

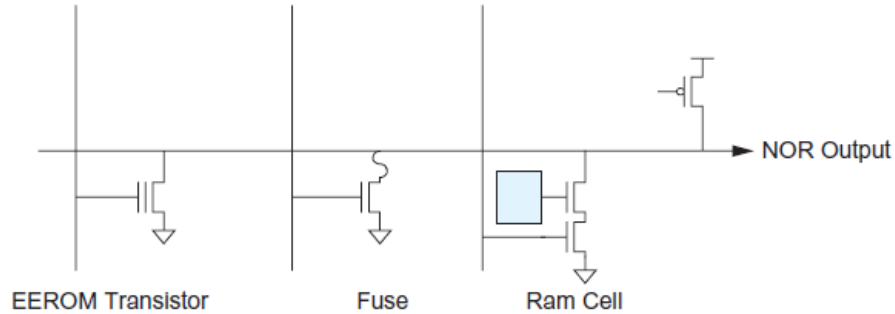
РВС  
Лекция 4-10

ТНК  
2019

# Специальные чипы:

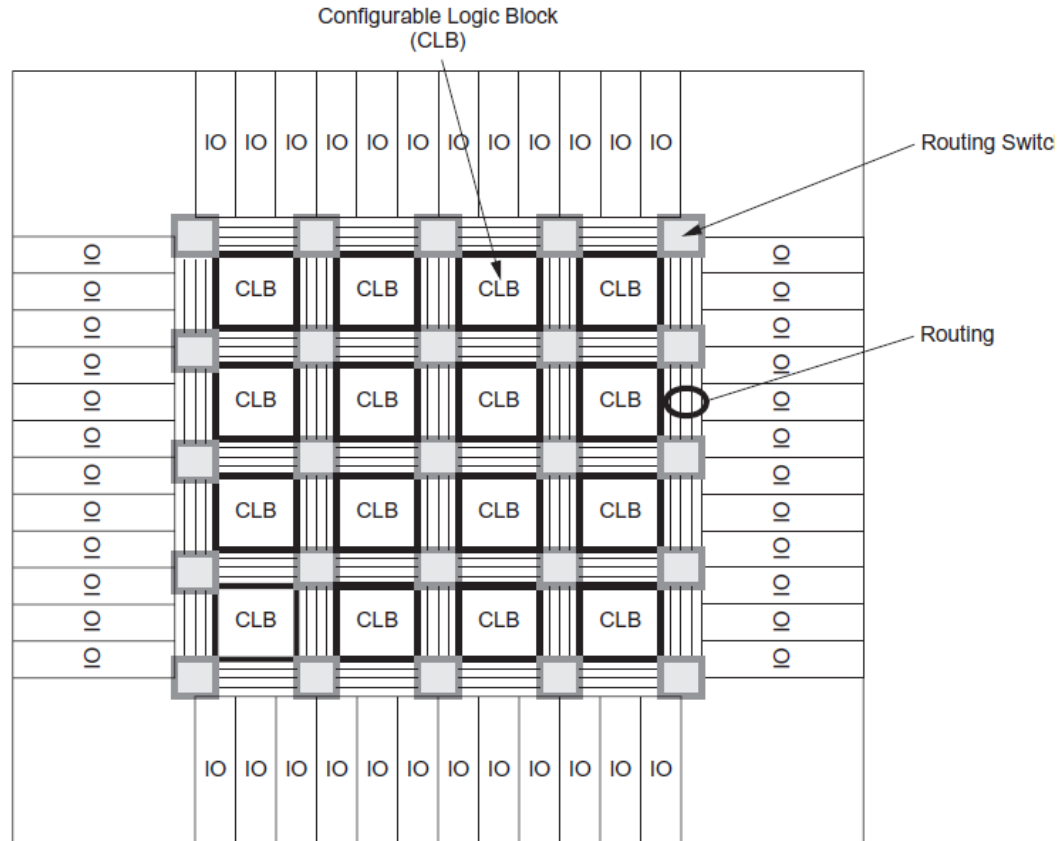
- Чипы с программируемыми логическими массивами
- Микросхемы с программируемыми соединениями
- Микросхемы с перепрограммируемой логикой и межсоединениями

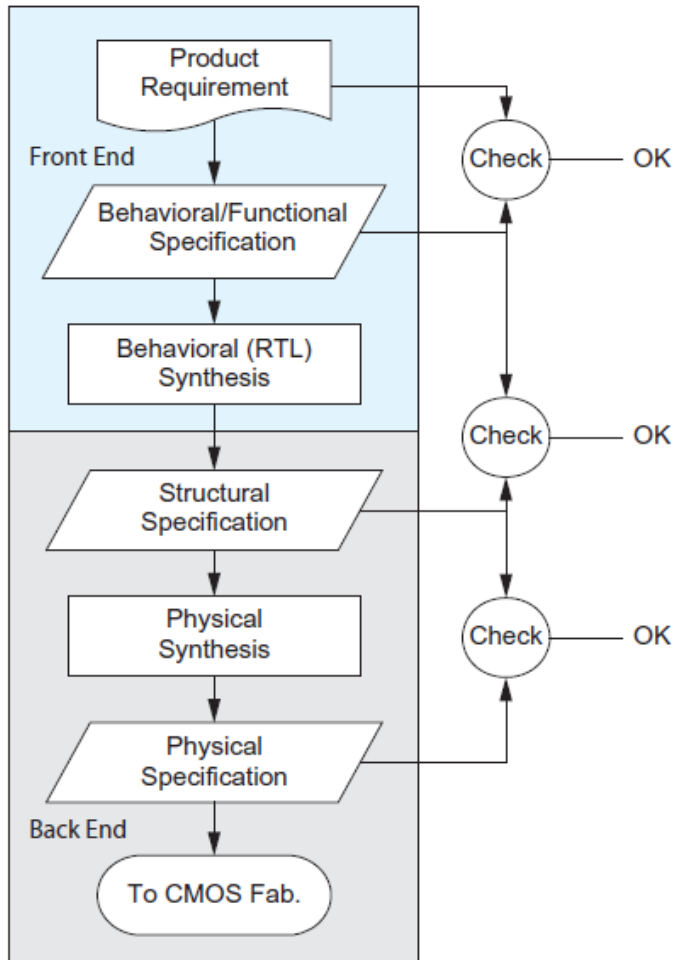
# Структура PLA NOR



PLA состоит из плоскости AND и плоскости OR для вычисления любой функции, выраженной в виде суммы произведений. Каждый транзистор в плоскости И и ИЛИ должен быть запрограммирован на присутствие или нет. Это может быть достигнуто путем полного заполнения плоскости AND и OR структурой NOR в каждом местоположении PLA. Каждый узел запрограммирован с помощью транзистора с плавающим затвором, плавкой вставки или транзистора с управлением от ОЗУ Первые две версии были способом программирования этих типов устройств при низкой плотности устройства.

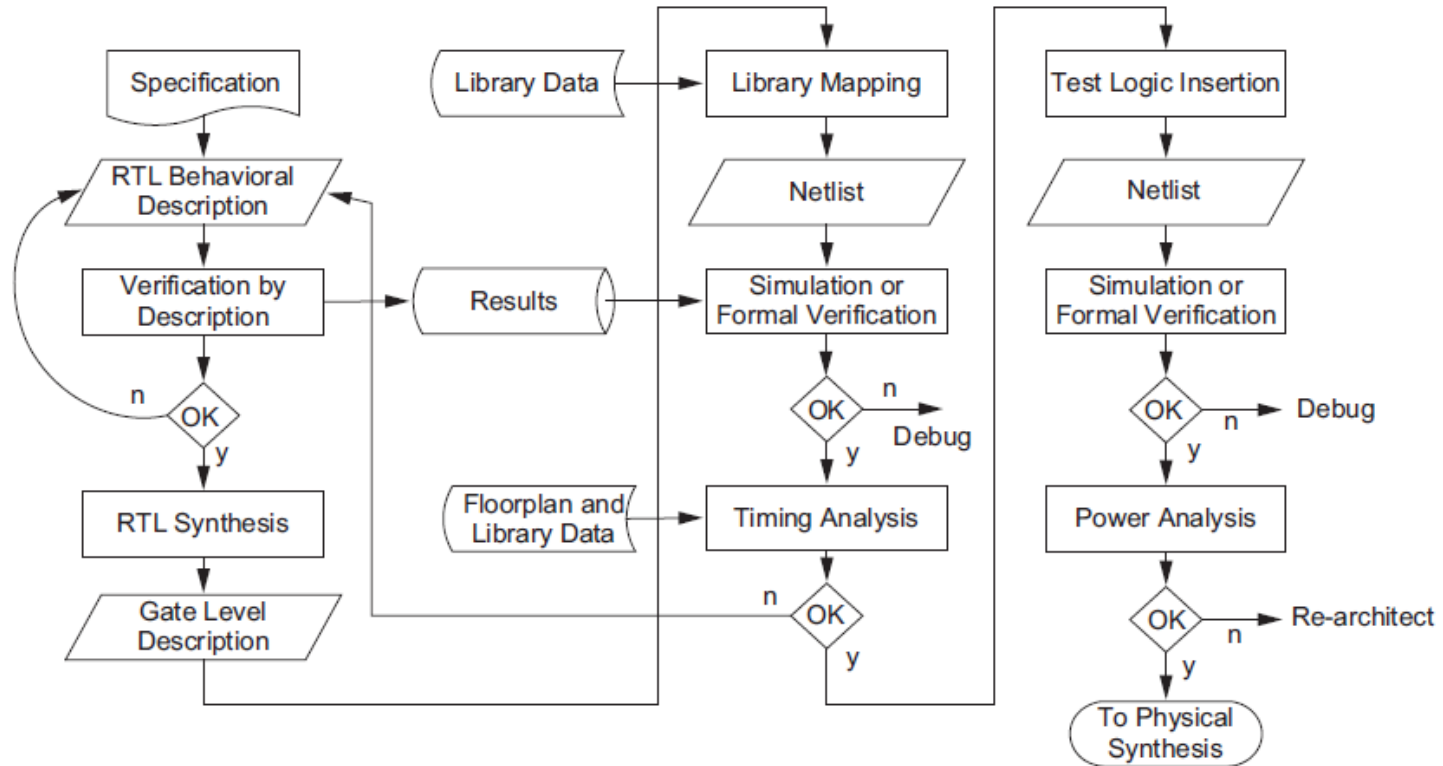
# Микросхема состоит из массива конфигурируемых логических блоков (CLB)





# Алгоритм разработки чипа ВС

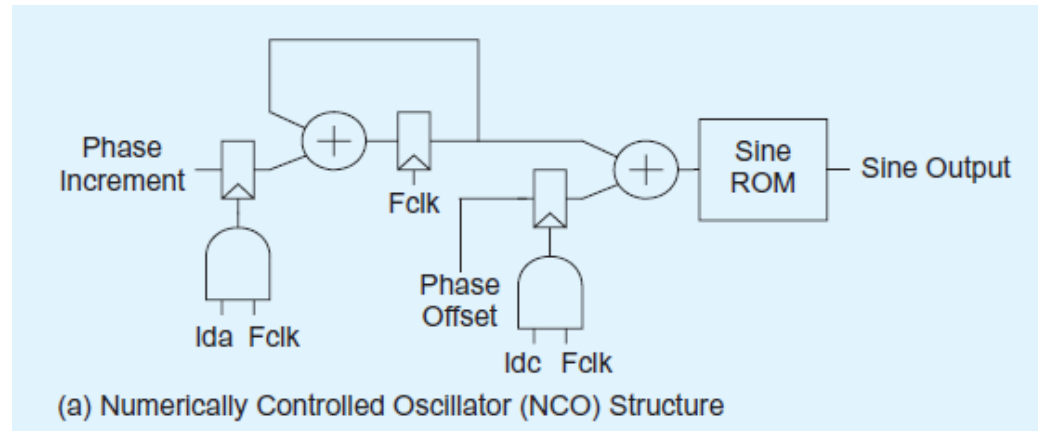
# САПР: Synopsys, Cadence Design Systems, Mentor Graphics и Synplicity



# Проектирование и проверка логики

- Проектирование начинается со спецификации, которая может быть текстовым описанием или описанием на языке спецификации системы. Разработчик (и) преобразуют это в описание поведения RTL в HDL, таком как Verilog или VHDL.

# Пример



NCO состоит из зарегистрированного сумматора, который увеличивается каждый такт на регистр приращения фазы. Это реализует счетчик фаз, который используется для пошагового просмотра справочной таблицы ПЗУ, которая обеспечивает преобразование фазы в амплитуду. Смещение фазы может быть добавлено к фазовому инкременту для выполнения фазовой модуляции. Благодаря этой структуре мы можем генерировать цифровую синусоидальную волну.



# Поведенческий Verilog для 8- битной реализации ранее созданного NCO

```
module nco #(parameter size = 8,  
             counter_size = 16,  
             table_size = 64)  
  (input  
   input [counter_size-1:0] initial_phase, phase_increment,  
   output [size-1:0] q);  
  
  reg [counter_size-1:0] phase;  
  
  wire [size-3:0] phase_part, inverted_adr, ROM_adr;  
  wire [size-2:0] ROM_data;  
  wire [size-1:0] wave_out;  
  
  // numerically controlled oscillator  
  // note that some constants are hardwired in the code below  
  
  // phase counter  
  always @(posedge fclock)  
    if (reset) phase <= initial_phase;  
    else phase <= phase + phase_increment;  
  
  // add offset and determine ROM address  
  assign phase_part = phase[counter_size-3:counter_size-8];  
  assign inverted_adr = 7'3f - phase_part;  
  assign ROM_adr = phase[counter_size-2] ? inverted_adr : phase_part;  
  
  // look up data in ROM and negate if appropriate  
  quarter_wave sine_table(ROM_adr, ROM_data);  
  assign wave_out = phase[counter_size-1] ? ~ROM_data : ROM_data;  
  assign q = wave_out + 8'h80 + phase[counter_size-1];  
endmodule
```

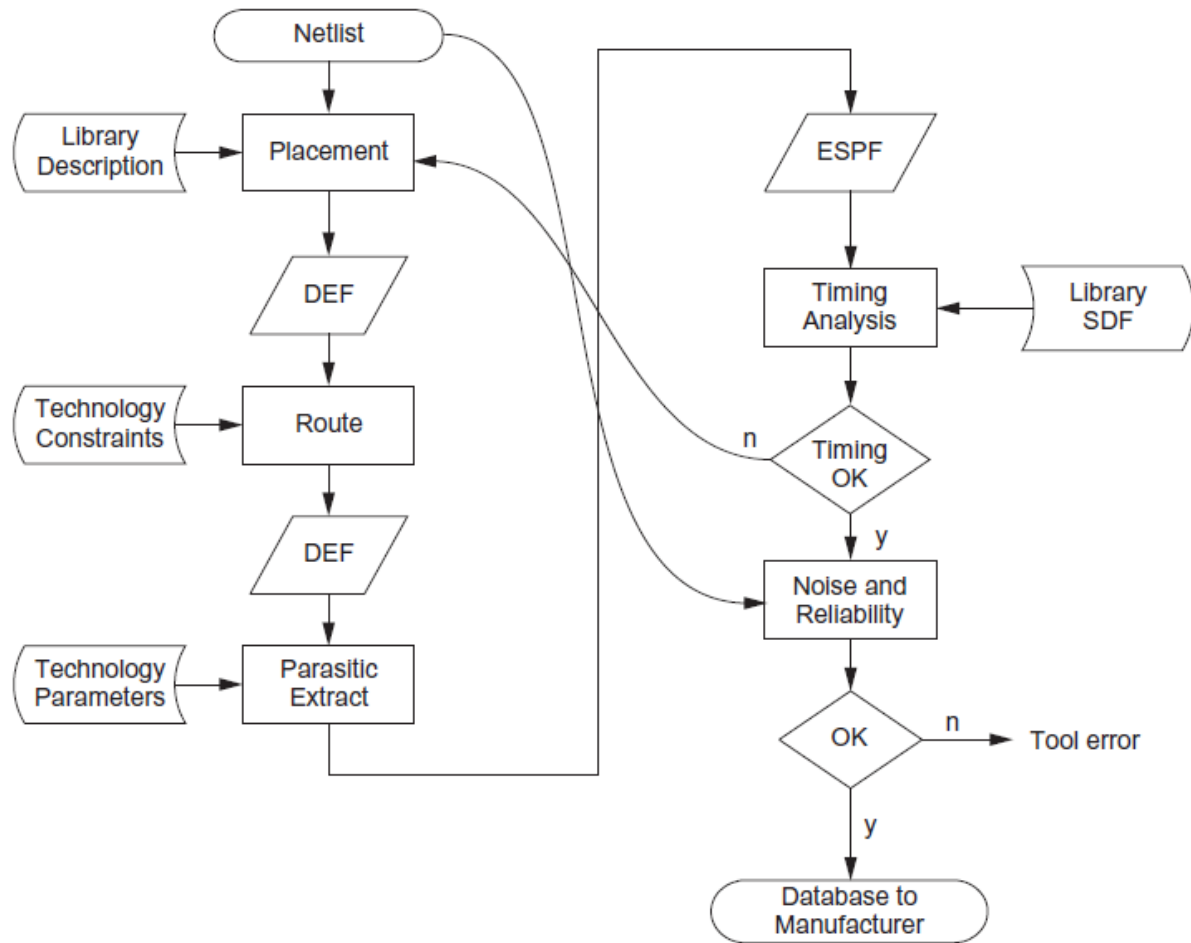
# RTL-синтез

- Следующим шагом является обобщение описания поведения. Это включает в себя преобразование RTL в вентиляные схемы и регистры, оптимизацию логики для улучшения скорости и площади и сопоставление вентиля со стандартной библиотекой ячеек.
- Другими шагами, выполняемыми на этом этапе, являются декомпозиция конечного автомата, оптимизация канала данных и оптимизация энергопотребления.
- Типичные продукты включают компилятор дизайна от Synopsys, компилятор RTL от Cadence и Synplify Pro от Synplicity.

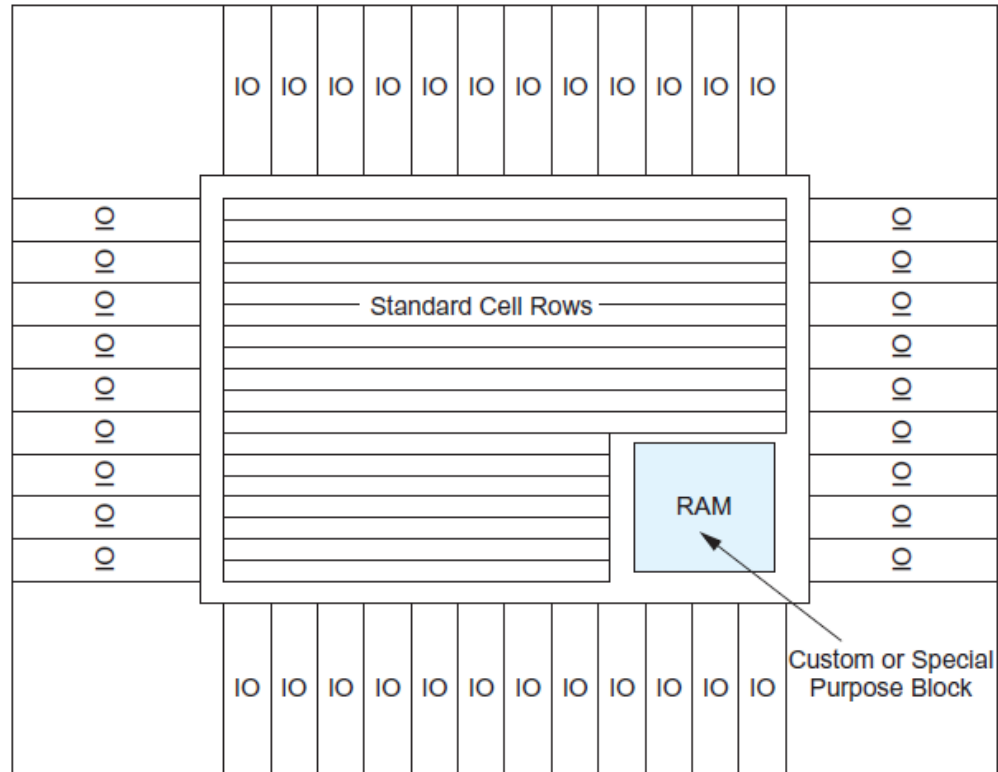
## Verilog. Схемотехника NCO

```
module nco_struct_mapped(input          fclock, reset,
                        input  [15:0] initial_phase, phase_increment,
                        output [7:0]  q);
    .
    .
    BUFX4 i_506(.A(n_355), .Y(q[7]));
    .
    MX2X1 i_00(.S0(reset), .B(initial_phase[15]), .A(nbus_1[15]),
              .Y(phase_0[15]));
    NAND2BX1 i_8(.AN(n_102), .B(n_101), .Y(n_104));
    XOR2X1 i_6(.A(phase[15]), .B(ROM_Table[6] ), .Y(n_103));
    .
    .
    DFFHQX1 phase_reg_0(.D(phase_0[15]), .CK(fclock), .Q(phase[15]));
    .
    .
    .
endmodule
```

Алгоритм  
физического  
дизайна  
(размещение и  
трассировка)



# Стандартная схема размещения в чипе



# Затраты на изготовление прототипа

Эти затраты ( $P_{total}$ ) являются фиксированными затратами на получение первых микросхем от поставщика. Они включают

- Стоимость маски
- Расходы на тестовое оборудование
- Пакетная оснастка

Стоимость фотомаски зависит от количества шагов, используемых в процессе, и точности, требуемой для каждого шага. Маски на слоях металлизации могут быть дешевле, чем на нижних слоях, потому что шаг не такой плотный.

показывает, как стоимость маски экспоненциально увеличивается [Donovan02, LaPedus07]. Стоимость полного набора масок в 45 нм процессе составляет около 5 миллионов долларов.

Испытательное приспособление состоит из узла зонда с печатной монтажной платой, предназначенного для проверки отдельных штампов на уровне пластин, и интерфейса с тестером. Расходы варьируются от 1000 до 50 000 долларов, в зависимости от сложности интерфейса электроники.

