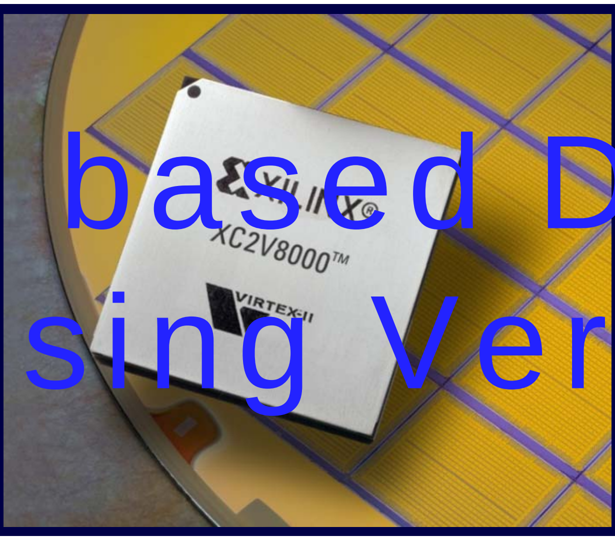
Training Course
on

FPGA based Digital Design using Verilog HDL



By

NAUMAN MIR
(HDL Designer)

**** Organized by*** **Skill Development Council,**
(*Ministry of Labour, Manpower and overseas Pakistani)*
Govt. of Pakistan.

Core Generator

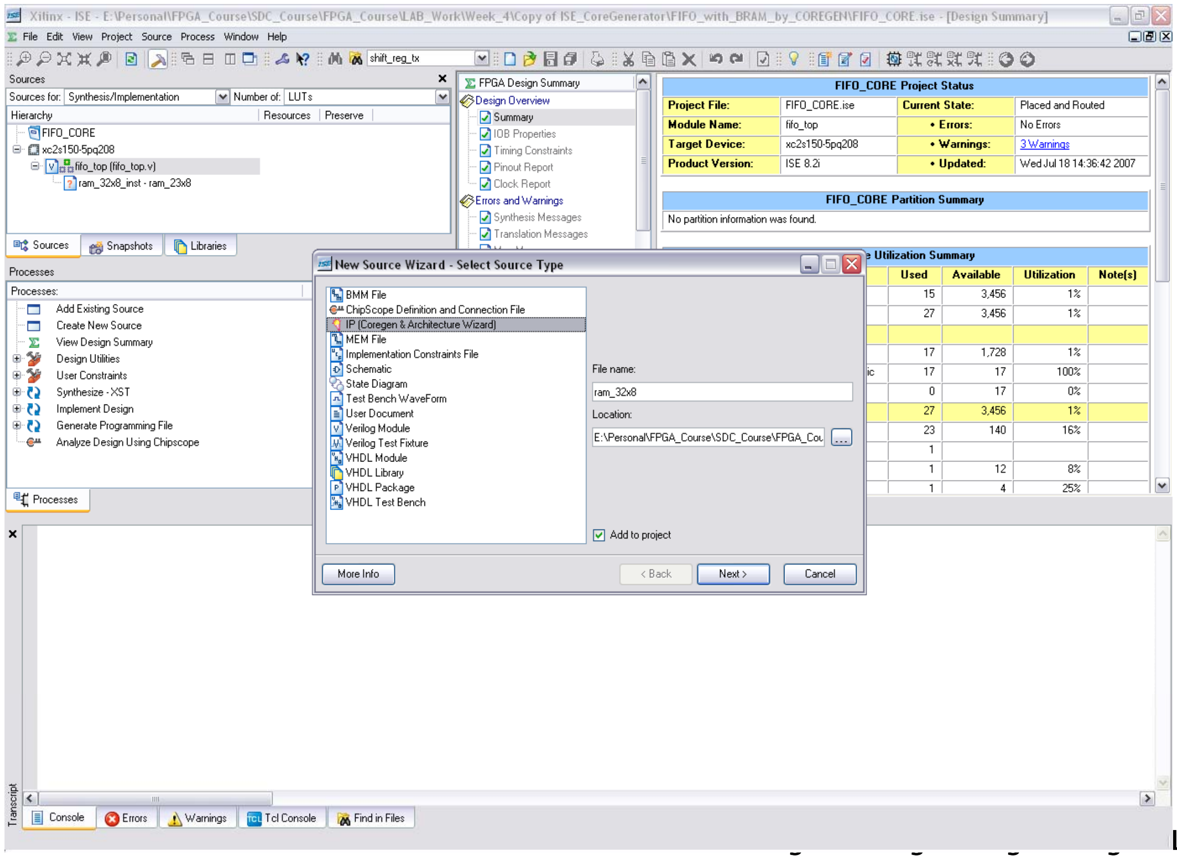
- **The CORE Generator System is a design tool that delivers parameterized cores optimized for Xilinx® FPGAs. It provides you with a catalog of ready-made functions ranging in complexity from simple arithmetic operators such as adders, accumulators, and multipliers, to system-level building blocks such as filters, transforms, FIFOs, and memories.**

Core Generator – contd.

- **Starting the CORE Generator from Xilinx ISE**

The CORE Generator can be opened from within Project Navigator in these ways:

- **If you have added a core to an ISE project, you can then open the CORE Generator GUI from within the ISE Project Navigator. Project Navigator will run on a PC.**


Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_COREGEN\FIFO_CORE.ise - [Design Summary]

Sources for: Synthesis/Implementation — Number of: LUTs

- FIFO_CORE
- xc2s150-5pq208
  - fifo_top (fifo_top.v)
    - ram_32x8_inst - ram_23x8

Processes:
- Add Existing Source
- Create New Source
- View Design Summary
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Analyze Design Using Chipscope

FPGA Design Summary
- Design Overview
  - Summary
  - IOB Properties
  - Timing Constraints
  - Pinout Report
  - Clock Report
- Errors and Warnings
  - Synthesis Messages
  - Translation Messages

FIFO_CORE Project Status			
Project File:	FIFO_CORE.ise	**Current State:**	Placed and Routed
Module Name:	fifo_top	**• Errors:**	No Errors
Target Device:	xc2s150-5pq208	**• Warnings:**	3 Warnings
Product Version:	ISE 8.2i	**• Updated:**	Wed Jul 18 14:36:42 2007

FIFO_CORE Partition Summary
No partition information was found.

e Utilization Summary

	Used	Available	Utilization	Note(s)
	15	3,456	1%	
	27	3,456	1%	
	17	1,728	1%	
ic	17	17	100%	
	0	17	0%	
	27	3,456	1%	
	23	140	16%	
	1			
	1	12	8%	
	1	4	25%	

New Source Wizard - Select Source Type

- BMM File
- ChipScope Definition and Connection File
- IP (Coregen & Architecture Wizard)
- MEM File
- Implementation Constraints File
- Schematic
- State Diagram
- Test Bench WaveForm
- User Document
- Verilog Module
- Verilog Test Fixture
- VHDL Module
- VHDL Library
- VHDL Package
- VHDL Test Bench

File name: ram_32x8

Location: E:\Personal\FPGA_Course\SDC_Course\FPGA_Cou

✓ Add to project

Core Generator – contd.

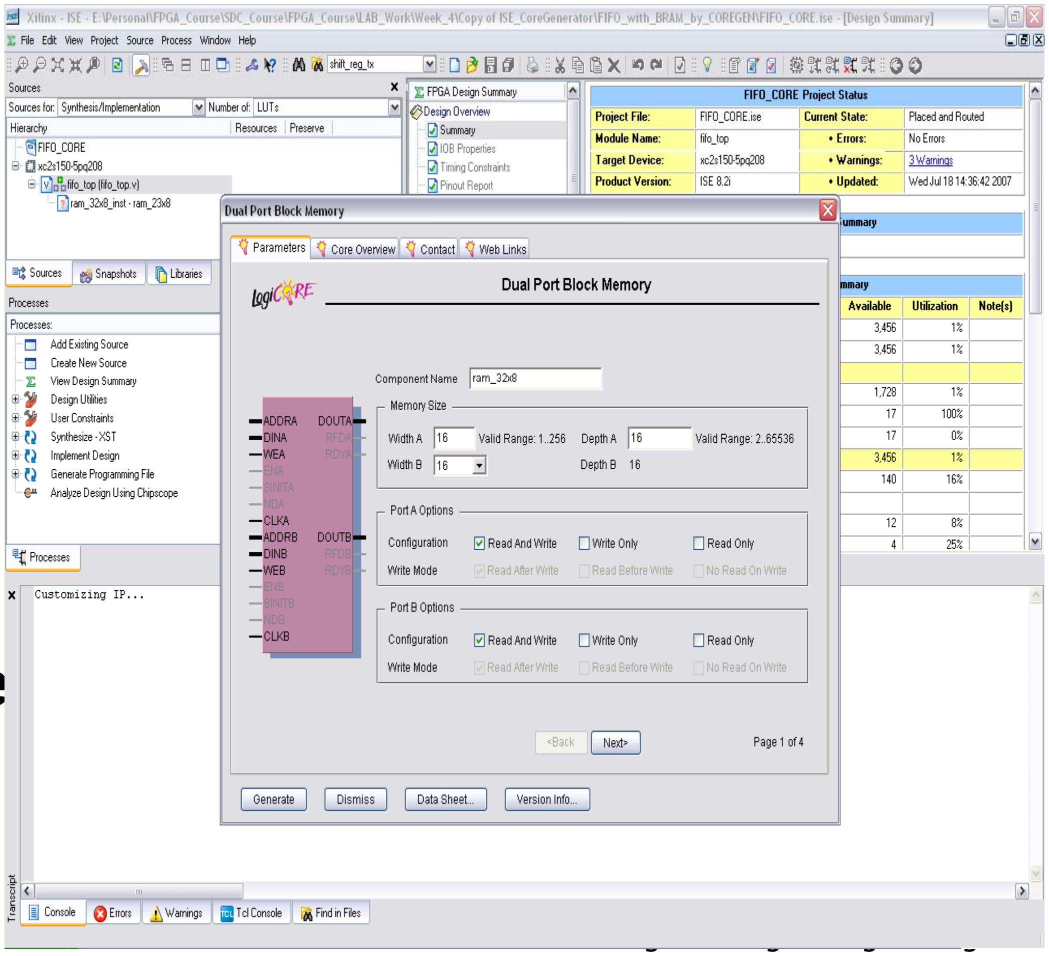
To open the CORE Generator GUI from within Xilinx ISE:

a. In Project Navigator, select an IP core name in the Sources in Project window.

b. Click the "+" icon next to the Coregen process in the Process-es for Current Source window.

The Manage Cores proce-ss shows in the process-es window.

c. Double-click on Manage Cores. The CORE Gener-ator window displays.


Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_COREGEN\FIFO_CORE.ise - [Design Summary]

Sources for: Synthesis/Implementation | Number of: LUTs

Hierarchy: FIFO_CORE / xc2s150-5pq208 / fifo_top (fifo_top.v) / ram_32x8_inst - ram_23x8

Processes: Add Existing Source, Create New Source, View Design Summary, Design Utilities, User Constraints, Synthesize - XST, Implement Design, Generate Programming File, Analyze Design Using Chipscope

FIFO_CORE Project Status

Project File:	FIFO_CORE.ise	Current State:	Placed and Routed
Module Name:	fifo_top	• Errors:	No Errors
Target Device:	xc2s150-5pq208	• Warnings:	3 Warnings
Product Version:	ISE 8.2i	• Updated:	Wed Jul 18 14:36:42 2007

Available	Utilization	Note(s)
3,456	1%	
3,456	1%	
1,728	1%	
17	100%	
17	0%	
3,456	1%	
140	16%	
12	8%	
4	25%	

Dual Port Block Memory

Component Name: ram_32x8

Memory Size: Width A 16 Valid Range: 1..256 Depth A 16 Valid Range: 2..65536; Width B 16 Depth B 16

Port A Options: Configuration [✓ Read And Write] Write Only Read Only; Write Mode [✓ Read After Write] Read Before Write No Read On Write

Port B Options: Configuration [✓ Read And Write] Write Only Read Only; Write Mode [✓ Read After Write] Read Before Write No Read On Write

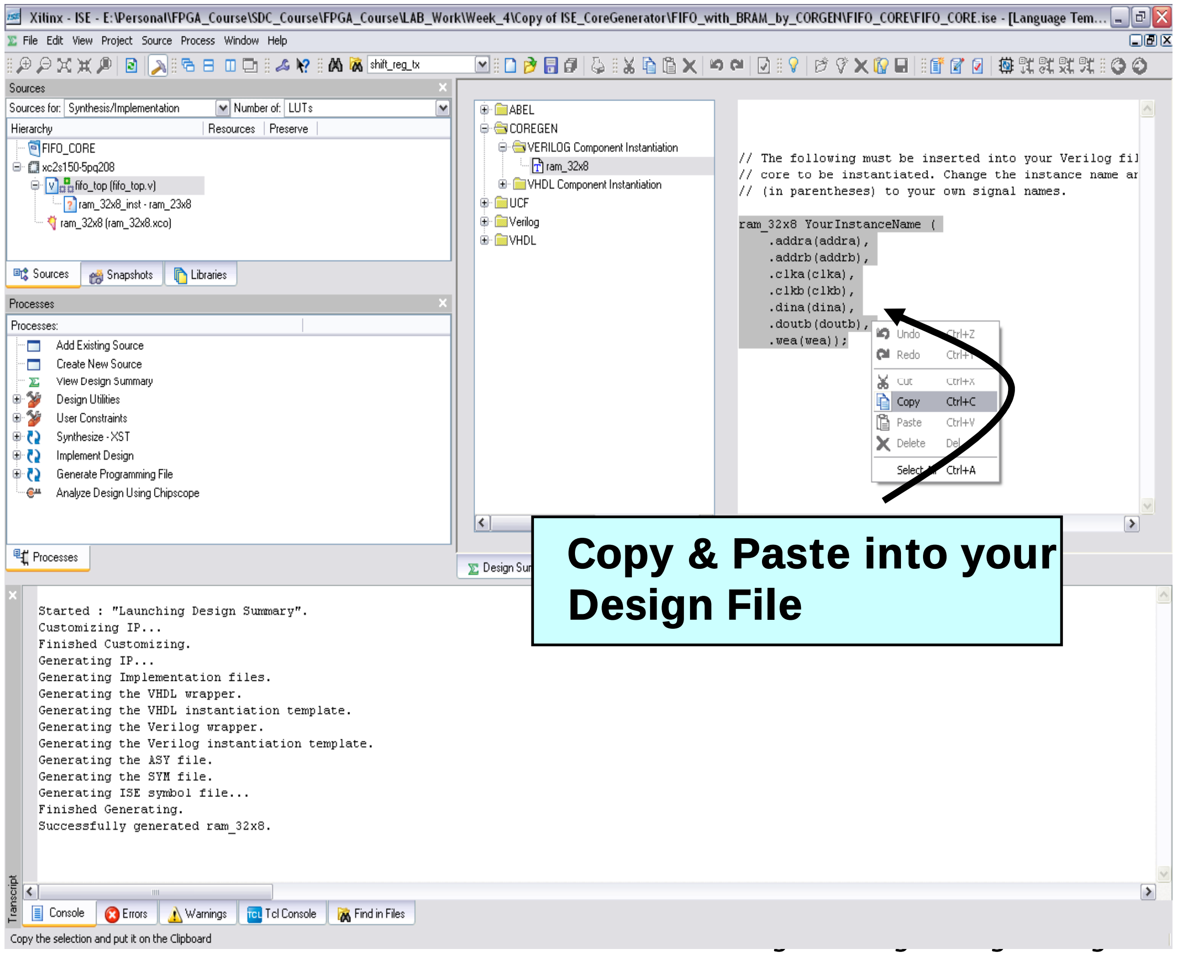
Page 1 of 4

```
Customizing IP...
```

Core Generator – contd.

❑ **To Instantiate an IP Core in an HDL Source**

1. **Select Edit > Language Template to open the Language Template window.**
2. **Click the "+" icon next to COREGEN in the Language Template window. The COREGEN directory will expand.**



```
// The following must be inserted into your Verilog fil
// core to be instantiated. Change the instance name ar
// (in parentheses) to your own signal names.

ram_32x8 YourInstanceName (
    .addra(addra),
    .addrb(addrb),
    .clka(clka),
    .clkb(clkb),
    .dina(dina),
    .doutb(doutb),
    .wea(wea));
```

```
Started : "Launching Design Summary".
Customizing IP...
Finished Customizing.
Generating IP...
Generating Implementation files.
Generating the VHDL wrapper.
Generating the VHDL instantiation template.
Generating the Verilog wrapper.
Generating the Verilog instantiation template.
Generating the ASY file.
Generating the SYM file.
Generating ISE symbol file...
Finished Generating.
Successfully generated ram_32x8.
```

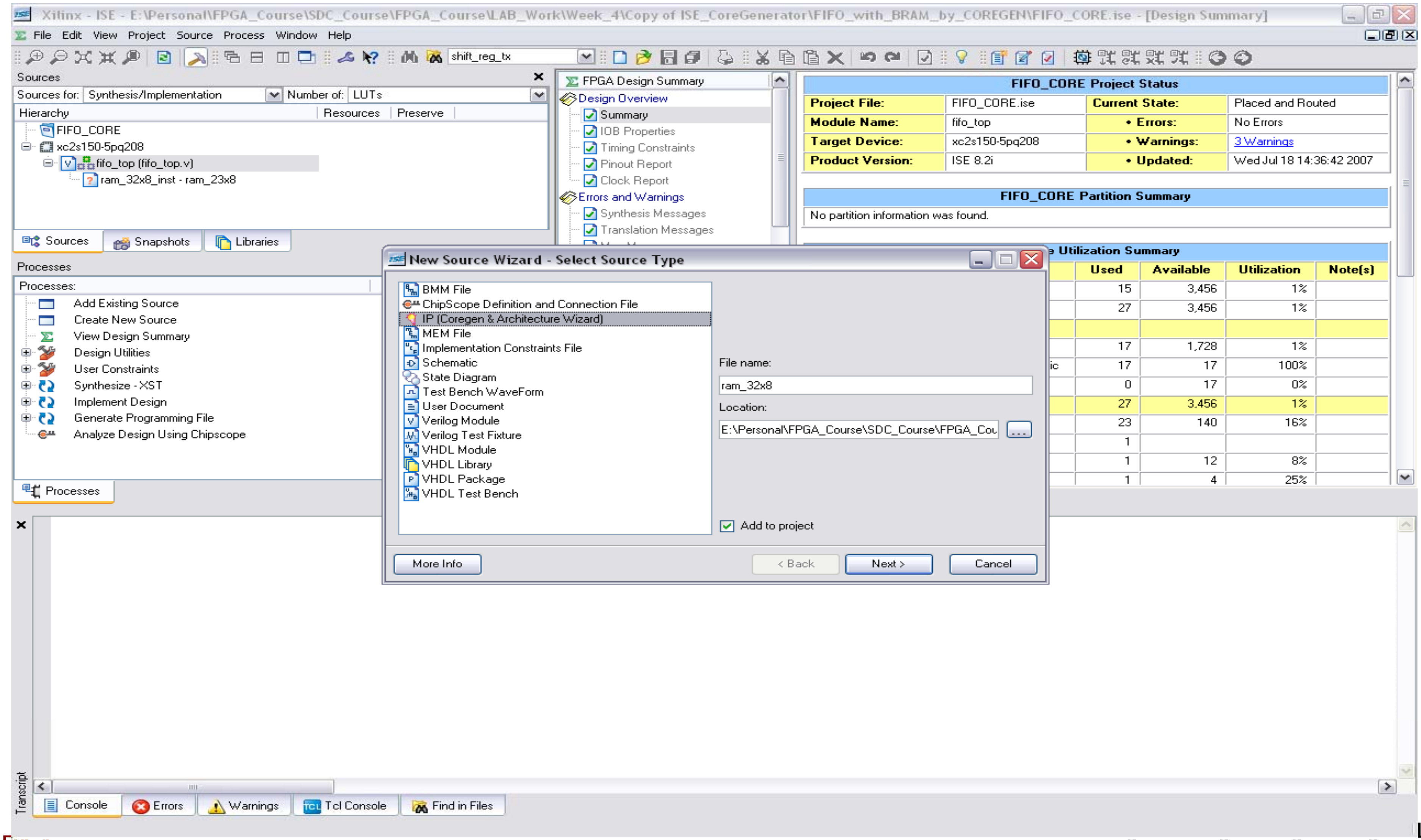
Copy & Paste into your Design File

Core Generator – contd.

3. Click the "+" on either VHDL Component Instantiation or Verilog Component Instantiation. The list of available instantiation templates will expand.
4. Click on the core to display the instantiation template you want to use. The template will display in the right pane of the Language Template window.
5. Cut and paste the template into the open HDL file in the ISE Text Editor window.

Design Example

➤ **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_COREGEN\FIFO_CORE.ise - [Design Summary]

Sources

Sources for: Synthesis/Implementation — Number of: LUTs

Hierarchy	Resources	Preserve
FIFO_CORE		
xc2s150-5pq208		
fifo_top (fifo_top.v)		
ram_32x8_inst - ram_23x8		

Processes

Processes:
- Add Existing Source
- Create New Source
- View Design Summary
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Analyze Design Using Chipscope

FPGA Design Summary
- Design Overview
  - Summary
  - IOB Properties
  - Timing Constraints
  - Pinout Report
  - Clock Report
- Errors and Warnings
  - Synthesis Messages
  - Translation Messages

FIFO_CORE Project Status

Project File:	FIFO_CORE.ise	**Current State:**	Placed and Routed
Module Name:	fifo_top	• **Errors:**	No Errors
Target Device:	xc2s150-5pq208	• **Warnings:**	3 Warnings
Product Version:	ISE 8.2i	• **Updated:**	Wed Jul 18 14:36:42 2007

FIFO_CORE Partition Summary

No partition information was found.

Utilization Summary

	Used	Available	Utilization	Note(s)
	15	3,456	1%	
	27	3,456	1%	
	17	1,728	1%	
ic	17	17	100%	
	0	17	0%	
	27	3,456	1%	
	23	140	16%	
	1			
	1	12	8%	
	1	4	25%	

New Source Wizard - Select Source Type

- BMM File
- ChipScope Definition and Connection File
- IP (Coregen & Architecture Wizard)
- MEM File
- Implementation Constraints File
- Schematic
- State Diagram
- Test Bench WaveForm
- User Document
- Verilog Module
- Verilog Test Fixture
- VHDL Module
- VHDL Library
- VHDL Package
- VHDL Test Bench

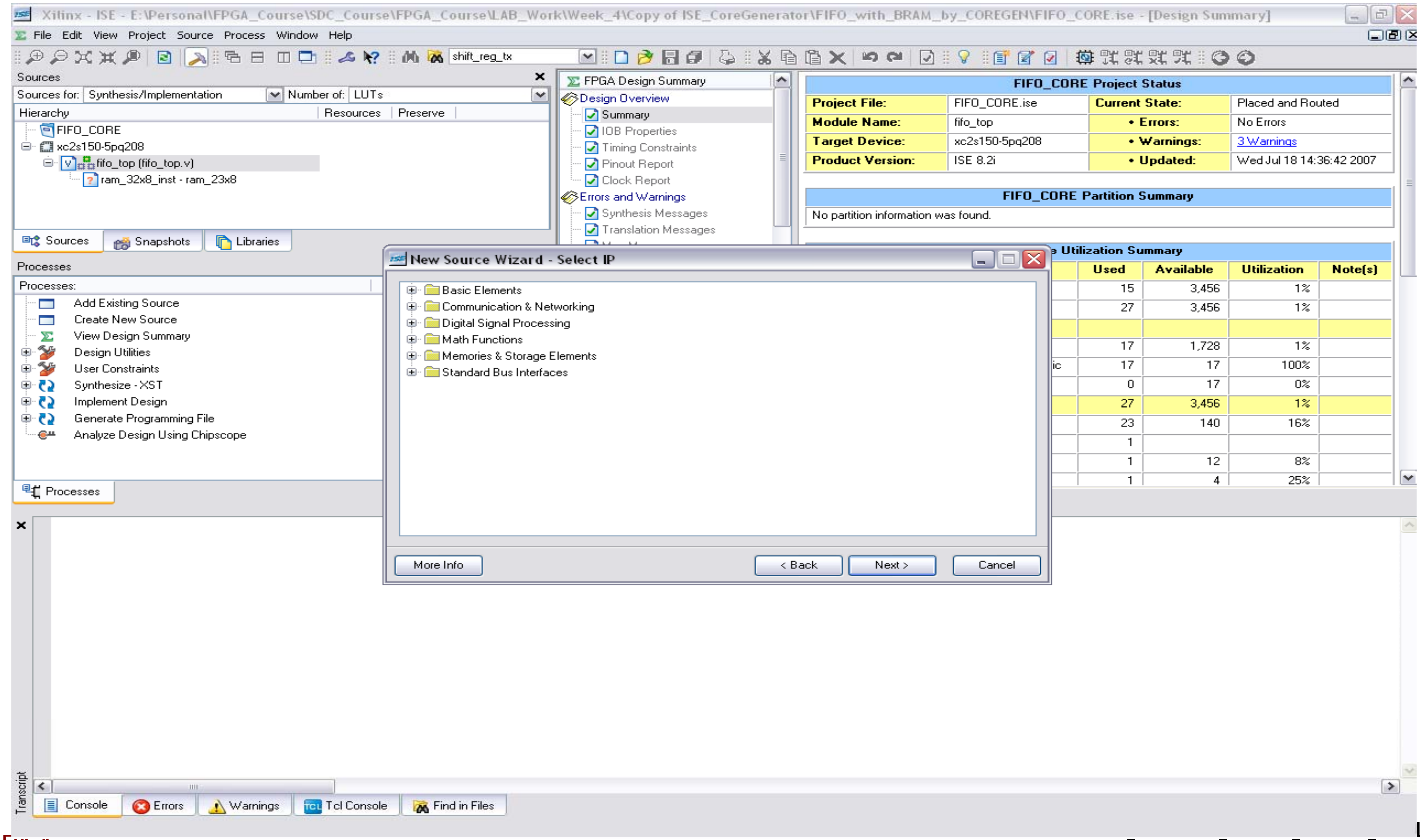
File name: ram_32x8

Location: E:\Personal\FPGA_Course\SDC_Course\FPGA_Cou

☑ Add to project

Design Example

- Generate Block RAM Core from Core Generator Tool (ISE 8.2i)



Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_COREGEN\FIFO_CORE.ise - [Design Summary]

File Edit View Project Source Process Window Help

Sources for: Synthesis/Implementation — Number of: LUTs

Hierarchy | Resources | Preserve

- FIFO_CORE
  - xc2s150-5pq208
    - fifo_top (fifo_top.v)
      - ram_32x8_inst - ram_23x8

Processes:

- Add Existing Source
- Create New Source
- View Design Summary
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Analyze Design Using Chipscope

FPGA Design Summary

- Design Overview
  - Summary
  - IOB Properties
  - Timing Constraints
  - Pinout Report
  - Clock Report
- Errors and Warnings
  - Synthesis Messages
  - Translation Messages

FIFO_CORE Project Status

Project File:	FIFO_CORE.ise	**Current State:**	Placed and Routed
Module Name:	fifo_top	**• Errors:**	No Errors
Target Device:	xc2s150-5pq208	**• Warnings:**	3 Warnings
Product Version:	ISE 8.2i	**• Updated:**	Wed Jul 18 14:36:42 2007

FIFO_CORE Partition Summary

No partition information was found.

Utilization Summary

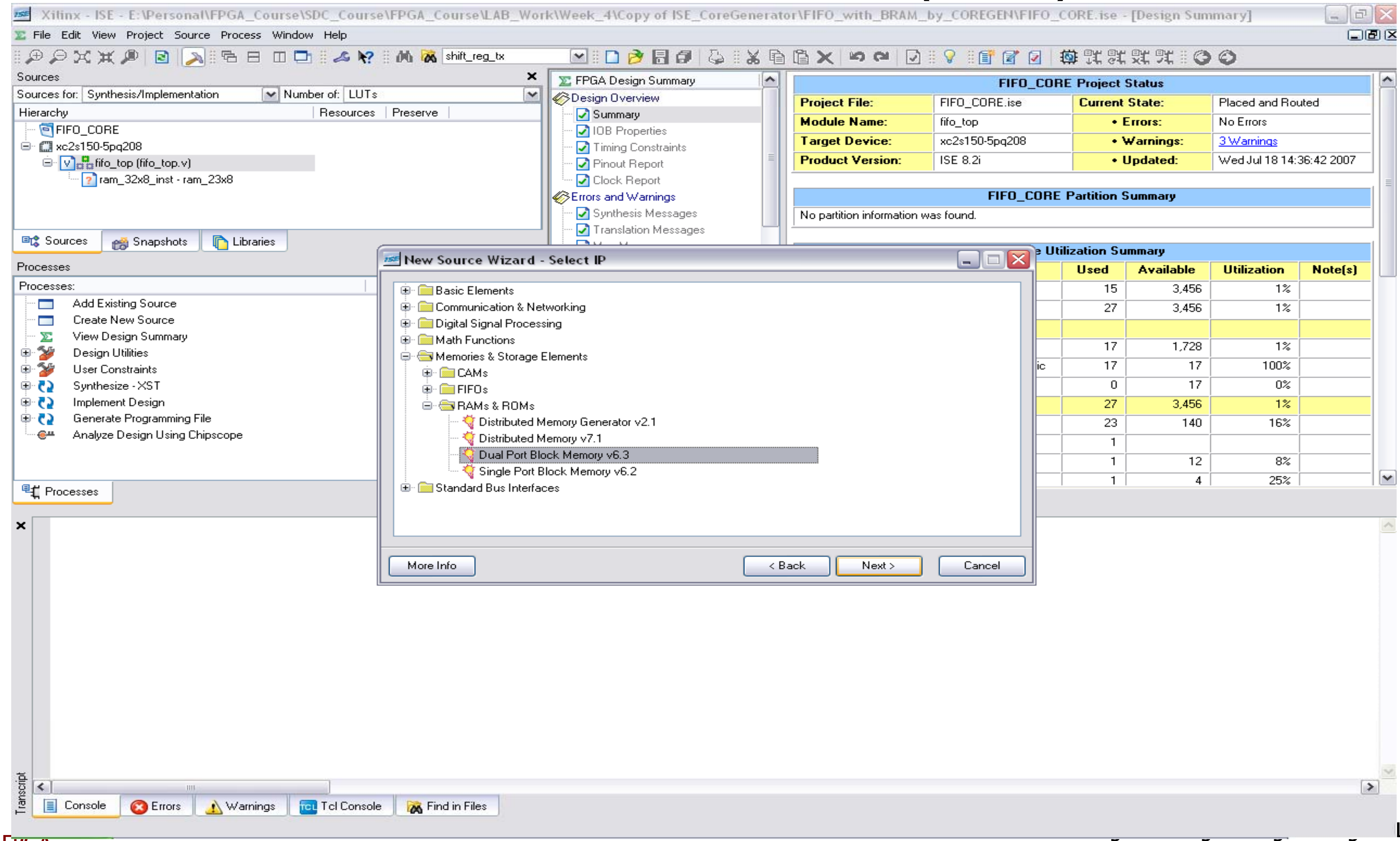
	Used	Available	Utilization	Note(s)
	15	3,456	1%	
	27	3,456	1%	
	17	1,728	1%	
ic	17	17	100%	
	0	17	0%	
	27	3,456	1%	
	23	140	16%	
	1			
	1	12	8%	
	1	4	25%	

New Source Wizard - Select IP

- Basic Elements
- Communication & Networking
- Digital Signal Processing
- Math Functions
- Memories & Storage Elements
- Standard Bus Interfaces

Design Example

- Generate Block RAM Core from Core Generator Tool (ISE 8.2i)



FIFO_CORE Project Status

Project File:	FIFO_CORE.ise	**Current State:**	Placed and Routed
Module Name:	fifo_top	**• Errors:**	No Errors
Target Device:	xc2s150-5pq208	**• Warnings:**	3 Warnings
Product Version:	ISE 8.2i	**• Updated:**	Wed Jul 18 14:36:42 2007

FIFO_CORE Partition Summary

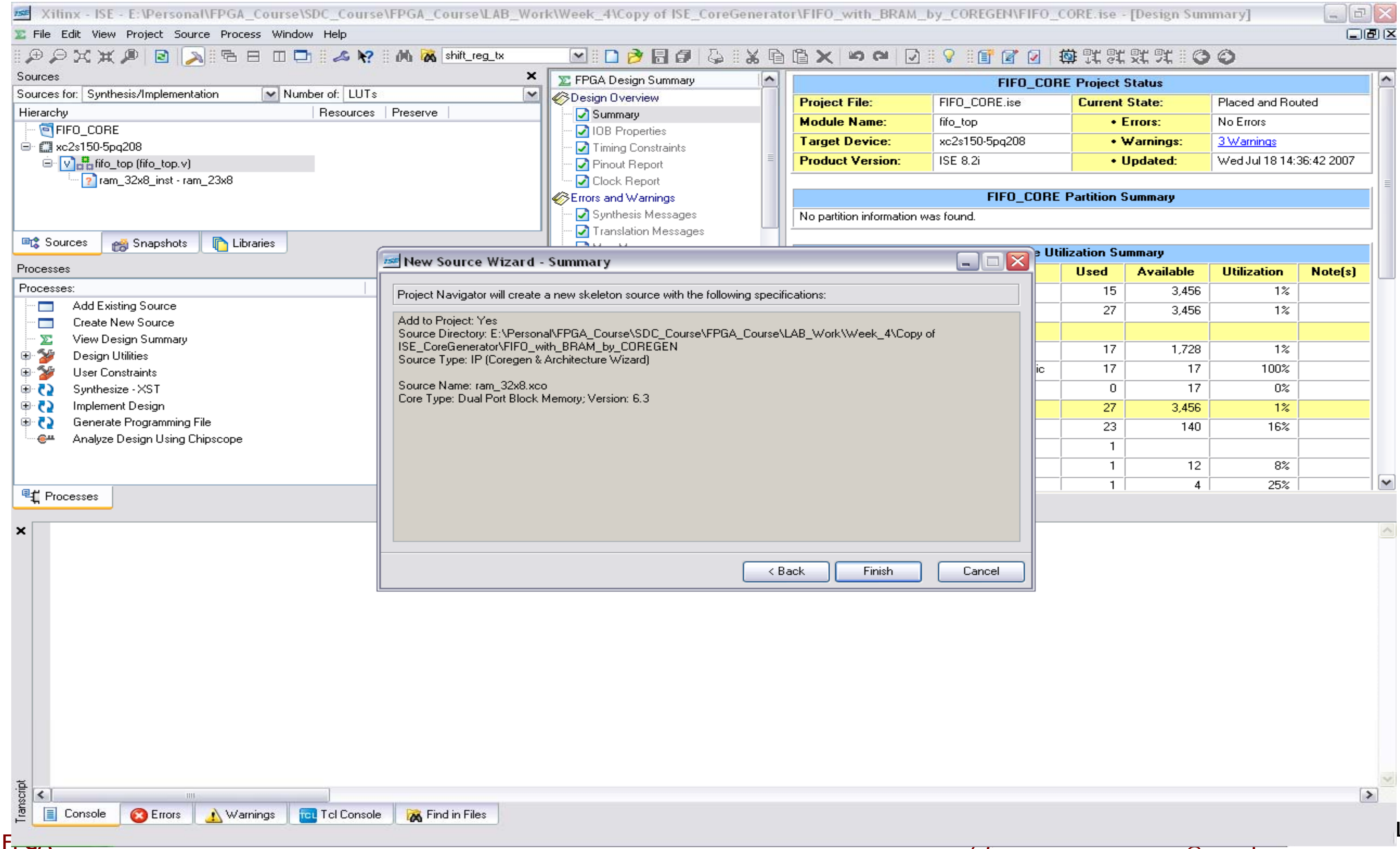
No partition information was found.

e Utilization Summary

	Used	Available	Utilization	Note(s)
	15	3,456	1%	
	27	3,456	1%	
	17	1,728	1%	
ic	17	17	100%	
	0	17	0%	
	27	3,456	1%	
	23	140	16%	
	1			
	1	12	8%	
	1	4	25%	

Design Example

- **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_COREGEN\FIFO_CORE.ise - [Design Summary]

Sources for: Synthesis/Implementation — Number of: LUTs

Hierarchy:
- FIFO_CORE
  - xc2s150-5pq208
    - fifo_top (fifo_top.v)
      - ram_32x8_inst - ram_23x8

Processes:
- Add Existing Source
- Create New Source
- View Design Summary
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Analyze Design Using Chipscope

FPGA Design Summary
- Design Overview
  - Summary
  - IOB Properties
  - Timing Constraints
  - Pinout Report
  - Clock Report
- Errors and Warnings
  - Synthesis Messages
  - Translation Messages

FIFO_CORE Project Status

Project File:	FIFO_CORE.ise	**Current State:**	Placed and Routed
Module Name:	fifo_top	**• Errors:**	No Errors
Target Device:	xc2s150-5pq208	**• Warnings:**	3 Warnings
Product Version:	ISE 8.2i	**• Updated:**	Wed Jul 18 14:36:42 2007

FIFO_CORE Partition Summary

No partition information was found.

Utilization Summary

	Used	Available	Utilization	Note(s)
	15	3,456	1%	
	27	3,456	1%	
	17	1,728	1%	
ic	17	17	100%	
	0	17	0%	
	27	3,456	1%	
	23	140	16%	
	1			
	1	12	8%	
	1	4	25%	

New Source Wizard - Summary

Project Navigator will create a new skeleton source with the following specifications:

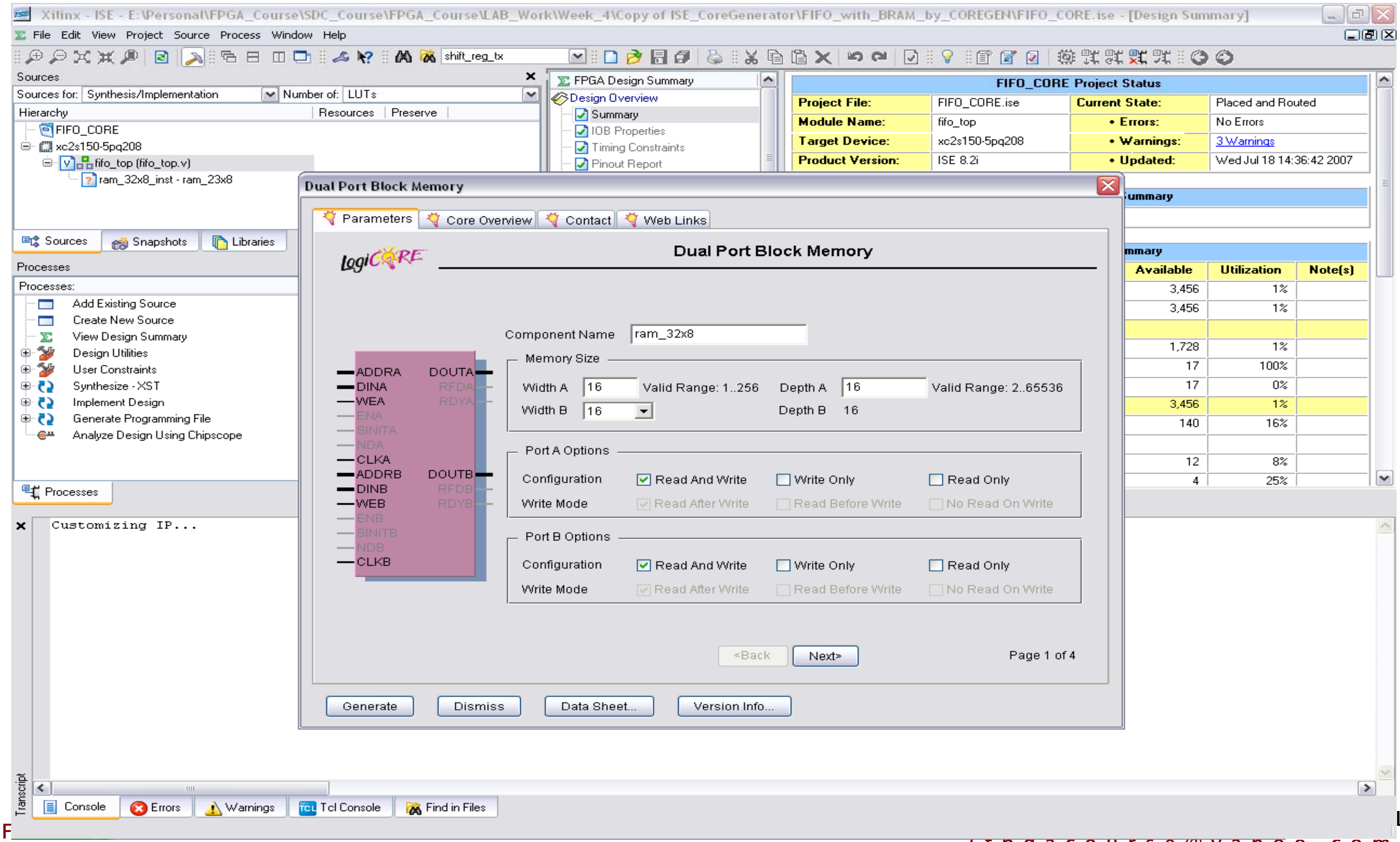
```
Add to Project: Yes
Source Directory: E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of
ISE_CoreGenerator\FIFO_with_BRAM_by_COREGEN
Source Type: IP (Coregen & Architecture Wizard)

Source Name: ram_32x8.xco
Core Type: Dual Port Block Memory; Version: 6.3
```

< Back | Finish | Cancel

Console | Errors | Warnings | Tcl Console | Find in Files

Design Example

- Generate Block RAM Core from Core Generator Tool (ISE 8.2i)



Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_COREGEN\FIFO_CORE.ise - [Design Summary]

Sources for: Synthesis/Implementation — Number of: LUTs

Hierarchy:
- FIFO_CORE
  - xc2s150-5pq208
    - fifo_top (fifo_top.v)
      - ram_32x8_inst - ram_23x8

Processes:
- Add Existing Source
- Create New Source
- View Design Summary
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Analyze Design Using Chipscope

FPGA Design Summary: Design Overview — Summary, IOB Properties, Timing Constraints, Pinout Report

FIFO_CORE Project Status

Project File:	FIFO_CORE.ise	Current State:	Placed and Routed
Module Name:	fifo_top	• Errors:	No Errors
Target Device:	xc2s150-5pq208	• Warnings:	3 Warnings
Product Version:	ISE 8.2i	• Updated:	Wed Jul 18 14:36:42 2007

Available	Utilization	Note(s)
3,456	1%	
3,456	1%	
1,728	1%	
17	100%	
17	0%	
3,456	1%	
140	16%	
12	8%	
4	25%	

Dual Port Block Memory

Component Name: ram_32x8

Memory Size:
- Width A: 16 — Valid Range: 1..256 — Depth A: 16 — Valid Range: 2..65536
- Width B: 16 — Depth B: 16

Port A Options:
- Configuration: [✓ Read And Write] Write Only, Read Only
- Write Mode: [✓ Read After Write] Read Before Write, No Read On Write

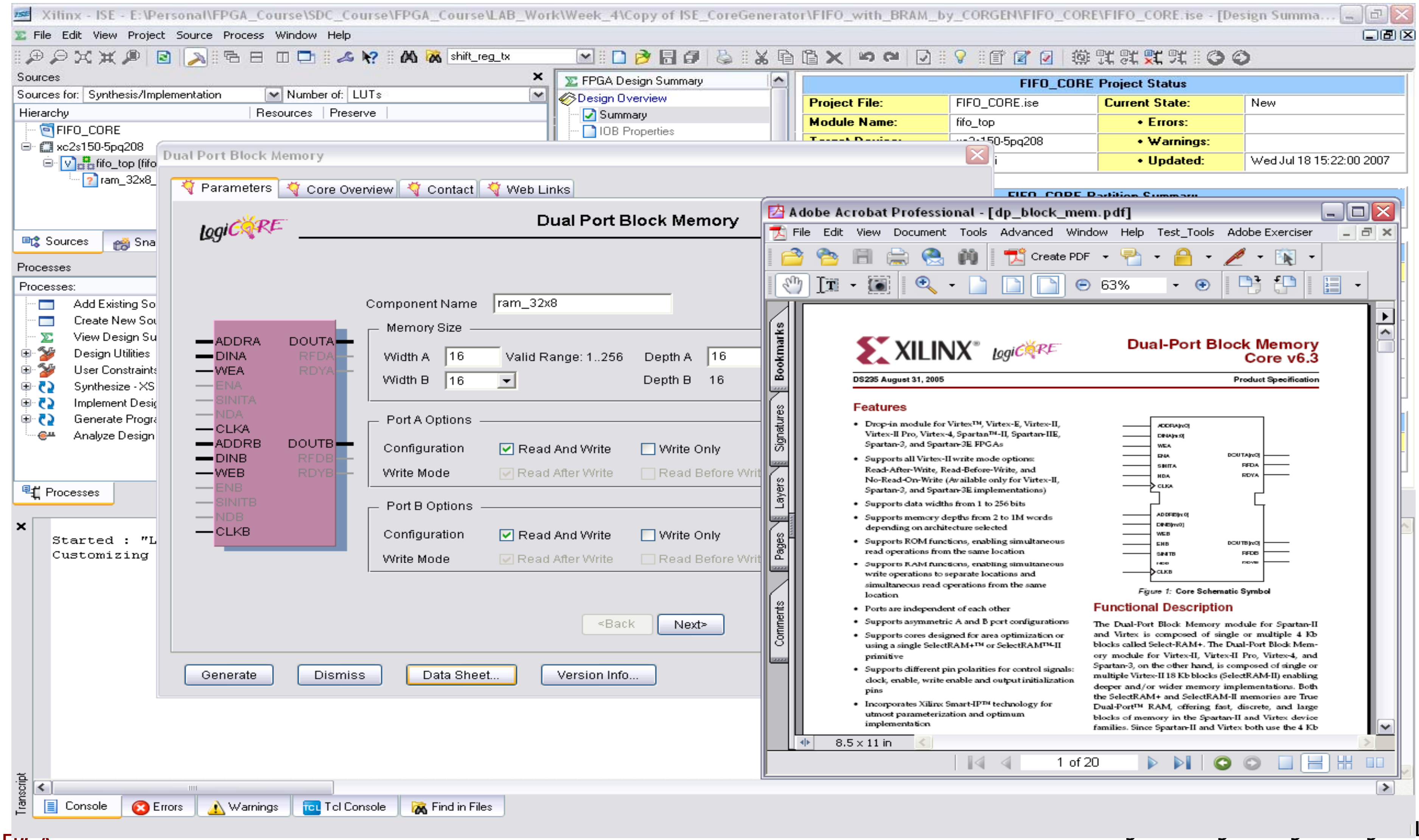
Port B Options:
- Configuration: [✓ Read And Write] Write Only, Read Only
- Write Mode: [✓ Read After Write] Read Before Write, No Read On Write

Page 1 of 4

```
Customizing IP...
```

Design Example

- **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_CORGEN\FIFO_CORE\FIFO_CORE.ise - [Design Summa...

Sources for: Synthesis/Implementation — Number of: LUTs

Hierarchy: FIFO_CORE / xc2s150-5pq208 / fifo_top (fifo... / ram_32x8...

FIFO_CORE Project Status			
Project File:	FIFO_CORE.ise	Current State:	New
Module Name:	fifo_top	• Errors:	
Target Device:	xc2s150-5pq208	• Warnings:	
		• Updated:	Wed Jul 18 15:22:00 2007

Dual Port Block Memory

Component Name: ram_32x8

Memory Size

Width A	16	Valid Range: 1..256	Depth A	16
Width B	16		Depth B	16

Port A Options
- Configuration: [✓ Read And Write] Write Only
- Write Mode: [✓ Read After Write] Read Before Write

Port B Options
- Configuration: [✓ Read And Write] Write Only
- Write Mode: [✓ Read After Write] Read Before Write

```
Started : "L...
Customizing
```

Adobe Acrobat Professional - [dp_block_mem.pdf]

Dual-Port Block Memory Core v6.3

DS235 August 31, 2005 — Product Specification

Features

- Drop-in module for Virtex™, Virtex-E, Virtex-II, Virtex-II Pro, Virtex-4, Spartan™-II, Spartan-IIE, Spartan-3, and Spartan-3E FPGAs
- Supports all Virtex-II write mode options: Read-After-Write, Read-Before-Write, and No-Read-On-Write (Available only for Virtex-II, Spartan-3, and Spartan-3E implementations)
- Supports data widths from 1 to 256 bits
- Supports memory depths from 2 to 1M words depending on architecture selected
- Supports ROM functions, enabling simultaneous read operations from the same location
- Supports RAM functions, enabling simultaneous write operations to separate locations and simultaneous read operations from the same location
- Ports are independent of each other
- Supports asymmetric A and B port configurations
- Supports cores designed for area optimization or using a single SelectRAM+™ or SelectRAM™-II primitive
- Supports different pin polarities for control signals: clock, enable, write enable and output initialization pins
- Incorporates Xilinx Smart-IP™ technology for utmost parameterization and optimum implementation

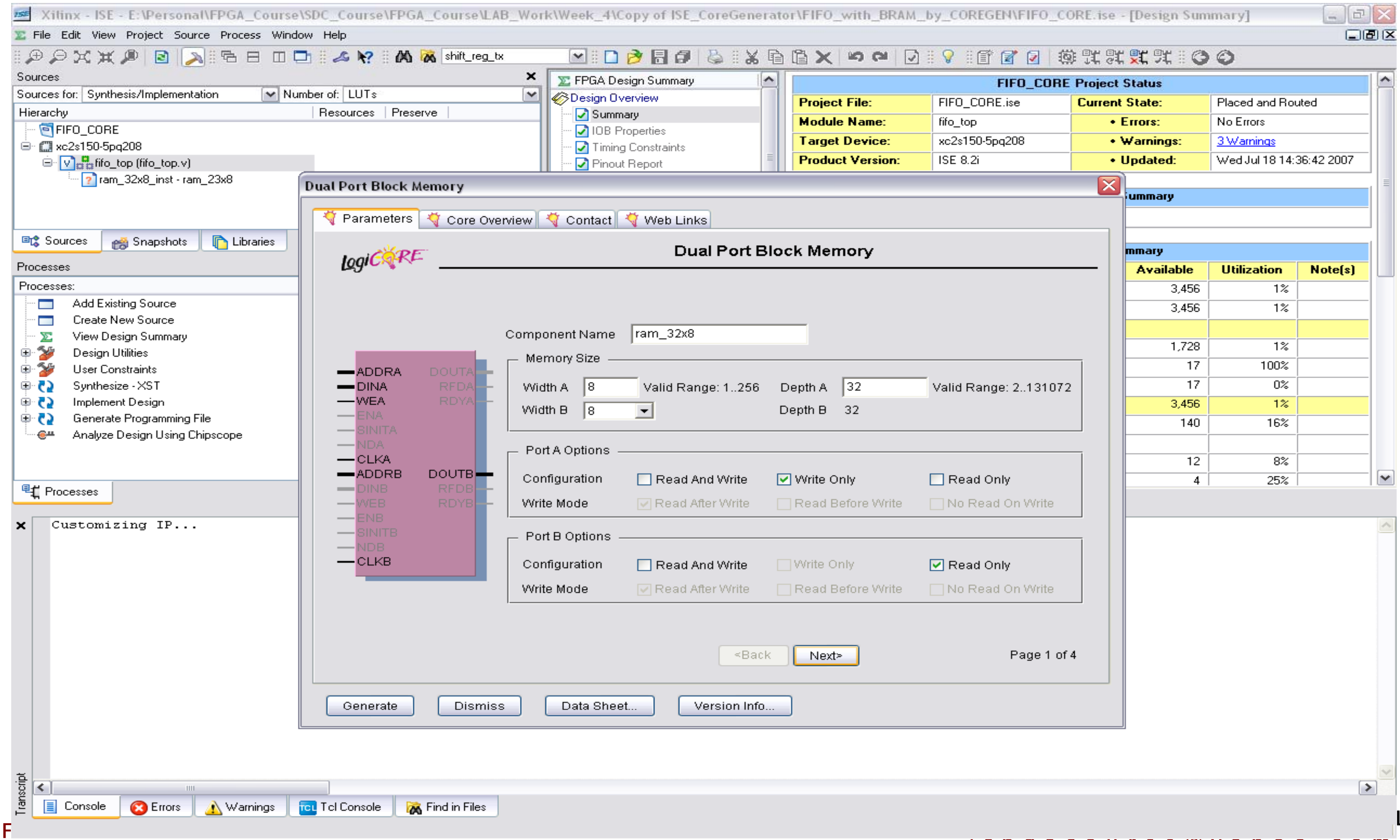
Figure 1: Core Schematic Symbol

Functional Description

The Dual-Port Block Memory module for Spartan-II and Virtex is composed of single or multiple 4 Kb blocks called Select-RAM+. The Dual-Port Block Memory module for Virtex-II, Virtex-II Pro, Virtex-4, and Spartan-3, on the other hand, is composed of single or multiple Virtex-II 18 Kb blocks (SelectRAM-II) enabling deeper and/or wider memory implementations. Both the SelectRAM+ and SelectRAM-II memories are True Dual-Port™ RAM, offering fast, discrete, and large blocks of memory in the Spartan-II and Virtex device families. Since Spartan-II and Virtex both use the 4 Kb

Design Example

➤ **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_COREGEN\FIFO_CORE.ise - [Design Summary]

File Edit View Project Source Process Window Help

Sources

Sources for: Synthesis/Implementation   Number of: LUTs

Hierarchy   Resources   Preserve

```
FIFO_CORE
  xc2s150-5pq208
    fifo_top (fifo_top.v)
      ram_32x8_inst - ram_23x8
```

Sources   Snapshots   Libraries

Processes

Processes:

- Add Existing Source
- Create New Source
- View Design Summary
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Analyze Design Using Chipscope

FPGA Design Summary

- Design Overview
  - Summary
  - IOB Properties
  - Timing Constraints
  - Pinout Report

FIFO_CORE Project Status

Project File:	FIFO_CORE.ise	**Current State:**	Placed and Routed
Module Name:	fifo_top	**• Errors:**	No Errors
Target Device:	xc2s150-5pq208	**• Warnings:**	3 Warnings
Product Version:	ISE 8.2i	**• Updated:**	Wed Jul 18 14:36:42 2007

ummary

mmary

Available	Utilization	Note(s)
3,456	1%	
3,456	1%	
1,728	1%	
17	100%	
17	0%	
3,456	1%	
140	16%	
12	8%	
4	25%	

Dual Port Block Memory

Parameters   Core Overview   Contact   Web Links

LogiCORE   **Dual Port Block Memory**

Component Name: ram_32x8

Memory Size

Width A	8	Valid Range: 1..256	Depth A	32	Valid Range: 2..131072
Width B	8		Depth B	32	

Port A Options

Configuration	☐ Read And Write	☑ Write Only	☐ Read Only
Write Mode	☑ Read After Write	☐ Read Before Write	☐ No Read On Write

Port B Options

Configuration	☐ Read And Write	☐ Write Only	☑ Read Only
Write Mode	☑ Read After Write	☐ Read Before Write	☐ No Read On Write

<Back   Next>   Page 1 of 4

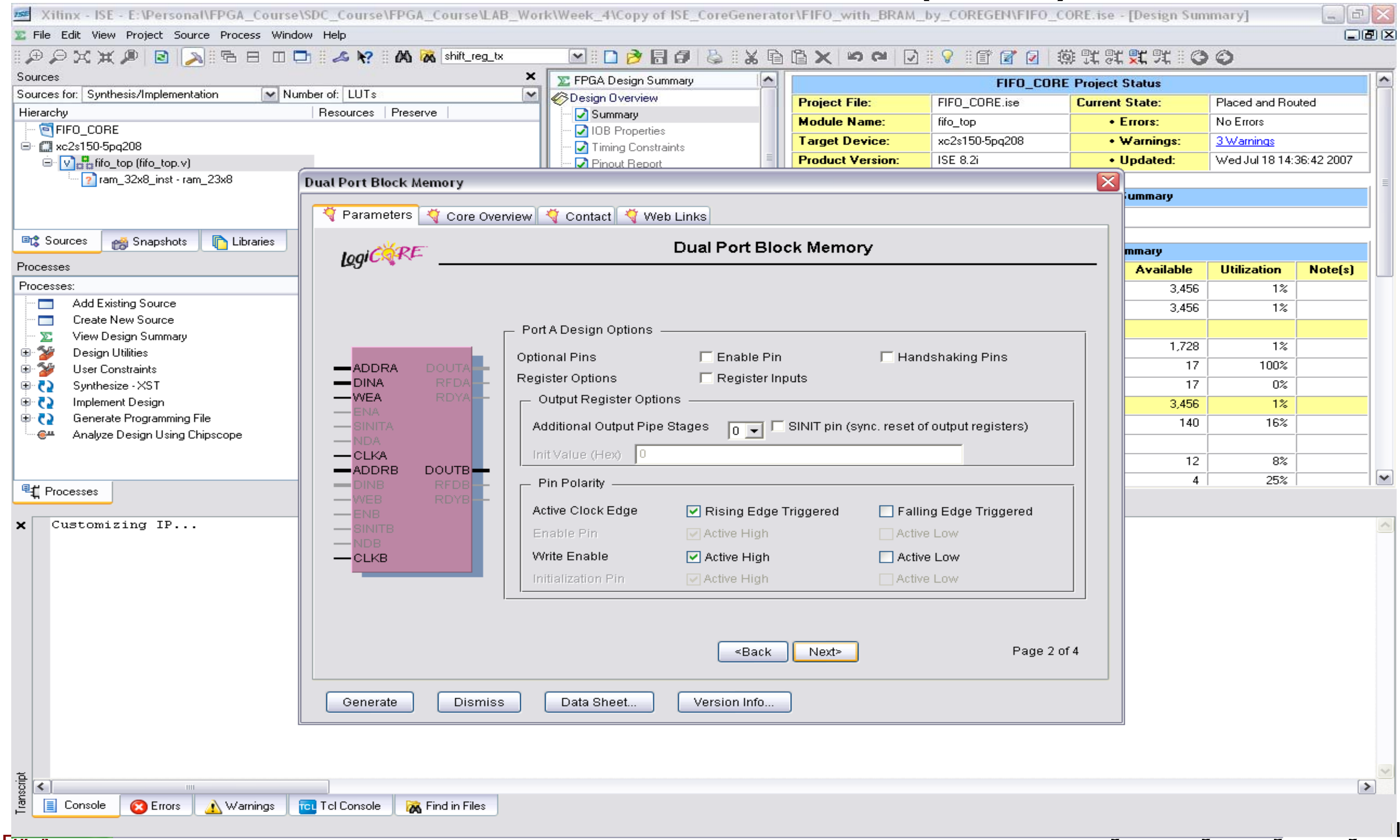
Generate   Dismiss   Data Sheet...   Version Info...

```
Customizing IP...
```

Console   Errors   Warnings   Tcl Console   Find in Files

Design Example

- **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_COREGEN\FIFO_CORE.ise - [Design Summary]

Sources for: Synthesis/Implementation — Number of: LUTs

Hierarchy:
- FIFO_CORE
- xc2s150-5pq208
  - fifo_top (fifo_top.v)
    - ram_32x8_inst - ram_23x8

Processes:
- Add Existing Source
- Create New Source
- View Design Summary
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Analyze Design Using Chipscope

```
Customizing IP...
```

FIFO_CORE Project Status			
Project File:	FIFO_CORE.ise	Current State:	Placed and Routed
Module Name:	fifo_top	• Errors:	No Errors
Target Device:	xc2s150-5pq208	• Warnings:	3 Warnings
Product Version:	ISE 8.2i	• Updated:	Wed Jul 18 14:36:42 2007

Available	Utilization	Note(s)
3,456	1%	
3,456	1%	
1,728	1%	
17	100%	
17	0%	
3,456	1%	
140	16%	
12	8%	
4	25%	

Dual Port Block Memory

Parameters | Core Overview | Contact | Web Links

Dual Port Block Memory

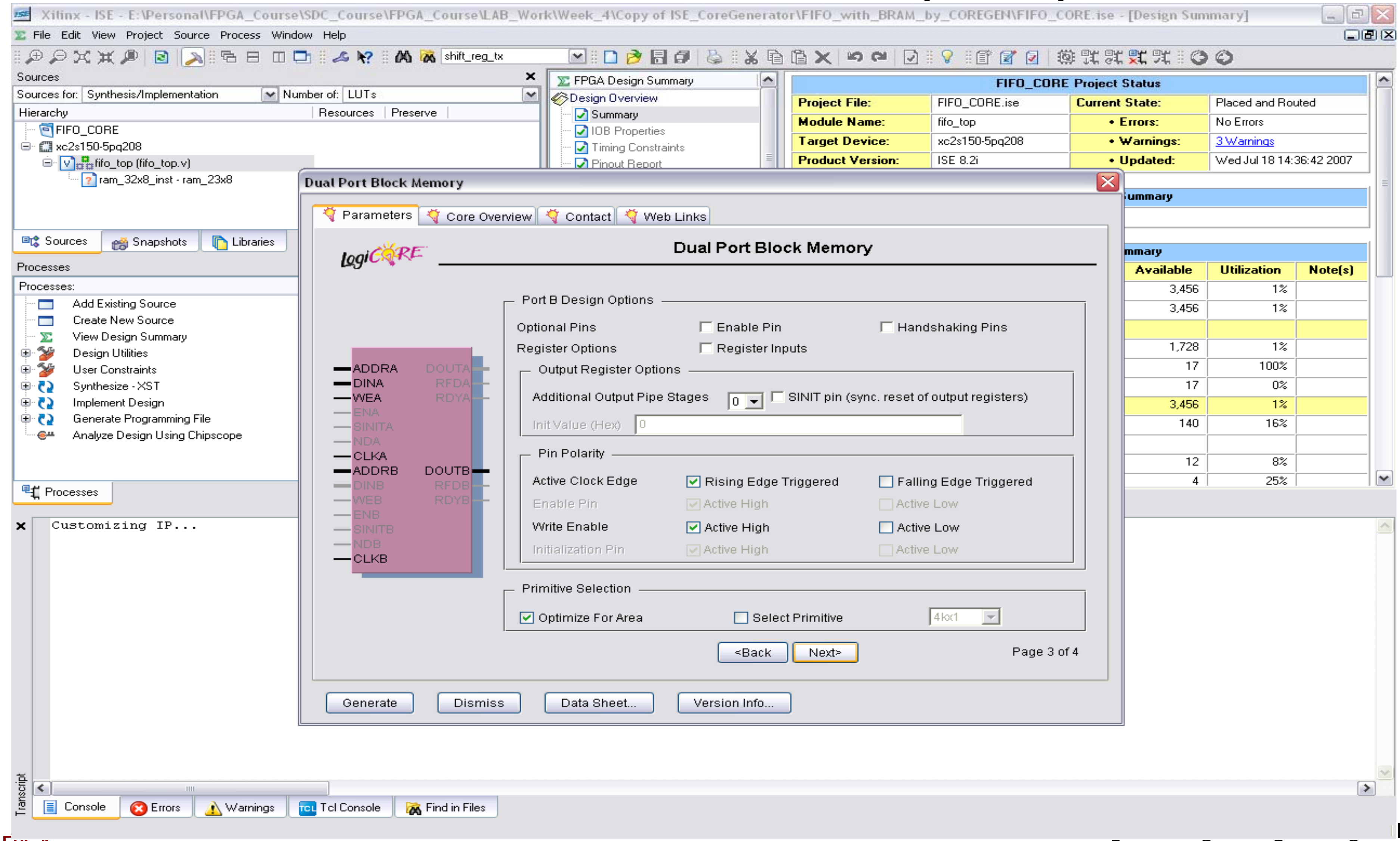
Port A Design Options
- Optional Pins: [ ] Enable Pin [ ] Handshaking Pins
- Register Options: [ ] Register Inputs
- Output Register Options
  - Additional Output Pipe Stages: 0 [ ] SINIT pin (sync. reset of output registers)
  - Init Value (Hex): 0

Pin Polarity
- Active Clock Edge: [✓] Rising Edge Triggered [ ] Falling Edge Triggered
- Enable Pin: [✓] Active High [ ] Active Low
- Write Enable: [✓] Active High [ ] Active Low
- Initialization Pin: [✓] Active High [ ] Active Low

Page 2 of 4

Design Example

➢ **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_COREGEN\FIFO_CORE.ise - [Design Summary]

Sources

Sources for: Synthesis/Implementation | Number of: LUTs

Hierarchy | Resources | Preserve

- FIFO_CORE
  - xc2s150-5pq208
    - fifo_top (fifo_top.v)
      - ram_32x8_inst - ram_23x8

Processes

Processes:

- Add Existing Source
- Create New Source
- View Design Summary
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Analyze Design Using Chipscope

FPGA Design Summary
- Design Overview
  - Summary
  - IOB Properties
  - Timing Constraints
  - Pinout Report

FIFO_CORE Project Status			
Project File:	FIFO_CORE.ise	**Current State:**	Placed and Routed
Module Name:	fifo_top	**• Errors:**	No Errors
Target Device:	xc2s150-5pq208	**• Warnings:**	3 Warnings
Product Version:	ISE 8.2i	**• Updated:**	Wed Jul 18 14:36:42 2007

ummary

mmary

Available	Utilization	Note(s)
3,456	1%	
3,456	1%	
1,728	1%	
17	100%	
17	0%	
3,456	1%	
140	16%	
12	8%	
4	25%	

Dual Port Block Memory

Dual Port Block Memory

Port B Design Options

Optional Pins: ☐ Enable Pin ☐ Handshaking Pins

Register Options: ☐ Register Inputs

Output Register Options

Additional Output Pipe Stages: 0 ☐ SINIT pin (sync. reset of output registers)

Init Value (Hex): 0

Pin Polarity

Active Clock Edge: ☑ Rising Edge Triggered ☐ Falling Edge Triggered

Enable Pin: ☑ Active High ☐ Active Low

Write Enable: ☑ Active High ☐ Active Low

Initialization Pin: ☑ Active High ☐ Active Low

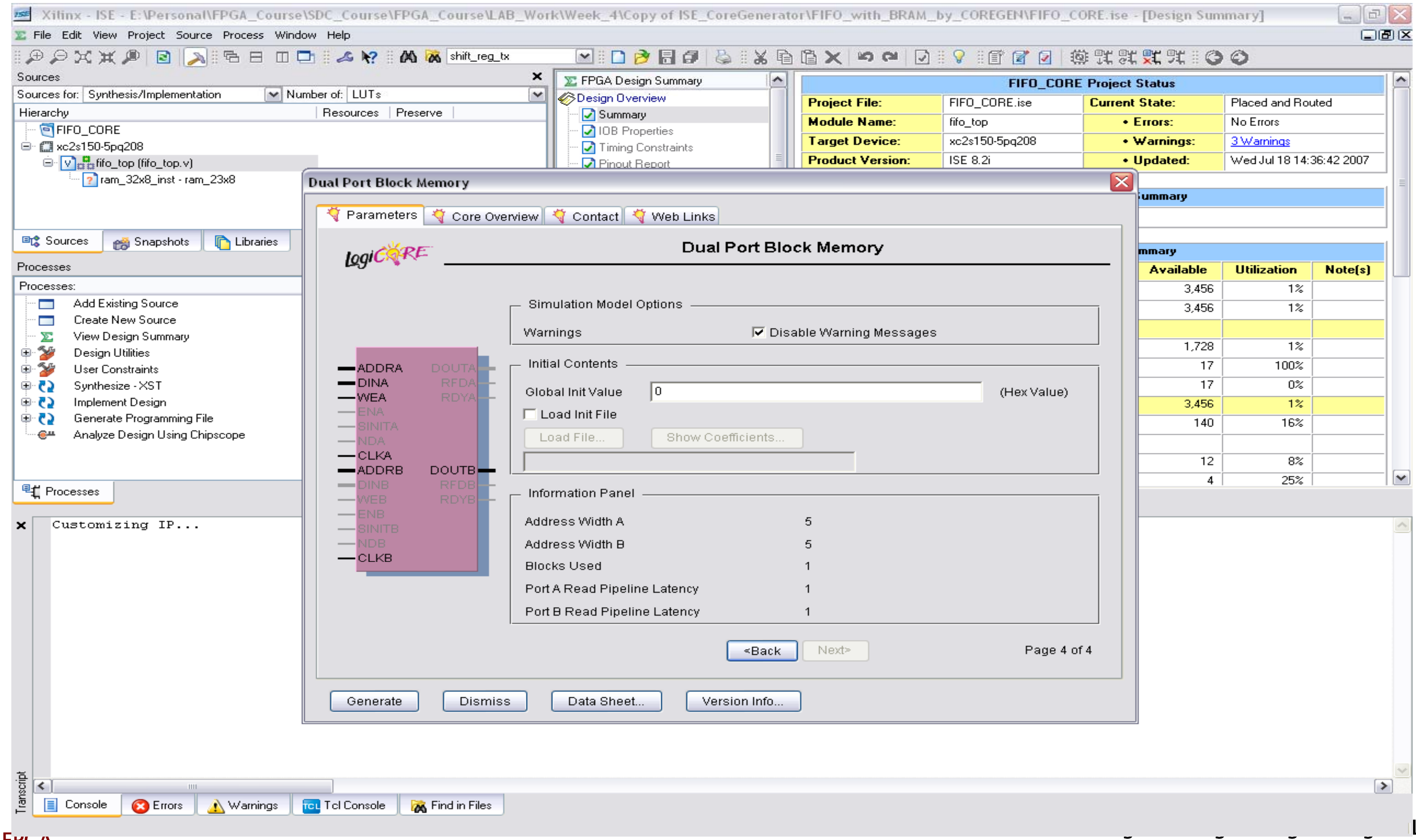
Primitive Selection

☑ Optimize For Area ☐ Select Primitive 4kx1

```
Customizing IP...
```

Design Example

- **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



- FIFO_CORE
  - xc2s150-5pq208
    - fifo_top (fifo_top.v)
      - ram_32x8_inst - ram_23x8

Processes:
- Add Existing Source
- Create New Source
- View Design Summary
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Analyze Design Using Chipscope

FIFO_CORE Project Status			
Project File:	FIFO_CORE.ise	Current State:	Placed and Routed
Module Name:	fifo_top	• Errors:	No Errors
Target Device:	xc2s150-5pq208	• Warnings:	3 Warnings
Product Version:	ISE 8.2i	• Updated:	Wed Jul 18 14:36:42 2007

Available	Utilization	Note(s)
3,456	1%	
3,456	1%	
1,728	1%	
17	100%	
17	0%	
3,456	1%	
140	16%	
12	8%	
4	25%	

Dual Port Block Memory

Dual Port Block Memory

Simulation Model Options

Warnings [✓] Disable Warning Messages

Initial Contents

Global Init Value: 0 (Hex Value)

[ ] Load Init File

Information Panel

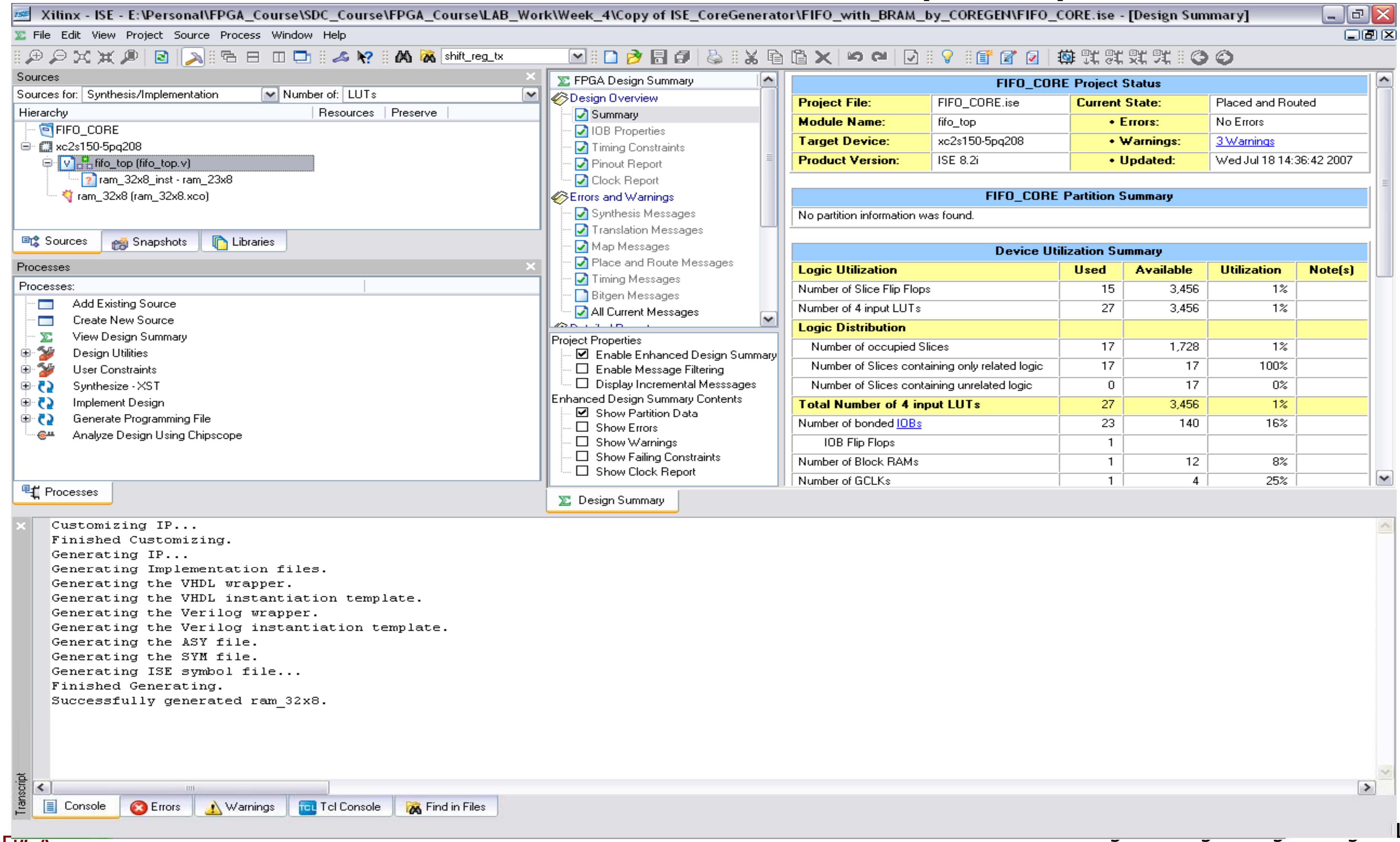
Address Width A	5
Address Width B	5
Blocks Used	1
Port A Read Pipeline Latency	1
Port B Read Pipeline Latency	1

Page 4 of 4

```
Customizing IP...
```

Design Example

- **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



Project Properties
- [✓] Enable Enhanced Design Summary
- [ ] Enable Message Filtering
- [ ] Display Incremental Messages

Enhanced Design Summary Contents
- [✓] Show Partition Data
- [ ] Show Errors
- [ ] Show Warnings
- [ ] Show Failing Constraints
- [ ] Show Clock Report

FIFO_CORE Project Status

Project File:	FIFO_CORE.ise	**Current State:**	Placed and Routed
Module Name:	fifo_top	**• Errors:**	No Errors
Target Device:	xc2s150-5pq208	**• Warnings:**	3 Warnings
Product Version:	ISE 8.2i	**• Updated:**	Wed Jul 18 14:36:42 2007

FIFO_CORE Partition Summary

No partition information was found.

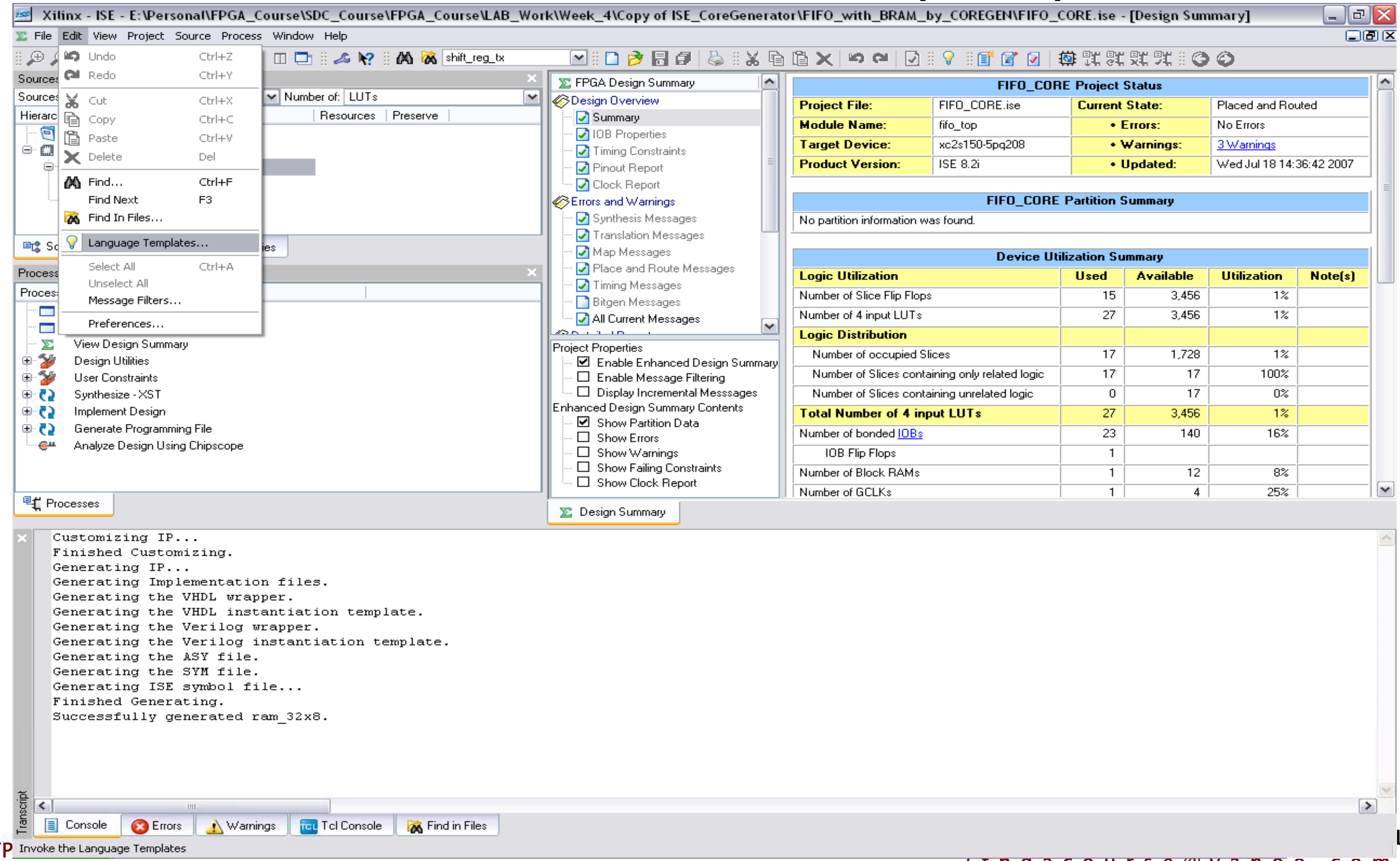
Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	15	3,456	1%	
Number of 4 input LUTs	27	3,456	1%	
Logic Distribution				
Number of occupied Slices	17	1,728	1%	
Number of Slices containing only related logic	17	17	100%	
Number of Slices containing unrelated logic	0	17	0%	
Total Number of 4 input LUTs	27	3,456	1%	
Number of bonded IOBs	23	140	16%	
IOB Flip Flops	1			
Number of Block RAMs	1	12	8%	
Number of GCLKs	1	4	25%	

```
Customizing IP...
Finished Customizing.
Generating IP...
Generating Implementation files.
Generating the VHDL wrapper.
Generating the VHDL instantiation template.
Generating the Verilog wrapper.
Generating the Verilog instantiation template.
Generating the ASY file.
Generating the SYM file.
Generating ISE symbol file...
Finished Generating.
Successfully generated ram_32x8.
```

Design Example

- **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



Project Properties
- [✓] Enable Enhanced Design Summary
- [ ] Enable Message Filtering
- [ ] Display Incremental Messsages

Enhanced Design Summary Contents
- [✓] Show Partition Data
- [ ] Show Errors
- [ ] Show Warnings
- [ ] Show Failing Constraints
- [ ] Show Clock Report

FIFO_CORE Project Status

Project File:	FIFO_CORE.ise	**Current State:**	Placed and Routed
Module Name:	fifo_top	**• Errors:**	No Errors
Target Device:	xc2s150-5pq208	**• Warnings:**	3 Warnings
Product Version:	ISE 8.2i	**• Updated:**	Wed Jul 18 14:36:42 2007

FIFO_CORE Partition Summary

No partition information was found.

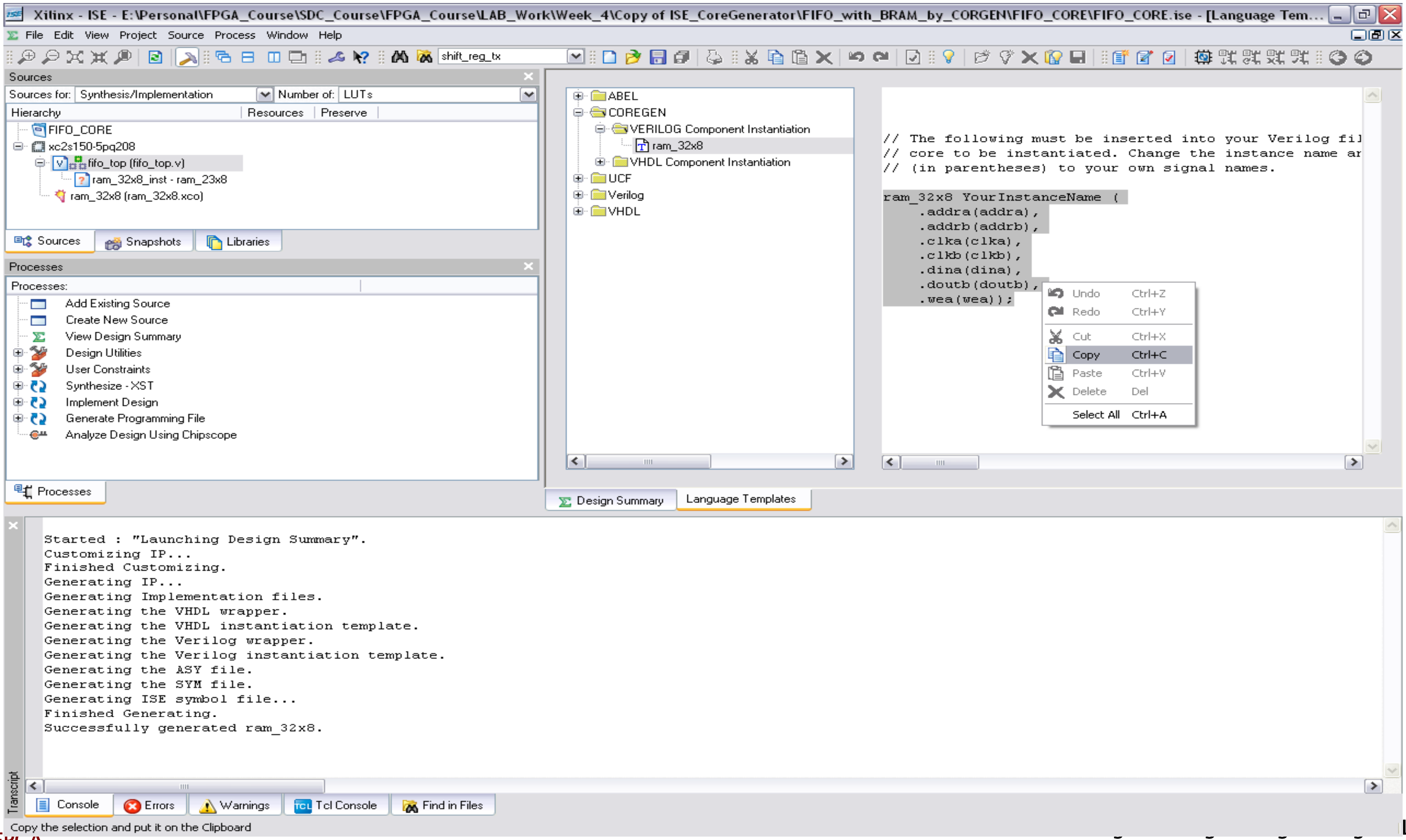
Device Utilization Summary

Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	15	3,456	1%	
Number of 4 input LUTs	27	3,456	1%	
Logic Distribution				
Number of occupied Slices	17	1,728	1%	
Number of Slices containing only related logic	17	17	100%	
Number of Slices containing unrelated logic	0	17	0%	
Total Number of 4 input LUTs	27	3,456	1%	
Number of bonded IOBs	23	140	16%	
IOB Flip Flops	1			
Number of Block RAMs	1	12	8%	
Number of GCLKs	1	4	25%	

```
Customizing IP...
Finished Customizing.
Generating IP...
Generating Implementation files.
Generating the VHDL wrapper.
Generating the VHDL instantiation template.
Generating the Verilog wrapper.
Generating the Verilog instantiation template.
Generating the ASY file.
Generating the SYM file.
Generating ISE symbol file...
Finished Generating.
Successfully generated ram_32x8.
```

Design Example

- **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



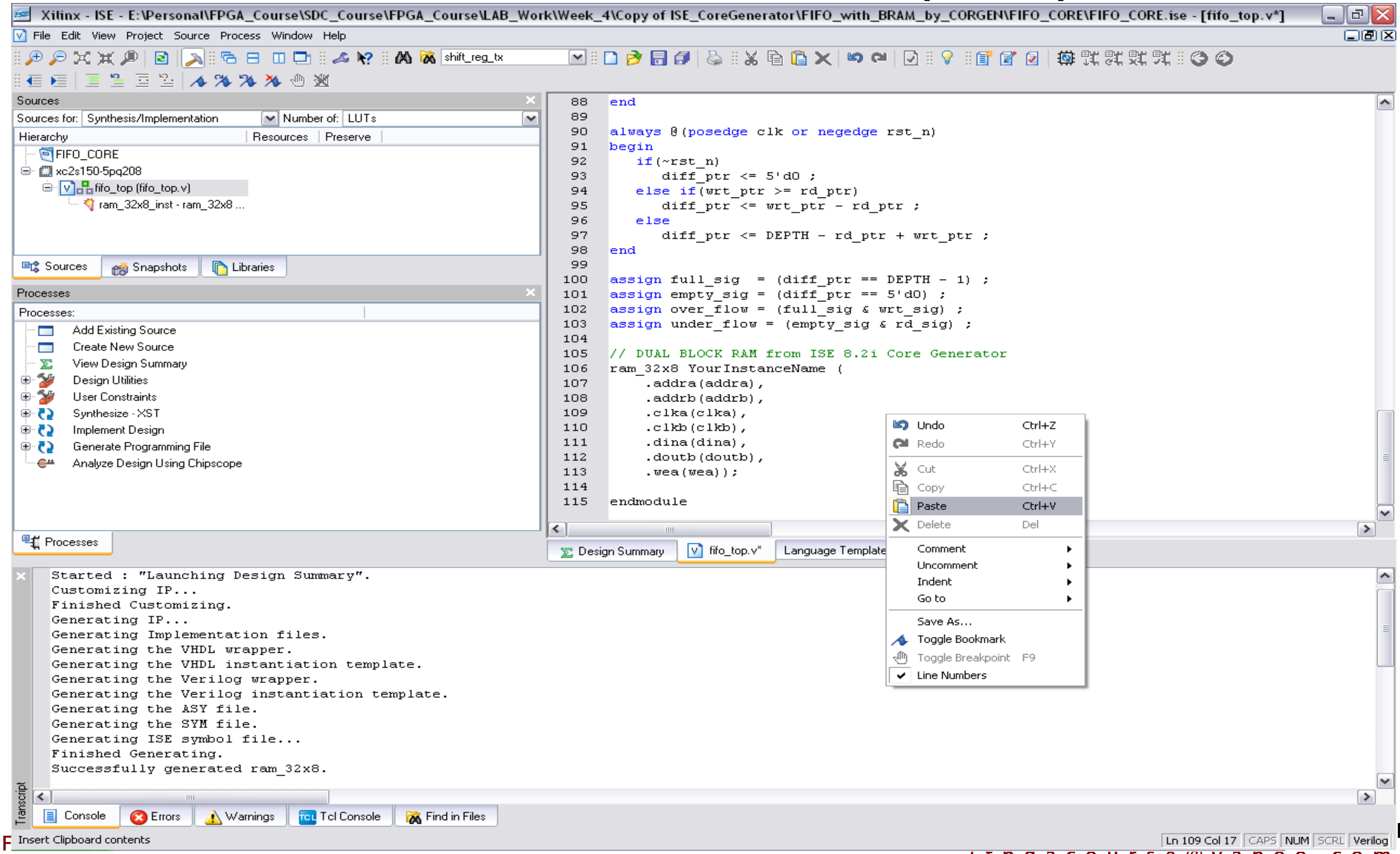
```
// The following must be inserted into your Verilog fil
// core to be instantiated. Change the instance name ar
// (in parentheses) to your own signal names.

ram_32x8 YourInstanceName (
    .addra(addra),
    .addrb(addrb),
    .clka(clka),
    .clkb(clkb),
    .dina(dina),
    .doutb(doutb),
    .wea(wea));
```

```
Started : "Launching Design Summary".
Customizing IP...
Finished Customizing.
Generating IP...
Generating Implementation files.
Generating the VHDL wrapper.
Generating the VHDL instantiation template.
Generating the Verilog wrapper.
Generating the Verilog instantiation template.
Generating the ASY file.
Generating the SYM file.
Generating ISE symbol file...
Finished Generating.
Successfully generated ram_32x8.
```

Copy the selection and put it on the Clipboard

Design Example

➢ **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**

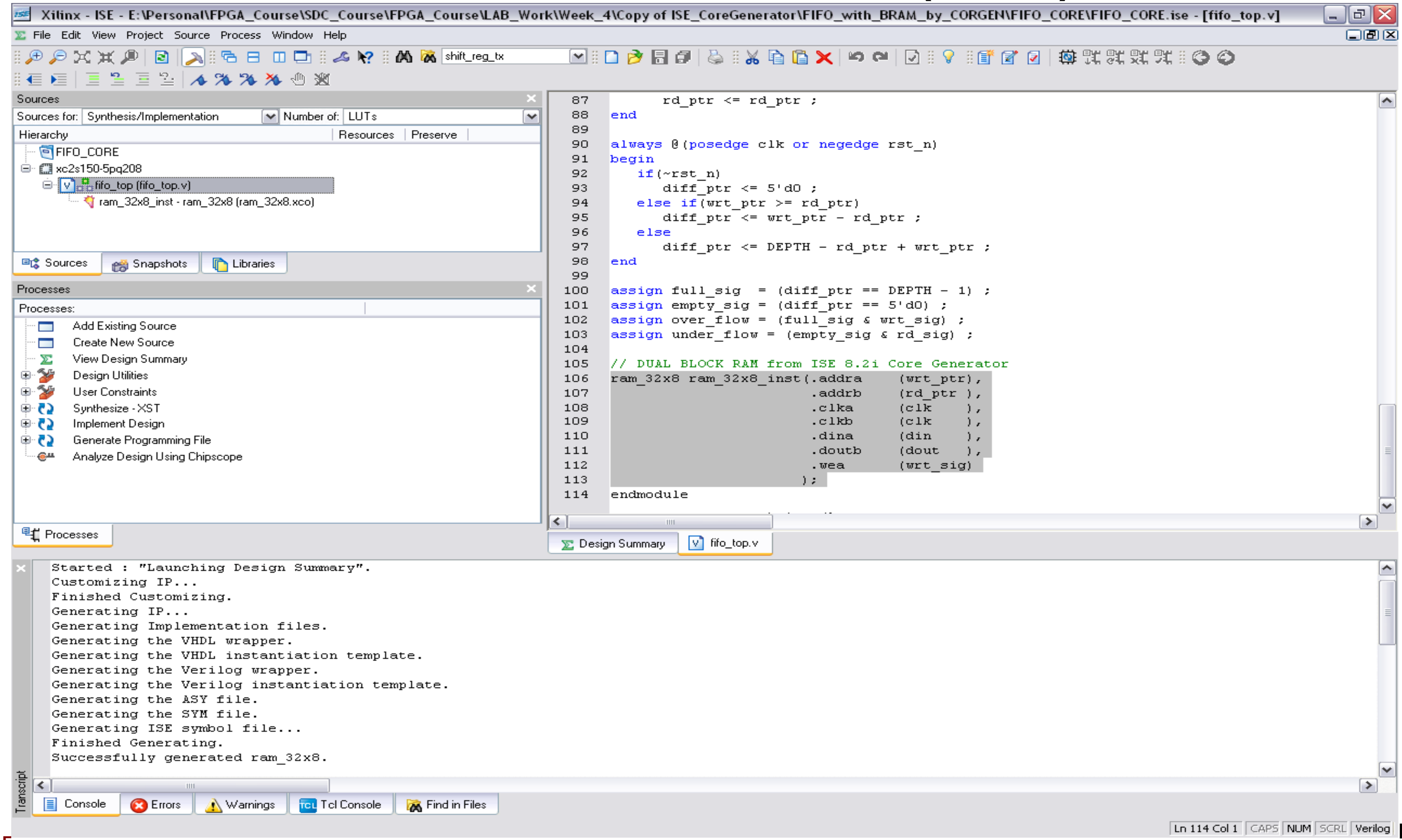


```verilog
end

always @(posedge clk or negedge rst_n)
begin
   if(~rst_n)
      diff_ptr <= 5'd0 ;
   else if(wrt_ptr >= rd_ptr)
      diff_ptr <= wrt_ptr - rd_ptr ;
   else
      diff_ptr <= DEPTH - rd_ptr + wrt_ptr ;
end

assign full_sig  = (diff_ptr == DEPTH - 1) ;
assign empty_sig = (diff_ptr == 5'd0) ;
assign over_flow = (full_sig & wrt_sig) ;
assign under_flow = (empty_sig & rd_sig) ;

// DUAL BLOCK RAM from ISE 8.2i Core Generator
ram_32x8 YourInstanceName (
      .addra(addra),
      .addrb(addrb),
      .clka(clka),
      .clkb(clkb),
      .dina(dina),
      .doutb(doutb),
      .wea(wea));


endmodule
```

```
Started : "Launching Design Summary".
Customizing IP...
Finished Customizing.
Generating IP...
Generating Implementation files.
Generating the VHDL wrapper.
Generating the VHDL instantiation template.
Generating the Verilog wrapper.
Generating the Verilog instantiation template.
Generating the ASY file.
Generating the SYM file.
Generating ISE symbol file...
Finished Generating.
Successfully generated ram_32x8.
```

Design Example

- **Generate Block RAM Core from Core Generator Tool (ISE 8.2i)**



```
      rd_ptr <= rd_ptr ;
end

always @(posedge clk or negedge rst_n)
begin
   if(~rst_n)
      diff_ptr <= 5'd0 ;
   else if(wrt_ptr >= rd_ptr)
      diff_ptr <= wrt_ptr - rd_ptr ;
   else
      diff_ptr <= DEPTH - rd_ptr + wrt_ptr ;
end

assign full_sig  = (diff_ptr == DEPTH - 1) ;
assign empty_sig = (diff_ptr == 5'd0) ;
assign over_flow = (full_sig & wrt_sig) ;
assign under_flow = (empty_sig & rd_sig) ;

// DUAL BLOCK RAM from ISE 8.2i Core Generator
ram_32x8 ram_32x8_inst(.addra     (wrt_ptr),
                       .addrb     (rd_ptr ),
                       .clka      (clk    ),
                       .clkb      (clk    ),
                       .dina      (din    ),
                       .doutb     (dout   ),
                       .wea       (wrt_sig)
                      );
endmodule
```

```
Started : "Launching Design Summary".
Customizing IP...
Finished Customizing.
Generating IP...
Generating Implementation files.
Generating the VHDL wrapper.
Generating the VHDL instantiation template.
Generating the Verilog wrapper.
Generating the Verilog instantiation template.
Generating the ASY file.
Generating the SYM file.
Generating ISE symbol file...
Finished Generating.
Successfully generated ram_32x8.
```

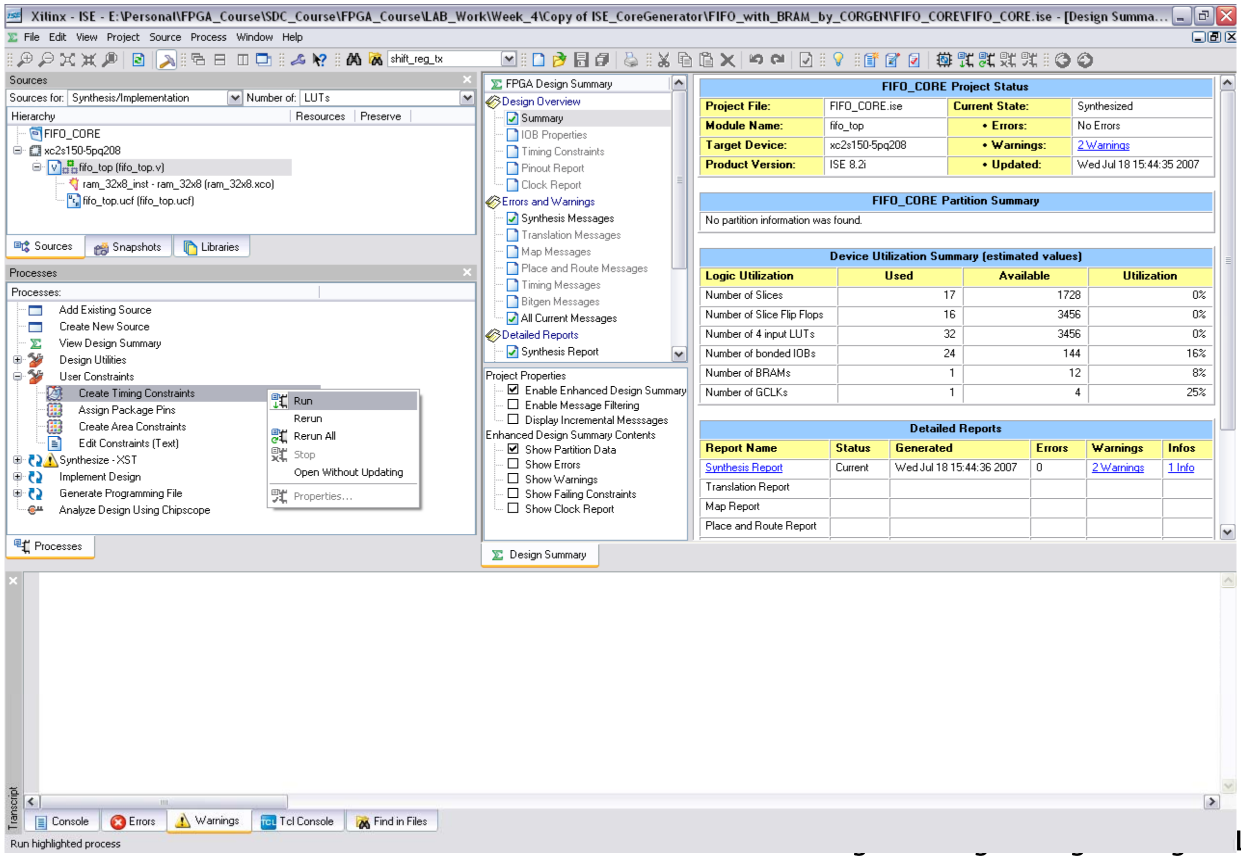
Constraints Editor

- **Constraints Editor:** **The Constraints Editor is a graphical user interface (GUI) tool for entering timing constraints and pin location constraints. The user interface simplifies constraint entry by guiding you through constraint creation without your needing to understand UCF (User Constraint File) file syntax.**

The Constraints Editor interface consists of a main window, four tab windows, a constraints window, an output window, and numerous dialog boxes.

Constraints Editor – contd.

➢ **Starting the Constraints Editor from the Project Navigator: Within the Project Navigator, you can launch the Constraints Editor from the Processes window. First select a design file in the Sources window. Then double-click Create Timing Constraints in the Processes window, which is located within User Constr-aints underneath Design Utilities.**


Xilinx - ISE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_CORGEN\FIFO_CORE\FIFO_CORE.ise - [Design Summa...

FIFO_CORE Project Status

Project File:	FIFO_CORE.ise	**Current State:**	Synthesized
Module Name:	fifo_top	**• Errors:**	No Errors
Target Device:	xc2s150-5pq208	**• Warnings:**	2 Warnings
Product Version:	ISE 8.2i	**• Updated:**	Wed Jul 18 15:44:35 2007

FIFO_CORE Partition Summary

No partition information was found.

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	17	1728	0%
Number of Slice Flip Flops	16	3456	0%
Number of 4 input LUTs	32	3456	0%
Number of bonded IOBs	24	144	16%
Number of BRAMs	1	12	8%
Number of GCLKs	1	4	25%

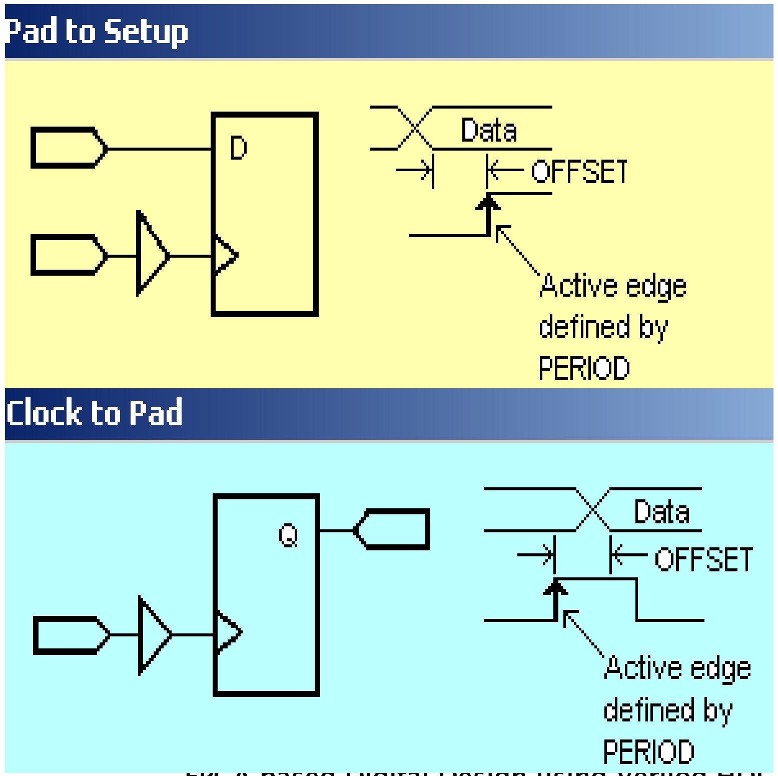
Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Wed Jul 18 15:44:36 2007	0	2 Warnings	1 Info
Translation Report					
Map Report					
Place and Route Report					

Constraints Editor – contd.

2. With the design loaded in the Constraints Editor, go to the Ports tab.
3. Right-click in the Location field corresponding to the desired I/O signal.
4. In the Location dialog box, enter the desired pin location.

- **Pad to Setup / Clock to Pad**
  Specifies the timing relationship between an external clock and data at the pins of a device. Operates on pads or predefined groups of pads.



Constraints Editor – contd.

- **Period**

  Defines a clock period.
- **Location**

  Locks a user-defined port to a device pin.
- **FAST/SLOW**

  Assigns a slew rate to selected port.
- **PULLUP/PULLDOWN**

  Signifies a pull level (PULLUP, PULLDOWN, or KEEPER) for a selected output port.KEEPER is used for Virtex devices only. When a 3-state buffer goes to high impedance,KEEPER keeps the input level of the buffer on the pad net.

Constraints Editor – contd.

- **DRIVE**

  This constraint assigns a signal strength to a selected port.

- **IOSTANDARD**

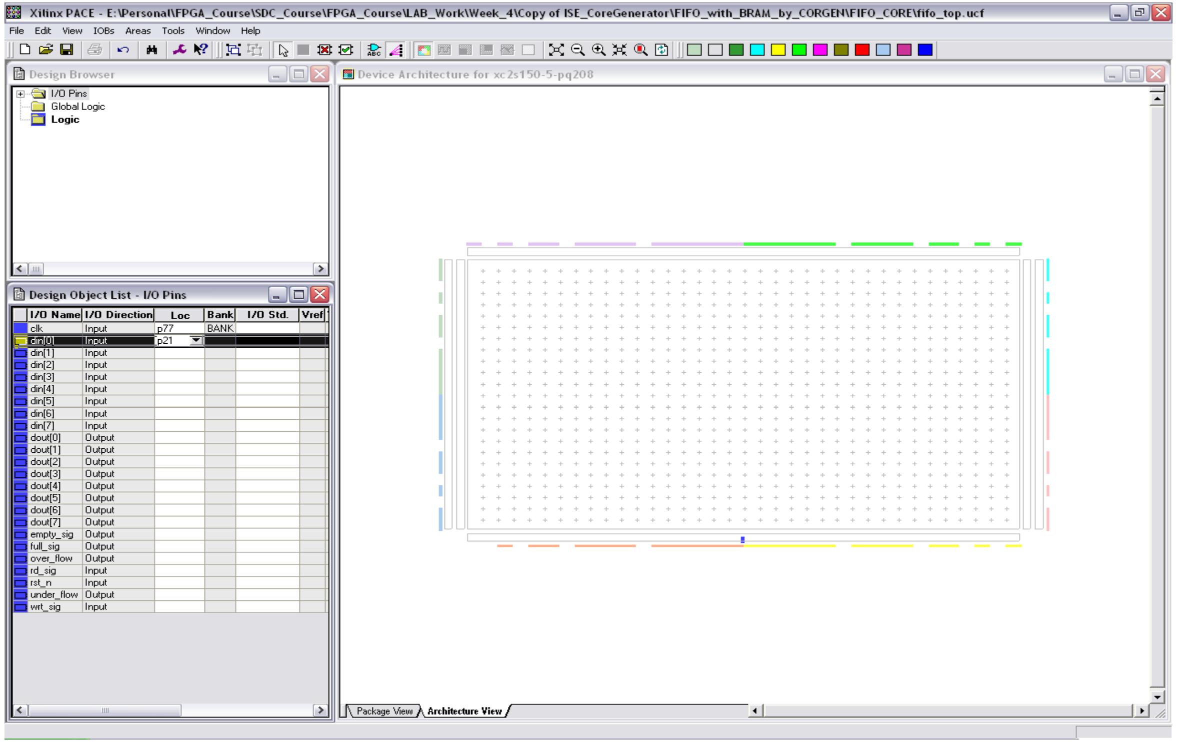
  Assigns an input/output standard (LVTTL, LVCMOS, and so forth) to a selected net attached to the port.

Constraints Editor – contd.

❑ Assigning Pins Using Xilinx PACE Tool..

To assign pin locations using the Xilinx PACE:

1. In the Processes for Source window, expand the User Constraints tree and double-click on the Assign Package Pins to display the Xilinx PAEC Tool.

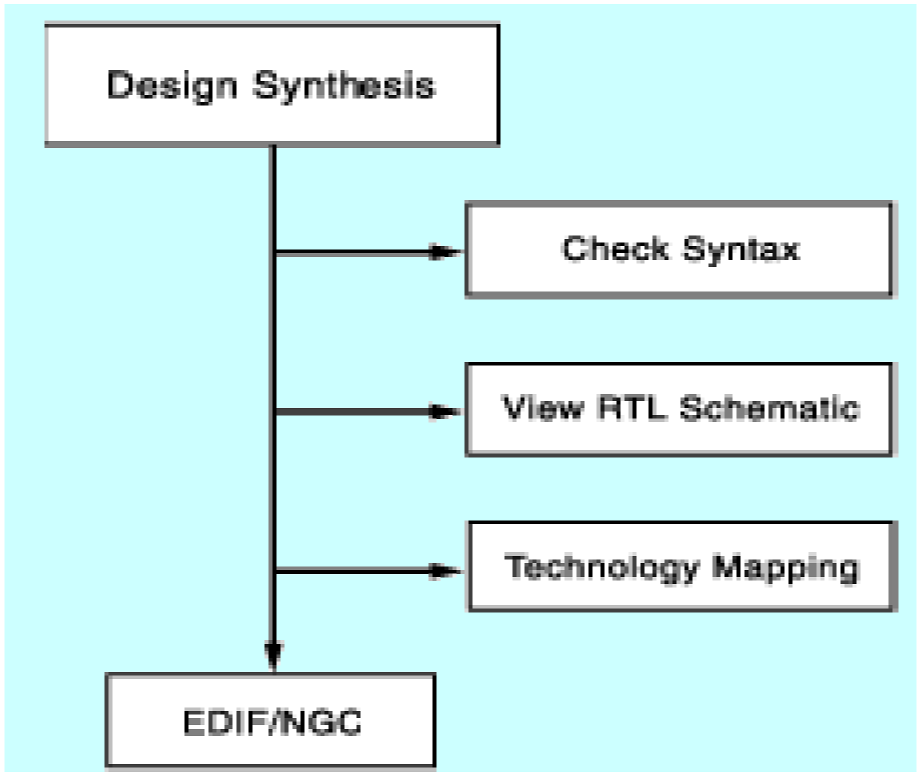

Xilinx PACE - E:\Personal\FPGA_Course\SDC_Course\FPGA_Course\LAB_Work\Week_4\Copy of ISE_CoreGenerator\FIFO_with_BRAM_by_CORGEN\FIFO_CORE\fifo_top.ucf
File
Edit
View
IOBs
Areas
Tools
Window
Help
Design Browser
I/O Pins
Global Logic
Logic
Device Architecture for xc2s150-5-pq208
Design Object List - I/O Pins
Package View
Architecture View

I/O Name	I/O Direction	Loc	Bank	I/O Std.	Vref
clk	Input	p77	BANK		
din[0]	Input	p21			
din[1]	Input				
din[2]	Input				
din[3]	Input				
din[4]	Input				
din[5]	Input				
din[6]	Input				
din[7]	Input				
dout[0]	Output				
dout[1]	Output				
dout[2]	Output				
dout[3]	Output				
dout[4]	Output				
dout[5]	Output				
dout[6]	Output				
dout[7]	Output				
empty_sig	Output				
full_sig	Output				
over_flow	Output				
rd_sig	Input				
rst_n	Input				
under_flow	Output				
wrt_sig	Input				

Design Synthesis

➢ Xilinx Synthesis Technology (XST) Flow:

XST is a Xilinx® tool that synthesizes HDL designs to create Xilinx® specific netlist files called NGC files. The NGC file is a netlist that contains both logical design data and constraints that takes the place of both EDIF and NCF files.

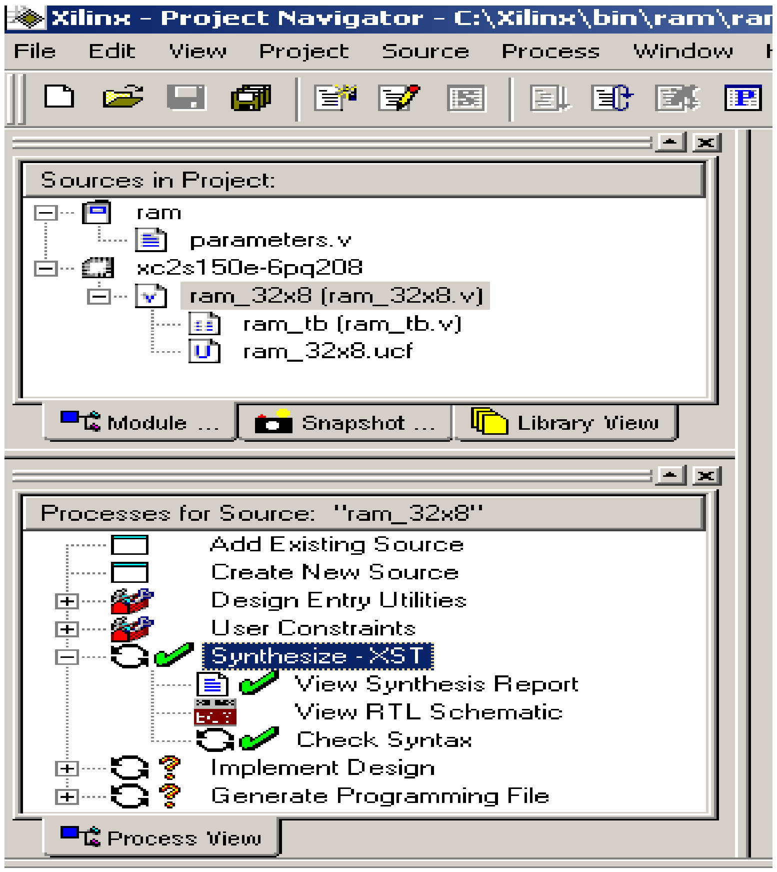


Xilinx Synthesis Technology (XST)

- **XST in Project Navigator:**

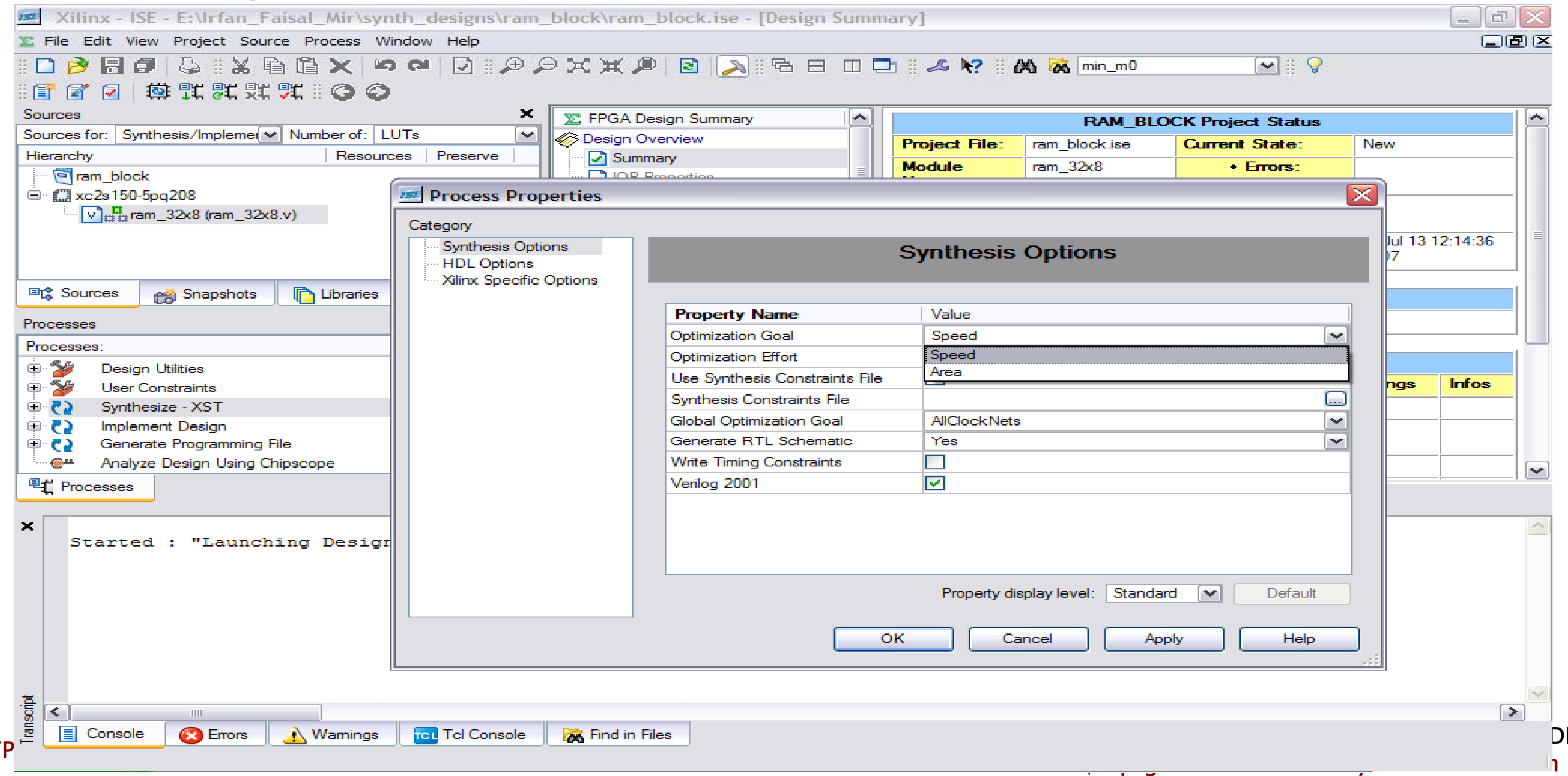
Before you synthesize your design, you can set a variety of options for XST. The following are the instructions to set the options and run XST from Project Navigator.

1. Select your top-level design in the Source window.



Xilinx Synthesis Technology (XST) – contd.

2. To set the options, right-click Synthesize - XST in the Process window.
3. Select Properties to display the Process Properties dialog box.



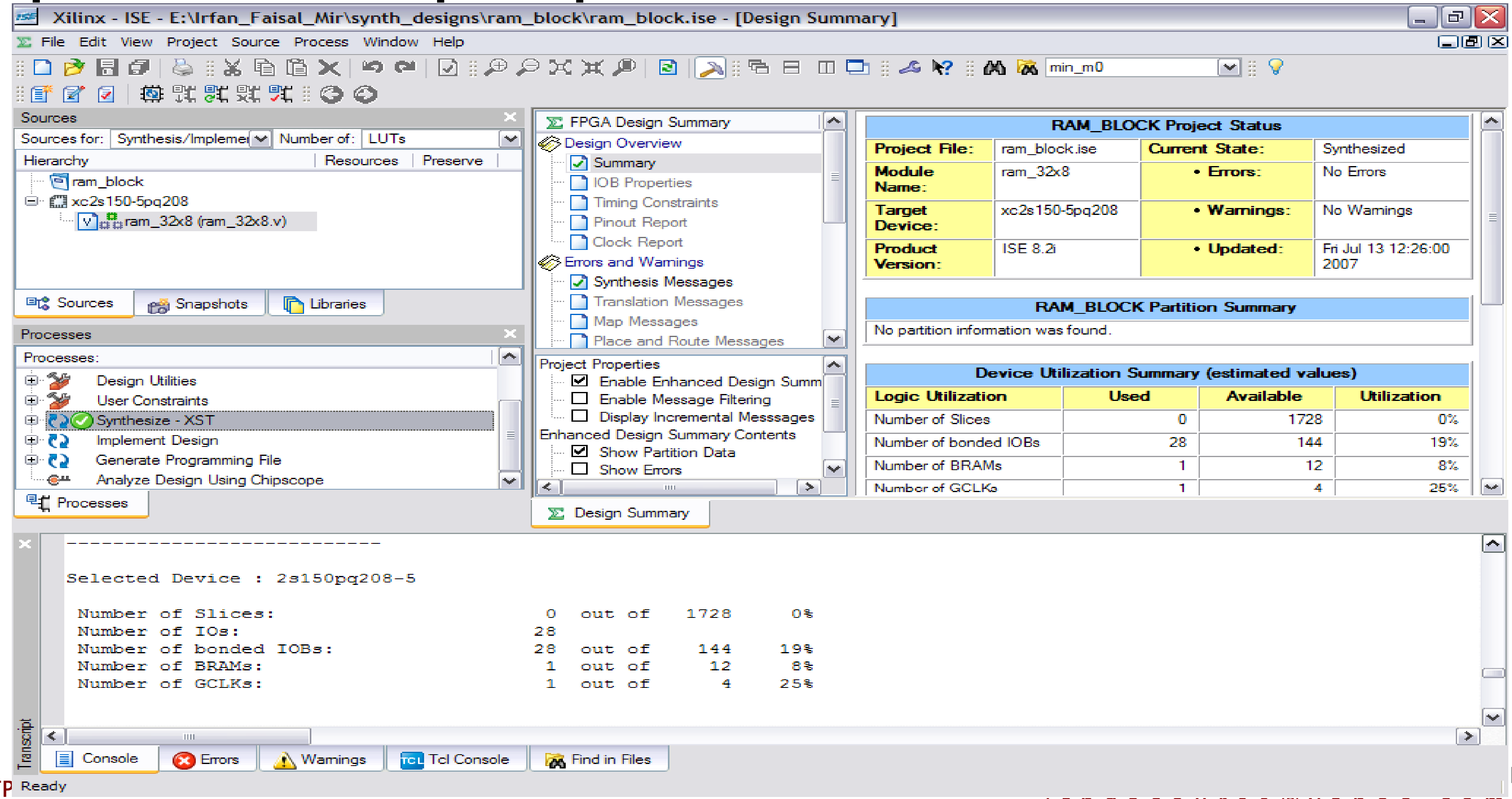
```
Started : "Launching Design
```

RAM_BLOCK Project Status			
Project File:	ram_block.ise	Current State:	New
Module	ram_32x8	• Errors:	

Property Name	Value
Optimization Goal	Speed
Optimization Effort	Speed / Area
Use Synthesis Constraints File	
Synthesis Constraints File	
Global Optimization Goal	AllClockNets
Generate RTL Schematic	Yes
Write Timing Constraints	☐
Verilog 2001	✓

Xilinx Synthesis Technology (XST) – contd.

4. When synthesis is complete, view the results by double-clicking View Synthesis Report. Following is a portion of a sample report.



Project Properties
- [✓] Enable Enhanced Design Summ
- [ ] Enable Message Filtering
- [ ] Display Incremental Messsages

Enhanced Design Summary Contents
- [✓] Show Partition Data
- [ ] Show Errors

RAM_BLOCK Project Status			
Project File:	ram_block.ise	**Current State:**	Synthesized
Module Name:	ram_32x8	**• Errors:**	No Errors
Target Device:	xc2s150-5pq208	**• Warnings:**	No Warnings
Product Version:	ISE 8.2i	**• Updated:**	Fri Jul 13 12:26:00 2007

RAM_BLOCK Partition Summary
No partition information was found.

Logic Utilization	Used	Available	Utilization
Number of Slices	0	1728	0%
Number of bonded IOBs	28	144	19%
Number of BRAMs	1	12	8%
Number of GCLKs	1	4	25%

```
---------------------------

Selected Device : 2s150pq208-5

 Number of Slices:                   0  out of   1728     0%
 Number of IOs:                     28
 Number of bonded IOBs:             28  out of    144    19%
 Number of BRAMs:                    1  out of     12     8%
 Number of GCLKs:                    1  out of      4    25%
```

Xilinx Synthesis Technology (XST) – contd.

- **HDL Coding Techniques for XST:**

Designs are usually made up of combinatorial logic and macros (for example, flip-flops, adders, subtractors, counters, FSMs, RAMs). The macros greatly improve performance of the synthesized designs. Therefore, it is important to use some coding techniques to model the macros so that they are optimally processed by XST.

NOTE: For detail please see the XST User Guide

Xilinx Synthesis Technology (XST) – contd.

➢ HDL Coding Techniques for XST:

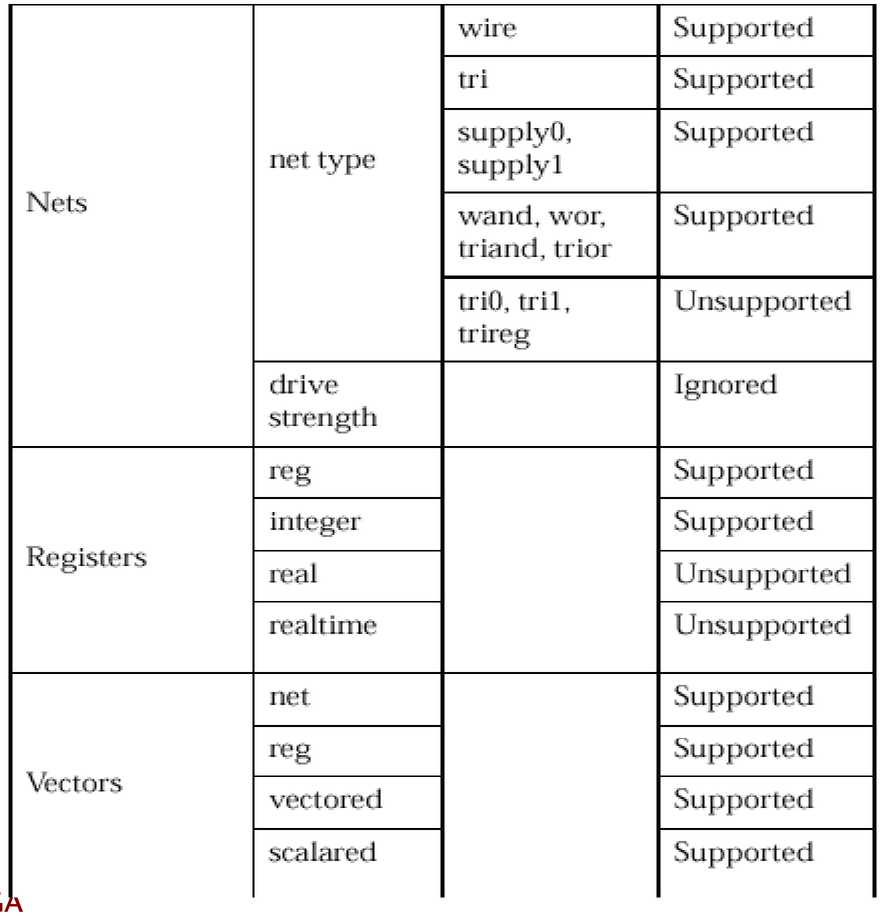
During its run, XST first tries to recognize (infer) as many macros as possible. Then all of these macros are passed to the Low Level Optimization step, either preserved as separate blocks or merged with surrounded logic in order to get better optimization results. This filtering depends on the type and size of a macro (for example, by default, 2-to-1 multiplexers are not preserved by the optimization engine). You have full control of the processing of inferred macros through synthesis constraints.

Xilinx Synthesis Technology (XST) – contd.

➢ Language Support Tables

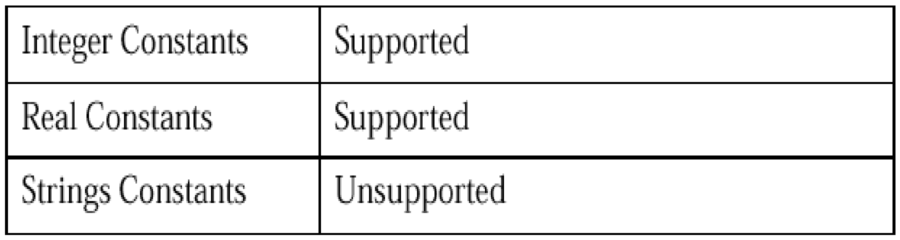
The following tables indicate which Verilog constructs are supported in XST.

Table **Data Types**

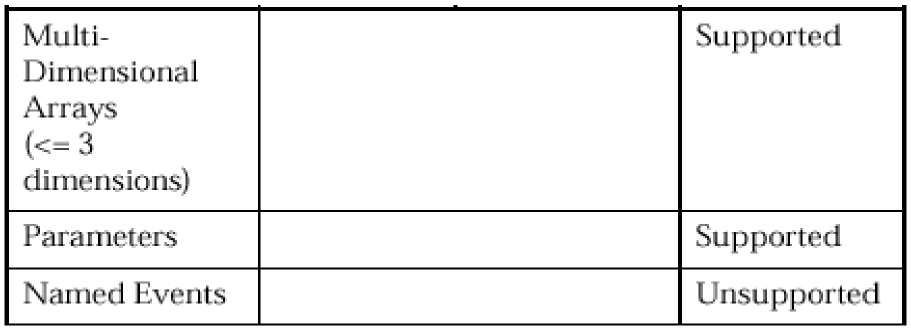


Nets	net type	wire	Supported
		tri	Supported
		supply0, supply1	Supported
		wand, wor, triand, trior	Supported
		tri0, tri1, trireg	Unsupported
	drive strength		Ignored
Registers	reg		Supported
	integer		Supported
	real		Unsupported
	realtime		Unsupported
Vectors	net		Supported
	reg		Supported
	vectored		Supported
	scalared		Supported

Table **Constants**

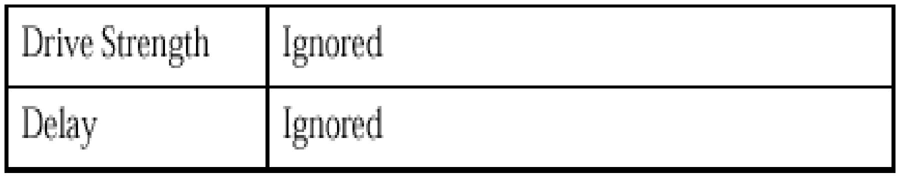


Integer Constants	Supported
Real Constants	Supported
Strings Constants	Unsupported



Multi-Dimensional Arrays (<= 3 dimensions)		Supported
Parameters		Supported
Named Events		Unsupported

Table **Continuous Assignments**



Drive Strength	Ignored
Delay	Ignored

Xilinx Synthesis Technology (XST) – contd.

➢ Language Support Tables

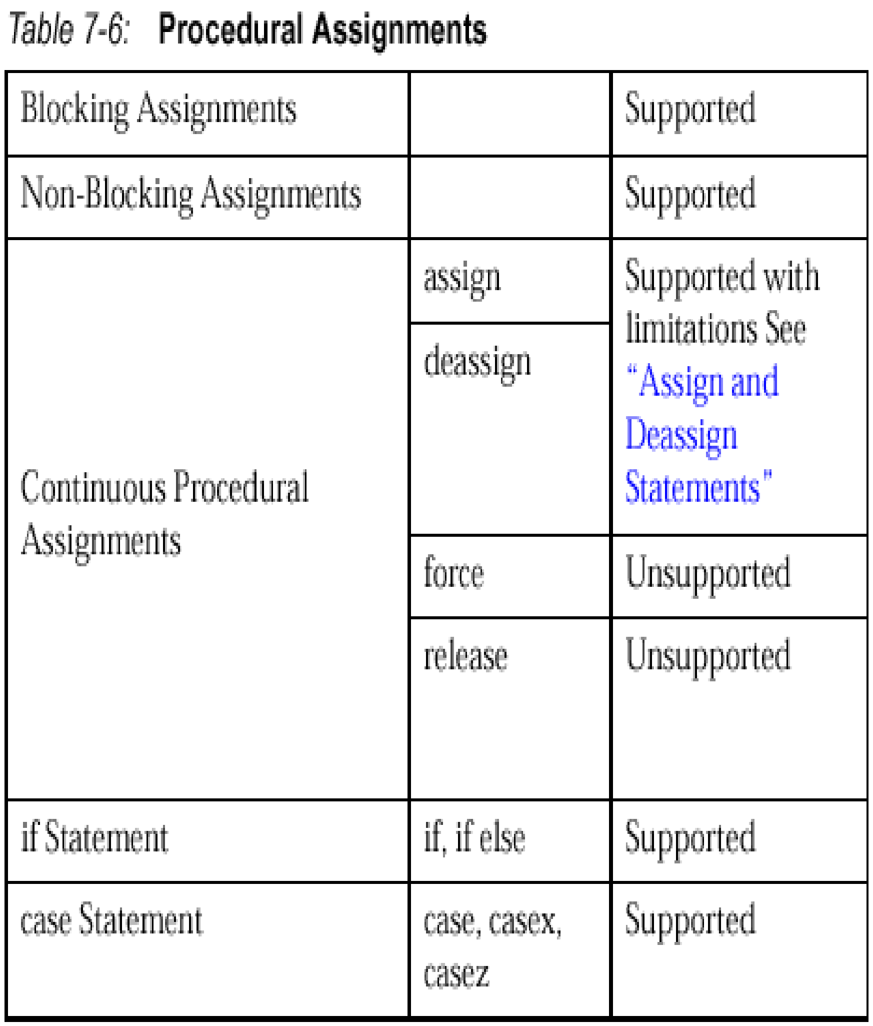
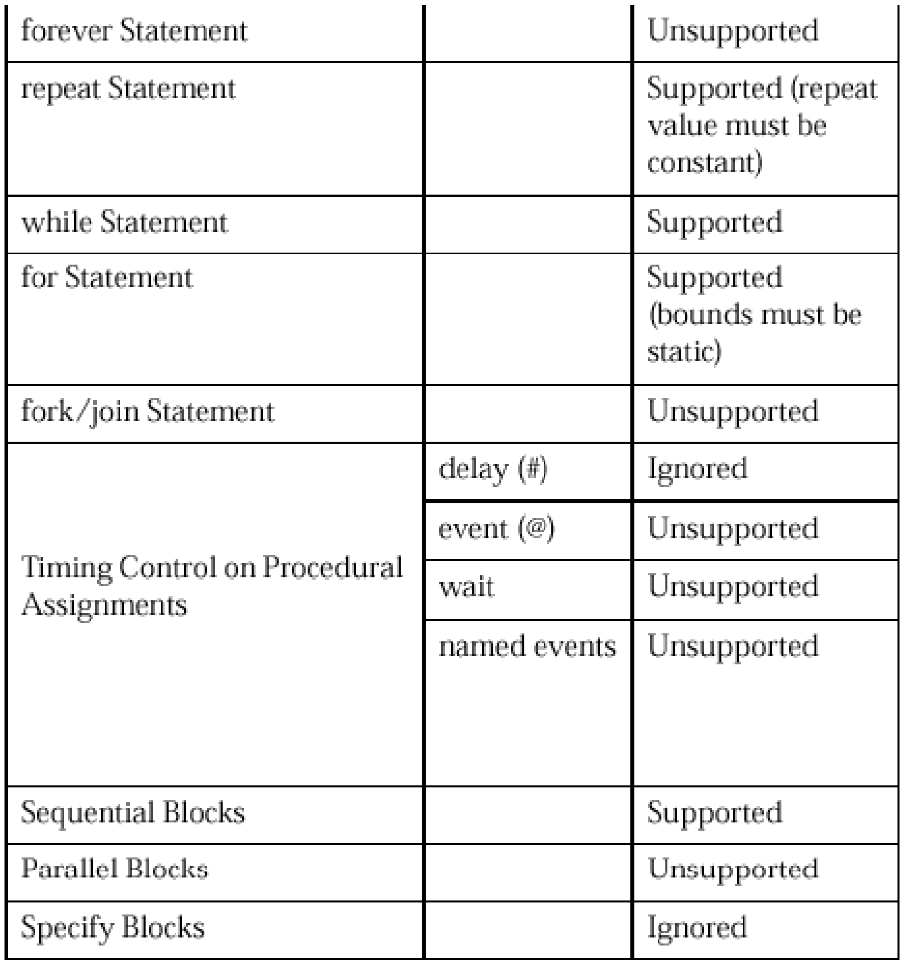



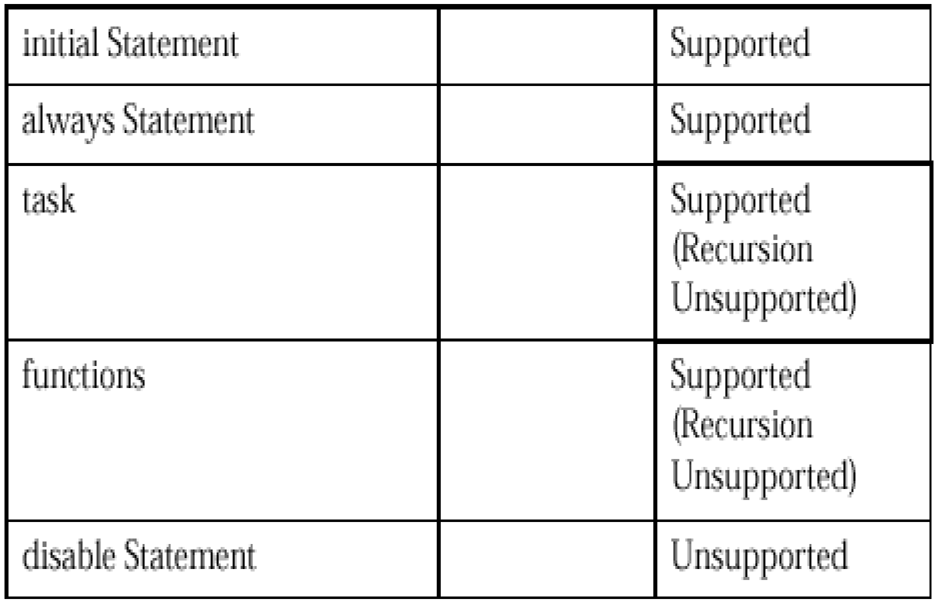
Table 7-6: **Procedural Assignments**

Blocking Assignments		Supported
Non-Blocking Assignments		Supported
Continuous Procedural Assignments	assign	Supported with limitations See "Assign and Deassign Statements"
	deassign	
	force	Unsupported
	release	Unsupported
if Statement	if, if else	Supported
case Statement	case, casex, casez	Supported
forever Statement		Unsupported
repeat Statement		Supported (repeat value must be constant)
while Statement		Supported
for Statement		Supported (bounds must be static)
fork/join Statement		Unsupported
Timing Control on Procedural Assignments	delay (#)	Ignored
	event (@)	Unsupported
	wait	Unsupported
	named events	Unsupported
Sequential Blocks		Supported
Parallel Blocks		Unsupported
Specify Blocks		Ignored

Xilinx Synthesis Technology (XST) – contd.

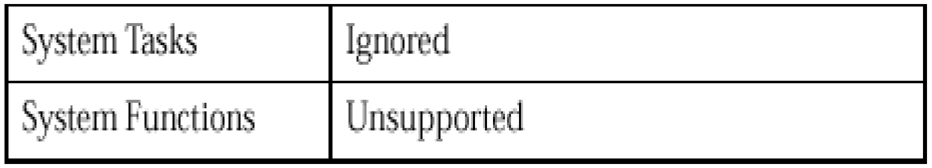
➢ Language Support Tables

Table 7-6: **Procedural Assignments**



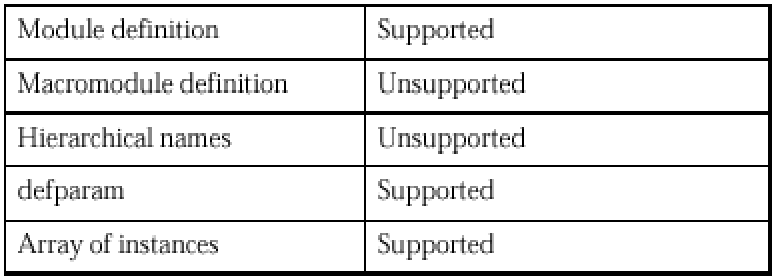
initial Statement		Supported
always Statement		Supported
task		Supported (Recursion Unsupported)
functions		Supported (Recursion Unsupported)
disable Statement		Unsupported

Table 7-7: **System Tasks and Functions**



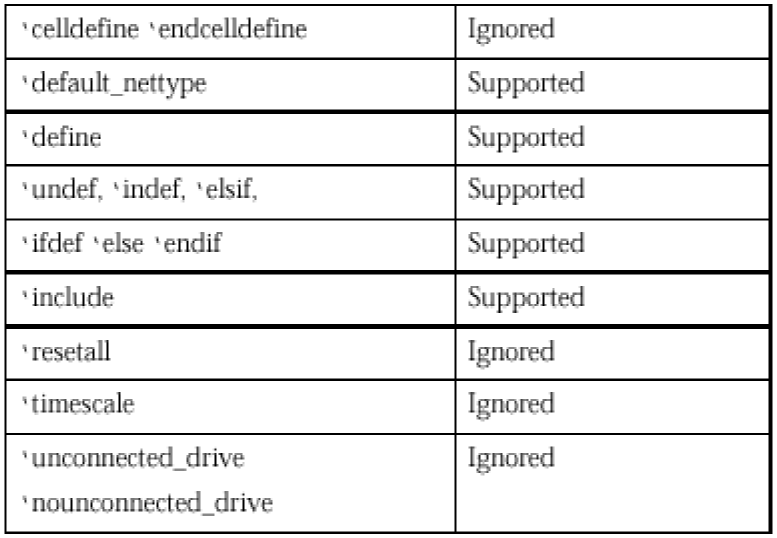
System Tasks	Ignored
System Functions	Unsupported

Table 7-8: **Design Hierarchy**



Module definition	Supported
Macromodule definition	Unsupported
Hierarchical names	Unsupported
defparam	Supported
Array of instances	Supported

Table 7-9: **Compiler Directives**

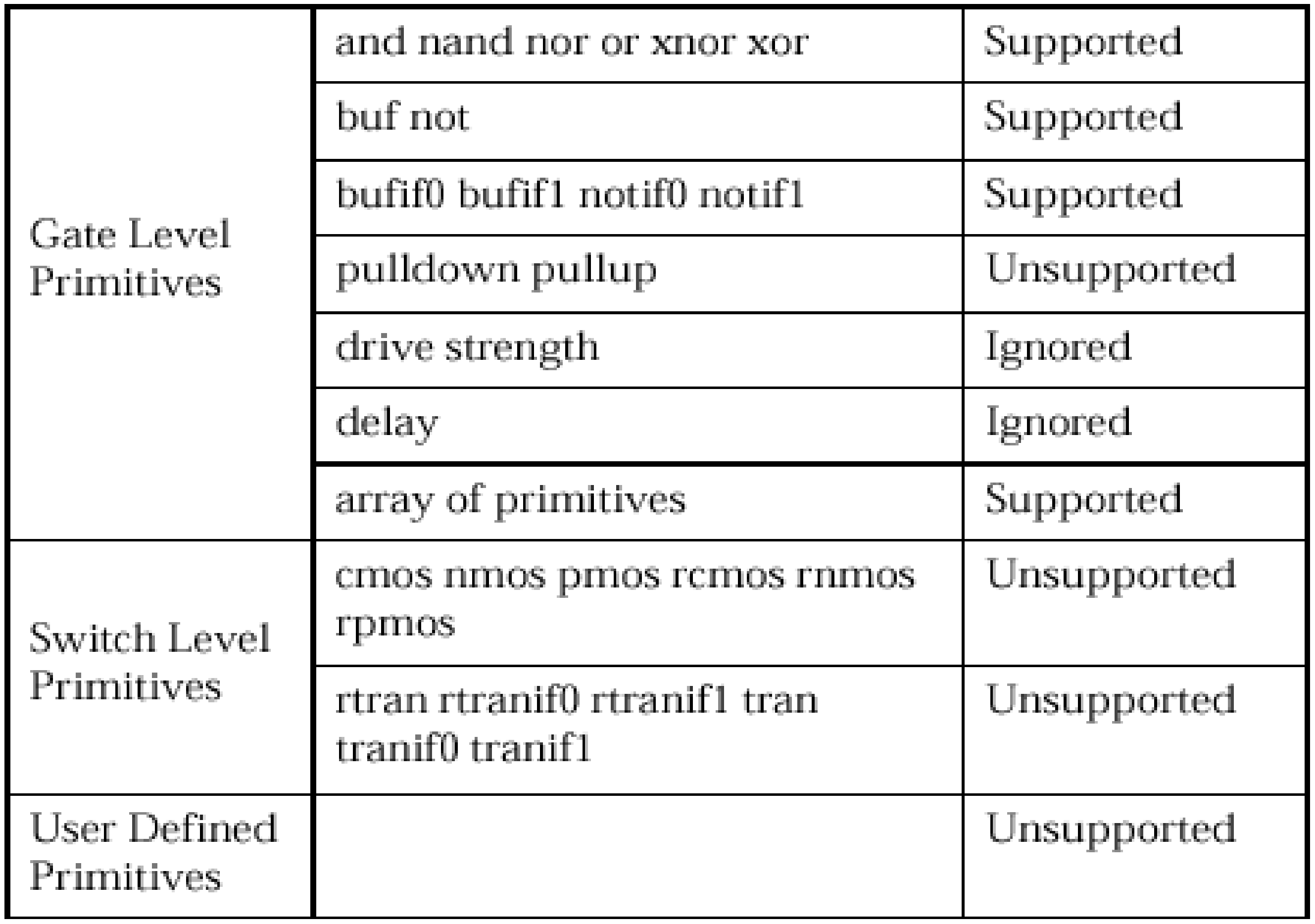


`celldefine `endcelldefine	Ignored
`default_nettype	Supported
`define	Supported
`undef, `indef, `elsif,	Supported
`ifdef `else `endif	Supported
`include	Supported
`resetall	Ignored
`timescale	Ignored
`unconnected_drive `nounconnected_drive	Ignored

Xilinx Synthesis Technology (XST) – contd.

➢ **Language Support Tables**

Table 7-10: **Primitives**



Gate Level Primitives	and nand nor or xnor xor	Supported
	buf not	Supported
	bufif0 bufif1 notif0 notif1	Supported
	pulldown pullup	Unsupported
	drive strength	Ignored
	delay	Ignored
	array of primitives	Supported
Switch Level Primitives	cmos nmos pmos rcmos rnmos rpmos	Unsupported
	rtran rtranif0 rtranif1 tran tranif0 tranif1	Unsupported
User Defined Primitives		Unsupported

Xilinx Synthesis Technology (XST) – contd.

➢ Verilog Reserved Keywords

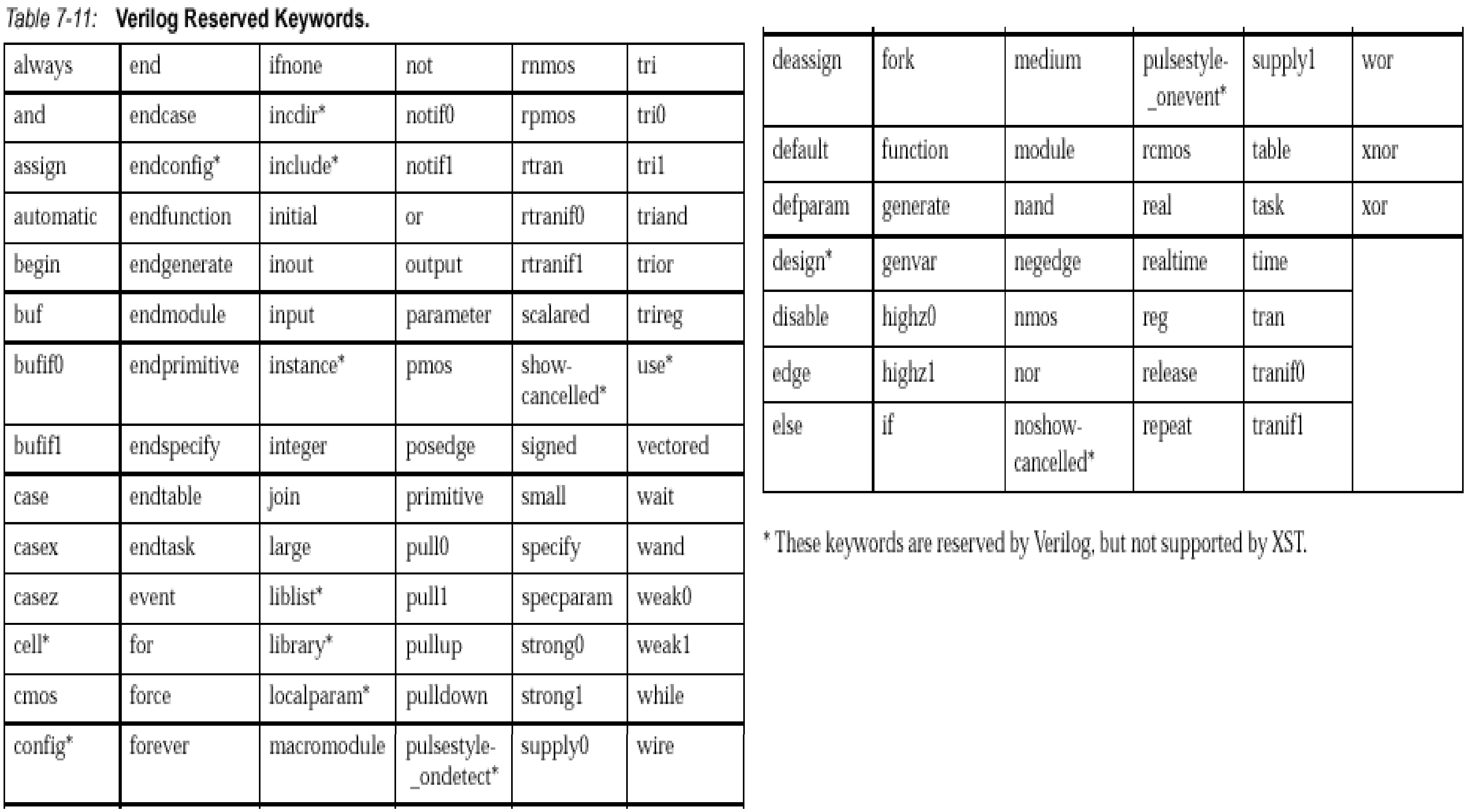


Table 7-11: **Verilog Reserved Keywords.**

always	end	ifnone	not	rnmos	tri
and	endcase	incdir*	notif0	rpmos	tri0
assign	endconfig*	include*	notif1	rtran	tri1
automatic	endfunction	initial	or	rtranif0	triand
begin	endgenerate	inout	output	rtranif1	trior
buf	endmodule	input	parameter	scalared	trireg
bufif0	endprimitive	instance*	pmos	show-cancelled*	use*
bufif1	endspecify	integer	posedge	signed	vectored
case	endtable	join	primitive	small	wait
casex	endtask	large	pull0	specify	wand
casez	event	liblist*	pull1	specparam	weak0
cell*	for	library*	pullup	strong0	weak1
cmos	force	localparam*	pulldown	strong1	while
config*	forever	macromodule	pulsestyle-_ondetect*	supply0	wire

deassign	fork	medium	pulsestyle-_onevent*	supply1	wor
default	function	module	rcmos	table	xnor
defparam	generate	nand	real	task	xor
design*	genvar	negedge	realtime	time	
disable	highz0	nmos	reg	tran	
edge	highz1	nor	release	tranif0	
else	if	noshow-cancelled*	repeat	tranif1	

* These keywords are reserved by Verilog, but not supported by XST.