

# Открытый маршрут проектирования ASIC и статический временной анализ

Коршунов Андрей Владимирович  
доцент кафедры ПКИМС МИЭТ

# План занятия

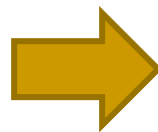
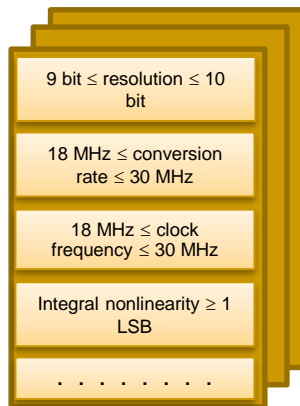
- 1) Режим занятия: лекция + демонстрация запуска программ
- 2) Повторение и самостоятельная работа после занятия в свободном режиме
  - Принципы проектирования
  - Логическая симуляция
  - Логический синтез
  - Временной анализ и ограничения
  - Библиотека
  - Реализация маршрута OpenLane

# Принципы проектирования

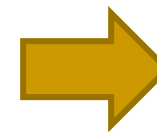
# Процесс проектирования ИС

- Преобразование спецификации в набор данных для изготовления

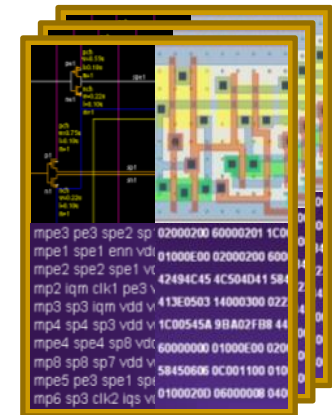
Спецификация



Процесс проектирования

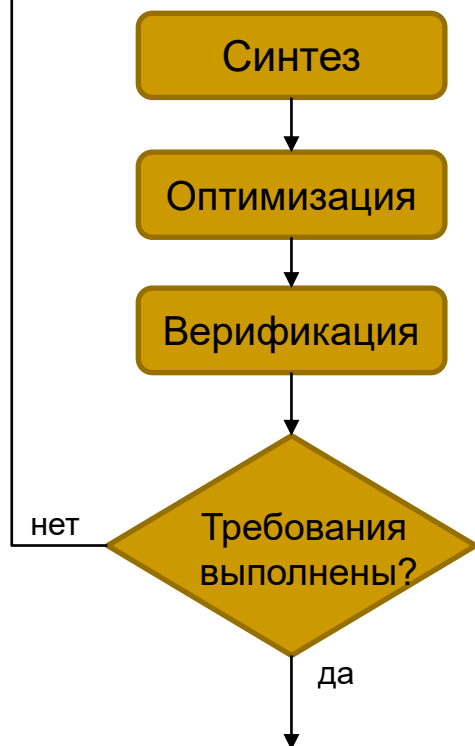


Проект ИС



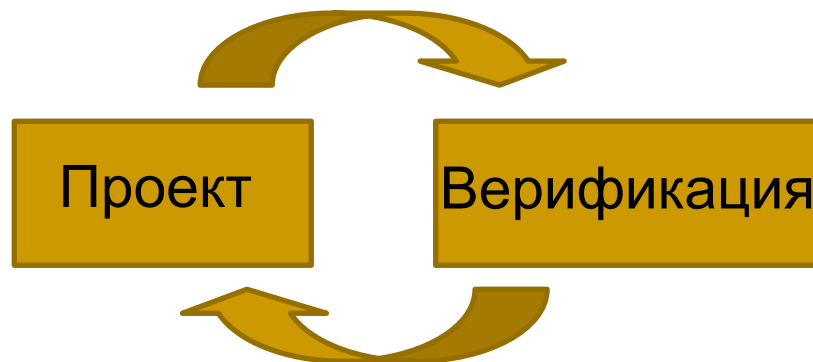
# Проблемы, решаемые на одном уровне проектирования

Менее детализированное



Более детализированное представление

Процес проектирования предполагает многократное прохождение этапа синтеза и последующей верификации результата



Синтез:

- Ручной
- Автоматизированный
- Автоматический

# Методы верификации

- Формальная верификация

- конструкция на разных уровнях проектирования имеет логически эквивалентную функцию

- Статический временной анализ

- Задержка пути рассчитывается как сумма задержек элементов без моделирования логики

- Симуляция

- Воспроизводится поведение объекта во времени

- Эмуляция

- Использование устройства, которое работает как проверяемая система, подача тестовых векторов, проверка выходного сигнала

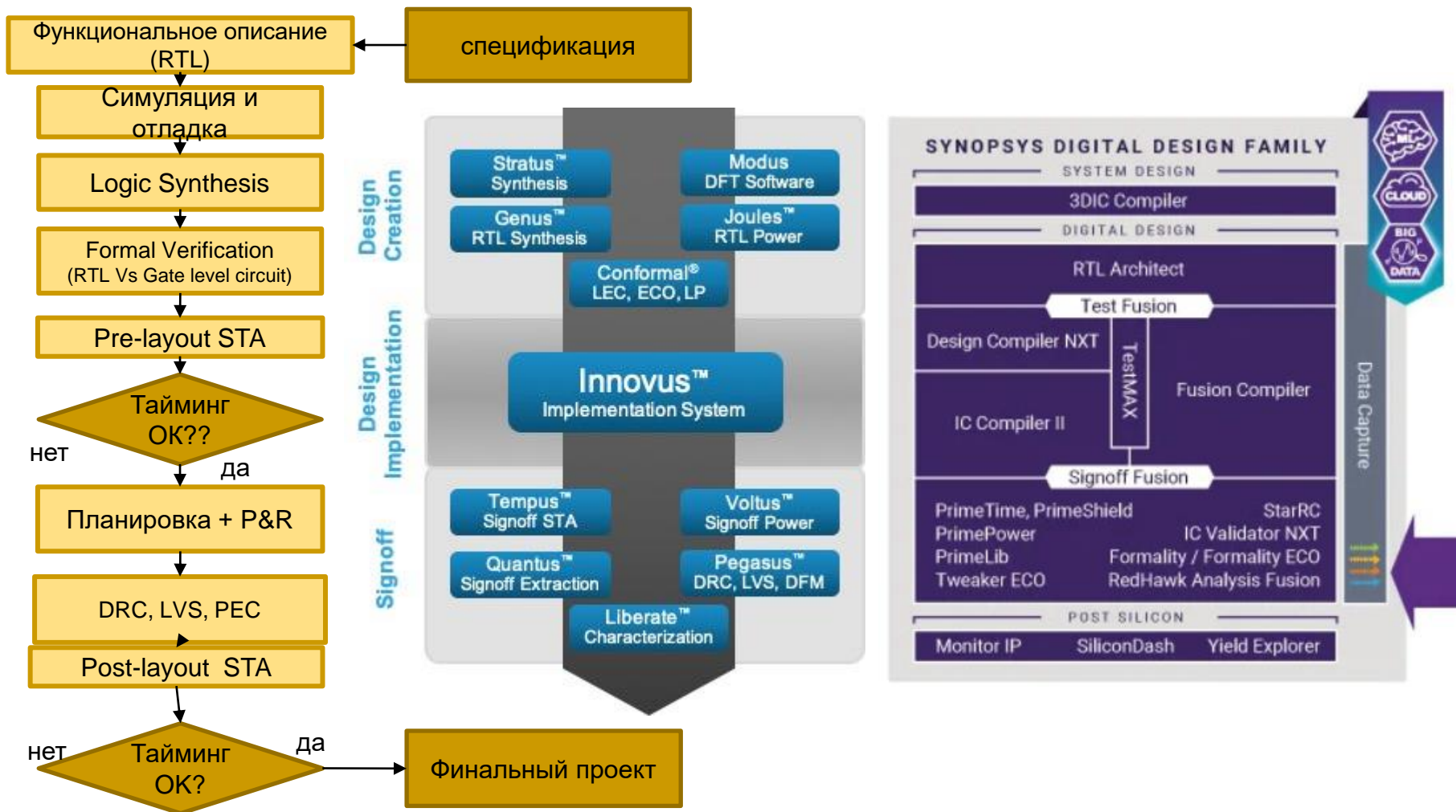
- Прототипирование

- Создание аппаратной реализации (обычно ПЛИС) проекта и его тестирование

# Обобщенный маршрут



# Традиционный маршрут RTL2GDS



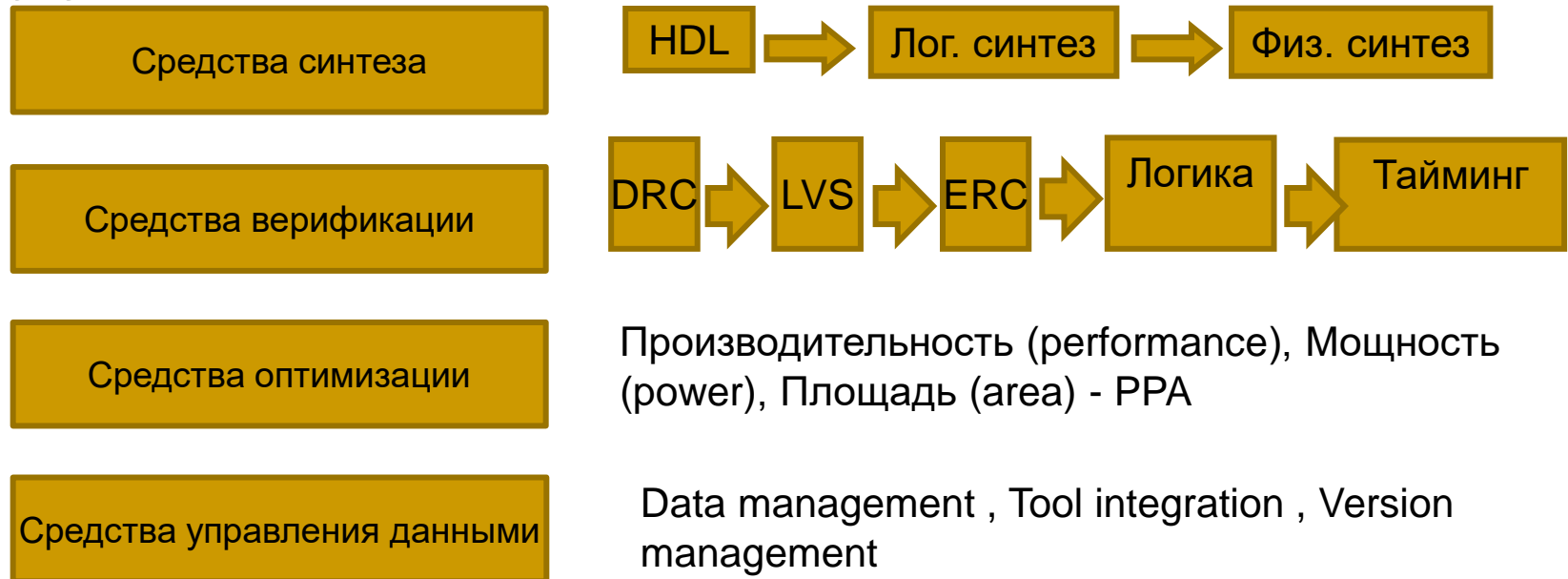


# Проблемы реального маршрута проетирования

- Новые виды верификации
  - Моделирование на уровне транзакций, использование утверждений (Transactions, assertions)
  - Статический анализ кода
- Анализ шумов, смешанные сигналы
- Разработка ограничений
- Учет трассировочных ресурсов
- Проектирование с учетом энергопотребления и целостности сигнала (SI)
- Сложная иерархическая планировка
- Проектирование для производства (DFT, DFM, DFY)

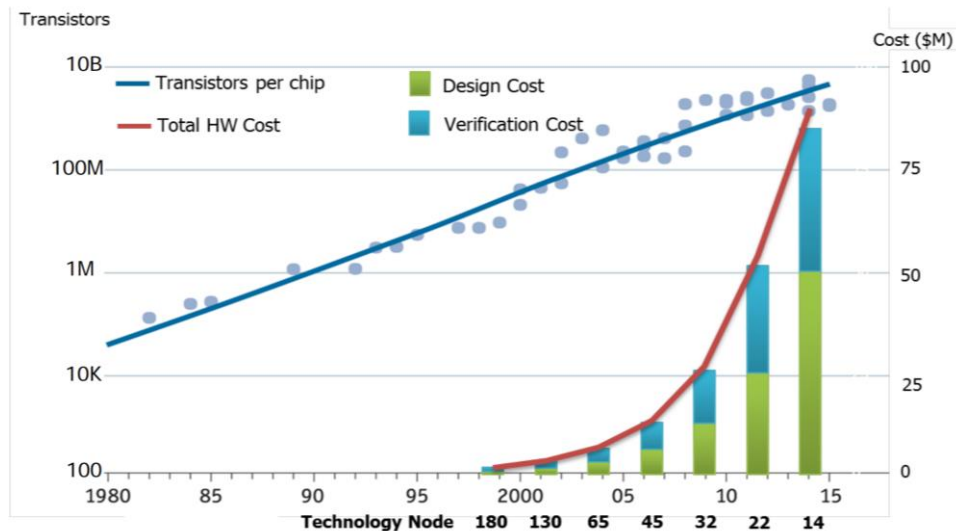
# Проектные операции и средства

- Операции при проектировании СБИС можно разделить еще по типам действий независимо от уровня представления схемы



# Открытый маршрут

# Кризис проектирования



- стоимость изготовления – находилась примерно на одном уровне, либо росла в разумных пределах
- но стоимость РАЗРАБОТКИ кремния находится вне контроля.
- стоимость САПР (100k\$/year) + разработчики

Проектирование ASIC на передовых техпроцессах:  
Огромные барьеры стоимости, экспертизы, риска

Сложно оценить SWaP\*, "PPA\*\*  
своих проектных идей

Коммерческие решения:

- сложное ПО с 1000 команд
- Ориентация коммерческих заказчиков
- пользователь-эксперт
- множество ручных операций
- Попытки получить наилучший результат
  - size-weight-and-power
  - power-performance-area

# Открытые маршруты

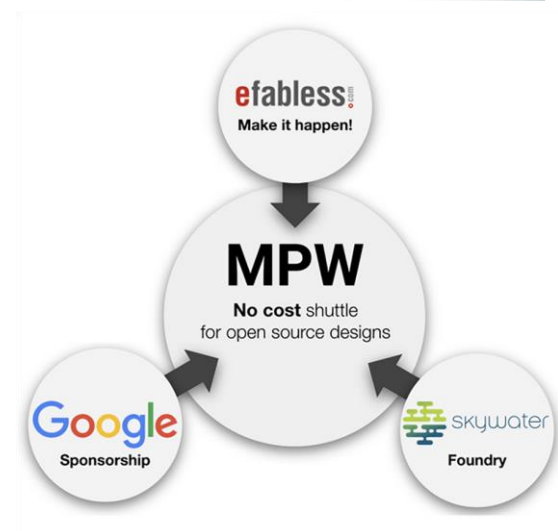
Инструмент	Поддержка	Временные рамки разработки	Текущая версия	Лицензия
qFlow	Р.Т. Эдвардс	2013 – н. вр.	1.4.98	GPL
OpenLANE	Сообщество разработчиков	2020 – н. вр.	02.00.18	Apache 2.0
Alliance/Coriolis	Сообщество разработчиков	1990 – н. вр.	28.10.2019	GPL, LGPL, Apache 2.0
SymbiFlow	Сообщество разработчиков	2018 – н. вр.	–	ISC

Камкин А.С., Смолов С.А., Чупилко М.М. Сравнение открытых маршрутов проектирования цифровой аппаратуры: qFlow, OpenLANE, Coriolis, SymbiFlow. Труды ИСП РАН, том 33, вып. 6, 2021 г., стр. 111-130

Несколько попыток собрать множество разнородных академических пакетов в единое решение, которое будет эффективно реализовывать:  
преобразование **RTL-модели** в топологию (**GDS II**)

В 2018 году инициатива DARPA по разработке новой экосистемы разработки электроники

В 2020 – OpenMPW от Google и Skywater для свободного выпуска чипов



Open-source PDK



Open-source EDA Tools



Free Fabrication

# OpenLane



Open-source PDK



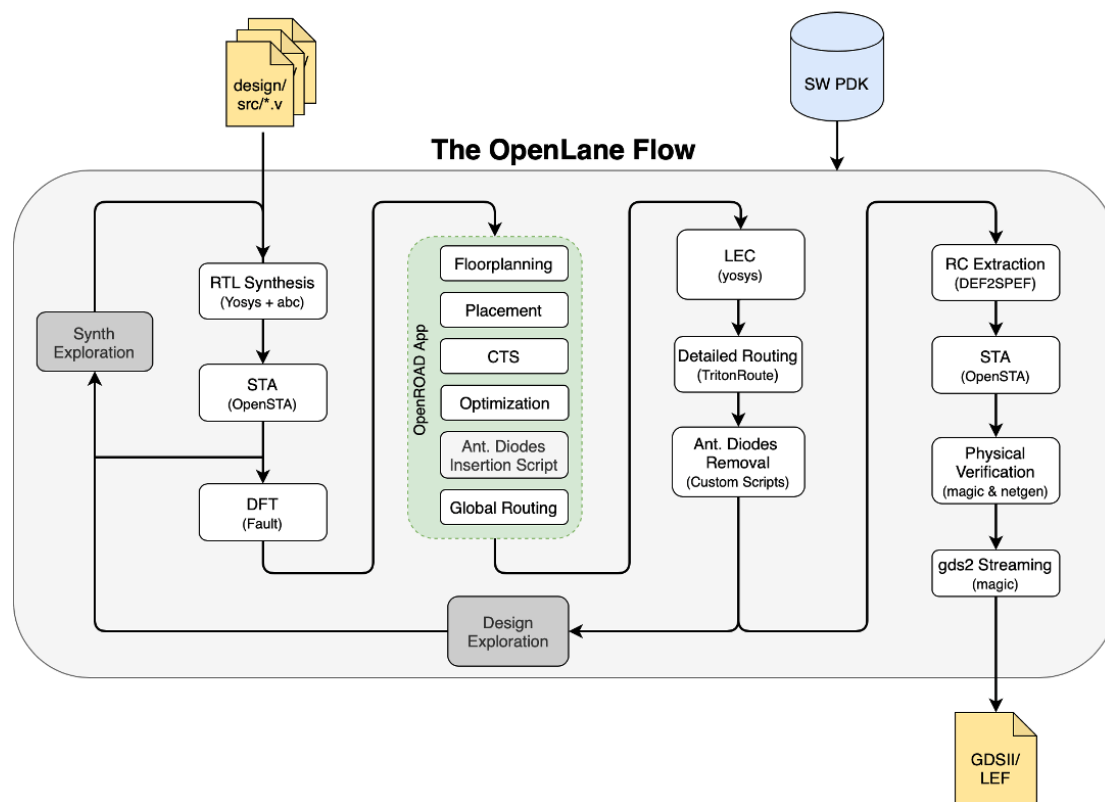
Open-source EDA Tools



Free Fabrication

Восемь запусков, 40 слотов в каждом шаттле, бесплатно для разработчиков полностью открытых проектов ИС и IP.

SkyWater 130 нм PDK с открытым исходным кодом



OpenLane - компилятор RTL в GDS, на основе OpenROAD, аналог GNU компилятор ПО, но для аппаратуры!!.

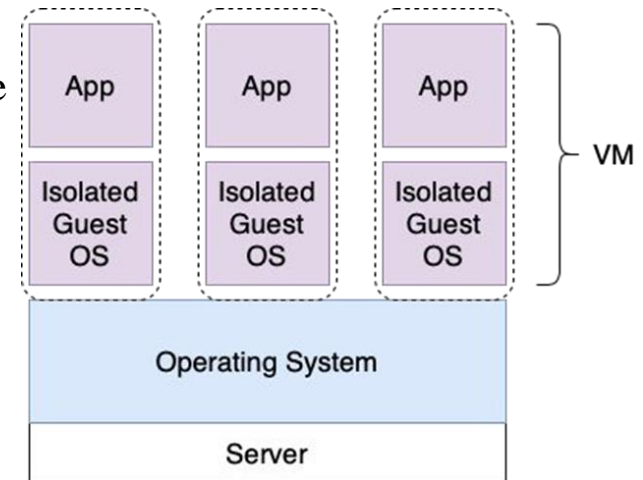
Автоматическая генерация аппаратное представление без необходимости детализации. Запуск доступен для «всех»!

# Проблемы с OpenSource

- Требуется определенные навыки работы с Linux, git и открытым ПО
- чем выше навыки разработки, тем лучше: управление OpenLane – make, shell, python, tcl + json, yaml + ....
- знакомство с контейнеризацией/виртуализацией

Короткая версия инструкции по установке:

1. Get Docker
2. Get Python 3.6
3. Get git
4. Get GNU Make
5. `cd $HOME`
6. `git clone https://github.com/The-OpenROAD-Project/OpenLane`
7. `cd OpenLane`
8. `make`
9. `make test` или `make mount`
10. `./flow.tcl --design xtea`



Полная версия инструкции по установке:

[https://openlane.readthedocs.io/en/latest/getting\\_started/installation.html](https://openlane.readthedocs.io/en/latest/getting_started/installation.html)

# Внутри OpenLane

## 1. Synthesis

1. `yosys/abc` - Perform RTL synthesis and technology mapping.
2. `openSTA` - Performs static timing analysis on the resulting netlist to generate timing reports

## 2. Floorplaning

1. `init_fp` - Defines the core area for the macro as well as the rows (used for placement) and the tracks (used for routing)
2. `ioplacer` - Places the macro input and output ports
3. `pdngen` - Generates the power distribution network
4. `tapcell` - Inserts welltap and decap cells in the floorplan

## 3. Placement

1. `RePLace` - Performs global placement
2. `Resizer` - Performs optional optimizations on the design
3. `OpenDP` - Performs detailed placement to legalize the globally placed components

## 4. CTS

1. `TritonCTS` - Synthesizes the clock distribution network (the clock tree)

## 5. Routing

1. `FastRoute` - Performs global routing to generate a guide file for the detailed router
2. `TritonRoute` - Performs detailed routing
3. `OpenRCX` - Performs SPEF extraction

## 6. Tapeout

1. `Magic` - Streams out the final GDSII layout file from the routed def
2. `KLayout` - Streams out the final GDSII layout file from the routed def as a back-up

## 7. Signoff

1. `Magic` - Performs DRC Checks & Antenna Checks
2. `KLayout` - Performs DRC Checks
3. `Netgen` - Performs LVS Checks
4. `CVC` - Performs Circuit Validity Checks

# OpenLane Output

```
<design_name>
├─ config.json/config.tcl
├─ runs
│   └─ <tag>
│       ├── config.tcl
│       ├── {logs, reports, tmp}
│       ├── cts
│       ├── signoff
│       ├── floorplan
│       ├── placement
│       ├── routing
│       └─ synthesis
│   └─ results
│       ├── final
│       ├── cts
│       ├── signoff
│       ├── floorplan
│       ├── placement
│       ├── routing
│       └─ synthesis
```

[https://openlane.readthedocs.io/en/latest/reference/interactive\\_mode.html](https://openlane.readthedocs.io/en/latest/reference/interactive_mode.html)



# OpenROAD

- OpenLane не отдельный пакет! Это набор скриптов, предназначенный для максимальной автоматизации процесса использования сторонних открытых программ – и ключевая среди них OpenROAD
- OpenROAD – свободная реализация аналог Verilog + libraries, constraints
- Два поддерживаемых режима – OpenLane
- и OpenROAD-flow-scripts – много, очень много tcl

## Open-Source PDKs

*реальные:*

Skywater130 - 130nm

GlobalFoundries180

*учебные*

Nangate45 - 45nm

ASAP7 - Predictive FinFET 7nm

Proprietary PDKs

GF180 - 180nm

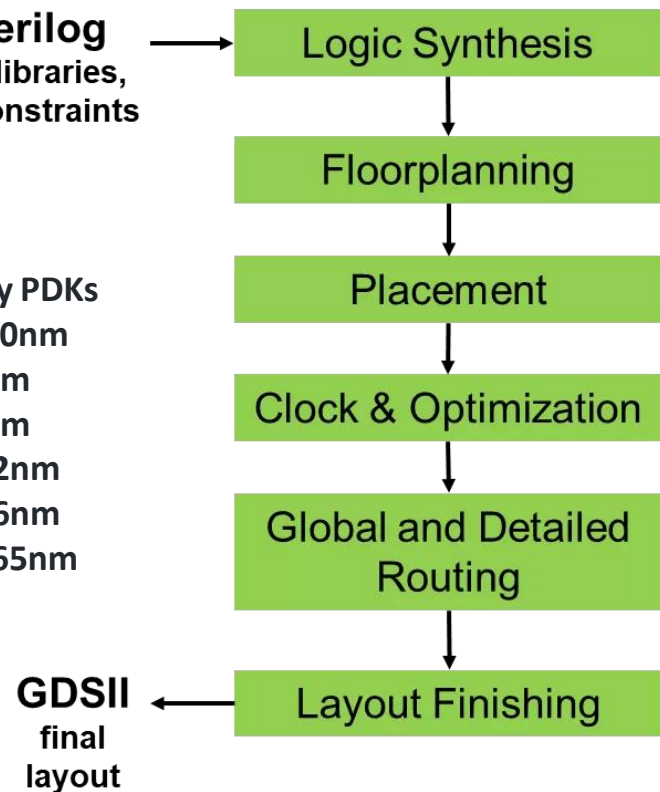
GF55 - 55nm

GF12 - 12nm

Intel22 - 22nm

Intel16 - 16nm

TSMC65 - 65nm



# Автоматизированный синтез

# Спецификация

## Спецификация

### Описание функционала

Схема должна включать освещение при нажатии кнопки или запрашивать подзарядку при низком заряде батареи

№	Описание параметра	Мин	Тип	Макс	Ед.
1.	Процесс	3.3V IO, SAED 90nm			
2.	Напряжение	1.08	1.2	1.32	В
3.	Температура	-40		125	°C
4.	Мощность			100	мВт
5.	Площадь			2	мкм <sup>2</sup>
6.	Частота	1 ГГц			
....	....	....	....	....	....
....	....	....	....	....	....
....	....	....	....	....	....

**Спецификация** - это список целей, которые должны быть достигнуты в данном проекте.

**Описание функционала** (поведение) должно быть переведено на язык описания аппаратуры (HDL), понятный инструментам EDA.

**Описание на уровне RTL** используется в качестве входных данных

```
if button1_pressed
    if (battery_charge > 10)
        turn_on_light();
    else
        prompt_for_recharge();
```

# Условия функционирования/рабочие условия (Operating Conditions)

## Спецификация

### Описание функционала

Схема должна включать освещение при нажатии кнопки или запрашивать подзарядку при низком заряде батареи

№	Описание параметра	Мин	Тип	Макс	Ед.
1.	Процесс	3.3V IO, SAED 90nm			
2.	Напряжение	1.08	1.2	1.32	В
3.	Температура	-40		125	°C
4.	Мощность			100	мВт
5.	Площадь			2	мкм <sup>2</sup>
6.	Частота	1 ГГц			
....	....	....	....	....	....
....	....	....	....	....	....
....	....	....	....	....	....

Характеристики зависят от условий работы (**operating conditions**)

Характеристики могут меняться в следствие

- Вариаций техпроцесса (**Process**)
- Изменения напряжения (**Voltage**)
- Изменения температуры (**Temperature**)

Необходим анализ работы проекта при различных комбинациях P, V, T (**PVT corners**)

- Fast , 1.08V, 125C
- Slow, 1.2V, 125C

Вариации техпроцесса : Fast (минимальные задержки), Slow (максимальные) и типичные

# Проектные ограничения

## Спецификация

### Описание функционала

Схема должна включать освещение при нажатии кнопки или запрашивать подзарядку при низком заряде батареи

№	Описание параметра	Мин	Тип	Макс	Ед.
1.	Процесс	3.3V IO, SAED 90nm			
2.	Напряжение	1.08	1.2	1.32	В
3.	Температура	-40		125	°C
4.	Мощность			100	мВт
5.	Площадь			2	мкм <sup>2</sup>
6.	Частота	1 ГГц			
....	....	....	....	....	....
....	....	....	....	....	....
....	....	....	....	....	....

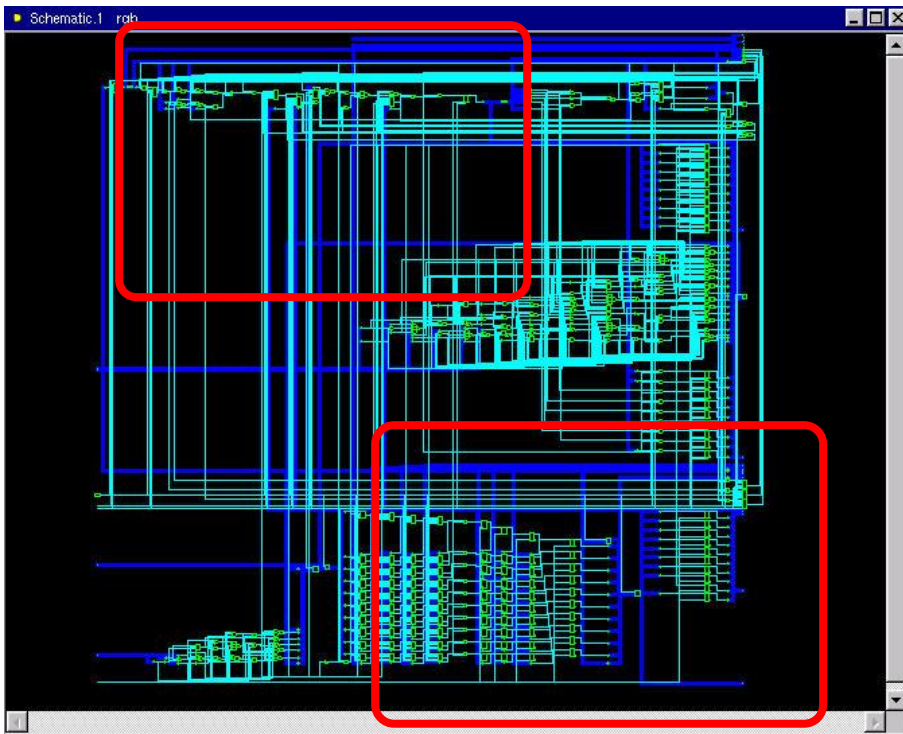
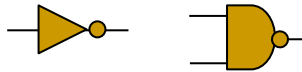
Характеристика проекта обычно описываются через ограничения (design constraints)

Мощность  $\leq 100$  мВт  
Площадь  $< 2$  мкм<sup>2</sup>  
Частота  $> 1$  ГГц

Проектные ограничения описанные в специальном формате (SDC) являются входными данным для синтеза

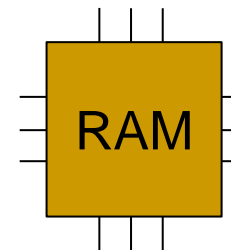
# Логическая схема

Цифровые стандартные ячейки



Сложнофункциональный  
(Intellectual property (IP)) блок

- Схема состоит из цифровых стандартных ячеек и IP-блоков  
Библиотеки стандартных ячеек и IP-блоков должны быть предоставлены как входные данные при синтезе



# Данные для логического синтеза



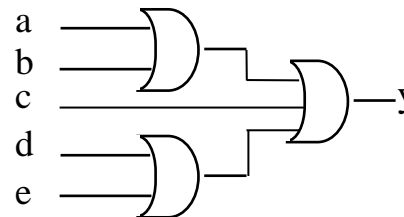
# Главные компромиссы оптимизации

Процесс проектирования схемы – компромисс между временными параметрами, мощностью и занимаемой схемой площадью

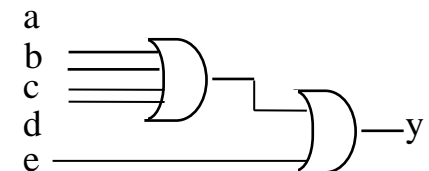
- Временная оптимизация
  - Цель: маленькие задержки
- Оптимизация мощности
  - Цель: низкая потребляемая мощность
- Оптимизация площади
  - Цель: маленькая площадь

Ячейка	Мощность
	2
	2.5
	3

Одинаковая функция:  $Y=a+b+c+d$



Полная мощность: ~6

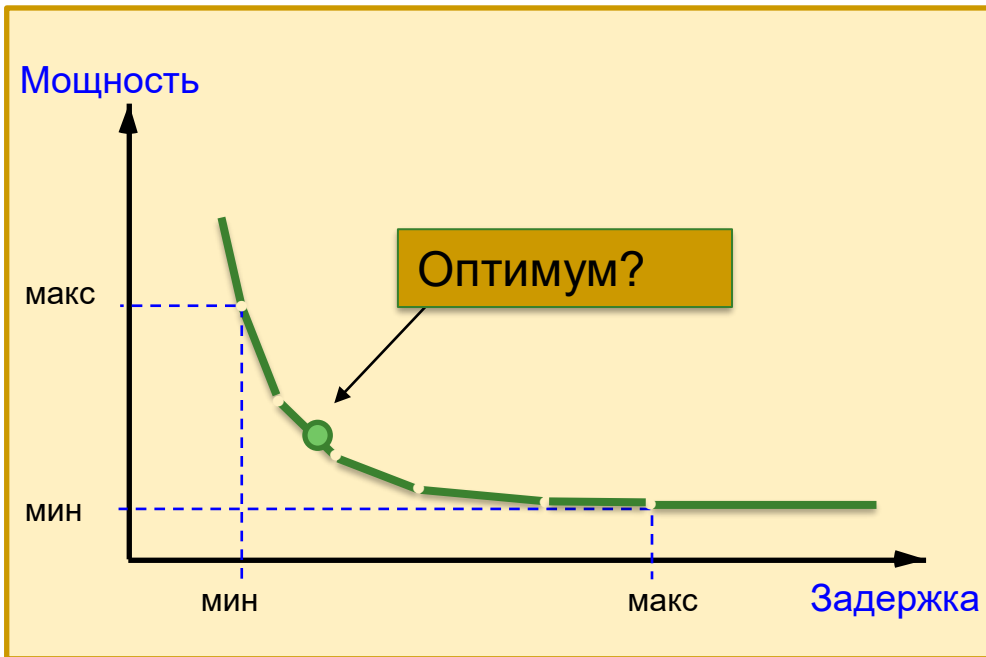


Полная мощность: ~5



# Временной анализ и ограничения

# Компромисс характеристик проекта (design trade-off)

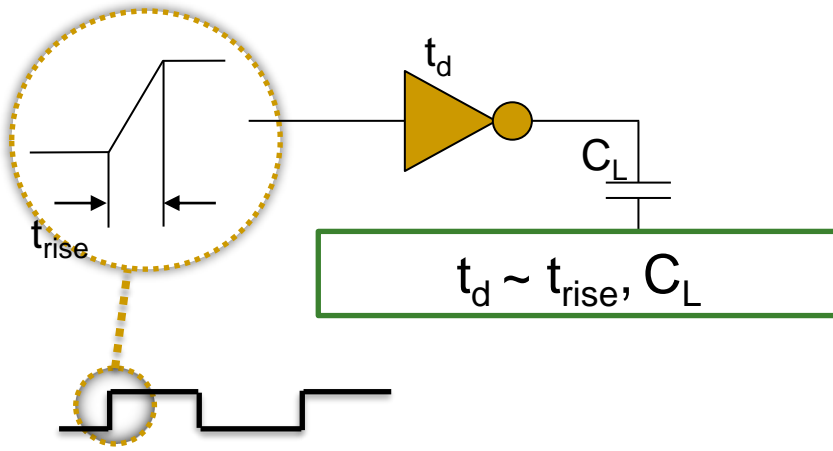
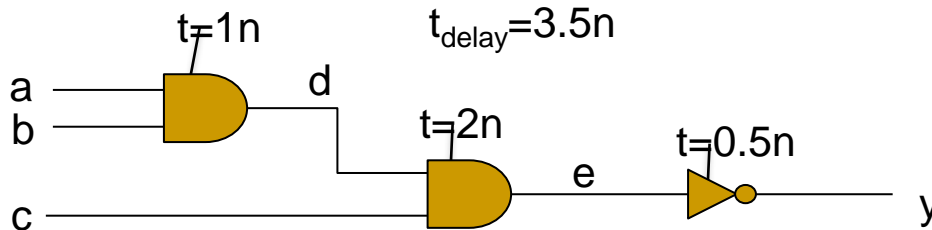


- При автоматизированном синтезе обычно рассчитывается задержка (вместо частоты)

Мощность  $\leq 100$  мВт  
Площадь  $< 2$  мкм<sup>2</sup>

Частота  $> 1$  ГГц **или**  
Задержка  $< 1$  нс

# Задержка пути

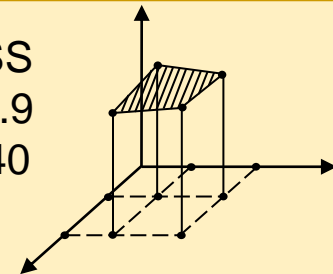


$t_{\text{rise}}$  10ps – 120ps  
 $C_{\text{load}}$  10fF – 50 fF

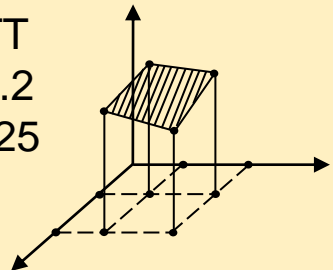
$t_d \sim f(\text{PVT}) !$

Process: TT, FF, SS, etc.  
 Voltage:  $\pm 10\%$   
 Temperature:  $-40 \square 125^\circ\text{C}$

P:SS  
 V:0.9  
 T:-40

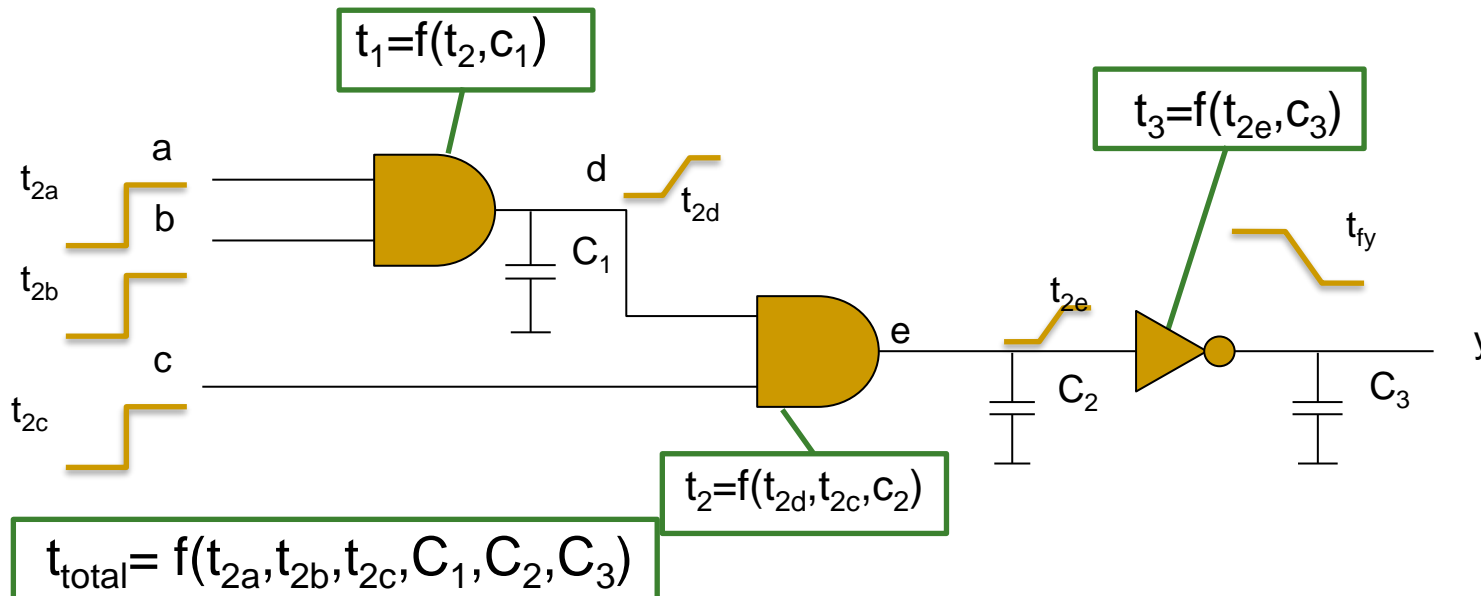


P:TT  
 V:1.2  
 T:125



# Расчет задержки

Для выбранного PVT



Работоспособность схемы может быть гарантирована только при отсутствии нарушений временных ограничений для всех путей и PVT углов!

# Временные отчеты

# Отчеты

```
g:.\nmanufacturability.rpt\nmetrics.csv\n\n---cts\n11-cts_sta.clock_skew.rpt\n11-cts_sta.max.rpt\n11-cts_sta.min.rpt\n11-cts_sta.rpt\n11-cts_sta.tns.rpt\n11-cts_sta.wns.rpt\n13-cts_rsz_sta.area.rpt\n13-cts_rsz_sta.clock_skew.rpt\n13-cts_rsz_sta.max.rpt\n13-cts_rsz_sta.min.rpt\n13-cts_rsz_sta.power.rpt\n13-cts_rsz_sta.rpt\n13-cts_rsz_sta.slew.rpt\n13-cts_rsz_sta.tns.rpt\n13-cts_rsz_sta.wns.rpt\n13-cts_rsz_sta.worst_slack.rpt\n\n---eco\n---floorplan\n3-initial_fp_core_area.rpt\n3-initial_fp_die_area.rpt\n\n---placement\n7-gpl_sta.clock_skew.rpt\n7-gpl_sta.max.rpt\n7-gpl_sta.min.rpt\n7-gpl_sta.rpt\n7-gpl_sta.tns.rpt\n7-gpl_sta.wns.rpt\n8-pl_rsz_sta.area.rpt\n8-pl_rsz_sta.clock_skew.rpt\n8-pl_rsz_sta.max.rpt\n8-pl_rsz_sta.min.rpt\n8-pl_rsz_sta.power.rpt\n8-pl_rsz_sta.rpt\n8-pl_rsz_sta.slew.rpt\n8-pl_rsz_sta.tns.rpt\n8-pl_rsz_sta.wns.rpt\n8-pl_rsz_sta.worst_slack.rpt
```

1. STA – на всех основных этапах!

2. Основные метрики WNS/TNS – worst/total negative slack

3. slack - запас на выполнения ограничения

Частота 1 ГГц / задержка 0,999 нс: **slack = 1 пс**

4. Одновременно анализируется мощность, площадь

5. Анализ проводится для нескольких PVT-углов

```
---routing\n15-rt_rsz_sta.area.rpt\n15-rt_rsz_sta.clock_skew.rpt\n15-rt_rsz_sta.max.rpt\n15-rt_rsz_sta.min.rpt\n15-rt_rsz_sta.power.rpt\n15-rt_rsz_sta.rpt\n15-rt_rsz_sta.slew.rpt\n15-rt_rsz_sta.tns.rpt\n15-rt_rsz_sta.wns.rpt\n15-rt_rsz_sta.worst_slack.rpt\n\n---signoff\n---synthesis\n1-synthesis.AREA 0.chk.rpt\n1-synthesis.AREA 0.stat.rpt\n1-synthesis_dff.stat\n1-synthesis_pre.stat\n2-syn_sta.area.rpt\n2-syn_sta.clock_skew.rpt\n2-syn_sta.max.rpt\n2-syn_sta.min.rpt\n2-syn_sta.power.rpt\n2-syn_sta.rpt\n2-syn_sta.slew.rpt\n2-syn_sta.tns.rpt\n2-syn_sta.wns.rpt\n2-syn_sta.worst_slack.rpt
```

# Отчеты

В отчете рассчитывается два времени:

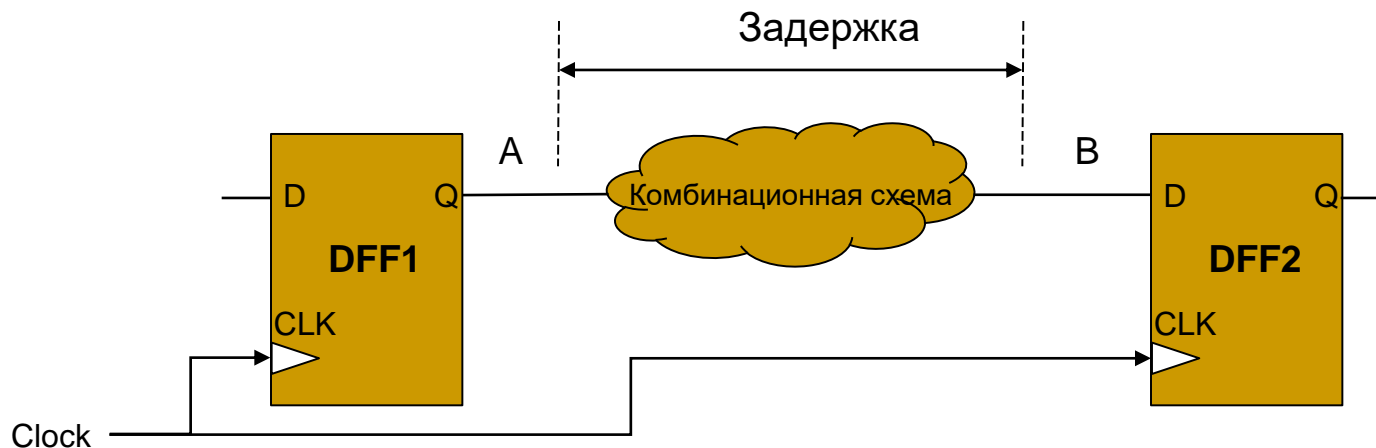
- Arrival – фактическое прохождение сигнала по схеме
- Определяется задержкой вентиля и цепей + [внешнее ограничение]
- Required требуемое:
- T – setup – [output] – [uncertainty]

----- Typical Corner -----					
Startpoint: addr[3] (input port clocked by config_clk)					
Endpoint: out (output port clocked by config_clk)					
Path Group: config_clk					
Path Type: max					
Corner: <del>typ</del>					
Fanout	Cap	Slew	Delay	Time	Description
-----					
			0.00	0.00	clock config_clk (rise edge)
			0.00	0.00	clock network delay (propagated)
			6.00	6.00	input external delay
		0.03	0.02	6.02	^ <del>addr</del> [3] (in)
2	0.00				<del>addr</del> [3] (net)
		0.03	0.00	6.02	^ input4/A (sky130_fd_sc_hd_clkbuf_1)
		0.13	0.14	6.16	^ input4/X (sky130_fd_sc_hd_clkbuf_1)
2	0.01				net4 (net)
		0.13	0.00	6.16	^ _101_/A (sky130_fd_sc_hd_clkbuf_2)
		0.12	0.20	6.36	^ _101_/X (sky130_fd_sc_hd_clkbuf_2)
5	0.02				_083_ (net)
		0.12	0.00	6.36	^ _106_/S0 (sky130_fd_sc_hd_mux4_1)
		0.08	0.58	6.94	v _106_/X (sky130_fd_sc_hd_mux4_1)
1	0.00				_088_ (net)
		0.08	0.00	6.94	v _107_/B (sky130_fd_sc_hd_and2b_1)
		0.03	0.17	7.11	v _107_/X (sky130_fd_sc_hd_and2b_1)
1	0.00				_089_ (net)
		0.03	0.00	7.11	v _115_/A3 (sky130_fd_sc_hd_o32a_1)
		0.06	0.31	7.42	v _115_/X (sky130_fd_sc_hd_o32a_1)
1	0.01				_035_ (net)
		0.06	0.00	7.42	v _116_/S (sky130_fd_sc_hd_mux2_1)
		0.07	0.35	7.77	v _116_/X (sky130_fd_sc_hd_mux2_1)
1	0.01				_036_ (net)
		0.07	0.00	7.77	v _117_/A (sky130_fd_sc_hd_clkbuf_1)
		0.03	0.10	7.87	v _117_/X (sky130_fd_sc_hd_clkbuf_1)
1	0.00				net25 (net)
		0.03	0.00	7.87	v output25/A (sky130_fd_sc_hd_buf_2)
		0.09	0.18	8.05	v output25/X (sky130_fd_sc_hd_buf_2)
1	0.03				out (net)
		0.09	0.00	8.05	v out (out)
				8.05	data arrival time
-----					
			30.00	30.00	clock config_clk (rise edge)
			0.00	30.00	clock network delay (propagated)
			-0.25	29.75	clock uncertainty
			0.00	29.75	clock <del>reconvergence</del> pessimism
			-6.00	23.75	output external delay
				23.75	data required time
-----					
				23.75	data required time
				-8.05	data arrival time
-----					
				15.70	slack (MET)

# Быстродействия и ограничения

## ■ Основное ограничение

- В синхронной схеме, сигналы на входах регистров должны удовлетворять требованиям setup/hold



Если задержка ( $t_{pd}$ )  $> T_{clock}$  то требование setup не соблюдается в точке B

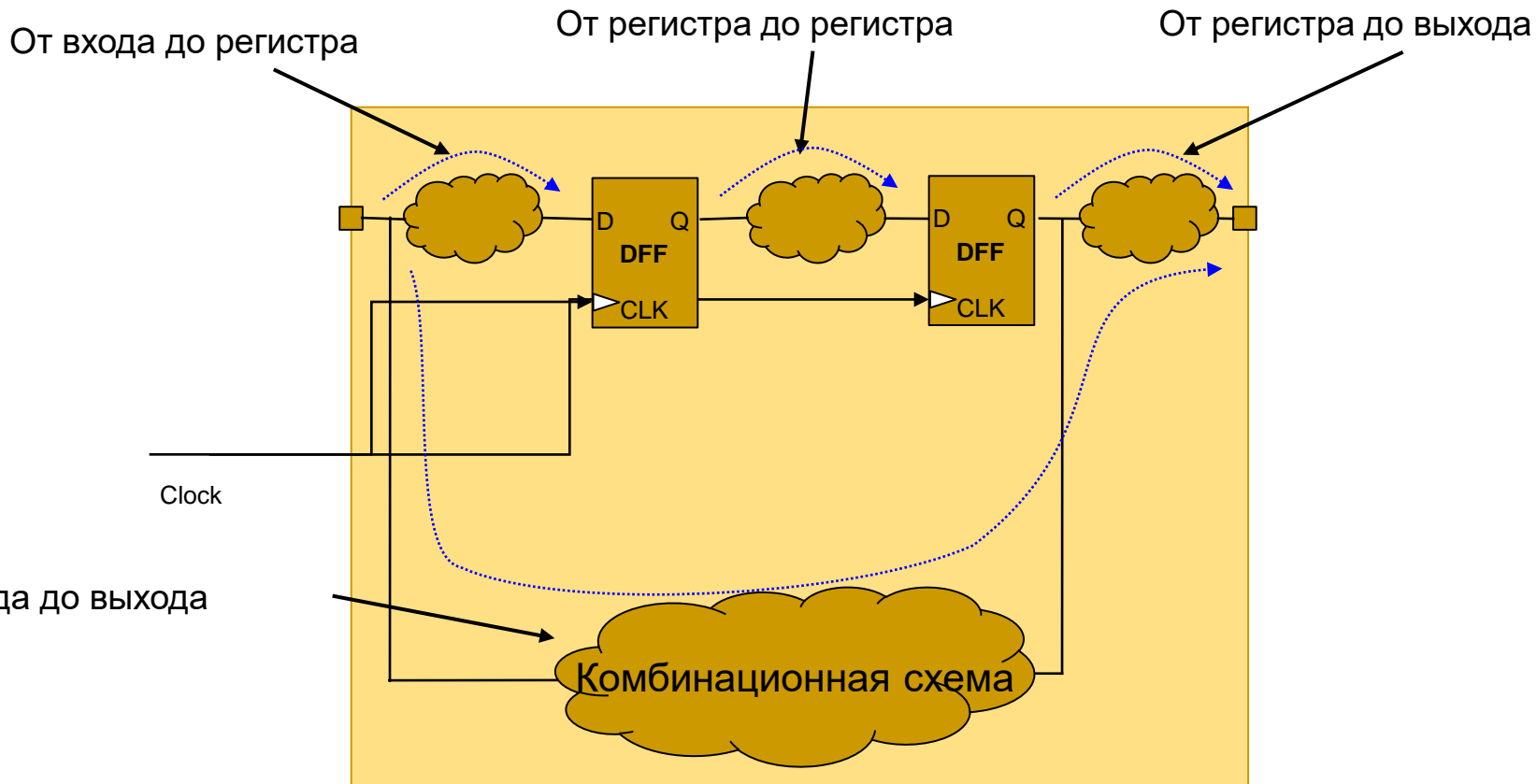
Если кратчайший путь ( $t_{cd}$ )  $< t_{hold}$  то требование hold в DFF2

## ■ Харрис/Харрис – раздел 3.5!



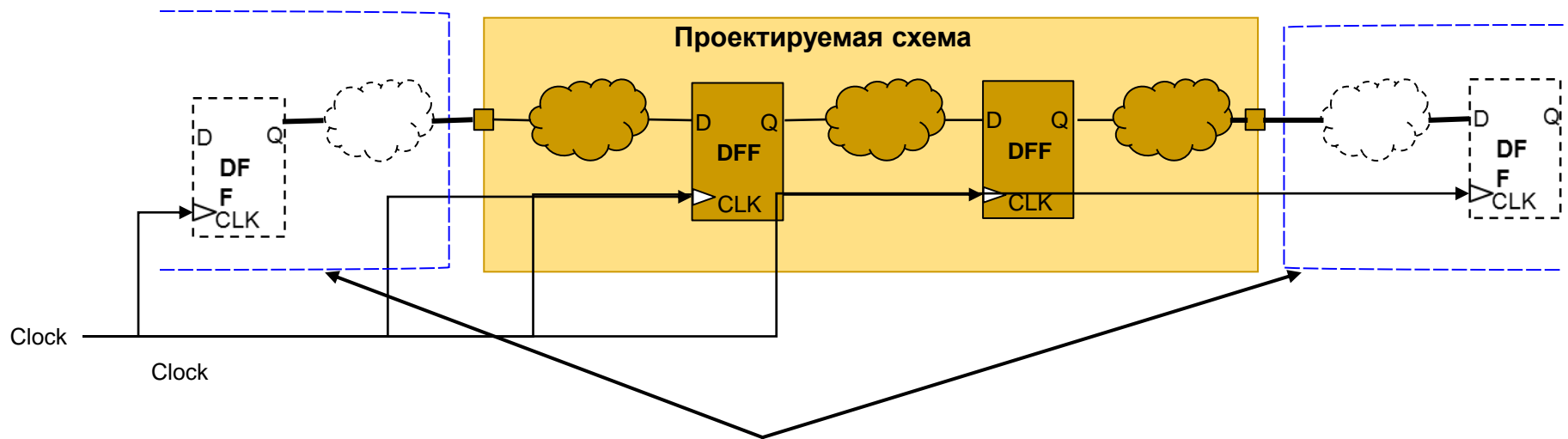
# Типы путей

Существует 4 типа путей в тактируемых схемах



# Тактируемая схема

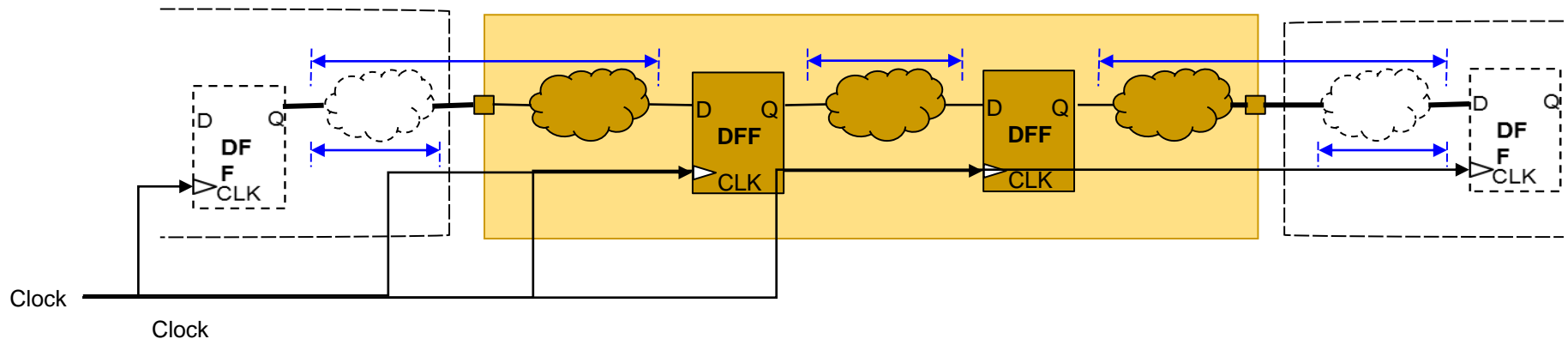
Для облегчения постановки ограничений, считается, что проект тактируется единым сигналом синхронизации.



Внешняя логика считается виртуальной, для того, чтобы использовать однообразные ограничения для всех частей схемы.

# Ограничение быстродействия: Setup/Hold

Межрегистровые схемы ограничены **периодом тактового сигнала** и **требованиями setup/hold** регистров

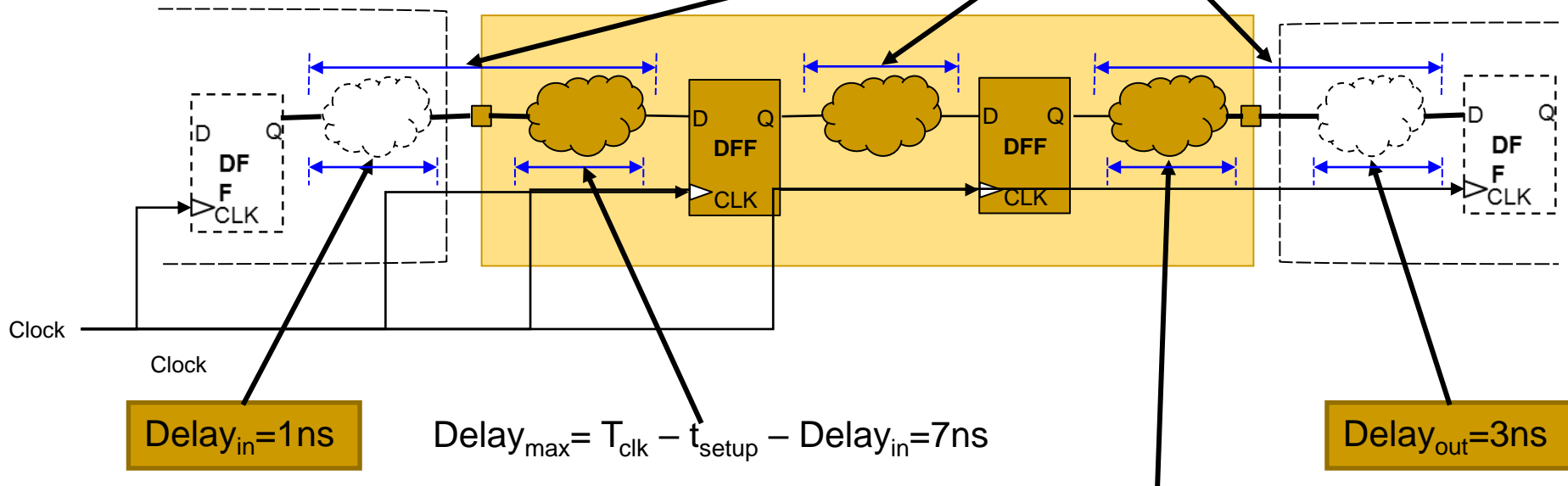


Для расчета задержек схем на входах/выходах необходимо предоставить задержки **виртуальной внешней логики**.

# Ограничение быстродействия: Пример

Период сигнала:  $T_{clk} = 10ns$   
 Время setup DFF:  $t_{setup} = 2ns$

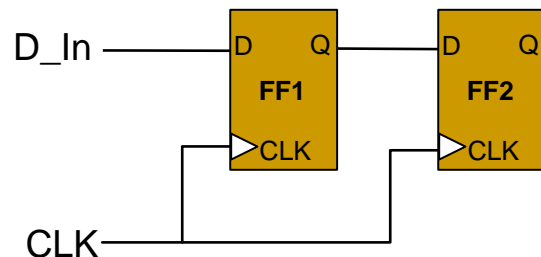
$$Delay_{max} = T_{clk} - t_{setup} = 8ns$$



```
create_clock -period 10 [get_ports Clk]
set_input_delay -max 1 -clock Clk [get_ports IN]
set_output_delay -max 3 -clock Clk [get_ports OUT]
```

# Ограничение быстродействия: Моделирование тактового сигнала

- Design Compiler считает что тактовый сигнал идеален
- Дерево синхронизации строится на этапе топологического проектирования
- Чтобы получить реалистичный проект, необходимо моделировать запаздывание тактового сигнала



Логическая схема

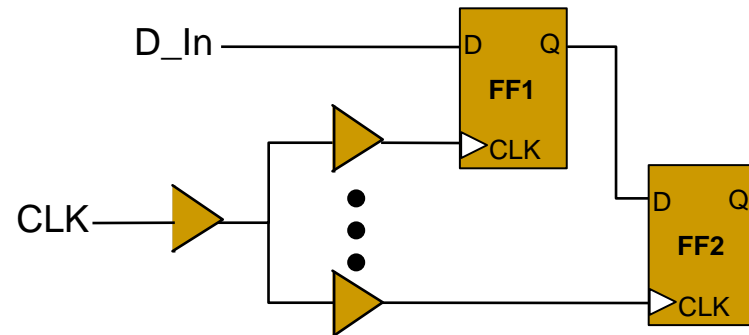
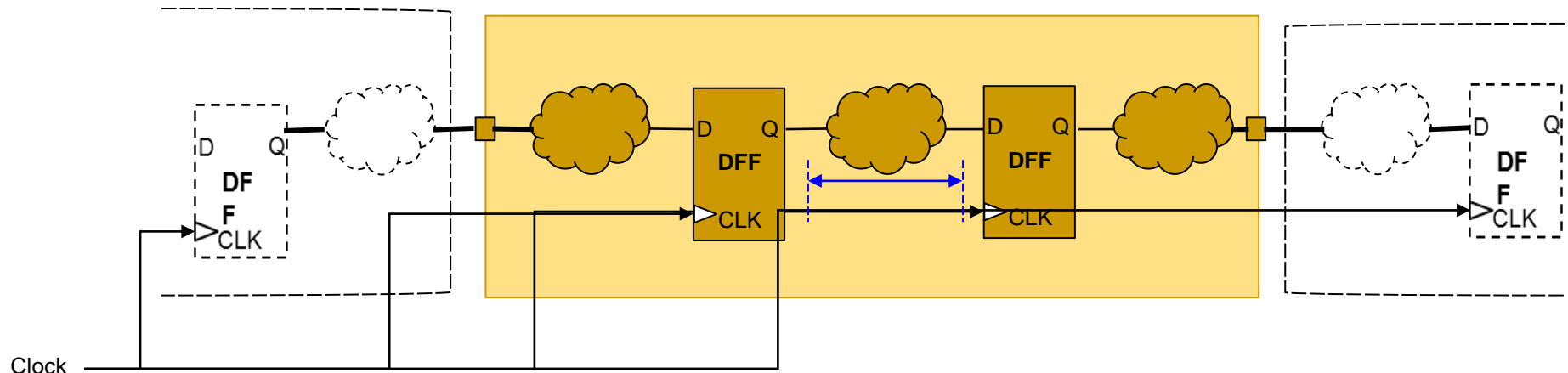
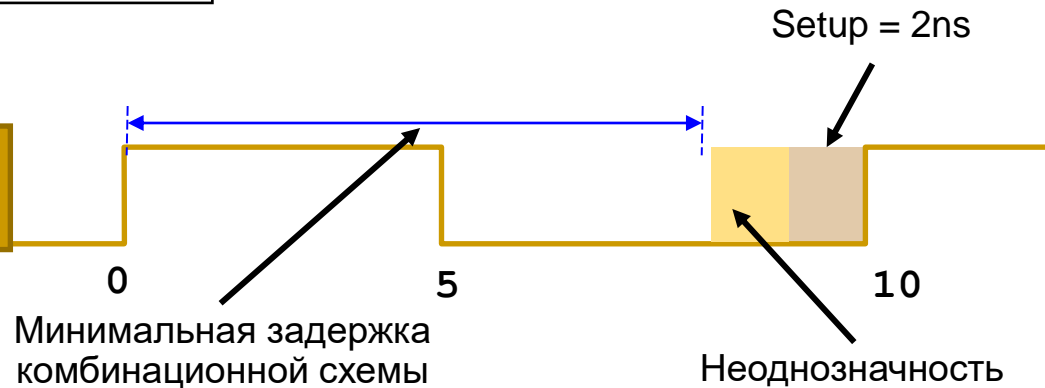


Схема после синтеза топологии

# Пример моделирования неоднозначности тактового сигнала



Период сигнала:  $T_{clk} = 1ns$   
Время setup DFF:  $t_{setup} = 100ps$

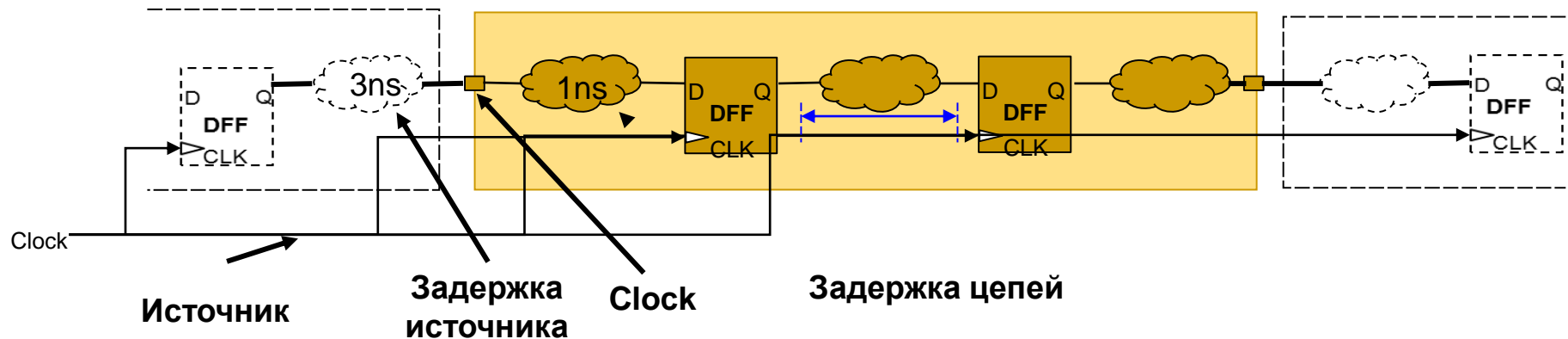


```
set_clock_uncertainty -setup 0.15 [get_clocks CLK]
```

# Моделирование задержки тактового сигнала

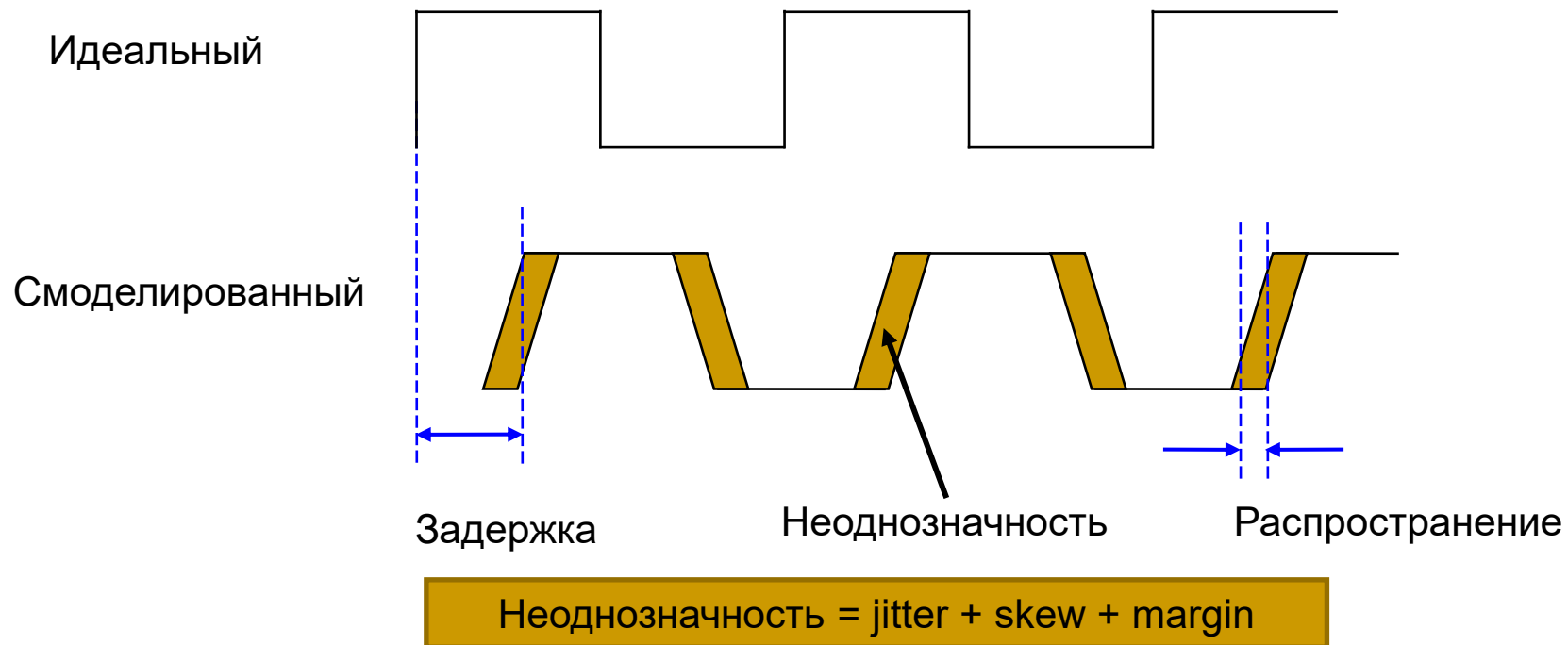
## ■ Задержка бывает двух типов:

- **Задержка источника:** задержка тактового сигнала от его источника до входных портов схемы (полезна когда устройство генерации синхроимпульсов не является частью схемы)
- **Задержка цепей:** средняя задержка распространения тактового сигнала внутри схемы



```
set_clock_latency -source -max 3 [get_clocks CLK]
set_clock_latency -max 1 [get_clocks CLK] ;# pre layout
# set_propagated_clock [get_ports CLK] ;# post layout
```

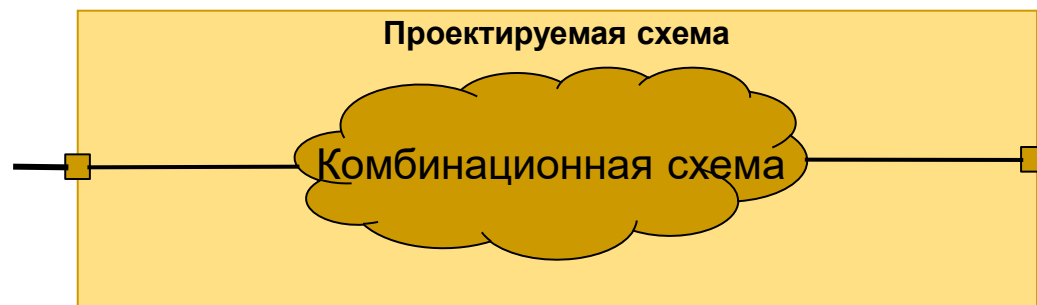
# Моделирование тактового импульса





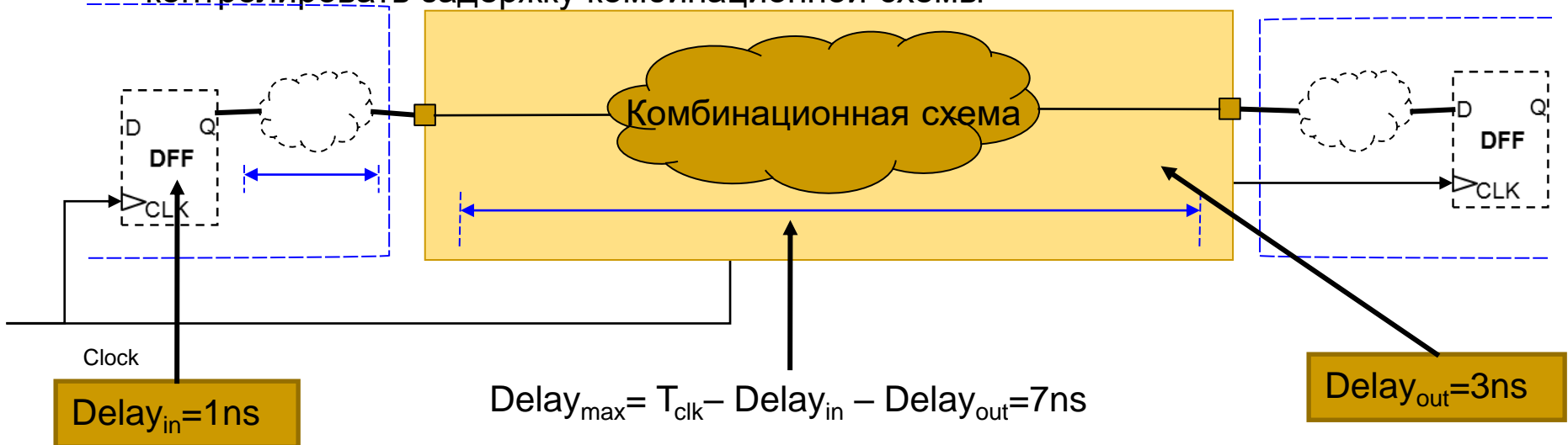
# Комбинационные схемы

- Иногда схема не содержит в себе регистров
- И нет тактового сигнала



# Комбинационные схемы (2)

- Комбинационная схема помещается в то же окружение, что и тактируемые схемы
- Для этого окружения определяется абстрактный тактовый сигнал, называемый виртуальным
- Заданием периода этого сигнала, задержек входов и выходов можно контролировать задержку комбинационной схемы



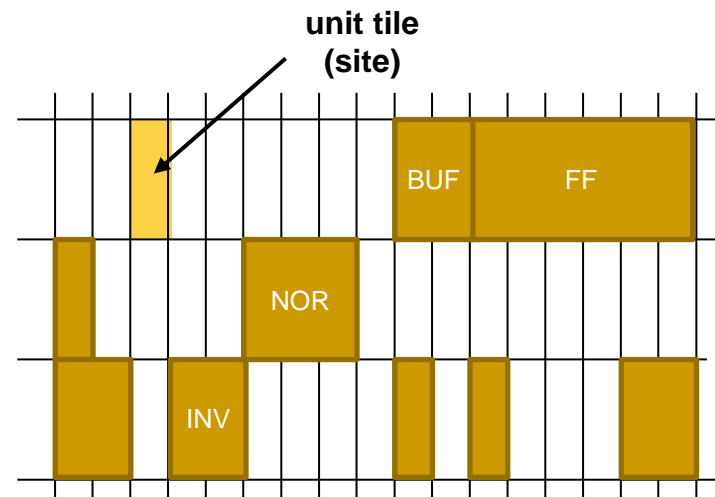
# Библиотека

# Физическая структура стандартной ячейки (1)

- Топология стандартной ячейки полностью зависит от задач, решаемых при проектировании
- Из-за того, что уменьшение площади под разводку более выгодно, чем уменьшение размеров ячеек при минимизации размеров кристалла, то предпочтительнее проектировать ячейки, удобные для работы программы создания межсоединений.

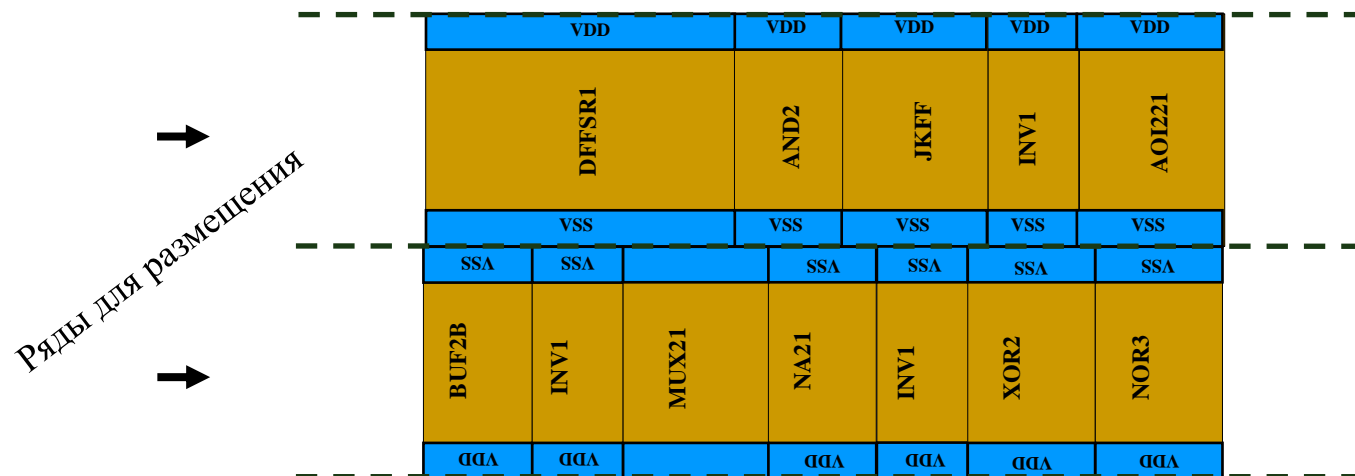
# Физическая структура стандартной ячейки : Unit Tile

- При размещении используется горизонтальная и вертикальная сетка
- Все ячейки библиотеки создаются кратными размеру «unit tile» ячейки

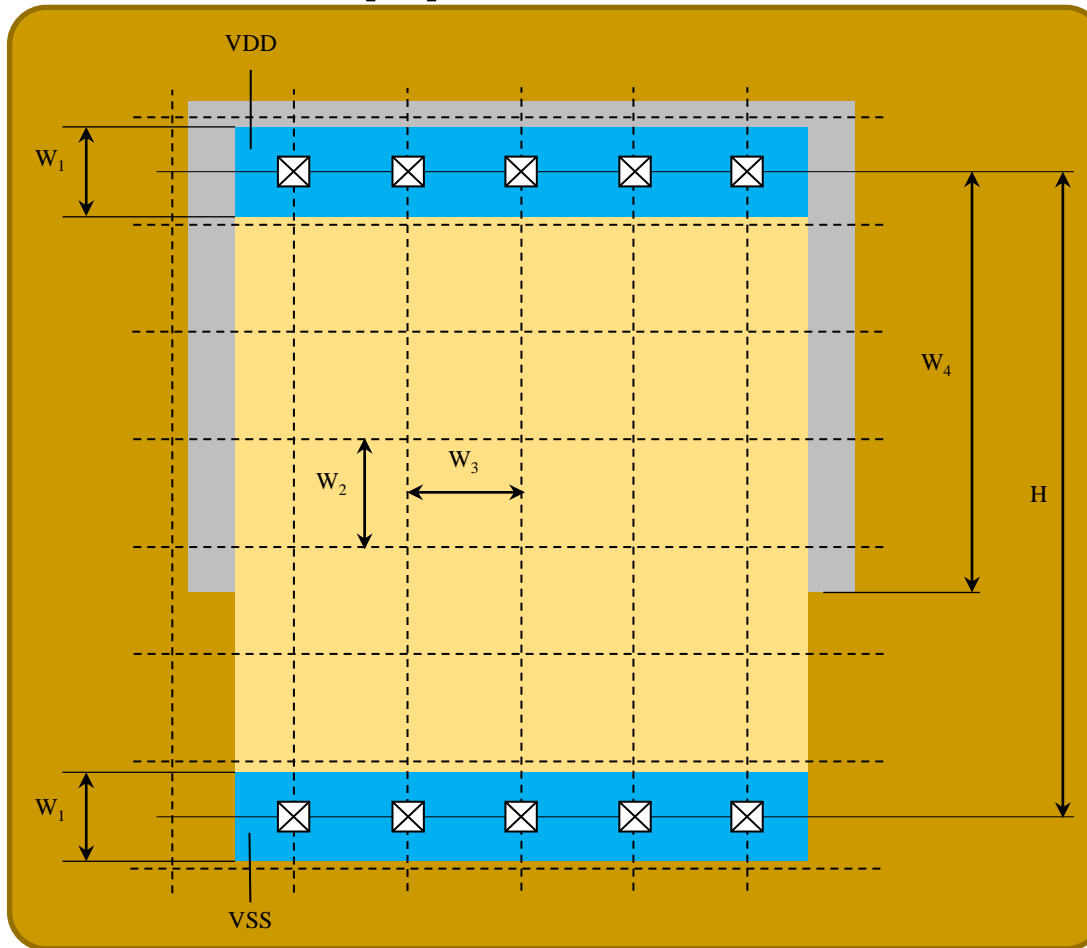


# Физическая структура стандартной ячейки (2)

- Ячейки располагаются в рядах, рядом друг с другом.
- Одну группу ячеек (ряд) продолжает другая группа.
- Ячейки на соседних рядах переворачиваются с целью использования одной шины питания.



# Физическая структура стандартной ячейки (3)



Параметр	Обозначение
Высота ячейки	$H$
Ширина шины питания	$W_1$
Вертикальная сетка	$W_2$
Горизонтальная сетка	$W_3$
Высота N-кармана	$W_4$

# Физическая структура стандартной ячейки (4)

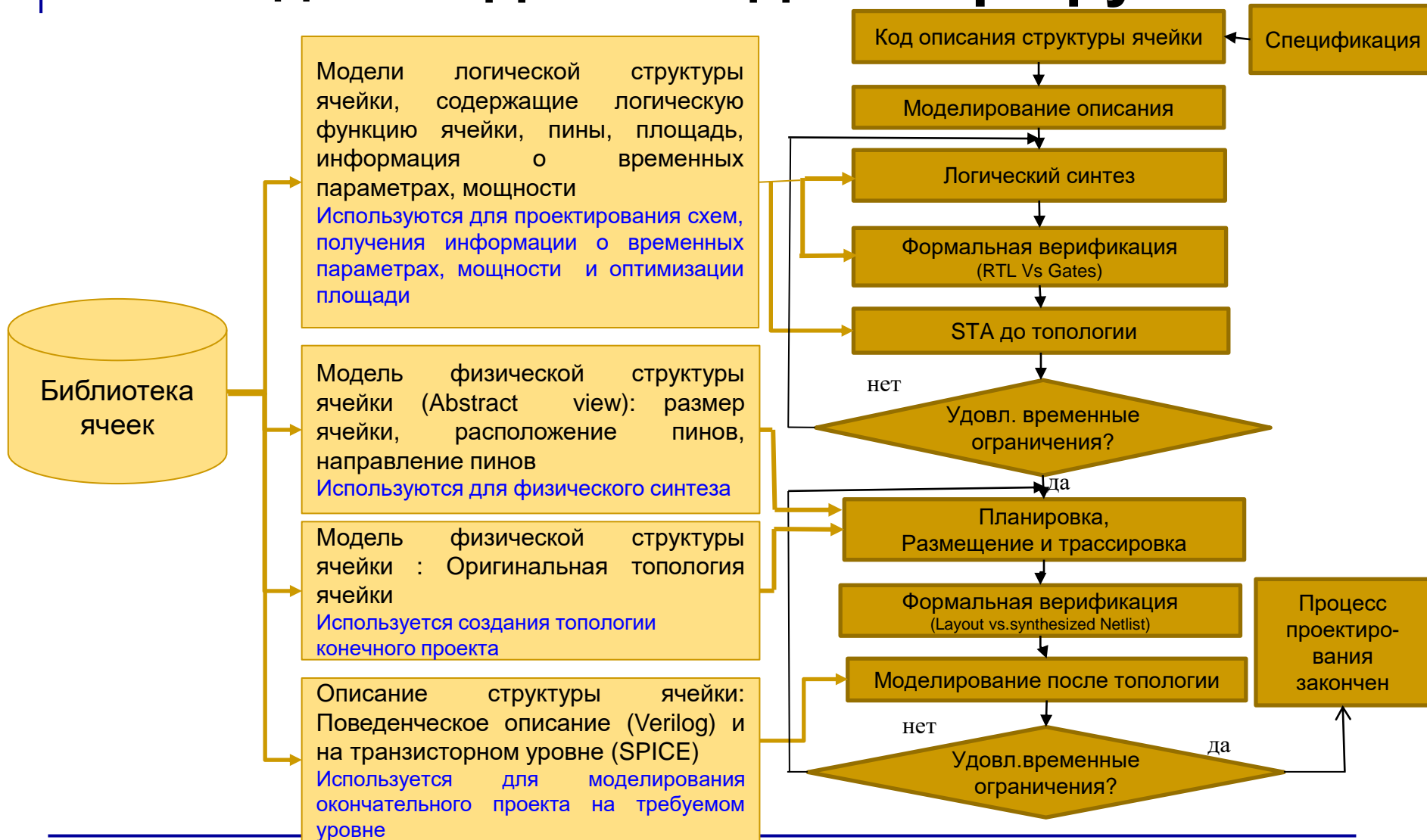
- Все вертикальные размеры в ячейках зафиксированы:
  - Высота ячейки
  - Высота N-кармана
  - Ширина шины
- Горизонтальный размер ограничен вертикальной сеткой
  - Ячейки должны быть кратны размерам сетки
- Размещение пинов очень важно для трассировки
  - Пины должны располагаться в вершинах сетки
  - Расстояние между пинами должно обеспечивать соединение без DRC ошибок



# Результаты работы

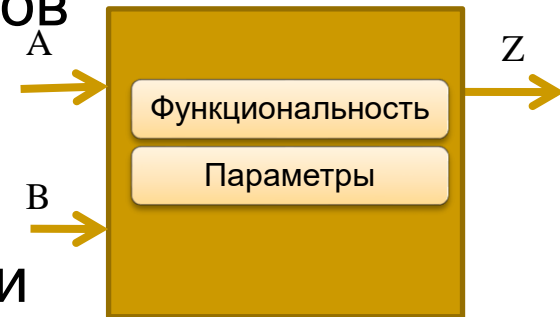
- Два типа результатов:
  - Данные: views, файлы
    - Требуются для маршрута проектирования или процесса, в котором будут использоваться ячейки
    - Документация, отчеты и т.д.
    - Требуются людям, использующим библиотеку и которые уже знакомы с библиотекой

# Необходимые данные для маршрута



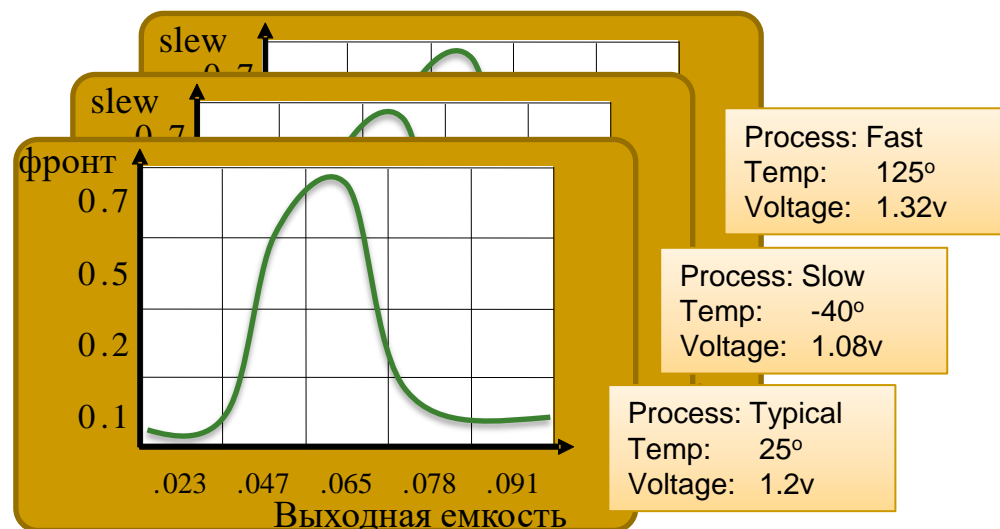
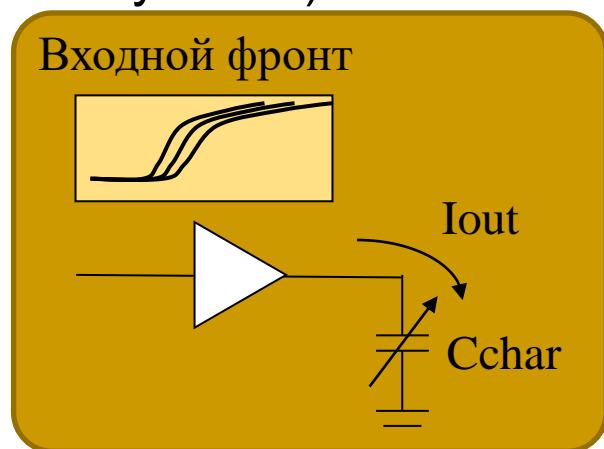
# Логическая модель ячейки

- Логическая модель ячейки создается путем ее характеристики
- Модель ячейки содержит
  - Имя ячейки, пины, направления пинов
  - Функциональность
  - Временные параметры
  - Параметры потребляемой мощности
  - Другие необходимые для EDA программы параметры
    - Емкости пинов
    - И т.д.



# Цель характеристики

- При характеристике рассчитываются параметры ячейки (например: задержка, выходной ток) в зависимости от входных переменных : выходная нагрузка, входной фронт, и т.д.
- Характеризация проводится для различных комбинаций рабочих условий: процесс, напряжение, температура (называемых PVT углами).



# Спецификация и документация библиотек

## Список ячеек

### ■ Требования быстродействия

- Скорость
- Площадь
- Мощность

Функциональность

Логика,  
Буферы,  
Инверторы,  
Триггеры,  
Мультиплексоры  
.....

Нагрузочная способность

1х, 2х, 3х, ...

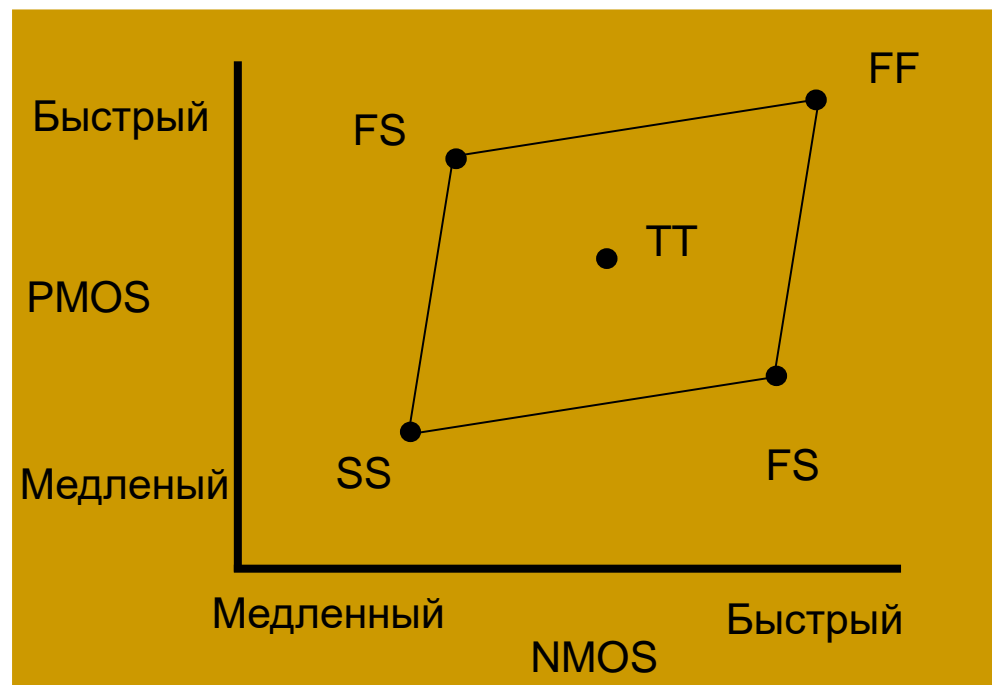
Для различных  
нагрузок

### Краевые случаи

Процесс	TT	SS	FF
Напряжение питания [V]	1.0	0.9	1.1
Температура [°C]	25	125	-40

# Процесс и окружающие условия

- Стандартные ячейки характеризуются по следующим параметрам
  - Разброс питания
  - Вариации процесса
  - Температурный разброс



# Условия характеристики

Название	Процесс (NMOS proc. – PMOS proc.)	Температура (Т)	Напряжение (V)	Примечание
TTNT1p20v	Typical - Typical	25	1.2	Typical corner
SSHT1p08v	Slow - Slow	125	1.08	Slow corner
FFLT1p32v	Fast - Fast	-40	1.32	Fast corner
FFHT1p32v	Fast - Fast	125	1.32	High leakage corner
SSLT1p32v	Slow - Slow	-40	1.32	Low temperature corners
SSLT1p08v	Slow - Slow	-40	1.08	
Низкое напряжение				
TTNT0p80v	Typical - Typical	25	0.80	
SSHT0p70v	Slow - Slow	125	0.70	
FFLT0p90v	Fast - Fast	-40	0.90	
FFHT0p90v	Fast - Fast	125	0.90	
SSLT0p90v	Slow - Slow	-40	0.90	
SSLT0p70v	Slow - Slow	-40	0.70	

# Файл логической модели библиотечной ячейки

- Synopsys Liberty Format (.lib)
  - Библиотека (.lib) – текстовый файл
  - Содержание:
    - Функция ячейки
    - Задержки
    - Rise/Fall времена
    - Площадь ячейки
    - Направления пинов
    - Емкости пинов
    - и т.д.

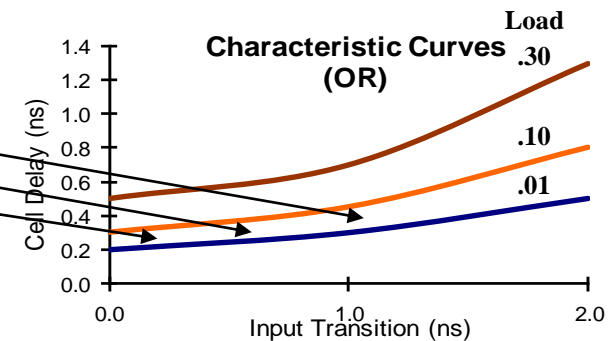
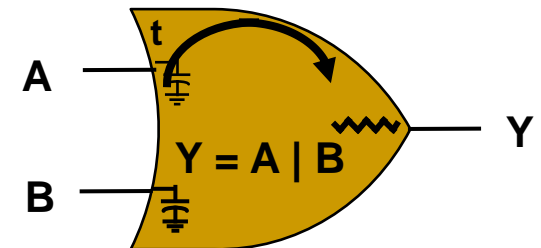
```
library (Digital_Std_Lib) {  
  technology (cmos);  
  delay_model : table_lookup;  
  cell(AND2) {  
    area : 2;  
    pin(A) {  
      direction : input;  
    }  
    pin(B) {  
      direction : input;  
    }  
    pin(Z) {  
      direction : output;  
      function : "A*B";  
      timing() {  
        related_pin : "A" ;  
        timing_type : "combinational" ;  
        cell_rise(...) { values("1.0020, 1.1280")  
        rise_transition(...) { values("0.2069, 0.3315")  
        cell_fall(...) { values("1.0720, 1.2060"); }  
        fall_transition(...) { values("0.2187, 0.3333"); }  
      }  
    }  
  } /* end of cell */  
} /* end of library*/
```



# Пример технологической библиотеки

## Формат Liberty (\*.lib)

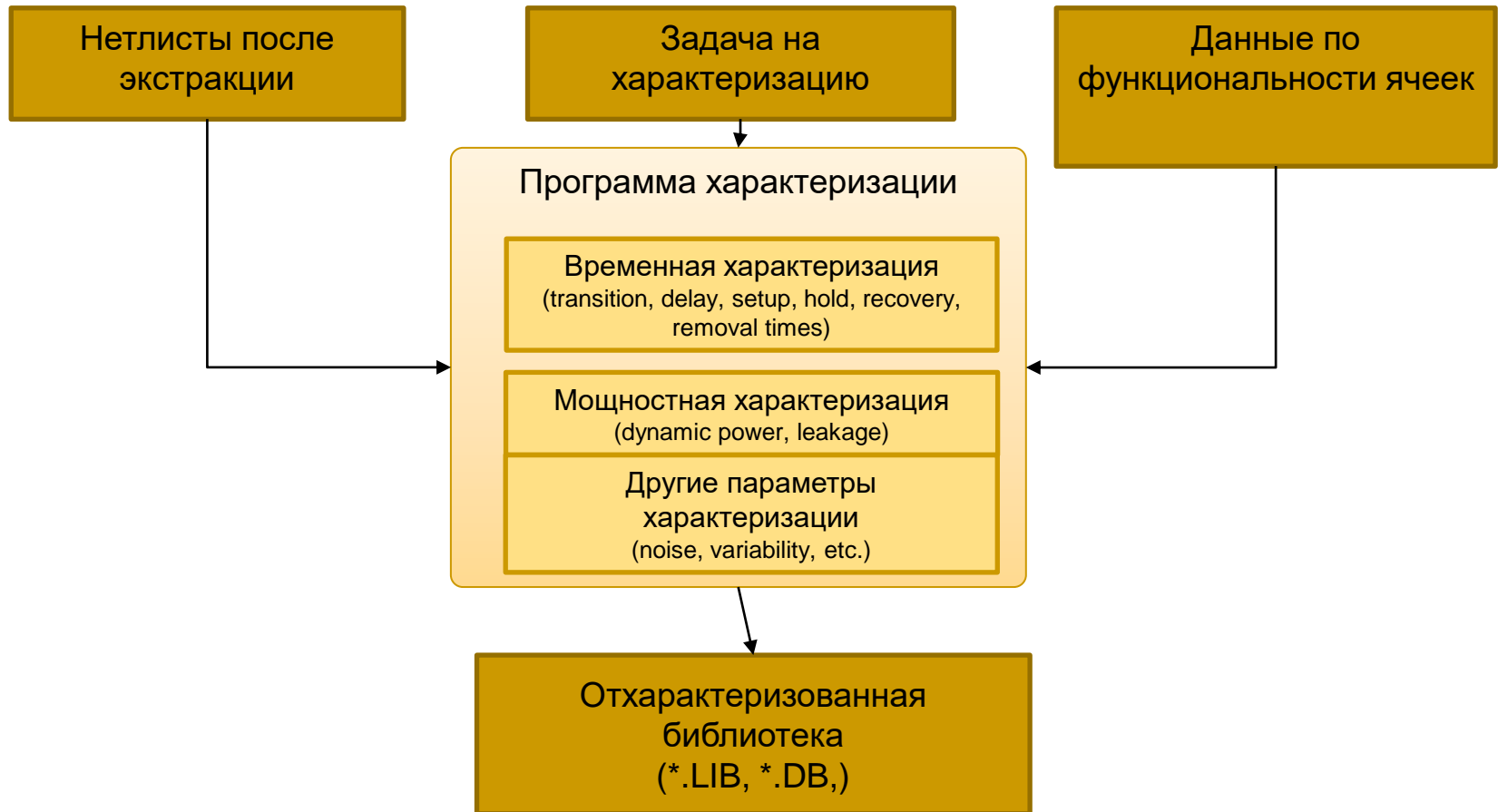
```
cell ( OR2_4x ) {                                     ← Название
  area : 8.000 ;                                       ← Площадь
  pin ( Y ) {
    direction : 2;                                     ← 2 = выход; 1 = вход
    timing ( ) {
      related_pin : "A" ;
      timing_sense : positive_unate ;
      rise_propagation (drive_3_table_1) {
        values ("0.2616, 0.2711, 0.2831,...")
      }
      rise_transition (drive_3_table_2) {
        values ("0.0223, 0.0254, ...")
      }
      . . . .
      function : "(A | B)";                             ← Функция выхода
      max_capacitance : 1.14810 ; ← КТО
      min_capacitance : 0.00220 ;
    }
  }
  pin ( A ) {
    direction : 1;
    capacitance : 0.012000;                             ← Электрические
                                                         характеристики
    . . . .
  }
```



# Как связаны углы и ограничения?

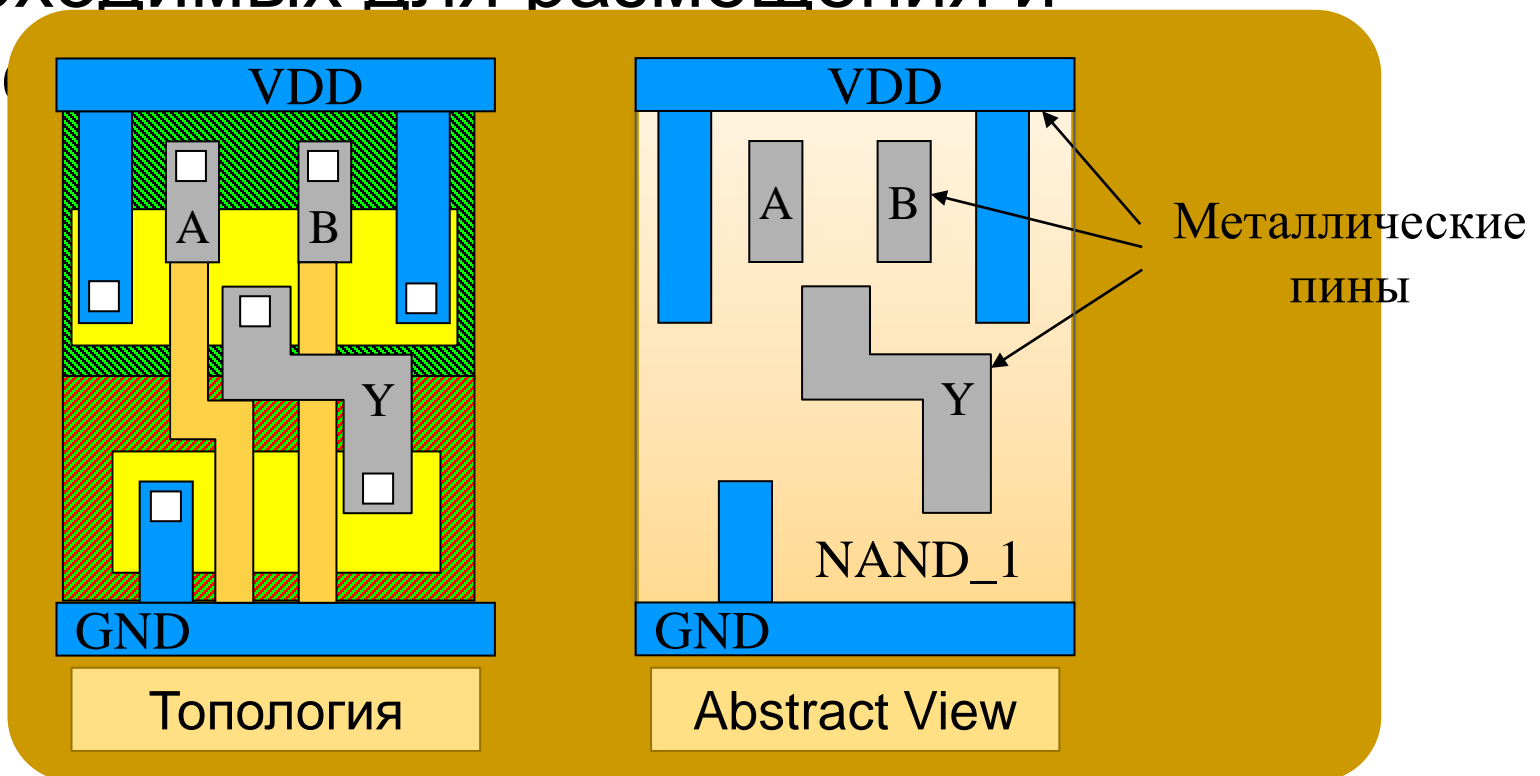
- Пример вопроса с собеседования от Юрия Панчула:
  - С производства пришли два чипа. Мы знаем, что в одном есть небольшое нарушение по setup, а в другом небольшое нарушение по hold. Как можно вернуть их работоспособность?

# Маршрут характеристики



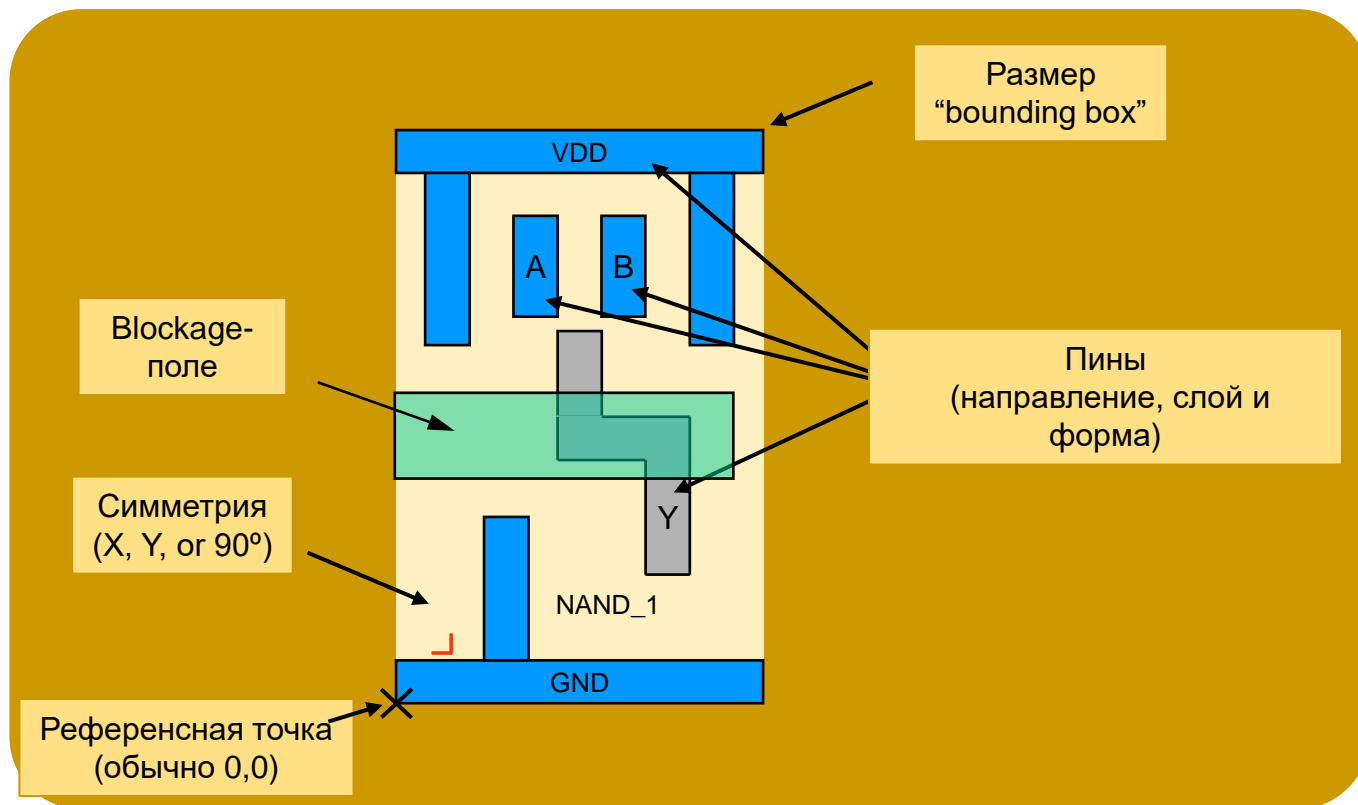
# (Abstract) вид ячейки

Вид «Abstract» содержит только минимальное количество данных, необходимых для размещения и трасс



# Содержимое (Abstract) вида

Вид «Abstract» содержит физическую информацию о стандартных и макро ячейках, необходимых для процесса размещения.



# Library Exchange Format (LEF)

- LEF – текстовый формат. Файл формата LEF содержит физическую абстрактную модель ячейки
  - Размеры ячейки
  - Геометрию пинов
  - Blockage области
  - Информация об антенна-эффekte у пинов
  - ...

```
MACRO single_port_bbb
CLASS BLOCK ;
FOREIGN single_port_bbb ;
ORIGIN 0 0 ;
SIZE 774 BY 547 ;
SYMMETRY X Y R90 ;
PIN OUT
DIRECTION INPUT ;
USE SIGNAL ;
PORT
LAYER M3 ;
RECT 420.180 625.650 420.960 625.810 ;
END
END OUT
OBS
LAYER M1 ;
RECT 0.000 0.000 774.000 547.000 ;
END
END single_port_bbb
```