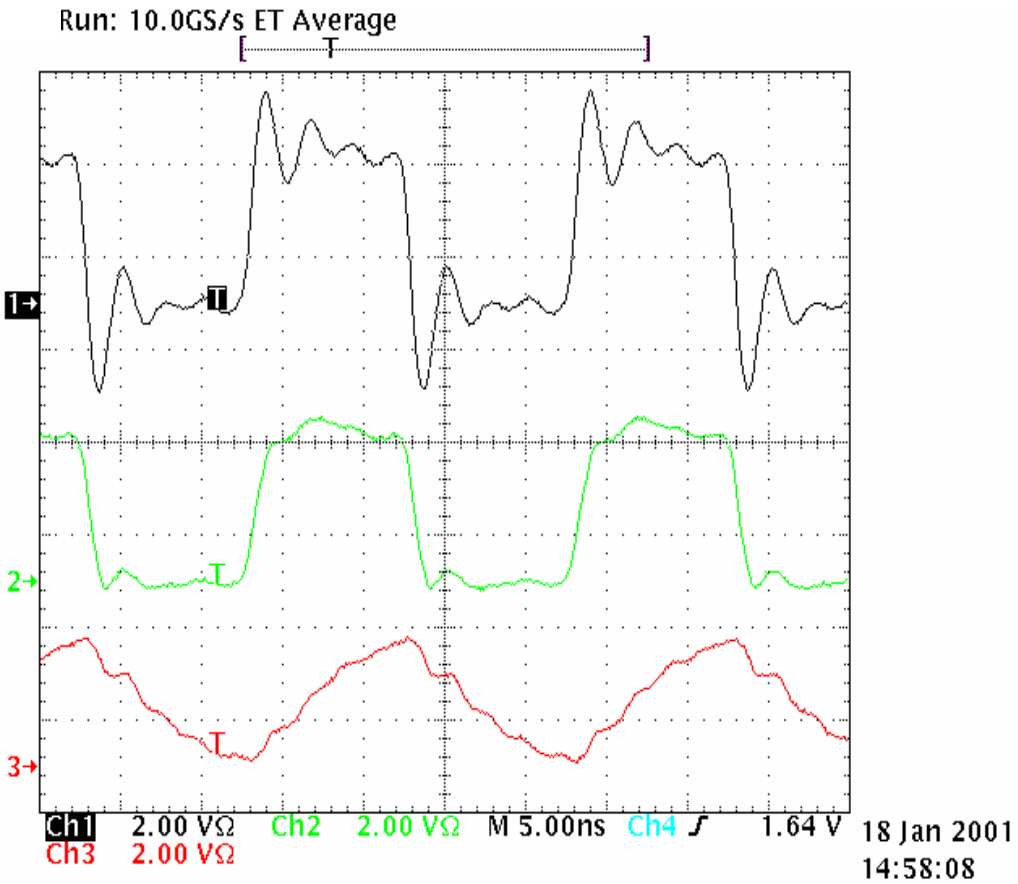


- On-Chip Termination
- Maximize I/O bandwidth
  - Matched I/O low noise signals
  - Full clock period yields faster system performance
- Reduce total board cost
  - Eliminate termination resistors
  - Delivers small broads, easier layout and less layers
- Increase system reliability
  - Greatly reduced overall components
  - Less chance of manufacturing of field failures
- Elimination of Stub reflection Noise
  - No traces between termination resistor and package pins
  - Termination if directly connected to I/O drives

Impedance Mismatches



< 50 Ω	Poor Signal Integrity
50 Ω	OK
> 50 Ω	Poor Signal Integrity



Summary

- CLB: Configurable Logic Block—contains four slices
- Slice: Contains four LUTs, four registers, and carry logic
- LUT: 6-input LUT—the basic element for implementing combinatorial logic
- CMT: two DCM and one PLL per CMT
- DCM: Digital Clock Manager—used to de-skew clock and create other clocks
- PLL: Phase-Locked Loop—reduces internal clock jitter
- Block RAM/FIFO: 36-kb block RAM or FIFO

Xilinx Tool Flow

Lessons

- Overview
- ISE Software
- ISE Simulator
- Summary

Foundation Series ISE Software

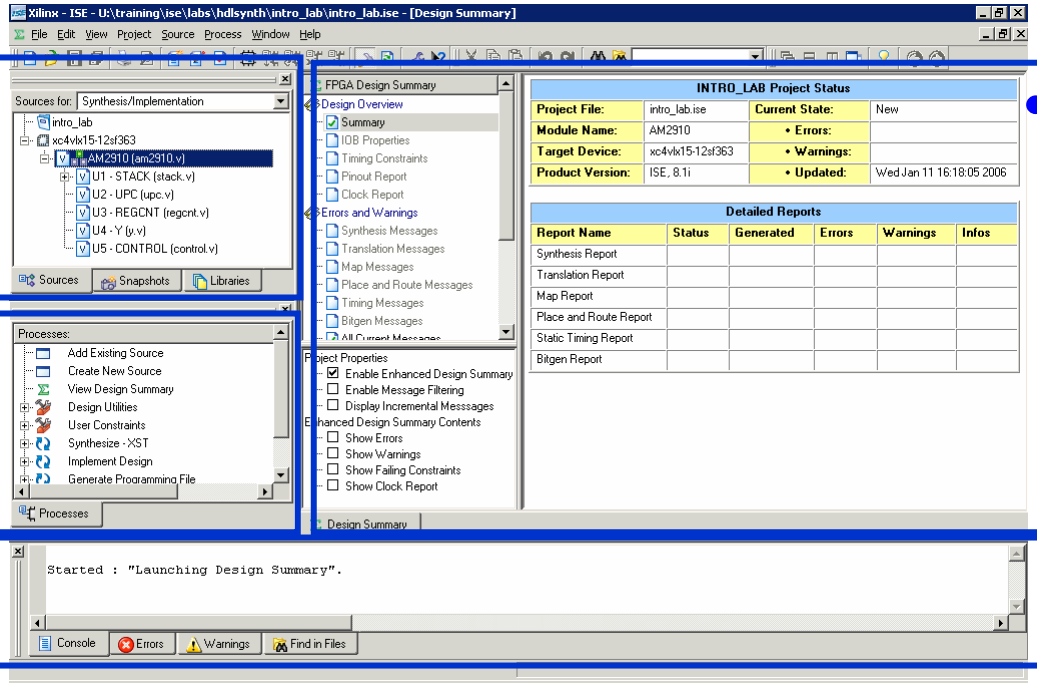
- Foundation Series™ ISE (Integrated Software Environment)

- For PC platforms:
  - Windows 2000 platform
  - Windows XP platform
  - Linux and WINE platform
- For workstation platforms:
  - Solaris Operating System platform

Project Navigator

Sources in Project

Processes for Source

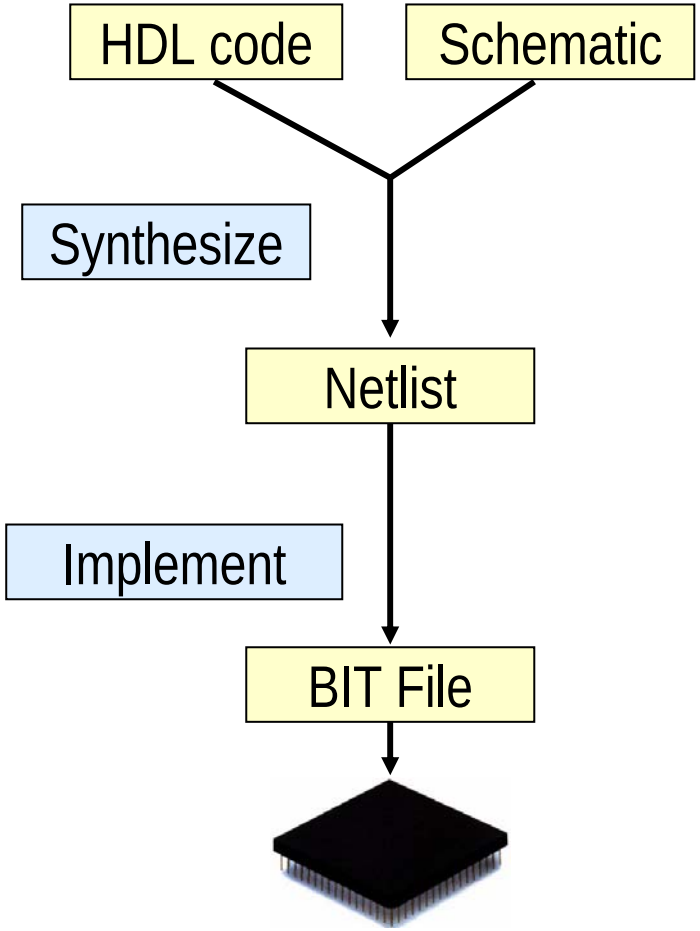


Design Summary (can also provide a Text Editor)

Message Console

Xilinx Design Process

- Step 1: Design entry
  - HDL (Verilog or VHDL) or schematic drawings
- Step 2: Synthesis
  - Translates V, VHD, and SCH files into an industry-standard Electronic Design Interchange Format (EDIF) file format
- Step 3: Implementation
  - Translate, Map, Place & Route
- Step 4: Configuration
  - Download a BIT file into the FPGA
- Simulation can occur after steps 1, 2, or 3



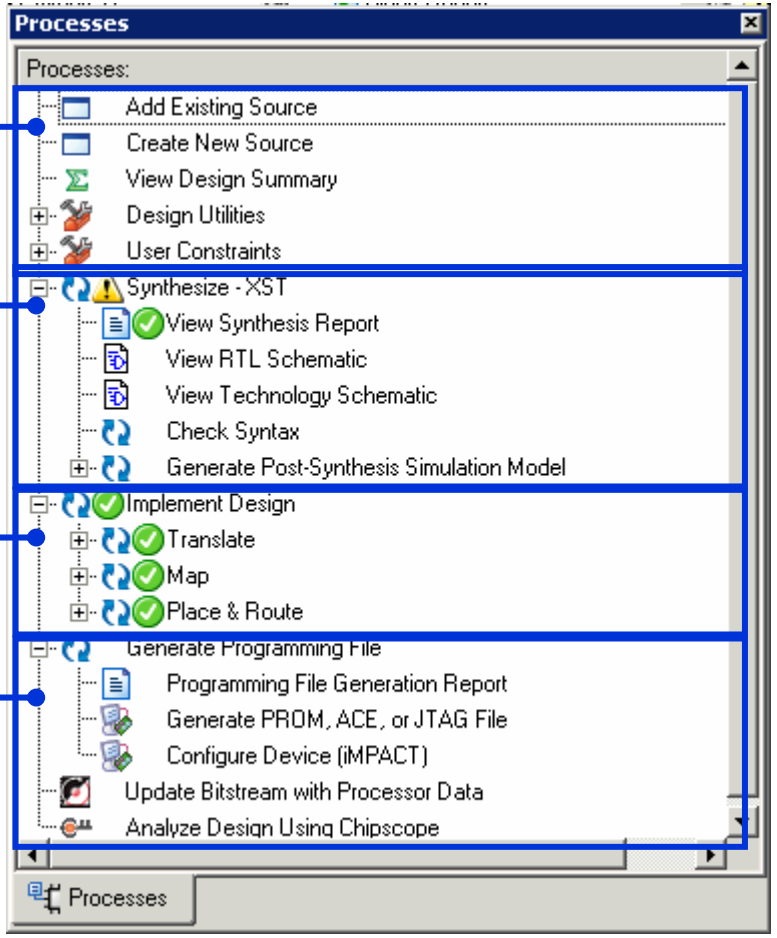
Tools and Processes

Step 1: Design entry

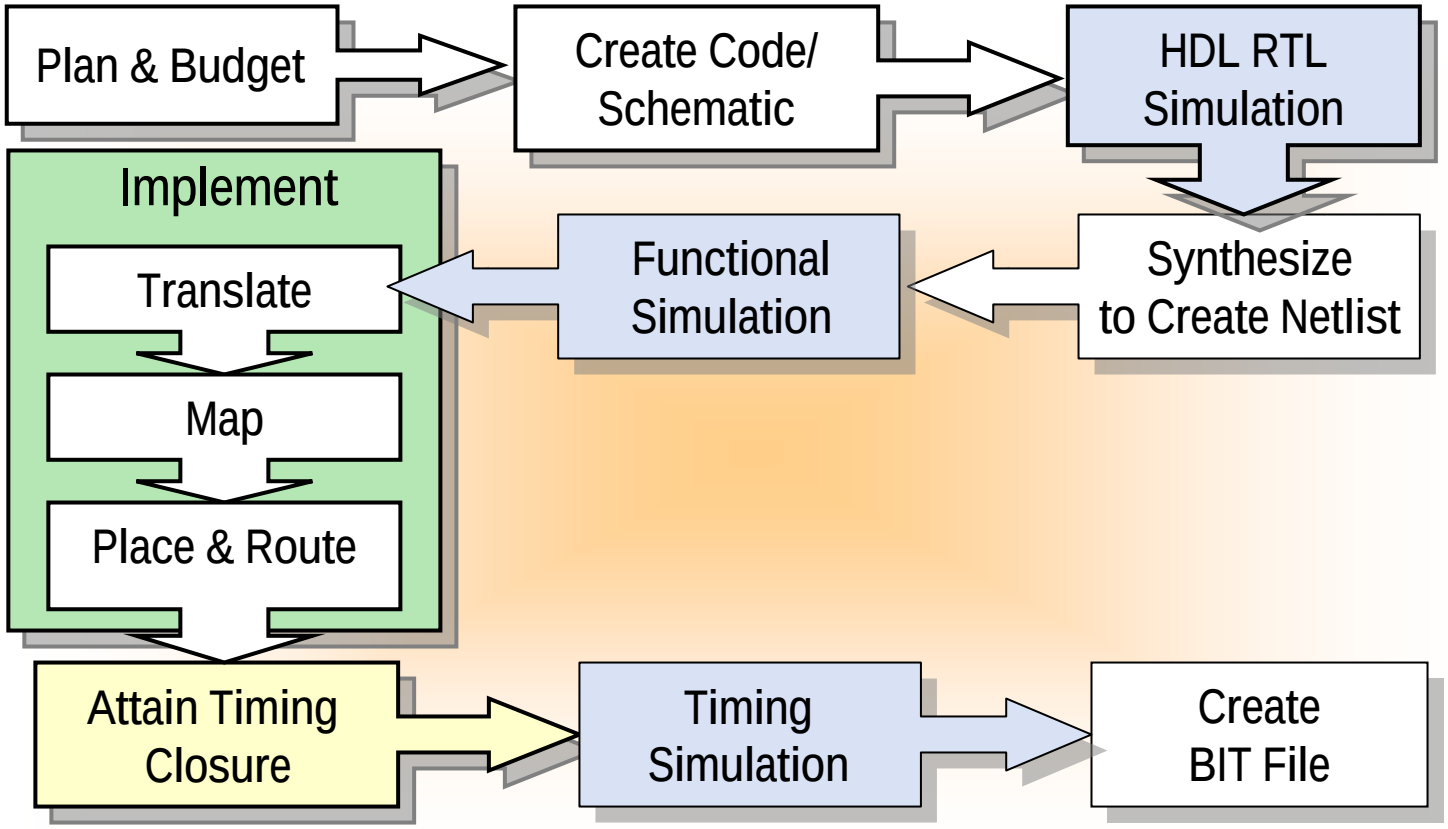
Step 2: Synthesis

Step 3: Implementation

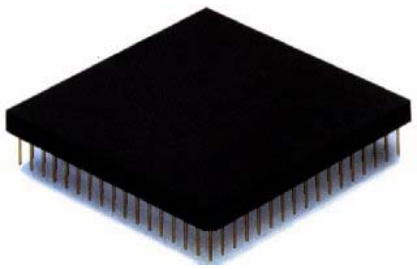
Step 4: Configuration



Implementing a Design into a Xilinx Device



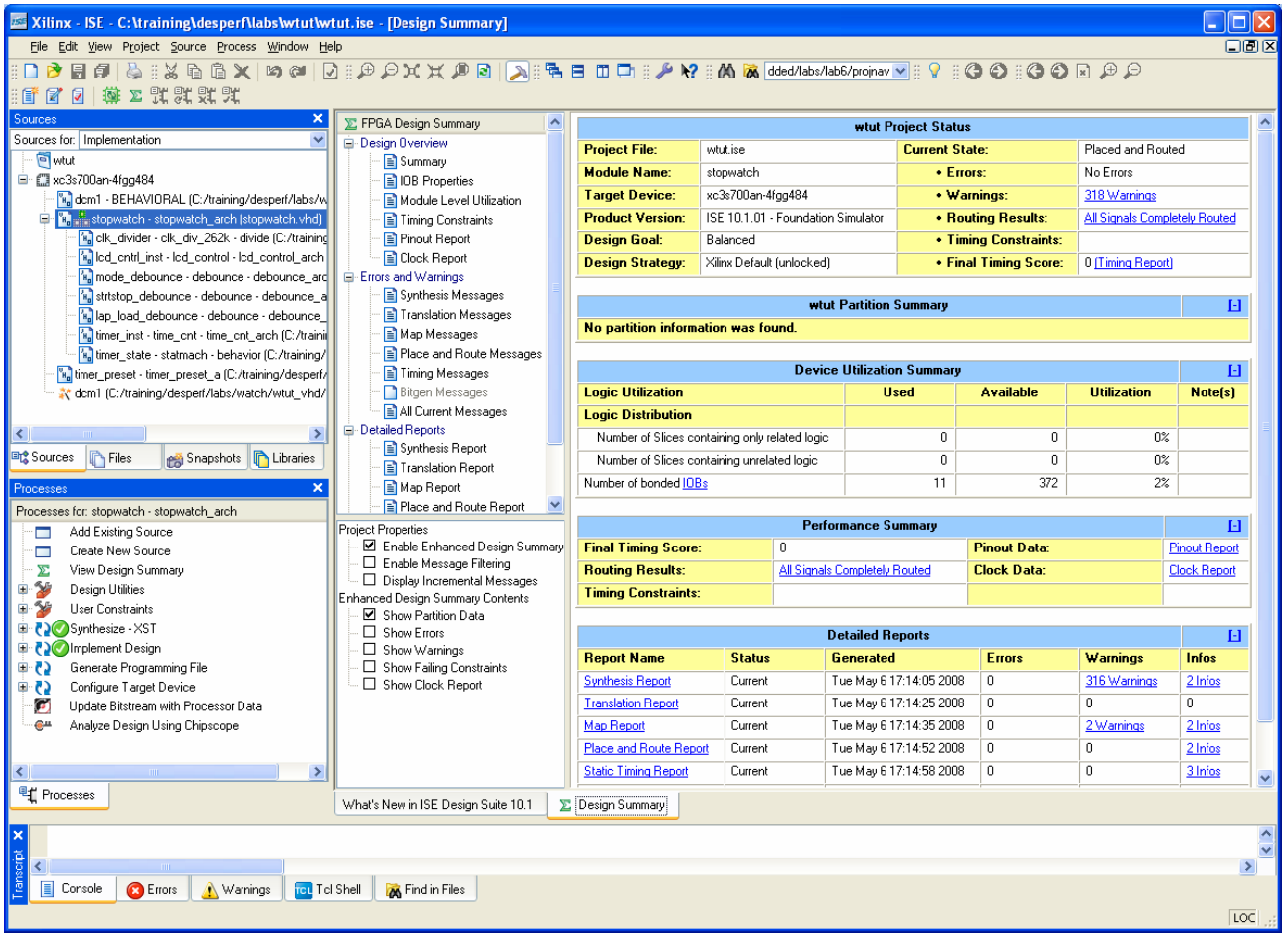
After Implementation, Create a File Called a Bitstream



Lessons

- Overview
- ISE Software
- ISE Simulator
- Summary

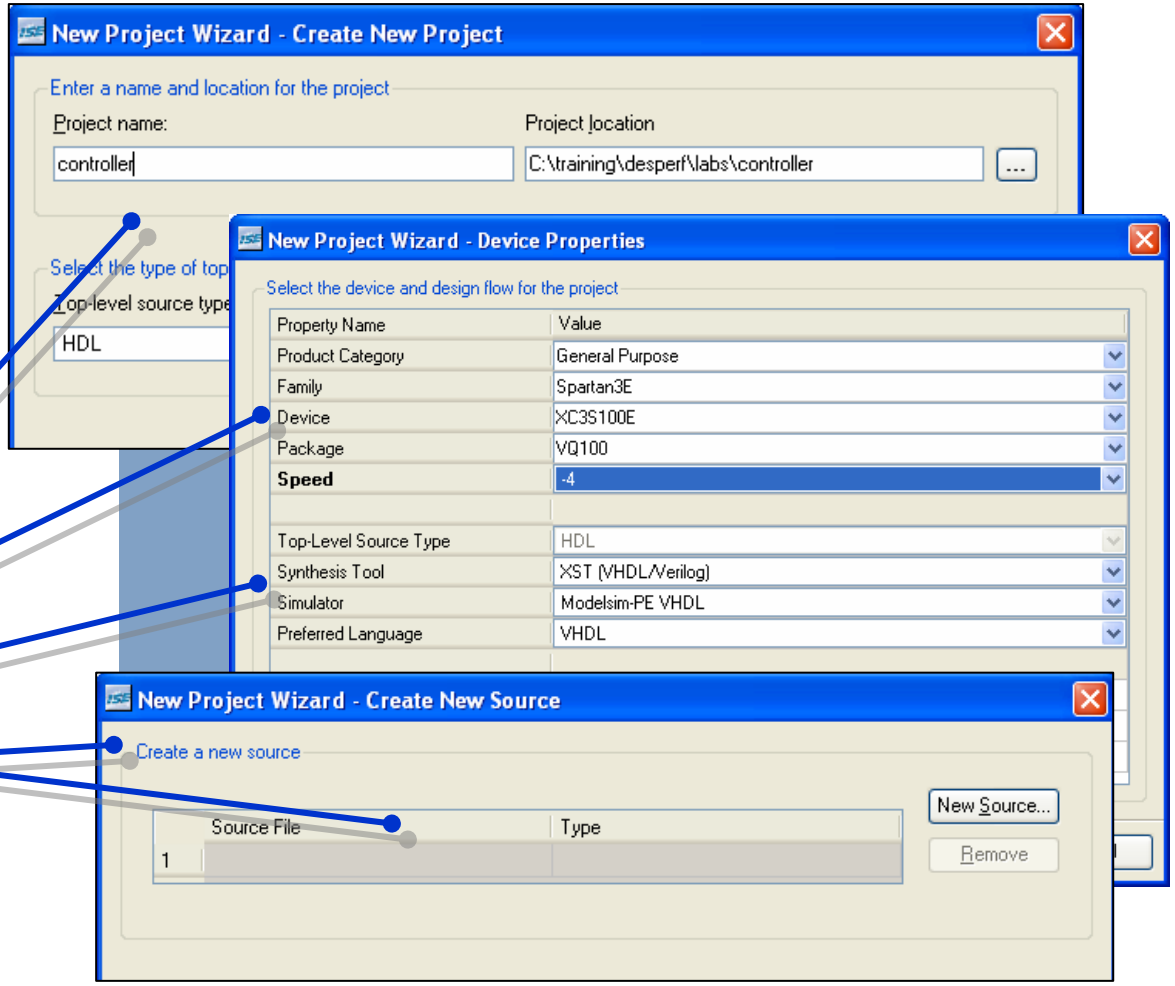
Project Navigator is the Graphical Interface to the ISE Tool Suite



E-ELEMENTS

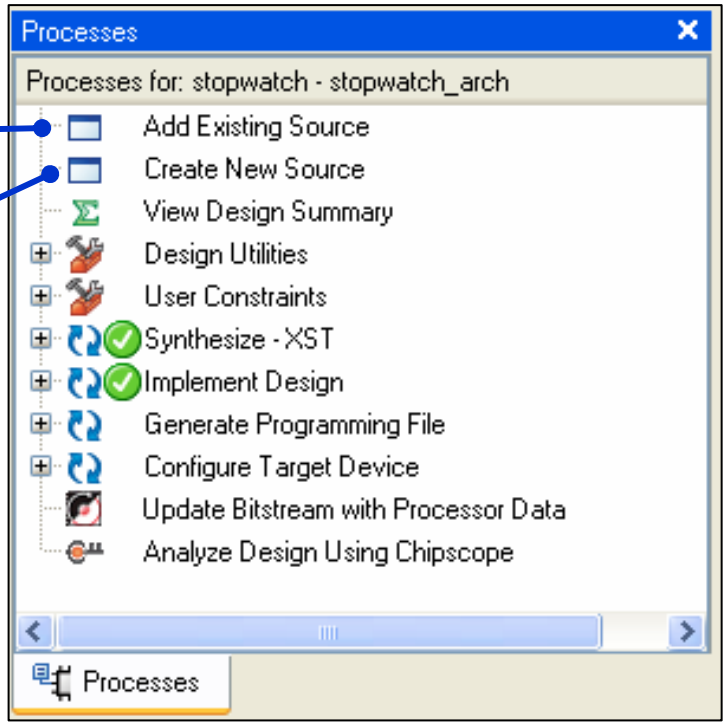
Creating a Project

- Select File → New Project
- New Project Wizard guides you through the process
  - Project name and location
  - Target device
  - Software flow
  - Create or add source files



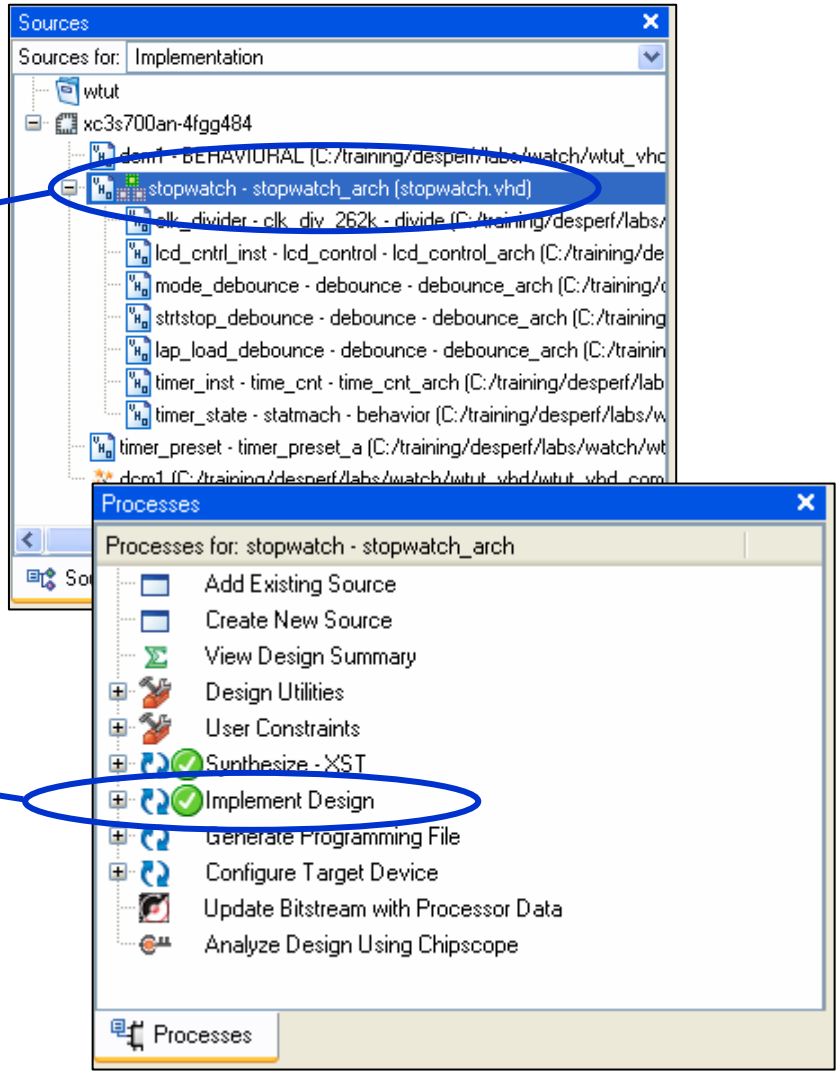
Creating and Adding Source Files

- Double-click **Add Existing Source** to include an existing source file
- Double-click **Create New Source** and choose the type of file to create a new source file
  - HDL
  - IP
  - Schematic
  - State diagram
  - Testbench
  - Constraints



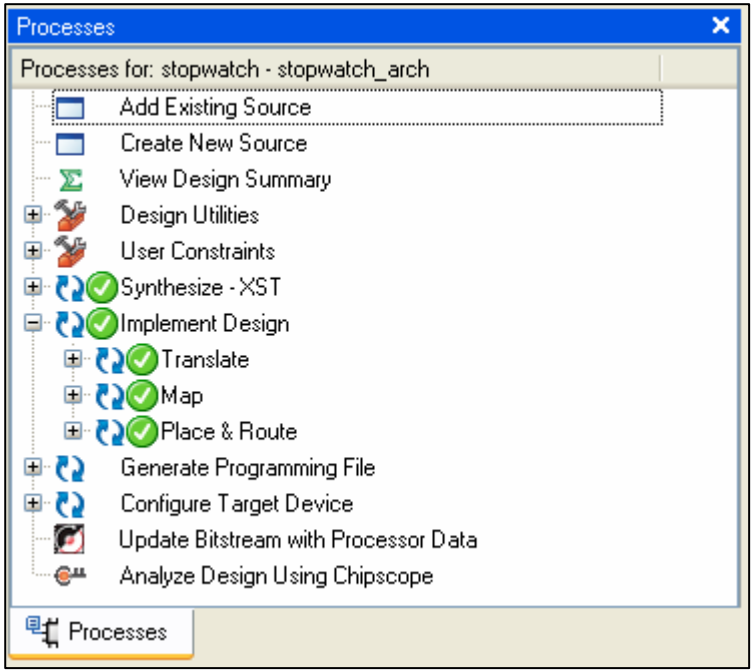
Implementing a Design

- Implement a design
  - Select the top-level source file in the Sources window
    - HDL, schematic, or EDIF, depending on your design flow
  - Double-click **Implement Design** in the Processes window



Checking the Implementation Status

- The ISE software will run all of the necessary steps to implement the design
  - Synthesize HDL code
  - Translate
  - Map
  - Place & Route



✓ = process was completed successfully

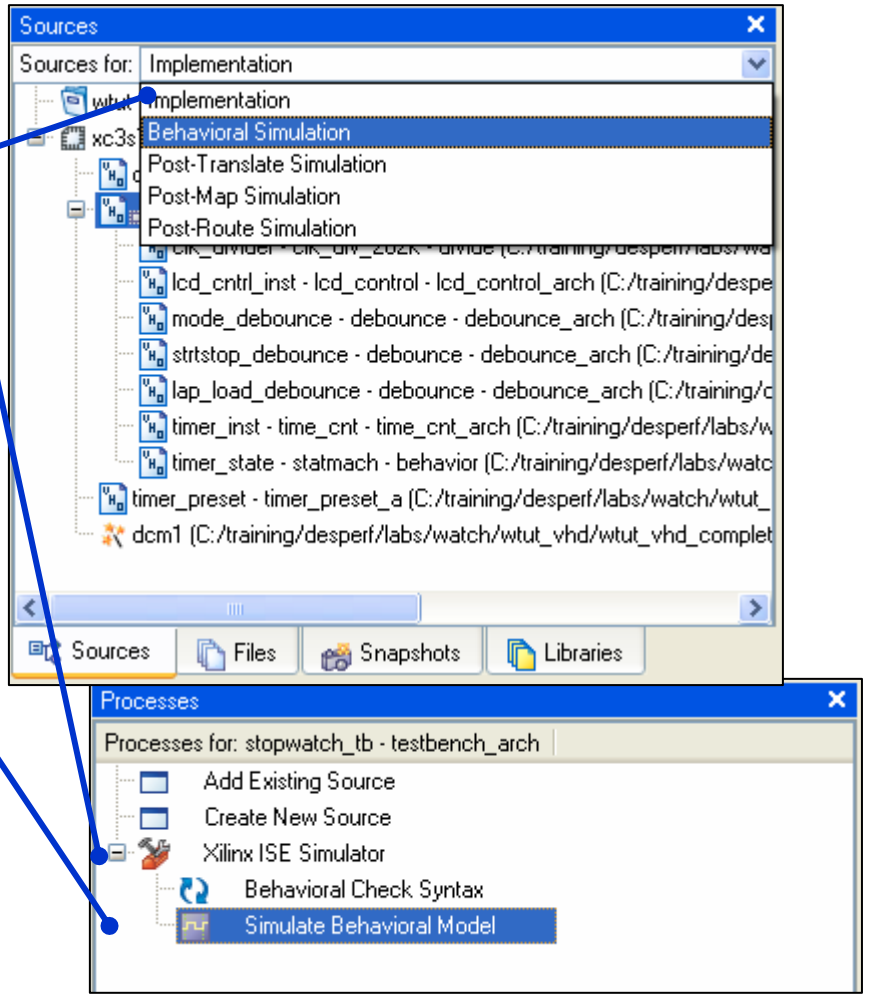
! = warnings

? = a file that is out of date

X = errors

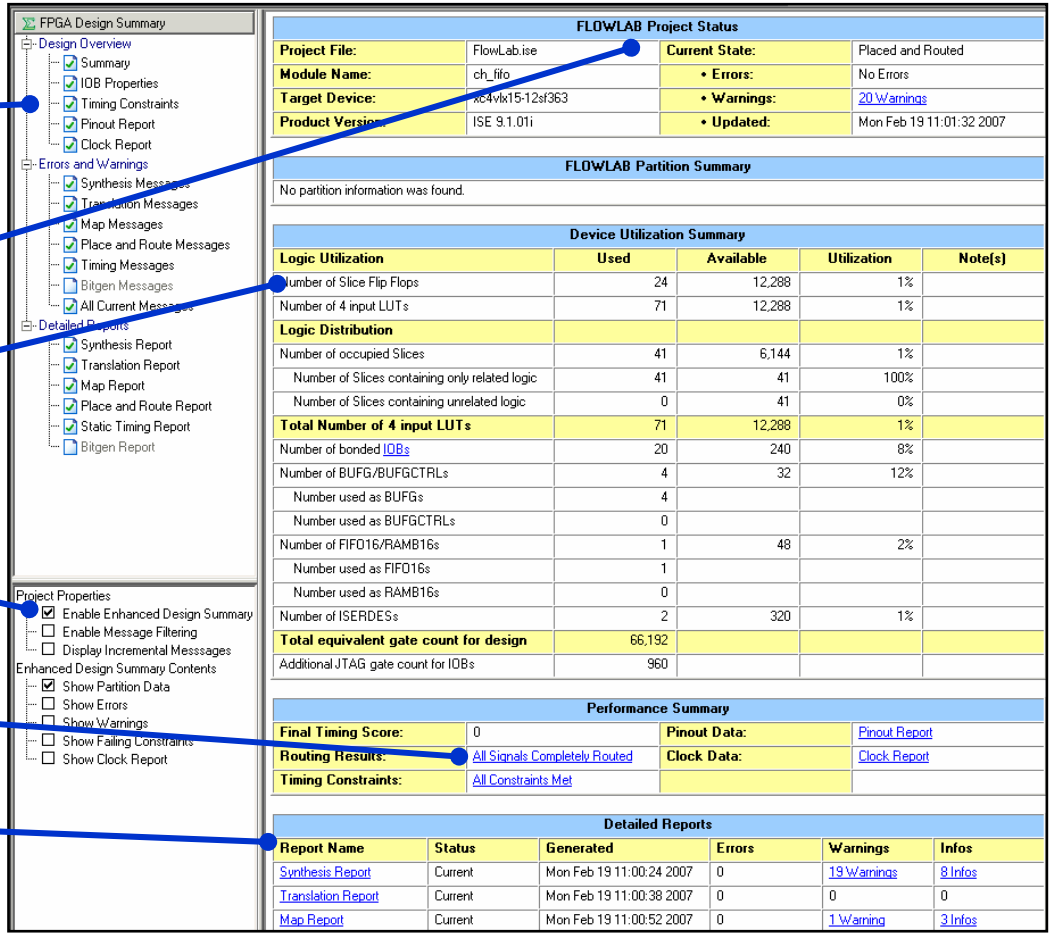
Simulating a Design

- Simulate a design
  - Select Sources for: Behavioral Simulation
  - Expand Xilinx ISE Simulator in the Processes window
  - Double-click Simulate Behavioral Model or Simulate Post-Place & Route Model
    - You can also simulate after Translate or after Map



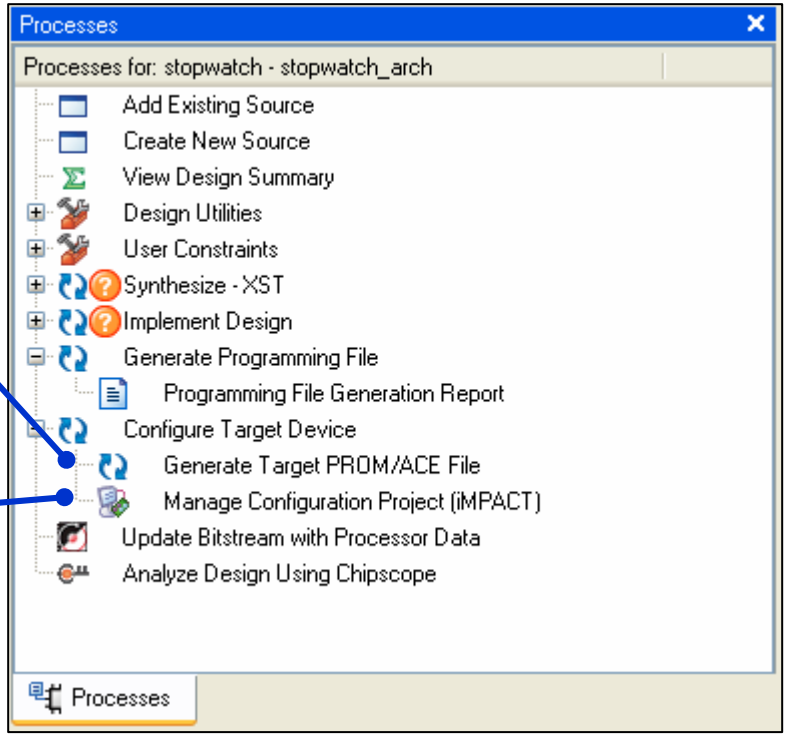
Design Summary Displays Design Data

- Quick view of reports, constraints
- Project status
- Device utilization
- Design summary options
- Performance and constraints
- Reports



Programming the FPGA

- There are two ways to program an FPGA
  - Through a PROM device
    - You must generate a file that the PROM programmer can understand
  - Directly from the computer
    - Use the iMPACT configuration tool



Lessons

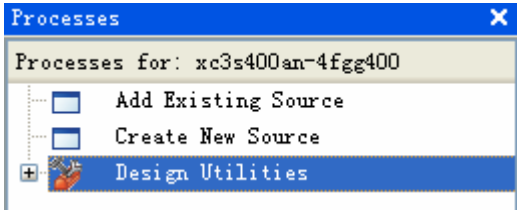
- Overview
- ISE Software
- ISE Simulator
- Summary

ISE Simulator

- First simulator created by Xilinx
- Supports VHDL and Verilog designs
- Includes a waveform editing tool for creating testbenches graphically
  - No need to learn HDL syntax
  - Waveforms can be converted to HDL for using in third-party simulators

Create New Source

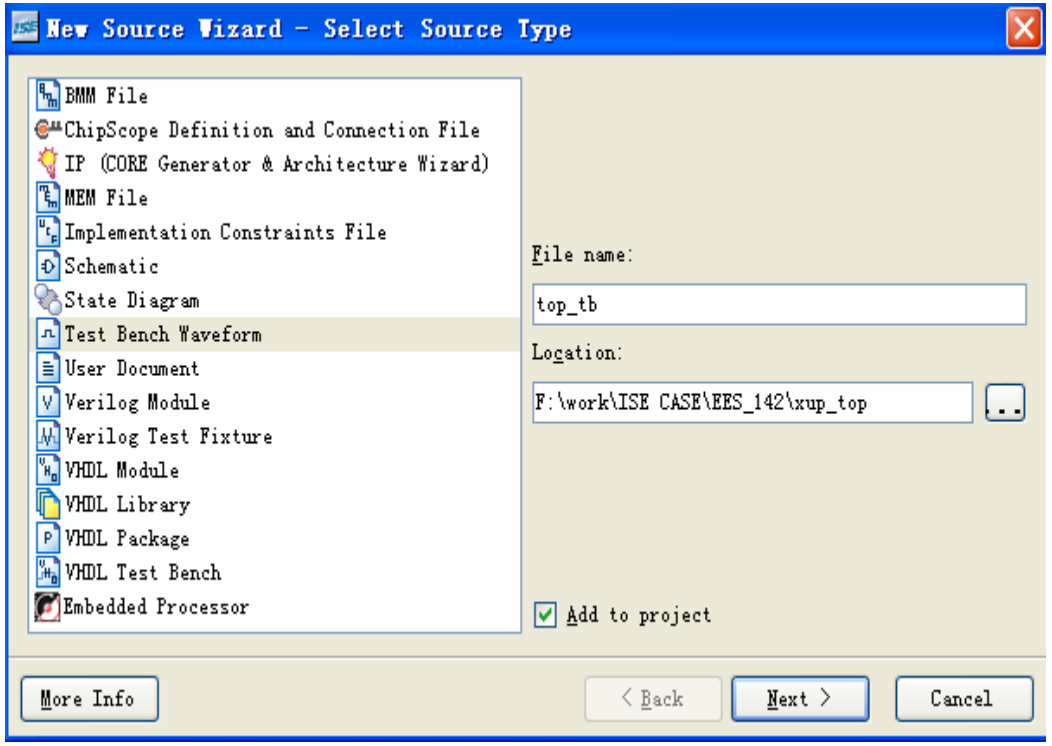
- In the Process window, double-click Create New Source



- Select source type

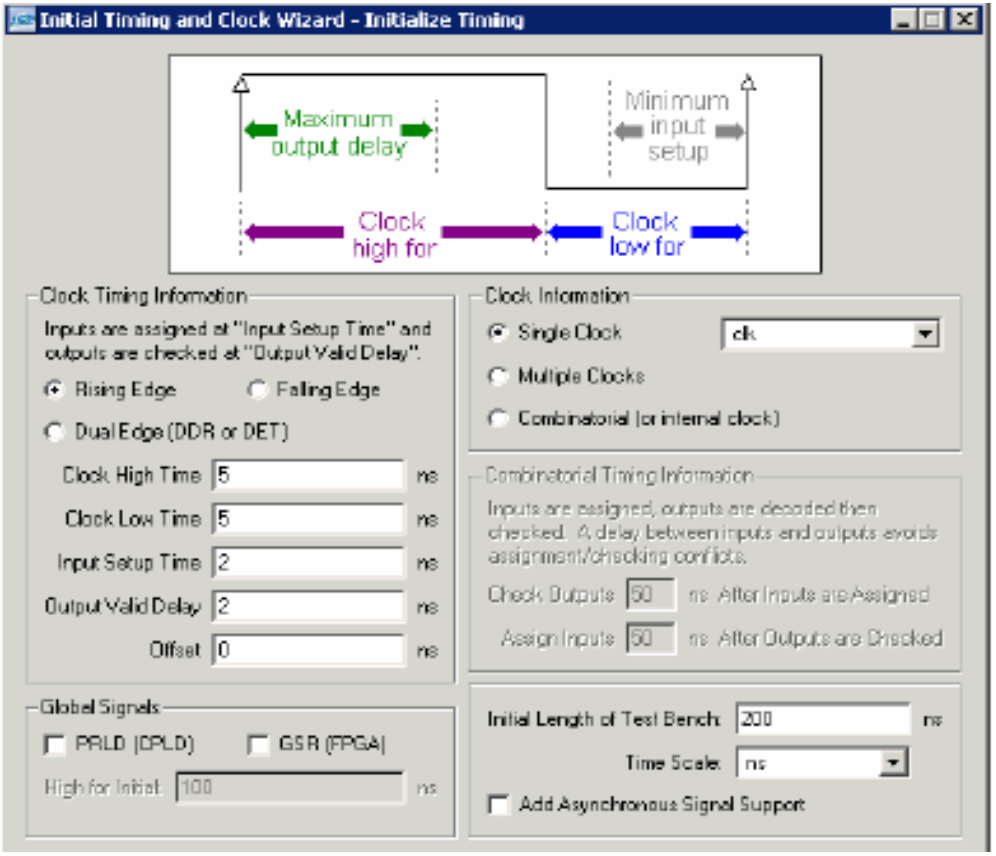
Test Bench Waveform

- Enter a *filename*



Initialize Timing

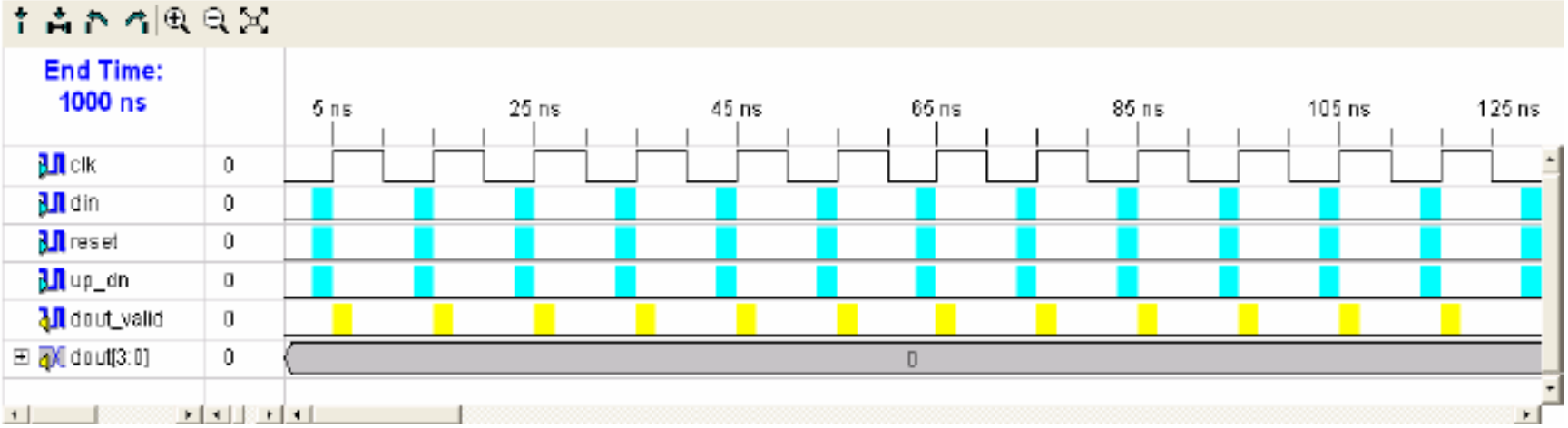
- Define basic timing
- relationships for singleclock,
- fully synchronous
- designs
  - Active edge
  - Clock waveform
  - Input setup time
  - Output valid delay
  - Initial offset
- Global reset support
- Length of testbench



E-ELEMENTS

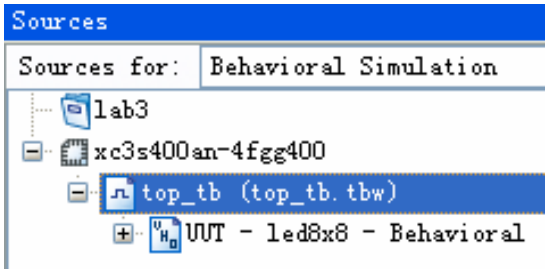
Waveform Editor

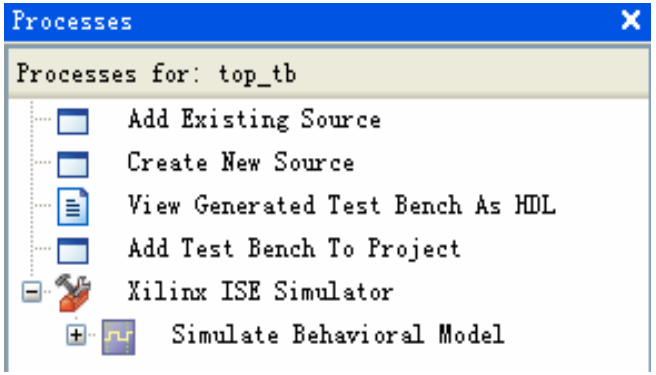
- Blue regions indicate setup time for inputs before a clock edge
- Yellow regions indicate delay for outputs after a clock edge
- Buses can be expanded to view individual signals
  - Default display radix is decimal
- Click a signal to toggle the signal level
  - For buses, click and type in a value



Running a Simulation

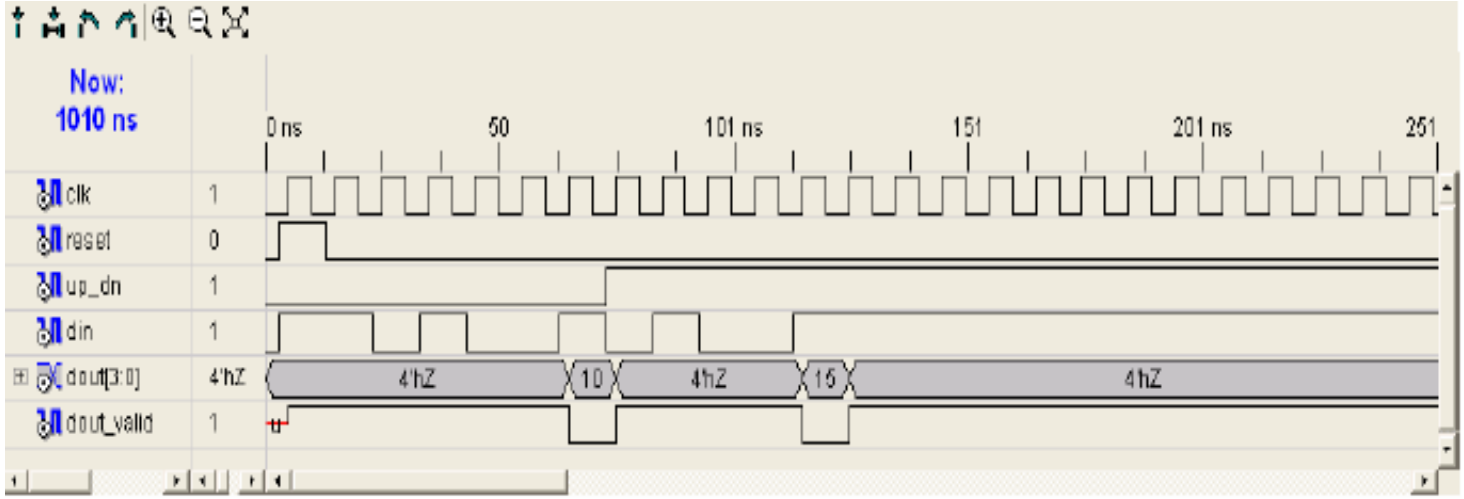
- With a testbench
  - Select a source for **Behavioral Simulation**
  - Select a testbench (.tbw) in the Sources window
  - Expand the **Xilinx ISE Simulator** process
  - Double-click **Simulate Behavioral Model**
  - Testbench automatically executes





Simulation Results

- Background is gray to indicate that this window is read-only
- Toolbar buttons for adding markers, measuring delays, and zooming
- Buses can be expanded to view individual signals



Lessons

- Overview
- ISE Software
- ISE Simulator
- Summary

Summary

- Implementation means more than place & route
- Xilinx provides a simple *pushbutton* tool to guide you through the design process
- The waveform editor helps you create a test bench without requiring knowledge of HDL syntax
- ISE Simulator supports VHDL and Verilog designs

CORE Generator System

E-ELEMENTS

Objectives

After completing this module, you will be able to:

- Describe the differences between LogiCORE™ and AllianceCORE™ solutions
- Identify two benefits of using cores in your designs
- Create customized cores by using the CORE Generator™ software system GUI
- Instantiate cores into your schematic or HDL design
- Run behavioral simulation on a design that contains cores

Outline

- Introduction
- Using the CORE Generator System
- CORE Generator Design Flows
- Summary

What are Cores?

- A *core* is a ready-made function that you can instantiate into your design as a *black box*
- Cores can range in complexity
  - Simple arithmetic operators, such as adders, accumulators, and multipliers
  - System-level building blocks, such as filters, transforms, and memories
  - Specialized functions, such as bus interfaces, controllers, and microprocessors
- Some cores can be customized

Benefits of Using Cores

- Save design time
  - Cores are created by expert designers who have in-depth knowledge of Xilinx FPGA architecture
  - Guaranteed functionality saves time during simulation
- Increase design performance
  - Cores that contain mapping and placement information have predictable performance that is constant over device size and utilization
  - The data sheet for each core provides performance expectations
    - Use timing constraints to achieve maximum performance

Types of Cores

- LogiCORE™ solutions
- AllianceCORE™ solutions

LogiCORE Solutions

- Typically customizable
- Fully tested, documented, and supported by Xilinx
- Many are pre-placed for predictable timing
- Many are unlicensed and provided for free with Xilinx software
  - More complex LogiCORE™ solution products are licensed
- VHDL and Verilog flow support for several EDA tools
- Schematic flow support for most cores

AllianceCORE Solutions

- Point-solution cores
  - Typically not customizable (some HDL versions are customizable)
- Sold and supported by Xilinx AllianceCORE™ solution partners
  - Partners can be contacted directly to provide customized cores
- All cores are optimized for Xilinx; some are pre-placed
- Typically supplied as an Electronic Design Interchange Format (EDIF) netlist
- VHDL and Verilog flow support; some schematic support

Sample Functions

- LogiCORE™ solutions
  - DSP functions
    - Time skew buffers, Finite Impulse Response (FIR) filters, and correlators
  - Math functions
    - Accumulators, adders, multipliers, integrators, and square root
  - Memories
    - Pipelined delay elements, single- and dual-port RAM
    - Synchronous FIFOs
  - PCI master and slave interfaces, PCI bridge

- AllianceCORE™ solutions
  - Peripherals
    - DMA controllers
    - Programmable interrupt controllers
    - UARTs
  - Communications and networking
    - ATM
    - Reed-Solomon encoders and decoders
    - T1 framers
  - Standard bus interfaces
    - PCMCIA, USB

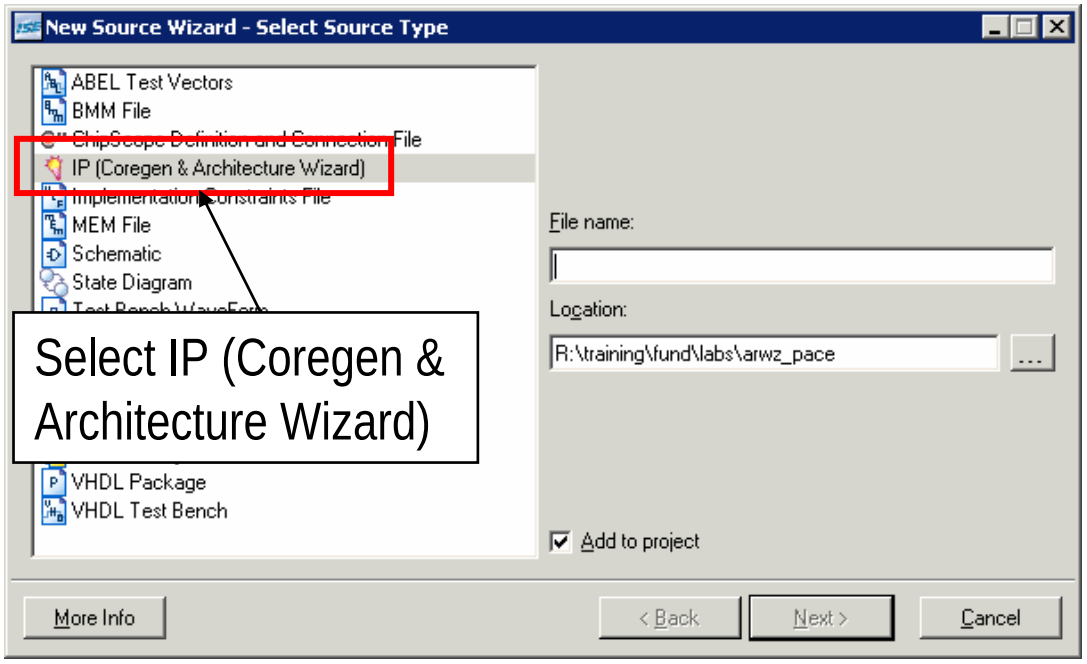
Outline

- Introduction
- Using the CORE Generator System
- CORE Generator Design Flows
- Summary

Invoking CORE Generator

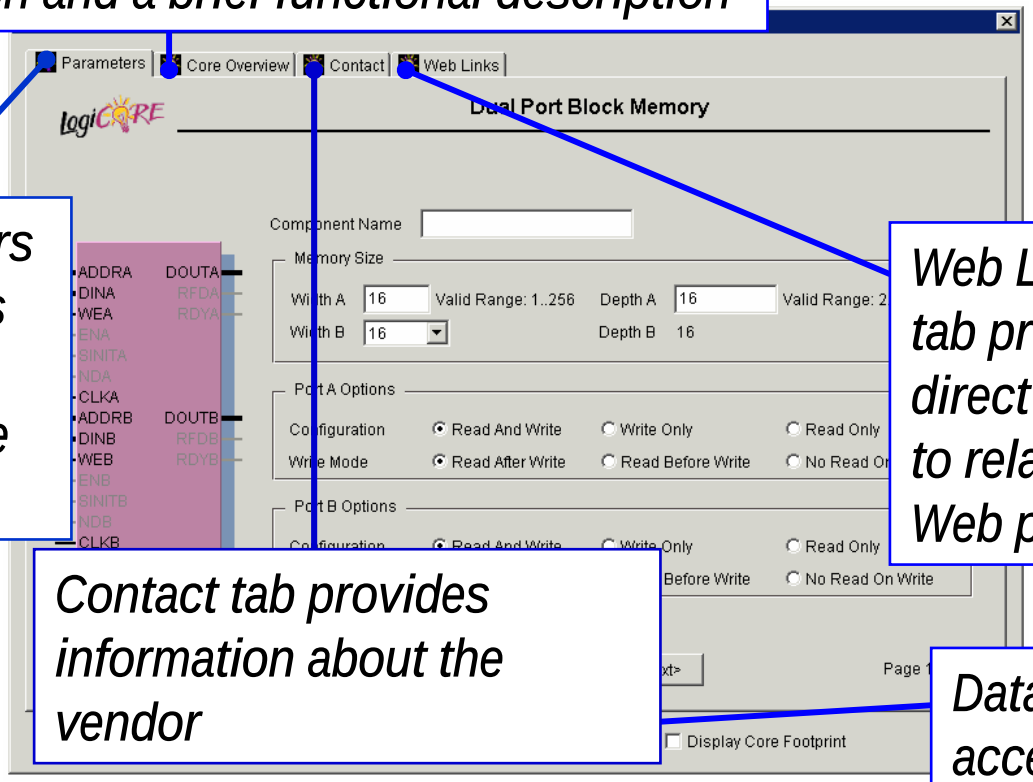
Can access from within ISE using the New Source Wizard

- From the Project Navigator, select **Project** → **New Source**
- Select **IP (CoreGen & Architecture Wizard)** and enter a *filename*
- Click **Next** and then select the **type of core**



Core Customize Window

Core Overview tab provides version information and a brief functional description



Parameters tab allows you to customize the core

Web Links tab provides direct access to related Web pages

Contact tab provides information about the vendor

Data sheet access

CORE Data Sheets

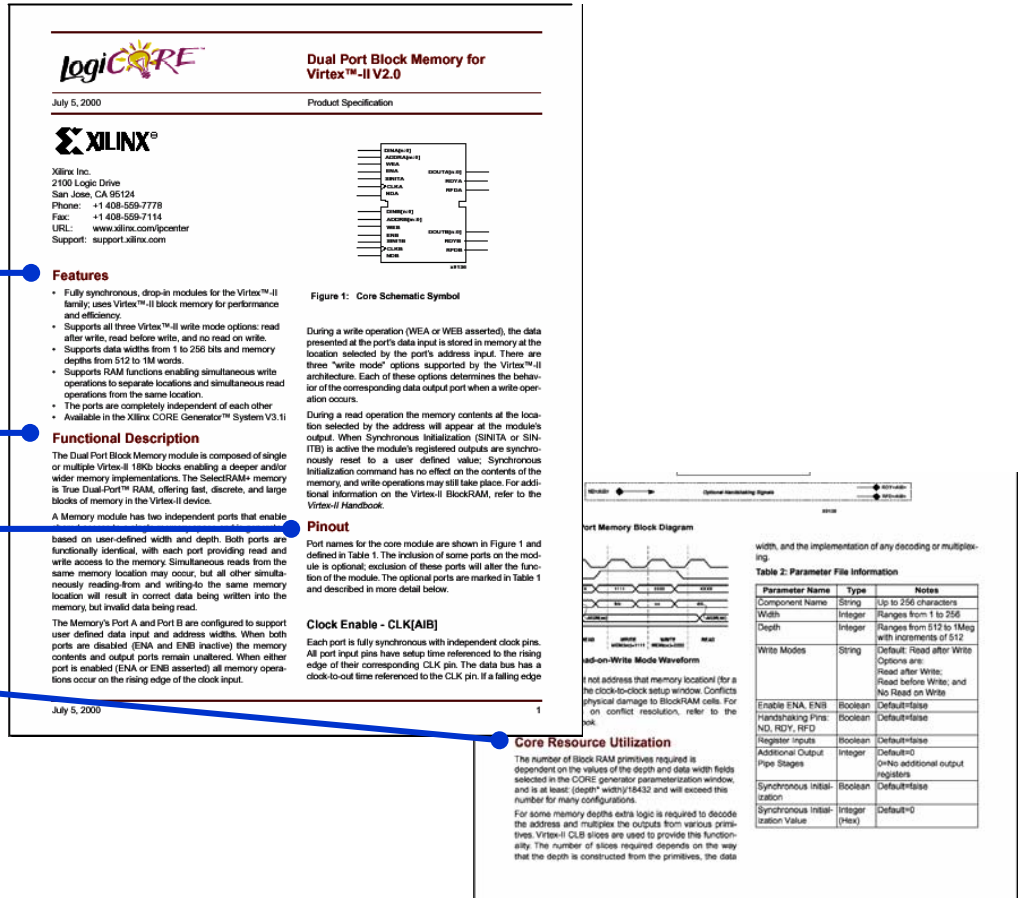
- Performance expectations (not shown)

Features

Functionality

Pinout

Resource utilization



LogiCORE

Dual Port Block Memory for Virtex™-II V2.0

July 5, 2000 — Product Specification

XILINX

Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com/ipcenter
Support: support.xilinx.com

Features

- Fully synchronous, drop-in modules for the Virtex™-II family; uses Virtex™-II block memory for performance and efficiency.
- Supports all three Virtex™-II write mode options: read after write, read before write, and no read on write.
- Supports data widths from 1 to 256 bits and memory depths from 512 to 1M words.
- Supports RAM functions enabling simultaneous write operations to separate locations and simultaneous read operations from the same location.
- The ports are completely independent of each other
- Available in the Xilinx CORE Generator™ System V3.1i

Functional Description

The Dual Port Block Memory module is composed of single or multiple Virtex-II 18Kb blocks enabling a deeper and/or wider memory implementations. The SelectRAM+ memory is True Dual-Port™ RAM, offering fast, discrete, and large blocks of memory in the Virtex-II device.

A Memory module has two independent ports that enable

based on user-defined width and depth. Both ports are functionally identical, with each port providing read and write access to the memory. Simultaneous reads from the same memory location may occur, but all other simultaneously reading-from and writing-to the same memory location will result in correct data being written into the memory, but invalid data being read.

The Memory's Port A and Port B are configured to support user defined data input and address widths. When both ports are disabled (ENA and ENB inactive) the memory contents and output ports remain unaltered. When either port is enabled (ENA or ENB asserted) all memory operations occur on the rising edge of the clock input.

Figure 1: Core Schematic Symbol

During a write operation (WEA or WEB asserted), the data presented at the port's data input is stored in memory at the location selected by the port's address input. There are three "write mode" options supported by the Virtex™-II architecture. Each of these options determines the behavior of the corresponding data output port when a write operation occurs.

During a read operation the memory contents at the location selected by the address will appear at the module's output. When Synchronous Initialization (SINITA or SINITB) is active the module's registered outputs are synchronously reset to a user defined value; Synchronous Initialization command has no effect on the contents of the memory, and write operations may still take place. For additional information on the Virtex-II BlockRAM, refer to the *Virtex-II Handbook*.

Pinout

Port names for the core module are shown in Figure 1 and defined in Table 1. The inclusion of some ports on the module is optional; exclusion of these ports will alter the function of the module. The optional ports are marked in Table 1 and described in more detail below.

Clock Enable - CLK[AIB]

Each port is fully synchronous with independent clock pins. All port input pins have setup time referenced to the rising edge of their corresponding CLK pin. The data bus has a clock-to-out time referenced to the CLK pin. If a falling edge

July 5, 2000 — 1

ort Memory Block Diagram

ad-on-Write Mode Waveform

Core Resource Utilization

The number of Block RAM primitives required is dependent on the values of the depth and data width fields selected in the CORE generator parameterization window, and is at least: (depth* width)/18432 and will exceed this number for many configurations.

For some memory depths extra logic is required to decode the address and multiplex the outputs from various primitives. Virtex-II CLB slices are used to provide this functionality. The number of slices required depends on the way that the depth is constructed from the primitives, the data

width, and the implementation of any decoding or multiplexing.

Table 2: Parameter File Information

Parameter Name	Type	Notes
Component Name	String	Up to 256 characters
Width	Integer	Ranges from 1 to 256
Depth	Integer	Ranges from 512 to 1Meg with increments of 512
Write Modes	String	Default: Read after Write Options are: Read after Write; Read before Write; and No Read on Write
Enable ENA, ENB	Boolean	Default=false
Handshaking Pins: ND, RDY, RFD	Boolean	Default=false
Register Inputs	Boolean	Default=false
Additional Output Pipe Stages	Integer	Default=0 0=No additional output registers
Synchronous Initialization	Boolean	Default=false
Synchronous Initialization Value	Integer (Hex)	Default=0

Outline

- Introduction
- Using the CORE Generator System
- CORE Generator Design Flows
- Summary

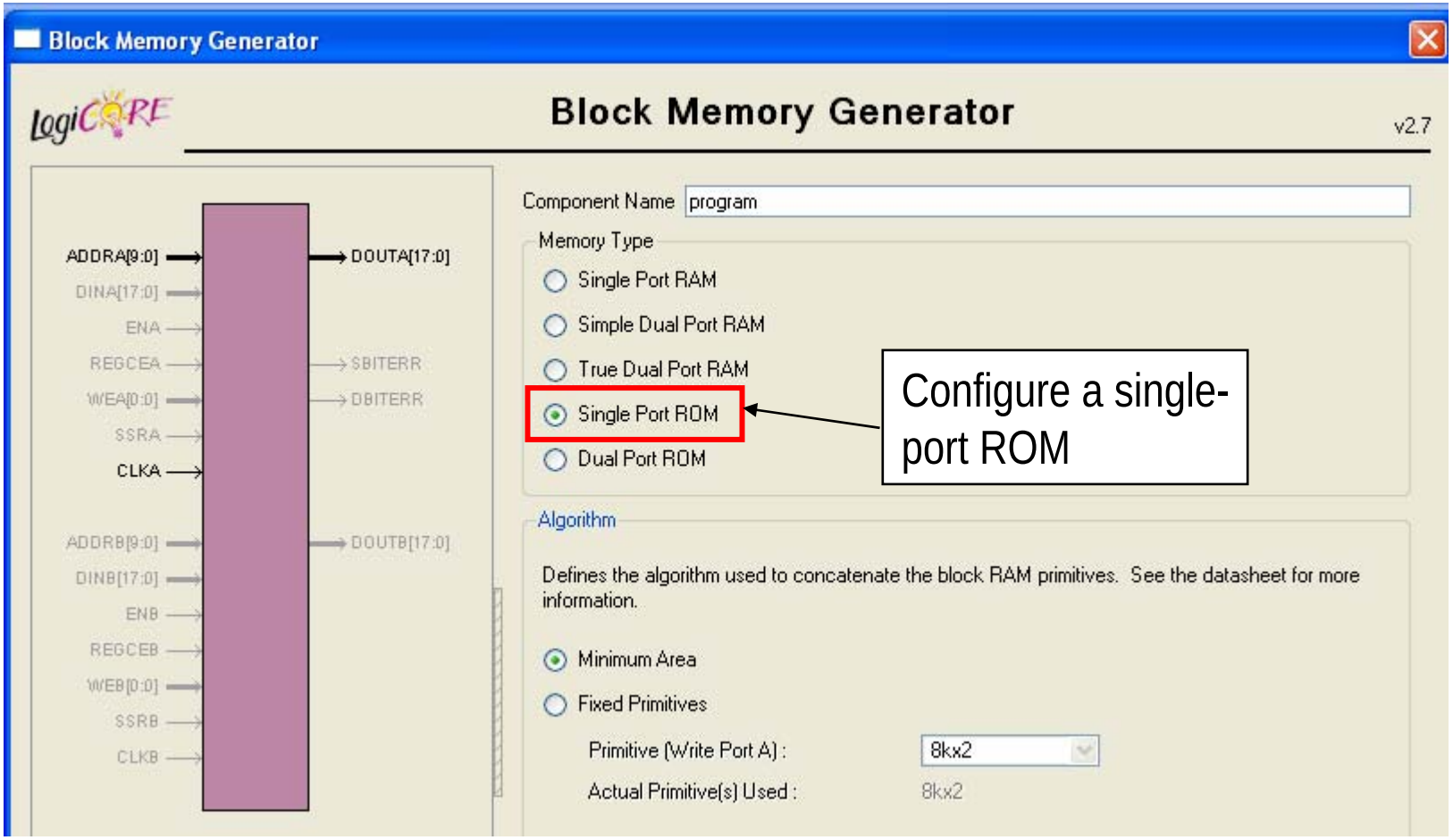
HDL Design Flow

Generate and integrate cores

- Generate a core
  - Netlist file (EDN, NGO)
  - Instantiation template files (VHO or VEO)
  - Behavioral simulation wrapper files (VHD or V)
- Instantiate the core
  - Cut and paste from the templates provided in the VEO or VHO file
- Perform a behavioral simulation
  - The ISE™ software automatically uses wrapper files when cores are present in the design
  - VHDL: Analyze the wrapper file for each core before analyzing the file that instantiates the core
- Synthesize and implement the design

Generate Core

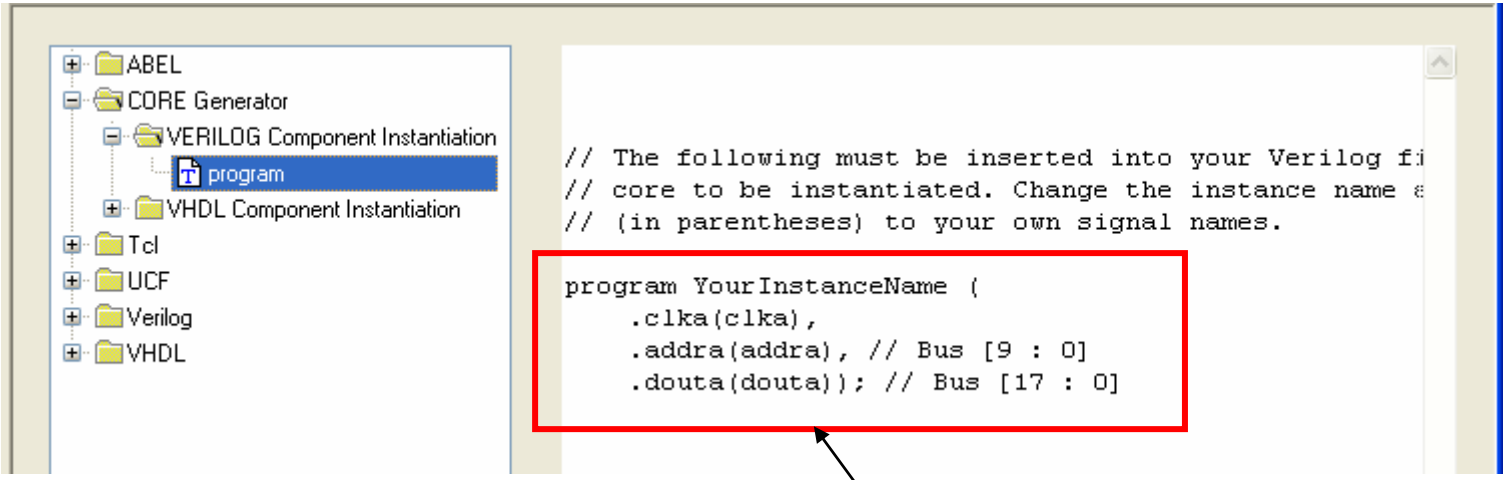
Parameterize the core using the GUI and generate the project files



Instantiate Core

Access HDL instantiation templates (VHO and VEO) from the ISE Language Templates

Go to Edit → Language Templates



Copy into the design
and connect

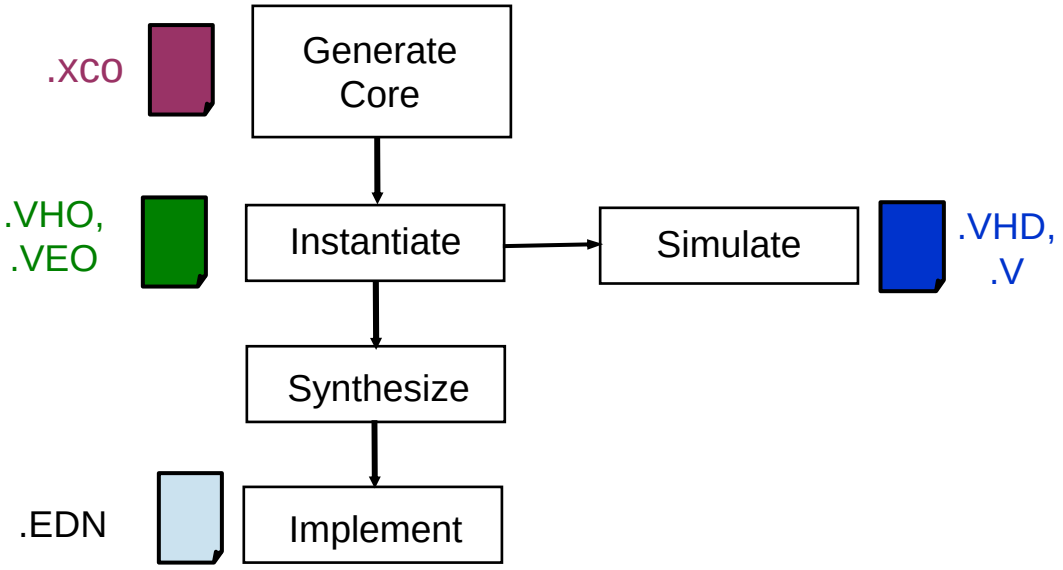
Perform Behavioral Simulation

XilinxCoreLib simulation library available for performing behavioral simulation on netlist cores

- Before your first behavioral simulation, you must run *compxlib.exe* to compile the XilinxCoreLib simulation library
  - Located in the *$XILINX\bin\<platform>* directory
- If you download new or updated cores, additional simulation models will be automatically extracted during installation

Synthesize and Implement

The core netlist will merge with the design during the translate phase of Implementation



Outline

- Introduction
- Using the CORE Generator System
- CORE Generator Design Flows
- Summary

Summary

- A core is a ready-made function that you can insert into your design
- LogiCORE™ solution products are sold and supported by Xilinx
- AllianceCORE™ solution products are sold and supported by AllianceCORE solution partners
- Using cores can save design time and provide increased performance
- Cores can be used in schematic or HDL design flows