



## Using Block SelectRAM+ Memory in Spartan-II FPGAs

XAPP173 (v1.0) November 23, 1999

Application Note

### Summary

The Spartan™-II FPGAs provide dedicated blocks of true dual-port RAM, known as Block SelectRAM™+ memory. This dedicated memory provides a cost-effective use of resources without sacrificing the existing distributed SelectRAM memory or logic resources. The Block SelectRAM+ memory is fully synchronous for easy timing analysis and is easily initialized at configuration. This additional integration capability makes the Spartan-II family ideal for cost-sensitive applications.

### Introduction

The Spartan-II Family provides Block SelectRAM+ memory, providing from four to 12 blocks of 4K bits of dual-port RAM. Both ports are configurable to any size from 4Kx1 to 256x16, allowing built-in bus width conversion. Each port is completely independent and fully synchronous, allowing the creation of flexible RAM structures. The dedicated RAM provides very high speed, comparable to discrete memories or ASIC memory cores.

The Block SelectRAM+ memory allows the Spartan-II FPGAs to handle on-chip memory requirements, which tend to grow faster than logic densities. It can be used for FIFOs to buffer data on and off chip, caches for high-speed parallel searches, ATM packet buffers, or simple bus-width converters. The integration of RAM and logic opens new application possibilities such as shift registers, state machines, and register stacks. Consolidating these memory-based functions into Spartan-II FPGAs further enhances the logic integration and cost-effectiveness of the family.

The Block SelectRAM+ memory is in addition to the popular distributed SelectRAM memory that can be built from the logic resources. The distributed SelectRAM memory is ideal for shallow memories that are closely integrated with logic, such as shift registers, pipeline registers, and scratch pad memories, or for applications requiring an asynchronous read capability. For very large blocks of memory, the Spartan-II FPGAs provide a number of SelectI/O interfaces for high-speed external memory access.

**Table 1** describes the depth and width aspect ratios for the Block SelectRAM+ memory. Blocks can be cascaded to create larger sizes, up to the total capabilities of a given device. The number of blocks per device is shown in **Table 2**.

### Synchronous Memory Basics

Dual-port and single-port synchronous RAM as discrete components have several different modes of operation for read and write operations. The four major modes of operation are:

#### Read Through (One Clock Edge)

The read address is registered on the read port clock edge and data appears on the output after the RAM access time. Some memories may place the latch/register at the outputs depending on the desire to have a faster clock-to-out versus setup time. This is generally considered to be an inferior solution since it changes the read operation to an asynchronous function with the possibility of missing an address/control line transition during the generation of the read pulse clock.

**Table 1: Block SelectRAM+ Port Aspect Ratios**

Width	Depth	ADDR Bus	Data Bus
1	4096	ADDR[11:0]	DATA[0]
2	2048	ADDR[10:0]	DATA[1:0]
4	1024	ADDR[9:0]	DATA[3:0]
8	512	ADDR[8:0]	DATA[7:0]
16	256	ADDR[7:0]	DATA[15:0]

**Table 2: Block RAM by Device**

Device	Blocks	Total Bits
XC2S15	4	16K
XC2S30	6	24K
XC2S50	8	32K
XC2S100	10	40K
XC2S150	12	48K

### Read Pipelined (two clock edges)

The read address is registered on the read port clock edge and the data is registered and appears on the output after the second read clock edge.

### Write Back (one clock edge)

The write address is registered on the write port clock edge and the data input is written to the memory and mirrored on the write port input.

### Write Through (one clock edge)

The write address is registered on the write port clock edge and the data is mirrored on the write port input as well as the read port output if the write and read addresses match. Data is written to and read from the memory in the same cycle.

The Spartan-II Block SelectRAM+ memory has the "read through" and "write back" functions. Simply by adding a CLB register to the outputs, designs requiring a "read pipeline" function can be done. This has the effect of improving the perceived clock-to-out timing of the Block SelectRAM+ memory with little device area cost.

### Block SelectRAM+ Characteristics

1. All inputs are registered with the port clock and have a set-up to clock timing specification.
2. All outputs have a read through or write back function depending on the state of the port WE pin. The outputs relative to the port clock are available after the clock-to-out timing specification.

3. The Block SelectRAM+ memories are true SRAM memories and do not have a combinatorial path from the address to the output. The LUT SelectRAM+ cells in the CLBs are still available with this function.
4. The ports are completely independent from each other (i.e., clocking, control, address, read/write function, and data width) without arbitration.
5. A write operation requires only one clock edge.
6. A read operation requires only one clock edge.
7. The output ports are latched with a self timed circuit to guarantee a glitch free read. The state of the output port will not change until the port executes another read or write operation.

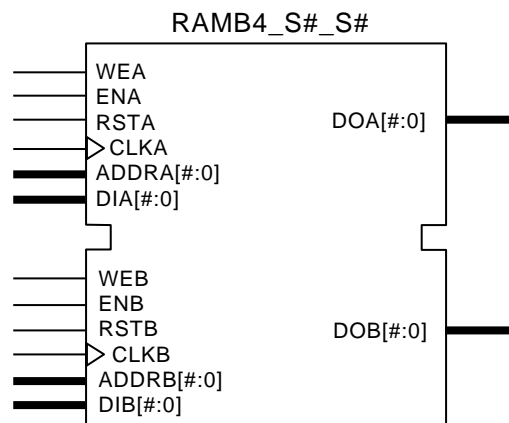
## Including Block SelectRAM+ Memory

Block SelectRAM+ memory can be included in a design by instantiating special library primitives or by using the CORE Generator System to create a custom memory. The primitives are available in all the configurations possible for a single block, from 4Kx1 to 256x16 single-port and dual-port memories. The CORE Generator System allows for any size memory that is possible in the Spartan-II architecture, up to 1M words.

Using or not using control inputs can control the functionality of the memory. For example, a ROM is created simply by not using the Write Enable and Data Inputs. If the Enable is tied High or unused the memory will always respond to writes and reads. If the Reset signal is tied Low or unused, the function that drives the outputs Low will not be available.

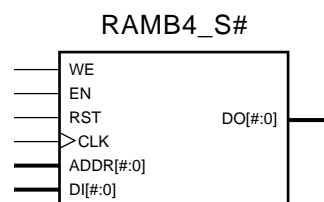
## Library Primitives

Figure 1 and Figure 2 show the two generic library Block SelectRAM+ primitives. Table 3 describes all of the available primitives for synthesis and simulation.



x130\_01\_091799

Figure 1: Dual-port Block SelectRAM+ Memory



X130\_02\_091799

Figure 2: Single-port Block SelectRAM+ Memory

**Table 3: Available Library Primitives**

Primitive	Port A Width	Port B Width
RAMB4_S1	1	N/A
RAMB4_S1_S1		1
RAMB4_S1_S2		2
RAMB4_S1_S4		4
RAMB4_S1_S8		8
RAMB4_S1_S16		16
RAMB4_S2	2	N/A
RAMB4_S2_S2		2
RAMB4_S2_S4		4
RAMB4_S2_S8		8
RAMB4_S2_S16		16
RAMB4_S4	4	N/A
RAMB4_S4_S4		4
RAMB4_S4_S8		8
RAMB4_S4_S16		16
RAMB4_S8	8	N/A
RAMB4_S8_S8		8
RAMB4_S8_S16		16
RAMB4_S16	16	N/A
RAMB4_S16_S16		16

## Port Signals

Each Block SelectRAM+ port operates independently of the other while accessing the same set of 4096 memory cells.

### Clock—CLK[A|B]

Each port is fully synchronous with independent clock pins. All port input pins have setup time referenced to the port CLK pin. The data output bus has a clock-to-out time referenced to the CLK pin.

### Enable—EN[A|B]

The enable pin affects the read, write and reset functionality of the port. Ports with an inactive enable pin keep the output pins in the previous state and do not write data to the memory cells.

### Write Enable—WE[A|B]

Activating the write enable pin allows the port to write to the memory cells. When active, the contents of the data input bus are written to the RAM at the address pointed to by the address bus, and the new data also reflects on the data out bus. When inactive, a read operation occurs and the contents of the memory cells referenced by the address bus reflect on the data out bus.

## Reset—RST[A|B]

The reset pin forces the data output bus latches to zero synchronously. This does not affect the memory cells of the RAM and does not disturb a write operation on the other port.

## Address Bus—ADDR(A/B[#:0])

The address bus selects the memory cells for read or write. The width of the port determines the required width of this bus as shown in [Table 1](#).

## Data In Bus—DI(A/B[#:0])

The data in bus provides the new data value to be written into the RAM. This bus and the port have the same width, as shown in [Table 1](#).

## Data Output Bus—DO(A/B[#:0])

The data out bus reflects the contents of the memory cells referenced by the address bus at the last active clock edge. During a write operation, the data out bus reflects the data in bus. The width of this bus equals the width of the port. The allowed widths appear in [Table 1](#).

## Inverting Control Pins

The four control pins (CLK, EN, WE and RST) for each port have independent inversion control as a configuration option. The inverters are automatically absorbed into the Block SelectRAM+ Structure.

## Address Mapping

Each port accesses the same set of 4096 memory cells using an addressing scheme dependent on the width of the port. The physical RAM locations addressed for a particular width are described in the following formula (of interest only when the two ports use different aspect ratios).

$$\text{Start} = ((\text{ADDR}_{\text{port}} + 1) * \text{Width}_{\text{port}}) - 1$$

$$\text{End} = \text{ADDR}_{\text{port}} * \text{Width}_{\text{port}}$$

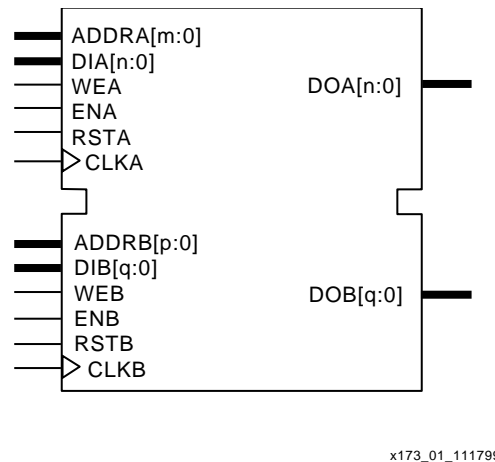
[Table 4](#) shows low order address mapping for each port width.

**Table 4: Port Address Mapping**

Port Width	Port Addresses																
1	4095...	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
2	2047...	07		06		05		04		03		02		01		00	
4	1023...	03				02				01				00			
8	511...	01								00							
16	255...	00															

## Xilinx CORE Generator System

The Xilinx CORE Generator System provides either a Single Port Block Memory or a Dual Port Block Memory (see [Figure 3](#)). Both support RAM, ROM, and Write Only functions, according to the control signals that are selected. Any size memory that can be created in the architecture is supported.



**Figure 3: Core Schematic Symbol**

Both modules are parameterizable as with most CORE Generator modules. The user can select the component name and choose to include or exclude control inputs, and choose whether to make them active High or active Low. For the Dual-Port Block Memory, once the dimensions for Port A are selected, only valid combinations for Port B will be displayed.

The user can specify the initial contents of the memory. The default will be zeroes for any addresses otherwise undefined, or the user can specify a different default value directly within the CORE Generator System. The radix can be selected to be 2, 10, or 16.

Non-default initial values can be specified in a Memory Initialization File, consisting of one line of binary data for every memory location. A default file is generated by the CORE Generator System. An alternative is the `.coe` file, which can not only define the initial contents in a radix of 2, 10, or 16, but also define all the other control parameters for the CORE Generator System.

The output of the CORE Generator System includes a report on options selected and device resources required. The number of Block RAMs required is shown. If a very deep memory is selected, some external multiplexing may be required, and this will be reported in terms of the number of slices required. In addition, the software reports the number of bits available in Block RAMs that are less than 100% utilized. For simulation purposes, the CORE Generator System creates VHDL or Verilog behavioral models.

## Conflict Resolution

The Block SelectRAM+ memory is a true dual-read/write port RAM that allows simultaneous access of the same memory cell from both ports. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window. The following lists specifics of port and memory cell write conflict resolution.

- If both ports write to the same memory cell simultaneously, violating the clock-to-clock setup requirement, consider the data stored as invalid.
- If one port attempts a read of the same memory cell the other simultaneously writes, violating the clock-to-clock setup requirement, the following occurs.
  - The write succeeds
  - The data out on the writing port accurately reflects the data written.
  - The data out on the reading port is invalid.

Conflicts do not cause any physical damage.

## Read and Write Operations

### Single Port Timing

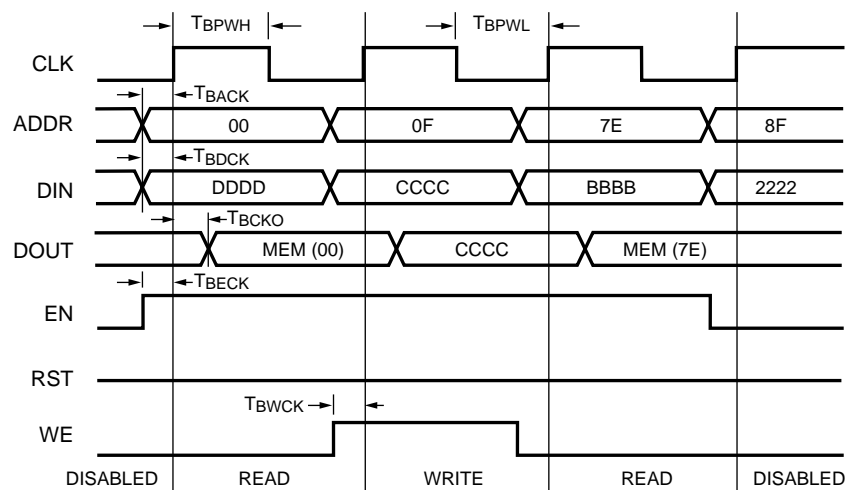
Figure 4 shows a timing diagram for a single port of a Block SelectRAM+ memory. The Block SelectRAM+ AC switching characteristics are specified in the datasheet. The Block SelectRAM+ memory is initially disabled.

At the first rising edge of the CLK pin, the ADDR, DI, EN, WE, and RST pins are sampled. The EN pin is High and the WE pin is Low indicating a read operation. The DO bus contains the contents of the memory location, 0x00, as indicated by the ADDR bus.

At the second rising edge of the CLK pin, the ADDR, DI, EN, WR, and RST pins are sampled again. The EN and WE pins are High indicating a write operation. The DO bus mirrors the DI bus. The DI bus is written to the memory location 0x0F.

At the third rising edge of the CLK pin, the ADDR, DI, EN, WR, and RST pins are sampled again. The EN pin is High and the WE pin is Low indicating a read operation. The DO bus contains the contents of the memory location 0x7E as indicated by the ADDR bus.

At the fourth rising edge of the CLK pin, the ADDR, DI, EN, WR, and RST pins are sampled again. The EN pin is Low indicating that the Block SelectRAM+ memory is now disabled. The DO bus retains the last value.



X130\_03\_091799

Figure 4: Timing Diagram for Single Port Block SelectRAM+ Memory

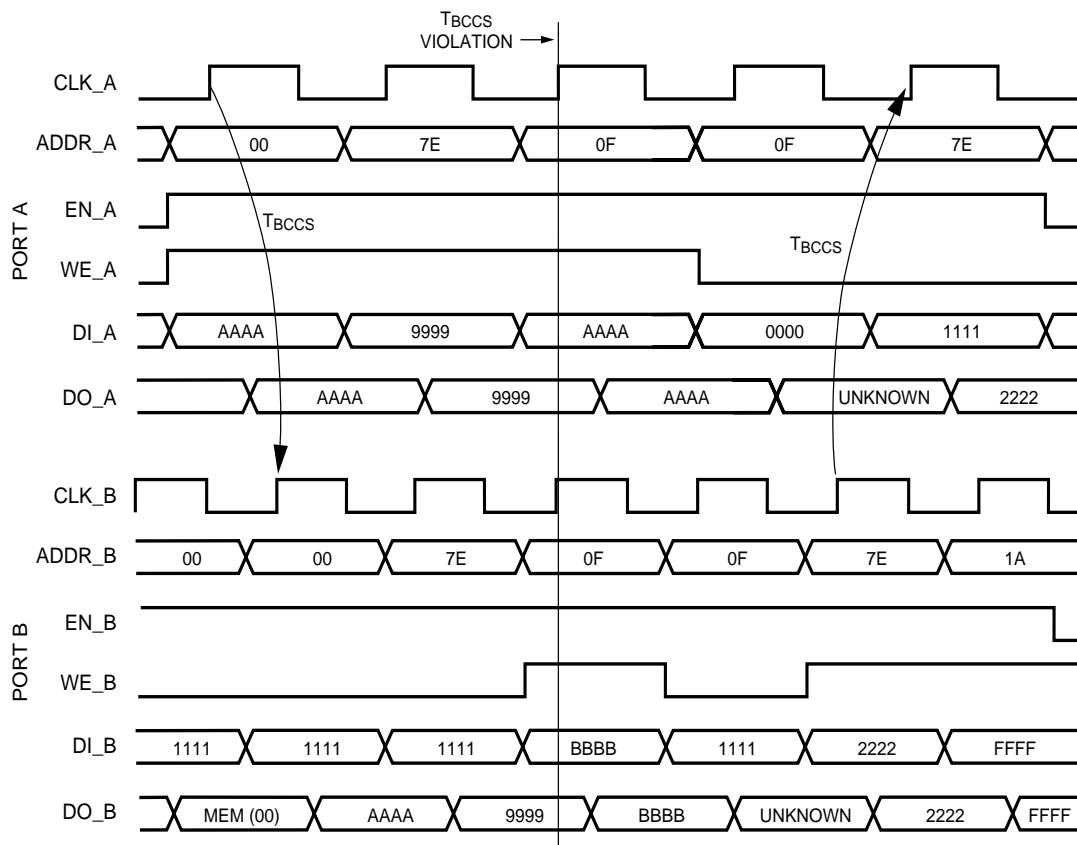
## Dual Port Timing

Figure 5 shows a timing diagram for a true dual-port read/write Block SelectRAM+ memory. The clock on port A has a longer period than the clock on Port B. The timing parameter  $T_{BCCS}$ , (clock-to-clock set-up) is shown on this diagram. The parameter  $T_{BCCS}$  is violated once in the diagram. All other timing parameters are identical to the single port version shown in Figure 4.

$T_{BCCS}$  is only of importance when the addresses of both ports are the same and at least one port is performing a write operation. When the clock-to-clock set-up parameter is violated for a WRITE-WRITE condition, the contents of the memory at that location will be invalid. When the clock-to-clock set-up parameter is violated for a WRITE-READ condition, the contents of the memory will be correct, but the read port will have invalid data. At the first rising edge of the CLKA, memory location 0x00 is to be written with the value 0xAAAA and is mirrored on the DOA bus. The last operation of Port B was a read to the same memory location 0x00. The DOB bus of Port B does not change with the new value on Port A, and retains the last read value. A short time later, Port B executes another read to memory location 0x00, and the DOB bus now reflects the new memory value written by Port A.

At the second rising edge of CLKA, memory location 0x7E is written with the value 0x9999 and is mirrored on the DOA bus. Port B then executes a read operation to the same memory location without violating the  $T_{BCCS}$  parameter and the DOB reflects the new memory values written by Port A.

At the third rising edge of CLKA, the  $T_{BCCS}$  parameter is violated with two writes to memory location 0x0F. The DOA and DOB busses reflect the contents of the DIA and DIB busses, but the stored value at 0x7E is invalid.



X130\_04\_091799

Figure 5: Timing Diagram for a True Dual-port Read/Write Block SelectRAM+ Memory



At the fourth rising edge of CLKA, a read operation is performed at memory location 0x0F and invalid data is present on the DOA bus. Port B also executes a read operation to memory location 0x0F and also reads invalid data.

At the fifth rising edge of CLKA a read operation is performed that does not violate the  $T_{BCCS}$  parameter to the previous write of 0x7E by Port B. The DOA bus reflects the recently written value by Port B.

## Initialization

The Block SelectRAM+ memory can initialize during the device configuration sequence. The 16 initialization properties of 64 hex values each (a total of 4096 bits) set the initialization of each RAM. These properties appear in [Table 5](#). Any initialization properties not explicitly set configure as zeros. Partial initialization strings pad with zeros. Initialization strings greater than 64 hex values generate an error.

The RAMs can be simulated with the initialization values using generics in VHDL simulators and parameters in Verilog simulators.

## Initialization in VHDL and Synopsys

The Block SelectRAM+ structures may be initialized in VHDL for both simulation and synthesis for inclusion in the EDIF output file. The simulation of the VHDL code uses a generic to pass the initialization. Synopsys FPGA Compiler does not presently support generics. The initialization values instead attach as attributes to the RAM by a built-in Synopsys `dc_script`. The `translate_off` statement stops synthesis translation of the generic statements. The "[VHDL Initialization Example](#)" on [page 10](#) illustrates a module that employs these techniques.

**Table 5: RAM Initialization Properties**

Property	Memory Cells
INIT_00	255 to 0
INIT_01	511 to 256
INIT_02	767 to 512
INIT_03	1023 to 768
INIT_04	1279 to 1024
INIT_05	1535 to 1280
INIT_06	1791 to 1536
INIT_07	2047 to 1792
INIT_08	2303 to 2048
INIT_09	2559 to 2304
INIT_0a	2815 to 2560
INIT_0b	3071 to 2816
INIT_0c	3327 to 3072
INIT_0d	3583 to 3328
INIT_0e	3839 to 3584
INIT_0f	4095 to 3840



```

begin
logic0 <='0';
logic1 <='1';

ram0: RAMB4_S8
--synopsys translate_off
generic map (
INIT_00 =>
X"0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF",
INIT_01 =>
X"FEDCBA9876543210FEDCBA9876543210FEDCBA9876543210FEDCBA9876543210")
--synopsys translate_on
port map (WE=>WE, EN=>logic1, RST=>logic0, CLK=>CLK, ADDR=>ADDR,
DI=>DIN, DO=>DOUT);

end BEHAVE;

```

## Initialization in Verilog and Synopsys

The Block SelectRAM+ structures may be initialized in Verilog for both simulation and synthesis for inclusion in the EDIF output file. The simulation of the Verilog code uses a defparam to pass the initialization. The Synopsys FPGA compiler does not presently support defparam. The initialization values instead attach as attributes to the RAM by a built-in Synopsys dc\_script. The translate\_off statement stops synthesis translation of the defparam statements. The following example illustrates a module that employs these techniques.

## Verilog Initialization Example

```

module MYMEM (CLK, WE, ADDR, DIN, DOUT);
input CLK, WE;
input [8:0] ADDR;
input [7:0] DIN;
output [7:0] DOUT;

wire logic0, logic1;

//synopsys dc_script_begin
//set_attribute ram0 INIT_00
"0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF"
-type string
//set_attribute ram0 INIT_01
"FEDCBA9876543210FEDCBA9876543210FEDCBA9876543210FEDCBA9876543210"
-type string
//synopsys dc_script_end

assign logic0 = 1'b0;
assign logic1 = 1'b1;

```

```

RAMB4_S8 ram0 (.WE(WE), .EN(logic1), .RST(logic0), .CLK(CLK),
.ADDR(ADDR), .DI(DIN), .DO(DOUT));

//synopsys translate_off

defparam ram0.INIT_00 =
256h' 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
;

defparam ram0.INIT_01 =
256h' FEDCBA9876543210FEDCBA9876543210FEDCBA9876543210FEDCBA9876543210
;

//synopsys translate_on

endmodule

```

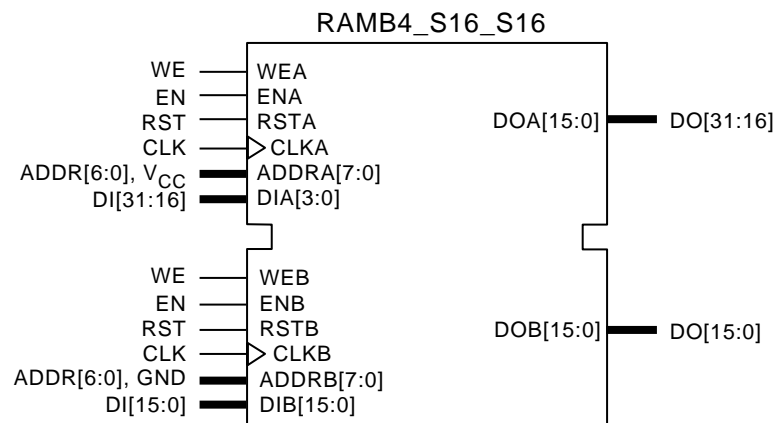
## Design Examples

Through creative use of the two sets of address lines in a dual-port RAM, non-standard memory sizes can be created. These include 128-word memories that do not waste the other half of the memory cells.

### Creating a 32-bit Single Port RAM

The true dual-read/write port functionality of the Block SelectRAM+ memory allows a single port, 128 deep by 32-bit wide RAM to be created using a single Block SelectRAM+ cell as shown in [Figure 6](#).

Interleaving the memory space, setting the LSB of the address bus of Port A to 1 ( $V_{CC}$ ), and the LSB of the address bus of Port B to 0 (GND), allows a 32-bit wide single port RAM to be created.



x130\_05\_091799

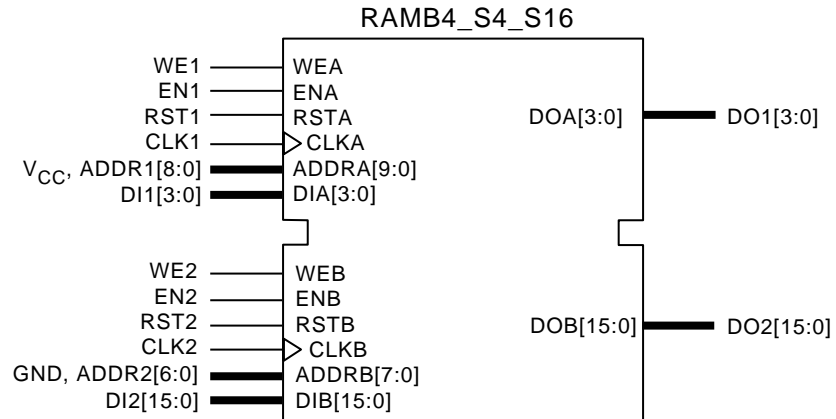
**Figure 6: Single Port 128 x 32 RAM**

In this example, a 512K x 4 RAM (Port A) and a 128 x 16 RAM (Port B) are created out of a single Block SelectRAM+ block. The address space for the RAM is split by fixing the MSB of Port A to 1 ( $V_{CC}$ ) for the upper 2K bits and the MSB of Port B to 0 (GND) for the lower 2K bits.

## Creating Two Single Port RAMs

The true dual-read/write port functionality of the Block SelectRAM+ memory allows a single RAM to be split into two single port memories of 2K bits each as shown in [Figure 7](#).

In this example, a 512K x 4 RAM (Port A) and a 128 x 16 RAM (Port B) are created out of a single Block SelectRAM+ structure. The address space for the RAM is split by fixing the MSB of Port A to 1 ( $V_{CC}$ ) for the upper 2K bits and the MSB of Port B to 0 (GND) for the lower 2K bits.



x130\_06\_092399

**Figure 7: 512 x 4 RAM and 128 x 16 RAM**

## Revision History

Date	Version #	Revision
11.23.99	1.0	Initial release.

© 1999 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners.