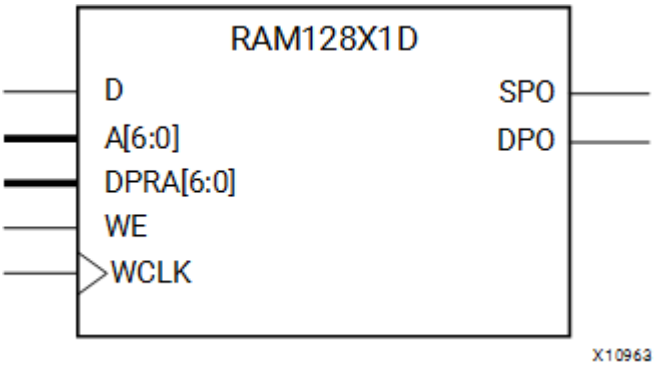
Vivado Design Suite 7 Series FPGA and Zynq 7000 SoC Libraries Guide (UG953)

RAM128X1D
RAM128X1S
RAM256X1S
RAM32M
RAM32X1D
RAM32X1S
RAM32X1S_1
RAM32X2S
RAM64M
RAM64X1D
RAM64X1S
RAM64X1S_1
RAMB18E1
RAMB36E1
ROM128X1
ROM256X1
ROM32X1
ROM64X1

RAM128X1D

Primitive: 128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)



Introduction

This design element is a 128-bit deep by 1-bit wide random access memory and has a read/write port that writes the value on the D input data pin when the write enable (WE) is high to the memory cell specified by the A address bus. This happens shortly after the rising edge of the WCLK and that same value is reflected in the data output SPO. When WE is low, an asynchronous read is initiated in which the contents of the memory cell specified by the A address bus is output asynchronously to the SPO output. The read port can perform asynchronous read access of the memory by changing the value of the address bus DPRA, and by outputting that value to the DPO data output.

Port Descriptions

Port	Direction	Width	Function
DPO	Output	1	Read/Write port data output addressed by A.
SPO	Output	1	Read port data output addressed by DPRA.
D	Input	1	Write data input addressed by A.
A	Input	7	Read/Write port address bus.

Port	Direction	Width	Function
DPRA	Input	7	Read port address bus.
WE	Input	1	Write Enable.
WCLK	Input	1	Write clock (reads are asynchronous).

If instantiated, the following connections should be made to this component:

- Tie the WCLK input to the desired clock source, the D input to the data source to be stored and the DPO output to an FDCE D input or other appropriate data destination.
- Optionally, the SPO output can also be connected to the appropriate data destination or else left unconnected.
- Connect the WE clock enable pin to the proper write enable source in the design.
- Connect the 7-bit A bus to the source for the read/write addressing and the 7-bit DPRA bus to the appropriate read address connections.

You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAM128X1D: 128-deep by 1-wide positive edge write,
asynchronous read
--            dual-port distributed LUT RAM (Mapped to two
SliceM LUT6s)
--            7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM128X1D_inst : RAM128X1D
generic map (
   INIT => X"00000000000000000000000000000000")
port map (
   DPO => DPO,     -- Read port 1-bit output
   SPO => SPO,     -- Read/Write port 1-bit output
   A => A,         -- Read/Write port 7-bit address input
   D => D,         -- RAM data input
   DPRA => DPRA,   -- Read port 7-bit address input
   WCLK => WCLK,   -- Write clock input
   WE => WE        -- RAM data input
);

-- End of RAM128X1D_inst instantiation
```

Verilog Instantiation Template

```
// RAM128X1D: 128-deep by 1-wide positive edge write,
asynchronous read  (Mapped to two SliceM LUT6s)
//            dual-port distributed LUT RAM
//            7 Series
// Xilinx HDL Language Template, version 2025.1
```

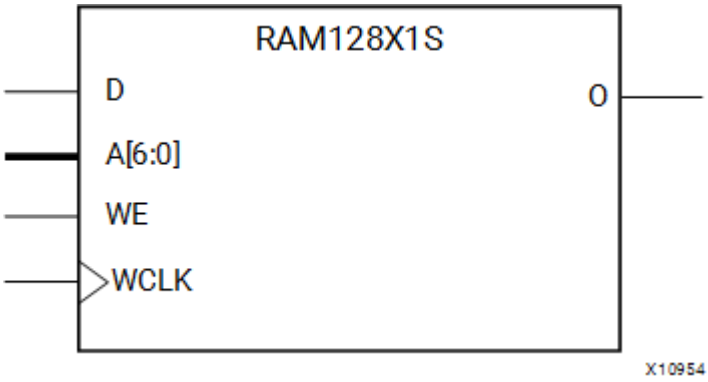
```
RAM128X1D #(
   .INIT(128'h00000000000000000000000000000000)
) RAM128X1D_inst (
   .DPO(DPO),   // Read port 1-bit output
   .SPO(SPO),   // Read/write port 1-bit output
   .A(A),       // Read/write port 7-bit address input
   .D(D),       // RAM data input
   .DPRA(DPRA), // Read port 7-bit address input
   .WCLK(WCLK), // Write clock input
   .WE(WE)      // Write enable input
);

// End of RAM128X1D_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM128X1S

Primitive: 128-Deep by 1-Wide Random Access Memory (Select RAM)



Introduction

This design element is a 128-bit deep by 1-bit wide random access memory with synchronous write and asynchronous read capability. This RAM is implemented using the LUT resources of the device (also known as Select RAM), and does not consume any of the block RAM resources of the device. If a synchronous read

capability is preferred, a register can be attached to the output and placed in the same slice as long as the same clock is used for both the RAM and the register. The RAM128X1S has an active-High write enable, WE, so that when that signal is High, and a rising edge occurs on the WCLK pin, a write is performed recording the value of the D input data pin into the memory array. The output O displays the contents of the memory cell addressed by A, regardless of the WE value. When a write is performed, the output is updated to the new value shortly after the write completes.

Port Descriptions

Port	Direction	Width	Function
O	Output	1	Read/Write port data output addressed by A.
D	Input	1	Write data input addressed by A.
A	Input	7	Read/Write port address bus.
WE	Input	1	Write Enable.
WCLK	Input	1	Write clock (reads are asynchronous).

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

If instantiated, the following connections should be made to this component:

- Tie the WCLK input to the desired clock source, the D input to the data source to be stored, and the O output to an FDCE D input or other appropriate data destination.
- Connect the WE clock enable pin to the proper write enable source in the design.
- Connect the 7-bit A bus to the source for the read/write.

You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAM128X1S: 128-deep x 1 positive edge write, asynchronous
read
--            single-port distributed RAM (Mapped to SliceM
LUT6)
--            7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM128X1S_inst : RAM128X1S
generic map (
   INIT => X"00000000000000000000000000000000")
port map (
   O => O,     -- 1-bit data output
   A0 => A0,   -- Address[0] input bit
```

```
   A1 => A1,   -- Address[1] input bit
   A2 => A2,   -- Address[2] input bit
   A3 => A3,   -- Address[3] input bit
   A4 => A4,   -- Address[4] input bit
   A5 => A5,   -- Address[5] input bit
   A6 => A6,   -- Address[6] input bit
   D => D,          -- 1-bit data input
   WCLK => WCLK,   -- Write clock input
   WE => WE        -- RAM data input
);

-- End of RAM128X1S_inst instantiation
```

Verilog Instantiation Template

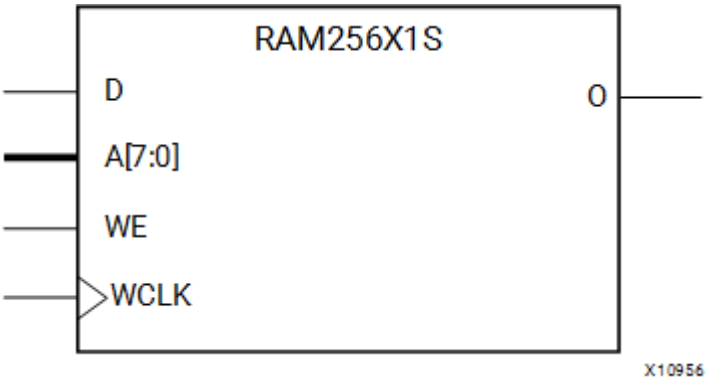
```
// RAM128X1S: 128 x 1 positive edge write, asynchronous read
single-port
//            distributed RAM (Mapped to two SliceM LUT6s)
//            7 Series
// Xilinx HDL Language Template, version 2025.1

RAM128X1S #(
   .INIT(128'h00000000000000000000000000000000) // Initial
contents of RAM
) RAM128X1S_inst (
   .O(O),        // 1-bit data output
   .A0(A0),      // Address[0] input bit
   .A1(A1),      // Address[1] input bit
   .A2(A2),      // Address[2] input bit
   .A3(A3),      // Address[3] input bit
   .A4(A4),      // Address[4] input bit
   .A5(A5),      // Address[5] input bit
   .A6(A6),      // Address[6] input bit
   .D(D),        // 1-bit data input
   .WCLK(WCLK),  // Write clock input
   .WE(WE)       // Write enable input
);

// End of RAM128X1S_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM256X1S

Primitive: 256-Deep by 1-Wide Random Access Memory (Select RAM)



Introduction

This design element is a 256-bit deep by 1-bit wide random access memory with synchronous write and asynchronous read capability. This RAM is implemented using the LUT resources of the device (also known as Select RAM), and does not consume any of the block RAM resources of the device. If a synchronous read capability is preferred, a register can be attached to the output and placed in the same slice as long as the same clock is used for both the RAM and the register. The RAM256X1S has an active-High write enable, WE, so that when that signal is High, and a rising edge occurs on the WCLK pin, a write is performed recording the value of the D input data pin into the memory array. The output O displays the contents of the memory cell addressed by A, regardless of the WE value. When a write is performed, the output is updated to the new value shortly after the write completes.

Port Descriptions

Port	Direction	Width	Function
O	Output	1	Read/Write port data output

Port	Direction	Width	Function
			addressed by A.
D	Input	1	Write data input addressed by A.
A	Input	8	Read/Write port address bus.
WE	Input	1	Write Enable.
WCLK	Input	1	Write clock (reads are asynchronous).

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

If instantiated, the following connections should be made to this component:

- Tie the WCLK input to the desired clock source, the D input to the data source to be stored, and the O output to an FDCE D input or other appropriate data destination.
- Connect the WE clock enable pin to the proper write enable source in the design.
- Connect the 8-bit A bus to the source for the read/write.

You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 256-bit value	All zeros	Specifies the initial contents of the RAM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAM256X1S: 256-deep by 1-wide positive edge write,
asynchronous read
--            single-port distributed LUT RAM (Mapped to four
SliceM LUT6s)
--            7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM256X1S_inst : RAM256X1S
generic map (
   INIT =>
X"0000000000000000000000000000000000000000000000000000000000000
00000")
port map (
   O => O,  -- Read/Write port 1-bit ouput
   A => A,  -- Read/Write port 8-bit address input
   D => D,  -- RAM data input
   WCLK => WCLK,  -- Write clock input
   WE => WE -- Write enable input
);

-- End of RAM256X1S_inst instantiation
```

Verilog Instantiation Template

```
// RAM256X1S: 256-deep by 1-wide positive edge write,
asynchronous read  (Mapped to four SliceM LUT6s)
//            single-port distributed LUT RAM
//            7 Series
// Xilinx HDL Language Template, version 2025.1
```

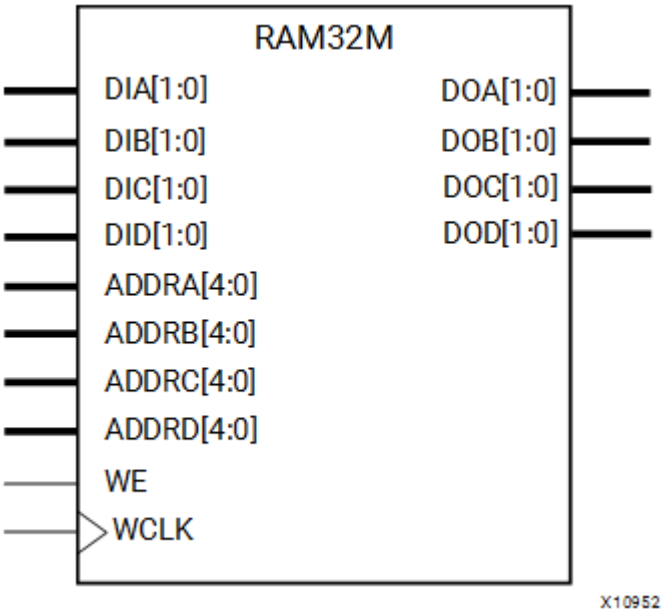
```
RAM256X1S #(

.INIT(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAM256X1S_inst (
   .O(O),       // Read/write port 1-bit output
   .A(A),       // Read/write port 8-bit address input
   .WE(WE),     // Write enable input
   .WCLK(WCLK), // Write clock input
   .D(D)        // RAM data input
);

// End of RAM256X1S_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM32M

Primitive: 32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)



Introduction

This design element is a 32-bit deep by 8-bit wide, multi-port, random access memory with synchronous write and asynchronous independent, 2-bit, wide-read capability. This RAM is implemented using the LUT resources of the device known as SelectRAM™+, and does not consume any of the Block RAM resources of the device. The RAM32M is implemented in a single slice and consists of one 8-bit write, 2-bit read port and three separate 2-bit read ports from the same memory, which allows for byte-wide write and independent 2-bit read access RAM.

- If the DIA, DIB, DIC, and DID inputs are all tied to the same data inputs, the RAM can become a 1 read/write port, 3 independent read port, 32x2 quad port memory.
- If DID is grounded, DOD is not used.
- If ADDRA, ADDRB, and ADDRC are tied to the same address, the RAM becomes a 32x6 simple dual port RAM.
- If ADDRD is tied to ADDRA, ADDRB, and ADDRC, then the RAM is a 32x8 single port RAM.

There are several other possible configurations for this RAM.

Port Descriptions

Port	Direction	Width	Function
DOA	Output	2	Read port data outputs addressed by ADDRA.
DOB	Output	2	Read port data outputs addressed by ADDRB.
DOC	Output	2	Read port data outputs addressed by ADDRC.
DOD	Output	2	Read/Write port data outputs addressed by ADDRD.
DIA	Input	2	Write data inputs addressed by ADDRD (read output is addressed by ADDRA).
DIB	Input	2	Write data inputs addressed by ADDRD (read output is addressed by

Port	Direction	Width	Function
			ADDRB).
DIC	Input	2	Write data inputs addressed by ADDRD (read output is addressed by ADDRC).
DID	Input	2	Write data inputs addressed by ADDRD.
ADDRA	Input	5	Read address bus A.
ADDRB	Input	5	Read address bus B.
ADDRC	Input	5	Read address bus C.
ADDRD	Input	5	8-bit data write port, 2-bit data read port address bus D.
WE	Input	1	Write Enable.
WCLK	Input	1	Write clock (reads are asynchronous).

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

This element can be inferred by some synthesis tools by describing a RAM with a synchronous write and asynchronous read capability. Consult your synthesis tool documentation for details on RAM inference capabilities and coding examples. You should instantiate this component if you have a need to implicitly specify the RAM function, or if you need to manually place or relationally place the component. If a synchronous read capability is desired, the outputs can be connected to an FDRSE (FDCPE if asynchronous reset is needed) in order to improve the output timing of the function. However, this is not necessary for the proper operation of the RAM. If

you want to have the data clocked on the negative edge of a clock, an inverter can be described on the clock input to this component. This inverter will be absorbed into the block giving the ability to write to the RAM on falling clock edges.
If instantiated, the following connections should be made to this component:

- Connect the WCLK input to the desired clock source
- Connect the DIA, DIB, DIC, and DID inputs to the data source to be stored
- Connect the DOA, DOB, DOC, and DOD outputs to an FDCE D input or other appropriate data destination, or leave unconnected if not used
- Connect the WE clock enable pin to the proper write enable source in the design
- Connect the ADDRD bus to the source for the read/write addressing
- Connect the ADDRA, ADDRB, and ADDRC buses to the appropriate read address connections

The optional INIT_A, INIT_B, INIT_C and INIT_D attributes let you specify the initial memory contents of each port using a 64-bit hexadecimal value. The INIT value correlates to the RAM addressing by the following equation: ADDRy[z] = INIT_y[2*z+1:2*z]. For instance, if the RAM ADDRC port is addressed to 00001, then the INIT_C[3:2] values would be the initial values shown on the DOC port before the first write occurs at that address. If left unspecified, the initial contents will be all zeros.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_A	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port A.
INIT_B	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port B.
INIT_C	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port C.
INIT_D	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port D.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAM32M: 32-deep by 8-wide Multi Port LUT RAM (Mapped to
four SliceM LUT6s)
--         7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM32M_inst : RAM32M
generic map (
   INIT_A => X"0000000000000000",   -- Initial contents of A
port
   INIT_B => X"0000000000000000",   -- Initial contents of B
port
   INIT_C => X"0000000000000000",   -- Initial contents of C
port
   INIT_D => X"0000000000000000")   -- Initial contents of D
port
port map (
   DOA => DOA, -- Read port A 2-bit output
   DOB => DOB, -- Read port B 2-bit output
   DOC => DOC, -- Read port C 2-bit output
   DOD => DOD, -- Read/Write port D 2-bit output
   ADDRA => ADDRA,   -- Read port A 5-bit address input
   ADDRB => ADDRB,   -- Read port B 5-bit address input
   ADDRC => ADDRC,   -- Read port C 5-bit address input
   ADDRD => ADDRD,   -- Read/Write port D 5-bit address input
   DIA => DIA, -- RAM 2-bit data write input addressed by
ADDRD,
               -- read addressed by ADDRA
   DIB => DIB, -- RAM 2-bit data write input addressed by
ADDRD,
               -- read addressed by ADDRB
   DIC => DIC, -- RAM 2-bit data write input addressed by
ADDRD,
```

```
                -- read addressed by ADDRC
   DID => DID, -- RAM 2-bit data write input addressed by
ADDRD,
                -- read addressed by ADDRD
   WCLK => WCLK,  -- Write clock input
   WE => WE       -- Write enable input
);
-- End of RAM32M_inst instantiation
```

Verilog Instantiation Template

```
// RAM32M: 32-deep by 8-wide Multi Port LUT RAM (Mapped to
four SliceM LUT6s)
//         7 Series
// Xilinx HDL Language Template, version 2025.1

RAM32M #(
   .INIT_A(64'h0000000000000000), // Initial contents of A
Port
   .INIT_B(64'h0000000000000000), // Initial contents of B
Port
   .INIT_C(64'h0000000000000000), // Initial contents of C
Port
   .INIT_D(64'h0000000000000000)  // Initial contents of D
Port
) RAM32M_inst (
   .DOA(DOA),     // Read port A 2-bit output
   .DOB(DOB),     // Read port B 2-bit output
   .DOC(DOC),     // Read port C 2-bit output
   .DOD(DOD),     // Read/write port D 2-bit output
   .ADDRA(ADDRA), // Read port A 5-bit address input
   .ADDRB(ADDRB), // Read port B 5-bit address input
   .ADDRC(ADDRC), // Read port C 5-bit address input
   .ADDRD(ADDRD), // Read/write port D 5-bit address input
   .DIA(DIA),     // RAM 2-bit data write input addressed by
ADDRD,
                  //   read addressed by ADDRA
   .DIB(DIB),     // RAM 2-bit data write input addressed by
ADDRD,
```

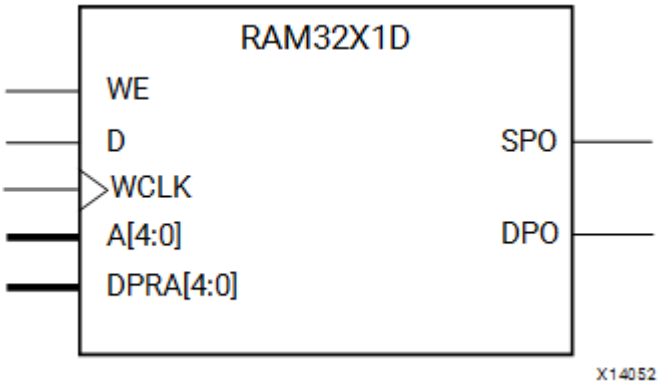
```
                    //   read addressed by ADDRB
   .DIC(DIC),       // RAM 2-bit data write input addressed by
ADDRD,
                    //   read addressed by ADDRC
   .DID(DID),       // RAM 2-bit data write input addressed by
ADDRD,
                    //   read addressed by ADDRD
   .WCLK(WCLK),     // Write clock input
   .WE(WE)          // Write enable input
);

// End of RAM32M_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM32X1D

Primitive: 32-Deep by 1-Wide Static Dual Port Synchronous RAM



Introduction

This design element is a 32-bit deep by 1-bit wide static dual port random access memory with synchronous write capability. The device has two separate address ports: the read address (DPRA4:DPRA0) and the write address (A4:A0). These two address ports are completely asynchronous. The read address controls the location of the data driven out of the output pin (DPO), and the write address controls the destination of a valid write transaction. When the write enable (WE) is Low,

transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is High, any positive transition on WCLK loads the data on the data input (D) into the memory cell selected by the 5-bit write address. For predictable performance, write address and data inputs must be stable before a Low-to-High WCLK transition. This RAM block assumes an active-High WCLK. WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block. You can initialize RAM32X1D during configuration using the INIT attribute. Mode selection is shown in the following logic table.
The SPO output reflects the data in the memory cell addressed by A4:A0. The DPO output reflects the data in the memory cell addressed by DPRA4:DPRA0. The write process is not affected by the address on the read address port.
You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

Logic Table

Inputs			**Outputs**	
WE (Mode)	**WCLK**	**D**	**SPO**	**DPO**
0 (read)	X	X	data_a	data_d
1 (read)	0	X	data_a	data_d
1 (read)	1	X	data_a	data_d
1 (write)	↑	D	D	data_d
1 (read)	↓	X	data_a	data_d

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 32-bit value	All zeros	Specifies the initial contents of the RAM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAM32X1D: 32 x 1 positive edge write, asynchronous read
--              dual-port distributed RAM (Mapped to SliceM
LUT6)
--              7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM32X1D_inst : RAM32X1D
generic map (
   INIT => X"00000000") -- Initial contents of RAM
port map (
   DPO => DPO,     -- Read-only 1-bit data output
   SPO => SPO,     -- R/W 1-bit data output
   A0 => A0,       -- R/W address[0] input bit
   A1 => A1,       -- R/W address[1] input bit
   A2 => A2,       -- R/W address[2] input bit
   A3 => A3,       -- R/W address[3] input bit
   A4 => A4,       -- R/W address[4] input bit
   D => D,         -- Write 1-bit data input
   DPRA0 => DPRA0, -- Read-only address[0] input bit
   DPRA1 => DPRA1, -- Read-only address[1] input bit
   DPRA2 => DPRA2, -- Read-only address[2] input bit
   DPRA3 => DPRA3, -- Read-only address[3] input bit
   DPRA4 => DPRA4, -- Read-only address[4] input bit
```

```
   WCLK => WCLK,   -- Write clock input
   WE => WE        -- Write enable input
);

-- End of RAM32X1D_inst instantiation
```

Verilog Instantiation Template

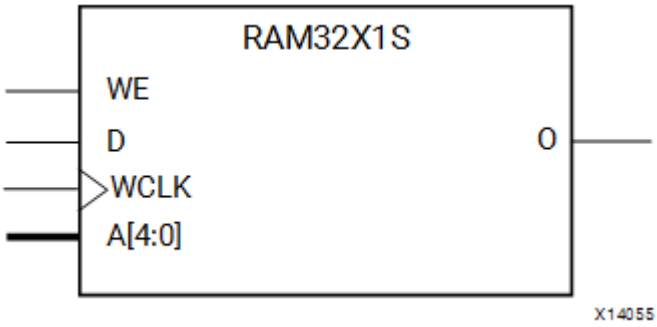
```
// RAM32X1D: 32 x 1 positive edge write, asynchronous read
dual-port
//           distributed RAM (Mapped to a SliceM LUT6)
//           7 Series
// Xilinx HDL Language Template, version 2025.1

RAM32X1D #(
   .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1D_inst (
   .DPO(DPO),     // Read-only 1-bit data output
   .SPO(SPO),     // Rw/ 1-bit data output
   .A0(A0),       // Rw/ address[0] input bit
   .A1(A1),       // Rw/ address[1] input bit
   .A2(A2),       // Rw/ address[2] input bit
   .A3(A3),       // Rw/ address[3] input bit
   .A4(A4),       // Rw/ address[4] input bit
   .D(D),         // Write 1-bit data input
   .DPRA0(DPRA0), // Read-only address[0] input bit
   .DPRA1(DPRA1), // Read-only address[1] input bit
   .DPRA2(DPRA2), // Read-only address[2] input bit
   .DPRA3(DPRA3), // Read-only address[3] input bit
   .DPRA4(DPRA4), // Read-only address[4] input bit
   .WCLK(WCLK),   // Write clock input
   .WE(WE)        // Write enable input
);

// End of RAM32X1D_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM32X1S

Primitive: 32-Deep by 1-Wide Static Synchronous RAM



Introduction

This design element is a 32-bit deep by 1-bit wide static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any positive transition on (WCLK) loads the data on the data input (D) into the memory cell selected by the 5-bit address (A4-A0). For predictable performance, address and data inputs must be stable before a Low-to-High (WCLK) transition. This RAM block assumes an active-High (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.
The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins.
You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

Logic Table

Inputs			Outputs
WE (Mode)	**WCLK**	**D**	**O**
0 (read)	X	X	Data

Inputs			Outputs
WE (Mode)	**WCLK**	**D**	**O**
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D	D
1 (read)	↓	X	Data

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 32-bit value	All zeros	Specifies initial contents of the RAM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;


-- RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
```

```
(Mapped to SliceM LUT6)
--           7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM32X1S_inst : RAM32X1S
generic map (
   INIT => X"00000000")
port map (
   O => O,       -- RAM output
   A0 => A0,     -- RAM address[0] input
   A1 => A1,     -- RAM address[1] input
   A2 => A2,     -- RAM address[2] input
   A3 => A3,     -- RAM address[3] input
   A4 => A4,     -- RAM address[4] input
   D => D,       -- RAM data input
   WCLK => WCLK, -- Write clock input
   WE => WE      -- Write enable input
);

-- End of RAM32X1S_inst instantiation
```

Verilog Instantiation Template

```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
(Mapped to a SliceM LUT6)
//           7 Series
// Xilinx HDL Language Template, version 2025.1

RAM32X1S #(
   .INIT(32'h00000000)  // Initial contents of RAM
) RAM32X1S_inst (
   .O(O),       // RAM output
   .A0(A0),     // RAM address[0] input
   .A1(A1),     // RAM address[1] input
   .A2(A2),     // RAM address[2] input
   .A3(A3),     // RAM address[3] input
   .A4(A4),     // RAM address[4] input
   .D(D),       // RAM data input
   .WCLK(WCLK), // Write clock input
```

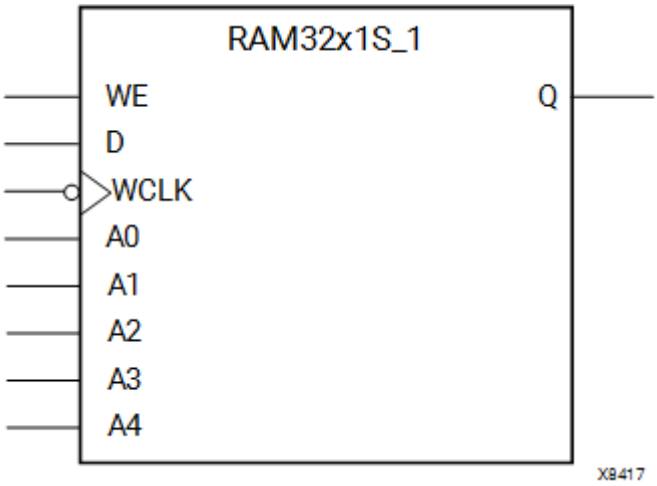
```
   .WE(WE)      // Write enable input
);

// End of RAM32X1S_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM32X1S_1

Primitive: 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



Introduction

This design element is a 32-bit deep by 1-bit wide static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the memory cell selected by the 5-bit address (A4:A0). For predictable performance, address and data inputs must be stable before a High-to-Low (WCLK) transition. This RAM block assumes an active-Low (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can use the INIT attribute to specify the initial contents of the RAM. If left

unspecified, the initial contents default to all zeros.

Logic Table

Inputs			Outputs
WE (Mode)	**WCLK**	**D**	**O**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↓	D	D
1 (read)	↑	X	Data
Data = memory cell addressed by bits A4:A0			

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 32-bit value	All zeros	Specifies the initial contents of the RAM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
(Mapped to SliceM LUT6)
--             7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM32X1S_1_inst : RAM32X1S_1
generic map (
   INIT => X"00000000")
port map (
   O => O,       -- RAM output
   A0 => A0,     -- RAM address[0] input
   A1 => A1,     -- RAM address[1] input
   A2 => A2,     -- RAM address[2] input
   A3 => A3,     -- RAM address[3] input
   A4 => A4,     -- RAM address[4] input
   D => D,       -- RAM data input
   WCLK => WCLK, -- Write clock input
   WE => WE      -- Write enable input
);

-- End of RAM32X1S_1_inst instantiation
```

Verilog Instantiation Template

```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
(Mapped to a SliceM LUT6)
//             7 Series
// Xilinx HDL Language Template, version 2025.1

RAM32X1S_1 #(
   .INIT(32'h00000000)  // Initial contents of RAM
)RAM32X1S_1_inst (
```

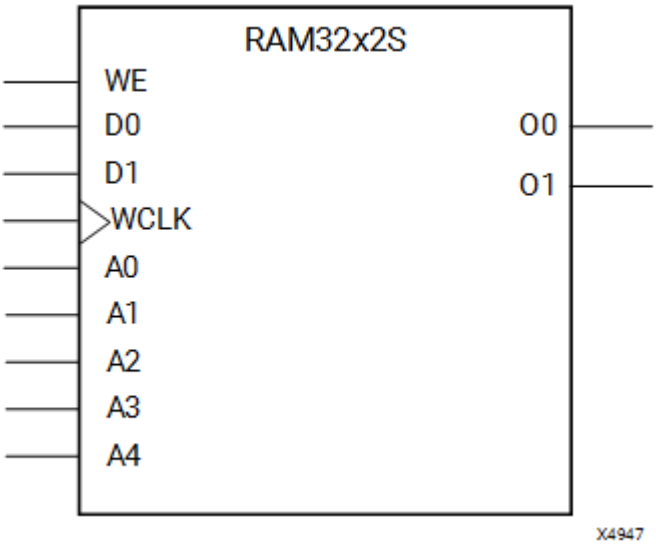
```
    .O(O),       // RAM output
    .A0(A0),     // RAM address[0] input
    .A1(A1),     // RAM address[1] input
    .A2(A2),     // RAM address[2] input
    .A3(A3),     // RAM address[3] input
    .A4(A4),     // RAM address[4] input
    .D(D),       // RAM data input
    .WCLK(WCLK), // Write clock input
    .WE(WE)      // Write enable input
);

// End of RAM32X1S_1_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM32X2S

Primitive: 32-Deep by 2-Wide Static Synchronous RAM



Introduction

This design element is a 32-bit deep by 2-bit wide static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected.

When (WE) is High, any positive transition on (WCLK) loads the data on the data input (D1-D0) into the word selected by the 5-bit address (A4-A0). For predictable performance, address and data inputs must be stable before a Low-to-High (WCLK) transition. This RAM block assumes an active-High (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block. The signal output on the data output pins (O1-O0) is the data that is stored in the RAM at the location defined by the values on the address pins.
You can use the INIT_00 and INIT_01 properties to specify the initial contents of RAM32X2S.

Logic Table

Inputs			**Outputs**
WE (Mode)	**WCLK**	**D**	**O0-O1**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D1:D0	D1:D0
1 (read)	↓	X	Data
Data = word addressed by bits A4:A0			

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Descriptions
INIT_00	HEX	Any 32-bit value	All zeros	INIT for bit 0 of RAM.
INIT_01	HEX	Any 32-bit value	All zeros	INIT for bit 1 of RAM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
(Mapped to SliceM LUT6)
--           7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM32X2S_inst : RAM32X2S
generic map (
   INIT_00 => X"00000000", --  INIT for bit 0 of RAM
   INIT_01 => X"00000000") --  INIT for bit 1 of RAM
port map (
   O0 => O0,     -- RAM data[0] output
   O1 => O1,     -- RAM data[1] output
   A0 => A0,     -- RAM address[0] input
   A1 => A1,     -- RAM address[1] input
   A2 => A2,     -- RAM address[2] input
   A3 => A3,     -- RAM address[3] input
   A4 => A4,     -- RAM address[4] input
   D0 => D0,     -- RAM data[0] input
   D1 => D1,     -- RAM data[1] input
   WCLK => WCLK, -- Write clock input
   WE => WE      -- Write enable input
);
```

```
-- End of RAM32X2S_inst instantiation
```

Verilog Instantiation Template

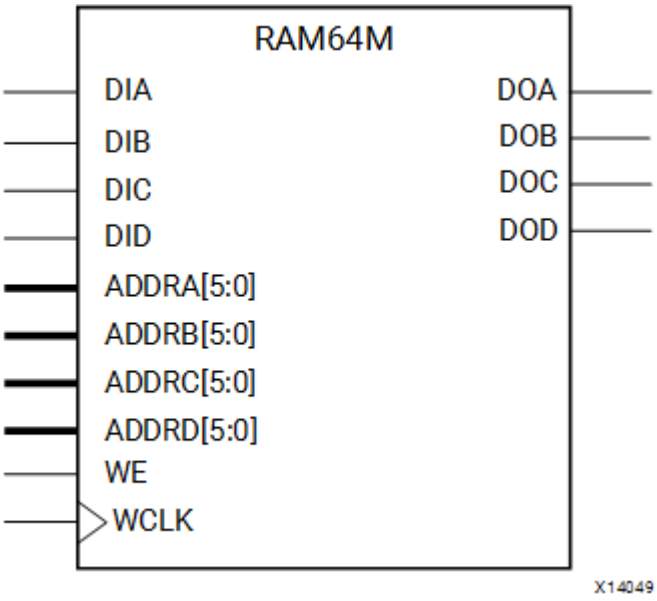
```
// RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
(Mapped to a SliceM LUT6)
//           7 Series
// Xilinx HDL Language Template, version 2025.1

RAM32X2S #(
   .INIT_00(32'h00000000), // INIT for bit 0 of RAM
   .INIT_01(32'h00000000)  // INIT for bit 1 of RAM
) RAM32X2S_inst (
   .O0(O0),     // RAM data[0] output
   .O1(O1),     // RAM data[1] output
   .A0(A0),     // RAM address[0] input
   .A1(A1),     // RAM address[1] input
   .A2(A2),     // RAM address[2] input
   .A3(A3),     // RAM address[3] input
   .A4(A4),     // RAM address[4] input
   .D0(D0),     // RAM data[0] input
   .D1(D1),     // RAM data[1] input
   .WCLK(WCLK), // Write clock input
   .WE(WE)      // Write enable input
);

// End of RAM32X2S_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM64M

Primitive: 64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)



Introduction

This design element is a 64-bit deep by 4-bit wide, multi-port, random access memory with synchronous write and asynchronous independent bit wide read capability. This RAM is implemented using the LUT resources of the device (also known as SelectRAM™+) and does not consume any of the block RAM resources of the device. The RAM64M component is implemented in a single slice, and consists of one 4-bit write, 1-bit read port, and three separate 1-bit read ports from the same memory allowing for 4-bit write and independent bit read access RAM.

- If the DIA, DIB, DIC, and DID inputs are all tied to the same data inputs, the RAM can become a 1 read/write port, 3 independent read port 64x1 quad port memory.
- If DID is grounded, DOD is not used.
- If ADDRA, ADDRB, and ADDRC are tied to the same address, the RAM becomes a 64x3 simple dual port RAM.
- If ADDRD is tied to ADDRA, ADDRB, and ADDRC, the RAM is a 64x4 single port RAM.

There are several other possible configurations for this RAM.

Port Descriptions

Port	Direction	Width	Function

Port	Direction	Width	Function
DOA	Output	1	Read port data outputs addressed by ADDRA.
DOB	Output	1	Read port data outputs addressed by ADDRB.
DOC	Output	1	Read port data outputs addressed by ADDRC.
DOD	Output	1	Read/Write port data outputs addressed by ADDRD.
DIA	Input	1	Write data inputs addressed by ADDRD (read output is addressed by ADDRA).
DIB	Input	1	Write data inputs addressed by ADDRD (read output is addressed by ADDRB).
DIC	Input	1	Write data inputs addressed by ADDRD (read output is addressed by ADDRC).
DID	Input	1	Write data inputs addressed by ADDRD.
ADDRA	Input	6	Read address bus A.
ADDRB	Input	6	Read address bus B.
ADDRC	Input	6	Read address bus C.
ADDRD	Input	6	4-bit data write port, 1-bit data read port address bus D.
WE	Input	1	Write Enable.
WCLK	Input	1	Write clock (reads are asynchronous).

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

This element can be inferred by some synthesis tools by describing a RAM with a synchronous write and asynchronous read capability. Consult your synthesis tool documentation for details on RAM inference capabilities and coding examples. Xilinx suggests that you instantiate this component if you have a need to implicitly specify the RAM function, or if you need to manually place or relationally place the component. If a synchronous read capability is desired, the outputs can be connected to an FDRE (FDCE if asynchronous reset is needed) to improve the output timing of the function. However, this is not necessary for the proper operation of the RAM. If you want to have the data clocked on the negative edge of a clock, an inverter can be described on the clock input to this component. This inverter will be absorbed into the block giving the ability to write to the RAM on falling clock edges.
If instantiated, the following connections should be made to this component:

- Connect the WCLK input to the desired clock source, the DIA, DIB, DIC
- Connect the DIA, DIB, DIC, and DID inputs to the data source to be stored
- Connect the DOA, DOB, DOC, and DOD outputs to an FDCE D input or other appropriate data destination, or leave unconnected if not used
- Connect the WE clock enable pin to the proper write enable source in the design
- Connect the ADDRD bus to the source for the read/write addressing
- Connect the ADDRA, ADDRB, and ADDRC buses to the appropriate read address connections

The optional INIT_A, INIT_B, INIT_C and INIT_D attributes let you specify the initial memory contents of each port using a 64-bit hexadecimal value. The INIT value correlates to the RAM addressing by the following equation: ADDRy[z] = INIT_y[z]. For instance, if the RAM ADDRC port is addressed to 00001, then the INIT_C[1] values would be the initial values shown on the DOC port before the first write occurs at that address. If left unspecified, the initial contents will default to all zeros.

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_A	HEX	Any 64-bit value	All zero	Specifies the initial contents of the RAM on port A.
INIT_B	HEX	Any 64-bit value	All zero	Specifies the initial contents of the RAM on port B.
INIT_C	HEX	Any 64-bit value	All zero	Specifies the initial contents of the RAM on port C.
INIT_D	HEX	Any 64-bit value	All zero	Specifies the initial contents of the RAM on port D.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAM64M: 64-deep by 4-wide Multi Port LUT RAM (Mapped to
four SliceM LUT6s)
--         7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM64M_inst : RAM64M
generic map (
   INIT_A => X"0000000000000000",   -- Initial contents of A
port
   INIT_B => X"0000000000000000",   -- Initial contents of B
port
   INIT_C => X"0000000000000000",   -- Initial contents of C
port
   INIT_D => X"0000000000000000")   -- Initial contents of D
port
```

```
port map (
   DOA => DOA, -- Read port A 1-bit output
   DOB => DOB, -- Read port B 1-bit output
   DOC => DOC, -- Read port C 1-bit output
   DOD => DOD, -- Read/Write port D 1-bit output
   ADDRA => ADDRA,   -- Read port A 6-bit address input
   ADDRB => ADDRB,   -- Read port B 6-bit address input
   ADDRC => ADDRC,   -- Read port C 6-bit address input
   ADDRD => ADDRD,   -- Read/Write port D 6-bit address input
   DIA => DIA, -- RAM 1-bit data write input addressed by
ADDRD,
               -- read addressed by ADDRA
   DIB => DIB, -- RAM 1-bit data write input addressed by
ADDRD,
               -- read addressed by ADDRB
   DIC => DIC, -- RAM 1-bit data write input addressed by
ADDRD,
               -- read addressed by ADDRC
   DID => DID, -- RAM 1-bit data write input addressed by
ADDRD,
               -- read addressed by ADDRD
   WCLK => WCLK,  -- Write clock input
   WE => WE       -- Write enable input
);
-- End of RAM64M_inst instantiation
```

Verilog Instantiation Template

```
// RAM64M: 64-deep by 4-wide Multi Port LUT RAM (Mapped to
four SliceM LUT6s)
//         7 Series
// Xilinx HDL Language Template, version 2025.1

RAM64M #(
   .INIT_A(64'h0000000000000000), // Initial contents of A
Port
   .INIT_B(64'h0000000000000000), // Initial contents of B
Port
   .INIT_C(64'h0000000000000000), // Initial contents of C
```

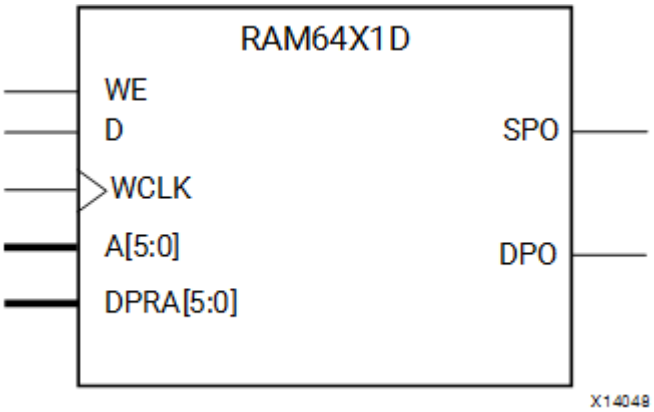
```
Port
   .INIT_D(64'h0000000000000000)  // Initial contents of D
Port
) RAM64M_inst (
   .DOA(DOA),     // Read port A 1-bit output
   .DOB(DOB),     // Read port B 1-bit output
   .DOC(DOC),     // Read port C 1-bit output
   .DOD(DOD),     // Read/write port D 1-bit output
   .DIA(DIA),     // RAM 1-bit data write input addressed by
ADDRD,
                  //   read addressed by ADDRA
   .DIB(DIB),     // RAM 1-bit data write input addressed by
ADDRD,
                  //   read addressed by ADDRB
   .DIC(DIC),     // RAM 1-bit data write input addressed by
ADDRD,
                  //   read addressed by ADDRC
   .DID(DID),     // RAM 1-bit data write input addressed by
ADDRD,
                  //   read addressed by ADDRD
   .ADDRA(ADDRA), // Read port A 6-bit address input
   .ADDRB(ADDRB), // Read port B 6-bit address input
   .ADDRC(ADDRC), // Read port C 6-bit address input
   .ADDRD(ADDRD), // Read/write port D 6-bit address input
   .WE(WE),       // Write enable input
   .WCLK(WCLK)    // Write clock input
);

// End of RAM64M_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM64X1D

Primitive: 64-Deep by 1-Wide Dual Port Static Synchronous RAM



Introduction

This design element is a 64-bit deep by 1-bit wide static dual port random access memory with synchronous write capability. The device has two separate address ports: the read address (DPRA5:DPRA0) and the write address (A5:A0). These two address ports are completely asynchronous. The read address controls the location of the data driven out of the output pin (DPO), and the write address controls the destination of a valid write transaction. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected.

When WE is High, any positive transition on WCLK loads the data on the data input (D) into the memory cell selected by the 6-bit (A0:A5) write address. For predictable performance, write address and data inputs must be stable before a Low-to-High WCLK transition. This RAM block assumes an active-High WCLK. WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The SPO output reflects the data in the memory cell addressed by A5:A0. The DPO output reflects the data in the memory cell addressed by DPRA5:DPRA0. The write process is not affected by the address on the read address port.

You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

Logic Table

Inputs			Outputs	
WE (mode)	**WCLK**	**D**	**SPO**	**DPO**
0 (read)	X	X	data_a	data_d

Inputs			Outputs	
WE (mode)	**WCLK**	**D**	**SPO**	**DPO**
1 (read)	0	X	data_a	data_d
1 (read)	1	X	data_a	data_d
1 (write)	↑	D	D	data_d
1 (read)	↓	X	data_a	data_d
data_a = memory cell addressed by bits A5:A0 data_d = memory cell addressed by bits DPRA5:DPRA0				

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;
```

```
-- RAM64X1D: 64 x 1 negative edge write, asynchronous read
--           dual-port distributed RAM (Mapped to SliceM
LUT6)
--           7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM64X1D_1_inst : RAM64X1D_1
generic map (
   INIT => X"0000000000000000") -- Initial contents of RAM
port map (
   DPO => DPO,     -- Read-only 1-bit data output
   SPO => SPO,     -- R/W 1-bit data output
   A0 => A0,       -- R/W address[0] input bit
   A1 => A1,       -- R/W address[1] input bit
   A2 => A2,       -- R/W address[2] input bit
   A3 => A3,       -- R/W address[3] input bit
   A4 => A4,       -- R/W address[4] input bit
   A5 => A5,       -- R/W address[5] input bit
   D => D,         -- Write 1-bit data input
   DPRA0 => DPRA0, -- Read-only address[0] input bit
   DPRA1 => DPRA1, -- Read-only address[1] input bit
   DPRA2 => DPRA2, -- Read-only address[2] input bit
   DPRA3 => DPRA3, -- Read-only address[3] input bit
   DPRA4 => DPRA4, -- Read-only address[4] input bit
   DPRA5 => DPRA5, -- Read-only address[5] input bit
   WCLK => WCLK,   -- Write clock input
   WE => WE        -- Write enable input
);

-- End of RAM64X1D_1_inst instantiation
```

Verilog Instantiation Template

```
// RAM64X1D: 64 x 1 positive edge write, asynchronous read
dual-port
//           distributed RAM (Mapped to a SliceM LUT6)
//           7 Series
// Xilinx HDL Language Template, version 2025.1
```

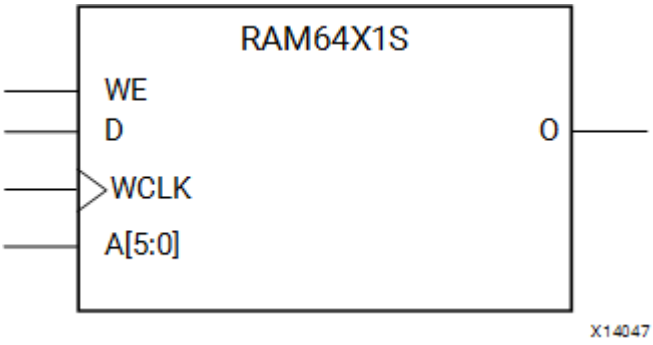
```
RAM64X1D #(
   .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1D_inst (
   .DPO(DPO),     // Read-only 1-bit data output
   .SPO(SPO),     // Rw/ 1-bit data output
   .A0(A0),       // Rw/ address[0] input bit
   .A1(A1),       // Rw/ address[1] input bit
   .A2(A2),       // Rw/ address[2] input bit
   .A3(A3),       // Rw/ address[3] input bit
   .A4(A4),       // Rw/ address[4] input bit
   .A5(A5),       // Rw/ address[5] input bit
   .D(D),         // Write 1-bit data input
   .DPRA0(DPRA0), // Read-only address[0] input bit
   .DPRA1(DPRA1), // Read-only address[1] input bit
   .DPRA2(DPRA2), // Read-only address[2] input bit
   .DPRA3(DPRA3), // Read-only address[3] input bit
   .DPRA4(DPRA4), // Read-only address[4] input bit
   .DPRA5(DPRA5), // Read-only address[5] input bit
   .WCLK(WCLK),   // Write clock input
   .WE(WE)        // Write enable input
);

// End of RAM64X1D_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM64X1S

Primitive: 64-Deep by 1-Wide Static Synchronous RAM



Introduction

This design element is a 64-bit deep by 1-bit wide static random access memory (RAM) with synchronous write capability. When the write enable is set Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is set High, any positive transition on WCLK loads the data on the data input (D) into the memory cell selected by the 6-bit address (A5:A0). This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block. The signal output on the data output pin (O) is the data that is stored in the memory cell defined by the values on the address pins.
You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

Logic Table

Mode selection is shown in the following logic table.

Inputs			**Outputs**
WE (mode)	**WCLK**	**D**	**O**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D	D
1 (read)	↓	X	Data

Inputs			Outputs
WE (mode)	**WCLK**	**D**	**O**
Data = memory cell addressed by bits A5:A0			

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Value	Default	Description
INIT	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAM64X1S: 64 x 1 positive edge write, asynchronous read
single-port distributed RAM (Mapped to SliceM LUT6)
--           7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM64X1S_inst : RAM64X1S
```

```
generic map (
   INIT => X"0000000000000000")
port map (
   O => O,         -- 1-bit data output
   A0 => A0,       -- Address[0] input bit
   A1 => A1,       -- Address[1] input bit
   A2 => A2,       -- Address[2] input bit
   A3 => A3,       -- Address[3] input bit
   A4 => A4,       -- Address[4] input bit
   A5 => A5,       -- Address[5] input bit
   D => D,         -- 1-bit data input
   WCLK => WCLK,   -- Write clock input
   WE => WE        -- Write enable input
);

-- End of RAM64X1S_inst instantiation
```

Verilog Instantiation Template

```
// RAM64X1S: 64 x 1 positive edge write, asynchronous read
single-port
//           distributed RAM (Mapped to a SliceM LUT6)
//           7 Series
// Xilinx HDL Language Template, version 2025.1

RAM64X1S #(
   .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_inst (
   .O(O),         // 1-bit data output
   .A0(A0),       // Address[0] input bit
   .A1(A1),       // Address[1] input bit
   .A2(A2),       // Address[2] input bit
   .A3(A3),       // Address[3] input bit
   .A4(A4),       // Address[4] input bit
   .A5(A5),       // Address[5] input bit
   .D(D),         // 1-bit data input
   .WCLK(WCLK),   // Write clock input
   .WE(WE)        // Write enable input
);
```

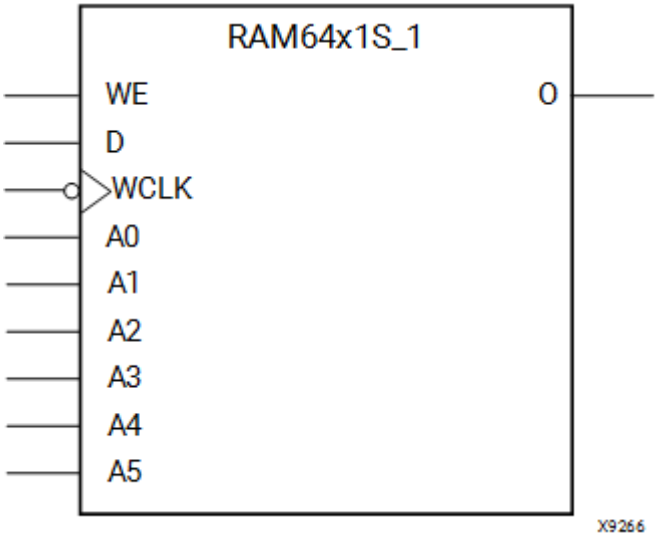
```
// End of RAM64X1S_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAM64X1S_1

Primitive: 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



Introduction

This design element is a 64-bit deep by 1-bit wide static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the memory cell selected by the 6-bit address (A5:A0). For predictable performance, address and data inputs must be stable before a High-to-Low (WCLK) transition. This RAM block assumes an active-Low (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the memory cell defined by the values on the address pins.

You can use the INIT attribute to specify the initial contents of the RAM. If left

unspecified, the initial contents default to all zeros.

Logic Table

Inputs			Outputs
WE (mode)	**WCLK**	**D**	**O**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↓	D	D
1 (read)	↑	X	Data
Data = memory cell addressed by bits A5:A0			

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAM64X1S_1: 64 x 1 negative edge write, asynchronous read
single-port distributed RAM (Mapped to SliceM LUT6)
--             7 Series
-- Xilinx HDL Language Template, version 2025.1

RAM64X1S_1_inst : RAM64X1S_1
generic map (
   INIT => X"0000000000000000")
port map (
   O => O,         -- 1-bit data output
   A0 => A0,       -- Address[0] input bit
   A1 => A1,       -- Address[1] input bit
   A2 => A2,       -- Address[2] input bit
   A3 => A3,       -- Address[3] input bit
   A4 => A4,       -- Address[4] input bit
   A5 => A5,       -- Address[5] input bit
   D => D,         -- 1-bit data input
   WCLK => WCLK,   -- Write clock input
   WE => WE        -- Write enable input
);

-- End of RAM64X1S_1_inst instantiation
```

Verilog Instantiation Template

```
// RAM64X1S_1: 64 x 1 negative edge write, asynchronous read
single-port
//             distributed RAM (Mapped to a SliceM LUT6)
//             7 Series
// Xilinx HDL Language Template, version 2025.1

RAM64X1S_1 #(
```

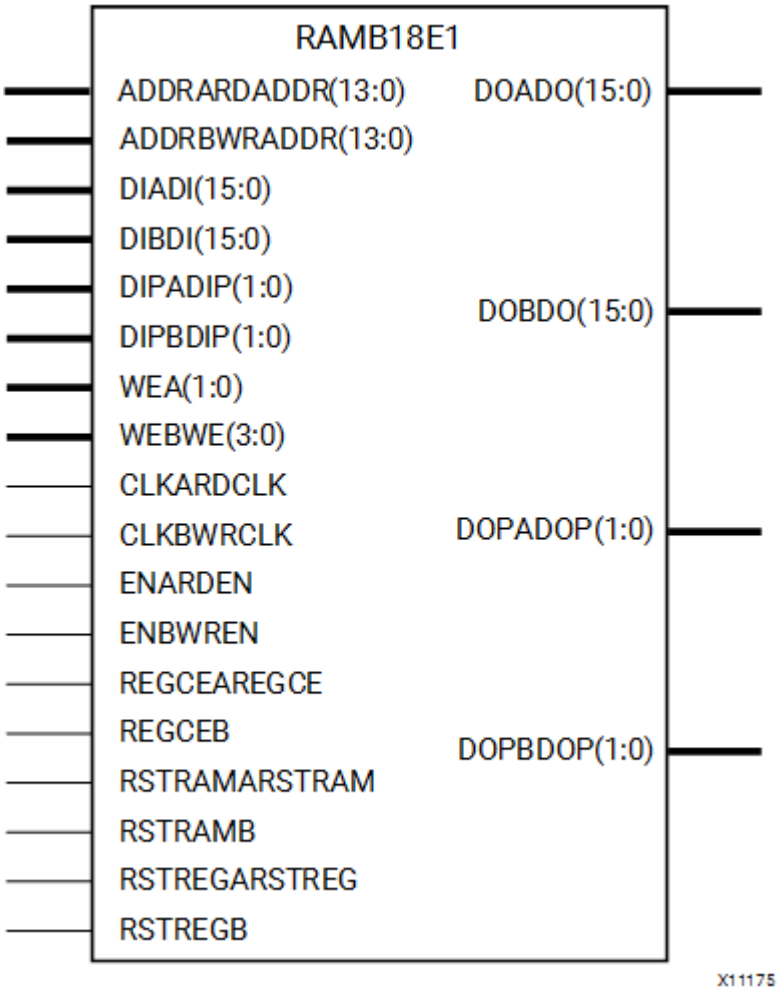
```
   .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_1_inst (
   .O(O),         // 1-bit data output
   .A0(A0),       // Address[0] input bit
   .A1(A1),       // Address[1] input bit
   .A2(A2),       // Address[2] input bit
   .A3(A3),       // Address[3] input bit
   .A4(A4),       // Address[4] input bit
   .A5(A5),       // Address[5] input bit
   .D(D),         // 1-bit data input
   .WCLK(WCLK),   // Write clock input
   .WE(WE)        // Write enable input
);

// End of RAM64X1S_1_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

RAMB18E1

Primitive: 18K-bit Configurable Synchronous Block RAM



Introduction

7 series devices contain several block RAM memories that can be configured as FIFOs, automatic error correction RAM, or general-purpose 36Kb or 18Kb RAM/ROM memories. These block RAM memories offer fast and flexible storage of large amounts of on-chip data. The RAMB18E1 allows access to the block RAM in the 18Kb configuration.

This element can be configured and used as a 1-bit wide by 16K deep to an 18-bit wide by 1024-bit deep true dual port RAM. This element can also be configured as a 36-bit wide by 512 deep simple dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component. However, the READ and WRITE ports can operate fully independent and asynchronous to each other, accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible, and an optional output register can be used to reduce the clock-to-out times of the RAM.

Port Descriptions

Port	Direction	Width	Function
ADDRARDADDR<13:0>	Input	14	Port A address input bus/Read address input bus.
ADDRBWRADDR<13:0>	Input	14	Port B address input bus/Write address input bus.
CLKARDCLK	Input	1	Rising edge port A clock input/Read clock input.
CLKBWRCLK	Input	1	Rising edge port B clock input/Write clock input.
DIADI<15:0>	Input	16	Port A data input bus/Data input bus addressed by WRADDR. When RAM_MODE="SDP", DIADI is the logical DI<15:0>.
DIBDI<15:0>	Input	16	Port B data input bus/Data input bus addressed by WRADDR. When RAM_MODE="SDP", DIBDI is the logical DI<31:16>.
DIPADIP<1:0>	Input	2	Port A parity data input bus/Data parity input bus addressed by WRADDR. When RAM_MODE="SDP", DIPADIP is the logical DIP<1:0>.
DIPBDIP<1:0>	Input	2	Port B parity data input bus/Data parity input bus addressed by WRADDR. When RAM_MODE="SDP", DIPBDIP is the logical DIP<3:2>.
DOADO<15:0>	Output	16	Port A data output bus/Data output bus addressed by RDADDR. When RAM_MODE="SDP", DOADO is the logical DO<15:0>.

Port	Direction	Width	Function
DOBDO<15:0>	Output	16	Port B data output bus/Data output bus addressed by RDADDR. When RAM_MODE="SDP", DOBDO is the logical DO<31:16>.
DOPADOP<1:0>	Output	2	Port A parity data output bus/Data parity output bus addressed by RDADDR. When RAM_MODE="SDP", DOPADOP is the logical DOP<1:0>.
DOPBDOP<1:0>	Output	2	Port B parity data output bus/Data parity output bus addressed by RDADDR. When RAM_MODE="SDP", DOPBDOP is the logical DOP<3:2>.
ENARDEN	Input	1	Port A RAM enable/Read enable.
ENBWREN	Input	1	Port B RAM enable/Write enable.
REGCEAREGCE	Input	1	Port A output register clock enable input/Output register clock enable input (valid only when DOA_REG=1).
REGCEB	Input	1	Port B output register clock enable (valid only when DOB_REG=1 and RAM_MODE="TDP").
RSTRAMARSTRAM	Input	1	Synchronous data latch set/reset to value indicated by SRVAL_A. RSTRAMARSTRAM sets/resets the BRAM data output latch when DO_REG=0 or 1. If DO_REG=1 there is a cycle of latency between the internal data latch node that is reset by RSTRAMARSTRAM and the DO output of the BRAM. This signal resets port A RAM output when

Port	Direction	Width	Function
			RAM_MODE="TDP" and the entire RAM output when RAM_MODE="SDP".
RSTRAMB	Input	1	Synchronous data latch set/reset to value indicated by SRVAL_B. RSTRAMB sets/resets the BRAM data output latch when DO_REG=0 or 1. If DO_REG=1 there is a cycle of latency between the internal data latch node that is reset by RSTRAMB and the DO output of the BRAM. Not used when RAM_MODE="SDP".
RSTREGARSTREG	Input	1	Synchronous output register set/reset to value indicated by SRVAL_A. RSTREGARSTREG sets/resets the output register when DO_REG=1. RSTREG_PRIORITY_A determines if this signal gets priority over REGCEAREGCE. This signal resets port A output when RAM_MODE="TDP" and the entire output port when RAM_MODE="SDP".
RSTREGB	Input	1	Synchronous output register set/reset to value indicated by SRVAL_B. RSTREGB sets/resets the output register when DO_REG=1. RSTREG_PRIORITY_B determines if this signal gets priority over REGCEB. Not used when RAM_MODE="SDP".
WEA<1:0>	Input	2	Port A byte-wide write enable. Not used when RAM_MODE="SDP". See User Guide for WEA mapping for

Port	Direction	Width	Function
			different port widths.
WEBWE<3:0>	Input	4	Port B byte-wide write enable/Write enable. See User Guide for WEBWE mapping for different port widths.

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	Yes
Macro support	Yes

Available Attributes

Attribute	Type	Allowed Values	Default	Description
RDADDR_COLLISION_HWCONFIG	STRING	"DELAYED_WRITE", "PERFORMANCE"	"DELAYED_WRITE"	When set to "PERFORMANCE" allows for higher clock performance (frequency) in READ_FIRST mode. If using the same clock on both ports of the RAM with "PERFORMANCE" mode, the address overlap collision rules apply where in "DELAYED_WRITE" mode, you can safely use the BRAM without incurring collisions.
SIM_COLLISION_CHECK	STRING	"ALL", "GENERATE_X_ONLY",	"ALL"	Allows modification of the simulation behavior so that if

Attribute	Type	Allowed Values	Default	Description
		"NONE", "WARNING_ONLY"		a memory collision occurs • "ALL" = warning produced and affected outputs/memory go unknown (X) • "WARNING_ONLY" = warning produced and affected outputs/ memory retain last value • "GENERATE_X_ONLY" = no warning and affected outputs/ memory go unknown (X) • "NONE" = no warning and affected outputs/ memory retain last value ✎ **Note:** Use this setting carefully. Setting it to a value other than "ALL" can mask design problems during simulation.
DOA_REG, DOB_REG	DECIMAL	0, 1	0	A value of 1 enables the output registers to the RAM enabling quicker clock-to-out from the RAM at the expense of an added clock cycle of read latency. A value of 0 allows a read in one clock cycle but will result in slower clock-to-out timing. Applies to port A/B in TDP

Attribute	Type	Allowed Values	Default	Description
				mode and up to 18 lower bits (including parity bits) in SDP mode.
INIT_A, INIT_B	HEX	18 bit HEX	18'h00000	Specifies the initial value on the port output after configuration. Applies to Port A/B in TDP mode and up to 18 lower bits (including parity bits) in SDP mode.
INIT_00 to INIT_3F	HEX	256 bit HEX	All zeros	Allows specification of the initial contents of the 16Kb data memory array.
INIT_FILE	STRING	String representing file name and location	None	File name of file used to specify initial RAM contents.
INITP_00 to INITP_07	HEX	256 bit HEX	All zeros	Allows specification of the initial contents of the 2Kb parity data memory array.
RAM_MODE	STRING	"TDP", "SDP"	"TDP"	Selects simple dual port (SDP) or true dual port (TDP) mode.
READ_WIDTH_A	DECIMAL	0, 1, 2, 4, 9, 18, 36, 72	0	Specifies the desired data width for a read on Port A, including parity bits. This value must be 0 if the Port A is not used. Otherwise, it should be set to the desired port width. In "SDP" mode, this is the read width including parity bits.
READ_WIDTH_B	DECIMAL	0, 1, 2, 4, 9,	0	Specifies the desired data

Attribute	Type	Allowed Values	Default	Description
		18		width for a read on Port B including parity bits. This value must be 0 if the Port B is not used. Otherwise, it should be set to the desired port width. Not used for "SDP" mode.
RSTREG_PRIORITY_A, RSTREG_PRIORITY_B	STRING	"RSTREG", "REGCE"	"RSTREG"	Selects register priority for RSTREG or REGCE. Applies to port A/B in TDP mode and up to 18 lower bits (including parity bits) in SDP mode.
SIM_DEVICE	STRING	"7SERIES"	"7SERIES"	Must be set to "7SERIES" in order to exhibit proper simulation behavior under all conditions.
SRVAL_A, SRVAL_B	HEX	18 bit HEX	18'h00000	Specifies the output value of the RAM upon assertion of the synchronous reset (RSTREG) signal.
WRITE_MODE_A, WRITE_MODE_B	STRING	"WRITE_FIRST", "NO_CHANGE", "READ_FIRST"	"WRITE_FIRST"	Specifies output behavior of the port being written to.

Attribute	Type	Allowed Values	Default	Description
				• "WRITE_FIRST": Written value appears on output port of the RAM • "READ_FIRST": Previous RAM contents for that memory location appear on the output port • "NO_CHANGE": Previous value on the output port remains the same. When RAM_MODE="SDP", WRITE_MODE can not be set to "NO_CHANGE". For simple dual port implementations you should set this attribute to "READ_FIRST" if using the same clock on both ports, or set it to "WRITE_FIRST" if using different clocks. This generally yields an improved collision or address overlap behavior.
WRITE_WIDTH_A	DECIMAL	0, 1, 2, 4, 9, 18	0	Specifies the desired data width for a write to Port A including parity bits. This value must be 0 if the port is not used. Otherwise should be set to the desired write width. Not used in SDP mode.

Attribute	Type	Allowed Values	Default	Description
WRITE_WIDTH_B	DECIMAL	0, 1, 2, 4, 9, 18, 36, 72	0	Specifies the desired data width for a write to Port B including parity bits. This value must be 0 if the port is not used. Otherwise should be set to the desired write width. In SDP mode, this is the write width including parity bits.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAMB18E1: 18K-bit Configurable Synchronous Block RAM
--           7 Series
-- Xilinx HDL Language Template, version 2025.1

RAMB18E1_inst : RAMB18E1
generic map (
   -- Address Collision Mode: "PERFORMANCE" or
"DELAYED_WRITE"
   RDADDR_COLLISION_HWCONFIG => "DELAYED_WRITE",
   -- Collision check: Values ("ALL", "WARNING_ONLY",
"GENERATE_X_ONLY" or "NONE")
   SIM_COLLISION_CHECK => "ALL",
   -- DOA_REG, DOB_REG: Optional output register (0 or 1)
   DOA_REG => 0,
   DOB_REG => 0,
   -- INITP_00 to INITP_07: Initial contents of parity memory
array
   INITP_00 =>
```

```
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_01 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_02 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_03 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_04 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_05 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_06 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_07 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   -- INIT_00 to INIT_3F: Initial contents of data memory
array
   INIT_00 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_01 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_02 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_03 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_04 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_05 =>
X"0000000000000000000000000000000000000000000000000000000000000000
```

```
00000",
   INIT_06 =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_07 =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_08 =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_09 =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0A =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0B =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0C =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0D =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0E =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0F =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_10 =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_11 =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_12 =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_13 =>
X"0000000000000000000000000000000000000000000000000000000000
```

```
00000",
   INIT_14 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_15 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_16 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_17 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_18 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_19 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1A =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1B =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1C =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1D =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1E =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1F =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_20 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_21 =>
X"0000000000000000000000000000000000000000000000000000000000000
```

```
00000",
   INIT_22 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_23 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_24 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_25 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_26 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_27 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_28 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_29 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_2A =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_2B =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_2C =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_2D =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_2E =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_2F =>
X"0000000000000000000000000000000000000000000000000000000000000
```

```
00000",
   INIT_30 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_31 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_32 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_33 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_34 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_35 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_36 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_37 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_38 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_39 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_3A =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_3B =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_3C =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_3D =>
X"0000000000000000000000000000000000000000000000000000000000000
```

```
00000",
   INIT_3E =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   INIT_3F =>
X"0000000000000000000000000000000000000000000000000000000000
00000",
   -- INIT_A, INIT_B: Initial values on output ports
   INIT_A => X"00000",
   INIT_B => X"00000",
   -- Initialization File: RAM initialization file
   INIT_FILE => "NONE",
   -- RAM Mode: "SDP" or "TDP"
   RAM_MODE => "TDP",
   -- READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per
port
   READ_WIDTH_A => 0,
-- 0-72
   READ_WIDTH_B => 0,
-- 0-18
   WRITE_WIDTH_A => 0,
-- 0-18
   WRITE_WIDTH_B => 0,
-- 0-72
   -- RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable
priority ("RSTREG" or "REGCE")
   RSTREG_PRIORITY_A => "RSTREG",
   RSTREG_PRIORITY_B => "RSTREG",
   -- SRVAL_A, SRVAL_B: Set/reset value for output
   SRVAL_A => X"00000",
   SRVAL_B => X"00000",
   -- Simulation Device: Must be set to "7SERIES" for
simulation behavior
   SIM_DEVICE => "7SERIES",
   -- WriteMode: Value on output upon a write ("WRITE_FIRST",
"READ_FIRST", or "NO_CHANGE")
   WRITE_MODE_A => "WRITE_FIRST",
   WRITE_MODE_B => "WRITE_FIRST"
)
port map (
   -- Port A Data: 16-bit (each) output: Port A data
   DOADO => DOADO,                 -- 16-bit output: A port
```

```
data/LSB data
   DOPADOP => DOPADOP,             -- 2-bit output: A port
parity/LSB parity
   -- Port B Data: 16-bit (each) output: Port B data
   DOBDO => DOBDO,                 -- 16-bit output: B port
data/MSB data
   DOPBDOP => DOPBDOP,             -- 2-bit output: B port
parity/MSB parity
   -- Port A Address/Control Signals: 14-bit (each) input:
Port A address and control signals (read port
   -- when RAM_MODE="SDP")
   ADDRARDADDR => ADDRARDADDR,     -- 14-bit input: A port
address/Read address
   CLKARDCLK => CLKARDCLK,         -- 1-bit input: A port
clock/Read clock
   ENARDEN => ENARDEN,             -- 1-bit input: A port
enable/Read enable
   REGCEAREGCE => REGCEAREGCE,     -- 1-bit input: A port
register enable/Register enable
   RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit input: A port
set/reset
   RSTREGARSTREG => RSTREGARSTREG, -- 1-bit input: A port
register set/reset
   WEA => WEA,                     -- 2-bit input: A port
write enable
   -- Port A Data: 16-bit (each) input: Port A data
   DIADI => DIADI,                 -- 16-bit input: A port
data/LSB data
   DIPADIP => DIPADIP,             -- 2-bit input: A port
parity/LSB parity
   -- Port B Address/Control Signals: 14-bit (each) input:
Port B address and control signals (write port
   -- when RAM_MODE="SDP")
   ADDRBWRADDR => ADDRBWRADDR,     -- 14-bit input: B port
address/Write address
   CLKBWRCLK => CLKBWRCLK,         -- 1-bit input: B port
clock/Write clock
   ENBWREN => ENBWREN,             -- 1-bit input: B port
enable/Write enable
   REGCEB => REGCEB,               -- 1-bit input: B port
register enable
   RSTRAMB => RSTRAMB,             -- 1-bit input: B port
```

```
set/reset
   RSTREGB => RSTREGB,             -- 1-bit input: B port
register set/reset
   WEBWE => WEBWE,                 -- 4-bit input: B port
write enable/Write enable
   -- Port B Data: 16-bit (each) input: Port B data
   DIBDI => DIBDI,                 -- 16-bit input: B port
data/MSB data
   DIPBDIP => DIPBDIP              -- 2-bit input: B port
parity/MSB parity
);

-- End of RAMB18E1_inst instantiation
```

Verilog Instantiation Template

```
// RAMB18E1: 18K-bit Configurable Synchronous Block RAM
//           7 Series
// Xilinx HDL Language Template, version 2025.1

RAMB18E1 #(
   // Address Collision Mode: "PERFORMANCE" or
"DELAYED_WRITE"
   .RDADDR_COLLISION_HWCONFIG("DELAYED_WRITE"),
   // Collision check: Values ("ALL", "WARNING_ONLY",
"GENERATE_X_ONLY" or "NONE")
   .SIM_COLLISION_CHECK("ALL"),
   // DOA_REG, DOB_REG: Optional output register (0 or 1)
   .DOA_REG(0),
   .DOB_REG(0),
   // INITP_00 to INITP_07: Initial contents of parity memory
array

.INITP_00(256'h0000000000000000000000000000000000000000000
0000000000000000000),

.INITP_01(256'h0000000000000000000000000000000000000000000
0000000000000000000),
```

```
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
   // INIT_00 to INIT_3F: Initial contents of data memory
array

.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
00000000000000000),
   // INIT_A, INIT_B: Initial values on output ports
   .INIT_A(18'h00000),
   .INIT_B(18'h00000),
   // Initialization File: RAM initialization file
   .INIT_FILE("NONE"),
   // RAM Mode: "SDP" or "TDP"
   .RAM_MODE("TDP"),
   // READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per
port
   .READ_WIDTH_A(0),
// 0-72
   .READ_WIDTH_B(0),
// 0-18
   .WRITE_WIDTH_A(0),
// 0-18
   .WRITE_WIDTH_B(0),
// 0-72
   // RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable
priority ("RSTREG" or "REGCE")
   .RSTREG_PRIORITY_A("RSTREG"),
   .RSTREG_PRIORITY_B("RSTREG"),
   // SRVAL_A, SRVAL_B: Set/reset value for output
   .SRVAL_A(18'h00000),
   .SRVAL_B(18'h00000),
   // Simulation Device: Must be set to "7SERIES" for
simulation behavior
   .SIM_DEVICE("7SERIES"),
   // WriteMode: Value on output upon a write ("WRITE_FIRST",
"READ_FIRST", or "NO_CHANGE")
   .WRITE_MODE_A("WRITE_FIRST"),
   .WRITE_MODE_B("WRITE_FIRST")
)
RAMB18E1_inst (
   // Port A Data: 16-bit (each) output: Port A data
   .DOADO(DOADO),                 // 16-bit output: A port
data/LSB data
   .DOPADOP(DOPADOP),             // 2-bit output: A port
parity/LSB parity
   // Port B Data: 16-bit (each) output: Port B data
   .DOBDO(DOBDO),                 // 16-bit output: B port
data/MSB data
```

```
   .DOPBDOP(DOPBDOP),            // 2-bit output: B port
parity/MSB parity
   // Port A Address/Control Signals: 14-bit (each) input:
Port A address and control signals (read port
   // when RAM_MODE="SDP")
   .ADDRARDADDR(ADDRARDADDR),    // 14-bit input: A port
address/Read address
   .CLKARDCLK(CLKARDCLK),        // 1-bit input: A port
clock/Read clock
   .ENARDEN(ENARDEN),            // 1-bit input: A port
enable/Read enable
   .REGCEAREGCE(REGCEAREGCE),    // 1-bit input: A port
register enable/Register enable
   .RSTRAMARSTRAM(RSTRAMARSTRAM), // 1-bit input: A port set/
reset
   .RSTREGARSTREG(RSTREGARSTREG), // 1-bit input: A port
register set/reset
   .WEA(WEA),                    // 2-bit input: A port
write enable
   // Port A Data: 16-bit (each) input: Port A data
   .DIADI(DIADI),                // 16-bit input: A port
data/LSB data
   .DIPADIP(DIPADIP),            // 2-bit input: A port
parity/LSB parity
   // Port B Address/Control Signals: 14-bit (each) input:
Port B address and control signals (write port
   // when RAM_MODE="SDP")
   .ADDRBWRADDR(ADDRBWRADDR),    // 14-bit input: B port
address/Write address
   .CLKBWRCLK(CLKBWRCLK),        // 1-bit input: B port
clock/Write clock
   .ENBWREN(ENBWREN),            // 1-bit input: B port
enable/Write enable
   .REGCEB(REGCEB),              // 1-bit input: B port
register enable
   .RSTRAMB(RSTRAMB),            // 1-bit input: B port set/
reset
   .RSTREGB(RSTREGB),            // 1-bit input: B port
register set/reset
   .WEBWE(WEBWE),                // 4-bit input: B port
write enable/Write enable
   // Port B Data: 16-bit (each) input: Port B data
```

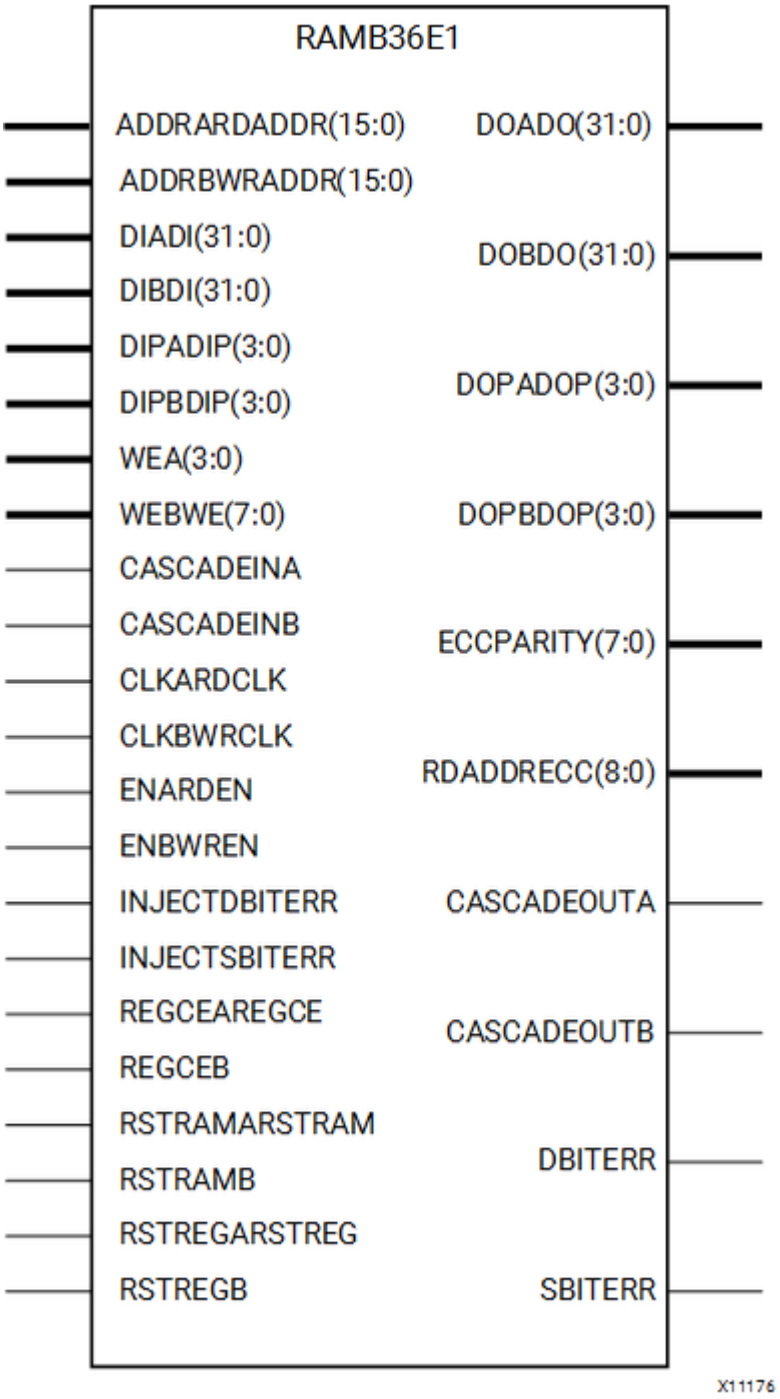
```
   .DIBDI(DIBDI),                 // 16-bit input: B port
data/MSB data
   .DIPBDIP(DIPBDIP)              // 2-bit input: B port
parity/MSB parity
);

// End of RAMB18E1_inst instantiation
```

Related Information

- *7 Series FPGAs Memory Resources User Guide* (UG473)

RAMB36E1

Primitive: 36K-bit Configurable Synchronous Block RAM



Introduction

7 series devices contain several block RAM memories that can be configured as FIFOs, automatic error correction RAM, or general-purpose 36 Kb or 18 Kb RAM/ROM memories. These block RAM memories offer fast and flexible storage of large amounts of on-chip data. The RAMB36E1 allows access to the block RAM in the 36 Kb configuration. This element can be cascaded to create a larger ram. This element can be configured and used as a 1-bit wide by 32K deep to a 36-bit wide by 1K deep true dual port RAM. This element can also be configured as a 72-bit wide by 512 deep simple dual port RAM. Both read and write operations are fully

synchronous to the supplied clock(s) to the component. However, the READ and WRITE ports can operate fully independent and asynchronous to each other, accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible, and an optional output register can be used to reduce the clock-to-out times of the RAM. Error detection and correction circuitry can also be enabled to uncover and rectify possible memory corruptions.

Port Descriptions

Port	Direction	Width	Function
ADDRARDADDR<15:0>	Input	16	Port A address input bus/Read address input bus.
ADDRBWRADDR<15:0>	Input	16	Port B address input bus/Write address input bus.
CASCADEINA	Input	1	Port A cascade input. Never use when RAM_MODE="SDP".
CASCADEINB	Input	1	Port B cascade input. Never use when RAM_MODE="SDP".
CASCADEOUTA	Output	1	Port A cascade output. Never use when RAM_MODE="SDP".
CASCADEOUTB	Output	1	Port B cascade output. Never use when RAM_MODE="SDP".
CLKARDCLK	Input	1	Rising edge port A clock input/Read clock input.
CLKBWRCLK	Input	1	Rising edge port B clock input/Write clock input.
DBITERR	Output	1	Status output from ECC function to indicate a double bit error was detected. EN_ECC_READ needs to be TRUE to use this functionality. Not used when RAM_MODE="TDP".

Port	Direction	Width	Function
DIADI<31:0>	Input	32	Port A data input bus/Data input bus addressed by WRADDR. When RAM_MODE="SDP", DIADI is the logical DI<31:0>.
DIBDI<31:0>	Input	32	Port B data input bus/Data input bus addressed by WRADDR. When RAM_MODE="SDP", DIBDI is the logical DI<63:32>.
DIPADIP<3:0>	Input	4	Port A parity data input bus/Data parity input bus addressed by WRADDR. When RAM_MODE="SDP", DIPADIP is the logical DIP<3:0>.
DIPBDIP<3:0>	Input	4	Port B parity data input bus/Data parity input bus addressed by WRADDR. When RAM_MODE="SDP", DIPBDIP is the logical DIP<7:4>.
DOADO<31:0>	Output	32	Port A data output bus/Data output bus addressed by RDADDR. When RAM_MODE="SDP", DOADO is the logical DO<31:0>.
DOBDO<31:0>	Output	32	Port B data output bus/Data output bus addressed by RDADDR. When RAM_MODE="SDP", DOBDO is the logical DO<63:32>.
DOPADOP<3:0>	Output	4	Port A parity data output bus/Data parity output bus addressed by RDADDR. When RAM_MODE="SDP", DOPADOP is the logical DOP<3:0>.

Port	Direction	Width	Function
DOPBDOP<3:0>	Output	4	Port B parity data output bus/Data parity output bus addressed by RDADDR. When RAM_MODE="SDP", DOPBDOP is the logical DOP<7:4>.
ECCPARITY<7:0>	Output	8	8-bit data generated by the ECC encoder used by the ECC decoder for memory error detection and correction. Not used if RAM_MODE="TDP".
ENARDEN	Input	1	Port A RAM enable/Read enable.
ENBWREN	Input	1	Port B RAM enable/Write enable.
INJECTDBITERR	Input	1	Inject a double bit error if ECC feature is used.
INJECTSBITERR	Input	1	Inject a single bit error if ECC feature is used.
RDADDRECC<8:0>	Output	9	ECC read address. Not used when RAM_MODE="TDP".
REGCEAREGCE	Input	1	Port A output register clock enable input/Output register clock enable input (valid only when DO_REG=1).
REGCEB	Input	1	Port B output register clock enable (valid only when DO_REG=1 and RAM_MODE="TDP").
RSTRAMARSTRAM	Input	1	Synchronous data latch set/reset to value indicated by SRVAL_A. RSTRAMARSTRAM sets/resets the BRAM data output latch when DO_REG=0 or 1. If DO_REG=1 there is a cycle of latency between the internal data latch node that is reset

Port	Direction	Width	Function
			by RSTRAMARSTRAM and the DO output of the BRAM. This signal resets port A RAM output when RAM_MODE="TDP" and the entire RAM output when RAM_MODE="SDP".
RSTRAMB	Input	1	Synchronous data latch set/reset to value indicated by SRVAL_B. RSTRAMB sets/resets the BRAM data output latch when DO_REG=0 or 1. If DO_REG=1 there is a cycle of latency between the internal data latch node that is reset by RSTRAMB and the DO output of the BRAM. Not used when RAM_MODE="SDP".
RSTREGARSTREG	Input	1	Synchronous output register set/reset to value indicated by SRVAL_A. RSTREGARSTREG sets/resets the output register when DO_REG=1. RSTREG_PRIORITY_A determines if this signal gets priority over REGCEAREGCE. This signal resets port A output when RAM_MODE="TDP" and the entire output port when RAM_MODE="SDP".
RSTREGB	Input	1	Synchronous output register set/reset to value indicated by SRVAL_B. RSTREGB sets/resets the output register when DO_REG=1. RSTREG_PRIORITY_B determines if this signal gets priority over REGCEB. Not used when RAM_MODE="SDP".

Port	Direction	Width	Function
SBITERR	Output	1	Status output from ECC function to indicate a single bit error was detected. EN_ECC_READ needs to be TRUE to use this functionality. Not used when RAM_MODE="TDP".
WEA<3:0>	Input	4	Port A byte-wide write enable. Not used when RAM_MODE="SDP". See User Guide for WEA mapping for different port widths.
WEBWE<7:0>	Input	8	Port B byte-wide write enable/Write enable. See User Guide for WEBWE mapping for different port widths.

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	Yes
Macro support	Yes

Available Attributes

Attribute	Type	Allowed Values	Default	Description
RDADDR_COLLISION_HWCONFIG	STRING	"DELAYED_WRITE", "PERFORMANCE"	"DELAYED_WRITE"	When set to "PERFORMANCE" allows for higher clock performance (frequency) in READ_FIRST mode. If using the same clock on both ports of the RAM with "PERFORMANCE" mode,

Attribute	Type	Allowed Values	Default	Description
				the address overlap collision rules apply where in "DELAYED_WRITE" mode, you can safely use the BRAM without incurring collisions.
SIM_COLLISION_CHECK	STRING	"ALL", "GENERATE_X_ONLY", "NONE", "WARNING_ONLY"	"ALL"	Allows modification of the simulation behavior so that if a memory collision occurs. • "ALL" = warning produced and affected outputs/memory go unknown (X) • "WARNING_ONLY" = warning produced and affected outputs/ memory retain last value • "GENERATE_X_ONLY" = no warning and affected outputs/ memory go unknown (X) • "NONE" = no warning and affected outputs/ memory retain last value ✎ **Note:** Use this setting carefully. Setting it to a value other than "ALL" can mask design problems during simulation.
DOA_REG, DOB_REG	DECIMAL	0, 1	0	A value of 1 enables the output registers to the RAM,

Attribute	Type	Allowed Values	Default	Description
				which gives you quicker clock-to-out from the RAM at the expense of an added clock cycle of read latency. A value of 0 allows a read-in-one clock cycle but will result in slower clock-to-out timing. The number of registers activated is the same as the port width and includes parity bits. In SDP mode, DOA_REG and DOB_REG should always be set to the same value.
EN_ECC_READ	BOOLEAN	FALSE, TRUE	FALSE	Enable the ECC decoder circuitry.
EN_ECC_WRITE	BOOLEAN	FALSE, TRUE	FALSE	Enable the ECC encoder circuitry.
INIT_A, INIT_B	HEX	36 bit HEX	All zeros	Specifies the initial value on the port output after configuration. In SDP mode, INIT_A and INIT_B should always be set to the same value.
INIT_00 to INIT_7F	HEX	256 bit HEX	All zeros	Allows specification of the initial contents of the 32 Kb data memory array.
INIT_FILE	STRING	String representing file name and location	None	File name of file used to specify initial RAM contents.
INITP_00 to INITP_0F	HEX	256 bit HEX	All zeros	Allows specification of the initial contents of the 4 Kb

Attribute	Type	Allowed Values	Default	Description
				parity data memory array.
RAM_EXTENSION_A, RAM_EXTENSION_B	STRING	"NONE", "LOWER", "UPPER"	"NONE"	Selects cascade mode. If not cascading two block RAMs to form a 64K x 1 RAM set to "NONE". If cascading RAMs, set to either "UPPER" or "LOWER" to indicate relative RAM location for proper configuration of the RAM. Not used if RAM_MODE="SDP".
RAM_MODE	STRING	"TDP", "SDP"	"TDP"	Selects simple dual port (SDP) or true dual port (TDP) mode.
READ_WIDTH_A, READ_WIDTH_B, WRITE_WIDTH_A, WRITE_WIDTH_B	DECIMAL	0, 1, 2, 4, 9, 18, 36, 72	0	Specifies the desired data width for a read/write on port A/B, including parity bits. This value must be 0 if the port is not used. Otherwise, it should be set to the desired port width.
RSTREG_PRIORITY_A, RSTREG_PRIORITY_B	STRING	"RSTREG", "REGCE"	"RSTREG"	Selects register priority for "RSTREG" or "REGCE". In SDP mode, RSTREG_PRIORITY_A and RSTREG_PRIORITY_B should always be set to the same value.
SIM_DEVICE	STRING	"7SERIES"	"7SERIES"	Must be set to "7SERIES" in order to exhibit proper simulation behavior under all conditions.

Attribute	Type	Allowed Values	Default	Description
SRVAL_A, SRVAL_B	HEX	36 bit HEX	All zeros	Specifies the output value of the RAM upon assertion of the synchronous reset (RSTREG) signal. In SDP mode, SRVAL_A and SRVAL_B should always be set to the same value.
WRITE_MODE_A, WRITE_MODE_B	STRING	"WRITE_FIRST", "NO_CHANGE", "READ_FIRST"	"WRITE_FIRST"	Specifies output behavior of the port being written to. • "WRITE_FIRST" = written value appears on output port of the RAM • "READ_FIRST" = previous RAM contents for that memory location appears on the output port • "NO_CHANGE" = previous value on the output port remains the same When RAM_MODE="SDP", WRITE_MODE can not be set to "NO_CHANGE". For simple dual port implementations, it is generally suggested to set WRITE_MODE to "READ_FIRST" if using the same clock on both ports and to set it to "WRITE_FIRST" if using different clocks. This

Attribute	Type	Allowed Values	Default	Description
				generally yields an improved collision or address overlap behavior when using the BRAM in this configuration.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- RAMB36E1: 36K-bit Configurable Synchronous Block RAM
--           7 Series
-- Xilinx HDL Language Template, version 2025.1

RAMB36E1_inst : RAMB36E1
generic map (
   -- Address Collision Mode: "PERFORMANCE" or
"DELAYED_WRITE"
   RDADDR_COLLISION_HWCONFIG => "DELAYED_WRITE",
   -- Collision check: Values ("ALL", "WARNING_ONLY",
"GENERATE_X_ONLY" or "NONE")
   SIM_COLLISION_CHECK => "ALL",
   -- DOA_REG, DOB_REG: Optional output register (0 or 1)
   DOA_REG => 0,
   DOB_REG => 0,
   EN_ECC_READ => FALSE,
-- Enable ECC decoder,

-- FALSE, TRUE
   EN_ECC_WRITE => FALSE,
-- Enable ECC encoder,

-- FALSE, TRUE
   -- INITP_00 to INITP_0F: Initial contents of the parity
```

```
memory array
   INITP_00 =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_01 =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_02 =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_03 =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_04 =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_05 =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_06 =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_07 =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_08 =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_09 =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_0A =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_0B =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_0C =
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_0D =
X"0000000000000000000000000000000000000000000000000000000000000
```

```
00000",
   INITP_0E =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INITP_0F =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   -- INIT_00 to INIT_7F: Initial contents of the data memory
array
   INIT_00 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_01 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_02 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_03 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_04 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_05 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_06 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_07 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_08 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_09 =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0A =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
```

```
   INIT_0B =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0C =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0D =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0E =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_0F =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_10 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_11 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_12 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_13 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_14 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_15 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_16 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_17 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_18 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
```

```
   INIT_19 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1A =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1B =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1C =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1D =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1E =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_1F =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_20 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_21 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_22 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_23 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_24 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_25 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_26 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
```

```
   INIT_27 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_28 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_29 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2A =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2B =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2C =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2D =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2E =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2F =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_30 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_31 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_32 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_33 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_34 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INIT_35 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_36 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_37 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_38 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_39 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3A =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3B =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3C =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3D =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3E =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3F =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_40 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_41 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_42 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INIT_43 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_44 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_45 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_46 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_47 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_48 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_49 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4A =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4B =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4C =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4D =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4E =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4F =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_50 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INIT_51 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_52 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_53 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_54 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_55 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_56 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_57 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_58 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_59 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5A =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5B =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5C =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5D =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5E =>
X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INIT_5F =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_60 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_61 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_62 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_63 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_64 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_65 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_66 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_67 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_68 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_69 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6A =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6B =>
X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6C =>
X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INIT_6D =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_6E =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_6F =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_70 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_71 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_72 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_73 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_74 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_75 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_76 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_77 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_78 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_79 =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_7A =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000",
```

```
   INIT_7B =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_7C =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_7D =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_7E =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   INIT_7F =>
X"0000000000000000000000000000000000000000000000000000000000000
00000",
   -- INIT_A, INIT_B: Initial values on output ports
   INIT_A => X"000000000",
   INIT_B => X"000000000",
   -- Initialization File: RAM initialization file
   INIT_FILE => "NONE",
   -- RAM Mode: "SDP" or "TDP"
   RAM_MODE => "TDP",
   -- RAM_EXTENSION_A, RAM_EXTENSION_B: Selects cascade mode
("UPPER", "LOWER", or "NONE")
   RAM_EXTENSION_A => "NONE",
   RAM_EXTENSION_B => "NONE",
   -- READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per
port
   READ_WIDTH_A => 0,
-- 0-72
   READ_WIDTH_B => 0,
-- 0-36
   WRITE_WIDTH_A => 0,
-- 0-36
   WRITE_WIDTH_B => 0,
-- 0-72
   -- RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable
priority ("RSTREG" or "REGCE")
   RSTREG_PRIORITY_A => "RSTREG",
   RSTREG_PRIORITY_B => "RSTREG",
   -- SRVAL_A, SRVAL_B: Set/reset value for output
   SRVAL_A => X"000000000",
```

```
   SRVAL_B => X"000000000",
   -- Simulation Device: Must be set to "7SERIES" for
simulation behavior
   SIM_DEVICE => "7SERIES",
   -- WriteMode: Value on output upon a write ("WRITE_FIRST",
"READ_FIRST", or "NO_CHANGE")
   WRITE_MODE_A => "WRITE_FIRST",
   WRITE_MODE_B => "WRITE_FIRST"
)
port map (
   -- Cascade Signals: 1-bit (each) output: BRAM cascade
ports (to create 64kx1)
   CASCADEOUTA => CASCADEOUTA,     -- 1-bit output: A port
cascade
   CASCADEOUTB => CASCADEOUTB,     -- 1-bit output: B port
cascade
   -- ECC Signals: 1-bit (each) output: Error Correction
Circuitry ports
   DBITERR => DBITERR,             -- 1-bit output: Double
bit error status
   ECCPARITY => ECCPARITY,         -- 8-bit output: Generated
error correction parity
   RDADDRECC => RDADDRECC,         -- 9-bit output: ECC read
address
   SBITERR => SBITERR,             -- 1-bit output: Single
bit error status
   -- Port A Data: 32-bit (each) output: Port A data
   DOADO => DOADO,                 -- 32-bit output: A port
data/LSB data
   DOPADOP => DOPADOP,             -- 4-bit output: A port
parity/LSB parity
   -- Port B Data: 32-bit (each) output: Port B data
   DOBDO => DOBDO,                 -- 32-bit output: B port
data/MSB data
   DOPBDOP => DOPBDOP,             -- 4-bit output: B port
parity/MSB parity
   -- Cascade Signals: 1-bit (each) input: BRAM cascade ports
(to create 64kx1)
   CASCADEINA => CASCADEINA,       -- 1-bit input: A port
cascade
   CASCADEINB => CASCADEINB,       -- 1-bit input: B port
cascade
```

```
   -- ECC Signals: 1-bit (each) input: Error Correction
Circuitry ports
   INJECTDBITERR => INJECTDBITERR, -- 1-bit input: Inject a
double bit error
   INJECTSBITERR => INJECTSBITERR, -- 1-bit input: Inject a
single bit error
   -- Port A Address/Control Signals: 16-bit (each) input:
Port A address and control signals (read port
   -- when RAM_MODE="SDP")
   ADDRARDADDR => ADDRARDADDR,     -- 16-bit input: A port
address/Read address
   CLKARDCLK => CLKARDCLK,         -- 1-bit input: A port
clock/Read clock
   ENARDEN => ENARDEN,             -- 1-bit input: A port
enable/Read enable
   REGCEAREGCE => REGCEAREGCE,     -- 1-bit input: A port
register enable/Register enable
   RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit input: A port
set/reset
   RSTREGARSTREG => RSTREGARSTREG, -- 1-bit input: A port
register set/reset
   WEA => WEA,                     -- 4-bit input: A port
write enable
   -- Port A Data: 32-bit (each) input: Port A data
   DIADI => DIADI,                 -- 32-bit input: A port
data/LSB data
   DIPADIP => DIPADIP,             -- 4-bit input: A port
parity/LSB parity
   -- Port B Address/Control Signals: 16-bit (each) input:
Port B address and control signals (write port
   -- when RAM_MODE="SDP")
   ADDRBWRADDR => ADDRBWRADDR,     -- 16-bit input: B port
address/Write address
   CLKBWRCLK => CLKBWRCLK,         -- 1-bit input: B port
clock/Write clock
   ENBWREN => ENBWREN,             -- 1-bit input: B port
enable/Write enable
   REGCEB => REGCEB,               -- 1-bit input: B port
register enable
   RSTRAMB => RSTRAMB,             -- 1-bit input: B port
set/reset
   RSTREGB => RSTREGB,             -- 1-bit input: B port
```

```
register set/reset
   WEBWE => WEBWE,                 -- 8-bit input: B port
write enable/Write enable
   -- Port B Data: 32-bit (each) input: Port B data
   DIBDI => DIBDI,                 -- 32-bit input: B port
data/MSB data
   DIPBDIP => DIPBDIP              -- 4-bit input: B port
parity/MSB parity
);

-- End of RAMB36E1_inst instantiation
```

Verilog Instantiation Template

```
// RAMB36E1: 36K-bit Configurable Synchronous Block RAM
//           7 Series
// Xilinx HDL Language Template, version 2025.1

RAMB36E1 #(
   // Address Collision Mode: "PERFORMANCE" or
"DELAYED_WRITE"
   .RDADDR_COLLISION_HWCONFIG("DELAYED_WRITE"),
   // Collision check: Values ("ALL", "WARNING_ONLY",
"GENERATE_X_ONLY" or "NONE")
   .SIM_COLLISION_CHECK("ALL"),
   // DOA_REG, DOB_REG: Optional output register (0 or 1)
   .DOA_REG(0),
   .DOB_REG(0),
   .EN_ECC_READ("FALSE"),
// Enable ECC decoder,

// FALSE, TRUE
   .EN_ECC_WRITE("FALSE"),
// Enable ECC encoder,

// FALSE, TRUE
   // INITP_00 to INITP_0F: Initial contents of the parity
memory array
```

```
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_08(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_09(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

```
.INITP_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INITP_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
   // INIT_00 to INIT_7F: Initial contents of the data memory array

.INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
00000000000000000),

.INIT_1A(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_1B(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_1C(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_1D(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_1E(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_1F(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_20(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_21(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_22(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_23(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_24(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_25(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_26(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_27(256'h0000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_40(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_41(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_42(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_43(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_44(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_45(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_46(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_47(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_48(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_49(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_4A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_4B(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_4C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_4D(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_4E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_4F(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_50(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_51(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_52(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_53(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_54(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_55(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_56(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_57(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_58(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_59(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_5A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_5B(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_5C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_5D(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_5E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_5F(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_60(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_61(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_62(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_63(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_64(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_65(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_66(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_67(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_68(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_69(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_6A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_6B(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_6C(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_6D(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000),

.INIT_6E(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_6F(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_70(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_71(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_72(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_73(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_74(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_75(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_76(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_77(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_78(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_79(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_7A(256'h0000000000000000000000000000000000000000000000000000000000000000),

.INIT_7B(256'h0000000000000000000000000000000000000000000000000000000000000000
```

```
00000000000000000),

.INIT_7C(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_7D(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_7E(256'h0000000000000000000000000000000000000000000000000
00000000000000000),

.INIT_7F(256'h0000000000000000000000000000000000000000000000000
00000000000000000),
   // INIT_A, INIT_B: Initial values on output ports
   .INIT_A(36'h000000000),
   .INIT_B(36'h000000000),
   // Initialization File: RAM initialization file
   .INIT_FILE("NONE"),
   // RAM Mode: "SDP" or "TDP"
   .RAM_MODE("TDP"),
   // RAM_EXTENSION_A, RAM_EXTENSION_B: Selects cascade mode
("UPPER", "LOWER", or "NONE")
   .RAM_EXTENSION_A("NONE"),
   .RAM_EXTENSION_B("NONE"),
   // READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per
port
   .READ_WIDTH_A(0),
// 0-72
   .READ_WIDTH_B(0),
// 0-36
   .WRITE_WIDTH_A(0),
// 0-36
   .WRITE_WIDTH_B(0),
// 0-72
   // RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable
priority ("RSTREG" or "REGCE")
   .RSTREG_PRIORITY_A("RSTREG"),
   .RSTREG_PRIORITY_B("RSTREG"),
   // SRVAL_A, SRVAL_B: Set/reset value for output
   .SRVAL_A(36'h000000000),
   .SRVAL_B(36'h000000000),
   // Simulation Device: Must be set to "7SERIES" for
```

```
simulation behavior
   .SIM_DEVICE("7SERIES"),
   // WriteMode: Value on output upon a write ("WRITE_FIRST",
"READ_FIRST", or "NO_CHANGE")
   .WRITE_MODE_A("WRITE_FIRST"),
   .WRITE_MODE_B("WRITE_FIRST")
)
RAMB36E1_inst (
   // Cascade Signals: 1-bit (each) output: BRAM cascade
ports (to create 64kx1)
   .CASCADEOUTA(CASCADEOUTA),     // 1-bit output: A port
cascade
   .CASCADEOUTB(CASCADEOUTB),     // 1-bit output: B port
cascade
   // ECC Signals: 1-bit (each) output: Error Correction
Circuitry ports
   .DBITERR(DBITERR),             // 1-bit output: Double bit
error status
   .ECCPARITY(ECCPARITY),         // 8-bit output: Generated
error correction parity
   .RDADDRECC(RDADDRECC),         // 9-bit output: ECC read
address
   .SBITERR(SBITERR),             // 1-bit output: Single bit
error status
   // Port A Data: 32-bit (each) output: Port A data
   .DOADO(DOADO),                 // 32-bit output: A port
data/LSB data
   .DOPADOP(DOPADOP),             // 4-bit output: A port
parity/LSB parity
   // Port B Data: 32-bit (each) output: Port B data
   .DOBDO(DOBDO),                 // 32-bit output: B port
data/MSB data
   .DOPBDOP(DOPBDOP),             // 4-bit output: B port
parity/MSB parity
   // Cascade Signals: 1-bit (each) input: BRAM cascade ports
(to create 64kx1)
   .CASCADEINA(CASCADEINA),       // 1-bit input: A port
cascade
   .CASCADEINB(CASCADEINB),       // 1-bit input: B port
cascade
   // ECC Signals: 1-bit (each) input: Error Correction
Circuitry ports
```

```
   .INJECTDBITERR(INJECTDBITERR), // 1-bit input: Inject a
double bit error
   .INJECTSBITERR(INJECTSBITERR), // 1-bit input: Inject a
single bit error
   // Port A Address/Control Signals: 16-bit (each) input:
Port A address and control signals (read port
   // when RAM_MODE="SDP")
   .ADDRARDADDR(ADDRARDADDR),     // 16-bit input: A port
address/Read address
   .CLKARDCLK(CLKARDCLK),         // 1-bit input: A port
clock/Read clock
   .ENARDEN(ENARDEN),             // 1-bit input: A port
enable/Read enable
   .REGCEAREGCE(REGCEAREGCE),     // 1-bit input: A port
register enable/Register enable
   .RSTRAMARSTRAM(RSTRAMARSTRAM), // 1-bit input: A port set/
reset
   .RSTREGARSTREG(RSTREGARSTREG), // 1-bit input: A port
register set/reset
   .WEA(WEA),                     // 4-bit input: A port
write enable
   // Port A Data: 32-bit (each) input: Port A data
   .DIADI(DIADI),                 // 32-bit input: A port
data/LSB data
   .DIPADIP(DIPADIP),             // 4-bit input: A port
parity/LSB parity
   // Port B Address/Control Signals: 16-bit (each) input:
Port B address and control signals (write port
   // when RAM_MODE="SDP")
   .ADDRBWRADDR(ADDRBWRADDR),     // 16-bit input: B port
address/Write address
   .CLKBWRCLK(CLKBWRCLK),         // 1-bit input: B port
clock/Write clock
   .ENBWREN(ENBWREN),             // 1-bit input: B port
enable/Write enable
   .REGCEB(REGCEB),               // 1-bit input: B port
register enable
   .RSTRAMB(RSTRAMB),             // 1-bit input: B port set/
reset
   .RSTREGB(RSTREGB),             // 1-bit input: B port
register set/reset
   .WEBWE(WEBWE),                 // 8-bit input: B port
```

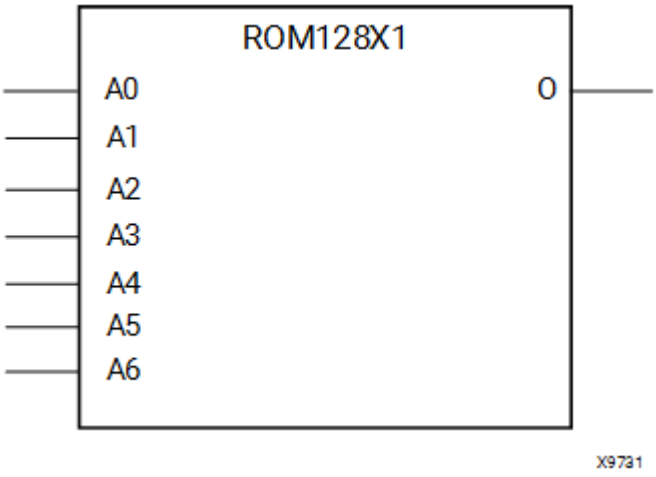
```
write enable/Write enable
   // Port B Data: 32-bit (each) input: Port B data
   .DIBDI(DIBDI),                 // 32-bit input: B port
data/MSB data
   .DIPBDIP(DIPBDIP)              // 4-bit input: B port
parity/MSB parity
);

// End of RAMB36E1_inst instantiation
```

Related Information

- *7 Series FPGAs Memory Resources User Guide* (UG473)

ROM128X1

Primitive: 128-Deep by 1-Wide ROM



Introduction

This design element is a 128-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 7-bit address (A6:A0). The ROM is initialized to a known value during configuration with the INIT parameter. The value consists of 32 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H.
An error occurs if INIT is not specified.

Logic Table

Input				Output
I0	**I1**	**I2**	**I3**	**O**
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

Design Entry Method

Instantiation	Yes
Inference	Recommended

IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 128-Bit Value	All zeros	Specifies the contents of the ROM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;
```

```
-- ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
--           7 Series
-- Xilinx HDL Language Template, version 2025.1

ROM128X1_inst : ROM128X1
generic map (
   INIT => X"00000000000000000000000000000000")
port map (
   O => O,   -- ROM output
   A0 => A0, -- ROM address[0]
   A1 => A1, -- ROM address[1]
   A2 => A2, -- ROM address[2]
   A3 => A3, -- ROM address[3]
   A4 => A4, -- ROM address[4]
   A5 => A5, -- ROM address[5]
   A6 => A6  -- ROM address[6]
);

-- End of ROM128X1_inst instantiation
```

Verilog Instantiation Template

```
// ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
(Mapped to two SliceM LUT6s)
//           7 Series
// Xilinx HDL Language Template, version 2025.1

ROM128X1 #(
   .INIT(128'h00000000000000000000000000000000) // Contents
of ROM
) ROM128X1_inst (
   .O(O),   // ROM output
   .A0(A0), // ROM address[0]
   .A1(A1), // ROM address[1]
   .A2(A2), // ROM address[2]
   .A3(A3), // ROM address[3]
   .A4(A4), // ROM address[4]
   .A5(A5), // ROM address[5]
```

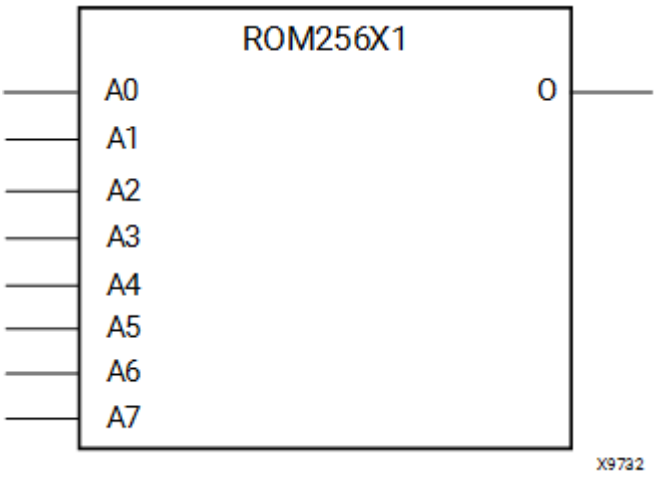
```
   .A6(A6)  // ROM address[6]
);

// End of ROM128X1_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

ROM256X1

Primitive: 256-Deep by 1-Wide ROM



Introduction

This design element is a 256-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 8-bit address (A7:A0). The ROM is initialized to a known value during configuration with the INIT parameter. The value consists of 64 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H.
An error occurs if the INIT is not specified.

Logic Table

Input	Output

I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 256-Bit Value	All zeros	Specifies the contents of the ROM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
--           7 Series
-- Xilinx HDL Language Template, version 2025.1

ROM256X1_inst : ROM256X1
generic map (
   INIT =>
X"0000000000000000000000000000000000000000000000000000000000000000
00000")
port map (
   O => O,   -- ROM output
   A0 => A0, -- ROM address[0]
   A1 => A1, -- ROM address[1]
   A2 => A2, -- ROM address[2]
   A3 => A3, -- ROM address[3]
   A4 => A4, -- ROM address[4]
   A5 => A5, -- ROM address[5]
   A6 => A6,  -- ROM address[6]
   A7 => A7  -- ROM address[7]
);

-- End of ROM256X1_inst instantiation
```

Verilog Instantiation Template

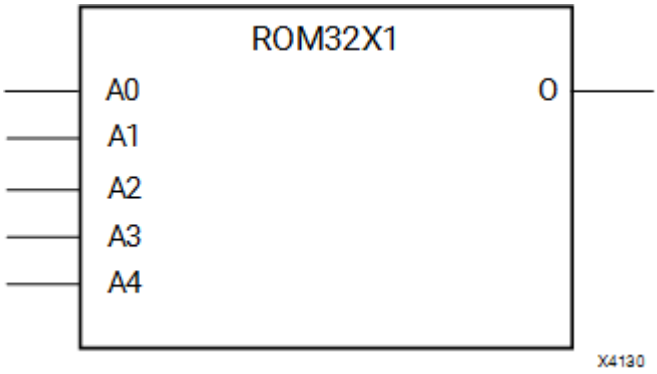
```
// ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
(Mapped to four SliceM LUT6s)
//           7 Series
// Xilinx HDL Language Template, version 2025.1

ROM256X1 #(

.INIT(256'h0000000000000000000000000000000000000000000000000
00000000000000) // Contents of ROM
) ROM256X1_inst (
   .O(O),   // ROM output
   .A0(A0), // ROM address[0]
   .A1(A1), // ROM address[1]
   .A2(A2), // ROM address[2]
   .A3(A3), // ROM address[3]
   .A4(A4), // ROM address[4]
   .A5(A5), // ROM address[5]
   .A6(A6), // ROM address[6]
   .A7(A7)  // ROM address[7]
);

// End of ROM256X1_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

ROM32X1

Primitive: 32-Deep by 1-Wide ROM



Introduction

This design element is a 32-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 5-bit address (A4:A0). The ROM is initialized to a known value during configuration with the INIT parameter. The value consists of eight hexadecimal digits that are written into the ROM from the most-significant digit A=1FH to the least-significant digit A=00H.
For example, INIT=10A78F39 produces the data stream: 0001 0000 1010 0111 1000 1111 0011 1001.
An error occurs if the INIT is not specified.

Logic Table

Input				**Output**
I0	**I1**	**I2**	**I3**	**O**
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)

Input				Output
I0	**I1**	**I2**	**I3**	**O**
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 32-Bit Value	All zeros	Specifies the contents of the ROM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
--          7 Series
-- Xilinx HDL Language Template, version 2025.1

ROM32X1_inst : ROM32X1
generic map (
   INIT => X"00000000")
port map (
   O => O,   -- ROM output
   A0 => A0, -- ROM address[0]
   A1 => A1, -- ROM address[1]
   A2 => A2, -- ROM address[2]
   A3 => A3, -- ROM address[3]
   A4 => A4  -- ROM address[4]
);
-- End of ROM32X1_inst instantiation
```

Verilog Instantiation Template

```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM (Mapped
to a SliceM LUT6)
//          7 Series
// Xilinx HDL Language Template, version 2025.1

ROM32X1 #(
   .INIT(32'h00000000) // Contents of ROM
) ROM32X1_inst (
   .O(O),   // ROM output
   .A0(A0), // ROM address[0]
   .A1(A1), // ROM address[1]
   .A2(A2), // ROM address[2]
   .A3(A3), // ROM address[3]
   .A4(A4)  // ROM address[4]
);
```

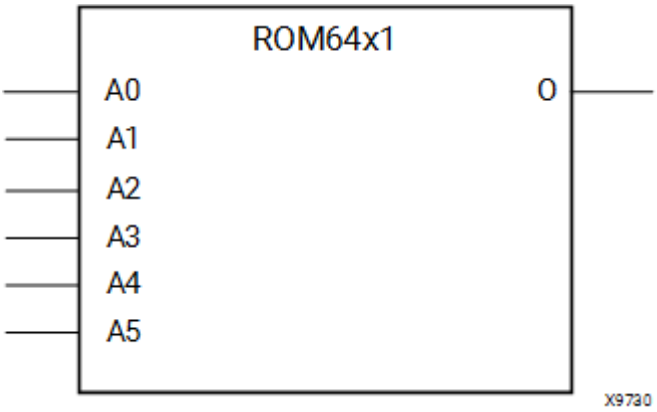
```
// End of ROM32X1_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)

ROM64X1

Primitive: 64-Deep by 1-Wide ROM



Introduction

This design element is a 64-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 6-bit address (A5:A0). The ROM is initialized to a known value during configuration with the INIT parameter. The value consists of 16 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H.
An error occurs if INIT is not specified.

Logic Table

Input				Output
I0	**I1**	**I2**	**I3**	**O**
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)

Input				Output
I0	**I1**	**I2**	**I3**	**O**
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 64-Bit Value	All zeros	Specifies the contents of the ROM.

VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;



-- ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
--          7 Series
-- Xilinx HDL Language Template, version 2025.1

ROM64X1_inst : ROM64X1
generic map (
   INIT => X"0000000000000000")
port map (
   O => O,   -- ROM output
   A0 => A0, -- ROM address[0]
   A1 => A1, -- ROM address[1]
   A2 => A2, -- ROM address[2]
   A3 => A3, -- ROM address[3]
   A4 => A4, -- ROM address[4]
   A5 => A5  -- ROM address[5]
);

-- End of ROM64X1_inst instantiation
```

Verilog Instantiation Template

```
// ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM (Mapped
to a SliceM LUT6)
//          7 Series
```

```
// Xilinx HDL Language Template, version 2025.1

ROM64X1 #(
   .INIT(64'h0000000000000000) // Contents of ROM
) ROM64X1_inst (
   .O(O),   // ROM output
   .A0(A0), // ROM address[0]
   .A1(A1), // ROM address[1]
   .A2(A2), // ROM address[2]
   .A3(A3), // ROM address[3]
   .A4(A4), // ROM address[4]
   .A5(A5)  // ROM address[5]
);

// End of ROM64X1_inst instantiation
```

Related Information

- *7 Series FPGAs Configurable Logic Block User Guide* (UG474)