Xilinx Tool Flow

Objectives

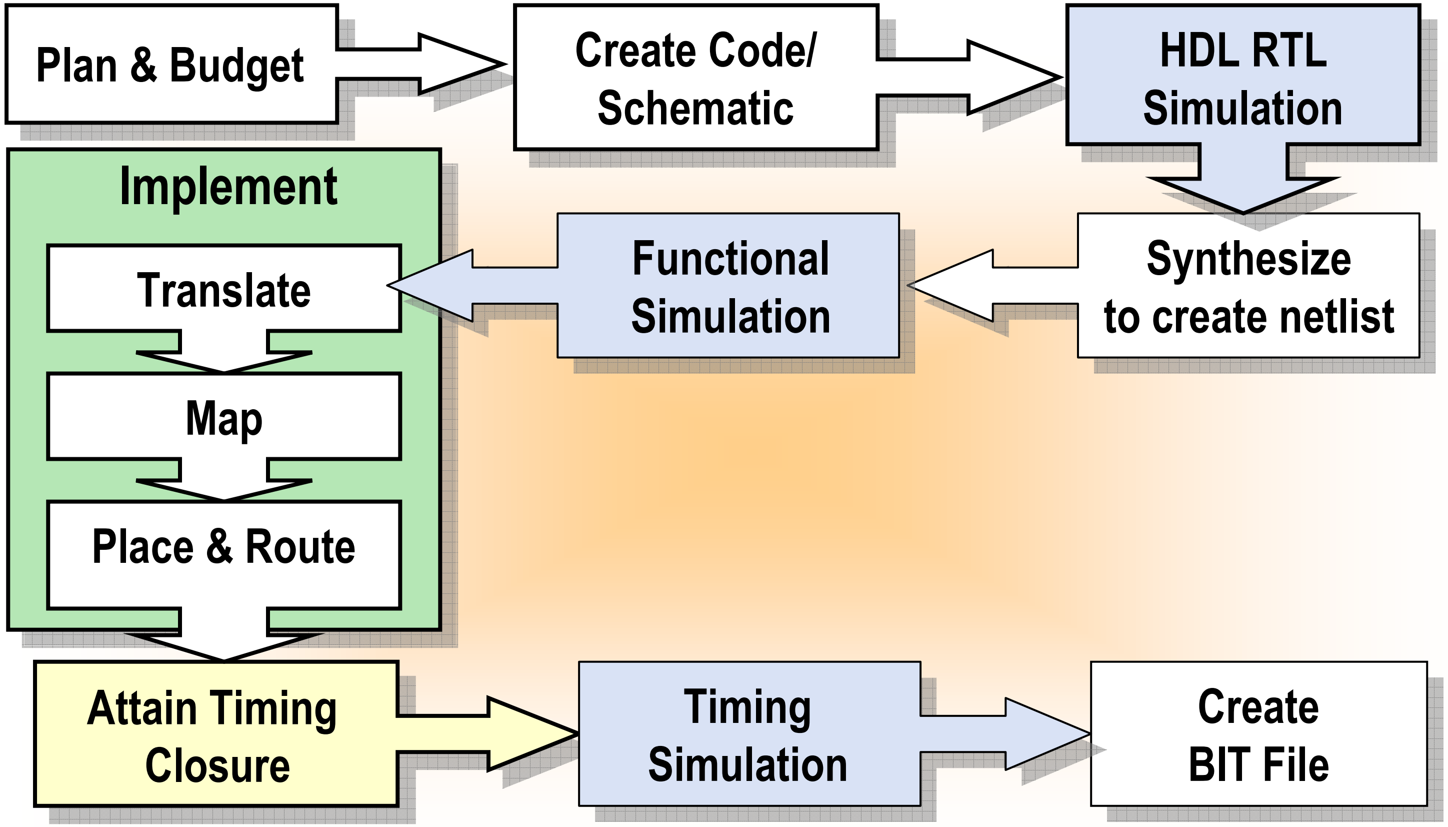
After completing this module, you will be able to:

- List the steps of the Xilinx design process
- Implement and simulate an FPGA design by using default software options

Outline

- Overview
- ISE
- Summary
- Lab 1: Xilinx Tool Flow Demo

Xilinx Design Flow



Plan & Budget

Create Code/ Schematic

HDL RTL Simulation

Synthesize to create netlist

Functional Simulation

Implement

Translate

Map

Place & Route

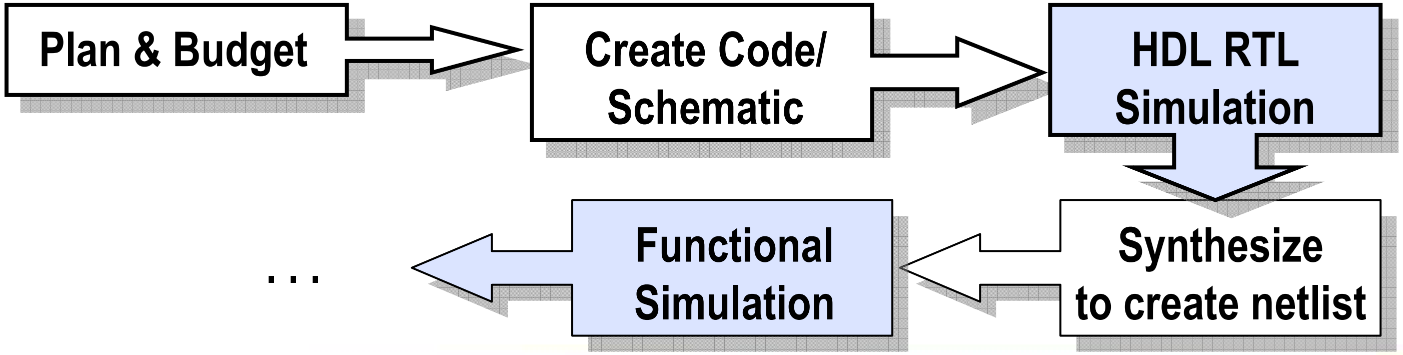
Attain Timing Closure

Timing Simulation

Create BIT File

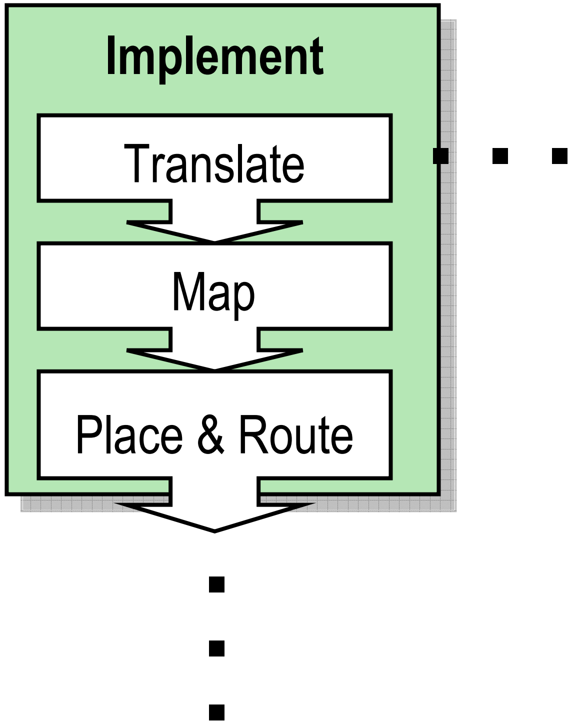
Design Entry

- Plan and budget
- Two design-entry methods: HDL or schematic
  - Architecture Wizard, CORE Generator™ system, and StateCAD tools are available to assist in design entry
- Whichever method you use, you will need a tool to generate an EDIF or NGC netlist to bring into the Xilinx implementation tools
  - Popular synthesis tools include: Synplify, Precision, FPGA Compiler II, and XST
- Simulate the design to ensure that it works as expected!



Xilinx Implementation

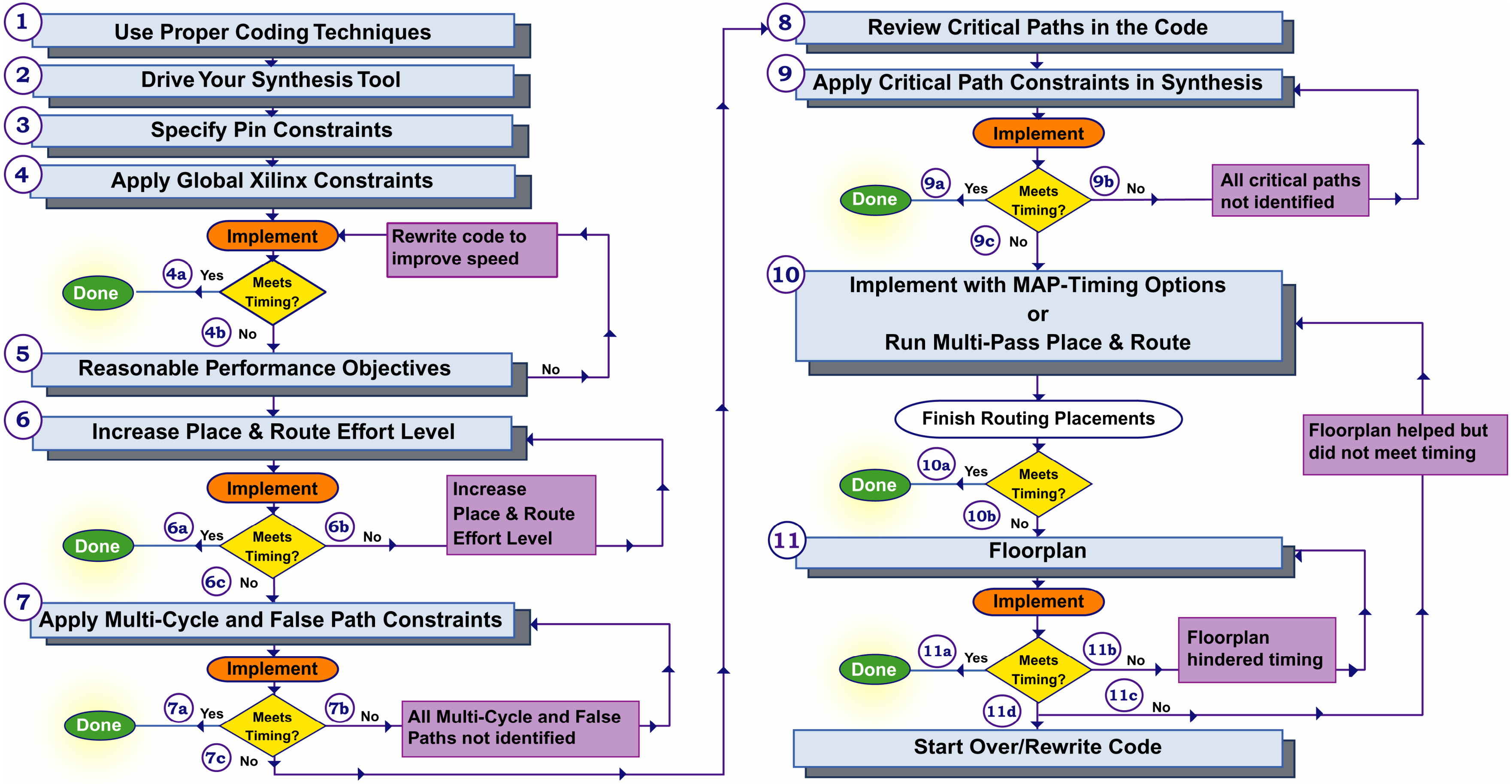
- Once you generate a netlist, you can implement the design
- There are several outputs of implementation
  - Reports
  - Timing simulation netlists
  - Floorplan files
  - FPGA Editor files
  - and more!



What is Implementation?

- More than just *Place & Route*
- Implementation includes many phases
  - **Translate:** Merge multiple design files into a single netlist
  - **Map:** Group logical symbols from the netlist (gates) into physical components (slices and IOBs)
  - **Place & Route:** Place components onto the chip, connect the components, and extract timing data into reports
- Each phase generates files that allow you to use other Xilinx tools
  - Floorplanner, FPGA Editor, XPower

Timing Closure



1 Use Proper Coding Techniques
2 Drive Your Synthesis Tool
3 Specify Pin Constraints
4 Apply Global Xilinx Constraints
Implement
Rewrite code to improve speed
4a Yes
Done
Meets Timing?
4b No
5 Reasonable Performance Objectives
No
6 Increase Place & Route Effort Level
Implement
6a Yes
Done
Meets Timing?
6b No
Increase Place & Route Effort Level
6c No
7 Apply Multi-Cycle and False Path Constraints
Implement
7a Yes
Done
Meets Timing?
7b No
All Multi-Cycle and False Paths not identified
7c No
8 Review Critical Paths in the Code
9 Apply Critical Path Constraints in Synthesis
Implement
9a Yes
Done
Meets Timing?
9b No
All critical paths not identified
9c No
10 Implement with MAP-Timing Options or Run Multi-Pass Place & Route
Finish Routing Placements
10a Yes
Done
Meets Timing?
10b No
Floorplan helped but did not meet timing
11 Floorplan
Implement
11a Yes
Done
Meets Timing?
11b No
Floorplan hindered timing
11c No
11d
Start Over/Rewrite Code

Download

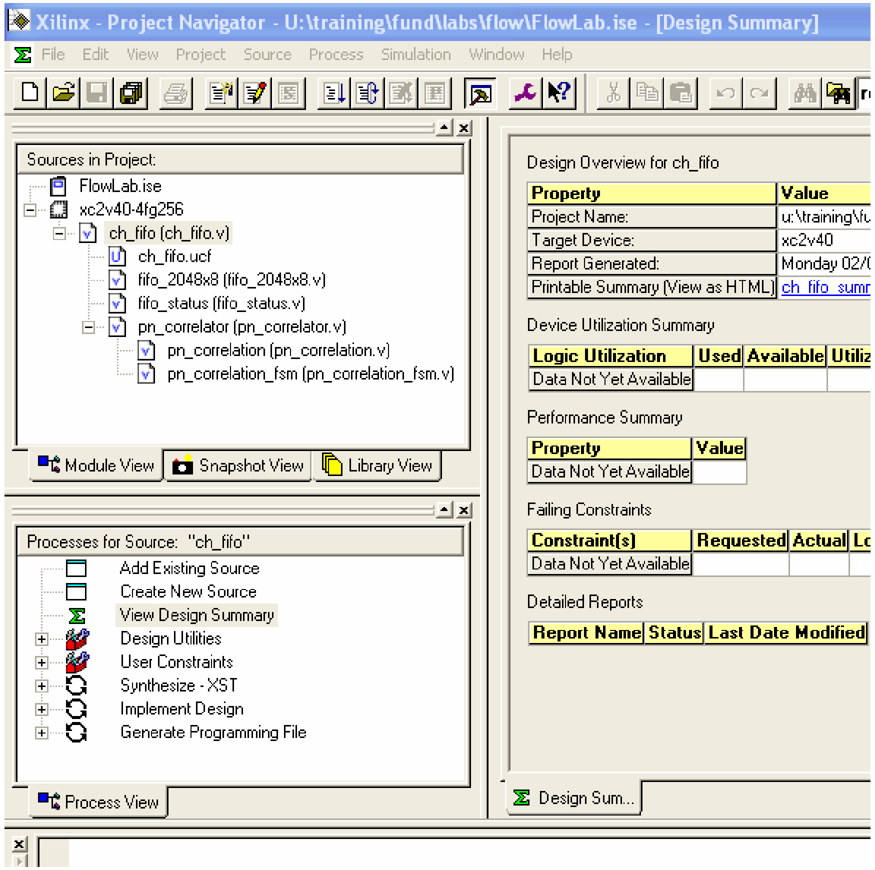
- Once a design is implemented, you must create a file that the FPGA can understand
  - This file is called a bitstream: a BIT file (.bit extension)
- The BIT file can be downloaded directly into the FPGA, or the BIT file can be converted into a PROM file, which stores the programming information

Outline

- Overview
- ISE
- Summary
- Lab 1: Xilinx Tool Flow Demo

ISE Project Navigator

- Built around the Xilinx design flow
  - Access to synthesis and schematic tools
    - Including third-party synthesis tools
  - Implement your design with a simple double-click
    - Fine-tune with easy-to-access software options


Xilinx - Project Navigator - U:\training\fund\labs\flow\FlowLab.ise - [Design Summary]
File Edit View Project Source Process Simulation Window Help
Sources in Project:
FlowLab.ise
xc2v40-4fg256
ch_fifo (ch_fifo.v)
ch_fifo.ucf
fifo_2048x8 (fifo_2048x8.v)
fifo_status (fifo_status.v)
pn_correlator (pn_correlator.v)
pn_correlation (pn_correlation.v)
pn_correlation_fsm (pn_correlation_fsm.v)
Module View
Snapshot View
Library View
Processes for Source: "ch_fifo"
Add Existing Source
Create New Source
View Design Summary
Design Utilities
User Constraints
Synthesize - XST
Implement Design
Generate Programming File
Process View
Design Sum...

Design Overview for ch_fifo

Property	Value
Project Name:	u:\training\fu
Target Device:	xc2v40
Report Generated:	Monday 02/(
Printable Summary (View as HTML)	ch_fifo_sumr

Device Utilization Summary

Logic Utilization	Used	Available	Utiliz
Data Not Yet Available			

Performance Summary

Property	Value
Data Not Yet Available	

Failing Constraints

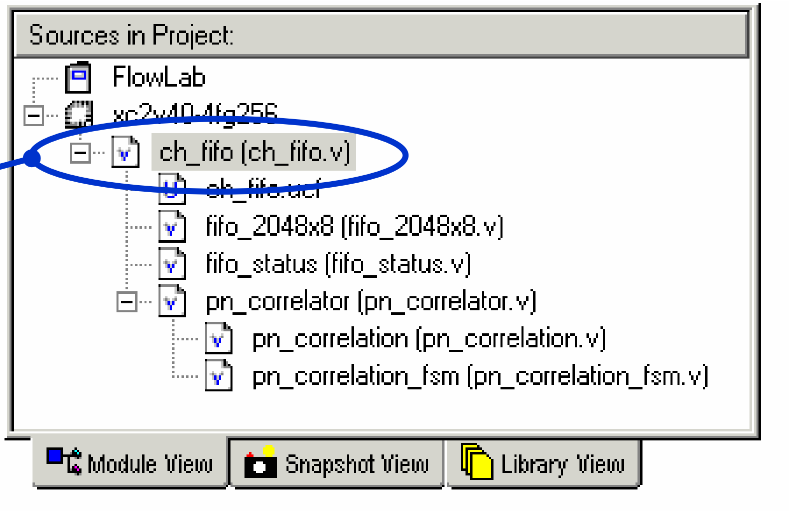
Constraint(s)	Requested	Actual	Lo
Data Not Yet Available			

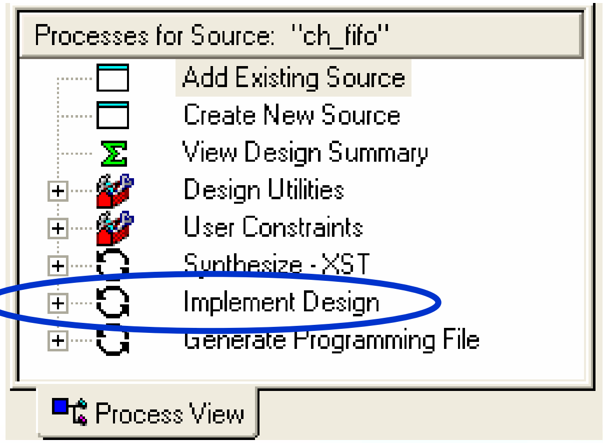
Detailed Reports

Report Name	Status	Last Date Modified

Implementing a Design

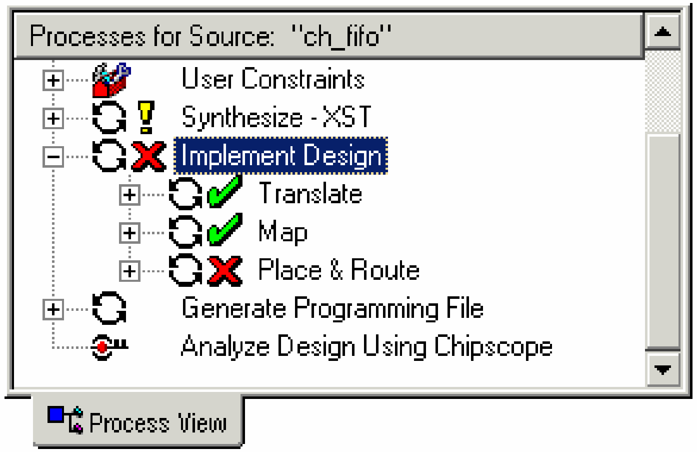
- Implement a design:
  - Select the **top-level source file** in the Sources in Project window
    - HDL, schematic, or EDIF, depending on your design flow
  - Double-click **Implement Design** in the Processes for Source window


Sources in Project:
FlowLab
xc2v40-4fg256
ch_fifo (ch_fifo.v)
ch_fifo.ucf
fifo_2048x8 (fifo_2048x8.v)
fifo_status (fifo_status.v)
pn_correlator (pn_correlator.v)
pn_correlation (pn_correlation.v)
pn_correlation_fsm (pn_correlation_fsm.v)
Module View
Snapshot View
Library View



Implementation Status

- The ISE™ software will run all of the necessary steps to implement the design
  - Synthesize HDL code
  - Translate
  - Map
  - Place & Route
- Progress and status are indicated by icons
  - Green check mark (✓) indicates that the process was completed successfully
  - Yellow exclamation point (!) indicates warnings
  - Yellow question mark (?) indicates a file that is out of date
  - Red X indicates errors

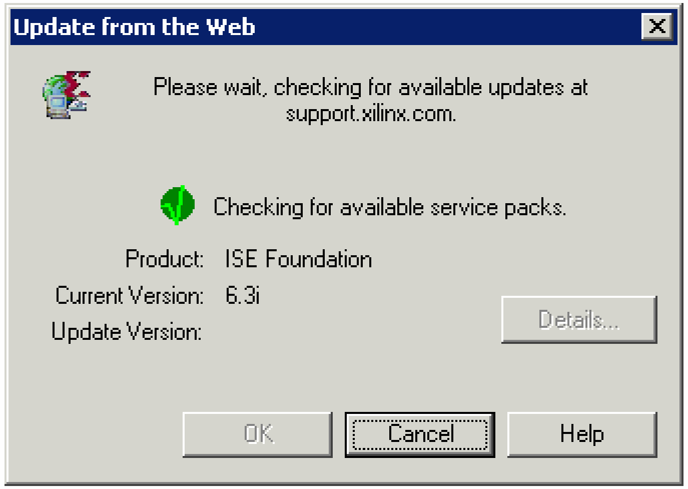


Processes for Source: "ch_fifo"

- User Constraints
- Synthesize - XST !
- Implement Design ✗
  - Translate ✓
  - Map ✓
  - Place & Route ✗
- Generate Programming File
- Analyze Design Using Chipscope

Software Update Center

- Automatically checks for Service Packs on the Web
- Alerts you when an update is available
- Supports Windows platforms only



Update from the Web

Please wait, checking for available updates at support.xilinx.com.

Checking for available service packs.

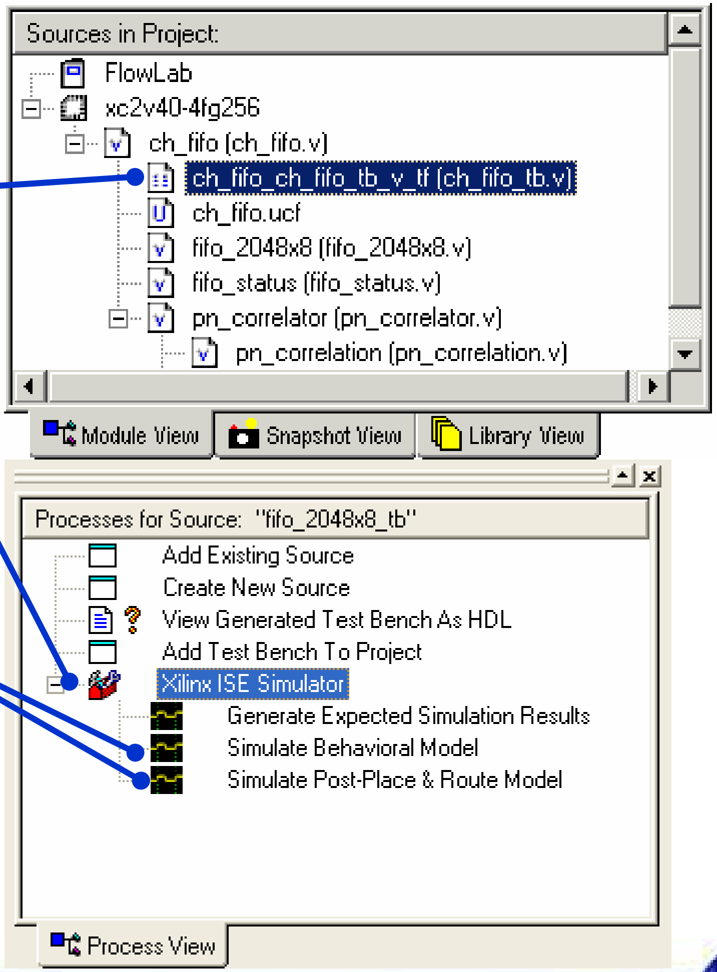
Product: ISE Foundation

Current Version: 6.3i

Update Version:

Simulating a Design

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



Sources in Project:

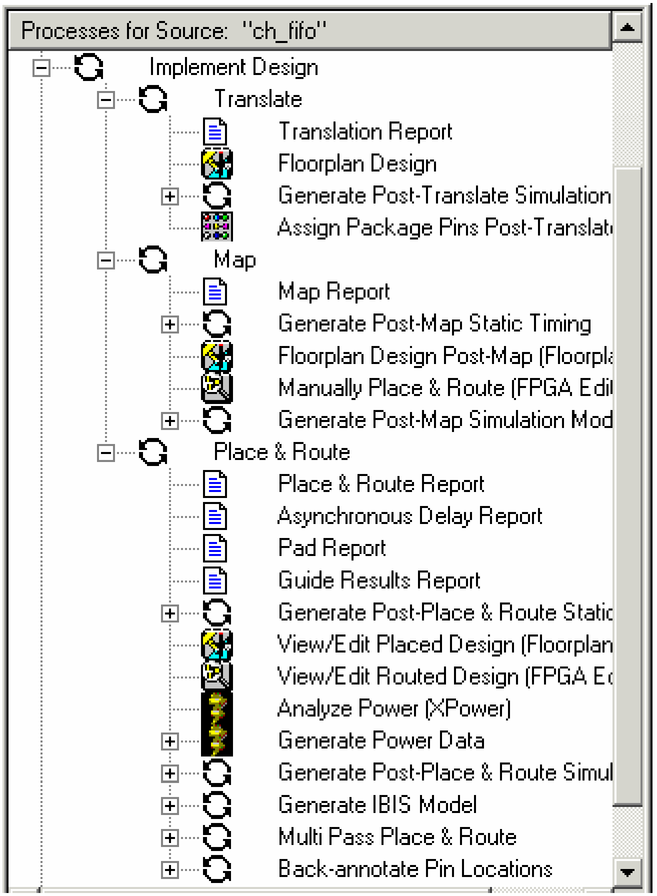
```
FlowLab
xc2v40-4fg256
  ch_fifo (ch_fifo.v)
    ch_fifo_ch_fifo_tb_v_tf (ch_fifo_tb.v)
    ch_fifo.ucf
    fifo_2048x8 (fifo_2048x8.v)
    fifo_status (fifo_status.v)
    pn_correlator (pn_correlator.v)
      pn_correlation (pn_correlation.v)
```

Processes for Source: "fifo_2048x8_tb"

```
Add Existing Source
Create New Source
View Generated Test Bench As HDL
Add Test Bench To Project
Xilinx ISE Simulator
  Generate Expected Simulation Results
  Simulate Behavioral Model
  Simulate Post-Place & Route Model
```

Subprocesses

- Expand each process to view subtools and subprocesses
  - Translate
    - Floorplan
    - Assign package pins
  - Map
    - Analyze timing
  - Place & Route
    - Analyze timing
    - Floorplan
    - FPGA Editor
    - Analyze power
    - Create simulation model

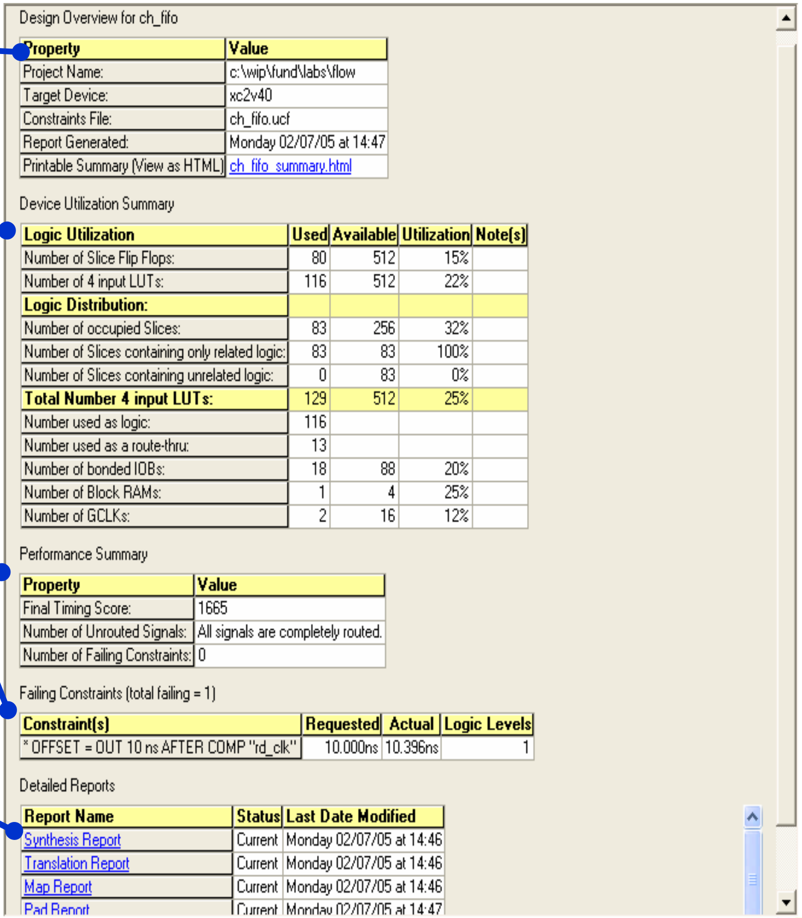


Processes for Source: "ch_fifo"

```
Implement Design
  Translate
    Translation Report
    Floorplan Design
    Generate Post-Translate Simulation
    Assign Package Pins Post-Translat
  Map
    Map Report
    Generate Post-Map Static Timing
    Floorplan Design Post-Map (Floorpl
    Manually Place & Route (FPGA Edit
    Generate Post-Map Simulation Mod
  Place & Route
    Place & Route Report
    Asynchronous Delay Report
    Pad Report
    Guide Results Report
    Generate Post-Place & Route Static
    View/Edit Placed Design (Floorplan
    View/Edit Routed Design (FPGA Ed
    Analyze Power (XPower)
    Generate Power Data
    Generate Post-Place & Route Simul
    Generate IBIS Model
    Multi Pass Place & Route
    Back-annotate Pin Locations
```

Project Summary

- Design Overview
- Device Utilization
- Performance and Constraints
- Reports



Design Overview for ch_fifo

Property	Value
Project Name:	c:\wip\fund\labs\flow
Target Device:	xc2v40
Constraints File:	ch_fifo.ucf
Report Generated:	Monday 02/07/05 at 14:47
Printable Summary (View as HTML)	ch_fifo_summary.html

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops:	80	512	15%	
Number of 4 input LUTs:	116	512	22%	
Logic Distribution:				
Number of occupied Slices:	83	256	32%	
Number of Slices containing only related logic:	83	83	100%	
Number of Slices containing unrelated logic:	0	83	0%	
Total Number 4 input LUTs:	129	512	25%	
Number used as logic:	116			
Number used as a route-thru:	13			
Number of bonded IOBs:	18	88	20%	
Number of Block RAMs:	1	4	25%	
Number of GCLKs:	2	16	12%	

Performance Summary

Property	Value
Final Timing Score:	1665
Number of Unrouted Signals:	All signals are completely routed.
Number of Failing Constraints:	0

Failing Constraints (total failing = 1)

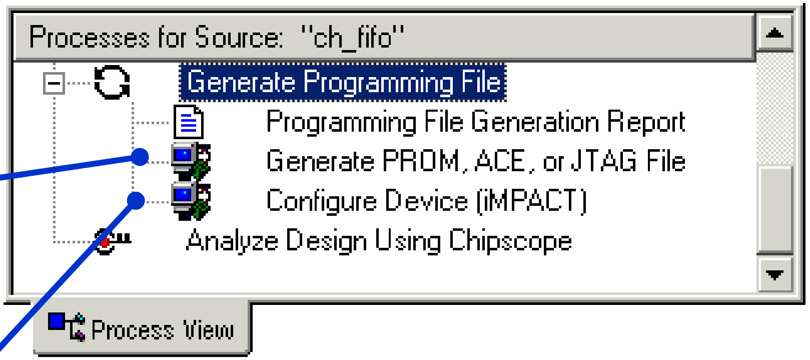
Constraint(s)	Requested	Actual	Logic Levels
* OFFSET = OUT 10 ns AFTER COMP "rd_clk"	10.000ns	10.396ns	1

Detailed Reports

Report Name	Status	Last Date Modified
Synthesis Report	Current	Monday 02/07/05 at 14:46
Translation Report	Current	Monday 02/07/05 at 14:46
Map Report	Current	Monday 02/07/05 at 14:46
Pad Report	Current	Monday 02/07/05 at 14:47

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



Processes for Source: "ch_fifo"

- Generate Programming File
  - Programming File Generation Report
  - Generate PROM, ACE, or JTAG File
  - Configure Device (iMPACT)
- Analyze Design Using Chipscope

Outline

- Overview
- ISE
- **Summary**
- Lab 1: Xilinx Tool Flow Demo

Review Questions

- What are the phases of the Xilinx design flow?
- What are the components of implementation, and what happens at each step?
- What are two methods of programming an FPGA?

Answers

- What are the phases of the Xilinx design flow?
  - Plan and budget, create code or schematic, RTL simulation, synthesize, functional simulation, implement, timing closure, timing simulation, and BIT file creation
- What are the components of implementation, and what happens at each step?
  - Translate: merges multiple design files into one netlist
  - Map: groups logical symbols into physical components
  - Place & Route: places components onto the chip and connects them
- What are two methods of programming an FPGA?
  - PROM
  - Xilinx iMPACT configuration tool

Summary

- Implementation means more than Place & Route
- Xilinx provides a simple *pushbutton* tool to guide you through the Xilinx design process

Where Can I Learn More?

- Complete design flow tutorials
  - www.xilinx.com → Documentation → Tutorials
- On implementation: Development System Reference Guide
  - www.xilinx.com → Documentation → Software Manuals
  - Documentation may also be installed on your local computer
- On simulation: ISIM Online Help
- Configuration Problem Solver
  - www.xilinx.com → Support → Problem Solvers → Configuration Problem Solver

Outline

- Overview
- ISE
- Summary
- **Lab 1: Xilinx Tool Flow Demo**

Software Demo

- Follow along with the instructor:
  - Launch the ISE™ Project Navigator
  - Create a project
  - Add design files to the project
  - Simulate the design
  - Implement the design
- After the design has been implemented, you will examine some reports in the next module