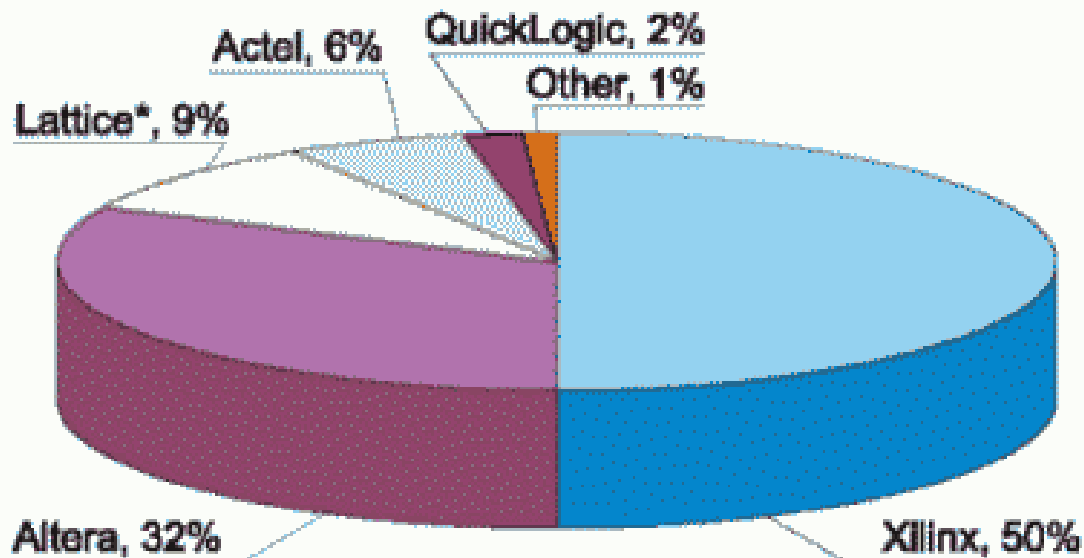


# Как FPGA это делает ?

Дмитрий Смехов

# Компании Xilinx и Altera



## Xilinx

- Основана в 1984

## Altera

- Основана в 1983

2015 — компания Intel покупает Altera

2020 — компания AMD покупает Xilinx

ASIC - application-specific integrated circuit

Интегральная схема для конкретного применения

FPGA — field-programmable gate array

CPLD — complex programmable logic device

ПЛИС — программируемая логическая интегральная схема

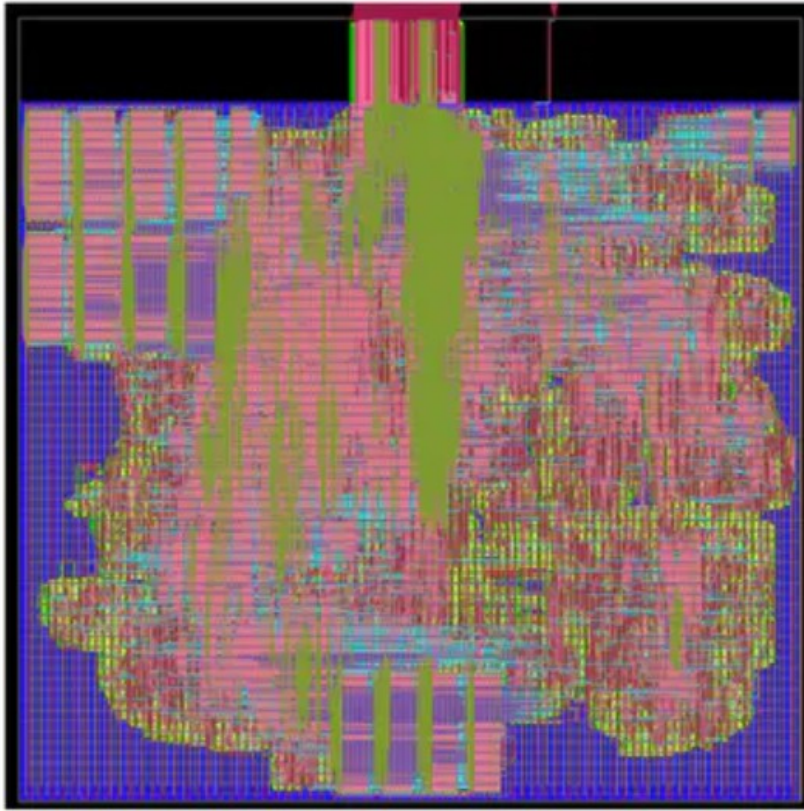
# Структура ASIC

Есть поле которое надо заполнить элементами.

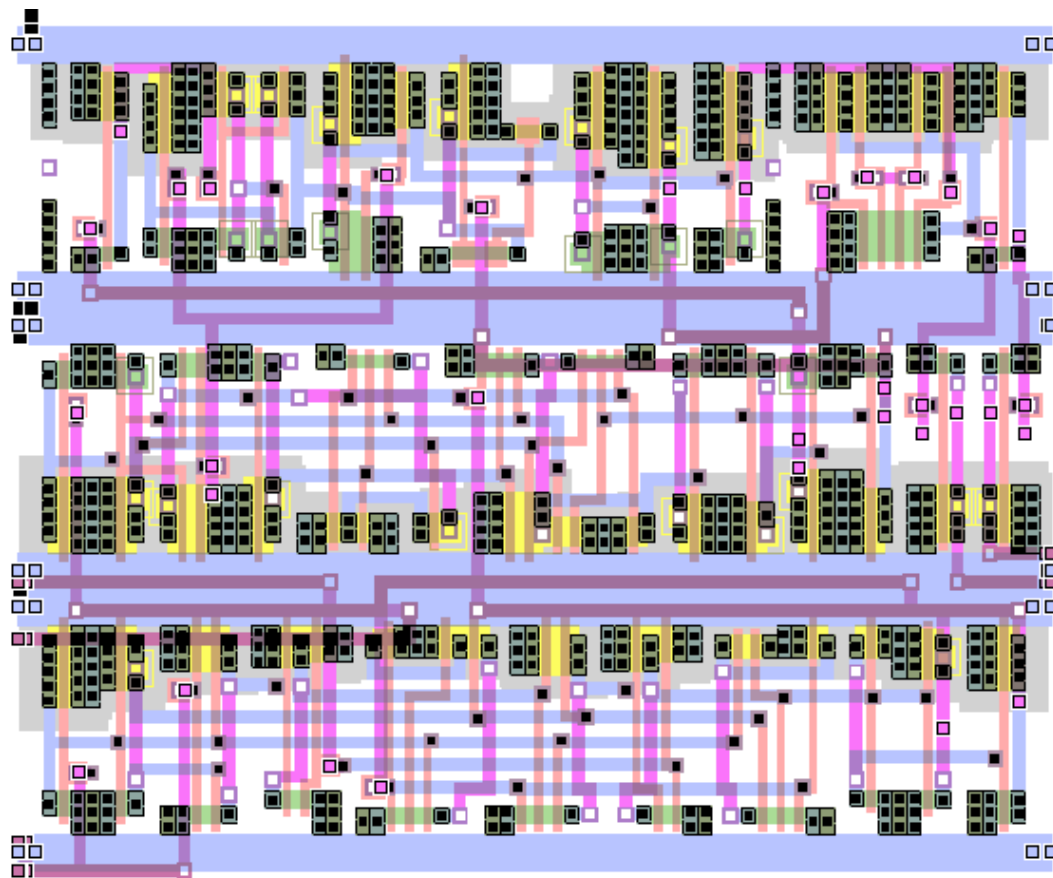
Каждый элемент занимает площадь и представлен в виде транзисторов и проводников

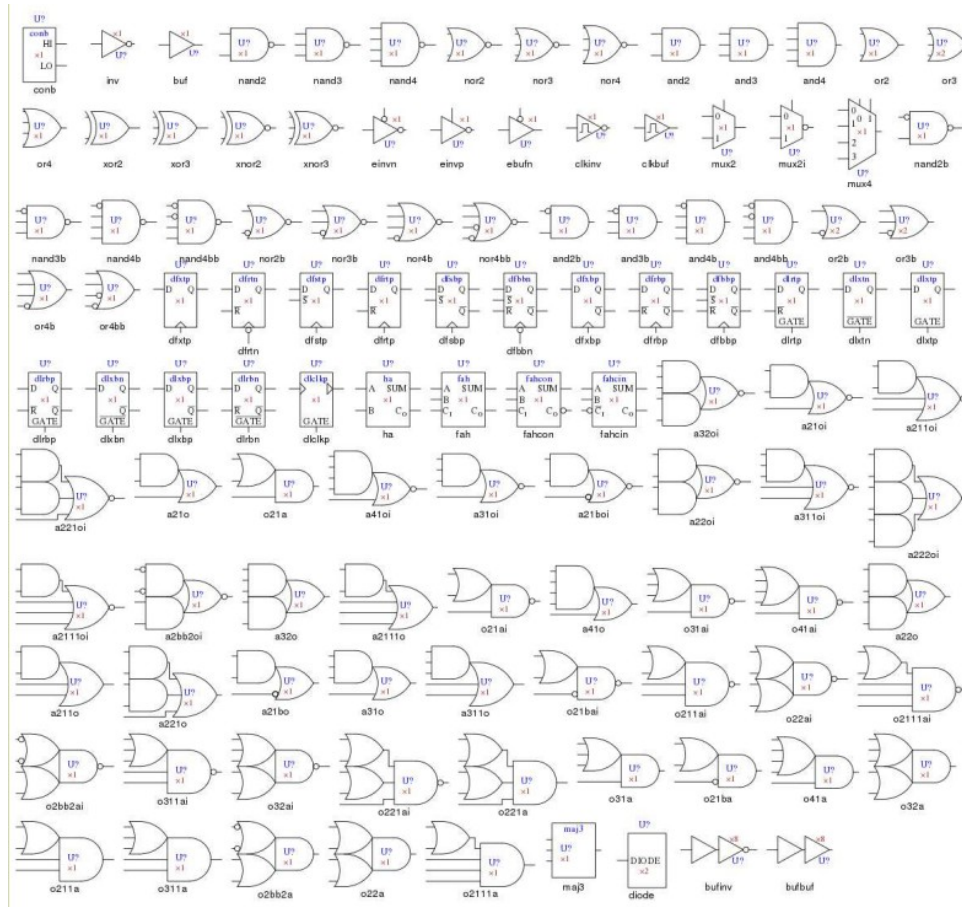
Доступные элементы

- Standard Cell Libraries
- IP Core

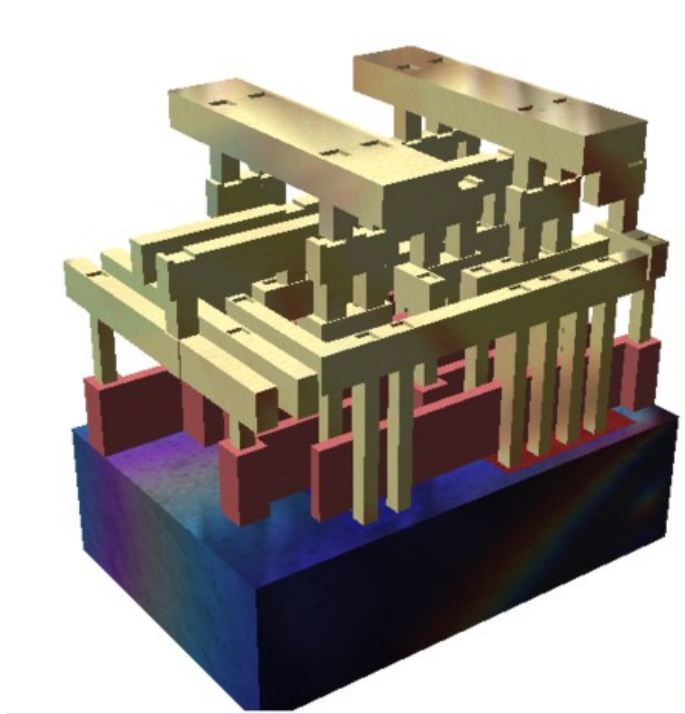


# Выравнивание по рядам





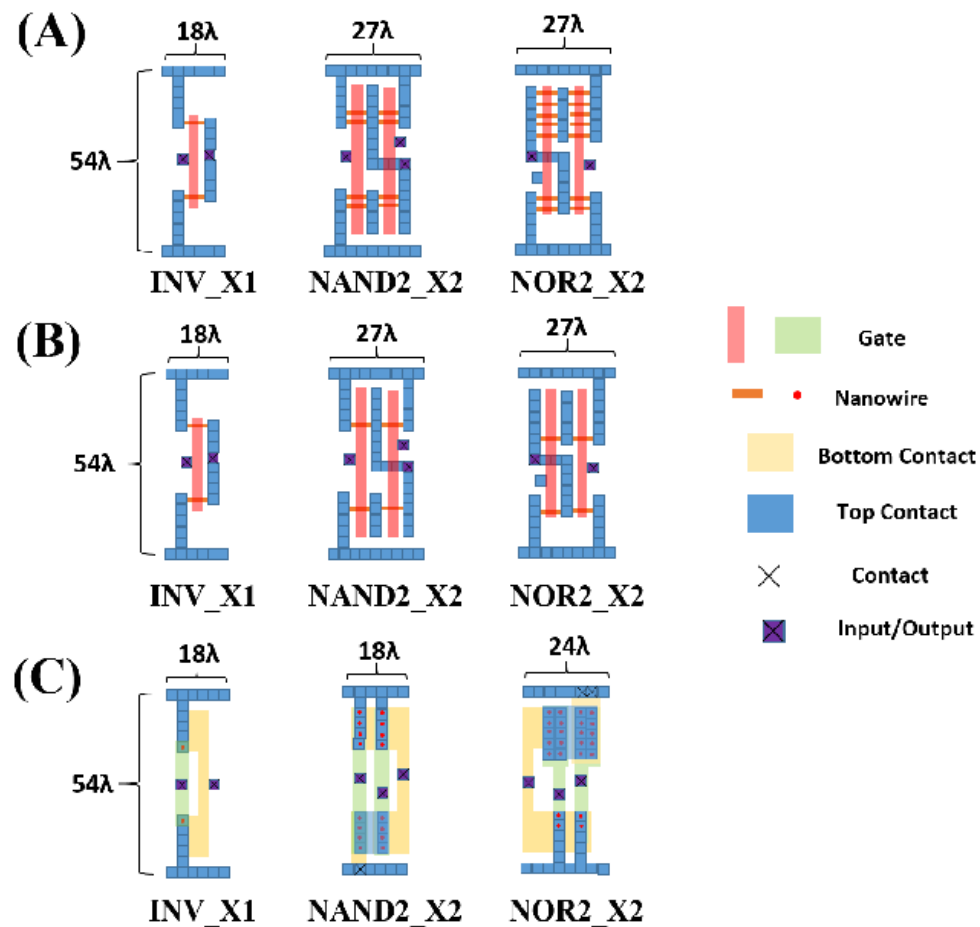
# Пример элемента



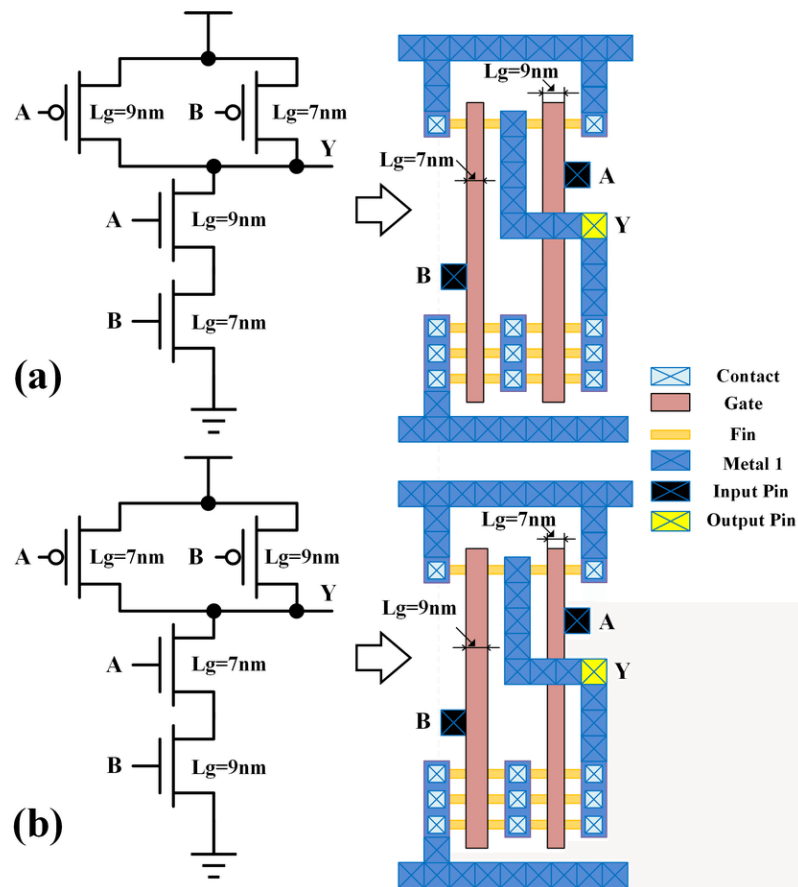
[https://en.wikipedia.org/wiki/Standard\\_cell](https://en.wikipedia.org/wiki/Standard_cell)



# Некоторые элементы



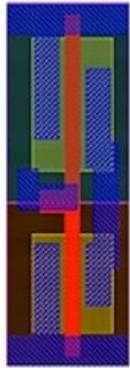
# СВЯЗЬ со схемой



# Различные типы библиотек

## High Density

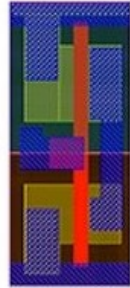
9 or 10-tracks high cells



Balanced transistor size  
for high density and good  
performance, low power

## Ultra-High Density

7 or 8-tracks high cells



Small transistors for high  
density and low power

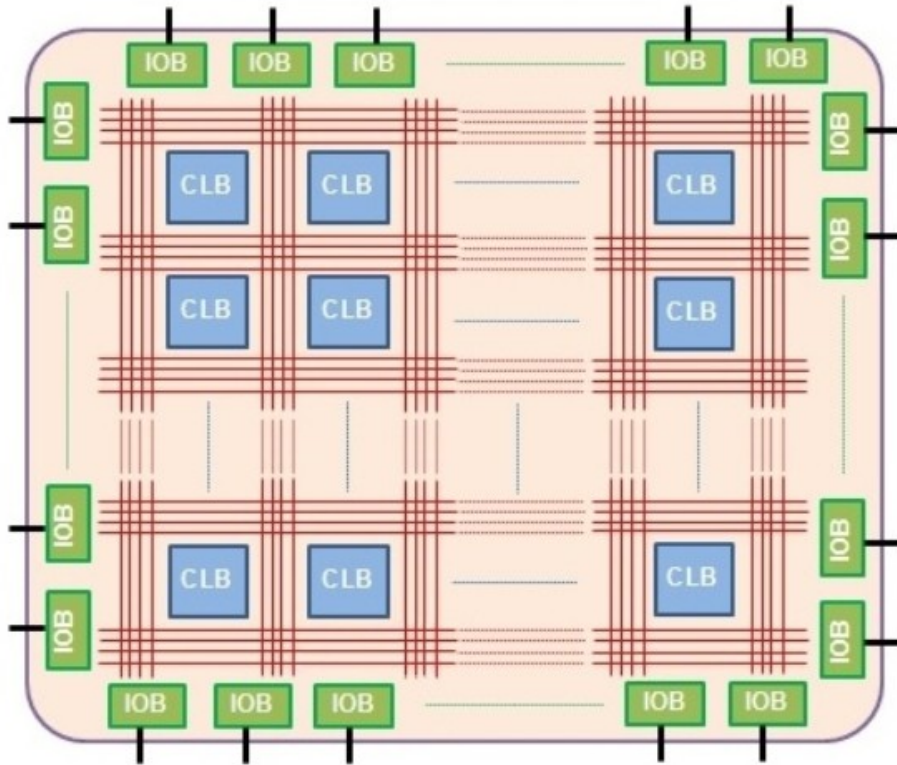
## High Performance

12-tracks high cells



Large transistors for optimal  
speed, but also low power  
features

# Структура FPGA



На физическом уровне м/с не изменяется

Основные компоненты:

- Блоки ввода-вывода
- Матрица соединений
- Конфигурируемый логический блок

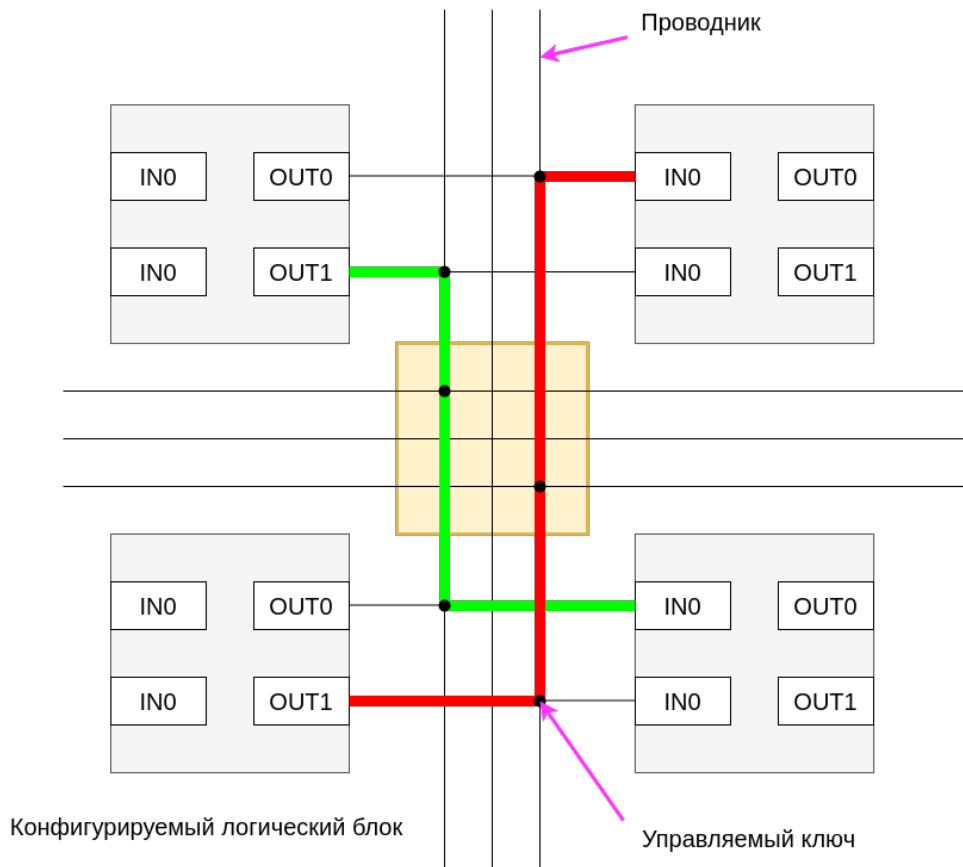
FPGA устанавливается на плату и взаимодействует с другими м/с. Требуется согласовать стандарт и направление передачи сигналов.

При неправильном конфигурировании возможно повреждение м/с.

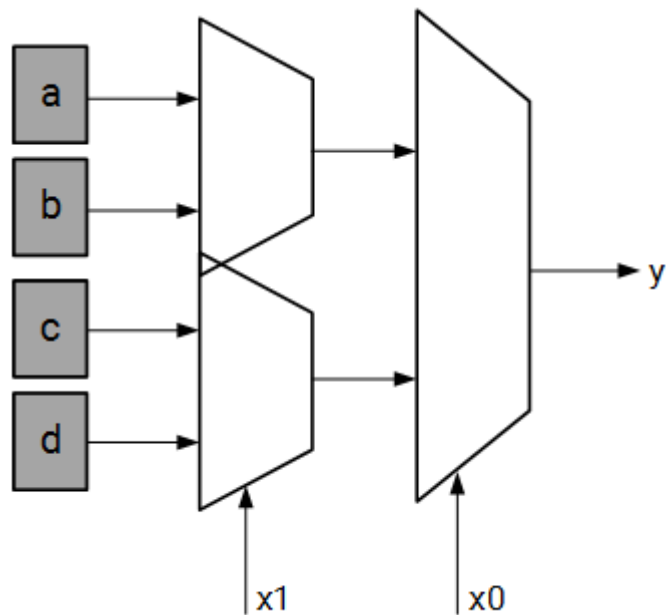
Современные ПЛИС поддерживают много стандартов:

- LVDS, LVCMOS и т.д.

# Матрица соединений

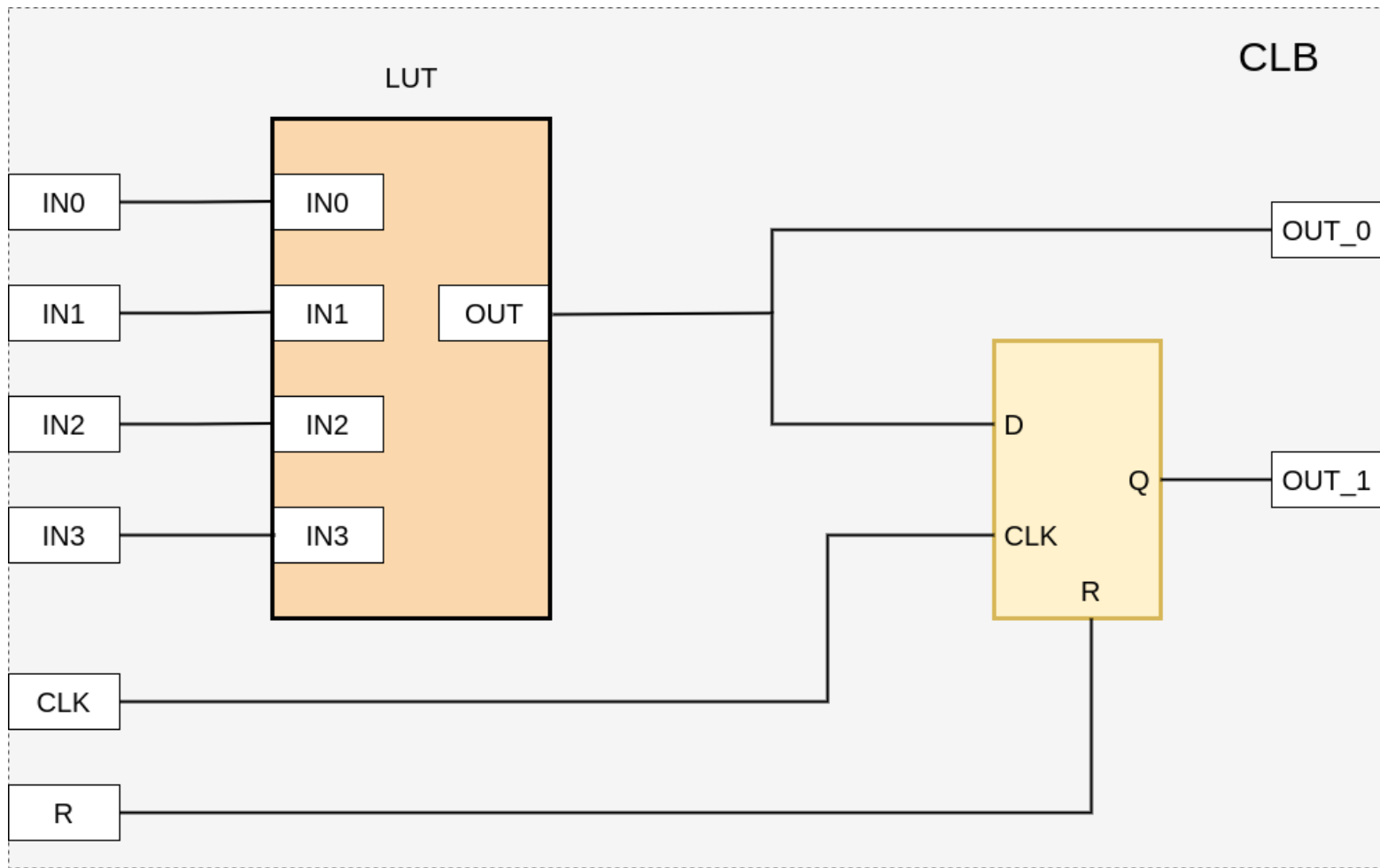


# Произвольная функция двух переменных



X13469-102417

# Конфигурируемый логический блок





## Таблица истинности для LUT

IN0	IN1	IN2	IN3	OUT
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3

1	1	0	1	D13
1	1	1	0	D14
1	1	1	1	D15

LUT - Look up table

Это память, требуется инициализация

Варианты:

- LUT на шесть входов
- Работа в качестве памяти
- Работа в качестве сдвигового регистра

Требуется загрузить конфигурацию

Варианты:

- Из встроенной FLASH памяти (характерно для CPLD)
- Через последовательную или параллельную шину из ПЗУ
- Через отладочный интерфейс JTAG
- Перезагрузка только части ПЛИС через работающий интерфейс той же самой ПЛИС

ПЛИС может содержать дополнительные аппаратные блоки

Варианты:

- PLL, MMCM — обеспечивают формирование тактовых частот
- Трансиверы — обеспечивают на скоростях 0.6 — 400 Гбит/с
- Процессор — как правило ARM
- Контроллер DDR — поддержка памяти DDR3, DDR4
- Контроллер PCIe — поддержка шины PCI Express
- Память HBM — динамическая память внутри м/с

Компания AMD/Xilinx

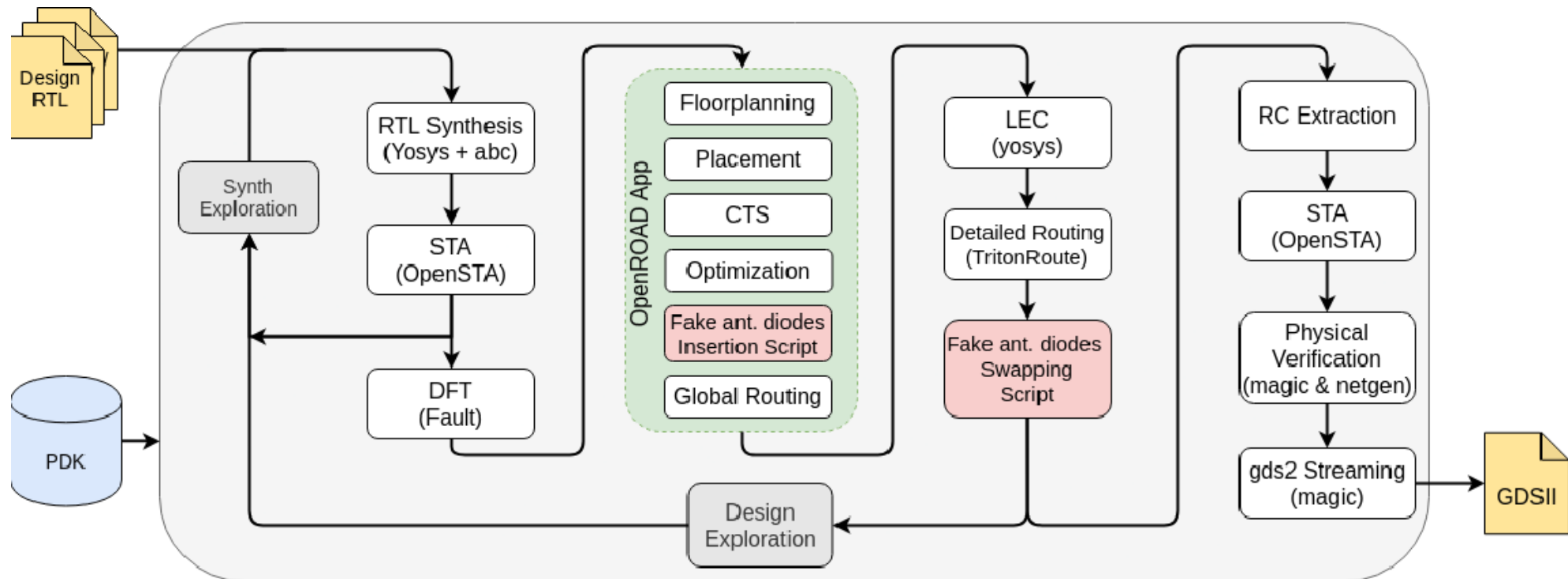
- Серия Zynq 7000 SoC
  - Single or Dual ARM Cortex-A9
- Серия Zynq UltraScale+ MPSoC
  - Dual or Quad Arm Cortex-A53
  - Dual Arm Cortex-R5F
  - Arm Mali-400MP2
- Серия Zynq UltraScale+ RFSoc
  - Quad Arm Cortex-A53
  - Dual Arm Cortex-R5F
  - RF-ADC, RF-DAC

Новый класс:

ACAP — Adaptive Compute Acceleration Platform

Основные свойства:

- Процессор: Dual-core Cortex-A72, Dual-core Cortex-R5
- FPGA
- **Матрица из 400 процессоров**
- NoC — network on chip
- Контроллеры: PCIe, DDR, MAC



## How to Design Your Own Chip – For FREE

Verilog File  
(Hardware  
description  
language)

```
default_nettype none
module inverter (
  input wire in,
  output out );

  assign out = !in;

endmodule
```

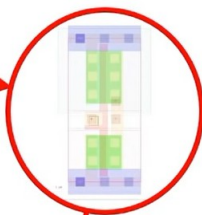
Openlane  
(software to create GDS – Silicon  
“Gerber” Files)

+

Skywater Open Source PDK  
(Process Design Kit e.g. Chip  
Design Rules, libraries, etc)

+

SKY130 Process Node  
(Chip Manufacturing Technology)



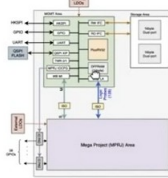
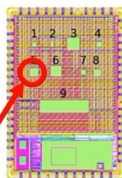
Efabless  
(Company manufacturing  
custom chips)

+

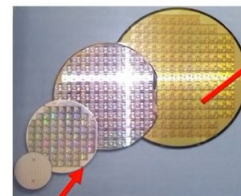
Open MPW  
Shuttle Program  
(Google sponsored chip  
manufacturing)

+

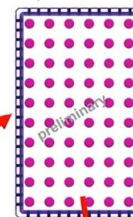
Caravel “harness” SoC  
(Chip Template)

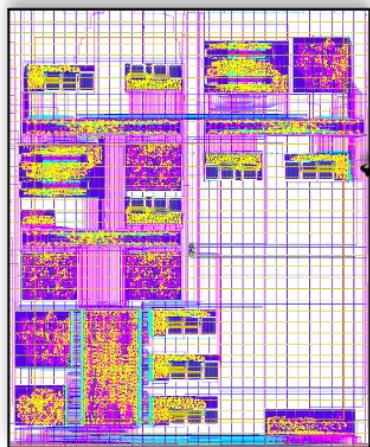


SkyWater Technology  
(Silicon manufacturing company –  
Semiconductor fabrication plant)

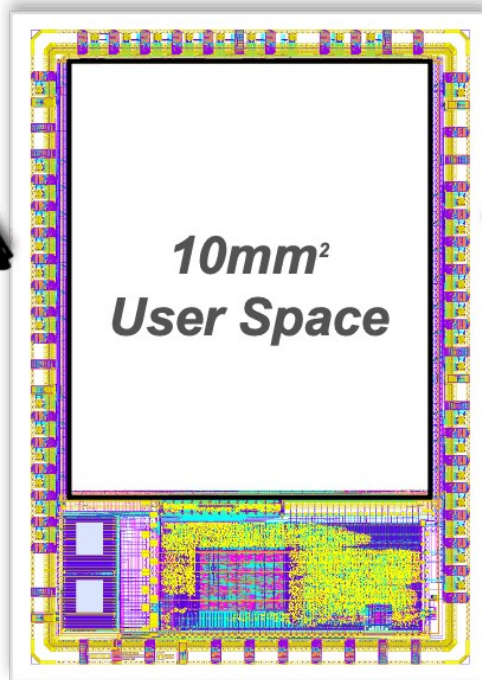
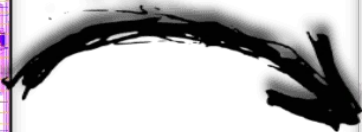


You will get:  
Chip + Board

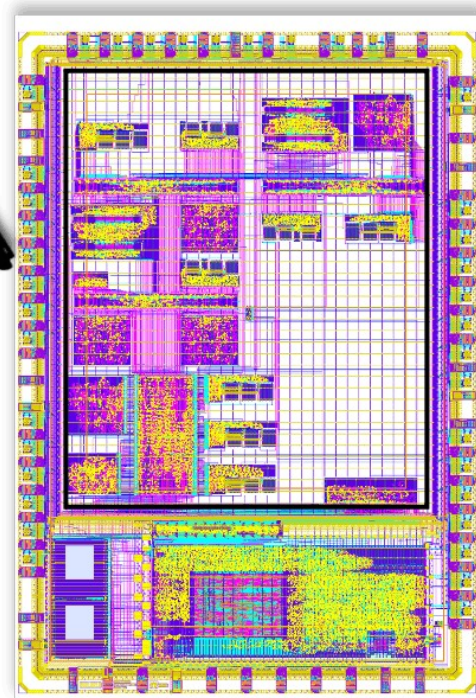




YOUR DESIGN



CARAVEL



INTEGRATED CHIP



Verilog:

**wire** - провод, значения 0, 1, X, Z и ещё 116

**reg** - регистр, 0,1,X,Z

SystemVerilog:

**logic** - объединяет wire и reg, значения 0,1,X,Z

**bit** - значения 0,1

А также:

- **byte, shortint, int, longint, string**
- **struct, interface, class**

# Битовый вектор

```
logic          a1; // простой сигнал
logic [31:0];   a2; // шина шириной 32 разряда
logic [7:0][31:0]; a3; // восемь шин по 32 разряда

logic [31:0]    a4[8]; // массив из восьми шин по 32 разряда

logic [7:0]      b1,b2;

assign b1 = a2[7:0];           // разряды 7:0
assign b2 = a2[16+:8];         // разряды 23:16
```

## Сравнение двух значений

- Операторы == и !=
- Операторы === и !==
- Операторы ==? и !=?
- Оператор inside

```
logic a1 = 4'010X  
logic a2 = 4'010X  
logic c1,c2
```

a1==a2 - результат X (неопределён)  
a1==a2 - результат 1 (истина)  
a1=? 4'0?1? - результат 1 (истина)

```
int b1;  
b1 inside { 5, 6, 8 }  
b1 inside { [10:20] }
```

# Операторы if else, case, casex, casez

```
if( rst )  
    b <= #1 0;  
else  
    b <= #1 c;
```

```
if( rst ) begin  
    b1 <= #1 0;  
    b2 <= #1 0;  
else begin  
    b1 <= #1 c1;  
    b2 <= #1 c2;  
end
```

```
case( adr[3:0] )  
    4'h0: reg_a <= #1 data;  
    4'h1: reg_b <= #1 data;  
    4'h2: reg_c <= #1 data;  
endcase
```

```
casez( interrupt_line )  
    4'b1???: num <= #1 4;  
    4'b01??: num <= #1 3;  
    4'b001?: num <= #1 2;  
    4'b0001: num <= #1 1;  
endcase
```

```
assign c = (a==2) ? 1 : 0
```

# always\_ff, always\_comb, always\_latch

```
module();
```

```
...
```

```
assign a = b;
```

```
always_ff @(posedge clk) begin
```

```
....
```

```
end
```

```
always_comb begin
```

```
...
```

```
end
```

```
endmodule
```

SystemVerilog — язык описания аппаратуры

- требуются средства для описания параллельных процессов
- при синтезе формируется параллельная структура
- при моделировании совмещаются последовательные и параллельные процессы
- операторы внутри always блока выполняются последовательно, в течении одного момента модельного времени