

Продолжение. Начало в № 2 '2010

Разработка VHDL-описаний цифровых устройств, проектируемых на основе ПЛИС фирмы Xilinx, с использованием шаблонов САПР ISE Design Suite

В одиннадцатой части статьи завершается изучение шаблонов VHDL-описаний 2-портовых ОЗУ, реализуемых на основе ресурсов блочной памяти Block RAM кристаллов программируемой логики семейства Spartan-3 [1, 19, 20]. Рассмотрены также образцы описания различных вариантов конфигурирования модуля блочной памяти ПЛИС указанного семейства в виде элемента 1-портового ОЗУ. Здесь же приведена подробная информация о шаблонах описаний элементов распределенной оперативной памяти.

Валерий ЗОТОВ
walerry@km.ru

8k/512 x 2/32 + 0/4 Parity bits (RAMB16_S2_S36) включает в себя шаблон описания элемента 2-портового ОЗУ, конфигурируемого на базе модуля блочной памяти кристаллов программируемой логики семейства Spartan-3, в котором первый порт имеет организацию 8192 слова × 2 разряда, а второй — 512 слов × 32 разряда. Основой этой конструкции является оператор создания экземпляра библиотечного элемента оперативной памяти RAMB16_S2_S36, второй порт которого поддерживает возможность выполнения операций с применением контроля четности.

```
-- RAMB16_S2_S36 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : (RAMB16_S2_S36_inst) and/or the port declarations
-- code : after the "=" assignment may be changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <----Cut code below this line and paste into the architecture
body---->
--
-- RAMB16_S2_S36: 8k/512 x 2/32 + 0/4 Parity bits Dual-Port RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
RAMB16_S2_S36_inst : RAMB16_S2_S36
generic map (
    INIT_A => X"0", -- Value of output RAM registers on Port A
at startup
    INIT_B => X"00000000", -- Value of output RAM registers
on Port B at startup
```

```
SRVAL_A => X"0", -- Port A output value upon SSR assertion
SRVAL_B => X"00000000", -- Port B output value upon SSR
assertion
    WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST,
READ_FIRST or NO_CHANGE
    WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST,
READ_FIRST or NO_CHANGE
    SIM_COLLISION_CHECK => "ALL", -- "NONE",
"WARNING", "GENERATE_X_ONLY", "ALL"
-- The following INIT_xx declarations specify the initial contents
of the RAM
-- Port A Address 0 to 2047, Port B Address 0 to 127
INIT_00 => X"0000000000000000000000000000000000000000000000000000",
INIT_01 => X"0000000000000000000000000000000000000000000000000000",
INIT_02 => X"0000000000000000000000000000000000000000000000000000",
...
INIT_0E => X"0000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000",
-- Port A Address 2048 to 4095, Port B Address 128 to 255
INIT_10 => X"0000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000",
...
INIT_1E => X"0000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000",
-- Port A Address 4096 to 6143, Port B Address 256 to 383
INIT_20 => X"0000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000",
...
INIT_2E => X"0000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000",
-- Port A Address 6144 to 8191, Port B Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000",
...
INIT_3E => X"0000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port B Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000",
INITP_01 => X"000000000000000000000000000000000000000000000000000",
-- Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000",
-- Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000",
-- Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000",
port map (
    DOA => DOA, -- Port A 2-bit Data Output
    DOB => DOB, -- Port B 32-bit Data Output
    DOPB => DOPB, -- Port B 4-bit Parity Output
```

```
ADDR_A => ADDR_A, -- Port A 13-bit Address Input
ADDR_B => ADDR_B, -- Port B 9-bit Address Input
CLKA => CLKA, -- Port A Clock
CLKB => CLKB, -- Port B Clock
DIA => DIA, -- Port A 2-bit Data Input
DIB => DIB, -- Port B 32-bit Data Input
DIPB => DIPB, -- Port B 4-bit parity Input
ENA => ENA, -- Port A RAM Enable Input
ENB => ENB, -- Port B RAM Enable Input
SSRA => SSRA, -- Port A Synchronous Set/Reset Input
SSRB => SSRB, -- Port B Synchronous Set/Reset Input
WEA => WEA, -- Port A Write Enable Input
WEB => WEB -- Port B Write Enable Input
);
-- End of RAMB16_S2_S36_inst instantiation
```

После включения представленной конструкции в состав описания разрабатываемого устройства следует указать идентификаторы цепей, подключаемых к интерфейсным портам формируемого экземпляра библиотечного примитива RAMB16_S2_S36. В этом шаблоне используются те же параметры настройки и система условных обозначений интерфейсных портов, что и в шаблонах 16k x 1 (RAMB16_S1_S1) и 1k x 16 + 2 Parity bits (RAMB16_S18_S18), информация о которых была приведена в девятой части статьи (КиТ. 2010. № 10). При этом нужно обратить внимание на то, что основные шины данных второго порта формируемого элемента оперативной памяти являются 32-разрядными, а дополнительные шины данных — 4-разрядными. Шина адреса второго порта содержит девять двоичных разрядов.

На рис. 100 показан условный графический образ элемента 2-портовой оперативной памяти с организацией портов 8192 слова × 2 разряда и 512 слов × 32 разряда, для создания описания которого используется шаблон 8k/512 x 2/32 + 0/4 Parity bits (RAMB16_S2_S36).

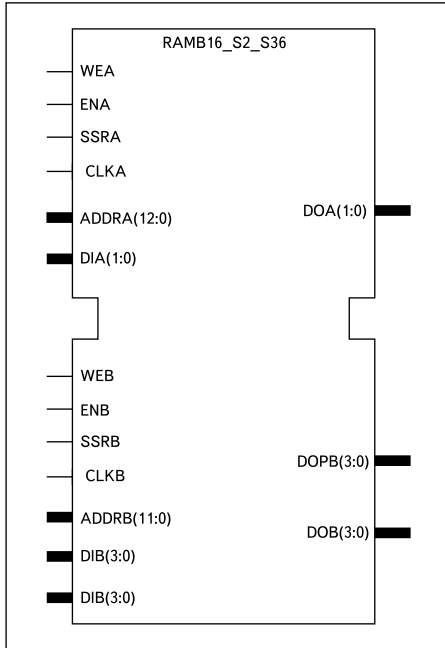


Рис. 100. Условный графический образ элемента 2-портового ОЗУ (шаблон 8k/512 x 2/32 + 0/4 Parity bits (RAMB16_S2_S36))

1k/512 x 16/32 + 2/4 Parity bits (RAMB16_S18_S36) является образцом VHDL-описания варианта конфигурирования модуля блочной памяти кристаллов программируемой логики семейства Spartan-3 в виде элемента 2-портового ОЗУ с организацией первого и второго порта 1024 слова × 16 разрядов и 512 слов × 32 разряда соответственно. Этот шаблон выполнен на основе использования библиотечного примитива *RAMB16_S18_S36*, который предоставляет возможность выполнения операций записи и чтения данных с использованием контроля четности для каждого порта.

```
-- RAMB16_S18_S36: In order to incorporate this function into the design,
-- VHDL: the following instance declaration needs to be placed
-- instance: in the architecture body of the design code. The
-- declaration: (RAMB16_S18_S36_inst) and/or the port declarations
-- code: after the "=" assignment may be changed to properly
--: reference and connect this function to the design.
--: All inputs and outputs must be connected.
--
-- Library: In addition to adding the instance declaration, a use
-- declaration: statement for the UNISIM.vcomponents library needs to be
-- for: added before the entity declaration. This library
-- Xilinx: contains the component declarations for all Xilinx
-- primitives: primitives and points to the models that will be used
--: for simulation.
```

```
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
```

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

```
-- <-----Cut code below this line and paste into the architecture
body----->
```

```
-- RAMB16_S18_S36: 1k/512 x 16/32 + 2/4 Parity bits Dual-Port RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
```

```
RAMB16_S18_S36_inst : RAMB16_S18_S36
generic map (
INIT_A => X"000000", -- Value of output RAM registers on
Port A at startup
INIT_B => X"0000000000", -- Value of output RAM registers
on Port B at startup

```

```
SRVAL_A => X"000000", -- Port A output value upon SSR
assertion
SRVAL_B => X"0000000000", -- Port B output value upon
SSR assertion
WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST,
READ_FIRST or NO_CHANGE
WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST,
READ_FIRST or NO_CHANGE
SIM_COLLISION_CHECK => "ALL", -- "NONE",
"WARNING", "GENERATE_X_ONLY", "ALL"
-- The following INIT_xx declarations specify the initial contents
of the RAM
-- Port A Address 0 to 255, Port B Address 0 to 127
INIT_00 => X"00000000000000000000000000000000000000000000000000000",
INIT_01 => X"00000000000000000000000000000000000000000000000000000",
INIT_02 => X"00000000000000000000000000000000000000000000000000000",
...
INIT_0D => X"00000000000000000000000000000000000000000000000000000",
INIT_0E => X"00000000000000000000000000000000000000000000000000000",
INIT_0F => X"00000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INIT_10 => X"00000000000000000000000000000000000000000000000000000",
INIT_11 => X"00000000000000000000000000000000000000000000000000000",
INIT_12 => X"00000000000000000000000000000000000000000000000000000",
...
INIT_1D => X"00000000000000000000000000000000000000000000000000000",
INIT_1E => X"00000000000000000000000000000000000000000000000000000",
INIT_1F => X"00000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INIT_20 => X"00000000000000000000000000000000000000000000000000000",
INIT_21 => X"00000000000000000000000000000000000000000000000000000",
INIT_22 => X"00000000000000000000000000000000000000000000000000000",
...
INIT_2D => X"00000000000000000000000000000000000000000000000000000",
INIT_2E => X"00000000000000000000000000000000000000000000000000000",
INIT_2F => X"00000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INIT_30 => X"00000000000000000000000000000000000000000000000000000",
INIT_31 => X"00000000000000000000000000000000000000000000000000000",
INIT_32 => X"00000000000000000000000000000000000000000000000000000",
...
INIT_3D => X"00000000000000000000000000000000000000000000000000000",
INIT_3E => X"00000000000000000000000000000000000000000000000000000",
INIT_3F => X"00000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Port A Address 0 to 255, Port B Address 0 to 127
INITP_00 => X"00000000000000000000000000000000000000000000000000000",
INITP_01 => X"00000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INITP_02 => X"00000000000000000000000000000000000000000000000000000",
INITP_03 => X"00000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INITP_04 => X"00000000000000000000000000000000000000000000000000000",
INITP_05 => X"00000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INITP_06 => X"00000000000000000000000000000000000000000000000000000",
INITP_07 => X"00000000000000000000000000000000000000000000000000000",
port map (
DOA => DOA, -- Port A 16-bit Data Output
DOB => DOB, -- Port B 32-bit Data Output
DOPA => DOPA, -- Port A 2-bit Parity Output
DOPB => DOPB, -- Port B 4-bit Parity Output
ADDRA => ADDR, -- Port A 10-bit Address Input
ADDRB => ADDR, -- Port B 9-bit Address Input
CLKA => CLKA, -- Port A Clock
CLKB => CLKB, -- Port B Clock
DIA => DIA, -- Port A 16-bit Data Input
DIB => DIB, -- Port B 32-bit Data Input
DIPA => DIPA, -- Port A 2-bit parity Input
DIPB => DIPB, -- Port B 4-bit parity Input
ENA => ENA, -- Port A RAM Enable Input
ENB => ENB, -- Port B RAM Enable Input
SSRA => SSRA, -- Port A Synchronous Set/Reset Input
SSRB => SSRB, -- Port B Synchronous Set/Reset Input
WEA => WEA, -- Port A Write Enable Input
WEB => WEB -- Port B Write Enable Input
);
-- End of RAMB16_S18_S36_inst instantiation
```

```
DOA => DOA, -- Port A 16-bit Data Output
DOB => DOB, -- Port B 32-bit Data Output
DOPA => DOPA, -- Port A 2-bit Parity Output
DOPB => DOPB, -- Port B 4-bit Parity Output
ADDRA => ADDR, -- Port A 10-bit Address Input
ADDRB => ADDR, -- Port B 9-bit Address Input
CLKA => CLKA, -- Port A Clock
CLKB => CLKB, -- Port B Clock
DIA => DIA, -- Port A 16-bit Data Input
DIB => DIB, -- Port B 32-bit Data Input
DIPA => DIPA, -- Port A 2-bit parity Input
DIPB => DIPB, -- Port B 4-bit parity Input
ENA => ENA, -- Port A RAM Enable Input
ENB => ENB, -- Port B RAM Enable Input
SSRA => SSRA, -- Port A Synchronous Set/Reset Input
SSRB => SSRB, -- Port B Synchronous Set/Reset Input
WEA => WEA, -- Port A Write Enable Input
WEB => WEB -- Port B Write Enable Input
);
-- End of RAMB16_S18_S36_inst instantiation
```

В элементах оперативной памяти, формируемых на базе библиотечного компонента *RAMB16_S18_S36*, разрядность входной и выходной основных шин данных первого порта составляет 16 бит, а второго порта — 32 бита. Кроме того, первый порт содержит дополнительные входную и выходную 2-разрядные шины данных, которые предназначены для организации контроля четности. В составе второго порта для этой цели предусмотрены дополнительные 4-разрядные шины данных. При этом шина адреса первого порта

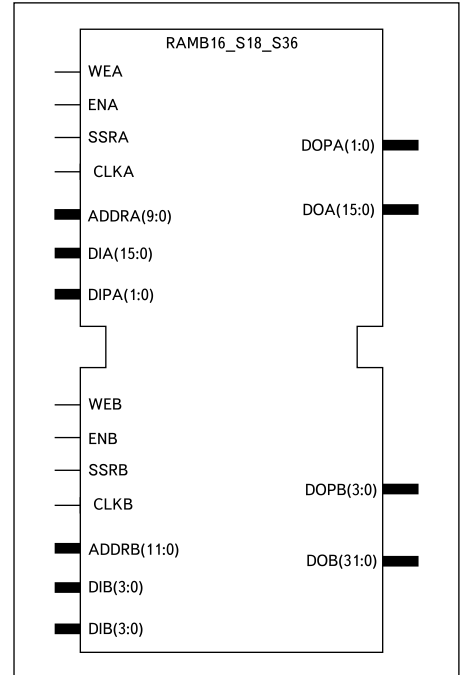


Рис. 101. Условный графический образ 2-портового ОЗУ (шаблон 1k/512 x 16/32 + 2/4 Parity bits (RAMB16_S18_S36))

содержит 10 двоичных разрядов, а второго — 9 разрядов.

Условный графический образ элемента 2-портового ОЗУ с организацией 1024 слова × 16 разрядов и 512 слов × 32 разряда, описание которого создается с помощью шаблона *1k/512 x 16/32 + 2/4 Parity bits (RAMB16_S18_S36)* для последующей реализации на основе модуля блочной памяти Block RAM ПЛИС семейства Spartan-3, изображен на рис. 101.

16k x 1 (RAMB16_S1) представляет собой шаблон описания элемента 1-портового ОЗУ с организацией 16 384 слова × 1 разряд, реализуемого на базе модуля блочной памяти Block RAM кристаллов программируемой логики семейства Spartan-3.

```
-- RAMB16_S1: In order to incorporate this function into the design,
-- VHDL: the following instance declaration needs to be placed
-- instance: in the architecture body of the design code. The
-- declaration: instance name (RAMB16_S1_inst) and/or the port
declarations
-- code: after the "=" assignment may be changed to properly
--: reference and connect this function to the design.
--: All inputs and outputs must be connected.
```

```
-- Library: In addition to adding the instance declaration, a use
-- declaration: statement for the UNISIM.vcomponents library needs to be
-- for: added before the entity declaration. This library
-- Xilinx: contains the component declarations for all Xilinx
-- primitives: primitives and points to the models that will be used
--: for simulation.
```

```
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
```

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

```
-- <-----Cut code below this line and paste into the architecture
body----->
```

```
-- RAMB16_S1: Single-Port RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
```

```

RAMB16_S1_inst : RAMB16_S1
generic map (
    INIT => X"0", -- Value of output RAM registers at startup
    SRVAL => X"0", -- Output value upon SSR assertion
    WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST,
    READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the initial contents
of the RAM
-- Address 0 to 4095
    INIT_00 => X"00000000000000000000000000000000000000000000",
    INIT_01 => X"00000000000000000000000000000000000000000000",
    INIT_02 => X"00000000000000000000000000000000000000000000",
...
    INIT_0D => X"00000000000000000000000000000000000000000000",
    INIT_0E => X"00000000000000000000000000000000000000000000",
    INIT_0F => X"00000000000000000000000000000000000000000000",
-- Address 4096 to 8191
    INIT_10 => X"00000000000000000000000000000000000000000000",
    INIT_11 => X"00000000000000000000000000000000000000000000",
    INIT_12 => X"00000000000000000000000000000000000000000000",
...
    INIT_1D => X"00000000000000000000000000000000000000000000",
    INIT_1E => X"00000000000000000000000000000000000000000000",
    INIT_1F => X"00000000000000000000000000000000000000000000",
-- Address 8192 to 12287
    INIT_20 => X"00000000000000000000000000000000000000000000",
    INIT_21 => X"00000000000000000000000000000000000000000000",
    INIT_22 => X"00000000000000000000000000000000000000000000",
...
    INIT_2D => X"00000000000000000000000000000000000000000000",
    INIT_2E => X"00000000000000000000000000000000000000000000",
    INIT_2F => X"00000000000000000000000000000000000000000000",
-- Address 12288 to 16383
    INIT_30 => X"00000000000000000000000000000000000000000000",
    INIT_31 => X"00000000000000000000000000000000000000000000",
    INIT_32 => X"00000000000000000000000000000000000000000000",
...
    INIT_3D => X"00000000000000000000000000000000000000000000",
    INIT_3E => X"00000000000000000000000000000000000000000000",
    INIT_3F => X"00000000000000000000000000000000000000000000")
port map (
    DO => DO, -- 1-bit Data Output
    ADDR => ADDR, -- 14-bit Address Input
    CLK => CLK, -- Clock
    DI => DI, -- 1-bit Data Input
    EN => EN, -- RAM Enable Input
    SSR => SSR, -- Synchronous Set/Reset Input
    WE => WE -- Write Enable Input
);
-- End of RAMB16_S1_inst instantiation

```

Основу приведенного шаблона образует оператор создания экземпляра библиотечного элемента **RAMB16_S1**. В этом библиотечном примитиве используются следующие настраиваемые параметры, предназначенные для определения начального содержимого ячеек и режимов функционирования формируемого элемента 1-портовой оперативной памяти:

- **INIT** — указывает начальное значение для выходного регистра создаваемого ОЗУ, которое заносится после завершения процесса конфигурирования кристалла (по умолчанию предлагается нулевое значение).
- **SRVAL** — определяет состояние выходного регистра при подаче активного уровня сигнала на вход сброса/установки (по умолчанию устанавливается нулевое значение).
- **WRITE_MODE** — устанавливает режим записи информации (порядок выполнения операций записи и чтения данных при одновременном обращении к ячейкам памяти).
- **INIT_00–INIT_3F** — предоставляют возможность инициализации содержимого соответствующих ячеек памяти (по умолчанию во все ячейки ОЗУ записывается нулевое значение).

В системе условных обозначений, используемых в описании интерфейса библиотечного примитива **RAMB16_S1**, представлены

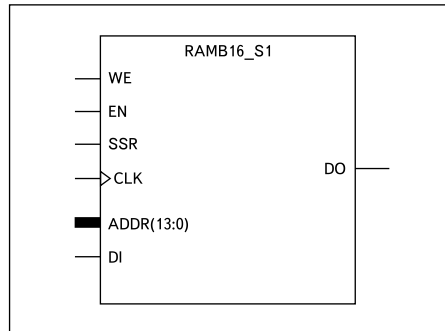


Рис. 102. Условный графический образ 1-портового ОЗУ (шаблон 16k x 1 (RAMB16_S1))

следующие идентификаторы входов и выходов:

- **DO** — выход элемента 1-портовой оперативной памяти;
- **ADDR** — 14-разрядная шина адреса;
- **CLK** — вход сигнала синхронизации;
- **DI** — информационный вход (вход записи);
- **EN** — вход сигнала разрешения;
- **SSR** — вход сигнала синхронного сброса/установки;
- **WE** — вход сигнала разрешения записи.

Для осуществления всех операций, включая сброс/установку, необходимо подать высокий логический уровень напряжения на вход сигнала разрешения EN. Выполнение операций производится по фронту тактового сигнала CLK. Занесение информационных данных с входа DI в ячейку, адрес которой определяется совокупностью значений сигналов, представленных на шине ADDR, осуществляется только при наличии высокого логического уровня напряжения на входе сигнала разрешения записи WE. Если на вход сигнала синхронного сброса/установки SSR поступает активный уровень напряжения, то выход элемента оперативной памяти переключается в состояние, которое определяется указанным значением параметра настройки SRVAL.

На рис. 102 показан условный графический образ элемента 1-портового ОЗУ с информационной емкостью 16 384 бита и организацией 16 384 слова × 1 разряд, описание которого создается с помощью шаблона **16k x 1 (RAMB16_S1)** для последующей реализации на основе модуля блочной памяти Block RAM кристаллов программируемой логики семейства Spartan-3.

1k x 16 + 2 Parity bits (RAMB16_S18) включает в себя образец VHDL-описания элемента 1-портовой оперативной памяти информационной емкостью 16 384 бита с организацией 1024 слова × 16 разрядов и дополнительным объемом 2048 бит, необходимым для осуществления контроля четности загружаемых и считываемых данных. В качестве основы этого шаблона используется экземпляр библиотечного примитива **RAMB16_S18**, который представляет соответствующий вариант конфигурирования модуля блочной памяти Block RAM ПЛИС семейства Spartan-3.

```

-- RAMB16_S18 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAMB16_S18_inst) and/or the port
declarations
-- code : after the "=" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs
to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;

```

```

-- Cut code below this line and paste into the architecture
body---->

```

```

-- RAMB16_S18: 1k x 16 + 2 Parity bits Single-Port RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4

RAMB16_S18_inst : RAMB16_S18
generic map (
    INIT => X"000000", -- Value of output RAM registers at
startup
    SRVAL => X"000000", -- Output value upon SSR assertion
    WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST,
    READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the initial contents
of the RAM
-- Address 0 to 255
    INIT_00 => X"00000000000000000000000000000000000000000000",
    INIT_01 => X"00000000000000000000000000000000000000000000",
    INIT_02 => X"00000000000000000000000000000000000000000000",
...
    INIT_0D => X"00000000000000000000000000000000000000000000",
    INIT_0E => X"00000000000000000000000000000000000000000000",
    INIT_0F => X"00000000000000000000000000000000000000000000",
-- Address 256 to 511
    INIT_10 => X"00000000000000000000000000000000000000000000",
    INIT_11 => X"00000000000000000000000000000000000000000000",
    INIT_12 => X"00000000000000000000000000000000000000000000",
...
    INIT_1D => X"00000000000000000000000000000000000000000000",
    INIT_1E => X"00000000000000000000000000000000000000000000",
    INIT_1F => X"00000000000000000000000000000000000000000000",
-- Address 512 to 767
    INIT_20 => X"00000000000000000000000000000000000000000000",
    INIT_21 => X"00000000000000000000000000000000000000000000",
    INIT_22 => X"00000000000000000000000000000000000000000000",
...
    INIT_2D => X"00000000000000000000000000000000000000000000",
    INIT_2E => X"00000000000000000000000000000000000000000000",
    INIT_2F => X"00000000000000000000000000000000000000000000",
-- Address 768 to 1023
    INIT_30 => X"00000000000000000000000000000000000000000000",
    INIT_31 => X"00000000000000000000000000000000000000000000",
    INIT_32 => X"00000000000000000000000000000000000000000000",
...
    INIT_3D => X"00000000000000000000000000000000000000000000",
    INIT_3E => X"00000000000000000000000000000000000000000000",
    INIT_3F => X"00000000000000000000000000000000000000000000",
-- The next set of INIT_xx are for the parity bits
-- Address 0 to 255
    INITP_00 => X"00000000000000000000000000000000000000000000",
    INITP_01 => X"00000000000000000000000000000000000000000000",
-- Address 256 to 511
    INITP_02 => X"00000000000000000000000000000000000000000000",
    INITP_03 => X"00000000000000000000000000000000000000000000",
-- Address 512 to 767
    INITP_04 => X"00000000000000000000000000000000000000000000",
    INITP_05 => X"00000000000000000000000000000000000000000000",
-- Address 768 to 1023
    INITP_06 => X"00000000000000000000000000000000000000000000",
    INITP_07 => X"00000000000000000000000000000000000000000000")
port map (
    DO => DO, -- 16-bit Data Output
    DOP => DOP, -- 2-bit parity Output
    ADDR => ADDR, -- 10-bit Address Input
    CLK => CLK, -- Clock
    DI => DI, -- 16-bit Data Input
    DIP => DIP, -- 2-bit parity Input
    EN => EN, -- RAM Enable Input
    SSR => SSR, -- Synchronous Set/Reset Input
    WE => WE -- Write Enable Input
);
-- End of RAMB16_S18_inst instantiation

```

В систему условных обозначений, используемую в описании интерфейса библиотечного примитива *RAMB16_S18*, входят идентификаторы входов и выходов, представленные при рассмотрении предыдущего шаблона. При этом следует обратить внимание на то, что в приведенной конструкции вместо одиночных информационных входа и выхода DI и DO применяются одноименные 16-разрядные шины данных. Кроме того, в шаблоне *1k x 16 + 2 Parity bits (RAMB16_S18)* присутствуют 2-разрядные дополнительные входная и выходная шины данных (DIP и DOP соответственно), которые предназначены для организации контроля четности. В отличие от предыдущего шаблона шина адреса библиотечного элемента *RAMB16_S18* содержит 10 двоичных разрядов.

На рис. 103 представлен условный графический образ элемента 1-портового ОЗУ с организацией 1024 слова \times 16 разрядов, для подготовки описания которого используется шаблон *1k x 16 + 2 Parity bits (RAMB16_S18)*.

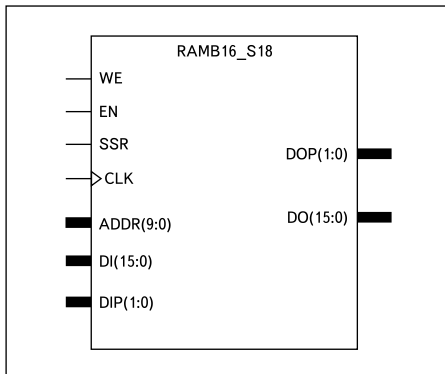


Рис. 103. Условный графический образ 1-портового ОЗУ (шаблон $1k \times 16 + 2$ Parity bits (*RAMB16_S18*))

2k x 8 + 1 Parity bit (RAMB16_S9) содержит конструкцию, предназначенную для подготовки описания варианта конфигурирования модуля блочной памяти кристаллов программируемой логики семейства Spartan-3 в виде элемента 1-портового ОЗУ с организацией 2048 слов \times 8 разрядов. Элементы оперативной памяти, формируемые с помощью этого шаблона, дают возможность выполнения операций с использованием контроля четности данных.

```
-- RAMB16_S9 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAMB16_S9_inst) and/or the port
declarations
-- code : after the ">" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs
to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
```

```
--
Library UNISIM;
use UNISIM.vcomponents.all;
-- <-----Cut code below this line and paste into the architecture
body---->
--
-- RAMB16_S9: 2k x 8 + 1 Parity bit Single-Port RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
RAMB16_S9_inst : RAMB16_S9
generic map (
  INIT => X"000", -- Value of output RAM registers at
startup
  SRVAL => X"000", -- Output value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST,
READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the initial contents
of the RAM
-- Address 0 to 511
INIT_00 => X"00000000000000000000000000000000000000000000",
INIT_01 => X"00000000000000000000000000000000000000000000",
INIT_02 => X"00000000000000000000000000000000000000000000",
...
INIT_0E => X"00000000000000000000000000000000000000000000",
INIT_0F => X"00000000000000000000000000000000000000000000",
-- Address 512 to 1023
INIT_10 => X"00000000000000000000000000000000000000000000",
INIT_11 => X"00000000000000000000000000000000000000000000",
INIT_12 => X"00000000000000000000000000000000000000000000",
...
INIT_1E => X"00000000000000000000000000000000000000000000",
INIT_1F => X"00000000000000000000000000000000000000000000",
-- Address 1024 to 1535
INIT_20 => X"00000000000000000000000000000000000000000000",
INIT_21 => X"00000000000000000000000000000000000000000000",
INIT_22 => X"00000000000000000000000000000000000000000000",
...
INIT_2E => X"00000000000000000000000000000000000000000000",
INIT_2F => X"00000000000000000000000000000000000000000000",
-- Address 1536 to 2047
INIT_30 => X"00000000000000000000000000000000000000000000",
INIT_31 => X"00000000000000000000000000000000000000000000",
INIT_32 => X"00000000000000000000000000000000000000000000",
...
INIT_3E => X"00000000000000000000000000000000000000000000",
INIT_3F => X"00000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 511
INITP_00 => X"00000000000000000000000000000000000000000000",
INITP_01 => X"00000000000000000000000000000000000000000000",
-- Address 512 to 1023
INITP_02 => X"00000000000000000000000000000000000000000000",
INITP_03 => X"00000000000000000000000000000000000000000000",
-- Address 1024 to 1535
INITP_04 => X"00000000000000000000000000000000000000000000",
INITP_05 => X"00000000000000000000000000000000000000000000",
-- Address 1536 to 2047
INITP_06 => X"00000000000000000000000000000000000000000000",
INITP_07 => X"00000000000000000000000000000000000000000000")
port map (
  DO => DO, -- 8-bit Data Output
  DOP => DOP, -- 1-bit parity Output
  ADDR => ADDR, -- 11-bit Address Input
  CLK => CLK, -- Clock
  DI => DI, -- 8-bit Data Input
  DIP => DIP, -- 1-bit parity Input
  EN => EN, -- RAM Enable Input
  SSR => SSR, -- Synchronous Set/Reset Input
  WE => WE -- Write Enable Input
);
-- End of RAMB16_S9_inst instantiation
```

Основные отличия приведенной конструкции от предыдущего шаблона проявляются в разрядности входных и выходных шин данных, а также шины адреса. Основные входная и выходная шины данных содержат 8 двоичных разрядов, а шина адреса — 11 разрядов. Для организации контроля четности предусмотрены дополнительные вход и выход данных.

На рис. 104 приведен условный графический образ 1-портового оперативного запоминающего устройства с организацией 2048 слов \times 8 разрядов и поддержкой выполнения операций с использованием контроля четности, формируемого с помощью шаблона *2k x 8 + 1 Parity bit (RAMB16_S9)*.

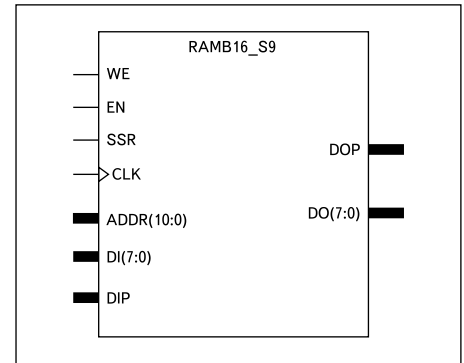


Рис. 104. Условный графический образ 1-портового ОЗУ (шаблон $2k \times 8 + 1$ Parity bit (*RAMB16_S9*))

4k x 4 (RAMB16_S4) является образцом VHDL-описания варианта конфигурирования модуля блочной памяти кристаллов программируемой логики семейства Spartan-3 в виде элемента 1-портового ОЗУ с организацией 4096 слов \times 4 разряда.

```
-- RAMB16_S4 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAMB16_S4_inst) and/or the port
declarations
-- code : after the ">" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
```

```
Library UNISIM;
use UNISIM.vcomponents.all;
-- <-----Cut code below this line and paste into the architecture
body---->
--
-- RAMB16_S4: 4k x 4 Single-Port RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
RAMB16_S4_inst : RAMB16_S4
generic map (
  INIT => X"0", -- Value of output RAM registers at startup
  SRVAL => X"0", -- Output value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST,
READ_FIRST or NO_CHANGE
-- The following INIT_xx declarations specify the initial contents
of the RAM
-- Address 0 to 1023
INIT_00 => X"00000000000000000000000000000000000000000000",
INIT_01 => X"00000000000000000000000000000000000000000000",
INIT_02 => X"00000000000000000000000000000000000000000000",
...
INIT_0D => X"00000000000000000000000000000000000000000000",
INIT_0E => X"00000000000000000000000000000000000000000000",
INIT_0F => X"00000000000000000000000000000000000000000000",
-- Address 1024 to 2047
INIT_10 => X"00000000000000000000000000000000000000000000",
INIT_11 => X"00000000000000000000000000000000000000000000",
INIT_12 => X"00000000000000000000000000000000000000000000",
...
INIT_1D => X"00000000000000000000000000000000000000000000",
INIT_1E => X"00000000000000000000000000000000000000000000",
INIT_1F => X"00000000000000000000000000000000000000000000",
-- Address 2048 to 3071
INIT_20 => X"00000000000000000000000000000000000000000000",
INIT_21 => X"00000000000000000000000000000000000000000000",
INIT_22 => X"00000000000000000000000000000000000000000000",
...
INIT_2D => X"00000000000000000000000000000000000000000000",
INIT_2E => X"00000000000000000000000000000000000000000000",
INIT_2F => X"00000000000000000000000000000000000000000000",
-- Address 3072 to 4095
```

```

INIT_30 => X"00000000000000000000000000000000",
INIT_31 => X"00000000000000000000000000000000",
INIT_32 => X"00000000000000000000000000000000",
...
INIT_3D => X"00000000000000000000000000000000",
INIT_3E => X"00000000000000000000000000000000",
INIT_3F => X"00000000000000000000000000000000")
port map (
  DO => DO, -- 4-bit Data Output
  ADDR => ADDR, -- 12-bit Address Input
  CLK => CLK, -- Clock
  DI => DI, -- 4-bit Data Input
  EN => EN, -- RAM Enable Input
  SSR => SSR, -- Synchronous Set/Reset Input
  WE => WE -- Write Enable Input
);
-- End of RAMB16_S4_inst instantiation

```

Основу этого шаблона образует оператор создания экземпляра библиотечного элемента **RAMB16_S4**, который соответствует указанному выше варианту конфигурирования модуля блочной памяти кристаллов программируемой логики семейства Spartan-3. При практическом использовании этой конструкции нужно обратить внимание на то, что разрядность входной и выходной шин данных формируемого элемента оперативной памяти составляет 4 бита, а разрядность шины адреса — 12 бит.

Условный графический образ элемента 1-портового ОЗУ с организацией 4096 слов × 4 разряда, описание которого формируется с помощью шаблона **4k x 4 (RAMB16_S4)**, показан на рис. 105.

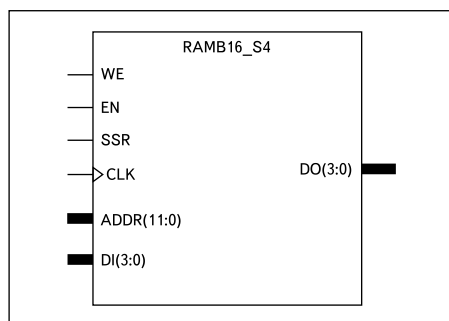


Рис. 105. Условный графический образ 1-портового ОЗУ (шаблон 4k x 4 (RAMB16_S4))

512 x 32 + 4 Parity bits (RAMB16_S36) представляет шаблон VHDL-описания 1-портового оперативного запоминающего устройства с организацией 512 слов × 32 разряда и поддержкой контроля четности записываемых и считываемых данных, которое предназначено для реализации на базе ресурсов блочной памяти кристаллов программируемой логики семейства Spartan-3.

```

-- RAMB16_S36 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAMB16_S36_inst) and/or the port
declarations
-- code : after the ">" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used

```

```

-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- RAMB16_S36: 512 x 32 + 4 Parity bits Single-Port RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
RAMB16_S36_inst : RAMB16_S36
generic map (
  INIT => X"0000000000", -- Value of output RAM registers at
startup
  SRVAL => X"0000000000", -- Output value upon SSR
assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST,
  READ_FIRST or NO_CHANGE
  -- The following INIT_xx declarations specify the initial contents
of the RAM
  -- Address 0 to 127
  INIT_00 => X"00000000000000000000000000000000",
  INIT_01 => X"00000000000000000000000000000000",
  INIT_02 => X"00000000000000000000000000000000",
  ...
  INIT_0E => X"00000000000000000000000000000000",
  INIT_0F => X"00000000000000000000000000000000",
  -- Address 128 to 255
  INIT_10 => X"00000000000000000000000000000000",
  INIT_11 => X"00000000000000000000000000000000",
  INIT_12 => X"00000000000000000000000000000000",
  ...
  INIT_1E => X"00000000000000000000000000000000",
  INIT_1F => X"00000000000000000000000000000000",
  -- Address 256 to 383
  INIT_20 => X"00000000000000000000000000000000",
  INIT_21 => X"00000000000000000000000000000000",
  INIT_22 => X"00000000000000000000000000000000",
  ...
  INIT_2E => X"00000000000000000000000000000000",
  INIT_2F => X"00000000000000000000000000000000",
  -- Address 384 to 511
  INIT_30 => X"00000000000000000000000000000000",
  INIT_31 => X"00000000000000000000000000000000",
  INIT_32 => X"00000000000000000000000000000000",
  ...
  INIT_3E => X"00000000000000000000000000000000",
  INIT_3F => X"00000000000000000000000000000000",
  -- The next set of INITP_xx are for the parity bits
  -- Address 0 to 127
  INITP_00 => X"00000000000000000000000000000000",
  INITP_01 => X"00000000000000000000000000000000",
  -- Address 128 to 255
  INITP_02 => X"00000000000000000000000000000000",
  INITP_03 => X"00000000000000000000000000000000",
  -- Address 256 to 383
  INITP_04 => X"00000000000000000000000000000000",
  INITP_05 => X"00000000000000000000000000000000",
  -- Address 384 to 511
  INITP_06 => X"00000000000000000000000000000000",
  INITP_07 => X"00000000000000000000000000000000")
port map (
  DO => DO, -- 32-bit Data Output
  DOP => DOP, -- 4-bit parity Output
  ADDR => ADDR, -- 9-bit Address Input
  CLK => CLK, -- Clock
  DI => DI, -- 32-bit Data Input
  DIP => DIP, -- 4-bit parity Input
  EN => EN, -- RAM Enable Input
  SSR => SSR, -- Synchronous Set/Reset Input
  WE => WE -- Write Enable Input
);
-- End of RAMB16_S36_inst instantiation

```

Основой этого шаблона является экземпляр библиотечного примитива **RAMB16_S36**, который представляет соответствующий вариант конфигурирования модуля блочной памяти ПЛИС семейства Spartan-3. Система условных обозначений входов и выходов, используемая в представленной конструкции, отличается от совокупности идентификаторов предыдущих шаблонов только разрядностью адресной и информационных шин. В шаблоне **512 x 32 + 4 Parity bits (RAMB16_S36)** входная и выходная основные

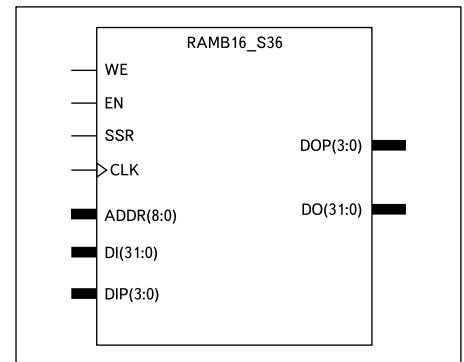


Рис. 106. Условный графический образ элемента 1-портового ОЗУ (шаблон 512 x 32 + 4 Parity bits (RAMB16_S36))

шины данных содержат 32 двоичных разряда, а шина адреса — 9 разрядов. В составе интерфейса присутствуют также дополнительные 4-разрядные шины данных (входная и выходная), используемые при выполнении операций с применением контроля четности.

На рис. 106 приведен условный графический образ элемента 1-портовой оперативной памяти с организацией 512 слов × 32 разряда и поддержкой контроля четности, описание которого формируется с помощью шаблона **512 x 32 + 4 Parity bits (RAMB16_S36)**.

8k x 2 (RAMB16_S2) включает в себя конструкцию, предназначенную для подготовки VHDL-описания варианта конфигурирования модуля блочной памяти в кристаллах программируемой логики семейства Spartan-3, в виде элемента 1-портового ОЗУ емкостью 16 384 бита с организацией 8192 слова × 2 разряда.

```

-- RAMB16_S2 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAMB16_S2_inst) and/or the port
declarations
-- code : after the ">" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- RAMB16_S2: 8k x 2 Single-Port RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
RAMB16_S2_inst : RAMB16_S2
generic map (
  INIT => X"0", -- Value of output RAM registers at startup
  SRVAL => X"0", -- Output value upon SSR assertion
  WRITE_MODE => "WRITE_FIRST", -- WRITE_FIRST,
  READ_FIRST or NO_CHANGE
  -- The following INITP_xx declarations specify the initial contents
of the RAM
  -- Address 0 to 2047
  INIT_00 => X"00000000000000000000000000000000",
  INIT_01 => X"00000000000000000000000000000000",

```

```

INIT_02 => X"00000000000000000000000000000000",
...
INIT_0D => X"00000000000000000000000000000000",
INIT_0E => X"00000000000000000000000000000000",
INIT_0F => X"00000000000000000000000000000000",
-- Address 2048 to 4095
INIT_10 => X"00000000000000000000000000000000",
INIT_11 => X"00000000000000000000000000000000",
INIT_12 => X"00000000000000000000000000000000",
...
INIT_1D => X"00000000000000000000000000000000",
INIT_1E => X"00000000000000000000000000000000",
INIT_1F => X"00000000000000000000000000000000",
-- Address 4096 to 6143
INIT_20 => X"00000000000000000000000000000000",
INIT_21 => X"00000000000000000000000000000000",
INIT_22 => X"00000000000000000000000000000000",
...
INIT_2D => X"00000000000000000000000000000000",
INIT_2E => X"00000000000000000000000000000000",
INIT_2F => X"00000000000000000000000000000000",
-- Address 6144 to 8191
INIT_30 => X"00000000000000000000000000000000",
INIT_31 => X"00000000000000000000000000000000",
INIT_32 => X"00000000000000000000000000000000",
...
INIT_3D => X"00000000000000000000000000000000",
INIT_3E => X"00000000000000000000000000000000",
INIT_3F => X"00000000000000000000000000000000")
port map (
  DO => DO, -- 2-bit Data Output
  ADDR => ADDR, -- 13-bit Address Input
  CLK => CLK, -- Clock
  DI => DI, -- 2-bit Data Input
  EN => EN, -- RAM Enable Input
  SSR => SSR, -- Synchronous Set/Reset Input
  WE => WE -- Write Enable Input
);
-- End of RAMB16_S2_inst instantiation

```

Представленный шаблон выполнен на основе библиотечного примитива *RAMB16_S2*. В приведенной конструкции в составе интерфейса присутствуют 2-разрядные шины данных и 13-разрядная шина адреса.

Условный графический образ элемента 1-портового ОЗУ с организацией 8192 слова × 2 разряда, описание которого создается с помощью шаблона *8k x 2 (RAMB16_S2)* для последующей реализации на основе модуля блочной памяти Block RAM ПЛИС семейства Spartan-3, изображен на рис. 107.

В каталоге *Distributed RAM*, входящем в состав подраздела *RAM/ROM* (Рис. 51. Кит. 2010. № 8), сосредоточены образцы описаний элементов оперативных запоминающих устройств различного типа, которые реализуются на базе ресурсов распределенной памяти ПЛИС семейства Spartan-3. В состав всех секций типа SLICEM кристаллов программируемой логики этого семейства входят две 4-входовые таблицы преобразования LUT (Look-Up Table), каждая из которых

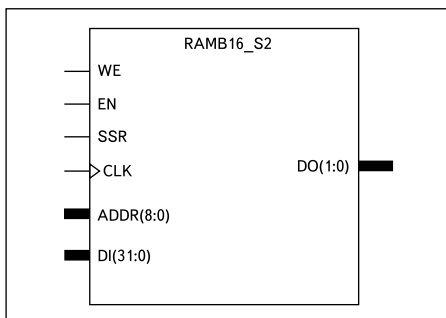


Рис. 107. Условный графический образ 2-портового ОЗУ (шаблон *8k x 2 (RAMB16_S2)*)

может конфигурироваться в виде элемента ОЗУ или ПЗУ с информационной емкостью 16 бит [1, 19]. Таким образом, одна секция указанного типа позволяет реализовать элемент запоминающего устройства емкостью 32 бита с различной разрядностью данных (32 слова × 1 разряд или 16 слов × 2 разряда).

Каталог *Distributed RAM* содержит две папки — *Dual-Port* и *Single-Port*. В папке *Dual-Port* находятся шаблоны описаний элементов 2-портовых распределенных оперативных запоминающих устройств с организацией 16 слов × 1 разряд. Папка *Single-Port* объединяет в себе образцы описаний элементов 1-портовой распределенной оперативной памяти с различной информационной емкостью, которые сгруппированы в подразделы в соответствии с количеством ячеек.

16 x 1 negedge write (RAM16X1D_1) содержит шаблон VHDL-описания элемента 2-портового ОЗУ с информационной емкостью 16 бит, синтезируемого в виде распределенной памяти в кристаллах программируемой логики семейства Spartan-3. Основу этой конструкции образует оператор создания экземпляра библиотечного примитива *RAM16X1D_1*.

```

-- RAM16X1D_1: In order to incorporate this function into the design,
-- VHDL: the following instance declaration needs to be placed
-- instance: in the architecture body of the design code. The
-- declaration: instance name (RAM16X1D_1_inst) and/or the port
-- declarations
-- code: after the "=" assignment may be changed to properly
--: reference and connect this function to the design.
--: All inputs and outputs must be connected.
--
-- Library: In addition to adding the instance declaration, a use
-- declaration: statement for the UNISIM.vcomponents library needs to be
-- for: added before the entity declaration. This library
-- Xilinx: contains the component declarations for all Xilinx
-- primitives: primitives and points to the models that will be used
--: for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
-- body----->
--
-- RAM16X1D_1: 16 x 1 negative edge write, asynchronous read dual-
-- port distributed RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
RAM16X1D_1_inst: RAM16X1D_1
generic map (
  INIT => X"0000"
)
port map (
  DPO => DPO, -- Read-only 1-bit data output for DPRA
  SPO => SPO, -- R/W 1-bit data output for A0-A3
  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
  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
  WCLK => WCLK, -- Write clock input
  WE => WE -- Write enable input
);
-- End of RAM16X1D_1_inst instantiation

```

В описании интерфейса библиотечного примитива *RAM16X1D_1* используется следующая система условных обозначений входов и выходов:

- SPO — информационный выход первого порта ОЗУ;
- DPO — информационный выход второго порта элемента оперативной памяти;
- A0–A3 — адресные входы первого порта;
- D — информационный вход данных;
- DPRA0–DPRA3 — адресные входы второго порта;
- WCLK — вход сигнала синхронизации;
- WE — вход разрешения записи.

В элементах 2-портовых запоминающих устройств, формируемых с помощью шаблона *16 x 1 negedge write (RAM16X1D_1)*, первый порт, который включает в себя адресные входы A0–A3 и выход SPO, позволяет выполнять как операции чтения, так и записи входных данных. Второй порт (с адресными входами DPRA0–DPRA3 и выходом DPO) предназначен только для чтения информации, хранящейся в ОЗУ. Загрузка входных данных в ячейки элемента 2-портовой распределенной оперативной памяти производится по спаду сигнала синхронизации при наличии высокого логического уровня на входе разрешения записи WE. Для инициализации содержимого создаваемых 2-портовых ОЗУ предусмотрен настраиваемый параметр INIT. Значение этого параметра задается в виде 4-разрядного шестнадцатеричного числа.

На рис. 108 показан условный графический образ элемента 2-портовой распределенной оперативной памяти с организацией 16 слов × 1 разряд, для подготовки описания которого используется шаблон *16 x 1 negedge write (RAM16X1D_1)*.

16 x 1 posedge write (RAM16X1D) представляет собой образец описания варианта конфигурирования таблицы преобразования LUT кристаллов программируемой логики семейства Spartan-3 в виде элемента 2-портового оперативного запоминающего устрой-

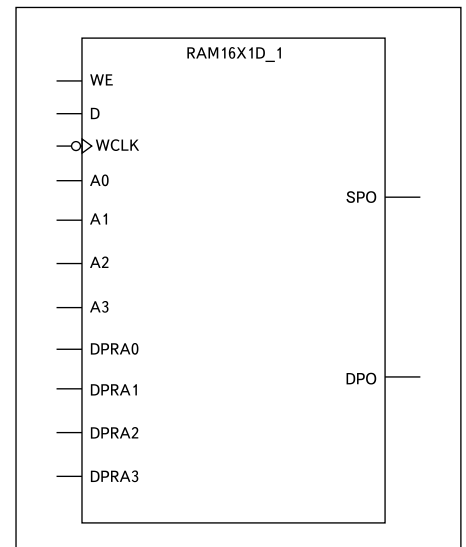


Рис. 108. Условный графический образ распределенного 2-портового ОЗУ (шаблон *16 x 1 negedge write (RAM16X1D_1)*)

ства с информационной емкостью 16 бит и организацией 16 слов \times 1 разряд. Основой этого шаблона является экземпляр библиотечного примитива *RAM16X1D*.

```
-- : In order to incorporate this function into the design,
-- VHDL: the following instance declaration needs to be placed
-- instance: in the architecture body of the design code. The
-- declaration: instance name (RAM16X1D_inst) and/or the port
declarations
-- code: after the "=" assignment maybe changed to properly
--: reference and connect this function to the design.
--: All inputs and outputs must be connected.
--
-- Library: In addition to adding the instance declaration, a use
-- declaration: statement for the UNISIM.vcomponents library needs
to be
-- for: added before the entity declaration. This library
-- Xilinx: contains the component declarations for all Xilinx
-- primitives: primitives and points to the models that will be used
--: for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- RAM16X1D: 16 x 1 positive edge write, asynchronous read dual-
port distributed RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
RAM16X1D_inst : RAM16X1D
generic map (
    INIT => X"0000"
)
port map (
    DPO => DPO, -- Read-only 1-bit data output for DPRA
    SPO => SPO, -- R/W 1-bit data output for A0-A3
    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
    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
    WCLK => WCLK, -- Write clock input
    WE => WE -- Write enable input
);
-- End of RAM16X1D_inst instantiation
```

В составе системы условных обозначений интерфейсных портов в представленной конструкции используются те же идентификаторы входов и выходов, что и в предыдущем шаблоне. Но при этом необходимо учитывать, что в отличие от библиотечного примитива *RAM16X1D_1*, используемого в составе предыдущего шаблона, в элементах ОЗУ, формируемых на основе компонента *RAM16X1D*, загрузка данных в ячейки памяти производится по фронту тактового сигнала при наличии активного уровня напряжения на входе разрешения записи WE.

На рис. 109 приведен условный графический образ элемента 2-портовой распределенной оперативной памяти, для создания описания которого используется шаблон *16 x 1 posedge write (RAM16X1D)*.

16 x 1 negedge write (RAM16X1S_1) является шаблоном описания элемента 1-портового оперативного запоминающего устройства информационной емкостью 16 бит с организацией 16 слов \times 1 разряд, синтезируемого на базе распределенной памяти кристаллов программируемой логики семейства Spartan-3.

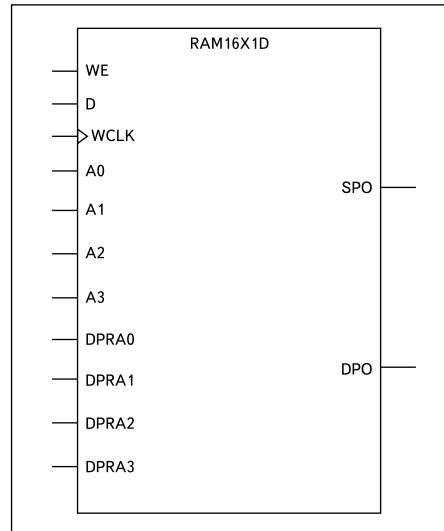


Рис. 109. Условный графический образ элемента 2-портового распределенного ОЗУ (шаблон 16 x 1 posedge write (RAM16X1D))

```
-- RAM16X1S_1 : In order to incorporate this function into the design,
-- VHDL: the following instance declaration needs to be placed
-- instance: in the architecture body of the design code. The
-- declaration: instance name (RAM16X1S_1_inst) and/or the port
declarations
-- code: after the "=" assignment maybe changed to properly
--: reference and connect this function to the design.
--: All inputs and outputs must be connected.
--
-- Library: In addition to adding the instance declaration, a use
-- declaration: statement for the UNISIM.vcomponents library needs to be
-- for: added before the entity declaration. This library
-- Xilinx: contains the component declarations for all Xilinx
-- primitives: primitives and points to the models that will be used
--: for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- RAM16X1S_1: 16 x 1 negedge write distributed (LUT) RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
RAM16X1S_1_inst : RAM16X1S_1
generic map (
    INIT => X"0000"
)
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
    D => D, -- RAM data input
    WCLK => WCLK, -- Write clock input
    WE => WE -- Write enable input
);
-- End of RAM16X1S_1_inst instantiation
```

В основе представленного шаблона лежит экземпляр библиотечного примитива *RAM16X1S_1*. Система условных обозначений, используемая в описании интерфейса этого компонента, отличается от совокупности идентификаторов входов и выходов библиотечного примитива *RAM16X1D_1* только наименованием выхода ОЗУ (O). Входные данные, поступающие на вход D, заносятся в ячейку элемента оперативной памяти по спаду тактового сигнала на входе WCLK

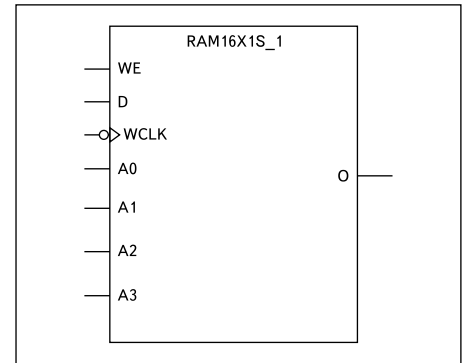


Рис. 110. Условный графический образ распределенного 1-портового ОЗУ (шаблон 16 x 1 negedge write (RAM16X1S_1))

при подаче высокого логического уровня напряжения на вход разрешения записи WE.

Условный графический образ элемента распределенного 1-портового ОЗУ с организацией 16 слов \times 1 разряд, описание которого формируется с помощью шаблона *16 x 1 negedge write (RAM16X1S_1)*, показан на рис. 110.

16 x 1 posedge write (RAM16X1S) представляет образец VHDL-описания варианта конфигурирования таблицы преобразования LUT ПЛИС семейства Spartan-3 в форме элемента 1-портового ОЗУ с информационной емкостью 16 бит и организацией 16 слов \times 1 разряд. Основой этого шаблона является оператор создания экземпляра библиотечного примитива *RAM16X1S*.

```
-- RAM16X1S: In order to incorporate this function into the design,
-- VHDL: the following instance declaration needs to be placed
-- instance: in the architecture body of the design code. The
-- declaration: instance name (RAM16X1S_inst) and/or the port
declarations
-- code: after the "=" assignment maybe changed to properly
--: reference and connect this function to the design.
--: All inputs and outputs must be connected.
--
-- Library: In addition to adding the instance declaration, a use
-- declaration: statement for the UNISIM.vcomponents library needs
to be
-- for: added before the entity declaration. This library
-- Xilinx: contains the component declarations for all Xilinx
-- primitives: primitives and points to the models that will be used
--: for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- RAM16X1S: 16 x 1 posedge write distributed (LUT) RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
RAM16X1S_inst : RAM16X1S
generic map (
    INIT => X"0000"
)
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
    D => D, -- RAM data input
    WCLK => WCLK, -- Write clock input
    WE => WE -- Write enable input
);
-- End of RAM16X1S_inst instantiation
```

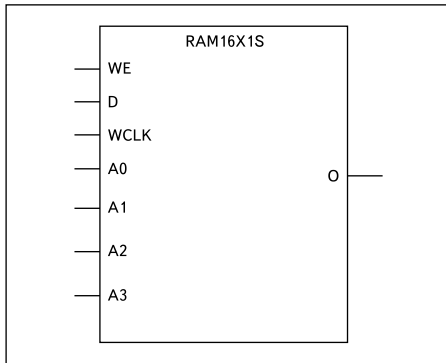


Рис. 111. Условный графический образ элемента 1-портового распределенного ОЗУ (шаблон 16 x 1 posedge write (RAM16X1S))

Элементы оперативной памяти, формируемые на основе библиотечного примитива *RAM16X1S*, отличаются от ОЗУ, выполняемых на базе компонента *RAM16X1S_1*, рассмотренного выше, только тем, что запись данных осуществляется по фронту сигнала синхронизации.

На рис. 111 изображен условный графический образ элемента распределенного 1-портового ОЗУ с информационной емкостью 16 бит и организацией 16 слов × 1 разряд, для подготовки описания которого используется шаблон *16 x 1 posedge write (RAM16X1S)*.

16 x 2 posedge write (RAM16X2S) включает в себя конструкцию, предназначенную для подготовки описания элемента 1-портового ОЗУ с информационной емкостью 32 бита и организацией 16 слов × 2 разряда, синтезируемого на основе ресурсов распределенной памяти кристаллов программируемой логики семейства Spartan-3. В качестве основы этого шаблона используется экземпляр библиотечного примитива *RAM16X2S*.

```
-- RAM16X2S : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAM16X2S_inst) and/or the port
declarations
-- code : after the "=" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs
to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- RAM16X2S: 16 x 2 posedge write distributed (LUT) RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
RAM16X2S_inst : RAM16X2S
generic map (
  INIT_00 => X"0000", -- INIT for bit 0 of RAM
  INIT_01 => X"0000") -- 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
  D0 => D0, -- RAM data[0] input
  D1 => D1, -- RAM data[1] input
  WCLK => WCLK, -- Write clock input
  WE => WE -- Write enable input
);
-- End of RAM16X2S_inst instantiation
```

В систему условных обозначений интерфейсных портов библиотечного примитива *RAM16X2S* кроме наименований входов и выходов, представленных выше при рассмотрении шаблонов описания элементов 2-портовых распределенных ОЗУ, входят следующие идентификаторы:

- O0 — выход нулевого разряда данных;
- O1 — выход первого разряда данных;
- D0 — информационный вход нулевого разряда данных;
- D1 — информационный вход первого разряда данных.

Для инициализации содержимого ячеек оперативной памяти, формируемых с помощью шаблона *16 x 2 posedge write (RAM16X2S)*, применяются настраиваемые параметры INIT_00 и INIT_01, значения которых задаются в виде 4-разрядных шестнадцатеричных чисел. При этом параметр INIT_00 определяет содержимое ячеек, соответствующих нулевому разряду данных, а параметр INIT_01 указывает значения, записываемые после завершения конфигурирования ПЛИС в ячейки, соответствующие первому разряду данных.

Условный графический образ элемента 1-портового ОЗУ с информационной емкостью 32 бита и организацией 16 слов × 2 разряда, описание которого создается на основе шаблона *16 x 2 posedge write (RAM16X2S)* для последующей реализации на базе таблиц преобразования LUT кристаллов программируемой логики семейства Spartan-3, показан на рис. 112.

32 x 1 negedge write (RAM32X1S_1) содержит образец описания элемента 1-портового ОЗУ с информационной емкостью 32 бита и организацией 32 слова × 1 разряд, выполняемого на основе ресурсов распределенной памяти ПЛИС семейства Spartan-3. Запись данных в оперативную память, формируемую на основе этого шаблона, производится по спаду тактового сигнала.

```
-- RAM32X1S_1 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAM32X1S_1_inst) and/or the port
declarations
-- code : after the "=" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
```

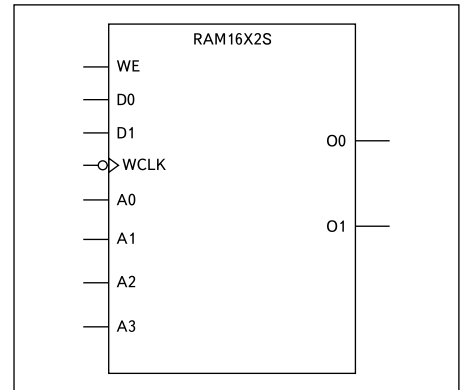


Рис. 112. Условный графический образ элемента 1-портового распределенного ОЗУ (шаблона 16 x 2 posedge write (RAM16X2S))

```
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
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
```

В отличие от предыдущих шаблонов описаний элементов распределенной оперативной памяти в составе интерфейса библиотечного примитива *RAM32X1S_1*, образующего основу представленной конструкции, присутствует дополнительный адресный вход A4 в соответствии с емкостью этого компонента. При практическом использовании шаблона *32 x 1 negedge write (RAM32X1S_1)* следует обратить внимание на то, что значения настраиваемого параметра INIT, определяющего начальное содержимое ячеек оперативной памяти, задаются в форме 8-разрядного шестнадцатеричного числа.

На рис. 113 изображен условный графический образ элемента 1-портового ОЗУ с информационной емкостью 32 бита и организацией 32 слова × 1 разряд, реализуемого на основе таблиц преобразования LUT кристаллов программируемой логики семейства Spartan-3.

32 x 1 posedge write (RAM32X1S) является шаблоном VHDL-описания элемента 1-портовой оперативной памяти с информационной емкостью 32 бита и организацией 32 слова × 1 разряд, предназначенного для

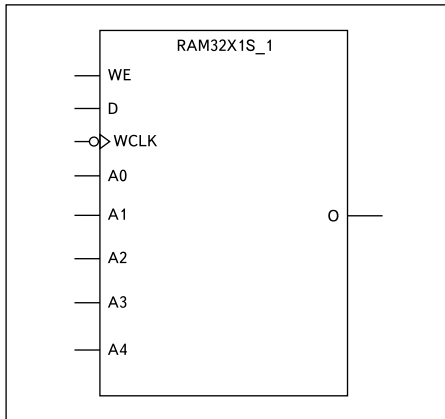


Рис. 113. Условный графический образ 1-портового ОЗУ (шаблон 32 x 1 negedge write (RAM32X1S_1))

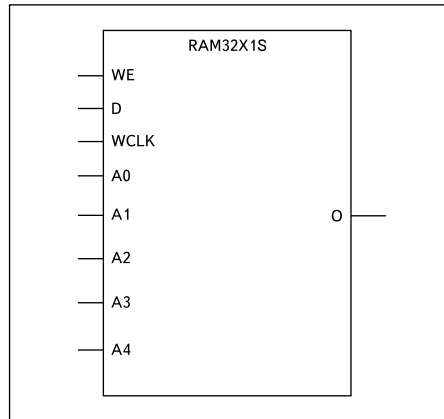


Рис. 114. Условный графический образ 1-портового распределенного ОЗУ (шаблон 32 x 1 posedge write (RAM32X1S))

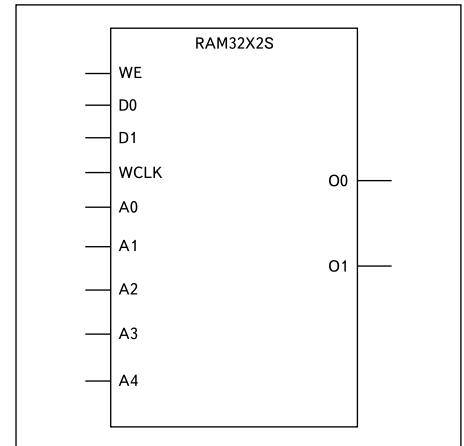


Рис. 115. Условный графический образ 1-портового ОЗУ (шаблон 32 x 2 posedge write (RAM32X2S))

реализации на базе таблиц преобразования ПЛИС семейства Spartan-3. Этот шаблон отличается от предыдущего только отсутствием инверсии тактового сигнала. Запись в элементы ОЗУ, описания которых создаются на основе библиотечного примитива **RAM32X1S**, осуществляется по фронту сигнала синхронизации.

32 x 2 posedge write (RAM32X2S) представляет собой образец описания элемента 1-портового оперативного запоминающего устройства с информационной емкостью 64 бита и организацией 32 слова × 2 разряда, реализуемого на базе ресурсов распределенной памяти кристаллов программируемой логики семейства Spartan-3.

с помощью двух предыдущих шаблонов, для инициализации содержимого ОЗУ, создаваемого на базе этого библиотечного компонента, необходимо определить значения двух настраиваемых параметров **INIT_00** и **INIT_01**. Значение каждого из этих параметров указывается в виде 8-разрядного шестнадцатеричного числа.

На рис. 115 показан условный графический образ элемента 1-портового ОЗУ с информационной емкостью 32 бита и организацией 32 слова × 2 разряда, реализуемого на основе таблиц преобразования LUT кристаллов программируемой логики семейства Spartan-3, для подготовки описания которого используется шаблон **32 x 2 posedge write (RAM32X2S)**.

64 x 1 negedge write (RAM64X1S_1) включает в себя шаблон VHDL-описания элемента 1-портового ОЗУ с информационной емкостью 64 бита и организацией 64 слова × 1 разряд, синтезируемого на базе таблиц преобразования ПЛИС семейства Spartan-3. В качестве основы этого шаблона используется экземпляр библиотечного примитива **RAM64X1S_1**. В элементах оперативной памяти, создаваемых на базе этого библиотечного компонента, запись данных в ячейки осуществляется по спаду тактового сигнала.

```
-- RAM32X1S : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAM32X1S_inst) and/or the port
declarations
-- code : after the "=" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
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
```

Условный графический образ элемента 1-портового распределенного ОЗУ с организацией 32 слова × 1 разряд, для формирования описания которого используется шаблон **32 x 1 posedge write (RAM32X1S)**, приведен на рис. 114.

```
-- RAM32X2S : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAM16X2S_inst) and/or the port
declarations
-- code : after the "=" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs
to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
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
```

Основу приведенной конструкции образуется оператор создания экземпляра библиотечного примитива **RAM32X2S**. В отличие от элементов оперативной памяти, формируемых

```
-- RAM64X1S_1 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAM64X1S_1_inst) and/or the port
declarations
-- code : after the "=" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
```

```

-- RAM64X1S_1: 64 x 1 negative edge write, asynchronous read
single-port distributed RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
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

```

Условный графический образ элемента 1-портового распределенного ОЗУ с организацией 64 слова × 1 разряд, формируемого с помощью шаблона *64 x 1 negedge write (RAM64X1S_1)*, изображен на рис. 116.

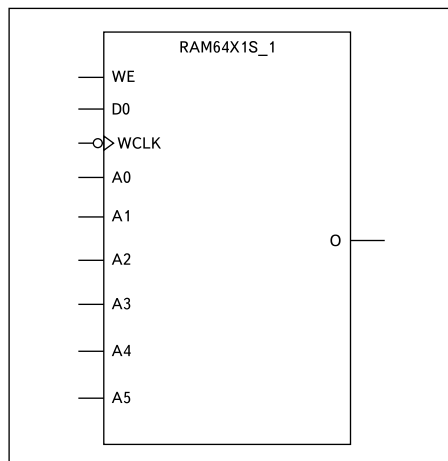


Рис. 116. Условный графический образ 1-портового ОЗУ (шаблон *64 x 1 negedge write (RAM64X1S_1)*)

64 x 1 posedge write (RAM64X1S) содержит образец описания элемента 1-портового ОЗУ с информационной емкостью 64 бита и организацией 64 слова × 1 разряд, предназначенного для реализации на базе ресурсов распределенной памяти кристаллов программируемой логики семейства Spartan-3. Этот шаблон выполнен на основе библиотечного примитива *RAM64X1S*, который отличается от компонента *RAM64X1S_1*, рассмотренного выше, записью данных по фронту сигнала синхронизации.

```

-- RAM64X1S : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (RAM64X1S_inst) and/or the port
declarations
-- code : after the "=" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs
to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.

```

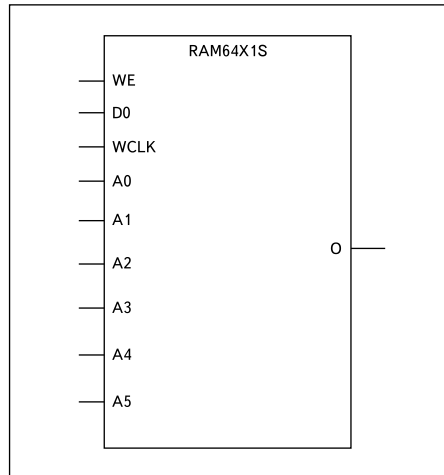


Рис. 117. Условный графический образ распределенного 1-портового ОЗУ (шаблон *64 x 1 posedge write (RAM64X1S)*)

```

--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- RAM64X1S: 64 x 1 positive edge write, asynchronous read single-
port distributed RAM
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
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

```

На рис. 117 представлен условный графический образ элемента 1-портового ОЗУ с информационной емкостью 64 бита, описание которого создается на основе шаблона *64 x 1 posedge write (RAM64X1S)*.

Продолжение следует

Литература

- Кузелин М. О., Кнышев Д. А., Зотов В. Ю. Современные семейства ПЛИС фирмы Xilinx. Справочное пособие. М.: Горячая линия – Телеком, 2004.
- Бибило П. Н. Основы языка VHDL. М.: Солон-Р, 2000.
- Бибило П. Н. Синтез логических схем с использованием языка VHDL. М.: Солон-Р, 2002.
- Уэйкерли Дж. Ф. Проектирование цифровых устройств. Т. 1. М.: Постмаркет, 2002.
- Поляков А. К. Языки VHDL и Verilog в проектировании цифровой аппаратуры. М.: Солон-Пресс, 2003.

- Зотов В. Инструментальный комплект Spartan-3 Starter Kit для практического освоения методов проектирования встраиваемых микропроцессорных систем на основе ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2005. № 7.
- Зотов В. Новый инструментальный комплект Spartan-3E Starter Kit для практического освоения методов проектирования встраиваемых микропроцессорных систем на основе ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2006. № 10.
- Зотов В. Новый инструментальный комплект Spartan-3A Starter Kit для практического освоения методов проектирования и отладки цифровых устройств с аппаратной и программной реализацией операций, реализуемых на основе ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2007. № 9.
- Зотов В. Новый инструментальный комплект от компании Avnet на основе ПЛИС FPGA семейства Spartan-3A фирмы Xilinx//Компоненты и технологии. 2008. № 8.
- Зотов В. Инструментальный модуль компании Avnet для отладки проектов встраиваемых систем, разрабатываемых на базе нового семейства ПЛИС FPGA фирмы Xilinx Virtex-5 FXT // Компоненты и технологии. 2008. № 9.
- Зотов В. Особенности архитектуры нового поколения высокопроизводительных ПЛИС FPGA фирмы Xilinx серии Virtex-6 // Компоненты и технологии. 2009. № 8.
- Зотов В. Особенности архитектуры нового поколения ПЛИС FPGA фирмы Xilinx серии Spartan-6 // Компоненты и технологии. 2009. № 9.
- Зотов В. Новое семейство высокопроизводительных ПЛИС с архитектурой FPGA фирмы Xilinx Virtex-6 HXT // Компоненты и технологии. 2010. № 1.
- IEEE Standard VHDL Language Reference Manual. IEEE Std 1076-2002.
- Суворова Е. А., Шейнин Ю. Е. Проектирование цифровых систем на VHDL. СПб.: БХВ-Петербург, 2003.
- Сергиенко А. М. VHDL для проектирования вычислительных устройств. Киев: ЧП «Корнейчук», ООО «ТИД «ДС», 2003.
- Зотов В. Проектирование цифровых устройств, реализуемых на базе ПЛИС FPGA фирмы Xilinx, с использованием средств CORE Generator // Компоненты и технологии. 2006. № 12. 2007. № 1.
- Зотов В. Разработка компонентов устройств цифровой обработки сигналов, реализуемых на базе аппаратных модулей DSP48E в ПЛИС FPGA серии Virtex-5, с помощью «мастера» Architecture Wizard САПР серии Xilinx ISE // Компоненты и технологии. 2008. № 12. 2009. № 1–7.
- Spartan-3 Generation FPGA User Guide. Xilinx, 2009.
- Spartan-3 FPGA Family: Complete Data Sheet. Xilinx, 2009.
- Spartan-3 Generation Configuration User Guide. Xilinx, 2009.