Distributed Memory V2.0

June 30, 2000 | Product Specification

Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com/support/techsup/appinfo
www.xilinx.com/ipcenter

Features

- Drop-in module for Virtex, Virtex™-E, Spartan™–II and Virtex™–II FPGAs
- Generates ROMs, single/dual-port RAMs and SRL16 based RAMs
- Supports data widths ranging from 1 to 64 bits wide
- Supports depths ranging from 16 to 256 words
- Optional registered output and pipelining
- Relationally Placed Macro (RPM) mapping and placement technology
- Incorporates Xilinx Smart-IP technology for maximum performance

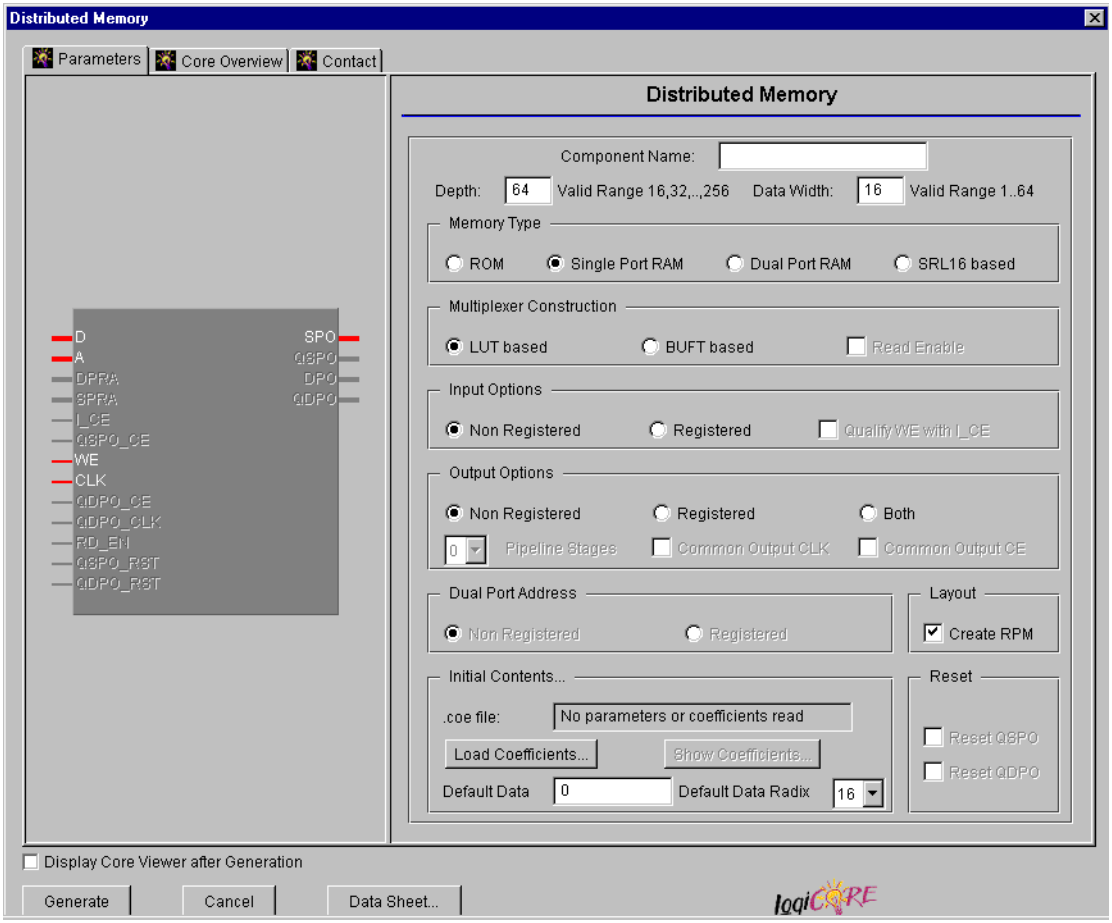

Figure 1: Main Distributed Memory Parameterization Screen

- To be used with version 3.1i and later of the Xilinx CORE Generator System

Functional Description

The distributed memory module is used to create memory structures using the Select-RAM. It can be used to create Read Only Memory (ROM), single port Random Access Memory (RAM), pseudo dual port RAM and SRL16 based. Data widths of up to 64 bits are supported with depths of up to 256 words. Options are available for simple registering of inputs and outputs in addition to pipelining capabilities. Optional asynchronous resets are available for the output registers. The module can optionally be generated as a Relationally Placed Macro (RPM) or as unplaced logic.

Pinout

Signal names are shown in Figures 2, 3, 4 and 5 and described in Table 1.

CORE Generator Parameters

The main CORE Generator parameterization screen for this module is shown in Figure 1. The parameters are as follows:

- **Component Name**: The component name is used as the base name of the output files generated for this module. Names must begin with a letter and must be composed from the following characters: a to z, 0 to 9 and "_" .
- **Depth**: Enter the required memory depth. The valid range is 16 to 256 in steps of 16. The default value is 64.
- **Data Width**: Enter the width of the memory. The valid range is 1 to 64. The default value is 16.
- **Memory Type**: Select the appropriate radio button for the types of memory required. The default setting is **Single Port RAM**.
  - **ROM**: A schematic diagram showing the structure of the ROM modules is shown in Figure 6. The Address register is optional (controlled by the setting of the **Input Options** parameter). Output registering and pipelining are also optional (controlled by the setting of the **Output Options** parameter). The CLK is not required if no registers are present. The reset is optional.
  - **Single Port RAM**: A schematic diagram showing the structure of the single port RAM modules is shown in Figure 8. The address and data registers are optional (controlled by the setting of the I**nput Options** parameter). Output registering and pipelining are also optional (controlled by the setting of the **Output Options** parameter). The reset is optional

Table 1: Core Signal Pinout

Signal	Signal 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{N:0] outputs.
SPRA	Input	Single Port Read Address. Port is only present on SRL16 based RAMs and defines memory location read out on the SPO(N:0) outputs.
DPRA[N:0]	Input	Dual Port Read Address. Port is only present on Dual port RAMs and defines memory location read out on the DPO{N:0] outputs.
SPO[P:0]	Output	Non-registered single port output bus. Non-registered data output bus for ROMs, single-port RAMs and SRL16 based. One of two non-registered output buses on dual-port RAMs.
QSPO[P:0]	Output	Registered single port output bus. Registered data output bus for ROMs, single-port RAMs and SRL16 based. One of two non-registered output buses on dual-port RAMs.
DPO[P:0]	Output	Non-registered dual port output bus - one of the non-registered data output bus 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.

Signal	Signal Direction	Description
CLK	Input	Write clock and register clock for ROMs, single-port RAMs and SRL16 based. 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.
RD_EN	Input	Read Enable
QSPO_CE	Input	On ROMs clock enable controls all input and output registers. On dual-port memories controls output register and pipeline registers in QSPO path.
QDPO_CE	Input	Only present on dual-port RAMs. Controls output register and pipeline registers in QDPO path.
QSPO_RST	Input	Single port registered output reset.
QDPO_RST	Input	Only available on dual-port RAMs. Dual port registered output reset.

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

- **Dual Port RAM**: A schematic diagram showing the structure of the dual port RAM modules is shown in Figure 9. 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 registering and pipelining 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 chosen for the **Common Output Clock** and

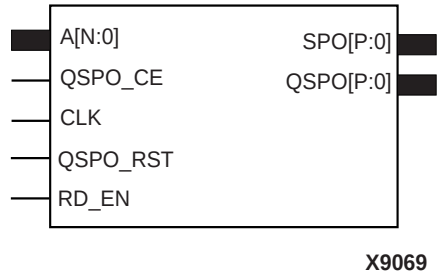

Figure 2: Core Schematic Symbol for ROMs

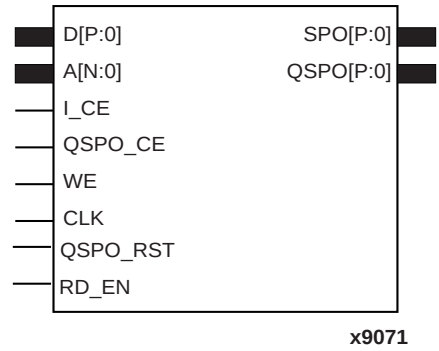

Figure 3: Core Schematic Symbol for single-port RAMs

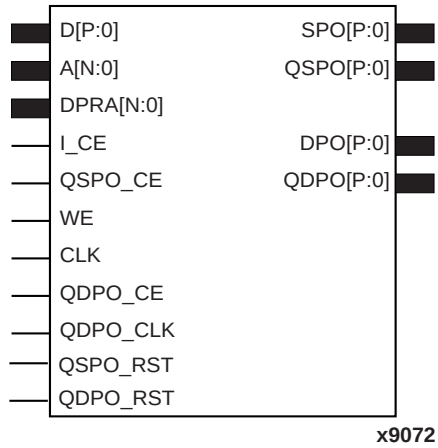

Figure 4: Core Schematic Symbol for dual-port RAMs

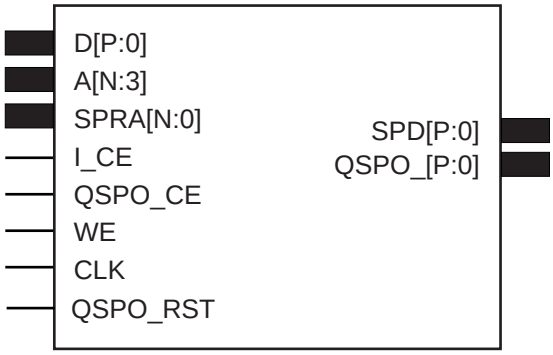

Figure 5: Core Schematic Symbol for SRL16-based RAM

Common Output CE parameters). Both resets are optional.

- **SRL16 based**: A schematic diagram showing the structure of the SRL16 based module is shown in Figure 7. The address and data registers are optional (controlled by the setting of the **Input Options** parameter). Output registering and pipelining are also optional (controlled by the setting of the **Output Options** parameter). The reset is optional.

  Note: The Dual Port RAM implementations are based on using the Select RAM feature of the LUTs. In this mode the LUTs behave as a pseudo dual port memory and for details of the fundamental operation refer to the Virtex Product Specification.

  Note: The QSPO_CE and QDPO_CE registers are automatically generated when input and outputs are both registered in order to ensure correct phasing of the CE signal relative to the addresses.

- **Multiplexer Construction**: For modules with depths greater than 16 (dual-port RAMs and SRL16 based RAMs) or 64 (ROMS and single-port RAMS) multiplexing is required in the output paths. This can be performed using multiplexers built in Look Up Tables (LUTs) or using the tristate buffer primitives (BUFTs). If the multiplexer is constructed using BUFTs, the output of the module cannot be registered, although one pipeline stage is optionally available. BUFT based multiplexers are not supported in the dual-port or SRL16 based RAM modules. The default setting is **LUT Based**.
- **Read Enable**: This optional read enable pin is only available when BUFT based multiplexers are used.
- **Input Options**: Select the appropriate radio button for the types of inputs required. The action of setting this parameter to **Registered** will have different effects depending on the **Memory Type** selection:
- **ROM**: An address register will be generated
- **Single Port RAM**, **Dual Port RAM** and **SRL16 based**: A register on the A[N:0] Address input, a data input register and a WE register will be generated.
- The default setting is **Non Registered**.
- **Note**: The QSPO_CE and QDPO_CE registers are automatically generated when input and outputs are both registered in order to ensure correct phasing of the CE signal relative to the addresses.
  - **Qualify WE with I_CE**: This parameter is only valid for single port RAM, dual port RAM, and SRL16 based RAM with **Input Options** set to **Registered**. When the checkbox is not checked the WE register has no clock enable control. When checked the WE register has a clock enable that is driven by the I_CE input.
  - **Dual Port Address**: This parameter is only valid for dual port RAMs. It controls the presence or absence of a register on the DPRA[N:0] inputs. The default setting is **Non Registered**.
  - **Output Options**: Select the appropriate radio button for the types of outputs required. The default setting is **Non Registered**.
  - **Pipeline Stages**: This parameter controls the amount of pipelining in the output side of the modules. Pipelining is only supported when the **Output Options** parameter is set to **Registered**. The amount of pipelining available depends on the **Memory Type** and the **Depth** of the memory. When 1 pipeline stage is selected the register is at the output of the module. A second pipeline stage adds registers at the outputs of the memory primitives. On modules that support a third pipeline stage the third register is added in the middle of the multiplexer. When a module is maximally pipelined its operating

Table 2: Amount of Pipelining Supported for Different Memory Types and Depths

Type of Memory	Number of Pipeline Stages supported		
	1 stage	2 stages	3 stages
ROM	≤ 64 words deep	> 64 words deep	
Single-port RAM	≤ 64 words deep	>-64 words deep	
Dual-port RAM	≤ 16 words deep	> 16 ≤ 128 words deep	> 128 words deep
SRL16 based	≤ 16 words deep	> 16 ≤ 128 words deep	> 128 words deep

speed will be maximized. The amount of pipelining supported for LUT-based **Multiplexer Construction** depends on the **Memory Type** and is shown in Table 2. For BUFT-based Multiplexer Construction there can only be one stage of pipelining. In this case the registers are placed at the outputs of the memory primitives.

- **Common Output Clock**: This checkbox is only enabled for registered dual port RAMs. If not checked, the SPO registers will be clocked by the CLK input and the DPO registers will be clocked from the QDPO_CLK input. The default is checked, where all output registers are be clocked from the CLK input.
- **Common Output CE**: This checkbox is only enabled for registered dual port RAMs and only if **Common Output Clock** is also checked. If **Common Output CE** is not checked, the SPO register clocks will be enabled by the QSPO_CE input and the DPO register clocks will be enabled from the QDPO_CE input. The default is checked, where all output register clocks are enabled by the QSPO_CE input.
- **Create RPM**: When this box is checked the module will be generated with relative location attributes attached. The resulting placement of the module will be in a column with two bits per slice. The default setting is to create an RPM.
- **Initial Contents...**: The initial values of the memory elements can be set with the use of a Coefficients file (COE), by loading the ".coe" file using the **Load Coefficients...** button. The initial contents can be viewed by selecting the **Show Coefficients...** button. For a description of the COE file refer to the section titled "Specifying Memory Contents using a COE file". The contents of the COE file are converted to a Memory Initialization File (MIF) with the values in a binary format. This file describes the true memory contents that are used by the core and the simulation models. For a description of the

Table 3: XCO File Values and Default Values

Parameter	XCO File Values	Default GUI Setting
component_name	ASCII text starting with a letter and based upon the following character set: a .. z, 0..9 and _	blank
depth	Integer in the range 16 to 256 in steps of 16	64
data_width	Integer in the range 1 to 64	16
memory_type	One of the following keywords: rom, single_port_ram, dual_port_ram, srl16_based	single_port_ram
multiplexer_construction	One of the following keywords: lut_based or buft_based	lut_based
read_enable	One of the following keywords: true, false	false
input_options	One of the following keywords: non_registered, registered	non_registered
qualify_we_with_i_ce	One of the following keywords: true, false	false
dual_port_address	One of the following keywords: non_registered, or registered	non_registered
output_options	One of the following keywords: non_registered, registered, both	non_registered
pipeline_stages	One of the following keywords: 0, 1, 2, or 3	0
common_output_clk	One of the following keywords: true, false	false
common_output_ce	One of the following keywords: true, false	false
create_rpm	One of the following keywords: true, false	true
coefficient_file	ASCII text starting with a letter and based upon the following character set: a...z, 0...9 and _, and must end with a ".coe" extension	blank
default_data	Numeric value in the Radix specified by the default_data_radix keyword whose value does not exceed 2 $^{DATA\ WIDTH}$ -1	0
default_data_radix	One of the following keywords: 2, 10, 16	16
reset_qspo	One of the following keywords: true, false	false
reset_qdpo	One of the following keywords: true, false	false

memory initialization file refer to the section entitled “MIF File description.”

- **Default Data**: Enter the initial value to be stored in any memory location not specified by another means. When no value is entered this field defaults to 0. Values may be entered in Binary, Decimal or Hex format, as defined by the **Default Data Radix** entry.
- **Default Data Radix**: Choose the radix of the **Default Data** value. Valid entries are 2, 10 and 16.

- **Reset**
  - **Reset QSPO**: This checkbox is only enabled when the core has a registered single port output. If checked, a reset pin will be available to asynchronously clear this output.
  - **Reset QDPO**: This checkbox is only enabled when the core has a registered dual port output. If checked, a reset pin will be available to reset this output.

Specifying Memory Contents using a COE File

The initial contents of the memory can be defined using a text file known as a Coefficient (COE) File. COE files must have a “.coe” extension.

The COE file consists of two parameters similar to an XCO file, but the end of each line is determined with the use of a semi-colon. The two parameters are:

- **memory_initialization_vector**: Each row of memory elements are defined with a binary, decimal or hexadecimal number whose equivalent binary value represents whether an individual memory element along the width of the row is set to a ‘1’ or a ‘0’. Each row of memory initialization is separated by a comma, up to the depth of the memory.
- **memory_initialization_radix:** The radix of the initialization value is specified here, with the choice being 2, 10 or 16.

An example of a COE file is shown below:

```
memory_initialization_vector = 23f4,721,11ff;

memory_initialization_radix = 16
```

MIF File Description

The COE file provides a wrapper to the user to allow memory contents to be initialized. However, the MIF file holds the actual binary data that is 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 corresponding to address 0, the second line corresponding to address 1, and so on. 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 the memory’s width.

Parameter Values in the XCO File

Names of XCO file parameters and their parameter values are identical to the names and values shown in the GUI, except that underscore characters (_) are used instead of spaces. The text in an XCO file is case insensitive.

Table 3 shows the XCO file parameters and values, as well as summarizing the GUI defaults. The following is an example of the CSET parameters in an XCO file:

```
CSET component_name = abc123
CSET depth = 64
CSET data_width = 16
CSET memory_type = Single_Port_RAM
CSET multiplexer_construction = LUT_based
CSET read_enable = FALSE
CSET input_options = Non_Registered
CSET qualify_we_with_i_ce = FALSE
CSET dual_port_address = Non_Registered
CSET output_options = Non_Registered
CSET pipeline_stages = 0
CSET common_output_clk = TRUE
CSET common_output_ce = TRUE
CSET create_rpm = TRUE
CSET coefficient_file = a.coe
CSET default_data = 0
CSET default_data_radix = 16
CSET reset_qspo = FALSE
CSET reset_qdpo = FALSE
```

Core Resource Utilization

When output registering or any style of pipelining is requested the registers used are in slices already used for the memory primitives or LUTs that are used to construct the multiplexers, so no additional slices are required.

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

If both input registering and output registering are requested, extra registers are required to implement the A Pipe Register and (for dual port RAMs only) the DPRA Pipe Register. This requires one additional flip-flop per bit of A[N:0] and (on dual port RAMs only) DPRA[N:0].

For an accurate measure of the usage of primitives, slices, and CLBs for a particular point solution, check the **Display Core Viewer after Generation** checkbox, in CoreGen.

Ordering Information

This core is downloadable free of charge from the Xilinx IP Center (www.xilinx.com/ipcenter), for use with the Xilinx Core Generator System version 3.1i and later. The Core Generator System 3.1i tool is bundled with the Alliance 3.1i and Foundation 2.1i implementation tools.

To order Xilinx software contact your local Xilinx sales representative at www.xilinx.com/company/sales.htm.

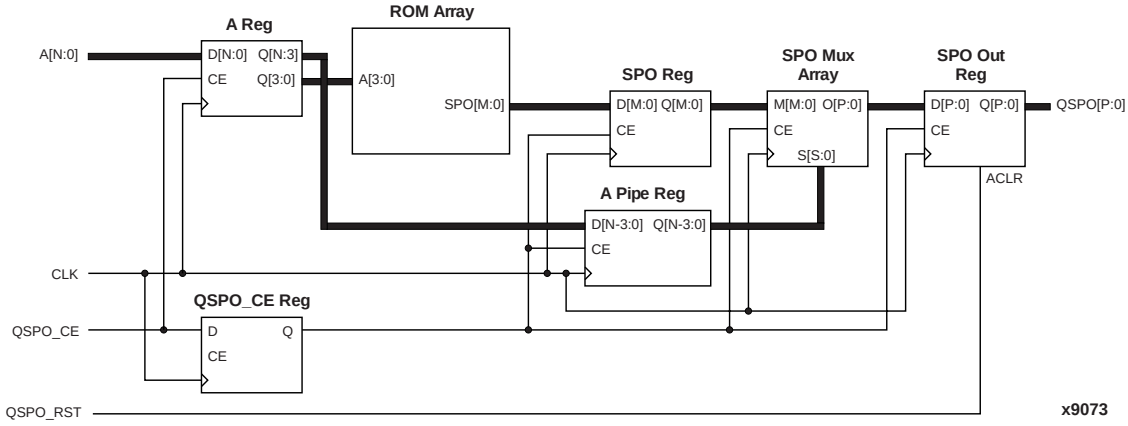

Figure 6: Schematic of a ROM module

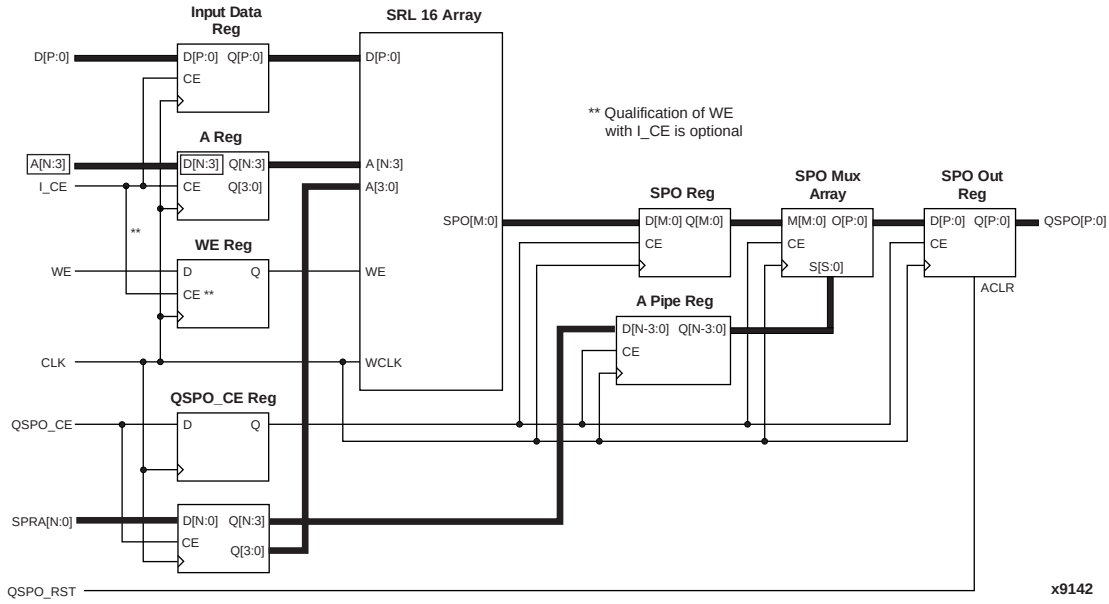

Figure 7: Schematic of an SRL16-Based RAM Module

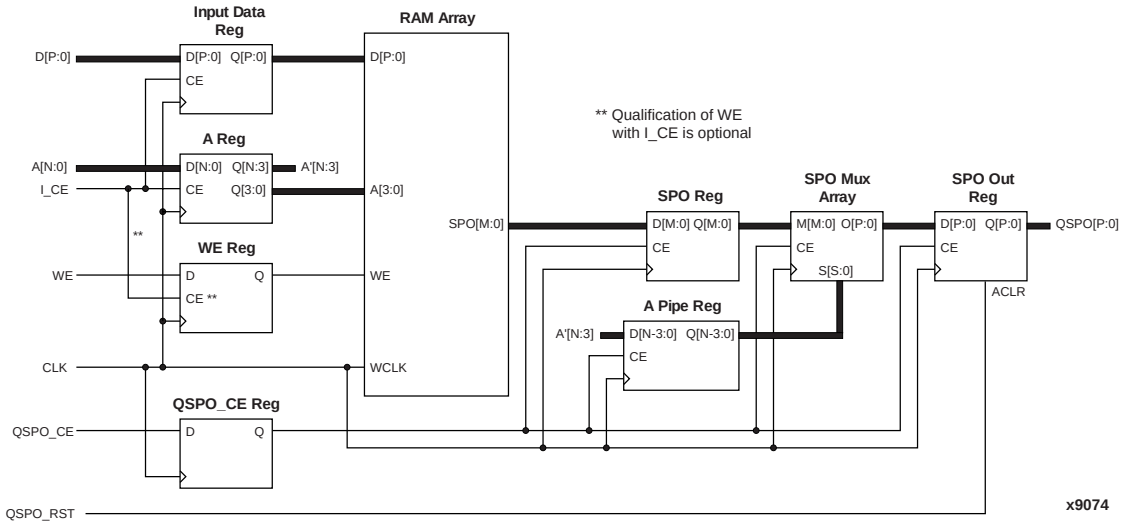

Figure 8: Schematic of a Single Port RAM Module

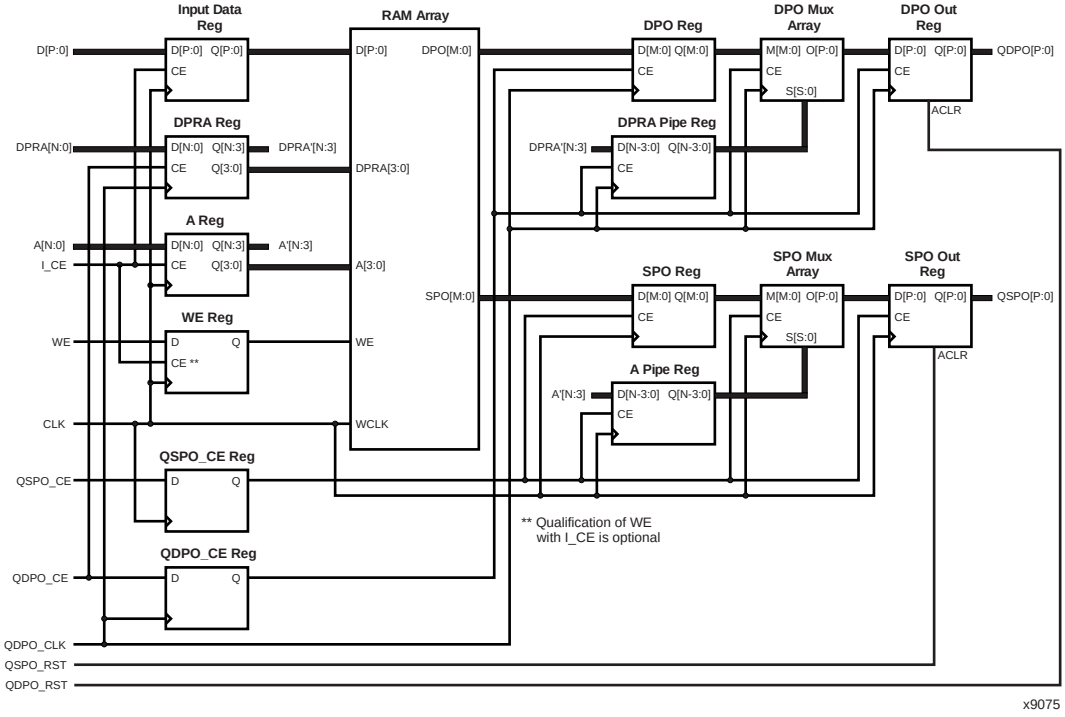

Figure 9: Schematic of a Dual Port RAM Module