Digital Design using VHDL and Xilinx FPGA

VHDL based synthesis

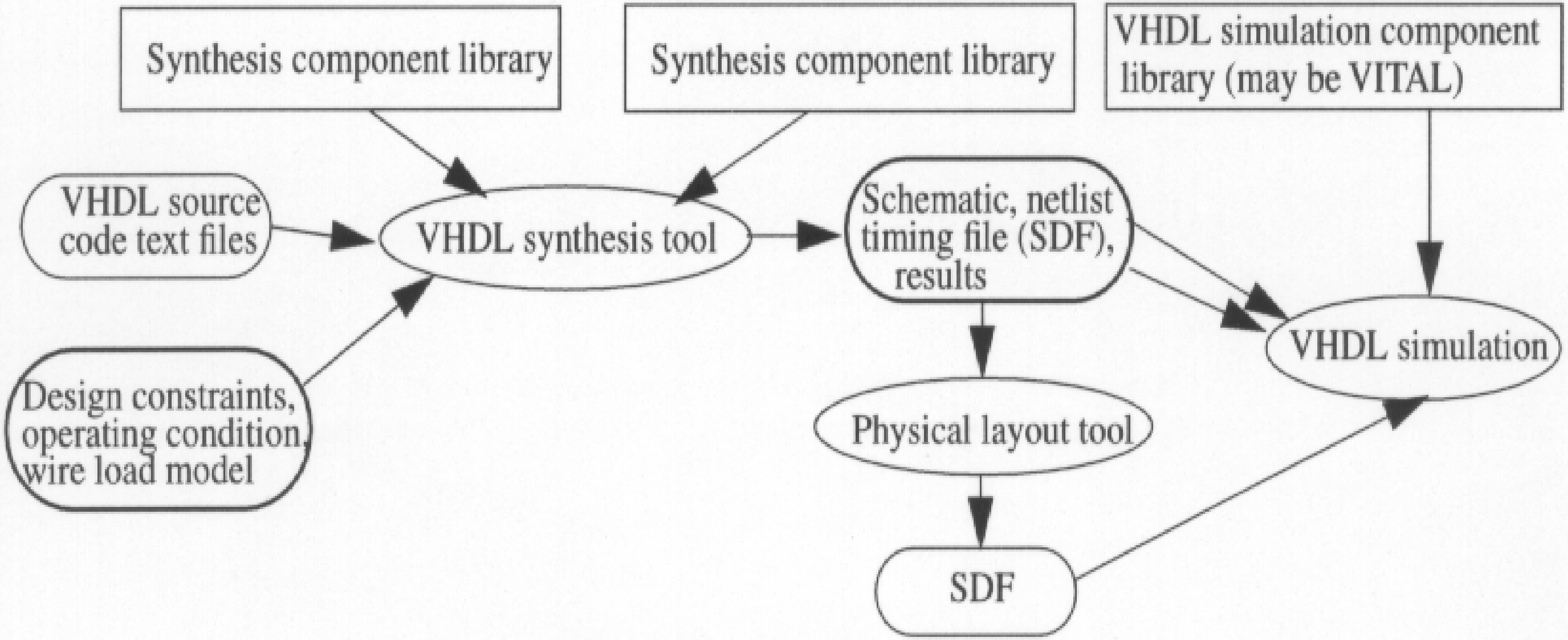

Synthesis component library
Synthesis component library
VHDL simulation component library (may be VITAL)
VHDL source code text files
VHDL synthesis tool
Schematic, netlist timing file (SDF), results
Design constraints, operating condition, wire load model
VHDL simulation
Physical layout tool
SDF

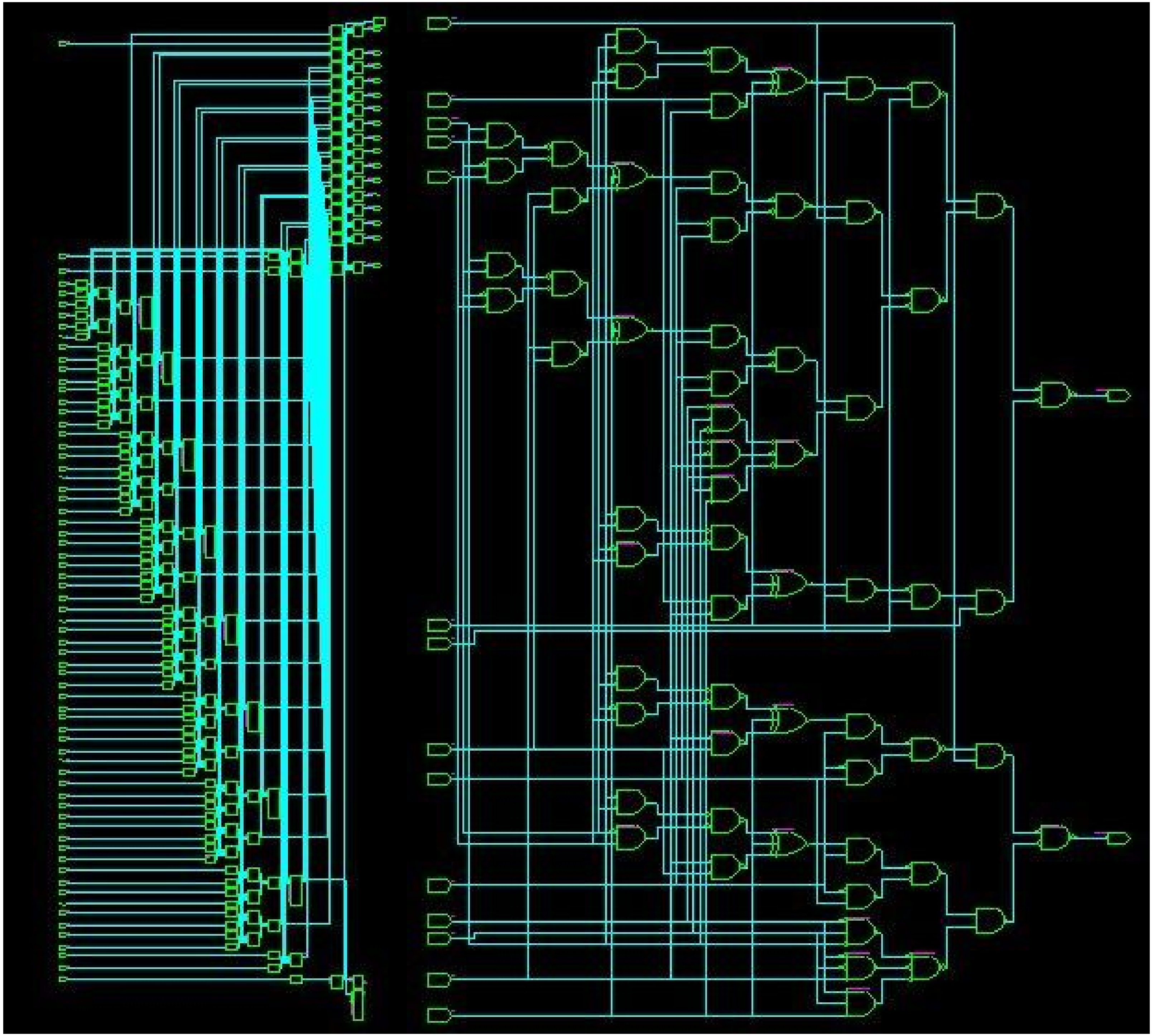
FIGURE 8.12 Synthesis framework.

VHDL code

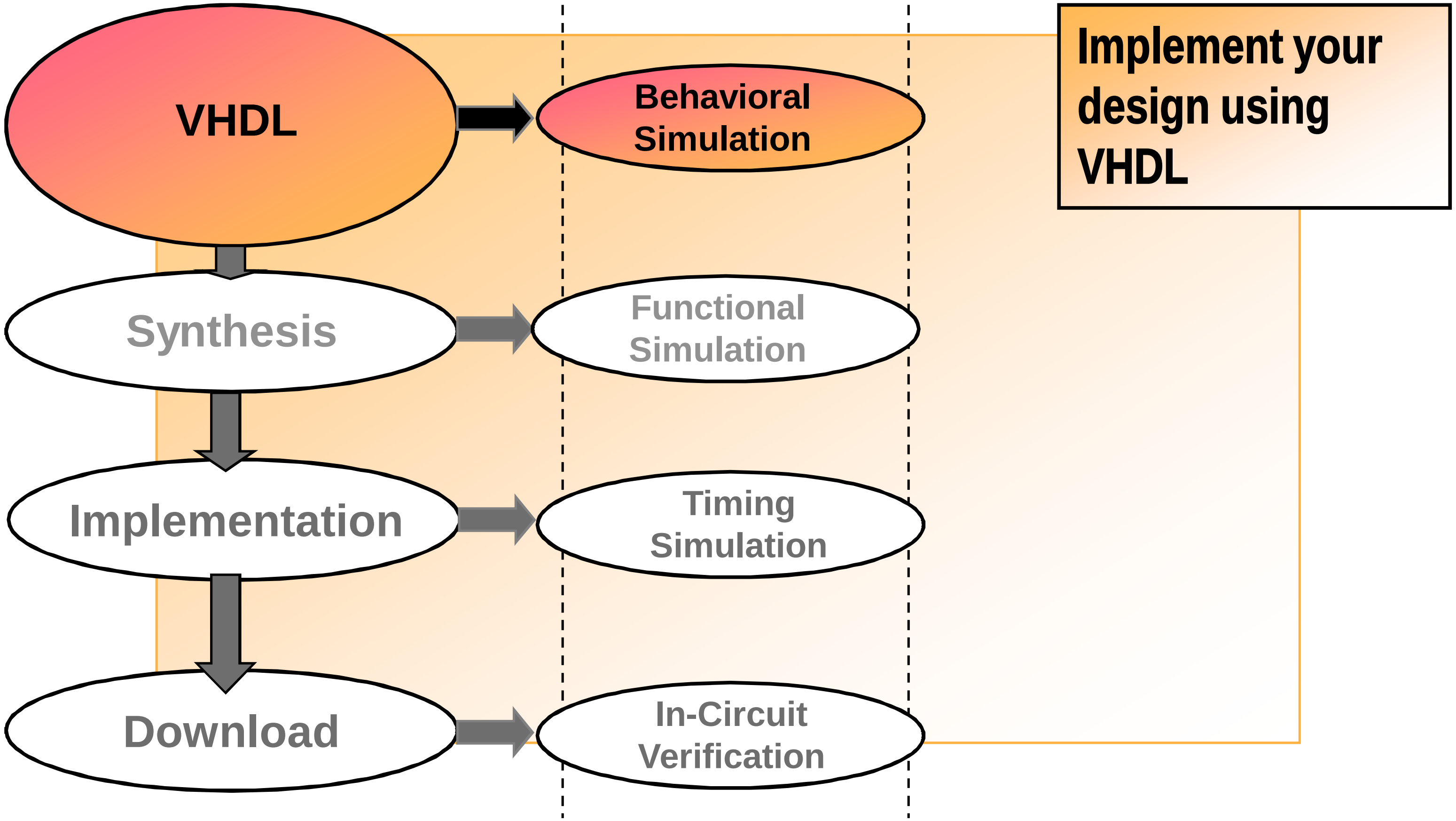
```
architecture RTL1 of RESOURCE is
begin
  seq : process (RSTn, CLOCK)
  begin
    if (RSTn = '0') then
      DOUT <= (others => '0');
    elsif (CLOCK'event and CLOCK = '1') then
      case SEL is
        when "00"   => DOUT <= unsigned(A) - 1;
        when "01"   => DOUT <= unsigned(B) - 1;
        when "10"   => DOUT <= unsigned(C) - 1;
        when others => DOUT <= unsigned(D) - 1;
      end case;
    end if;
  end process;
end RTL1;
```

Synthesized schematic for RTL1 of resource

delay 57 ns
area 65
number of flip-flops 16

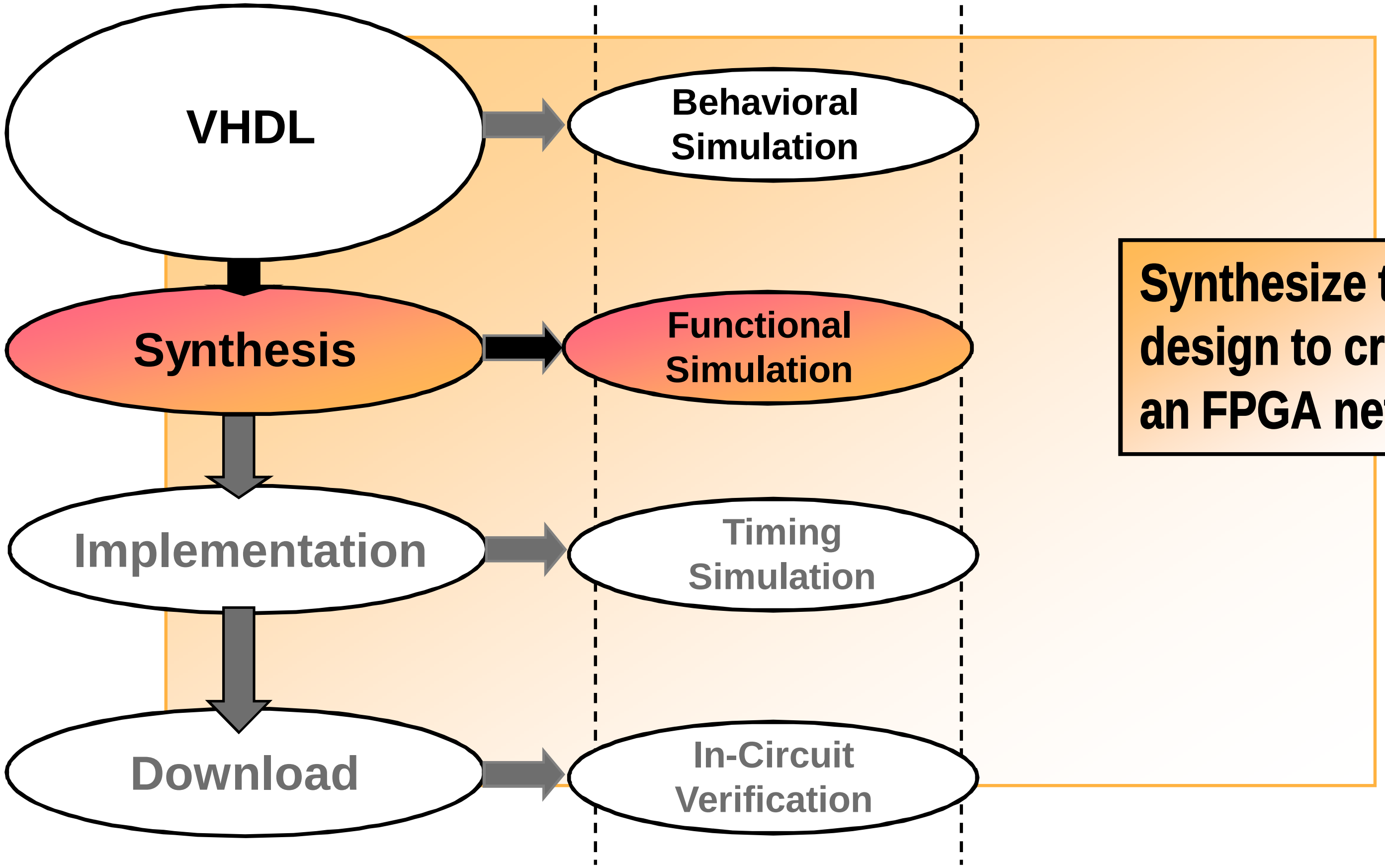


HDL Design Verification



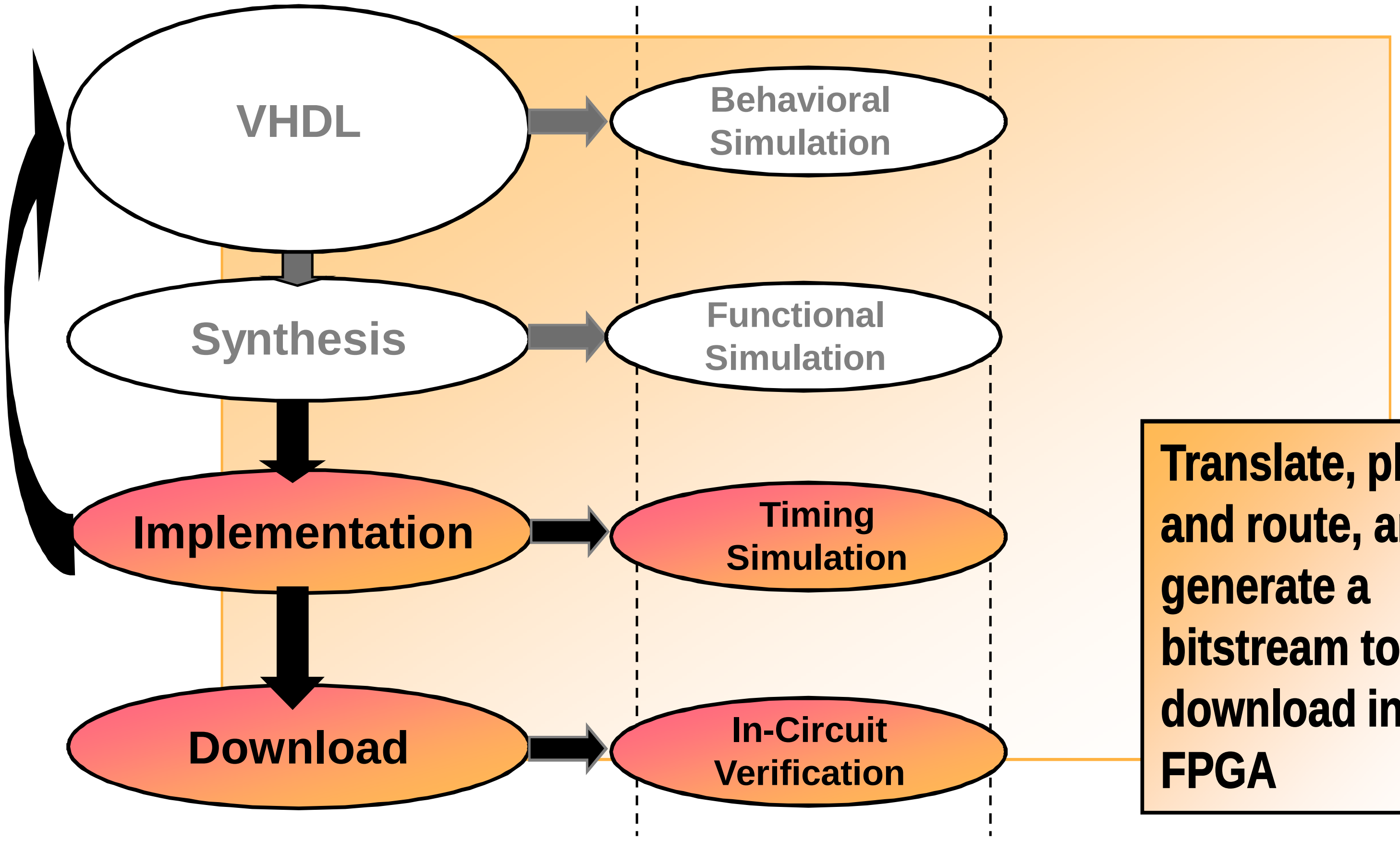
Implement your design using VHDL

Synthesis Design Verification



Synthesize the design to create an FPGA netlist

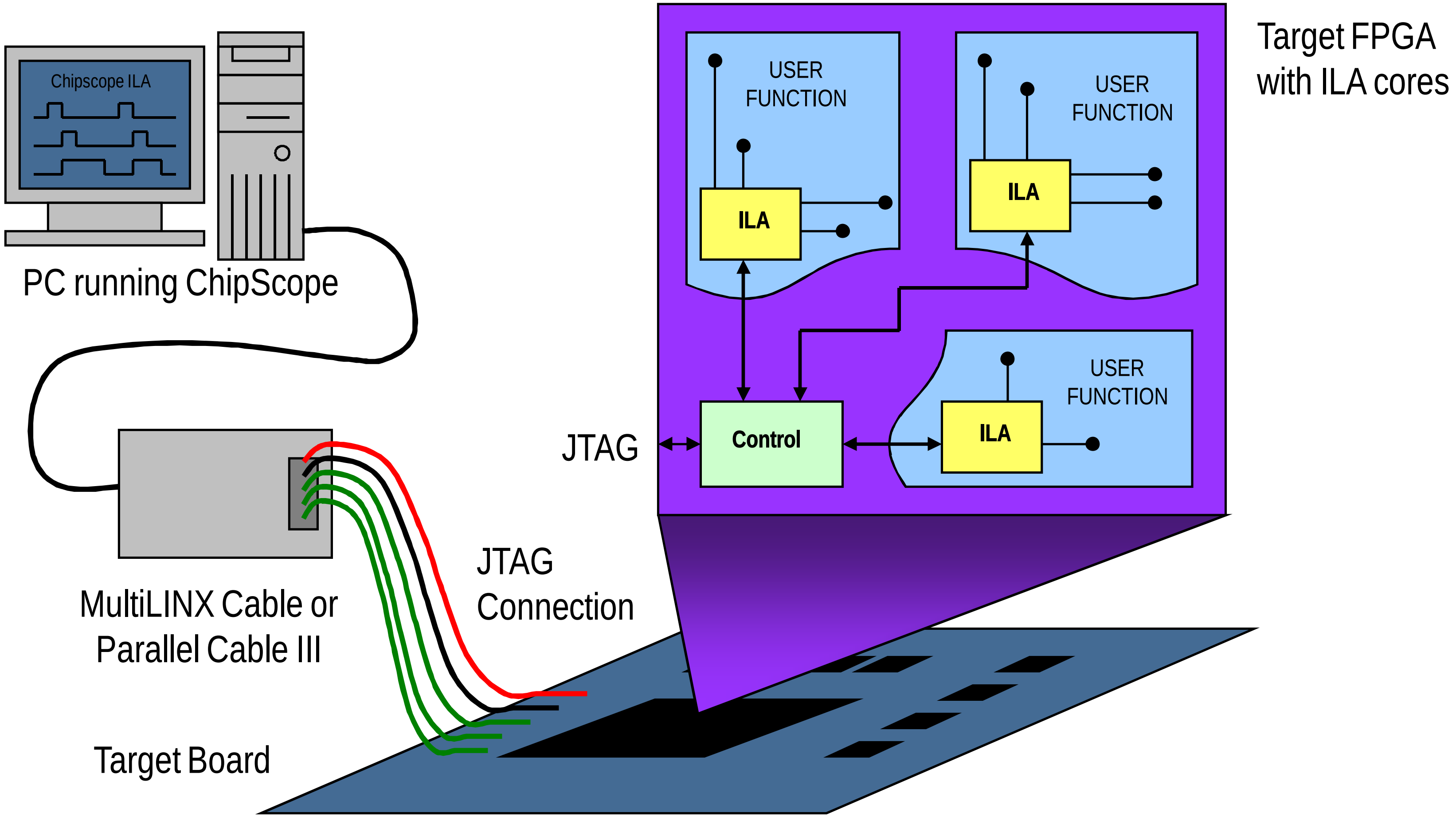
Implementation
Design Verification



Translate, place and route, and generate a bitstream to download in the FPGA

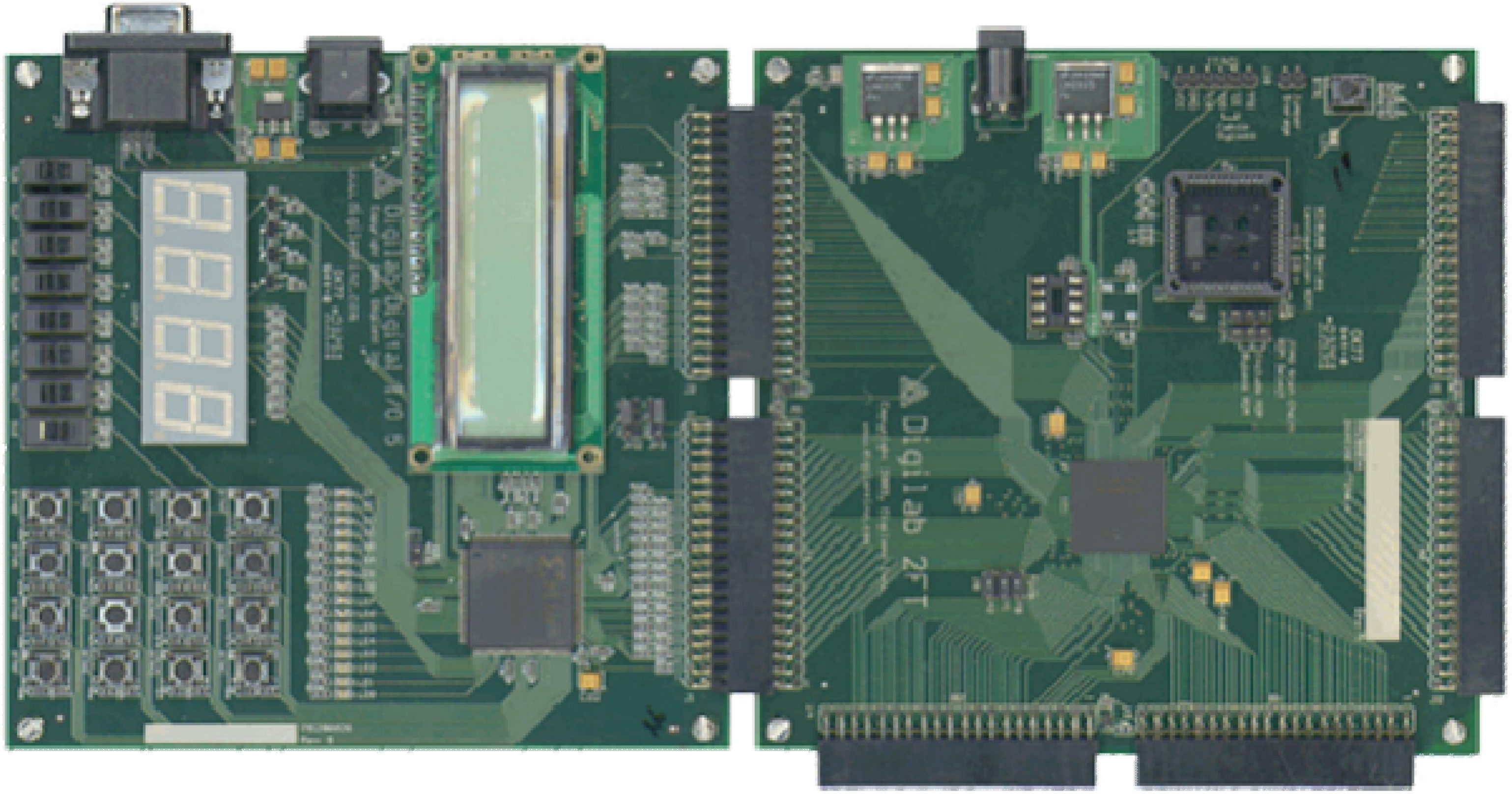
On-Chip Verification

ChipScope ILA System Diagram

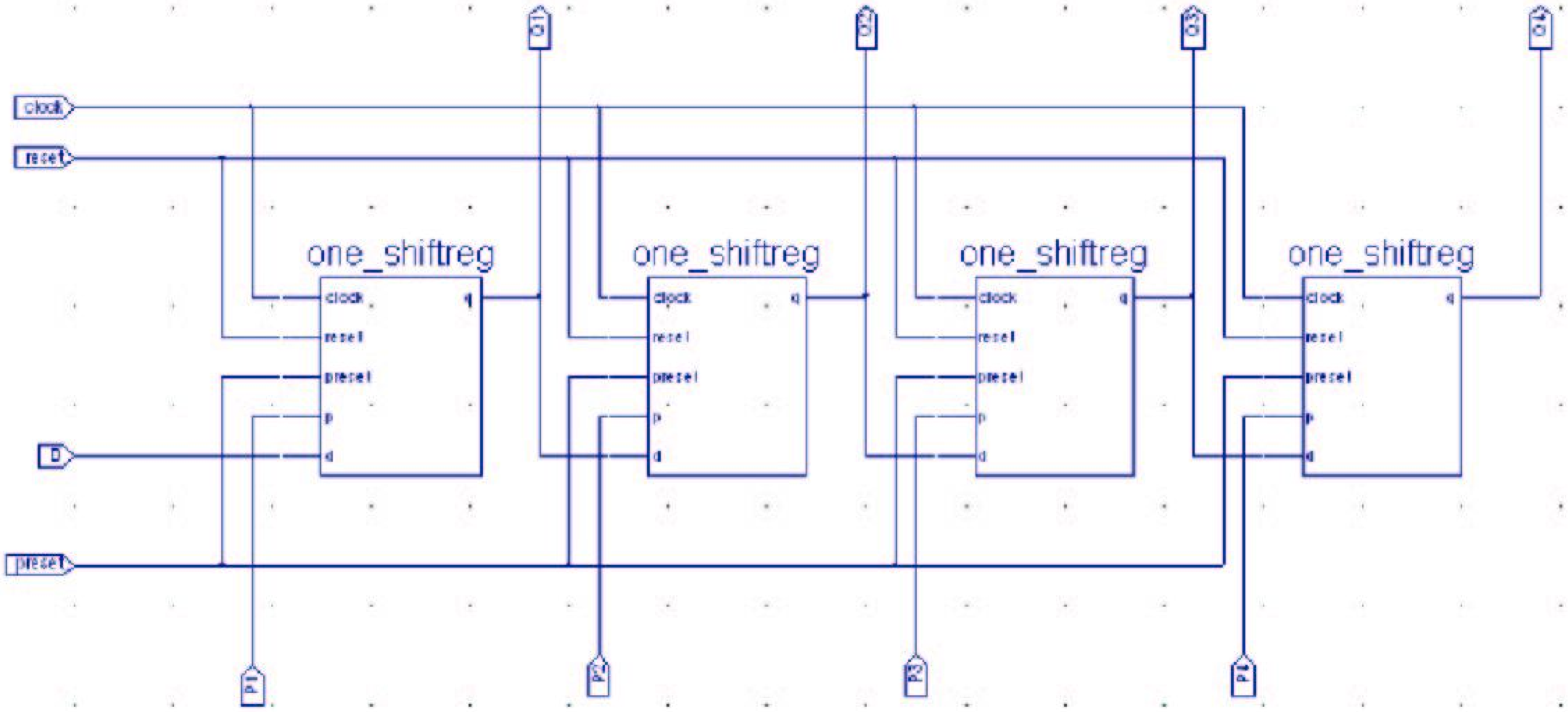


Digilab D2FT & DIO5 Boards

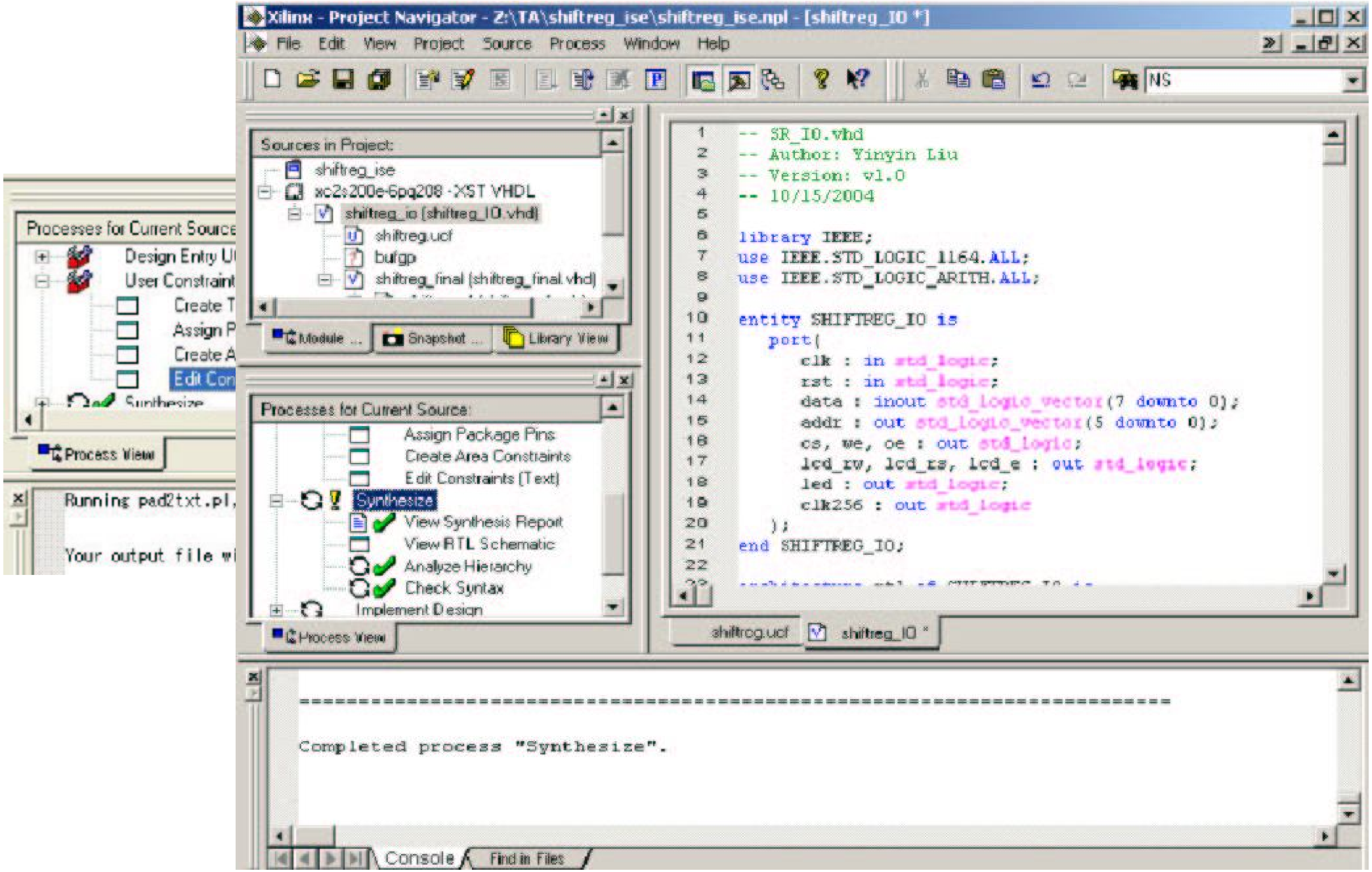
The Digilab 2FT/DIO5 board combination is an FPGA-based development platform with a large FPGA and I/O devices to support a wide range of digital circuits, including a complete computer system.



4-bit Shift Register



4-bit Shift Register



```
-- SR_IO.vhd
-- Author: Yinyin Liu
-- Version: v1.0
-- 10/15/2004

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;

entity SHIFTREG_IO is
    port(
        clk : in std_logic;
        rst : in std_logic;
        data : inout std_logic_vector(7 downto 0);
        addr : out std_logic_vector(5 downto 0);
        cs, we, oe : out std_logic;
        lcd_rw, lcd_rs, lcd_e : out std_logic;
        led : out std_logic;
        clk256 : out std_logic
    );
end SHIFTREG_IO;
```

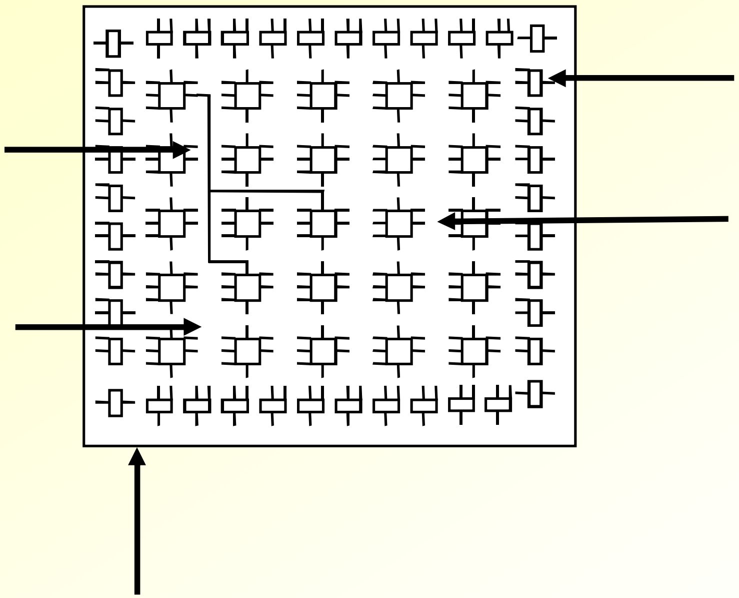
```
==========================================================================

Completed process "Synthesize".
```

```
Running pad2txt.pl,

Your output file wi
```

Xilinx FPGA Architecture



Programmable Interconnect

Tristate Buffers

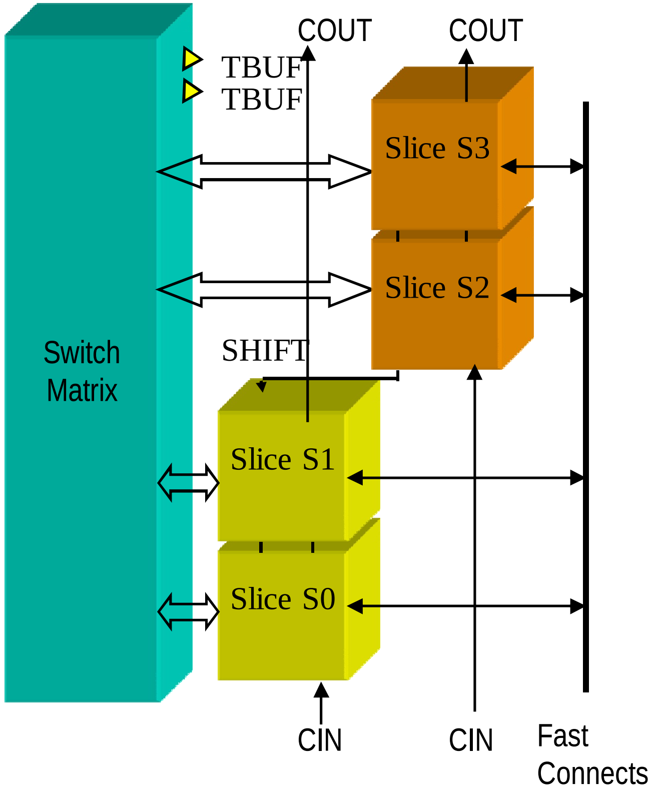
I/O Blocks (IOBs)

Configurable Logic Blocks (CLBs)

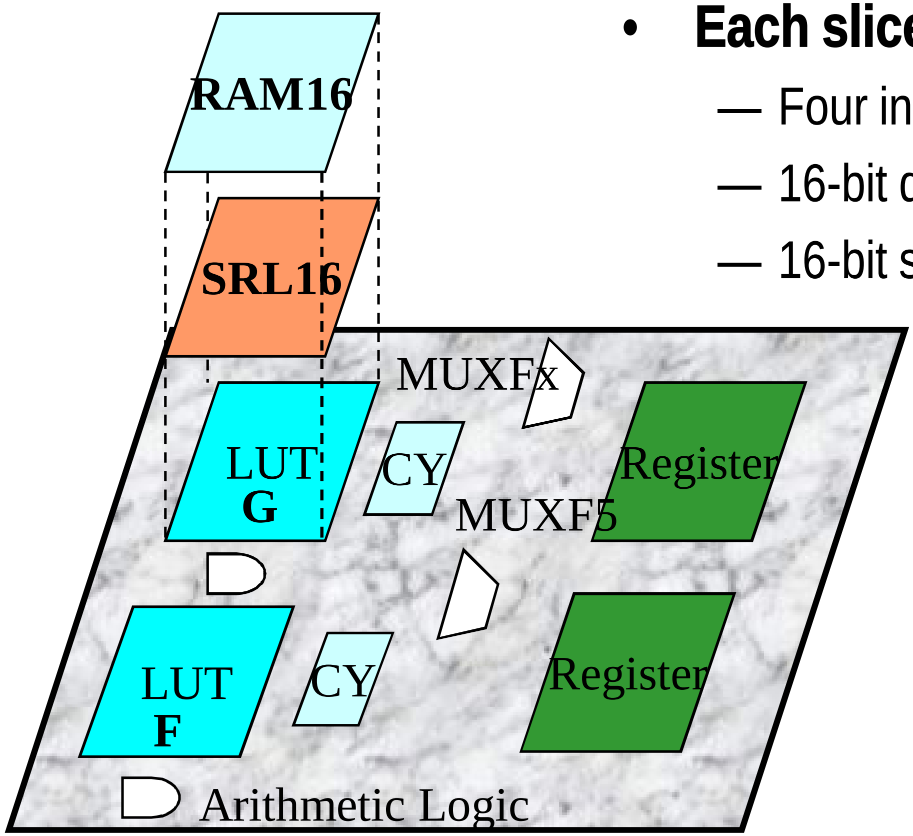
Global Resources

CLB

- **Flexible resources**
  - — Wide-input functions
    - 16:1 multiplexer in 1 CLB
  - — Fast arithmetic functions
    - Two dedicated carry chains
  - — Cascadable shift registers in LUT
    - 128-b shift register in 1 CLB
- **Ease of Performance**
  - — Direct routing enabling high speed



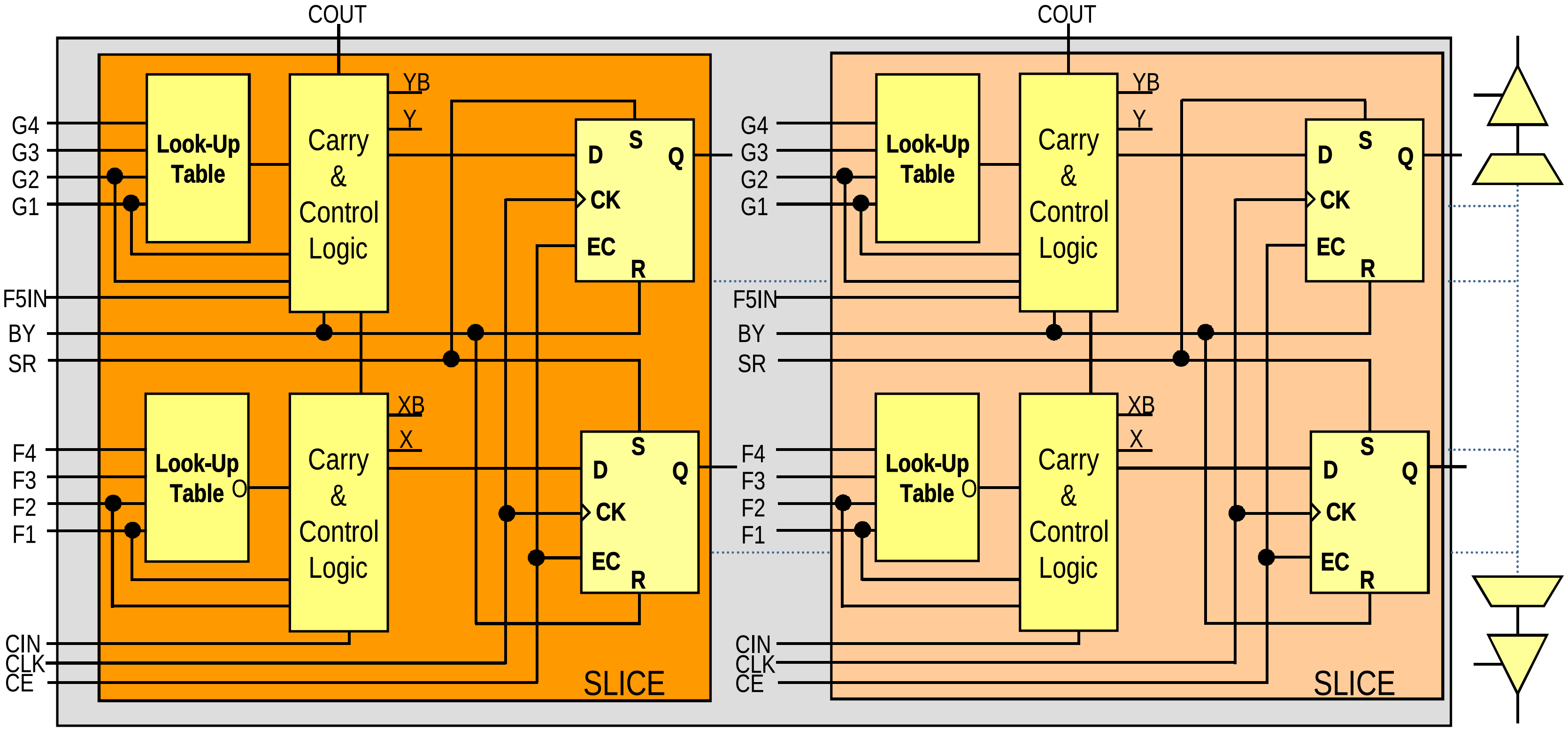
Slice



- **Each slice contains two:**
  - — Four inputs lookup tables
  - — 16-bit distributed SelectRAM
  - — 16-bit shift register

- **Each register:**
  - — D flip-flop
  - — Latch
- **Dedicated logic:**
  - — Muxes
  - — Arithmetic logic
    - — MULT_AND
    - — Carry Chain

CLB Structure



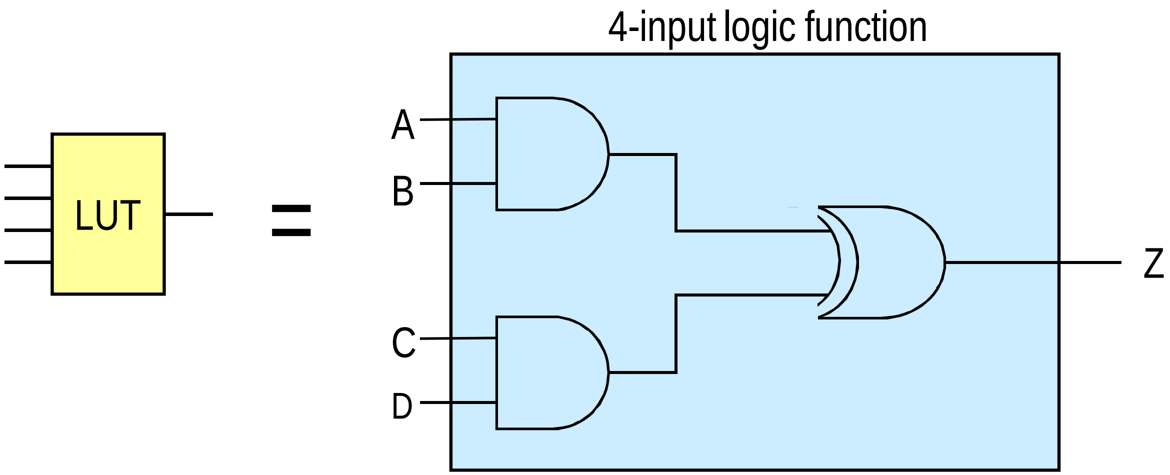
- Each slice has 2 LUT-FF pairs with associated carry logic
- Two 3-state buffers (BUFT) associated with each CLB, accessible by all CLB outputs

Four-Input LUT

- Implements combinatorial logic
  - Any 4-input logic function
  - Cascaded for wide-input functions

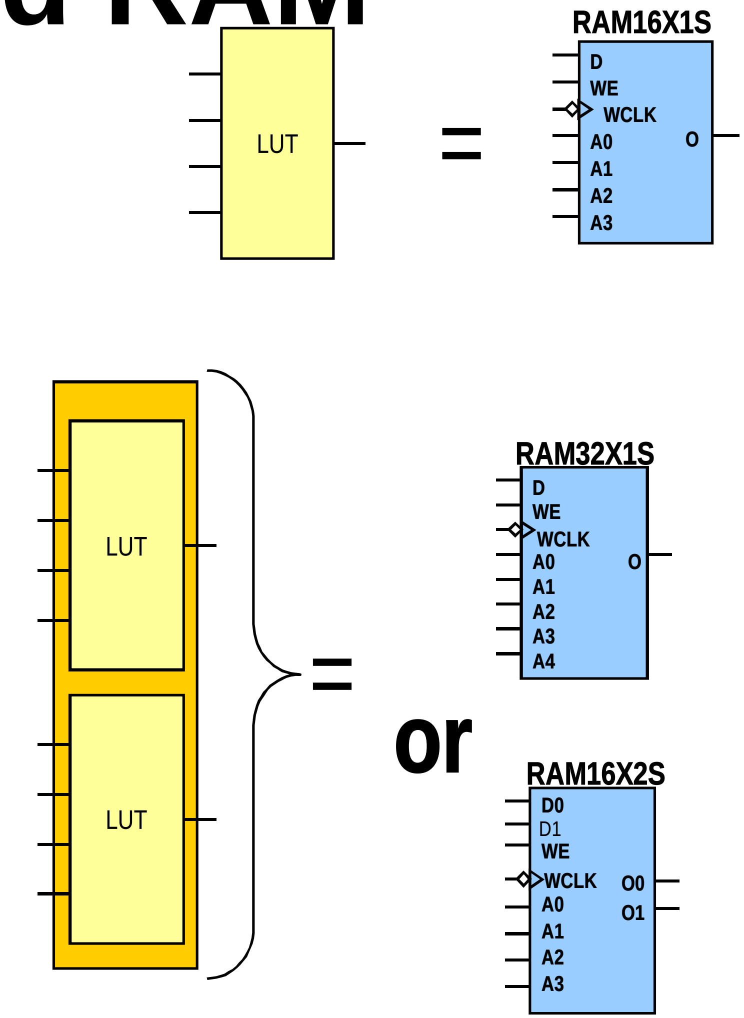
Truth Table

Inputs(ABCD)	Output(Z)
0000	0
0001	0
0010	1
0011	0
……	..
1110	1
1111	1



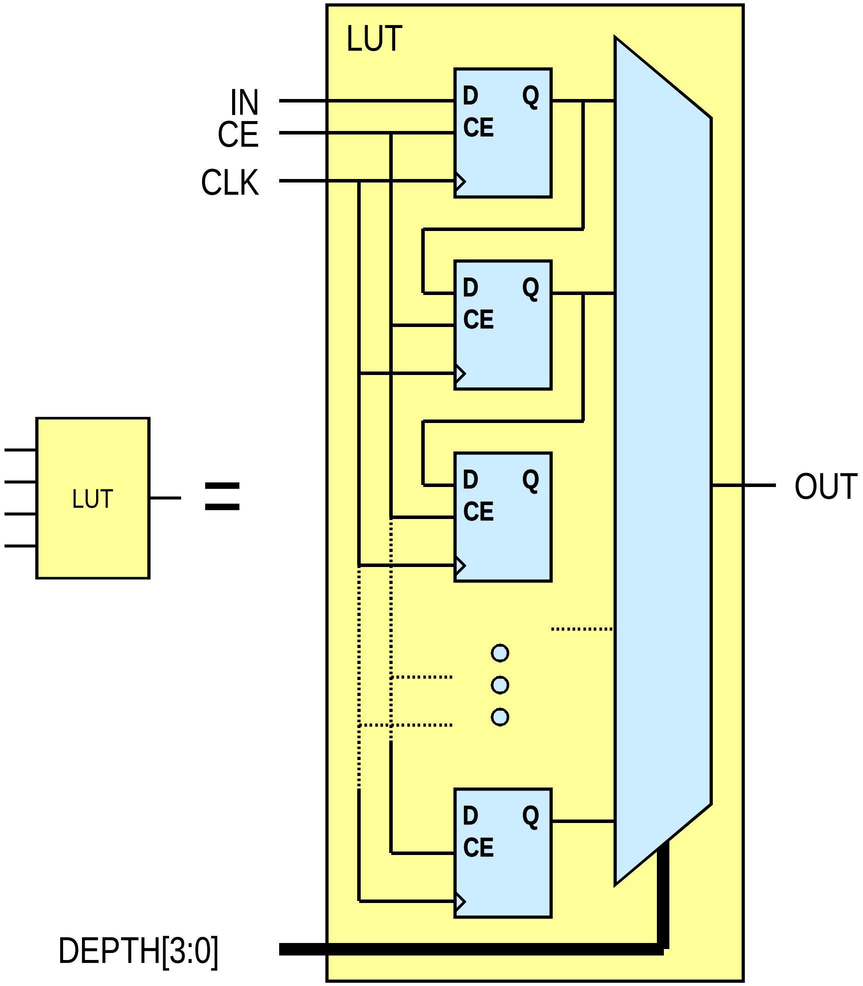
Distributed RAM

- CLB LUT configurable as Distributed RAM
  - A LUT equals 16x1 RAM
  - Implements Single and Dual-Ports
  - Cascade LUTs to increase RAM size
- Synchronous write
- Synchronous/Asynchronous read
  - Accompanying flip-flops used for synchronous read

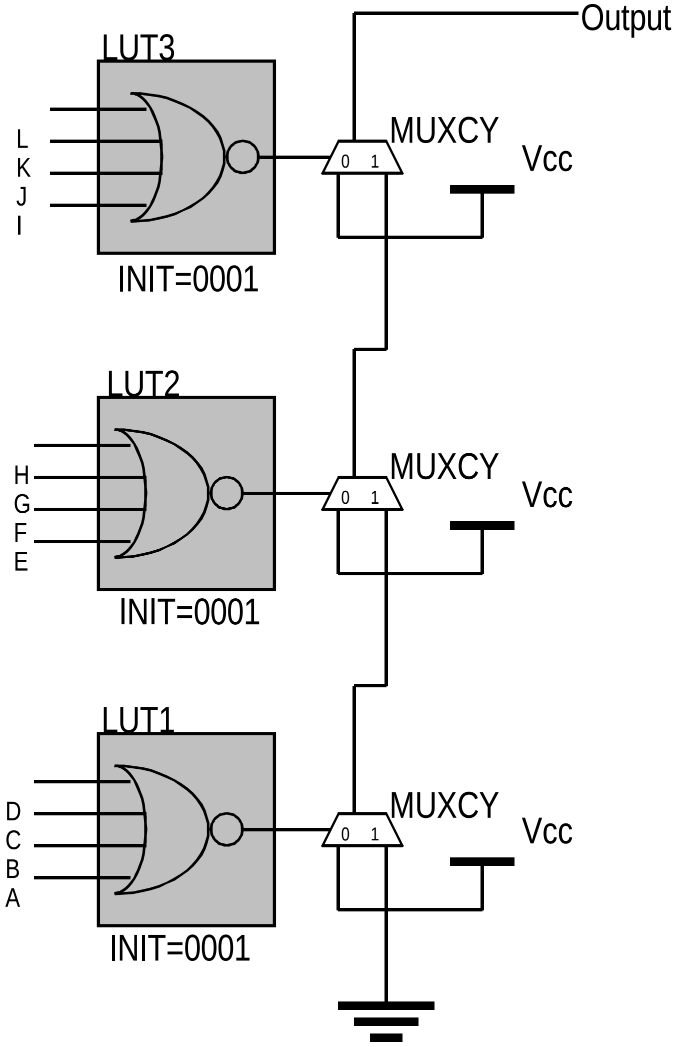


Shift Register

- Each LUT can be configured as shift register
  - Serial in, serial out
- Dynamically addressable delay up to 16 cycles
- For programmable pipeline
- Cascade for greater cycle delays
- Use CLB flip-flops to add depth



12- Input OR Function

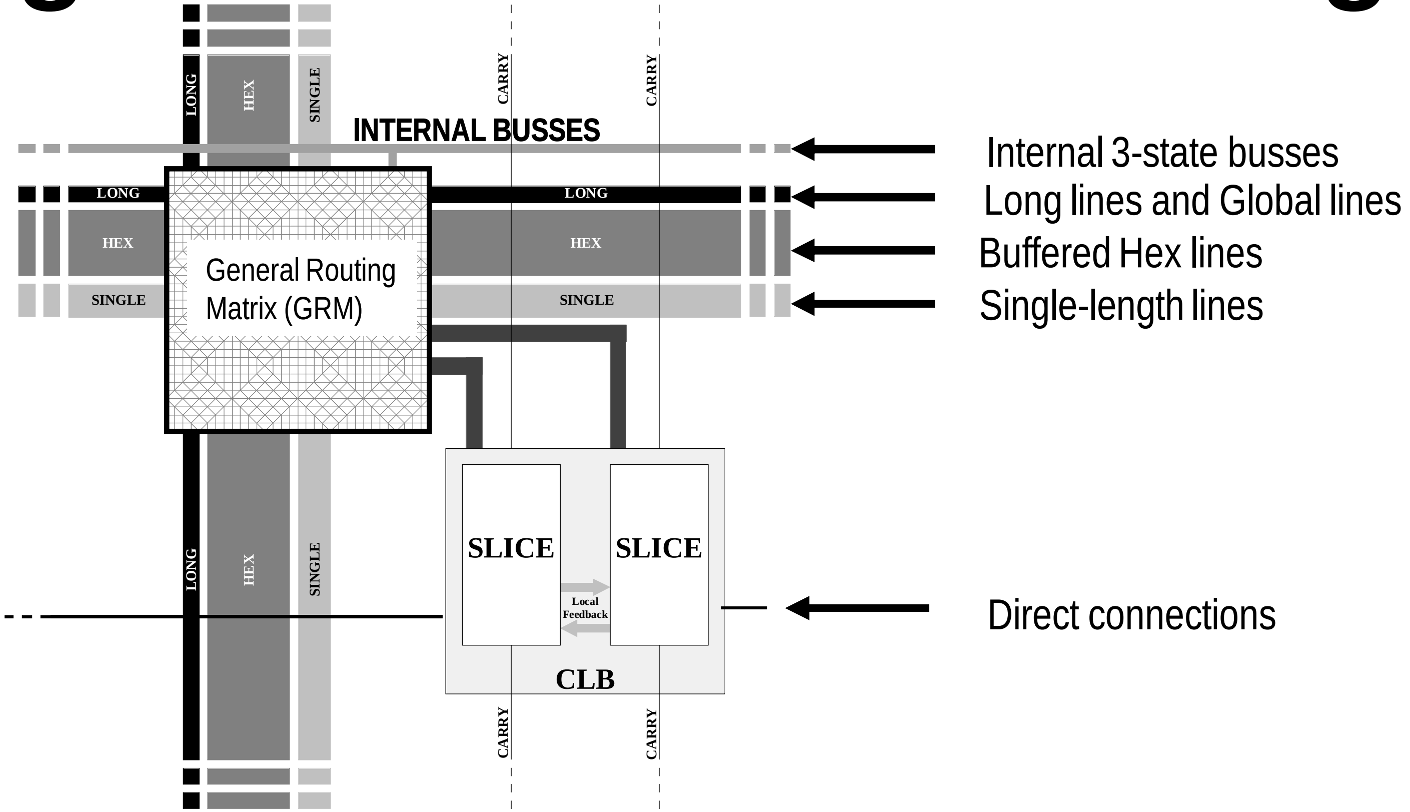


4-Input NOR Truth Table

Inputs(ABCD)	Output(Z)	Output(HEX)
0000	1	1
0001	0	
0010	0	
0011	0	
……	..	..
1011	0	..
1100	0	0
1101	0	
1110	0	
1111	0	

- Utilization
  - 3 LUTs and 3 MUXCYs
  - As opposed to 4 LUTs
- Performance
  - 1 logic level
  - As opposed to 2 logic levels

High-Performance Routing



Internal 3-state busses

Long lines and Global lines

Buffered Hex lines

Single-length lines

Direct connections

- Local routing
  - Direct connections
- General Routing Matrix (GRM)
  - Single line, Long line, Hex line

- Dedicated routing
  - Internal 3-state bus
- Global routing
  - Primary Clock Buffer lines, Secondary lines

Improved Clock-to-out Using DLL

- Spartan-II clock-to-out delays reduced over 50%

Output standard = LVTTL Fast 16mA

(OBUF_F_16)

Temp=room, Vdd=2.5V, Vcco=3.3V

Waveforms:

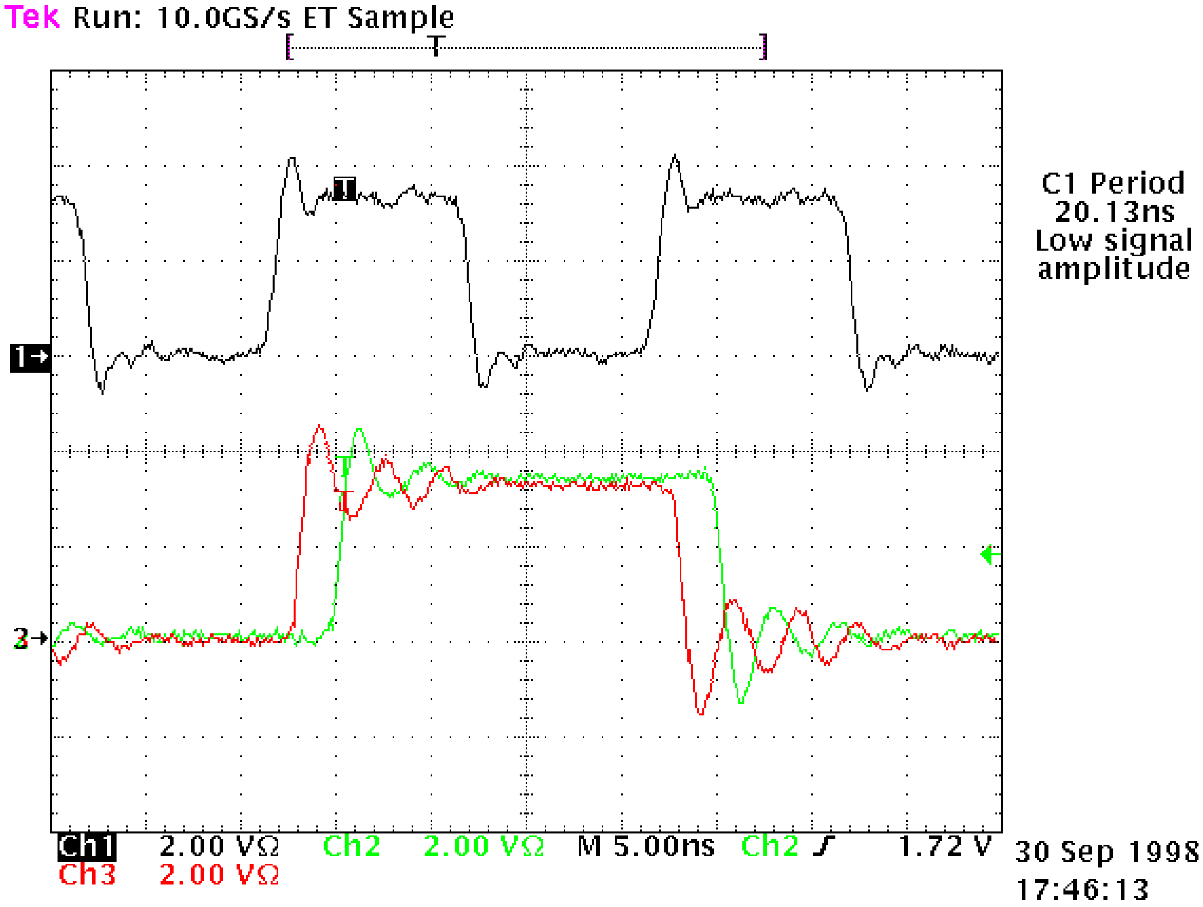
1: CLKIN

2: DATA OUT (no DLL)

3: DATA OUT (DLL deskewed)

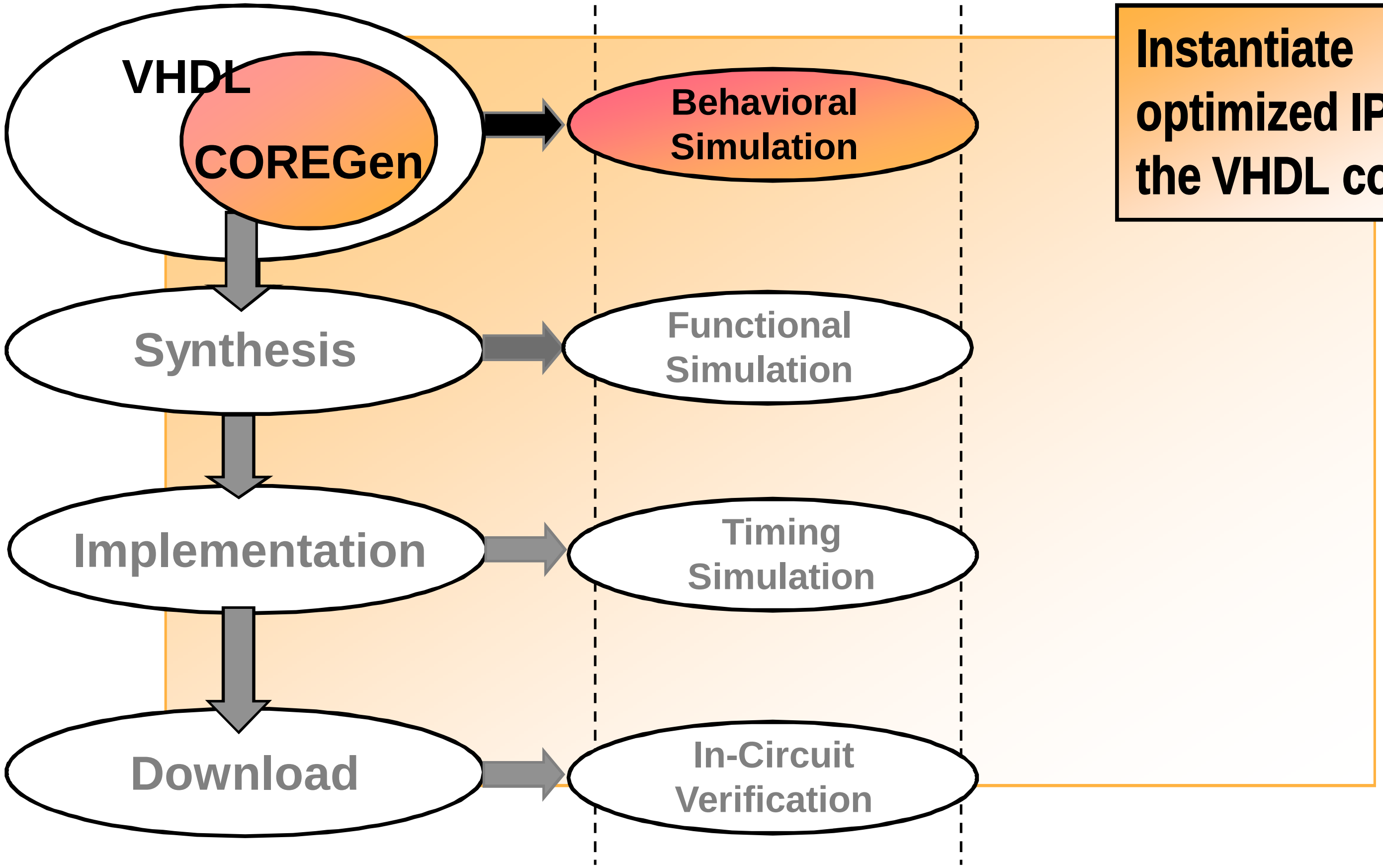
Timing

w/o DLL		w/ DLL	
r->r	r->f	r->r	r->f
3.6n	3.5n	1.4n	1.4n



CORE Generator

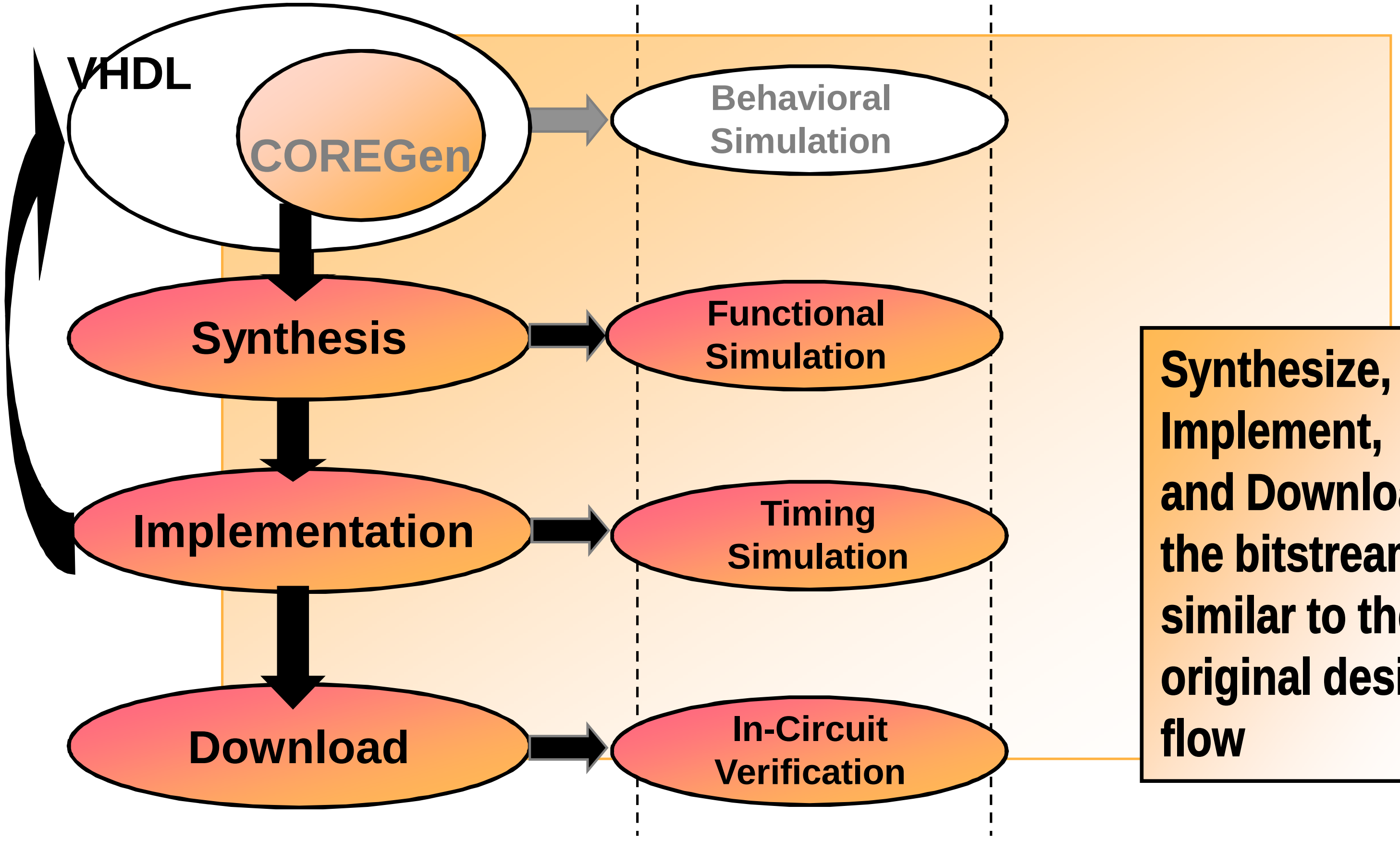
Design Verification



Instantiate optimized IP within the VHDL code

Synthesize, Implement, Download

Design Verification



Synthesize, Implement, and Download the bitstream, similar to the original design flow

Xilinx IP Solutions

DSP Functions

$P Additive White Gaussian Noise (AWGN)
$P Reed Solomon
$ 3GPP Turbo Code
$P Viterbi Decoder
P Convolution Encoder
$P Interleaver/De-interleaver
P LFSR
P 1D DCT
P 2D DCT
P DA FIR
P MAC
P MAC-based FIR filter
Fixed FFTs 16, 64, 256, 1024 points
P FFT 16- to 16384- points
P FFT - 32 Point
P Sine Cosine Look-Up Tables
$P Turbo Product Code (TPC)
P Direct Digital Synthesizer
P Cascaded Integrator Comb
P Bit Correlator
P Digital Down Converter

Math Functions

P Multiplier Generator
- Parallel Multiplier
- Dyn Constant Coefficient Mult
- Serial Sequential Multiplier
- Multiplier Enhancements
P Pipelined Divider
P CORDIC

Base Functions

P Binary Decoder
P Twos Complement
P Shift Register RAM/FF
P Gate modules
P Multiplexer functions
P Registers, FF & latch based
P Adder/Subtractor
P Accumulator
P Comparator
P Binary Counter

Memory Functions

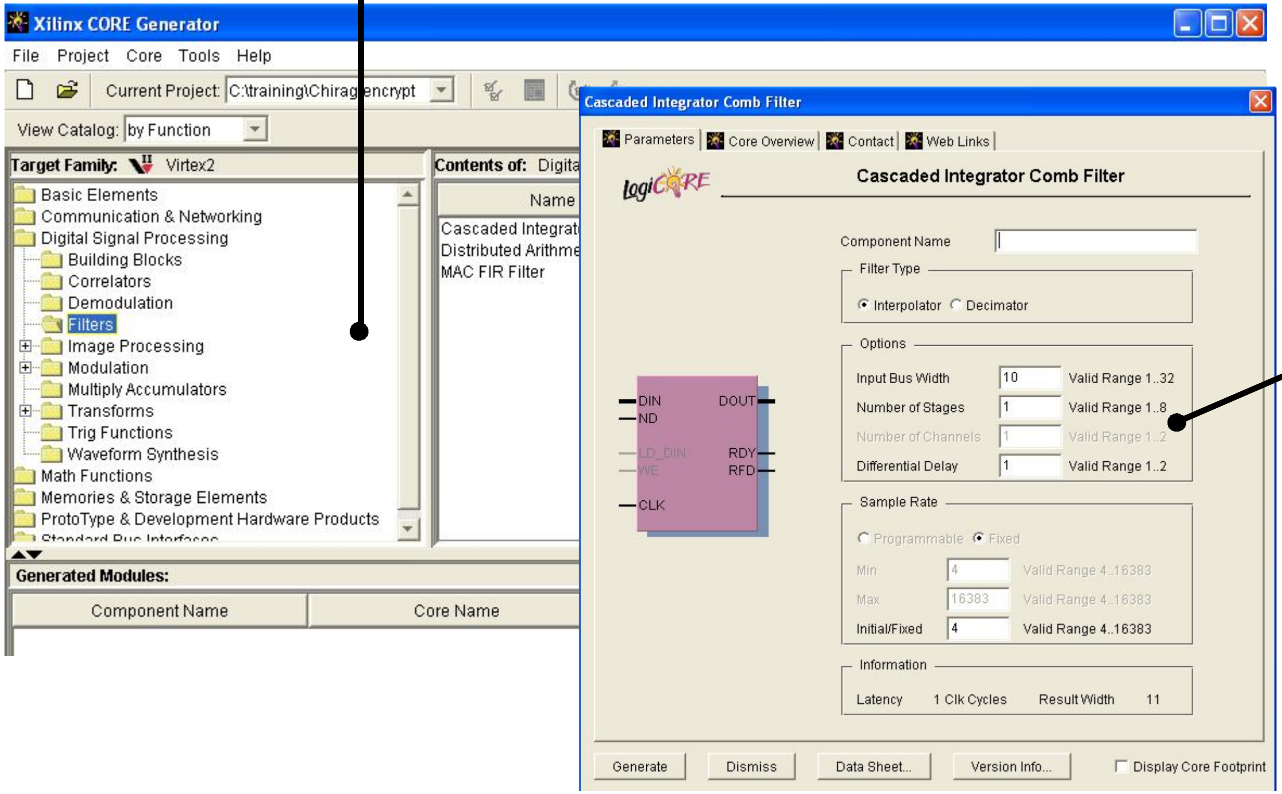
P Asynchronous FIFO
P Block Memory modules
P Distributed Memory
P Distributed Mem Enhance
P Sync FIFO (SRL16)
P Sync FIFO (Block RAM)
P CAM (SRL16)
P CAM (Block RAM)

IP CENTER
http://www.xilinx.com/ipcenter

Key: $ = License Fee, P = Parameterized, S = Project License Available,
BOLD = Available in the Xilinx Blockset for the System Generator for DSP

Xilinx CORE Generator

List of available IP from LogiCORE or Alliance CORE



Current Project: C:\training\Chirag\encrypt

View Catalog: by Function

Target Family: Virtex2

- Basic Elements
- Communication & Networking
- Digital Signal Processing
  - Building Blocks
  - Correlators
  - Demodulation
  - Filters
  - Image Processing
  - Modulation
  - Multiply Accumulators
  - Transforms
  - Trig Functions
  - Waveform Synthesis
- Math Functions
- Memories & Storage Elements
- ProtoType & Development Hardware Products
- Standard Bus Interfaces

Contents of: Digita

Name
Cascaded Integrat
Distributed Arithme
MAC FIR Filter

Generated Modules:

Component Name	Core Name

Cascaded Integrator Comb Filter

Cascaded Integrator Comb Filter

Component Name

Filter Type: ● Interpolator ○ Decimator

Options

Input Bus Width	10	Valid Range 1..32
Number of Stages	1	Valid Range 1..8
Number of Channels	1	Valid Range 1..2
Differential Delay	1	Valid Range 1..2

Sample Rate: ○ Programmable ● Fixed

Min	4	Valid Range 4..16383
Max	16383	Valid Range 4..16383
Initial/Fixed	4	Valid Range 4..16383

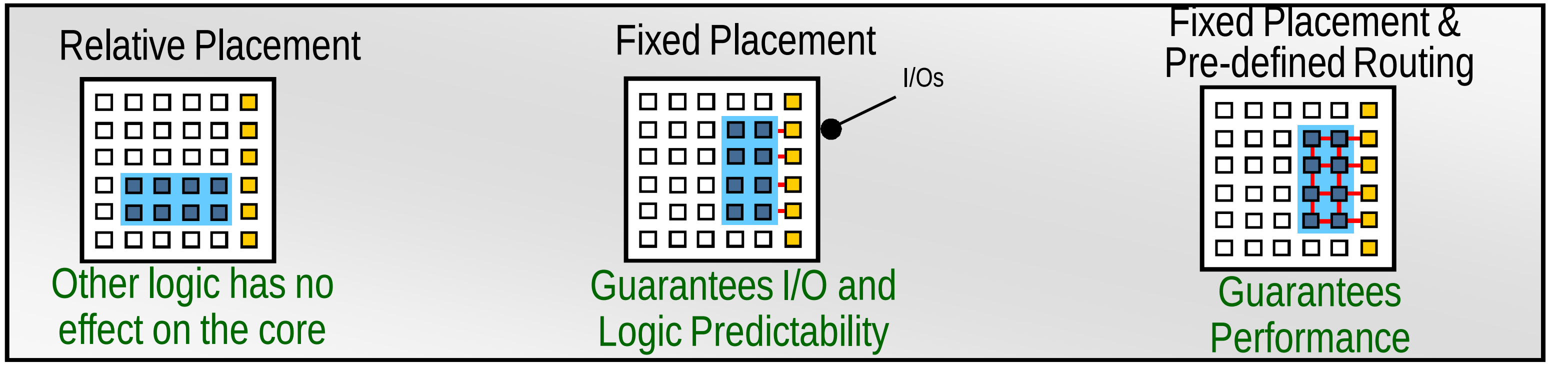
Information

Latency 1 Clk Cycles Result Width 11

Fully Parameterizable

Xilinx Smart-IP Technology

- Pre-defined placement and routing enhances performance and predictability



Relative Placement

Other logic has no effect on the core

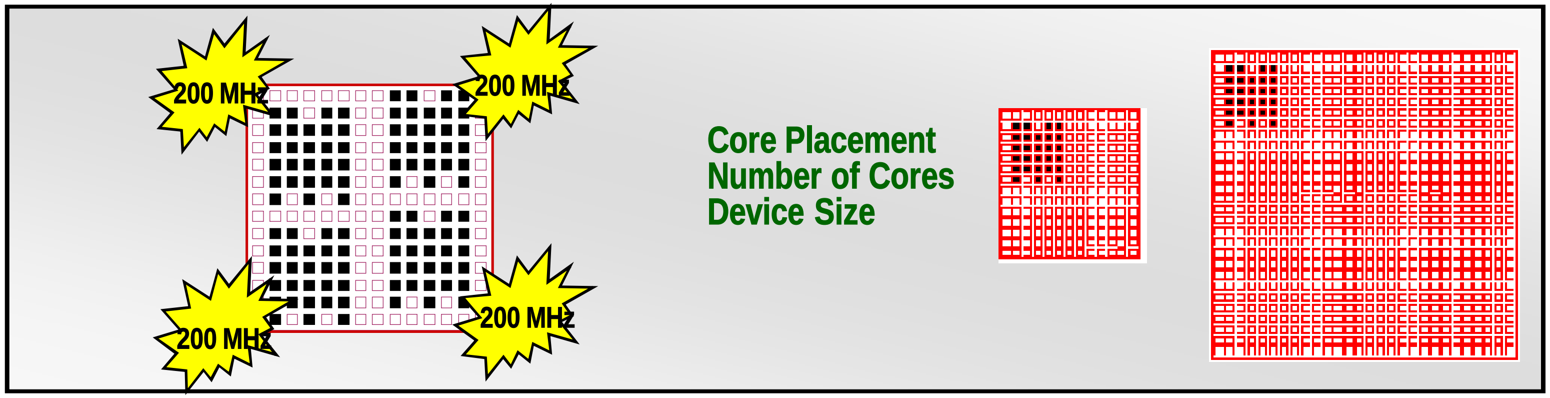
Fixed Placement

Guarantees I/O and Logic Predictability

Fixed Placement & Pre-defined Routing

Guarantees Performance

- Performance is independent of:



Core Placement
Number of Cores
Device Size

MATLAB

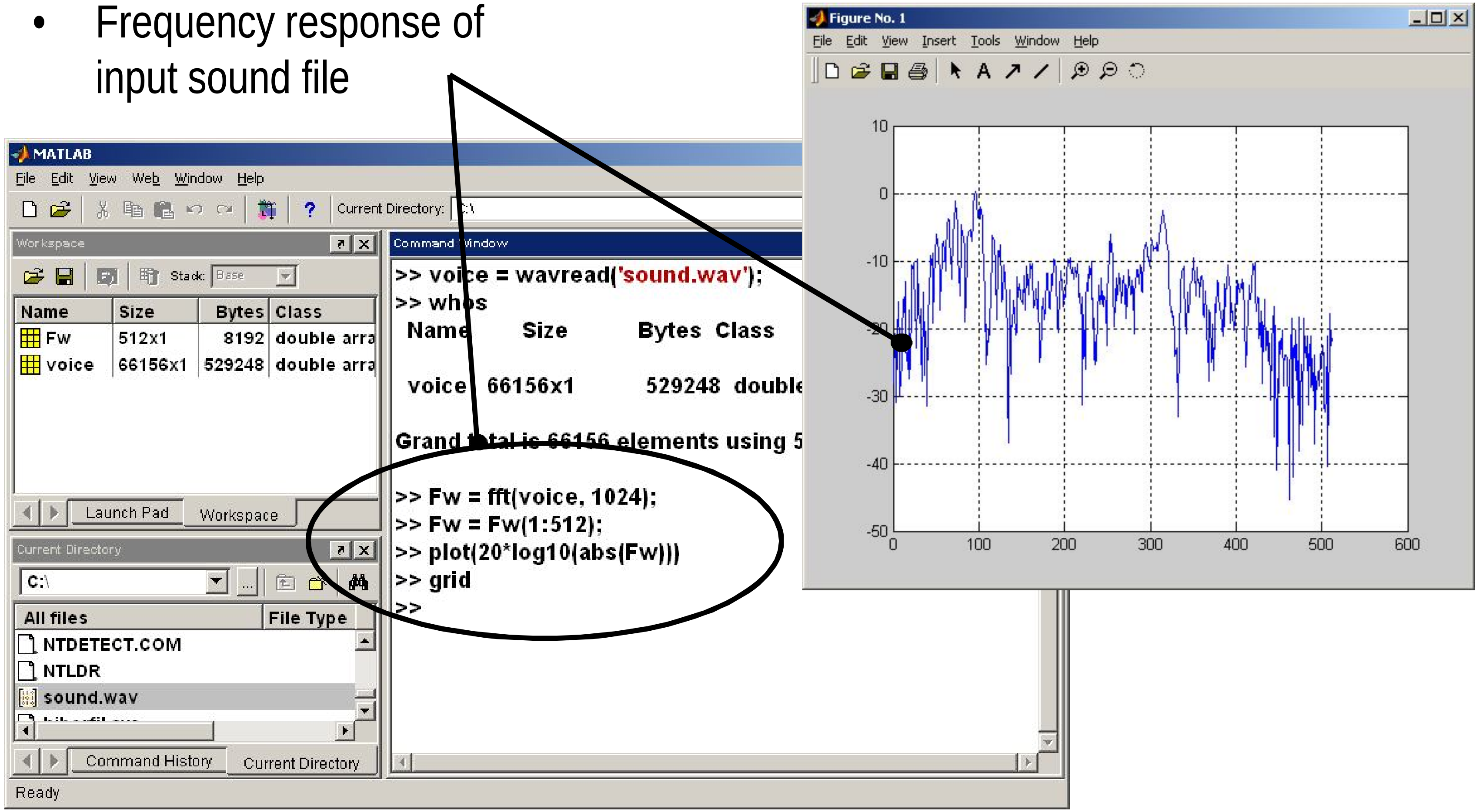
The MathWorks

- MATLAB™, the most popular system design tool, is a programming language, interpreter, and modeling environment
  - Extensive libraries for math functions, signal processing, DSP, communications, and much more
  - Visualization: large array of functions to plot and visualize your data and system/design
  - Open architecture: software model based on base system and domain-specific plug-ins

MATLAB

The MathWorks

- Frequency response of input sound file



Name	Size	Bytes	Class
Fw	512x1	8192	double arra
voice	66156x1	529248	double arra

All files	File Type
NTDETECT.COM	
NTLDR	
sound.wav	

```
>> voice = wavread('sound.wav');
>> whos
  Name       Size          Bytes  Class

  voice    66156x1         529248  double

Grand total is 66156 elements using 5

>> Fw = fft(voice, 1024);
>> Fw = Fw(1:512);
>> plot(20*log10(abs(Fw)))
>> grid
>>
```

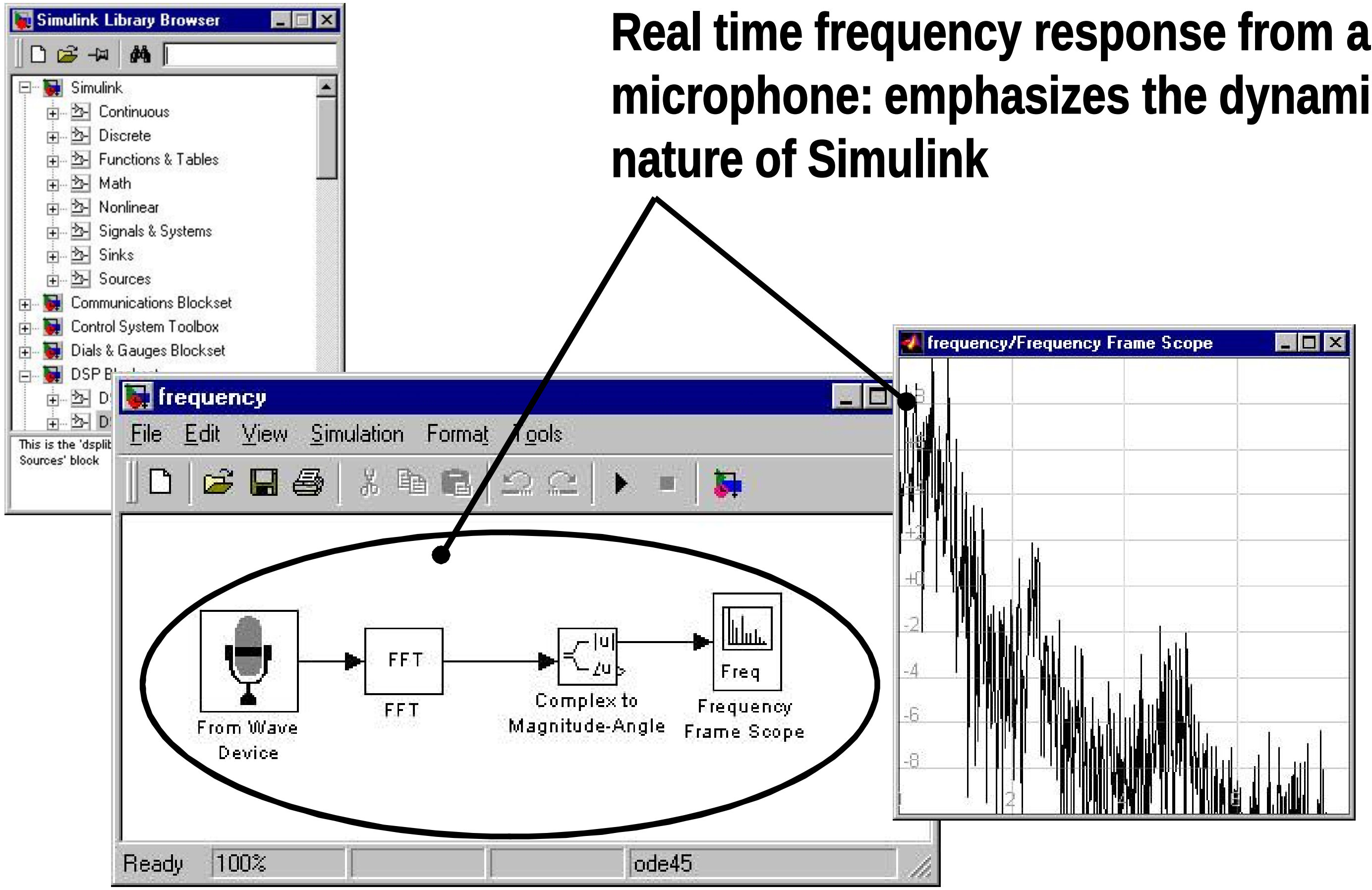
Simulink

The MathWorks

- Simulink™ - Visual data flow environment for modeling and simulation of dynamical systems
  - Fully integrated with the MATLAB engine
  - Graphical block editor
  - Event-driven simulator
  - Models parallelism
  - Extensive library of parameterizable functions
    - Simulink Blockset - math, sinks, sources
    - DSP Blockset - filters, transforms, etc.
    - Communications Blockset - modulation, DPCM, etc.

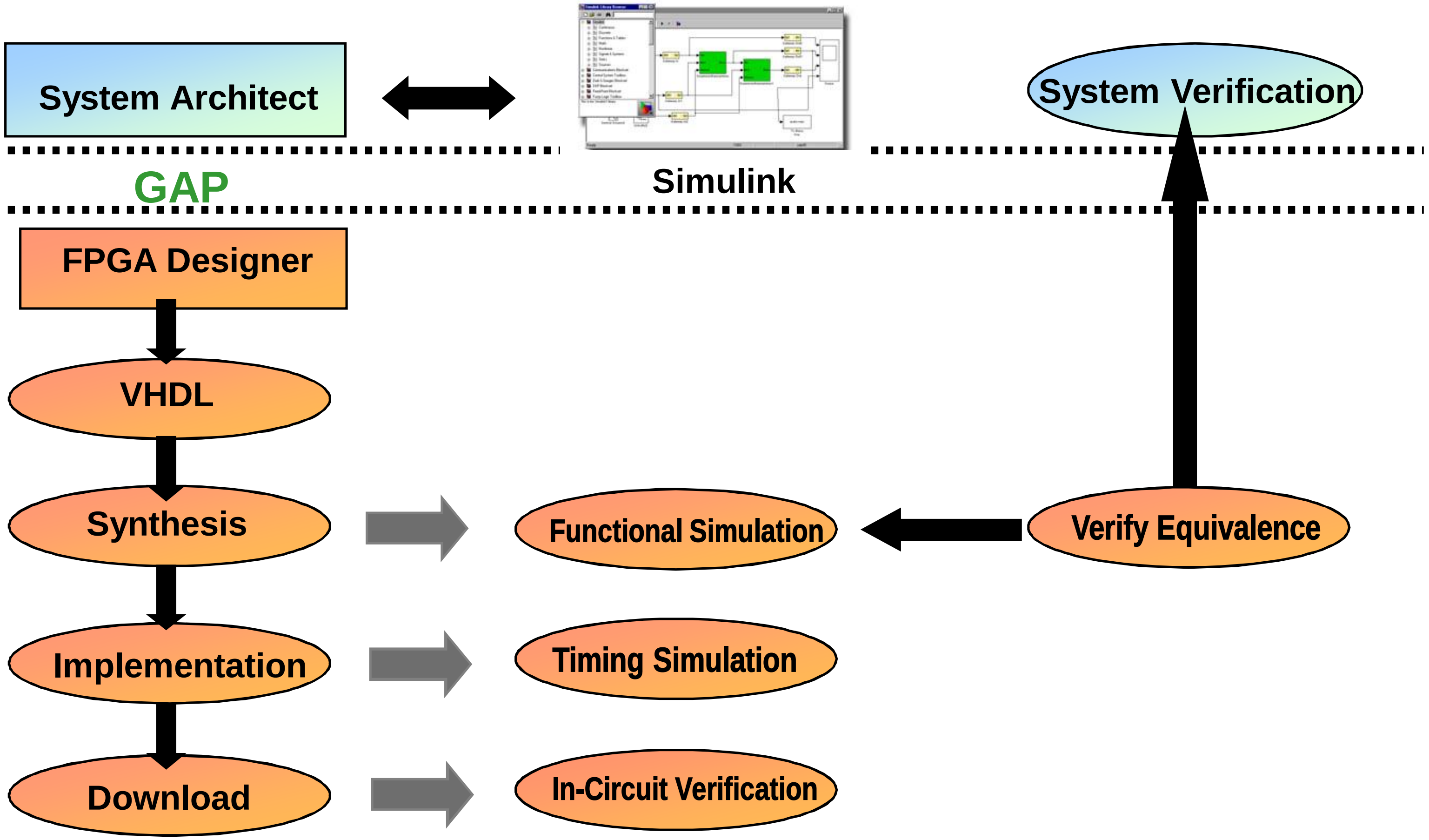
MATLAB/Simulink

The MathWorks



Real time frequency response from a microphone: emphasizes the dynamic nature of Simulink

Traditional Simulink FPGA Flow



System Architect

Simulink

System Verification

GAP

FPGA Designer

VHDL

Synthesis

Implementation

Download

Functional Simulation

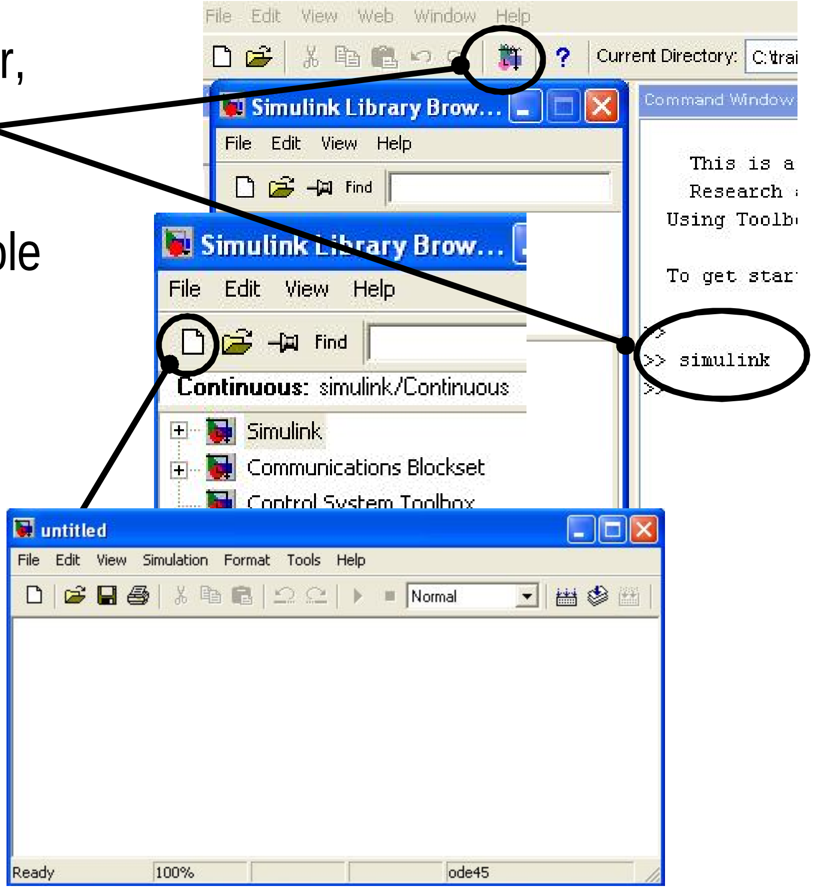
Timing Simulation

In-Circuit Verification

Verify Equivalence

Creating a System Generator Design

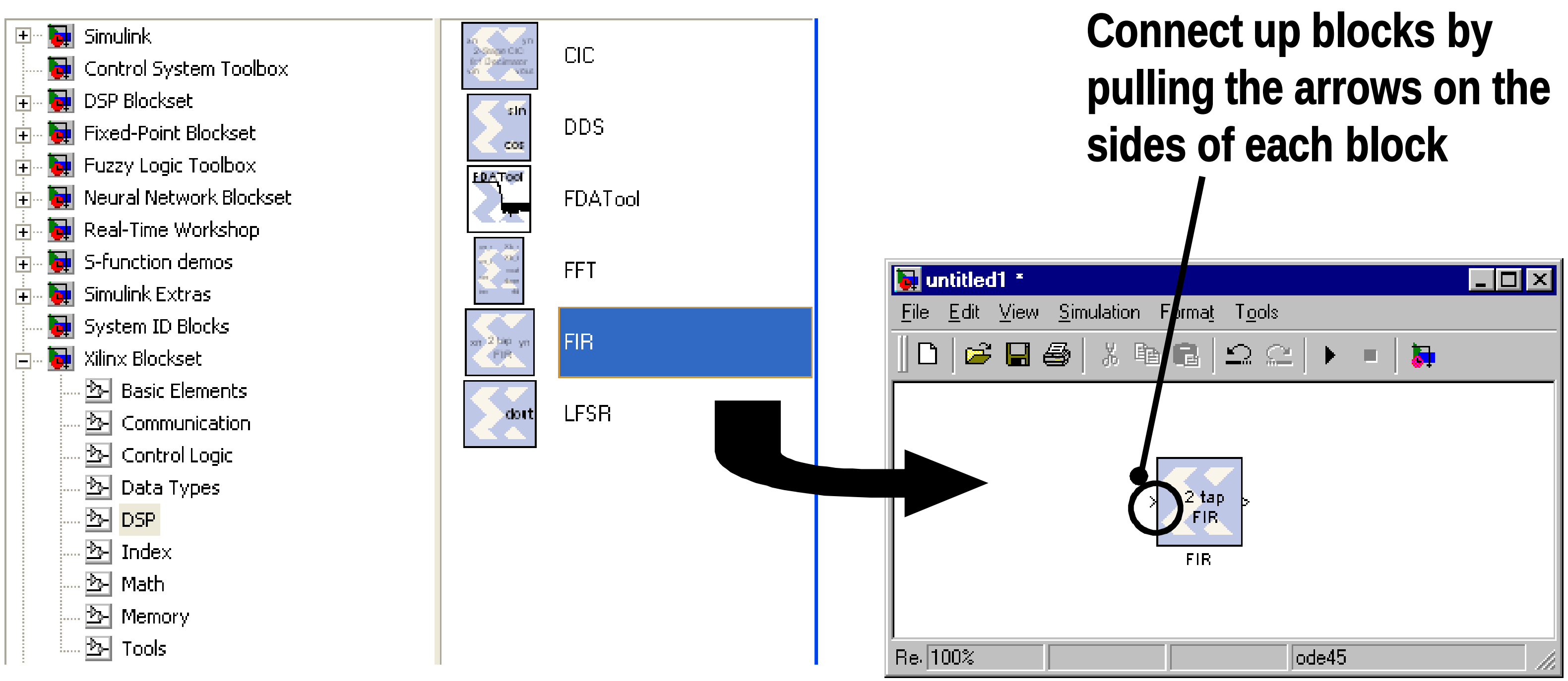
- Invoke Simulink library browser
- To open the Simulink library browser, click the Simulink library browser button
  or type “Simulink” in MATLAB console
- The library browser contains all the blocks available to designers
- Start a new design by clicking the new sheet button



```
This is a
Research
Using Toolb
To get star
>>
>> simulink
>>
```

Creating a System Generator Design

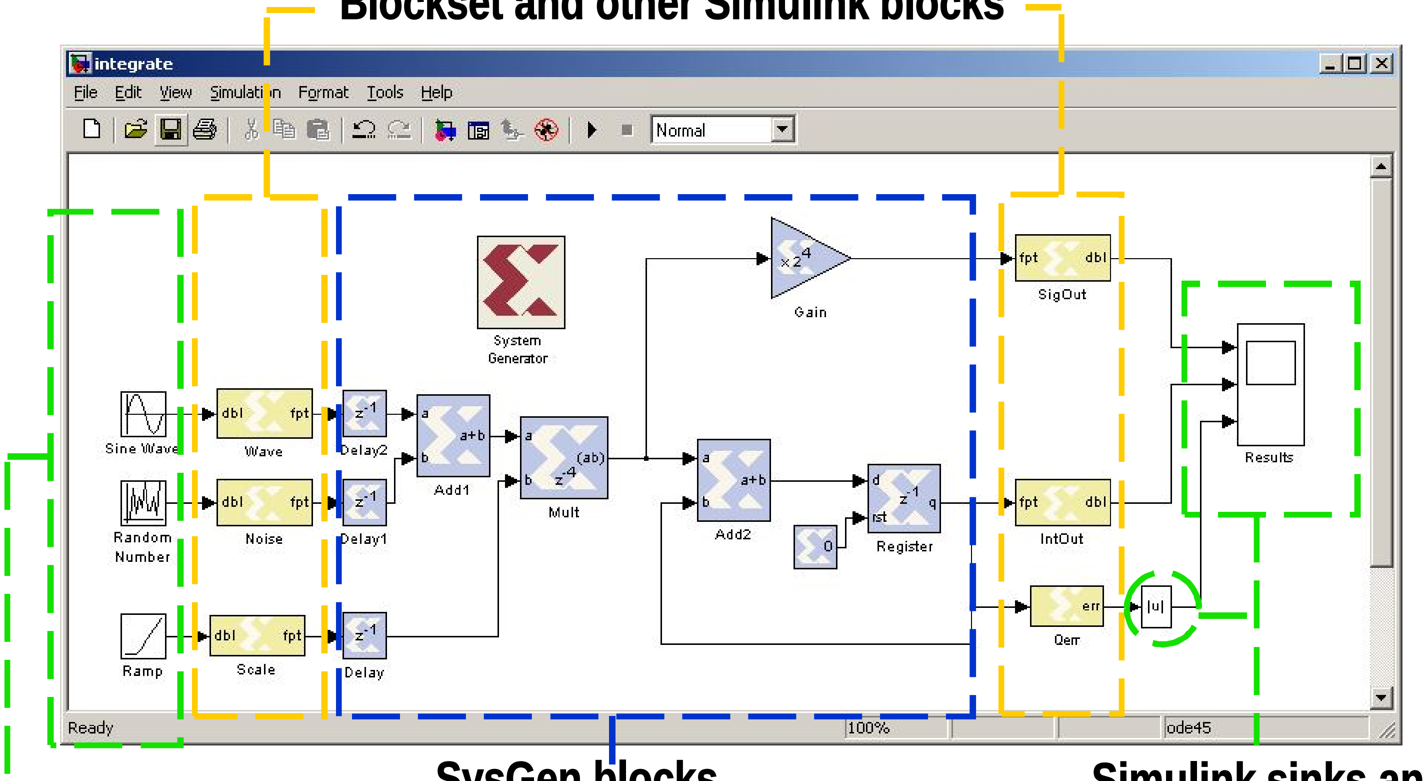
- Build the design by dragging and dropping blocks from the Xilinx blockset onto your new sheet.
- Design Entry is similar to a schematic editor



Connect up blocks by pulling the arrows on the sides of each block

Creating a System Generator Design

I/O blocks used as interface between the Xilinx Blockset and other Simulink blocks



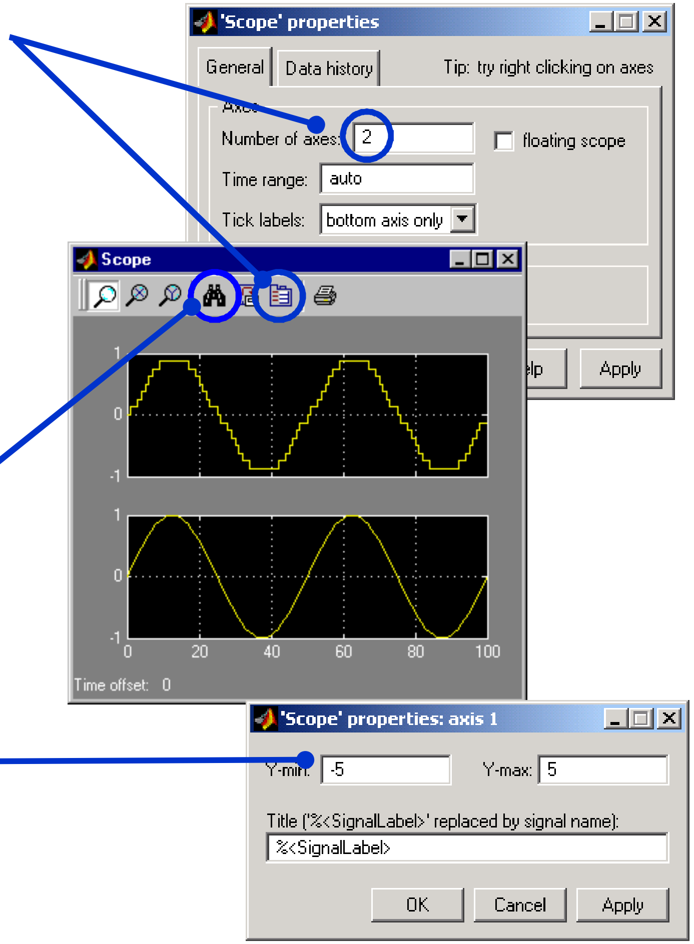
Simulink sources

SysGen blocks realizable in Hardware

Simulink sinks and library functions

Using the Scope

- Click Properties to change the number of axes displayed and the time range value (X-axis)
- Use the Data History tab to control how many values are stored and displayed on the scope
  - Also can direct output to workspace
- Click Autoscale to quickly let the tools configure the display to the correct axis values
- Right-click on the Y-axis to set its value



'Scope' properties

General | Data history

Tip: try right clicking on axes

Axes

Number of axes: 2

□ floating scope

Time range: auto

Tick labels: bottom axis only

'Scope' properties: axis 1

Y-min: -5

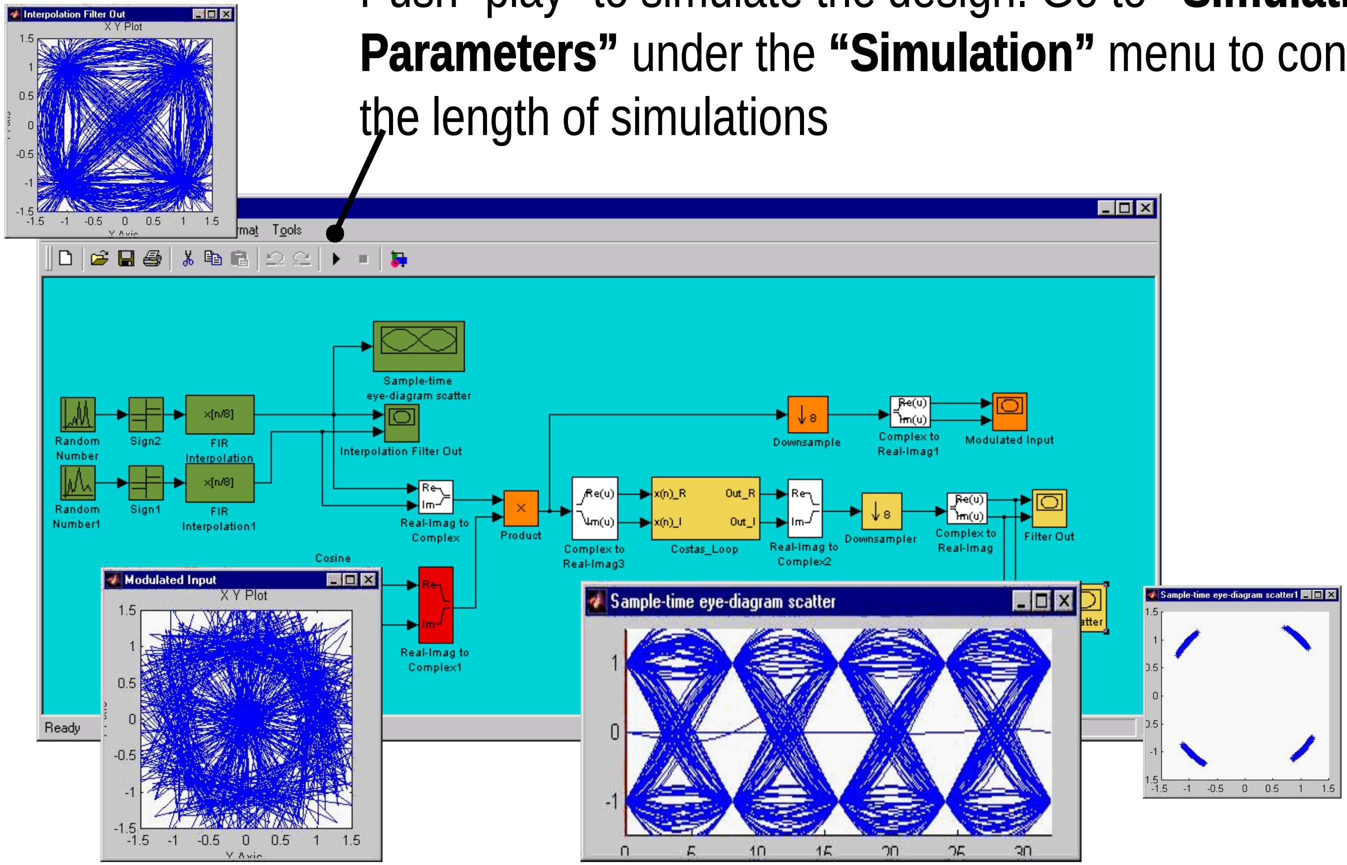
Y-max: 5

Title ('%<SignalLabel>' replaced by signal name): %<SignalLabel>

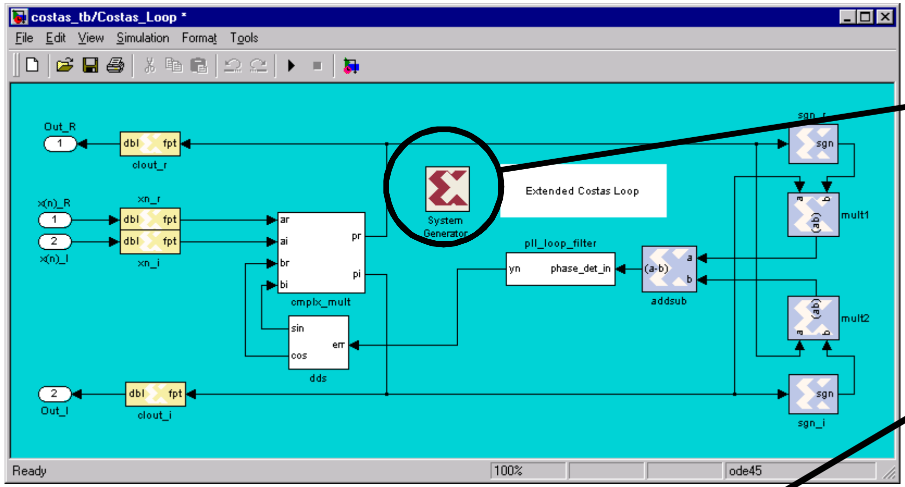
Design and Simulate

in Simulink

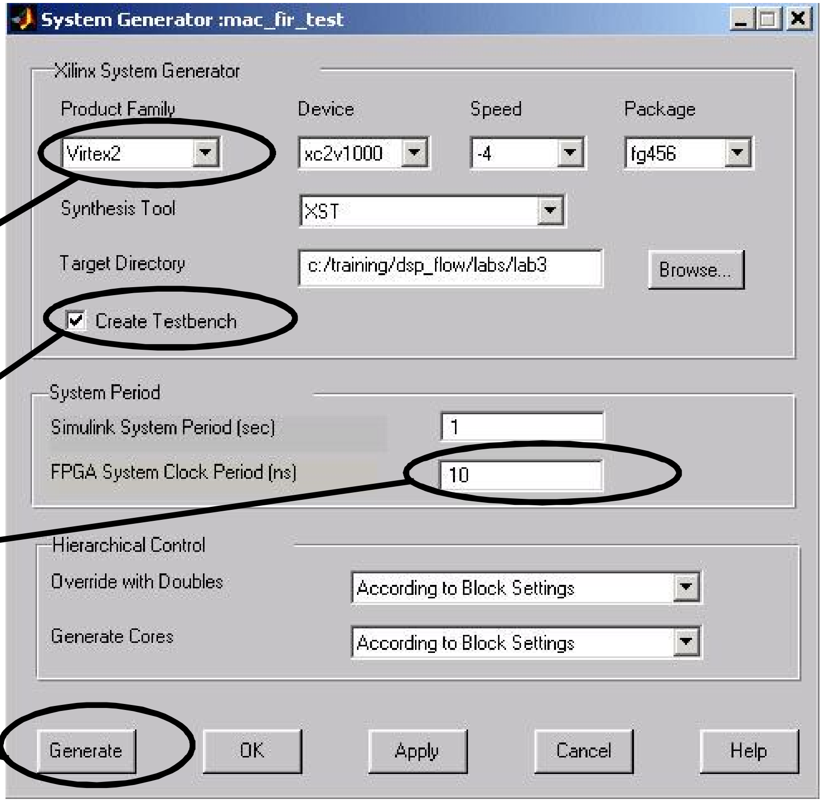
Push “play” to simulate the design. Go to **“Simulation Parameters”** under the **“Simulation”** menu to control the length of simulations



Generate the VHDL Code



Once complete, double-click the System Generator token


System Generator :mac_fir_test

Xilinx System Generator

Product Family	Device	Speed	Package
Virtex2	xc2v1000	-4	fg456

Synthesis Tool: XST

Target Directory: c:/training/dsp_flow/labs/lab3 Browse...

☑ Create Testbench

System Period

Simulink System Period (sec): 1

FPGA System Clock Period (ns): 10

Hierarchical Control

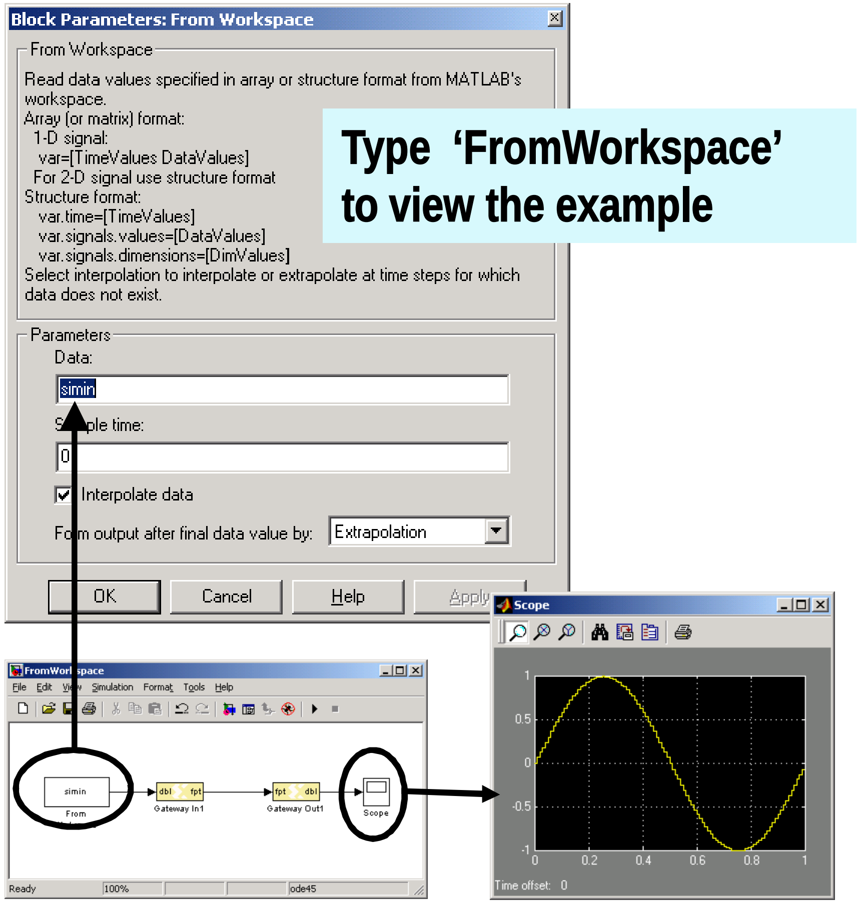
Override with Doubles: According to Block Settings

Generate Cores: According to Block Settings

- Select the target device
- Select to generate the testbench
- Set the System clock period desired
- Generate the VHDL

Inputting Data
from the Workspace

- “From Workspace” block can be used to input MATLAB data to a Simulink model
- Format:
  - t = 0:time_step:final_time;
  - x = func(t);
  - make these into a matrix for Simulink
- Example:
  - In the MATLAB console, type:
    ```
    t = 0:0.01:1;
    x = sin(2*pi*t);
    simin = [t', x'];
    ```



Block Parameters: From Workspace

From Workspace

Read data values specified in array or structure format from MATLAB's workspace.
Array (or matrix) format:
 1-D signal:
  var=[TimeValues DataValues]
 For 2-D signal use structure format
Structure format:
  var.time=[TimeValues]
  var.signals.values=[DataValues]
  var.signals.dimensions=[DimValues]
Select interpolation to interpolate or extrapolate at time steps for which data does not exist.

Parameters

Data: simin

Sample time: 0

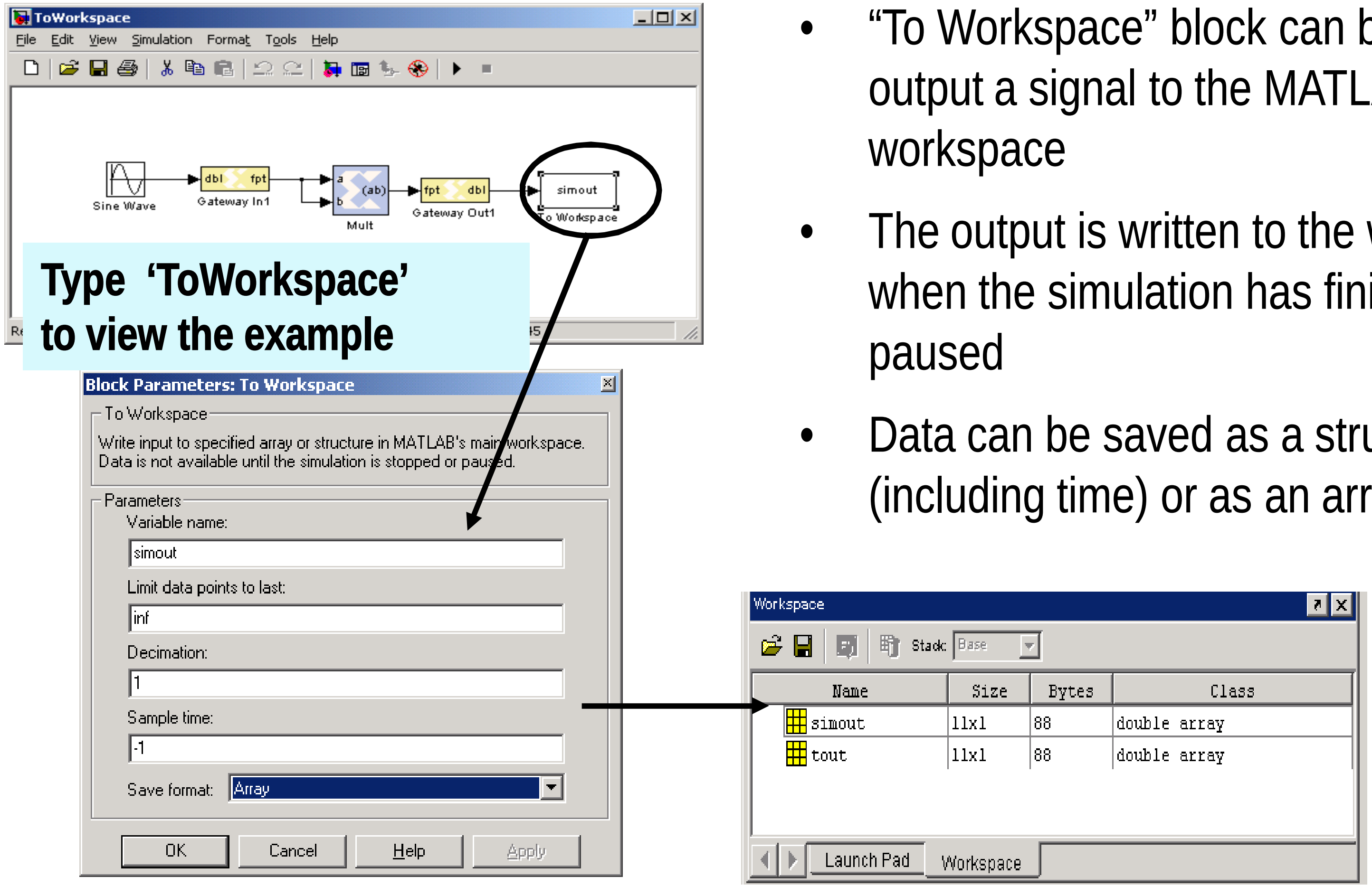
☑ Interpolate data

Form output after final data value by: Extrapolation

Type ‘FromWorkspace’ to view the example

Outputting Data

to the Workspace



Type 'ToWorkspace' to view the example

Block Parameters: To Workspace

To Workspace

Write input to specified array or structure in MATLAB's main workspace. Data is not available until the simulation is stopped or paused.

Parameters

Variable name: simout

Limit data points to last: inf

Decimation: 1

Sample time: -1

Save format: Array

Workspace

Name	Size	Bytes	Class
simout	11x1	88	double array
tout	11x1	88	double array

- "To Workspace" block can be used to output a signal to the MATLAB workspace
- The output is written to the workspace when the simulation has finished or is paused
- Data can be saved as a structure (including time) or as an array

Assignment No. 26

- Discuss FPGA programming in Xilinx through MATLAB.