Using Distributed SelectRAM Memory

Introduction

In addition to 18Kb SelectRAM blocks, Virtex-II devices feature distributed SelectRAM modules. Each function generator or LUT of a CLB resource can implement a 16 x 1-bit synchronous RAM resource. Distributed SelectRAM memory writes synchronously and reads asynchronously. However, a synchronous read can be implemented using the register that is available in the same slice. This 16 x 1-bit RAM is cascadable for a deeper and/or wider memory implementation, with a minimal timing penalty incurred through specialized logic resources.

Distributed SelectRAM modules up to a size of 128 x 1 are available as primitives. Two 16 x 1 RAM resources can be combined to form a dual-port 16 x 1 RAM with one dedicated read/write port and a second read-only port. One port writes into both 16 x1 RAMs simultaneously, but the second port reads independently.

This section provides generic VHDL and Verilog reference code examples implementing *n*-bit-wide single-port and dual-port distributed SelectRAM memory.

Distributed SelectRAM memory enables many high-speed applications that require relatively small embedded RAM blocks, such as FIFOs, which are close to the logic that uses them.

Virtex-II Distributed SelectRAM memories can be generated using the CORE Generator Distributed Memory module (V2.0 or later). The user can also generate Distributed RAM-based Asynchronous and Synchronous FIFOs using the CORE Generator.

Single-Port and Dual-Port RAM

Data Flow

Distributed SelectRAM memory supports the following:

- Single-port RAM with synchronous write and asynchronous read
- Dual-port RAM with one synchronous write and two asynchronous read ports

As illustrated in the Figure 2-49, the dual port has one read/write port and an independent read port.

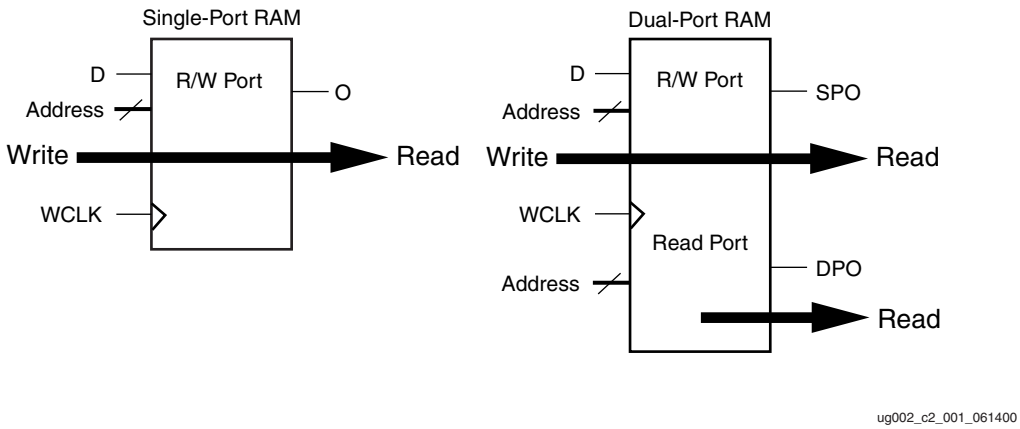

Figure 2-49: **Single-Port and Dual-Port Distributed SelectRAM**

Any read/write operation can occur simultaneously with and independently of a read operation on the other port.

Write Operations

The write operation is a single clock-edge operation, with a write enable that is active High by default. When the write enable is Low, no data is written into the RAM. When the write enable is High, the clock edge latches the write address and writes the data on D into the RAM.

Read Operation

The read operation is a combinatorial operation. The address port (single or dual port) is asynchronous with an access time equivalent to the logic delay.

Read During Write

When new data is synchronously written, the output reflects the data in the memory cell addressed (transparent mode). The timing diagram in Figure 2-50 illustrates a write operation, with the previous data read on the output port, before the clock edge and then the new data.

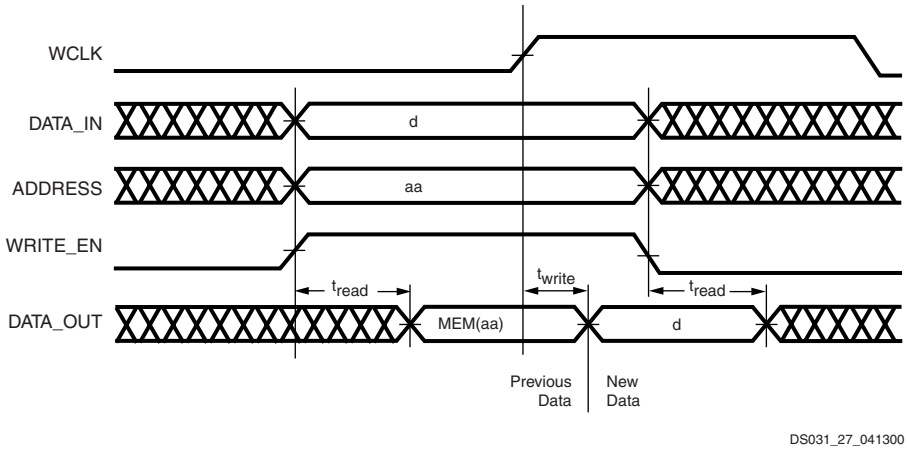

Figure 2-50: **Write Timing Diagram**

Characteristics

- A write operation requires only one clock edge.
- A read operation requires only the logic access time.
- Outputs are asynchronous and dependent only on the logic delay.
- Data and address inputs are latched with the write clock and have a setup-to-clock timing specification. There is no hold time requirement.
- For dual-port RAM, one address is the write and read address, the other address is an independent read address.

Library Primitives

Seven library primitives from 16 x 1-bit to 128 x 1-bit are available. Four primitives are single-port RAM and three primitives are True Dual-Port RAM, as shown in Table 2-15.

Table 2-15: **Single-Port and Dual-Port Distributed SelectRAM**

Primitive	RAM Size	Type	Address Inputs
RAM16X1S	16 bits	single-port	A3, A2, A1, A0
RAM32X1S	32 bits	single-port	A4, A3, A2, A1, A0
RAM64X1S	64 bits	single-port	A5, A3, A2, A1, A0
RAM128X1S	128 bits	single-port	A6, A4, A3, A2, A1, A0
RAM16X1D	16 bits	dual-port	A3, A2, A1, A0
RAM32X1D	32 bits	dual-port	A4, A3, A2, A1, A0
RAM64X1D	64 bits	dual-port	A5, A4, A3, A2, A1, A0

The input and output data are 1-bit wide. However, several distributed SelectRAM memories can be used to implement wide memory blocks.

Figure 2-51 shows generic single-port and dual-port distributed SelectRAM primitives. The A and DPRA signals are address busses.

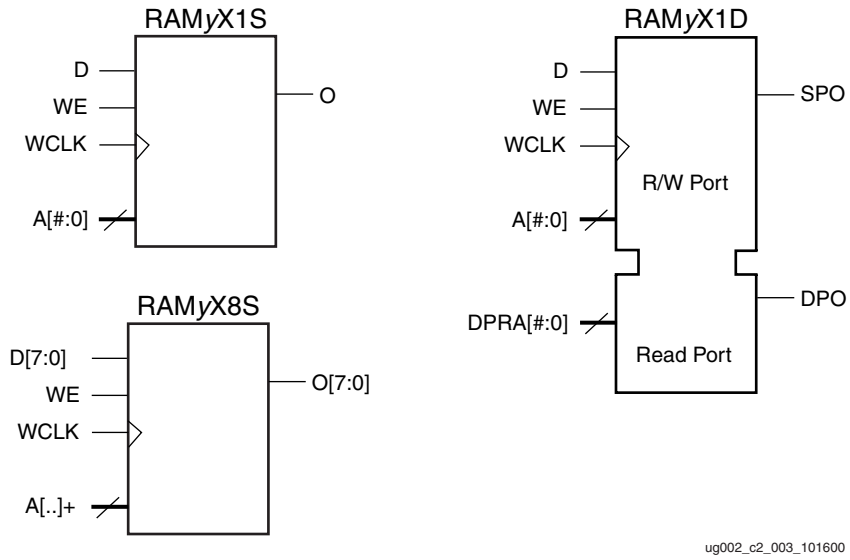

Figure 2-51: **Single-Port and Dual-Port Distributed SelectRAM Primitive**

As shown in Table 2-16, wider library primitives are available for 2-bit, 4-bit, and 8-bit RAM.

Table 2-16: **Wider Library Primitives**

Primitive	RAM Size	Data Inputs	Address Inputs	Data Outputs
RAM16x2S	16 x 2-bit	D1, D0	A3, A2, A1, A0	O1, O0
RAM32X2S	32 x 2-bit	D1, D0	A4, A3, A2, A1, A0	O1, O0
RAM64X2S	64 x 2-bit	D1, D0	A5, A4, A3, A2, A1, A0	O1, O0
RAM16X4S	16 x 4-bit	D3, D2, D1, D0	A3, A2, A1, A0	O3, O2, O1, O0
RAM32X4S	32 x 4-bit	D3, D2, D1, D0	A4,A3, A2, A1, A0	O3, O2, O1, O0
RAM16X8S	16 x 8-bit	D <7:0>	A3, A2, A1, A0	O <7:0>
RAM32X8S	32 x 8-bit	D <7:0>	A4,A3, A2, A1, A0	O <7:0>

VHDL and Verilog Instantiation

VHDL and Verilog instantiations templates are available as examples (see "VHDL and Verilog Templates" on page 221).

In VHDL, each template has a component declaration section and an architecture section. Each part of the template should be inserted within the VHDL design file. The port map of the architecture section should include the design signal names.

The SelectRAM_*x*S templates (with *x* = 16, 32, 64, or 128) are single-port modules and instantiate the corresponding RAM*x*X1S primitive.

SelectRAM_*x*D templates (with *x* = 16, 32, or 64) are dual-port modules and instantiate the corresponding RAM*x*X1D primitive.

Ports Signals

Each distributed SelectRAM port operates independently of the other while reading the same set of memory cells.

Clock - WCLK

The clock is used for the synchronous write. The data and the address input pins have setup time referenced to the WCLK pin.

Enable - WE

The enable pin affects the write functionality of the port. An inactive Write Enable prevents any writing to memory cells. An active Write Enable causes the clock edge to write the data input signal to the memory location pointed to by the address inputs.

Address - A0, A1, A2, A3 (A4, A5, A6)

The address inputs select the memory cells for read or write. The width of the port determines the required address inputs. Note that the address inputs are not a bus in VHDL or Verilog instantiations.

Data In - D

The data input provides the new data value to be written into the RAM.

Data Out - O, SPO, and DPO

The data out O (Single-Port or SPO) and DPO (Dual-Port) reflects the contents of the memory cells referenced by the address inputs. Following an active write clock edge, the data out (O or SPO) reflects the newly written data.

Inverting Control Pins

The two control pins (WCLK and WE) each have an individual inversion option. Any control signal, including the clock, can be active at 0 (negative edge for the clock) or at 1 (positive edge for the clock) without requiring other logic resources.

GSR

The global set/reset (GSR) signal does not affect distributed SelectRAM modules.

Attributes

Content Initialization - INIT

With the INIT attributes, users can define the initial memory contents after configuration. By default distributed SelectRAM memory is initialized with all zeros during the device configuration sequence. The initialization attribute INIT represents the specified memory

contents. Each INIT is a hex-encoded bit vector. Table 2-17 shows the length of the INIT attribute for each primitive.

Table 2-17: **INIT Attributes Length**

Primitive	Template	INIT Attribute Length
RAM16X1S	SelectRAM_16S	4 digits
RAM32X1S	SelectRAM_32S	8 digits
RAM64X1S	SelectRAM_64S	16 digits
RAM128X1S	SelectRAM_128S	32 digits
RAM16X1D	SelectRAM_16S	4 digits
RAM32X1D	SelectRAM_32S	8 digits
RAM64X1D	SelectRAM_64S	16 digits

Initialization in VHDL or Verilog Codes

Distributed SelectRAM memory structures can be initialized in VHDL or Verilog code for both synthesis and simulation. For synthesis, the attributes are attached to the distributed SelectRAM instantiation and are copied in the EDIF output file to be compiled by Xilinx Alliance Series™ tools. The VHDL code simulation uses a `generic` parameter to pass the attributes. The Verilog code simulation uses a `defparam` parameter to pass the attributes.

The distributed SelectRAM instantiation templates (in VHDL and Verilog) illustrate these techniques (see "VHDL and Verilog Templates" on page 221).

Location Constraints

The CLB has four slices S0, S1, S2 and S3. As an example, in the bottom left CLB, the slices have the coordinates shown below: S

Slice S3	Slice S2	Slice S1	Slice S0
X1Y1	X1Y0	X0Y1	X0Y0

Distributed SelectRAM instances can have LOC properties attached to them to constrain placement. The RAM16X1S primitive fits in any LUT of slices S0 or S1.

For example, the instance `U_RAM16` is placed in slice X0Y0 with the following LOC properties:

```
INST "U_RAM16" LOC = "SLICE_X0Y0";
```

The RAM16X1D primitive occupies half of two slices, as shown in Figure 2-52. The first slice (output SPO) implements the read/write port with the same address A[3:0] for read

and write. The second slice implements the second read port with the address DPRA[3:0] and is written simultaneously with the first slice to the address A[3:0].

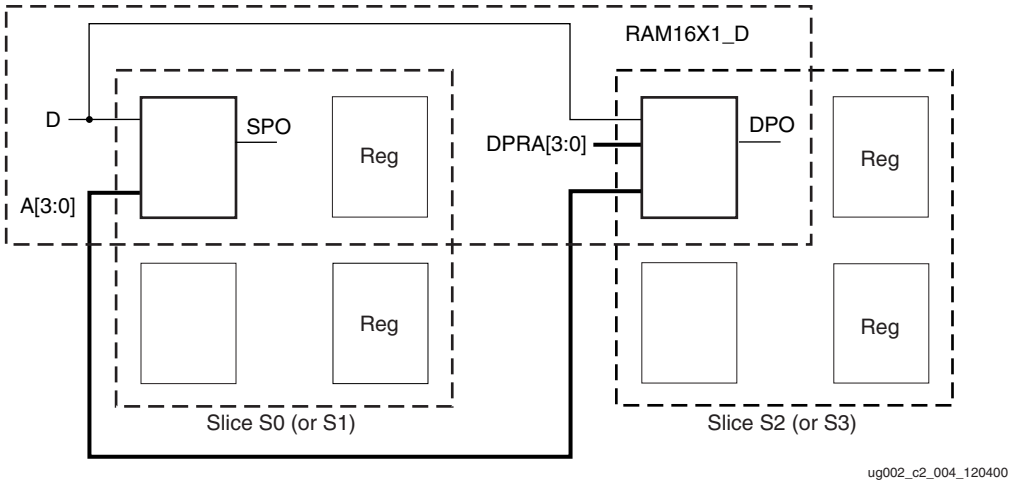

Figure 2-52: **RAM16X1_D Placement**

In the same CLB module, the dual-port RAM16X1_D either occupies half of slices S0 (X0Y0) and S2 (X1Y0), or half of slices S1 (X0Y1) and S3 (X1Y1).

If a dual-port 16 x 2-bit module is built, the two RAM16X1_D primitives occupy two slices, as long as they share the same clock and write enable, as illustrated in Figure 2-53.

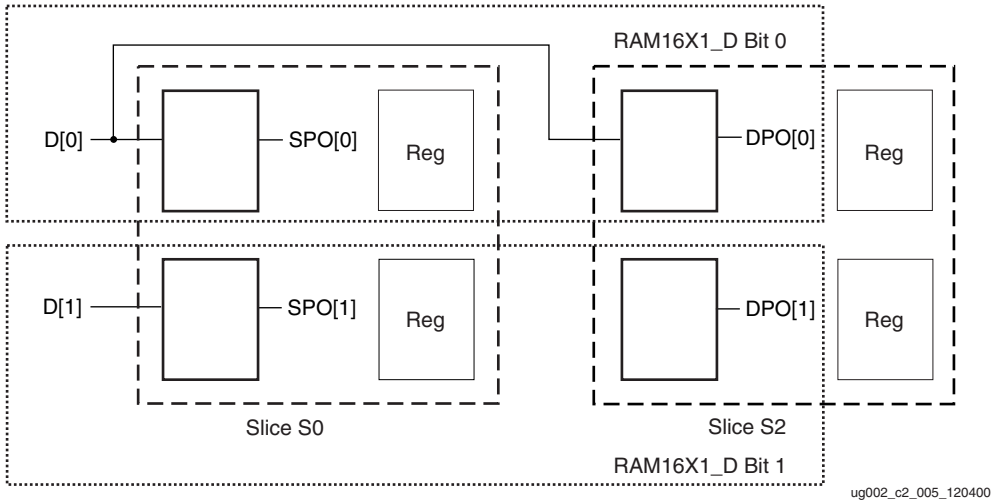

Figure 2-53: **Two RAM16X1_D Placement**

A RAM32X1S primitive fits in one slice, as shown in Figure 2-54.

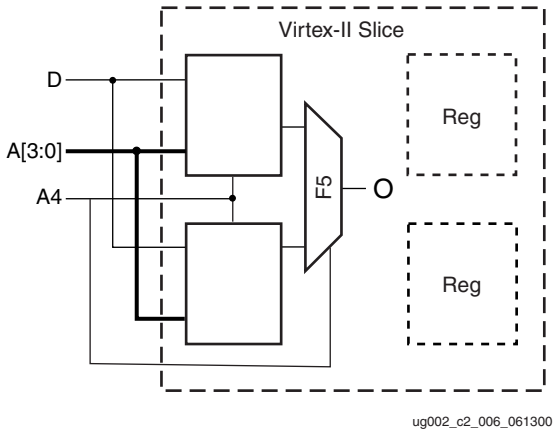

Figure 2-54: **RAM32X1_S Placement**

Following the same rules, a RAM32X1_D primitive fits in two slices, with one slice implementing the read/write port and the second slice implementing the second read port.

The RAM64X1_S primitive occupies two slices and the RAM64X1_D primitive occupies four slices (one CLB element), with two slices implementing the read/write port and two other slices implementing the second read port. The RAM64X1_S read path is built on the MUXF5 and MUXF6 multiplexers.

The RAM128X1_S primitive occupies four slices, equivalent to one CLB element.

Distributed SelectRAM placement locations use the slice location naming convention, allowing LOC properties to transfer easily from array to array.

Applications

Creating Larger RAM Structures

The memory compiler program generates wider and/or deeper memory structures using distributed SelectRAM instances. Along with an EDIF file for inclusion in a design, this program produces VHDL and Verilog instantiation templates and simulation models.

Table 2-18 shows the generic VHDL and Verilog distributed SelectRAM examples provided to implement *n*-bit-wide memories.

Table 2-18: **VHDL and Verilog Submodules**

Submodules	Primitive	Size	Type
XC2V_RAM16XN_S_SUBM	RAM16X1S	16 words x *n*-bit	single-port
XC2V_RAM32XN_S_SUBM	RAM32X1S	32 words x *n*-bit	single-port
XC2V_RAM64XN_S_SUBM	RAM64X1S	64 words x *n*-bit	single-port
XC2V_RAM128XN_S_SUBM	RAM128X1S	128 words x *n*-bit	single-port
XC2V_RAM16XN_D_SUBM	RAM16X1D	16 words x *n*-bit	dual-port
XC2V_RAM32XN_D_SUBM	RAM32X1D	32 words x *n*-bit	dual-port
XC2V_RAM64XN_D_SUBM	RAM64X1D	64 words x *n*-bit	dual-port

By using the read/write port for the write address and the second read port for the read address, a FIFO that can read and write simultaneously is easily generated. Simultaneous access doubles the effective throughput of the memory.

VHDL and Verilog Templates

VHDL and Verilog templates are available for all single-port and dual-port primitives. The number in each template indicates the number of bits (for example, SelectRAM_16S is the template for the 16 x 1-bit RAM); S indicates single-port, and D indicates dual-port.

In VHDL, each template has a component declaration section and an architecture section. Each part of the template should be inserted within the VHDL design file. The port map of the architecture section should include the design signal names.

The following are single-port templates:

- SelectRAM_16S
- SelectRAM_32S
- SelectRAM_64S
- SelectRAM_128S

The following are dual-port templates:

- SelectRAM_16D
- SelectRAM_32D
- SelectRAM_64D

Templates for the SelectRAM_16S module are provided in VHDL and Verilog code as examples.

VHDL Template

```
--
-- Module: SelectRAM_16S
--
-- Description: VHDL instantiation template
--              Distributed SelectRAM
--              Single Port 16 x 1
--              can be used also for RAM16X1S_1
--
-- Device: Virtex-II Family
--
-------------------------------------------------------------------
--
-- Components Declarations:
--
component RAM16X1S
-- pragma translate_off
   generic (
-- RAM initialization ("0" by default) for functional simulation:
         INIT : bit_vector := X"0000"
         );
-- pragma translate_on
  port (
         D    : in std_logic;
         WE   : in std_logic;
         WCLK : in std_logic;
         A0   : in std_logic;
         A1   : in std_logic;
         A2   : in std_logic;
         A3   : in std_logic;
         O    : out std_logic
         );
end component;
--
-------------------------------------------------------------------
--
-- Architecture  section:
--
-- Attributes for RAM initialization ("0" by default):
attribute INIT: string;
--
attribute INIT of U_RAM16X1S: label is "0000";
--
-- Distributed SelectRAM Instantiation
U_RAM16X1S: RAM16X1S
  port map (
    D      => , -- insert input signal
    WE     => , -- insert Write Enable signal
    WCLK   => , -- insert Write Clock signal
    A0     => , -- insert Address 0 signal
    A1     => , -- insert Address 1 signal
    A2     => , -- insert Address 2 signal
    A3     => , -- insert Address 3 signal
    O      =>   -- insert output signal
  );
--
-------------------------------------------------------------------
```

Verilog Template

```
//
// Module: SelectRAM_16S
//
// Description: Verilog instantiation template
//              Distributed SelectRAM
//              Single Port 16 x 1
//              can be used also for RAM16X1S_1
//
// Device: Virtex-II Family
//
//-----------------------------------------------------------------------
//
//
// Syntax for Synopsys FPGA Express
// synopsys translate_off

  defparam

       //RAM initialization ("0" by default) for functional simulation:
  U_RAM16X1S.INIT = 16'h0000;
// synopsys translate_on


//Distributed SelectRAM Instantiation
RAM16X1S U_RAM16X1S (  .D(),      // insert input signal
                 .WE(),     // insert Write Enable signal
                .WCLK(),   // insert Write Clock signal
                 .A0(),     // insert Address 0 signal
                 .A1(),     // insert Address 1 signal
                 .A2(),     // insert Address 2 signal
                 .A3(),     // insert Address 3 signal
                 .O()       // insert output signal
                  );

// synthesis attribute declarations
  /* synopsys attribute

  INIT "0000"
  */
```