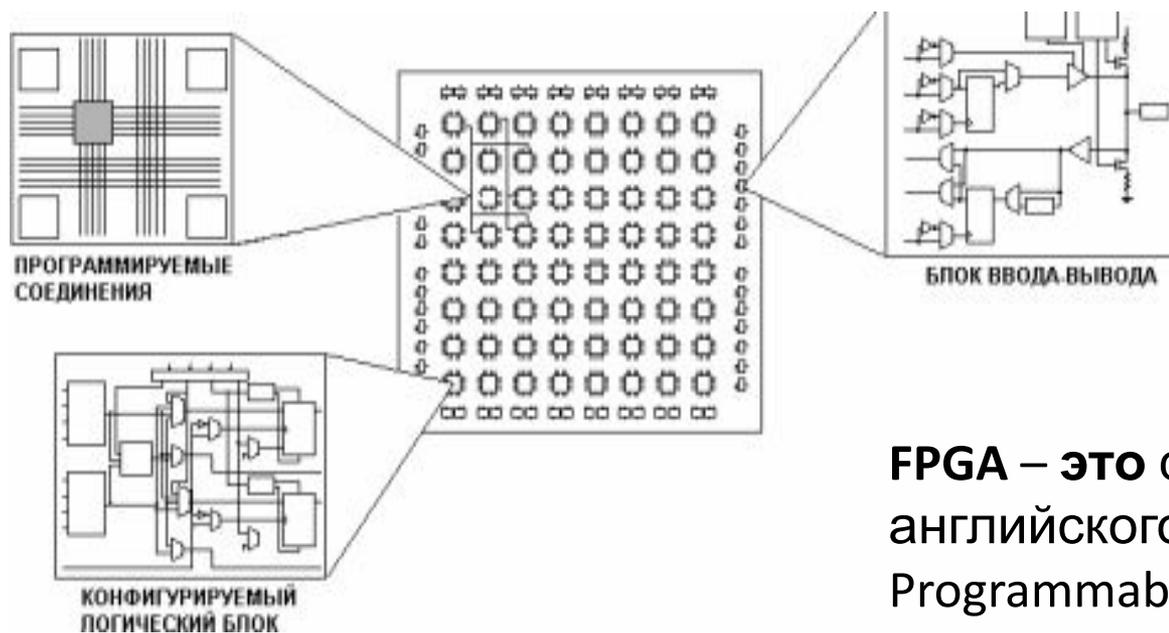


Лекция 5. Основные элементы ПЛИС



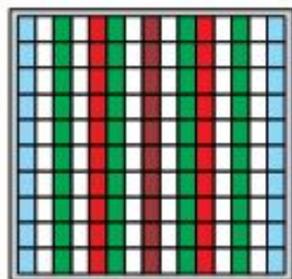
Основные элементы ПЛИС

В наиболее типичном варианте ПЛИС, выполненная по технологии FPGA, состоит из прямоугольной матрицы конфигурируемых логических блоков (Configurable Logic Blocks, CLB), блоков ввода-вывода (Input/Output Block, IOB) и дополнительных блоков. Между CLB располагаются программируемые трассировочные линии. Между матрицей CLB и блоками ввода-вывода имеются отдельные межсоединения, которые и обеспечивают подключение внешних сигналов.

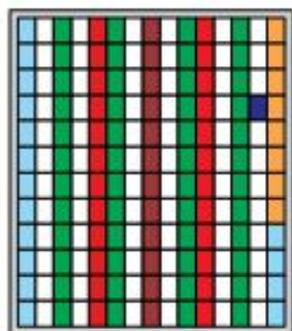


FPGA – это сокращение от английского словосочетания Field Programmable Gate Array

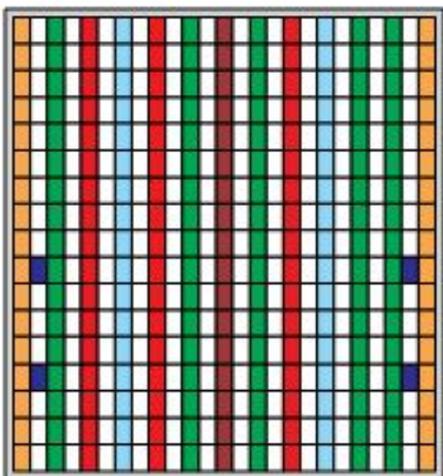
Основные элементы ПЛИС



Artix-7 FPGA



Kintex-7 FPGA



Virtex-7 FPGA

- Конфигурируемые логические блоки CLB
- Блочная память Block RAM
- Секции цифровой обработки сигналов DSP48E1
- Последовательные высокоскоростные приемопередатчики
- Комбинированные модули управления синхронизацией MMCM
- Модули интерфейса PCI Express
- Блоки ввода/вывода

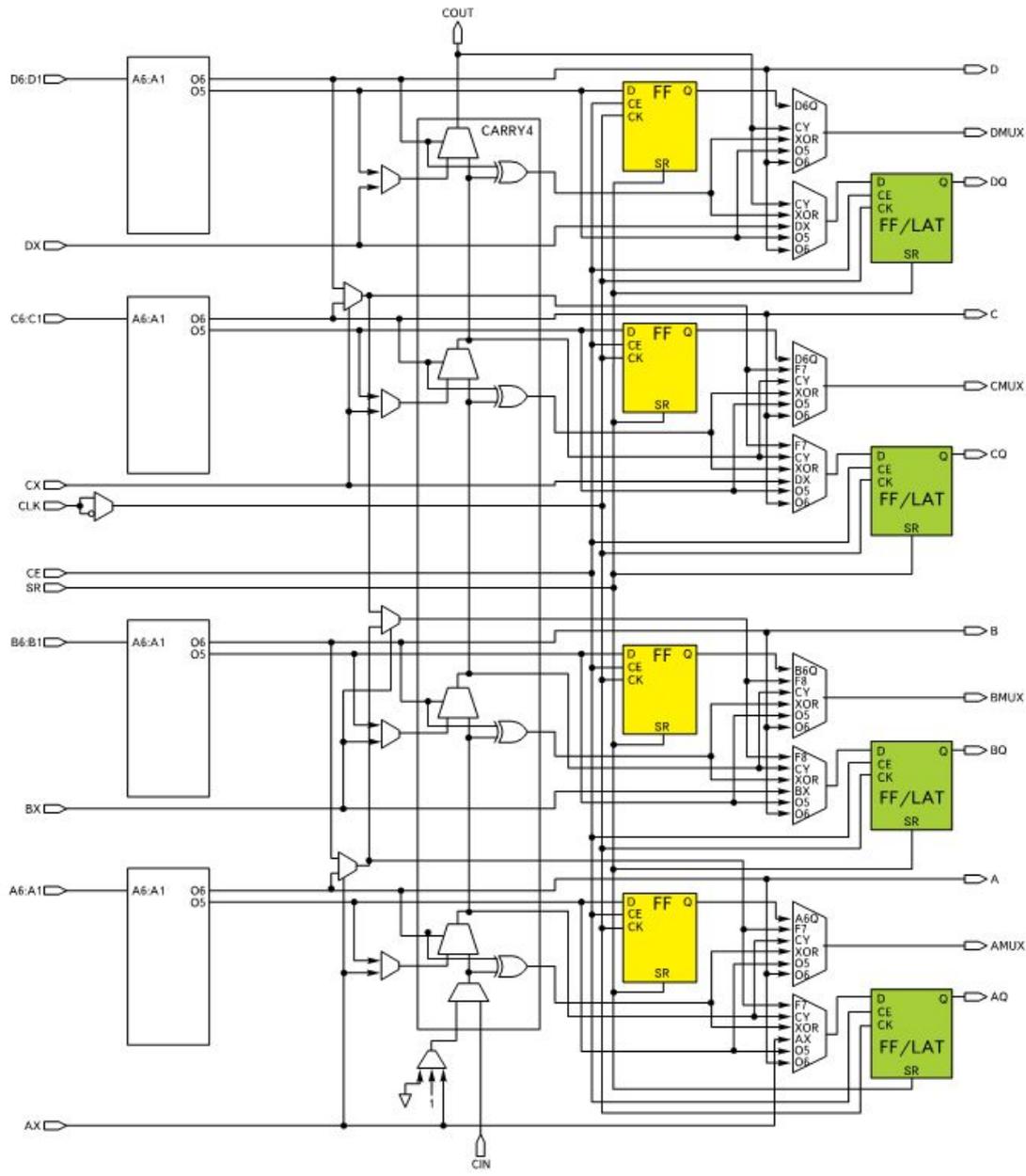
Современные ПЛИС

также содержат блоки памяти (BRAM),

секции цифровой обработки сигналов (DSP) и в отдельных случаях - аналого- цифровые и цифроаналоговые преобразователи.

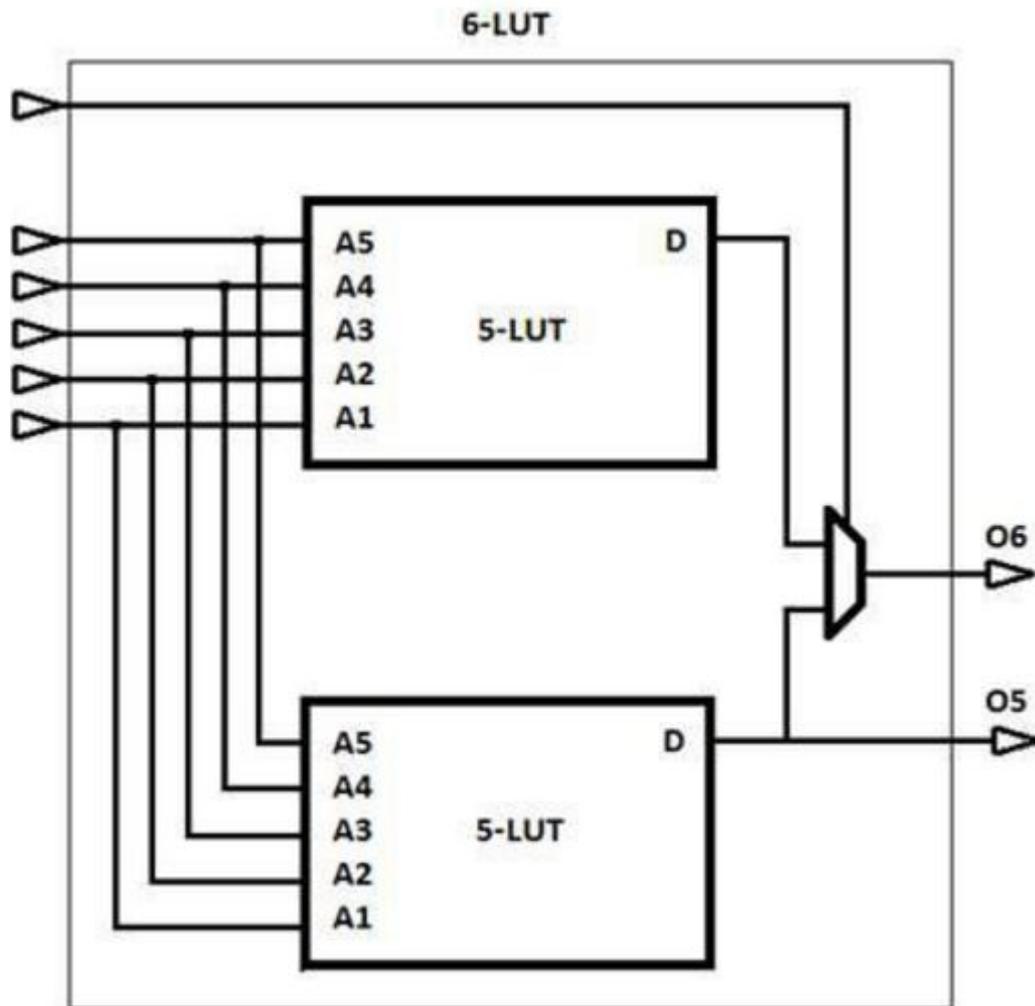
Каждый модуль двухпортовой блочной памяти емкостью 36 кбит, который может использоваться в виде двух независимых блоков объемом 18 кбит, оснащен встроенной системой обнаружения и коррекции ошибок (ECC) и специальной схемой управления, необходимой для организации запоминающих устройств, функционирующих по принципу «первым вошел – первым вышел»

Основные элементы ПЛИС



В состав секций SLICEL входит логика ускоренного переноса и мультиплексоры, предназначенные для расширения функциональных возможностей конфигурируемых логических блоков CLB. В секциях типа SLICEM, кроме логики ускоренного переноса и мультиплексоров расширения количества аргументов выполняемых функций, применяются таблицы преобразования, которые могут конфигурироваться как элементы 64-разрядной распределенной памяти или как сдвиговые регистры различной разрядности (до 32 разрядов).

Основные элементы ПЛИС



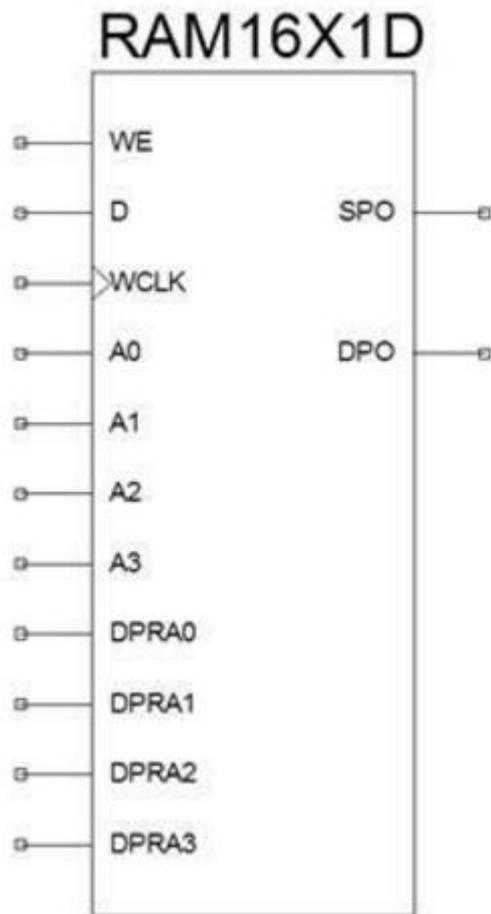
Основной элемент для реализации логических функций ЛОГИЧЕСКИЙ ГЕНЕРАТОР 6LUT. Блоки 5LUT-статическая память.

Логический генератор (LUT) реализует любую функцию с не более чем 6 операндами.

Кроме того возможно использование LUT блоков SLICEM в режиме с частично совмещенными входами, когда память разбивается на два фрагмента по 32 бита (формируются две логические функции с не более чем 5 операндами каждая).

При этом, поскольку в самой LUT всего 6 входов, часть входов должна быть обобщена. Такой режим автоматически

Основные элементы ПЛИС



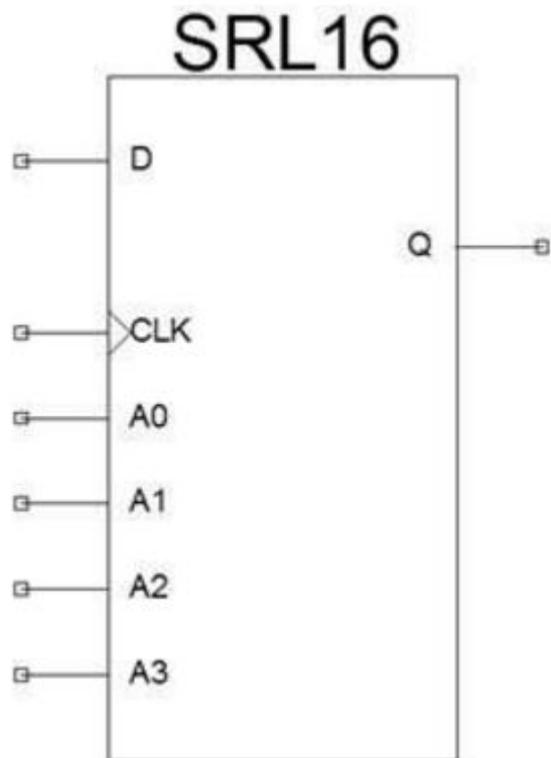
Логические генераторы (LUT) могут также использоваться в режимах распределенной памяти (distributed memory) и сдвиговых регистров (SHR).

Поскольку LUT представляет собой элемент статической памяти, который хранит таблицу истинности в явном виде, в SLICEM его можно использовать по прямому назначению – в качестве обычного компонента памяти.

Показанный элемент распределенной памяти имеет следующие сигналы:

- we – сигнал разрешения записи;
- d – данные для записи;
- wclk – вход тактового сигнала;
- a – адрес для записи;
- dpra – адрес для чтения, второй порт (dualportreadaddress);
- spo – выход первого порта (singleportoutput);
- dpo – выход второго порта (dualportoutput).

Основные элементы ПЛИС



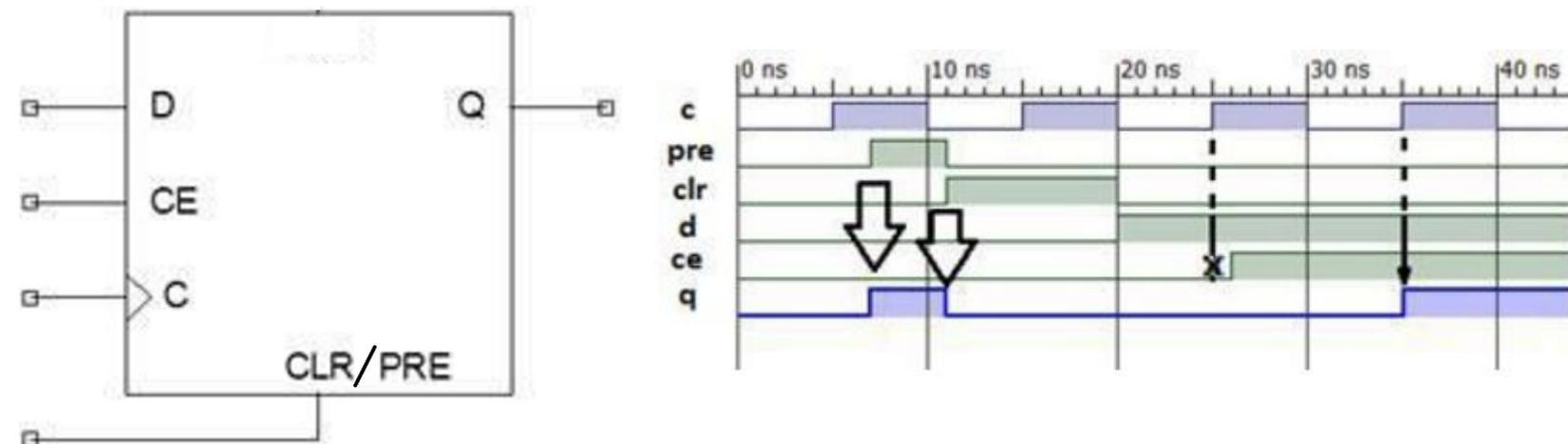
LUT может функционировать и как сдвиговый регистр, использующий следующие сигналы:

имеет следующие сигналы:

- d – данные для записи в сдвиговый регистр;
- clk – тактовый сигнал;
- a – адресный вход;
- q – выход данных.

LUT в режиме сдвигового регистра может использоваться совместно с триггером той же логической ячейки, увеличивая глубину сдвигового регистра на 1.

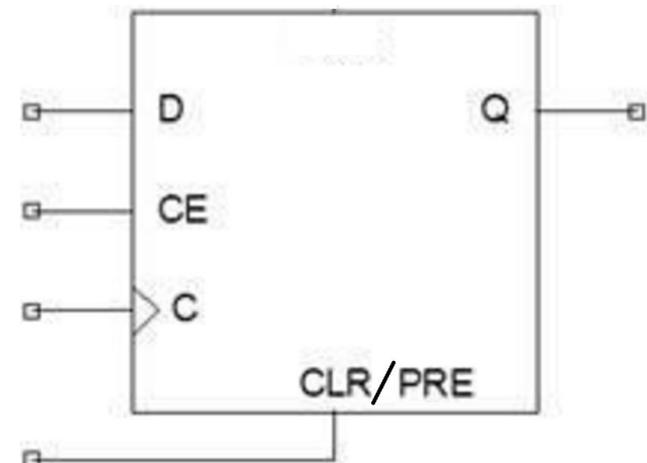
Основные элементы ПЛИС



Временные диаграммы работы триггера. Сигналы PRE, CLR действуют асинхронно, то есть установка триггера может осуществляться без привязки к фронту тактового сигнала.

Для логической секции FPGA Xilinx существует понятие controlset. Оно соответствует набору управляющих сигналов (clk, ce, sr), которые должны быть идентичны для всех триггеров секции. Это означает, например, что невозможно использовать сигнал ce только для одного триггера в секции. Следует обратить внимание, что тактовый сигнал также включен в controlset, т.е. все триггеры должны тактироваться одним и тем же тактовым сигналом. На практике такое ограничение означает, что при необходимости использовать разные управляющие сигналы соответствующие триггеры₈ будут помещены в разные секции.

Основные элементы ПЛИС



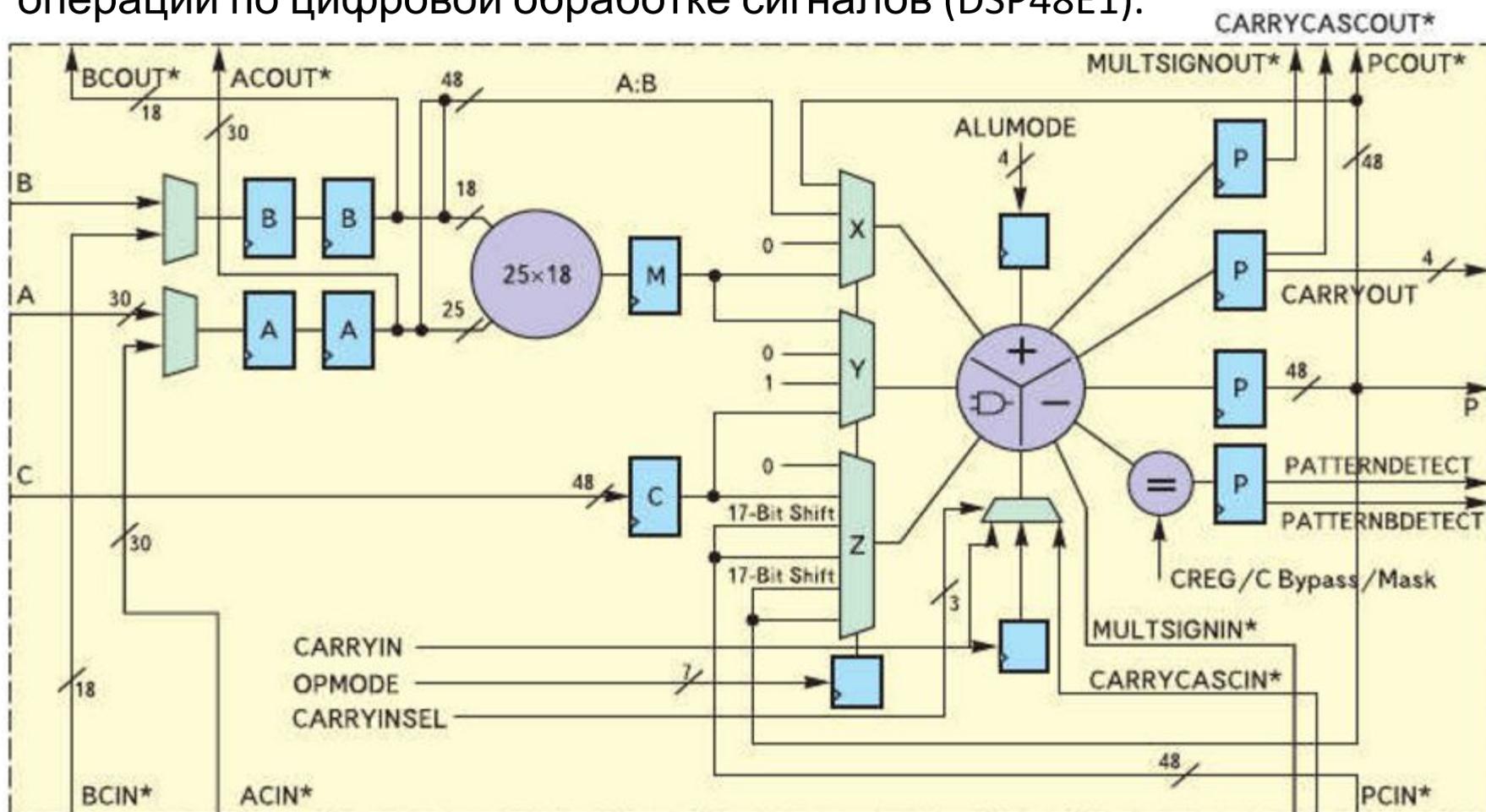
Триггеры FPGA имеют гарантированное начальное значение после включения питания. Значения всех триггеров (и других компонентов системы) записаны в конфигурационном файле, и инициализация выполняется с помощью специальных трассировочных ресурсов. Поэтому отдельные цепи сброса, для обеспечения начального состояния триггера, являются избыточными и ухудшают проект. Самый простой способ начальной установки триггера – объявление сигнала с начальным значением:

```
signal reg1 : std_logic := '1';
```

В этом примере на VHDL объявляется триггер с именем reg1, которому присваивается начальное значение 1. Это значение будет корректно учитываться средствами моделирования, и будет записано в конфигурационный файл FPGA.

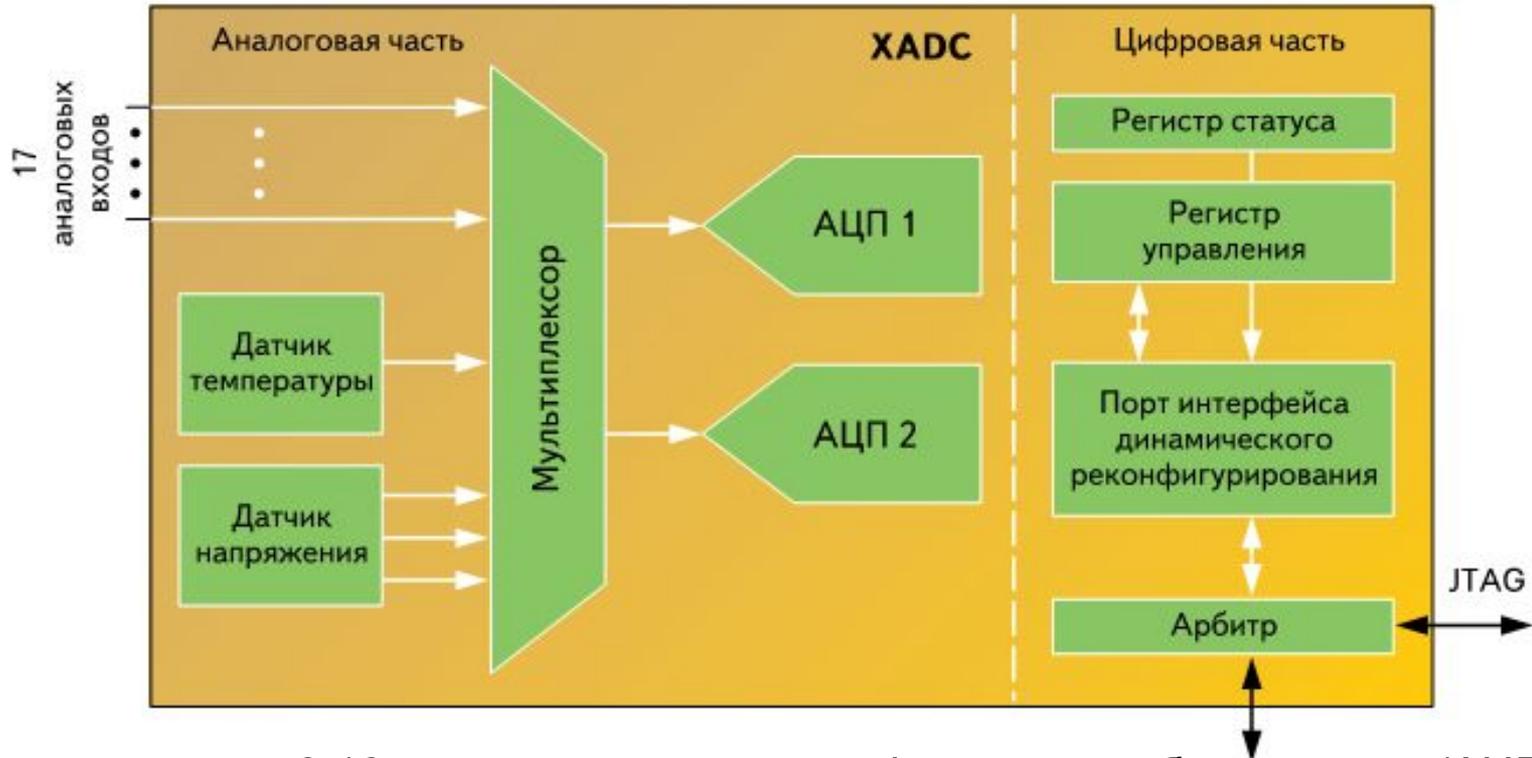
Основные элементы ПЛИС

В ПЛИС семейства VIRTEX (ARTIX-7 и Kintex-7) содержатся многофункциональные блоки, предназначенные для стандартных операций по цифровой обработке сигналов (DSP48E1).



Вычислительные возможности блока ЦОС видны из схемы-АЛУ и блок умножения.

В ПЛИС Xilinx 7-й серии содержатся аналого-цифровые блоки XADC



Каждый блок содержит 2 12-разрядных аналого-цифровых преобразователя (АЦП) и мультиплексор. Кроме того, к аналого-цифровому блоку относятся внутрикристалльные датчики напряжения питания и температуры.

Аналого-цифровой блок XADC поддерживает 17 внешних входных аналоговых каналов. Коммутация аналоговых сигналов на входы АЦП осуществляется с помощью мультиплексора. Каждый АЦП позволяет выполнять преобразование входного аналогового сигнала с частотой дискретизации 1 млн выборок/с и точностью 0,1%. Наличие блока XADC в составе ПЛИС позволяет реализовать устройства цифровой обработки низкочастотных сигналов на базе одного кристалла без внешних АЦП.

Семейство Artix-7

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP48E1 Slices ⁽²⁾	Block RAM Blocks ⁽³⁾			CMTs ⁽⁴⁾	PCIe ⁽⁵⁾	GTPs	XADC Blocks	Total I/O Banks ⁽⁶⁾	Max User I/O ⁽⁷⁾
		Slices ⁽¹⁾	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)						
XC7A15T	16,640	2,600	200	45	50	25	900	5	1	4	1	5	250
XC7A35T	33,280	5,200	400	90	100	50	1,800	5	1	4	1	5	250
XC7A50T	52,160	8,150	600	120	150	75	2,700	5	1	4	1	5	250
XC7A75T	75,520	11,800	892	180	210	105	3,780	6	1	8	1	6	300
XC7A100T	101,440	15,850	1,188	240	270	135	4,860	6	1	8	1	6	300
XC7A200T	215,360	33,650	2,888	740	730	365	13,140	10	1	16	1	10	500

Наименьшие значения потребляемой мощности и низкой стоимостью, ориентированы на применение в составе серийно выпускаемой аппаратуры.

CLB - общее количество триггеров

CMT – блоки управления синхронизацией

PCIe- интерфейсы к PCI express

XADC- Аналого- цифровые преобразователи

Семейство Kintex-7

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP Slices ⁽²⁾	Block RAM Blocks ⁽³⁾			CMTs ⁽⁴⁾	PCIe ⁽⁵⁾	GTXs	XADC Blocks	Total I/O Banks ⁽⁶⁾	Max User I/O ⁽⁷⁾
		Slices ⁽¹⁾	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)						
XC7K70T	65,600	10,250	838	240	270	135	4,860	6	1	8	1	6	300
XC7K160T	162,240	25,350	2,188	600	650	325	11,700	8	1	8	1	8	400
XC7K325T	326,080	50,950	4,000	840	890	445	16,020	10	1	16	1	10	500
XC7K355T	356,160	55,650	5,088	1,440	1,430	715	25,740	6	1	24	1	6	300
XC7K410T	406,720	63,550	5,663	1,540	1,590	795	28,620	10	1	16	1	10	500
XC7K420T	416,960	65,150	5,938	1,680	1,670	835	30,060	8	1	32	1	8	400
XC7K480T	477,760	74,650	6,788	1,920	1,910	955	34,380	8	1	32	1	8	400

Увеличено число ЦОС DSP48E1 и модулей двухпортовой блочной памяти Block RAM. Используются высокоскоростные приемопередатчики RocketIO типа GTX, поддерживающие скорость приема и передачи данных до 10,3125 Гбит/с.

Содержат аппаратный модуль интерфейса PCI Express, который поддерживает скорости передачи данных 2,5 и 5 Гбит/с

Семейство Virtex-7

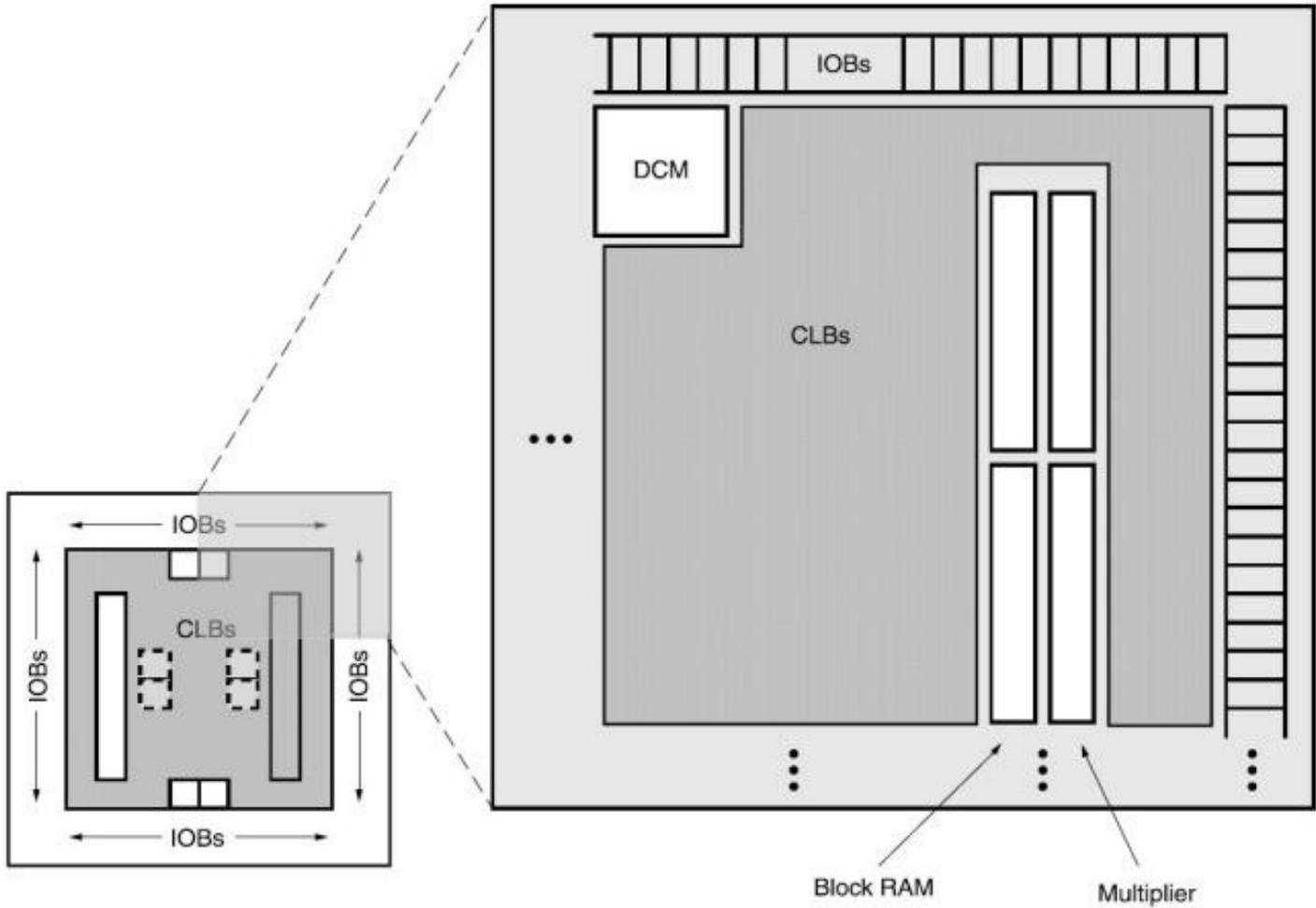
Device ⁽¹⁾	Logic Cells	Configurable Logic Blocks (CLBs)		DSP Slices ⁽³⁾	Block RAM Blocks ⁽⁴⁾			CMTs ⁽⁵⁾	PCIe ⁽⁶⁾	GTX	GTH	GTZ	XADC Blocks	Total I/O Banks ⁽⁷⁾	Max User I/O ⁽⁸⁾	SLRs ⁽⁹⁾
		Slices ⁽²⁾	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)									
XC7V585T	582,720	91,050	6,938	1,260	1,590	795	28,620	18	3	36	0	0	1	17	850	N/A
XC7V2000T	1,954,560	305,400	21,550	2,160	2,584	1,292	46,512	24	4	36	0	0	1	24	1,200	4
XC7VX330T	326,400	51,000	4,388	1,120	1,500	750	27,000	14	2	0	28	0	1	14	700	N/A
XC7VX415T	412,160	64,400	6,525	2,160	1,760	880	31,680	12	2	0	48	0	1	12	600	N/A
XC7VX485T	485,760	75,900	8,175	2,800	2,060	1,030	37,080	14	4	56	0	0	1	14	700	N/A
XC7VX550T	554,240	86,600	8,725	2,880	2,360	1,180	42,480	20	2	0	80	0	1	16	600	N/A
XC7VX690T	693,120	108,300	10,888	3,600	2,940	1,470	52,920	20	3	0	80	0	1	20	1,000	N/A
XC7VX980T	979,200	153,000	13,838	3,600	3,000	1,500	54,000	18	3	0	72	0	1	18	900	N/A
XC7VX1140T	1,139,200	178,000	17,700	3,360	3,760	1,880	67,680	24	4	0	96	0	1	22	1,100	4
XC7VH580T	580,480	90,700	8,850	1,680	1,880	940	33,840	12	2	0	48	8	1	12	600	2
XC7VH870T	876,160	136,900	13,275	2,520	2,820	1,410	50,760	18	3	0	72	16	1	6	300	3

Super logic regions (SLRs) are the constituent parts of FPGAs that use SSI technology. Virtex-7 HT devices use SSI technology to connect SLRs with 28.05 Gb/s transceivers.

Увеличено число ЦОС DSP48E1 и модулей двухпортовой блочной памяти Block RAM. Используются высокоскоростные приемопередатчики RocketIO (MGT) типа GTX+GTH+GTZ, поддерживающие скорость приема и передачи данных до 10,3125 (очень много) Гбит/с.

Содержат аппаратный модуль (блок) интерфейса PCI Express, который поддерживает (PCI Express Gen 1, 2 и 3) скорости передачи данных 2,5 и 5 Гбит/с

Организация работы со статической памятью в ПЛИС



Работа со статической памятью

Внутри ПЛИС Xilinx есть специальный аппаратный ресурс – блочная память или BRAM (Block RAM).

Эти блоки выполнены аппаратно, т.е. часть логики кристалла зафиксирована под память, и не может быть использована по другому назначению. Количество таких блоков зависит от конкретного типа кристалла ПЛИС и может быть до нескольких сотен. Каждый блок имеет два аппаратных порта, которые могут работать на разных тактовых частотах.

Другой ресурс, который можно использовать как память – это сами конфигурационные блоки ПЛИС. Из конфигурационных блоков можно собрать логическую схему статической памяти. Тогда биты информации, хранящиеся в такой памяти будут «размазаны» или «распределены» по большей (по сравнению с BRAM, организованным локально) площади кристалла, и такая память будет называться распределенной памятью (Distributed RAM).

Работа со статической памятью

Сравнивая блочную и распределенную память, надо сказать следующее:

Блочная память – это аппаратные блоки.

Блочная память – дорогостоящий ресурс. Количество ее ограничено. При этом, если требуется объем памяти меньший чем объем одного блока (18 или 36 Кбит), то все равно будет использован весь блок, и неиспользованная его часть просто пропадет. При этом распределенная память может быть создана конкретного объема: ни одного лишнего конфигурационного блока не будет задействовано.

При использовании блочной памяти ее разные порты могут работать на разных частотах. Это удобно использовать для синхронизации между разными блоковыми доменами.

Схемы как блочной так и распределенной памяти создаются с использованием программы генератора процессорного ядра. В ней можно выбрать необходимый тип памяти, объем, ширины шин данных, особенности и интерфейса. После создания IP блока, работать с памятью надо через интерфейс, подобный рассмотренному на следующем слайде.

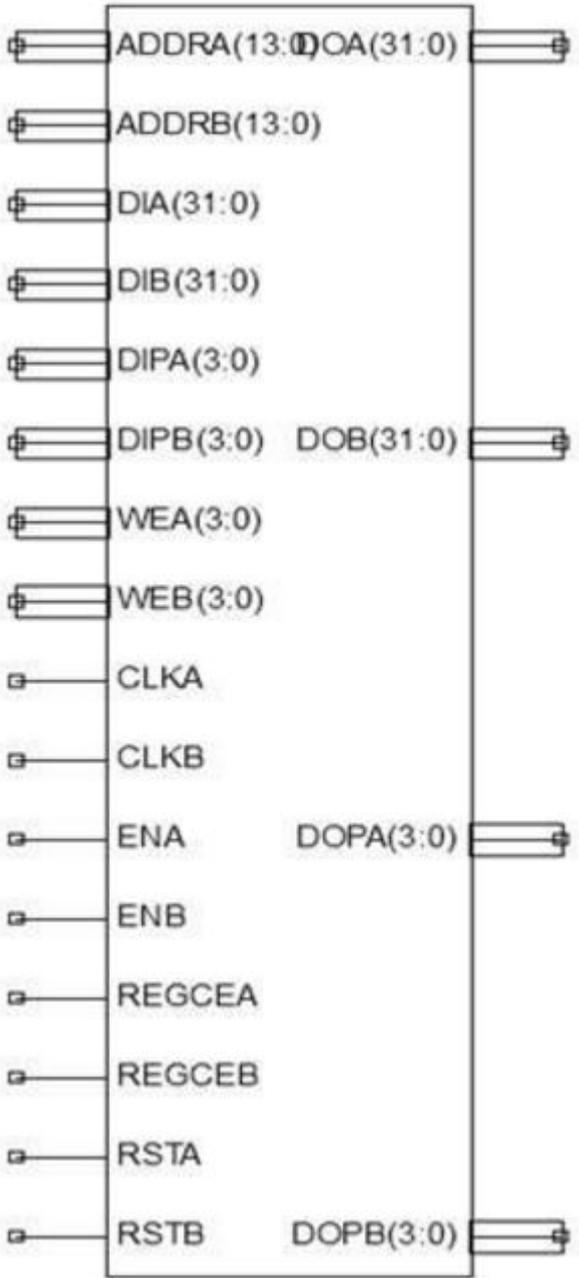
В VHDL есть возможность описать массив ячеек статической памяти с помощью сигналов типа массив и стандартных операторов.

Организация работы со статической памятью в ПЛИС

```
entity static_ram
PORT (
  clk : IN STD_LOGIC; -- тактовый сигнал
  rst : IN STD_LOGIC; -- сигнал сброса
  en : IN STD_LOGIC; -- сигнал разрешения работы (опционально)
  we : IN STD_LOGIC; -- write enable, сигнал определяющий операцию
записи или чтения
  addr : IN STD_LOGIC_VECTOR(3 DOWNTO 0); -- шина адреса ячейки
  din : IN STD_LOGIC_VECTOR(15 DOWNTO 0); -- входная шина данных
  dout : OUT STD_LOGIC_VECTOR(15 DOWNTO 0) -- выходная шина данных
);
END static_ram;
```

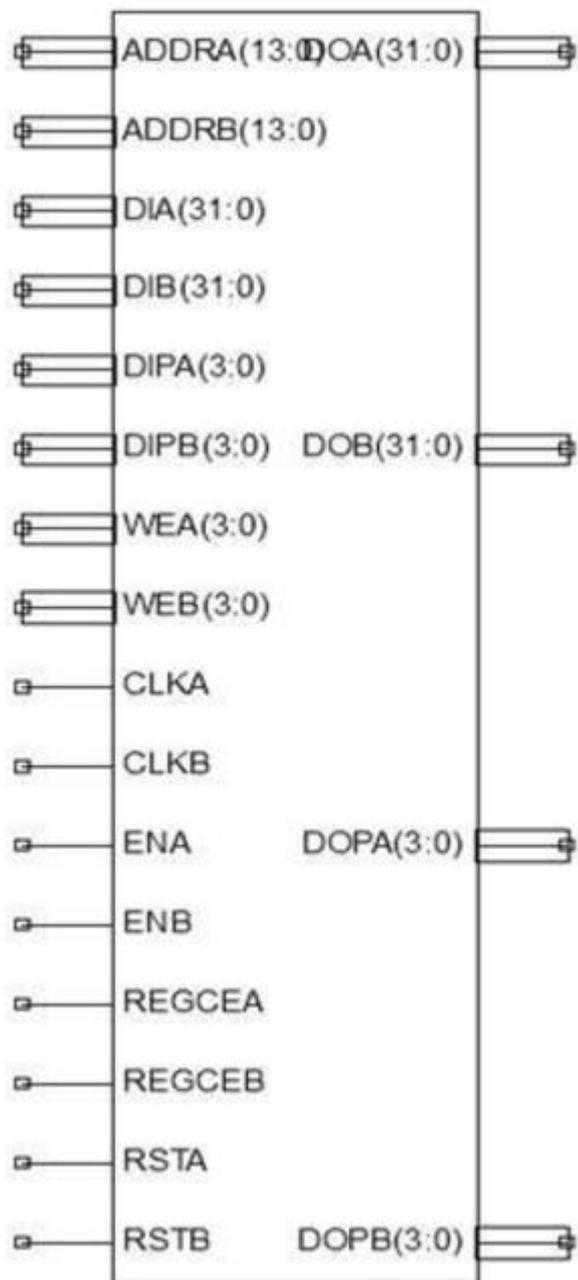
Организация работы со статической памятью в ПЛИС

Блок памяти имеет следующие порты:



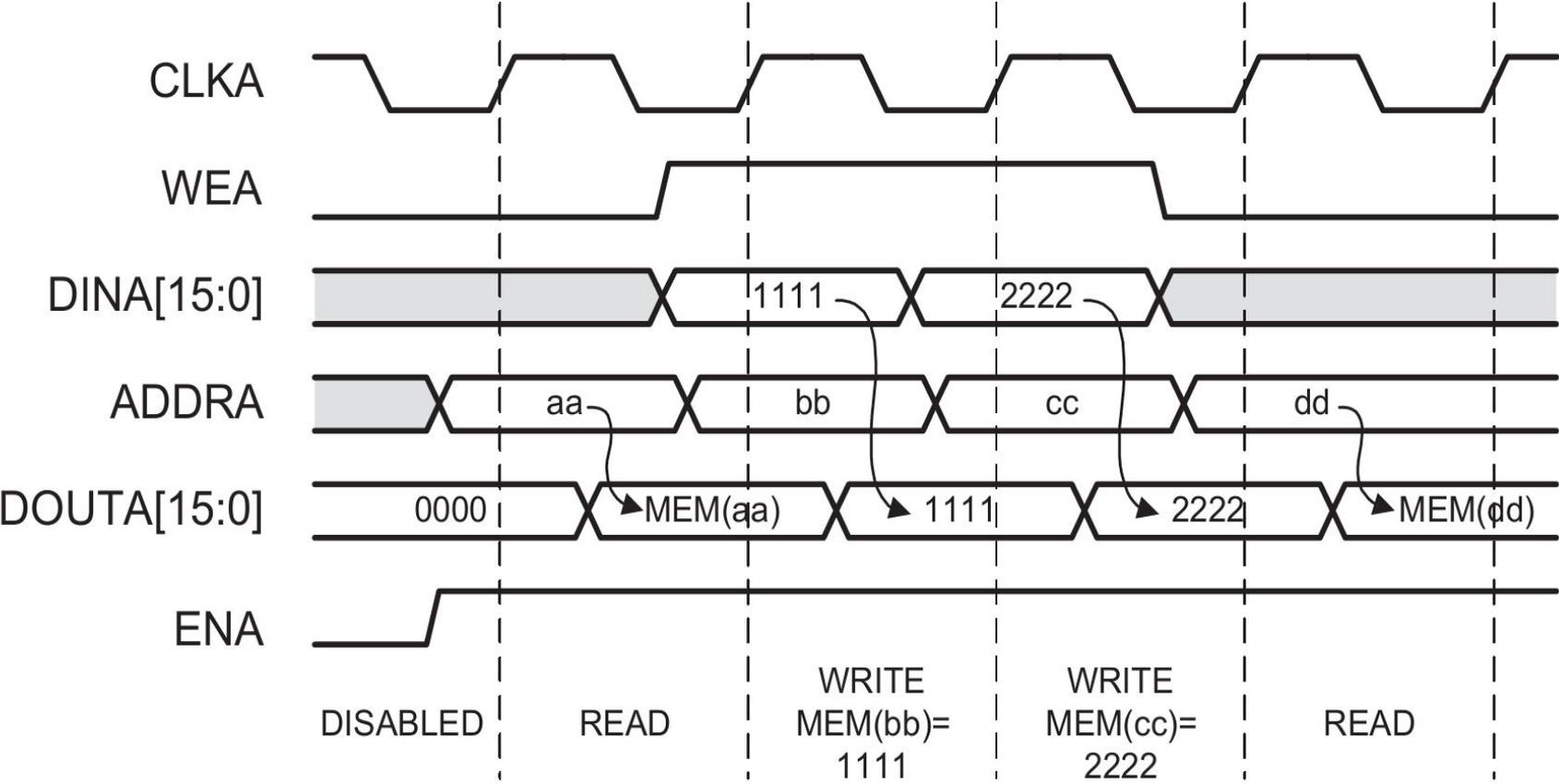
- addra, addrb – адреса портов А и В соответственно;
- dia, dib – данные для записи для портов А и В (32 бита);
- dipa, dipb – дополнительные данные для записи (4 бита);
- wea, web – входы разрешения записи (побайтно);
- clka, clkb – тактовые сигналы для портов;
- ena, enb – входы разрешения работы блока памяти (при чтении состояние выходов не обновляется, если нет разрешающего сигнала);
- regcea, regceb – разрешение работы выходных регистров;
- rsta, rstb – сброс выходных регистров (не влияет на содержимое массивов памяти);
- doa, dob – выходы данных для портов А и В (32 бита);
- dora, dorb – дополнительные выходы данных для портов А и В (4 бита)

Организация работы со статической памятью в ПЛИС



Физически размещенные в FPGA блоки памяти являются 18-битными. Такая разрядность позволяет реализовывать схемы контроля четности, когда каждые 8 бит имеют дополнительный 9-й бит для хранения бита четности. Соответственно, каждые 16 бит имеют 2 дополнительных бита четности, а 32 – 4 бита. Для удобства работы с дополнительными битами в графическом представлении модуля они выделены в отдельные шины *dip_a*, *dip_b*, *dop_a*, *dop_b*. Дополнительные биты не являются автоматически заполняемыми и представляют собой разряды, доступные для записи в них произвольных значений. Разработчик может выбирать требуемую ему разрядность, включая 1, 2, 4, 9, 18 или 36 бит, в особых случаях - до 72 бит.

Организация работы со статической памятью в ПЛИС



Организация работы со статической памятью в ПЛИС

EN	WE	Операция
1	0	Чтение
1	1	Запись
0	0	Память не работает
0	1	Память не работает

Организация работы со статической памятью в ПЛИС: двухпортовая память

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
entity rams_16 is
port(clka : in std_logic;
      clkb : in std_logic;
      ena : in std_logic;
      enb : in std_logic;
      wea : in std_logic;
      web : in std_logic;
      addr_a : in std_logic_vector(5 downto 0);
      addr_b : in std_logic_vector(5 downto 0);
      dia : in std_logic_vector(15 downto 0);
      dib : in std_logic_vector(15 downto 0);
      doa : out std_logic_vector(15 downto 0);
      dob : out std_logic_vector(15 downto 0));
end rams_16;
```

Организация работы со статической памятью в ПЛИС: двухпортовая память

```
architecture syn of rams_16 is
type ram_type is array (63 downto 0) of std_logic_vector(15 downto 0);
shared variable RAM : ram_type;
begin
process (CLKA)
begin
if CLK 'event and CLK = '1' then
    if ENA = '1' then
        if WEA = '1' then
            RAM(conv_integer(ADDRA)) := DIA;
        end if;
        DOA <= RAM(conv_integer(ADDRA));
    end if;
end if;
end process;
```

Организация работы со статической памятью в ПЛИС

```
process (CLKB)
begin
if CLKB 'event and CLKB = '1' then
    if ENB = '1' then
        if WEB = '1' then
            RAM(conv_integer(ADDRB)) := DIB;
        end if;
        DOB <= RAM(conv_integer(ADDRB));
    end if;
end if;
end process;
end syn;
```

Термин:

IP cores (IP-ядра, от англ. Intellectual Property — интеллектуальный продукт. В FPGA (ПЛИС) под Hard IP-Core понимают специализированные области кристалла, выделенные для определенных функций. В этих областях реализованы блоки неизменной структуры, спроектированные по методологии ASIC (как области типа БМК или схем со стандартными ячейками), оптимизированные для заданной функции и не имеющие средств её программирования.

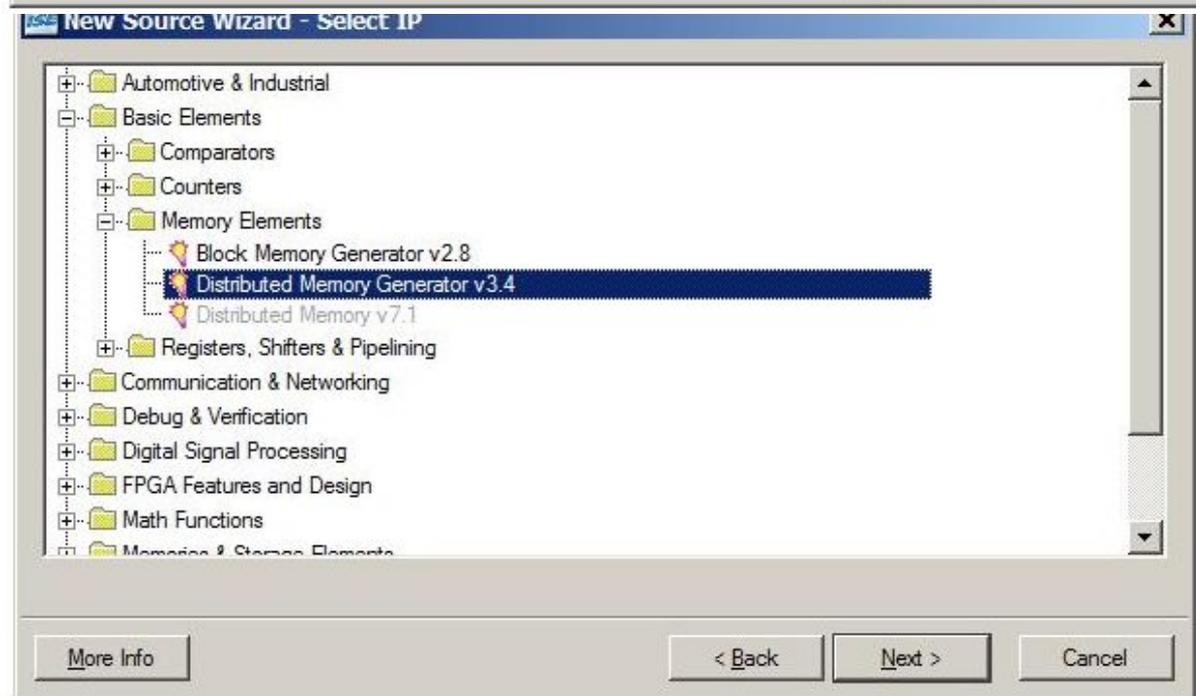
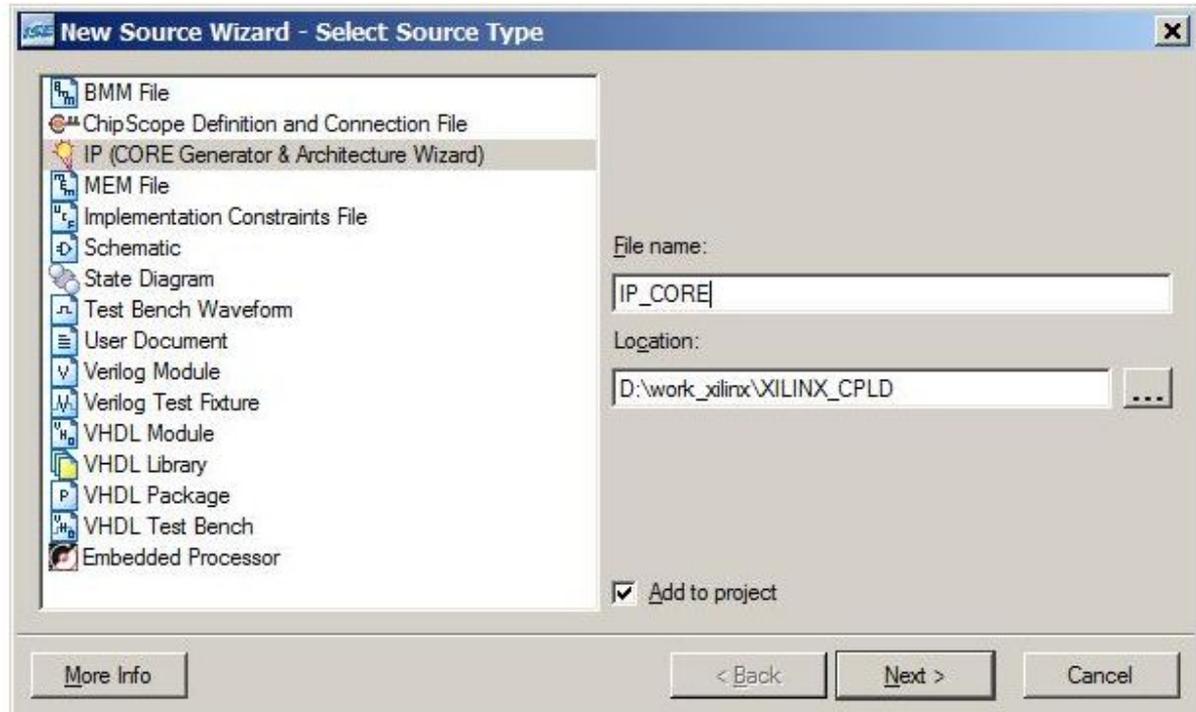
В случае использования данного вида ядер размер площади, используемой на кристалле, сокращается, улучшаются характеристики быстродействия, но в данном случае происходит потеря универсальности.

Термин:

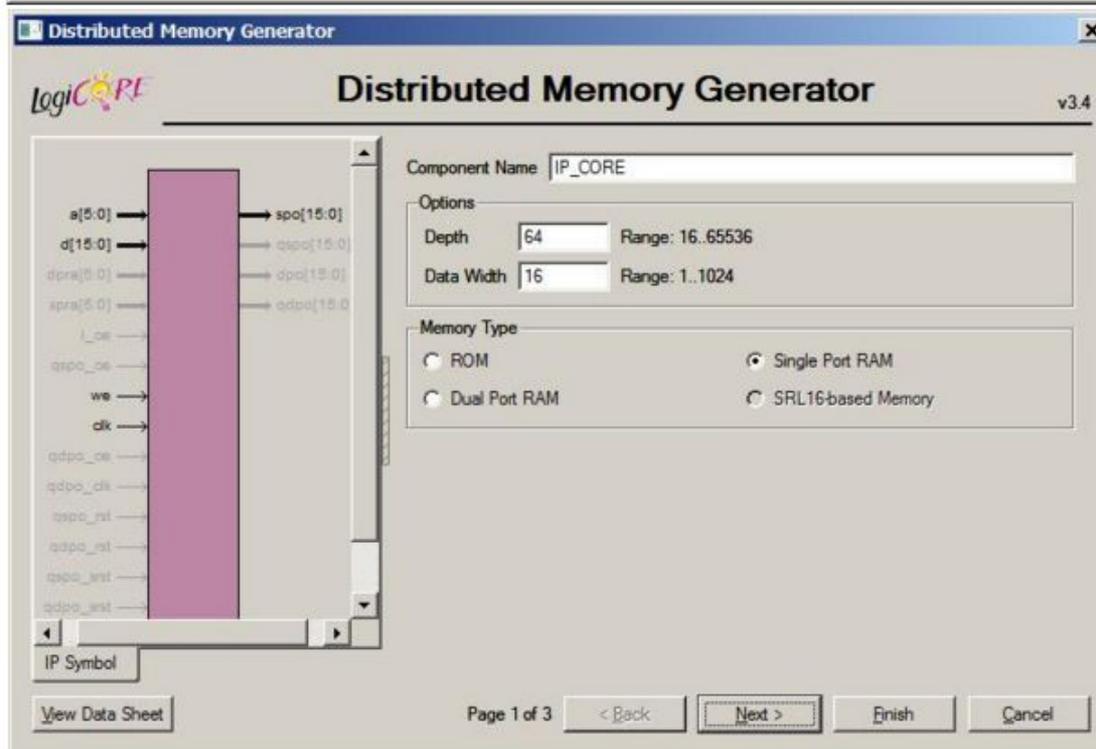
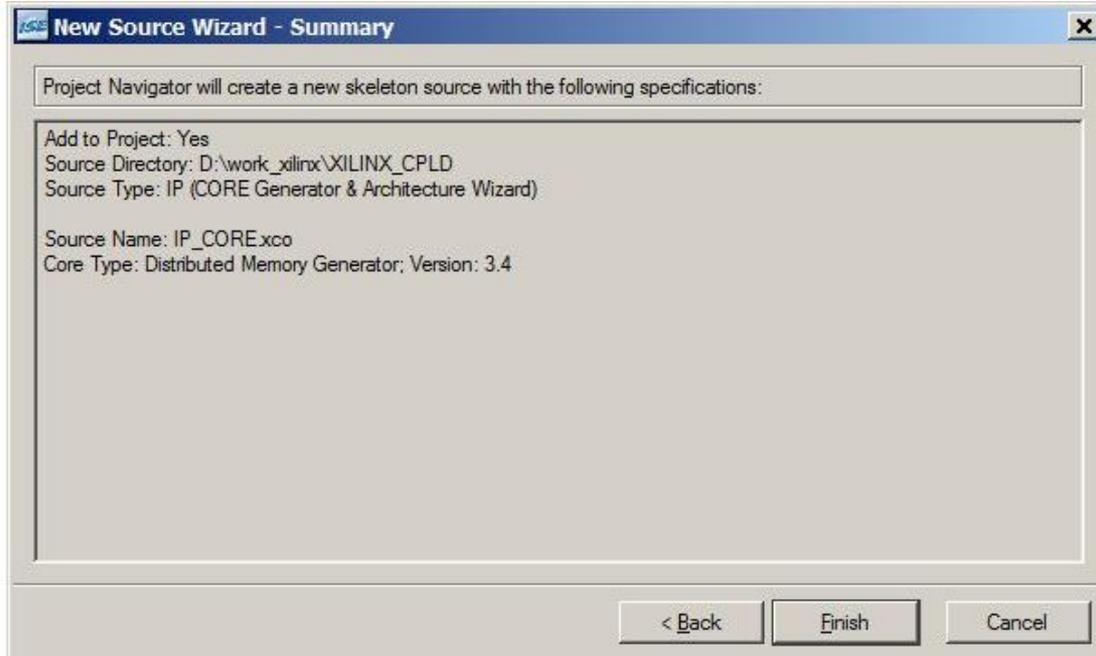
Приложение CoreGenerator входит в состав САПР фирмы Xilinx и предлагает разработчику графический интерфейс быстрого конфигурирования стандартных цифровых блоков для реализации их в ПЛИС (IP-ядер). С помощью CoreGenerator имеется возможность создания шаблонов на языке Verilog, которые потом будут использованы как instance (см. componentinstantiation) в проектах пользователя. Генерация компонентов с помощью CoreGenerator обладает рядом преимуществ:

- IP-ядра, как правило, представляют собой верифицированные компоненты, имеющие качественное размещение на кристалле, и вследствие этого способные работать на высокой тактовой частоте;
- ряд специфичных ресурсов ПЛИС, например, модули формирования тактовой частоты, достаточно сложны для освоения, однако вносят в проект полезные функциональные возможности, поэтому рекомендуются к использованию; соответственно, CoreGenerator предоставляет возможности для их быстрого конфигурирования с последующим добавлением к проекту.

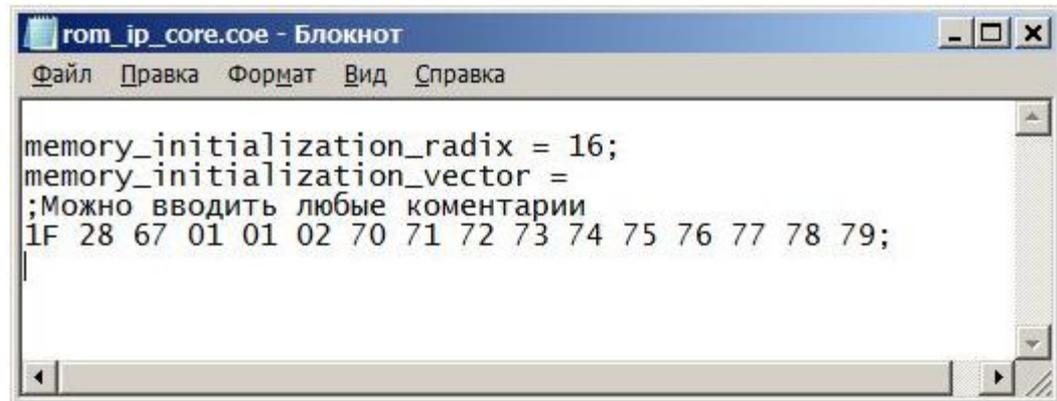
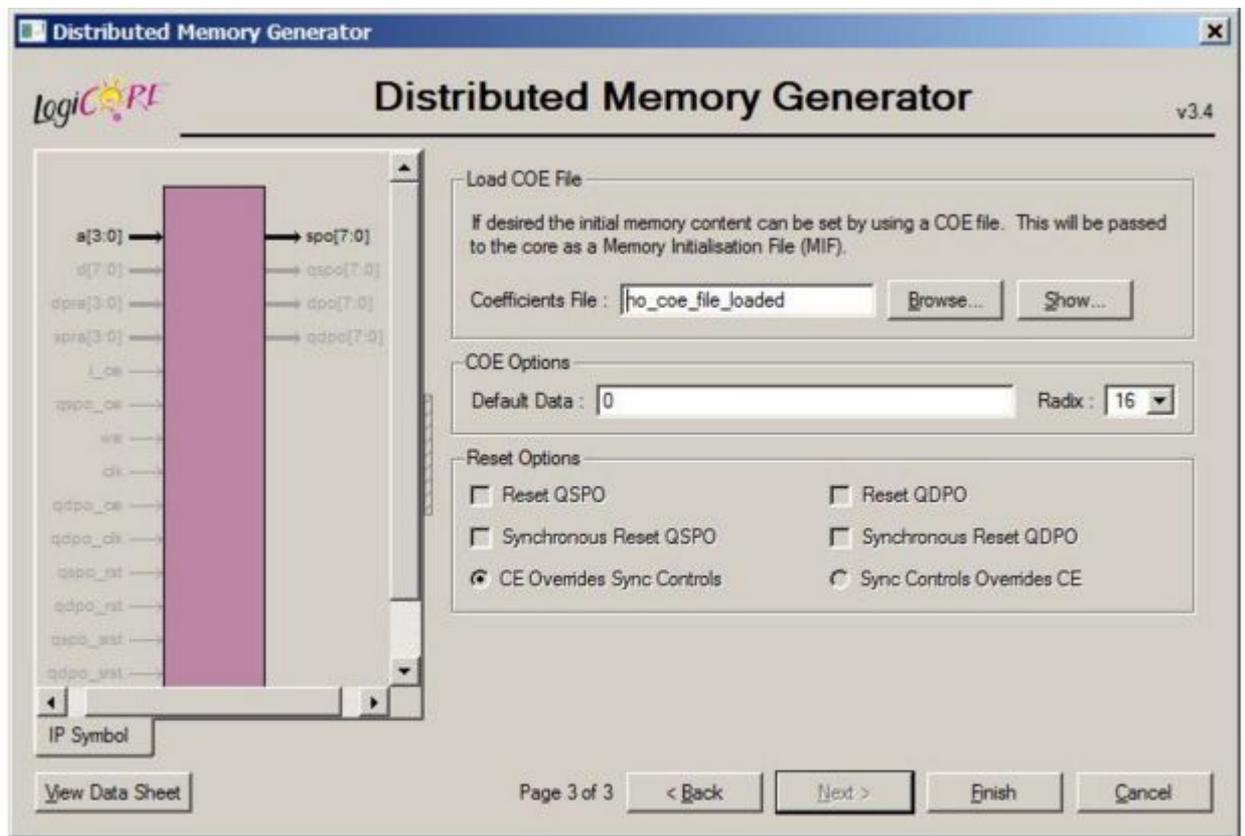
Запускаем генератор и выбираем тип IP ядра



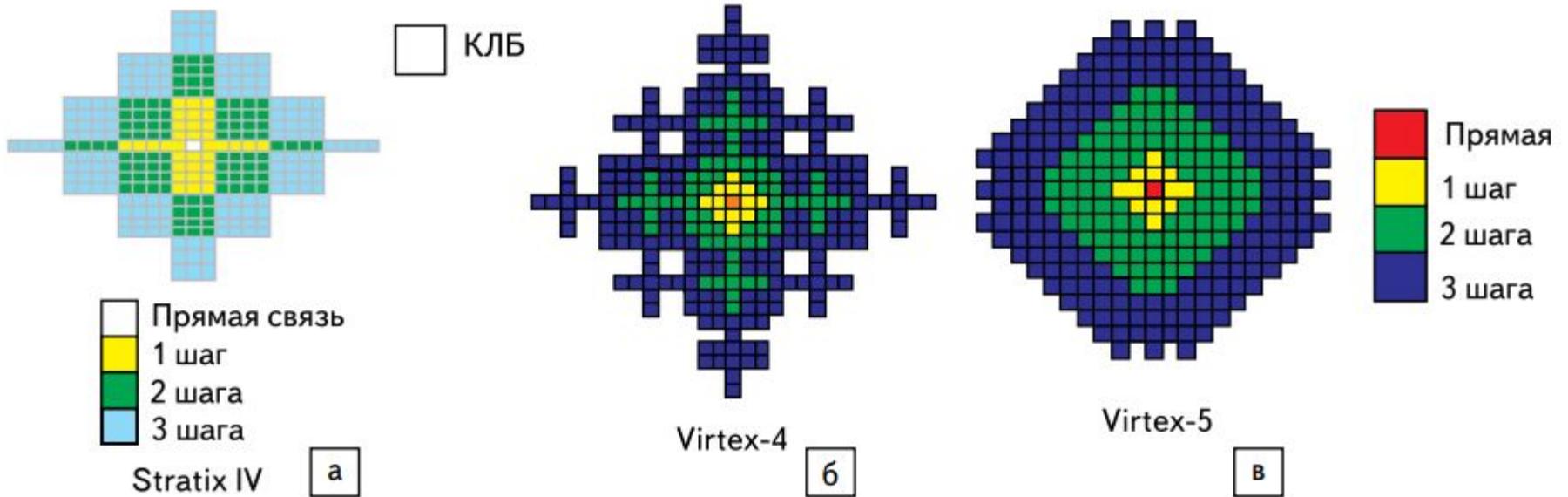
Выбираем параметры памяти, в частности, в частности ширину данных (**Data Width**), количество данных (**Depth – глубина**), тип создаваемой памяти (**Memory Type**)



Создаем и подключаем файлcoe- файл коэффициентов, после этого формируется память по заданным параметрам



Коммутационная система ПЛИС



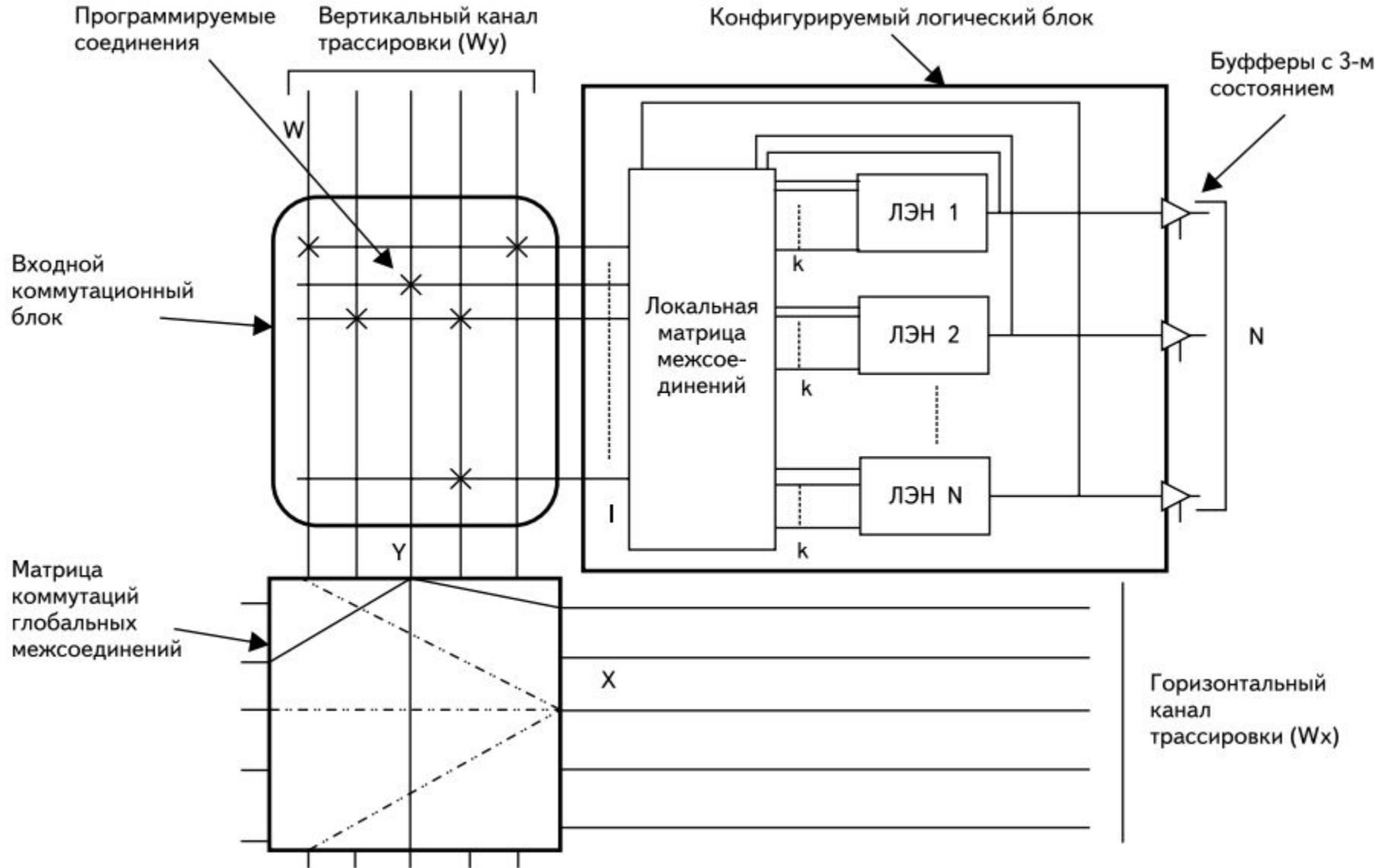
Трассировочные ресурсы ПЛИС:

а) Stratix IV фирмы Altera с использованием технологии DirectDrive;

б, в) фирмы Xilinx с использованием технологии ExpressFabric:

б) Virtex-4; в) Virtex-5

Коммутационная система ПЛИС



Пример организации взаимодействия между логическими блоками

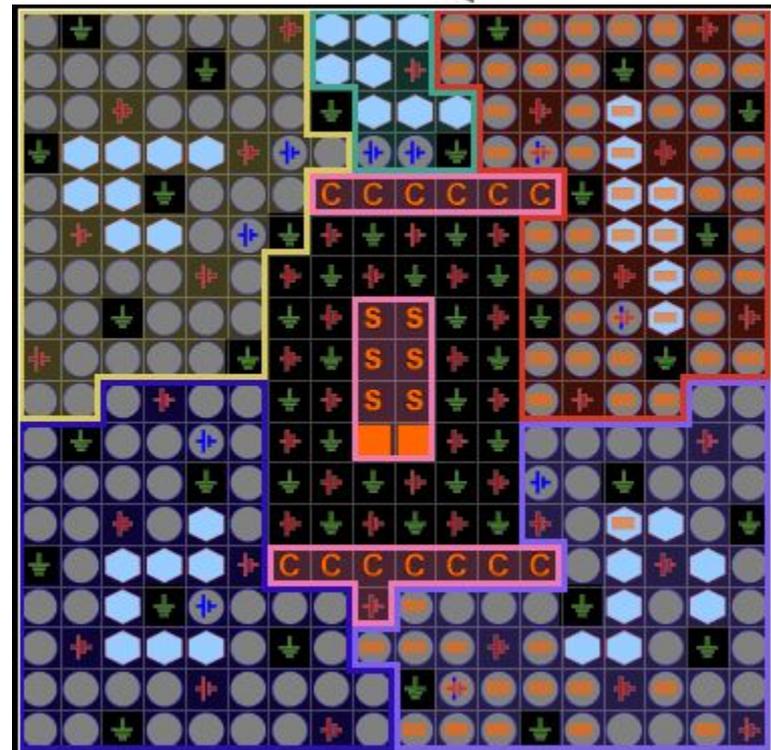
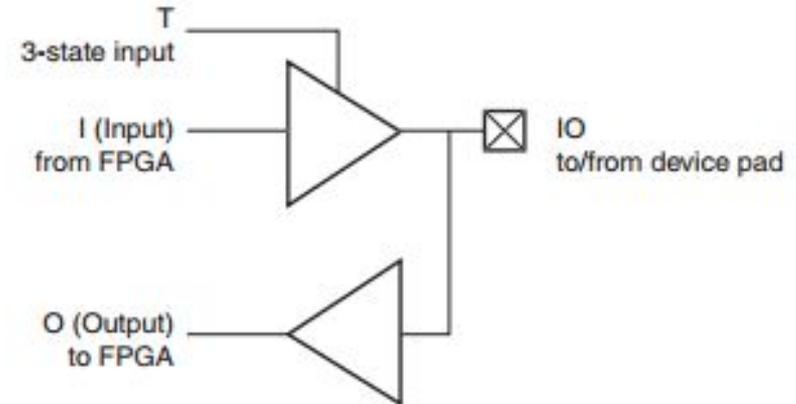
Блоки ввода- вывода

Каждый доступный разработчику ввод- вывод (pin) может быть сконфигурирован как вход, выход или универсальный вход-выход.

Зачастую возможен выбор рабочих напряжений, для разработчика- чем меньше напряжение, тем на большей частоте можно работать, но тем меньше помехозащищенность.

Внешние контакты ПЛИС в САПР Vivado

```
set_property IOSTANDARD LVCMOS18 [all_inputs]
set_property IOSTANDARD LVCMOS18 [all_outputs]
create_clock -period 1.600 -name CLK [get_ports clk]
set_property PACKAGE_PIN N15 [get_ports clk]
...
```



Благодарю за внимание!