

Краткое введение в ПЛИС/FPGA

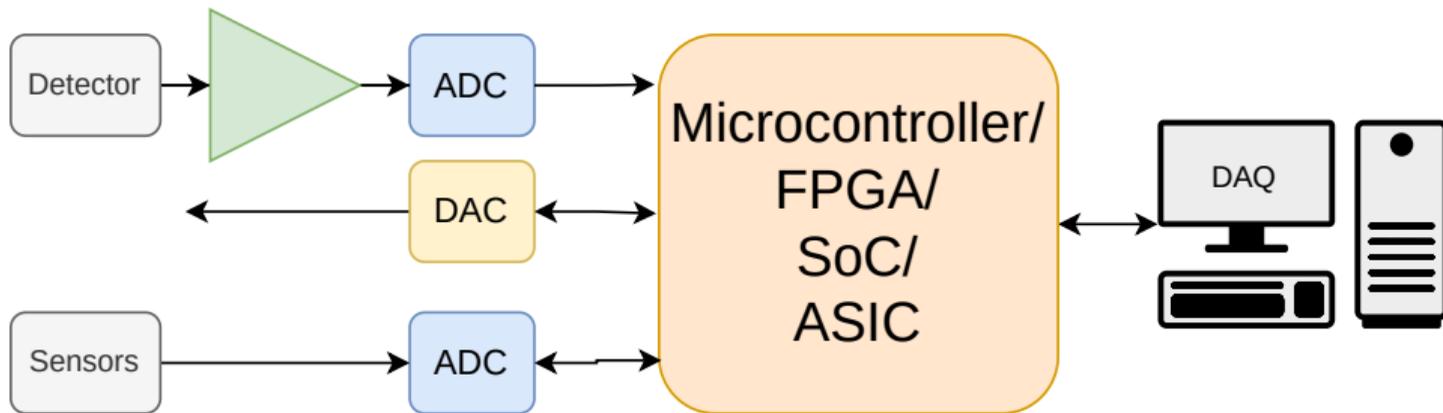
В.Шебалин

25.06.2025,

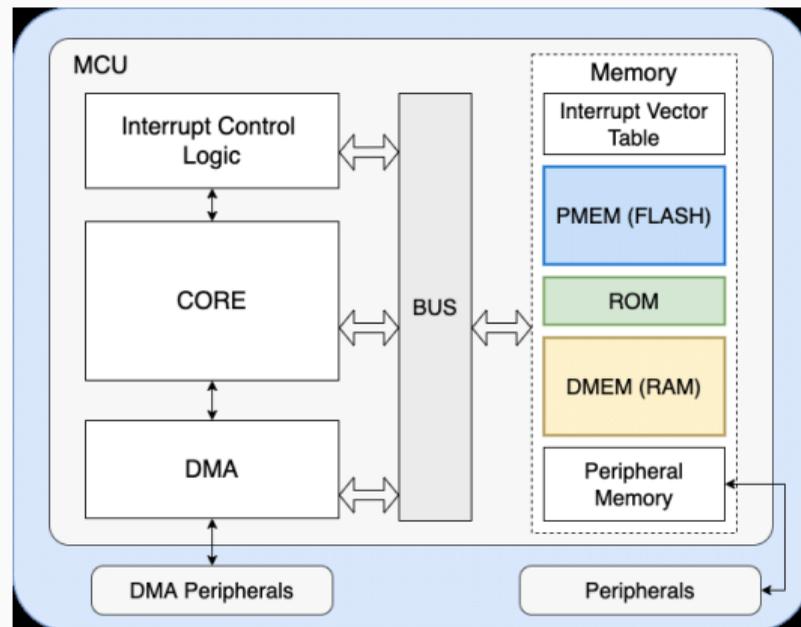
Выездная конференция дискуссионного клуба ФЭЧ ИЯФ на базе отдыха Разлив
(ВКДКФЭЧИЯФнБОР)

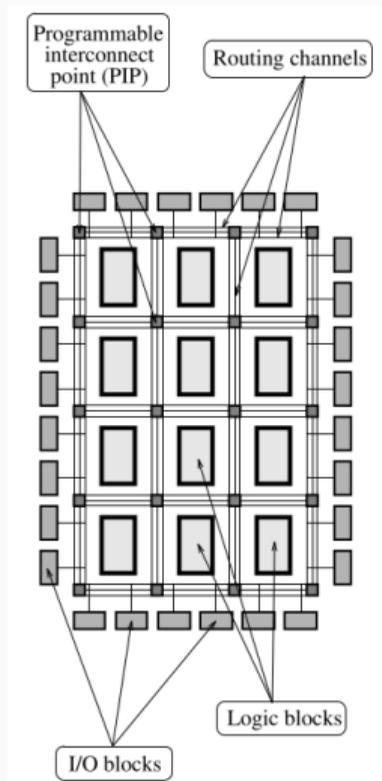
ИЯФ СОРАН

- **FPGA**(Field Programmable Gate Array) = **ПЛИС**(Программируемая логическая интегральная схема - более корректно использовать по отношению к CPLD) = **ППВМ**(Программируемая пользователем вентиляльная матрица)
- Наряду с микроконтроллерами и СБИС(ASIC) широко используется в считывающей электронике в экспериментах ФЭЧ.
- Disclaimer: mixed language design below.



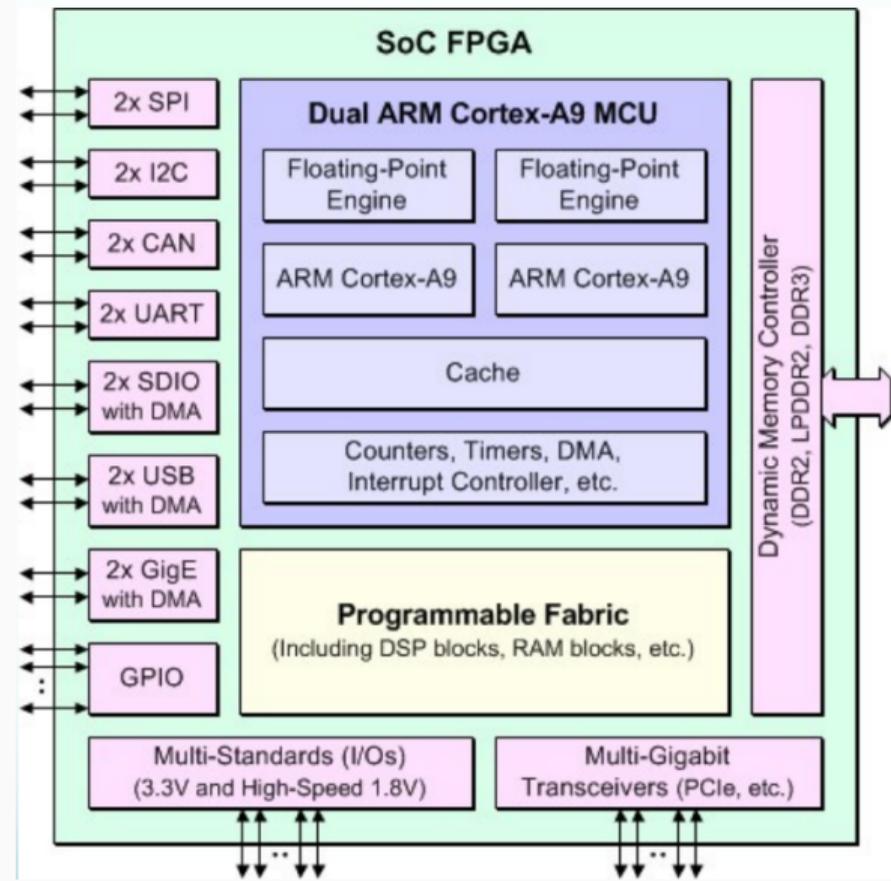
- Ядро микроконтроллера – ЦПУ
- Программа, исполняемая ЦПУ, создается с использованием языка высокого уровня, как правило – C/C++. Результатом компиляции программы является набор инструкций процессора.
- Инструкции исполняются последовательно

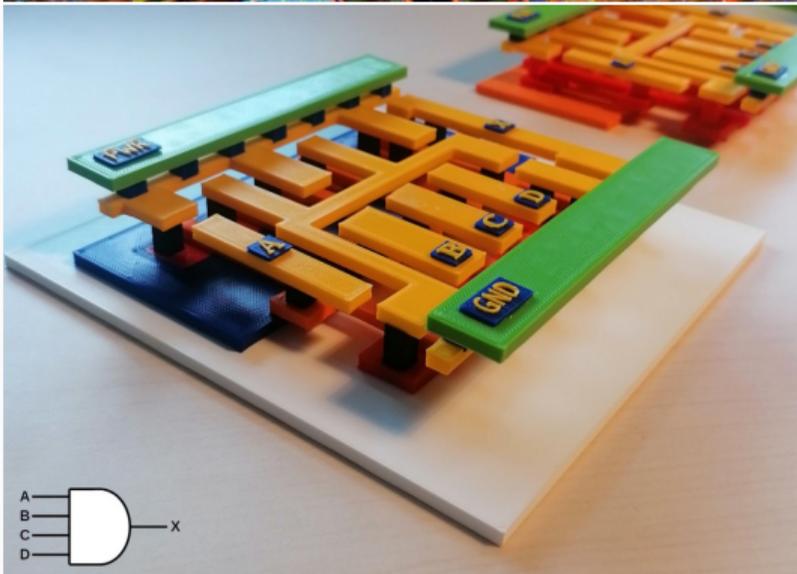
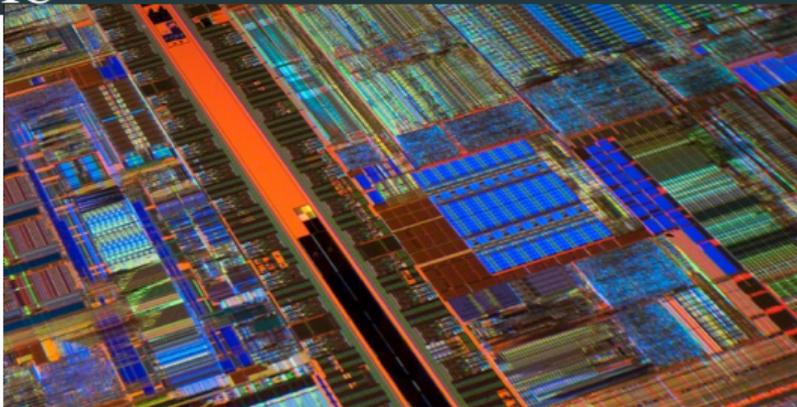




- Состоит из конфигурируемых логических блоков (CLB) и матрицы мажсоединений (interconnect array).
- Путем конфигурации можно создать произвольную логическую схему.
- Результатом синтезирования (synthesis, аналог компиляции в обычном программировании) является netlist - схема из логических вентиляей.

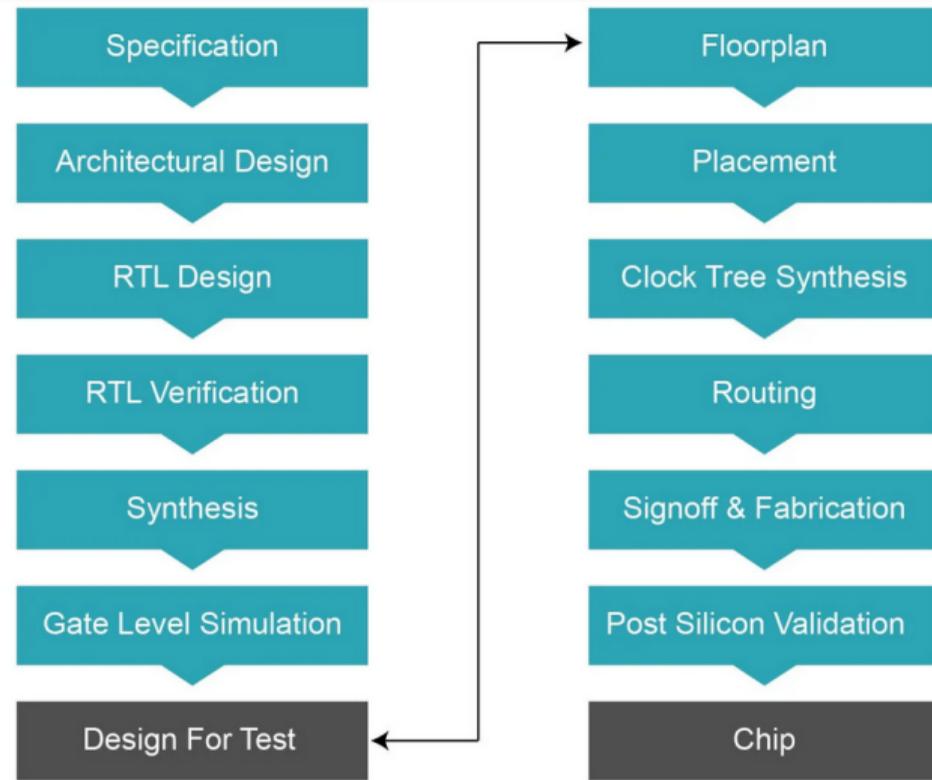
CPU+FPGA=SoC(SystemOnChip)





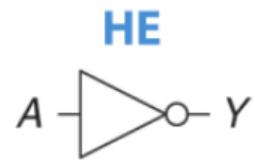
- Application Specific Integrated Circuit (ASIC) = интегральная схема специального назначения = сверхбольшая интегральная схема (СБИС).
- Gate-array design - дизайн на основе базовых матричных кристаллов. Схема конструируется из предоставляемых заводом логических блоков, заказчик разводит соединение между ними.
- Full-custom design - схема строится на основе транзисторов и элементов аналоговой электроники.
- Standard cells - построение дизайна на основе "библиотечных" компонентов для логических вентилей, триггеров и прочих элементов.

ASIC: design flow



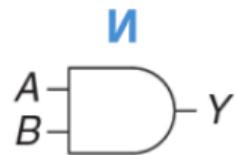
MCU vs FPGA vs ASIC

	MCU	FPGA	ASIC
Cost	low	moderate	high
Speed	low	high	high+
Power	low	moderate	low
Flexibility	low	high	none
Development time	low	moderate	high
Entry threshold	low	moderate	high
Programming language	c/c++	Verilog/VHDL	Special EDA, Verilog, VHDL



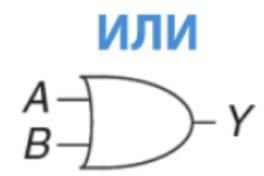
$$Y = \bar{A}$$

A	Y
0	1
1	0



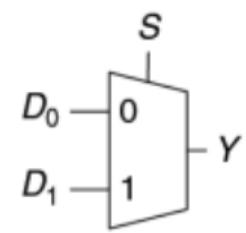
$$Y = AB$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



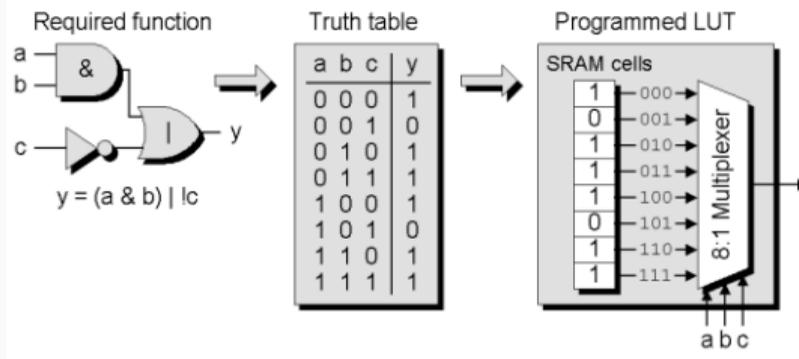
$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

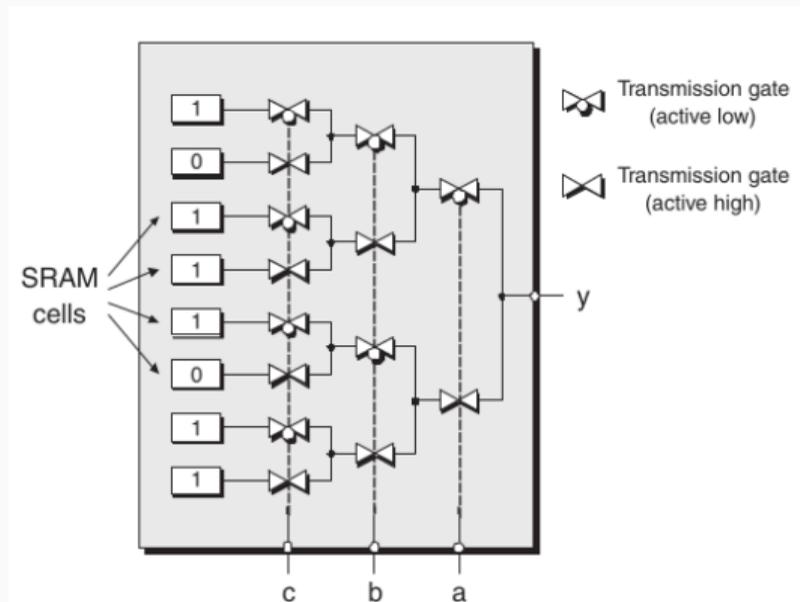


S	D ₁	D ₀	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

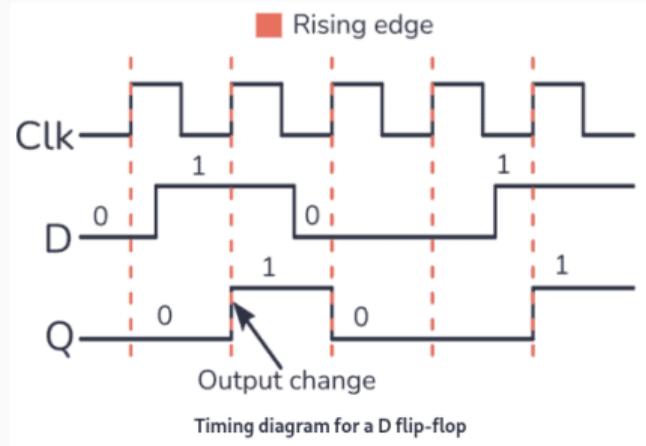
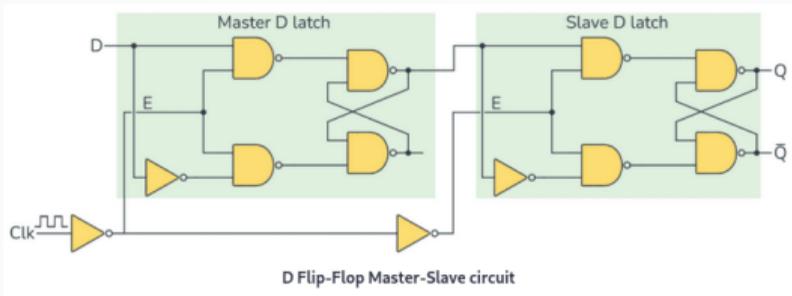
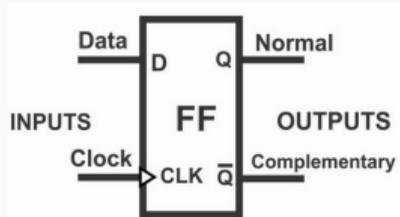
Look-Up-Table (LUT)



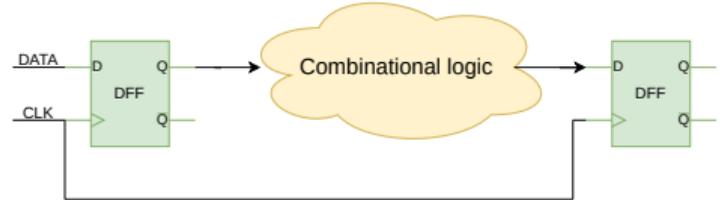
- Look-Up-Table = Таблица поиска / справочная таблица / таблица подстановок.
- Позволяет задать произвольное логическое выражение.
- Может служить элементом хранения данных.



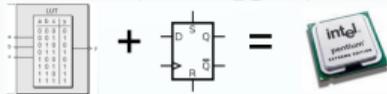
Synchronous(sequential) logic



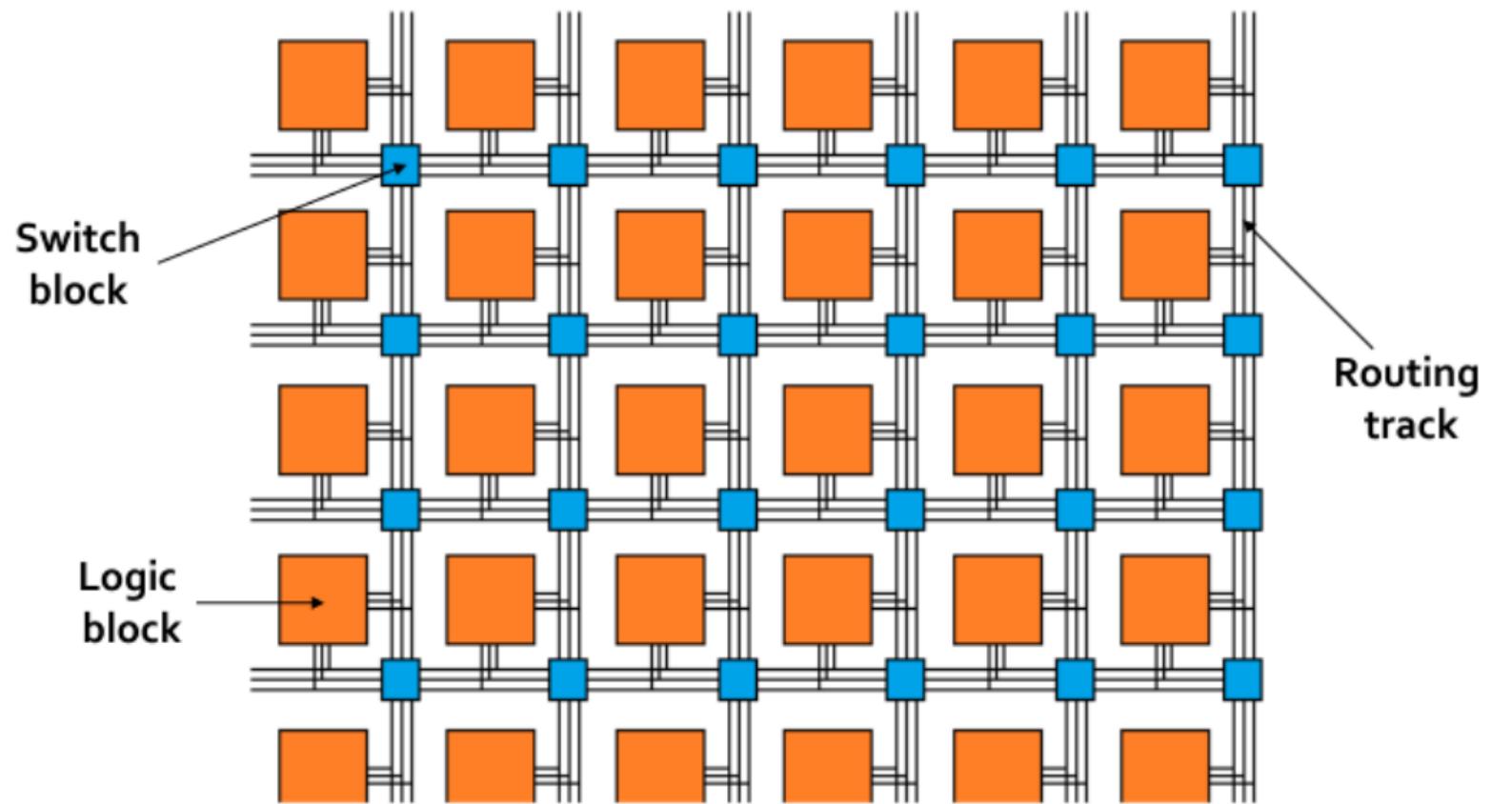
DFF + Combinatorial logic = Sequential logic



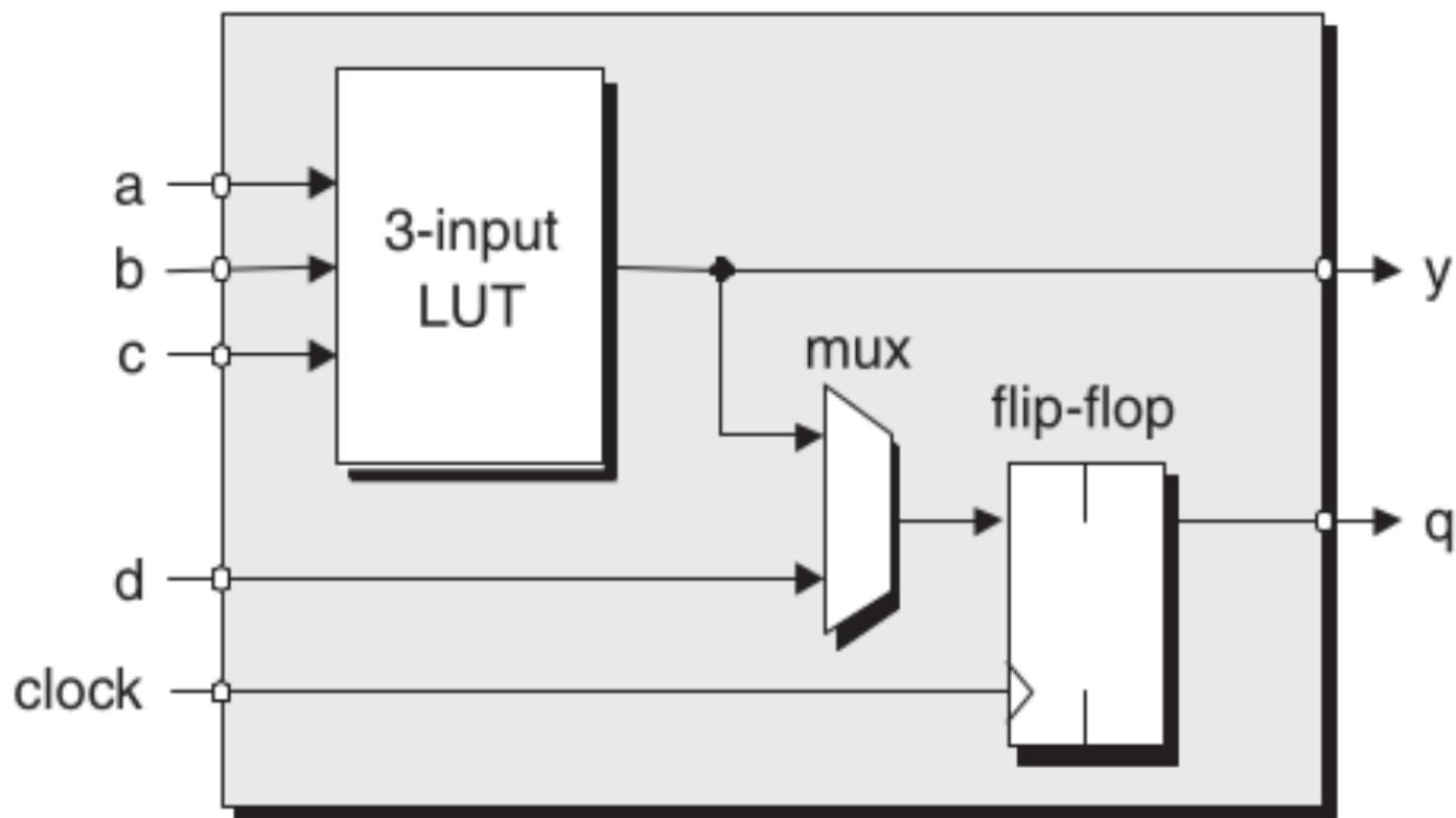
С помощью таблиц поиска и D-триггеров можно создать любую цифровую схему.



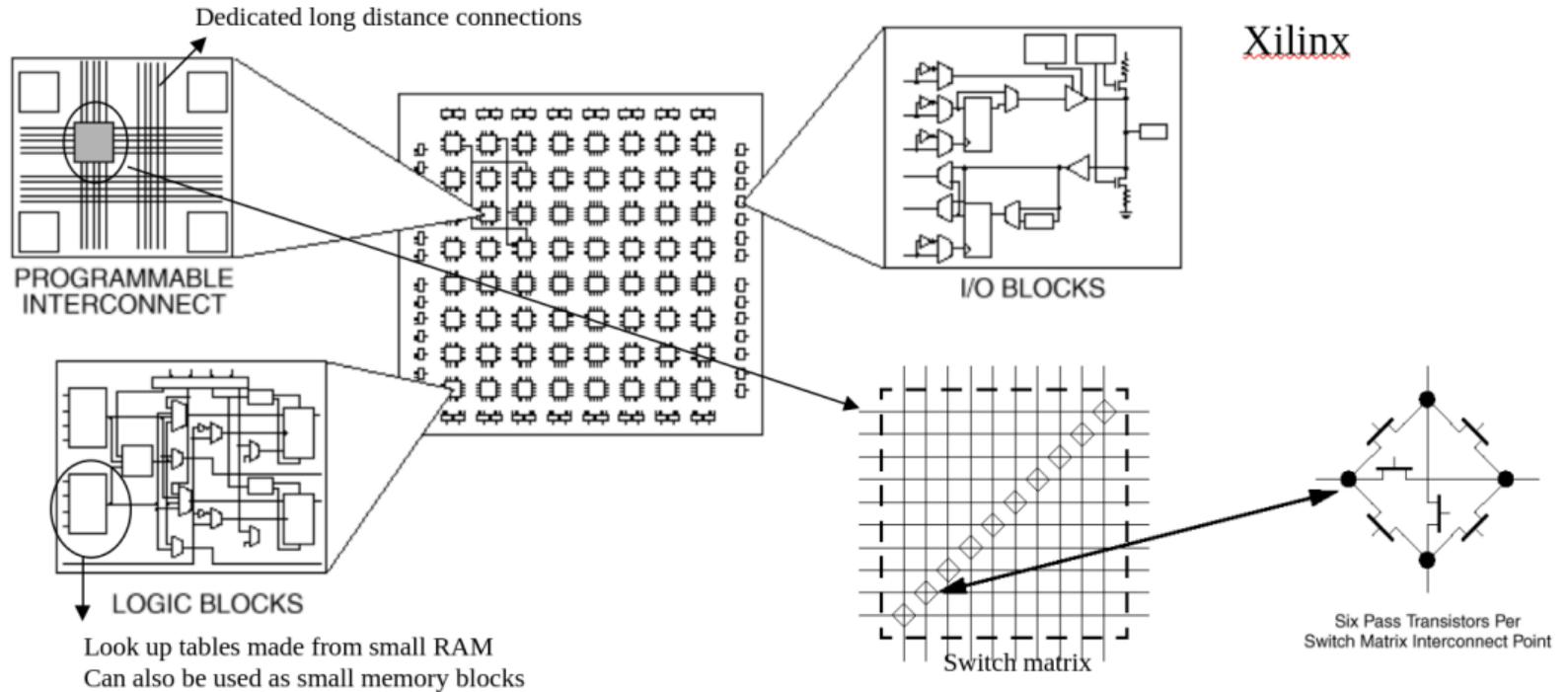
FPGA architecture



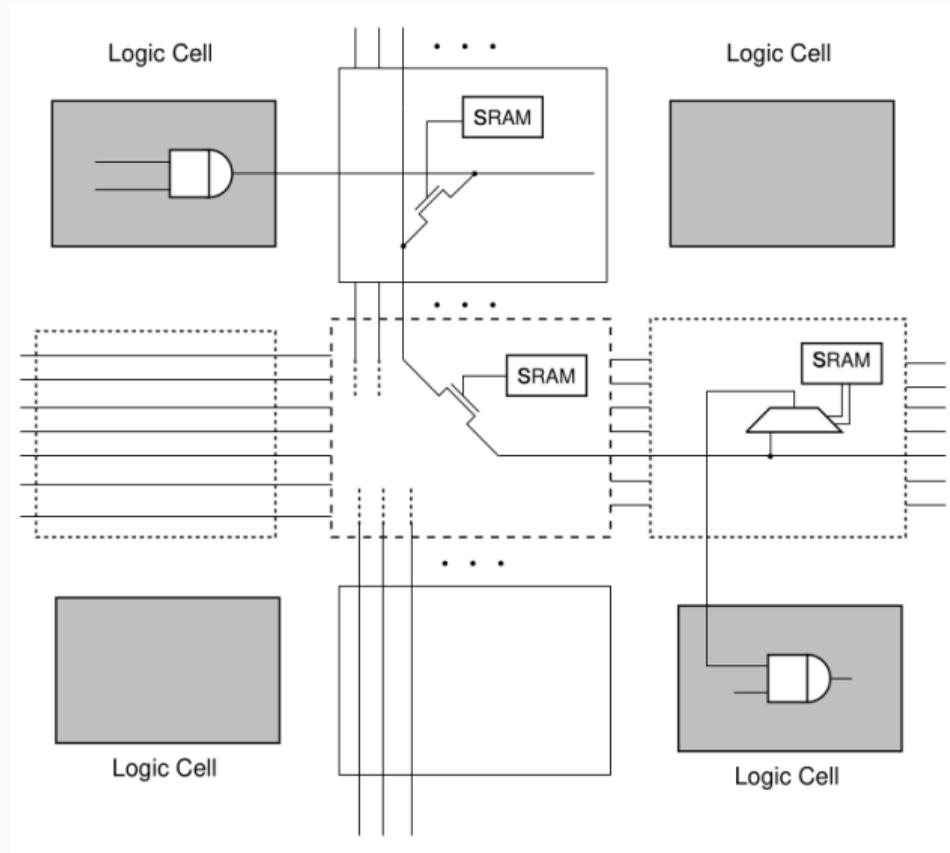
Programmable logic block (PLB)



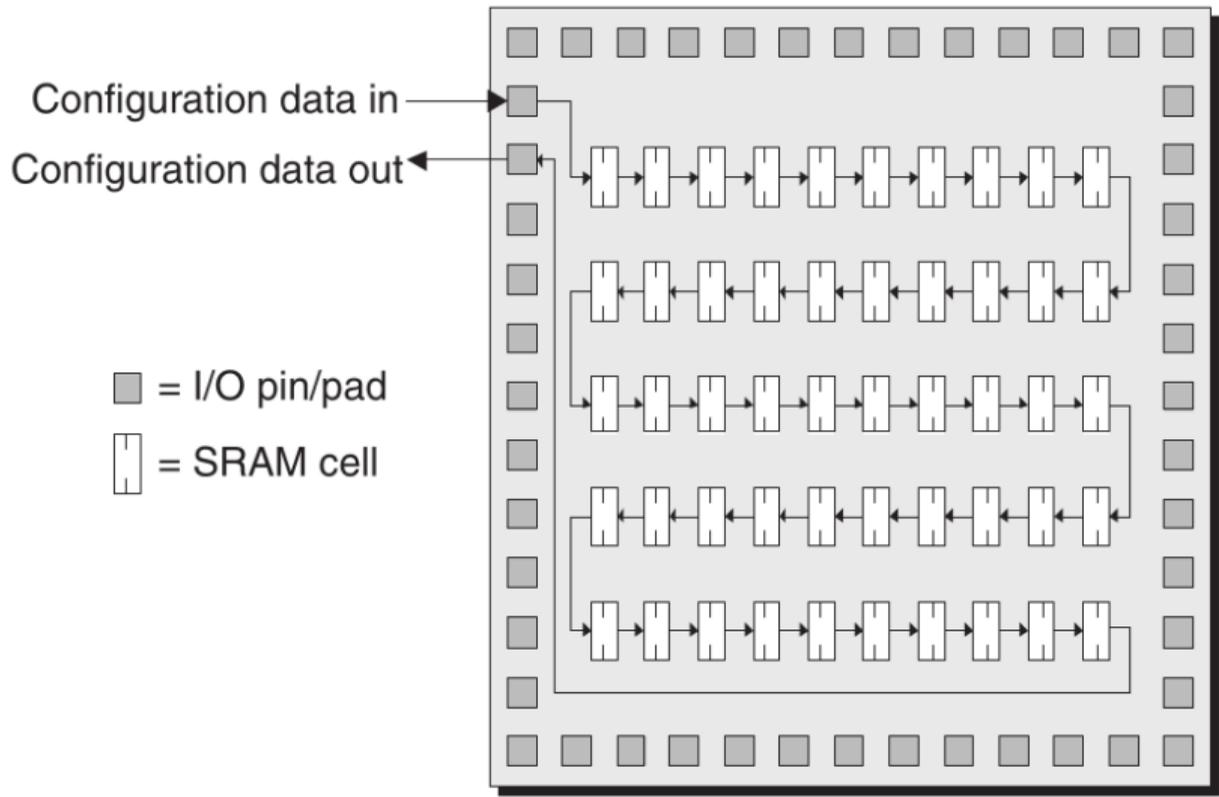
FPGA: Routing



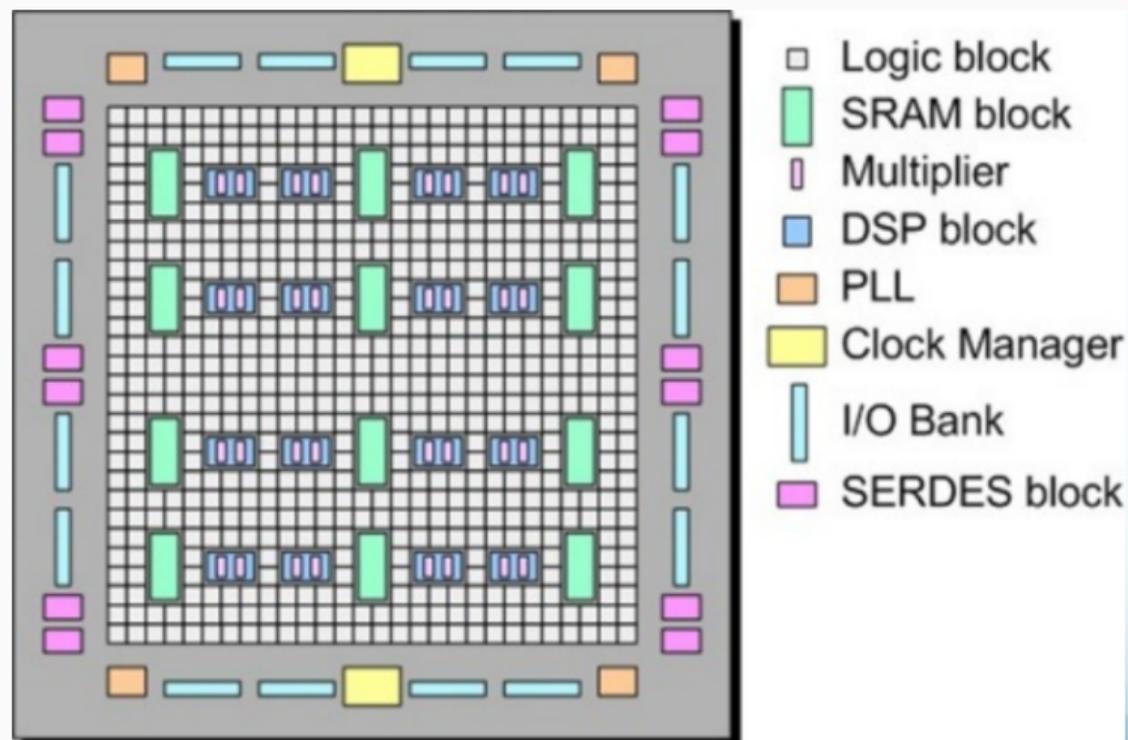
FPGA: Routing configuration



FPGA: Configuration



Other components inside FPGA

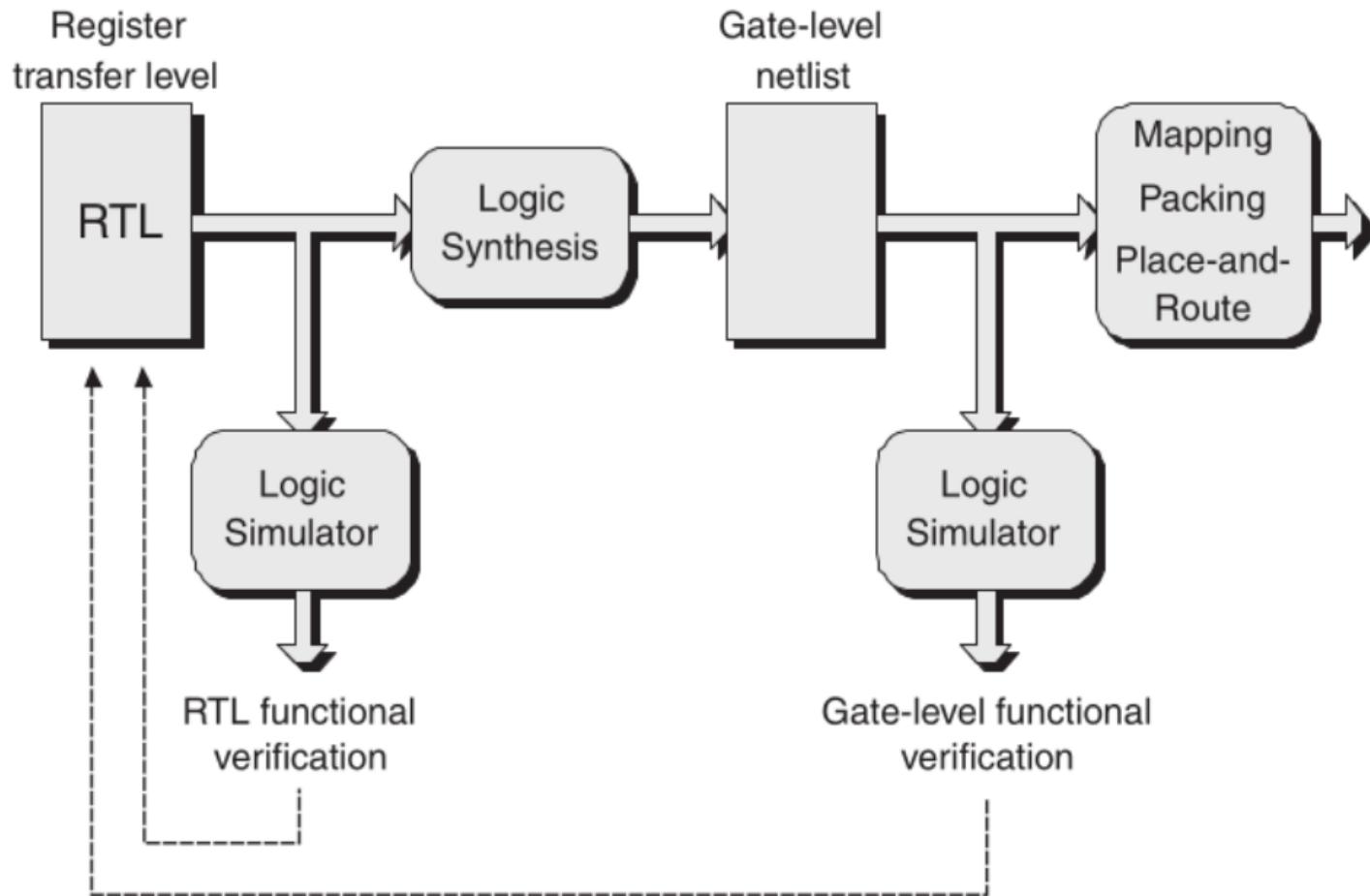


Most significant vendors

- Xilinx
 - First company to produce FPGAs in 1985
 - About 45-50% market share, today
 - SRAM based CMOS devices
- Intel FPGA (formerly Altera)
 - About 40-45% market share
 - SRAM based CMOS devices
- Microsemi (Actel)
 - Anti-fuse FPGAs
 - Flash based FPGAs
 - Mixed Signal
- Lattice Semiconductor
 - SRAM based with integrated Flash PROM
 - low power



FPGA: design flow



Hardware description languages (HDL)

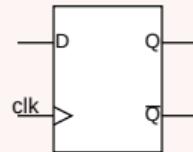
- Hardware description language – язык описания аппаратуры.
- Используются для описания логических схем.
- В отличие от языков программирования, которые транслируются в машинный код - инструкции последовательно выполняемые ЦПУ, HDL транслируется в схему из логических вентилей и D-триггеров.
- Основные HDL:
 - **VHDL** – адаптированный язык программирования ADA, был разработан в 1983г. по заказу министерства обороны США
 - **Verilog/SystemVerilog** – разработан в 1984г. в компании Gateway Design.
 - **AHDL** – язык описания аппаратуры, разработанный фирмой Altera, и использовавшийся только для их устройств. Современными программами разработки более не поддерживается, но все еще используется, например, на КМД-3.

```
library ieee;  
use ieee.std_logic_1164.all;
```

Подключенные
библиотеки

```
entity dff is  
port(  
  clk: in std_logic;  
  d:   in std_logic;  
  q:   out std_logic);  
end dff;
```

"Цоколевка"
компонента, описание
портов ввода/вывода



```
architecture behavioral of dff is  
begin  
  process (clk) begin  
    if rising_edge(clk) then  
      q <= d;  
    end if;  
  end process;  
end behavioral;
```

Архитектура - внутренняя
схема компонента

Последовательно
"исполняемый" блок,
синтезирование приводит
к созданию
комбинаторной схемы с
D-триггерами на выходе

VHDL trivial structural example

```
architecture structural of two_dff is
```

```
  signal q1 : std_logic := '0';
```

```
begin
```

```
  dff_1 : entity work.dff
```

```
  port map(
```

```
    clk => clk,
```

```
    d   => din,
```

```
    q   => q1
```

```
  );
```

```
  dff_2 : entity work.dff
```

```
  port map(
```

```
    clk => clk,
```

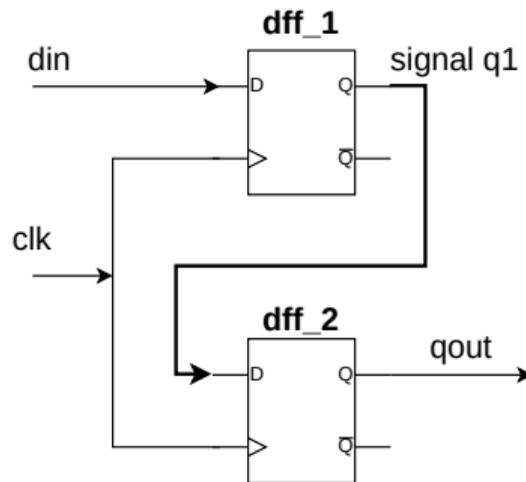
```
    d   => q1,
```

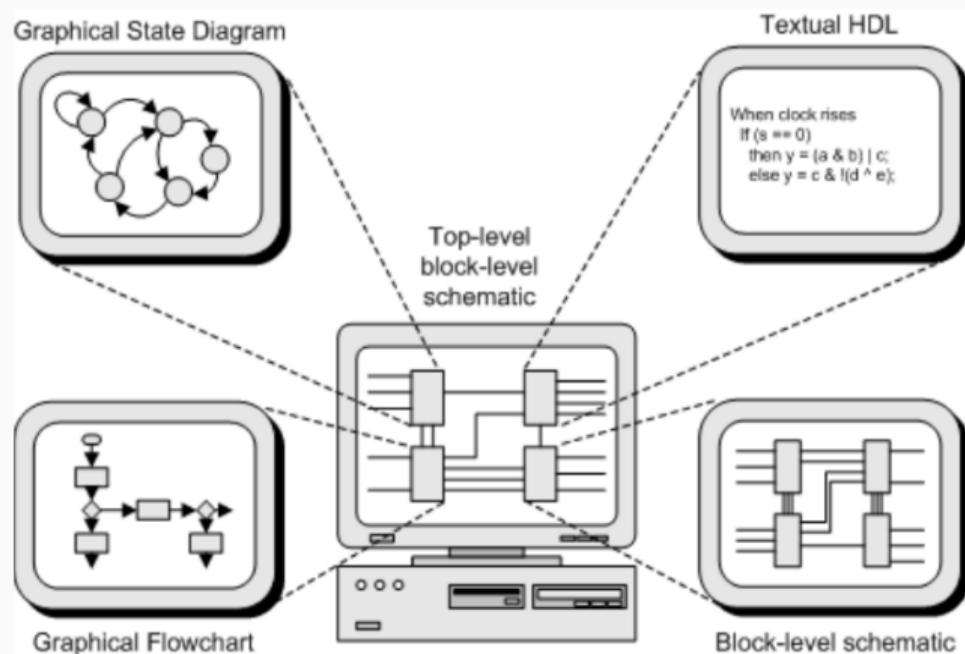
```
    q   => qout
```

```
  );
```

```
end structural;
```

Промежуточный "сигнал", может служить либо соединительным "проводом" либо регистром





- В области традиционного программирования физики давно "импортозаместили" программистов. В иностранных группах идет процесс перенимания задач проектирования цифровой логики физиками у сертифицированных инженеров.
- Несмотря на меньшую интеллектуальную сложность в сравнении с вычислением матричных элементов, присутствует критический дефицит кадров как в научной сфере, так и в промышленности.