

Основы проектирования аппаратных ускорителей систем искусственного интеллекта

Лекция 2 - Введение



Лекция 2

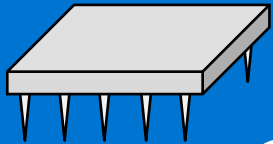
План лекции

- Уровни абстракции при проектировании цифровых СБИС. Математические модели, используемые для описания различных уровней абстракции цифровой СБИС. Комбинационные и последовательные схемы.
- Упрощенный маршрут проектирования современных цифровых СБИС.
- Меры качества разработки цифровых СБИС.

Уровни абстракции при проектировании цифровых СБИС

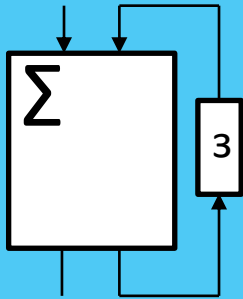
Лекция 2

Уровни абстракции при проектировании цифровых СБИС

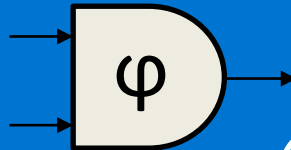


Системный уровень

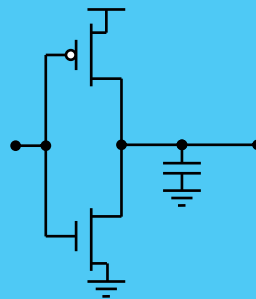
Автоматный (поведенческий) уровень



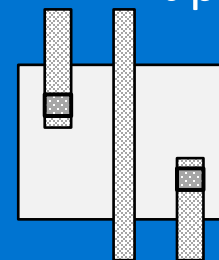
Логический (схемный) уровень



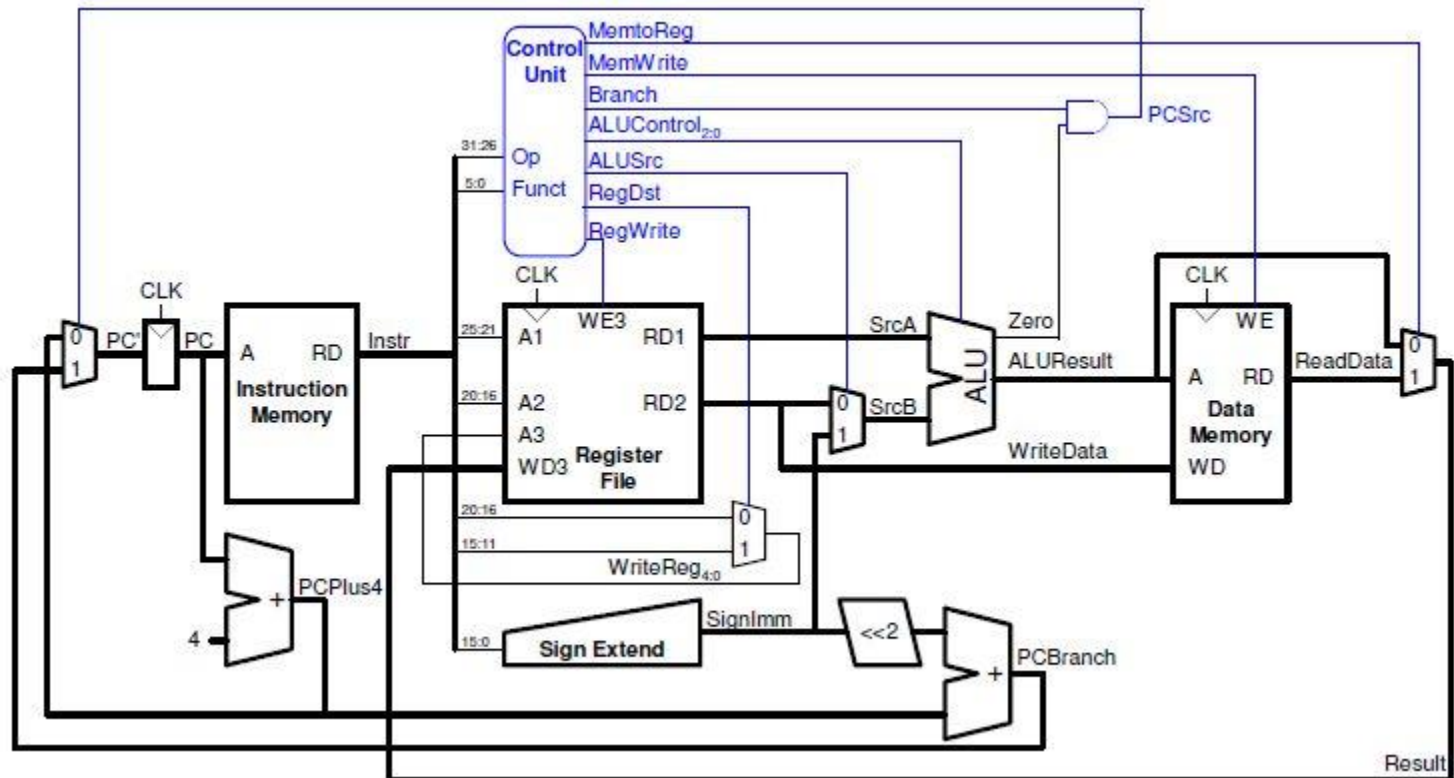
Транзисторный уровень



Уровень топологии



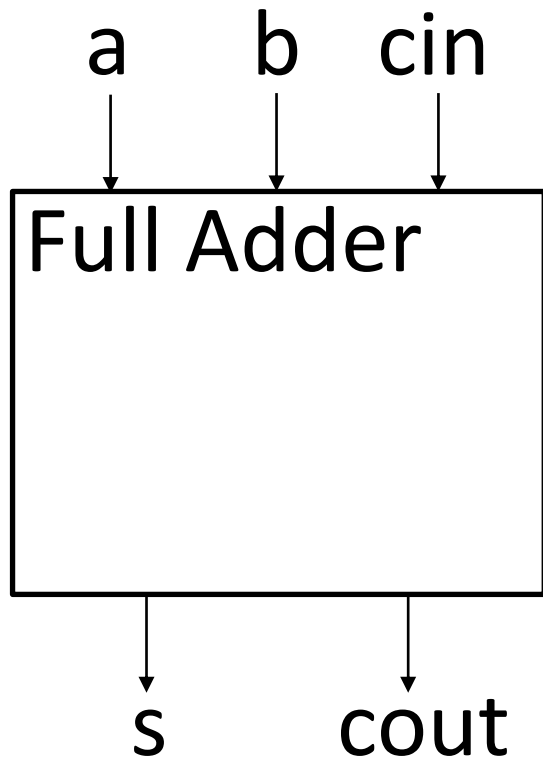
Системный уровень



Системный уровень

- Моделирование системы взаимодействующих процессов/сигналов
- Подходы к моделированию:
 - взаимодействие систем/компонент системы
 - система команд (instruction set simulation)
 - микроархитектура
 - использование языков описания аппаратуры (Verilog, SystemVerilog, SystemC)

Поведенческий уровень



```
`timescale 1ns / 1ps
module FullAdder (
    input a,
    input b,
    input cin,
    output s,
    output cout );

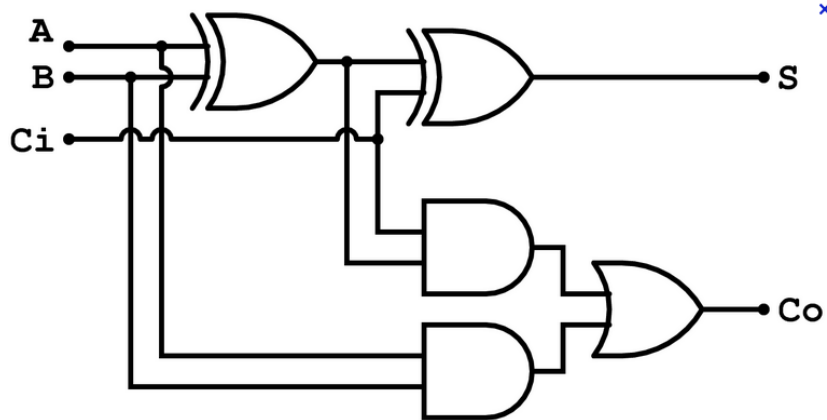
    assign {cout,s} = a + b + cin;

endmodule
```


Поведенческий уровень

- Моделирование поведения/функционирования процесса/сигнала
- Подходы к моделированию:
 - использование языков описания аппаратуры (Verilog, VHDL)
 - register-transfer level (RTL)
 - использование автоматов и других математических моделей

Логический (схемный) уровень



```
`timescale 1ns / 1ps
```

```
module FullAdder (
```

```
    input a,
```

```
    input b,
```

```
    input cin,
```

```
    output s,
```

```
    output cout );
```

```
// wires (from ands to or)
```

```
wire w1, w2, w3;
```

```
// carry-out circuitry
```

```
and( w1, a, b );
```

```
and( w2, a, cin );
```

```
and( w3, b, cin );
```

```
or( cout, w1, w2, w3 );
```

```
// sum
```

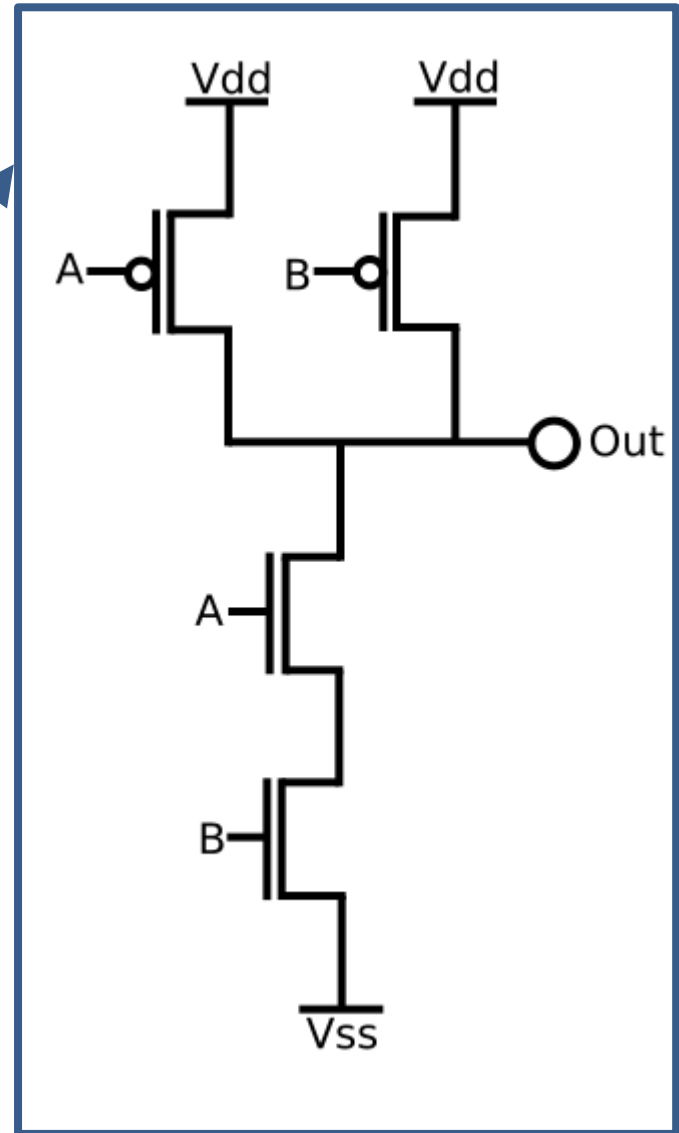
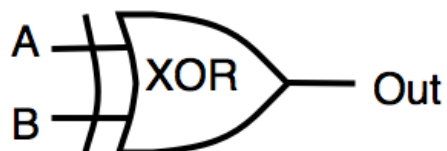
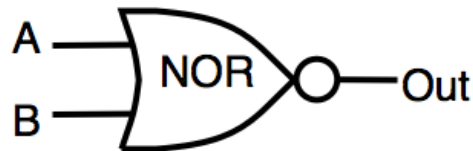
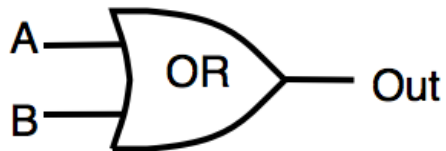
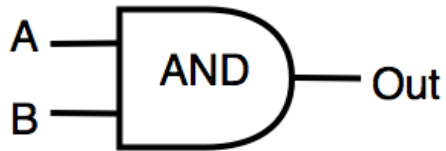
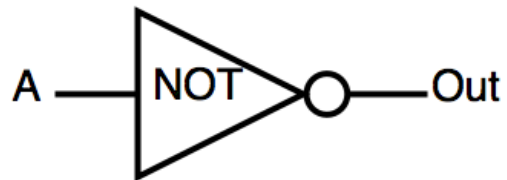
```
xor( s, a, b, cin );
```

```
endmodule
```

Логический (схемный) уровень

- Моделирование структуры и основных элементов блока, реализующего заданный процесс/сигнал
- Подходы к моделированию:
 - использование языков описания аппаратуры (Verilog, VHDL)
 - netlist, gate-level design
 - математические модели схем
 - схемы из функциональных элементов (СФЭ) и их обобщения
 - And-Inverter Graphs (AIG)
 - Binary Decision Diagrams (BDD)
 - и др.

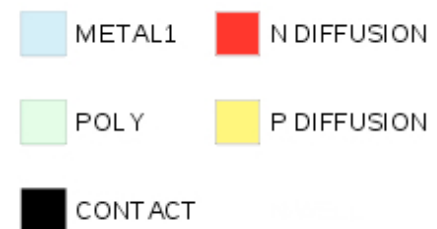
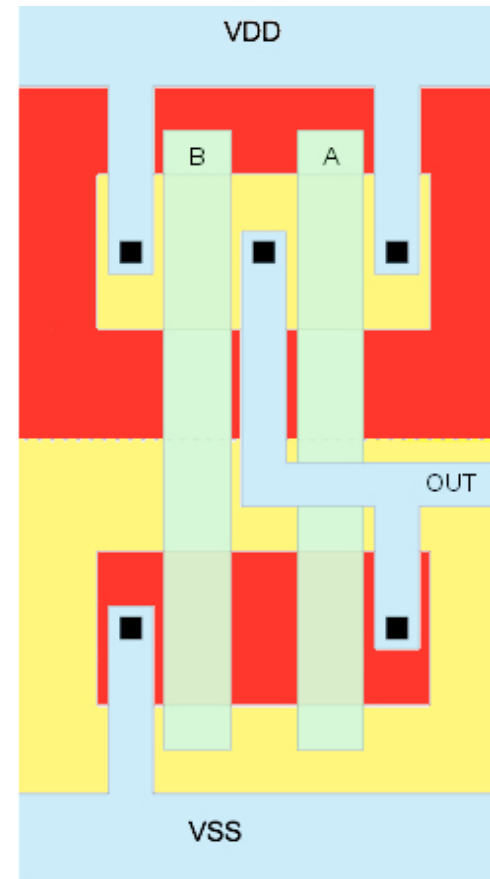
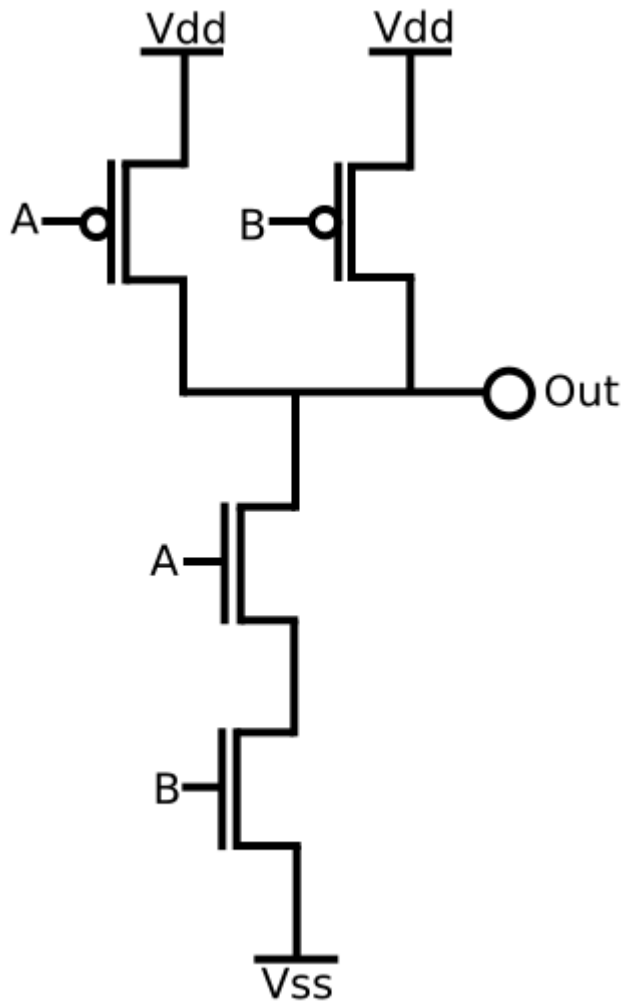
Транзисторный уровень



Транзисторный уровень

- Моделирование структуры основных логических элементов интегральной схемы
- Определение/оценка основных физических характеристик логических элементов (размер, задержка, энергопотребление и др.)
- Подходы к моделированию:
 - использование различных транзисторных моделей схем
 - имитационной моделирование
 - SPICE моделирование

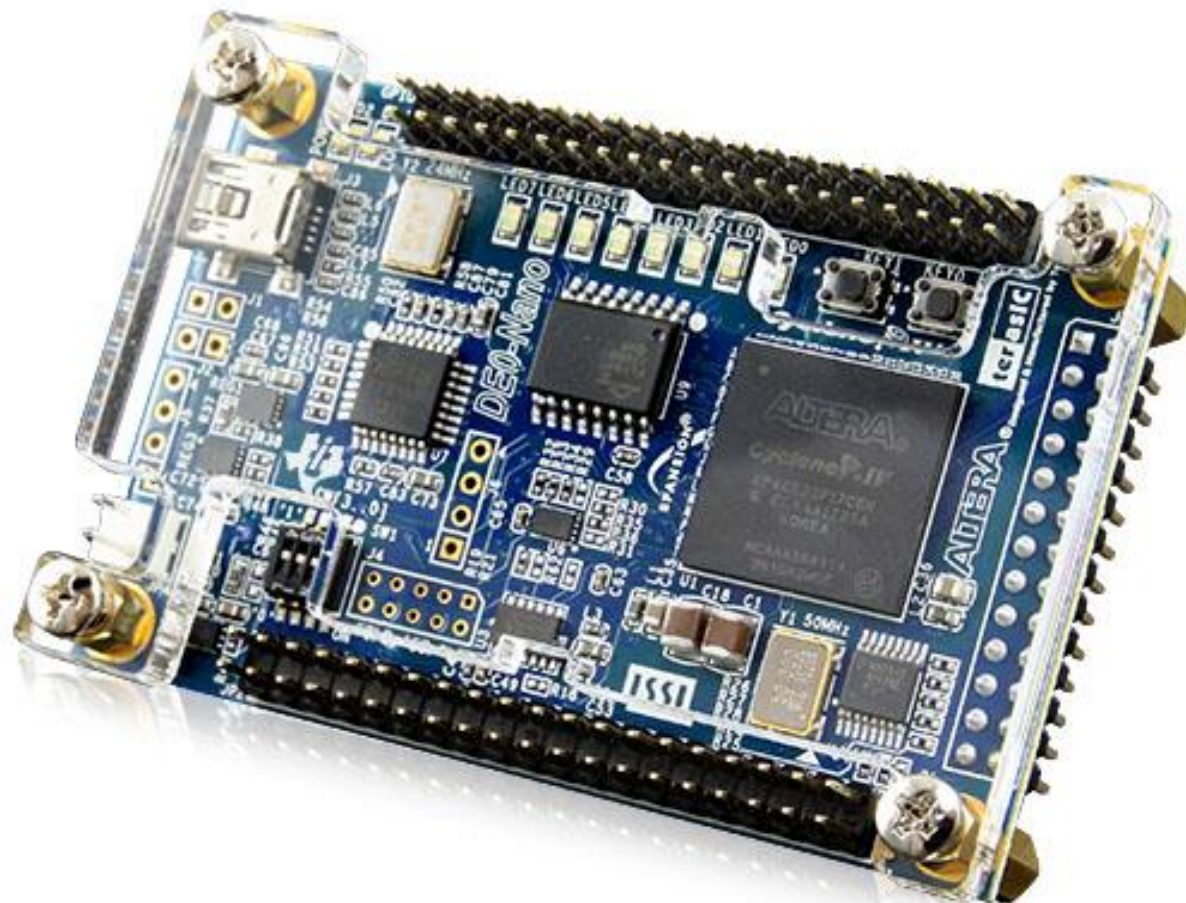
Уровень топологии



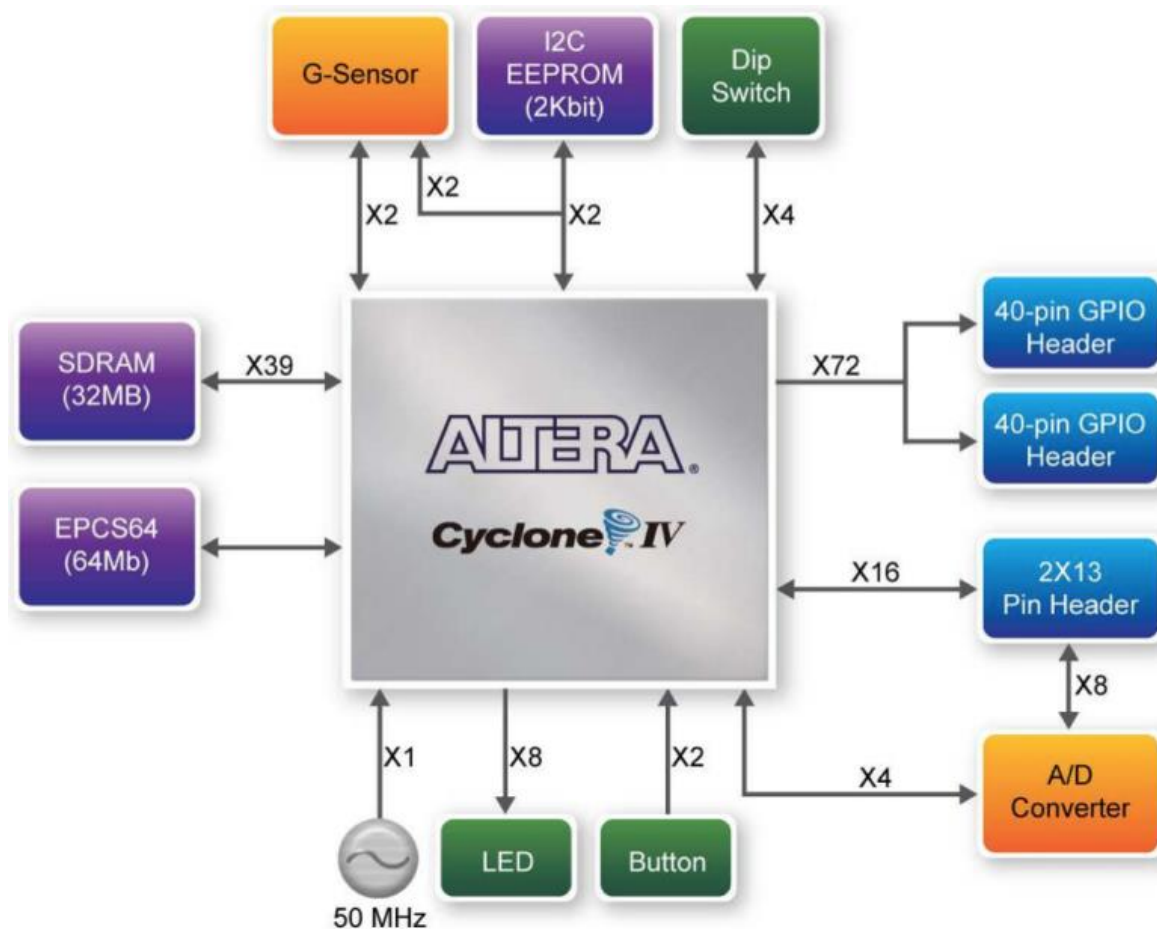
Уровень топологии

- Моделирование топологии (структуры и геометрии всех слоев) проектируемого устройства
- Основные задачи:
 - Design Rule Check (DRC)
 - Layout vs Schematics (LVS)
 - Оптимизация топологии и повышение выхода годных
 - Optical Proximity Correction(OPC)
 - Double/Triple patterning

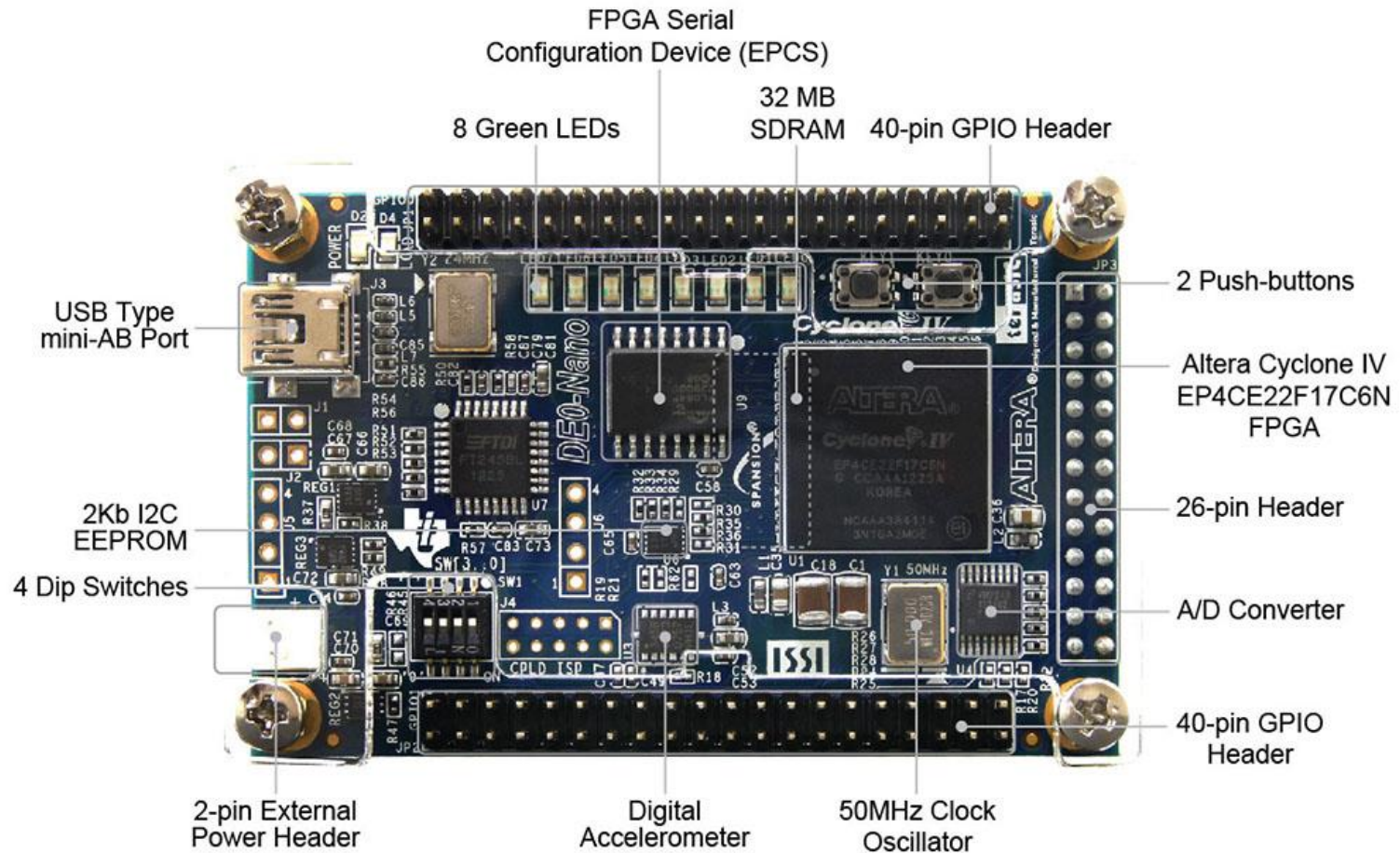
Пример: DE0-Nano



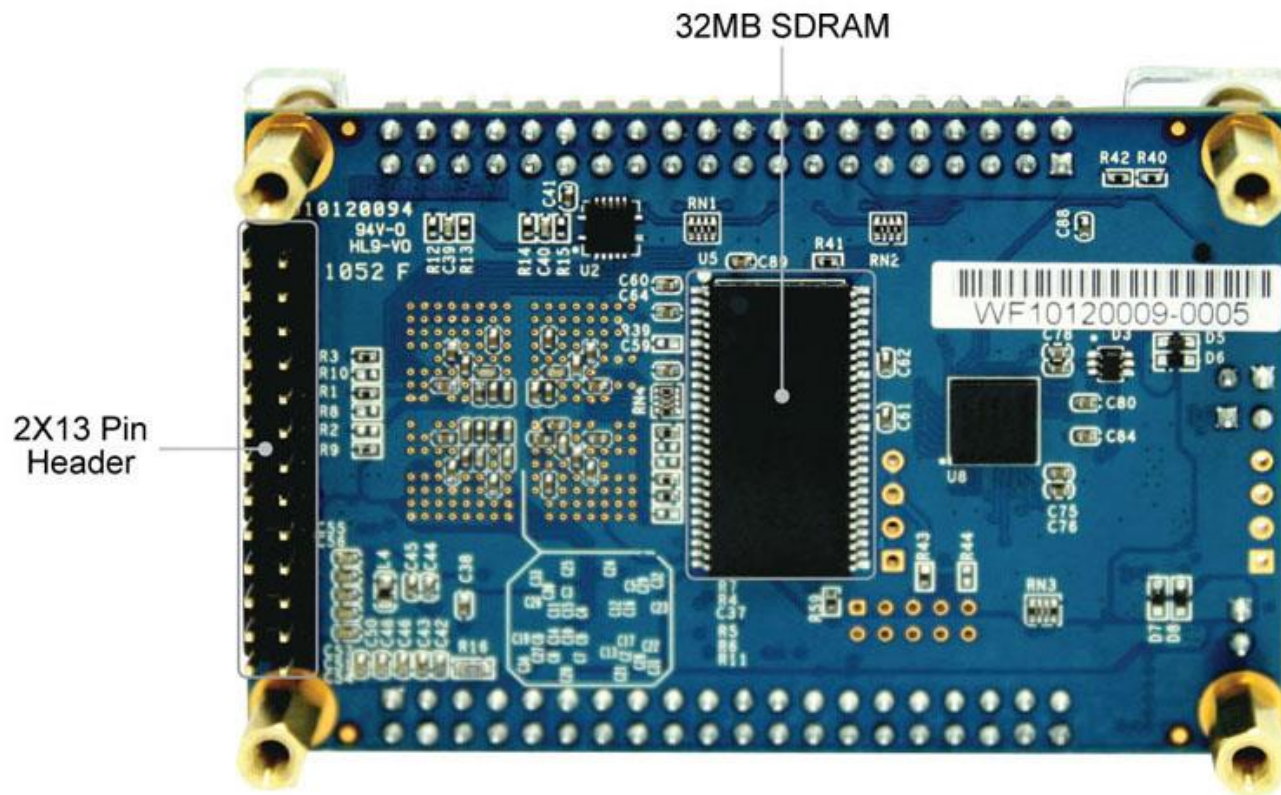
Структура DE0-Nano



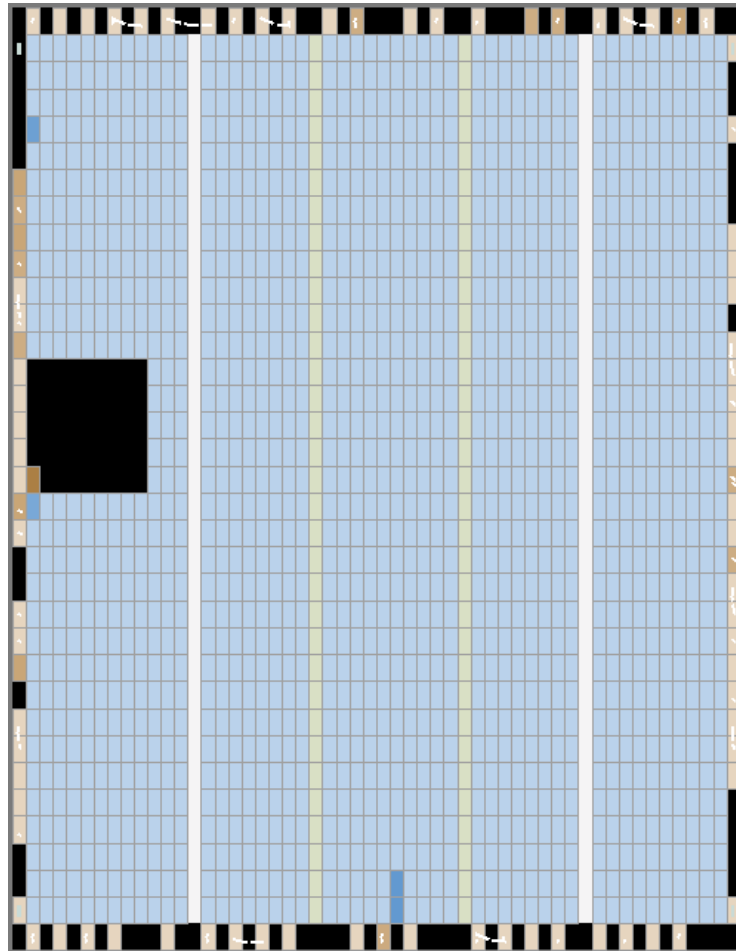
Структура DE0-Nano



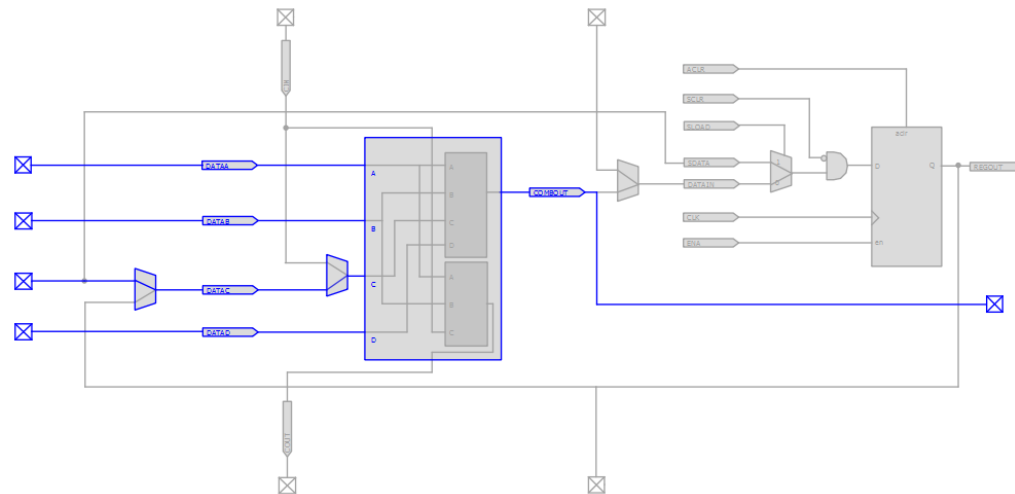
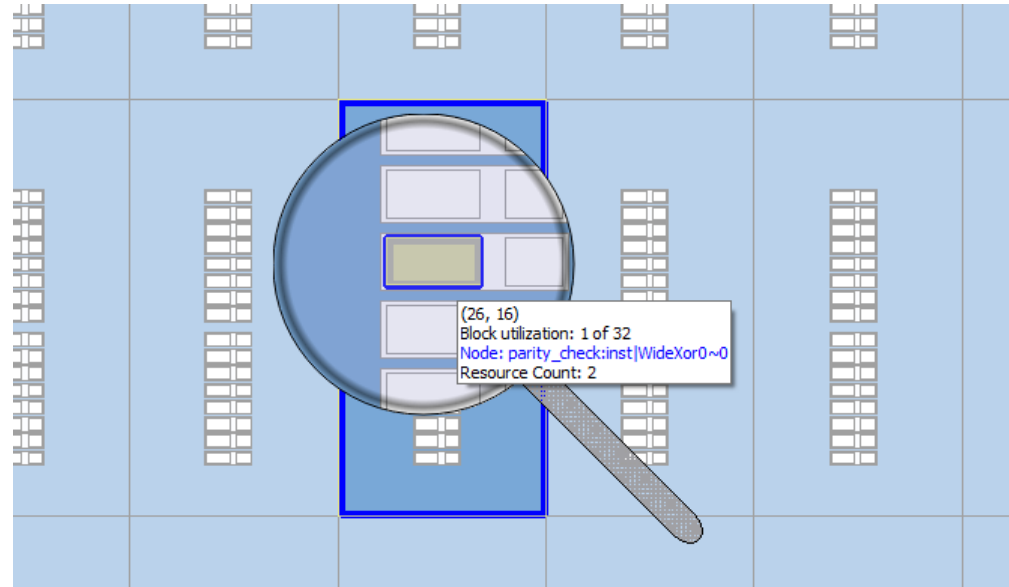
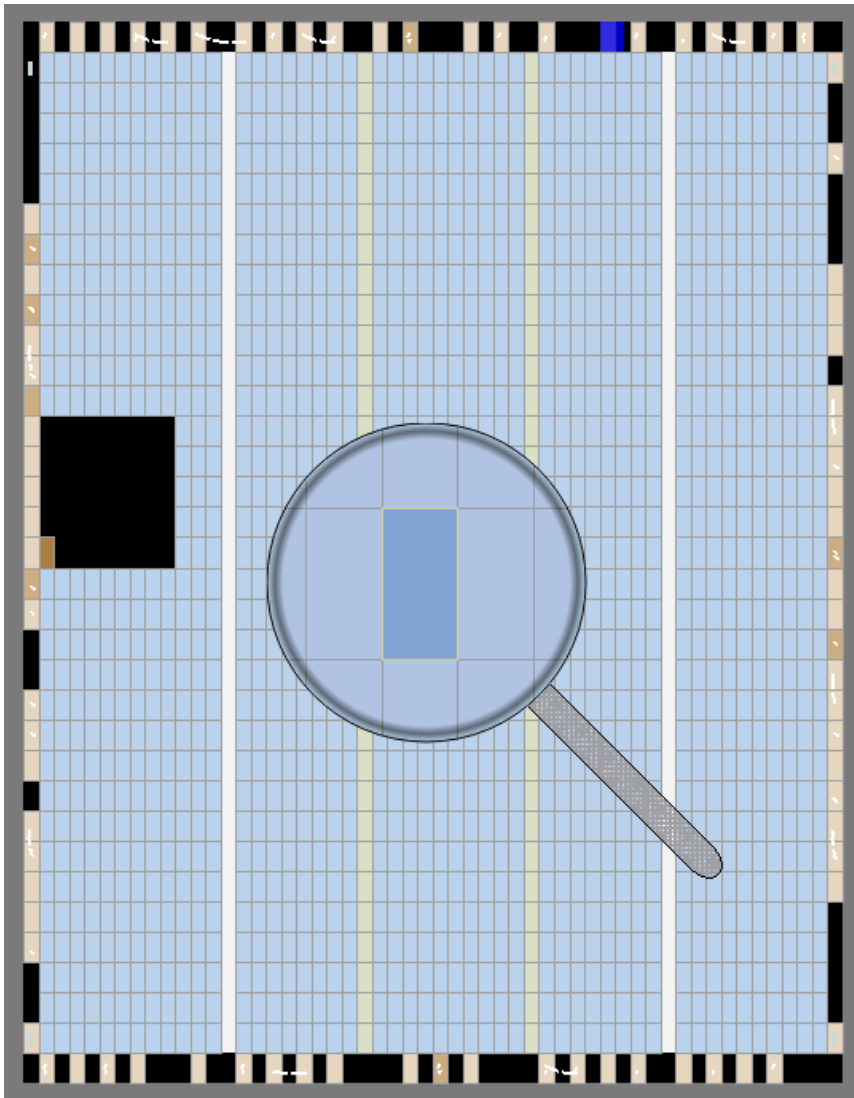
Структура DE0-Nano



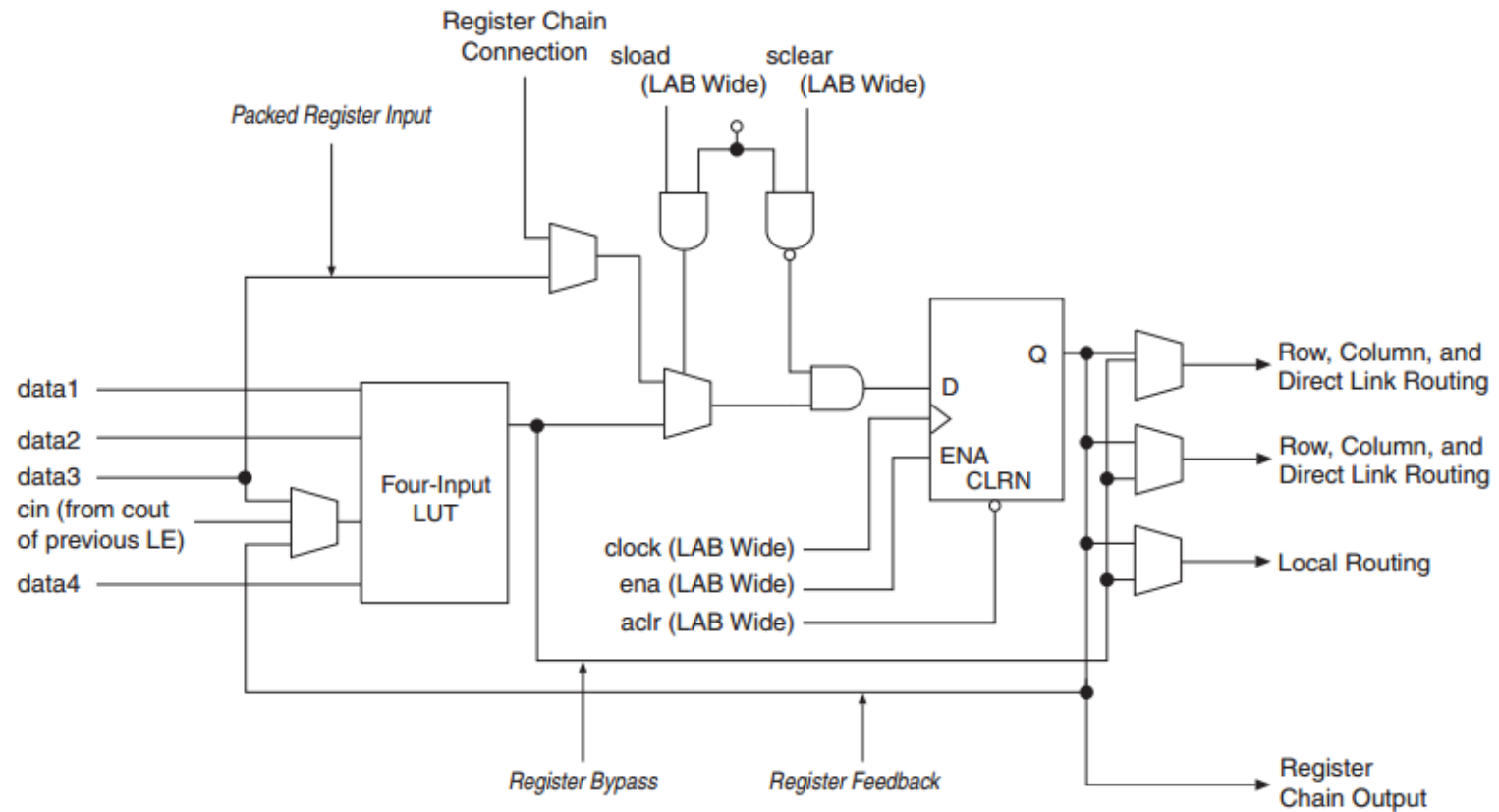
Топология ПЛИС Altera Cyclone IV



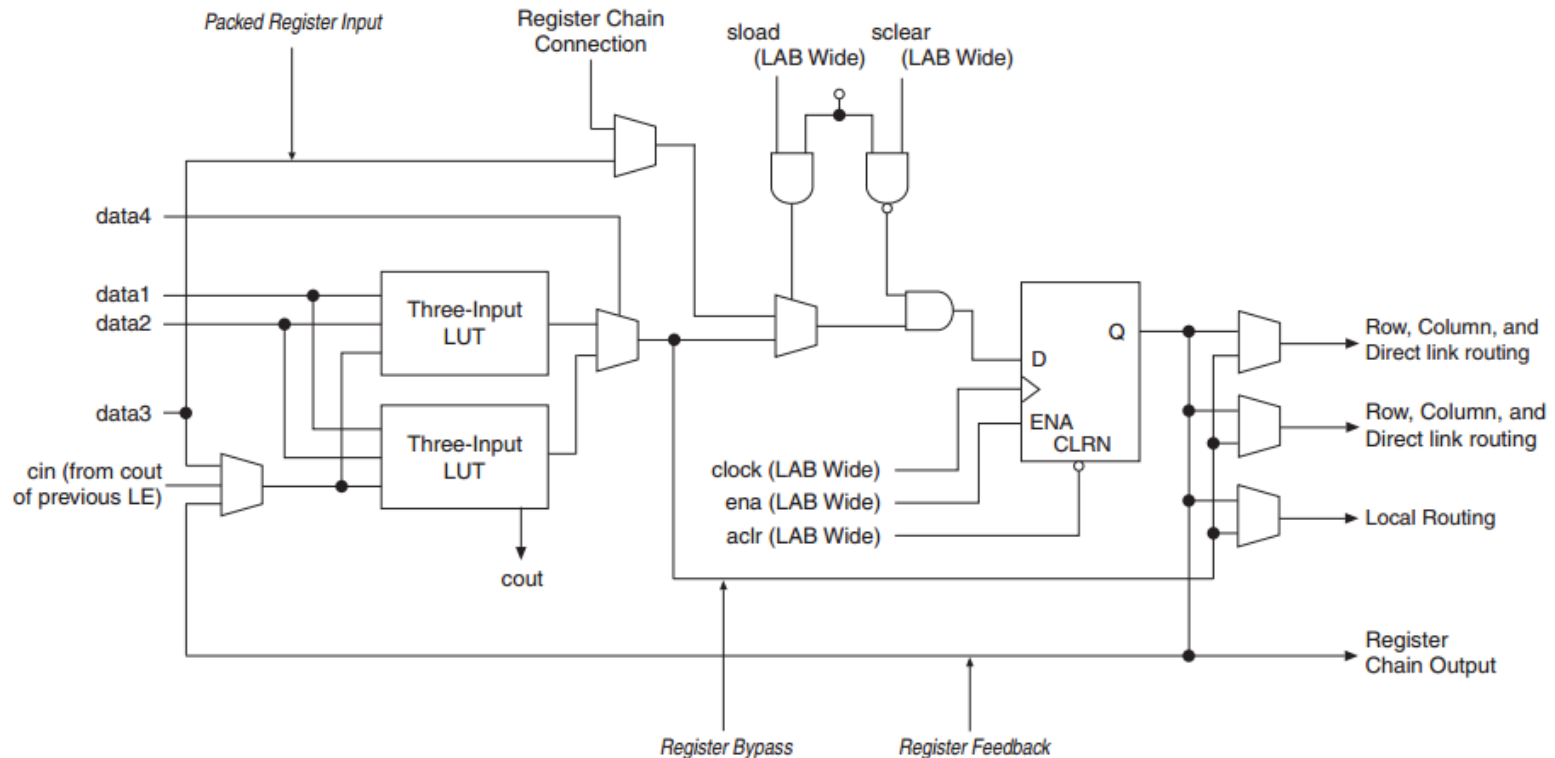
Структура логических элементов Altera Cyclone IV



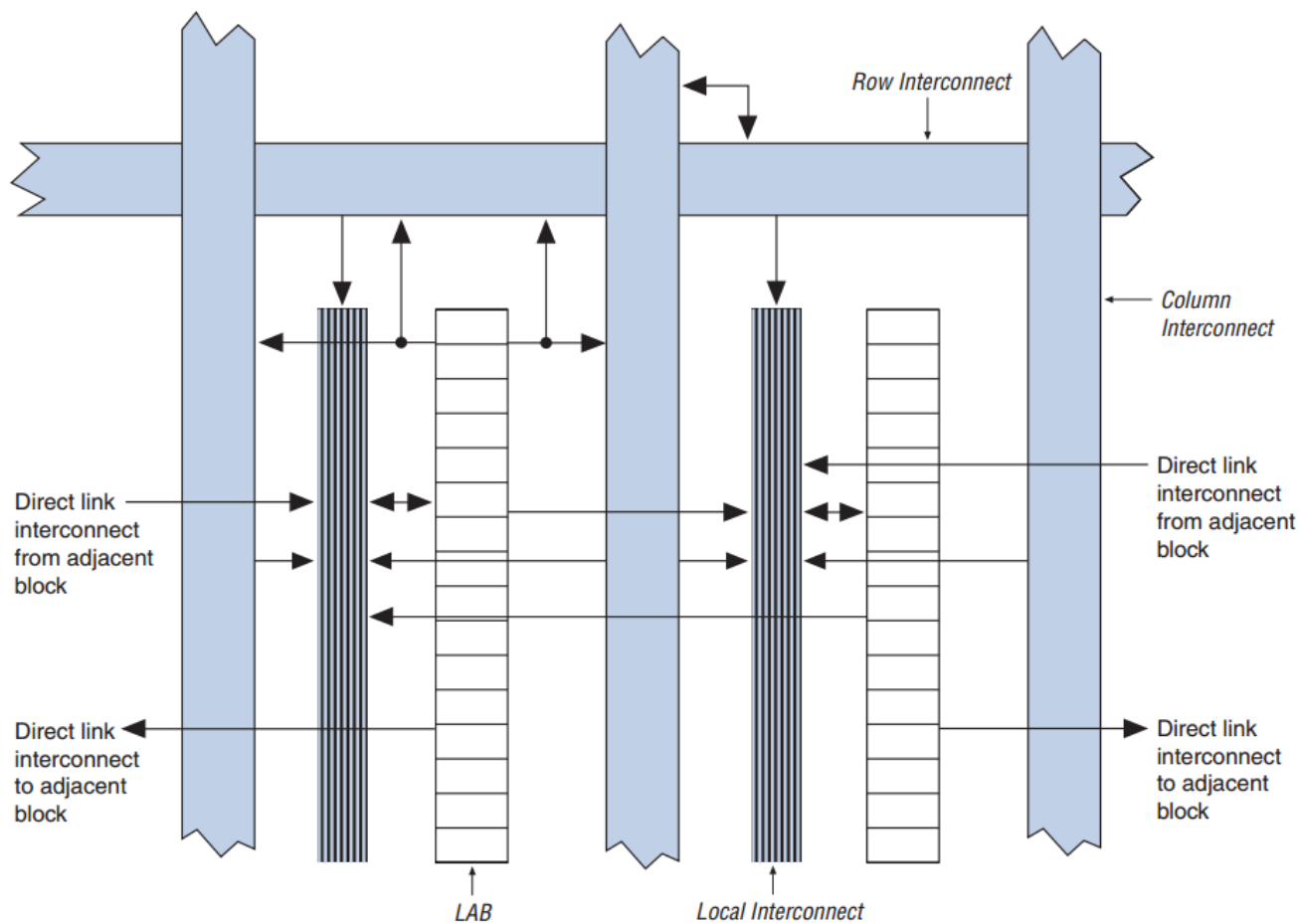
Стандартный режим работы логического элемента Altera Cyclone IV



Арифметический режим логического элемента Altera Cyclone IV



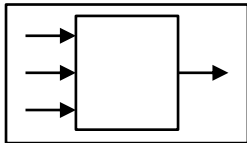
Структура логических массивов Altera Cyclone IV



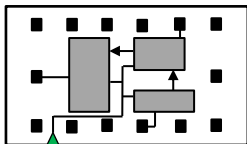
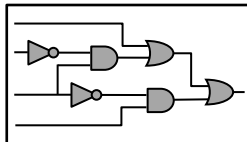
Упрощенный маршрут проектирования цифровых СБИС

Лекция 2

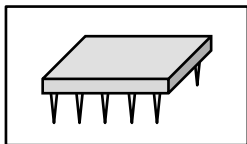
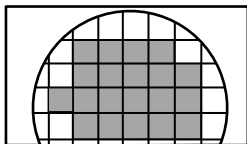
Упрощенный маршрут проектирования



ENTITY test is
port a: in bit;
end ENTITY test;



DRC
LVS
ERC



Спецификация системы

Проектирование архитектуры

Функциональное проектирование

Логическое проектирование

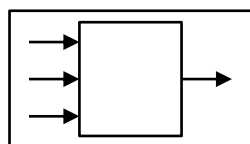
Физическое проектирование

Верификация топологии

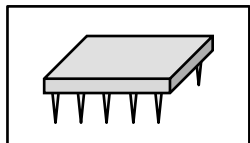
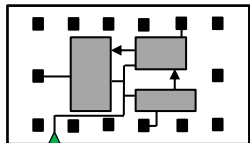
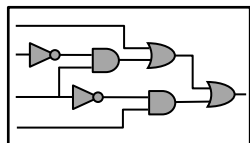
Изготовление

Корпусирование и финальное тестирование

Маршрут проектирования на базе ПЛИС



ENTITY test is
port a: in bit;
end ENTITY test;



Спецификация системы



Проектирование архитектуры



Логическое проектирование



Физическое проектирование

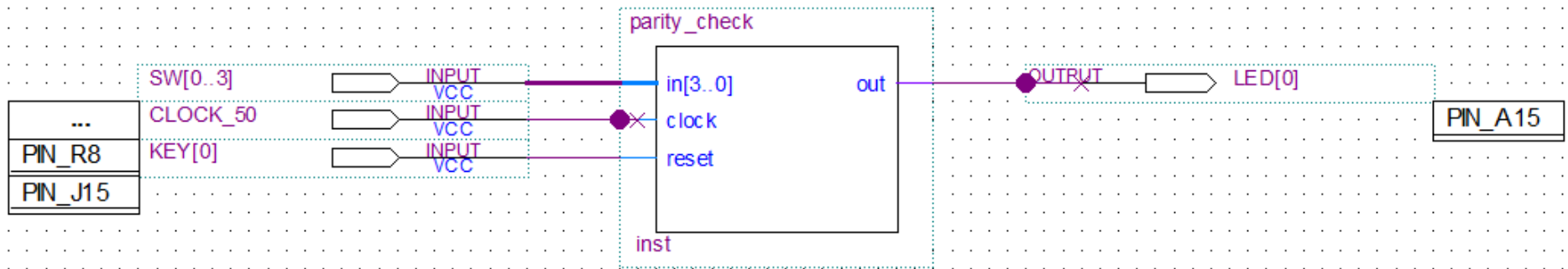


Временной анализ



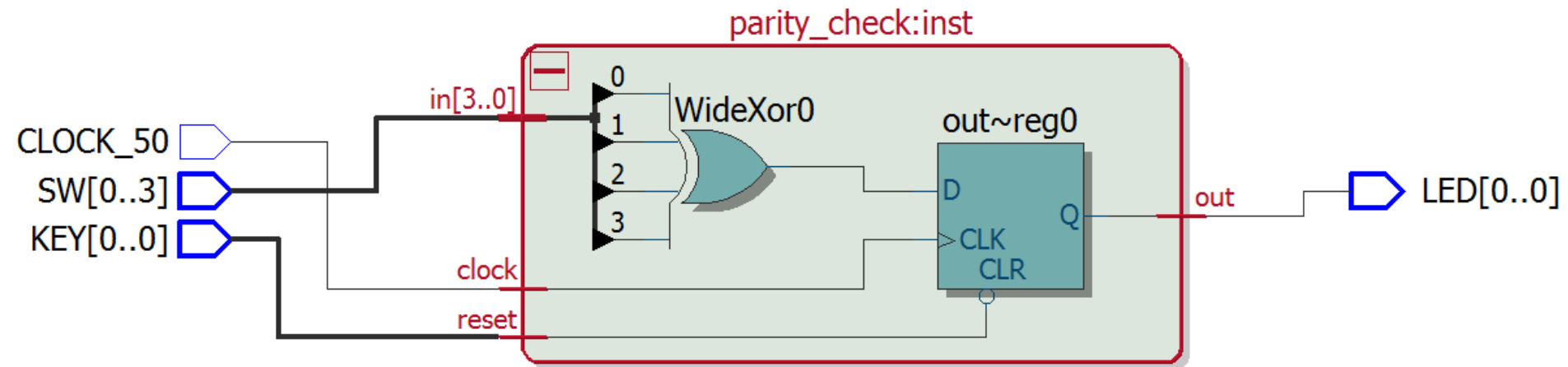
Программирование ПЛИС

Пример: счетчик четности

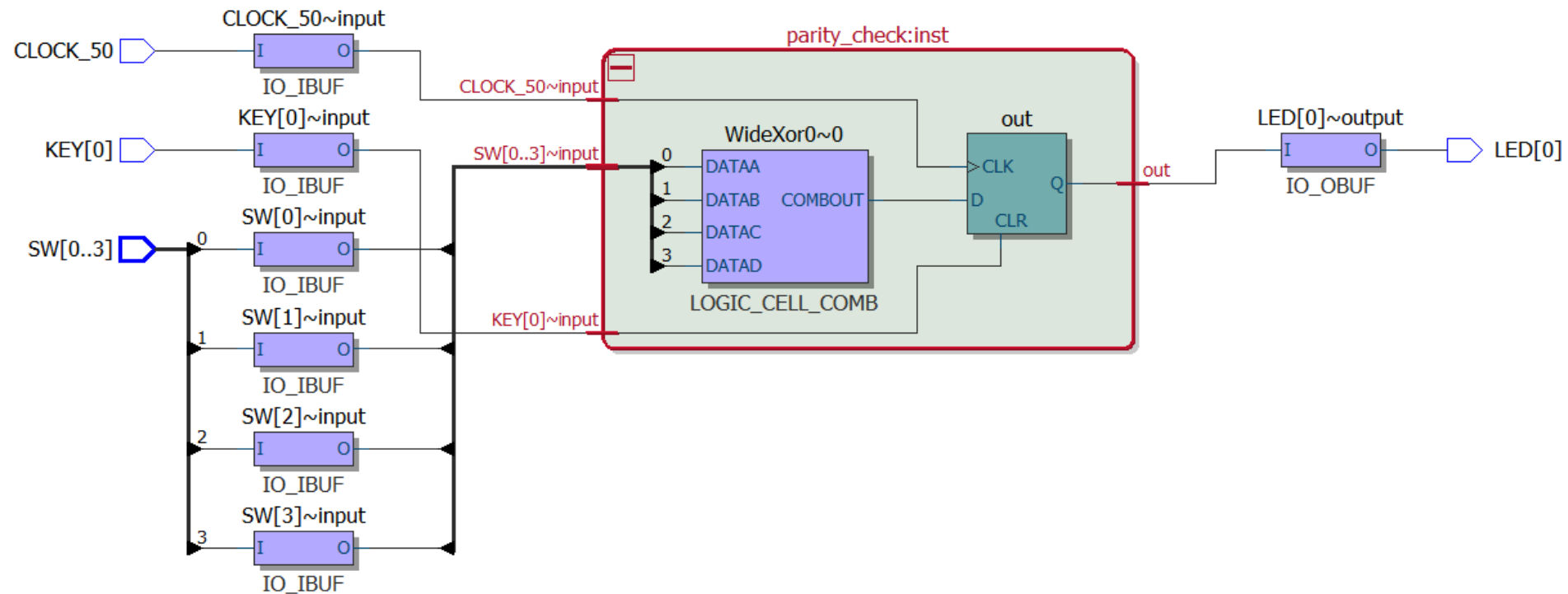


```
1 module parity_check(in, out, clock, reset);
2
3   input wire [3:0] in;
4   input wire clock, reset;
5
6   output reg out;
7
8   always @ (posedge clock or negedge reset)
9     if (~reset)
10      out <= 1'b0;
11    else
12      out <= ^in;
13 endmodule
```

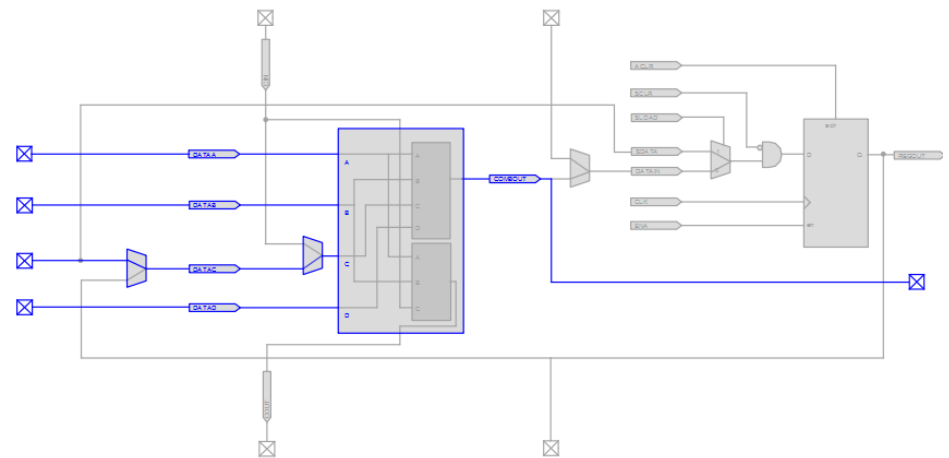
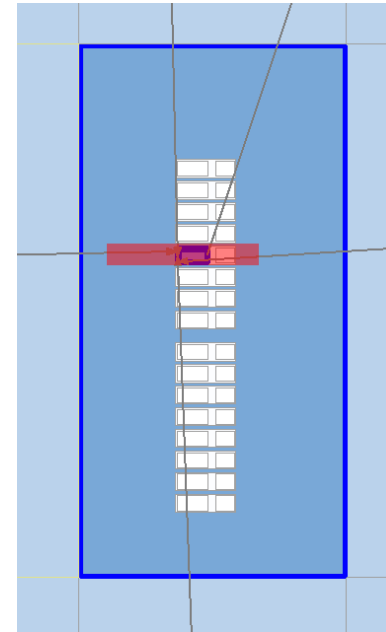
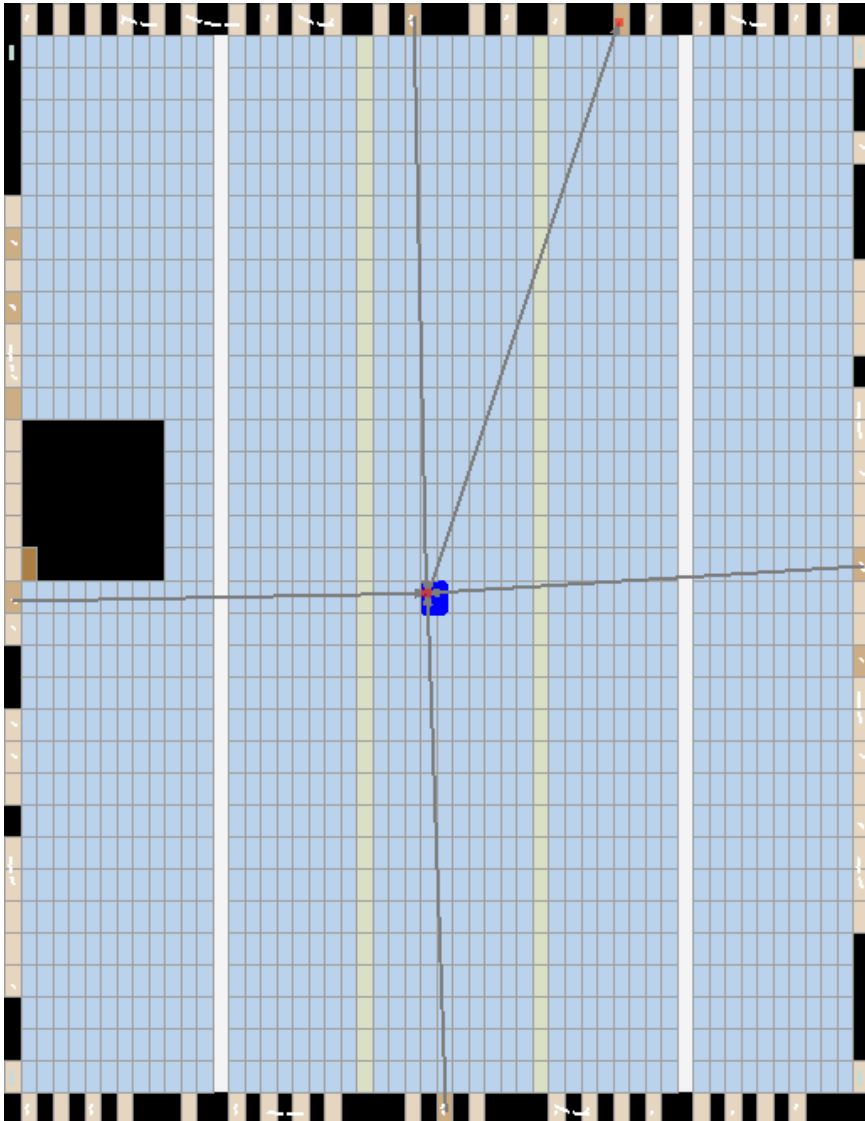
Счетчик четности – логическое проектирование



Счётчик четности – привязка к библиотеке



Счетчик четности – физическое проектирование



Меры качества разработки цифровой интегральной схемы

Лекция 2

Метрики проектирования

- Как оценить «качество» проектируемой интегральной схемы?
 - Цена
 - Надежность
 - Скорость/производительность(задержка, частота работы)
 - Энергопотребление