Distributed Memory Generator v6.1

DS322 March 1, 2011 **Product Specification**

Introduction

The Xilinx LogiCORE™ IP Distributed Memory Generator core uses Xilinx Synthesis Technology (XST) to create a variety of distributed memories.

Features

- Generates read-only memories (ROMs), single, simple dual, and dual-port random access memories (RAMs), and SRL16-based memories
- Supports data depths ranging from 16–65,536 words
- Supports data widths ranging from 1–1024 bits
- Optional registered inputs and outputs
- Optional pipelining when output is registered

LogiCORE IP Facts					
Core Specifics					
Supported Device Family(1)	Kintex-7, Virtex-7, Virtex-6, Virtex-5, Virtex-4, Spartan-6, Spartan-3/XA, Spartan-3E/XA, Spartan-3A/3AN/3A DSP				
Resources Used	I/O	LUTs	FFs	Block RAMs	
	Varies based on core configuration			None	
CLB/REG	Varies based on core configuration				
Provided with Core					
Documentation	Product Specification				
Design File Formats	NGC Netlist				
Behavioral Models	VHDL, Verilog				
Design Tool Requirements					
Xilinx Implementation Tools	ISE 13.1				
Verification	Mentor Graphics ModelSim 6.6d				
Simulation	Mentor Graphics ModelSim 6.6d				
Synthesis	XST				
Support					
Provided by Xilinx, Inc.					

1. For the complete list of supported devices, see Table 1, page 2 and the release notes for this core.

Overview

The Distributed Memory Generator core is used to create memory structures using LUT distributed RAM resources. The core can create the following memory types:

- Distributed ROM, page 3
- Distributed Single-Port RAM, page 5
- Distributed Dual-Port RAM, page 6
- Distributed SRL16-Based RAM, page 8 (excluding Kintex™-7, Virtex®-7, Virtex-6, Virtex-5 and Spartan®-6 devices)
- Distributed Simple Dual-Port RAM, page 9

For detailed information about each memory type, see the respective memory type in the Functional Description, page 3.

Options are available for simple registering of inputs and outputs. Optional asynchronous and synchronous resets are available for the output registers. For timing information, see the Xilinx Product Specification for the specific target architecture.

Applications

Applications for Distributed Memory are diverse and numerous. Examples include

- Using the distributed ROM as a very large look-up table
- Using the single-port RAM as scratch pad memory for the embedded PowerPC™ microprocessors used within the Kintex-7, Virtex-7, Virtex-6, Virtex-5, and Virtex-4 families, or for the MicroBlaze™ or PicoBlaze™ processors
- Using the simple dual and dual-port RAM within an asynchronous FIFO
- Using the SRL16-based RAM for pattern generation

A large number of Xilinx IP cores rely on the Distributed Memory Generator internally to implement memory structures.

Supported Devices

Table 1 shows the families and sub-families supported by the Distributed Memory Generator.

Table 1: **Supported FPGA Families and Sub-Families**

FPGA Family	Sub-Family
Spartan-3	
Spartan-3E	
Spartan-3A	
Spartan-3AN	
Spartan-3A DSP	
Spartan-6	LX/LXT
Virtex-4	LX/FX/SX
Virtex-5	LXT/FXT/SXT/TXT
Virtex-6	CXT/HXT/LXT/SXT
Virtex-7	
Kintex-7	

Feature Summary

Depth. In all supported FPGA families, the depth can range from 16–65536 words in multiples of 16.

Width. The width of each word can be anywhere in the range of 1–1024 bits.

Optional Input Registering. The following inputs to the memory can be registered or non-registered. When input registering is used, these inputs can be clock-enabled.

- Write Address
- Data
- Write Enable
- Output Register Clock Enable
- Dual-port RAM Read Address
- Simple Dual-port RAM Read Address

Optional Output Registering and Pipelining. The memory can be non-registered, registered, or both. The output registers can have a variety of controls, including

- Asynchronous Reset
- Synchronous Reset
- Clock Enable

In addition, the Clock Enable and Synchronous Reset can be configured so the Synchronous Reset overrides the Clock Enable, and vice versa. In single-port, simple dual-port and dual-port distributed RAM cores with registered outputs, an additional pipeline register may be added to the output path to improve the operating speed at the expense of an additional cycle of latency.

Functional Description

The following sections provide a functional description and illustration of each of the four memory types.

Distributed ROM

The Distributed Memory Generator uses LUT-based distributed ROM resources to create 16-bit deep, 1-bit wide ROMs, and generates a fabric-based bus multiplexer to create a deeper and wider ROM. The content of this memory is defined by supplying an input coefficient (COE) file to the CORE Generator™ software when the memory is generated, after which the content is fixed.

To create the distributed ROM, the core parses the initialization data provided by the user-supplied COE file. From that data, any necessary logic or optional registering is inferred and created. Figure 1 shows the distributed ROM schematic symbol, and Figure 2 illustrates one of the possible

implementations of a distributed ROM core. For distributed ROM timing information, see the user guide for the specific FPGA family-related architecture.

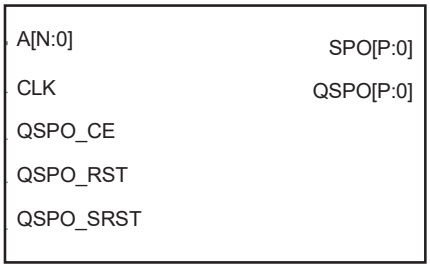

Figure 1: **Distributed ROM Schematic Symbol**

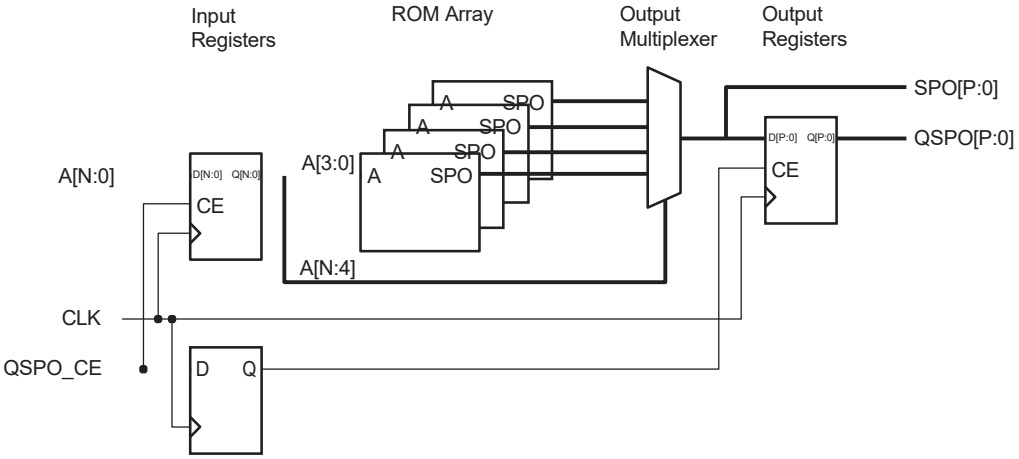

Figure 2: **Distributed ROM Module Schematic**

Distributed Single-Port RAM

The distributed single-port RAM uses the single-port distributed RAM resource of the LUT. Writes to the Single-Port RAM are synchronous to the clock (CLK). However, read operations can be asynchronous (SPO) or synchronous to the clock (QSPO). Figure 3 illustrates the distributed single-port RAM schematic symbol, and Figure 4 illustrates its internal implementation. If a pipeline register is added to a registered core (not illustrated), the additional register is re-timed into the SPO MUX array. For timing information about the distributed single-port RAM, see the appropriate user guide for the target FPGA architecture.

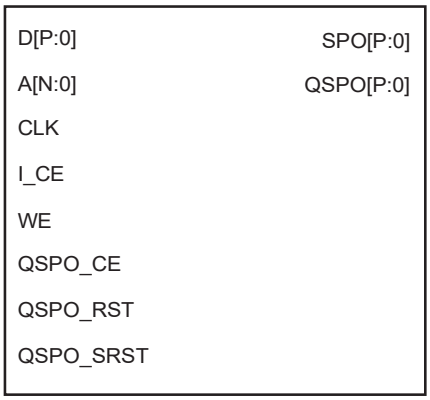

Figure 3: **Distributed Single-Port RAM Schematic Symbol**

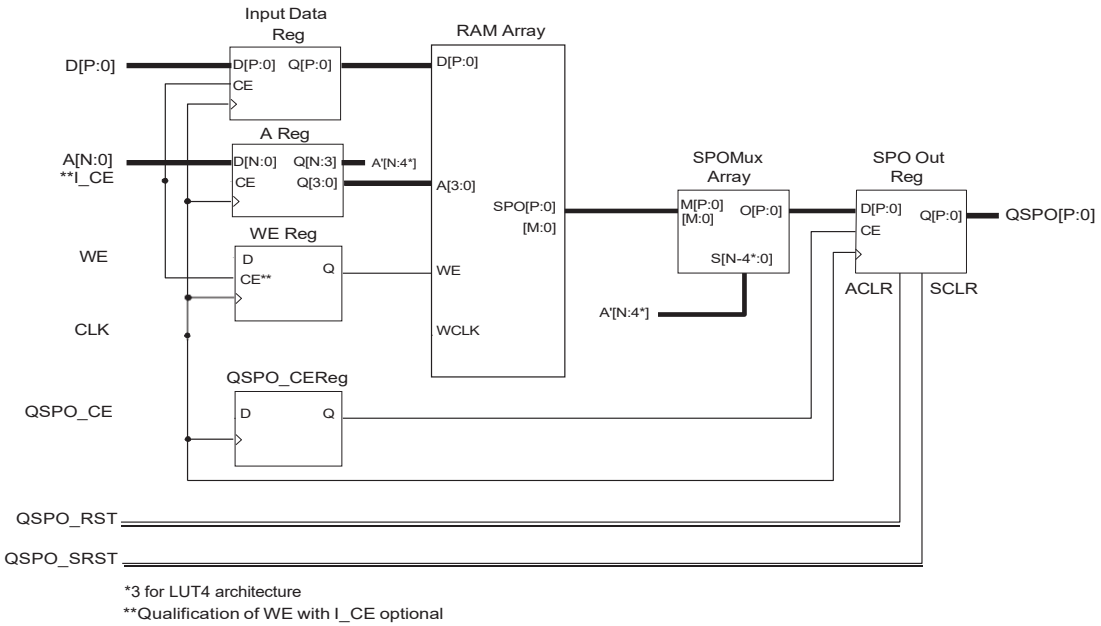

Figure 4: **Distributed Single-Port RAM Module Schematic**