Distributed Memory Generator v4.1

DS322 December 17, 2008 **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- 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	Virtex®-5, Virtex-4, Spartan®-3/XA, Spartan-3E, Spartan-3A/3AN/3A DSP For additional Xilinx devices, see "New Architecture."
Core Resources	
CLB/REG	Varied, based on core parameters
DCM	None
BUFG	None
IOB/Rocket IO™ Transceivers	None
PPC	None
IOB-FF/TBUFs	None
Provided with Core	
Documentation	Product Specification
Design File Formats	EDIF or NGC netlist
Design Tool Requirements	
Xilinx Implementation Tools	ISE® v11.1
Supported Behavioral Models	VHDL, Verilog
Synthesis	XST
Support	
Provided by Xilinx, Inc.	

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"
- "Distributed Single-Port RAM"
- "Distributed Dual-Port RAM"
- "Distributed SRL16-Based RAM" (excluding Virtex-5)

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 Virtex-4, and Virtex-5 families, or for the MicroBlaze™ or PicoBlaze™ processors
- Using the 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.

Feature Summary

Depth. In the Virtex-4, Virtex-5, Spartan-3, Spartan-3E, and Spartan-3A/3AN/3A DSP 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, it can be clock-enabled.

- Write Address
- Data
- Write Enable
- Output Register Clock Enable
- 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 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

This 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 platform-related architecture.

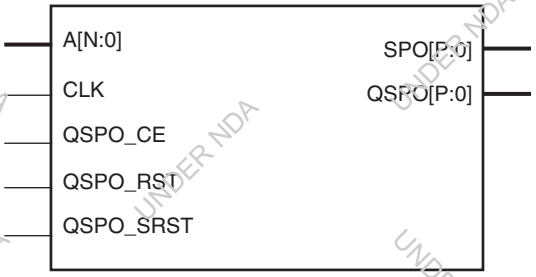

Figure 1: **Distributed ROM Schematic Symbol**

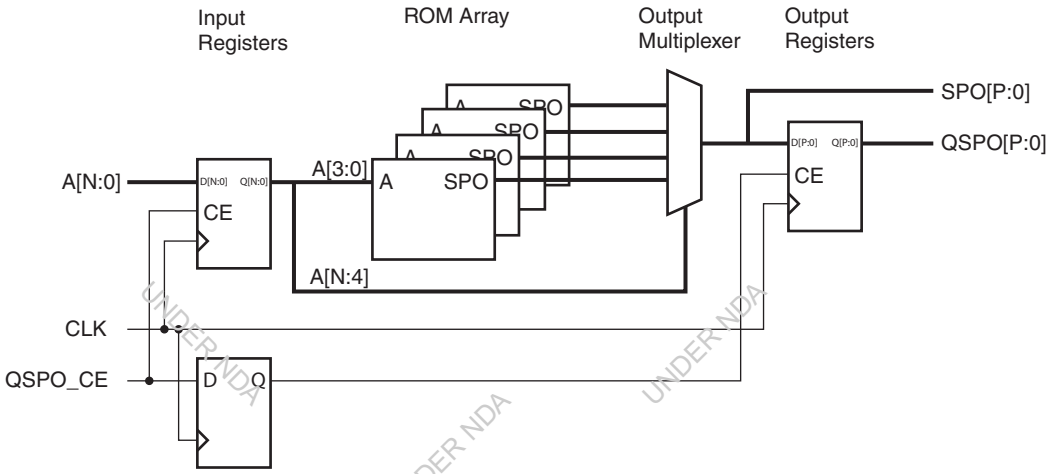

Figure 2: **Distributed ROM Module Schematic**

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 platform-specific user guide for the target FPGA architecture.

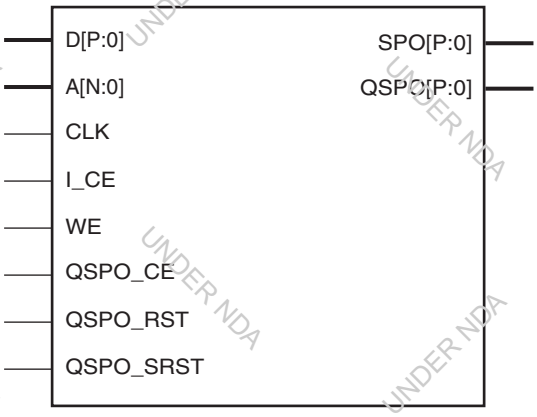

Figure 3: **Distributed Single-Port RAM Schematic Symbol**

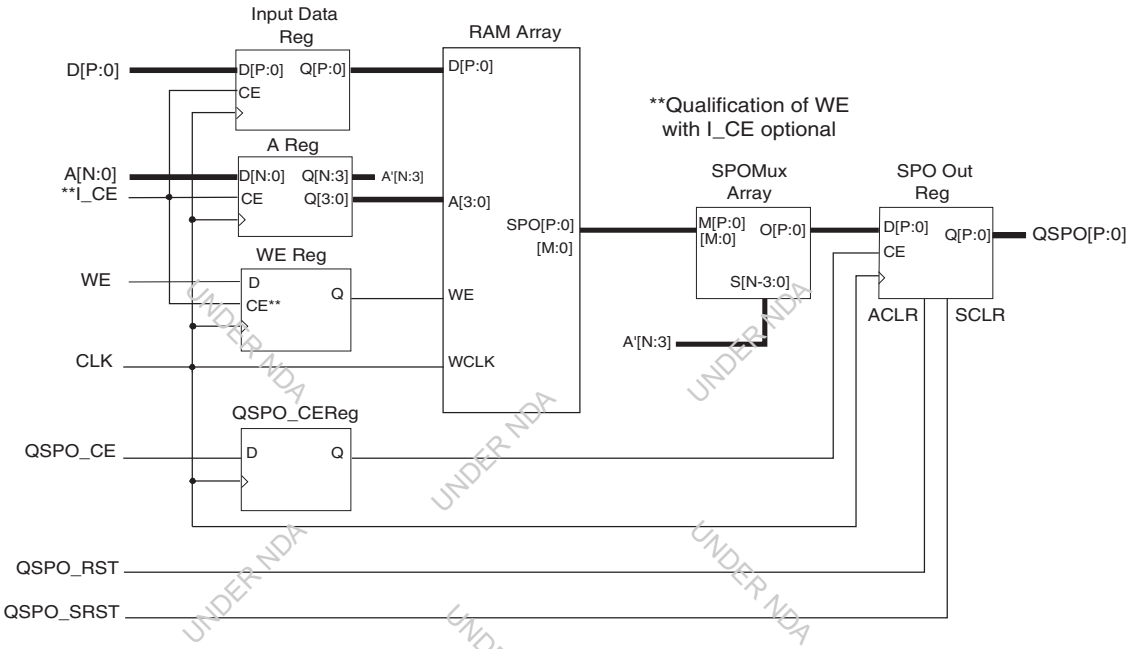

Figure 4: **Distributed Single-Port RAM Module Schematic**

Distributed Dual-Port RAM

Writes to the distributed dual-port RAM are synchronous to the clock (CLK). However, read operations from the distributed dual-port RAM can be asynchronous or synchronous with respect to either of the two clocks (CLK or QDPO_CLK). Figure 5 illustrates the dual-port RAM schematic symbol, showing the relevant ports. Figure 6 shows the internal implementation of the distributed dual-port RAM. If a pipeline register is added to a registered core (not illustrated), the additional register is re-timed into the SPO MUX array and DPO MUX array.

In realizing the most simple dual-port RAM, two LUTs sharing a common write logic are used. When RAM is written *to*, both LUTs continue to share common content, but have different address buses for reading. In this way, different content can be addressed and read from the SPO and DPO outputs, and use different clock domains where registered output is used.

The width of the data buses must be identical on input and output. If width conversion is required from a memory instance, consider using the Block Memory Generator core.

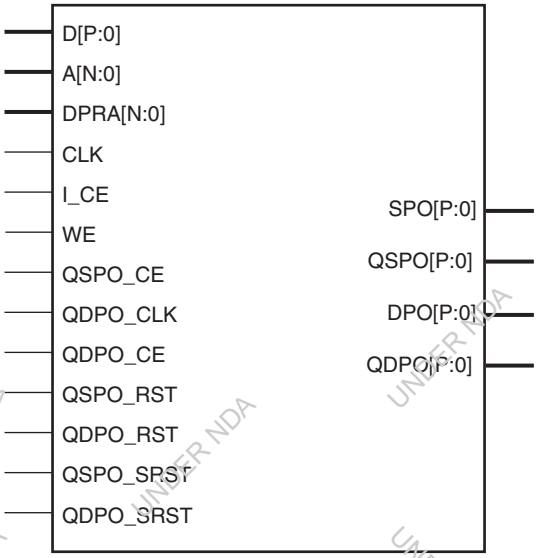

Figure 5: **Dual-Port RAM Schematic Symbol**

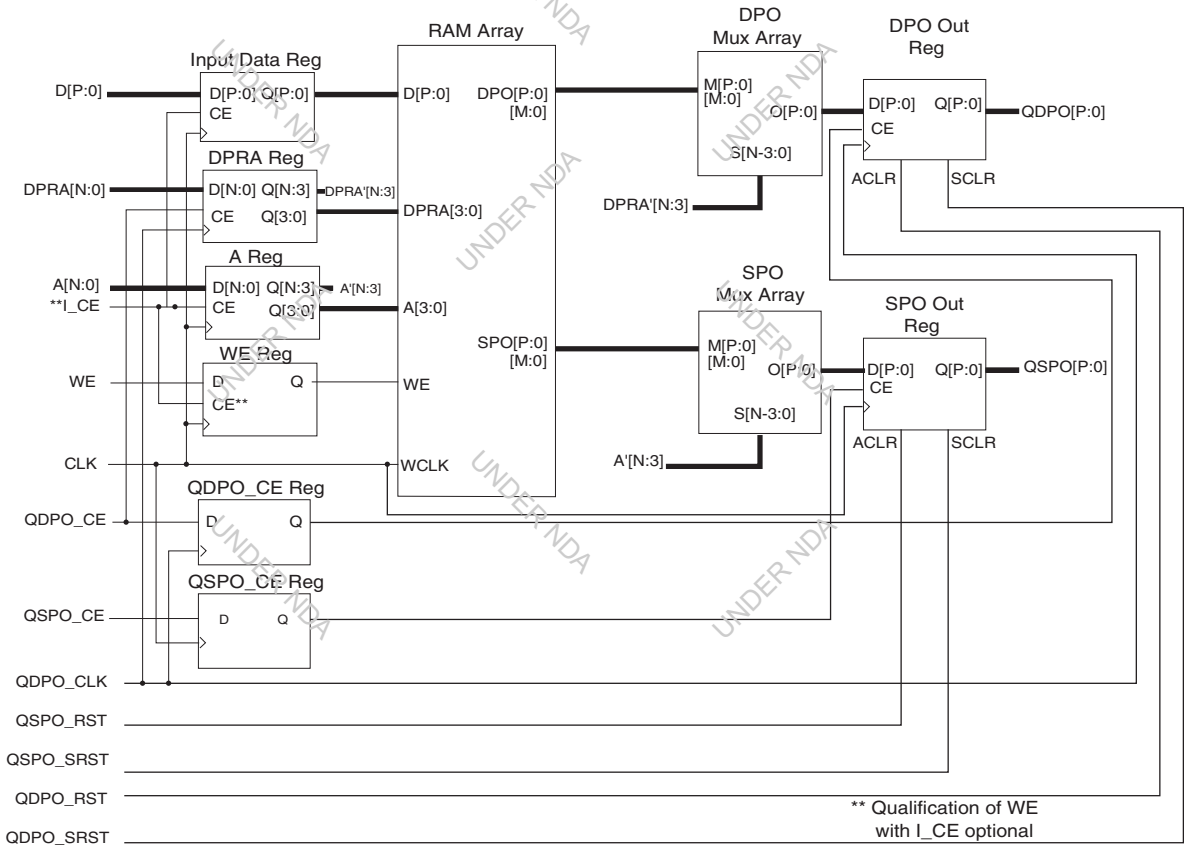

Figure 6: **Dual-Port RAM Module Schematic**

Distributed SRL16-Based RAM

Note: The Virtex-5 device family does not support SRL16-based memories.

Figure 7 shows the schematic symbol of the distributed SRL16-based RAM. Note that Address Bus A, used to write data to a specific address in the SRL16 array, does not contain the full range of bits normally required for a memory of a given depth. Address Bus A is missing the four *least* significant bits because the four address bits on each SRL16 are driven by the four least significant bits from the read address bus SPRA. (The address pins on the SRL16 determine where in the SRL16 the data is read *from*, not where data is written *to*.)

Where written data is placed in the address space is determined by the specific SRL16 written to, and where previous data has gone, because data is sequentially written into the SRL16. This means that the four least-significant bits in address bus A are not required.

Because 16-bit deep shift registers are being used in place of the LUT4 primitives, the SRL16-based RAM is not a genuine RAM. Specifically, random access is given to addresses the client of the memory wants to read *from* to access the SRL16 holding the data of interest. To write data to a specific location in memory, the application using the memory has to enable a specific SRL16 or set of SRL16s (when the width of the RAM is greater than 1 bit), and then assert the Write Enable for the RAM. The Write Enable actually drives the Clock Enable on the SRL16, allowing data to be shifted into the SRL16s.

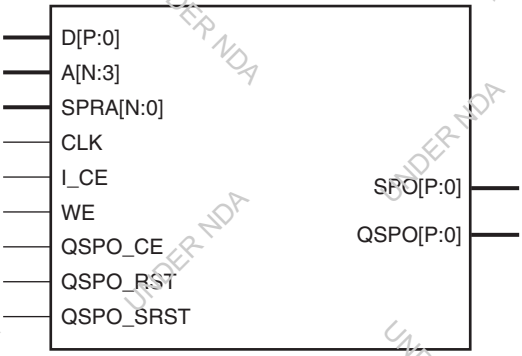

Figure 7: **SRL16-Based RAM Schematic Symbol**

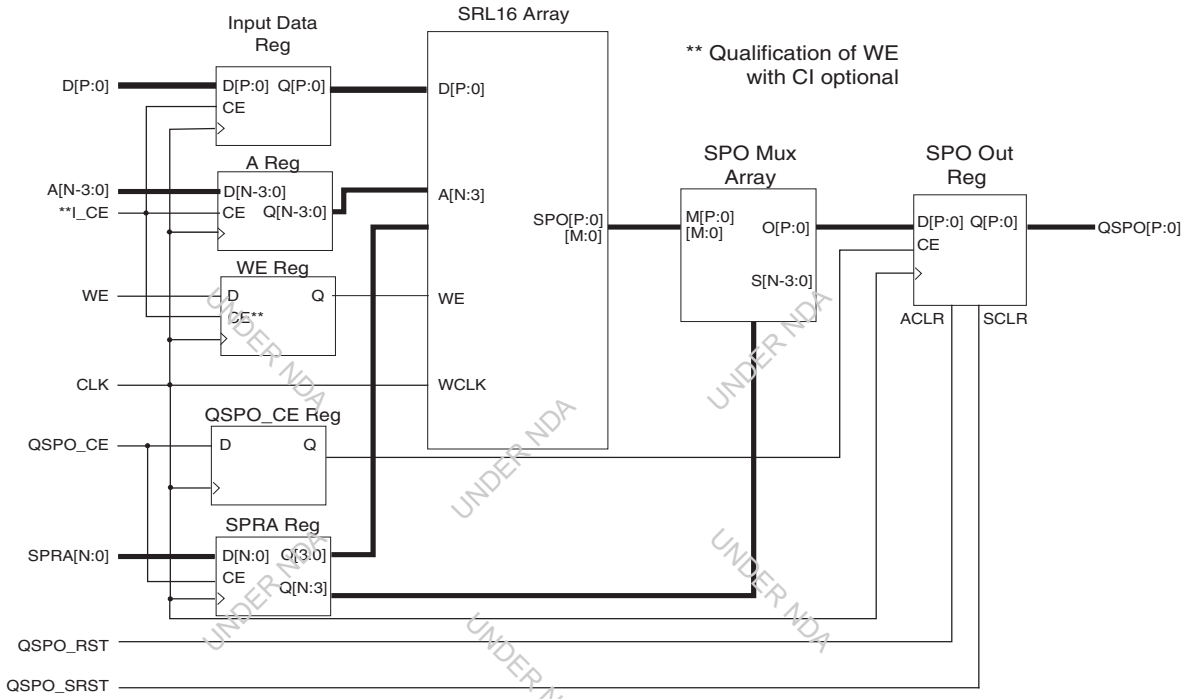

Figure 8: **SRL16-Based RAM Module Schematic**

Waveforms

Detailed timing diagrams for Distributed Memory cores vary depending on the target architecture. Examples can be found in the user guide for the target architecture. See the Xilinx Support Library for available user guides.

Signal List

Table 1 defines the Distributed Memory core signals.

Table 1: **Core Signal Pinout**

Name	Direction	Description
D[P:0]	Input	Data input to be written into the memory for single-port, dual-port and SRL16-based RAMs.
A[N:0]	Input	Address inputs. Only address input for ROMs and single-port RAMs. On SRL16-based RAMs, it defines the most significant bits (4 and up) of the memory locations written to. On dual- port memories, it defines memory location written to and memory location read out on the SPO[P:0] outputs.
SPRA[N:0]	Input	Single-Port Read Address. Present only on SRL16-based RAMs and defines memory location read out on the SPO[P:0] outputs.
DPRA[N:0]	Input	Dual-Port Read Address. Present only on dual-port RAMs and defines memory location read out on the DPO[P:0] outputs.
SPO[P:0]	Output	Non-registered single-port output bus. Non-registered data output bus for ROMs, single-port, and SRL16-based RAMs. One of two non-registered output buses on dual-port RAMs.

Table 1: **Core Signal Pinout *(Cont'd)***

Name	Direction	Description
QSPO[P:0]	Output	Registered single-port output bus. Registered data output bus for ROMs, single-port, and SRL16-based RAMs. One of two registered output buses on dual-port RAMs.
DPO[P:0]	Output	Non-registered dual-port output bus. One of the non-registered data output buses for dual-port RAMs. Data stored at the address location specified by DPRA[N:0] appears at this port.
QDPO[P:0]	Output	Registered dual-port output bus. One of two registered output buses on dual-port RAMs.
CLK	Input	Write clock and register clock for ROMs, single-port, and SRL16-based RAMs. On dual-port RAMs signal is the write clock and register clock for single-port input and output registers.
QDPO_CLK	Input	On dual-port RAMs, signal is the write clock and register clock for dual-port RAM input and output registers
WE	Input	Write Enable
I_CE	Input	Input Clock Enable. Signal is present for RAMs which have registered inputs. The clock enable controls input data register, address register and WE register.
QSPO_CE	Input	On ROMs, clock enable controls all input and output registers. On dual-port memories, controls output registers in QSPO path.
QDPO_CE	Input	Present only on dual-port RAMs. Controls output registers in QDPO path.
QSPO_RST	Input	Single-port registered output asynchronous reset.
QDPO_RST	Input	Available only on dual-port RAMs. Dual-port registered output asynchronous reset.
QSPO_SRST	Input	Single-port registered output synchronous reset
QDPO_SRST	Input	Available only on dual-port RAMs. Dual-port registered output synchronous reset

Note: All control inputs are Active High. If an Active Low input is required for a specific control pin, an inverter must be placed in the path to the pin; the inverter is absorbed appropriately during mapping.

Generating the Core

The Distributed Memory Generator can be found in two places in the CORE Generator graphical user interface (GUI) View by Function pane.

To access the Distributed Memory Generator, do one of the following:

- Choose Basic Elements > Memory Elements.
- Choose Memories & Storage Elements > RAMs & ROMs.

CORE Generator Parameter Screens

The Distributed Memory Generator GUI uses three screens.

- Main Screen (Figure 9)
- Input, Dual-Port, and Output Options Screen (Figure 10)
- Initial Content and Reset Options Screen (Figure 11)

All the screens share common tabs and buttons to provide information about the core and perform specific actions, such as generating the core and navigating among screens.

Main Screen

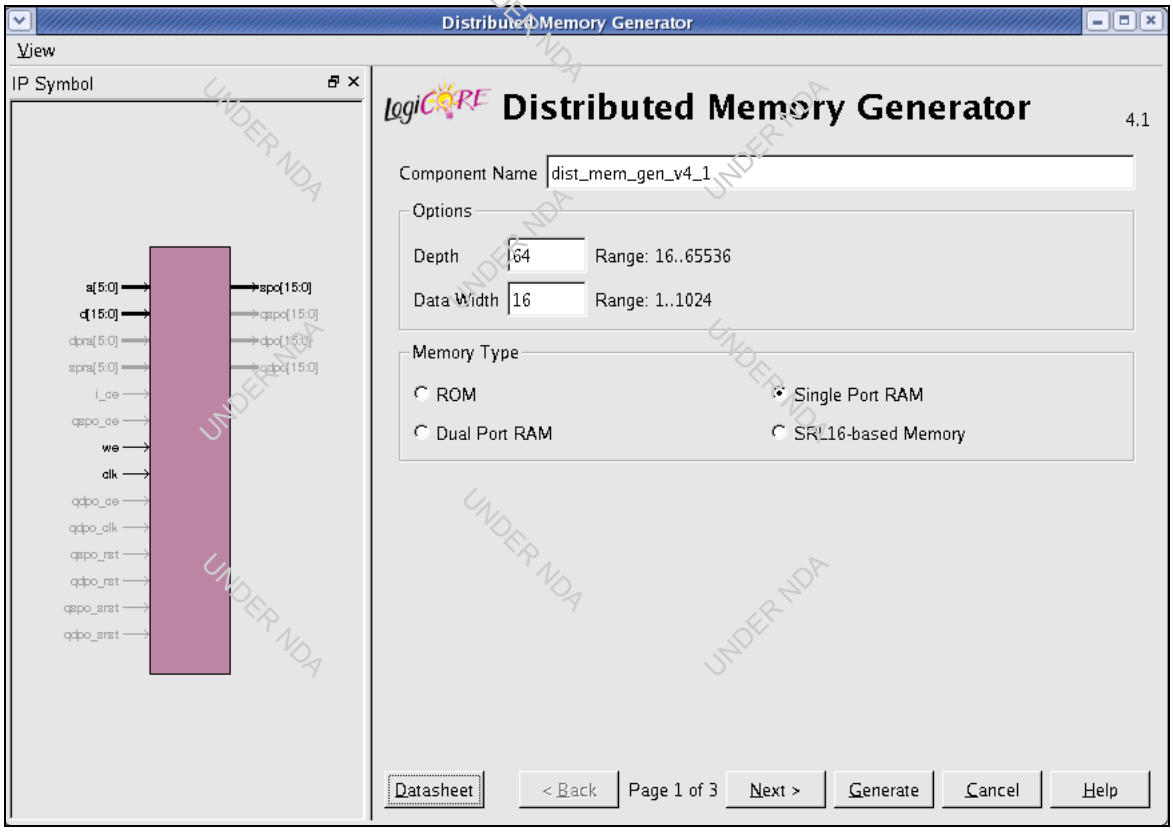

Figure 9: **Main Screen**

Component Name. The base name of the output files generated for the core. Names must begin with a letter and be composed of any of the following characters: a to z, 0 to 9 and "_."

Depth. Enter a value in the valid range from 16–65536, in steps of 16. Default value is 64. Note that the maximum depth is reduced to 4096 for all but the Virtex-4, Virtex-5, Spartan-3, Spartan-3A/3AN/3A DSP, and Spartan-3E devices.

Data Width: Enter the width of the memory in the valid range from 1–1024. The default value is 16.

Memory Type: Select one of four options. The default setting is single-port RAM.

- **ROM**. Figure 2 illustrates a schematic of the structure of the ROM modules. The address register is optional (controlled by the setting of the Input Options parameter). Output registering is also optional (controlled by the setting of the Output Options parameter). Note that the clock is not required when these registers are not present. The resets and clock enables are optional.
- **Single-Port RAM**. Figure 4 illustrates a schematic of the structure of the single-port RAM modules. The address and data registers are optional (controlled by the setting of the Input Options parameter). Output registering is also optional (controlled by the setting of the Output Options parameter). The resets and clock enables are optional. The clock CLK is always required because writes to the single-port RAM are synchronous to that clock.
- **Dual-Port RAM**. Figure 6 illustrates a schematic of the structure of the dual-port RAM modules. The address and data registers are optional (controlled by the setting of the Input Options parameter). The dual-port read address register is optional (controlled by the setting of the Dual-Port Address parameter). Output registers for both output ports are also optional (controlled by the setting of the Output Options parameter). When registered outputs are selected, the two output ports can be clocked by the same or different clock signals and can have the same or different clock enables (based on the settings selected for the Common Output Clock and Common Output CE parameters). All resets and clock enables are optional. Note that the clock CLK is always required, because writes to the RAM are synchronous to that clock.
- **SRL16-Based RAM**. Figure 8 illustrates a schematic of the structure of the SRL16-based RAM module. The address and data registers are optional (controlled by the setting of the Input Options parameter). Output registers are also optional (controlled by the setting of the Output Options parameter). The resets and clock enables are optional. When the project family is set to Virtex-5, this option will be greyed-out.

Input, Dual-Port, and Output Options Screen

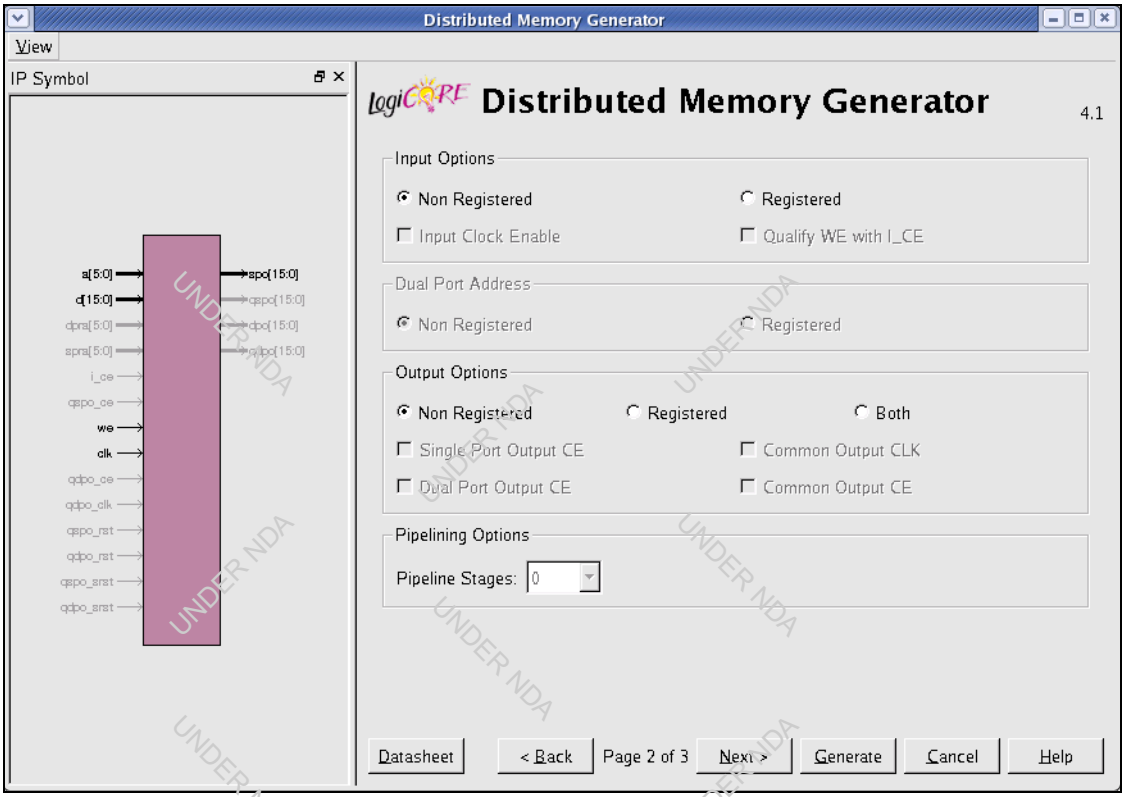

Figure 10: **Distributed Memory Options Screen**

Input Options. Select an option for the required input types. Non-registered is the default setting. Selecting Registered produces different effects depending on the selected memory type. For ROM, an address register is generated; for single-port RAM, dual-port, and SRL16-based RAM, a register on the A[N:0] address input, a data input register, and a WE register are generated.

- **Input Clock Enable.** An optional input available when Input Options are set to Registered and Memory Type *is not* a ROM**.**
- **Qualify WE with I_CE**. Valid only for single-port RAM, dual-port RAM, and SRL16-based RAM with Input Options set to Registered and Input Clock Enable selected. When deselected, the WE register has no clock-enable control. When selected, the WE register has a clock enable driven by the I_CE input.

Dual-Port Address. Valid only for dual-port RAMs, and controls the presence or absence of a register on the DPRA[N:0] inputs. Non-registered is the default setting.

Pipelining Options: When registered single-port RAMs and dual-port RAMs are selected, an optional pipeline register can be placed into the output path.

- **Pipeline Stages**. Select ’0’ for no pipeline registers and ’1’ for a single pipeline stage in the output multiplexer.

Output Options: Select an option for the required output types. Non-registered is the default setting.

- **Single-Port Output Clock Enable**. Enabled for registered output memory or for input registered ROM to provide this optional pin.

- **Dual-Port Output Clock Enable**. Enabled only for output registered dual-port RAMs to provide this optional pin.
- **Common Output CLK**. Enabled only for registered dual-port RAMs. If not selected, the SPO registers are clocked by the CLK input and the DPO registers are clocked from the QDPO_CLK input. Default setting is selected, where all output registers are clocked from the CLK input.
- **Common Output CE**. Enabled only for registered dual-port RAMs and only if Common Output Clock and Dual-Port Output Clock Enable are also selected. If Common Output CE is deselected, the SPO register clocks are enabled by the QSPO_CE input and the DPO register clocks are enabled from the QDPO_CE input. Default setting is selected, where all output register clocks are enabled by the QSPO_CE input.

Initial Content and Reset Options

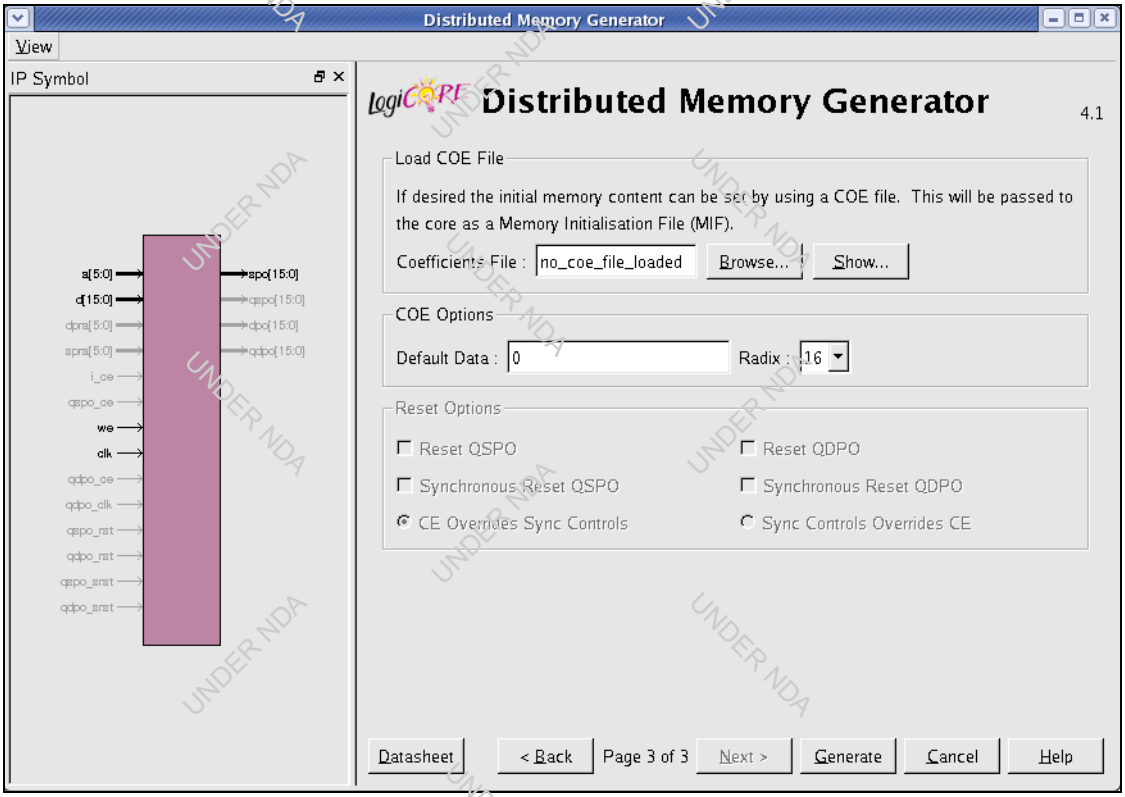

Figure 11: **Initial Content and Reset Options Screen**

Load COE File

The initial values of the memory elements can be set using a COE file.

- To load the COE file, click Browse.
- To view the initial contents, click Show.

For a description of the COE file, see "Specifying Memory Contents Using a COE File."

COE Options

Enter the initial value to be stored in memory locations not otherwise initialized in the COE file.

- Default Data: When no value is entered, the field defaults to 0. Values can be entered in binary, decimal, or hexadecimal formats, as defined by the Default Data Radix entry.
- **Radix**: Choose the radix of the Default Data value. Valid entries are 2, 10, and 16.

Reset Options

- **Reset QSPO**: Enabled only when the core has a registered single-port output. If selected, an asynchronous single-port output reset pin is available.
- **Reset QDPO**: Enabled only when the core has a registered dual-port output. If selected, an asynchronous dual-port output reset pin is available.
- **Synchronous Reset QSPO**: Enabled only when the core has a registered single-port output. If selected, a synchronous single-port output reset pin is available.
- **Synchronous Reset QDPO**: Enabled only when the core has a registered dual-port output. If selected, a synchronous dual-port output reset pin is available.
- **CE Overrides Sync Controls**: Enabled only when one of the synchronous reset options has been selected and an output clock enable has been selected. When selected, the synchronous control signals are qualified by the clock enable pin.
- **Sync Controls Overrides CE**: Enabled only when one of the synchronous reset options has been selected and an output clock enable has been selected. When selected, the synchronous control signals operate regardless of the state of the output clock enable signals.

Specifying Memory Contents Using a COE File

The initial contents of the memory can be defined using the COE file, which consists of two parameters: memory_initialization_radix and memory_initialization_vector. Each line is terminated by a semicolon.

- **memory_initialization_radix**. The radix of the initialization value is specified here, with the choices being 2, 10, or 16.
- **memory_initialization_vector**. Each row of memory elements are defined with a binary, decimal, or hexadecimal number having an equivalent binary value that represents whether an individual memory element along the width of the row is set to '1' or '0.' Each row of memory initialization is separated by a comma or white space, up to the depth of the memory. Negative values are not allowed.

An example COE file:

```
; Sample Initialization file for a 16x32 distributed ROM
memory_initialization_radix = 16;
memory_initialization_vector =
23f4 0721 11ff ABe1 0001 1 0A 0
23f4 0721 11ff ABe1 0001 1 0A 0
23f4 721 11ff ABe1 0001 1 A 0
23f4 721 11ff ABe1 0001 1 A 0;
```

MIF File Description

The COE file provides a high-level method for specifying initial memory contents. During core generation, the COE file is converted into a MIF file, which holds the actual binary data used to initialize the memory in the core and simulation models. The MIF file consists of one line of text per memory location. The first line in the file corresponds to address 0, and the second line corresponds to address 1, and

so forth. The text on each line must be the initialization value (MSB first) for the corresponding memory address in binary format, with exactly one binary digit per bit of memory width.

Note: For HDL simulations, the MIF file must reside in the simulation directory.

Core Implementation

Functional Simulation

VHDL and Verilog behavioral models of the Distributed Memory Generator core are provided with the XilinxCoreLib library for use within a simulation environment. Note that neither a test bench nor test fixture is provided with the Distributed Memory Generator.

The functional simulation model for this core is behavioral. Certain conditions such as out-of-range writes are not modelled in a cycle-accurate manner. Run timing simulation to more closely simulate cycle-based behavior.

Synthesis

Synthesis of the memory produced by the Distributed Memory Generator is performed with XST by the CORE Generator software.

Xilinx Tools

See the LogiCORE IP Facts table on page 1.

Static Timing Analysis

Static timing analysis can be performed using `trce`, following `ngdbuild`, `map`, and `par`.

Gate-level Simulation

If desired, the CORE Generator software can create a UniSim-based VHDL or Verilog model for gate level simulation, which is delivered with the NGC netlist. Alternatively, the netlist produced by the CORE Generator for the desired memory can be processed using `ngdbuild` and `netgen` to produce a SimPrim-based model for simulation.

Verification

Verification of the netlist produced by the CORE Generator is completed by a cycle-by-cycle comparison against the VHDL and Verilog behavioral models of the core contained within the XilinxCoreLib behavioral model library.

Core Resource Utilization

When designing distributed memories, the LUTs in the slices are used to implement memory primitives or to construct multiplexers. The registers available in these slices are used for input and output registering.

When input registering is requested, extra registers are used. One per control bit (WE, QSPO_CE, and QDPO_CE) and one per bit of data and address (D[P:0], A[N:0], and SPRA[N:0]/DPRA[N:0]).

Ordering Information

The Distributed Memory core is a free core provided under the SignOnce IP Site License. It can be generated using the Xilinx CORE Generator system v11.1 or higher. The CORE Generator system is shipped with Xilinx ISE Foundation Series Development software.

For more information about the core, see the Distributed Memory Generator product page at:

www.xilinx.com/products/ipcenter/DIST_MEM_GEN.htm

New Architecture

All functions of the Distributed Memory Generator are supported for Virtex-6 and Spartan-6 FPGAs.

Related Information

Xilinx products are not intended for use in life-support appliances, devices, or systems. Use of a Xilinx product in such application without the written consent of the appropriate Xilinx officer is prohibited.

Revision History

Date	Version	Revision
10/20/04	1.0	Initial draft.
4/28/05	1.1	Initial Xilinx release.
1/11/06	2.0	Updated version, ISE to 8.1i, fixed licensing section.
7/13/06	3.0	Updated for IP1i minor release. Added Virtex-5 support.
9/21/06	4.0	Updated core version to 3.2; added support for Spartan-3A.
2/15/07	5.0	Updated core version to 3.3; Xilinx tools 9.1i; added support for Spartan-3AN.
4/02/07	5.5	Added support for Spartan-3A DSP devices.
10/15/07	6.0	Updated core version to 3.4; Corrected Figure 6.
3/24/08	7.0	Updated core to version 3.4; ISE tools to 10.1.
12/17/08	7.0.1	Updated core to version 4.1; ISE tools to 11.1.

Notice of Disclaimer