

---

# Developing the Nanometer

# ASIC

Specs to Silicon



## 0. Инфраструктура семинаров DNA

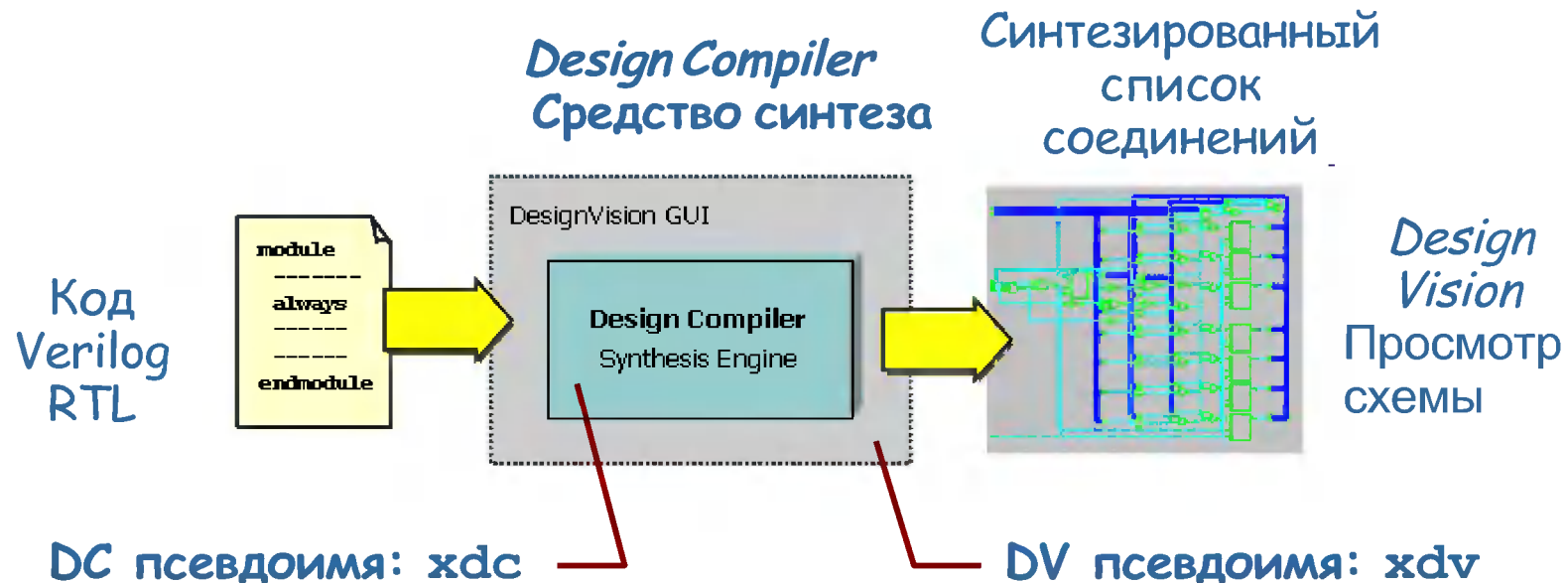
- Подтема 2.4 данной презентации была подготовлена в сотрудничестве с Юрием Панчулом, старшим инженером по разработке аппаратуры компании Imagination Technologies, группа разработки микропроцессора MIPS I6400
- Перевод на русский язык выполнен Александром Барабановым, доцентом кафедры компьютерной инженерии факультета радиофизики, электроники и компьютерных систем Киевского национального университета имени Тараса Шевченко

**Эти лекции и лабораторные работы позволят вам:**

- Понять **все этапы** процесса разработки ASIC.
- Получить базовый **практический** опыт использования средств EDA для: синтеза; моделирования; проверки формальной эквивалентности.
- Научиться идентифицировать типичные **проблемы**: нарушение временных требований; логические ошибки; расфазировка тактовых импульсов; нарушения правил DRC; ошибки производства.
- Научиться отслеживать основные шаги—наложение маски, окисление, имплантация, и т.д.—при **производстве** микросхем на реальном кремниевом кристалле.
- Познакомиться с **современными технологиями**: функциональное покрытие; анализ ошибок задержек; транзисторы finFET; 3-D упаковка.

# Что такое синтез?

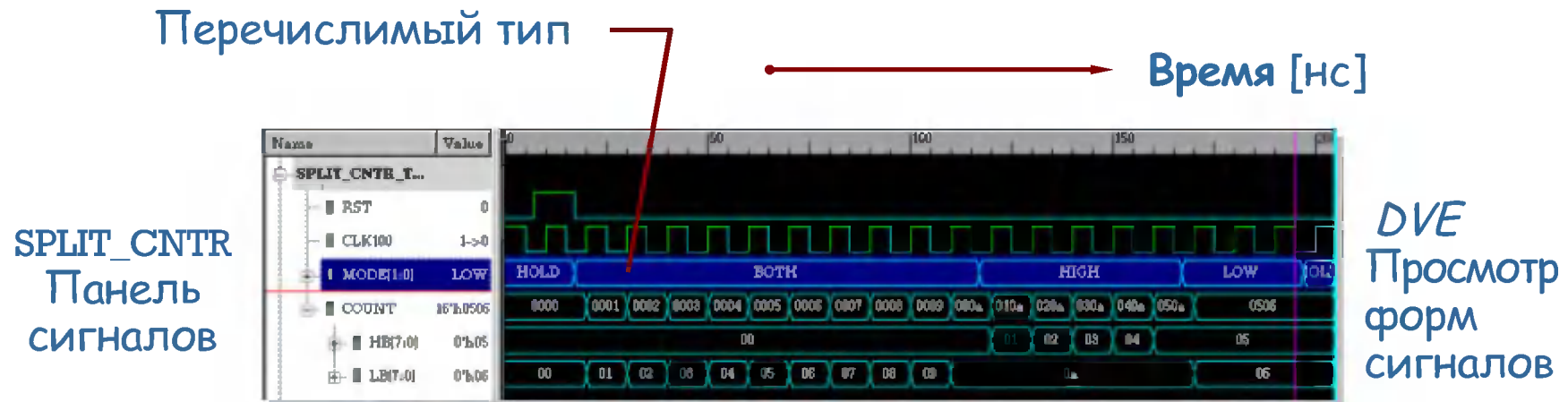
0-4



- Средства логического синтеза компилируют **исходный код** в **логические элементы**.
- Использует элементы, триггеры из библиотеки **стандартных ячеек** (не показана).
- **Оптимизация** логики для быстродействия, меньшей площади или энергопотребления.
- Оценивание **результатов** в сводках временных характеристик, необходимой площади и энергопотребления.

# Что такое моделирование?

0-5



VCS Compile-and-Elaborate псевдоимя: `xsim`  
Discovery Visual Environment псевдоимя: `xdve`

- Симулятор VCS компилирует файлы **исходного кода в модель**.
- Код может включать **тестовый стенд** для проверки проекта.
- Исполняемая модель **загружается в GUI**, и **выполняется некоторое время**.
- Формы сигналов, листинги, сообщения, проверки позволяют обнаружить **ошибки**.

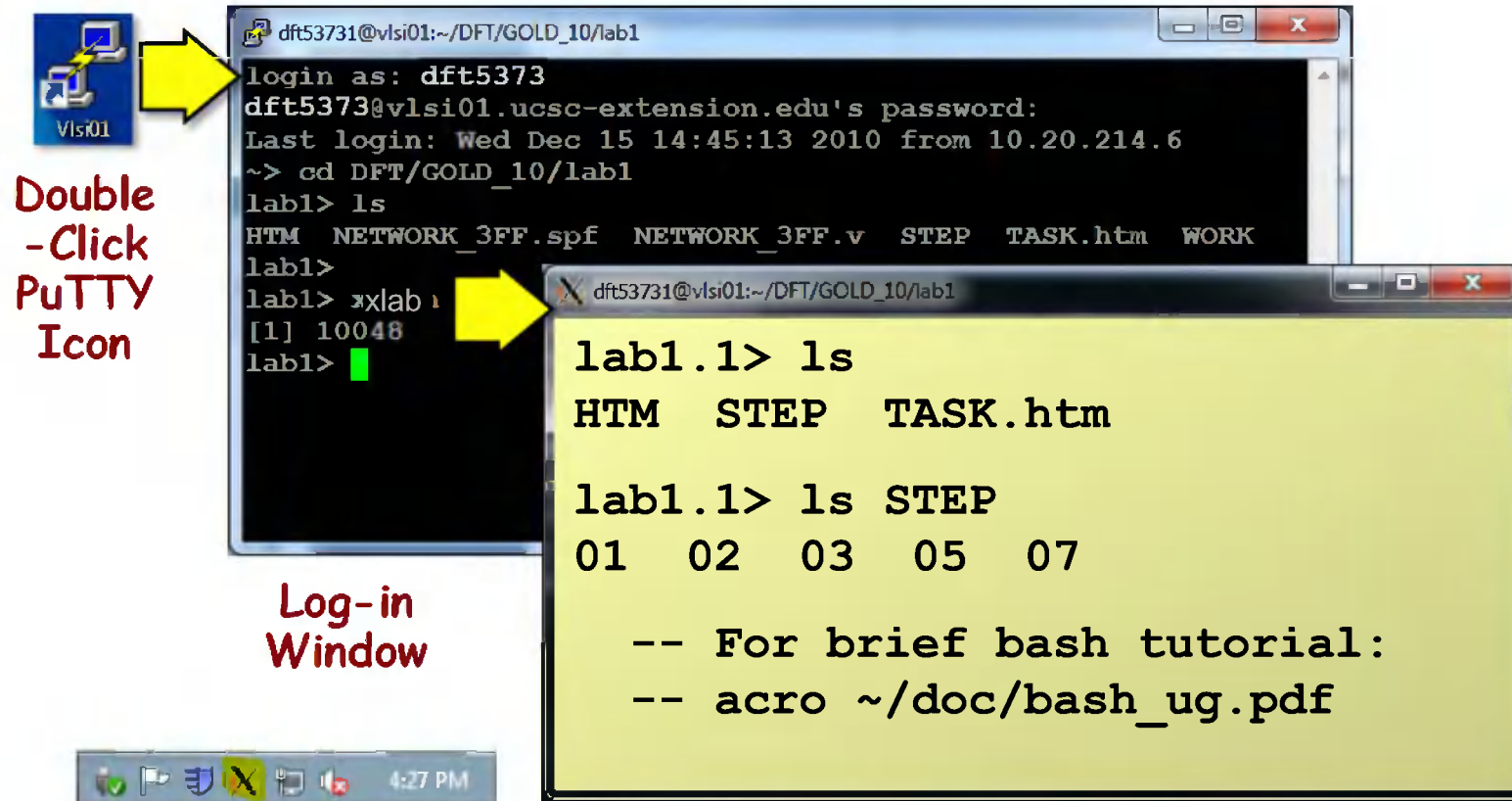
# Что такое формальная проверка? 0-6



- Средства FEC тщательно **сравнивают** две формы одного проекта .
- Код уровня RTL часто сравнивают со списком соединения уровня логических элементов .
- Не нужны векторы моделирования; проверяются **все** состояния .
- Все обнаруженные **несоответствия** идентифицируются для отладки .

# Доступ к учетной записи Linux

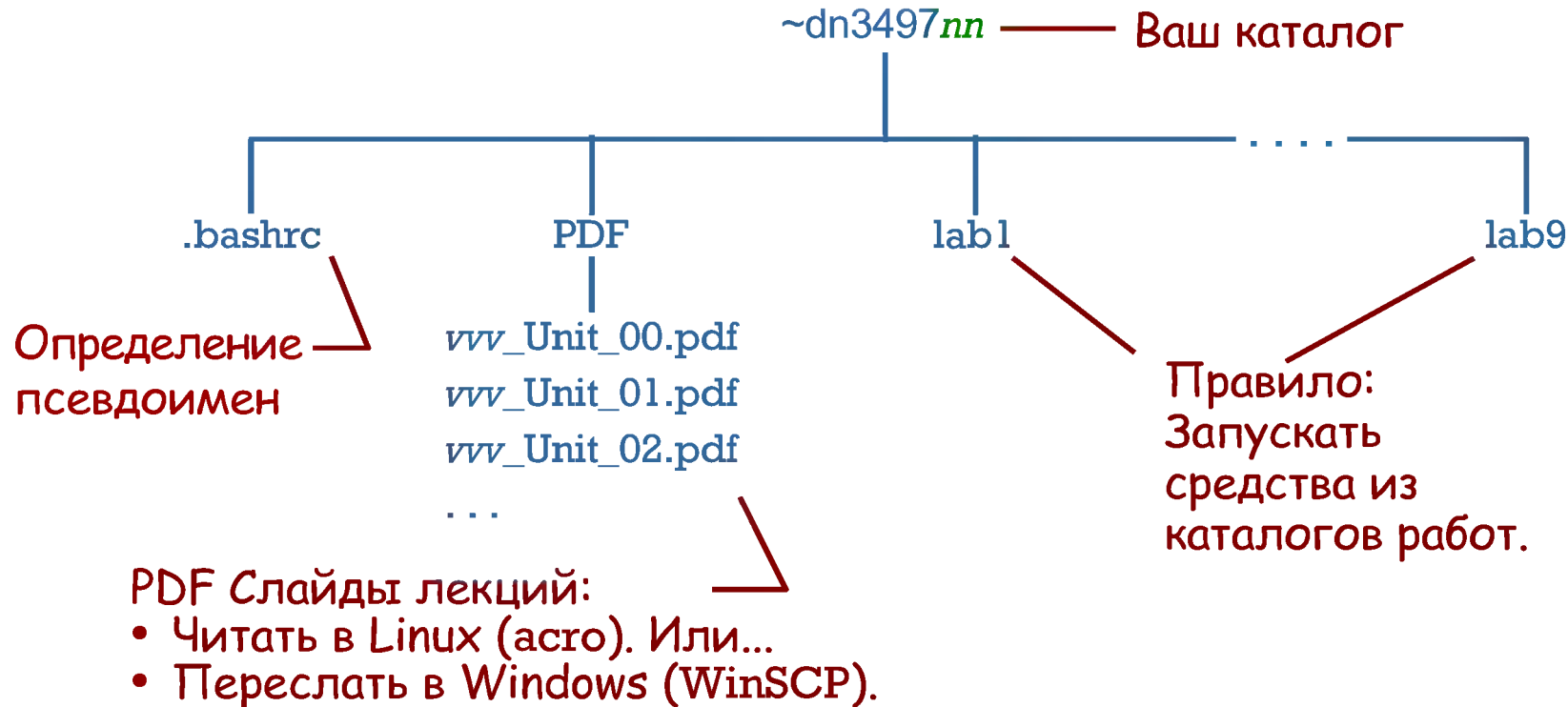
0-7



1. Выполняется ли X дисплей на эмуляторе **Xming**?
2. Откройте сессию **PuTTY** на удаленном сервере **vlsi02**. Войдите в систему.
3. Для открытия окна рабочего терминала введите псевдоимя **xlab**.

# Ваша учетная запись Linux (1/2)

0-8

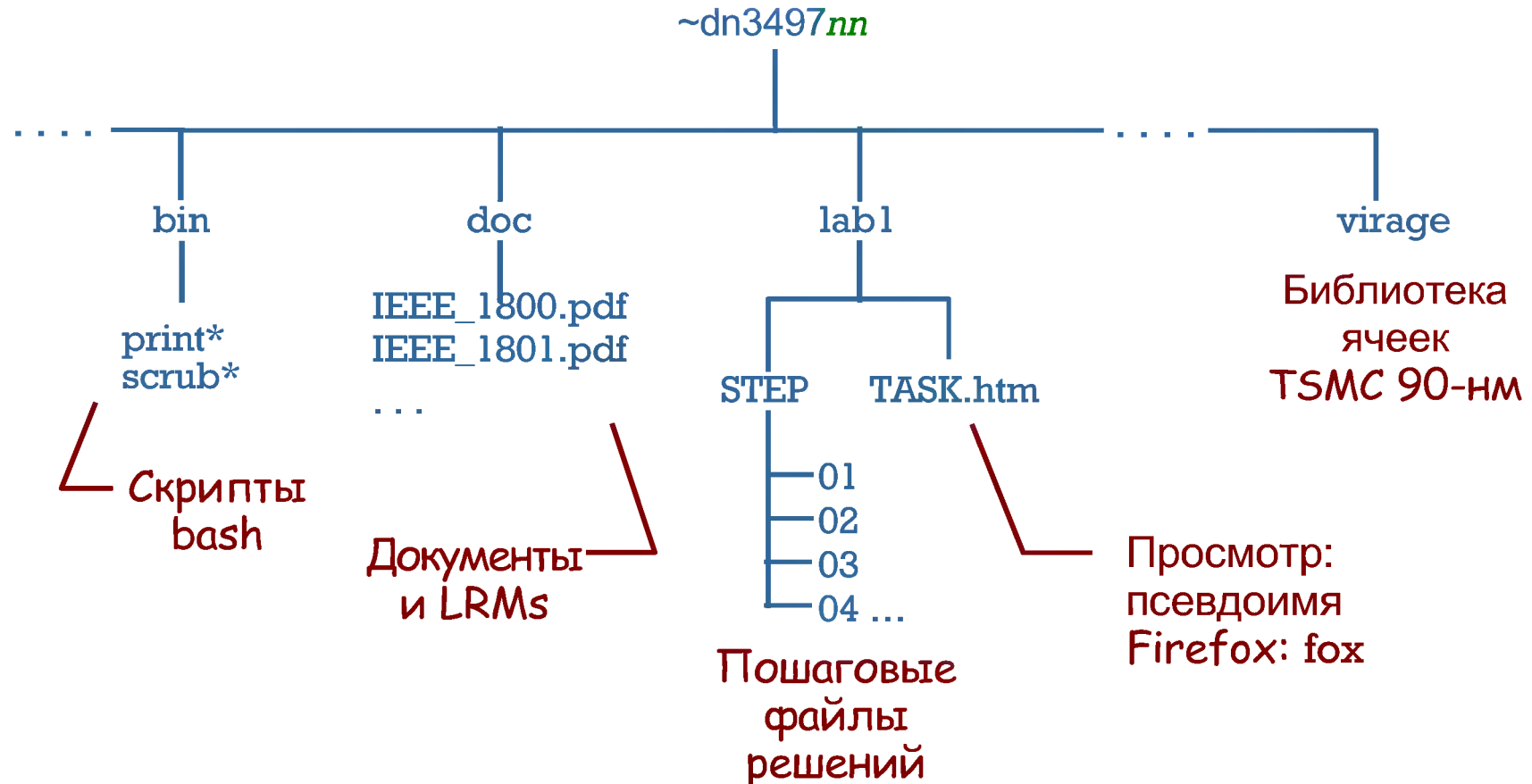


- В вашей учетной записи хранятся и материалы **лекций** и файлы **лабораторных работ**.
- WinSCP позволяет загрузить файлы Linux на Windows.
- Не изменяйте **пароль** учетной записи — иначе вы не получите обновления.



# Ваша учетная запись Linux (2/2)

0-9



- Каталог **лабораторной работы** содержит он-лайн указания и все файлы.
- Для получения пошаговых указаний просмотрите файл **TASK.htm**.
- На каждом **шаге** вы можете просмотреть файлы решений в каталоге **STEP**.

# Обозначения в коде

# 0-10

```
TALLY.sv

/* MATRIX TALLY MODULE
 * Inputs 3x4 array of bits.
 * Counts total number of 1s.
 */
typedef bit [1:3][1:4] MAT_t;
module TALLY(
  input  MAT_t MAT,
  output bit [3:0] COUNT
);
  always_comb
  begin:TOTAL
    COUNT = '0;
    for(int ROW = 1; ROW <= 3; ROW++)
      for(int COL=1; COL <= 4; COL++)
        COUNT += MAT[ROW][COL];
  end: TOTAL
endmodule: TALLY
```

Комментарий

Операторы деклараций

Выполняемые операторы

Условие

Выбор

- Язык чувствителен к регистру; ключевые слова должны быть в **нижнем регистре**.
- Для удобства восприятия в курсе для всех идентификаторов используется **верхний регистр**.

# Типичное задание лабораторной работы

# 0-11

Numbered Steps

Can Paste Commands into Tools

Click Middle on a Link to Open New Tab

Search: Ctrl-F

Tab Between Related Files

Open File: Ctrl-O

Resize Text: Ctrl±

Scroll Through Steps

Lab 4. Physical Level DFT - Mozilla Firefox

10 OPTIMIZE DFT PLACEMENT

During the placement phase of physical design, every standard cell is optimally placed on silicon. Its frame is positioned at a definite point in the x-y plane. In particular, the exact (x, y) coordinates of each scan cell are known. The Manhattan distance between any two scan cells is thus determined.

This is the earliest point in the physical-design flow where we can successfully perform the reoptimization step known as *scan reordering*. The tool-specific option that enables it is

```
place_opt -optimize_dft
```

Tool Insight Stand-Alone Version

This tool also supports a stand-alone `optimize_dft` command. It's intended for advanced users wanting more manual control, especially for designs with more than 20 scan chains.

For details on this alternative flow, consult the man pages for `set_optimize_dft_options` and `optimize_dft`.

Make a copy of the baseline TCL script. Name it `REORDER.tcl`. You'll need to edit just two command-line options.

Scroll to the floorplanning phase. Locate the command below. Delete the `-ignore_scan` option. The command should now be

```
create_fp_placement
```

This tells ICC not to ignore `.scandef` data, but to initially place leaf cells with some regard for optimal scan ordering.

Scroll to the placement phase. Modify `place_opt` by inserting the option below. This *enables* the scan reordering algorithm during the placement process.

```
$ PLACEMENT PHASE
$ Enable scan reordering:
place_opt -optimize_dft
report_qor
```

Before we run this modified script, let's briefly take a look at the underlying algorithm. Scan reordering finds an optimal repositioning of all scan flops such that total chain length — the sum of *Manhattan distances* between successive cells — is minimized. This reduces the chip's overall wire length.

The algorithm is constrained by the `scandef` file. Any cells marked `FLOATING` can be repositioned anywhere inside the core. `ORDERED` cells must be repositioned as an unbroken segment.

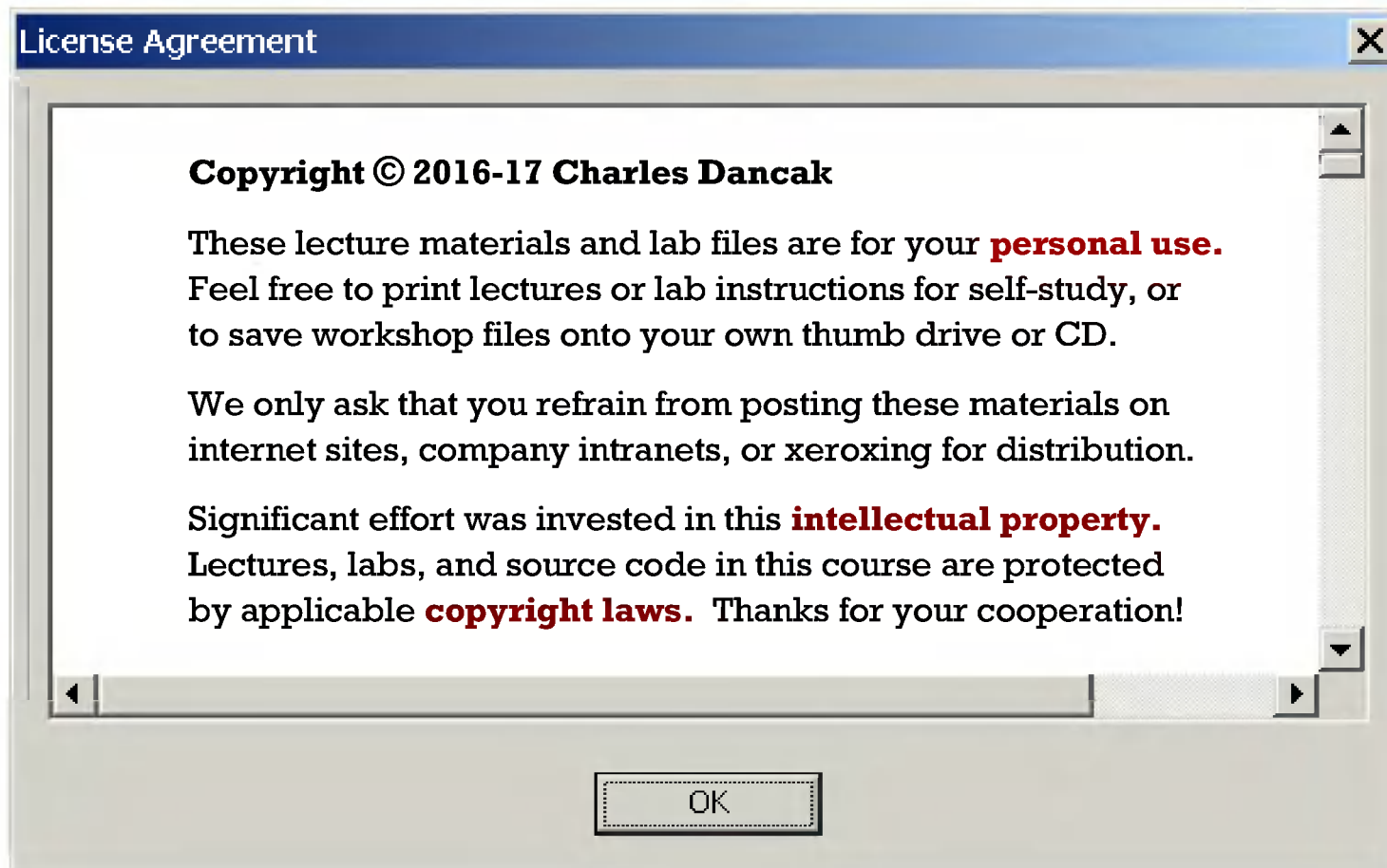
Find: Done

Для просмотра в Linux введите в каталоге `lab1`, введите: `fox TASK.htm`

# Выбор лабораторных работ 0-12

Доступные работы				
Работа	Уровень	Описание работы	Проект работы	Средства
1.1	Базовый	Code and lint a split-counter.	SPLIT_CNTR	lint
1.2	Базовый	Synthesize a split-counter.	SPLIT_CNTR	DC
1.3	Базовый	Debug split-counter functionality.	SPLIT_CNTR	VCS
2.0	Усложн.	Formal Equivalence Checking	NETWORK_3FF	Frmly
4.0	Усложн	Physical Design Tutorial	RISC_CHIP	ICC

- Каждый должен выполнить базовые работы 1.1—1.3.
- Если остается возможность, также выполните работы 2 и/или 4, исходя из ваших интересов.
- Работа выполняется где вам **удобно**; выполнение занимает работы от 3 до 8 часов.
- Для получения **оптимального результата**, тщательно выполняйте инструкции.



- Материалы предназначены только для **зарегистрированных студентов**.
- Не следует их размещать на корпоративных или любых других сайтах.

Приглашение  
оболочки  
Linux

```
> acro file.pdf //Adobe Reader, Linux
> fox TASK.htm //Lab instructions.
> lint file.pdf //Syntax-check code.
> print L213 file.pdf //Print code.
> xdv //Launch Design Vision GUI.
> xsim file.pdf //VCS compile simv.
> xdve //VCS Discovery Visual Env.
> xicc //Launch IC Compiler.
> xlab //Open 4 shell windows.
> type xsim //Display any alias.
```

Средства  
Synopsys EDA

- Использование **псевдоимен** гарантирует использование соответствующей версии средств EDA.
- Команда Linux **type** позволяет просмотреть синтаксис имен-псевдонимов.
- Все псевдоимена определены в файле настроек **~/.bashrc**.

# Ресурсы и рабочие места

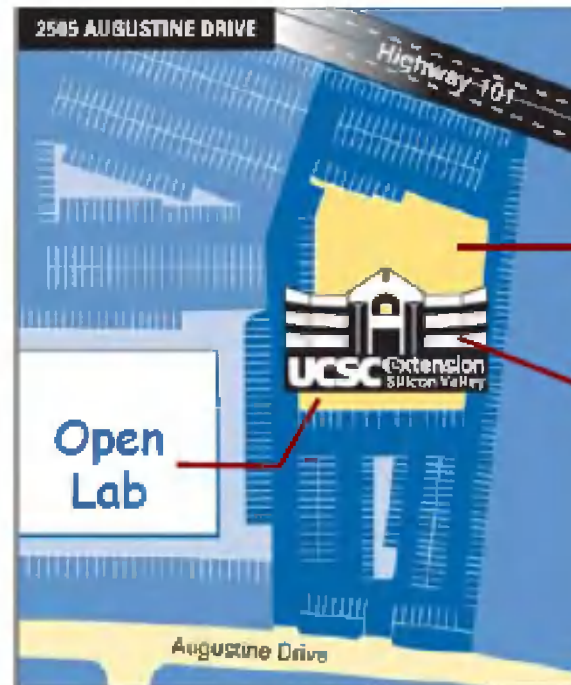
0-15

## Сеть UCSC:

SSID: **student**

Login: **ucsc-guest**

Pswd: **EnrollNow!**



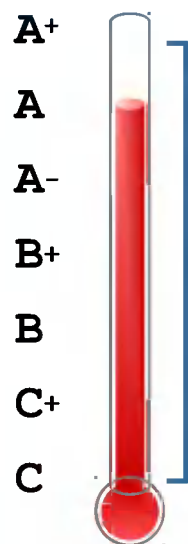
- Когда открыто здание, **Open Lab** всегда доступна .
- Можно использовать ПК с доступом к серверу **vlsi02**.
- Беспроводная сеть имеет **ограниченный** радиус действия.
- **Найдите меру** при печати—или приносите свою бумагу.

---

# Схема аттестации

0-16

Ваша оценка основана на:



## Активность в аудитории:

- Выполнения лабораторных работ, в аудитории или Open Lab.
- Вклад в работу в аудитории; работа в команде.
- Регулярное посещение—кроме командировок, выставок и т.д.

- Пожалуйста, следуйте элементарным правилам вежливости, выключайте свои **телефоны** во время занятий.
- Пожалуйста, не **печатайте** на клавиатуре во время лекций.
- Разный уровень подготовки—старайтесь избегать **примитивных вопросов**.

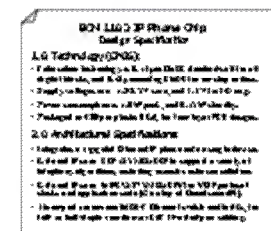


---

# Developing the Nanometer ASIC

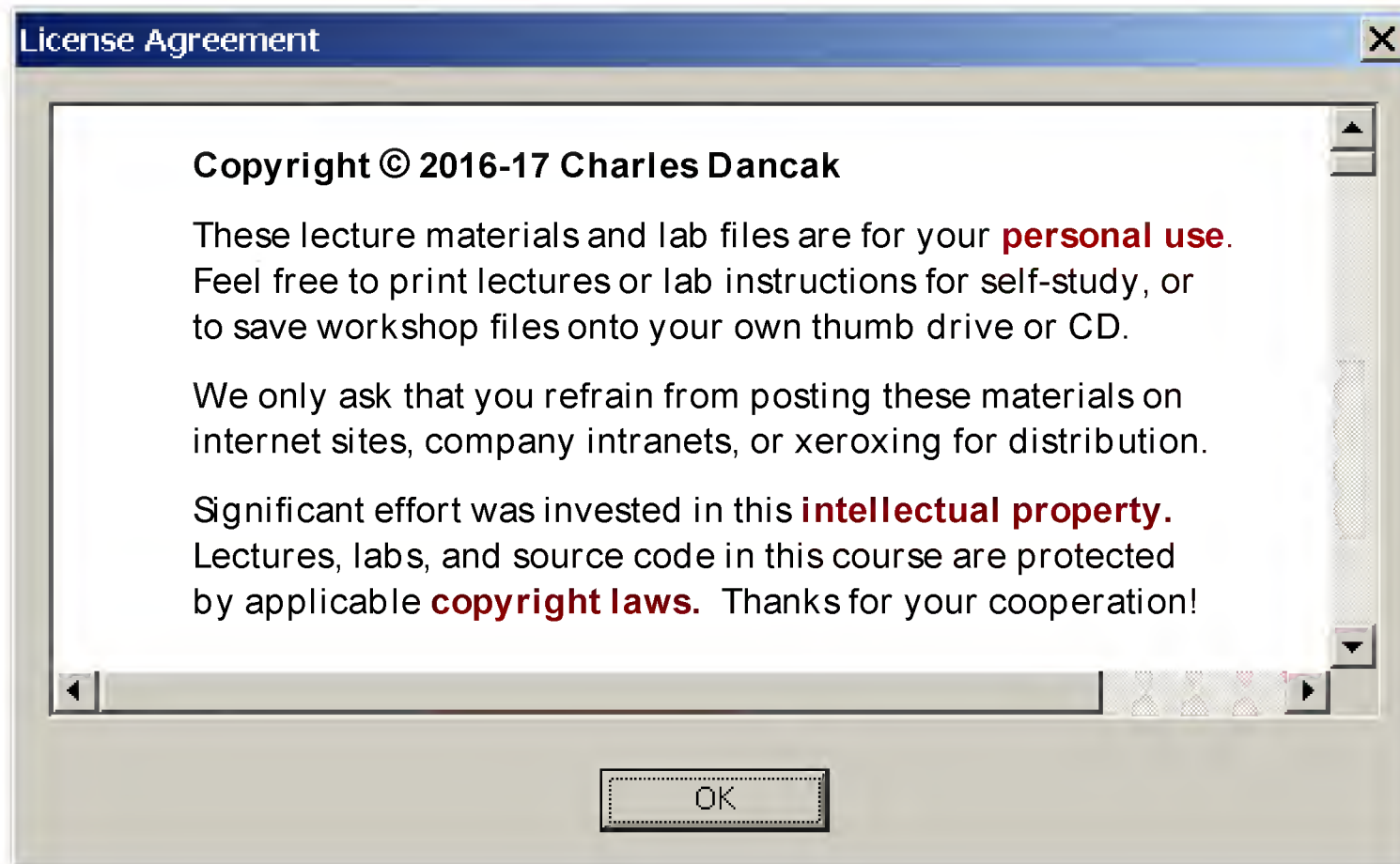


## 1. Стадия спецификации



# Авторское право

1-2

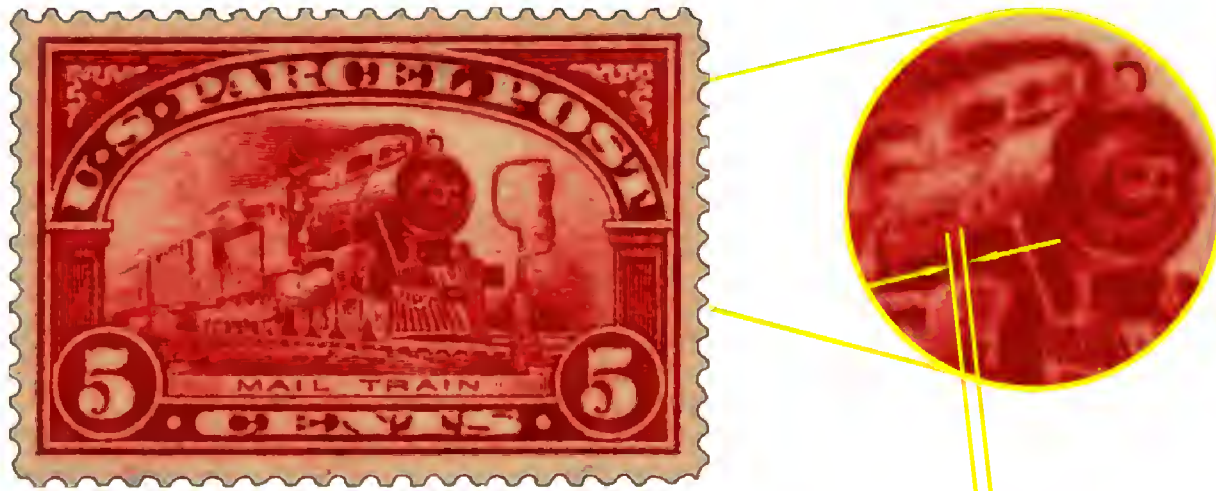


Чтобы организовать семинары на своем предприятии, свяжитесь с Quantum Consulting по тел: **650/380-9121**

## **Основные концепции ASIC**

- Аналогия с почтовой маркой
- Первые интегральные схемы
- Закон Мура
- Управление сложностью

# Аналогия: Почтовая марка (1/2) 1-4

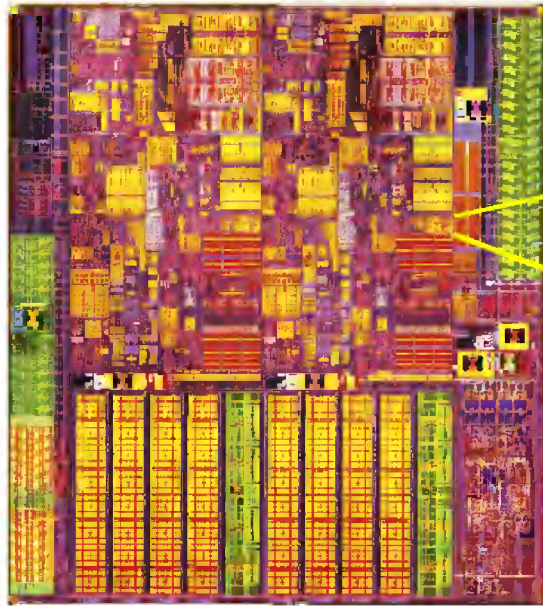


Старинная почтовая марка  
(Штриховая гравюра)

Разрешение: 5 mil  
(0.127 миллиметра)

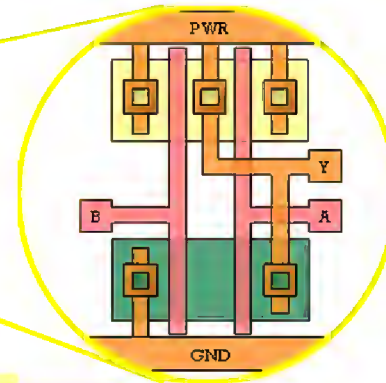
- Бескорпусная **микросхема ASIC** подобна почтовой марке.
- Она представляет собой прямоугольную поверхность с мелкими выгравированными **элементами**.
- Каждый цвет печатается **отдельной** металлической печатной формой.

# Аналогия: Почтовая марка (2/2) 1-5



**Кристалл процессора  
Intel 32-nm  
Westmere**

✦ Не в масштабе:  
Кристалл видим  
невооруженным глазом -  
но для вставки использован  
**SEM** (сканирующий  
электронный микроскоп)



**Стандартная  
ячейка  
элемента NAND  
(4 транзистора)**

**Разрешение: 32 nm  
(0.000032 миллиметра)**

(Фотолитографический процесс)

- Детали кристалла при технологической норме 32-nm мельче в **4,000×** раз.
- Для «печати» всех слоев нужно **40–50 фотошаблонов (masks)**.

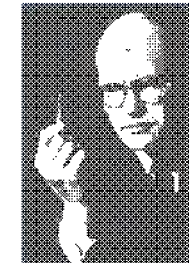
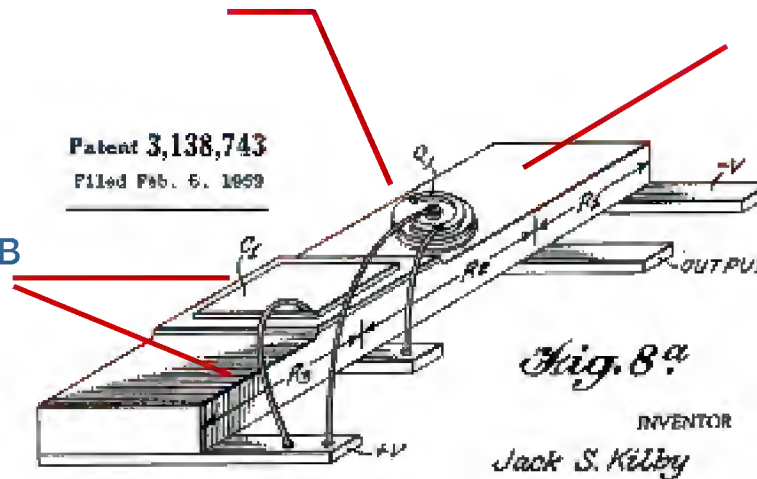
# Первые интегральные схемы

1-6

Один транзистор

Полупроводниковая подложка

Несколько резисторов  
и конденсаторов



Jack Kilby  
(Джек Килби), TI ('58)

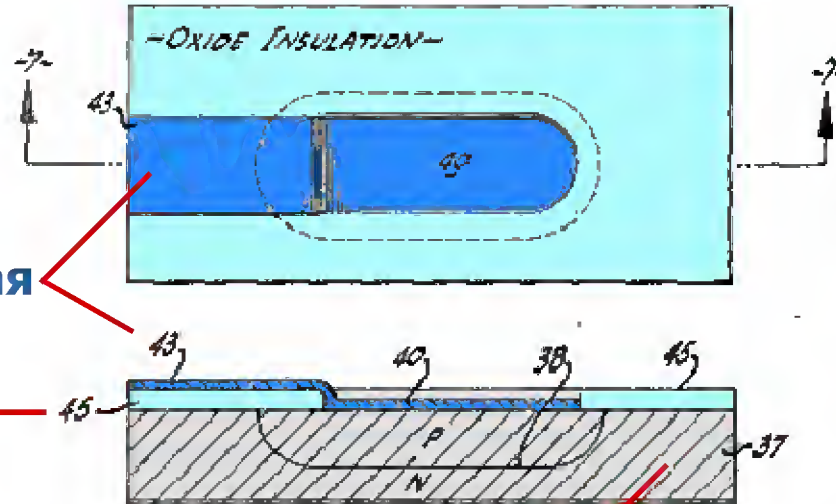
...миниатюрная электронная схема, ... в которой компоненты интегрированы в объем полупроводникового материала...

- Радикальный переход от **дискретных** компонентов на макетной плате.
- Генератор **интегрирован** в монолитный блок.
- Компоненты сформированы в нем электрически и физически.

# От проволок к полоскам

1-7

Patent 2,981,877  
Filed July 30, 1959



Металлическая  
плоска (M1)

Оксидный слой  
(толщина 1мкм)

Кремниевая  
подложка



Bob Noyce (Боб Нойс),  
Intel ('59)

...Представьте себе электронику в виде  
твёрдого блока... Без соединительных проводов  
*[Английский специалист по радарам, 1952]*

- Золотые **проволочки** Kilby заменены алюминиевыми **полосками**.
- Металл осаждается на оксидную пленку и **прилипает** к ней.
- Изолирующая пленка ( $\text{SiO}_2$ ) выращивается путем **окисления** кремния.

# Больше слоев металла

1-8

Верхний  
слой  
металла  
(M6)



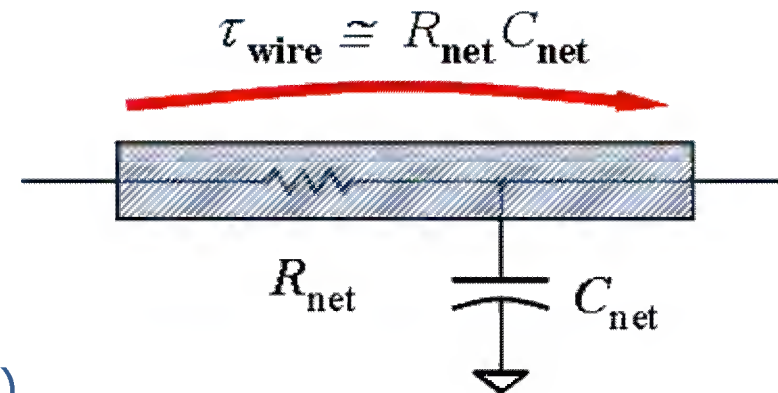
Нижний  
слой  
металла  
(M1)

**1997 Кристалл IBM**

(SEM шестислойной медной металлизации)

- Через несколько десятилетий использование шести слоев металла стало общепринятой практикой.
- Соседние слои **изолируются** друг от друга путем осаждения оксидных слоев.
- Вместо Al используется Cu с малым сопротивлением, что повышает **быстродействие**.

**Задержка распространения по  
металлической линии**







# Управление сложностью

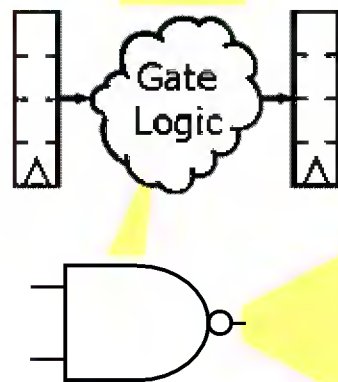
1-10



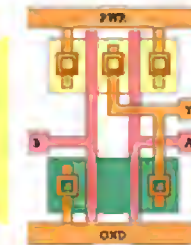
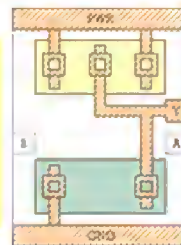
← **Ключевой принцип**  
Сложность SOC (Системы на кристалле) управляется путем работы на более высоких уровнях абстракции и скрывания деталей низкого уровня.



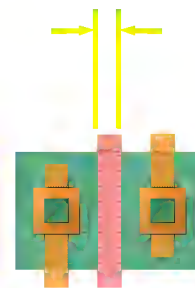
Отдельный логические элемент



Стандартная ячейка



$l = 32$  нм  
( $\sim 10^2$  атомов)



На топологическом уровне размеры действительно имеют **нанометровый** масштаб.

## ***n*MOS(нМОП)-транзистор:**

- Легированные области
- Поликремниевый затвор
- Аналогия с выключателем
- Работа включателя
- Библиотека ячеек
- Скорость переключения

# Проводящие области

# 1-12

## Замечание:

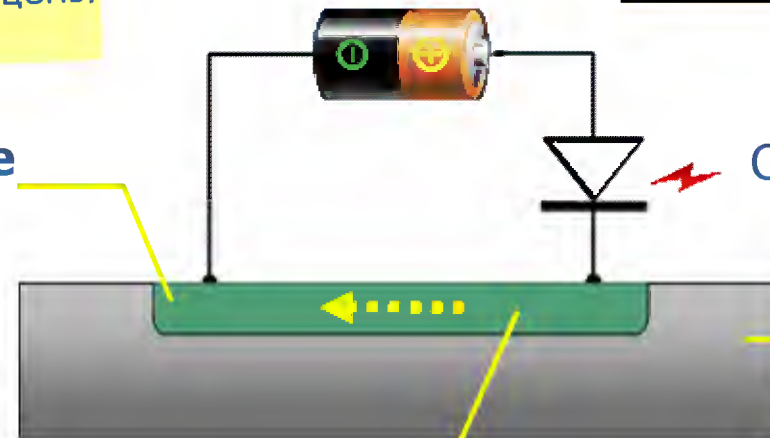
Ток протекает только через **замкнутую** цепь.

## Ключевой принцип

Добавление легирующих атомов к Si делает его хорошим проводником.

**Легируемые области**  
(*n*-типа)

**Батарея (1 В)**



Свет светодиода

**Подложка:**  
Чистый кремний –  
только полупроводник.

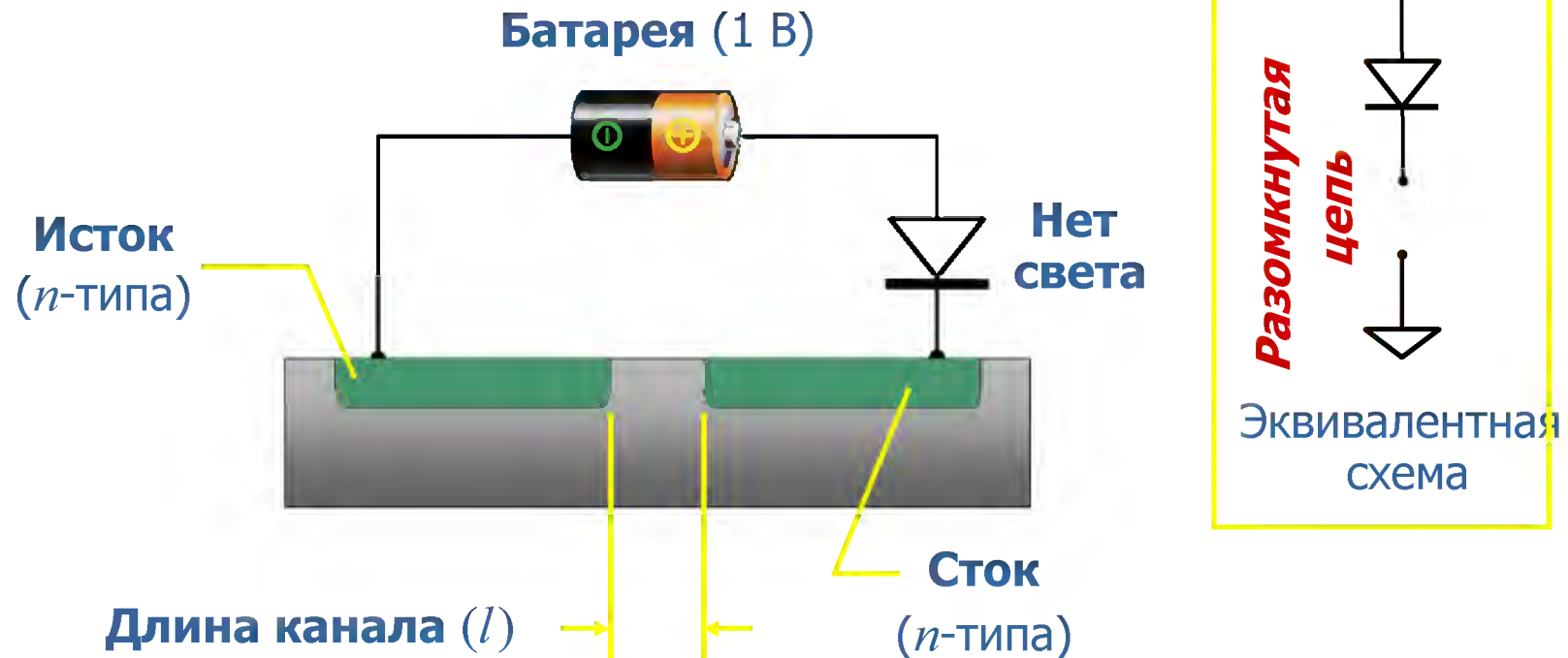
Толщина кристалла  
показана не в масштабе

**Подвижные электроны (*n*):**  
Обеспечивают прохождение тока.

- Чистый кремний, **полупроводник**, плохо проводит ток.
- Добавление **легирующих** атомов (P) обеспечивает появление подвижных **электронов**.
- Сильно *n*-легируемые области обеспечивают прохождение тока  $V/R$ .

# Участок между проводящими областями

1-13

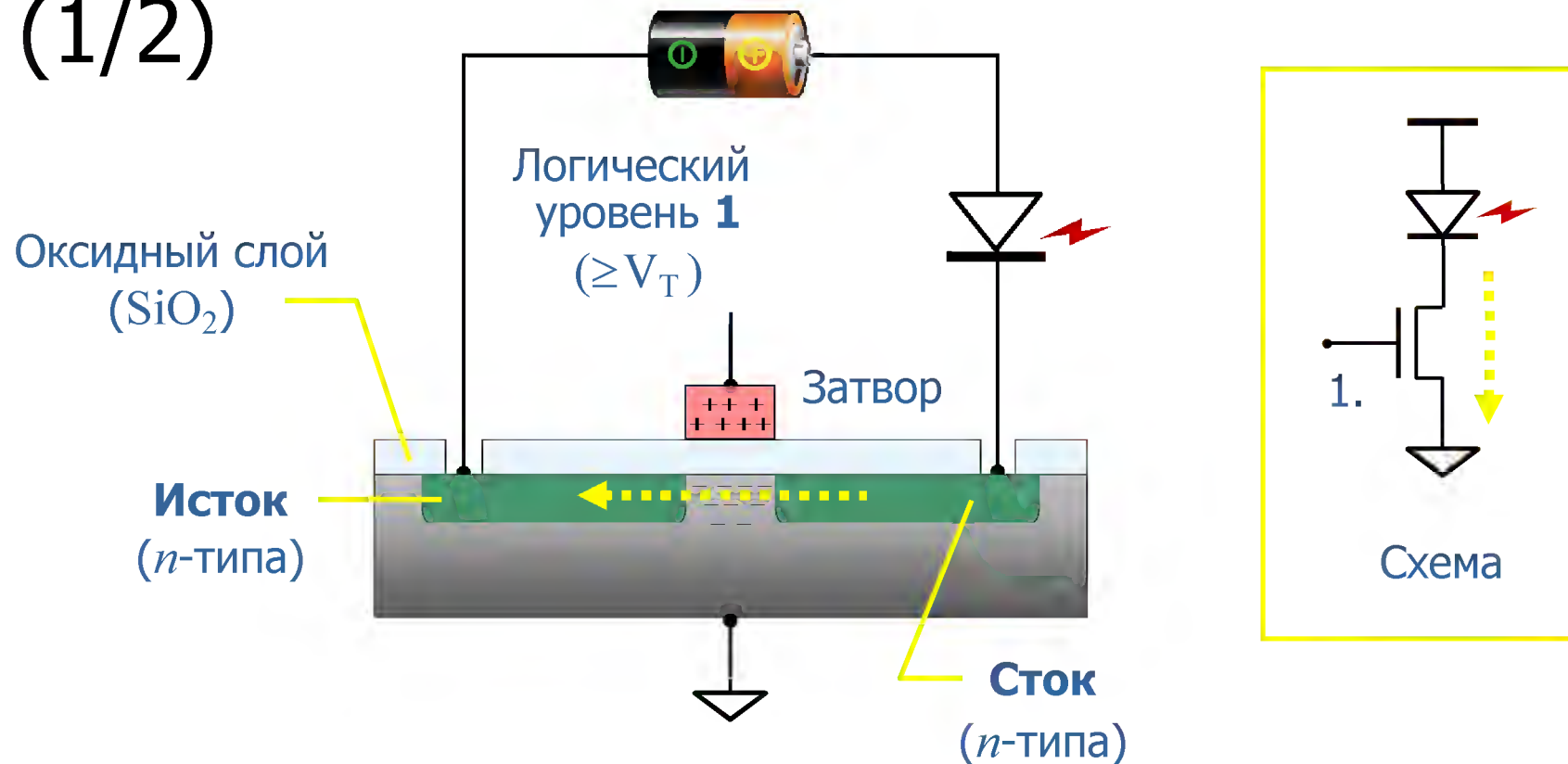


- Нелегированный **канал** разделяет области истока и стока.
- Электрическая цепь разорвана, ток не протекает.
- Пока мы не **внесем** подвижные электроны в канал.

# Маленький выключатель

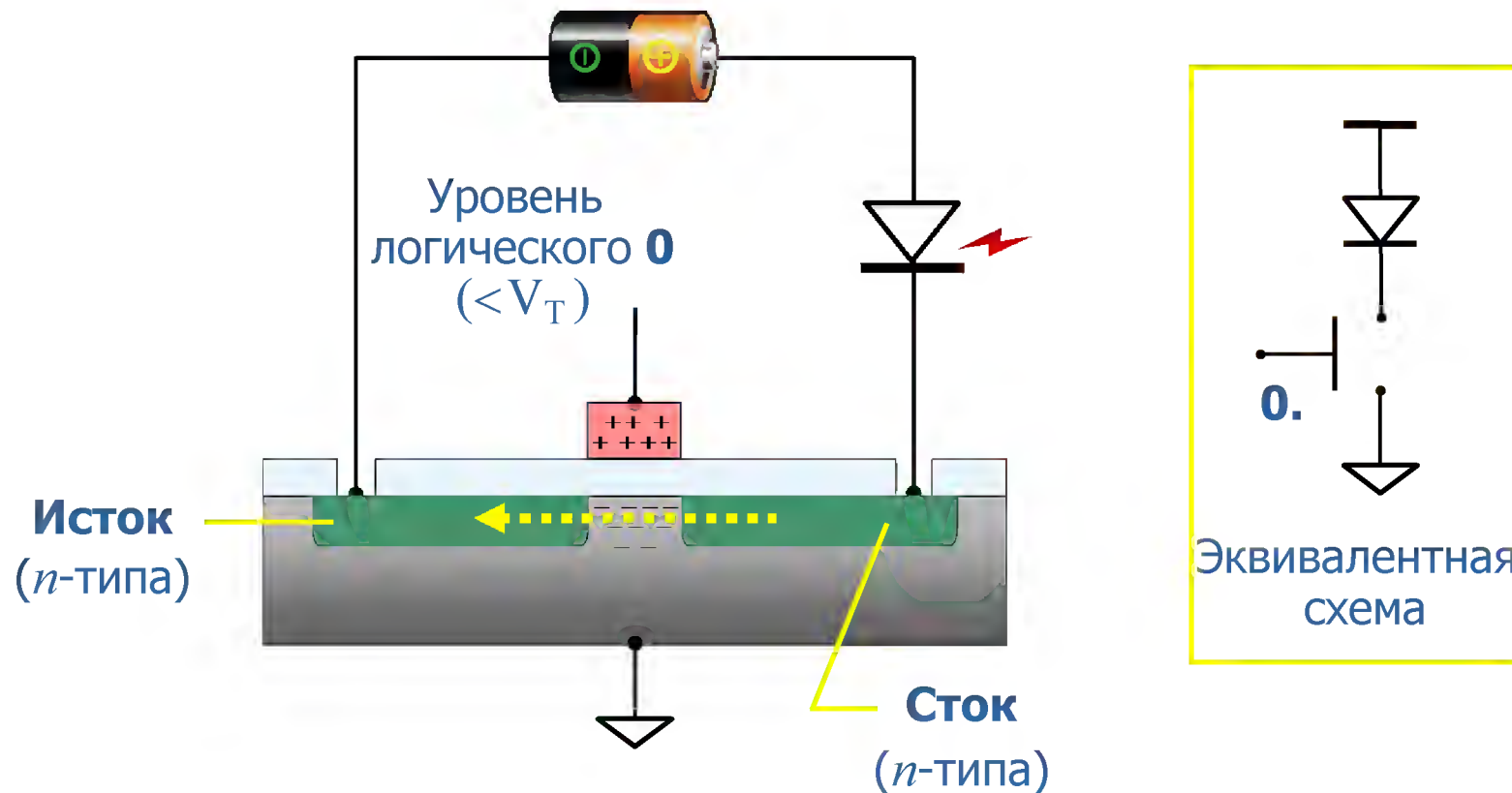
(1/2)

1-14



- Поликремниевый **электрод затвора** осаждается на оксид.
- **Конденсатор** ( $C_{ox}$ ) формируется затвором, изолятором, подложкой.
- Когда он заряжен, подвижные электроны **появляются** в области канала.

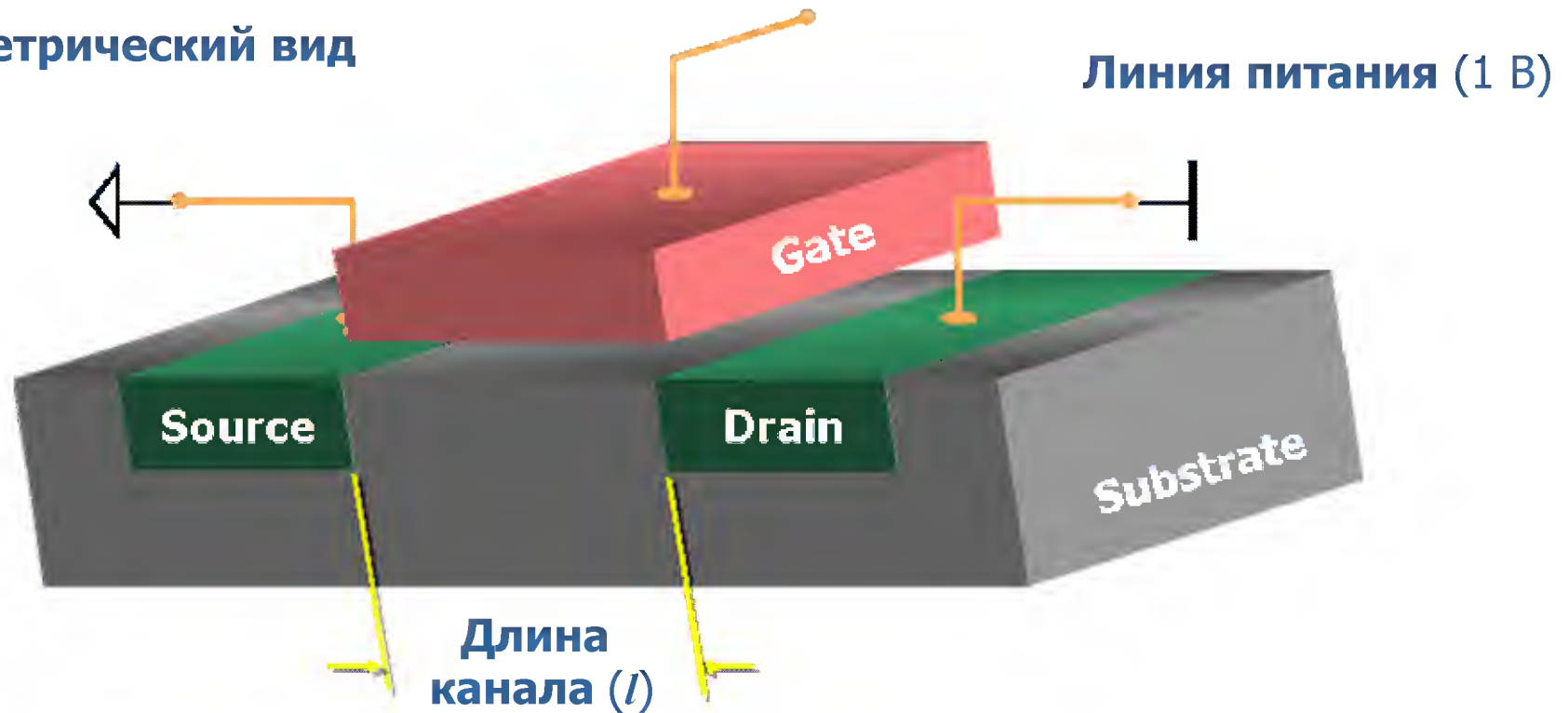
# Маленький выключатель (2/2) 1-15



- Когда разряжен, подвижные электроны **отсутствуют** в канале.
- Ток исток-сток не протекает; транзистор **закрыт**.
- Сокращение MOS: **m**etal/**p**oly-**o**xide-**s**emiconductor; МОП: металл/поли-окисел-полупроводник.

# *n*MOS(нМОП)-транзистор (1/2) 1-16

Изометрический вид

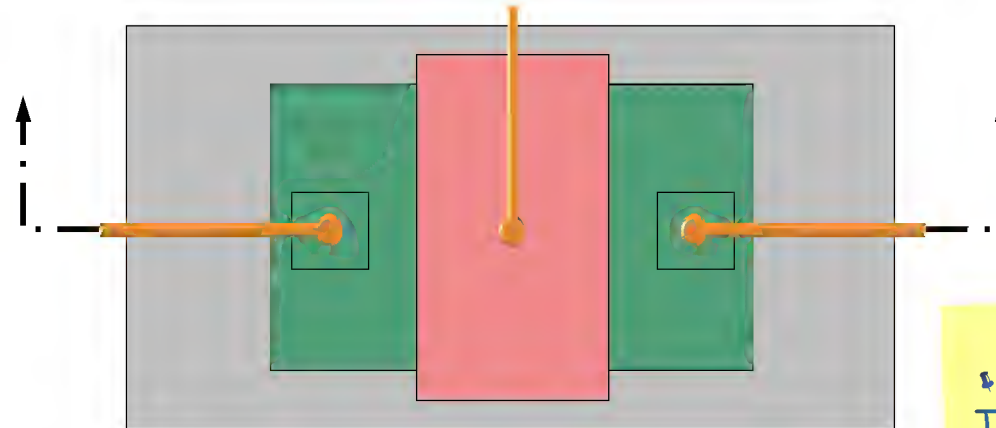


- МОП транзистор в цифровой схемотехнике ведет себя подобно **выключателю**.
- При подаче 1 он **замкнут**, при 0 - **разомкнут**.
- Оксидный изолятор ( $\text{SiO}_2$ ) для простоты не показан.

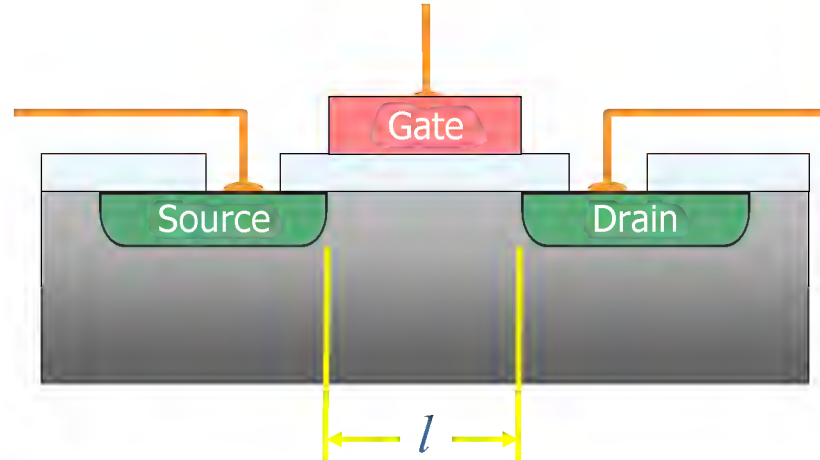


# nMOS(нМОП)-транзистор (2/2) 1-17

Вид сверху



Сечение

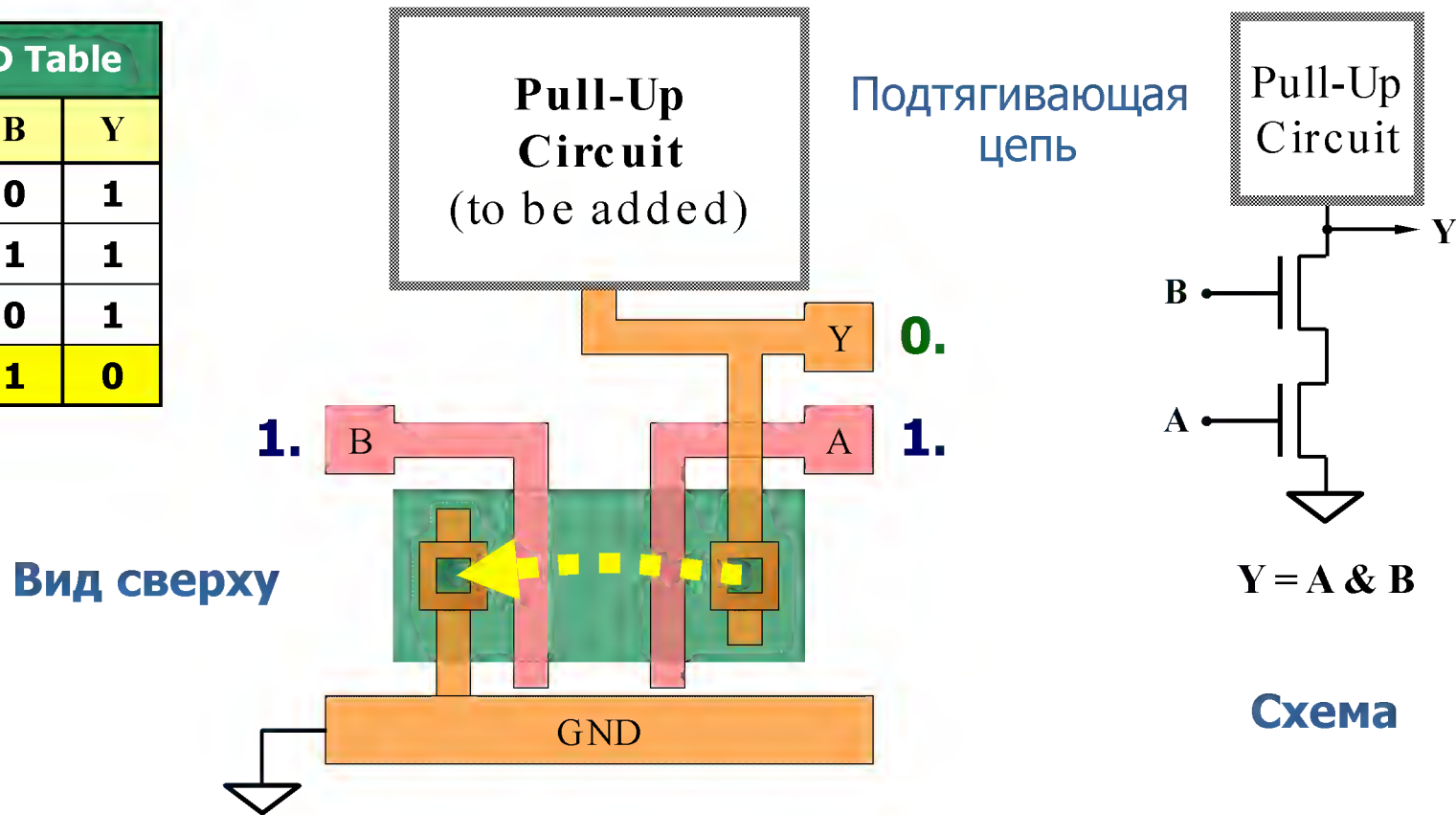


✦ Масштаб:  
Транзистор показан в  $\times 10^6$  большим, чем он есть на самом деле. В действительности подложка намного толще.

Длина  $l$  является классической **технологической нормой**, важной характеристикой технологического процесса.

# Работа элемента И-НЕ (NAND) 1-18

NAND Table		
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

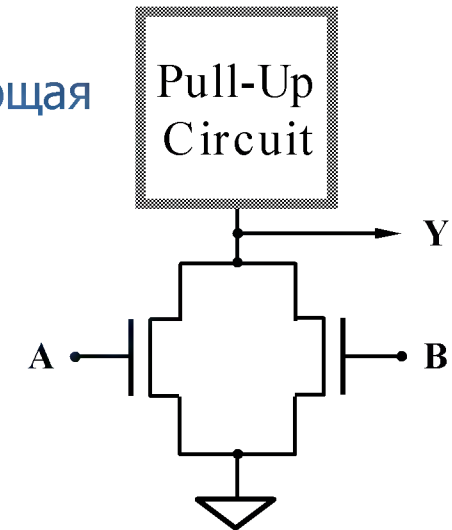
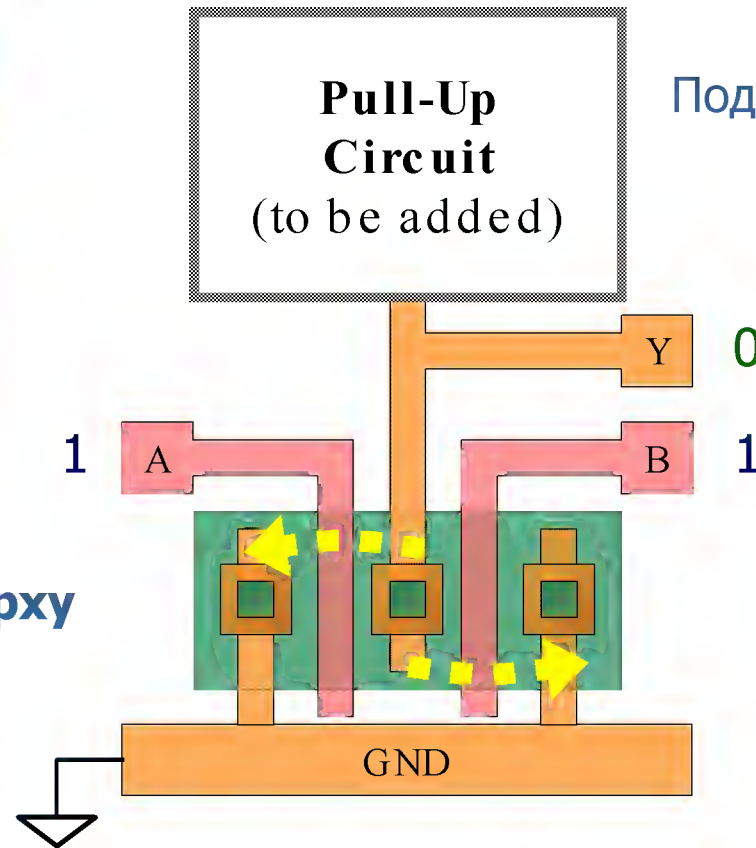


- Два последовательно соединенных транзистора реализуют операцию И-НЕ.
- Когда A и B равны 1, выход Y связан с 0.
- В **других** случаях Y подтягивается к 1 (не показано).

# Работа элемента ИЛИ-НЕ (NOR) 1-19

NOR Table		
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Вид сверху



$$Y = A | B$$

Схема

- Два **параллельно** соединенных транзистора реализуют операцию ИЛИ-НЕ.
- Когда **A** и **B** (или оба) равны 1, выход **Y** связан с 0.
- В **других** случаях **Y** подтягивается к 1 (не показано).

# Упражнение 1.1

1-20

Идентифицируйте следующую логическую функцию

Выход Y есть функция трех входов:

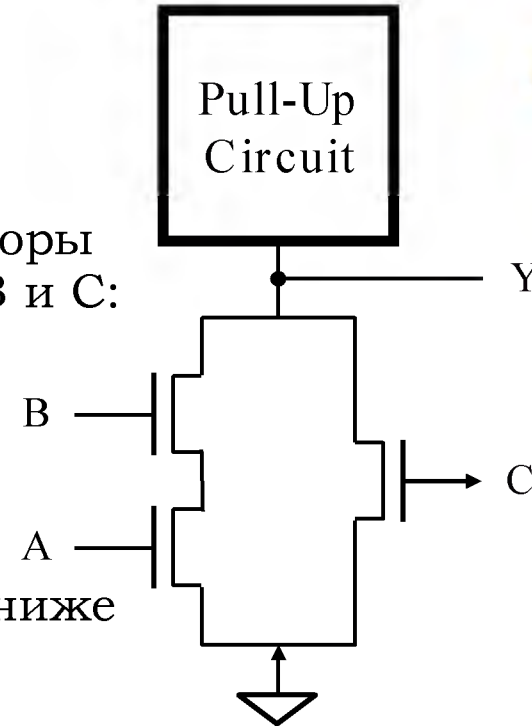
$$Y = F(A, B, C)$$

Проанализируйте параллельно и последовательно соединенные транзисторы и запишите выражение для Y через A, B и C:

Y =

Вы можете использовать приведенные ниже операторы SystemVerilog:

- P & Q    Операция И операндов P и Q.
- P | Q    Операция ИЛИ операндов P и Q.
- ~P      Операция НЕ операнда P.
- (...)    Используется для группирования.



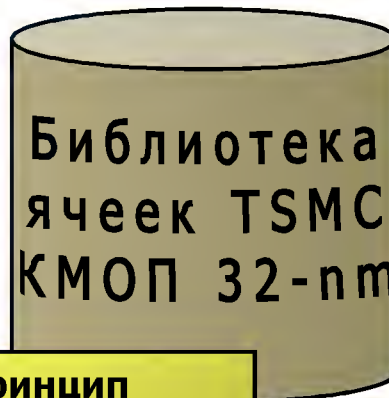
# Библиотека ячеек

# 1-21

## Элемент И-НЕ:

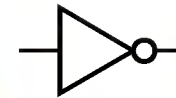
- Функция:  $\sim(A \& B)$
- Площадь:  $0.4 \mu\text{m}^2$
- Мощность (пВт)
- Задержка (пс)

tsmc\_32nm.lib

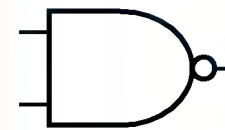
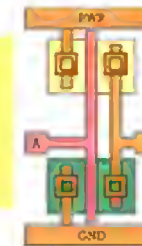


### ← Ключевой принцип

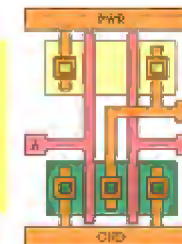
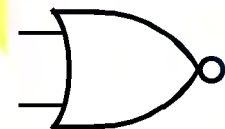
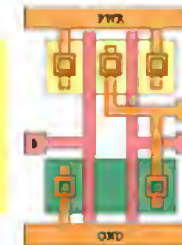
Характеристики ячейки очень сильно зависят от технологической нормы, особенностей процесса, изготовителя.



Логическая область

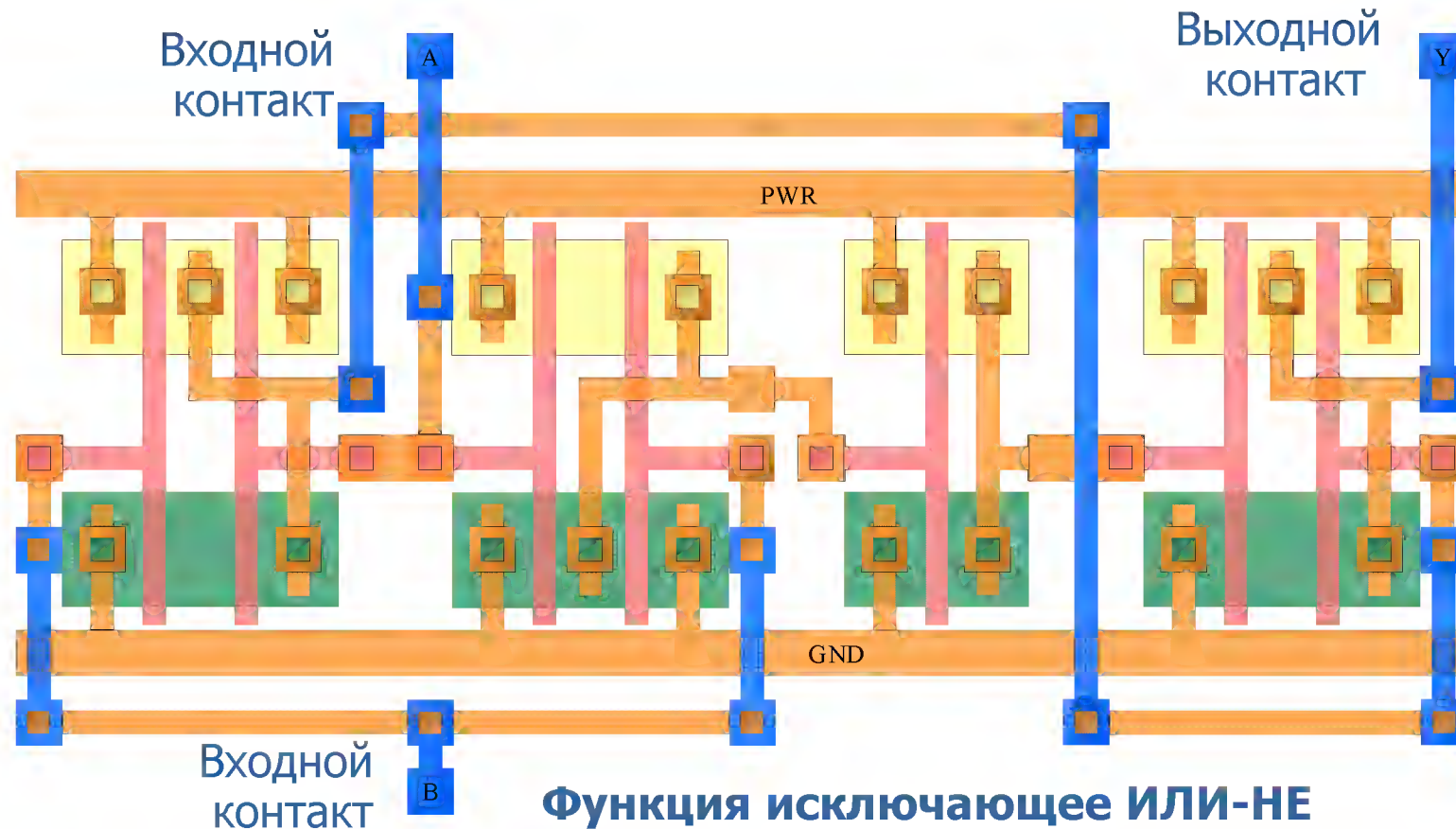


Физическая область



- В файле содержатся ячейки для сотен полезных **логических функций**.
- Включая логические элементы, триггеры, защелки, мультиплексоры и т.д.
- Приведены **характеристики** каждой ячейки:  
Площадь, быстродействие, потребляемая мощность.

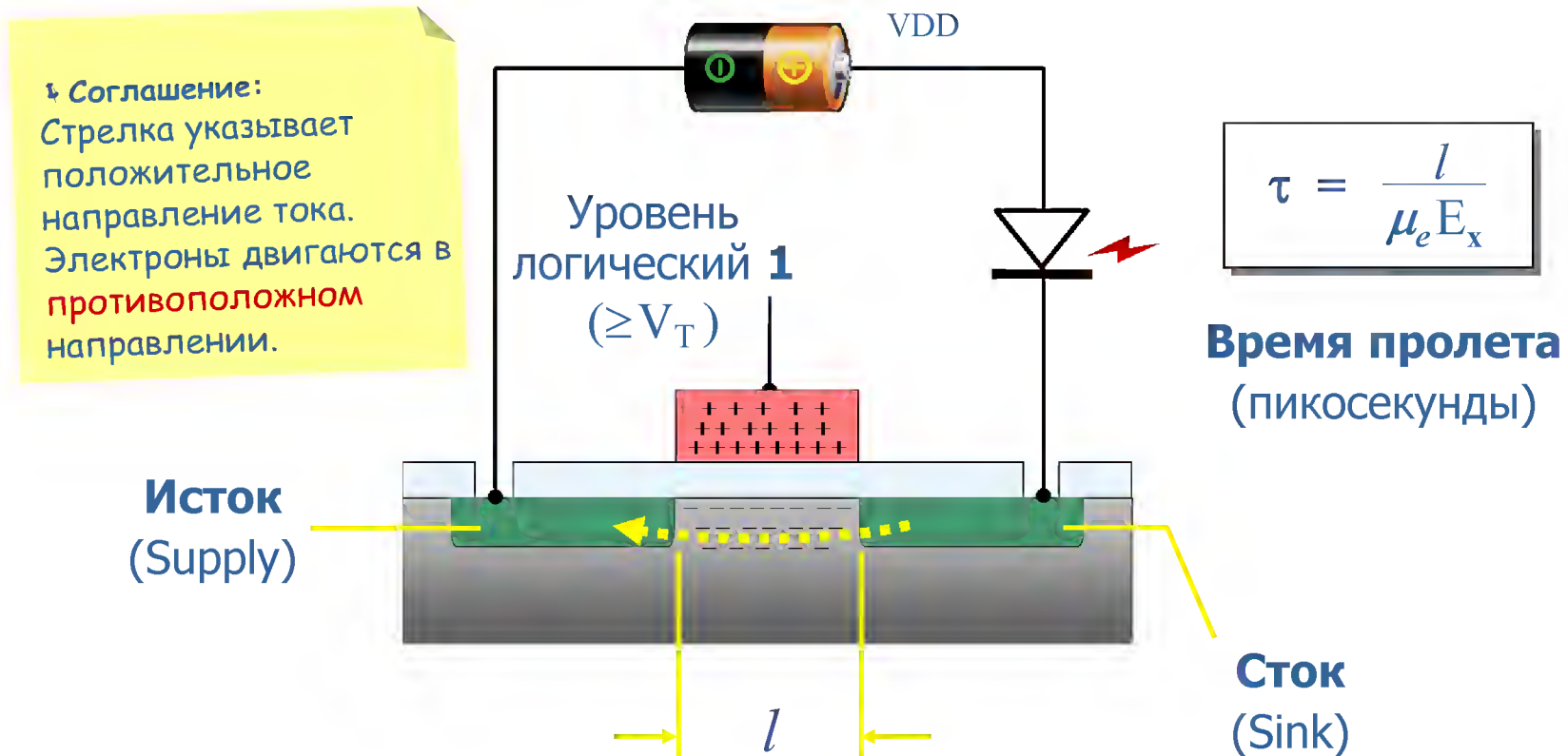
# Трассировка соединений стандартных ячеек 1-22



- Одинаковые по высоте, примыкающие стандартные ячейки формируют ряды.
- Трассировщик проектирует соединения контакт-контакт в металлическом слое.
- Показанная логическая функция сравнивает два бита на равенство.

# Скорость переключения

1-23



- Транзисторы определяют основное быстродействие технологии.
- Ограничивается **временем пролета**  $\tau$  всей длины канала  $l$ .
- Таким образом, уменьшение  $l$  повышает **быстродействие**.

---

# Основные концепции:

# 1-24

1. Кремний - **полупроводник**—он плохо проводит ток.
2. **Легирующие атомы** формируют проводящие области исток-сток.
3. МОП транзистор формируется истоком и стоком, которые разделены **каналом**, с поликремниевым затвором.
4. Конденсатор метал-оксид-полупроводник (МОП) заряжается и разряжается при приложении уровня логической **1** или **0**.
5. В результате МОП-транзистор действует подобно маленькому выключателю.
6. **Скорость** переключения выше для коротких каналов *l*.
7. Два последовательно соединенные транзисторы реализуют функцию И; два параллельно соединённые - функцию ИЛИ.
8. **Библиотека ячеек** содержит ячейки для примерно 700 логических функций.



## **Стадия спецификации:**

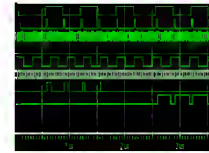
- Описание изделия
- Бизнес-возможности
- Отличие изделия
- Таблица маркетинговых особенностей
- Оценка реализуемости микросхемы

# Шесть основных этапов

# 1-26

## 1. Спецификация:

Описать замысел микросхемы на бумаге.



## 1. Спецификация

## 2. Проектирование

## 2. Проектирование:

Реализация замысла проекта путем создания его описания на уровне регистровой передачи (RTL).

```
1 /* RTL Verilog Code
2 * Four-bit serialiser,
3 * with loader control.
4 */
5 module SERIAL4
6 output wire TX, //Serial.
7 input wire [3:0] DMS_DATA,
8 input wire LOAD_SET, HXZ, CLK
9 );
10 reg [3:0] FF; //Four flops.
11
12 always @ (posedge CLK)
13 if (LOAD_SET)
14 FF <= DMS_DATA; //Load data.
15 else
16 FF <= {1'b1,FF[3:1]}; //Shift.
17
18 assign TX = FF[0]!HXZ; //OR.
19
20 endmodule
```

## 3. Верификация:

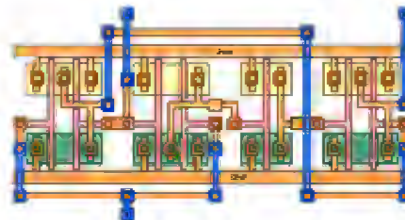
Подтвердить, что RTL-код соответствует первоначальным намерениям.

## 3. Верификация

## 4. Имплементация

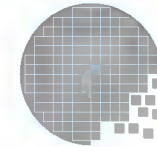
## 4. Имплементация:

Синтезировать RTL-код в стандартные 32-нм ячейки. Размещение и трассировка.



## 5. Производство:

Изготовление набора масок. Формирование каждого слоя на поверхности кремния.

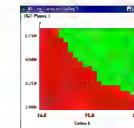


## 5. Производство

## 6. Отбраковка

## 6. Отбраковка:

Проверка микросхем перед использованием.



- Разработка новой ASIC включает **СОТНИ** шагов.
- Мы разделим разработку на шесть отдельных этапов.

# Изделие на бумаге

1-27



Размер кристалла

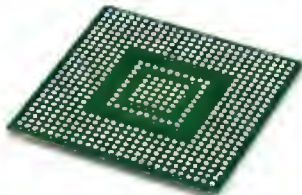
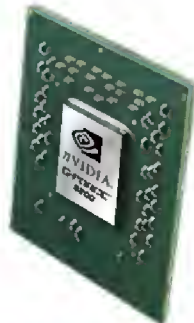
Тактовая частота



Особенности изделия

Feature	Market	Power	Package	Δcost
Video Decoder Fmt	MPEG2	1.0 W	731-pin BGA	+5%
Audio Decoder Fmt	Dolby 7.1	1.1 W	731-pin BGA	+10%
	MP3	1.2 W	731-pin BGA	+20%
Disk Format	Blu-Ray	1.3 W	731-pin BGA	+30%
	HDDVD	1.4 W	731-pin BGA	+30%
Video Resolution	1080i	1.5 W	731-pin BGA	+23%
	1080p60	1.6 W	731-pin BGA	+30%

Тип корпуса



Количество контактов корпуса

Потребляемая мощность



Напряжение питания

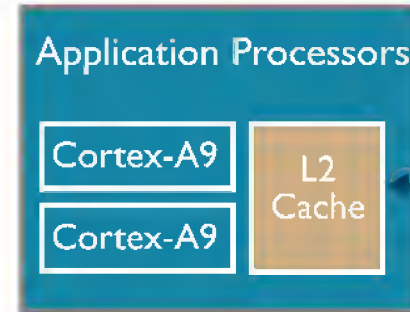
**BCM1103 IP Phone Chip**  
Product Requirements Document

**1.0 Technology (CMOS):**

- Fabrication technology is 0.13µm CMOS standard-cell for all digital blocks, and 0.18µm analog CMOS for on-chip codecs.
- Supply voltages are: 1.2/2.5V core, and 3.3V for I/O ring.
- Power consumption is 1.0W peak, and 0.15W standby.
- Packaged in 420-pin plastic BGA, for four-layer PCB designs.

**2.0 Architectural Specifications:**

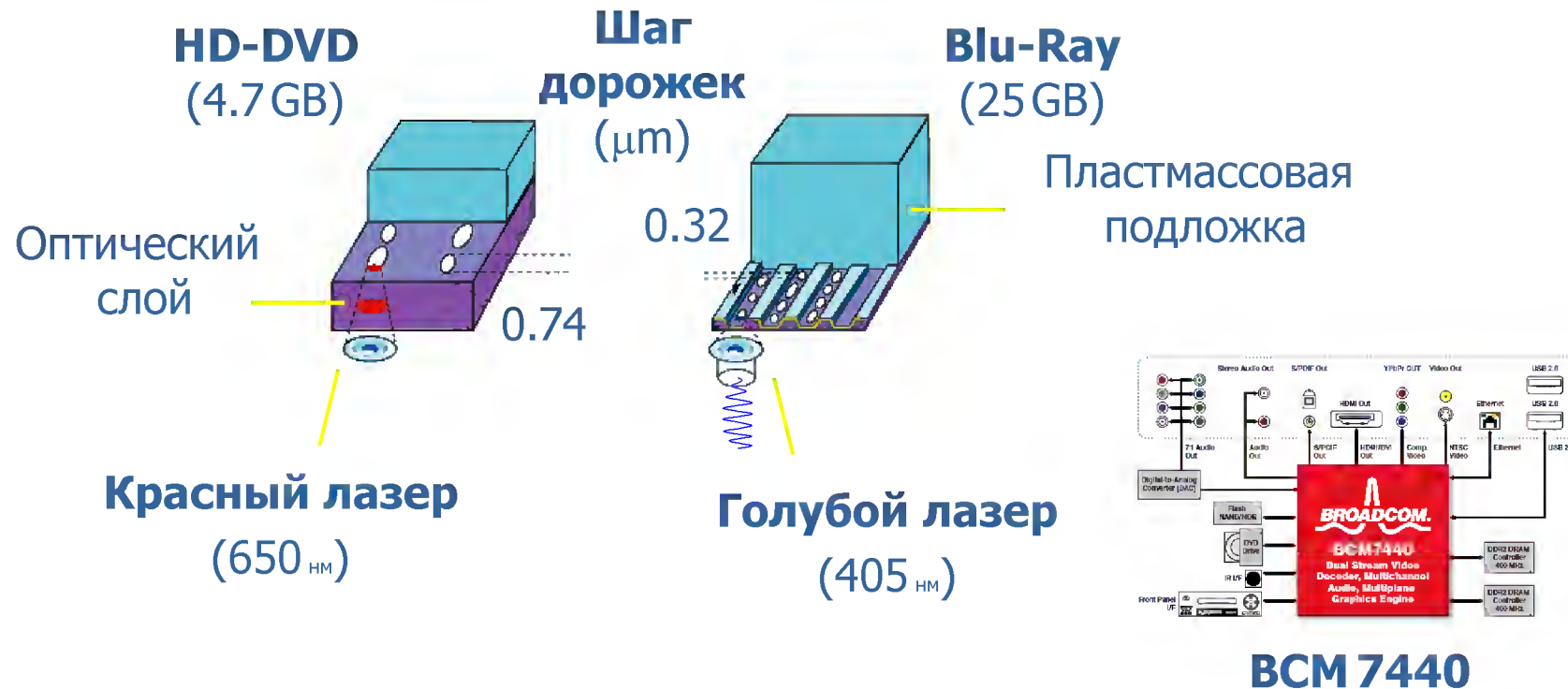
- Integrates a 1-gigabit Ethernet IP phone onto a single device.
- External IP core: ZEP 125 MHz DSP to support a variety of telephony algorithms, including acoustic echo cancellation.
- External IP core: MIP831 27.5 MHz CPU for VOIP protocol stacks and application code (C on top of Broadcom AXI).
- Three-port 10/100/1000 BASE-T Ethernet switch and MACs, for full- or half-duplex mode over CAT5 twisted-pair cabling.



Блоки IP (Интеллектуальной собственности)

Технологический процесс КМОП

- Документ PRD (Требования к изделию) перед кодированием описывает ASIC на бумаге.
- Тактовая частота, размер кристалла, и мощность - просто цели фирмы.



- Хорошее PRD (Требования к изделию) определяют **бизнес-возможности**, которым надо удовлетворить.
- Часто такие возможности возникают в результате технологического **прорыва**.
- 7440 используется в форматах DVD высокого разрешения.

# Таблица маркетинговых характеристик 1-29

Feature	Market	Power	Package	$\Delta$ cost
Video Decoder Fmt	MPEG2	1.0 W	731-pin BGA	+5%
Audio Decoder Fmt	Dolby 7.1	1.1 W	731-pin BGA	+10%
	MP3 (MPEG 1 layer 3)	1.2 W	731-pin BGA	+20%
Disk Format	Blu-Ray	1.3 W	731-pin BGA	+30%
	HD DVD	1.4 W	731-pin BGA	+30%
Video Resolution	1080i	1.5 W		
	1080p60	1.6 W		

→ **Ключевой принцип**

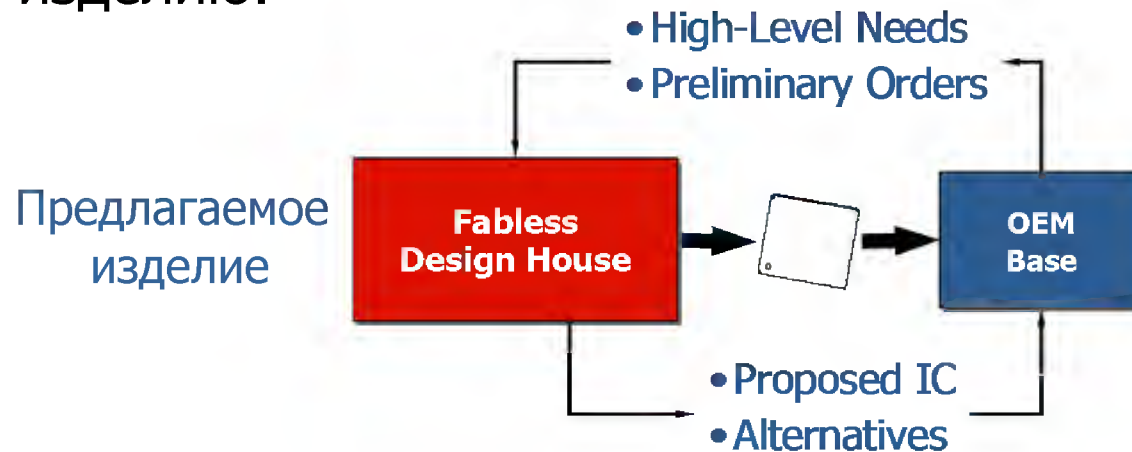
Площадь кристалла стоит денег.  
Коэффициент увеличения стоимости  
равен 6¢/mm<sup>2</sup>.

- Беспощадное **соревнование** определяет развитие микроэлектроники.
- Баланс **цена/производительность** критически важен.
- В таблице показано **влияние** добавляемой возможности на общую цену.

# Оценка реализуемости микросхемы 1-30

Как определить, можно ли реализовать новую микросхему?

- ✦ Коллектив, разрабатывающий спецификацию, должен выполнить несколько **итераций** для согласования требуемых характеристик и цены микросхемы.
- ✦ Новых характеристики могут потребовать проведения технических **исследований**.
- ✦ Отклик на требования рынка - рассмотрение **альтернатив**.
- ✦ Ключевые **потребители** могут повышать требования к изделию.



## Практический пример:

- Спецификация SM 1103
- Блок-схема архитектуры
- Функциональные требования
- Подходы к имплементации



# 1103 IP-телефон для Ethernet

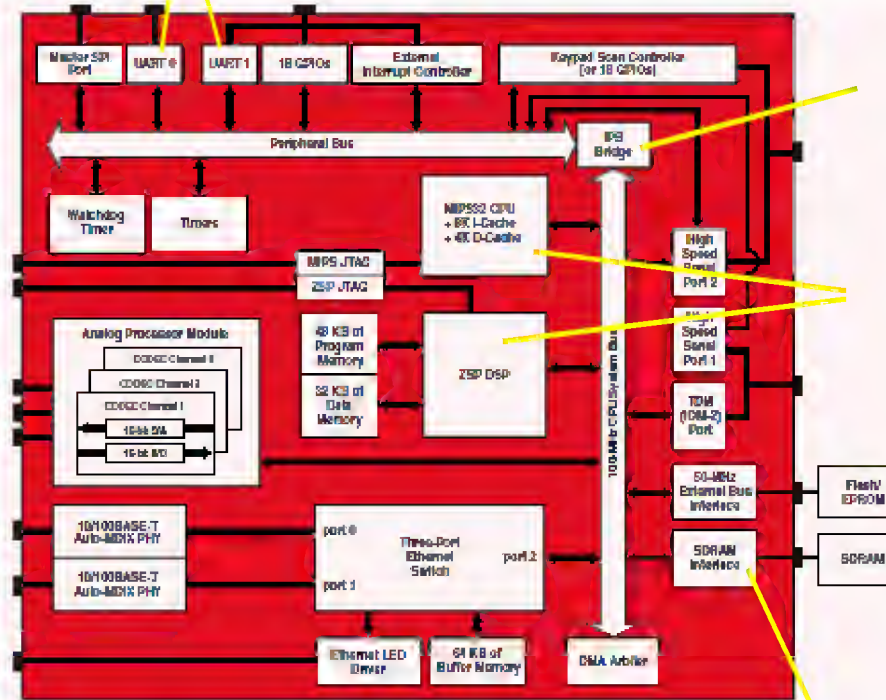
# 1-32

Внутренние IP-блоки:

Блоки UART

Цифровые функции

Аналоговые функции



Мост между структурами периферийной и системной шин

Внешние IP-блоки

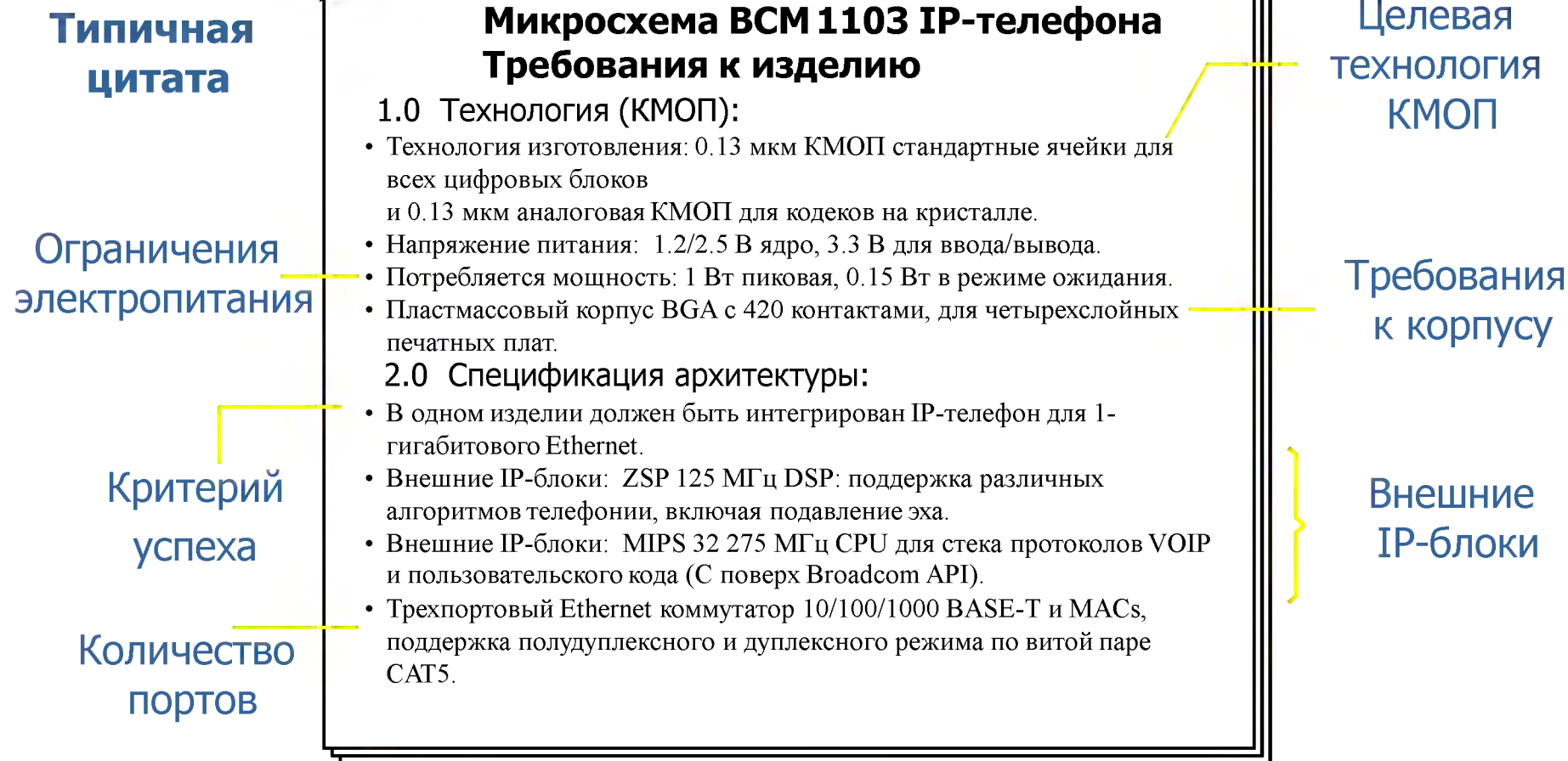
Внешняя память

Dual Data Rate (DDR)

- Схема архитектуры 1103 демонстрирует ключевые решения.
- Включает в себя системную и периферийную шину и мост.



# Спецификация проекта 1103 (1/3) 1-33



Вначале формируется **высокоуровневая** спецификация архитектуры .

# Спецификация проекта 1103 (2/3) 1-34

Типичная цитата

Функциональные требования

Что, но не каким образом

## Микросхема IP-телефона ВСМ 1103 Требования к изделию (продолжение)

### 2.1 Функциональные требования (цифровая часть):

#### 2.1.1 Контроллер клавиатуры (9×9):

- Контроллер должен обеспечивать борьбу с дребезгом контактов на аппаратном уровне, обрабатывать множественные нажатия и перебор клавиш.
- Контакты ввода/вывода блока контроллера можно будет переконфигурировать как контакты ввода/вывода общего назначения (GPIO) путем установки бита режима [5].

#### 2.1.2 Трехпортовый коммутатор Ethernet:

- Трехпортовый коммутатор должен обнаруживать активность несущей и передавать по активному порту Ethernet кадры в соответствии со стандартом 802.3.
- Он должен принимать/отправлять кадры со скоростью до 1,000 Мбит/с.
- Он должен обнаруживать конфликты в сети и сигнализировать о них.
- Он должен блокировать ненужный трафик: анализировать локальные MAC-адреса и сохранять их в кэш-памяти на 64 элемента.

Keyscan Controller  
(or 18 GPIOs)

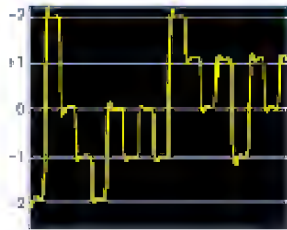
Функциональные  
блоки



Далее будут перечислены все функциональные требования

# Спецификация проекта 1103 (3/3) 1-35

Пятиуровневая  
логика



Подробности  
имплементации

## Микросхема IP-телефона ВСМ 1103 Требования к изделию *(продолжение)*

### 3.0 Спецификация имплементации (Цифровая часть):

#### 3.0.1 Контроллер клавиатуры (9×9):

- Логика кодировщика должна преобразовывать нажатую клавишу в ее ASCII код с помощью таблицы преобразования в ПЗУ 128×8 и выходного регистра.
- *Ненулевой* код в регистре будет инициализировать последовательность вызова.

#### 3.0.2 Трехпортовый коммутатор Ethernet:

- Трехпортовый коммутатор на 1000 Мбит/с будет передавать кадры Ethernet по четырем парам CAT-5 UTP, восемь бит за раз.
- Он будет использовать *эхоподавление* между +TD и -TD для преодоления ограничения в 100 Мбит/с пары CAT-5 UTP.
- Он будет использовать *пятиуровневую* амплитудную модуляцию (PAM-5).
- Каждые восемь битов данных будут передаваться четырьмя трехбитовыми символами с использованием LFSR- скремблирования с многочленом  $x^4 + x^3 + x + 1$ .
- Трехбитовые символы будут отображаться на уровни напряжения

...	
17:	h23
18:	h39
19:	h00
20:	h36
21:	h00
22:	h00
23:	h00
24:	h33
...	
7F:	h00

Таблица  
преобразо-  
вания в ПЗУ



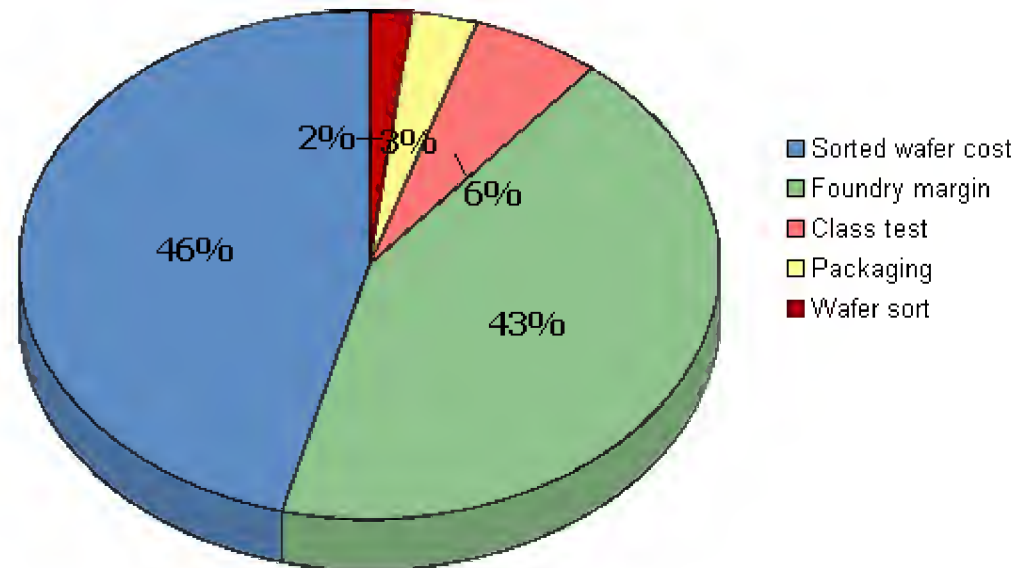
UTP Кабель

Далее детально рассмотрены детали **имплементации**.

## **Планирование проекта:**

- Экосистема ASIC
- Основные задачи планирования
- Пластины с различными микросхемами

Breakdown of Wafer Cost



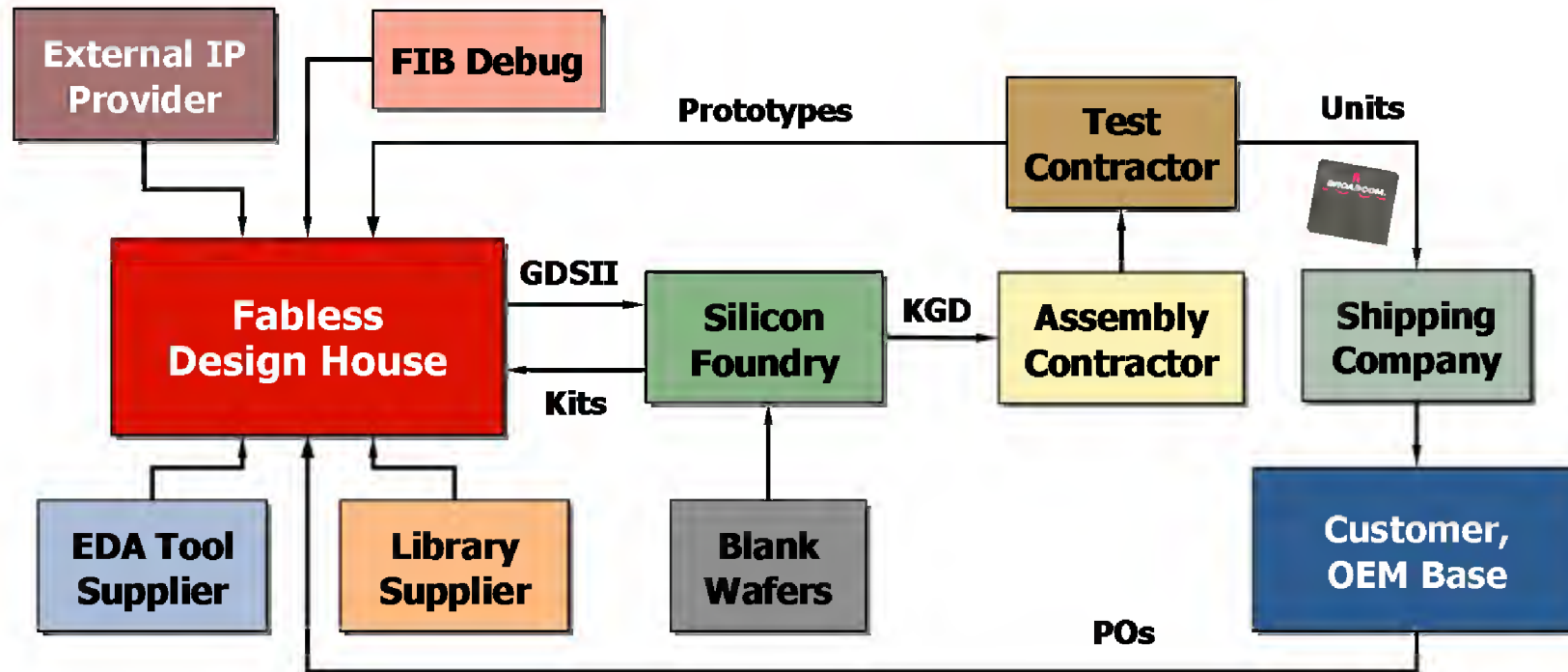
- Изготовление пластин является массовым производством.
- Но сотни шагов на сложном оборудовании очень дороги.
- Предприятие по производству интегральных схем должно заработать свою прибыль, амортизация оборудования.

# Цена пластин (2/2)

1-38

TSMC 28-nm High-Performance HKMG CMOS					
10-Layer Cu, Double Gate-Oxide (DGO), 55-mask, 300-mm wafers, Fab 14 (Taiwan)					
Fabrication	Units	Amount	Package/Test	Units	Amount
Blank wafer cost	\$/wafer	\$125	Package cost	\$/package	\$0.12
Foundry margin	\$/wafer	\$2,500	Class-test cost	\$/device	\$0.20
Unsorted wafer cost	\$/wafer	\$5,200	Total device cost	\$/device	\$3.32
Wafer sort cost	\$/wafer	\$200			
Sorted wafer cost	\$/wafer	\$5,400	<b>Packaging Notes:</b>		
Die yield	%	90	• Package type: TSCP54 wire-bond.		
Gross die per wafer	–	2000	• Very high-volume packaging (> \$50M).		
Good die per wafer	–	1800	• No copper wire; no resistors/capacitors.		
Per good sorted die	\$/die	\$3.00			

- Параметры цены сильно зависят от технологии и размера кристалла.
- В таблице приведены типичные слагаемые цены.
- При фиксированной цене пластины, выход годных изделий очень важен.



- Фирмы разработчики микросхем без собственных производственных мощностей сфокусированы на инновационных проектах.
- Их специалисты могут тесно сотрудничать с предприятиями по производству интегральных схем.
- Надежность планирования проектов обеспечивается взаимными **обязательствами** партнеров.

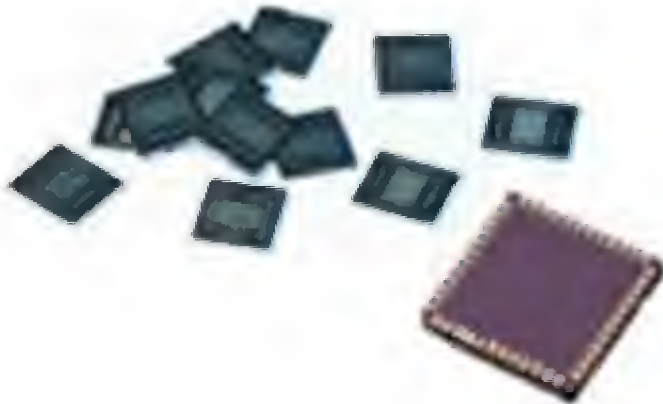




# Шатл-пластины (2/2)

1-41

## Расписание MPW



**Поставляемые изделия**  
(В корпусе или без него)

MPW	Process	DRC Clean GDSII
MPW1	SOLD OUT	Oct 08
MPW2	SOLD OUT	Feb 09
MPW3	SOLD OUT	Jun 09
MPW4	32LP	Nov 09
MPW6	28 nm compatible	1H 10

⚡ Предупреждение:  
Зафиксированное время  
используй-или-  
потеряешь. Слишком поздно  
опоздаешь на корабль.

- Экономичные запуски для малых объемов, **раз в месяц**.
- Проектировщик должен представить на предприятие по производству файл GDSII после проверки **DRC (соблюдения проектных норм)**.
- Предприятие возвращает **микросхемы в корпусах** или полупроводниковую пластину **разрезанную на кристаллы**.

---

# Основные задачи планирования 1-42

## Планирование проекта включает следующее:

- ✦ **Выбор** основных партнеров в цепи поставок и оформление обязательств с ними (например, GlobalFoundries)
- ✦ Расписание изготовления **шатл-пластин** и цены.
- ✦ **Опции производства:**  $V_T$  области; двойные канавки; poly 2.
- ✦ **Поставщики** внешних IP-блоков: программные, аппаратные, топологические?  
Тестовые векторы? Модели для моделирования? Даты выпуска?
- ✦ Оценка **стоимости** микросхемы: разработка, на одно изделие.
- ✦ **Жизненный цикл** изделия и **планы на будущее.**

---

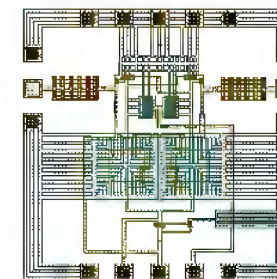
# Developing the Nanometer

# ASIC

## Specs to Silicon

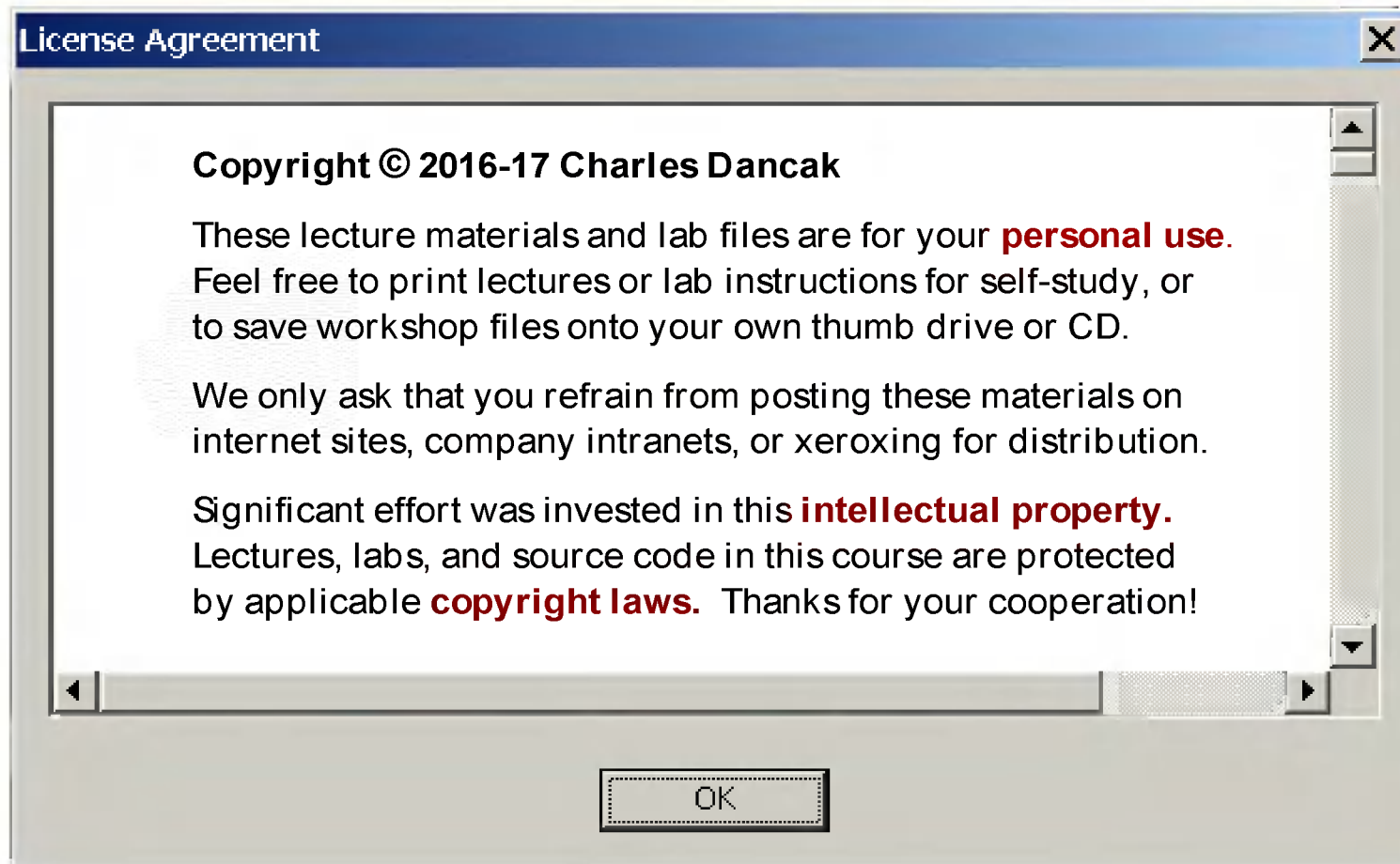


## 2. Стадия проектирования (Цифровая и аналоговая части)



# Авторское право

2-2



**Чтобы организовать семинары на своем предприятии, свяжитесь с Quantum Consulting по тел: 650/380-9121**

# Стадия проектирования

# 2-3



## 2. Проектирование:

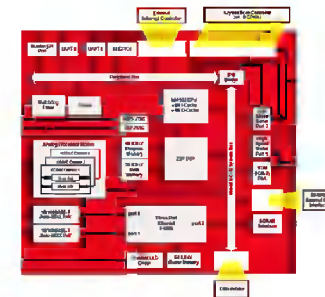
Реализация замысла проекта путем создания описания его цифровой части на уровне регистровой передачи (RTL), а аналоговой - на уровне GDSII.

- После подписания PRD (Требований к изделию) начинается **практическое проектирование**.
- Каждая цифровая функция описывается в файле RTL кода.
- Каждая **аналоговая** макроячейка проектируется вручную до разработки ее топологии.

## Логические блоки RTL

- Блок-схема ASIC
- Иерархия блоков
- Проектирование на основе HDL
- Стиль разработки HDL кода
- Логический синтез
- Временные характеристики одного цикла

RTL блоки



# Блок-схема ASIC

# 2-5

**BCM  
1103**

**Внутренние IP-блоки:**  
UART ядра.

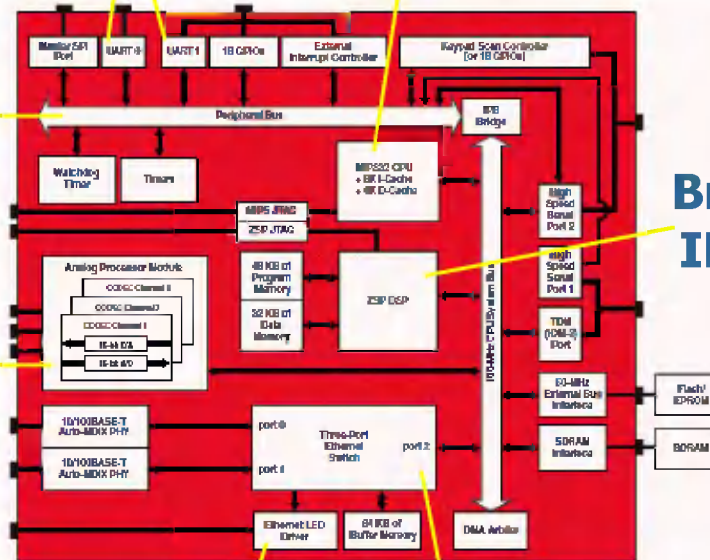
**Пользовательская программа:**  
Написана на C, поверх API  
цифровой обработки сигналов.

**Шинная  
структура**

**Аналоговые  
макроячейки**

**Внешние  
IP-блоки**

**Память  
(вне микросхемы)**



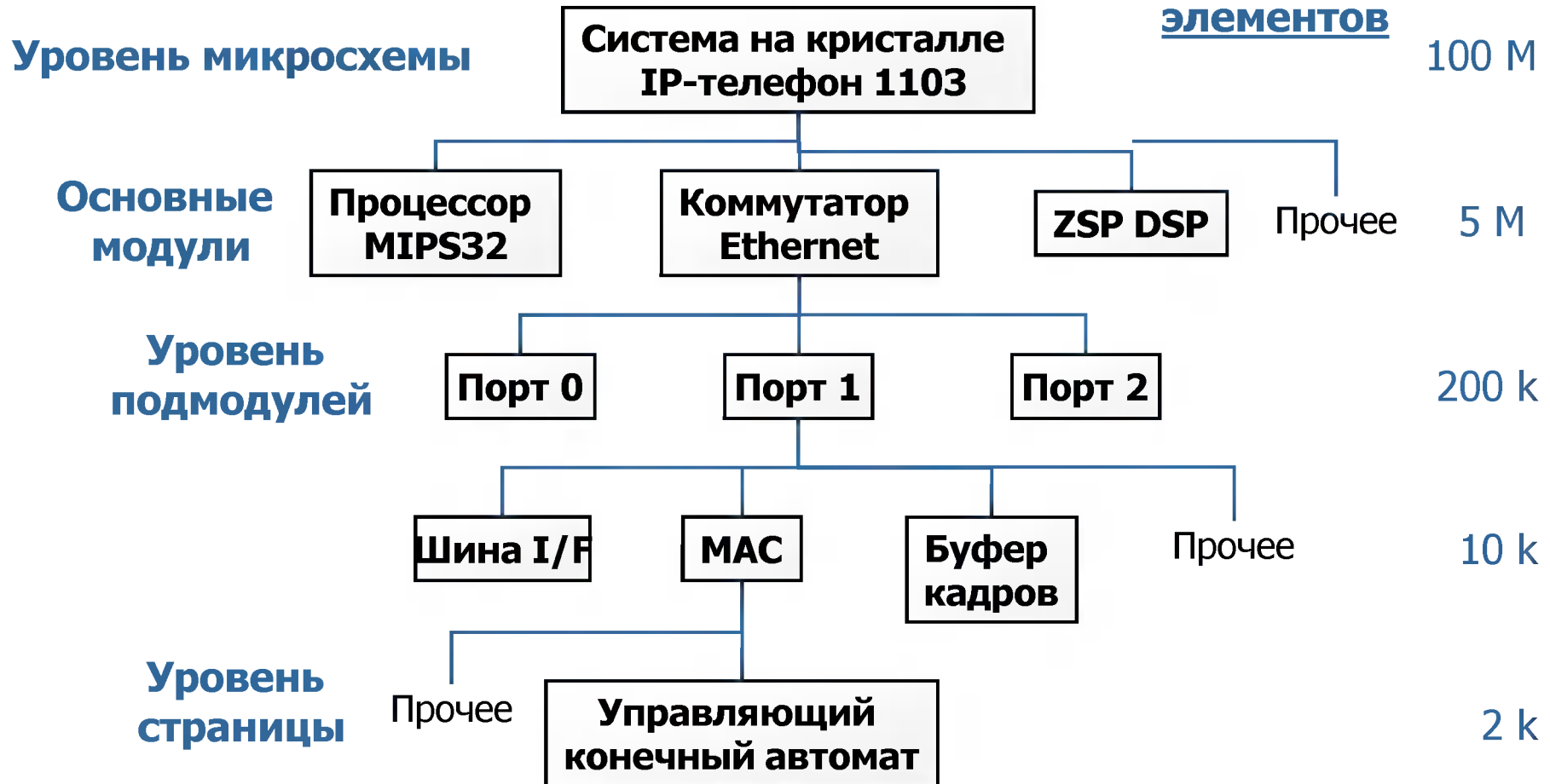
**Модуль RTL:**  
Логика, определенная  
пользователем.

**Блоки подсистемы:**  
Сами являются иерархическими  
блоками.

- Процесс проектирования осуществляется исходя из PRD (Требований к изделию).
- Разработку пользовательских логических блоков поручают проектировщикам RTL.

# Иерархия блоков

Количество  
элементов **2-6**



- Для управления сложностью, пользовательские блоки разделяются.
- Коммутатор Ethernet содержит 3 идентичных порта.

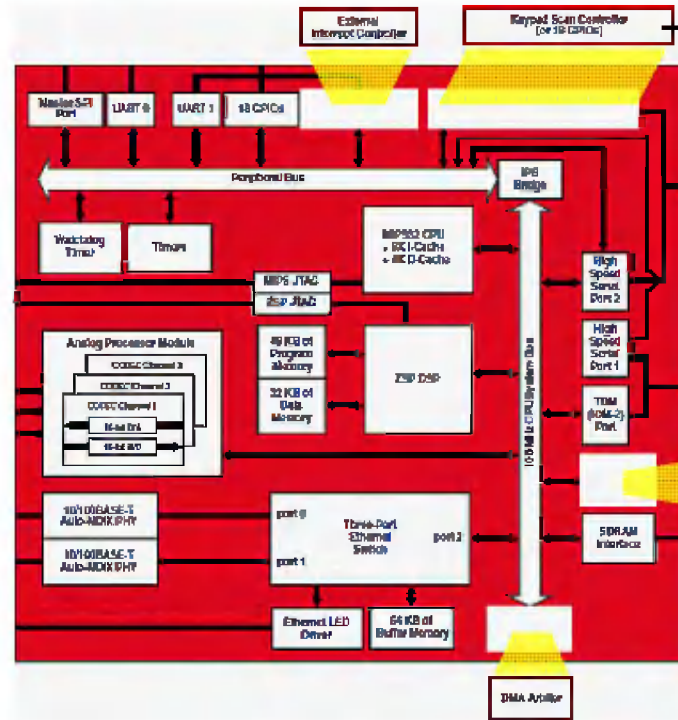


# Типичный RTL блок

2-7

**BCM  
1103**

Уровень микросхемы: Код верхнего уровня микросхемы **1103** включает в себя все остальные подмодули.



Прочие типичные RTL модули.

Шинный интерфейс:: Типичный RTL блок.

```
SERIAL4.v
1 /* RTL Verilog Code
2 * Four-bit serializer,
3 * with jtag control.
4 */
5 module SERIAL4(
6     output wire TX, //Serial
7     input wire [3:0] MAC_DATA,
8     input wire LOAD_SHIFT, MAX, CLK
9 );
10     reg [3:0] FF; //Four Flip-Flops.
11
12     always @(posedge CLK)
13         if (LOAD_SHIFT) //Load data.
14             FF <= MAC_DATA;
15         else
16             FF <= [1'b1,FF[3:1]]; //Shift.
17     assign TX = FF[0]MAX; //Out.
18
19
20 endmodule
```

SystemVerilog: Язык описания аппаратуры, подобный C.

- Цифровые логические блоки часто описываются на языке **SystemVerilog**.
- Стандартный язык описания **аппаратуры** (HDL).
- Модели HDL позволяют описать функции объектов, создание их экземпляров и соединение.

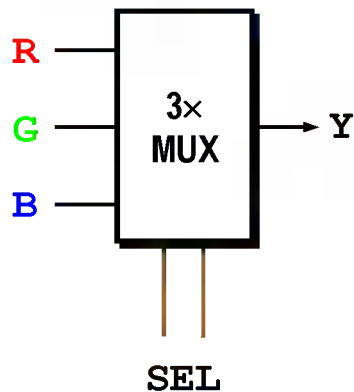
# Проектирование мультиплексора (MUX)

## 2-8



### Видео RGB:

Выбирается один из битов данных R, G, B и подается на выход Y.



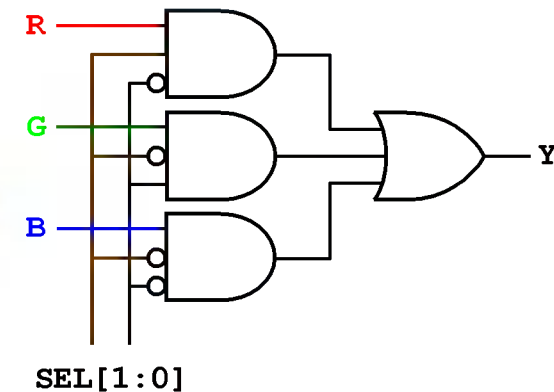
**Блок RTL**  
(3× MUX)

3:1 Multiplexer				
SEL	R	G	B	Y
00	?	?	B	B
01	?	G	?	G
10	R	?	?	R
11	?	?	?	?

### Таблица истинности:

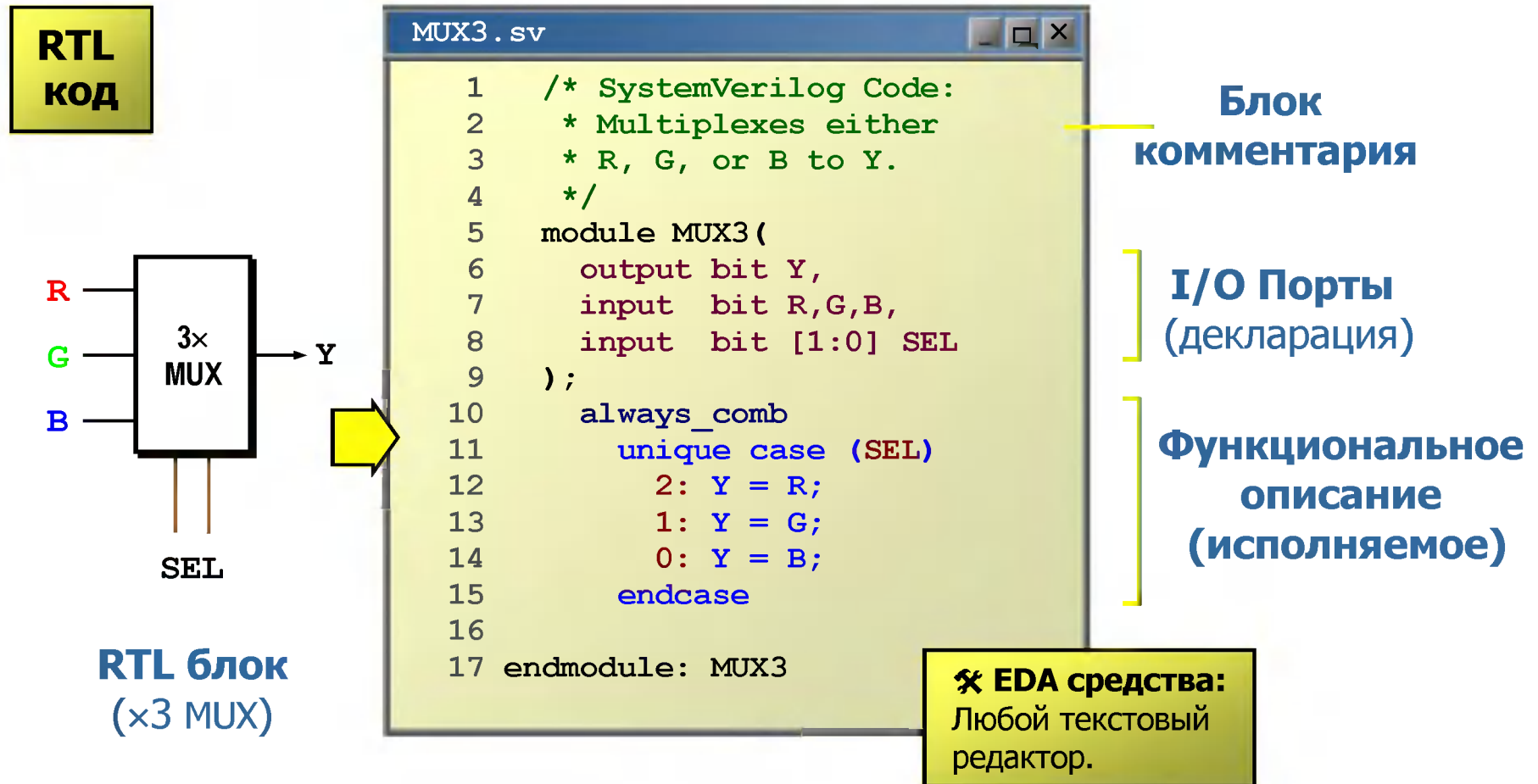
Определяет Y для всех комбинаций входов.  
? означает безразличное состояние (*don't-care*).

### Схема уровня логических элементов



- Перед использованием средств EDA, блоки синтезируются **вручную**.
- Проектировщик работает с таблицей истинности для разработки схемы MUX3.

# Проектирование на основе HDL 2-9



- Код HDL описывает работу блока MUX3.
- Готово для синтеза в набор соединенных логических элементы.

# Стиль разработки HDL кода 2-10

Ошибочный тип  
(ширина 32 бита)

Все диапазона  
(Допускается  
только 0..3)

```
MUX3 . sv
1  /* SystemVerilog Code:
2  * Multiplexes either
3  * R, G, or B to Y.
4  */
5  module MUX3 (
6      output bit Y,
7      input  bit R,G,B,
8      input  int [1:0] SEL
9  );
10     always_comb
11         unique case (SEL)
12             4: Y = R;
13             1: Y = #1 G;
14             0: Y = B;
15         endcase
16
17 endmodule: MUX3
```

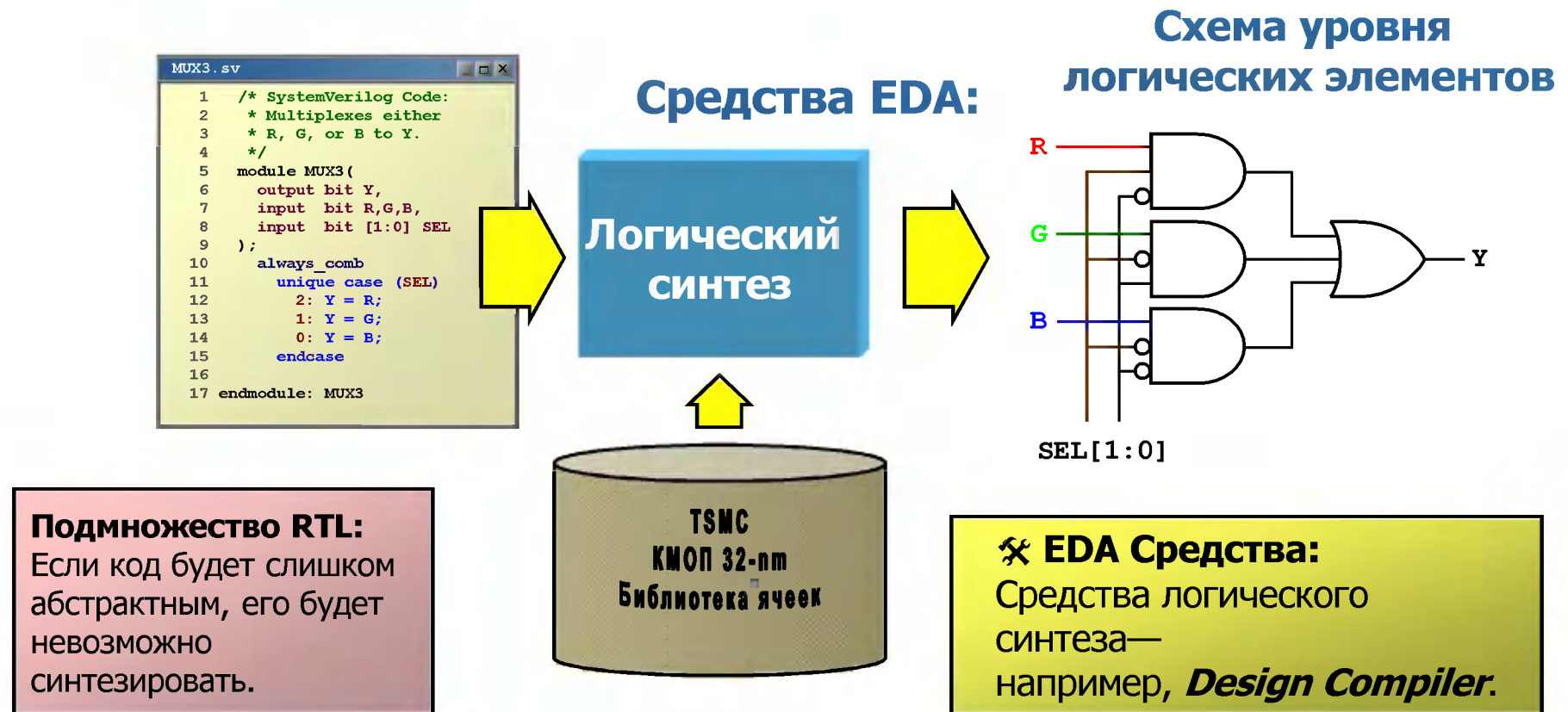
✘ EDA Средства:  
Любое средство  
контроля стиля.

Нарушение  
синтаксиса RTL  
(единичная задержка)

Функциональный код:  
Описывает реализацию  
таблицы истинности - не  
задержки.

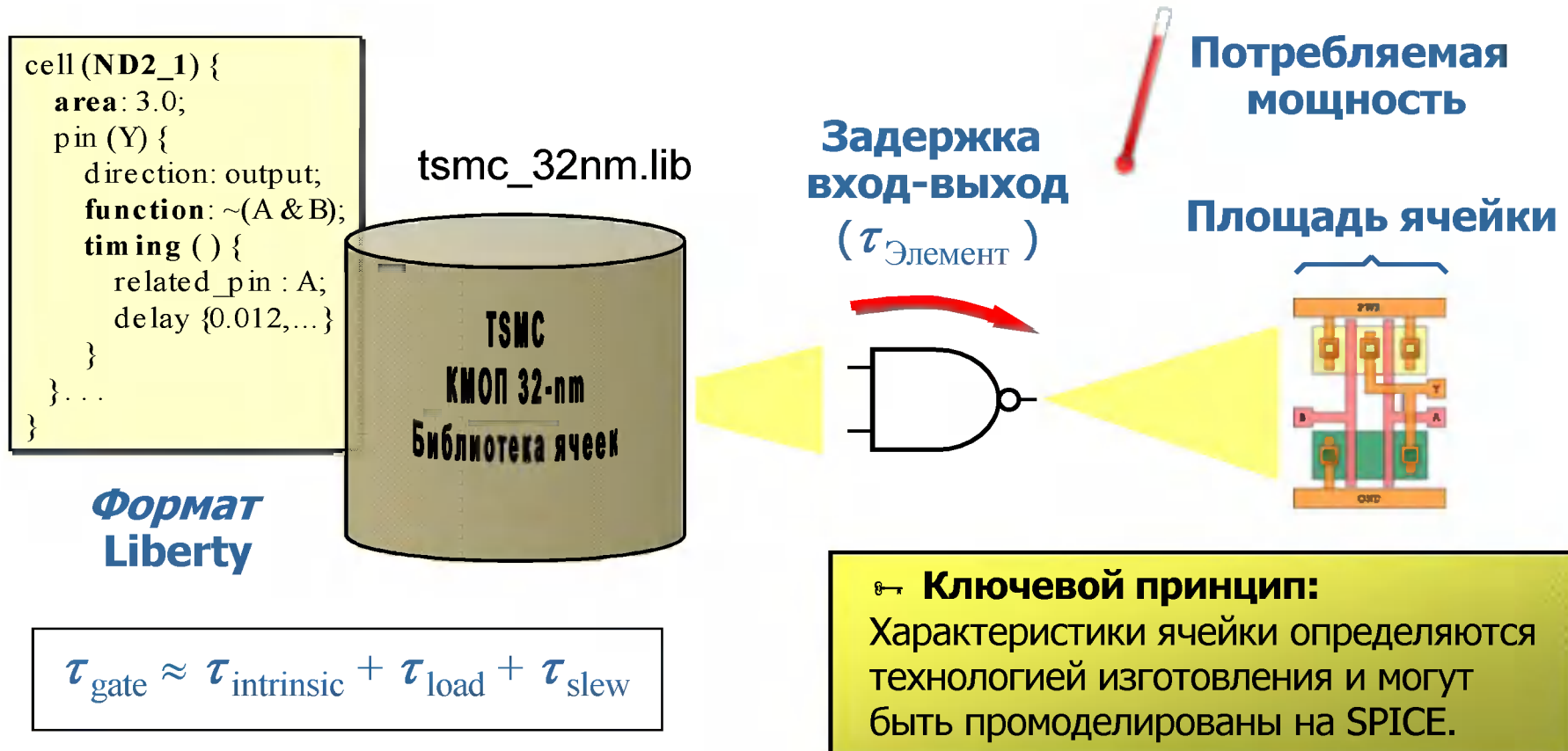
- Средства контроля стиля (Lint) выделяют **синтаксические ошибки** до синтеза.
- Предупреждают о рискованном коде или **несинтезируемом синтаксисе**.

# Проектирование на основе HDL 2-11



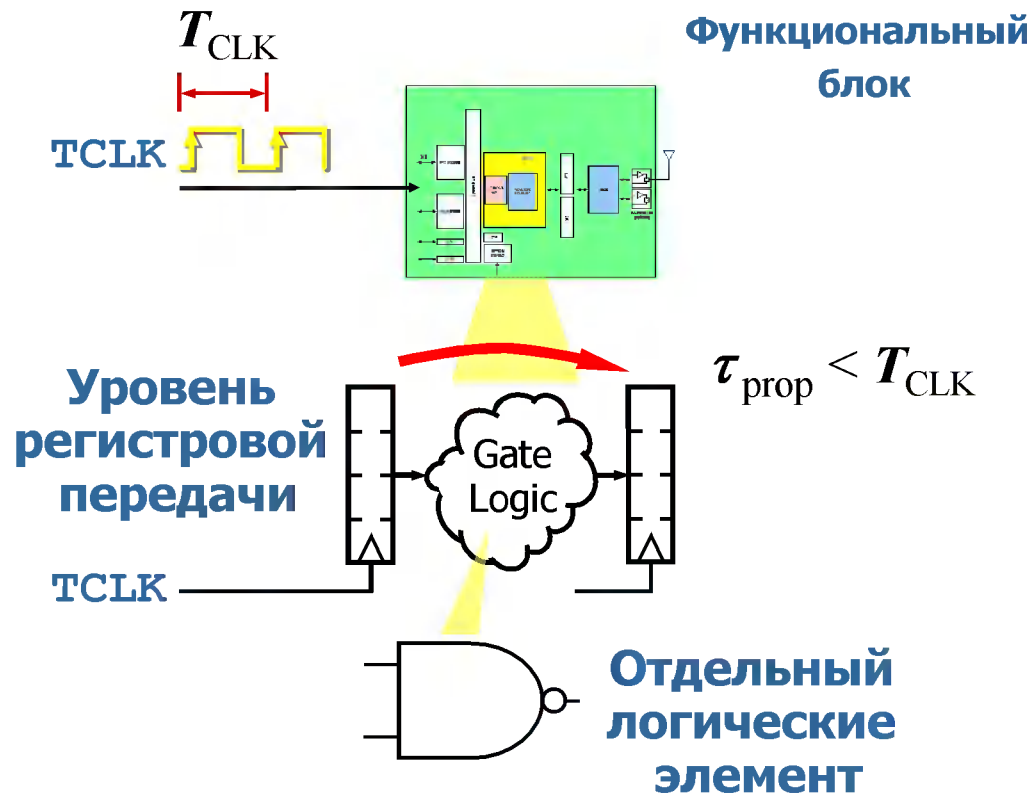
- Средства синтеза могут **читать** HDL код из файла.
- Затем они компилируют HDL-конструкции в соответствующую логику.
- Оптимизация для повышения быстродействия - затем уменьшение общей площади.

# Библиотека стандартных ячеек 2-12



- Средства синтеза используют **библиотеку ячеек** конкретного изготовителя.
- Большинство ячеек характеризуются **силой драйвера** от  $\times 1$  до  $\times 16$ .
- **Характеристики** ячеек указываются в компактном формате **Liberty**.

# Временные характеристики одного цикла 2-13



- **Тактовый сигнал** всей микросхемы загружает данные во входной регистр.
- Загруженные данные начинают распространяться через облако логических элементов.
- Они должны дойти до выходного регистра за **один** период тактового сигнала.

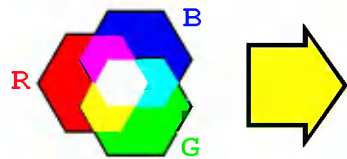
## **Нарушение временных характеристик:**

- **Неоптимальная схема**
- **Задержка в критическом пути**
- **Оптимальная схема MUX3**
- **Анализ пути**
- **Роль RTL-проектировщика**



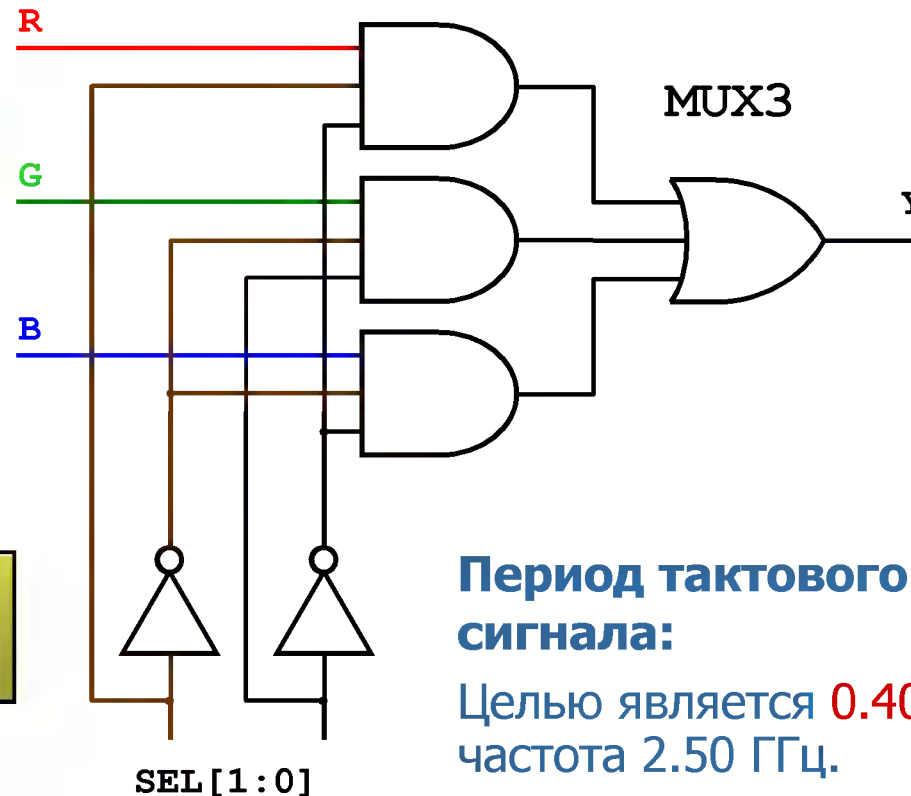
# Неоптимальная схема MUX3

2-15



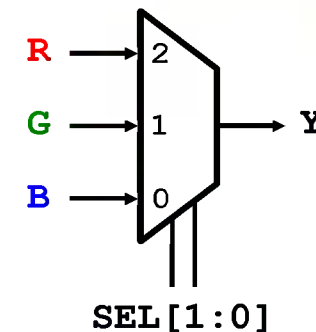
## Цифровое ТВ:

Выбирается один из битов данных **R**, **G**, **B** и подается на выход **Y**.



**Период тактового сигнала:**

Целью является **0.40 нс** — тактовая частота **2.50 ГГц**.



- Средства синтеза могут оптимизировать всю схему для повышения быстродействия.
- Парадигма печенья Oreo: Рассматриваются параметры **одного** периода.
- Эта реализация на элементах И и ИЛИ не является оптимальной.

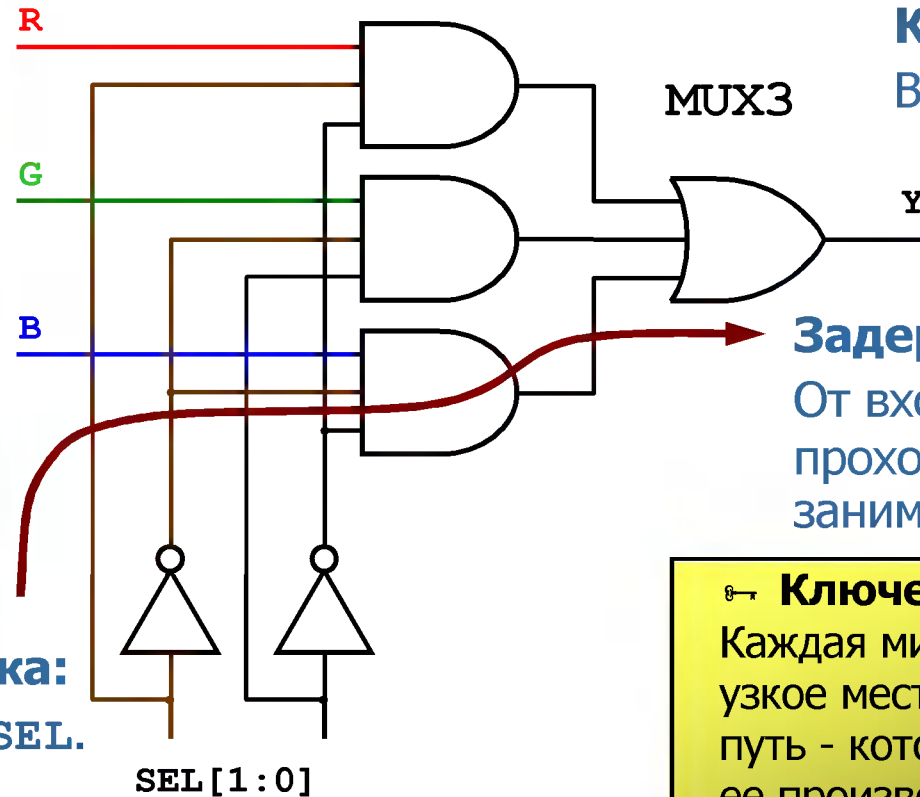
# Задержка в критическом пути

# 2-16

↳ Перерисованная схема:  
Чтобы отобразить неявные задержки, следует показать настоящие инверторы.

**Критический путь**

**Начальная точка:**  
Один из входов SEL.



**Конечная точка:**  
Выходной порт Y.

**Задержка пути:**  
От входа SEL до Y,  
прохождение данных  
занимает **0.44 нс.**

↳ **Ключевой принцип:**  
Каждая микросхема имеет узкое место - критический путь - который ограничивает ее производительность

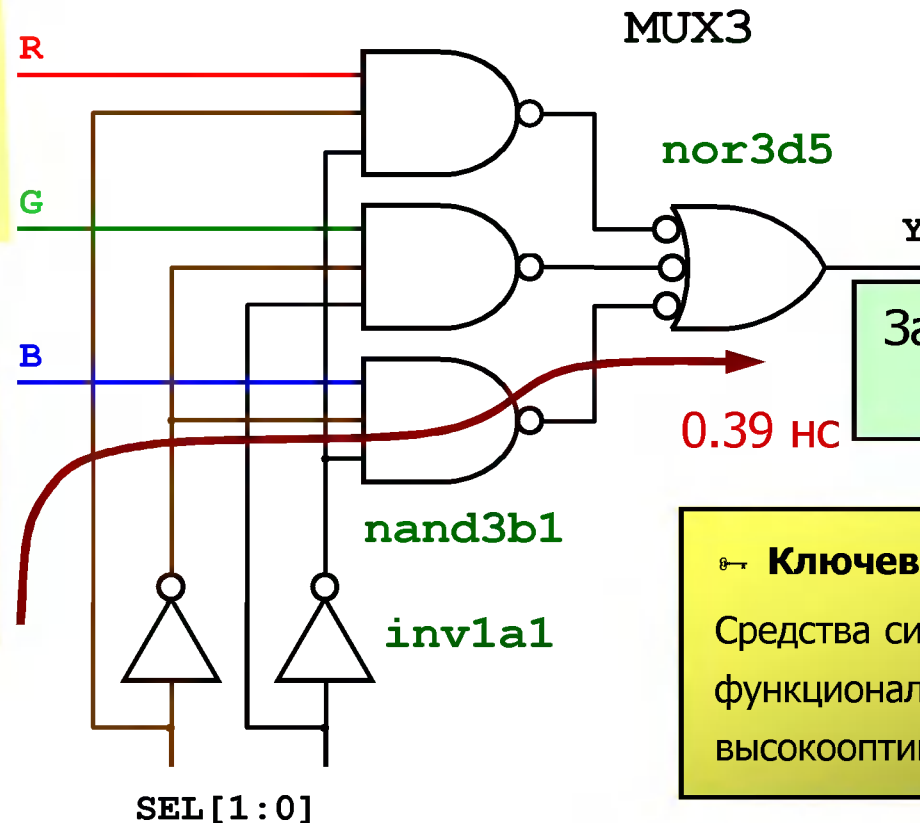
- Критический путь **самый длинный**, от одного из входов до выхода.
- Временной анализ показывает, что прохождение данных занимает **слишком много времени!**

# Оптимальная схема MUX3

2-17

↳ Эквивалентность:  
Последовательный  
инверсии **уничтожаются**—  
получается  
эквивалентная схема.

Синтезированный  
логические  
элементы  
(И-НЕ, ИЛИ-НЕ)



Задержка меньше  
на 50 пс!

← **Ключевой принцип:**

Средства синтеза создают  
функционально эквивалентную,  
высокооптимизированную схему.

- Средства синтеза заменяют элементы И и ИЛИ на И-НЕ и ИЛИ-НЕ.
- Простая временная оптимизация использует теорему де Моргана.
- Временной анализ показывает сокращение критического пути на **50 пс**.

# Анализ пути

2-18

**Критический путь Поэлементный**

**EDA Средства:**  
Статический временной анализ — например, *Prime Time*.

**Задержка элемента**

Delay to Gate Pin or Port	Incr [ns]	Path [ns]
External Delay	.00	.00
MUX3/SEL[1]	.00	.00
MUX3/U1/Y (inv1a1)	.04	.04
MUX3/U2/Y (nand3b1)	.15	.19
MUX3/U3/Y (nand3b1)	.00	.39
MUX3/U4/Y (nand3b1)	.00	.39

**Общая задержка**

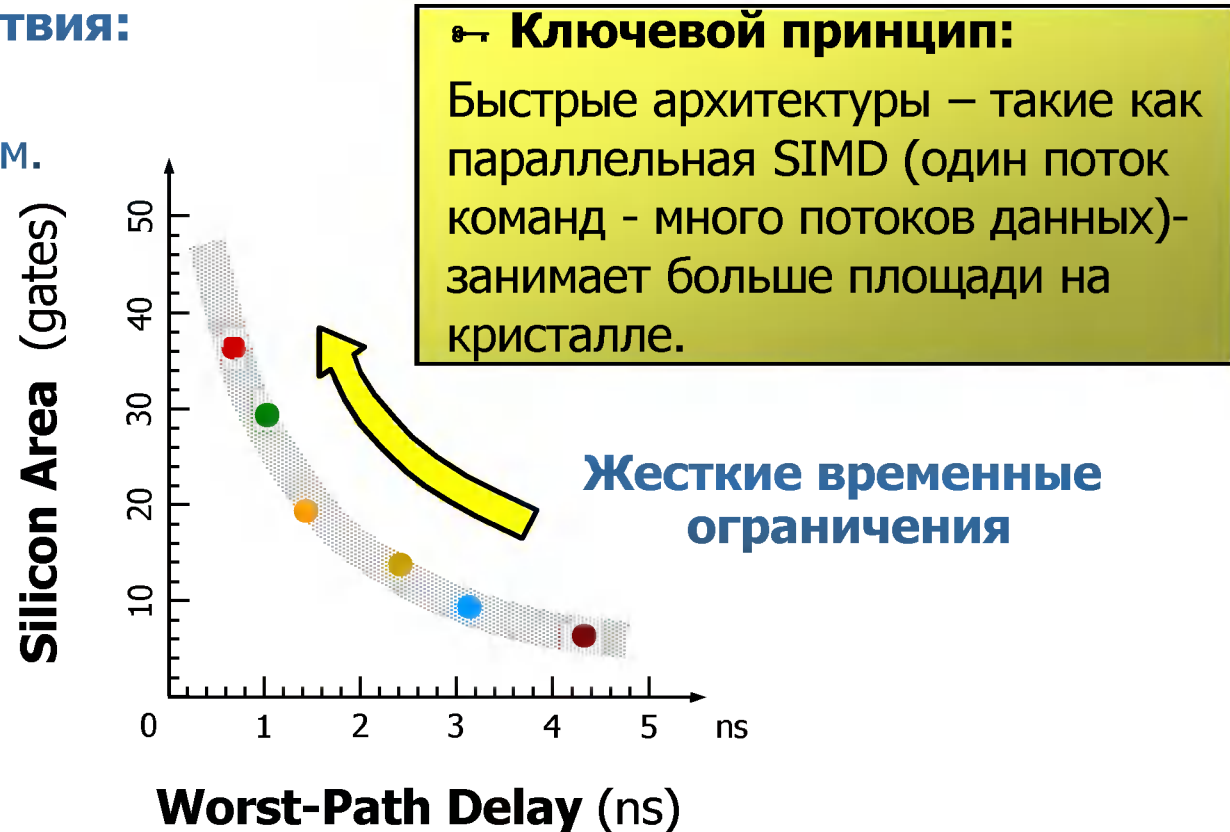
**Временные требования удовлетворены!**

- Средства статического анализа (STA) рассматривают схему, как диаграмму PERT (Program Evaluation and Review Technique, система планирования и руководства разработками) со многими **путями**.
- По умолчанию, сообщается о **самом длинном** (наихудшем) пути.
- Временные требования **удовлетворены** (так как  $\tau_{prop} < T_{CLK}$ ) с запасом 10 пс.

# Компромисс площадь/время 2-19

## Повышение быстродействия:

- Переработать HDL код.
- Использовать параллелизм.
- Продублировать тракт данных.
- Быстрая DesignWare.
- Сокращение обработки.



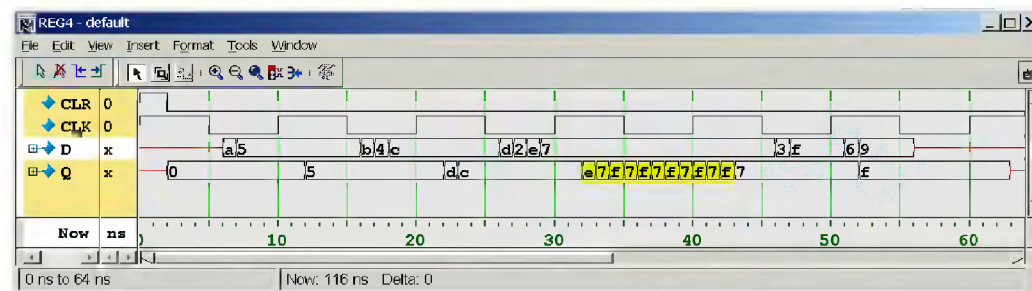
- Средства синтеза могут создать **альтернативную** имплементацию.
- Указанные ограничения указывают какая имплементация будет оптимальной.
- Неотъемлемый компромисс: Уменьшение **задержки** блока приводит к увеличению его **площади**.

# Роль RTL-проектировщика 2-20

Проектировщик RTL-блоков часто выполняет следующие действия:

- ↳ Отлаживает свой код при **моделировании** работы блока.
- ↳ Проводит исследовательский **синтез** отлаженного кода.
- ↳ **Временные** требования удовлетворены? Удовлетворены требования к **площади** и потребляемой **мощности** блока?
- ↳ Выполнение требований проектирования для тестирования (DFT). Запуск проверки правил DFT, исправление нарушений.

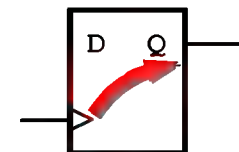
Временные  
диаграммы  
моделирования  
уровня блока



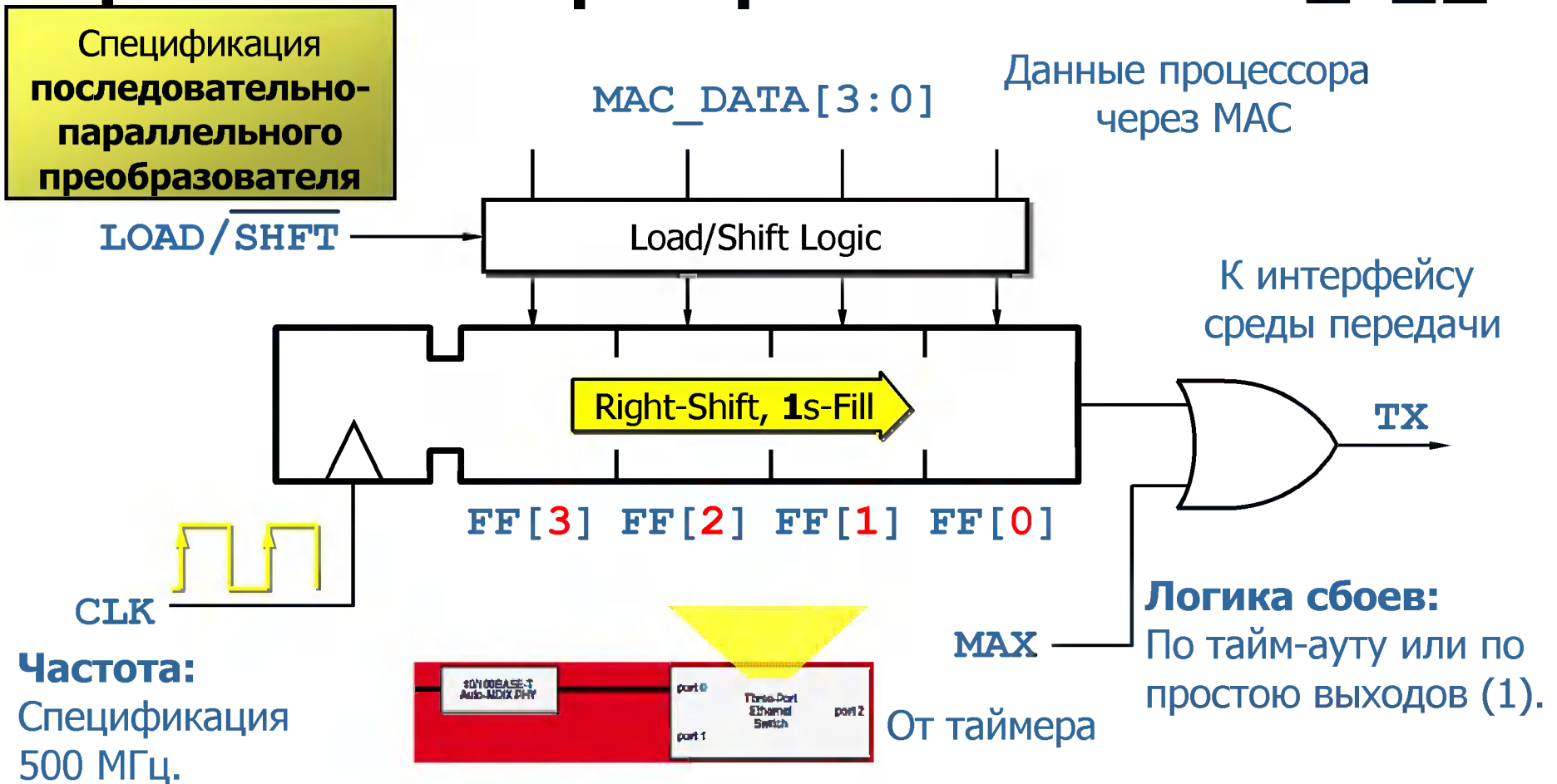
## Практический пример:

- Спецификация последовательно-параллельного преобразователя
- Описание HDL
- Синтезированная логическая схема
- Отчет о временных характеристиках
- Влияние задержек в соединениях

FF[1]



# Практический пример: Блок SERIAL4 2-22



- В нашем последнем примере рассматривается чисто **комбинационная** логическая схема.
- В этом примера разработки 1103 используются **триггеры** FF[3:0].



# Описание SERIAL4 на HDL

# 2-23

Последовательно-параллельный преобразователь:  
RTL код

```
SERIAL4.sv
1  /* SystemVerilog Code:
2  * Four-bit serializer,
3  * with jabber control.
4  */
5  module SERIAL4(
6  output bit TX, //Serial.
7  input bit [3:0] MAC_DATA,
8  input bit LOAD_SHFT, MAX, CLK
9  );
10 bit [3:0] FF; //Four flops.
11
12 always_ff @(posedge CLK)
13     if (LOAD_SHFT)
14         FF <= MAC_DATA; //Load data.
15     else
16         FF <= {1'b1,FF[3:1]}; //Shift.
17
18     assign TX = FF[0] | MAX; //OR.
19
20 endmodule: SERIAL4
```

← **Ключевой принцип:**  
HDL описывает параллельные процессы: `always` и `assign` отдельные потоки.

Внутренние сигналы

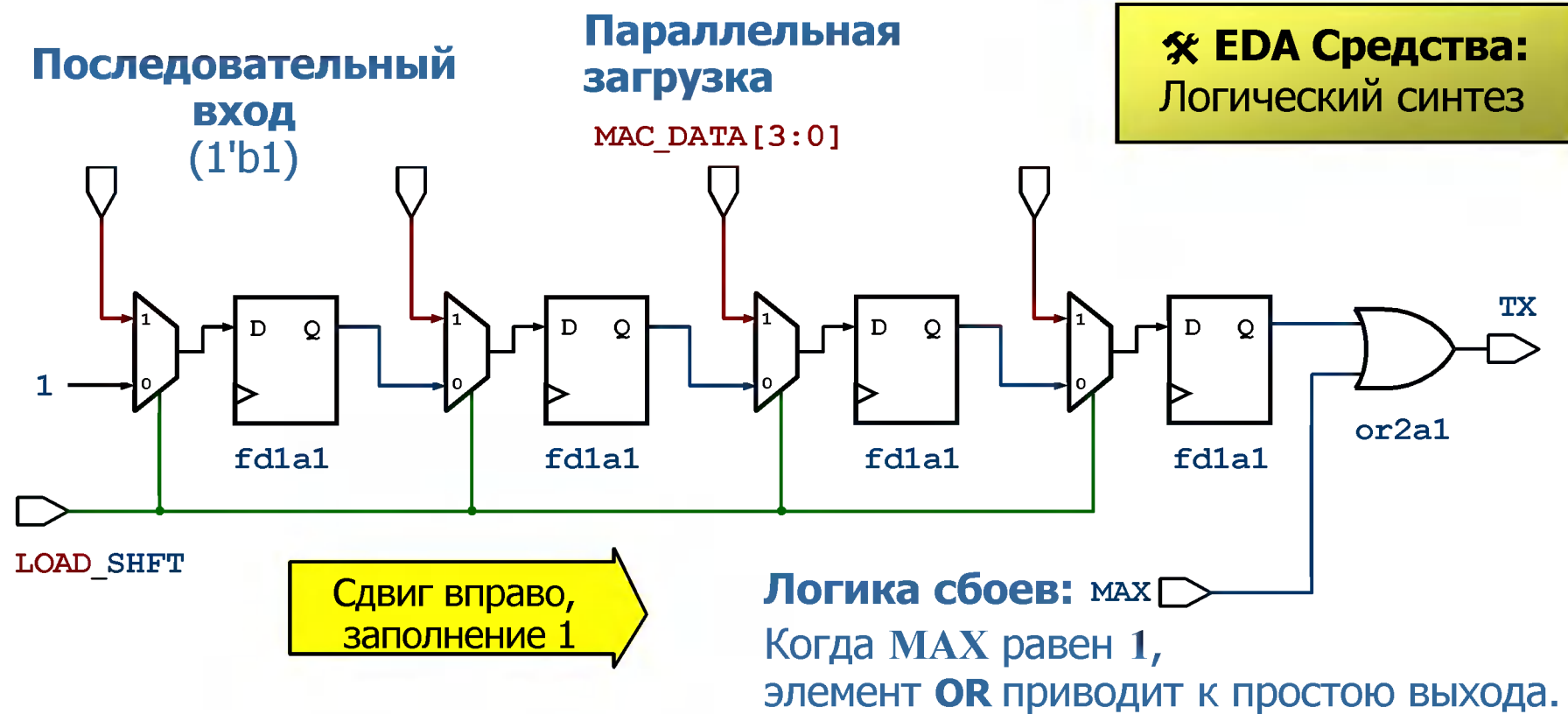
if-else  
(Загрузка/Сдвиг)

Запуск фронтом  
(передним фронтом тактового сигнала)

Оператор SystemVerilog  
OR

Код RTL описывает **функциональность** блока: загрузку и сдвиг.

# Синтезированная схема SERIAL4 2-24



- Оператор if-else синтезируется как схема **MUX2**.
- Оператор OR реализуется с помощью стандартной ячейки **or2a1**.
- Но будет ли надежно выполняться **сдвиг** при тактовой частоте 500 MHz?

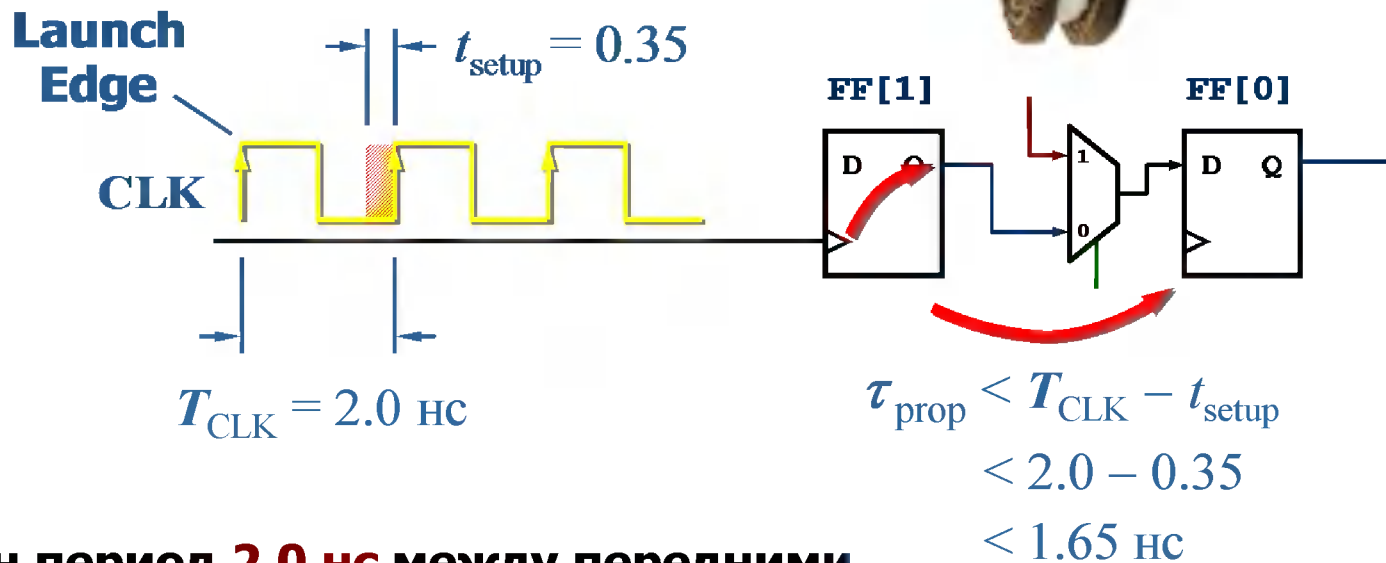
# Временные характеристики одного цикла

# 2-25

```
dc> create_clock -period 2.0 CLK
```



Парадигма печенья Oreo



- Указан период **2.0 нс** между передними фронтами такого сигнала CLK.
- За один период данные должны **пройти** от FF[1] до FF[0].
- Более точно, один период должен быть **меньшим**, чем время установки FF[0].
- Будет ли задержка тракта FF[1]-MUX2 **удовлетворять** ограничениям?

# Отчет о временных характеристиках SERIAL4 2-26

```
SERIAL4 Timing Report
*****
Report: timing
Module: SERIAL4
Library: tsmc_65nm
*****
Startpoint: FF[1]
Endpoint: FF[0] (setup: 0.35 ns)
Delay to Gate      Incr
Pin or Port        [ns]          [ns]
-----
FF[1]/Q (fd1a1)    .88             .88
FF[1] (net)        --             --
MUX2/U5/A (mux2a1) .00             .88
MUX2/U5/Y (mux2a1) .34             1.22
MUX2 (net)         --             --
FF[0]/D (fd1a1)   .00             1.22
```

**Перед трассировкой:**  
Перед трассировкой задержки в соединениях оцениваются - или просто игнорируются.

**EDA Средства:**  
Статический временной анализ — например, *Prime Time*.

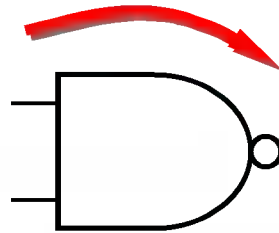
Общая задержка

**Временные требования удовлетворены!**

- Данные **поступают** на вход D FF[0] с **запасом** по времени.
- В отчете указан положительный резерв времени **1.65 – 1.22**, или **0.43 нс**.
- Однако, это через чур **ОПТИМИСТИЧНО** - мы игнорировали задержки в **соединениях!**

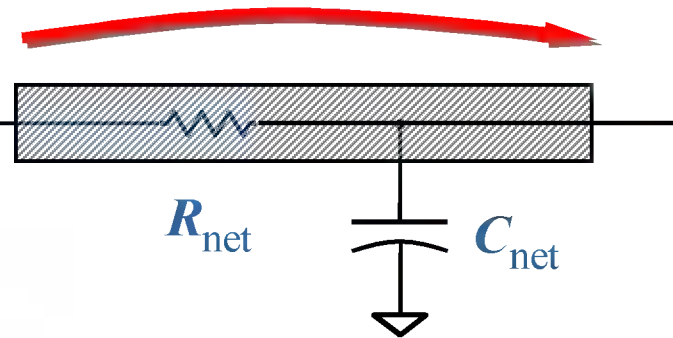
# Влияние задержек в соединениях 2-27

Задержка  
элемента  
( $\tau_{gate}$ )



Задержка в соединении

$$\tau_{wire} \cong R_{net} C_{net}$$



→ **Ключевой принцип:**

Задержки в сложных соединениях сложнее контролировать, чем задержки в элементах.

$$\tau_{path} \cong \tau_{gate} + \tau_{wire}$$

**Задержка пути:**  
(Элемент+соединение)

- Задержка элемента **уменьшается** при уменьшении длины  $l$  канала транзистора.
- Но проводники имеют большее сопротивление, если их сечение мало.
- Поэтому временной анализ **после трассировки** может показать **нарушения!**

# Список соединений SERIAL4 2-28

Последовательно-  
параллельный  
преобразователь:  
Список соединений

Внутренние  
сигналы

Двухвходовый  
мультиплексор

```
SERIAL4.vg
1  module SERIAL4 (TX,MAC_DATA,MAX,
2     LOAD_SHFT, CLK
3  );
4     output TX;
5     input [3:0] MAC_DATA;
6     input LOAD_SHFT, MAX, CLK;
7     wire N3, N2, N1, N0, FF3_reg;
8     wire FF2_reg, FF1_reg, FF0_reg;
9
10    SDN_FDPQ_1 FF3 (.Q(FF3_reg),
11        .D(N3), .CP(CLK)
12    );
21    SDN_MUX2_1 MUX3 (.Y(N3),
22        .A(MAC_DATA[3]), .B(1'b1)
23    );
32    SDN_OR2A1 OR1 (.Y(TX),
33        .A(FF0_reg), .B(MAX)
34    );
35  endmodule
```

список соединений  
Verilog:

Текстовое описание  
экземпляров логических  
элементов и триггеров и  
соединений между ними.

D-триггер  
(передний фронт)

Двухвходовый  
элемент ИЛИ

Список соединений описывает **соединение** стандартных ячеек.

---

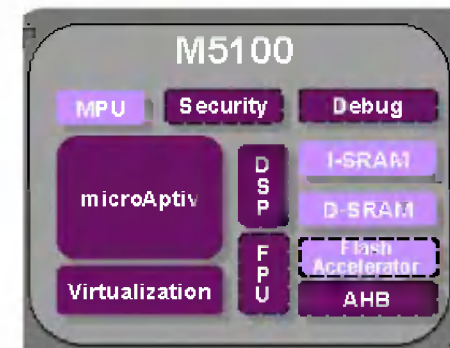
# Основные концепции:

2-29

1. Средства синтеза читают описание HDL и компилируют его в код схемы для конкретной технологии.
2. Синтезируемое описание ограничено **RTL** подмножеством языка - оно не может быть слишком абстрактно.
3. Средства синтеза не только компилируют схему, но и оптимизируют ее временные характеристики, площадь, потребляемую мощность.
4. Статический временной анализ - встроенный или автономный - указывает наихудший путь и обнаруживает любые нарушения.
5. Нарушения временных характеристик можно исправить разными способами, но часто нужно идти на компромисс по потребляемой мощности и/или площади.
6. Существенные проблемы могут быть связаны со временными нарушениями из-за наличия неожиданного долгих **соединений**.

## Обзор IP-ядер

- Временная шкала интеграции IP
- Первые конвейерные архитектуры MIPS
- Современная эволюция ядер MIPS
- Компромисс площадь/время





# Временная шкала интеграции 2-31

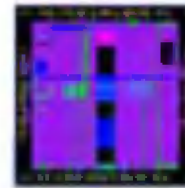
## IP

Внешние  
IP-блоки



Окончание  
разработки  
RTL кода

ARM



Предупреждение:  
График поставщиков IP-блоков должен быть согласован с контрольными точками проекта — такими, как окончание разработки RTL кода.

1. Спецификация

2. Проектирование

3. Верификация

4. Имплементация

5. Производство

6. Отбраковка

Цифровые IP-блоки:

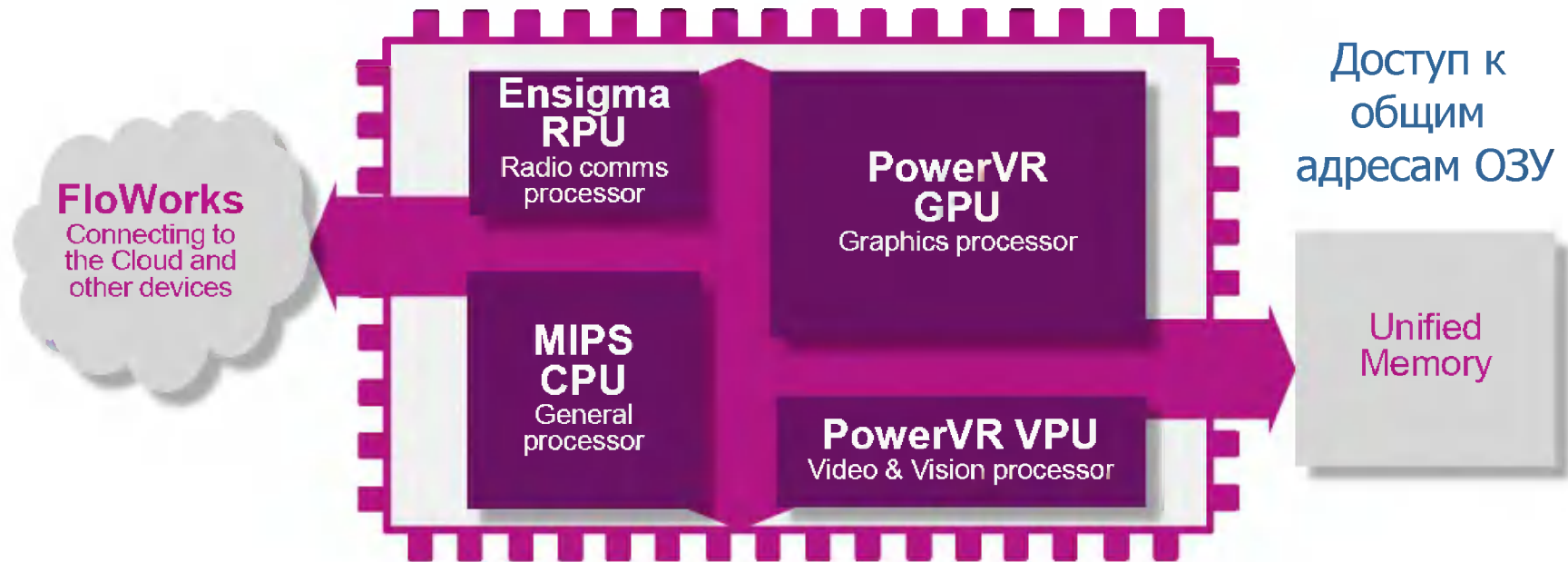
- Полученные извне
- Спроектированные внутри
- Модифицированные пользователем

Топологические  
макроячейки

- При проектировании на основе блоков интеллектуальной собственности (IP), поставщики системы на кристалле (SOC) повторно используют коммерческие IP-блоки.
- Или существующие IP-блоки из предыдущих проектов.
- Или новые функциональные блоки, которые созданы внутренними командами разработчиков.
- Поставщики IP-блоков, такие как ARM, тесно сотрудничают со своими основными клиентами.

# Ядра Imagination MIPS

2-32

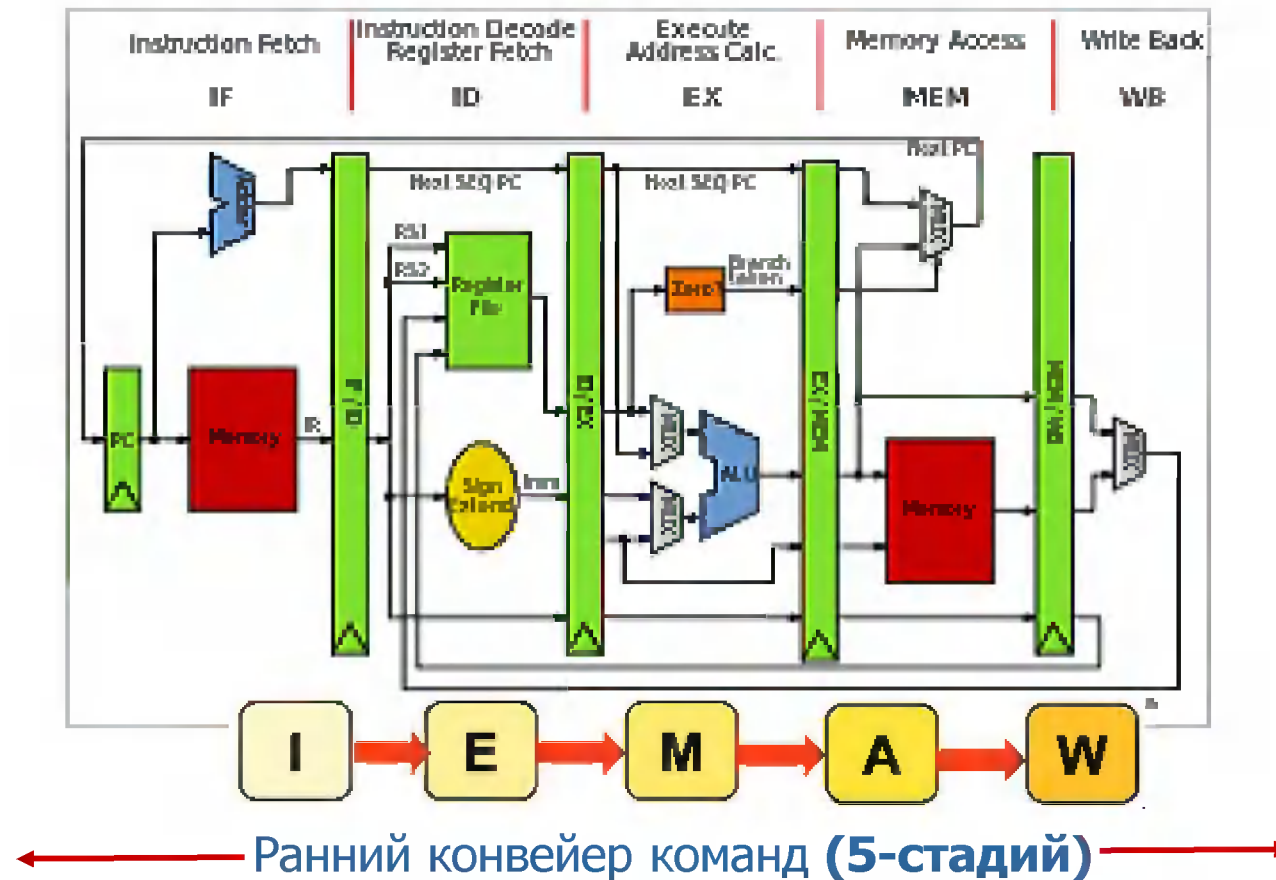


- Чем **различаются** различные проекты SOC, которые повторно используют IP-блоки?
- Команда разработчиков SOC должна сформировать правильное предложение продукта.
- Реализовать SOC с правильным **сочетанием**: усилий, цены, потребляемой мощности.
- Уделить внимание обеспечению эксплуатационных характеристик и поддержке заказчиков.

# Первые конвейерные архитектуры MIPS

2-33

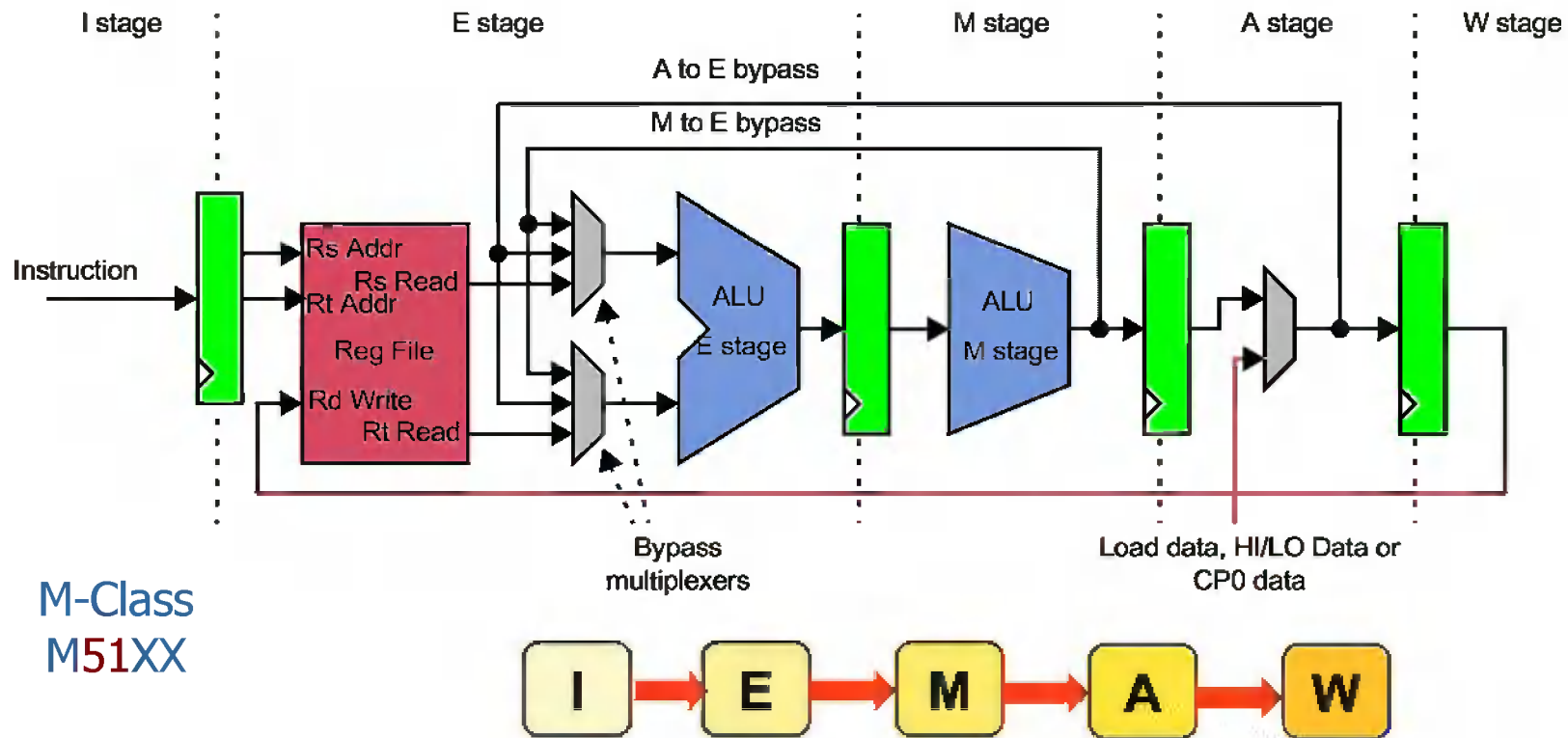
Устаревшая  
архитектура  
R2000



- Тонкая архитектура, т.е. **загрузка-сохранение** (регистр-регистр).
- Арифметические и логические операции выполняются между регистрами процессора.
- **Конвейер** постоянно развивался, становился глубже, было добавлено множество возможностей.

# Пример младшего ядра M51XX

# 2-34

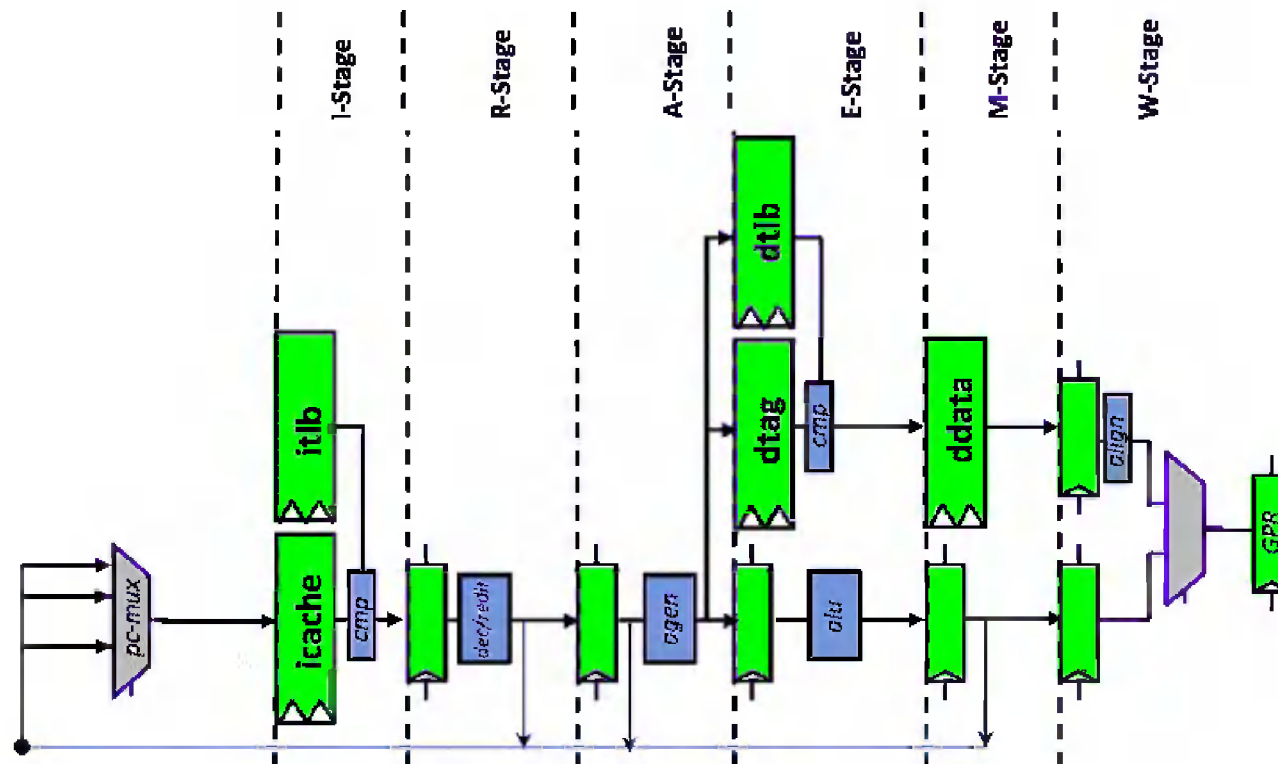


- Единственный конвейер с последовательным выполнением команд—глубина та же, но **задержки сбалансированы**.
- Варианты: Алгоритмы цифровой обработки с фиксированной точкой; векторная обработки и т.д.
- Ядро характеризуется минимальной площадью и энергопотреблением.

# Пример младшего ядра M62XX

2-35

M-Class  
M62XX

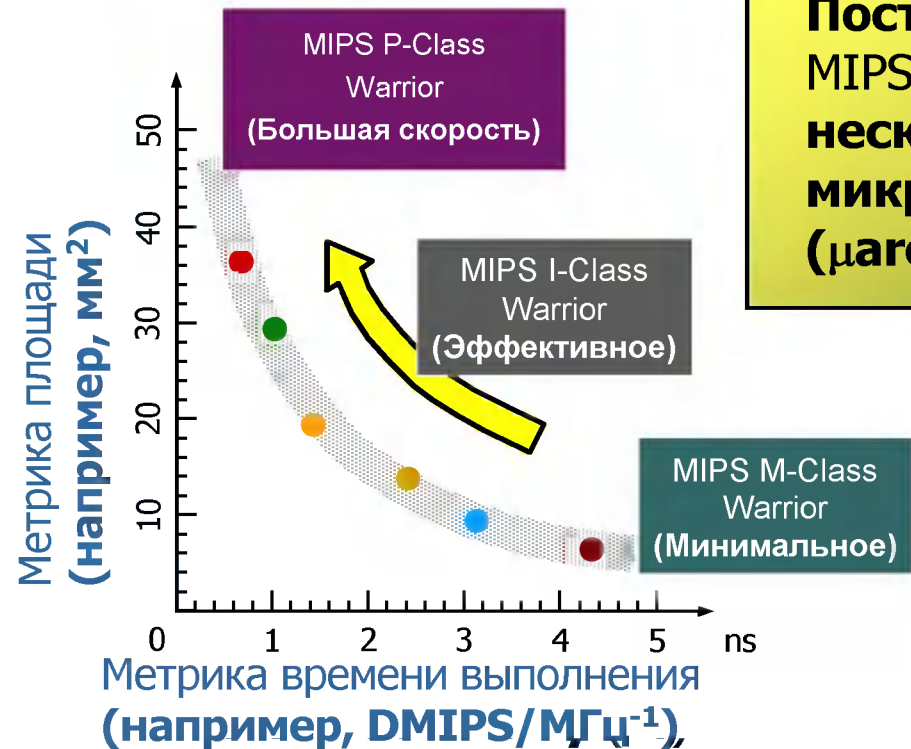


- Конвейер более **глубокий**, чем в M51XX—6 стадий, вместо 4 или 5.
- Компромисс:  $f_{max}$  больше в **1.3 раз**, но отношение DMIPS/МГц немного меньше.
- В целом, M62XX обеспечивает **более высокую производительность** чем M51XX.

# Компромисс площадь/время 2-36

## I-Class:

- Выполняется Linux
- Хабы для интернета вещей (IoT)
- Модемы LTE
- Помощник водителя автомобиля



← Ключевой принцип:  
**Поставщики IP—ARM, MIPS—предлагают несколько линий микроархитектуры (march).**

## M-Class:

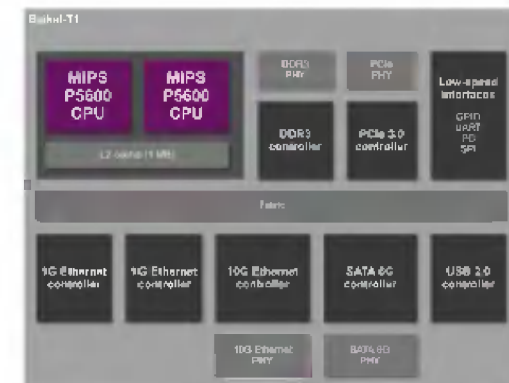
- Простые микроконтроллеры
- Встроенные системы
- Датчики (для интернета вещей)

- Младшие: оптимизированы - минимальная площадь, минимальная потребляемая мощность.
- Среднего уровня: достаточная производительность; меньшая потребляемая мощность. Эффективные!
- Высокого уровня: терминалы, планшеты, рынок минисерверов. Быстрые.

## IP-блоки средне-высокого уровня:

- Примеры среднего уровня
- Аппаратная многопоточность
- Максимальная производительность
- Многоядерные кластеры
- Когерентность кэш-памяти
- Конфигурируемые  $\mu$ Ps

SOC Байкал T1



# Среднего уровня: interAptiv 2-38



Модемы

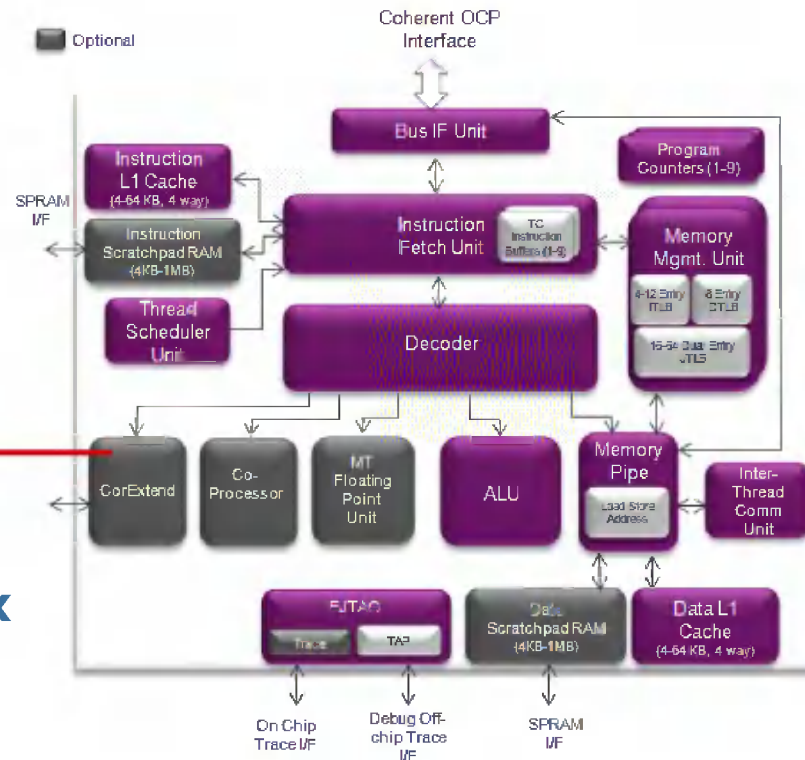


Камеры



Устройства хранения данных

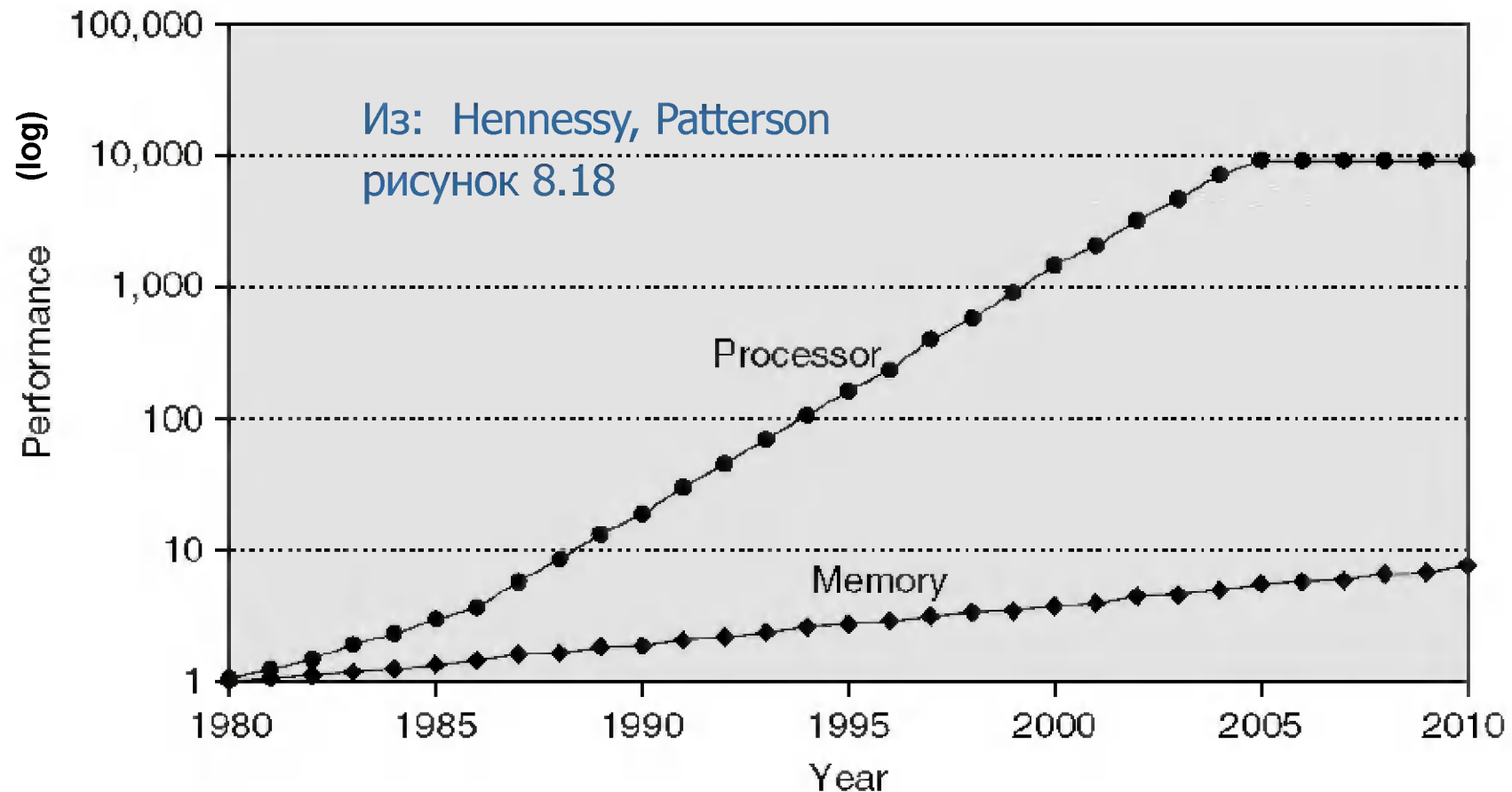
Технология конфигурируемых процессоров



- Оптимальный выбор для многих микросхем, производительность достаточна для выполнения Linux.
- Потребляемая мощность достаточно мала для применения в мобильных и переносных устройствах и в интернете вещей.
- Потребляемая мощность уменьшается путем **остановки тактового сигнала**, например, для латентных запросов шины.
- Девятистадийный конвейер с последовательным выполнением команд и аппаратной многопоточностью.



# Побочный эффект закона Мура 2-39

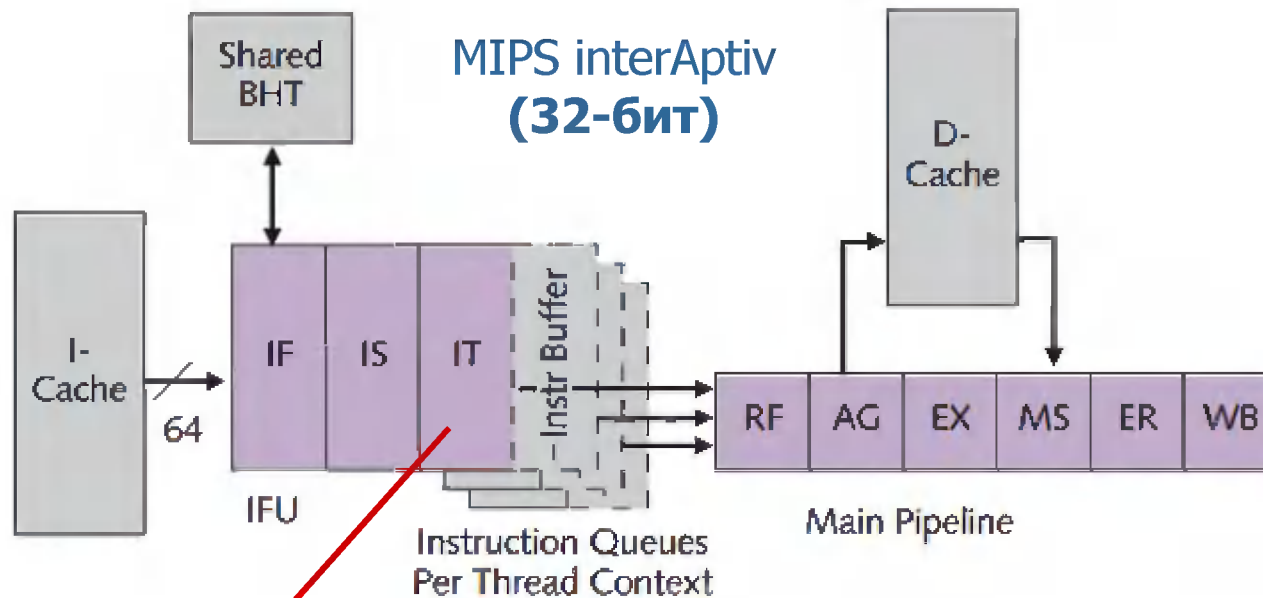


- Существует тенденция ускоренного роста скорости арифметико-логических устройств по сравнению с ростом скорости доступа к памяти.
- При **промахе** доступа к кэш-памяти процессор может простаивать  $10^2$  циклов.
- Процессор может потерять **50%** своего времени на такие паузы.

# Аппаратная многопоточность

2-40

(1/3)



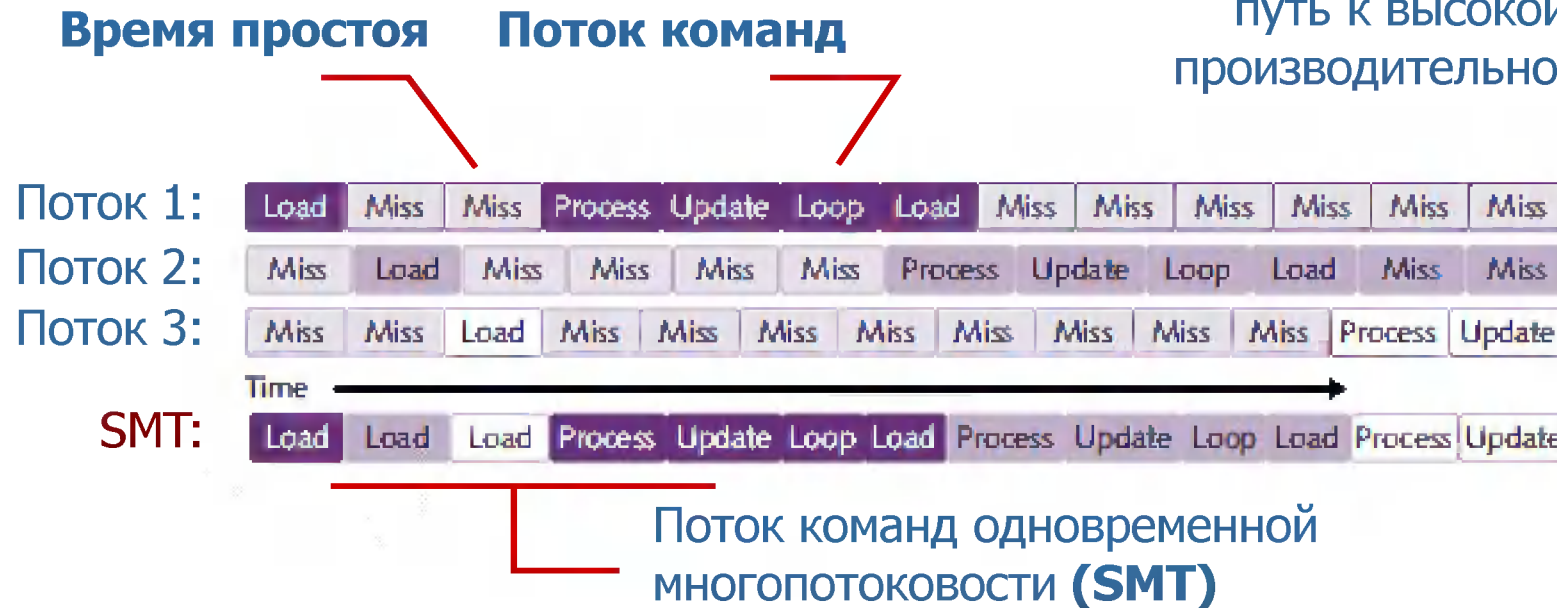
Третья стадия выборки команды,  
Instruction-Fetch Third (**Стадия IT**)

- Ядро interAptiv имеет чуть более глубокий конвейер для поддержки **SMT (одновременной многопоточности)**.
- Дополнительная стадия IT **назначает контекст** для выбранной только что команды.
- Таким образом, отдельные потоки ссылаются на **корректные** регистры, состояния.

# Аппаратная многопоточность (2/3)

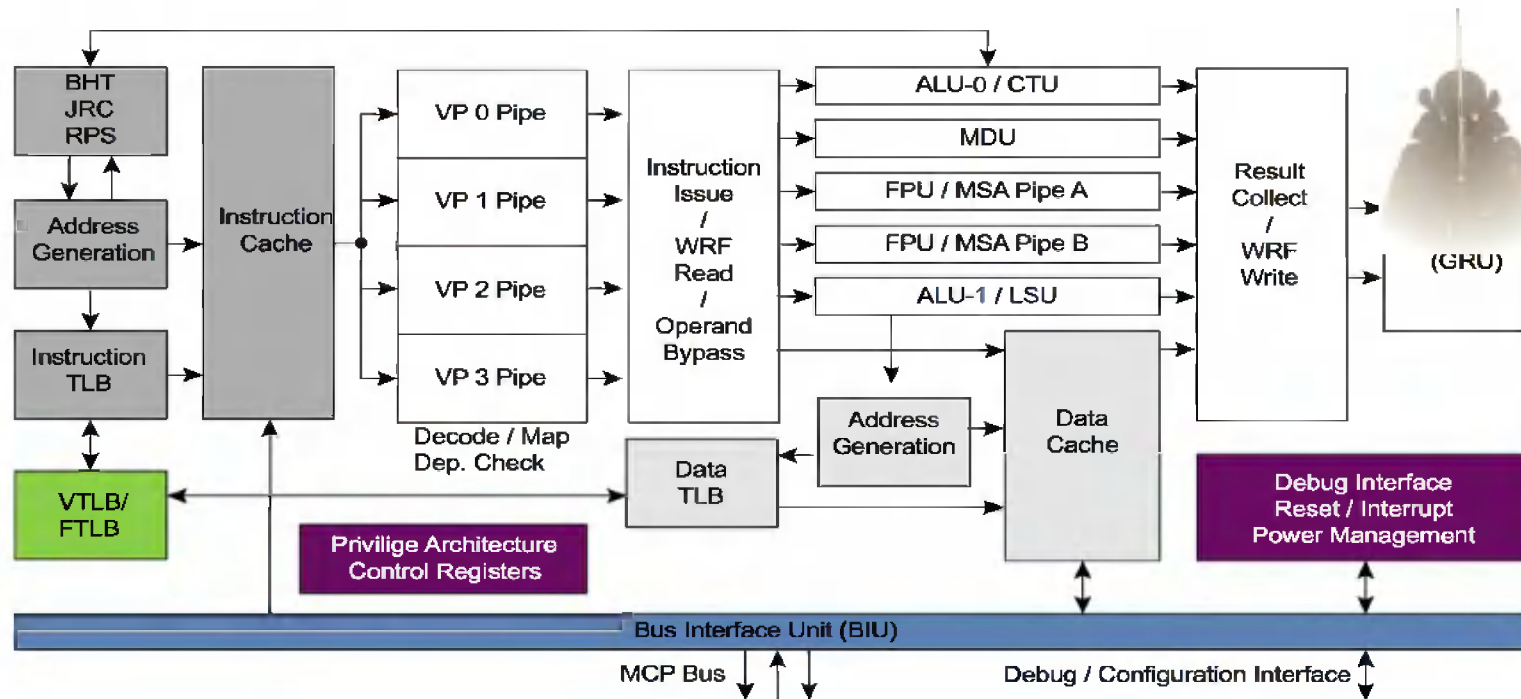
## 2-41

Энергоэффективный  
путь к высокой  
производительности



- При промахе кэш-памяти время простоя используется для выполнения **другого потока**.
- Для аппаратной реализации многопоточности, каждый поток имеет собственный счетчик команд, регистры общего назначения.
- Возможность **мгновенного** (за один цикл) переключения контекста—основа SMT.
- В отличие от программной реализации, отсутствует обработка прерываний и сохранение контекста.

# Аппаратная многопоточность (3/3) 2-42



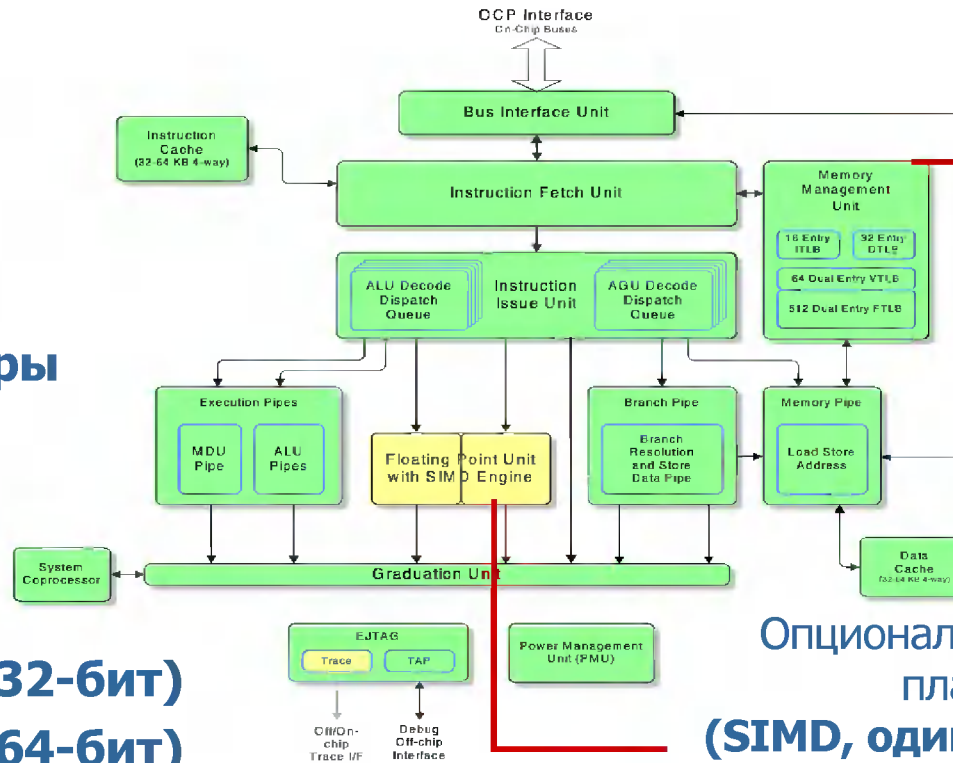
- Как реализовать SMT без нарушения ограничений по потребляемой мощности?
- Поддерживаемое **количество потоков** ( $\leq$  глубина) является критическим выбором.
- 64-Разрядное ядро среднего уровня I-class I6400 **Samurai** использует четыре конвейера.
- Таким образом, обеспечивается одновременная многопоточность с **четырьмя** потоками.

# Максимальная производительность

# 2-43



## Микросерверы



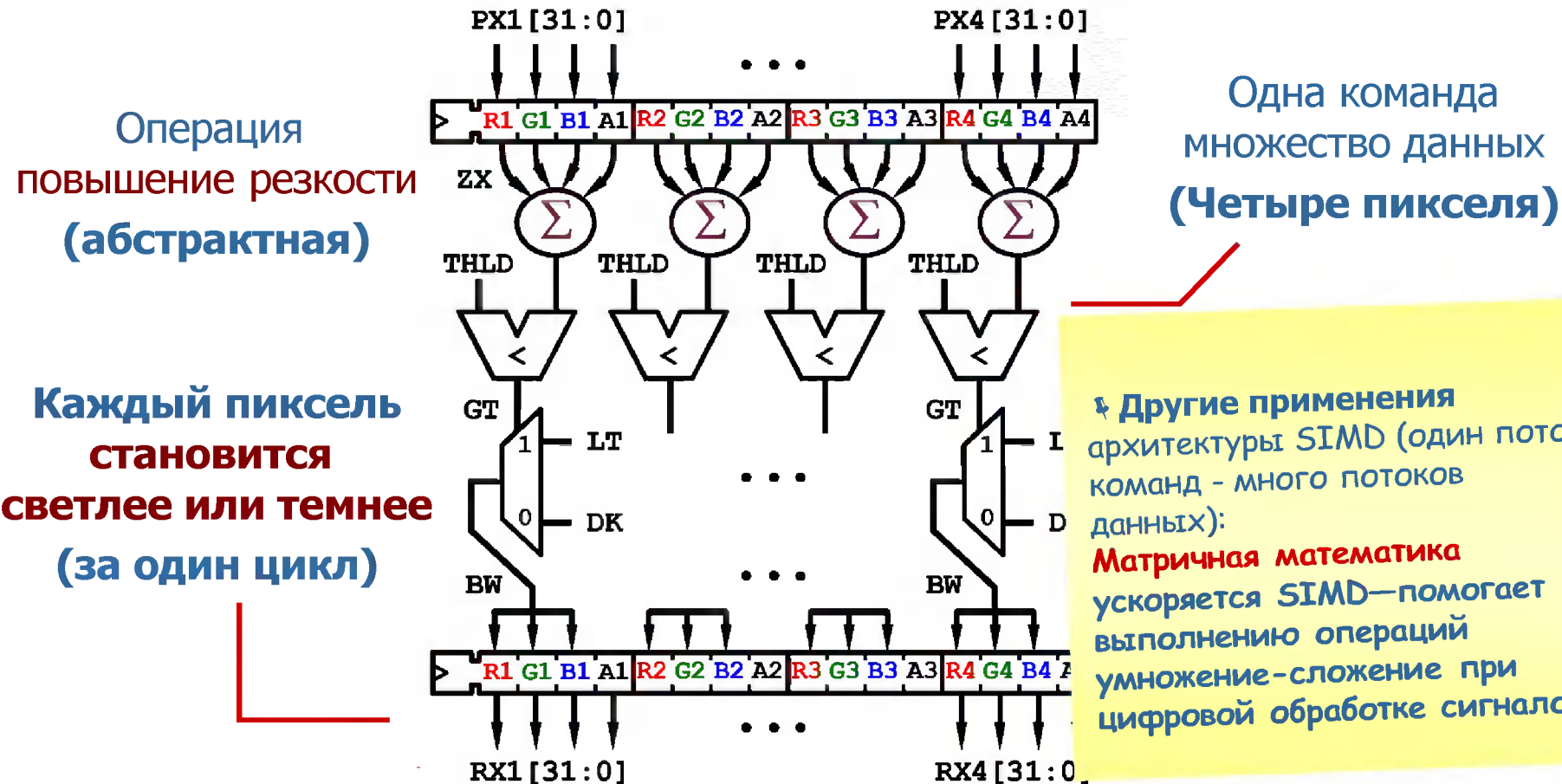
Модуль управления памятью/буфер ассоциативной трансляции (Поддержка виртуальной памяти ОС)

Опционально блок арифметики с плавающей точкой (SIMD, один поток команд - много потоков данных)

MIPS P5600 (32-бит)  
MIPS P6600 (64-бит)

- **Наивысшая** однопоточковая производительность. Идеально подходит для приложений Java.
- Глубокий конвейер целочисленной арифметики; суперскалярная архитектура с внеочередным выполнением.
- Возможность внеочередного выполнения команд **минимизирует** приостановки.

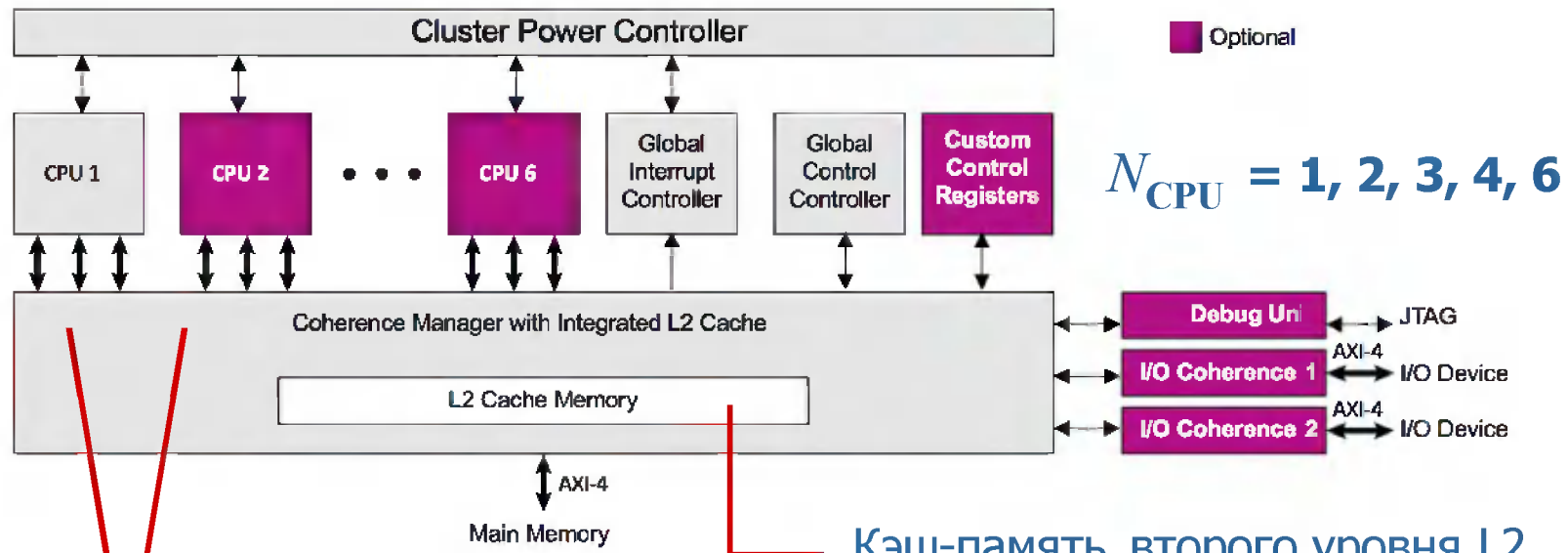
# Ускорение для мультимедийных приложений 2-44



- SIMD позволяет **параллельно** выполнять сложные арифметические операции.
- Каждая команда выполняется **синхронно** на всех параллельных блоках данных.

# Многоядерные кластеры

2-45



Отдельные кэш-памяти  
первого уровня L1

Интерфейс  
шины AXI

Кэш-память второго уровня L2  
(общая для всех)

- Включает  $N$  процессорных ядер, каждое со своей **отдельной** кэш-памятью первого уровня L1.
- Менеджер когерентности гарантирует отброс неактуальных данных.
- Поддерживает **вмешательство** в восстановление линии. Суперкластерные возможности.

# Когерентность кэш-памяти 2-46

Протокол **MESI**

Используется во всех многоядерных кластерах MIPS (средний-высокий уровень)

Неактуальное состояние: **Записывающий процессор выдает сигнал недоверности.**



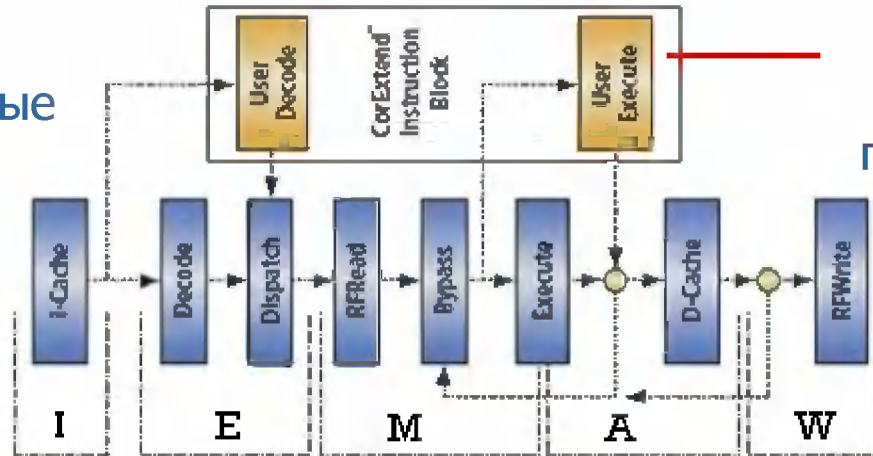
✦ **Аппаратная реализация конечного автомата:**  
Один конечный автомат на каждую кэш-память.  
Входные данные поступают с собственного процессора или шины.

- Области кэш-памяти, которые совместно используются  $N$  ядрами, должны быть **консистентными**.
- **Совместно** используемые блоки кэш-памяти на являются проблемой при **чтении**.
- Однако, **запись** слова требует более **исключительного** доступа.
- Встроенный конечный автомат **упрощает** задачу программиста.



# Конфигурируемая обработка 2-47

Cor Extend  
Команды, определенные  
пользователем



Параллельный блок  
команд,  
определенных  
пользователем (UDI)  
(тесно  
интегрирован)

Команды, определенные  
пользователем  
(UDI)

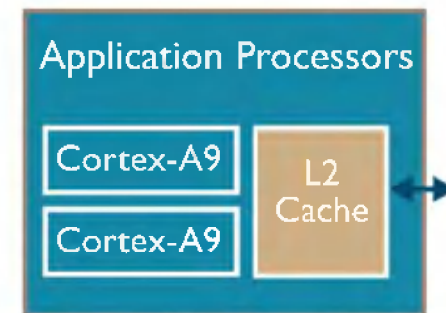


- Расширяет стандартный набор MIPS командами, **определенными пользователем**.
- Работают с **регистрами** общего назначения, промежуточными данными и т.д.
- Тесно связан с блоком исполнения команд четко определенными сигналами.
- Несколько специализированных команда могут **значительно повысить скорость**.

## Характеристики IP-блоков:

- Программные/аппаратные IP-блоки
- IP-блоки в топологическом формате
- Топологические IP-ядра ARM
- Протоколы шины ARM

Типичные  
IP-блоки



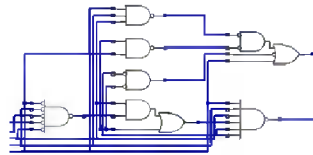
# Форматы IP-ядер (1/2)

2-49

Программные  
IP-блоки  
(RTL код)

```
1 /* Verilog code:
2 * Four-bit serializer,
3 * with jabber control.
4 */
5 module SERIAL4(
6   output wire TX, //serial.
7   input wire [3:0] MAC_DATA,
8   input wire LOAD_SHFT, MAX, CLK
9 );
10   reg [3:0] FF; //Four flops.
11
12   always @(posedge CLK)
13     if (LOAD_SHFT)
14       FF <- MAC_DATA; //Load data.
15     else
16       FF <- {1'b1,FF[3:1]}; //Shift.
17
18   assign TX = FF[0] MAX; //ok.
19
20 endmodule
```

Аппаратные  
IP-блоки  
(Список  
соединений)



Аспекты:

- **Модифицируемый пользователем RTL код**
- Синтезируются вместе с прочим RTL-кодом.

Аспекты:

- Можно подстраивать
- Могут синтезироваться по требованиям пользователя
- Может быть запутанным RTL— нечитабельным, защищенным!

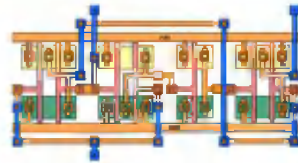
↳ Экономика IP:  
Программные IP-блоки **можно модифицировать** — но поставщики защищают свои продукты.

- В целом, существует тенденция к наибольшему распространению **программных IP-блоков**.
- Передается **RTL код** (возможно зашифрованный или сгенерированный).
- Передача читаемого RTL более распространена для **младших ядер**.
- Последователи ядер контролируют процесс логического синтеза, ход физического проектирования.

# Форматы IP-ядер (2/2)

2-50

Топологические  
IP-блоки  
(Топологический  
чертеж)



Аспекты:

- Оптимизированы под одну технологию
- Пользователь не может изменить
- Физически спроектированы поставщиком.

Поставляется:

- Файл **GDSII** топологии.
- Список соединений Verilog:
- Файлы .lef, .lib.
- Список соединений LVS.
- Модели для моделирования.
- Тестовые векторы для автоматической тестирующей установки



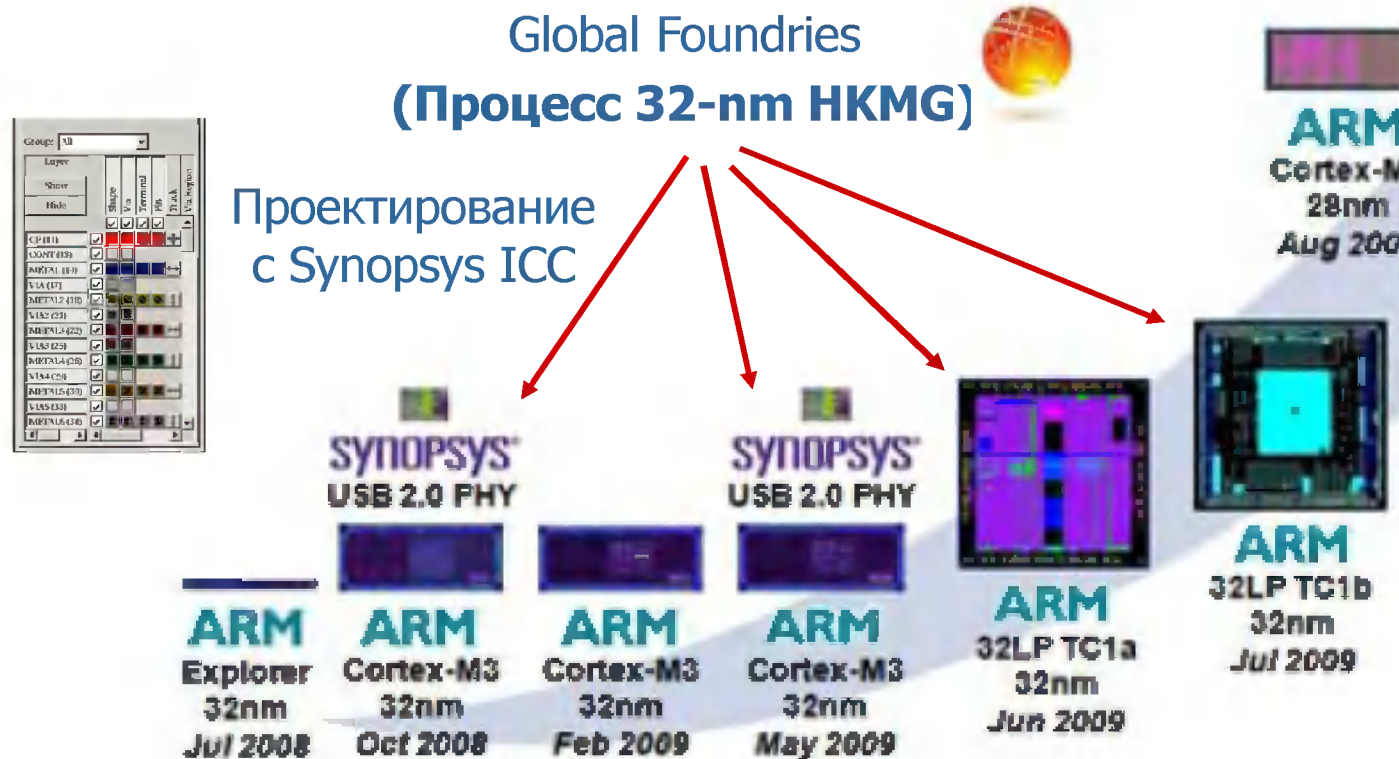
↳ Экономика IP:

Топологические IP-блоки могут быть в 10 дешевле из-за сложности изменения их пользователем.

- Топологические IP-блоки невозможно изменить, они, как правило, самые дешевые.
- Распространенный формат для **устаревших** ядер, поставляются на условия «как есть», без гарантий.
- Аналоговые IP-блоки или высокоскоростные цифровые интерфейсы всегда **ТОПОЛОГИЧЕСКИЕ**.

# Пример ARM: Топологические IP-блоки

2-51



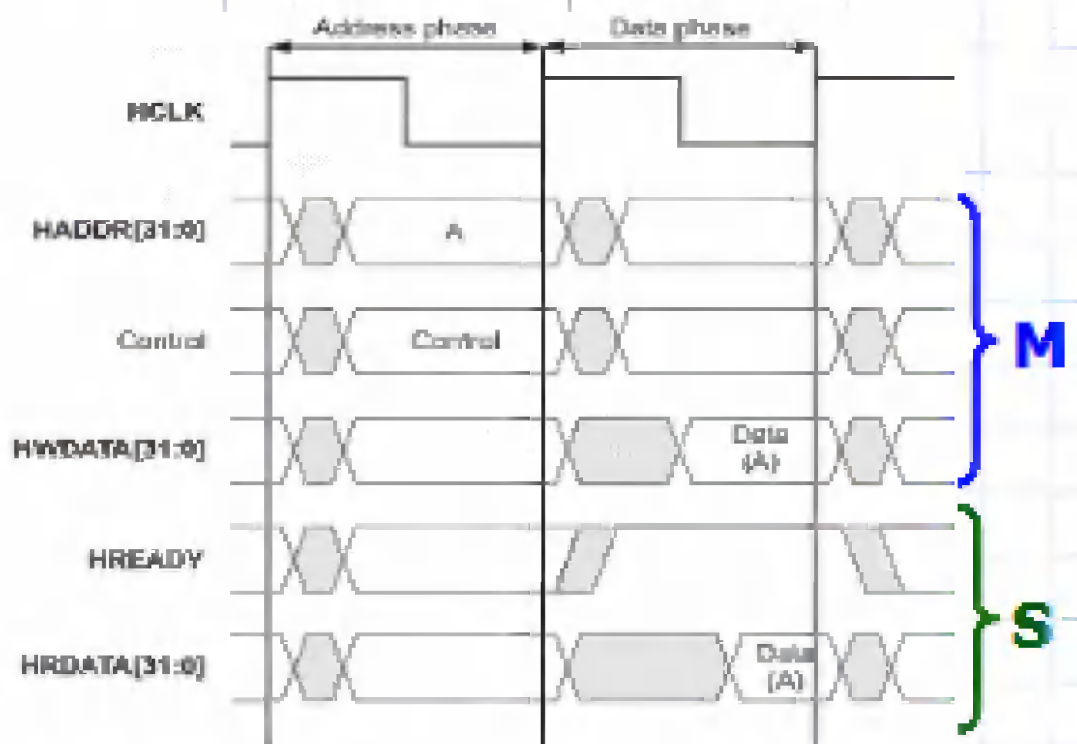
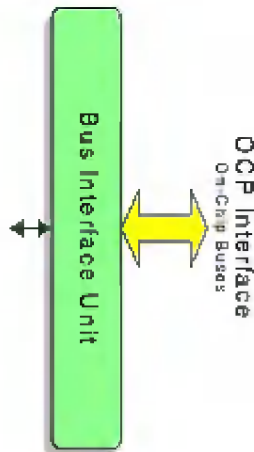
- Мягкий физический слой может каждый раз создавать различные файлы GDSII.
- Жесткий физический слой имеет согласованные задержки, меньшее дрожание фазы и ее разброс.
- Топологические IP-блоки могут разрабатываться одновременно с целевым технологическим процессом.

# Пример: Протокол шины

2-52

Протокол АНВ

Современная шина  
для микроконтроллеров  
Архитектуры (АМВА)



- **AMBA** компании ARM является открытым стандартом для внутренних соединений на кристалле.
- Соединяет и управляет функциональными блоками в микросхемах систем на кристалле.
- Таким образом, АНВ, АХІ, и т.д. работают **как** с ядрами ARM, **так** и MIPS.
- Шины работают параллельно с **целочисленным конвейером** MIPS.

---

# Прочие проблемы

# 2-53

При приобретении и использовании IP-блоков могут возникнуть следующие вопросы:

- ✦ Интегрировать IP-блоки в кристалл при синтезе RTL (программные) или при разработке топологии (топологические блоки)?
- ✦ Оценить влияние блоков на **бюджет** всего кристалла: какая часть кристалла будет занята большими блоками IP?
- ✦ Интегрировать IP-блоки в общую стратегию **DFT** (проектирование для тестирования).
- ✦ Обеспечить раннюю доступность **моделей для моделирования** - особенно для топологических блоков (для которых RTL-код недоступен).
- ✦ Корпоративные сайты для проектировщиков для доступа к IP-блокам, отслеживания обновлений и исправления ошибок.

## Процесс проектирования ВЧ/Аналоговых изделий:

- Аналоговая иерархия
- Проектирование устройств ВЧ (идеальный случай)
- Проектирование устройств ВЧ (реальные эффекты)
- Практический пример: Генератор, управляемый напряжением (VCO, ГУН)

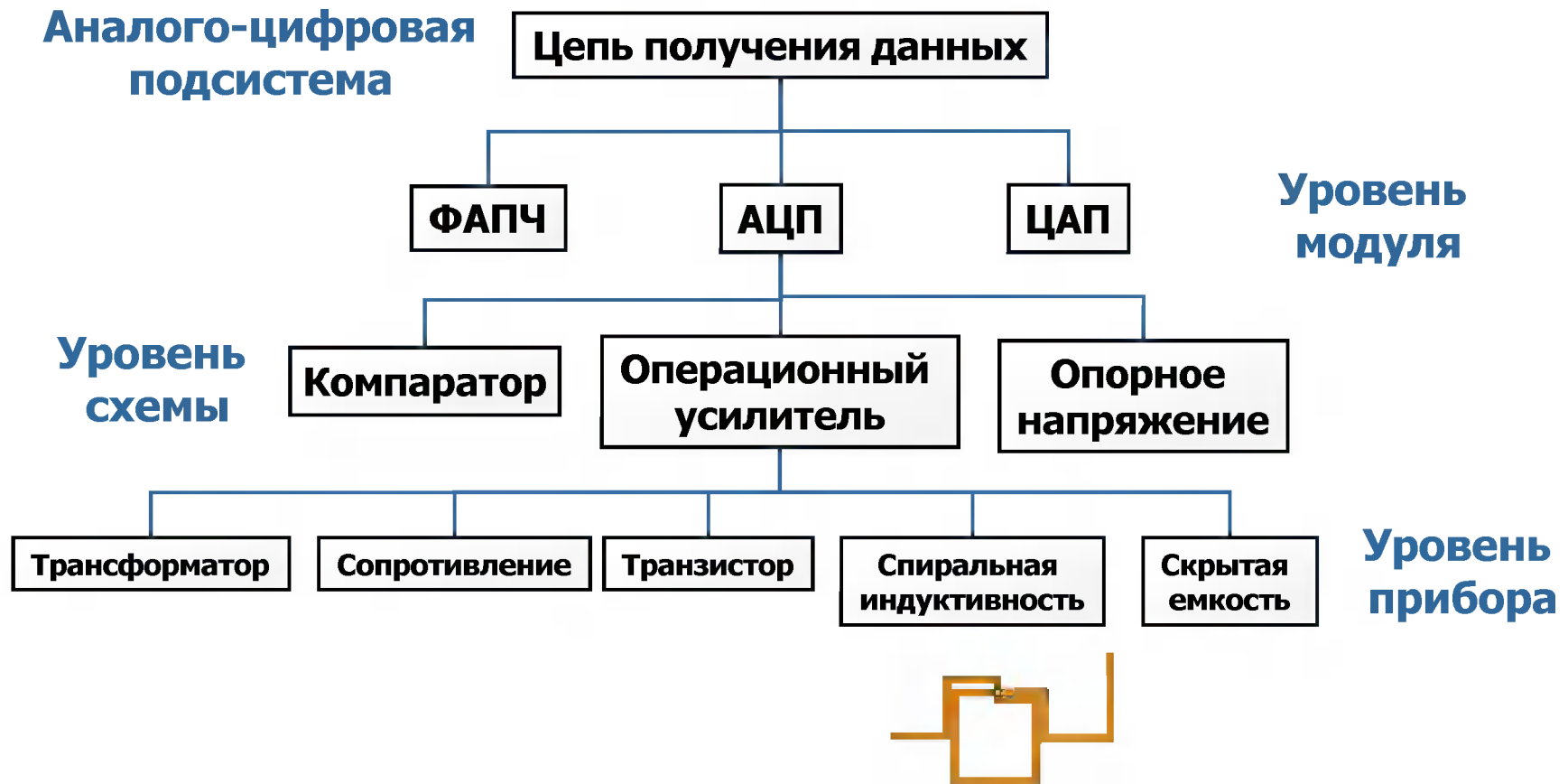
**ВЧ КМОП  
макроячейка**





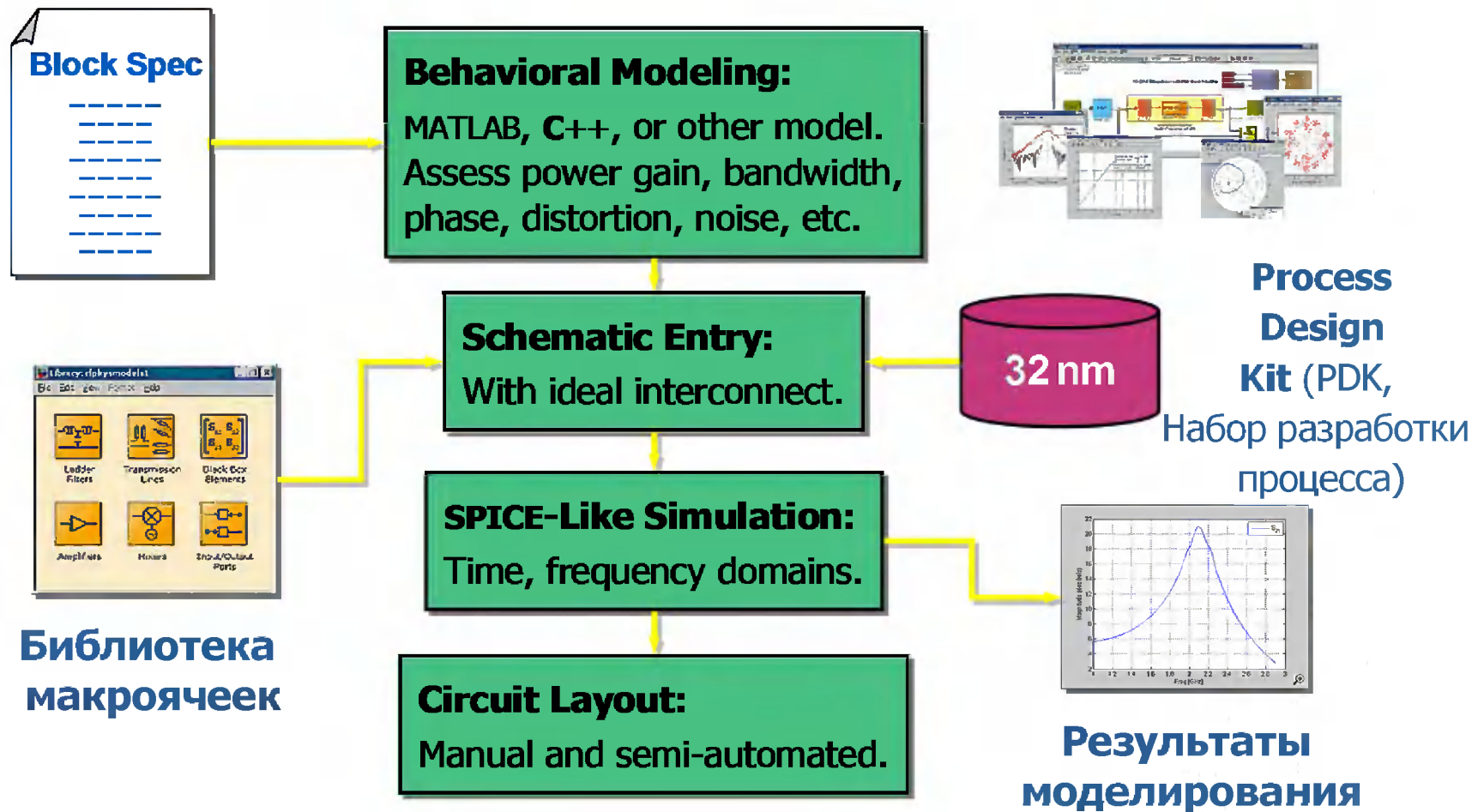
# Иерархия аналоговых блоков

2-55



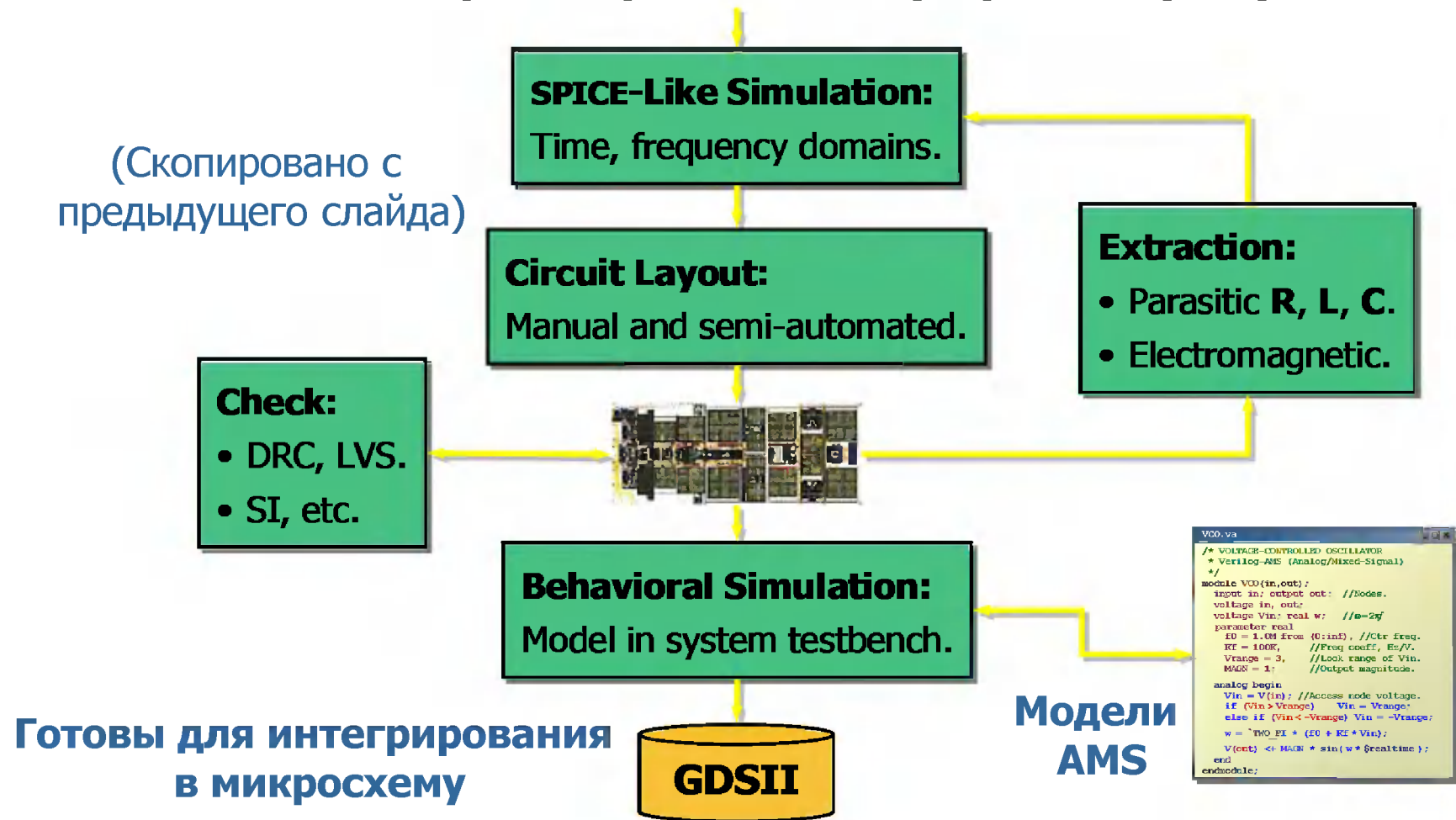
- Подобно цифровым блокам, аналоговые макроячейки могут быть иерархическими
- На нижнем уровне (листа) могут быть структуры из **R**, **L** и **C**.

# Последовательность проектирования ВЧ устройств (1/2) 2-56



- В начале проектирования детализируется поведенческая модель схемы.
- Топология, связанная с технологией, разрабатывается вручную.

# Последовательность проектирования ВЧ устройств (2/2) 2-57

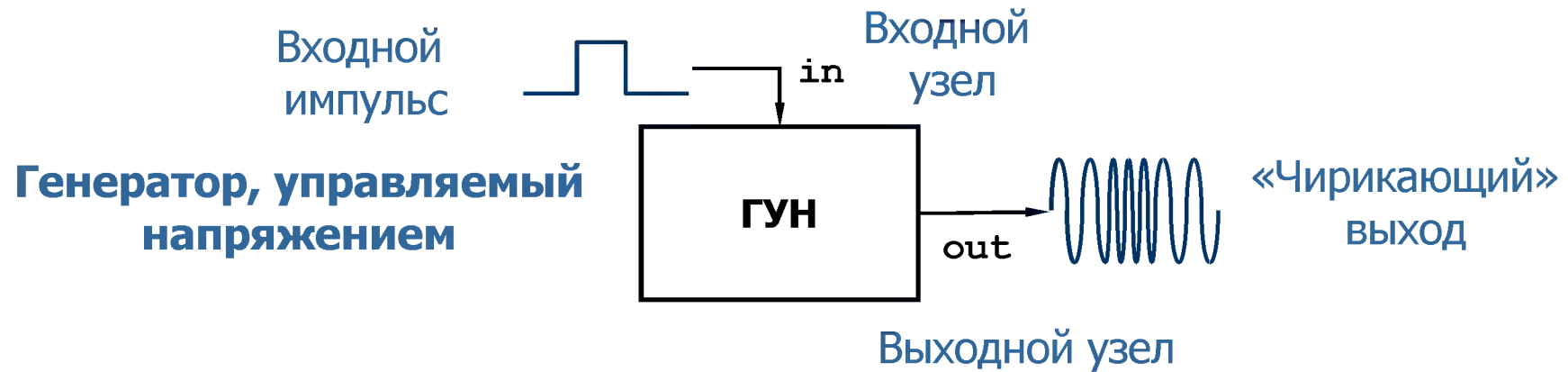


- Проектирование с учетом технологии производства извлекает паразитные эффекты из КМОП топологии.
- Набор изготовителя (PDK) моделирует аналоговые слои.

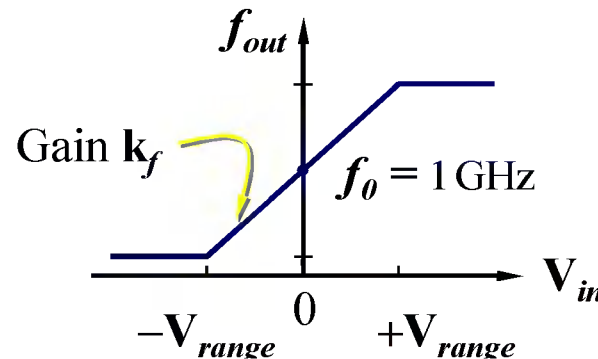
## Практический пример:

### Генератор, управляемый напряжением (VCO, ГУН) (1/5)

# 2-58



Передаточная кривая



**Поведение:**

Частота линейно изменяется с  $V_{in}$  центр в  $f_0$  с коэффициентом  $k_f$ .

- ГУН является распространенным **блоком** систем на кристалле.
- Документ PRD (Требования к изделию) определяет **параметры**, такие как диапазон напряжений и коэффициент.

# Практический пример: Генератор, управляемый напряжением (VCO, ГУН) (2/5) **2-59**

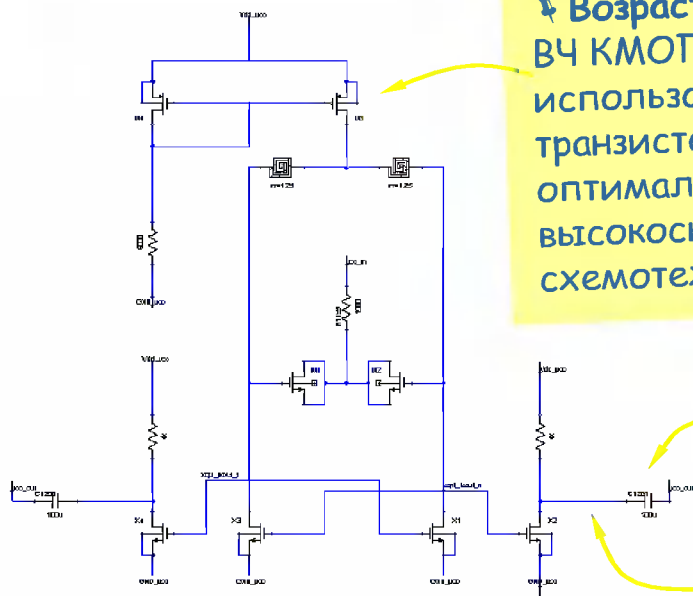
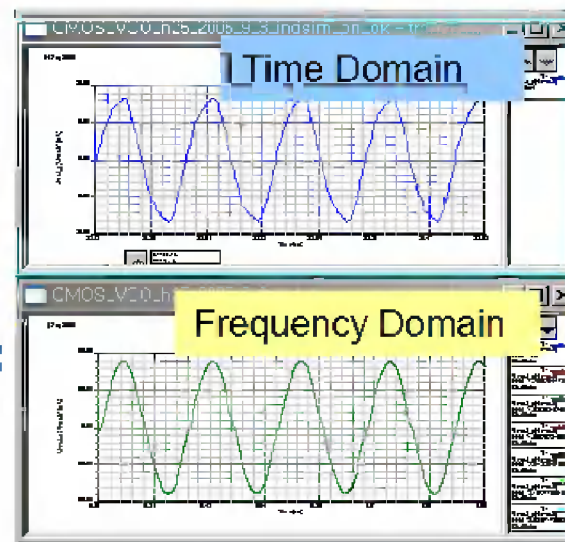


Схема (емкостный трёхточечный генератор)

↗ Возрастающие проблемы: ВЧ КМОП схемы требуют использования МОП транзисторов, которые будут оптимальными для высокоскоростной аналоговой схемотехники, а не цифровой.

Явно заданная  $C$ :  
Специально спроектирована.

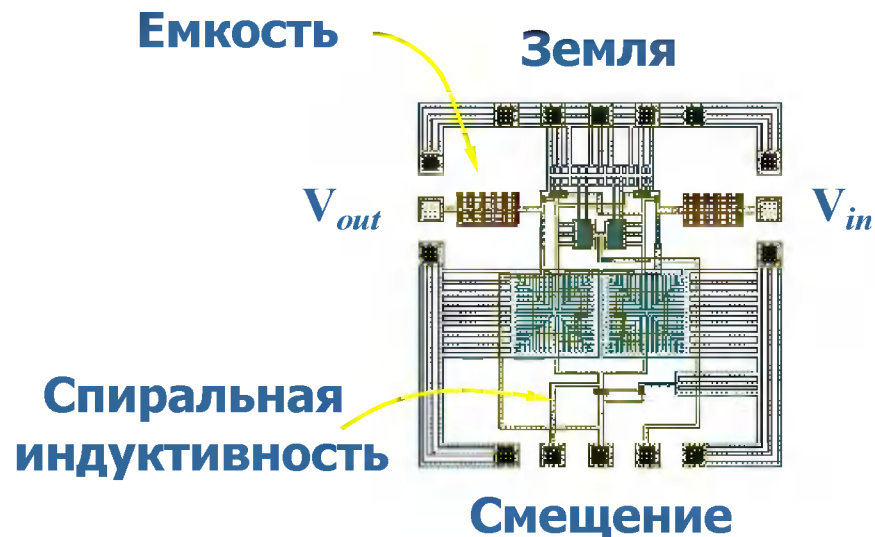
Паразитная  $C$ :  
Случайная  $C$   
должна быть оценена.



Моделирование, подобное SPICE

- Человек начинает проектирование с известного емкостного трёхточечного генератора.
- Идеальные соединения будут позднее связаны с паразитными  $R$ ,  $L$ ,  $C$ .

# Практический пример: Генератор, управляемый напряжением (VCO, ГУН) (3/5) **2-60**



## Моделирование L:

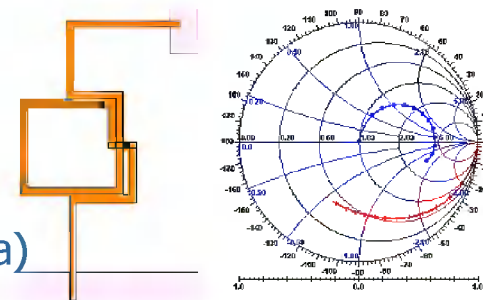
- Модели поставляет изготовитель.
- Часто толстый верхний слой металла.
- Используется программа расчета электромагнитных полей (2-D или 3-D).

## Топология схемы:

- В основном проектируется вручную.
- Полуавтоматическая трассировка.

## Индуктивность L

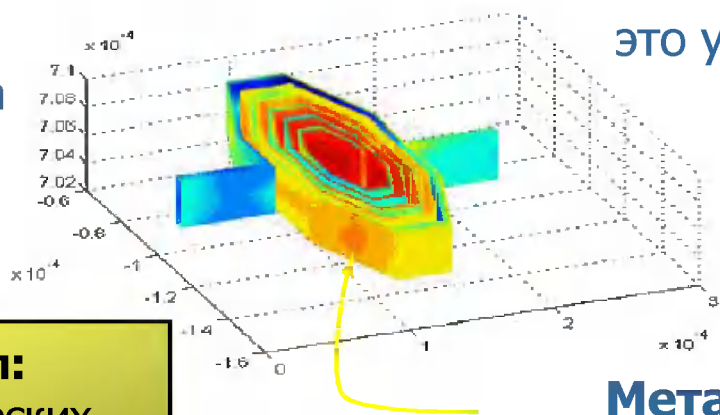
(Оптимизируется с помощью диаграмм Смита)



- Программа расчета электромагнитных полей приближенно решает уравнения **Максвелла**.
- Моделируют критическую **связь** между элементами схемы.
- Связь с **цифровой** частью микросхемы происходит через подложку.

## Практический пример: Генератор, управляемый напряжением (VCO, ГУН) (4/5)

Спиральная  
индуктивность на  
кристалле.



Металлические  
боковые станки (На 65 нм)

↔ **Ключевой принцип:**  
Уменьшение геометрических  
размеров и повышение  
частоты делает поверхностные  
эффекты более важными.

**Скин-эффект:**

Ток протекает около поверхности  
металла — увеличивается  $R$ ,  
это уменьшает добротность  $Q$ .

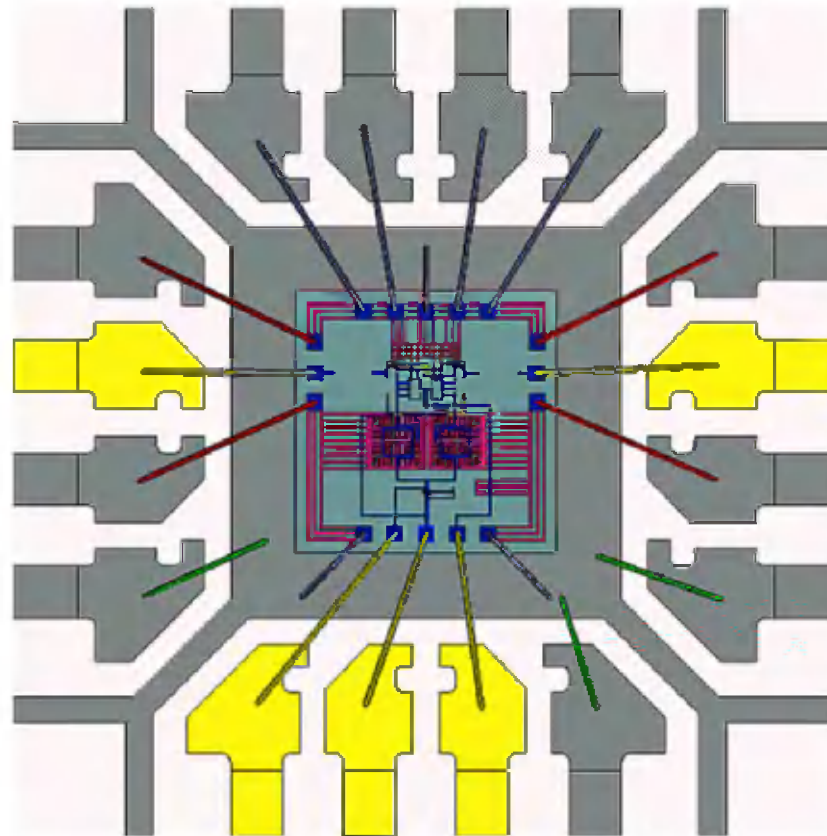
- Программа расчета электромагнитных полей учитывает скин-эффект на ВЧ.
- Высокочастотные токи концентрируются на **поверхности** металла.

Практический пример:

Генератор, управляемый напряжением (VCO, ГУН) (5/5)

2-62

Корпусированный  
кристалл



Поверхностный  
эффект:

Связывающие  
проводники добавляют  
паразитную  
индуктивность  $L$ .

Паразитная  $L$ :  
Около 1 нГн/мм,  
зависит от  $l$ ,  
угла связи и т.д.

- ВЧ проекты очень чувствительны к параметрам **корпуса**.
- Кристалл с шариками припоя на нижней стороне обеспечивают **намного меньшую  $L$** , чем золотые проволочки.



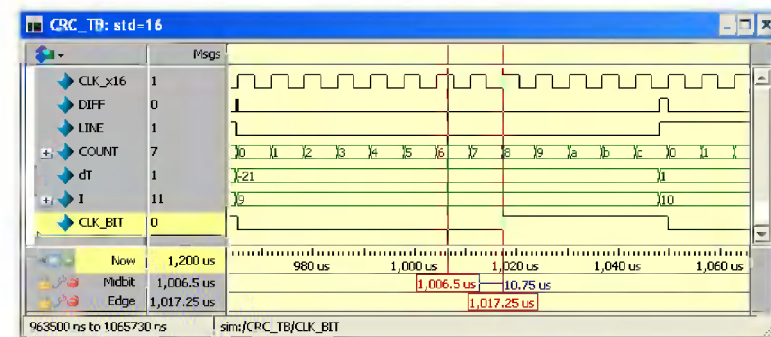
# Developing the Nanometer

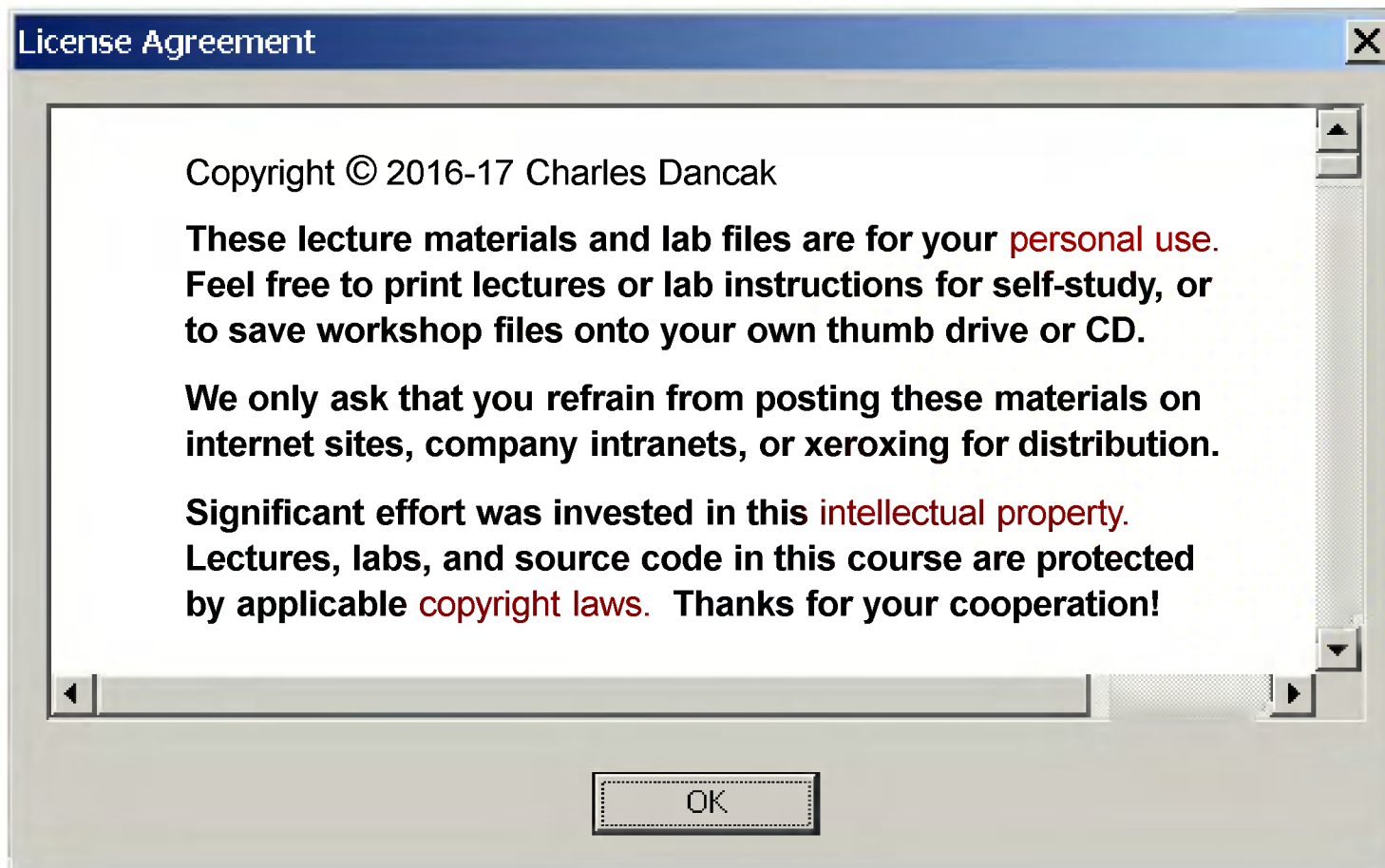
# ASIC

Specs to Silicon



## 3. Стадия верификации

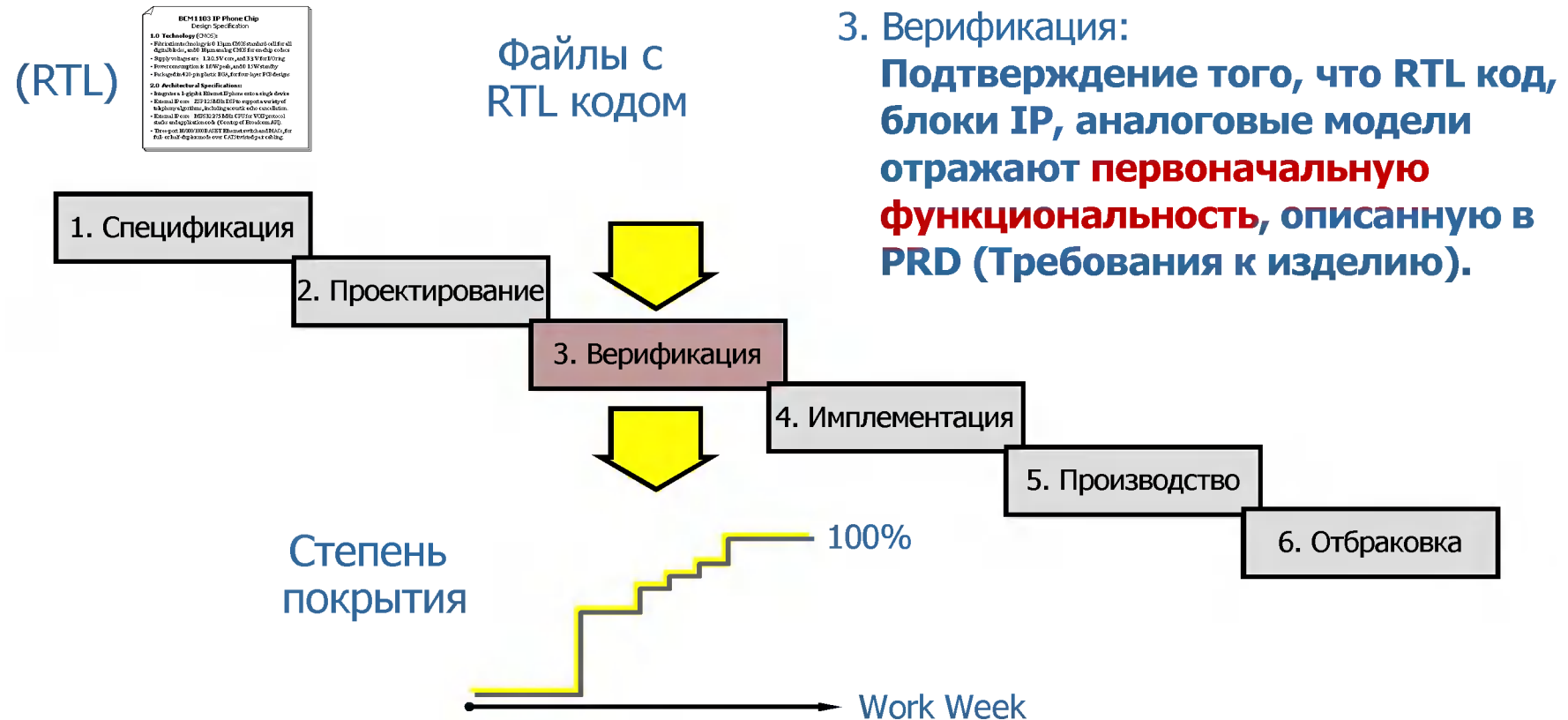




- Материалы предназначены только для **зарегистрированных студент**
- Не следует размещать их на корпоративных или любых других сайт

# Стадия верификации

3-3



- При практическом проектировании создается несколько сотен **файлов проекта**.
- Верификация проверяет соответствие кода в них первоначально определенной **функциональности**.
- Процесс **контролируется** до достижения уверенности в 100%.

# Методы верификации

3-4

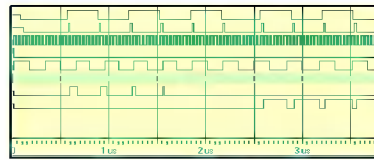
Формальная проверка  
(Подтема 3.4)

Общая стратегия  
верификации

```
SRIM4.v
1 /* RTL Verilog Code:
2 * Four-bit serialiser,
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 */
21
22 module SRIM4
23     output wire Y, //Serial
24     input wire [3:0] RSC_DATA,
25     input wire LOAD_SHIFT, RSC, CLK
26 );
27     reg [3:0] RT; //Four rings.
28
29     always @(posedge CLK)
30     if (LOAD_SHIFT) //Load data.
31         RT <- RSC_DATA;
32     else
33         RT <- (RT<<1) | RT[3]; //Shift.
34
35     assign TX = RT[0] | RSC; //OR.
36 endmodule
```

RTL код

Традиционное  
моделирование  
(Подтема 3.2)



Данные направленного  
тестирования

CRV (ограниченная  
случайная  
верификация) с  
функциональным  
покрытием



(Подтема 3.3)

Утверждения  
(сообщения)  
(SystemVerilog)

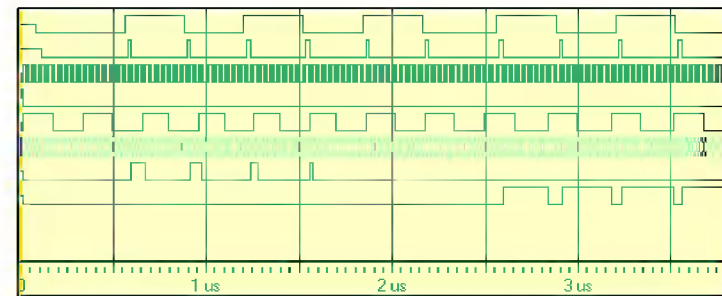
Аппаратная эмуляция  
(Подтема 3.5)



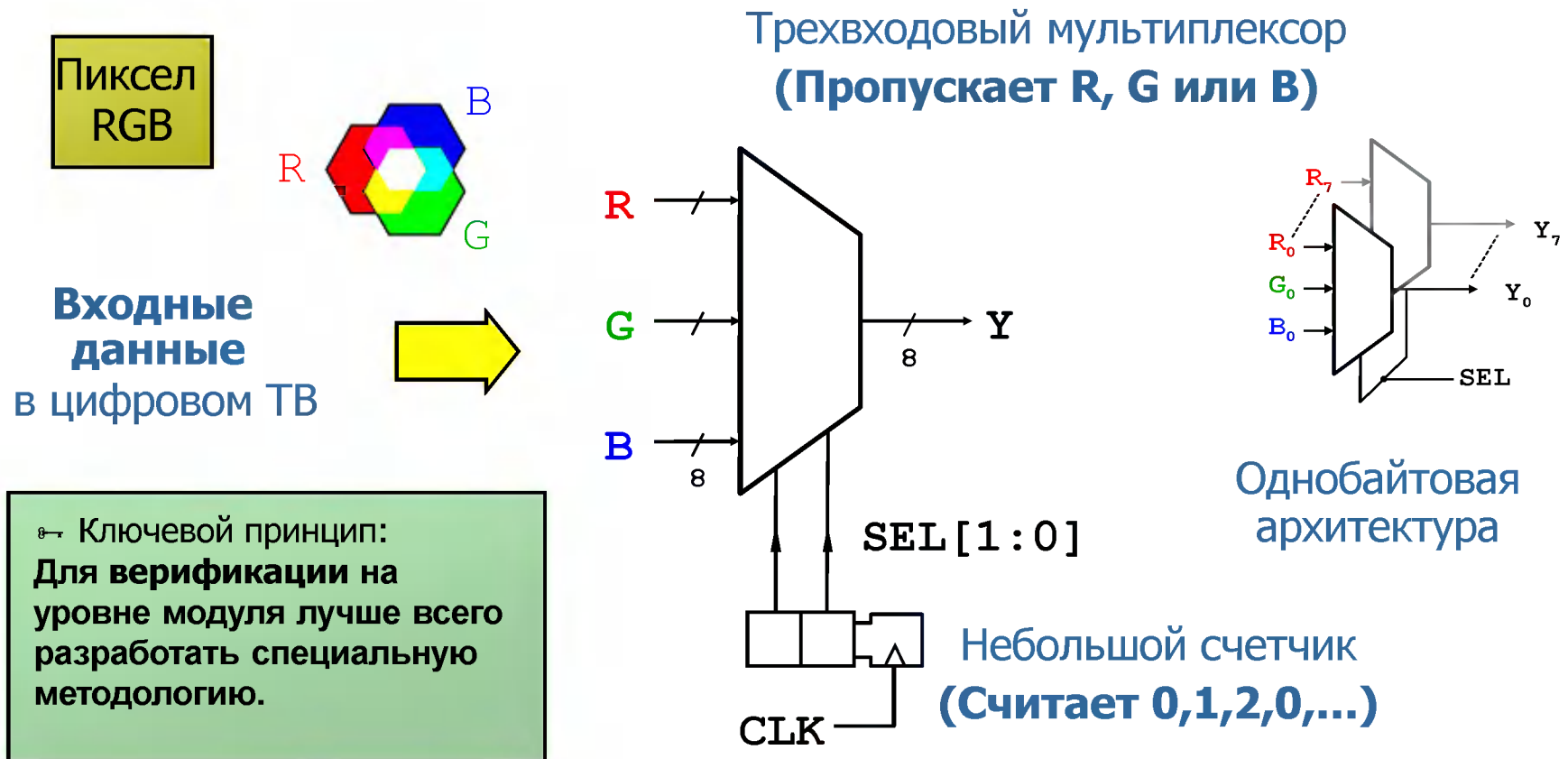
## Моделирование на уровне модуля:

- Практический пример: Мультиплексор RGB
- Обнаружение фатальных ошибок
- Среда тестирования уровня модуля

Временная  
диаграмма  
(Захват фазы в  
цифровой ФАПЧ)



# Практический пример: Мультиплексор RGB 3-6



- В блок входит **MUX3** и двухбитовый **счетчик вверх**.
- Он **по очереди** выводит байт B, G или R на выход Y.
- Требования к изделию указывают, что двухбитовый вход SEL **никогда** не должны достигать 3.

# Описание RTL Мультиплексор RGB

3-7

```
1  module RGBMUX(  
2      output bit [7:0] Y,  
3      input  bit [7:0] R,G,B,  
4          bit CLK  
5  );  
6      bit [1:0] SEL;  
7      always_ff @(posedge CLK)  
8          if (SEL!=2) SEL <= SEL+1;  
9          else SEL <= '0;  
10  
11     always_comb  
12         unique case (SEL)  
13             2: Y = R;  
14             1: Y = G;  
15             0: Y = B;  
16         endcase  
17  
18     //Detect invalid count:  
19     assert property (SEL!=3);  
20  
21 endmodule: RGBMUX
```

Порты ввода/вывода (8 бит)

Типичное RTL описание

Процесс счетчика

MUX3 процесс

Формальное свойство

Утверждение (параллельное)

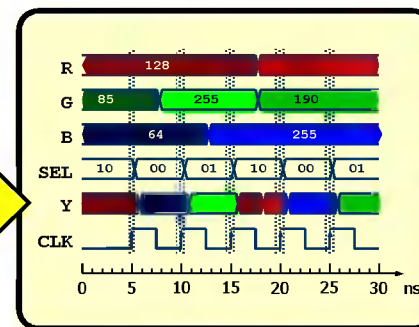
В этот код входит оператор утверждения о выполнении условия:  $SEL \neq 3$ .

# Моделирование: Код мультиплексора RGB 3-8

```
RGBMUX.sv
1 module RGBMUX(
2   output bit [7:0] Y,
3   input bit [7:0] R,G,B,
4   bit CLK
5 );
6 bit [1:0] SEL;
7 always_ff @(posedge CLK)
8   if (SEL!=2) SEL <= SEL+1;
9   else SEL <= '0;
10
11 always_comb
12   unique case (SEL)
13     2: Y = R;
14     1: Y = G;
15     0: Y = B;
16   endcase
17
18 //Detect invalid count:
19   assert property (SEL!=3);
20
21 endmodule: RGBMUX
```

RTL код

SystemVerilog:  
Программа  
моделирования



Вывод временных диаграмм

Входной сигнал  
Подается  
вручную

```
@07.5 ns: force G 255
@12.5 ns: force B 255
@17.5 ns: force R 200
...
```

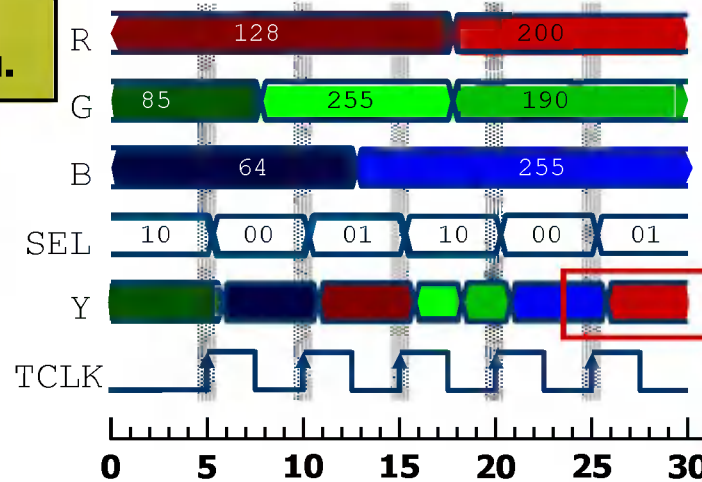
- Описание RTL компилируется в **моделируемую** модель.
- В это примере входные сигналы подаются **вручную** через командную строку.
- Но свободно ли RTL описание RGBMUX от **всех** логических ошибок?



# Обнаружение фатальных ошибок

3-9

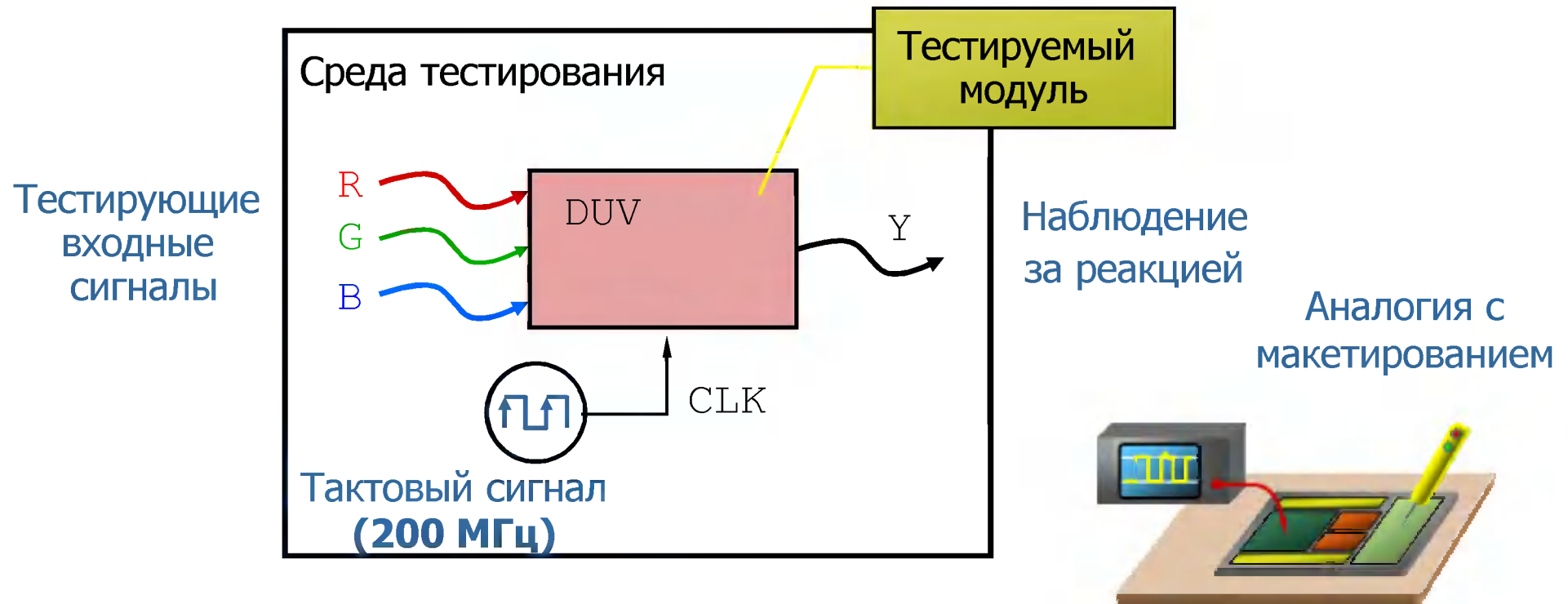
✂ Средства EDA:  
Программа  
моделирования HDL;  
просмотрщик  
временных диаграмм.



```
12  case (SEL)
13      2: Y = G;
14      1: Y = R;
15      0: Y = B;
16  endcase
```

Фатальная ошибка:  
R и G переставлены  
местами.

- В этот фрагмент кода RTL закралась **фатальная ошибка** проектировщика.
- По невнимательности код выбора R и G **переставлен местами**.
- Изучение временной диаграммы показывает **красный** сигнал - а должен быть **зеленый**.



- Ввод входных сигналов вручную **утомителен** и может привести к ошибкам.
- Как и настоящий лабораторный стенд, среда тестирования содержит тестируемый модуль (DUT).
- Под **управлением программы** на него подаются входные сигналы и анализируются его реакция.

# Простая среда тестирования (2/3)

3-11

Среда тестирования:  
Оболочка,  
не имеющая портов

Тактовый генератор  
(200 МГц)

```
RGBMUX_TB.sv
1 //Testbench for RGBMUX:
2 module RGBMUX_TB();
3     timeunit 1ns;
4     bit [7:0] Y,R,G,B;
5     bit CLK;
6 //Copy of RGBMUX:
7     RGBMUX DUV(. *);
8
9 //TV system clock:
10    initial begin
11        #2.5 CLK = '0;
12        forever #2.5 CLK = ~ CLK;
13    end
14
15 //Apply RGB inputs:
16    initial begin:TEST
17        @(negedge CLK) G = 255;
18        @(negedge CLK) B = 255;
19        @(negedge CLK) R = 200;
20    end: TEST
21 endmodule: RGBMUX_TB
```

Единица задержки (нс)

Экземпляр тестируемого модуля

Подаваемые входные сигналы

Часто входные сигналы подаются посередине между фронтами тактового сигнала

# Простая среда тестирования (3/3)

3-12

Среда  
тестирования  
с самопроверкой

```
RGBMUX_TB.sv
1 //Testbench for RGBMUX:
2 module RGBMUX_TB();
3     .
4     .
15 //Self-checking test:
16     initial begin:TEST
17         @(negedge CLK) G = 255;
18         CHECK_Y(255);
19         @(negedge CLK) B = 255;
20         CHECK_Y();
21         @(negedge CLK) R = 200;
22         CHECK_Y(200);
23     end: TEST
24     .
25     .
26     .
27     .
28     .
29     .
30     .
31     .
32     .
33     .
34     .
35     .
36     .
37     .
38     .
39     .
40     .
41     .
42     .
43     .
44     .
45     .
46     .
47     .
48     .
49     .
50     .
51     .
52     .
53     .
54     .
55     .
56     .
57     .
58     .
59     .
60     .
61     .
62     .
63     .
64     .
65     .
66     .
67     .
68     .
69     .
70     .
71     .
72     .
73     .
74     .
75     .
76     .
77     .
78     .
79     .
80     .
81     .
82     .
83     .
84     .
85     .
86     .
87     .
88     .
89     .
90     .
91     .
92     .
93     .
94     .
95     .
96     .
97     .
98     .
99     .
100    .
101    .
102    .
103    .
104    .
105    .
106    .
107    .
108    .
109    .
110    .
111    .
112    .
113    .
114    .
115    .
116    .
117    .
118    .
119    .
120    .
121    .
122    .
123    .
124    .
125    .
126    .
127    .
128    .
129    .
130    .
131    .
132    .
133    .
134    .
135    .
136    .
137    .
138    .
139    .
140    .
141    .
142    .
143    .
144    .
145    .
146    .
147    .
148    .
149    .
150    .
151    .
152    .
153    .
154    .
155    .
156    .
157    .
158    .
159    .
160    .
161    .
162    .
163    .
164    .
165    .
166    .
167    .
168    .
169    .
170    .
171    .
172    .
173    .
174    .
175    .
176    .
177    .
178    .
179    .
180    .
181    .
182    .
183    .
184    .
185    .
186    .
187    .
188    .
189    .
190    .
191    .
192    .
193    .
194    .
195    .
196    .
197    .
198    .
199    .
200    .
201    .
202    .
203    .
204    .
205    .
206    .
207    .
208    .
209    .
210    .
211    .
212    .
213    .
214    .
215    .
216    .
217    .
218    .
219    .
220    .
221    .
222    .
223    .
224    .
225    .
226    .
227    .
228    .
229    .
230    .
231    .
232    .
233    .
234    .
235    .
236    .
237    .
238    .
239    .
240    .
241    .
242    .
243    .
244    .
245    .
246    .
247    .
248    .
249    .
250    .
251    .
252    .
253    .
254    .
255    .
256    .
257    .
258    .
259    .
260    .
261    .
262    .
263    .
264    .
265    .
266    .
267    .
268    .
269    .
270    .
271    .
272    .
273    .
274    .
275    .
276    .
277    .
278    .
279    .
280    .
281    .
282    .
283    .
284    .
285    .
286    .
287    .
288    .
289    .
290    .
291    .
292    .
293    .
294    .
295    .
296    .
297    .
298    .
299    .
300    .
301    .
302    .
303    .
304    .
305    .
306    .
307    .
308    .
309    .
310    .
311    .
312    .
313    .
314    .
315    .
316    .
317    .
318    .
319    .
320    .
321    .
322    .
323    .
324    .
325    .
326    .
327    .
328    .
329    .
330    .
331    .
332    .
333    .
334    .
335    .
336    .
337    .
338    .
339    .
340    .
341    .
342    .
343    .
344    .
345    .
346    .
347    .
348    .
349    .
350    .
351    .
352    .
353    .
354    .
355    .
356    .
357    .
358    .
359    .
360    .
361    .
362    .
363    .
364    .
365    .
366    .
367    .
368    .
369    .
370    .
371    .
372    .
373    .
374    .
375    .
376    .
377    .
378    .
379    .
380    .
381    .
382    .
383    .
384    .
385    .
386    .
387    .
388    .
389    .
390    .
391    .
392    .
393    .
394    .
395    .
396    .
397    .
398    .
399    .
400    .
401    .
402    .
403    .
404    .
405    .
406    .
407    .
408    .
409    .
410    .
411    .
412    .
413    .
414    .
415    .
416    .
417    .
418    .
419    .
420    .
421    .
422    .
423    .
424    .
425    .
426    .
427    .
428    .
429    .
430    .
431    .
432    .
433    .
434    .
435    .
436    .
437    .
438    .
439    .
440    .
441    .
442    .
443    .
444    .
445    .
446    .
447    .
448    .
449    .
450    .
451    .
452    .
453    .
454    .
455    .
456    .
457    .
458    .
459    .
460    .
461    .
462    .
463    .
464    .
465    .
466    .
467    .
468    .
469    .
470    .
471    .
472    .
473    .
474    .
475    .
476    .
477    .
478    .
479    .
480    .
481    .
482    .
483    .
484    .
485    .
486    .
487    .
488    .
489    .
490    .
491    .
492    .
493    .
494    .
495    .
496    .
497    .
498    .
499    .
500    .
501    .
502    .
503    .
504    .
505    .
506    .
507    .
508    .
509    .
510    .
511    .
512    .
513    .
514    .
515    .
516    .
517    .
518    .
519    .
520    .
521    .
522    .
523    .
524    .
525    .
526    .
527    .
528    .
529    .
530    .
531    .
532    .
533    .
534    .
535    .
536    .
537    .
538    .
539    .
540    .
541    .
542    .
543    .
544    .
545    .
546    .
547    .
548    .
549    .
550    .
551    .
552    .
553    .
554    .
555    .
556    .
557    .
558    .
559    .
560    .
561    .
562    .
563    .
564    .
565    .
566    .
567    .
568    .
569    .
570    .
571    .
572    .
573    .
574    .
575    .
576    .
577    .
578    .
579    .
580    .
581    .
582    .
583    .
584    .
585    .
586    .
587    .
588    .
589    .
590    .
591    .
592    .
593    .
594    .
595    .
596    .
597    .
598    .
599    .
600    .
601    .
602    .
603    .
604    .
605    .
606    .
607    .
608    .
609    .
610    .
611    .
612    .
613    .
614    .
615    .
616    .
617    .
618    .
619    .
620    .
621    .
622    .
623    .
624    .
625    .
626    .
627    .
628    .
629    .
630    .
631    .
632    .
633    .
634    .
635    .
636    .
637    .
638    .
639    .
640    .
641    .
642    .
643    .
644    .
645    .
646    .
647    .
648    .
649    .
650    .
651    .
652    .
653    .
654    .
655    .
656    .
657    .
658    .
659    .
660    .
661    .
662    .
663    .
664    .
665    .
666    .
667    .
668    .
669    .
670    .
671    .
672    .
673    .
674    .
675    .
676    .
677    .
678    .
679    .
680    .
681    .
682    .
683    .
684    .
685    .
686    .
687    .
688    .
689    .
690    .
691    .
692    .
693    .
694    .
695    .
696    .
697    .
698    .
699    .
700    .
701    .
702    .
703    .
704    .
705    .
706    .
707    .
708    .
709    .
710    .
711    .
712    .
713    .
714    .
715    .
716    .
717    .
718    .
719    .
720    .
721    .
722    .
723    .
724    .
725    .
726    .
727    .
728    .
729    .
730    .
731    .
732    .
733    .
734    .
735    .
736    .
737    .
738    .
739    .
740    .
741    .
742    .
743    .
744    .
745    .
746    .
747    .
748    .
749    .
750    .
751    .
752    .
753    .
754    .
755    .
756    .
757    .
758    .
759    .
760    .
761    .
762    .
763    .
764    .
765    .
766    .
767    .
768    .
769    .
770    .
771    .
772    .
773    .
774    .
775    .
776    .
777    .
778    .
779    .
780    .
781    .
782    .
783    .
784    .
785    .
786    .
787    .
788    .
789    .
790    .
791    .
792    .
793    .
794    .
795    .
796    .
797    .
798    .
799    .
800    .
801    .
802    .
803    .
804    .
805    .
806    .
807    .
808    .
809    .
810    .
811    .
812    .
813    .
814    .
815    .
816    .
817    .
818    .
819    .
820    .
821    .
822    .
823    .
824    .
825    .
826    .
827    .
828    .
829    .
830    .
831    .
832    .
833    .
834    .
835    .
836    .
837    .
838    .
839    .
840    .
841    .
842    .
843    .
844    .
845    .
846    .
847    .
848    .
849    .
850    .
851    .
852    .
853    .
854    .
855    .
856    .
857    .
858    .
859    .
860    .
861    .
862    .
863    .
864    .
865    .
866    .
867    .
868    .
869    .
870    .
871    .
872    .
873    .
874    .
875    .
876    .
877    .
878    .
879    .
880    .
881    .
882    .
883    .
884    .
885    .
886    .
887    .
888    .
889    .
890    .
891    .
892    .
893    .
894    .
895    .
896    .
897    .
898    .
899    .
900    .
901    .
902    .
903    .
904    .
905    .
906    .
907    .
908    .
909    .
910    .
911    .
912    .
913    .
914    .
915    .
916    .
917    .
918    .
919    .
920    .
921    .
922    .
923    .
924    .
925    .
926    .
927    .
928    .
929    .
930    .
931    .
932    .
933    .
934    .
935    .
936    .
937    .
938    .
939    .
940    .
941    .
942    .
943    .
944    .
945    .
946    .
947    .
948    .
949    .
950    .
951    .
952    .
953    .
954    .
955    .
956    .
957    .
958    .
959    .
960    .
961    .
962    .
963    .
964    .
965    .
966    .
967    .
968    .
969    .
970    .
971    .
972    .
973    .
974    .
975    .
976    .
977    .
978    .
979    .
980    .
981    .
982    .
983    .
984    .
985    .
986    .
987    .
988    .
989    .
990    .
991    .
992    .
993    .
994    .
995    .
996    .
997    .
998    .
999    .
1000   .
endmodule: RGBMUX_TB
```

Подпрограмма  
CHECK\_Y

Вызов CHECK\_Y  
(Используется значение по умолчанию)

Аргумент  
(Есть значение по умолчанию)

Печать сообщения об ошибке

Реальный отклик Y анализируется **после** переднего фронта импульса такого сигнала.

## Функциональное моделирование:

- Уровни верификации
- Тестовый среды разных уровней
- Рандомизированные данные
- Стратегия CRV  
(ограниченная случайная верификация)
- Встроенные утверждения

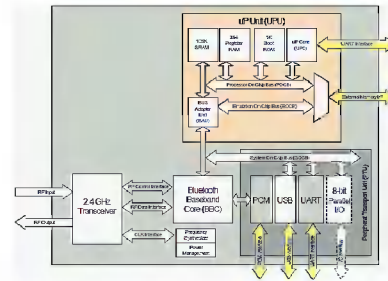
### Ячейка ATM

VPI [11:4]		
VPI [3:0]	VCI [15:12]	
VCI [11:4]		
VCI [3:0]	CLP	PT
HEC		
PAYLOAD [0]		
PAYLOAD [1]		
PAYLOAD [2]		
...		
PAYLOAD [46]		
PAYLOAD [47]		

# Уровни верификации

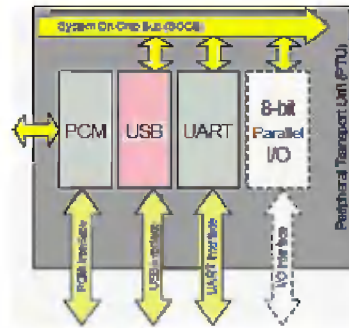
3-14

Уровень SOC



BSM 2033:  
**Верификация взаимодействия**  
 между подсистемами - а не  
 функционирования блоков.  
 Включает ВЧ часть.

Уровень подсистемы



PTU (Модуль периферийной  
 передачи):  
**Подход «снизу-вверх»**  
 гарантирует то, что блоки UART,  
 РСМ, USB уже разработаны.

Уровень функционального блока



UART:  
 Большой объем работы по  
 верификации.  
**Гарантия полной реализации**  
 всех требований. Автономная  
 верификация функций,  
 интерфейсов.

Уровень узла



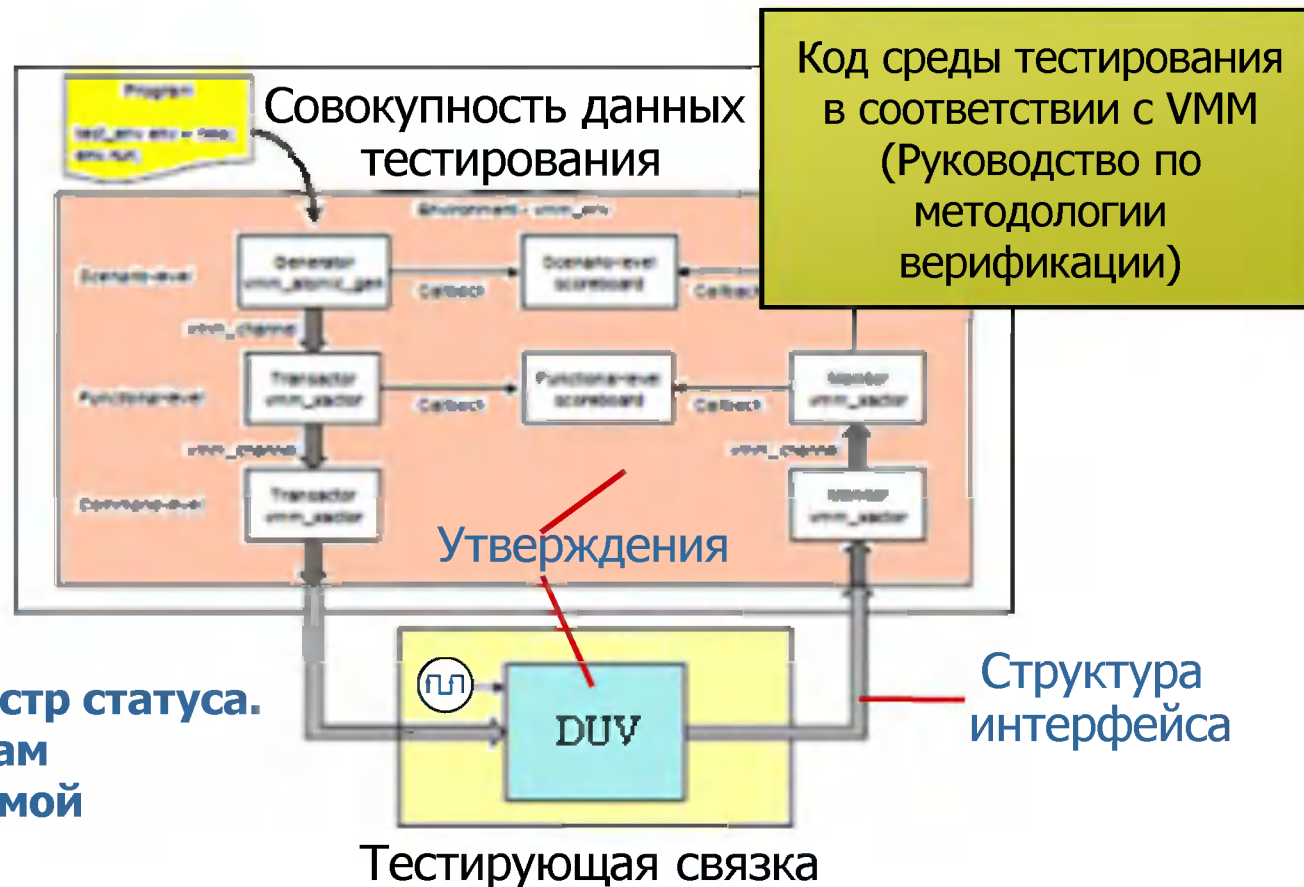
- Часто **сначала** выполняется верификация независимых функциональных блоков.
- Верификация более высокого уровня ограничивается обнаружением ошибок в **интерфейсах**.

# Тестовые среды разных уровней

3-15

Уровень сценария:  
**Тестируемая микросхема откликается на, например, изменение громкости или изменение каналов.**

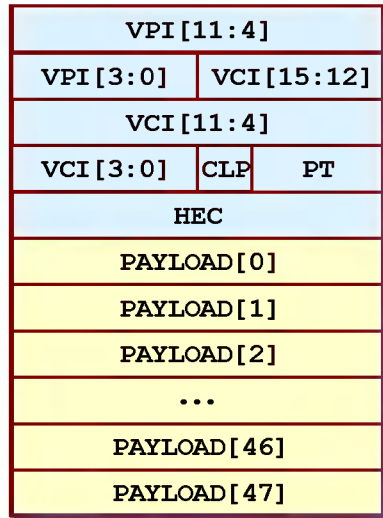
Уровень команд:  
**Например, считать регистр статуса. Имеет доступ к контактам ввода/вывода тестируемой микросхемы.**



- Среда тестирования тестирования уровня микросхемы обычно разделена на **уровни** для повторного использования.
- Руководство по методологии верификации (VMM) **стандартизирует** организацию кода.
- Как и ранее, входные сигналы подаются на тестируемое устройство, анализируется его реакция.

# Типичные объекты данных

3-16



vmm\_xactor



DUV

```
ATM_CELL.sv
1 //VMM base class: vmm_data
2 class ATM_CELL;
3     bit [11:0] VPI;
4     bit [15:0] VCI;
5     bit        CLP;
6     bit [2:0]  PT;
7     bit [ 7:0] HEC;
8     byte PAYLOAD [48];
9     function void FILL(byte BYTE);
10         this.VPI = 12'hABC;
11         for (int I=0; I<48; I++)
12             this.PAYLOAD[I] = BYTE;
13     endfunction: FILL
14 endclass: ATM_CELL
15
20 initial
21 begin:TEST
22     ATM_CELL CELL;
23     CELL = new;
24     CELL.FILL(8'h0A);
25 end: TEST
...
```

Данные элемента:  
**Для 53-битовой ячейки ATM (асинхронной передачи)**

Функции элемента:  
**Заполняет VPI (идентификатор виртуального пути) и полезные данные**

Новый объект:  
**Заполняется командой**

Класс многократно используемых **шаблонов** для порождения новых объектов.



# Рандомизация полезных данных ATM 3-17

```
ATM_CELL.sv
1 class ATM_CELL;
2   bit [11:0] VPI;
3   bit [15:0] VCI;
4   . . .
5   bit [2:0] PT;
6   bit [ 7:0] HEC;
7   rand byte PAYLOAD [48];
8   function void PRINT(int N);
9     $display("Item %d: %b %h",
10            N,PAYLOAD [47],PAYLOAD [47]);
11   endfunction: PRINT
12 endclass: ATM_CELL
. . .
20 initial
21 begin:TEST
22   ATM_CELL CELL = new;
23   for (int I=1; I<=12; I++)
24   begin
25     CELL.randomize();
26     CELL.PRINT(I);
27   end
28 end: TEST
. . .
```

Полезные данные:  
Случайная переменная

Печать хвоста списка

Заполнение полезных данных

```
Transcript
File Edit View Window
Transcript
# run -all
# 1: 1100101101100110 52070
# 2: 1011111110100111 49063
# 3: 0110100110010000 27024
# 4: 1111001011010111 62167
# 5: 1100100111111101 51709
# 6: 1110010110100010 58786
# 7: 1000101111011110 35806
# 8: 1100110100100010 52514
# 9: 0011110110101100 15788
# 10: 1110000001111011 57467
# 11: 0011001111010111 13271
# 12: 1011010001101000 46184
```

Генерация 12 наборов случайных данных

Элементы полезных данных сейчас объявлены как **случайная** переменная.

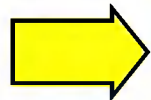
# Стратегия CRV

3-18

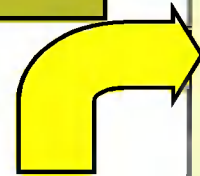
CRV:

Ограниченная случайная верификация, основана на методе `randomize`.

Повторное моделирование:  
Устранение обнаруженных ошибок.



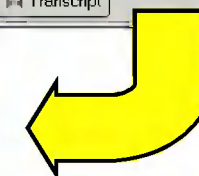
Начальное число генератора случайных чисел  
(32'h9A8B\_7C6D)



```
ATM_Cell.sv
1 class ATM_CELL;
2   bit [11:0] VPI;
3   bit [15:0] VCI;
4
5   bit [2:0] PT;
6   bit [ 7:0] HEC;
7   rand byte PAYLOAD [48];
8   function void PRINT(int N);
9     $display("Item 8d: 8b 8h",
10            N, PAYLOAD[47], PAYLOAD[47]);
11   endfunction: PRINT
12 endclass: ATM_CELL
13
14
15
16
17
18
19
20 initial
21 begin: TEST
22   ATM_CELL CELL = new;
23   for (int I=1; I<=12; I++)
24     begin
25       CELL.randomize();
26       CELL.PRINT(I);
27     end
28 end: TEST
```



```
Transcript
File Edit View Window
Transcript:
# run -all
# 1: 1100101101100110 52070
# 2: 1011111110100111 49063
# 3: 0110100110010000 27024
# 4: 1111001011010111 62167
# 5: 1100100111111101 51709
# 6: 1110010110100010 58786
# 7: 1000101111011110 35806
# 8: 1100110100100010 52514
# 9: 0011110110101100 15788
# 10: 1110000001111011 57467
# 11: 0011001111010111 13271
# 12: 1011010001101000 46184
```



Изменение начального числа:  
В командной строке или при вызове процедуры.

```
vcs ... +ntb_random_seed=32'h9A8B_7C6D filenames
```

- Пользователь создает **случайный поток** данных для АТМ.
- Случайные входные сигналы часто проявляют **неожиданные** ошибки.

# Ограниченная рандомизация 3-19

Корректный идентификатор пути  
Случайная переменная  
(В диапазоне 0-4095)

Сила средств проектирования:  
Средства EDA могут сгенерировать случайные комбинации, о которых вы никогда не думали!

```
ATM_CELL.sv
1 class ATM_CELL;
2   rand bit [11:0] VPI;
3   rand bit [15:0] VCI;
4   . . .
5   bit [2:0] PT;
6   bit [ 7:0] HEC;
7   rand byte PAYLOAD [48];
8   . . .
12  constraint PATH_ID {
13    VPI inside {[0:1],[255:4095]}
14  }
15 endclass: ATM_CELL
16 . . .
20 initial
21 begin:TEST
22   ATM_CELL CELL = new;
23   for (int I=1; I<=12; I++)
24   begin
25     CELL.randomize();
26     CELL.PRINT(I);
27   end
28 end: TEST
29 . . .
```

Ограничение:  
Явно указанный диапазон

Применение ограничений

Раздомозированные значения VPI попадают только в **явно указанный** диапазон.

# Встроенные утверждения

3-20

//Декларация свойства:

```
property req5gnt;  
  @(posedge clk) disable iff (! rst)  
  $rose(req) |-> ##[1:5] gnt;
```

//Параллельное утверждение:

```
assert property (req5gnt);
```

Explicit  $\text{clk}\uparrow$



- Оператор утверждения проверяет **каждый фронт** тактового сигнала, ведется протокол сбоев.
- Свойством, о котором делается утверждение, может быть вся **последовательность событий**.

## Контроль процесса:

- Когда верификация завершена?
- Традиционные метрики
- Функциональное покрытие
- Заполнение провалов покрытия
- CRV (ограниченная случайная верификация) с метрикой покрытия

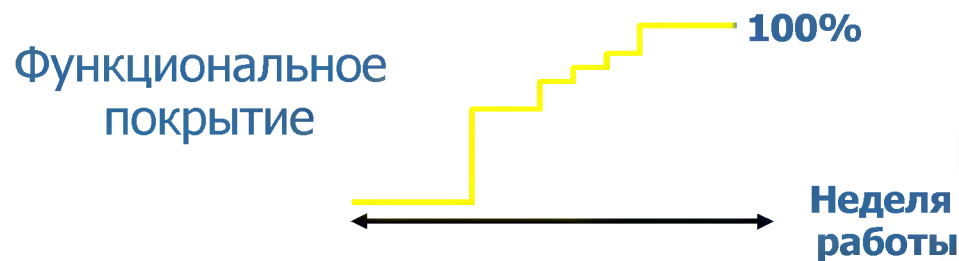
Провалы  
покрытия



# Когда верификация завершена? 3-22

Основные производители приводят следующие метрики:

- ✦ 40 миллиардов **случайных циклов** выполняются без ошибок.
- ✦ Все **направленные** тесты в плане верификации выполнены.
- ✦ Цели **функционального покрытия** достигнуты.
- ✦ Частота обнаружения ошибок **уменьшалась**.
- ✦ Достигнут **предельный срок** календарного плана.



✦ Практическое правило: Для оценки усилий на верификацию важно не количество логических элементов, а количество **разработчиков**. Каждый имеет свои предположения о проекте.

**Функциональное** покрытие дополняет **традиционные** метрики.

# Метрика частоты обнаружения ошибок 3-23



Система отслеживания ошибок  
(Внутренняя)

- Традиционная метрика: количество обнаруженных ошибок за рабочий день.
- Неожиданные скачки могут свидетельствовать о скрытых проблемах.

Выделенная  
функция  
UART 1.1

Выдержка из  
плана  
верификации

## BCM 2033 Verification Plan

### 1.1 Features (UART Interface):

- Request-to-send (**RTS**) pin must be asserted to inform modem that UART is ready to exchange data based on RS-232 protocol.
- Clear-to-send (**CTS**) pin must be asserted to inform UART that modem is ready to exchange data.

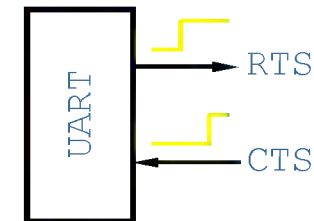
### 1.2 Features (UART Function):

- Characters shall be sent and received serially, with LSB first.
- Each character shall have 5, 6, 7, or 8 data bits per frame.
- The parity bit shall be even, odd, mark, space, or none according to mode bits 5:3 of 16550-type line control register (LCR).

### 1.3 Features (UART Architecture):

- An empty condition of the 16-byte FIFO buffer shall set bit 5 of UART 16550-compatible line status register (LSR).
- Bits 7:4 of the interrupt ID register (IIR) are reserved at 1100.

BCM2033  
Приемопередатчик  
Bluetooth



Аппаратное  
управление обменом  
с подтверждение  
установления связи

- Выделенные функции выбраны из плана для контроля.
- Эта группа функций обеспечивает **выборку RTS, CTS** в одно и тоже время.



---

# Функциональное покрытие

3-25

Поиск провалов в покрытии отдельной функции:

- ✦ Функциональное покрытие отслеживает процент функций проекта, которые испытаны набором тестовых данных.
- ✦ Метрика определяется количеством испытанных функции **спецификации проекта**—в отличии от кода реализации RTL.
- ✦ Требует простой идентификации **отдельной функции** в плане верификации.
- ✦ При покрытии **100%**, **все** существенные функции должны быть протестированы—задача верификации **выполнена**.
- ✦ Но реальное **достижение 100%** **является сложной задачей**.

# Группа покрытия

3-26

```
UART_TB.sv
1  class MODEM_XCHG;
2      rand bit RTS;
3      rand bit CTS;
   . . .
9  endclass: MODEM_XCHG
   . . .
11 MODEM_XCHG XCHG_0 = new;
   . . .
13 covergroup UART_1_1;
14     coverpoint XCHG_0.RTS;
15     coverpoint XCHG_0.CTS;
16 endgroup: UART_1_1
   . . .
18 initial
19 begin:TEST
20     UART_1_1 CHECK = new;
21     XCHG_0.randomize();
22     UART_IF.RTS <= XCHG_0.RTS;
23     UART_IF.CTS <= XCHG_0.CTS;
24     CHECK.sample();
   . . .
30 end: TEST
   . . .
```

Интерфейс UART  
Случайная  
переменная.

Выбранная  
функция  
(UART 1.1)

Группа покрытия  
Выбираются  
вместе

Обмен данными  
рандомизирован

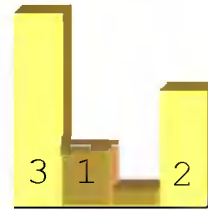
Группа  
выбрана

Величины RTS, CTS рандомизированы, поданы на вход блока и затем выбраны.

# Заполнение провалов покрытия

3-27

```
repeat(6)
begin
  XCHG_0.randomize();
  UART_IF.RTS <= XCHG_0.RTS;
  UART_IF.CTS <= XCHG_0.CTS;
  CHECK.sample();
end
```



База данных

Пока не завершено:  
**Нужно заполнить провал при значении сигналов 10**

Попадания:



= 6

Сигналы 00    Сигналы 01    Сигналы 10    Сигналы 11

```
option.at_least = 1; //Минимум для попадания
```

- Отчет о покрытии показывает покрытие функции на уровне **75%**.
- Для **полного** покрытия нужно продолжить верификацию.

# CRV (ограниченная случайная верификация) 3-28 с метрикой покрытия

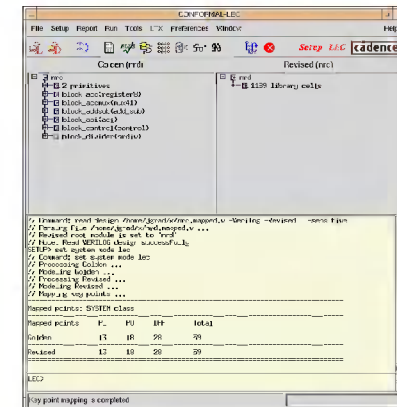


- Прямое кодирование для функционального покрытия является **нетривиальной** задачей.
- Но современные средства EDA позволяют выполнить почти **исчерпывающее** тестирование.

## Проверка эквивалентности:

- Где использует формальная проверка эквивалентности (FEC)?
- Типичные применения формальной проверки эквивалентности
- Упрощенный алгоритм формальной проверки эквивалентности
- Успешная проверка

Коммерческие средства  
формальной проверки  
эквивалентности



# Рекомендуемые контрольные точки

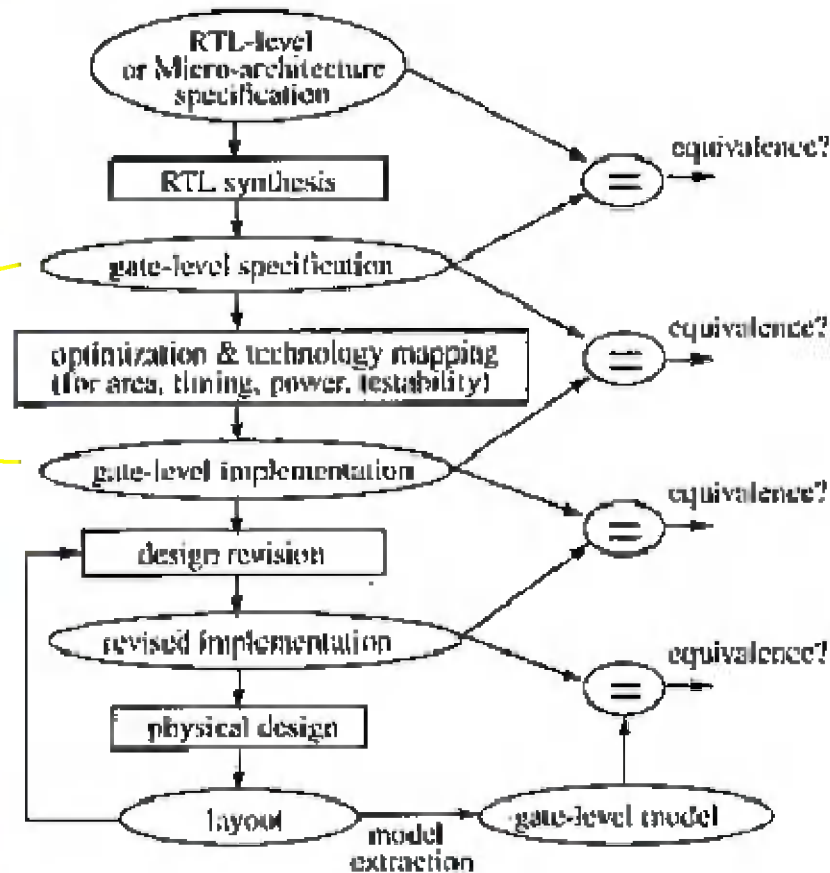
3-30

✂ Средства EDA:  
Средства проверки  
логической  
эквивалентности (FEC).

Синтезированный  
список соединений

Оптимизация по  
энергопотреблению.

→ Ключевой принцип:  
Функциональная  
эквивалентность проверяется  
*без* необходимости  
моделирования обоих  
проектов.



- Проверка эквивалентности указывает на **функциональные** различия.
- Сравнивает **референсную** и **измененную** базы данных проекта.

- ✦ Добавление средств для тестирования (DFT):  
Синтезированный список соединений изменяется: в него добавляются **цепочки сканирования**, фиксирующие триггеры-защелки, проходные мультиплексоры.
- ✦ Логическая оптимизация:  
Восстановление синхронизации регистров; арифметические преобразования; управление прохождением тактовых сигналов и проектирование для **уменьшения энергопотребления**.
- ✦ Физическое проектирование:  
Трассировка проекта и добавление **буферов** или элементов с **мощным выходом**. Синтез дерева тактовых сигналов.
- ✦ Указания о конструкторском техническом изменении (Engineering Change Orders, ECOs):  
Любое изменение, которое предусматривает **ручное редактирование** списка соединений.

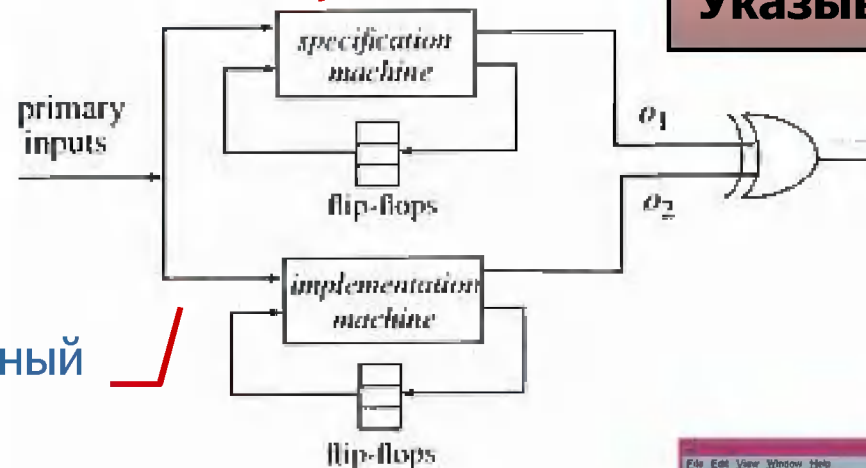
# Упрощенный алгоритм формальной проверки эквивалентности

3-32

Референсный проект

Подаются одинаковые входные сигналы

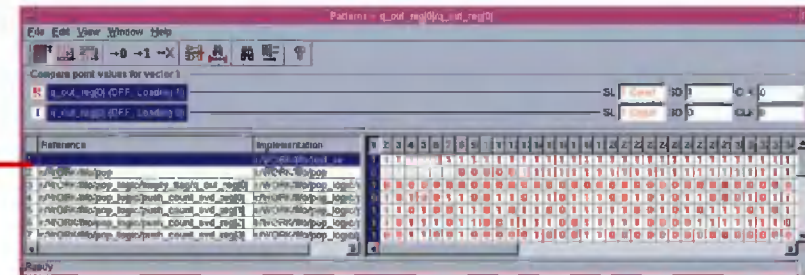
Измененный проект



Выход разницы:  
Указывает расхождения.

Классический алгоритм:  
Оба конечные автоматы переходят из состояния в состояние синхронно, до появления **расхождения**.

Ошибочная установка:  
Сигнал разрешения сканирования (SE) **не установлен**, что приводит в расхождению.

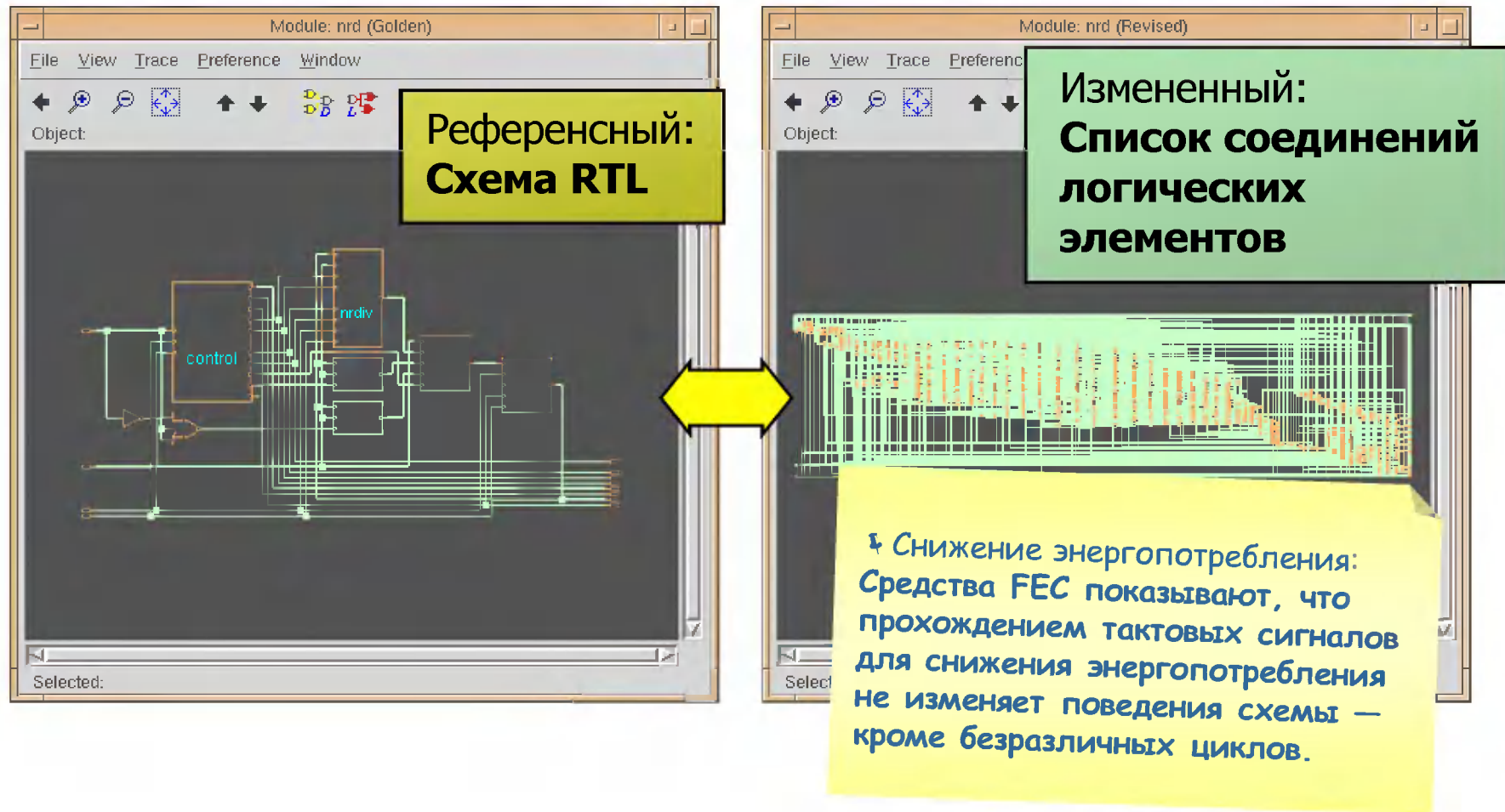


- Для сравнения фрагментов схем используется **дискретная математика** (бинарная диаграмма решений и т.д.).
- Фрагменты схемы **заканчиваются** первичными выходами или контактами Q триггеров.



# Успешная проверка

3-33



- Схемы уровней RTL и логических элементов выглядят по-разному.
- Но средства формальной проверки эквивалентности (FEC) доказывают их **функциональную эквивалентность**.

## Аппаратная эмуляция

- Необходимость эмуляции
- Концепции эмуляции
- Этапы аппаратной эмуляции
- Меньшее время выхода на рынок

Платформа  
эмуляции



# Концепции эмуляции

3-35

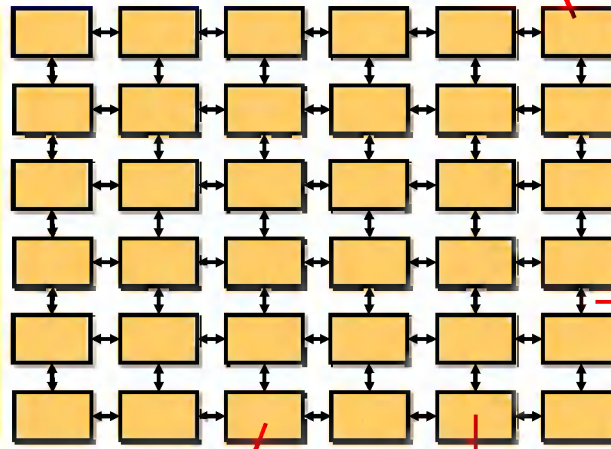
✂ Средства EDA:  
**Аппаратный эмулятор**

## Конструкция RTL

```
// Небольшой счетчик  
always_ff @(posedge CLK)  
  if (SEL!=2) ++SEL;  
  else SEL = '0;
```

✂ Не на реальной скорости:  
Выполняется на ~10 МГц.  
Медленнее, чем на  
номинальной скорости, но  
**быстрее**, чем в любой  
программе моделирования.

Платформа  
эмуляции



**Сигнализация о событиях**

**Утверждение**

**Уровень логических элементов**

```
//Экземпляр буфера:  
TRIBUF T0 (OUT, IN, ~OE);
```

```
//Обнаружение недопустимого элемента:  
assert property (SEL!=3);
```

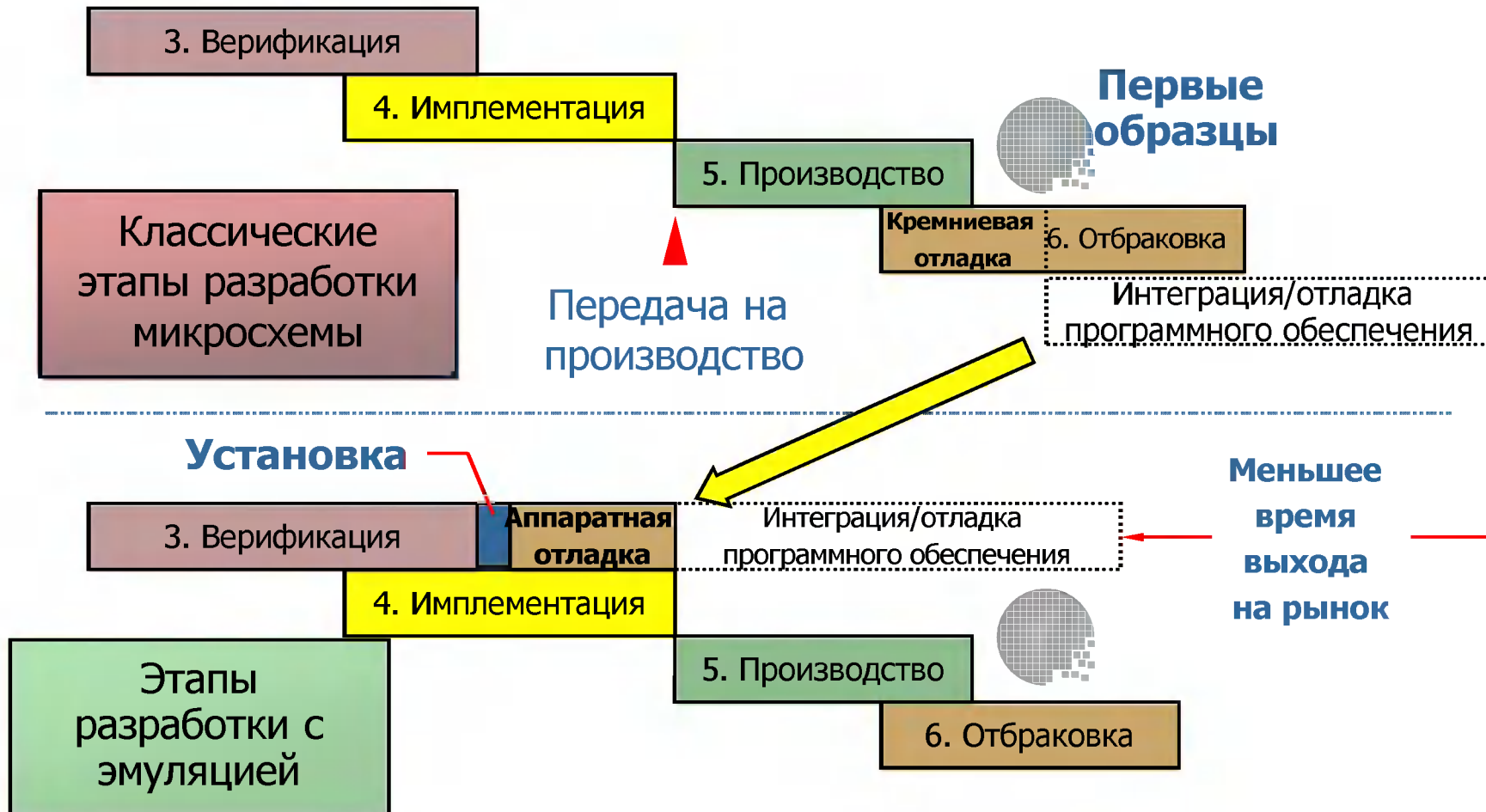
- Код HDL компилируется и отображается на **FPGA-подобные** кластеры.
- Большой массив низкоскоростных **аппаратных моделей** SOC.

# Этапы аппаратной эмуляции

3-36



- Эмулятор компонует RTL-блоки, IP-блоки, встроенное ОЗУ.
- Параллельно **моделируются** несинтезируемые блоки.



- Аппаратная модель может раньше использоваться для отладки программного обеспечения.
- Можно начать после передачи проекта на производство, на несколько недель ранее получения первых образцов микросхем.

## ВЧ/Аналоговая верификация:

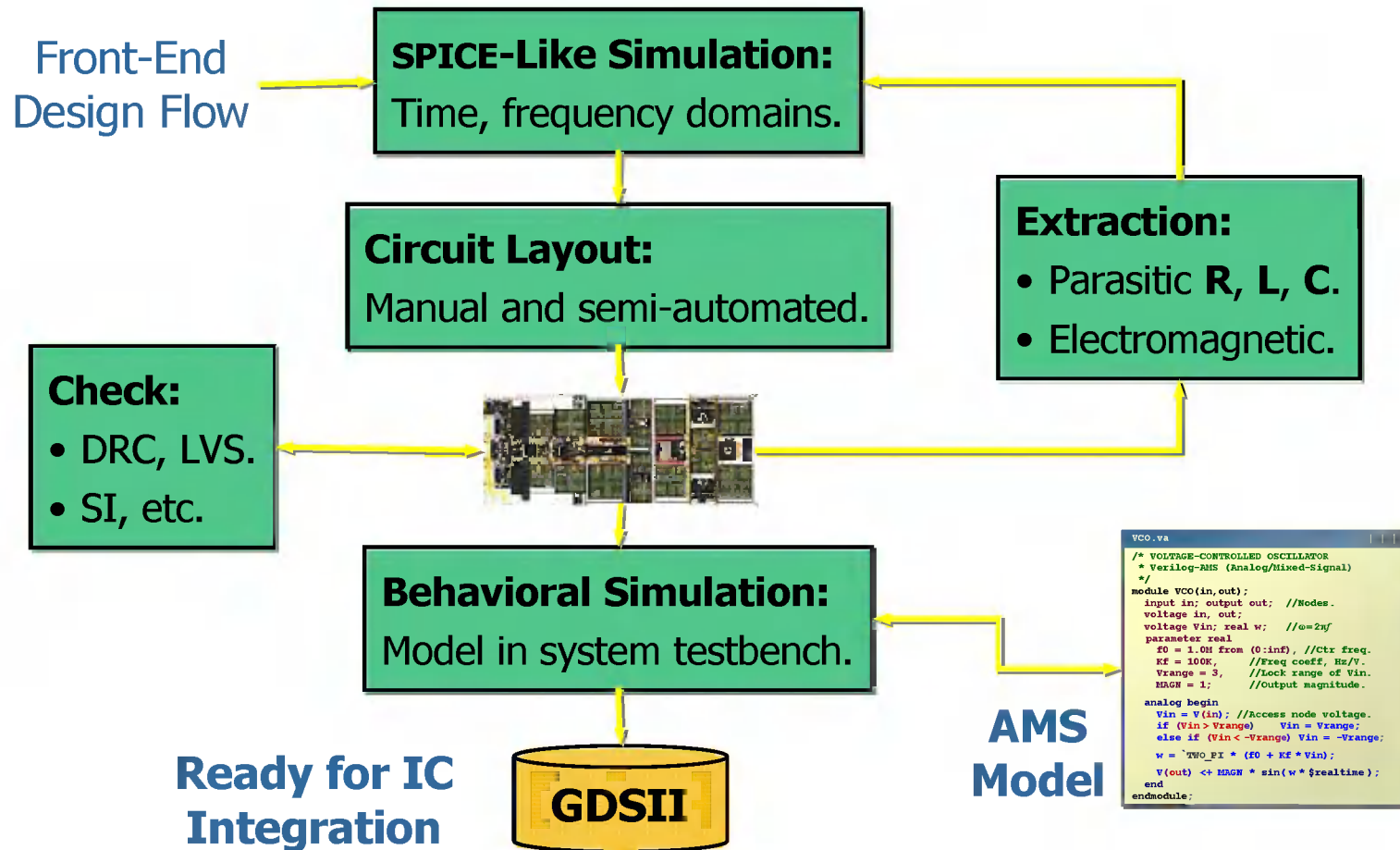
- Этапы ВЧ верификации
- Повторное рассмотрение: Модель ГУН
- Аналоговое совместное моделирование

ВЧ КМОП  
макроячейка



# Этапы ВЧ верификации

3-39



- Дополнительный шаг: Вручную разработать Verilog-AMS модель.
- Расширение синтаксиса, обеспечивающее поддержку моделирования аналоговых и смешанных (аналого-цифровых) устройств

# Повторное рассмотрение: Модель ГУН 3-40

```
VCO.va
/* VOLTAGE-CONTROLLED OSCILLATOR
 * Verilog-AMS (Analog/Mixed-Signal)
 */
module VCO(in,out);
  input in; output out; //Nodes.
  voltage in, out;
  voltage Vin; real w; //ω=2πf
  parameter real
    f0 = 1.0G from (0:inf), //Ctr freq.
    Kf = 100M, //Freq coeff, Hz/V.
    Vrange = 3, //Lock range of Vin.
    MAGN = 1; //Output magnitude.

  analog begin
    Vin = V(in); //Access node voltage.
    if (Vin > Vrange) Vin = Vrange;
    else if (Vin < -Vrange) Vin = -Vrange;
    w = `TWO_PI * (f0 + Kf * Vin);
    V(out) <+ MAGN * sin(w * $realtime);
  end
endmodule;
```

Имена узлов

Коэффициент передачи

Оператор, определяющий ВЫХОД

Несинтезируемое описание:  
**Слишком абстрактное - не RTL.**

Объявление переменных

Ограничение  $V_{in}$  вне диапазона

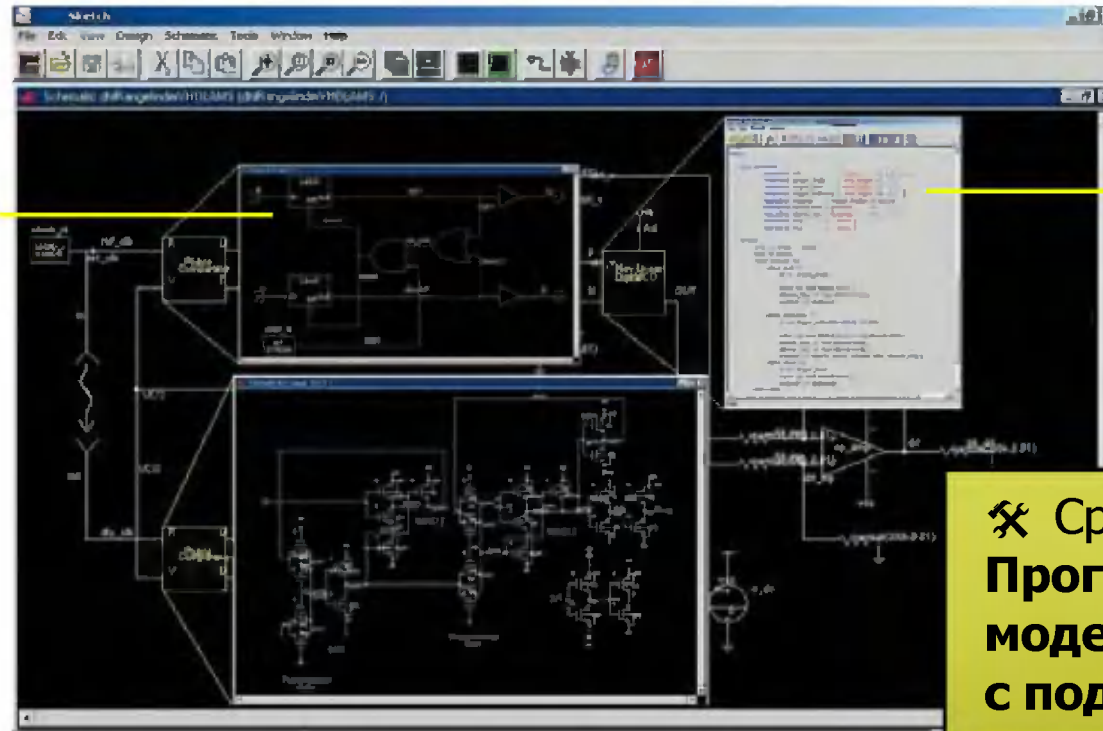
Ключевым AMS расширением языка Verilog является блок **аналогового** кода.



# Совместное моделирование Verilog-AMS

3-41

Цифровая  
логика



- Абстрактные модели AMS моделируются совместно с цифровой логикой.
- Можно обнаружить **интерфейсные** ошибки - неправильная полярность или ошибки соединений.
- Очень важно для аналоговых макроячеек, взаимодействующих с цифровыми блоками.

1. Перед началом имплементации необходимо **верифицировать** тысячи строк кода - от RTL блоков до IP ядер.
2. Традиционное моделирование не позволяет верифицировать сложные SOC (системы на кристалле). Необходимы дополнительные методы.
3. Верификация **CRV** (ограниченная случайная верификация) с функциональным покрытием вытеснила направленные наборы данных тестирования. При верификации используются операторы **утверждений**.
4. FEC (формальная проверка эквивалентности) является статическим методом **формальной** проверки эквивалентности двух версий проекта.
5. Эмуляция создает низкоскоростную **аппаратную модель**, которая позволяет выполнить отладку системы до получения первых образцов микросхем.
6. AMS модели **RF CMOS** и **аналоговых** макроячеек могут моделироваться совместно с цифровой частью микросхемы.

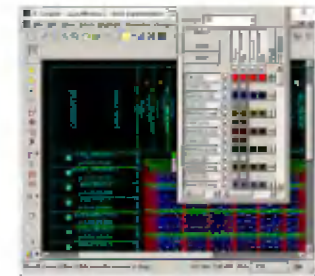
---

# Developing the Nanometer ASIC

From Specs to Silicon

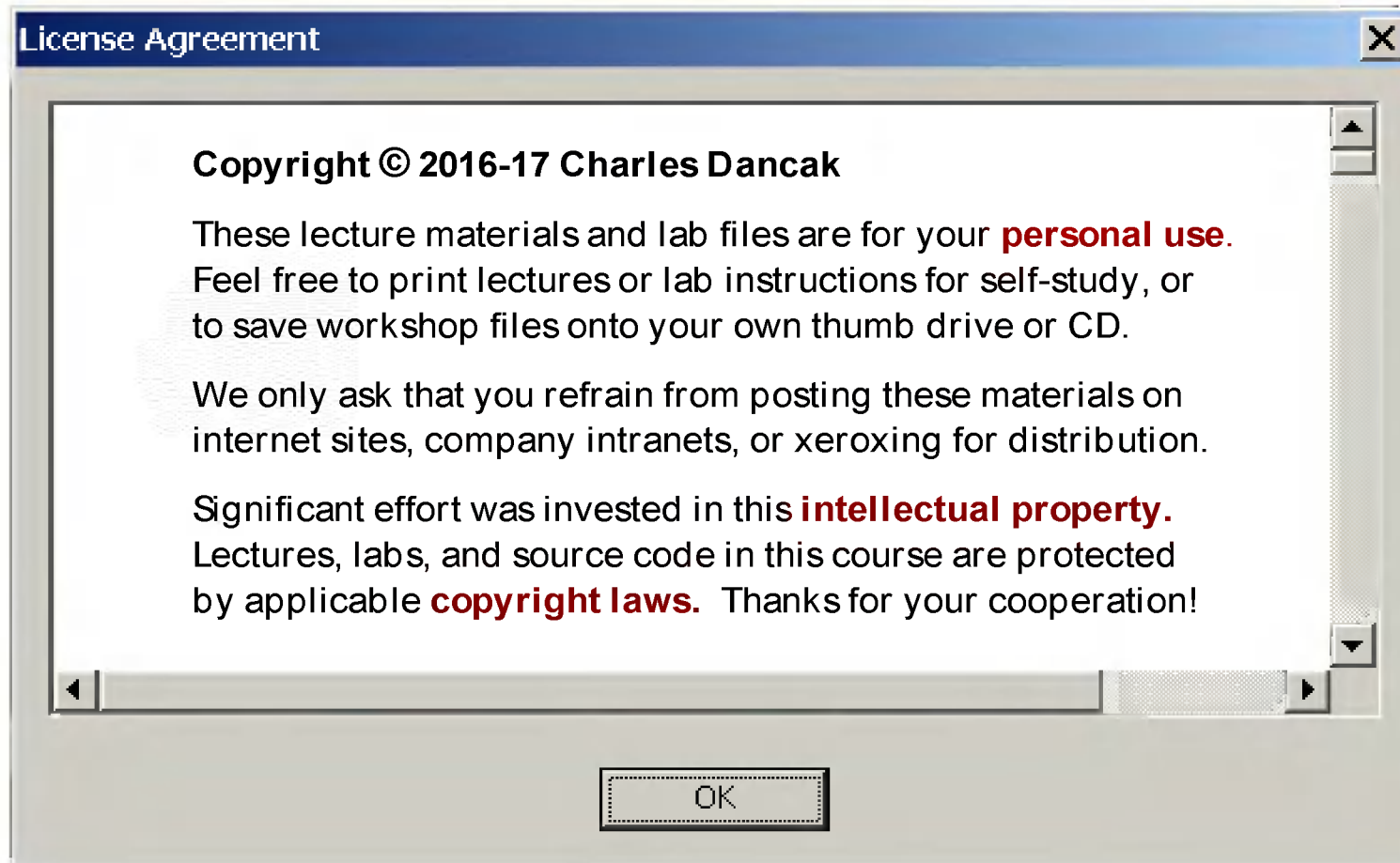


## 4. Стадия имплементации



# Авторское право

4-2



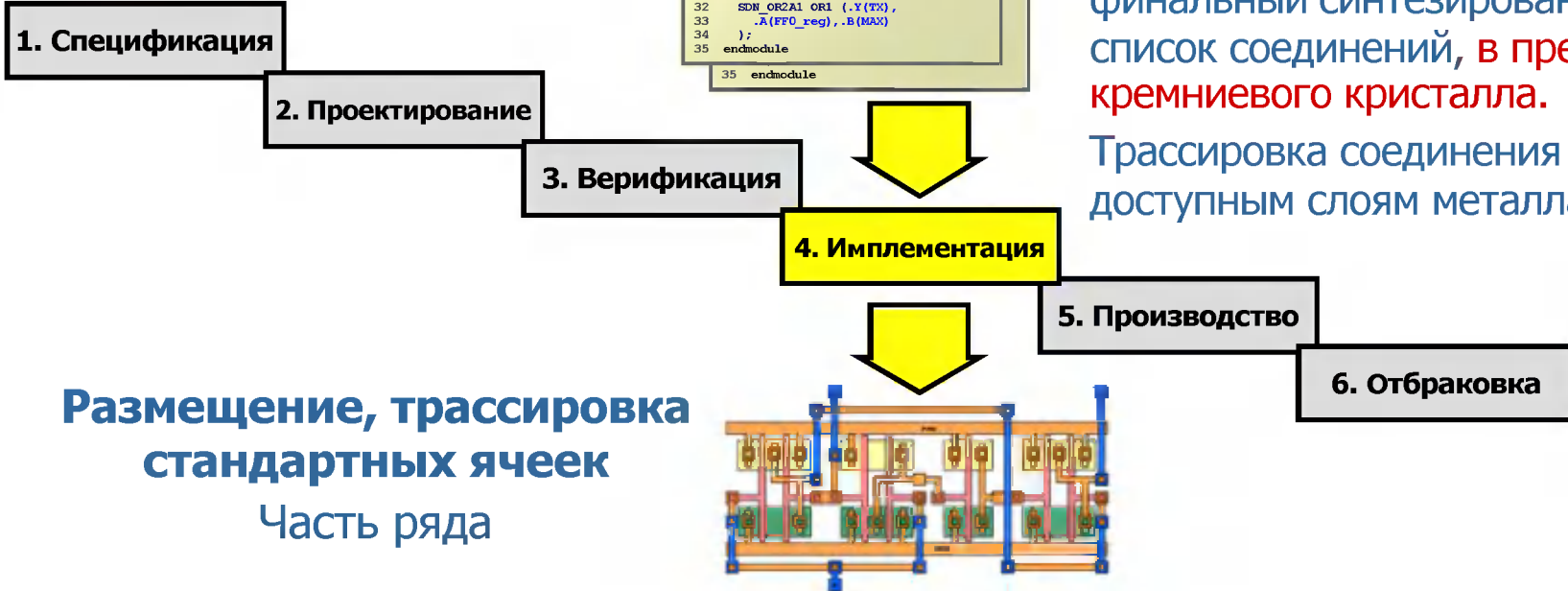
**Чтобы организовать семинары на своем предприятии, свяжитесь с Quantum Consulting по тел: 650/380-9121**

# Стадия имплементации

4-3

Файлы соединений Verilog

```
SERIAL4.vg
1 module SERIAL4 (TX, MAC_DATA, MAX,
2   LOAD_SHFT, CLK
3 );
4   output TX;
5   input [3:0] MAC_DATA;
6   input LOAD_SHFT, MAX, CLK;
7   wire N3, N2, N1, N0, FF3_reg;
8   wire FF2_reg, FF1_reg, FF0_reg;
9
10  SDN_FDPQ_1 FF3 (.Q(FF3_reg),
11   .D(N3), .CP(CLK)
12 );
13  . . .
21  SDN_MUX2_1 MUX3 (.Y(N3),
22   .A(MAC_DATA[3]), .B(1'b1)
23 );
32  SDN_OR2A1 OR1 (.Y(TX),
33   .A(FF0_reg), .B(MAX)
34 );
35 endmodule
```



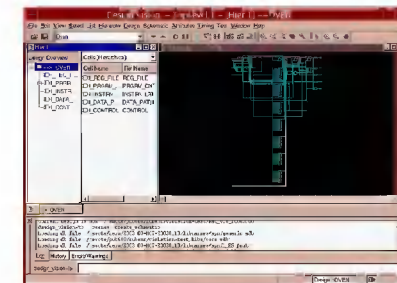
**4. Имплементация:**  
Размещение логических элементов, триггеров, макроячеек, объединенных в финальный синтезированный список соединений, **в пределах кремниевого кристалла.**  
Трассировка соединения по доступным слоям металла.

- Полностью верифицированные блоки проекта передаются для разработки **ТОПОЛОГИИ.**
- Происходит переход из **логической** области в **физическую.**
- Каждый шаг нужно выполнять с учетом особенностей **физической реализации.**

## Финальный логический синтез

- Модели нагрузки линий передач
- Идеализированные тактовые сигналы
- Из логической области в физическую

Типичное  
средство  
синтеза (GUI)



# Модели нагрузки линий передач 4-5

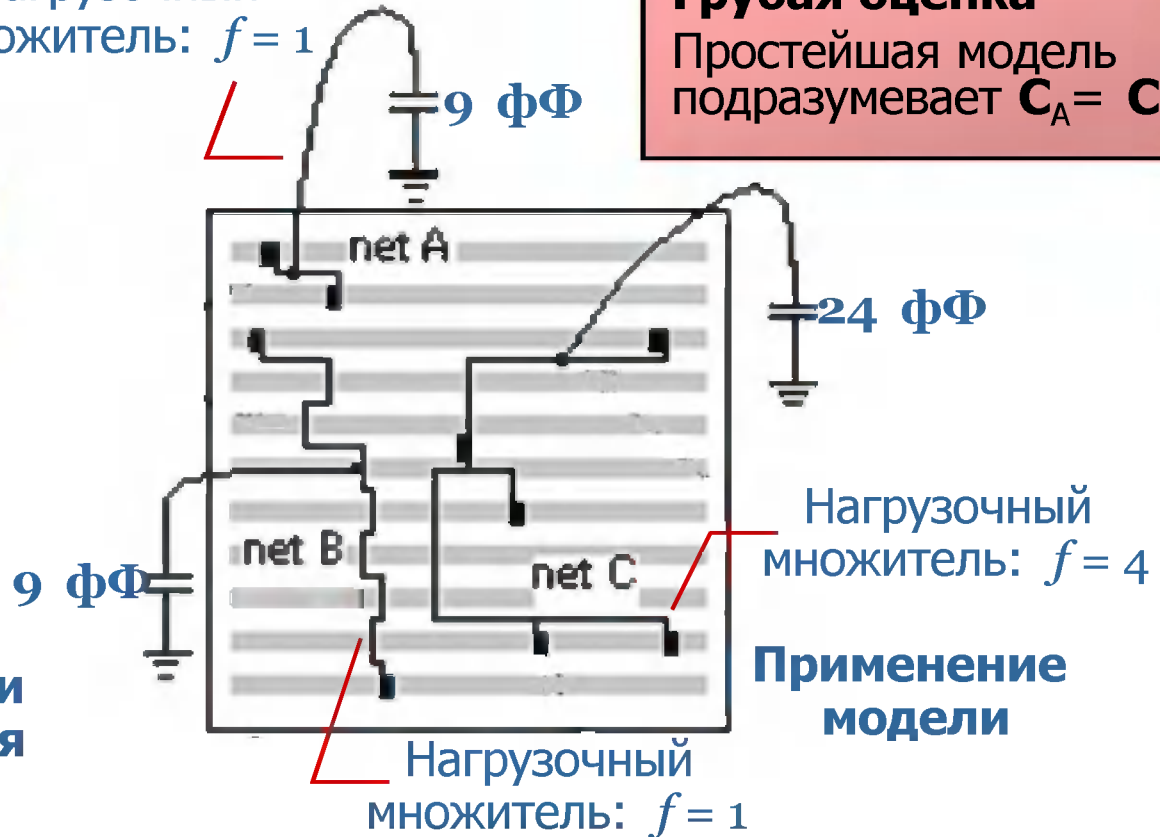
**Внутри RTL блока**  
(~20,000 элементов)

Нагрузочный множитель:  $f = 1$

**Грубая оценка**  
Простейшая модель подразумевает  $C_A = C_B$ .

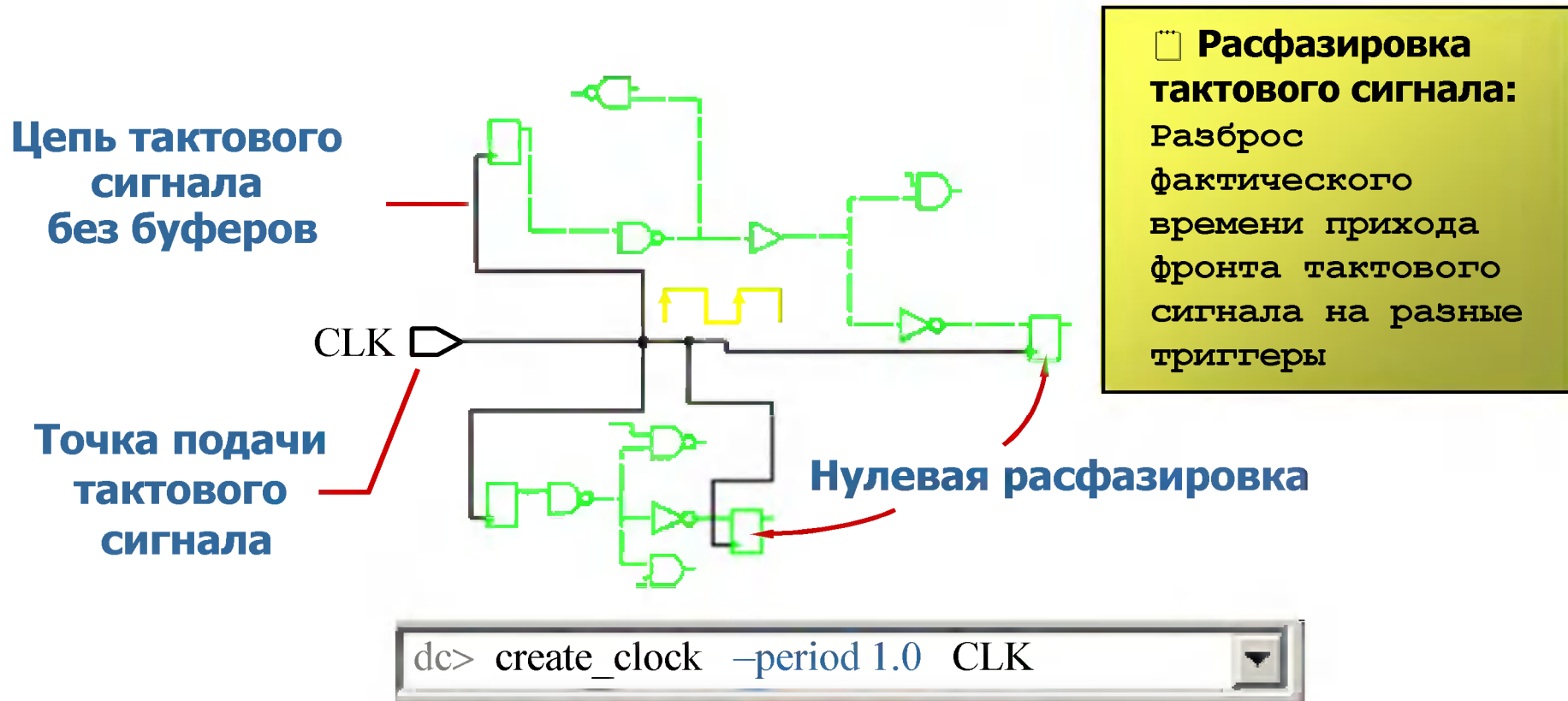
Wire Capacitance			
$f$	10k	20k	30k
1	...	9 fF	...
2	...	12	...
3	...	19	...
4	...	24	...
5	...	30	...

Модель нагрузки линии передачи изготовителя



- Перед трассировкой задержки в соединениях **сложно** рассчитать.
- Библиотечные **модели нагрузки соединений** обеспечивают **оценку**.
- Статистически задержки с цепях оценивается исходя из нагрузочного множителя и размера блока.

# Идеализированные тактовые сигналы 4-6



- Обычно средства синтеза рассматривают тактовые сигналы как **идеальные**.
- **Не** добавляют буферы в CLK-цепи с большим нагрузочным множителем.
- Синтез дерева тактовых сигналов лучше выполнять средствами, которые учитывают особенности **физической реализации**.



# Из логической области в физическую 4-7

Файлы списков соединений  
элементов уровня элементов

✂ EDA средства:

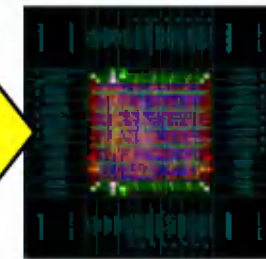
Средства размещения и  
трассировки

—например, *IC Compiler*.

Логический  
синтез

```
SERIAL4.vg
SER 1 module SERIAL4 (TX, MAC_DATA, MAX,
2 LOAD_SHEFT, CLK
3 );
4 output TX;
5 input [3:0] MAC_DATA;
6 input LOAD_SHEFT, MAX, CLK;
7 wire N3, N2, N1, N0, FFS_reg;
8 wire FFS2_reg, FFS1_reg, FFS0_reg;
9
10 SDR_FOPQ_1 FFS (.Q(FFS_reg),
11 .D(N3), .CE(CLK)
12 );
13
14 SDR_MUX2_1 MUX3 (.Y(N3),
15 .A(MAC_DATA[3]), .B(1'b1)
16 );
17
18 SDR_OR2A1 OR1 (.Y(N0),
19 .A(FFS_reg), .B(MUX3)
20 );
21
22 endmodule
23
24 SERIAL4.mdc
25 # TCL SCRIPT
26 # Synopsys Design Constraints
27 # For SERIAL4 Module
28
29 # Clocking constraints:
30 create_clock -period 2.8 CLK
31 set_clock_reach network CLK
32 set_clock_latency 1.8 CLK
33 set_clock_uncertainty 1
34 .1 -setup CLK
35
36 # Timing constraints:
37 set_input_delay -clock CLK \
38 0.75 MUX3[*]
39 set_output_delay -clock CLK \
40 0.8 TX
41
42 # Area constraints:
43 set_max_area 1000
```

IC Compiler



Топология кристалла  
после трассировки

Временные требования  
выполняются:  
Положительный запас —  
или минимум  $\approx 0$  нс.

Скрипт SDC

TSMC  
КМОП 32-нм  
Библиотека ячеек  
(Физическая)

- Файлы Verilog списков соединений уровня элементов передаются в ИСС.
- Превращает схему в **геометрический** план.
- Ограничения проекта передаются с помощью файлов SDC.

## Шаги разработки плана размещения:

- Начальный план размещения
- Рекомендации по разработке плана размещения
- Пробы планов размещения
- Размещение контактов ввода/вывода
- Сеть подачи питания

План размещения  
в офисе



# Начальный план размещения 4-9

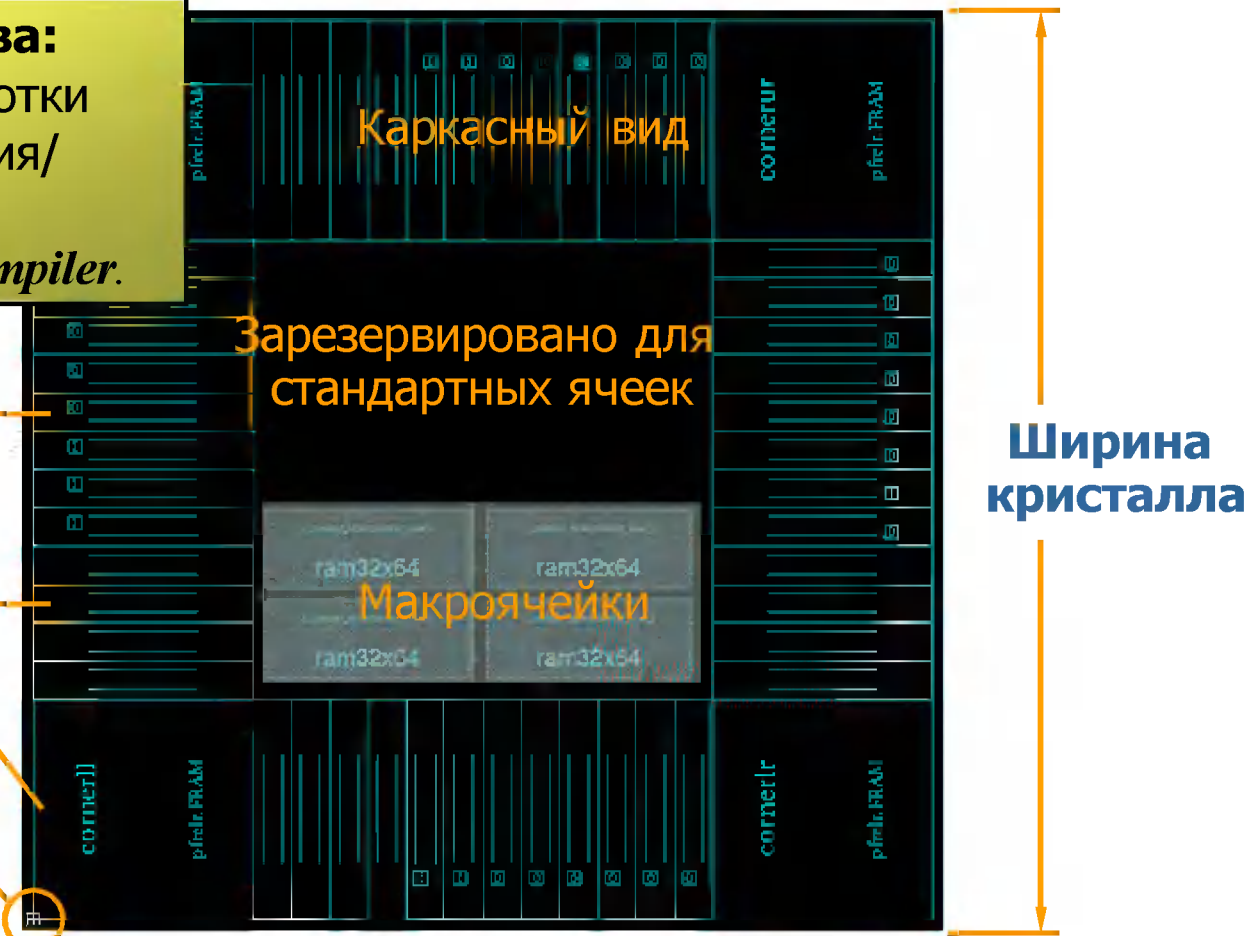
✂ **EDA Средства:**  
Средство разработки  
плана/размещения/  
трассировки  
например, *IC Compiler*.

Контактная  
площадка  
ввода/вывода  
Контактная  
площадка VDD

Угловая  
контактная  
площадка

(x, y)

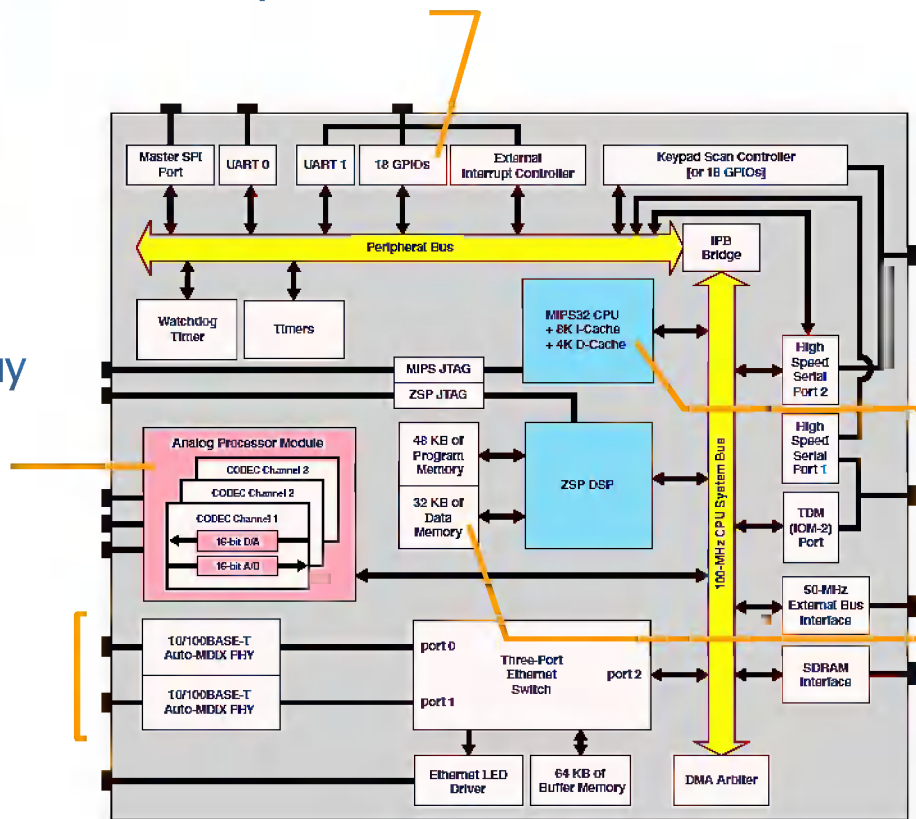
Точка отсчета



- При разработке плана блоки размещаются на **кремниевом кристалле**.
- Между контактными площадками и ядром резервируется место для **каналов трассировки**.

**BCM 1103**  
Телефон  
VOIP

## Кластерные блоки



**Аналоговые макроячейки:**  
Ближе в аналоговому вводу/выводу. Изолированы от цифровой части.

**Контактные площадки ввода/вывода:**  
(Высокоскоростной физический слой)

**Большие IP-блоки:**  
Располагаются в центре для минимизации трассировки.

**ОЗУ программ:**  
Ближе к процессорному ядру.

- План размещения оказывает влияние на возникновение узких мест трассировки и на ее длину.
- Для удовлетворения временным требованиям часто требуется **исследование** плана размещения.


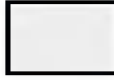
# Пробы планов размещения (1/2)

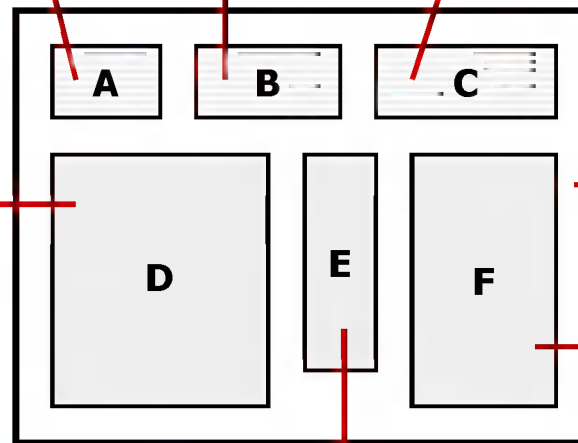
# 4-11

**BCM 1103**  
(Упрощенная)

**ZSP DSP**  
Блок IP

**UART**   **SPI**   **Шина I/F**

 Стандартная ячейка  
 Топологическая макроячейка



**Канал трассировки**

**Аналоговый кодек**

**Область ядер**

**ОЗУ DSP**

Управляется командами:  
Изучение огромного количества **опций** и **переменных** команд ICC может занять существенное время при повторных разработках плана размещения.

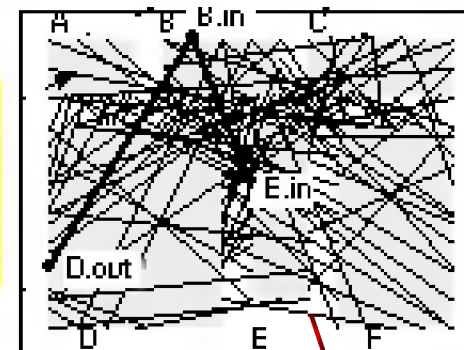
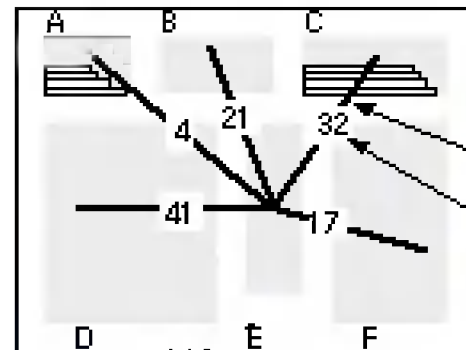
```
icc> initialize_floorplan -io2core 80
```

- Топологические блоки IP **интегрируются** в проектирование микросхемы на этом этапе.
- Изменение в макроячейках означают задержки выполнения календарного плана.

# Пробы планов размещения (2/2)

# 4-12

Попытка 1



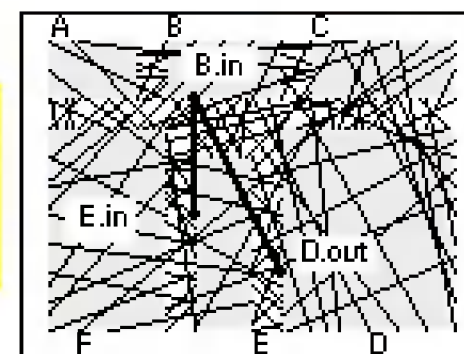
Перегруженный клубок соединений

Отражение  
B



Линии связи

Попытка 2



Улучшенный клубок соединений

Перестановка  
D, F

- Блоки **вручную** поворачиваются, зеркально отражаются, меняются местами.
- Видно, что в попытке 2 клубок соединений менее перегружен.

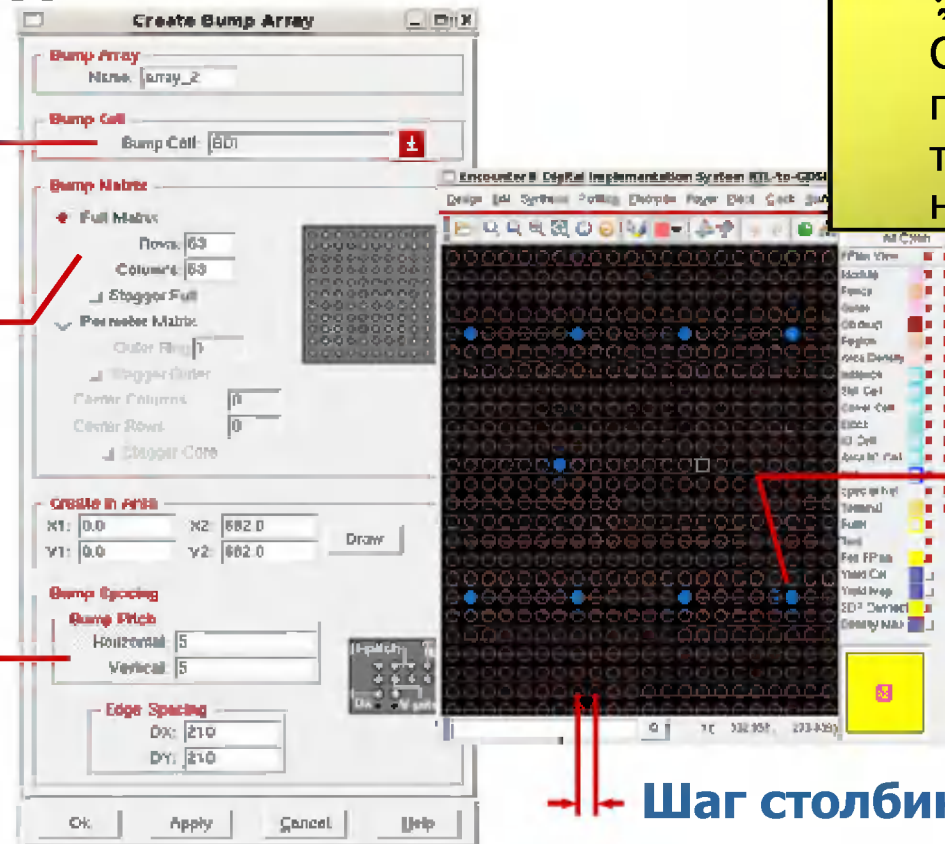
# Размещение контактных столбиков ввода/вывода

# 4-13

Ячейка контактного столбика (Библиотека ввода/вывода)

Размер массива

Шаг столбиков



✘ EDA Средства: Средство разработки плана/размещения/трассировки например, *Encounter*.

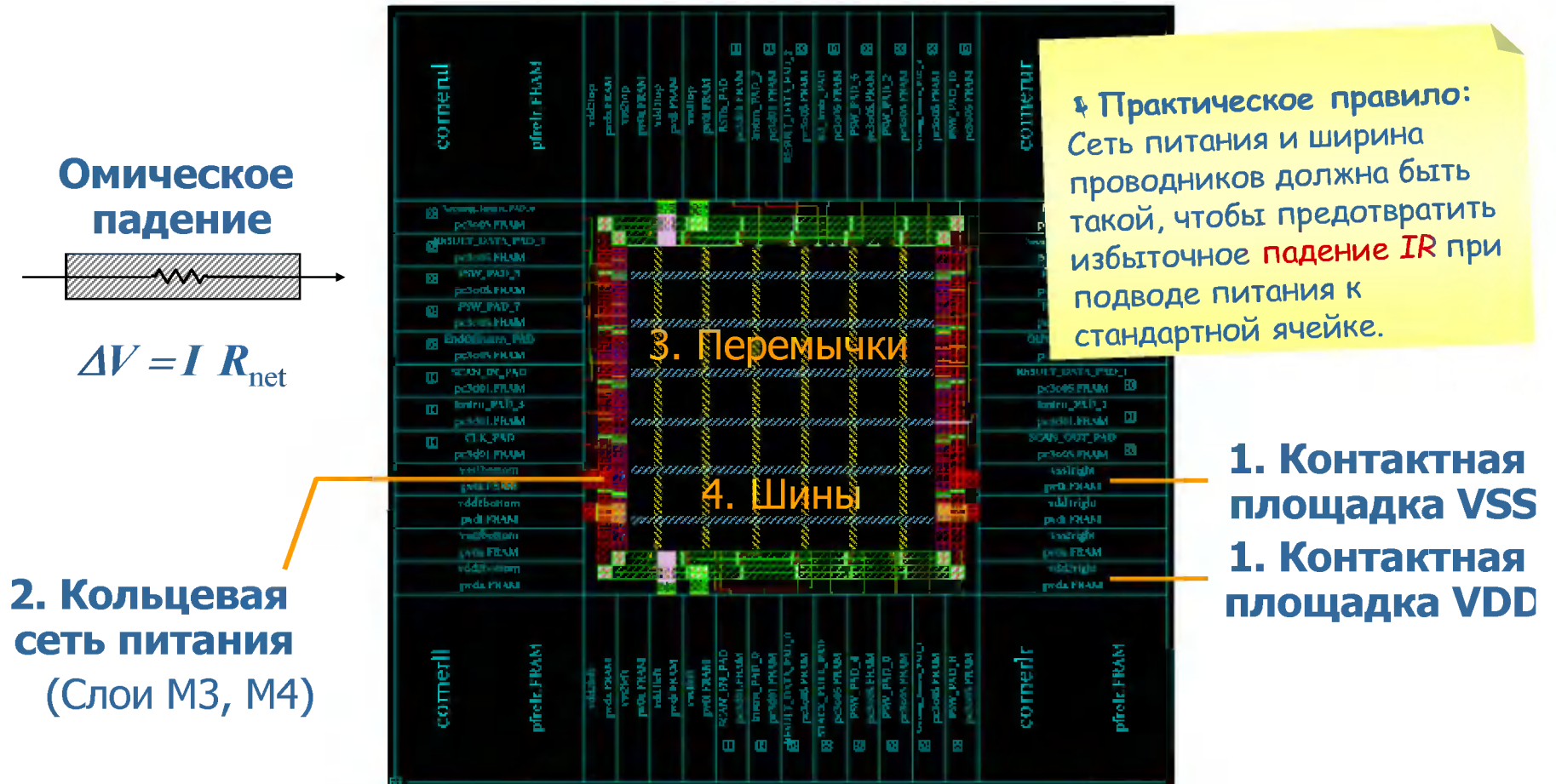
Назначенные контактные столбики (сигналам ввода/вывода)

Шаг столбиков

- Часть разработки плана размещения: расставляются контактные столбики для ввода/вывода, питания: VDD, VSS.
- Контактные столбики, организованные в массив, назначаются внешним сигналам кристалла.
- Буферы ввода/вывода (не показаны) расположены около периферии кристалла.

# Сеть подачи питания

4-14



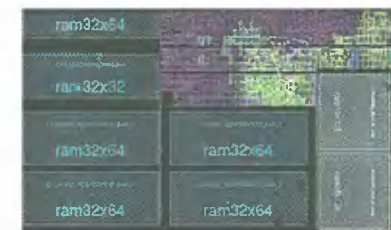
- При разработке плана размещения проектируется **кольцевая** сеть питания и **перемычки**.
- Перемычки в свою очередь подводят энергию к **шинам** локальных стандартных ячеек.



## Шаги по размещению ячеек

- Размещение стандартных ячеек
- Переход из топологической области в физическую
- Финальное размещение ячеек
- Оптимизация временных характеристик
- Синтез дерева тактовых сигналов.

Размещение ячеек  
(между  
макроячейками)



# Размещение стандартных ячеек 4-16

**ВСМ 1103**  
(Упрощенная)

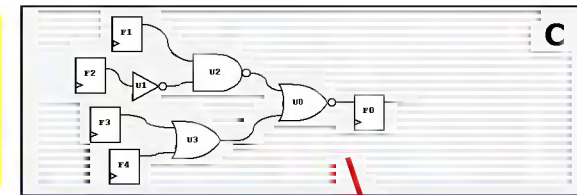
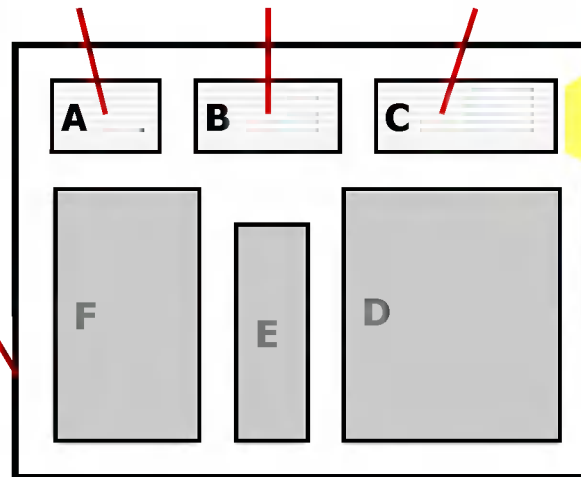
## Блоки RTL:

Определенная пользователем логика или программные IP-блоки



Стандартная ячейка

Область ядер



## Логические элементы:

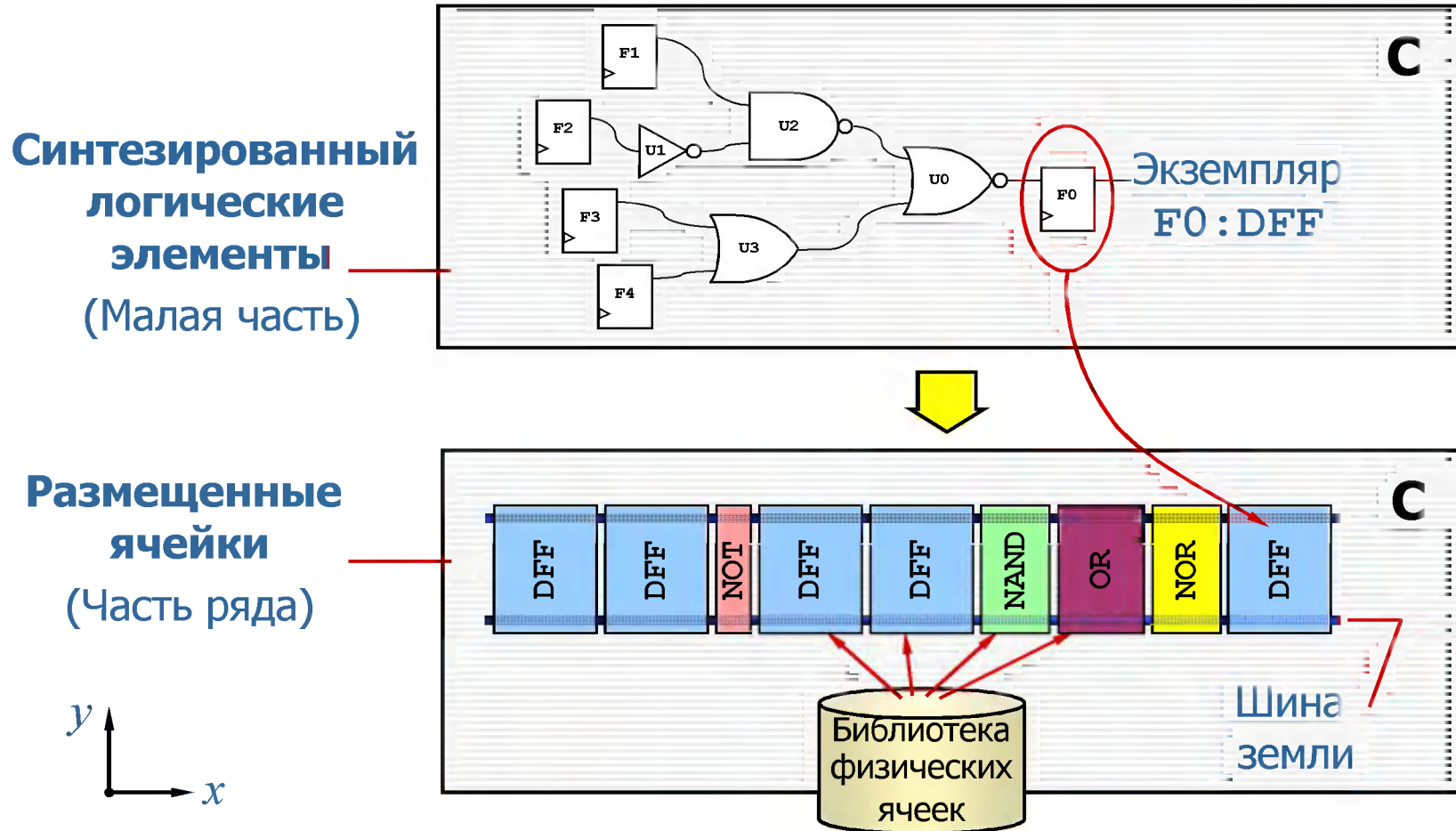
Синтезированный список соединений является только **ТОПОЛОГИЧЕСКИМ**—нет фиксированного ( $x, y$ ) расположения элементов, триггеров.

```
icc> place_opt
```

- Незарезервированная площадь используется для **рядов** стандартных ячеек.
- Размещение стандартных ячеек обычно выполняется **командой**.
- Оптимальное **размещение** является условием эффективной **трассировки**.

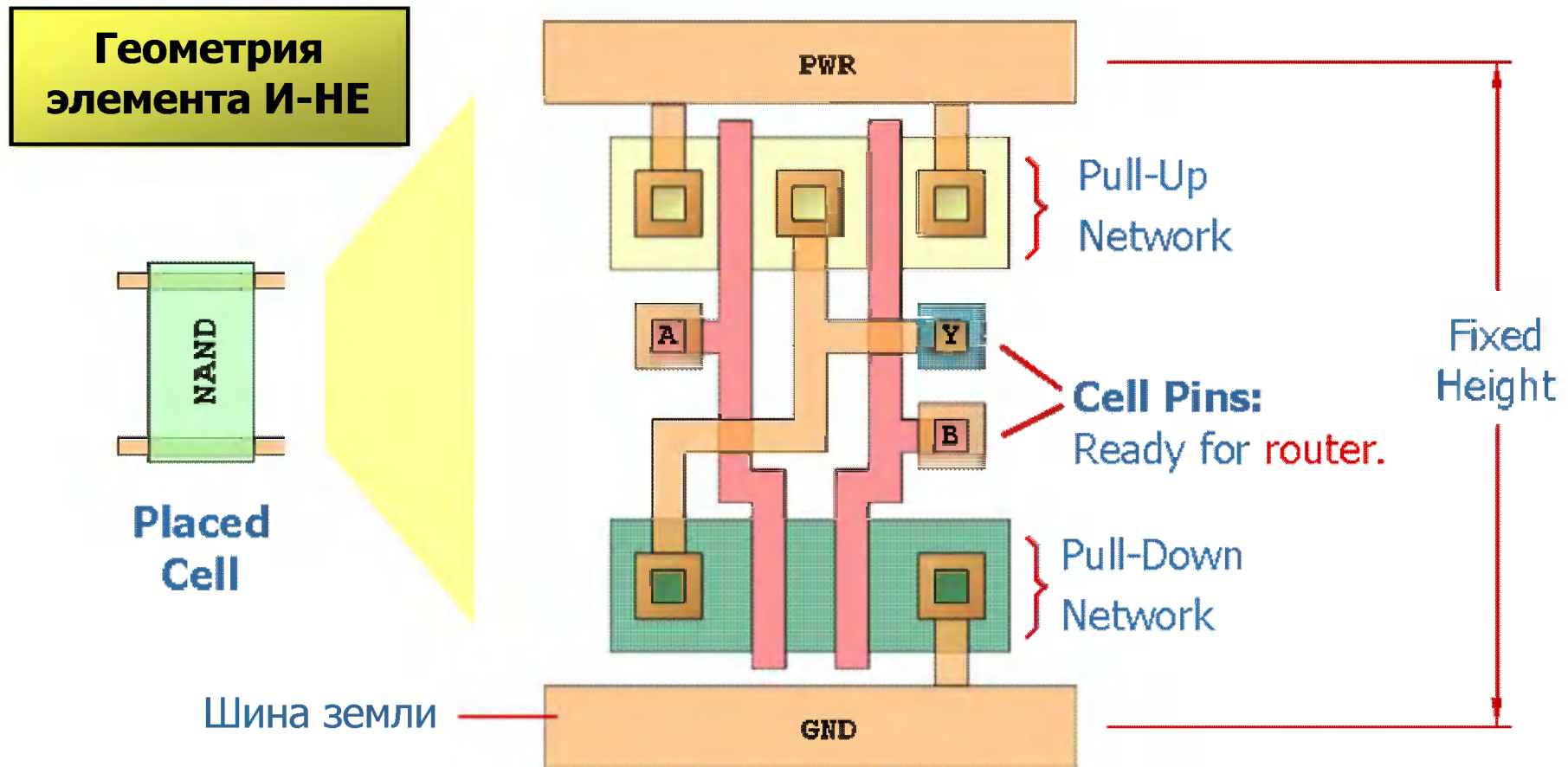
# Из топологической области в физическую

# 4-17



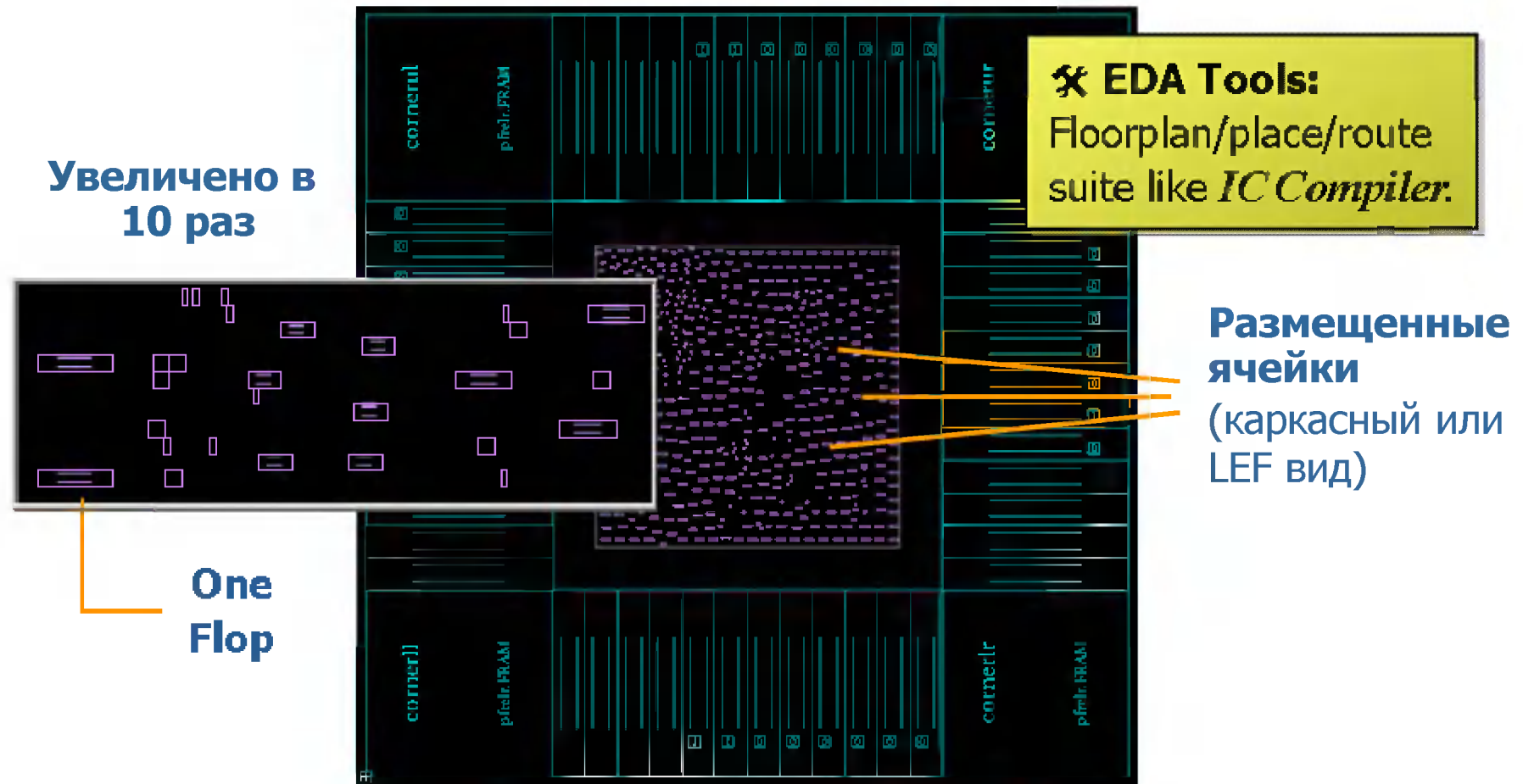
- Во время размещения стандартные ячейки располагаются в **рядах**.
- Напряжение подводится общими шинами **питания, земли**.

# Внутри стандартной ячейки 4-18



- **Геометрия** каждой ячейки описана в физической библиотеке ячеек.
- Трассировщик использует информацию о **расположении** ( $x, y$ ) и слое металла каждого контакта.

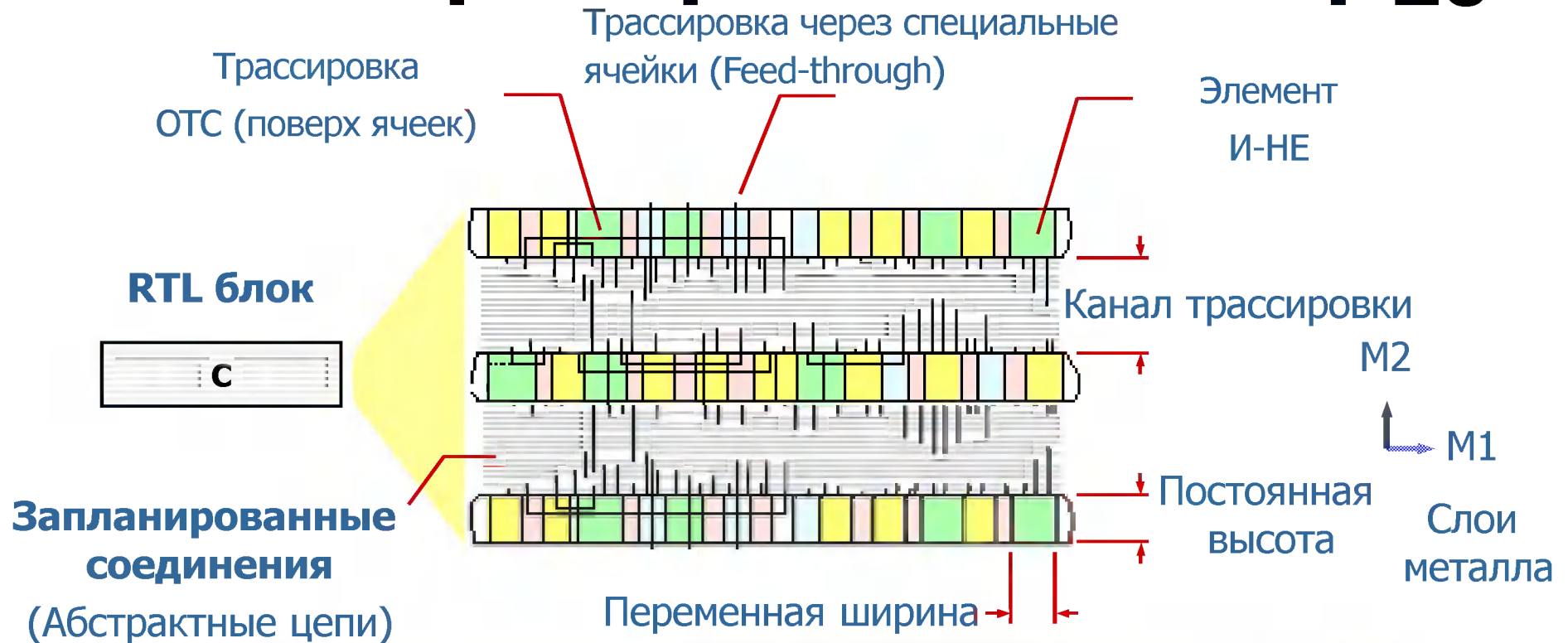
# Окончательное расположение ячеек 4-19



- Каждая стандартная ячейка сейчас **находится** в некоторой  $(x, y)$  позиции.
- Они организованы в горизонтальные ряды, но пока не соединены.

# Каналы трассировки

4-20

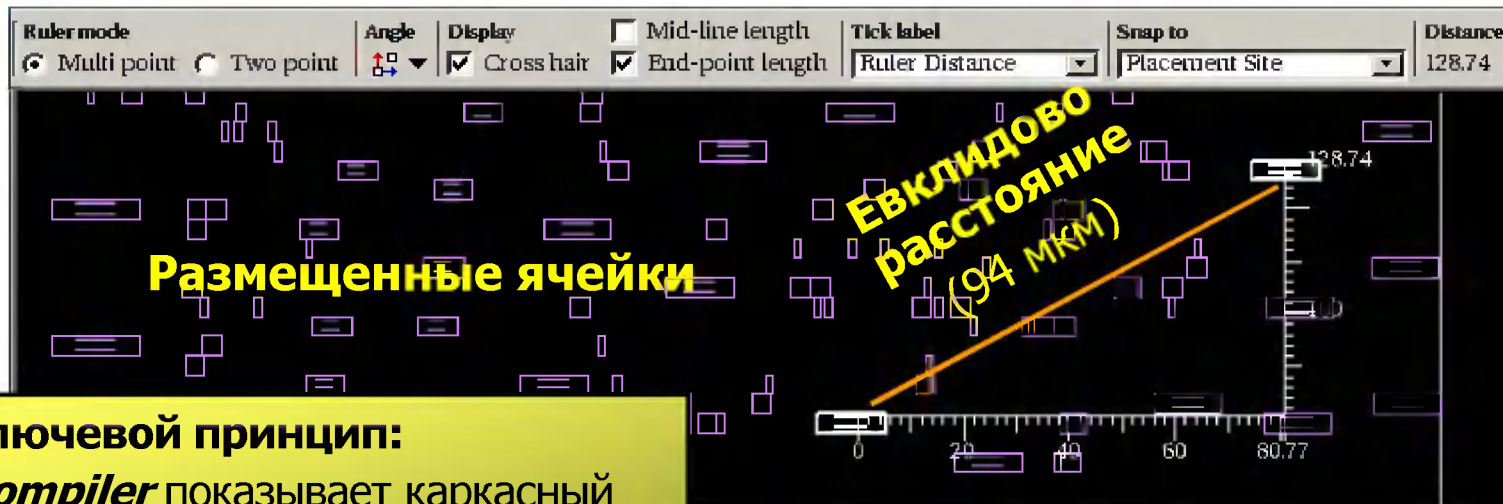


```
icc> initialize_floorplan -core_utilization 0.40
```

- Трассировочные **каналы** не распространены при технологической норме **32 нм**.
- Десять слоев металла **достаточно** для большинства соединений.
- Для использования ядра на **100%** соседние ряды примыкают друг к другу.

# Манхэттенская геометрия

# 4-21



Размещенные ячейки

Евклидово  
расстояние  
(94 мкм)

Манхэттенское расстояние  
(81 + 48 = 129 мкм)

↔ Ключевой принцип:

*IC Compiler* показывает каркасный (FRAME или LEF) вид, не такой детализированный как вид транзисторного уровня (CELL).

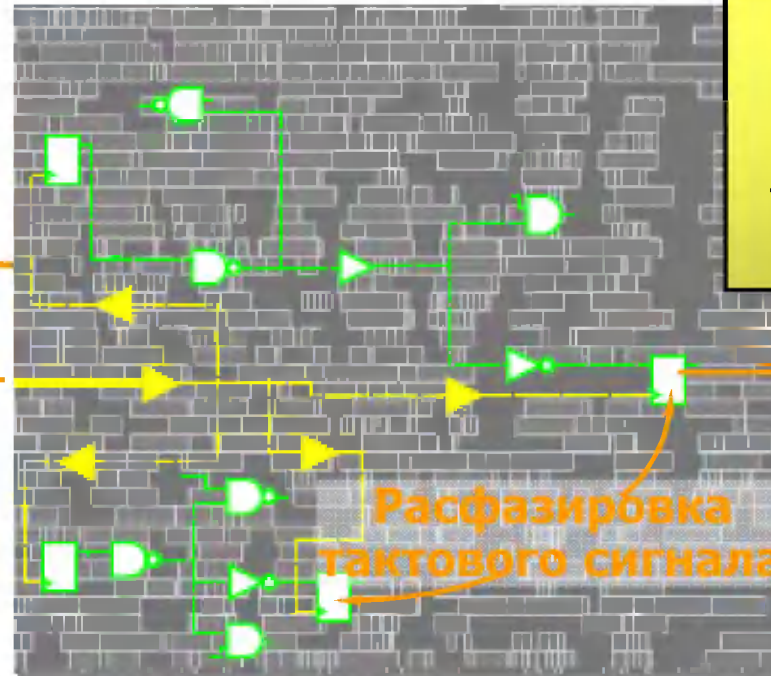
- Трассировка соединений миллионов ячеек является сложной (**NP-полной**) проблемой.
- Вычисления упрощаются при использовании **Манхэттенской** геометрии.
- Перемежающимся слоям металла назначены **предпочтительные** направления (↔ или ↓).
- Таким образом, показанные выше сегмент **север-юг** может быть проложен а слое **METAL2**.

# Синтез дерева тактовых сигналов.

# 4-22

Дерево тактовых сигналов  
(Один уровень)

Вход тактового сигнала  
(create\_clock CLK)



✂ EDA Средства:  
Средство разработки  
плана/размещения/  
трассировки  
например, *IC Compiler*.

Тактируемые  
триггеры  
(передний фронт)

Расфазировка  
тактового сигнала:

```
icc> clock_opt -fix_hold_all_clocks
```

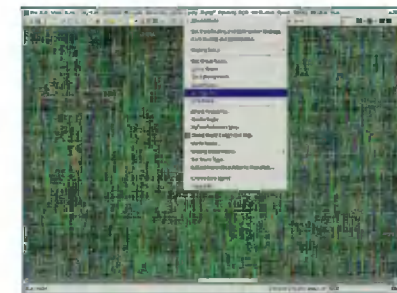
- Каждый триггер уже **расположен** в определенном  $(x, y)$  месте кристалла.
- Средства проектирования могут определить точную Манхэттенскую **длину** каждой линии передачи тактового сигнала.
- Они трассируют линии, при этом пытаются сбалансировать задержки для **минимизации расфазировки**.



## Этапы трассировки

- Глобальная трассировка
- Борьба с перегрузками
- Детальная трассировка
- Необходимость буферизации
- Обратная аннотация

Проект после  
трассировки



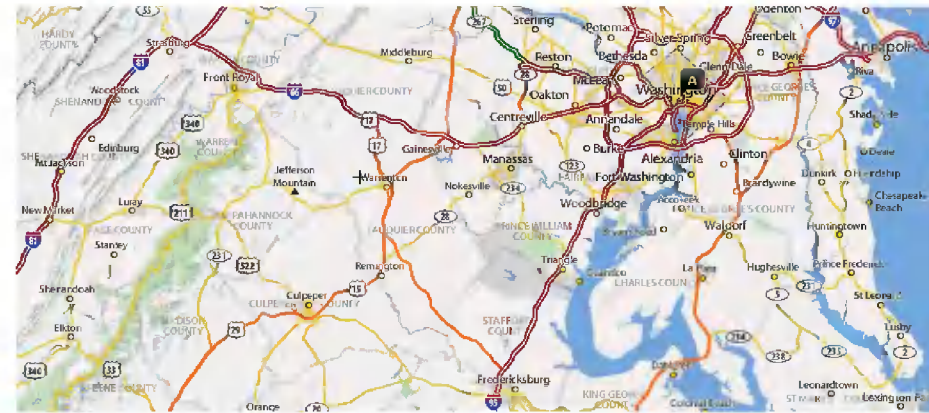
# Глобальная и детальная трассировки 4-24



## Глобальная трассировка:

Определяет оптимальный путь в глобальном масштабе от контакта к контакту, **без учета** ширины линий, локальных препятствие, или DFM (учета технологических требований) . Проявляет узкие места и обходные пути вокруг неявных и определенных пользователем запретов (преград).

- Для управления **сложностью** трассировка выполняется в два этапа.
- Глобальная трассировка выполняется дорожками нулевой ширины от начальной точке к конечной.
- Детальная трассировка учитывает ширину **линий** проектных норм (DRC); вставляет переходные отверстия.



## Детальная трассировка:

Конкретизирует глобальные пути с учетом **DRC** (проектных норм) ширины.

Добавляют переходные отверстия. Выполняет DFM (проектирование с учетом технологических требований). Разрешает нарушения DRC (проектных норм) в зафиксированном окне.

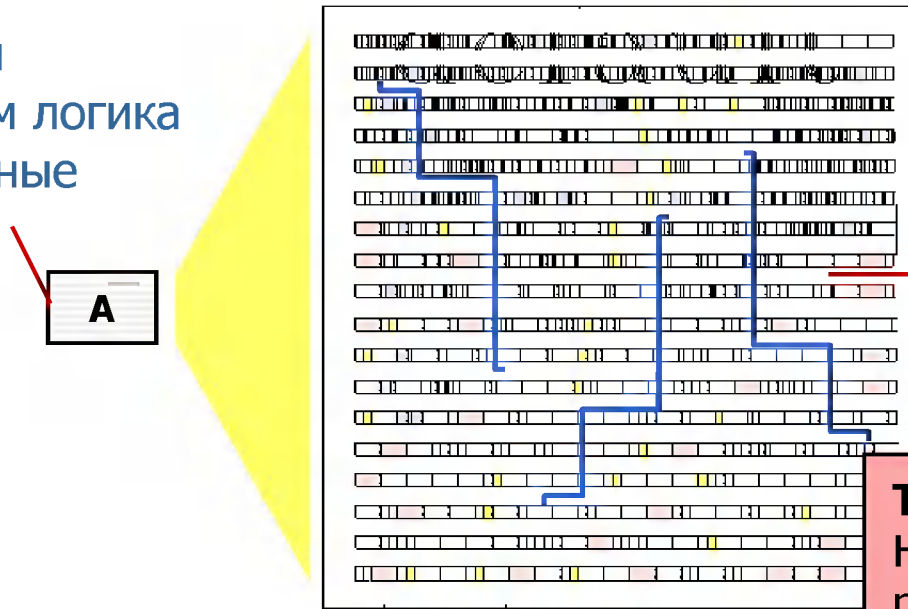
Выполняет оптимизацию временных характеристик.

# Глобальная трассировка внутри блока А

# 4-25

## RTL блок

Определенная пользователем логика или программные IP-блоки.



## Длинные трассы:

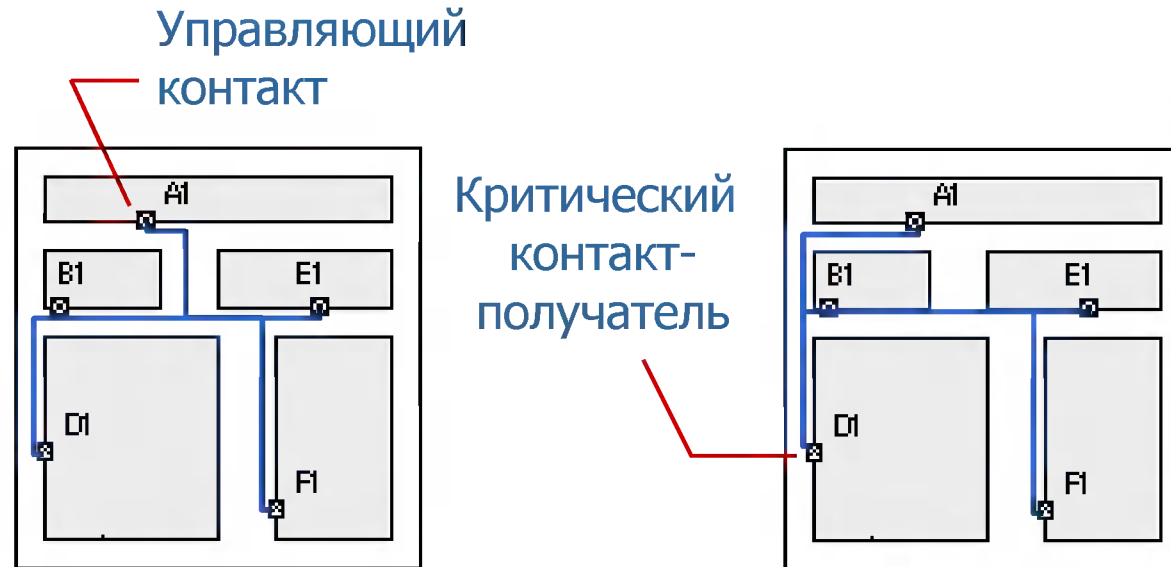
Эти цепи требуют **изменения** слоя и прохода через специальные ячейки (feedthroughs)

## Только дорожки:

Не учитываются детали реализации: ширина дорожки, локальные препятствия, избыточные переходные отверстия.

```
icc> route_opt
```

- Целью глобальной трассировки является минимизация общей **длины цепей**.
- Но **приоритетным** является удовлетворение временным требованиям.



## 1. Минимальная длина

A1 управляет разветвленной на 4 контакта (B1 - F1) цепью.

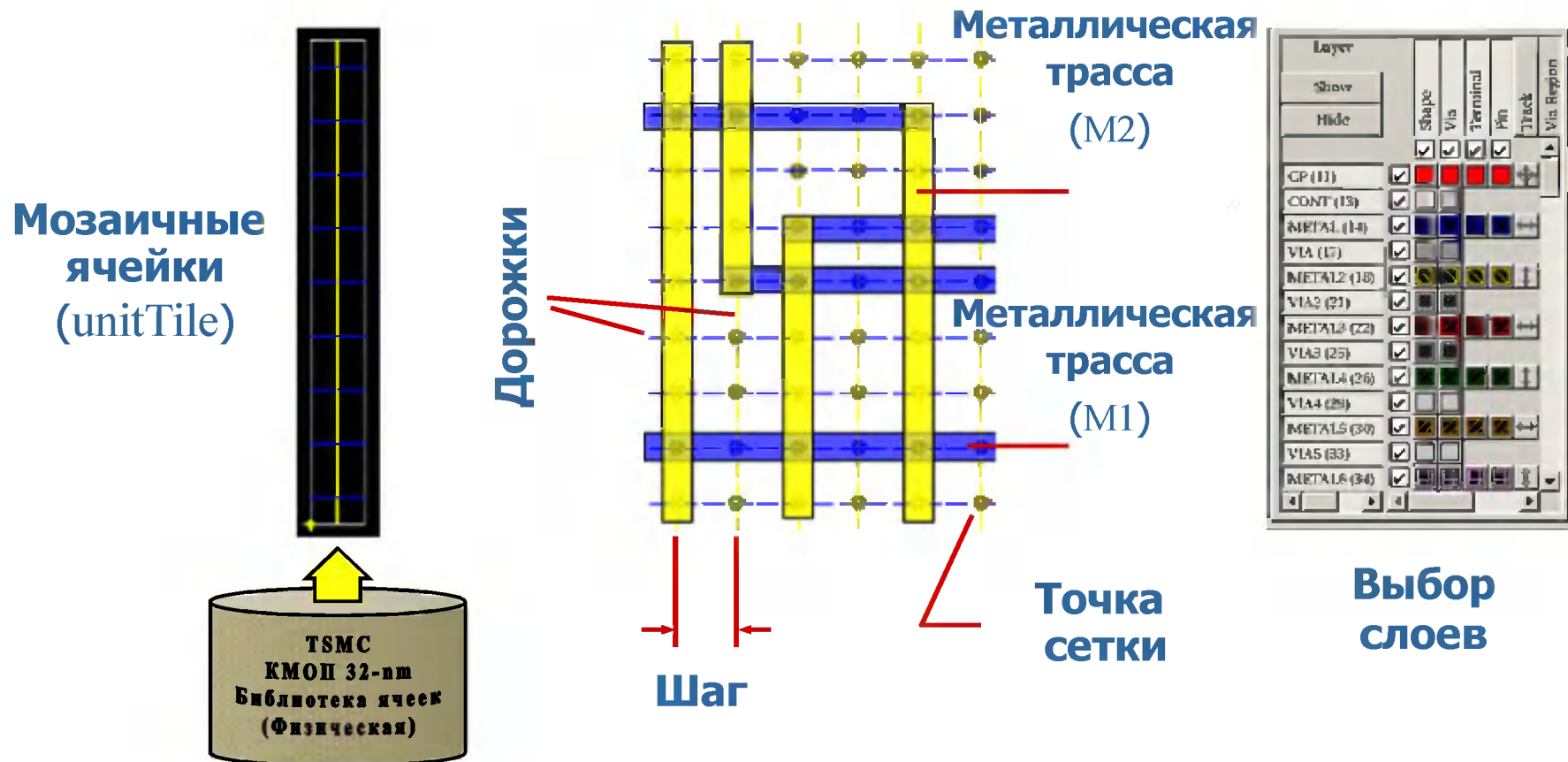
## 2. Минимальная задержка:

Кратчайший путь от управляющего контакта A1 к D1. Большая общая длина.

- В случае 1 глобальный трассировщик минимизирует общую **длину** цепи.
- В случае 2, для удовлетворения **временным требованиям**, оптимизируется сегмент A1–D1
- Первое приближение, т.к. задержки RC пока неизвестны.

# Сеточная трассировка

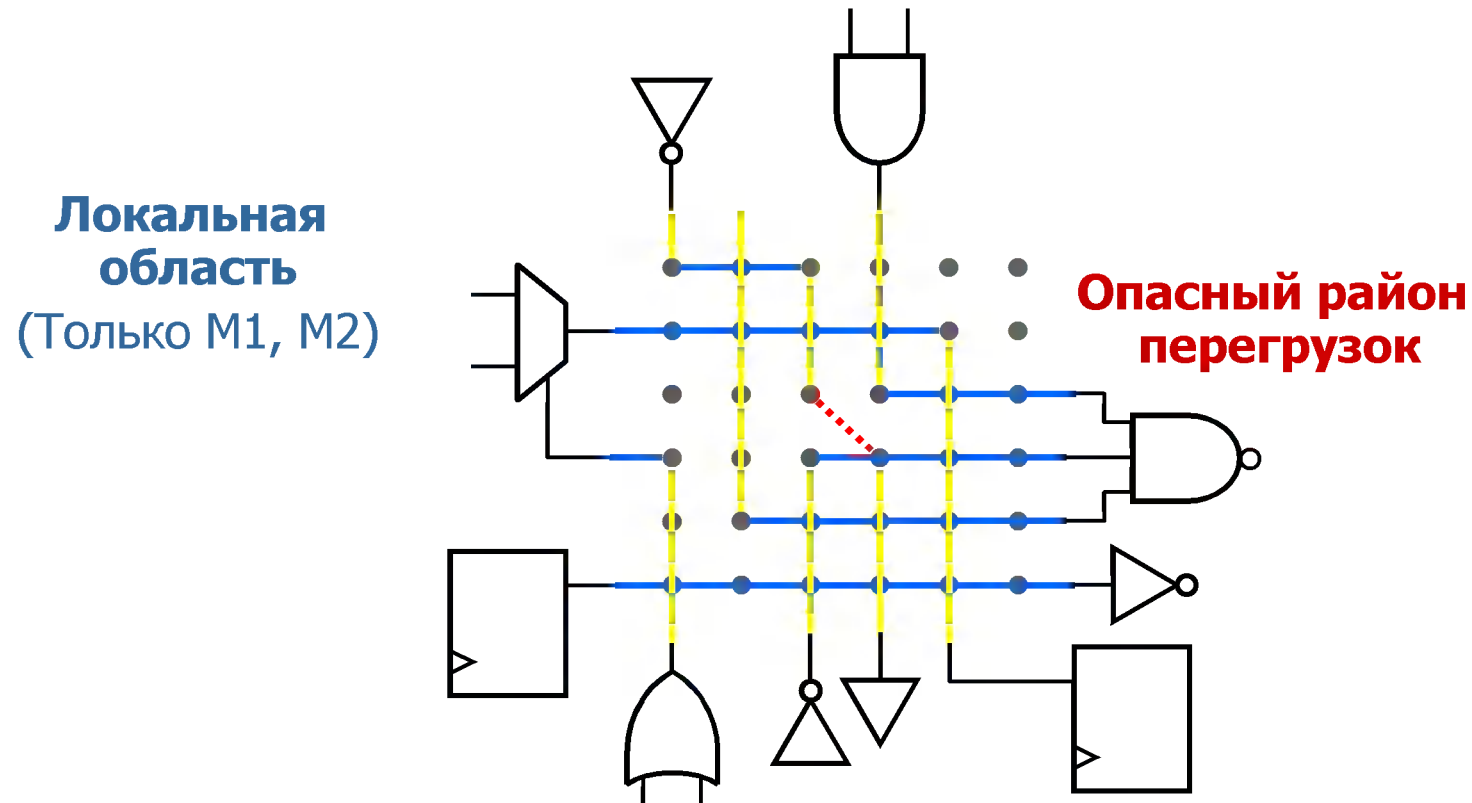
# 4-27



- Трассы располагаются по **центру** линий сетки (дорожек).
- Ширина трасс и их шаг должен удовлетворять 32-нм проектным нормам.
- Глобальный трассировщик располагает дорожки, но не заботится о их реальной ширине.

# Что такое перегрузки?

4-28



- Требования **превышают** возможности прокладки дорожек.
- Длинные **обходные дорожки** могут привести к нарушению временных требований.
- Проектировщик топологии должен находить и устранять **опасные области**.

# Разрешение перегрузок (1/2)

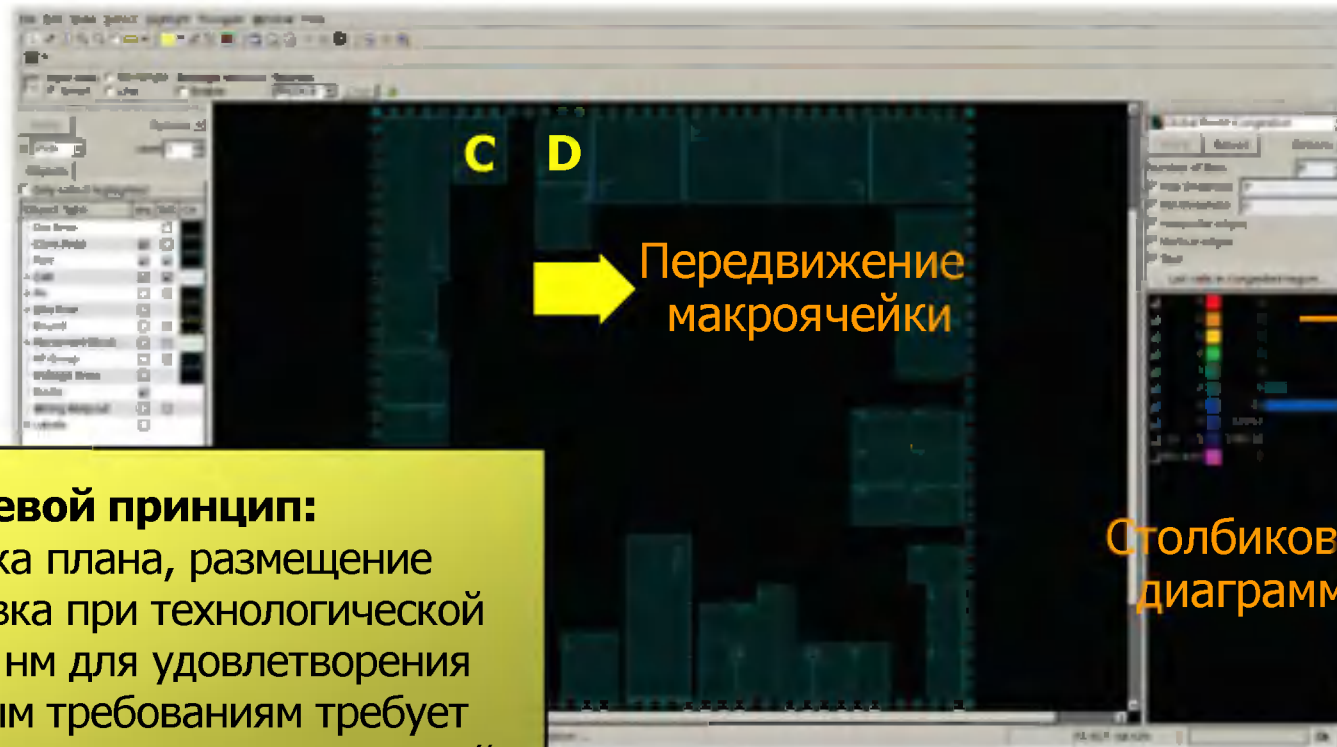
4-29



- Обратите внимание на очень узкий канал между макроячейками C и D.
- Это приводит к появлению опасной области перегрузок при глобальной трассировке.
- Гистограмма показывает много цепей в красных и желтых областях.
- Разрешение опасных областей может потребовать больше итераций разработки плана кристалла.

# Разрешение перегрузок (2/2)

4-30



## ↪ Ключевой принцип:

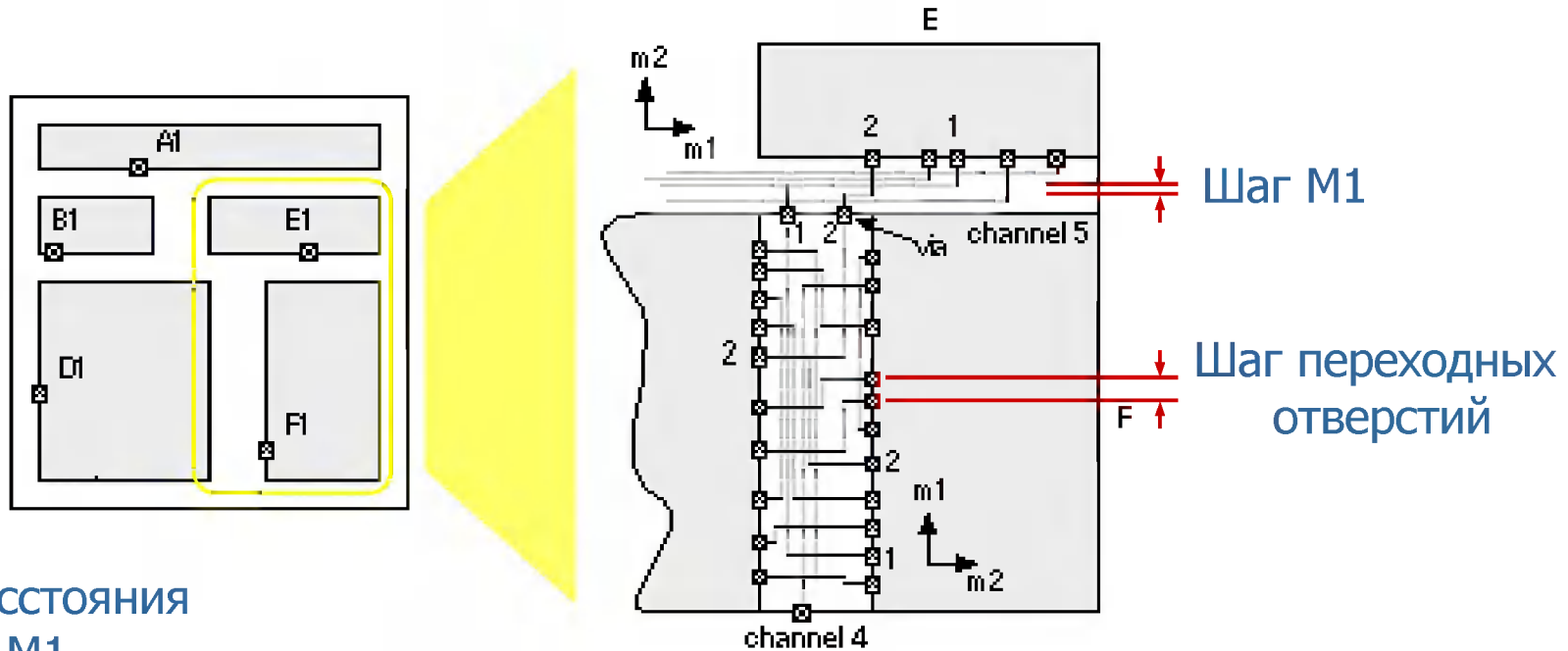
Разработка плана, размещение трассировка при технологической норме 32 нм для удовлетворения временным требованиям требует выполнения нескольких *итераций*.

- Простой сдвиг D вправо улучшает план кристалла.
- На гистограмме больше **нет** цепей в опасных зонах.
- Эта итерация приближает проект к выполнению **временных требований**.

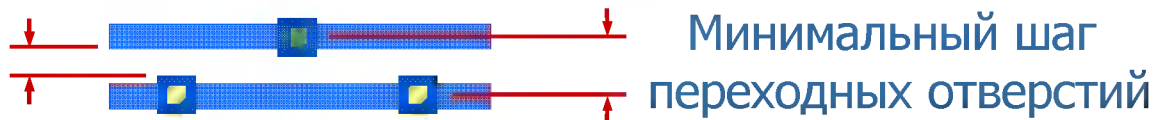


# Детальная трассировка

# 4-31



## Правило расстояния M1-to-M1

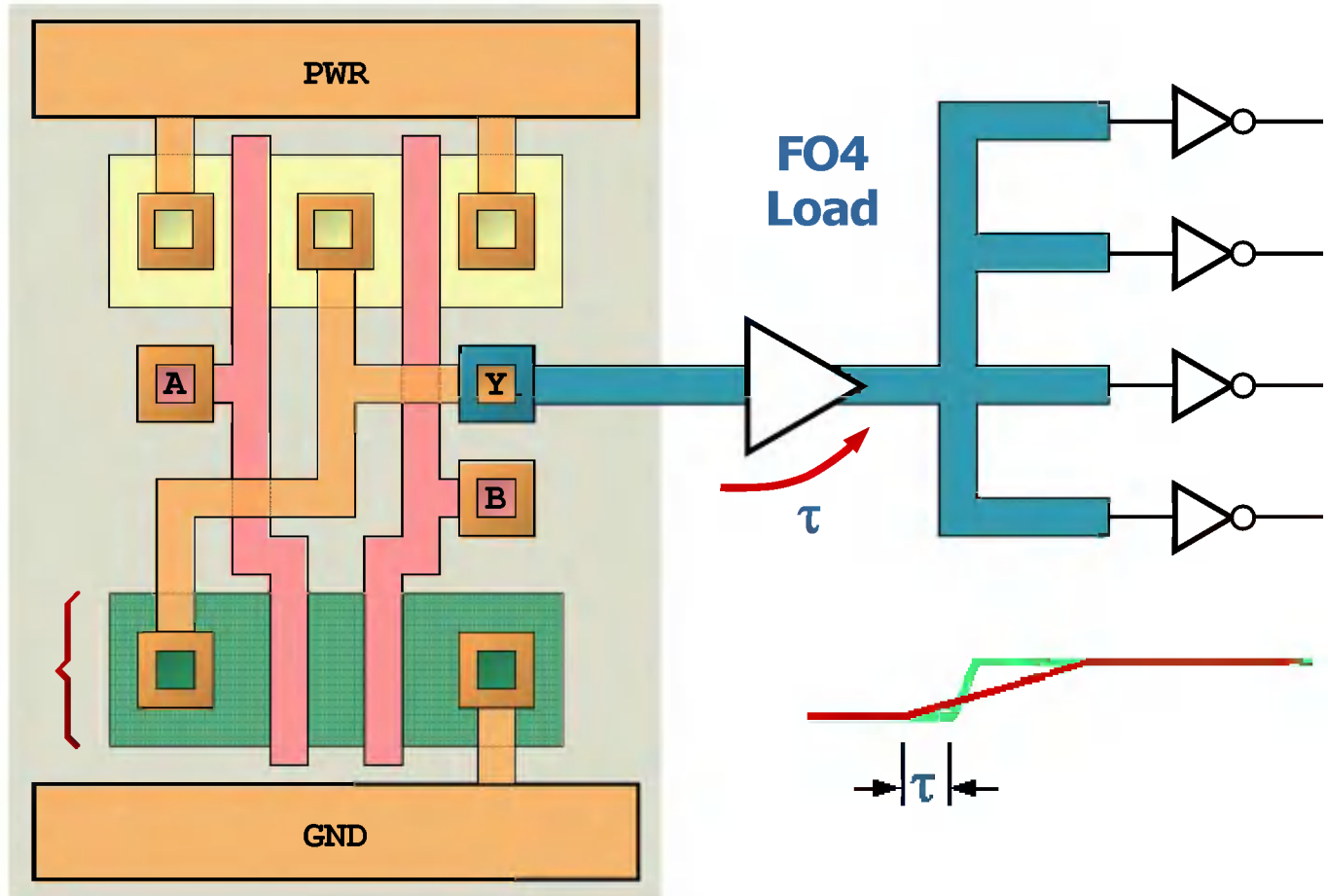


- Детальный трассировщик устанавливает реальные размеры дорожкам, которые проложены глобальным трассировщиком.
- Принимаются определенные в проектных нормах размеры линий и переходных отверстий и расстояния между ними.
- Может заставить выполнять указания **DFM (проектирование с учетом технологических требований)** - например устранить избыточные переходные отверстия.

# Увеличение размеров и буферизация 4-32

**Буфер:**  
Более мощный драйвер управляет большей нагрузкой.

**Увеличение размеров:**  
Увеличение ширины для увеличения нагрузочной способности.



- **ICС выполняет ограниченный набор задач синтеза, подобных увеличению размеров ячеек.**
- **Разрешает большую нагрузку путем добавления мощных буферов.**

# Извлечение параметров RC 4-33

Оттрассированные соединения  
(Реальные проводники—  
характеризуются  $R_{net}$   $C_{net}$ )

Линия питания

⚡ : Требуется много  
времени:

При разработке сложной  
SOC полная итерация  
трассировки может  
занять **16-36 часов**.

Оптимальный канал

M2  
M1  
Слои металла

Шина земли

**Слои металла:**

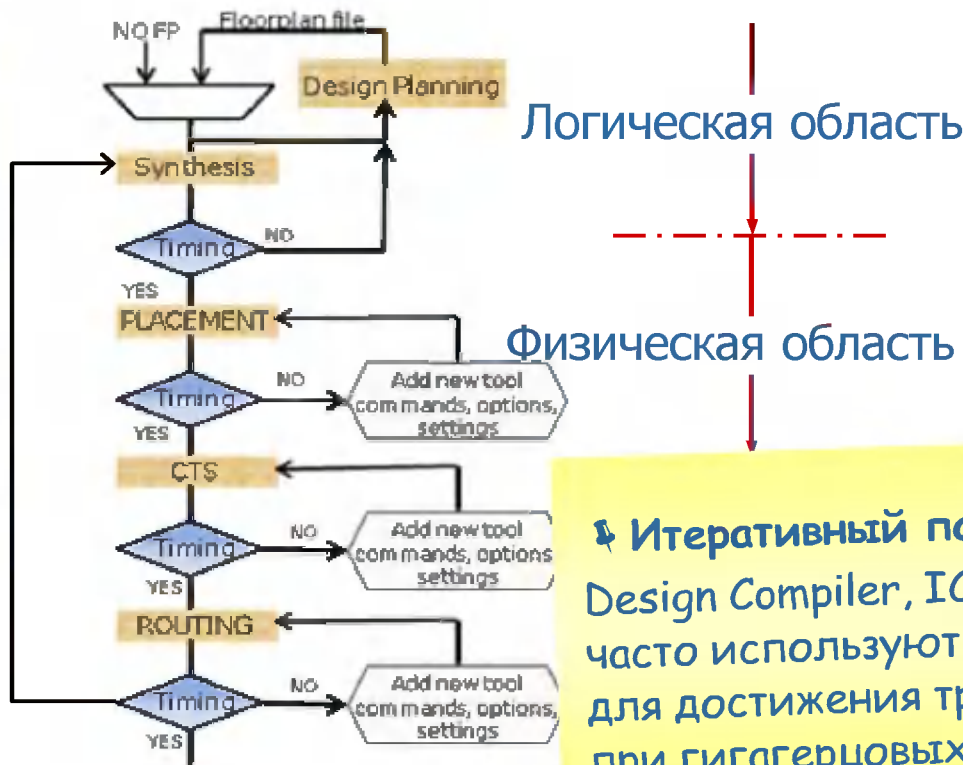
При технологической норме  
32 нм обычно используется  
8-11 уровней металла.

- После детальной трассировки средств разработки позволяют **извлечь** реальные значения RC.
- Серьезные нарушения временных требований могут потребовать **повторного синтеза**.

# Синтез с обратной аннотацией 4-34

## Обратная аннотация:

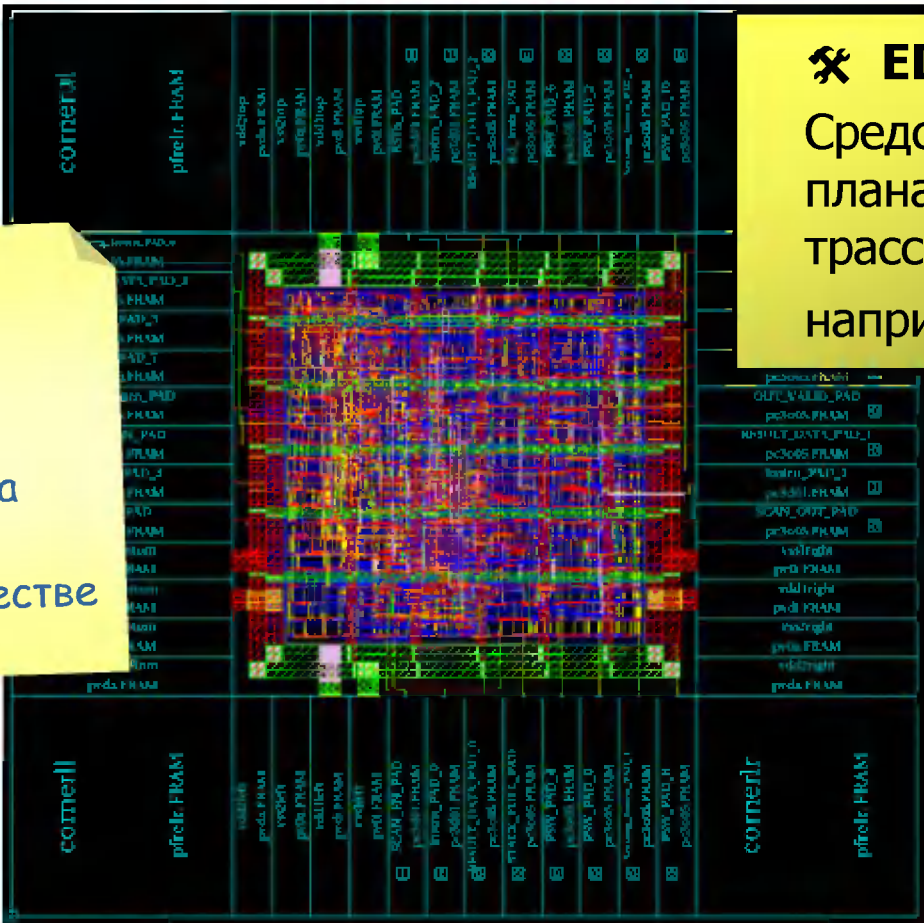
Извлекает *RC* задержки, передает их в средства синтеза для более точного повторного синтеза той же схемы.



- **ICC записывает** задержки в стандартной формате задержек (Standard Delay Format, SDF).
- **DC считывает** их и выполняет точный **повторный синтез**.
- При технологических нормах 32 нм, требуется несколько шагов обратной связи.

# Завершенная трассировка

# 4-35



**✂ EDA Средства:**  
Средство разработки  
плана/размещения/  
трассировки  
например, *IC Compiler*.

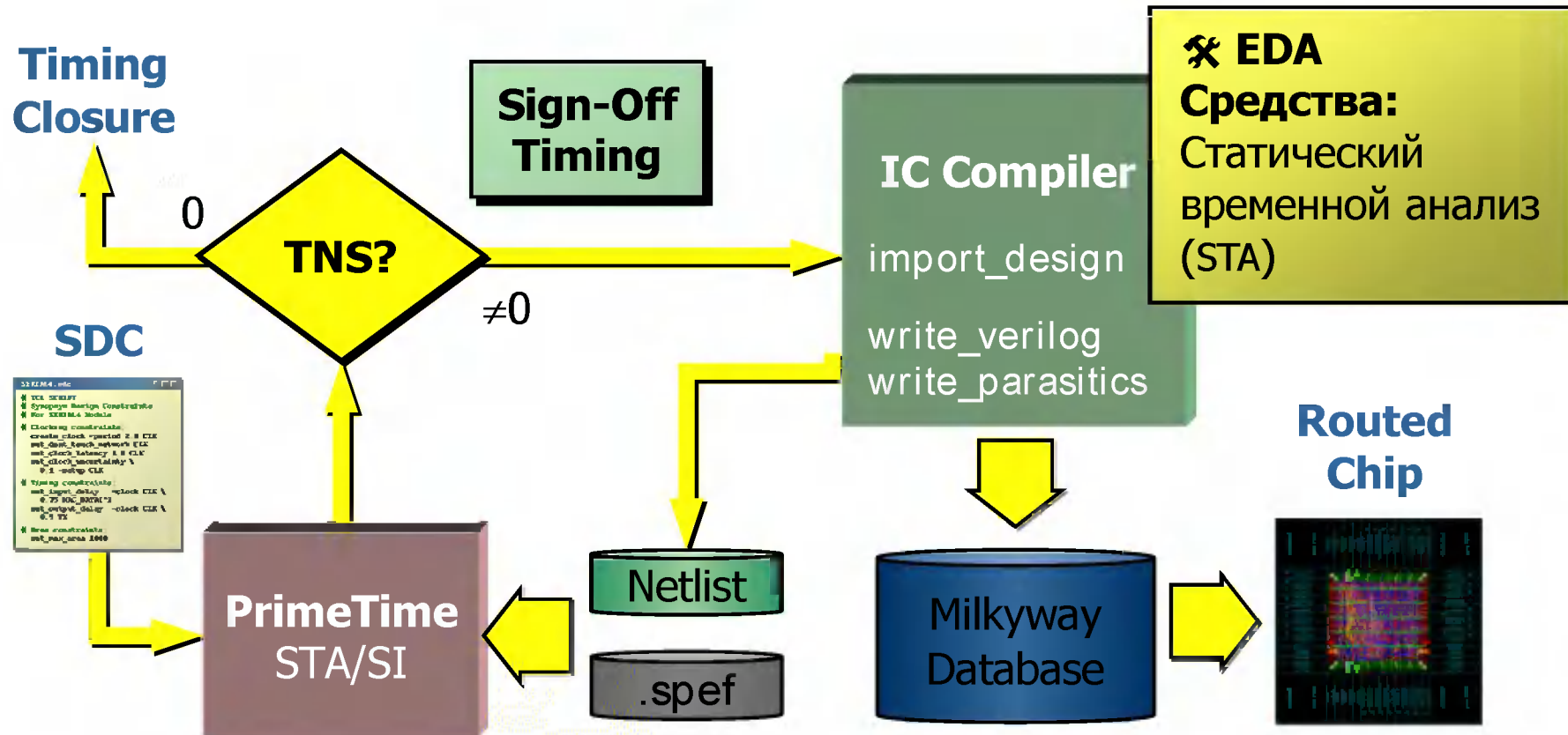
**📌 Качество трассировки:**  
Общая длина всех  
сигнальных трасс на  
кристалле часто  
используется в качестве  
метрики.

- Все соединения контакт-контакт списка соединений теперь **оттрассированы**.
- Перед передачей на производства все временны требования к ASIC должны быть **выполнены**.

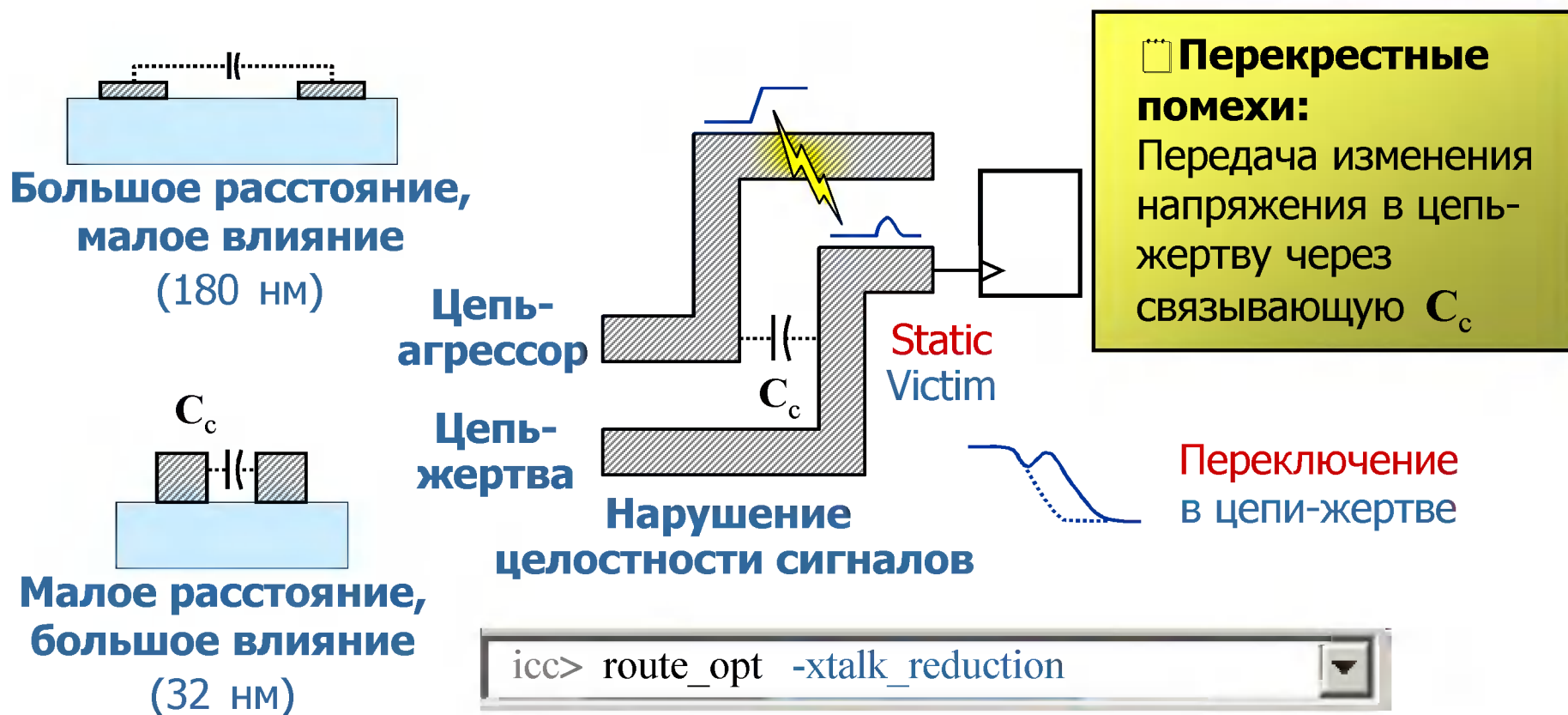
## **Выполнение требований целостности сигналов и временных требований**

- **Выполнение временных требований**
- **Выполнение требований целостности сигналов**
- **Проверка соблюдения проектных норм и исправление ошибок**

# Выполнение временных требований 4-37



- В финальный **список** соединений входят деревья тактовых сигналов, добавленные буферы, увеличенные элементы.
- Паразитные R и C в формате SPEF учитываются в **уточненном списке** соединений.
- Завершающие этап использования средств STA вычисляет задержки; анализирует временные параметры списка соединений.

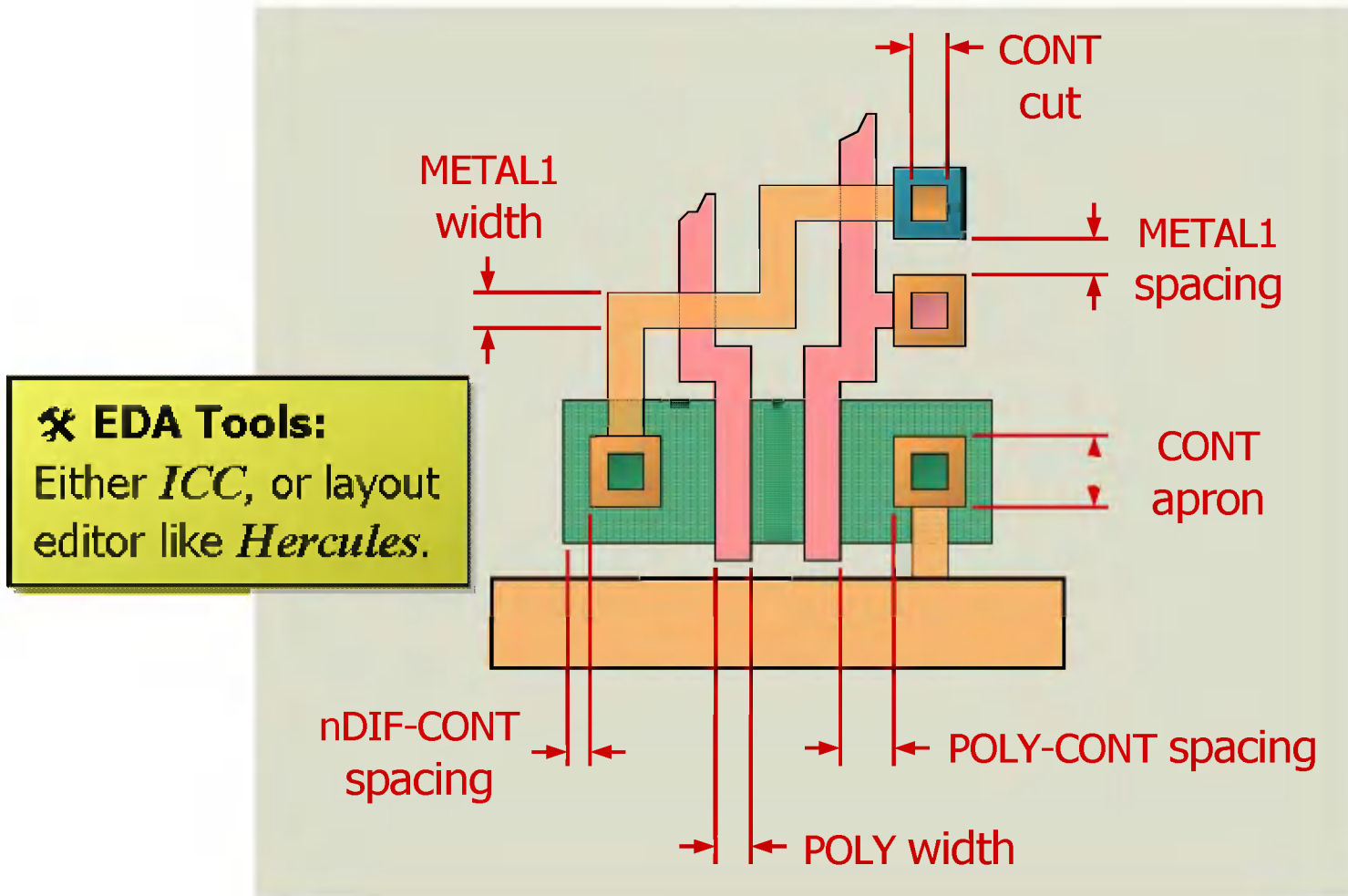


- ВЧ энергия может быть передана через связывающий импеданс.
- При уменьшении масштаба, паразитная емкостная ( $C_c$ ) связь проводников **увеличивается**.
- Трассировщик может разнести проводники, изменить их размеры или поменять их слои.



## **Физическая верификация:**

- **Исправление нарушений правил проектирования**
- **Заключительная проверка DRC/ERC/LVS (правил проектирования/электрических правил/соответствия топологии и схемы)**
- **Исправление оптической близости**
- **Контрольная точка проекта: Передача работы**



- Правила проектирования определяют **высоту** и **ширину** стандартных ячеек.
- Технологичность требует строгого соблюдения всех правил проектирования.

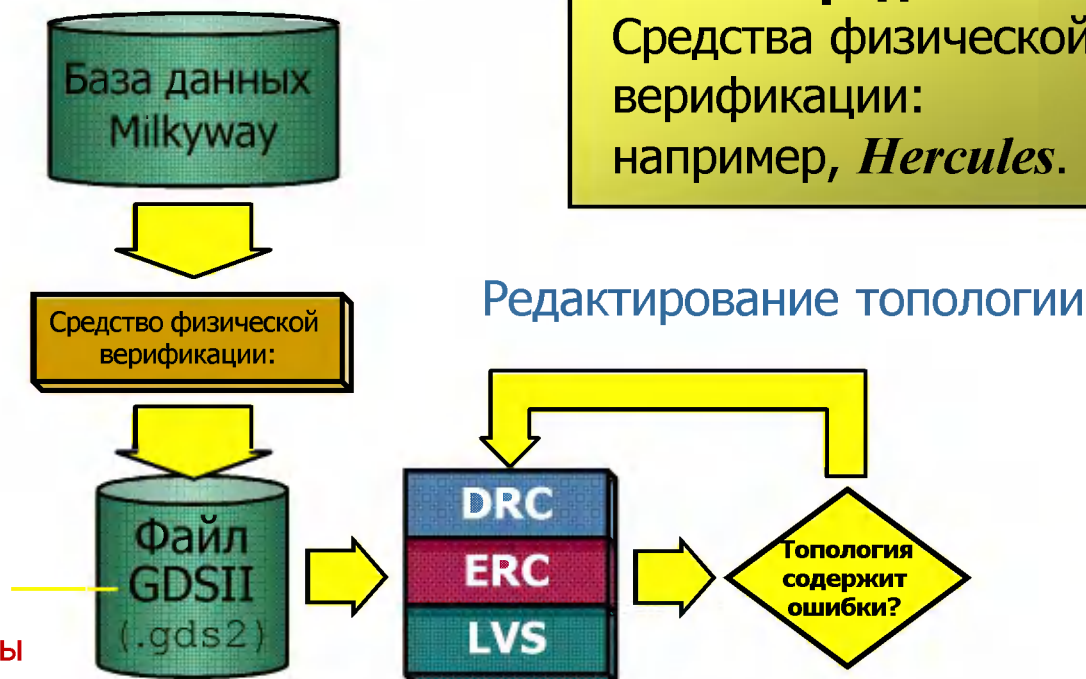
# Заключительная проверка DRC/ERC/LVS

# 4-41

↘ **Формат фотошаблонов:**  
Развивающийся синтаксис OASYS заменяет архаичный GDSII.  
Обеспечивает **меньший** размер файлов, ускоряет обработку.

## GDS:

Формат Graphical Design System (системы графического проекта). **Многоугольники** для маски каждого слоя всего кристалла.  
Размер файла двоичного файла: **терабайты**



## ✂ EDA Средства:

Средства физической верификации:  
например, *Hercules*.

- **Финальная DRC** проверяет **все** правила проектирования топологии данной технологии.
- **ERC** проверяет топологические правила электромиграции и т.д.
- Проверка **LVS** извлекает схему из топологии и сравнивает ее с первоначальной.

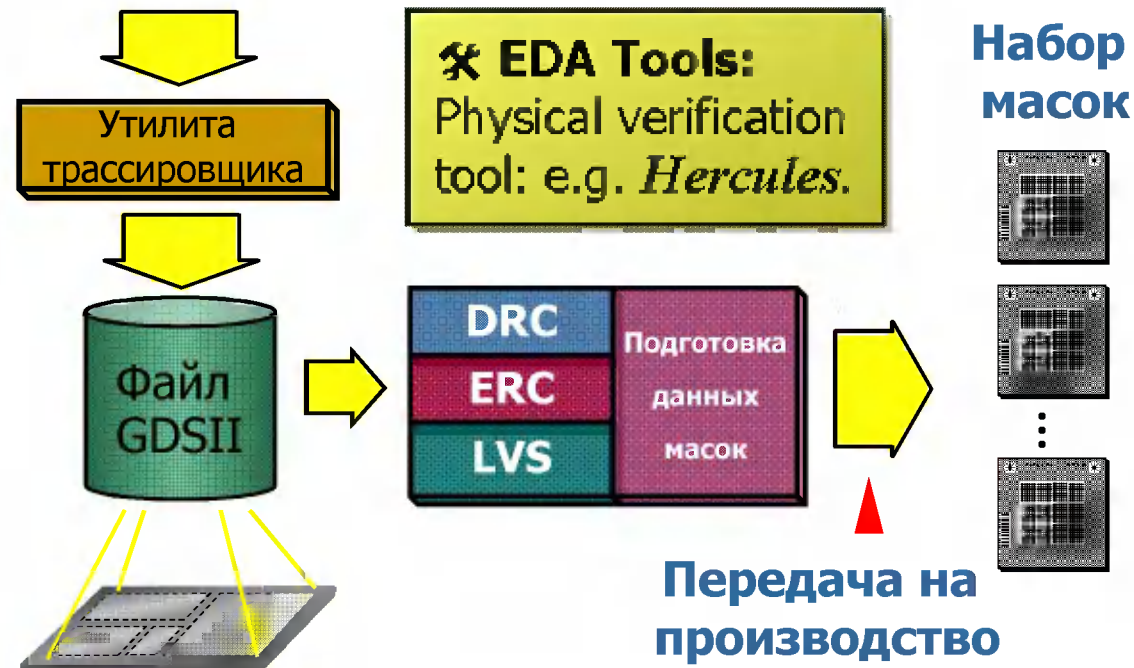
# Контрольная точка проекта: Передача работы 4-42

Этапы передачи проекта на производство

База данных проекта после трассировки

## GDSII:

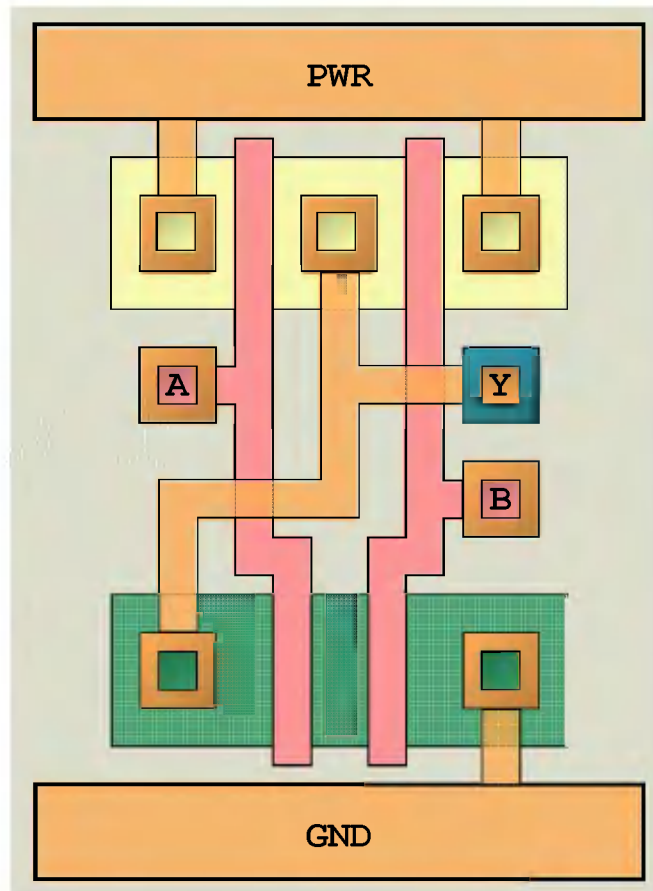
База данных графики послойно записывается в **маски**, которые используются при реализации проекта **кремнии**.



- Огромная база данных геометрии всего кристалл теперь **не содержит** ошибок правил проектирования.
- При передаче на производство финальный файл GDSII пересылается через ftp изготовителю масок.
- Для проверки правильности передачи используются контрольные суммы или повторная передача.

# Что такое GDSII? (1/2)

4-43



```
//GDSII Shape:  
BGNSTR  
AMD POLY1  
BOUNDARY 5  
435 185  
455 185  
. . .  
475 365  
475 385  
. . .  
455,415  
. . .  
435 515  
415,515  
. . .  
415,395  
. . .  
435 185  
ENDEL  
ENDSTR
```

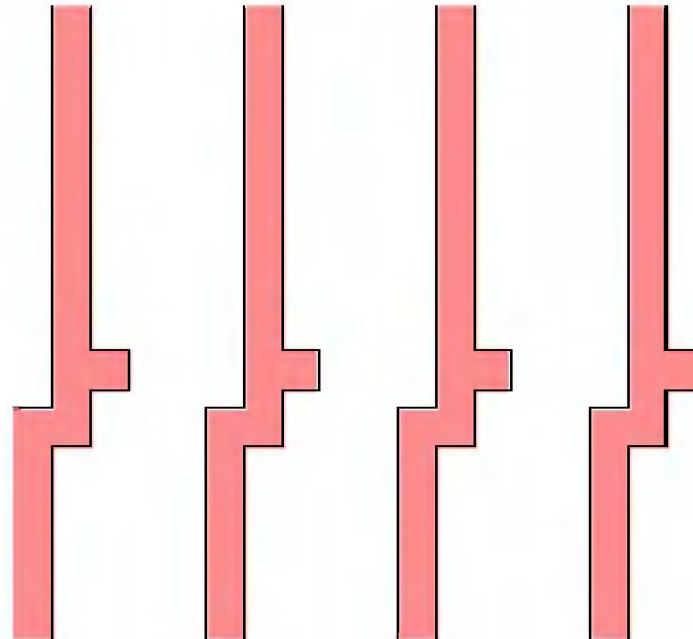
- **Формы GDSII** являются границами (BOUNDARY) и путями (PATH) фиксированной ширины.
- Граница (BOUNDARY) является **замкнутым контуром**, имеющим три или более вершин.

# Что такое GDSII? (2/2)

4-44

```
//GDSII Layout:
```

```
BGNSTR  
NAME=BLOCK  
  SREF POLY1  
    [1, 0]  
  SREF POLY1  
    [2, 0]  
  SREF POLY1  
    [3, 0]  
  SREF POLY1  
    [4, 0]  
ENDSTR
```



- **Ссылка SREF** помещает копию структуры в точку с координатами (x, y).
- Каждый конструктивный элемент GDSII представляет собой запись байтов **переменной длины**.
- **Потоковость** обозначает запись файла на несколько катушек ленты.

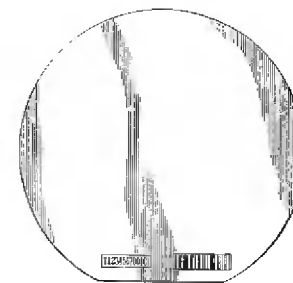
---

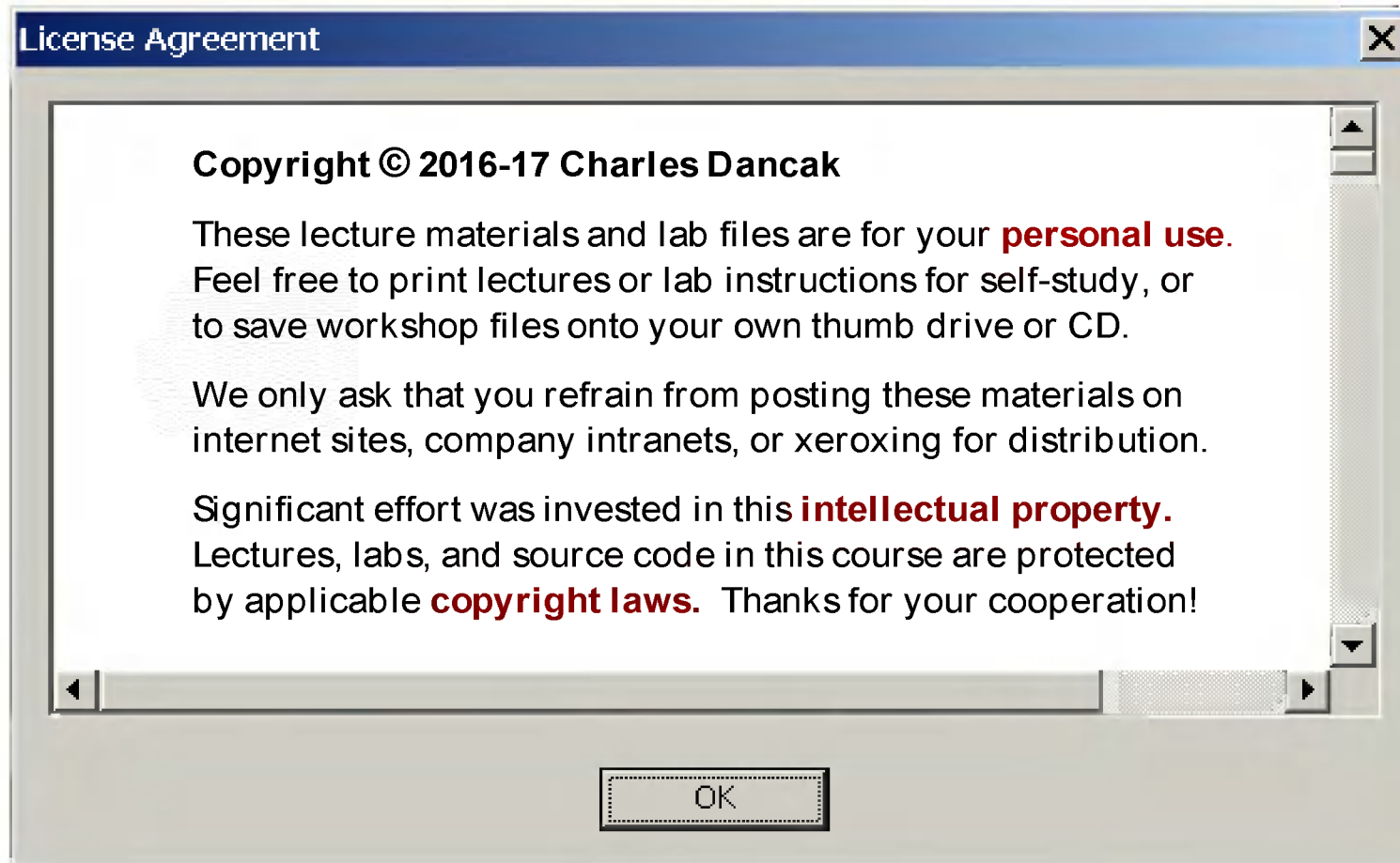
# Developing the Nanometer ASIC

From Specs to Silicon



## 5. Стадия производства



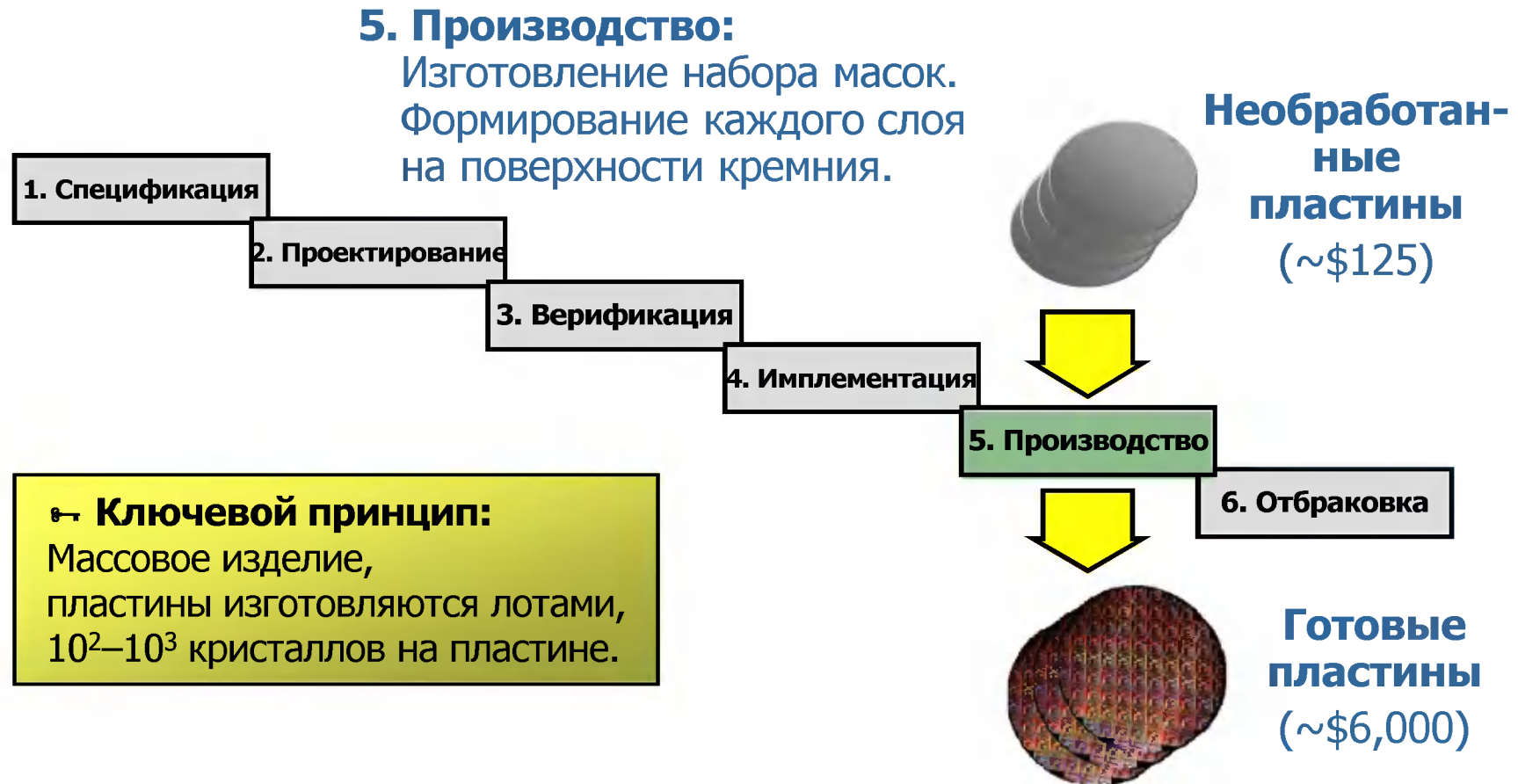


**Чтобы организовать семинары на своем предприятии, свяжитесь с Quantum Consulting по тел: 650/380-9121**



# Стадия производства

5-3

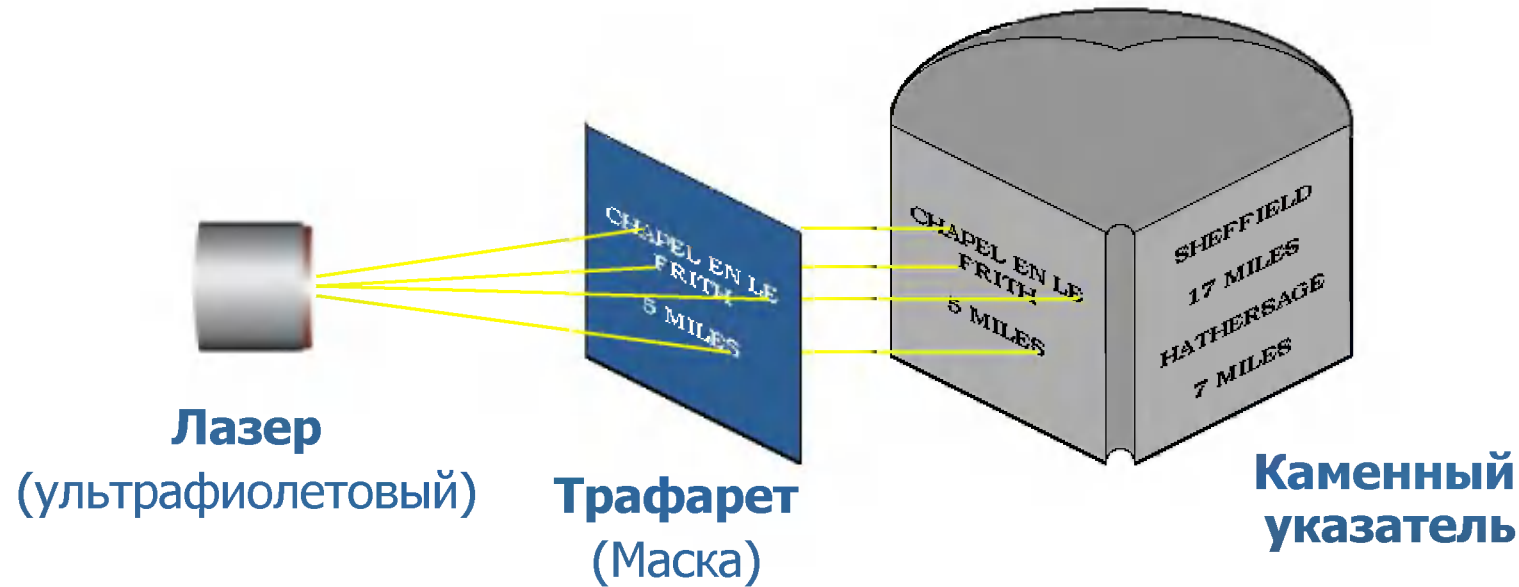


- Полностью подготовленные проекты **реализуются в кремнии** на предприятии производству микросхем.
- Продолжительность цикла по изготовлению обычно около **6–8 недель**.

## **Изготовление масок и их использование**

- **Аналогия с литографией**
- **На предприятии по производству масок**
- **Отображение данных проекта на набор масок**
- **От масок к пластине**
- **Устройство последовательного шагового экспонирования**
- **Фундаментальный предел**

# Аналогия: Печать на камне 5-5

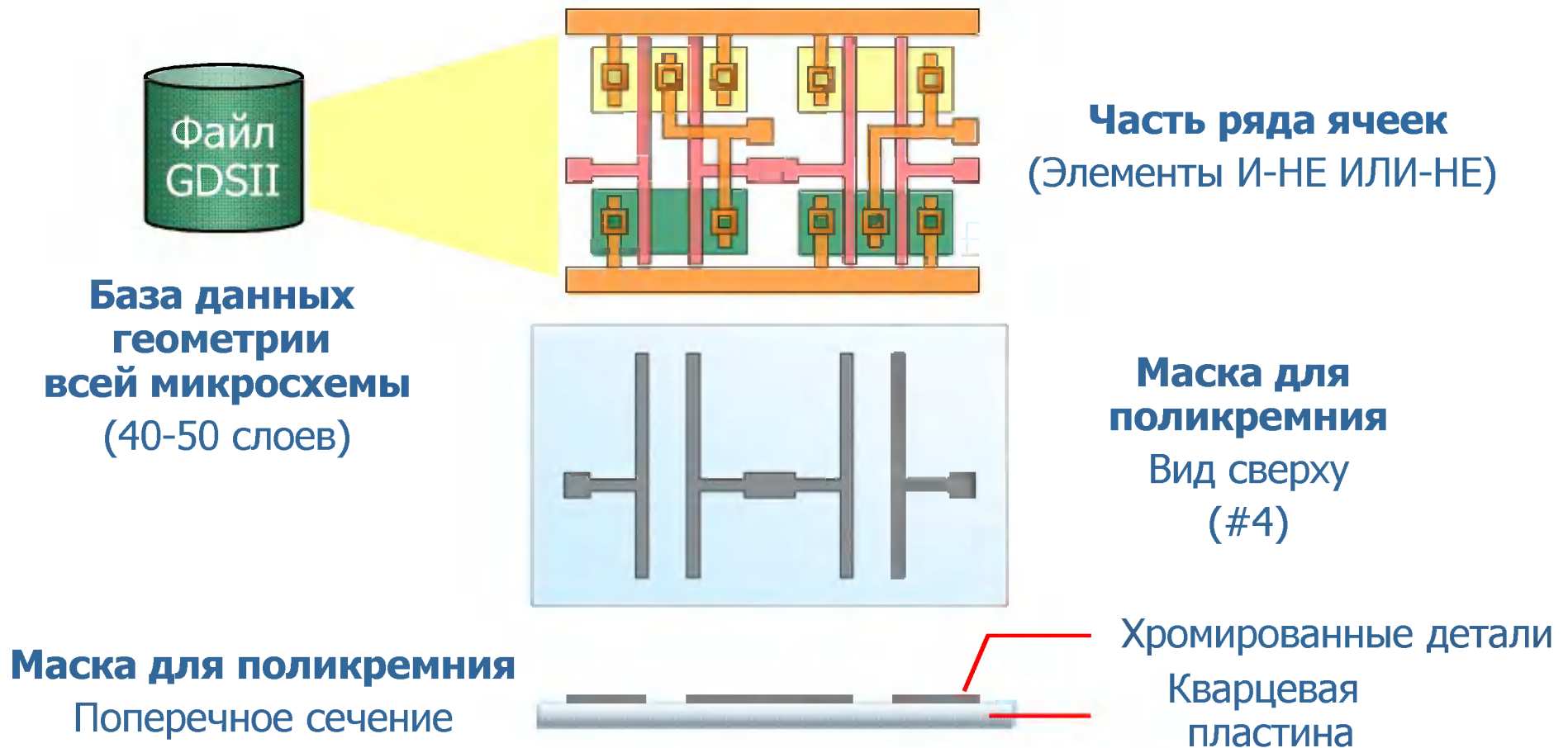


## Литография

Процесс переноса краски с печатной формы на плоский камень.  
От греческого *litho*, камень.

- Лазер светит только через точные **разрезы** в трафарете.
- На камне отпечатывается точная **копия** шаблон.
- Подобно трафарету, при нанесении изображения на пластине используется **фотомаска**.

# На предприятии по производству масок 5-6



База данных геометрии всей микросхемы (40-50 слоев)

Часть ряда ячеек (Элементы И-НЕ ИЛИ-НЕ)

Маска для поликремния Вид сверху (#4)

Маска для поликремния Поперечное сечение

Хромированные детали  
Кварцевая пластина

- При передаче на производство файл **GDSII** пересылается через ftp изготовителю масок.
- Геометрия **каждого слоя** переносится на чистую маску.

# От данных проекта к набору масок

# 5-7



- Двигающийся **электронный пучок** избирательно удаляет хром.
- Результат: точная увеличенная в 4 раза копия **шаблона** геометрии одного слоя.

# От масок на пластину

5-8

**Координатная метка**  
(Предназначена для выравнивания)



🔗 Жаргон:  
Часто фотооригиналы называются просто «масками»

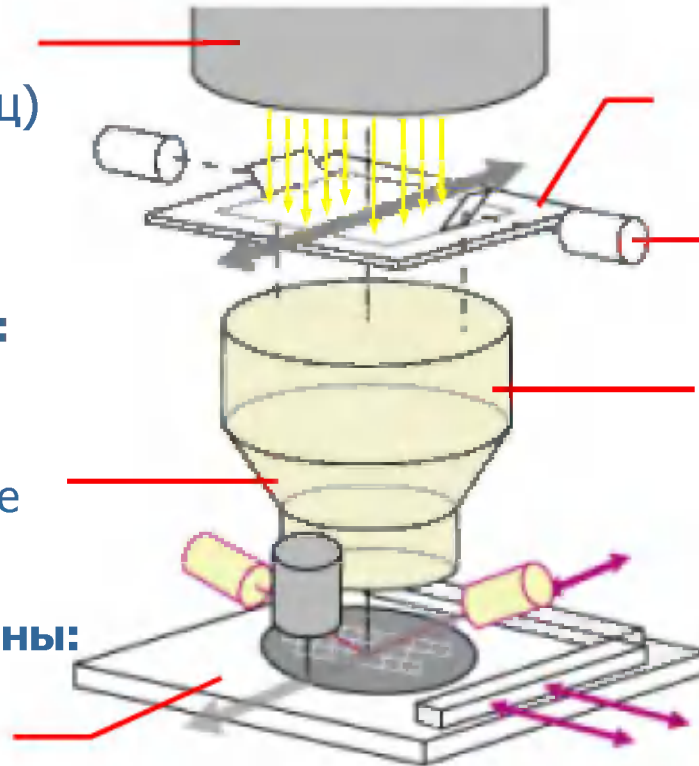
↪ **Ключевой принцип:**  
Каждый слой, выполняемый на кристалле, требует одну индивидуальную маску.

- Фотолитография **переносит** один шаблон на пластину.
- Раннее **маска** экспонировала все кристаллы на пластине за **один раз**.
- **Фотооригинал** переносит шаблон для нескольких кристаллов, затем **передвигается** на новое место.

# Устройство последовательного шагового экспонирования

# 5-9

**ArF лазер**  
(45 Вт на 100 Гц)



**Фотооригинал**

**Метка выравнивания:**  
Общая для фотооригиналов

**Конденсор:**  
Собирает ультрафиолетовый свет. Обеспечивает равномерное освещение маски.

**Интерферометр:**  
Положение *xuz* платформы

**Уменьшающая оптика:**

В отличие от увеличивающей — уменьшает изображение в 4 раза.

**Платформа для пластины:**

- ЧПУ *xuz* движение
- Воздушные подшипники

- В первых устройствах последовательного экспонирования фотооригинал был **неподвижным**.
- Пластина **пошагово передвигалась**, и весь фотооригинал экспонировался.
- **Сканирующее** устройство последовательного экспонирования передвигает и то и другое, выполняет пошаговое экспонирование.

# Устройство последовательного шагового экспонирования

# 5-10

## Высокая производительность:

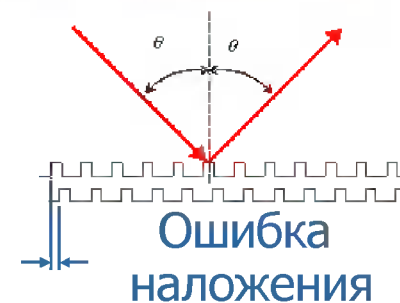
До 200 пластин/час.

Одна пластина экспонируется — следующая устанавливается.



Область показана на последнем слайде.

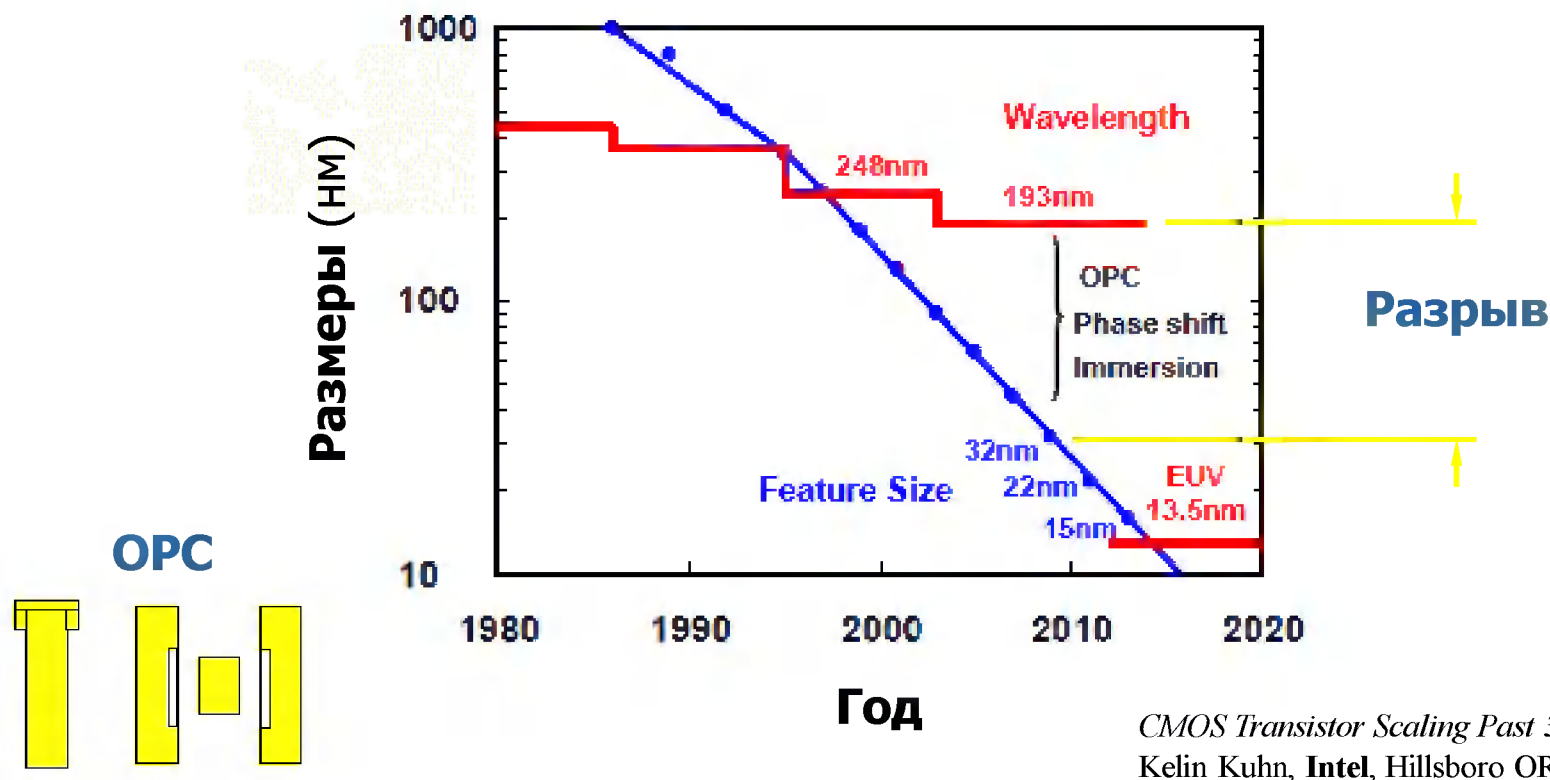
↳ Экономика производства: Самыми дорогими устройствами являются литографические установки. Они могут быть **узким местом** всего производства. Производительность критически важна.



- Такое устройство последовательного экспонирования обычно разрешает ширину линий **32 нм**.
- Такая точность не должна ограничивать **производительность** пластин.
- Производительность литографии может определять общую производительность **всего** предприятия.



# Разрыв в разрешающей способности 5-11



- Невероятно, но ультрафиолет **193 nm** позволяет разрешить детали **меньшего размера**.
- Необходима коррекция оптической близости (**OPC**) и другие коррекции.
- На **32 nm** требуется большая числовая апертура (**NA**).

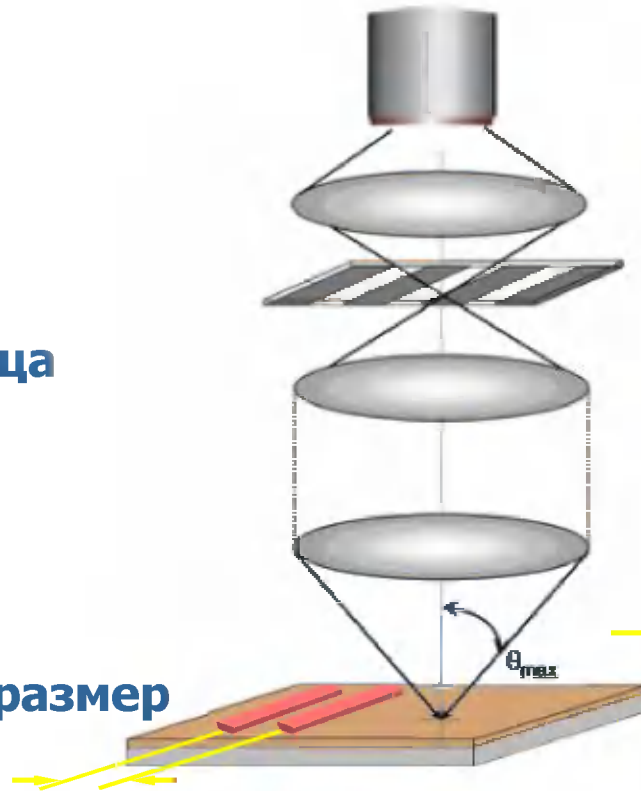
# Фундаментальный предел

5-12

$$CD = k_1 \frac{\lambda}{NA}$$

Рэлеевская граница  
(в нм)

Критический размер  
(CD)



↳ Что такое NA?

Числовая апертура -  
способность линзы  
собрать все порядки  
дифракции.

$$NA = n \sin \theta_{max}$$

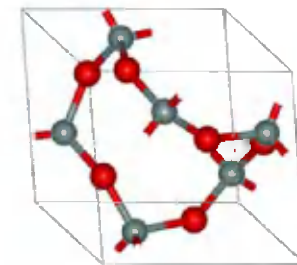
Показатель  
преломления  $n$   
(1.00 для воздуха)

- Способность устройства последовательного экспонирования **разрешать линии** имеет фундаментальный предел.
- Разрешение **меньших** критических размеров (CD) фундаментально требует **уменьшения**  $\lambda$ .
- Следующий качественный скачок возможно будет связан с использованием **глубокого (экстремального) ультрафиолетового излучения (EUV)**.

## **Основы технологии производства:**

- **Выращивание оксидных слоев**
- **Наложение фоторезиста**
- **Удаление ненужных участков оксида**
- **Имплантация легированных областей**

Молекулярная  
структура  
 $\text{SiO}_2$

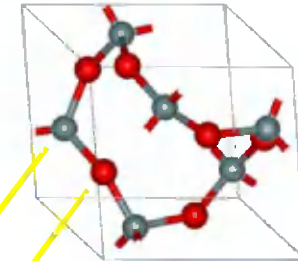


# Выращивание оксидных слоев 5-14

## Поперечное сечение

... формирует оксидный слой 5... внутренне объединенный с покрывающей поверхностью 2...

[Noyce Patent 2,981,877]

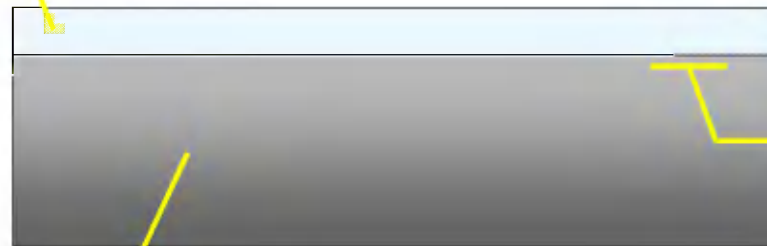


Атом Si

Атом O  
(Общий)

5. Оксидный слой  
(SiO<sub>2</sub>)

~1,000°C



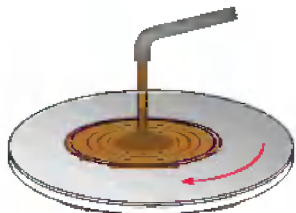
2. Высококачественная поверхность

Кремниевая подложка

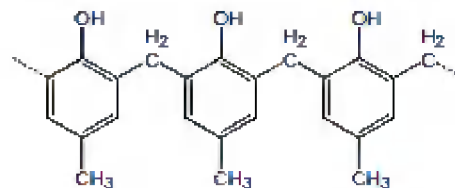
- Чистый кремний легко **окисляется** при температуре **~1,000°C** и создает SiO<sub>2</sub>.
- Атомы Si и O<sub>2</sub> соединяются и формируют **монолитный** слой.
- Подобно кварцу SiO<sub>2</sub> фактически является **совершенным** изолятором.

# Наложение фоторезиста

5-15



Нанесение фоторезиста  
(Центрифугирование)  
(~1,000 оборотов минуту)

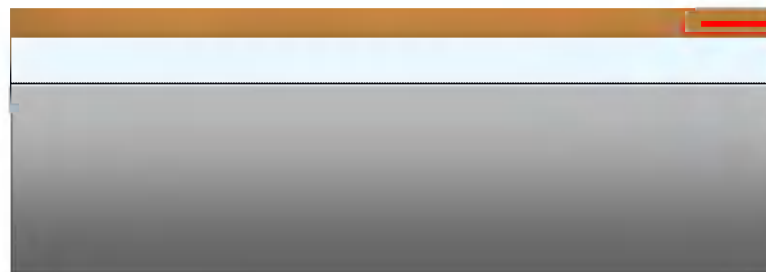


**Novolak (новолак, новолачная фенолформальдегидная смола):**  
Основной фоторезист микроэлектронной революции.

⚡ Не в масштабе:  
Фоторезист на самом деле намного **толще** оксида — около 1  $\mu\text{m}$ .

**Предварительный нагрев:**

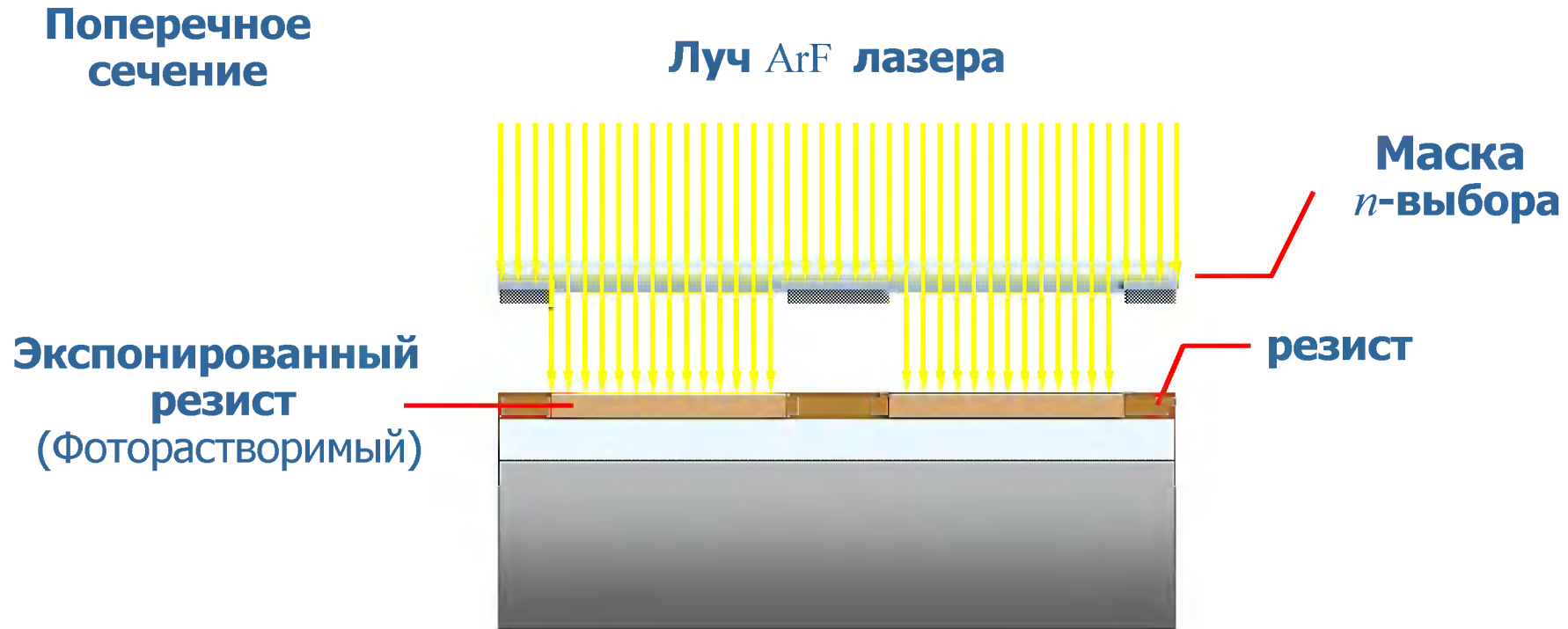
Удаляет растворитель, повышает плотность резиста.



Покрытие резистом

- Фоторезист равномерно наносится на всю платину путем **центрифугирования**.
- Ультрафиолетовые фотоны **разрушают** межмолекулярные связи между полимерами.
- **Позитивный** резист становится **растворимым** после экспонирования.

# Удаление ненужных участков оксида (1/3) 5-16



- На это шаге шаблон маски **переносится** на резист.
- После экспонирования фоторезист становится **растворимым**.
- Неэкспонированный резист остается нетронутым—точная **копия** геометрии маски.

---

# Удаление ненужных участков оксида (2/3) 5-17

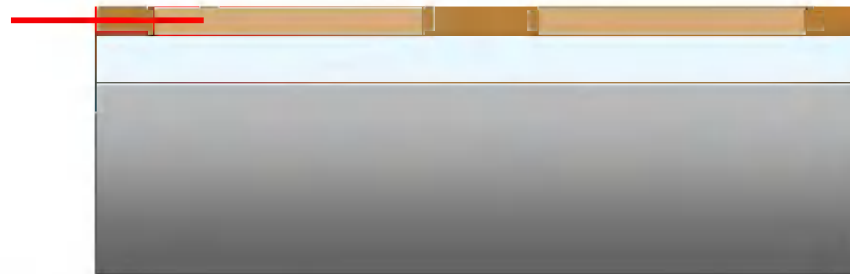


Проявление фоторезиста

Модульная установка  
(Центрифугирование/  
Нагрев/Проявления)



Экспонированный резист  
(удаляется проявителем)



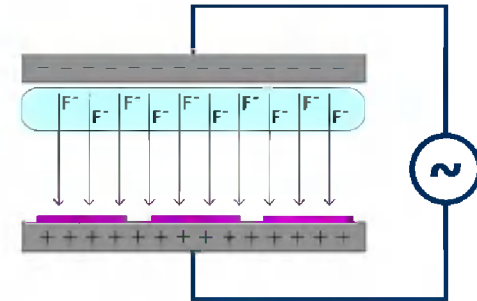
Финальный нагрев:  
Проявленный резист становится твердой эффективной маской.

- Экспонированный резист **вымывается** щелочным проявителем.
- Финальный нагрев **превращает** проявленный резист в твердую маску.
- Твердую маска может **защитить** участки от травления.

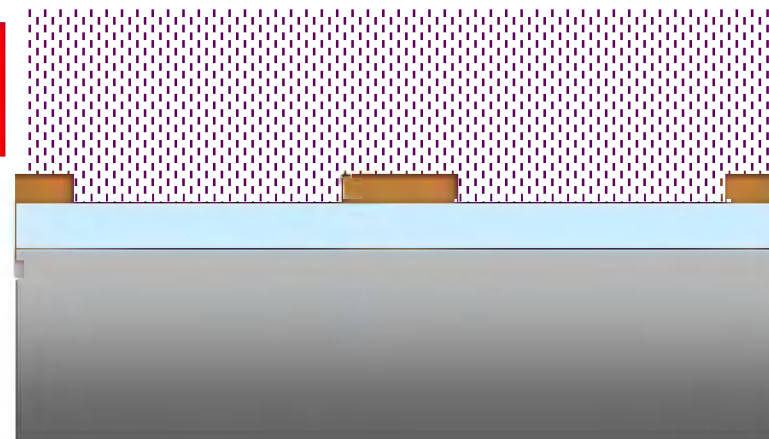
# Удаление ненужных участков оксида (3/3) 5-18

Поперечное сечение

Камера плазменного травления (14 МГц)



Плазменное травление  
(Фторуглеродная плазма,  $C_2F_6$ )



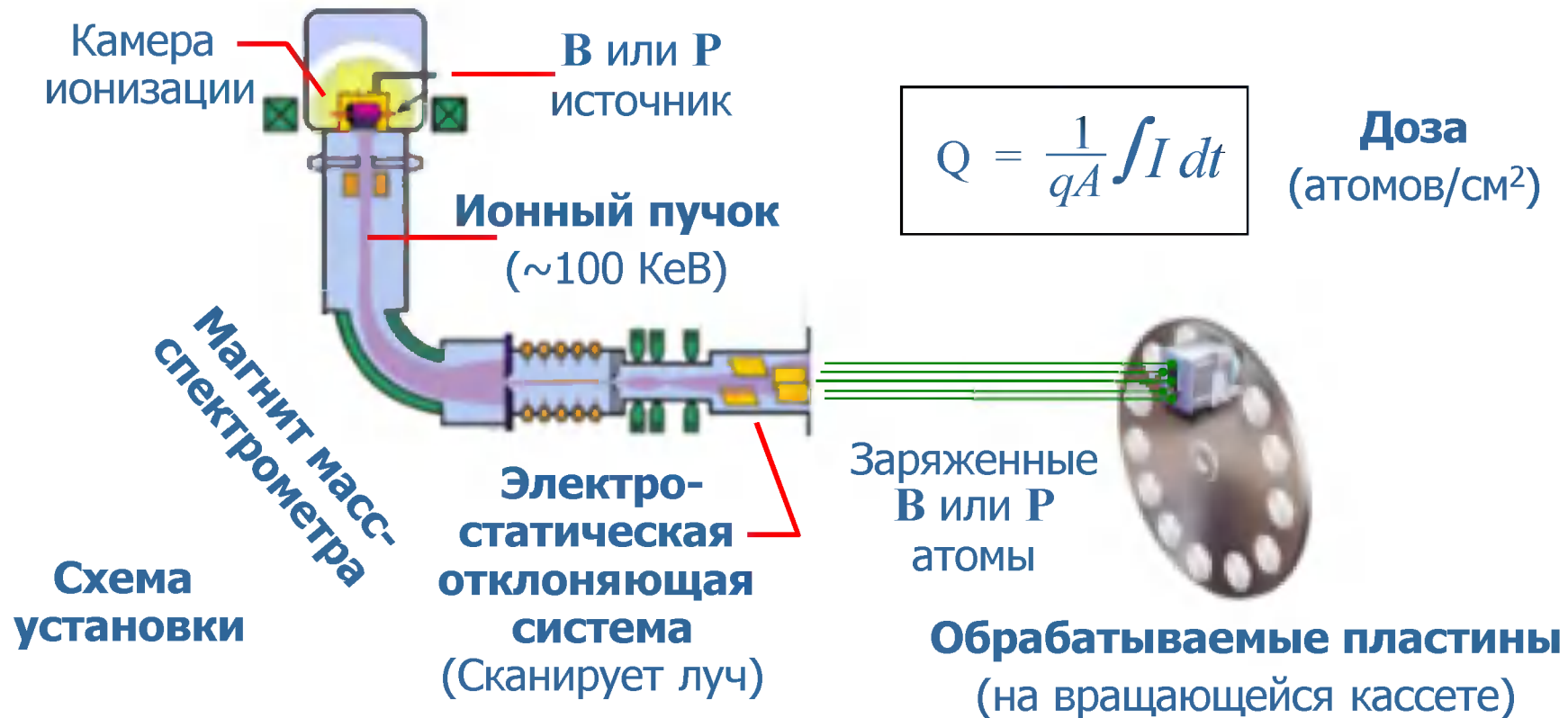
Резист удален

- Шаблон переносится с резиста на оксид с помощью **травления**.
- В прошлом **мокрое** травление использовало плавиковую кислоту, что приводило к **боковому травлению** стенок.
- Сухое (плазменное) травление обеспечивает **практически вертикальные** стенки.



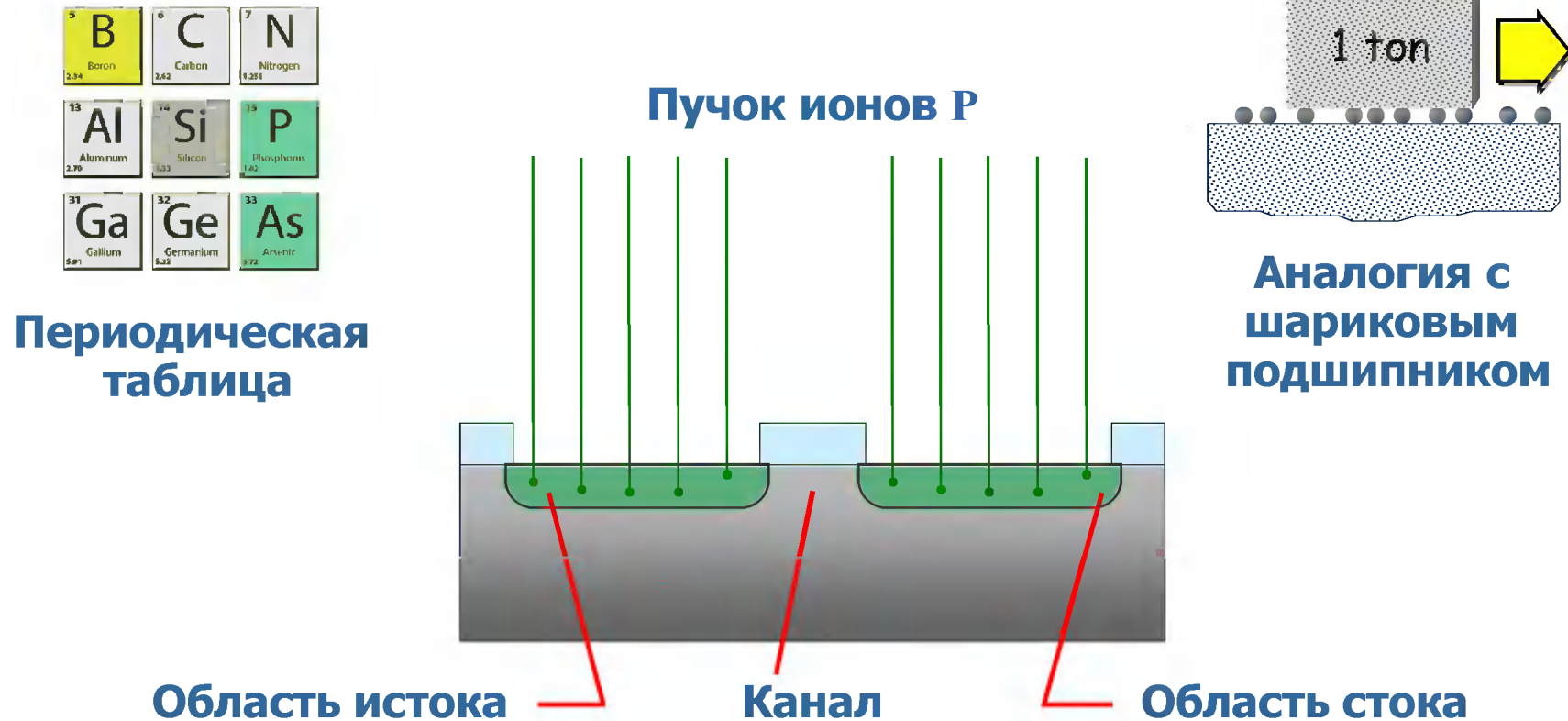
# Ионная имплантация

5-19



- Имплантация **вводит** легирующие атомы в приповерхностный слой.
- **Параметры** легирования—глубина и доза—тщательно контролируются.
- Отжиг позволяет атомам **занять свои места** в нарушенной кристаллической решетке.

# Имплантированные области 5-20

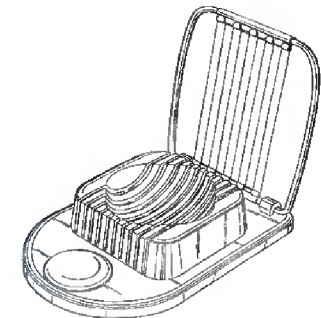


- Каждый атом **донора**, попавший в решетку, добавляет **электрон**.
- Подвижные электроны формируют проводящие области **n**-типа.
- Эти области разделены непроводящим **каналом**.

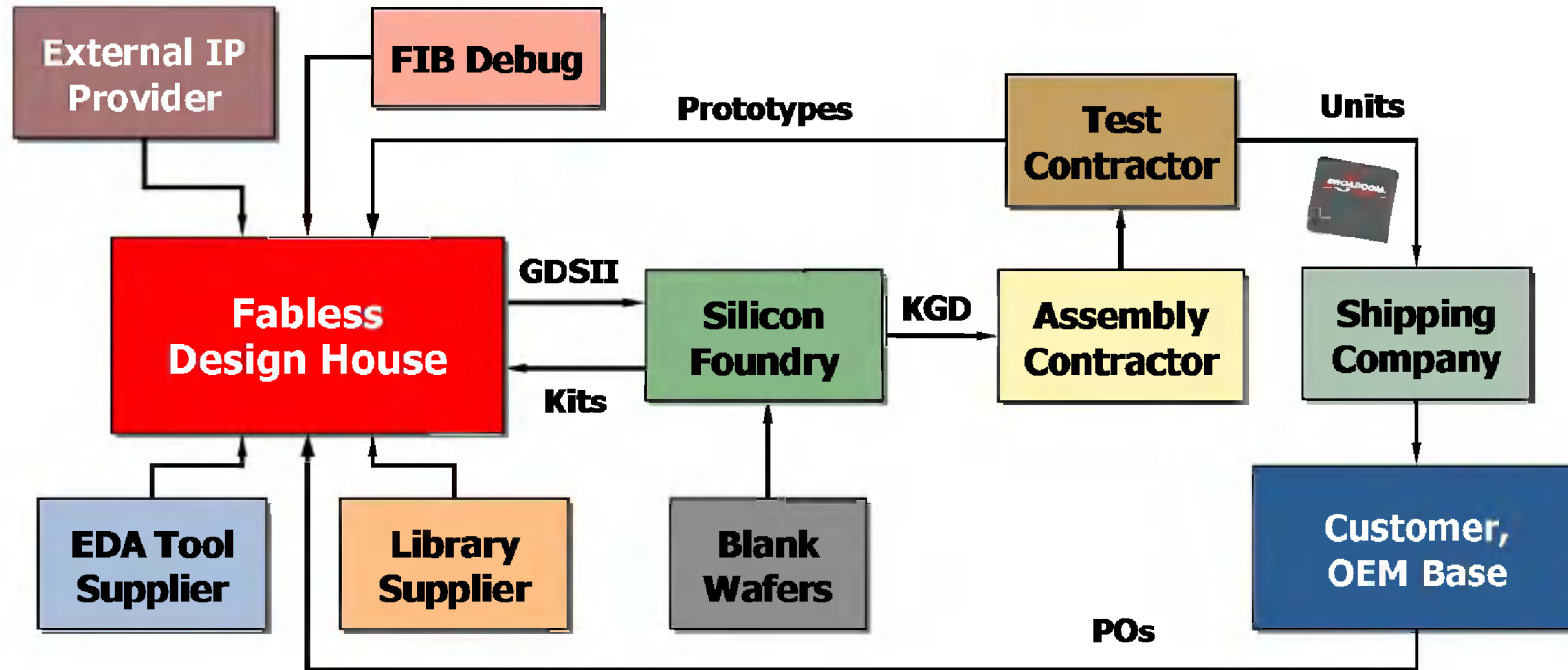
## Необработанные кремниевые пластины

- Экономика обработки пластин
- От спецификации к кремнию
- Искусство выращивания кристаллов
- Планарная КМОП технология
- На предприятии по производству микросхем

Кухонная машинка  
для нарезки яиц



# Экономика обработки пластин 5-22

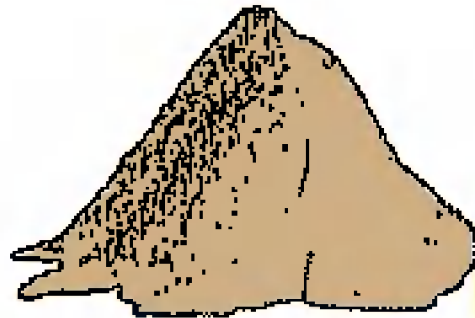


- Предприятия по производству микросхем, такие как TSMC, **покупают** необработанные пластины у поставщиков.
- Выращивание бездефектных кремниевых кристаллов само по себе является **искусством**.
- **Необработанная 300-мм пластина** стоит несколько сотен долларов.

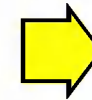
# От спецификации к кремнию

5-23

**Si для микроэлектроники:**  
Поликремниевые куски далее  
очищаются до 99.999999%  
чистоты.



**Песок:**  
Песчинки (Грязные  $\text{SiO}_2$ )  
кремнезема.  
Используется для получения 99%  
чистого поликристаллического Si.

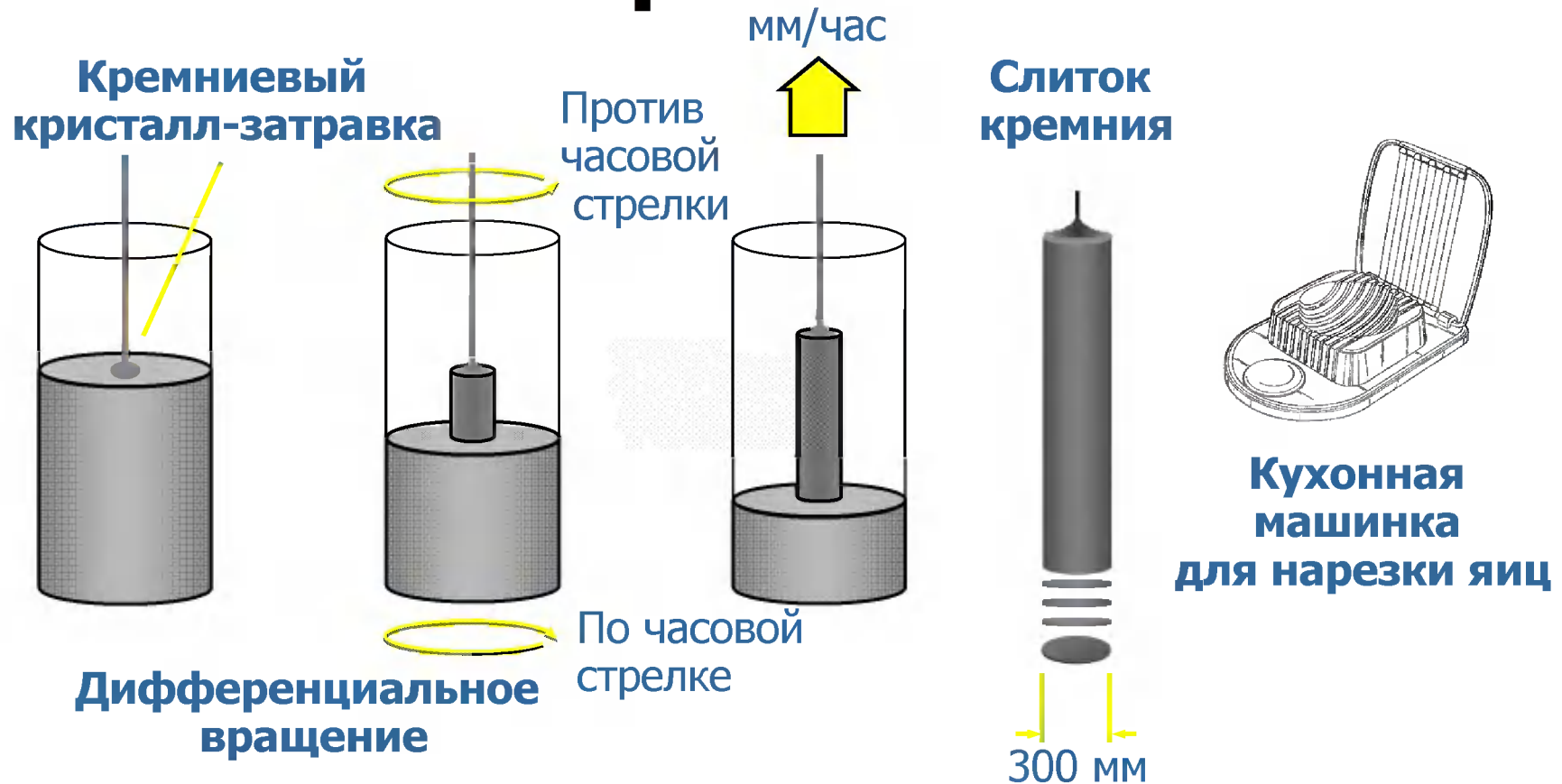


**Расплавленный Si:**  
Поликремниевые куски  
помещены в кварцевый  
тигель; нагреваются чуть  
выше температуры плавления.

- Кремний - один из самых **распространенных** на земле элементов.
- Но он должен быть **монокристаллическим** и **чистым** (несколько промилле, миллионная часть).
- Последовательно выполняются много шагов очистки и выращивания кристалла.

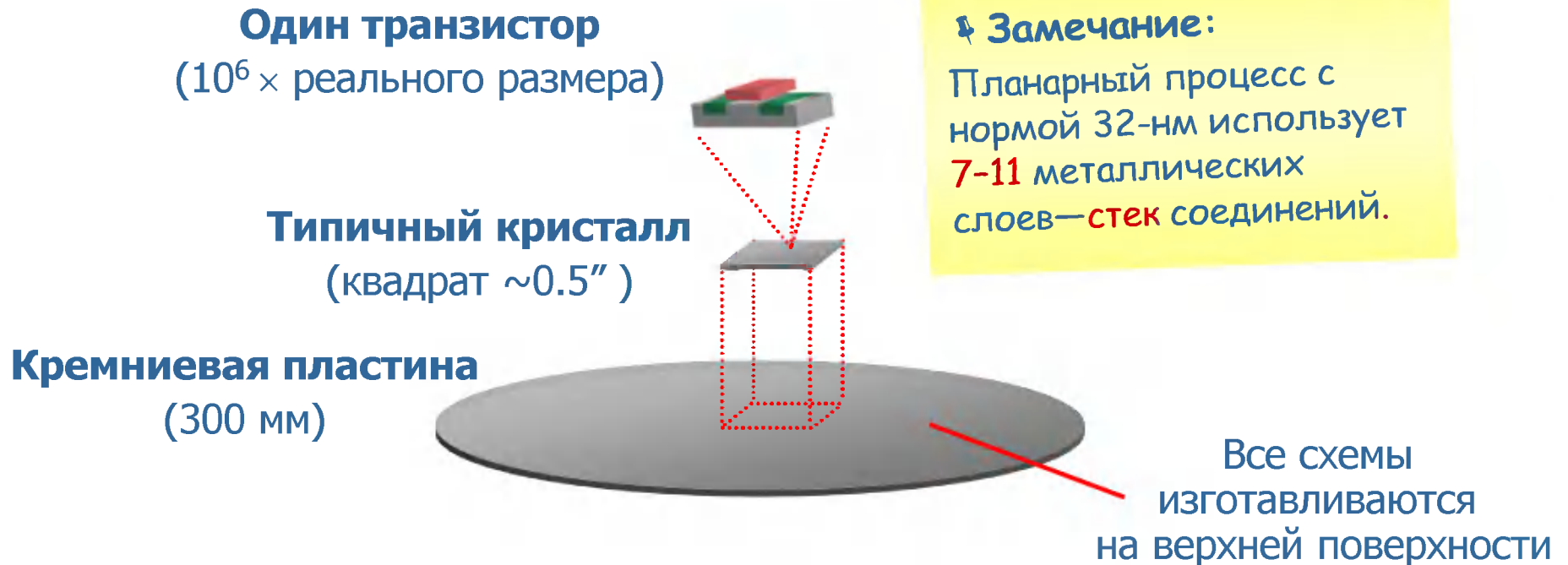
# Вытягивание кристалла

5-24



- Метод Чохральского позволяет вырастить **монокристаллический** кремний.
- Вращение обеспечивает **однородность**—получается **круглый** слиток .
- Слиток разрезается на тонкие **пластины** с помощью алмазной проволочной пилы.

# Планарная КМОП технология 5-25



- Большинство КМОП микросхем все еще изготавливаются по **планарной** технологии.
- Элементы формируются на **поверхности** кремния, без перекрытия.
- Разрезание слитка на пластины **доводит до максимума** площадь поверхности.

# На предприятии по производству микросхем 5-26

Участок  
производственной  
линии



Необработанные  
пластины  
на входе



Бар-код  
или OCR ID

- На предприятии поддерживаются **сверхчистые** условия.
- Каждая необработанная пластина **помечается** уникальным идентификатором.
- Партии пластин поочередно обрабатываются установками всей производственной **линии**.



## **Первые этапы технологического процесса (Front-End of Line, FEOL):**

- **Изоляция мелкими канавками**
- **Самосовмещенные поликристаллические затворы**
- **Имплантиция истока/стока**
- **Нанометровые улучшения**

**Кассета  
на 28 пластин**

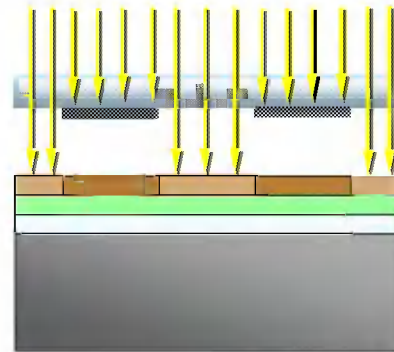
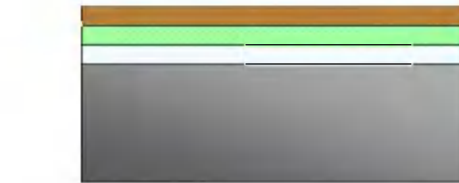


# Изоляция канавками (1/2)

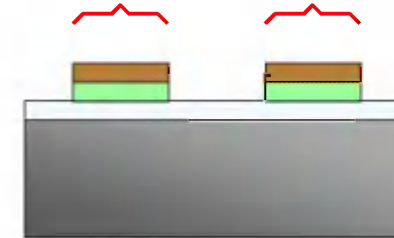
5-28



✦ **Высокая плотность:**  
Для реализация плотной упаковки старые техники заменяются изоляцией мелкими канавками (STI).



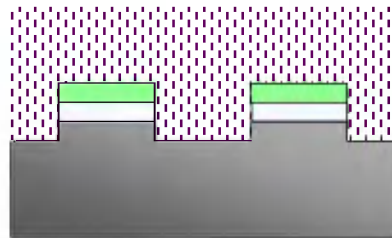
Активные области



- Нитрид кремния **блокирует** дальнейшее окисление поверхности.
- Устанавливает границы **активных областей**, в которых будут созданы транзисторы.

# Изоляция канавками (2/2)

5-29



⑤ Канавки, вытравленные в подложку



⑥ Выращивание оксидного поля.



⑦ Химико-механическая (CMP) полировка

## Предварительный обзор:

Канавки будут изолировать соседние *n*, *p* транзисторы.



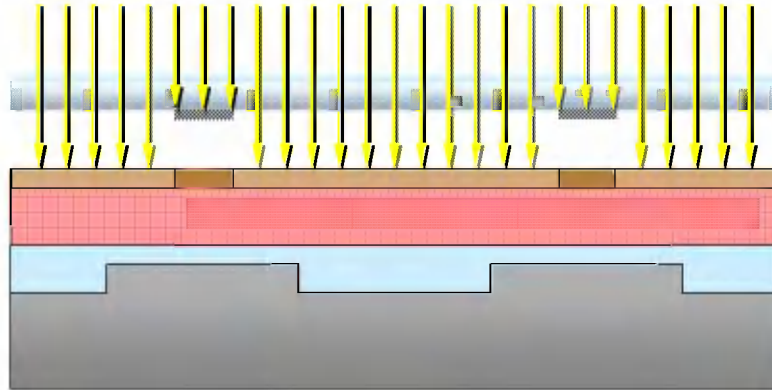
‡ Дамасская технология: Древняя технология заполнения канавок с избытком и последующее полирование для получения плоской поверхности.

- Мелкие ( $1/2$  мкм) канавки **изолируют** соседние транзисторы.
- Устраняют паразитное **взаимодействие** между соседями.

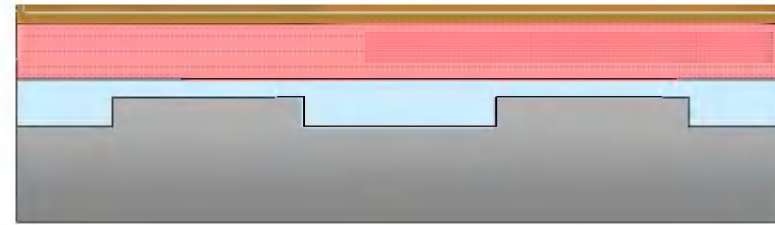
# Формирование поликремниевых 5-30 затворов



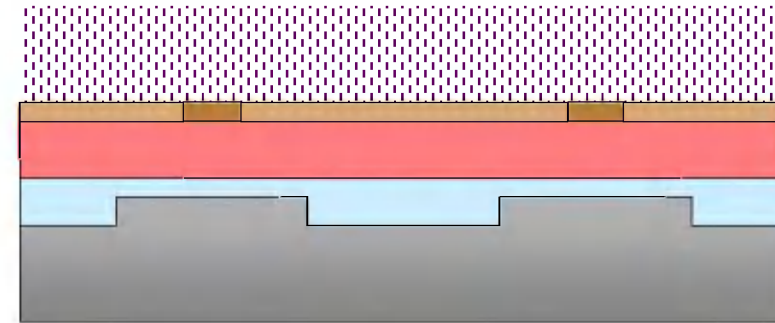
① Выращивание подзатворного оксида с высоким  $k$



③ Шаг: маска выбора ПОЛИКРЕМНИЯ (#4)



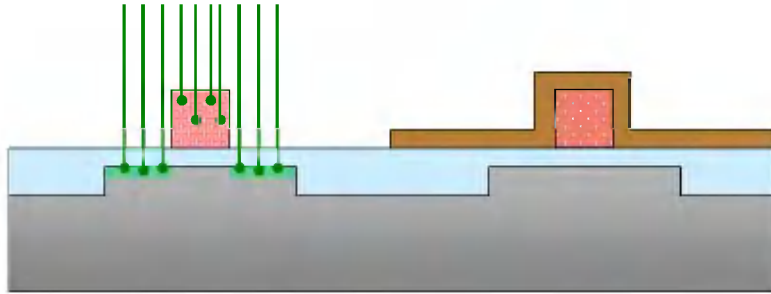
② Осаждение поликремния; центрифугирование фоторезиста



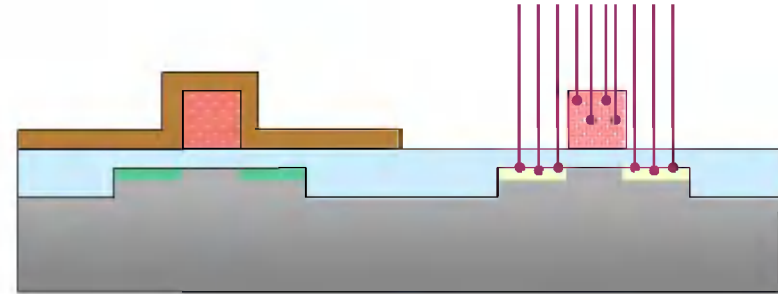
④ Вытравливание поликремниевых затворов

- На этом этапе формируется **МОП** сандвич для транзисторов.
- Поверх предельно тонкого оксида осаждается **поликремний**, затем он травится.

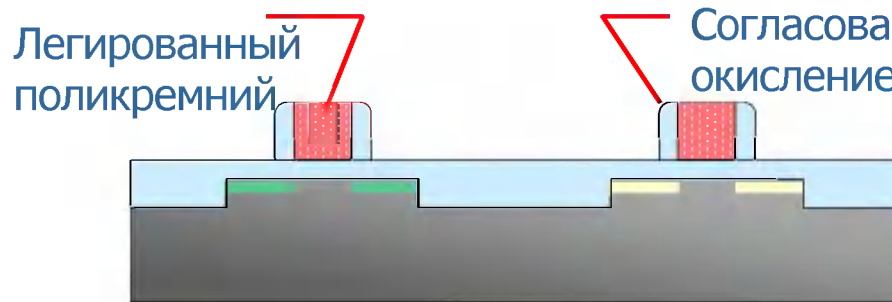
# Имплантация истока и стока 5-31



- ① Шаг: маска выбора  $n$  (#5);  
Формирование LDD (слабо-легированного стока):  $n$  имплантация



- ② Шаг: маска выбора  $p$  (#6);  
Формирование LDD (слабо-легированного стока)  $p$  имплантация



- ③ Выращивание согласованного оксида для защиты области слабо-легированного стока (LDD)

- Легирование **блокировано**, когда поликремний маскирует имплантируемую примесь.
- Поликремниевый затвор **сам совмещается** с областями истока/стока.



- ④ Шаг повторного выбора маски (#5, 6).  
Имплантация истоков и стоков

# Нанометровые улучшения

5-32

④ **Силицидная заглушка:**  
Повышает проводимость поликремния

⑥ **Металлический затвор:**  
Титановый сплав заменил поликремний — совместим с оксидом  $\text{HfO}_2$ .

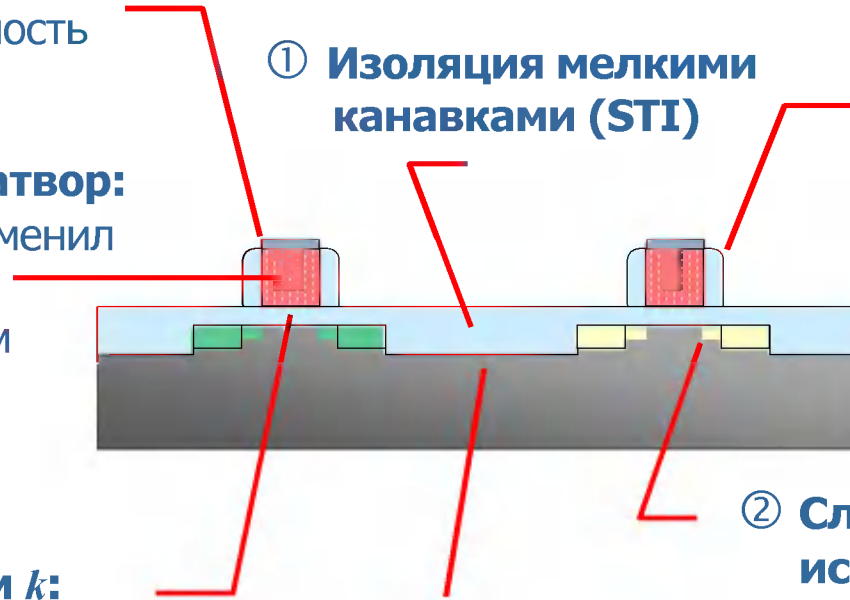
⑤ **Оксид с высоким  $k$ :**  
 $\text{HfO}_2$  заменил  $\text{SiO}_2$ .  
 $k \approx 15$  в сравнении с 4

⑦ **Напряженный кремний:**  
Подвижность увеличена за счет механического напряжения

① **Изоляция мелкими канавками (STI)**

③ **Окисление разделительной области:**  
Маскирует область слаболегированного стока (LDD)

② **Слаболегированный исток/сток:**  
Последовательное сопротивление уменьшает побочные эффекты горячих носителей

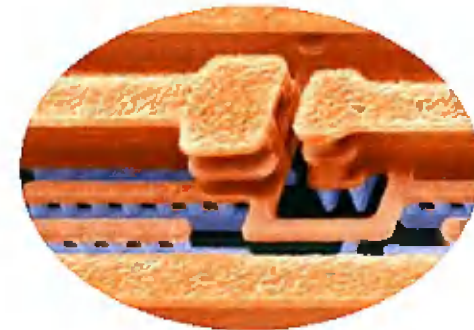


- Каждое уменьшение размера требует дальнейшей **модернизации** процесса.
- Предел: На  $12 \text{ \AA}$ , подзатворный оксид имеет толщину только **5 молекул**.
- Поэтому 32-нм КМОП технология использует **металлический затвор с диэлектриком с высоким  $k$** .

## **Заключительные этапы технологического процесса (Back-End of Line, BEOL):**

- **Нанесение диэлектрика перед металлизацией (PMD)**
- **Подсоединение к транзисторам**
- **Дамасская технология: медь**
- **Создание контактных столбиков пластин, сортировка**

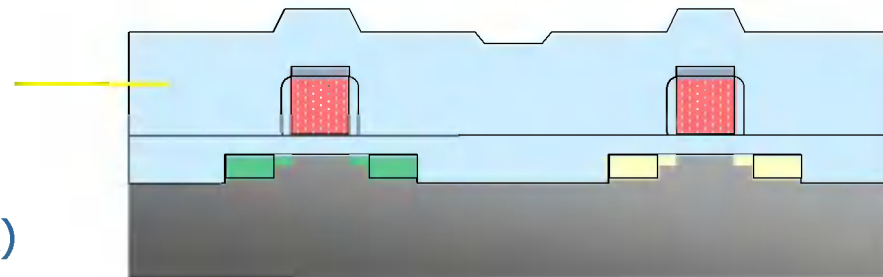
**Медные  
соединения**



# Нанесение диэлектрика перед металлизацией 5-34

(PMD)

Изолирующий  
слой оксида  
(Силикат с малым  $k$ )



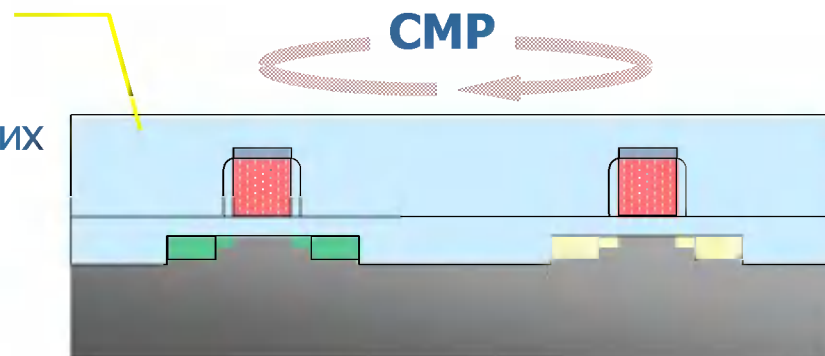
Нагревающие  
катушки

Низкотемпературное  
химическое осаждение  
из паровой фазы (CVD)

A schematic diagram of a CVD reactor. It shows a horizontal tube with a blue filament inside. Above the tube are two red rectangular heating coils. A yellow arrow points from the text 'Нагревающие катушки' to the coils.

Выравнивание  
поверхности:

Лучше для последующих  
слоев, несмотря  
на утончение  
изолирующего слоя.



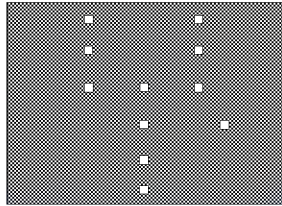
Химико-механическая  
полировка

A schematic diagram of a chemical-mechanical polishing process. It shows a substrate being polished by a rotating head. A yellow arrow points from the text 'Химико-механическая полировка' to the substrate.

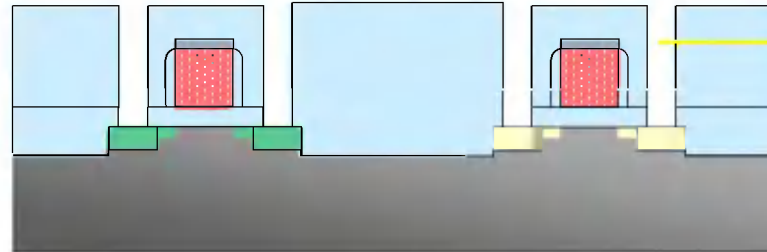
- На **завершающих** этапах технологического процесса транзисторы **соединяются** металлом.
- Вначале наносится и выравнивается изолирующий оксид.
- В отличие от подзатворного оксида, этот слой выполняется из диэлектрика с малым  $k$ .



# Подсоединение к транзисторам 5-35



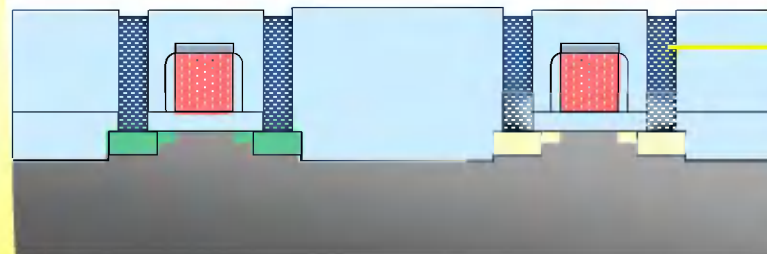
**Маскирование контактов**



## Проблемы заполнения:

Отверстие слишком узкое и высокое для заполнения обычными металлами.

✦ Бороздки/отверстия  
Сопротивление бороздок и отверстий достигает 2-20  $\Omega$ .  
Использование дублирующих отверстий уменьшает общее  $R_{net}$ .

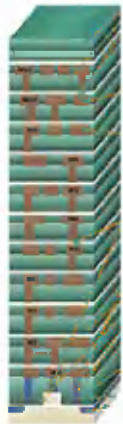


## Вольфрамовый столбик:

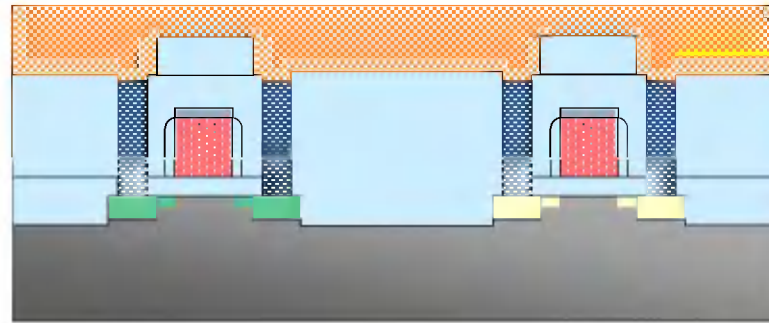
С помощью химического осаждения из паровой фазы отверстие заполняется вольфрамом (W), затем используется Дамасская технология.

- Маленькие бороздки позволяют сформировать металлические контакты к областям **истока/стока**.
- Контакты к поликремнию (тут не показаны) формируются вне активной области.
- Полное **заполнение** бороздок при технологической норме 32-нм является сложной проблемой.

# Дамасская технология: медь 5-36



**3-D вид:**  
11 слоев **Cu**  
межсоединения  
для 65 нм процесса.



**Гальваническое покрытие:**

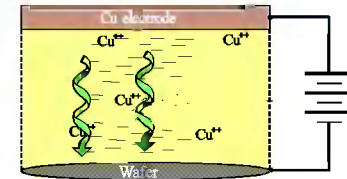
Медь с избытком покрывает весь маскирующий шаблон.

**Химико-механическая полировка (CMP)**



**Выравнивание поверхности:**

Остаются только линии **Cu**, которые соединяют оба стока инвертора и т.д.

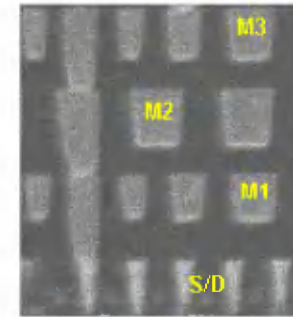


- Ранее для металлических линий использоваться **алюминий** (Al).
- Медь (Cu) - отличный проводник, но с ней сложно работать.
- Сложно травить—должно быть выполнено **гальваническое покрытие** Cu.

# Стек соединений

5-37

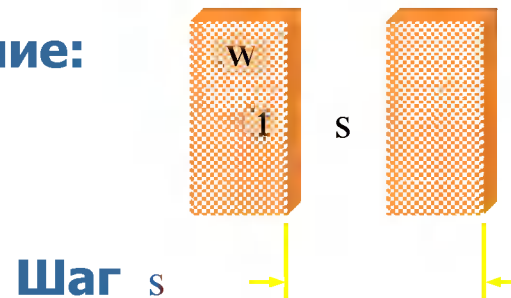
Metal Layers at 40 nm			
Layer	Pitch (nm)	Thick (nm)	Aspect Ratio
Metal 9 (RDL)	30,000	7,000	0.4
Metal 7-8	~700	~600	1.8
Metal 4-6	~300	~250	1.8
Metal 1-3	160	144	1.8



Стек соединений  
(часть)

## Форматное соотношение:

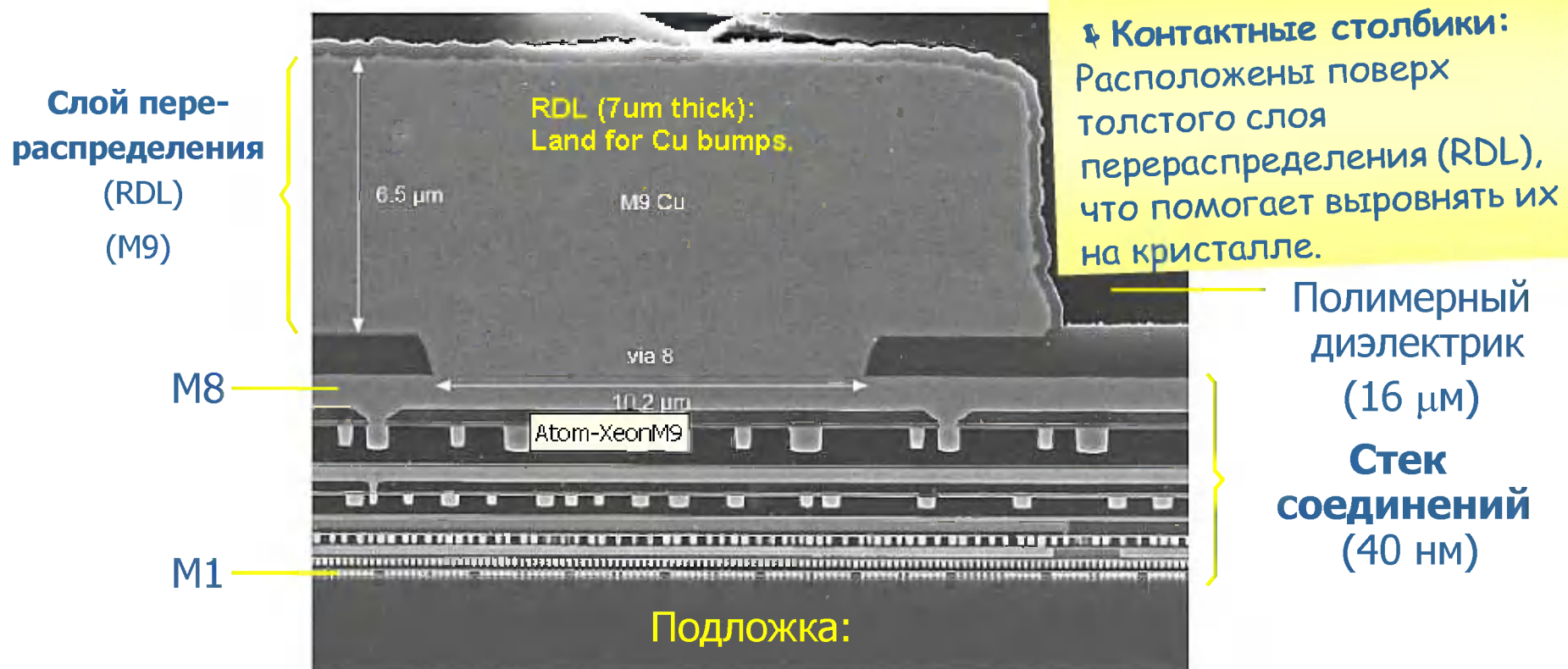
Толщина/ширина в поперечном сечении.



↳ **Высокий, тонкий:**  
В микронной технологии отношение  $< 1$ . На 32 нм форматное соотношение  $\approx 2$ .

- Нижние слои металла имеют **малый шаг**, но они медленные.
- Более высокие уровни более **толстые**, с меньшей задержкой по всему кристаллу.
- Стек соединений: **иерархия** уровней трассировки.

# Слой перераспределения (RDL) 5-38



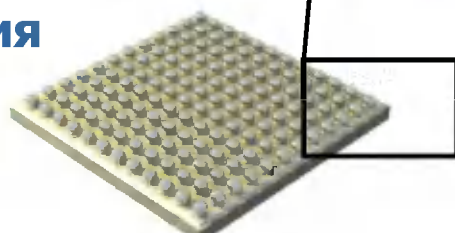
- Слой перераспределения **подводит** внешние сигналы микросхемы к контактному столбику.
- Макроскопические **контактные столбики** расположены поверх **площадок** микронной толщины.
- Помогают **выравнивать** рассеивание тепла и подвод питания.

# Создание контактных столбиков на пластине 5-39

## Преимущества:

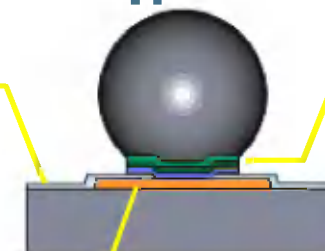
- Нет связывающих проволочек.
- Малая паразитная  $L$ .
- Очень большое количество входов/выходов.
- Распределение питания, земли по ядру.

## После создания контактных столбиков



Пассивация

УВМ (металлизация под столбиком)



Площадка

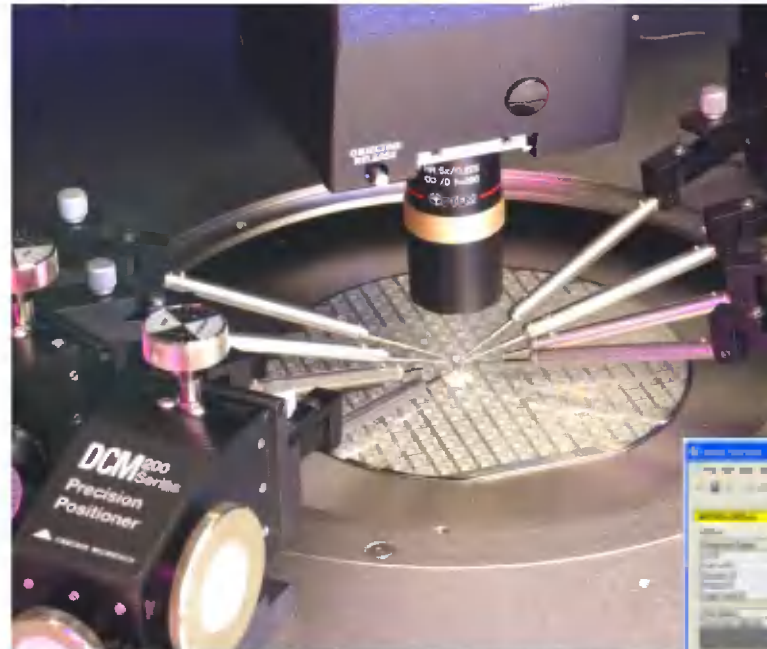
Контактный столбик из припоя

- При корпусировании методом **перевернутого кристалла**, ячейки ввода/вывода соединятся с контактными столбиками.
- Столбики **располагаются** как точки из припоя, затем он плавится.
- При сортировке микросхемы тестируются на пластине, **подсоединение осуществляется через контактные столбики.**

# Тестирование пластин при сортировке 5-40

## Виды тестов:

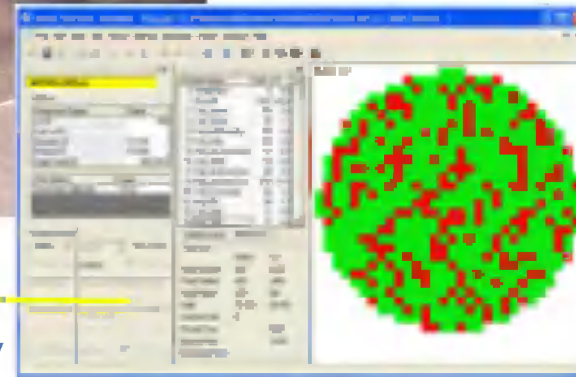
- Параметрические
- Для микросхем: ATPG (автоматическая генерация тестовых шаблонов), BIST (встроенный самоконтроль)
- Встроенное ОЗУ
- Iddq (ток утечки)
- Частоты круговой генерации



Нет красным точкам!  
Плохие кристаллы больше не **отмечаются красной**. Вся информация записывается в базу данных. Партия теперь сортируется за час.

## Карта отсортированных пластин

(Автоматизированная системам контроля на основе STIL, стандартного языка интерфейсного тестирования)



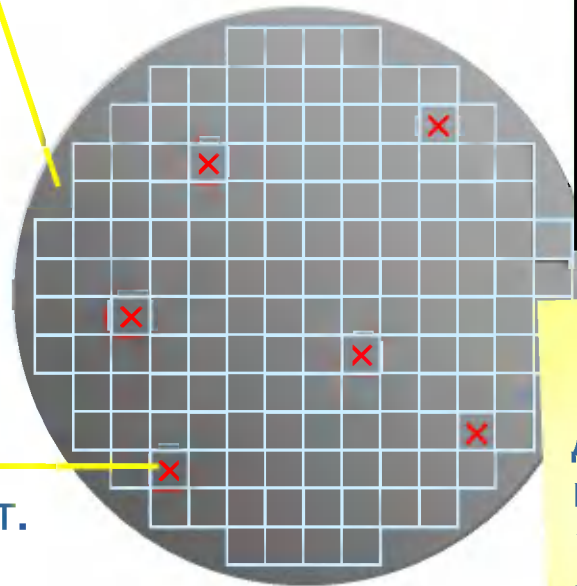
- Не **все** кристаллы работоспособны - некоторые имеют **дефекты** изготовления.
- При **сортировке** микросхемы тестируются на пока они все еще находятся на пластине, подсоединение осуществляется через контактные столбики.
- Только **бездефектные кристаллы (known-good die, KGD)** отправляются на корпусирование.

# Отслеживание выхода сортировки

# 5-41

## Пример выхода годных изделий:

Годных кристаллов: 142  
Всего кристаллов: 148  
Выход сортировки: 96%



**Плохой кристалл:**  
Не прошел хоть один тест.

$$Y_{sort} = 142 \div 148 = 0.96$$

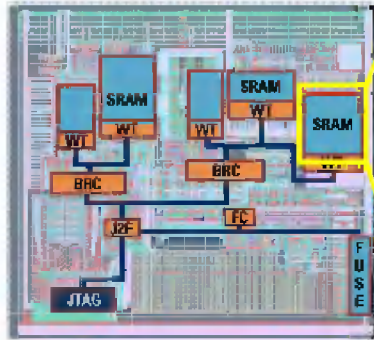
→ **Ключевой принцип:**  
При цене завершённой пластины в \$4к, кристалл стоит \$28—в основном определяется  $Y$ .

↳ **Практическое правило:**  
Для увеличения выхода годных изделий, делать размер кристалла минимальным для уменьшения вероятности отказа, которая растёт с площадью.

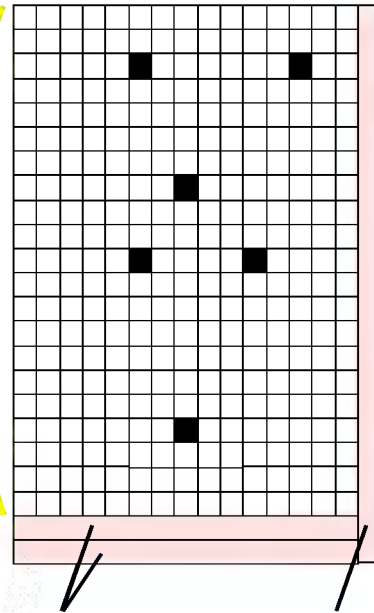
- Отношение количества бездефектных кристаллов к их общему количеству называется **выходом сортировки**:  $Y_{sort}$
- **Повышение** выхода уменьшает плотность дефектов со временем.
- Таким образом  $Y_{sort}$  улучшилась после отладки нового технологического процесса 32-нм.

# Ремонт встроенного ОЗУ

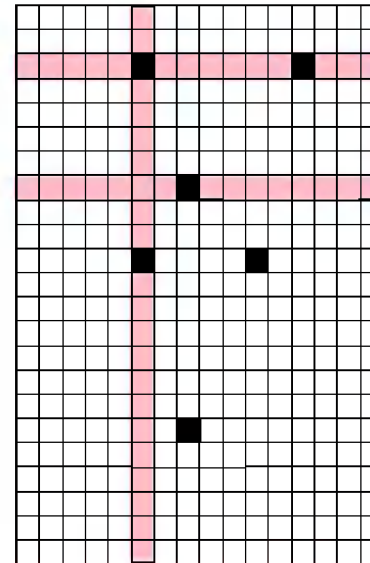
5-42



Блоки  
встроенного ОЗУ

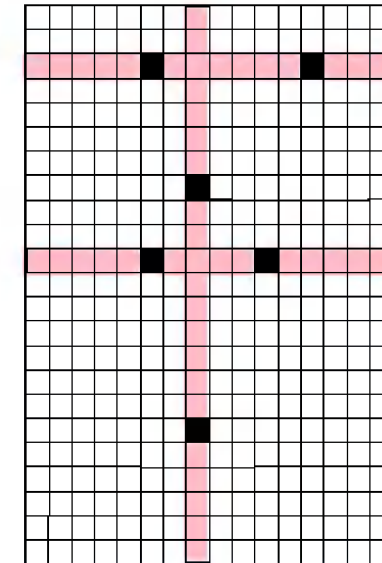


Запасные строки  
Запасные столбики



**Недостаточная замена**

Статическое ОЗУ все еще содержит 2 критических дефекта.



**Оптимальная замена:**

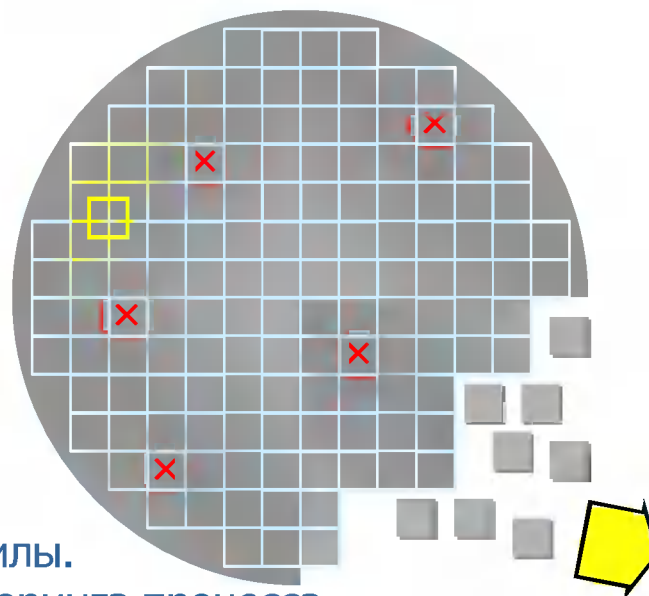
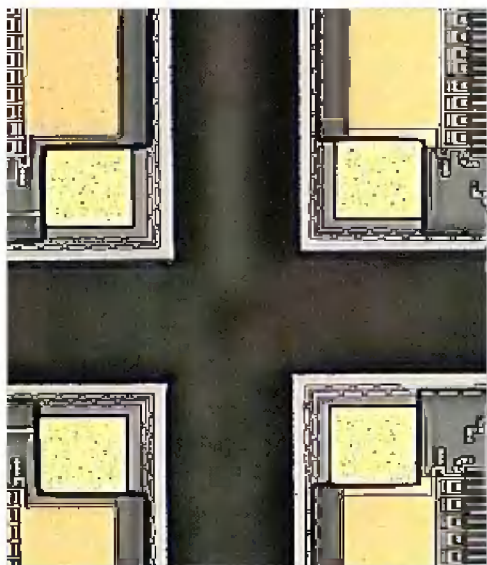
Все дефектные строки и столбцы заменены.

- Часто дефекты внутрикристаллического ОЗУ/ПЗУ уменьшают выход годных изделий.
- Замена плохих ячеек **запасными** может поднять  $Y_{sort}$ .
- Перераспределение адресов и замена происходит при **сортировке**.



# Разрезание (скрайбирование) пластины

# 5-43



**Пила для пластин**  
(алмазная)

## Загрузка - разгрузка:

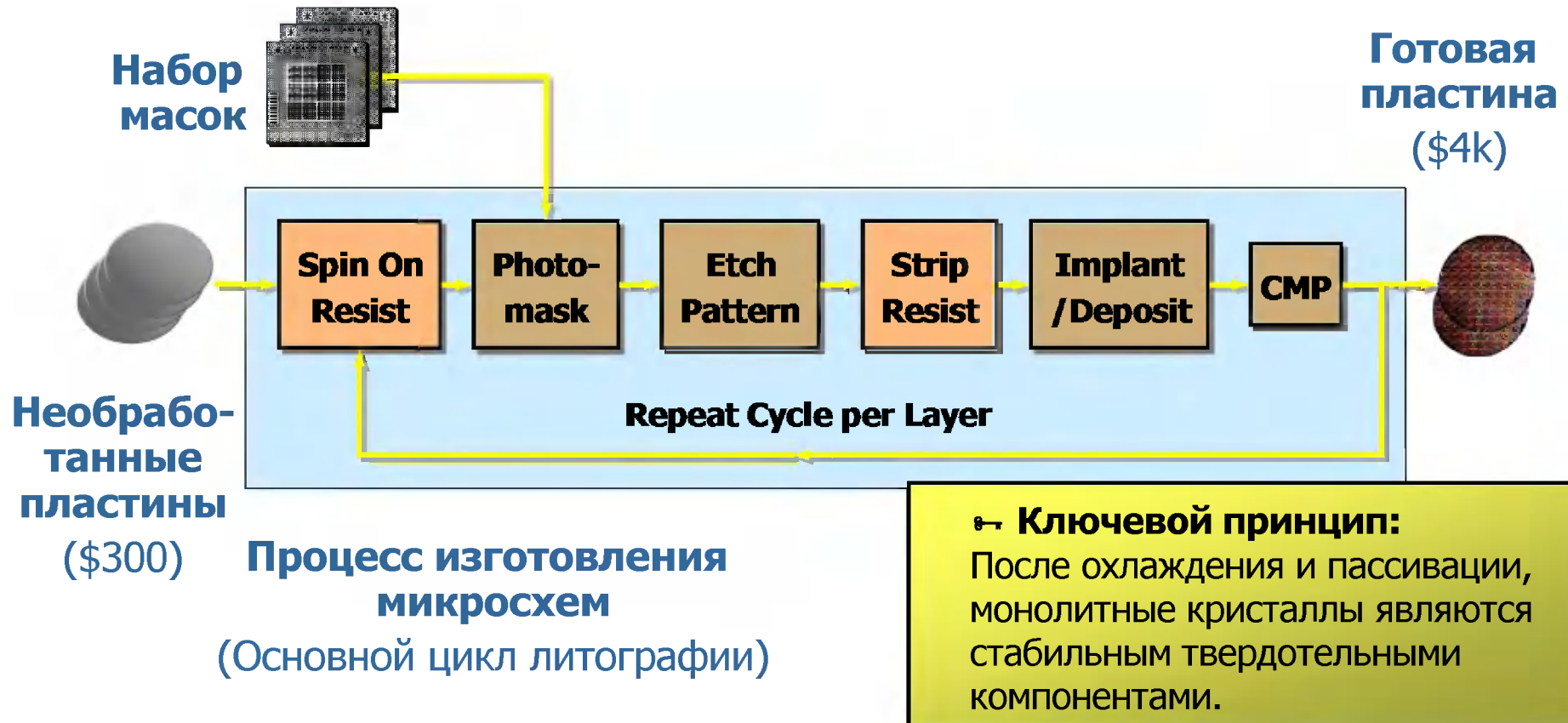
Хорошие кристаллы помещаются в контейнер; отправляются на сборочное предприятие.

## Линии нарезки:

*Дорожки* направляют лезвие пилы.  
Содержат структуры для мониторинга процесса.

- Разделение **пластины** на отдельные **кристаллы** называется **скрайбированием**.
- Резка выполняется сверхтонкой алмазной **пилой**.
- Для **корпусирования** отправляются только **хорошие** кристаллы, их размещают в контейнере, защищающем от статического электричества.

# От необработанной пластины к готовой 5-44

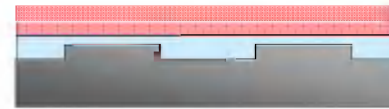


- Основной цикл **повторяется**—с изменениями—для каждой маски.
- Типичный 32-нм КМОП процесс включает **40–50** масок.
- Каждое **изменение процесса** неизбежно затрагивает другие шаги.

# Ключевые шаги процесса: (1/2) 5-45



1. Оксидная прослойка



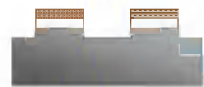
6. Подзатворный оксид



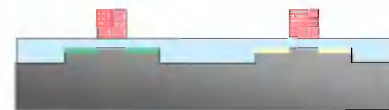
2. Активные области  
(Маска #1)



7. Поликремниевые  
затворы (Маска #4)



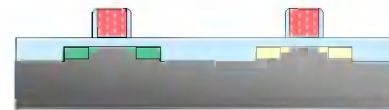
3. Вытравливание  
канавок



8. Имплантация слабо-  
легированного стока  
(Маски #5, 6)



4. Выращивание  
оксидного поля



9. Имплантация стока/  
истока (Маски #5,6)



5. Изоляция мелкими  
канавками (STI)

Первые этапы технологического процесса (FEOL) формирования инвертора

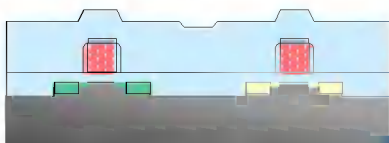
# Ключевые шаги процесса: (2/2) 5-46



10. Силицид



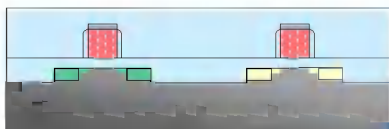
14. Заполнение  
контактов



11. Нанесение  
диэлектрика  
перед  
металлизацией



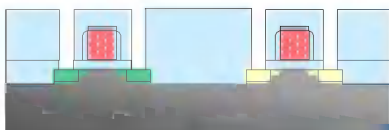
15. Гальваническое  
нанесение меди



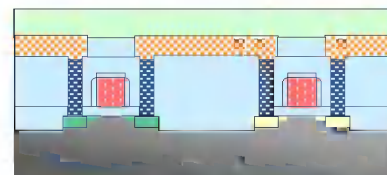
12. Химико-  
механическая  
полировка (CMP)



16. Метал 1  
(Маска #8)



13. Травление  
отверстий  
(Маска #7)



17. Осаждение  
диэлектрика,  
изолирующего  
металлические  
слои (IMD)

## Завершающие этапы

## технологического процесса (BEOL) формирования инвертора

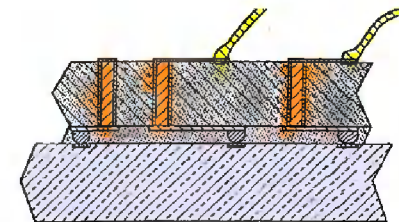
---

# Developing the Nanometer ASIC

## From Specs to Silicon

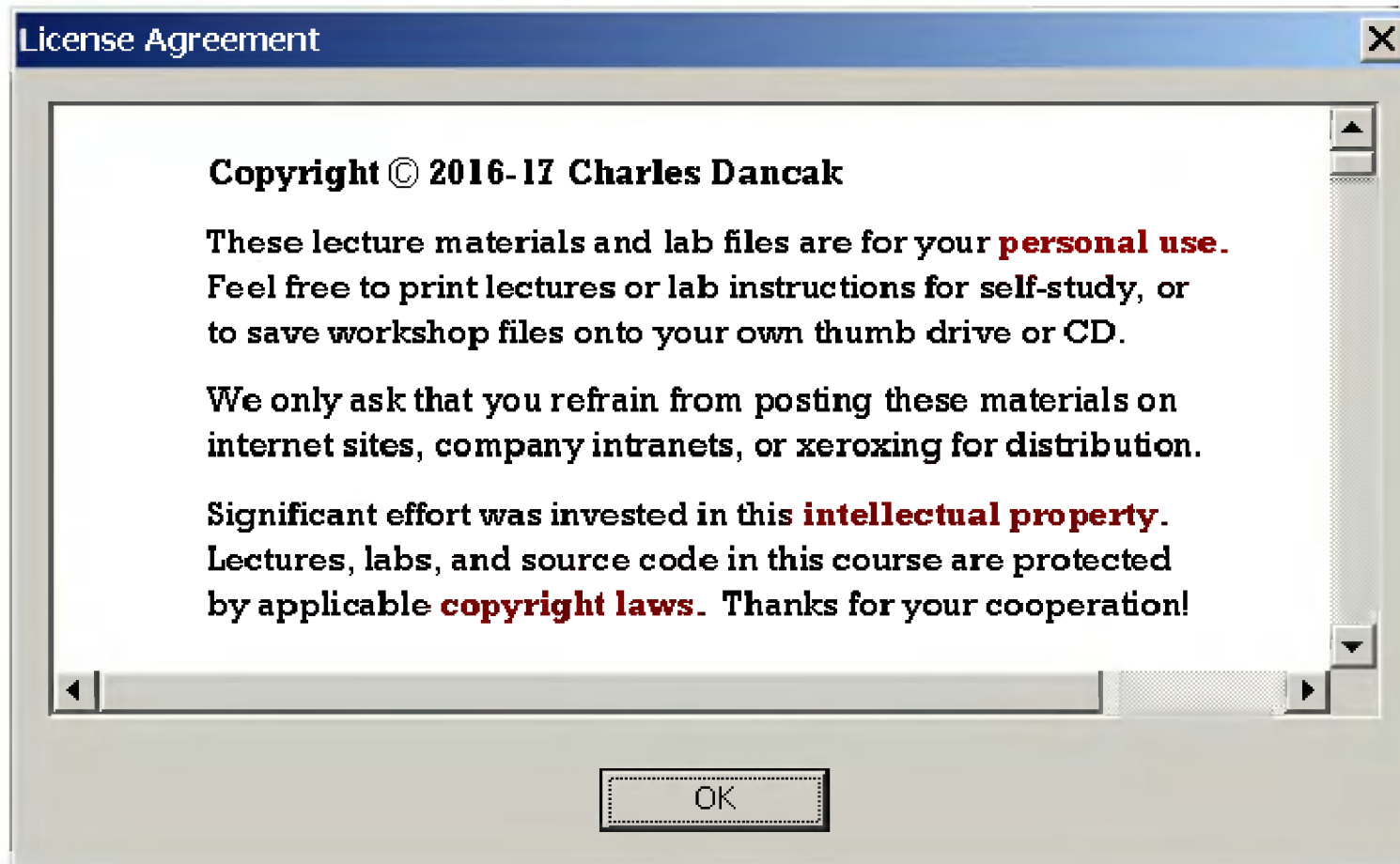


## 6. Корпусирование; Тестирование; Стадия отбраковки



# Авторское право

## 6-2

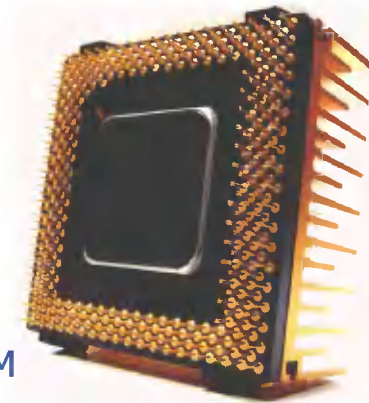


**Чтобы организовать семинары на своем предприятии, свяжитесь с Quantum Consulting по тел: 650/380-9121**

## **Корпусирование кристалла и сборка:**

- **Цель корпусирования**
- **Развитие технологии корпусирования**
- **Технологии проволочных соединений и перевернутого кристалла**
- **Распределение/трассировка**
- **Паразитные эффекты при корпусировании**
- **Рассеивание тепла**

**Корпус Pentium**  
с ребристым теплоотводом



# Шаг корпусирования

6-4

## Корпусирование изделия:

Готовые бездефектные кристаллы (KGD) помещаются в **корпуса** - с использованием проволочных соединений или технологии перевернутого кристалла.

1. Спецификация

2. Проектирование

3. Верификация

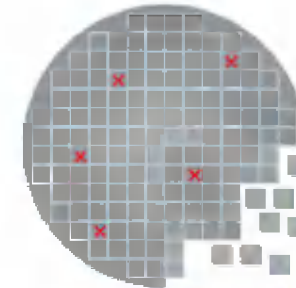
4. Имплементация

5. Производство

6. Отбраковка

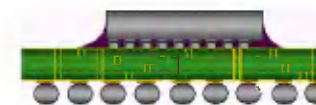
### → **Ключевой принцип:**

Эффективный, дешевый корпус микросхемы является существенным конкурентным преимуществом.



Бездефектные кристаллы

Собранный корпус FCBGA



- Важный вопрос дифференциации изделий—нужно решать **не слишком поздно**.
- Опции корпусирования выбираются на стадии **спецификации**.
- Корпус разрабатывается **совместно** и параллельно с разработкой микросхемы.

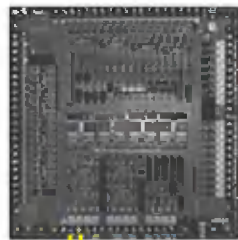


# Цель корпусирования

6-5

- ⌘ Физически **защитить** микроскопические элементы на поверхности кристалла от царапин и нагрузок.
- ⌘ Химически герметизировать кристалл, **изолировать** его от внешних разъедающих и загрязняющих веществ.
- ⌘ Электрически **соединить** входы/выходы кристалла с печатной платой.
- ⌘ Рассеивать в окружающую среду **тепло**, выделяемое кристаллом.

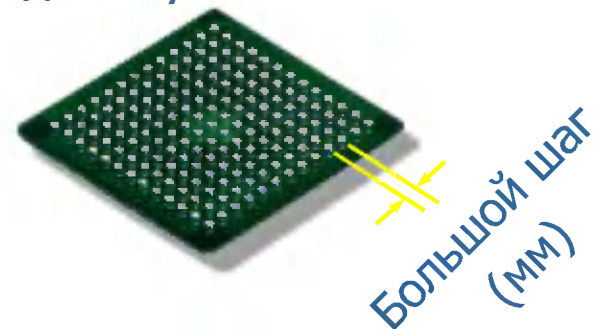
BCM2033  
Bluetooth  
Кристалл



→ || ← Малый шаг  
(мкм)

Корпус Ball-Grid Array  
(BGA, массив шариковых выводов)

Вид снизу



# Развитие технологии корпусирования 6-6



- Рост **возможностей** микросхемы требует наличия большего количества контактов.
- Больше контактных площадок для **подвода питания**, защиты от помех.
- Доля корпуса в общей цене изделия **возрастает**.
- Индустрия движется к **инновационным** трехмерным корпусам.

# Технологии проволочных соединений и перевернутого кристалла (1/2)

6-7

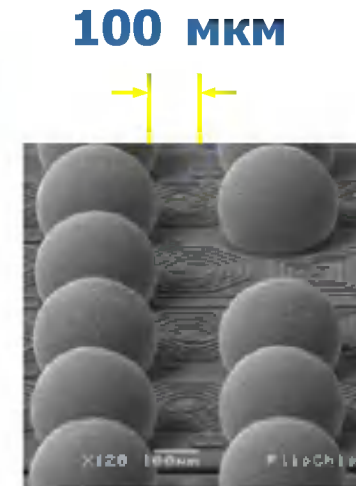


- Перевернутый кристалл - это **технология**, а не конкретный тип корпуса.
- Активная поверхность перевернутого кристалла при монтаже ориентирована **ВНИЗ**.
- Контактные столбики кристалла имеют **непосредственный контакт** с подложкой корпуса.
- Таким образом, технология перевернутого кристалла позволяет **обойтись без связывающих проволочек**.

# Технологии проволочных соединений и перевернутого кристалла (2/2)

6-8

Wire-Bond vs. Flip-Chip		
Feature	Wire-Bond	Flip-Chip
Die Orientation	Face-Up	Face-Down
Area for I/Os	Perimeter Only	Entire Die
I/O Count	$10^2$	$10^3$
Inductance ( $L$ )	1 nH/mm of wire	50 pH per ball
On-Chip Pitch	100 $\mu\text{m}$	200 $\mu\text{m}$
Heat Dissipation	Moderate	Superior



Шарики припоя перевернутого кристалла

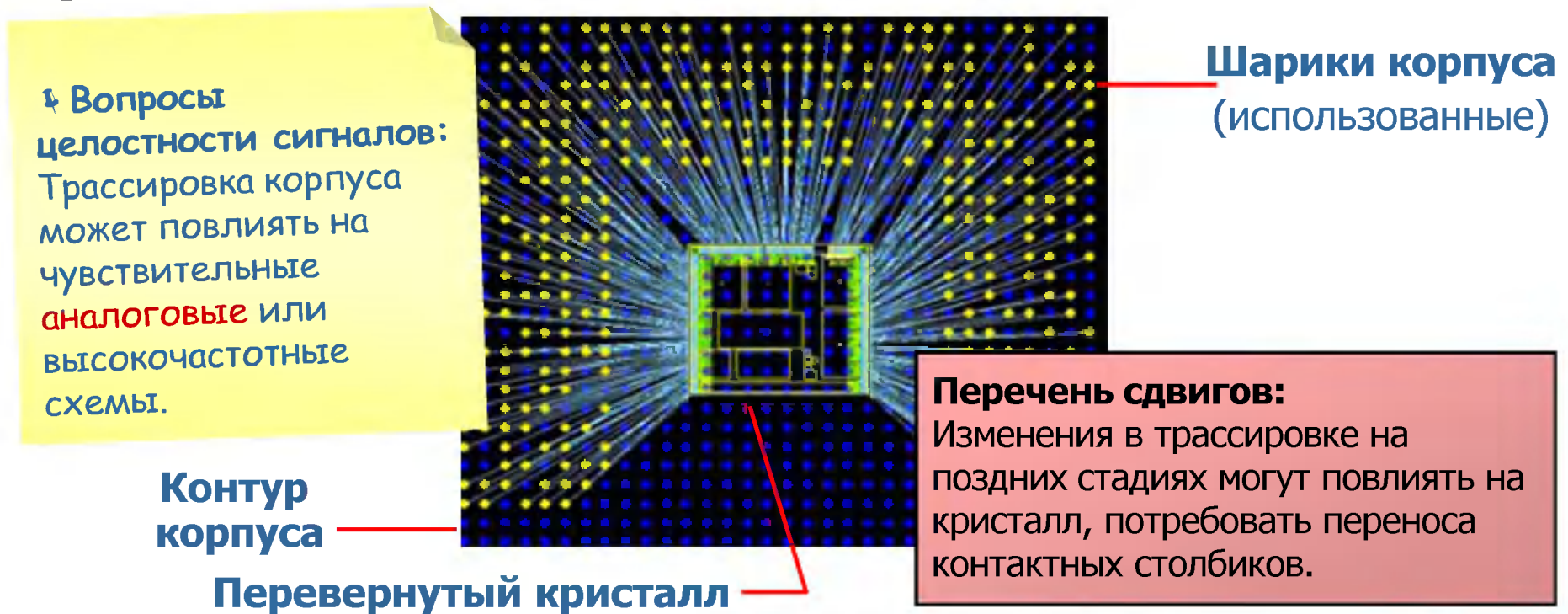
- Выбор между технологиями проволочных соединений и перевернутого кристалла является сложным **компромиссом**.
- Обе технологии **зрелые** и **экономичные**.
- Основным преимуществом технологии перевернутого кристалла является **большее количество вводов/выводов**.

# Перевернутый кристалл BGA 6-9



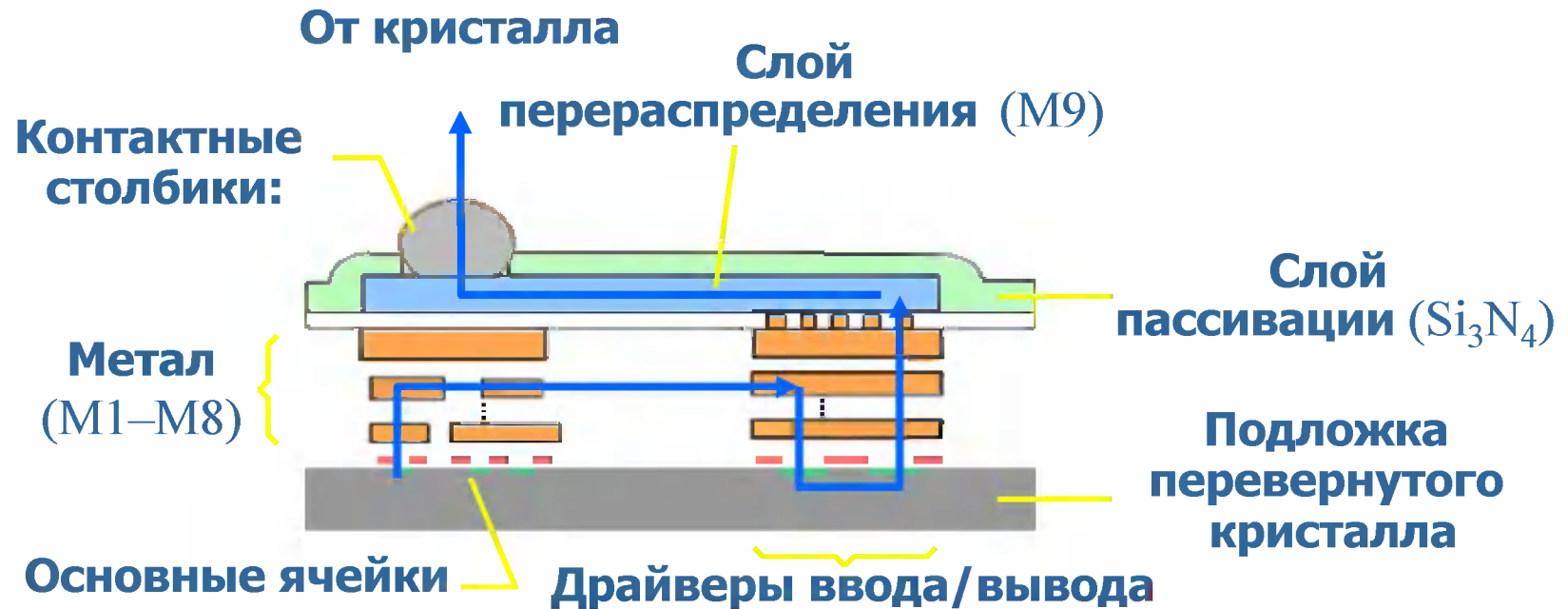
- Компания IBM разработала **controlled collapse chip connect (C4, соединение кристалла управляемым сжатием)**.
- Все входы/выходы соединятся **за один раз**, а не индивидуально, как при использовании проволочек.
- Обусловлена ростом **сложности входов/выходов** и **эксплуатационными качествами**.

# Трассировка корпуса перевернутого кристалла 6-10



- Соединение кристалла и корпуса приводит к новым **проблемам**.
- Массивы входов/выходов могут повлиять на защиту от **статического электричества** и **фиксация уровня**.
- Трассировка корпуса может повлиять на топологию слоя перераспределения (RDL) кристалла.

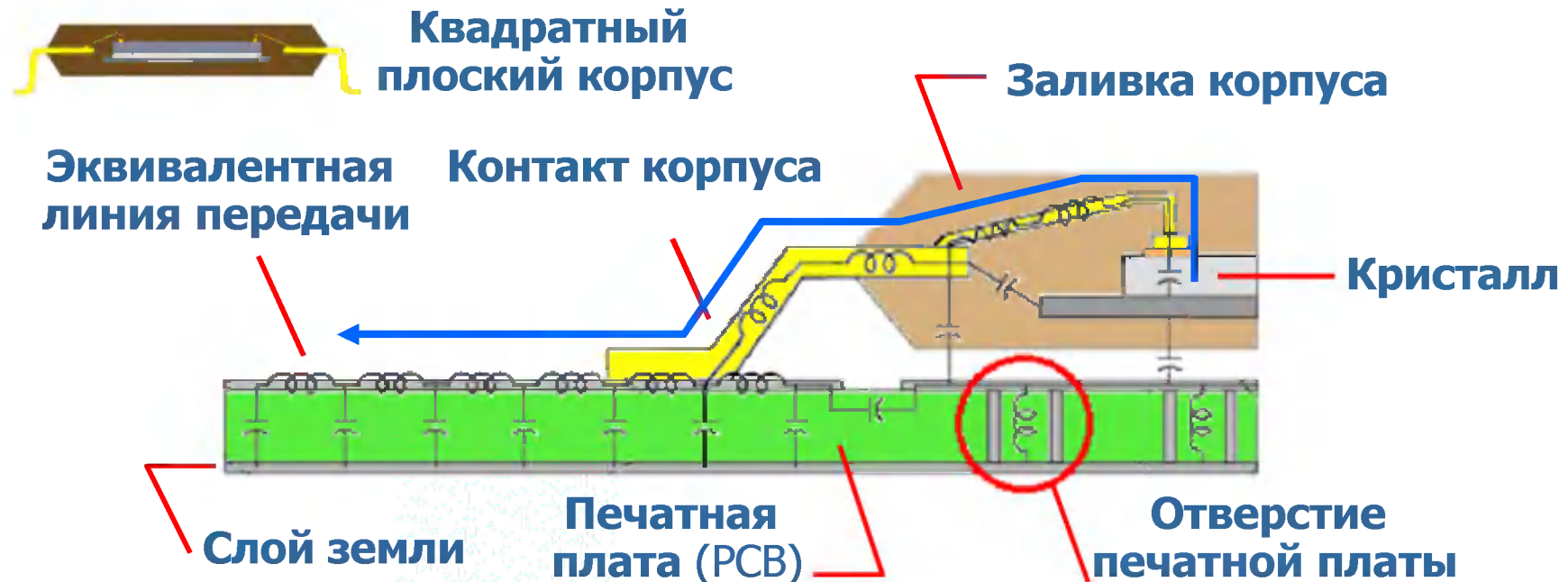
# Распределение/трассировка 6-11



- Первоначально слой перераспределения связывал **контактные столбики** с периферийными **площадками**.
- Густо расположенные контактные площадки **распределяются** по удаленным друг от друга контактными столбикам.
- Теперь служат как **площадки** для контактных столбиков и соединения с драйверами.
- Показан путь от основной ячейки к **выходному** драйверу, а затем из кристалла.

# Паразитные эффекты при корпусировании

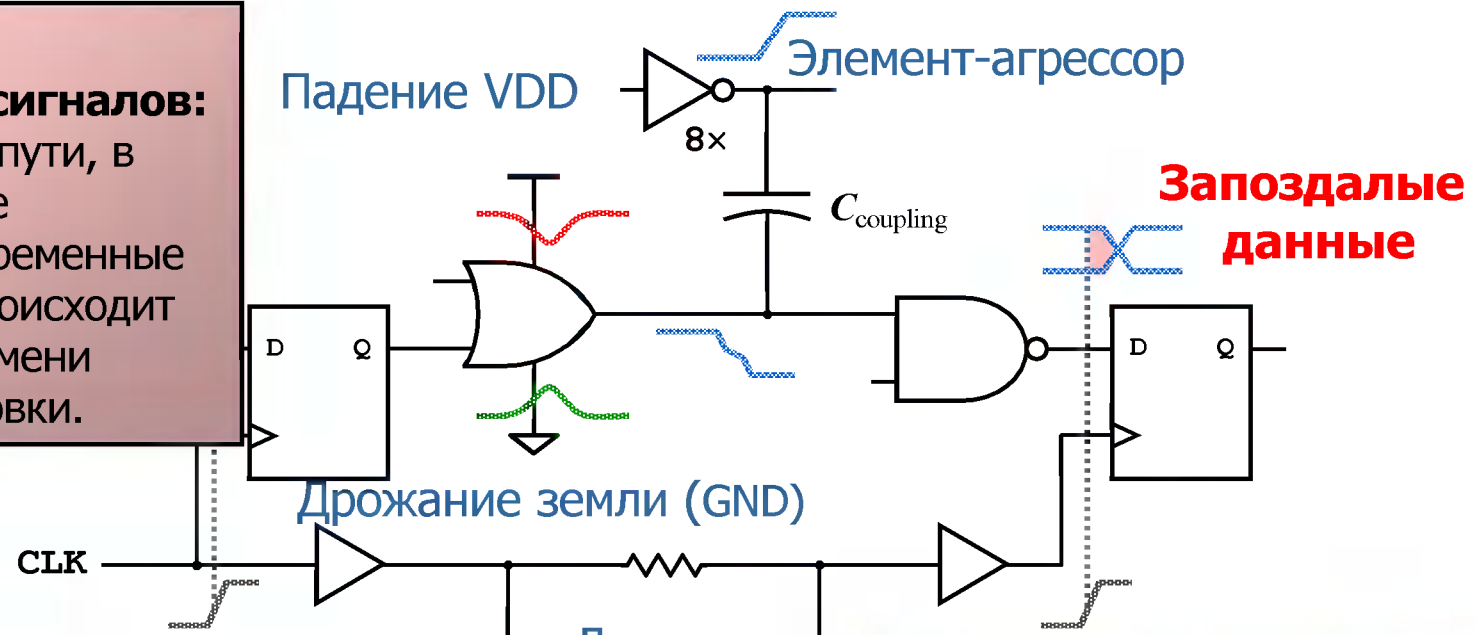
6-12



- Выходной сигнал кристалла проходит через соединения **разных уровней**.
- Паразитные сопротивления, емкости и индуктивности (RLC) влияют на **целостность** сигналов и источников питания.
- На частотах выше 1 ГГц это приводит к искажениям, перекрестным наводкам, «звону».
- Точный анализ должен включать паразитные параметры **корпуса**.



**Нарушение целостности сигналов:**  
 В критическом пути, в котором раннее выполнялись временные требования, происходит нарушение времени времени установки.



	Total chip power	Package parasitics	Jitter
1	1.0 to 1.2W	L=583 pH, C=4pF	660.838ps
2	1.0 to 1.2W	L=541pH, C=4pF	643.873ps
3	1.0 to 1.2W	L=583pH, C=10nF	614.463ps
4	0.8 to 0.9W	L=583pH, C=4pF	466.984ps
5	1.0 to 1.2W	L= 0pH, C=4pF	160.484ps

**Дрожание фазы тактового сигнала (Jitter):**  
 Разброс времени поступления фронта.

- **Выходные сигналы кристалла определяют моменты изменения потребляемой мощности.**
- **Переключение может вызвать падение VDD и дрожание земли.**
- **Результирующий шум питания является одной из причин дрожание фазы тактового сигнала (clock jitter).**

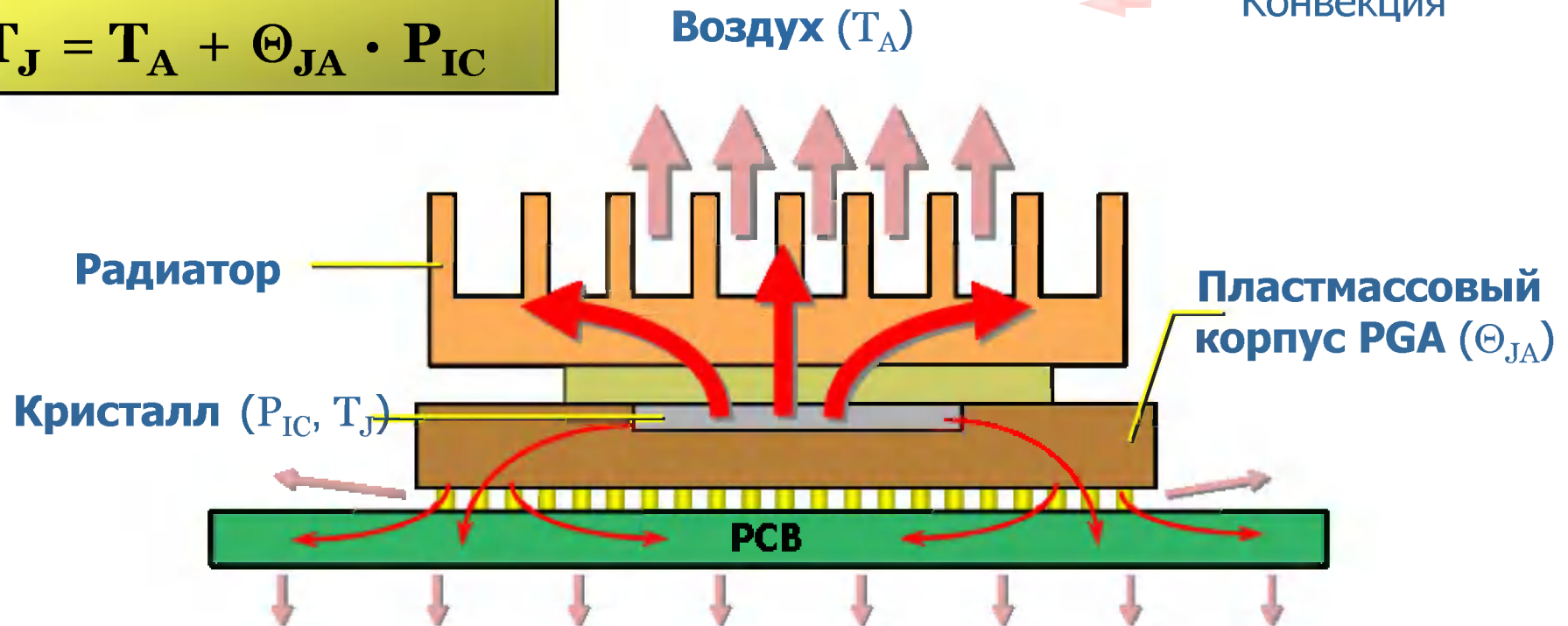
# Рассеивание тепла

6-14

Тепловой закон Ома:

$$T_J = T_A + \Theta_{JA} \cdot P_{IC}$$

← Теплопроводность  
← Конвекция

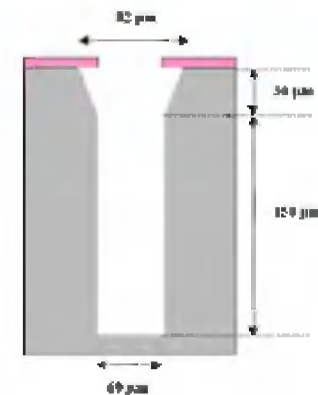


- В основном тепло рассеивается в окружающий воздух за счет **теплопроводности**.
- Конвекция (поток теплого воздуха) уносит меньше тепла.
- **Тепловое сопротивление** ( $\Theta_{JA}$ ) корпуса определяет  $\Delta T$ .

## Трехмерные корпуса:

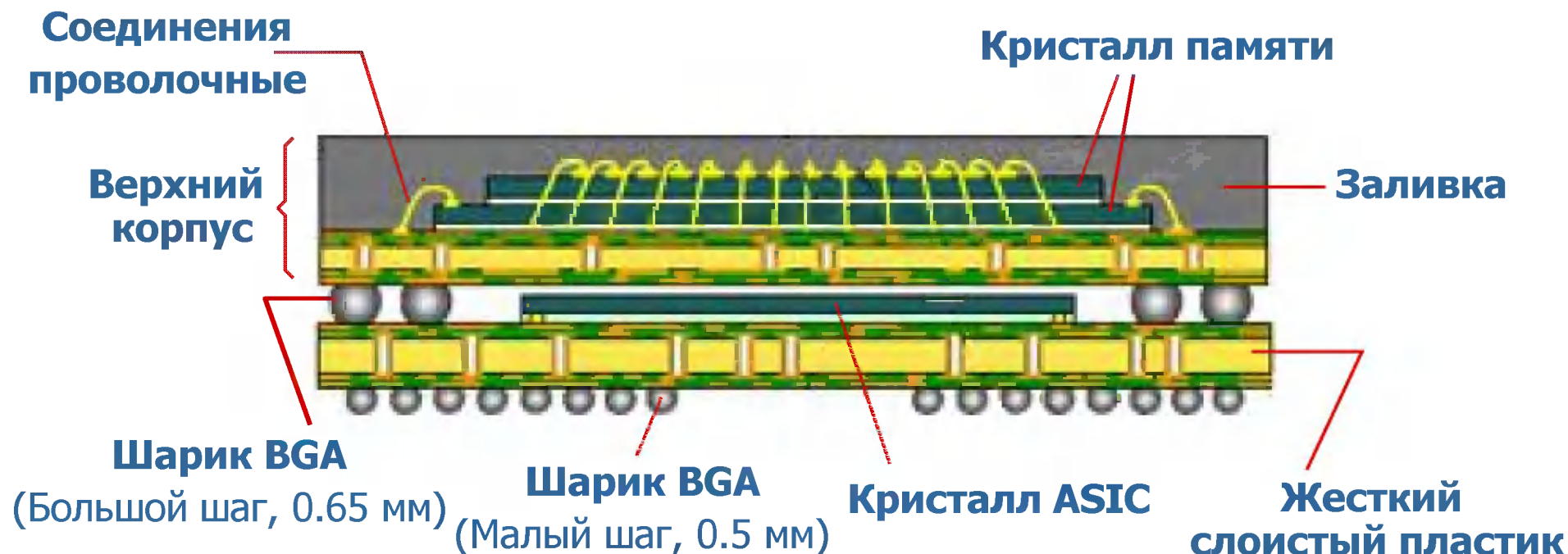
- Корпус-на-корпусе
- Этажерка кристаллов 3DIC
- Средства разработки топологии 3DIC
- Пример 3DIC

Конусообразное  
слепое отверстие  
в кремнии (TSV)



# Корпус-на-корпусе (POP)

6-16



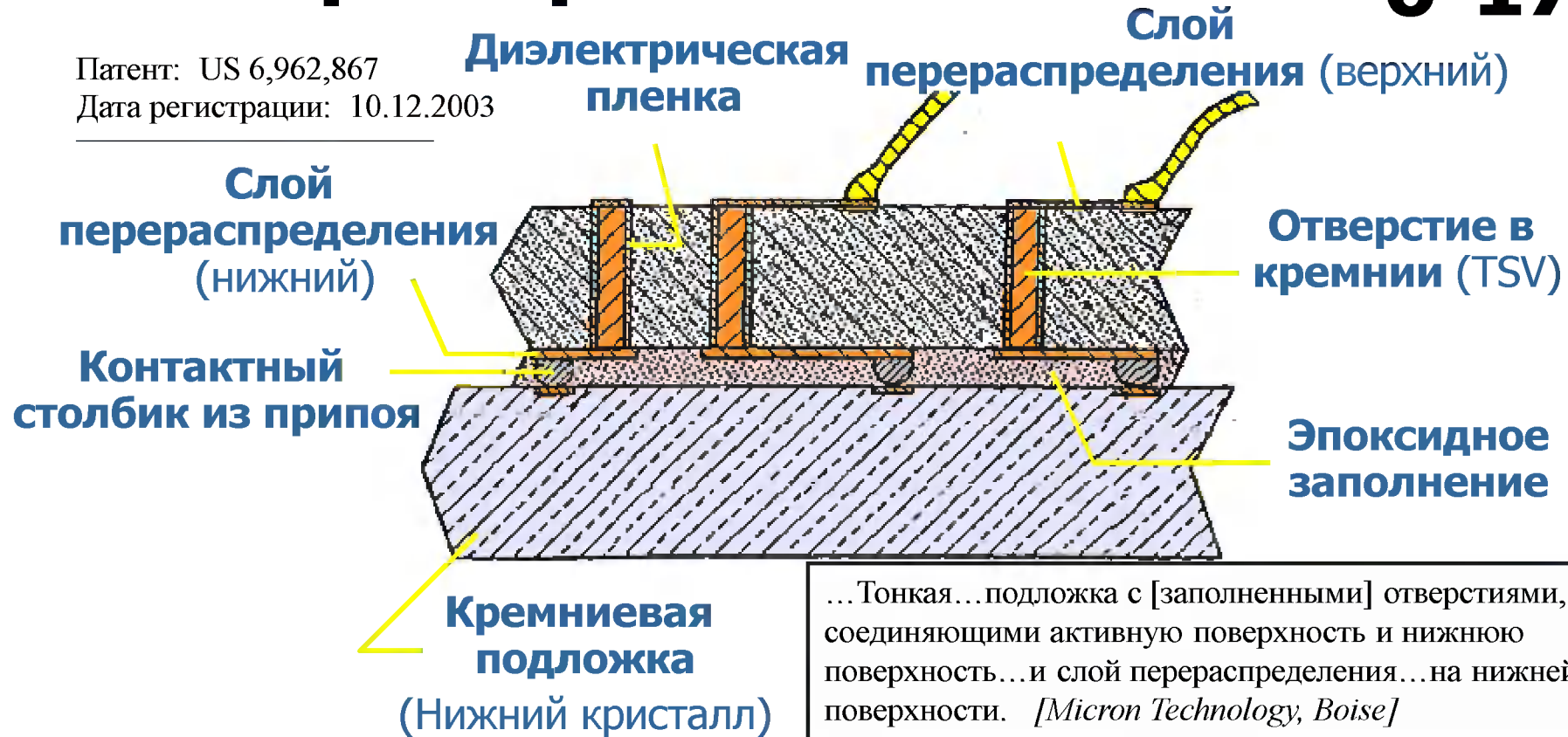
- Этажерка из **двух** корпусов, один - логика, другой - ОЗУ.
- Уменьшает **длину соединений** между кристаллами телефона и ОЗУ.
- ОЗУ можно **протестировать** отдельно от корпуса микросхемы сотового телефона.
- Используется на рынках **мобильных телефонов** и **цифровых фотокамер** и им подобным.

# Этажерка кристаллов 3DIC

6-17

Патент: US 6,962,867

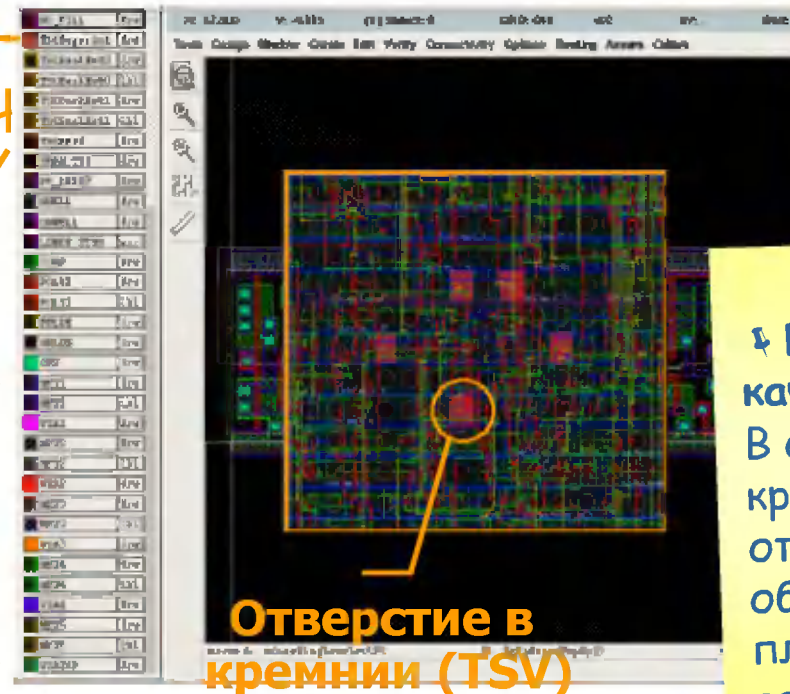
Дата регистрации: 10.12.2003



- 3D технология объединяет кристаллы малой толщины **вертикальными соединениями**.
- Глубокие вытравленные отверстия в кремнии проходят **через весь** верхний кристалл.
- Высокая степень травления достигается использованием реактивного ионного травления (RIE) плазмой ( $SF_6$ ).

# Средства разработки топологии 3DIC 6-18

Отверстие в кремнии (TSV)  
Слой перераспределения RDL (нижний)  
Контактная площадка (нижняя)  
Металл (M1–M5)



✂ Средства EDA:

Редактор топологии Cadence (с поддержкой 3D): *Virtuoso*.

✂ Высокие эксплуатационные качества:

В отличие от технологии кристалл-на-кристалле, отверстия в кремнии обеспечивают **большую** плотность отверстий, **короткие** соединения.

- Короткие соединения позволяют использовать **маломощные** драйвера линий.
- Отверстия в кремнии позволяют интегрировать **гетерогенные** кристаллы.
- Цифровые, ВЧ (RF), микроэлектромеханические (MEMS) и оптические кристаллы в **одном** корпусе.

# Пример корпуса 3D IC

6-19

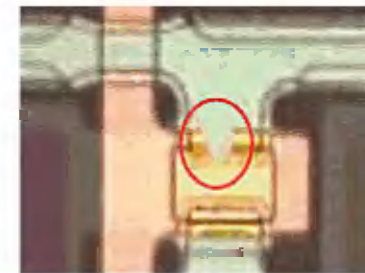


- Apple A6 является **3D-IC** TSMC 28-нм с малым энергопотреблением.
- Сердце iPad, включает в себя кристалл процессора и два кристалла ОЗУ.
- 3D обещает рост производительности на **30%** при потреблении **50%** мощности.

## Контроль в процессе изготовления:

- Физические дефекты
- D-алгоритм
- Проектирование для периферийного сканирования
- Тестирование IP-ядер
- Тестирования уровня платы
- Встроенное самотестирование

Дефект производства





# Проектирование для тестирования (DFT) и процесс тестирования

# 6-21

## Стратегия проектирования для тестирования (DFT) в масштабах кристалла



→ **Ключевой принцип:**  
Всестороннее тестирование является важным фактором экономической эффективности системы на кристалле.

## Изменение порядка в цепочках сканирование

**Кодирование для DFT:**

- Выполнение указаний DFT.
- Выполнение проверок DFT кода RTL.

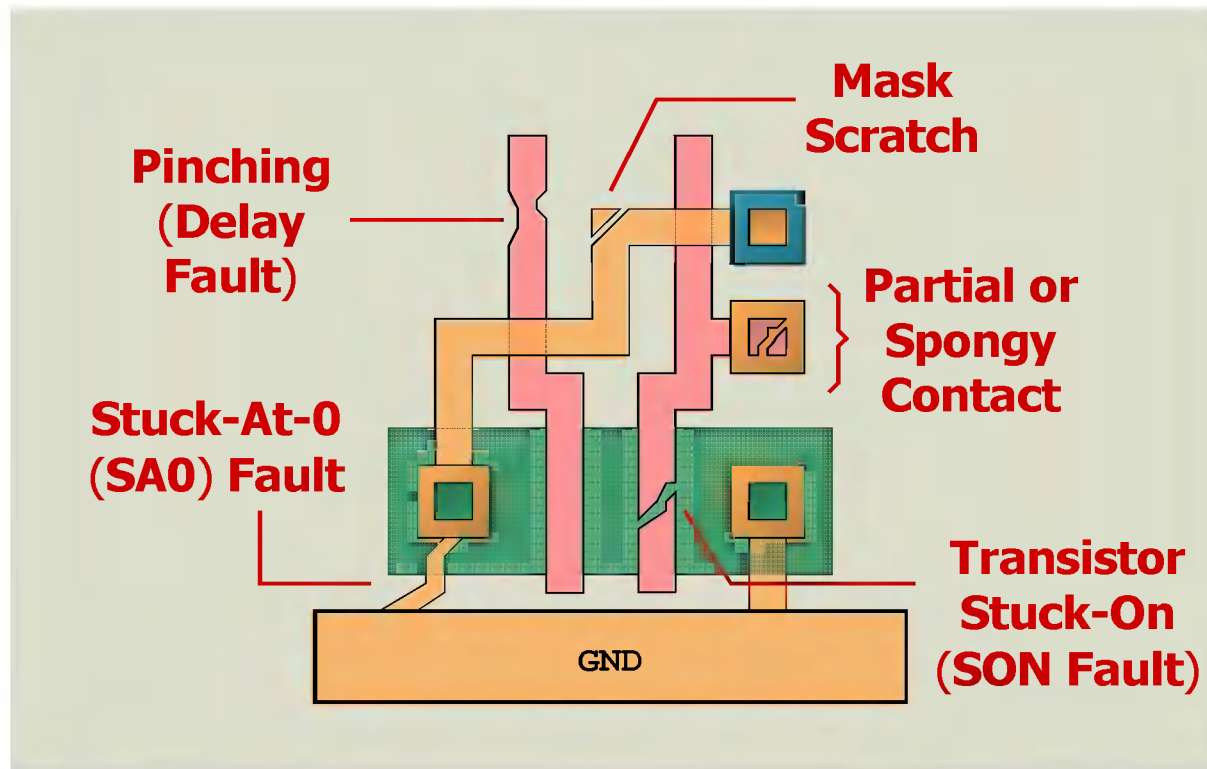
**Синтез DFT:**

- Исправление нарушений DFT.
- Добавление цепочек сканирования.
- Синтез логики встроенного самотестирования (BIST).

**Контроль в процессе производства**

## Характеристика

- Проектирование для тестирования и выполнение тестирования **охватывает** большую часть цикла разработки ASIC.
- Целью является достижение высокой **тестопригодности** блоков кристалла.
- Включает синтез нефункциональной логики, такой как логика встроенного самотестирования (BIST).



- Физические дефекты неизбежно возникают в процессе **производства**.
- Слишком разнообразны для непосредственного обнаружения; вместо этого используется **логические** модели отказов.
- Основной моделью все еще являются константные отказы: **SA0** или **SA1**.

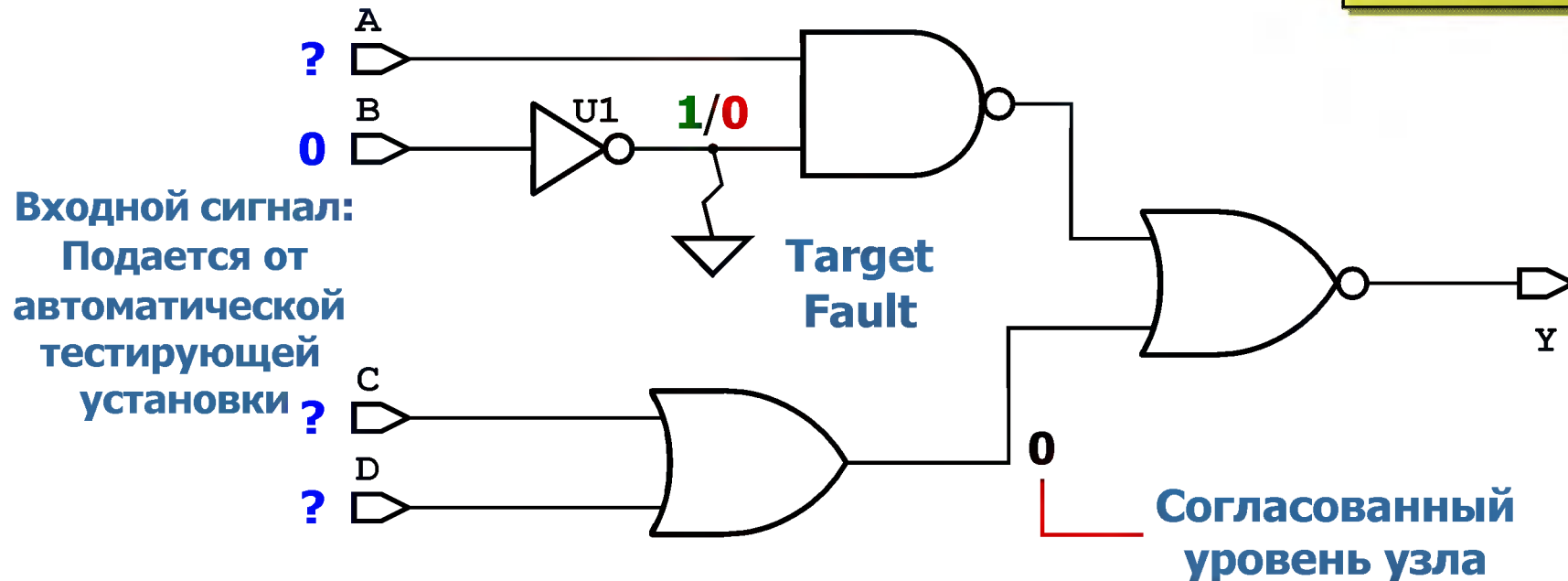
# D-алгоритм (1/2)

6-23

Network  $\mathcal{N}_1$   
(A Tiny Chip)

Активировать отказ

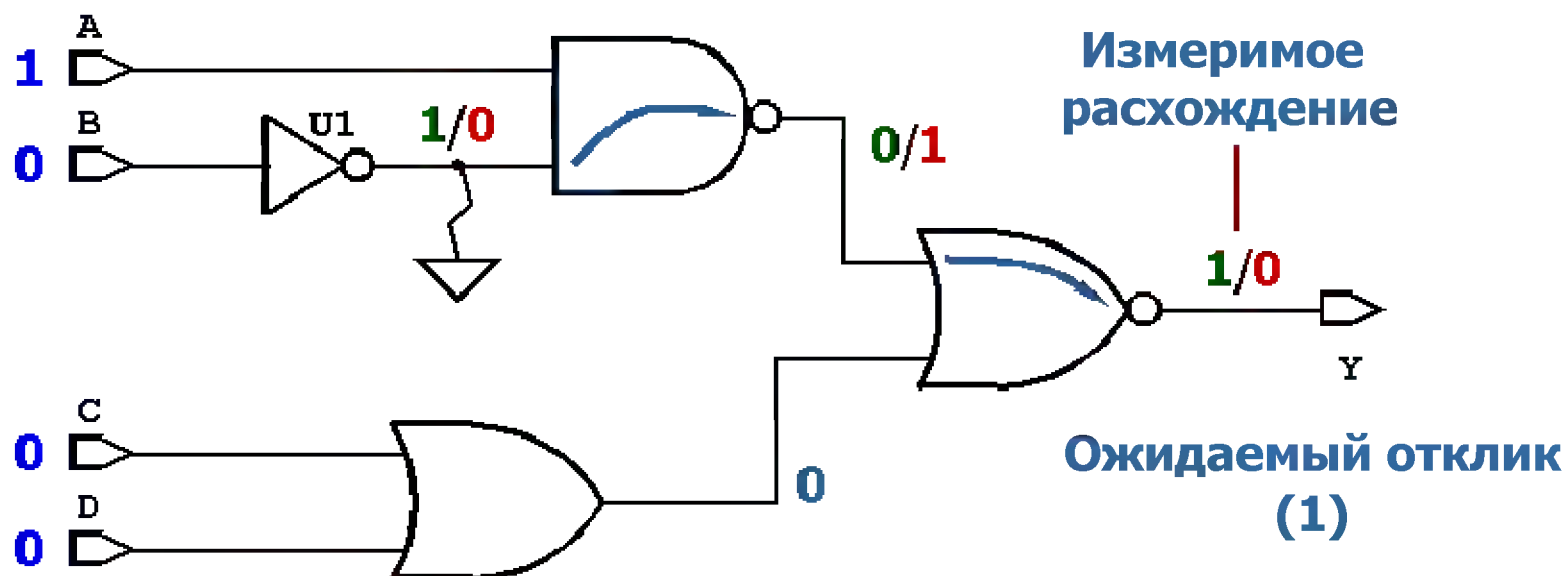
(Установить на узле высокий потенциал)



- Из списка выбирается **целевой отказ**: SA0 на выходе инвертора U1/Y.
- Автоматическая тестирующая установка подает **воздействие** для перевода отказавшего узла в противоположное состояние **1**.
- Не используются зонды - реакция распространяется на выходной контакт.

# D-алгоритм (2/2)

6-24



- Ожидаемый **безошибочный** отклик выхода  $Y$ , который измеряется автоматической тестирующей установкой, равен **1**.
- Если вместо этого наблюдается ошибочный выход, то **тестовый шаблон** приводит к сбою.
- Кристалл со сбоем отбраковывается—или отправляется в лабораторию для **анализа дефектов**.

# Типичная программа тестирования 6-25

Программа тестирования для цепи  $N1$

Тестовый шаблон:  
Дефект: SA0 отказ в U1/Y.

✦ Структурные отказы:  
Для обнаружения 100% структурных отказов в логической цепи  $N1$  нужно намного меньше, чем  $2^4$  функциональных шаблонов.

```
STIL 1.0;  
.  
.  
Pattern "N1_Burst" {  
  Vector {■■■■■■■■■■}  
  Vector {■■■■■■■■■■}  
  Vector {■■■■■■■■■■}  
  Vector {■■■■■■■■■■}  
  Vector {ALL=1000 1;}  
  Vector {■■■■■■■■■■}  
  Vector {■■■■■■■■■■}  
  Vector {■■■■■■■■■■}  
  Vector {■■■■■■■■■■}  
}
```

✦ Средства EDA:  
*TetraMAX ATPG*

Программа тестирования для  $N1$

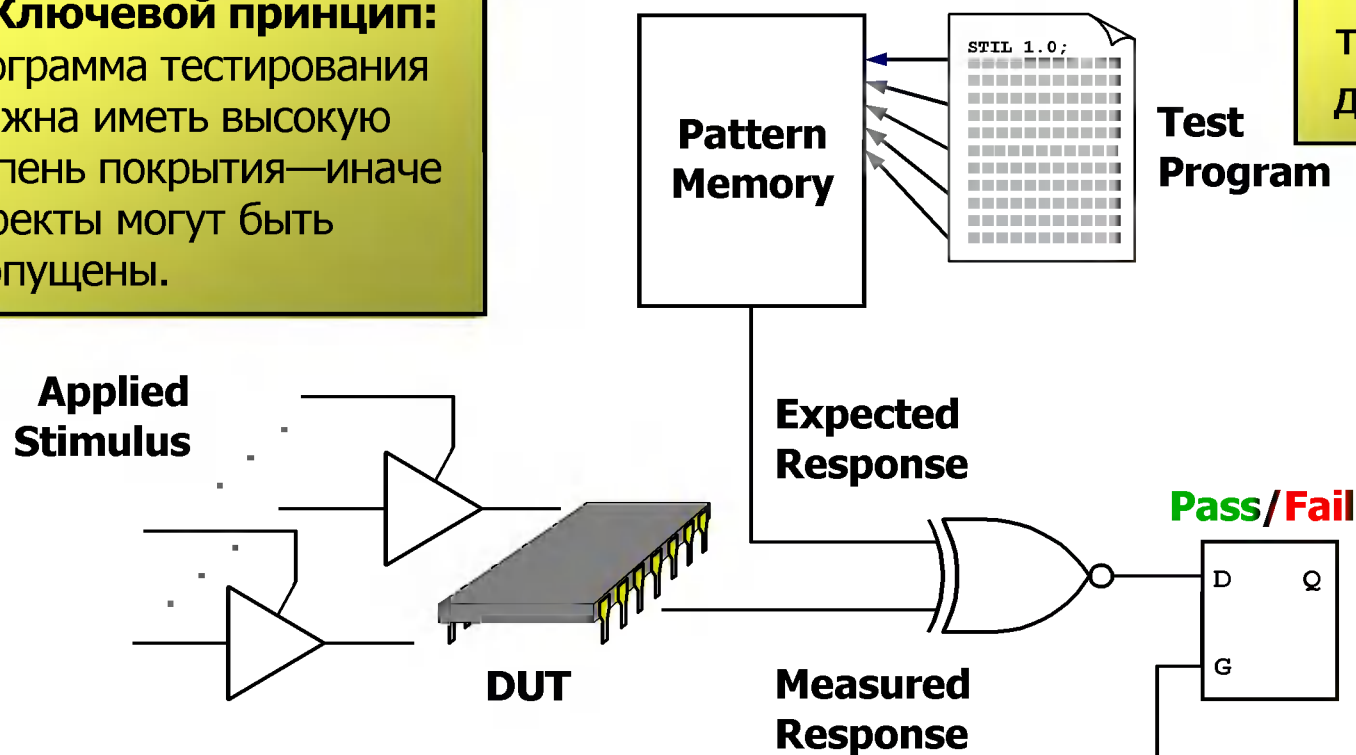
Входной сигнал ALL = 1000 1 — Ожидаемый отклик

- Автоматический генератор тестовых комбинаций (ATPG) генерирует тестовые шаблоны для каждой целевой ошибки в  $N1$ .
- Полный набор шаблонов генератора тестовых комбинаций (ATPG) составляет готовую программу для автоматической тестирующей установки.

# Внутри автоматической тестирующей установки 6-26

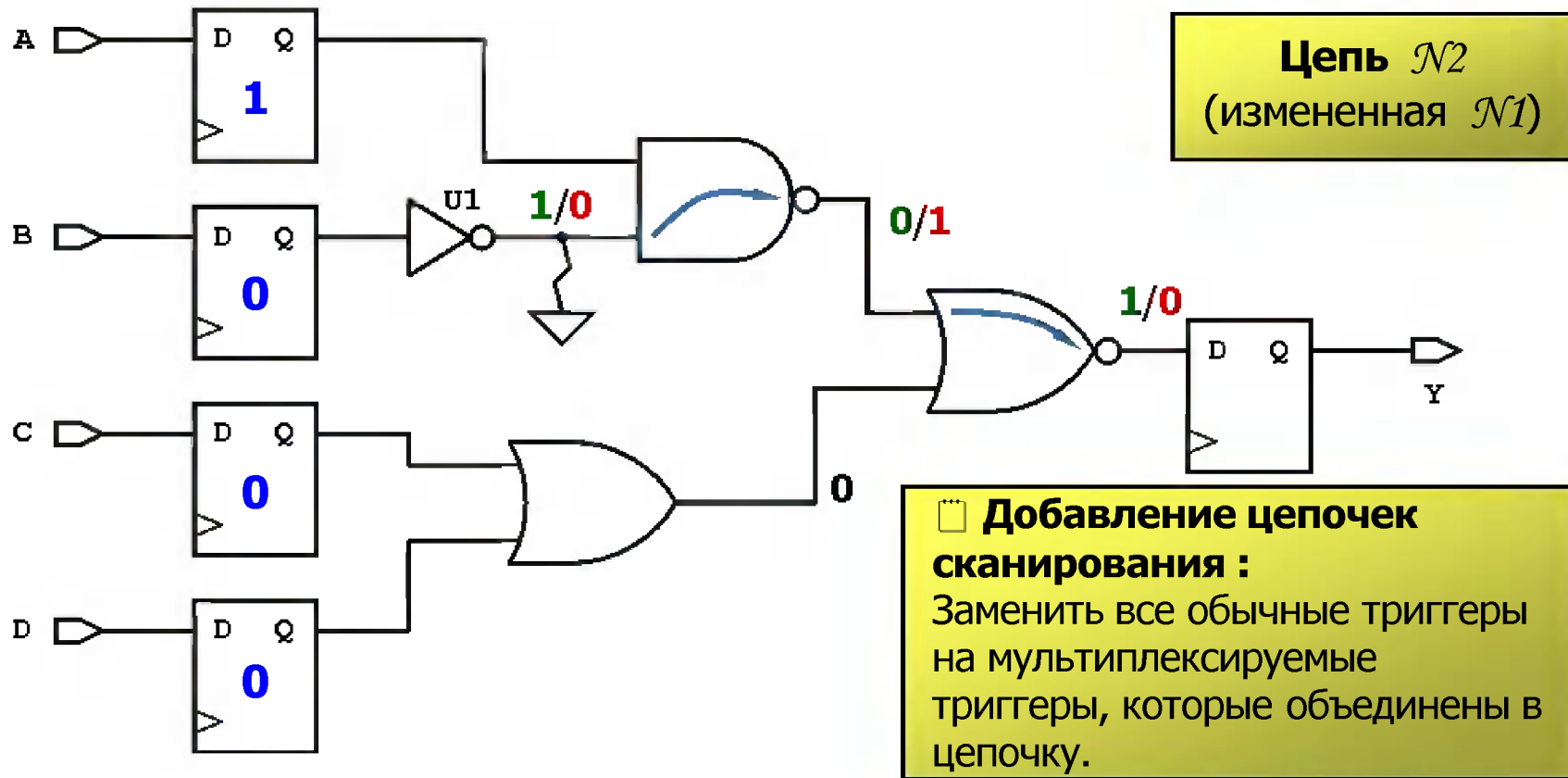
↔ **Ключевой принцип:**  
Программа тестирования должна иметь высокую степень покрытия—иначе дефекты могут быть пропущены.

Программа тестирования для цепи *N1*



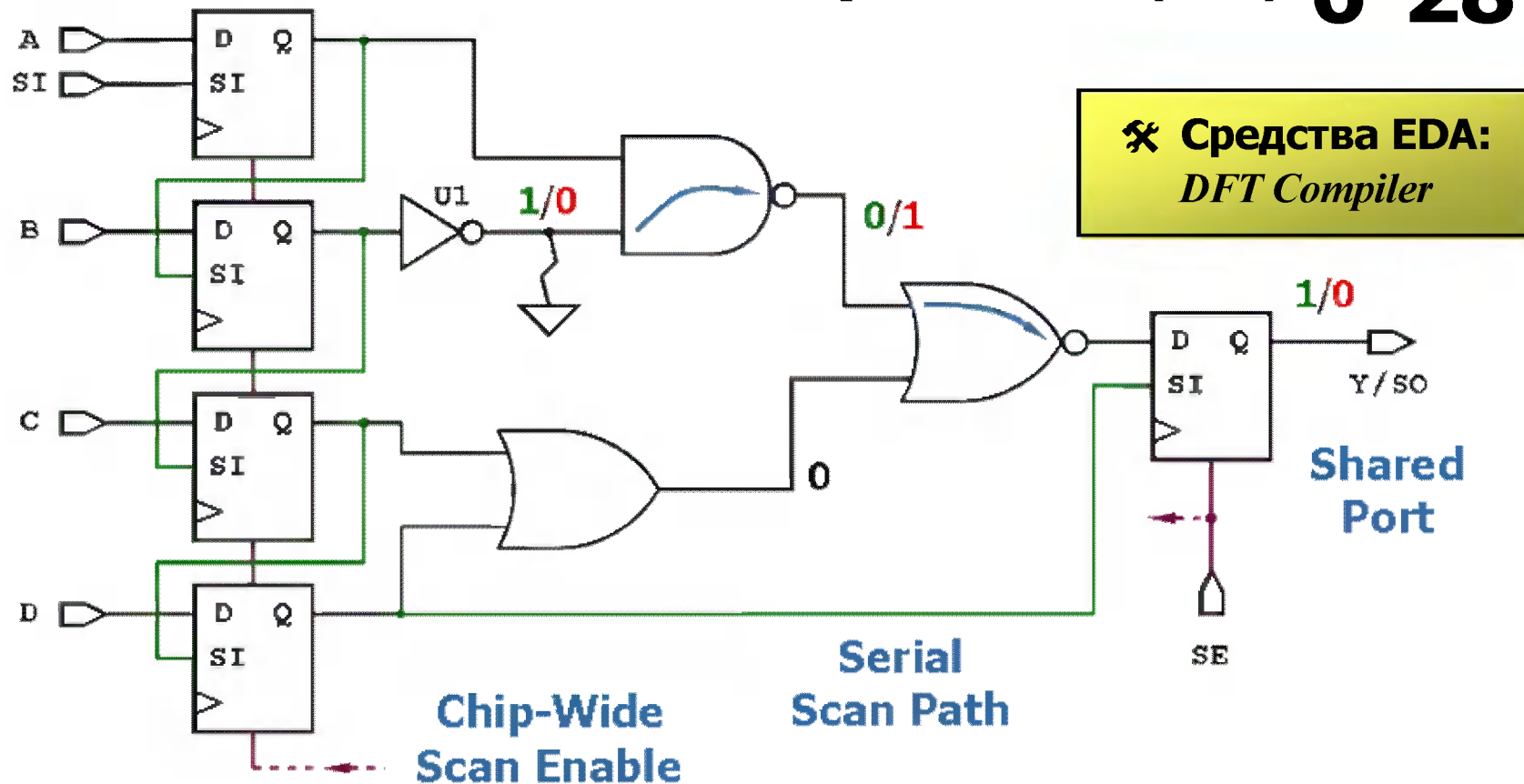
- Автоматическая тестирующая установка подает стимулирующие воздействия шаблона, затем **стробирует** отклик.
- Шаблон выполняется **медленно** (10 МГц), чтобы оставить достаточно времени для установления состояния.
- Шаблон приводит к ошибке, если реальный отклик тестируемого устройства (DUT) отличается от ожидаемого.

# Добавление цепочек сканирования (1/2) 6-27



- Но как D-алгоритм может **добраться** до логики между регистрами стадий?
- Автоматическая тестирующая установка должна как-то ввести тестирующее воздействие в триггеры.
- Простой подход: вставить сканируемые триггеры и вдвинуть воздействие в них.

# Добавление цепочек сканирования (2/2) 6-28



- Теперь выполнение того же тестового шаблона занимает **пять** циклов, а не один.
- Для последовательной загрузки цепочки сканирования нужно активировать SE и вдвинуть **1000**.
- После этого деактивировать SE. Подать тактовый импульс чтобы зафиксировать **1/0**. Проследить за SO.



# Исправление нарушений DFT 6-29

✂ Средства EDA:  
*DFT Compiler*

Дополнительный контакт

ASIC\_TEST

Тактовый сигнал  
автоматической  
тестирующей  
установки  
(~10 МГц)

CLK

Clock  
Divider

Обходной мультиплексор

Нарушение правил  
проектирования для  
тестирования

Любая из этих проблем делает  
невозможной загрузку (выгрузку)  
данных сканирования.

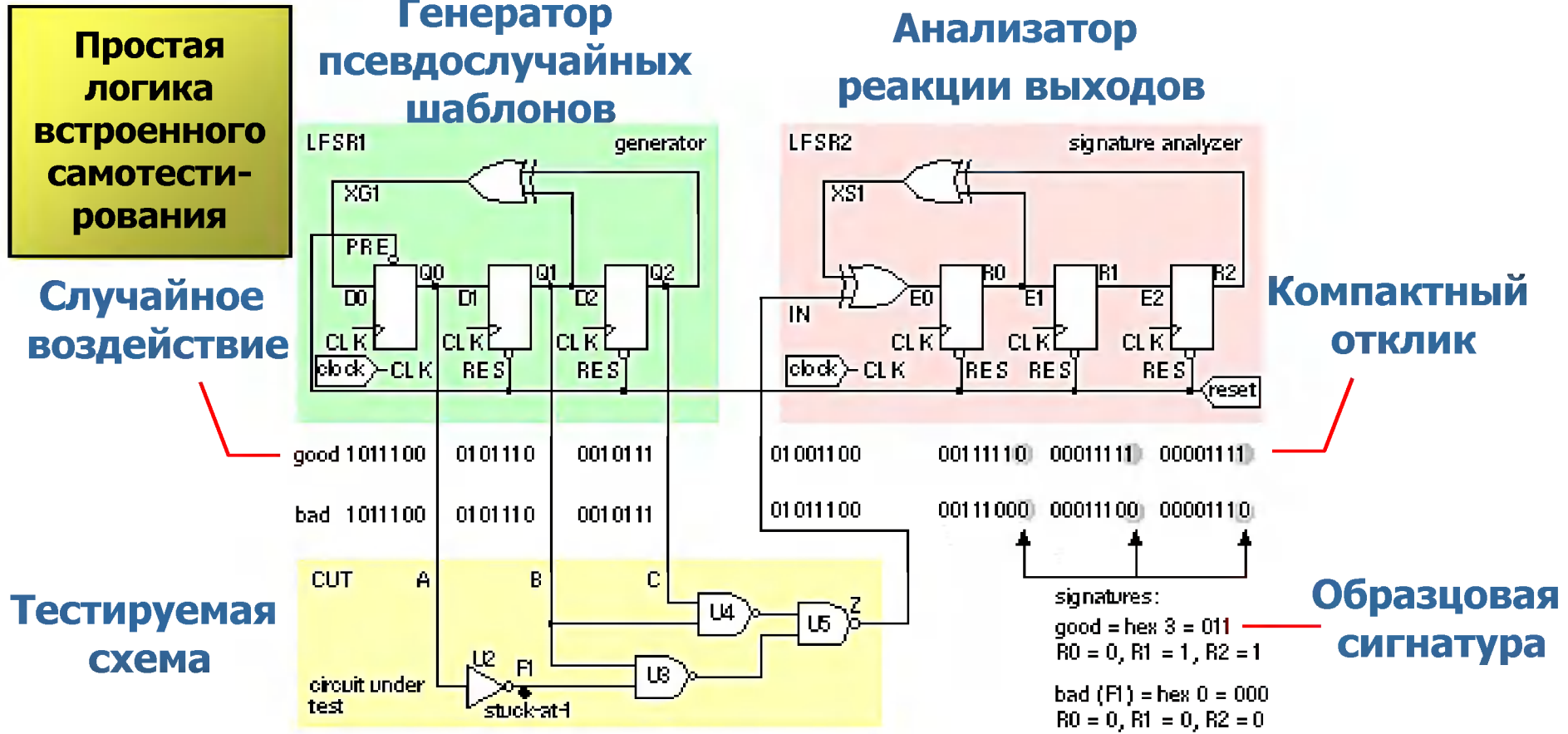
Key Principle:

ATE clock must control  
each scan flop directly.

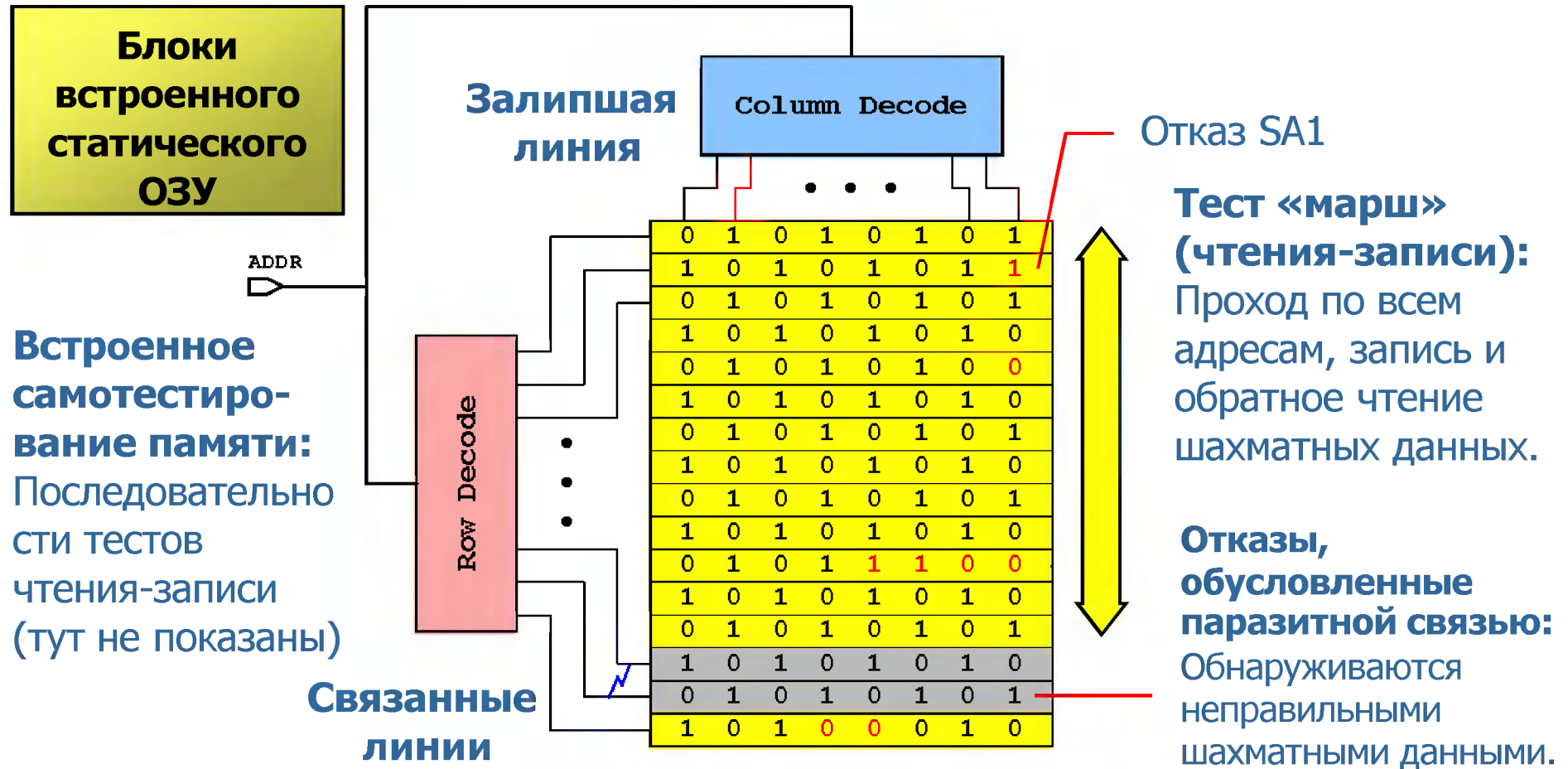
- Надежное выполнение операций сканирования обычно требует **специальных действий для проектирования для тестирования**.
- Тут добавлен мультиплексор для **обхода** делителя тактовой частоты при сканировании.
- Это выполняется только тогда, когда `ASIC_TEST` имеет **высокий уровень**. Добавляется небольшая расфазировка тактовых импульсов.

# Альтернатива: Логика встроенного самотестирования (BIST)

# 6-30

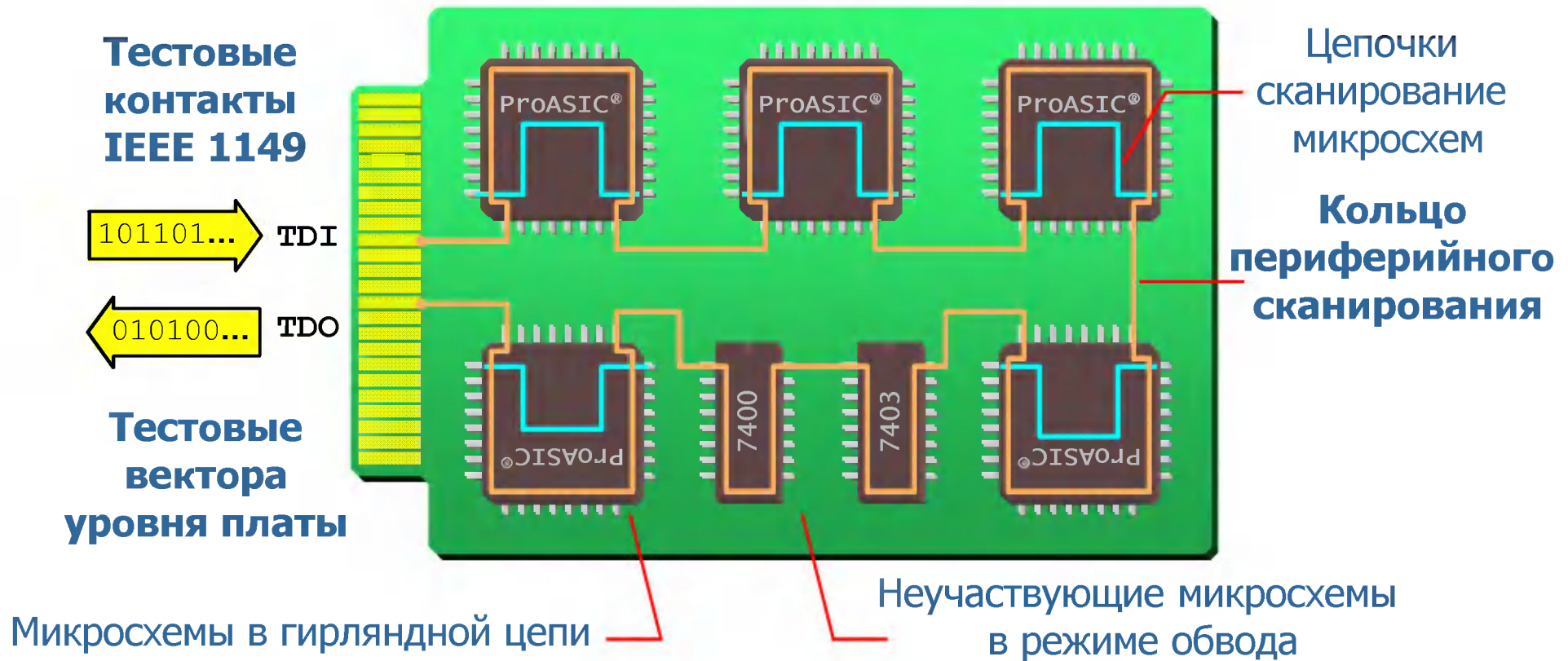


- Встроенное самотестирование (BIST) является альтернативой цепочек сканирования.
- Источник шаблонов и анализатор реализован **в кремнии**— автоматическая тестирующая установка не нужна.



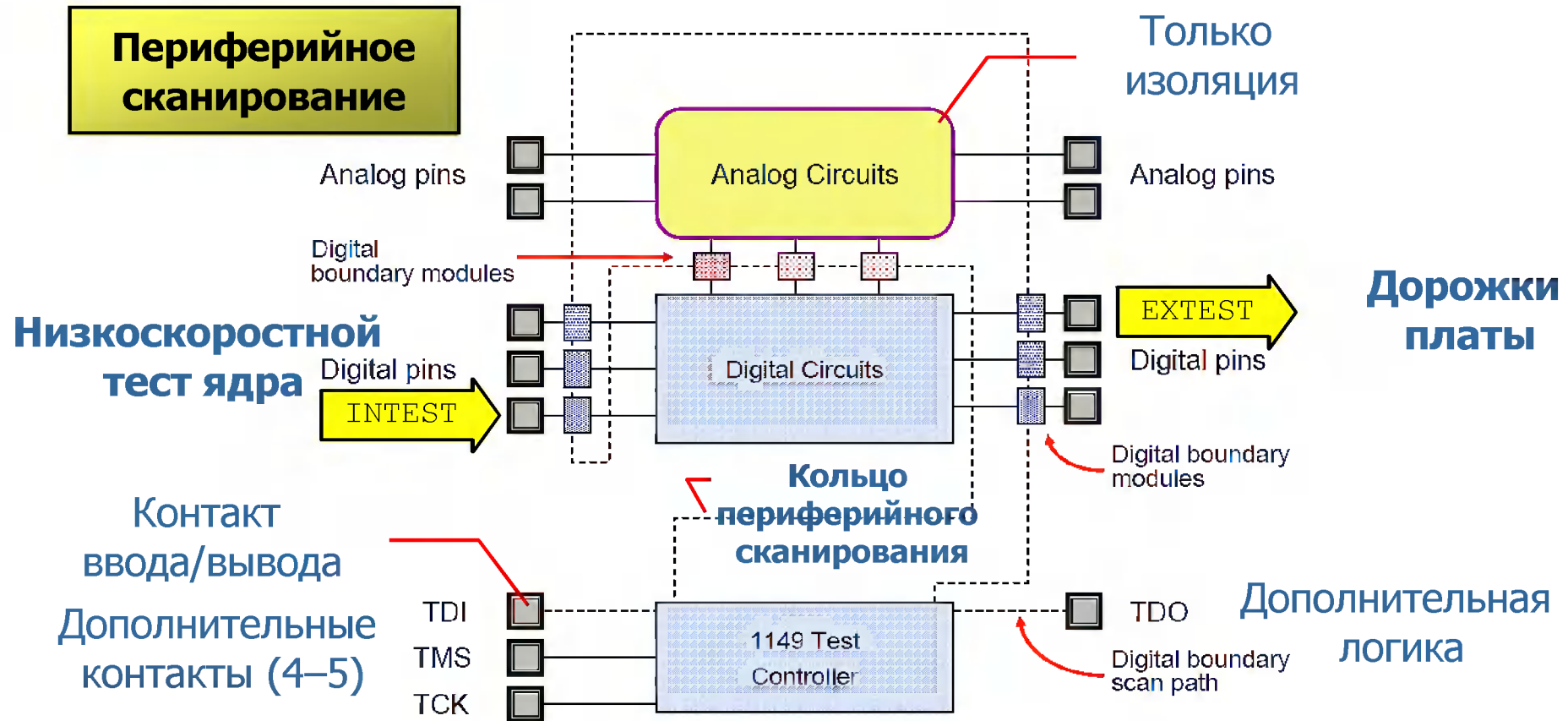
- ОЗУ/ПЗУ на кристалле состоят из **плотно** расположенных транзисторов.
- Встроенное самотестирование памяти должно обнаруживать характерные для памяти ошибки, обусловленные, например, паразитной связью.

# Поддержка тестирования уровня платы 6-32



- Поддержка тестирования **уровня печатной платы**: дорожки платы и **ASIC**.
- Синтезирует с использованием **периферии** совместимых кристаллов.

# Цифровая архитектура 1149.1 6-33

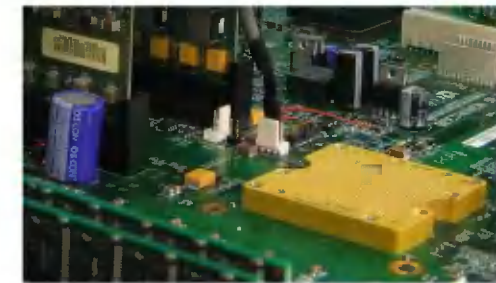


© S.K. Sunter (2004) Logic Vision

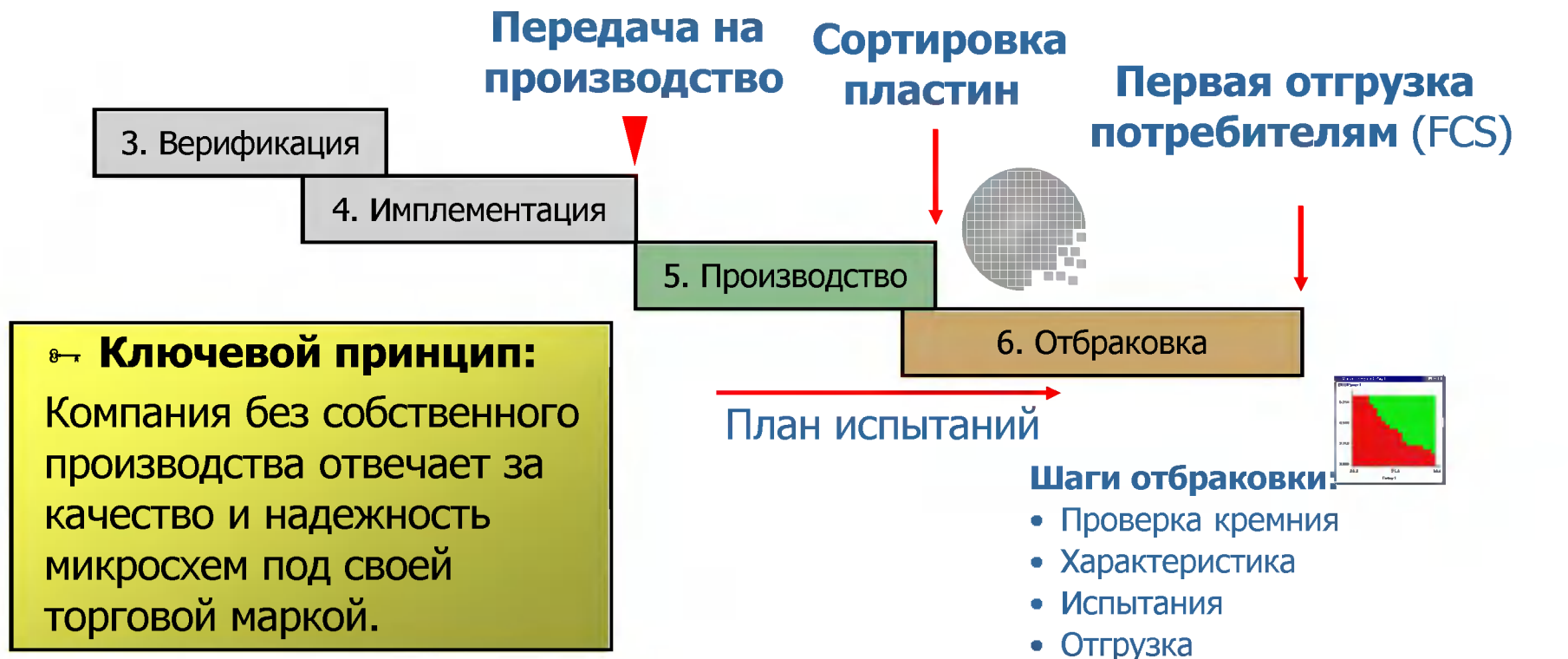
- Кольцо периферийного сканирования поддерживает последовательную загрузку и выгрузку.
- Биты воздействия поступают через **TDI**, а биты отклика выходят через **TDO**.

## Этапы приемочных испытаний:

- Плата для испытаний
- Переработка металлических масок
- Микрохирургия сфокусированным ионным пучком (FIB)
- Испытания критически важны



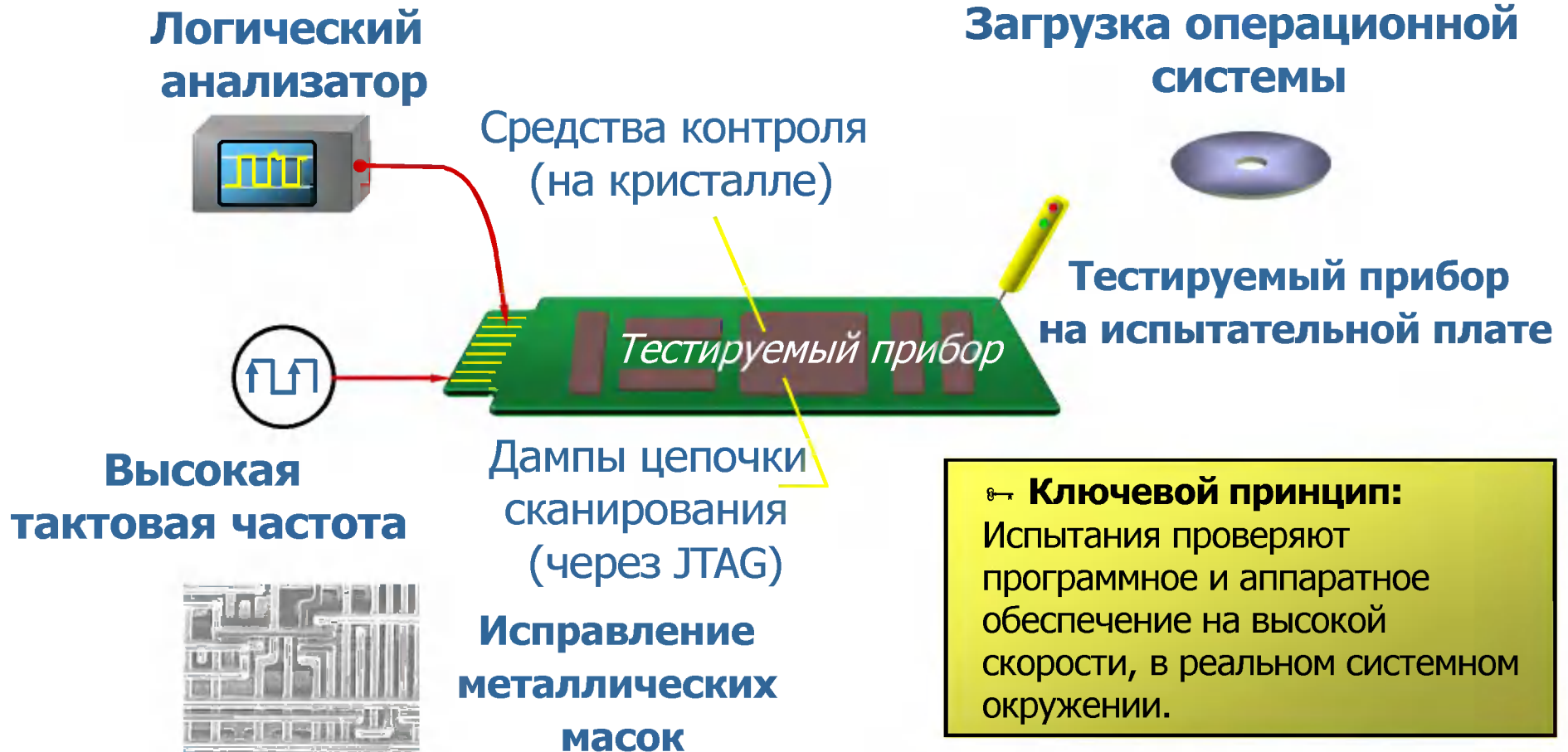
**Тестируемый прибор  
на испытательной плате**



- **Испытание гарантирует **готовность ASIC** к продажам.**
- **Уровень качества часто определяется **маркетинговой нишей**.**
- **В некоторых случаях уровень качества определяется **ключевыми потребителями**.**

# Плата для испытаний

6-36



- Корпусированный тестируемый прибор сейчас работает на **целевой** тактовой частоте.
- На плате находятся **FPGA драйверы** и **периферийные** микросхемы.



# Микрохирургия сфокусированным ионным пучком

# 6-37

**A. Травление линии:**  
Разрушает соединение  
металлического слоя 1.

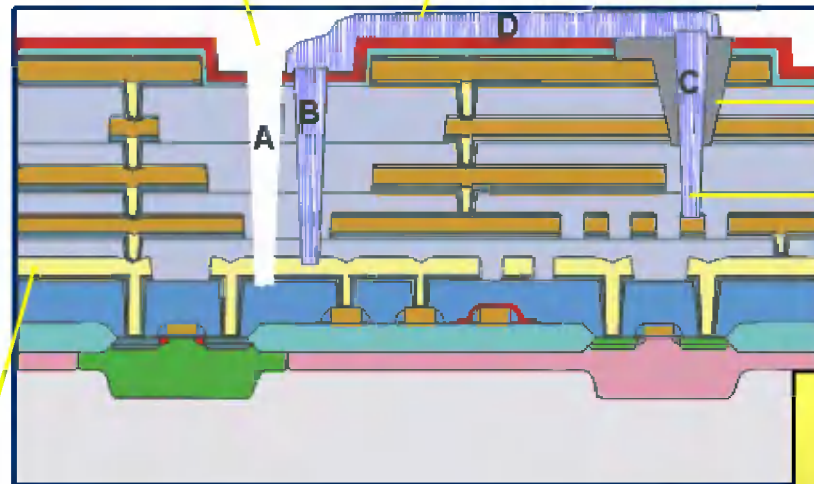
**D. Полоска:**  
Соединяет  
отверстия.

 **Вольфрам (W)**

**B. Добавление  
отверстия:**  
До металлического  
слоя 1.

**Добавленный SiO<sub>2</sub>**  
**C. Добавление  
отверстия:**  
До металлического  
слоя 2.

**Слои металла 1**



**Кремниевая  
подложка**

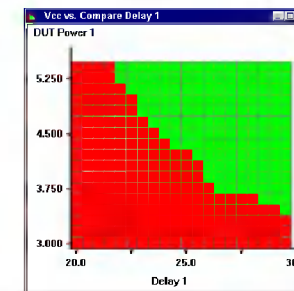
(Пять слоев металла)

- Фрезерование сфокусированным ионным пучком похоже на **электронную микроскопию**.
- Тяжелые ионы Ga **редактируют схему**, обеспечивают исправление маски.
- Часто ключевым потребителям нужен **рабочий образец** исправлений.
- Сложнее использовать с перевернутыми кристаллами; нужно травить заднюю поверхность кристалла.

## Шаг определения характеристик:

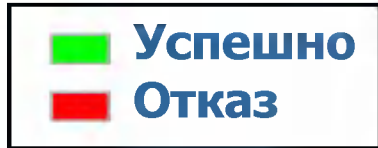
- График Shmoо
- Четыре угла процесса
- Флуктуации процесса

### График Shmoо

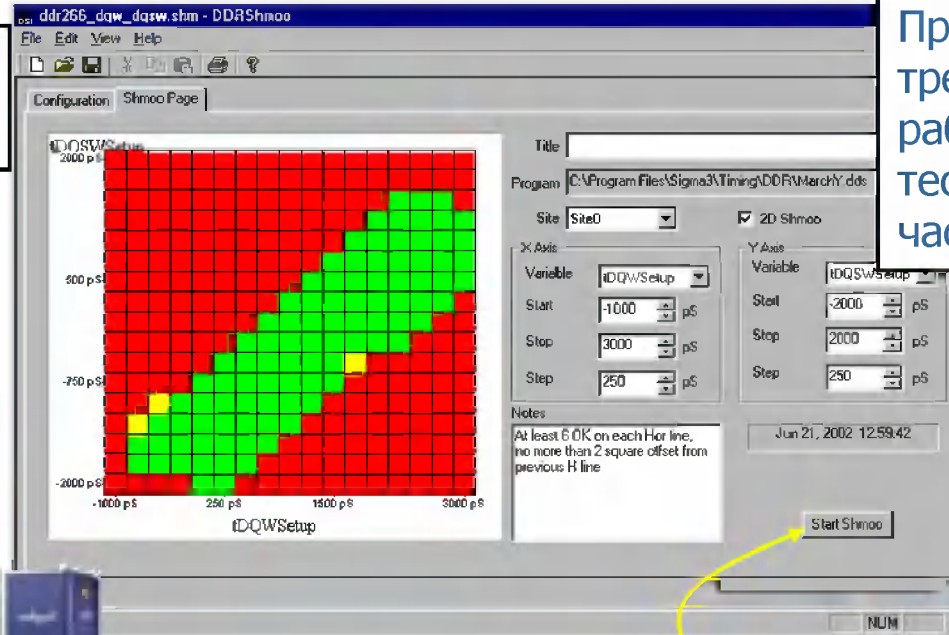


# График Shmoo

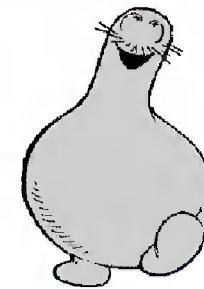
6-39



Окно автоматической тестирующей установки



**Каплеобразный:**  
При построении в трехмерном пространстве рабочая область тестируемого прибора часто подобна груше.



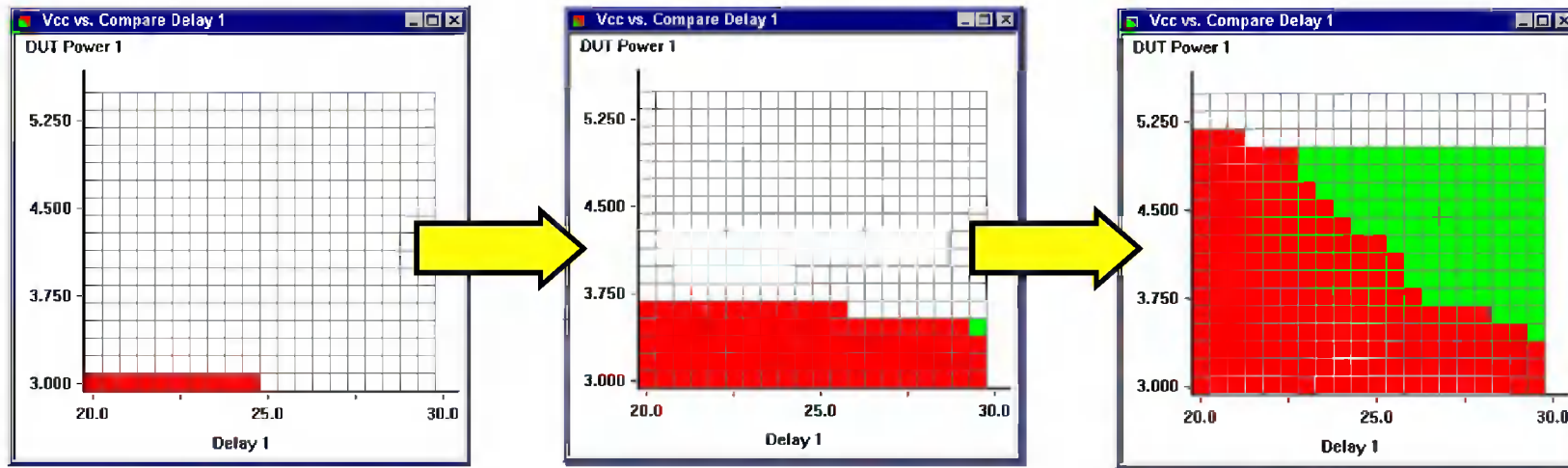
Shmoo

Автоматическая тестирующая установка с испытываемым прибором

Запуск построения графика

- График показывает **области**, в которых тестируемый прибор работает **корректно**.
- Может варьироваться **напряжение** питания, внешняя **температура** и **т.д.**

# Shmoos: Практический пример (1/2) 6-40



## Запуск построения графика:

- Питание установлено на 3 В.
- Наличие корректного выхода проверяется в моменты 20 нс, 20.5, 21, 21.5,...

## Первое прохождение теста:

- Увеличение  $V_{DD}$  по 150 мВ.
- При 3.7 В, выходные данные появляются момент времени 30 нс.

## Построение формы:

- 283 точек данных.
- Рабочая область определена.

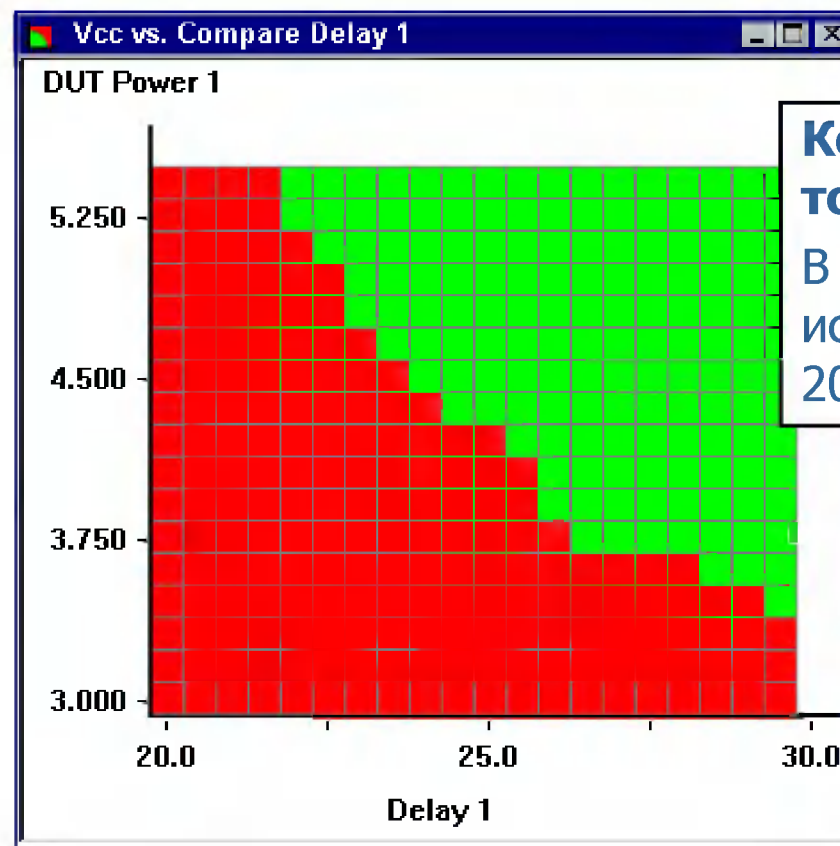
- При **низких** VDD, тестируемый прибор **не** будет работать быстро.
- Тестирование находит задержку появления **корректного выходного сигнала**.

# Shmoos: Практический пример (2/2)

# 6-41



Все 340 точек данных получены от **одного** тестируемого прибора.



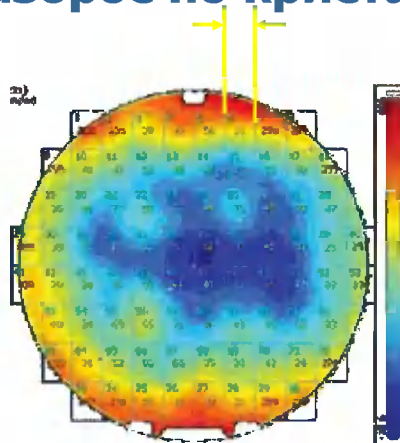
**Количество точек:**  
В этом примере используется всего 20 × 17 элементов.

- **Завершенный график показывает область, в которой тестируемый прибор работает.**
- **В условиях недостатка питания происходит постоянная деградация характеристик.**

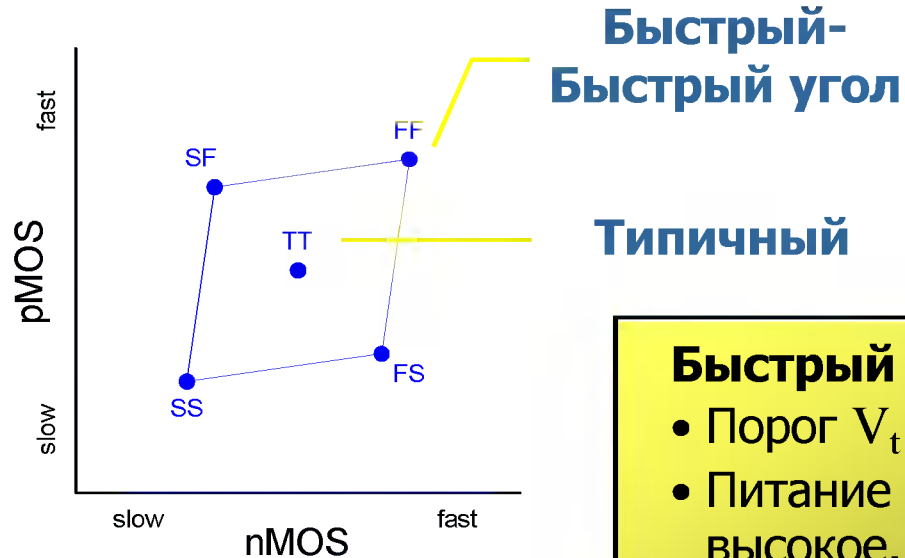
# Четыре угла процесса

6-42

OCV (on-chip variation,  
разброс по кристаллу)



Разброс по пластине



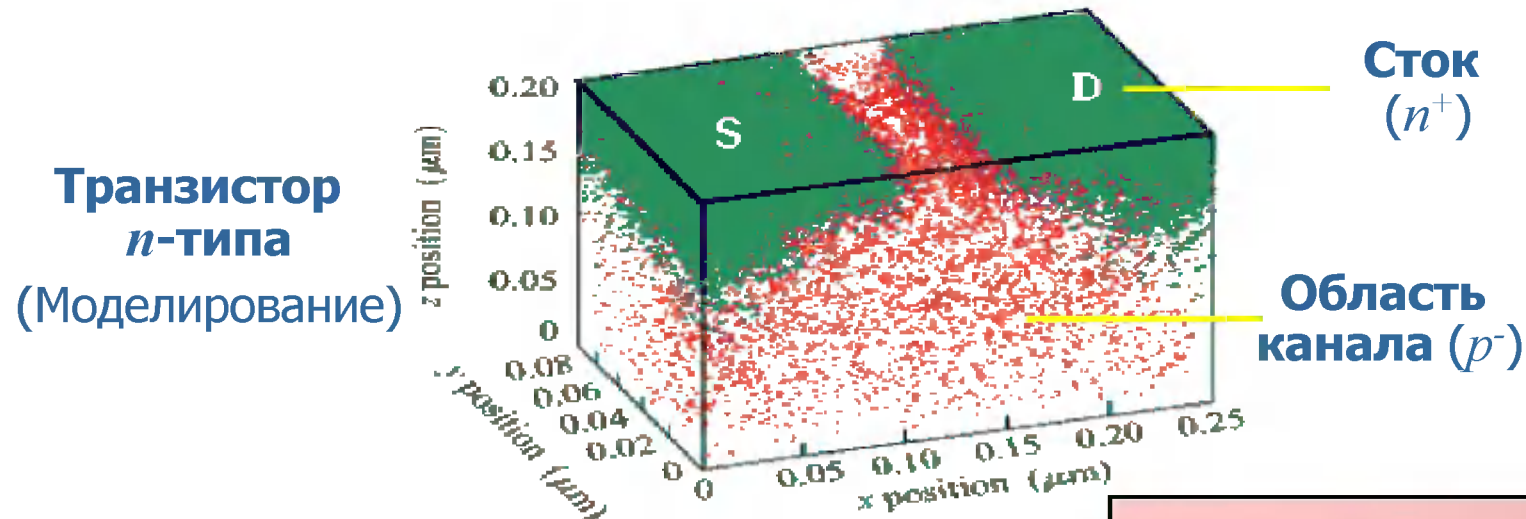
Ключевые параметры  
( $V_t$ ,  $t_{ox}$ , и т.д.)

## Быстрый угол

- Порог  $V_t$ : низкий.
- Питание  $V_{DD}$ : высокое.
- Длина  $l_{eff}$ : малая
- Оксид  $t_{ox}$ : тонкий.

- Углы процесса описывают четыре **наихудших** случая.
- Образцы с разбросом (**медленный** или **быстрый**) поступают с производства.
- Если ASIC работает во **всех** углах, то вероятно она будет работать **везде**.

# Флуктуации МОП процесса 6-43



*IBM Journal of Research  
and Development, Mar 2009*

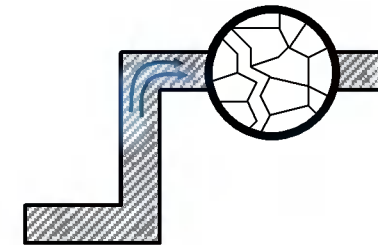
## Атомный масштаб:

Менее  $10^2$  легирующих атомов в канале. Имеет значение расположение отдельных атомов.

- При технологической норме 32 нм, случайные **флуктуации** уменьшают выход годных изделий.
- Разброс легирующей примеси в МОП канале - только один пример.
- Меньший контроль над **длиной** ( $l$ ) или **пороговым** напряжением ( $V_T$ ).

## Стадии испытаний:

- **Механизмы возможных отказов**
- **U-образная кривая**
- **Передача потребителю**
- **Подготовка первой отгрузке потребителю (FCS)**



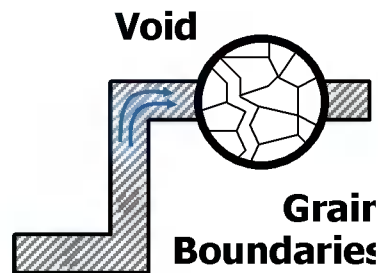
**Электромиграция**



# Механизмы возможных отказов

6-45

## Электромиграции в металлических линиях



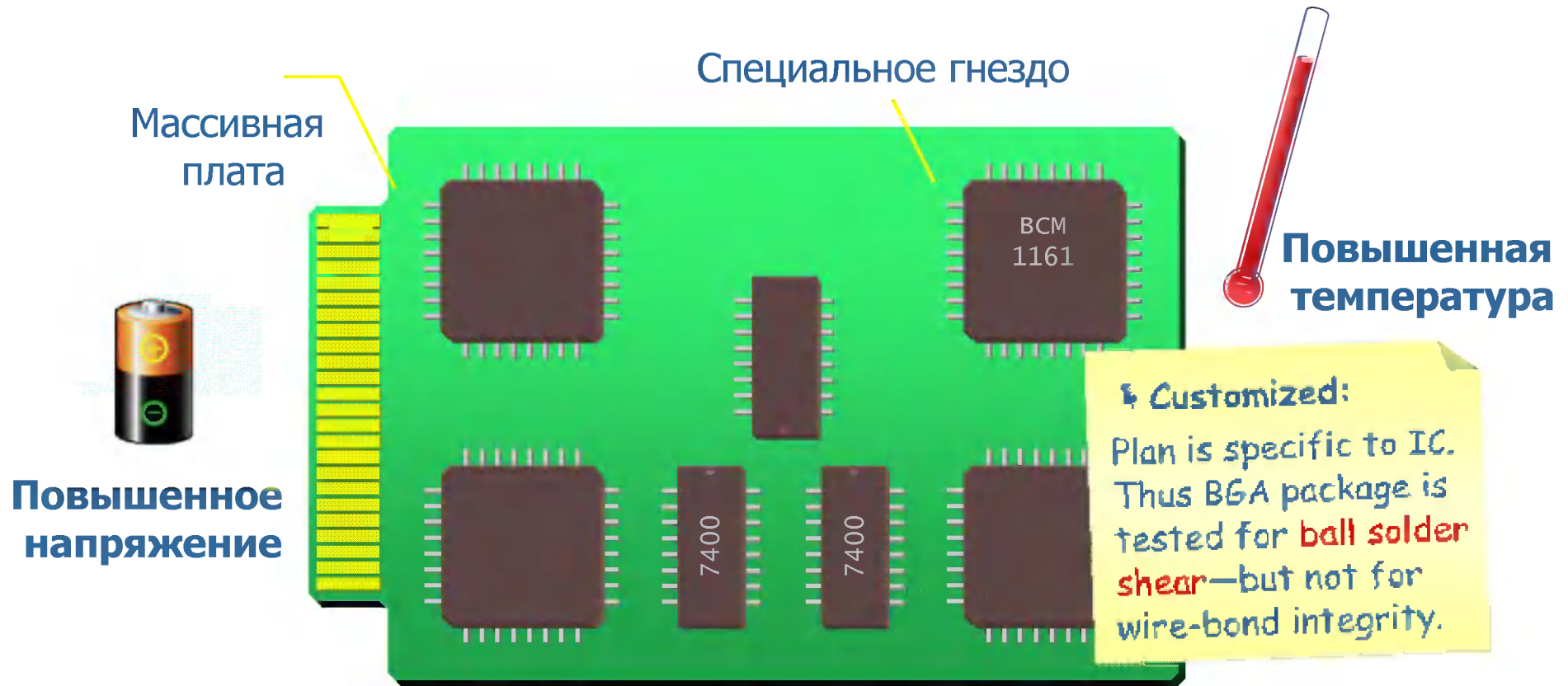
Stress	Failure Mechanism
Thermal & Voltage	Oxide breakdown over time
	Silicon substrate defects
	Electromigration
	Contact, via defects
Thermal Cycling ( $\Delta T$ )	Assembly defects
	Package cracking
	IMD cracking
I, V, T	Parasitic bipolar action (latch-up)
Voltage	Electrostatic sensitivity (ESD)
Thermal, Moisture	Contaminant-induced $V_t$ shifts
	Metal corrosion

### ASIC Qualification Plan

- State customer requirements.
- State end-usage conditions: Electrical, thermal, mechanical.
- Compliance with design rules and reliability guidelines.
- Acceptable outgoing or incoming quality level (AOQL or IQL).
- Reliability trials and failure modes.
- Verification of device functionality over specified operating conditions.
- Package verification trials.
- Production testing sign-off.

- Подобно дефектам производства, существует множество **механизмов отказов** с течением времени.
- При технологической норме 32 нм возникают новые проблемы, такие как надежность **HKMG** (диэлектрик с высоким  $k$ , металлический затвор).

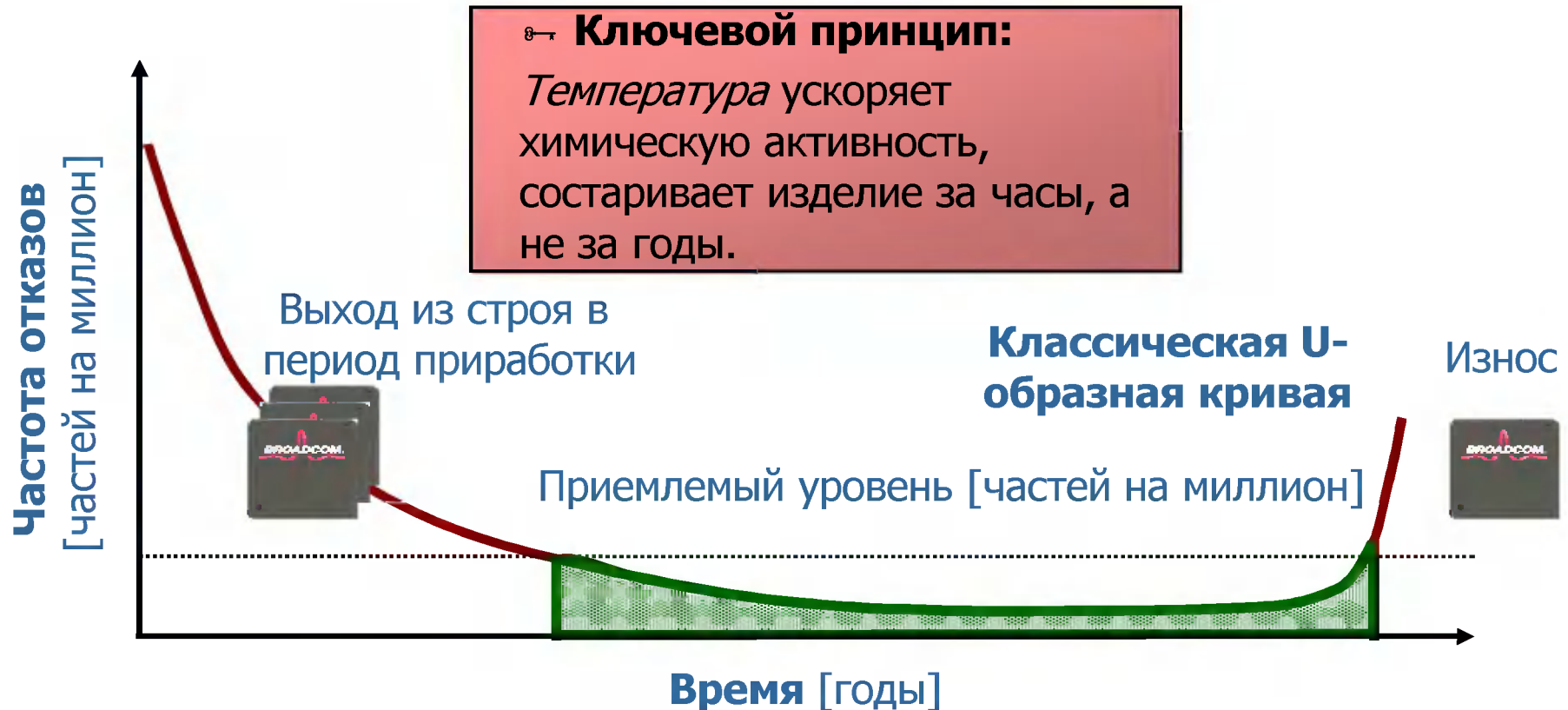
# Плата тестирования ресурса 6-46



- Процессы возникновения отказов **ускоряются** при высоких  $T_J$  или  $V_{DD}$ .
- Данные испытаний **экстраполируются** для предсказания реального времени жизни.
- Формула Блека и подобные ей позволяют оценить среднюю наработку да отказа, обусловленную электромиграцией.

# U-образная кривая

6-47



- Статистика показывает, как часто изделия отказывают **с течением времени**.
- Частота отказов **самая высокая** для очень молодых и очень старых изделий.

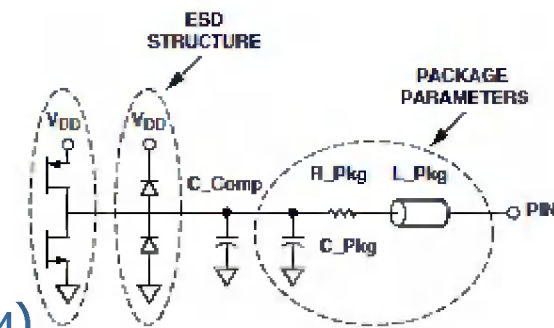
# Передача потребителю

6-48

## Элементы контрольной таблицы ASIC конечного пользователя:

- ✦ **Профили** скорость/мощность, различные условия использования.
- ✦ **Конфигурационные таблицы управления энергопотреблением.**
- ✦ **Конфигурационные таблицы внутренних мультиплексоров.**
- ✦ **Карта прерываний.** Файлы программных интерфейсов.
- ✦ **Файлы BSDL**, который описывают кольца периферийного сканирования.
- ✦ **Корпус:** распиновка BGA, площадь корпуса и т.д.
- ✦ **Модели IBIS** приемников/передатчиков ввода/вывода.

**Модель IBIS**  
(Драйвер с тремя состояниями)



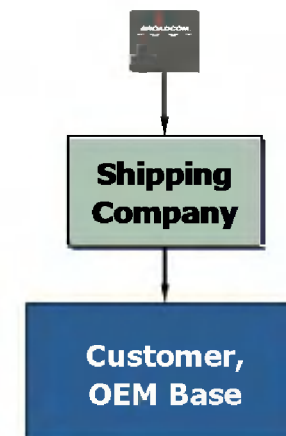
---

# Подготовка первой отгрузке потребителю (FCS)

# 6-49

## Действия перед первой отгрузкой потребителю:

- ✦ Отправка опытных образцов ключевым потребителям, для проверки и испытаний на их оборудовании.
- ✦ Анализ **возвратов от потребителей**, определение причин и назначение коллективов для их устранения.
- ✦ Планирование **серийного производства**, повышение выхода готовых изделий, усовершенствование программы тестирования.
- ✦ Гарантирование бесперебойной работы **цепочки поставок**, предсказание объемов продаж.



---

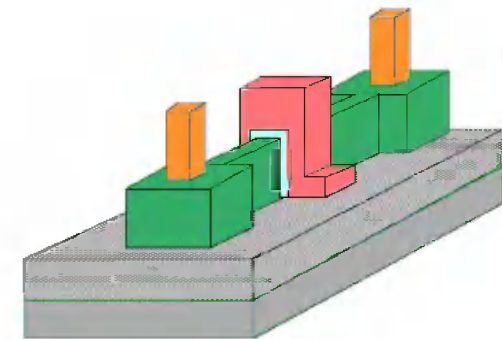
# Developing the Nanometer

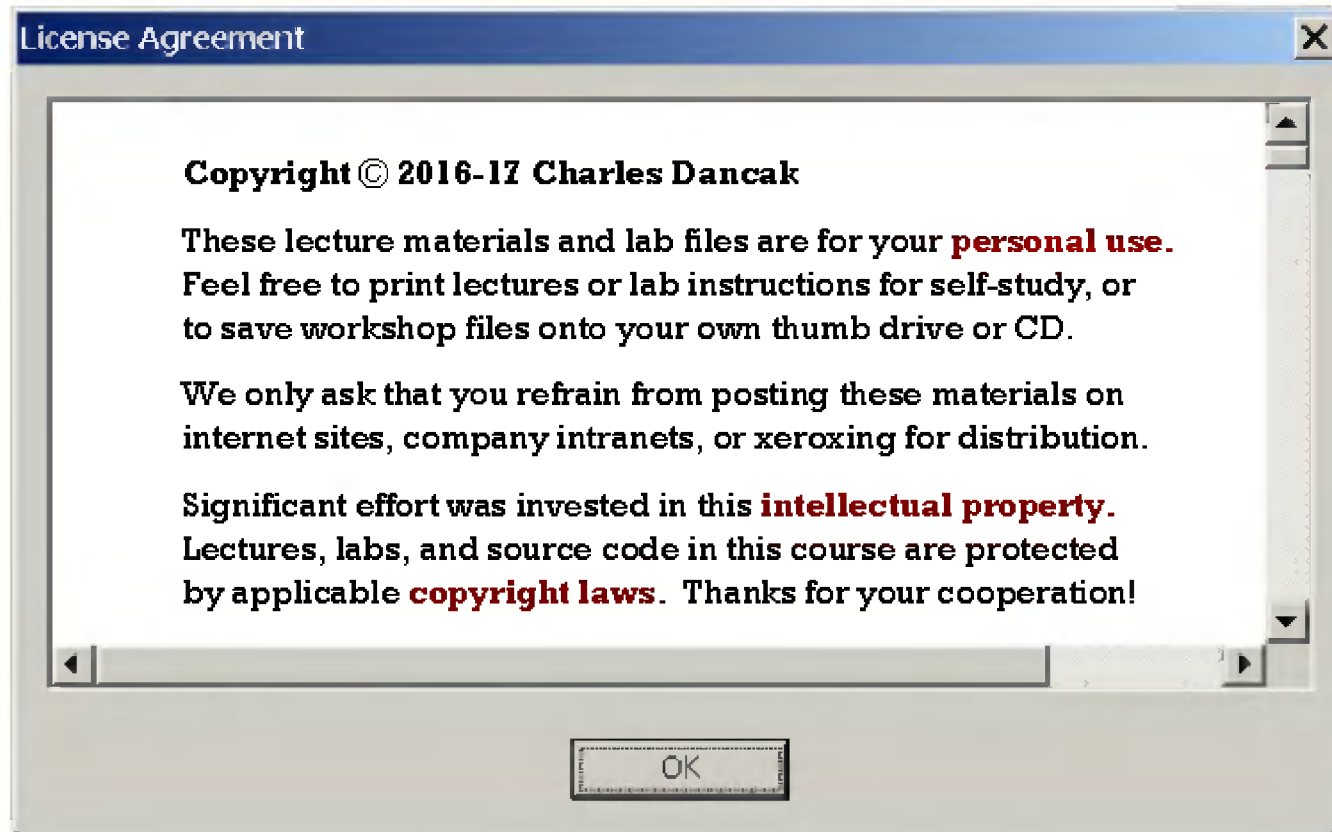
# ASIC

From Specs to Silicon



## А. За пределами планарной КМОП технологии





- **Материалы предназначены только для зарегистрированных студентов.**
- **Не следует размещать их на корпоративных или любых других сайтах.**

## **Ограничения планарной технологии КМОП:**

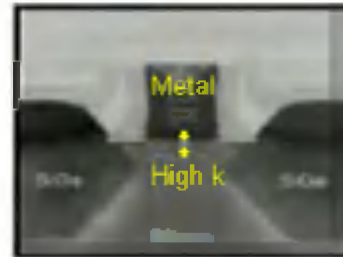
- **Конец эры масштабирования**
- **Побочные эффекты короткого канала**
- **Ultimate Transistor (предельный транзистор)**



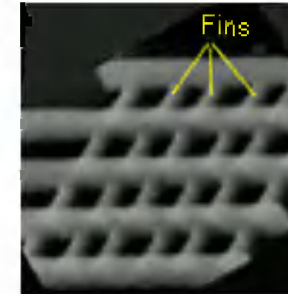
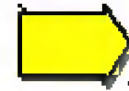
# Конец эры масштабирования А-4



**Strained Si (65 nm):**  
Stress enhances  $\mu$   
(**mobility**) of carriers.



**HKMG (32 nm):**  
Replaced thin  $\text{SiO}_2$   
with **thicker**  $\text{HfO}_2$ .  
Same effective  $C_{\text{ox}}$ .

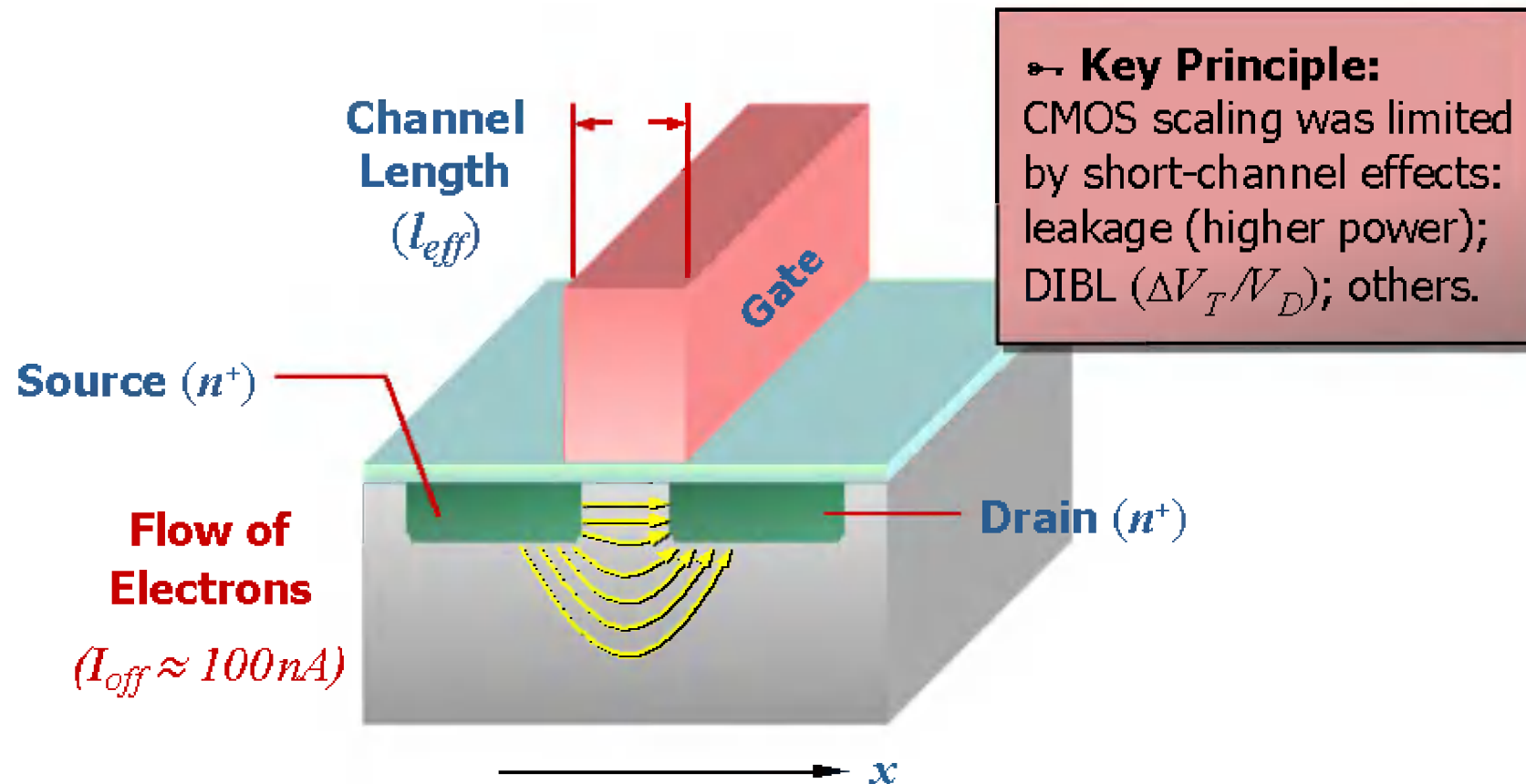


**FinFET (22 nm):**  
Band-aid fixes no longer  
adequate. Quantum leap  
to **nonplanar** technology.

- В течении десятилетий в микроэлектронике все размеры **уменьшались** в  $\sqrt{2}$  раз.
- Возможности простого масштабирования были исчерпаны примерно при технологической норме 90 нм.
- Для перехода на 65 нм была применена новая физическая идея (повышение  $\mu$ ): Канал из **напряженного кремния**.
- Для перехода на 32 нм были применены новые материалы (тончайший подзатворный оксид): **HKMG (диэлектрик с высоким k, металлический затвор)**.

# Боковые эффекты короткого канала

# A-5

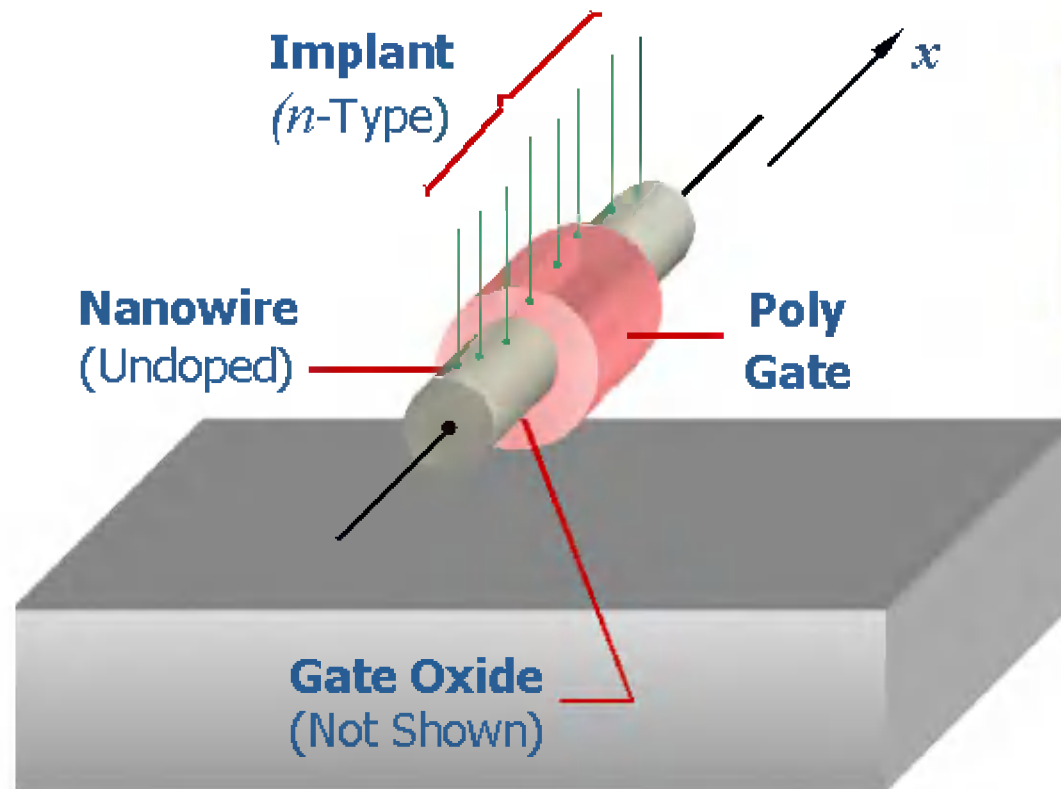


- Плохой побочный эффект: Большой ток утечки при закрытом транзисторе.
- Затвор не может управлять током утечки по путям, проходящим далеко от поликремния.
- Простое уменьшение толщины подзатворного оксида более не решает эту проблемы.

# Ultimate Transistor (предельный транзистор)

# A-6

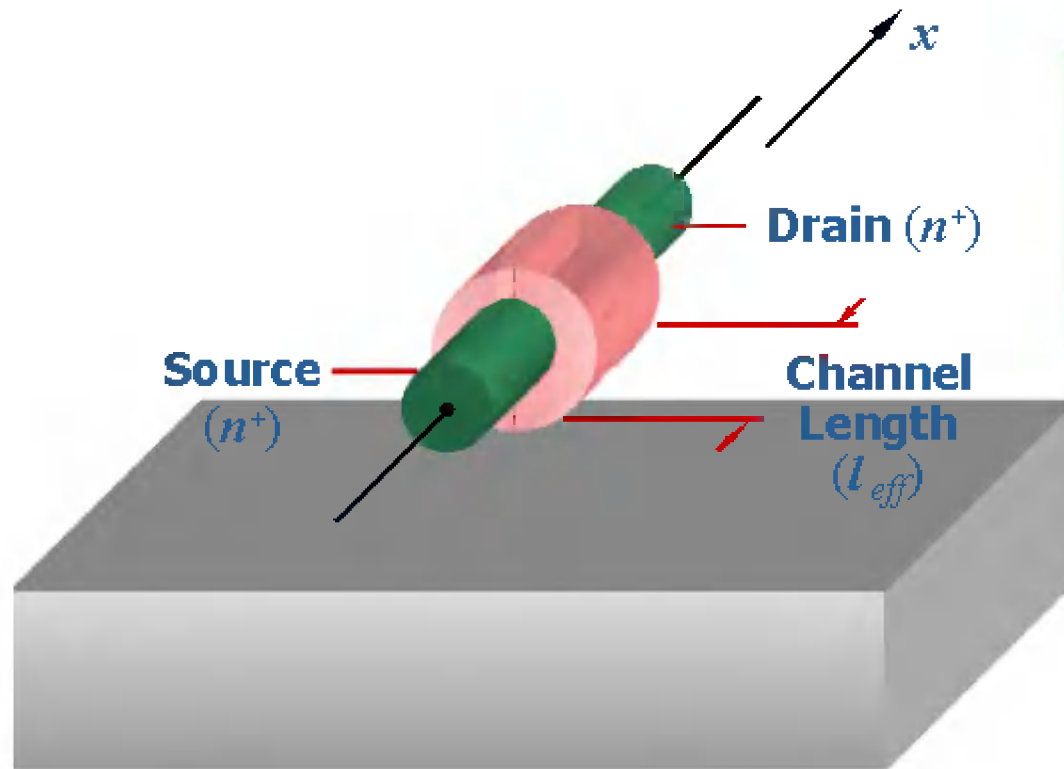
(1/2)



**Still Experimental:**  
Nanowires chemically synthesized in a lab—without lithography.

- Кремниевая **нанопроволока** изготовлена из нелегированного полупроводника.
- Проволока изолирована подзатворным оксидом с высоким  $\kappa$  (не видим).
- Опоясывающий затвор окружает проволоку—подобно первому обороту шланга.

# Ultimate Transistor (предельный транзистор) A-7 (2/2)



**Low Leakage:**  
A wrap-around gate  
has more control of  
channel:  $I_{off} \approx 0 \text{ nA}$ .

Inspired By:  
Paolo Gargani, *Intel* Fellow,  
ITRS Chairman

- Имплантация формирует  $n^+$  **исток/сток** на открытых концах проволоки.
- Часть между ними маскируется поликремнием, остается нелегированный канал.
- Сохранятся поведение длинного канала, даже при уменьшении  $l_{eff}$ .

## **Одиночный $n$ МОП плавниковый транзистор (FinFET):**

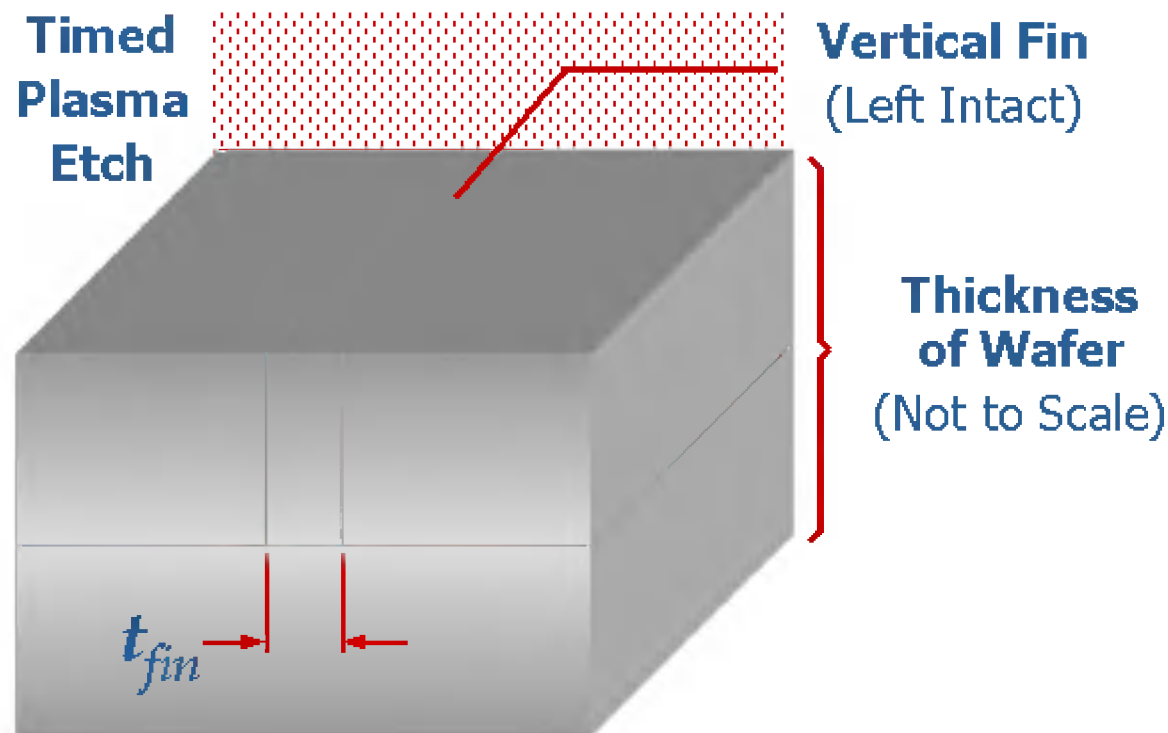
- **Травление вертикального плавника**
- **Добавление стека затвора**
- **Легирование истока и стока**
- **Критические размеры плавникового транзистора**

# Одиночный $n$ МОП плавниковый транзистор (FinFET) (1/4)

A-9



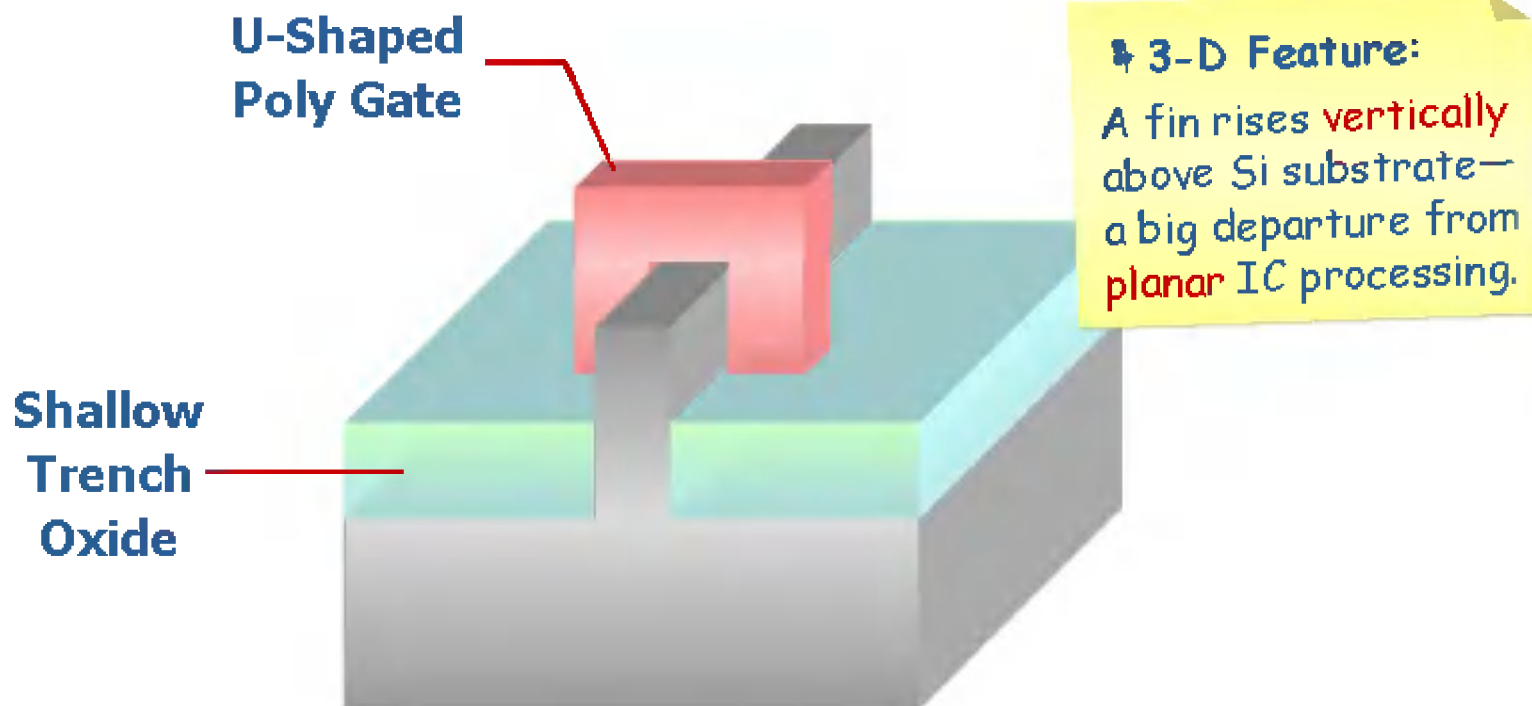
Chenming Hu,  
UC Berkeley  
(2011)



- **Спланированное по времени** травление удаляет на заданную глубину незащищенный кремний.
- Формируется **вертикальный «плавник»** нелегированного кремния с крутыми стенками.
- Тонкий **«плавник»** играет роль **нанопроволки**.
- **«Плавник»** должен быть **тонким**—иначе затвор не сможет контролировать глубокие пути токов утечек.

# Одиночный *n*МОП плавниковый транзистор (FinFET) (2/4)

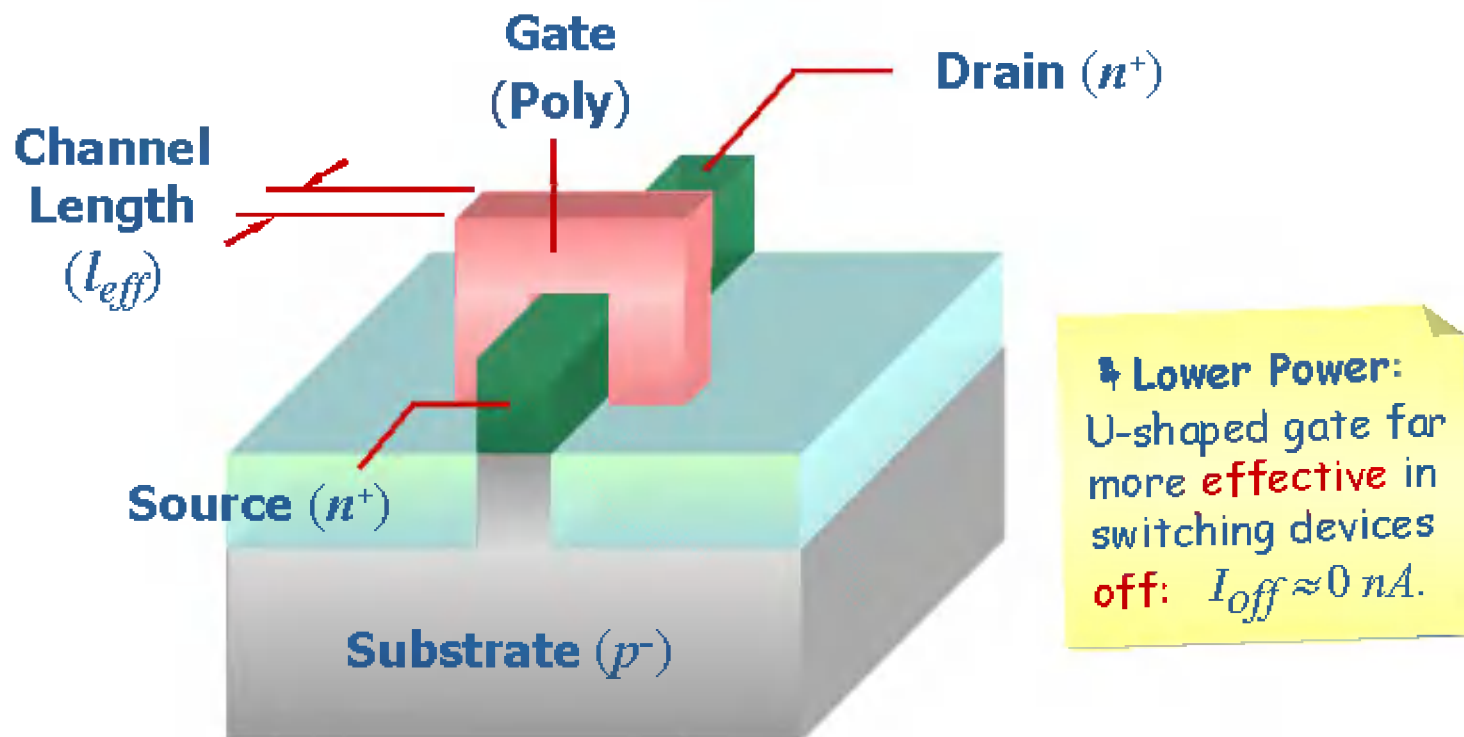
# A-10



- Выращивается канавка **изолирующего оксида**, «плавник» выступает над ней.
- «Плавник» покрывается тонким **подзатворным оксидом** с высоким  $k$  (не показан).
- Поверх «плавника» осаждаются поликремний, затем из него формируется **затвор**.
- Альтернативный стек затвора **HKMG** (диэлектрик с высоким  $k$ , металлический затвор): Оксид  $\text{HfO}_2$ , затвор  $\text{TiN}$ .

# Одиночный $n$ МОП плавниковый транзистор (FinFET) (3/4)

# A-11

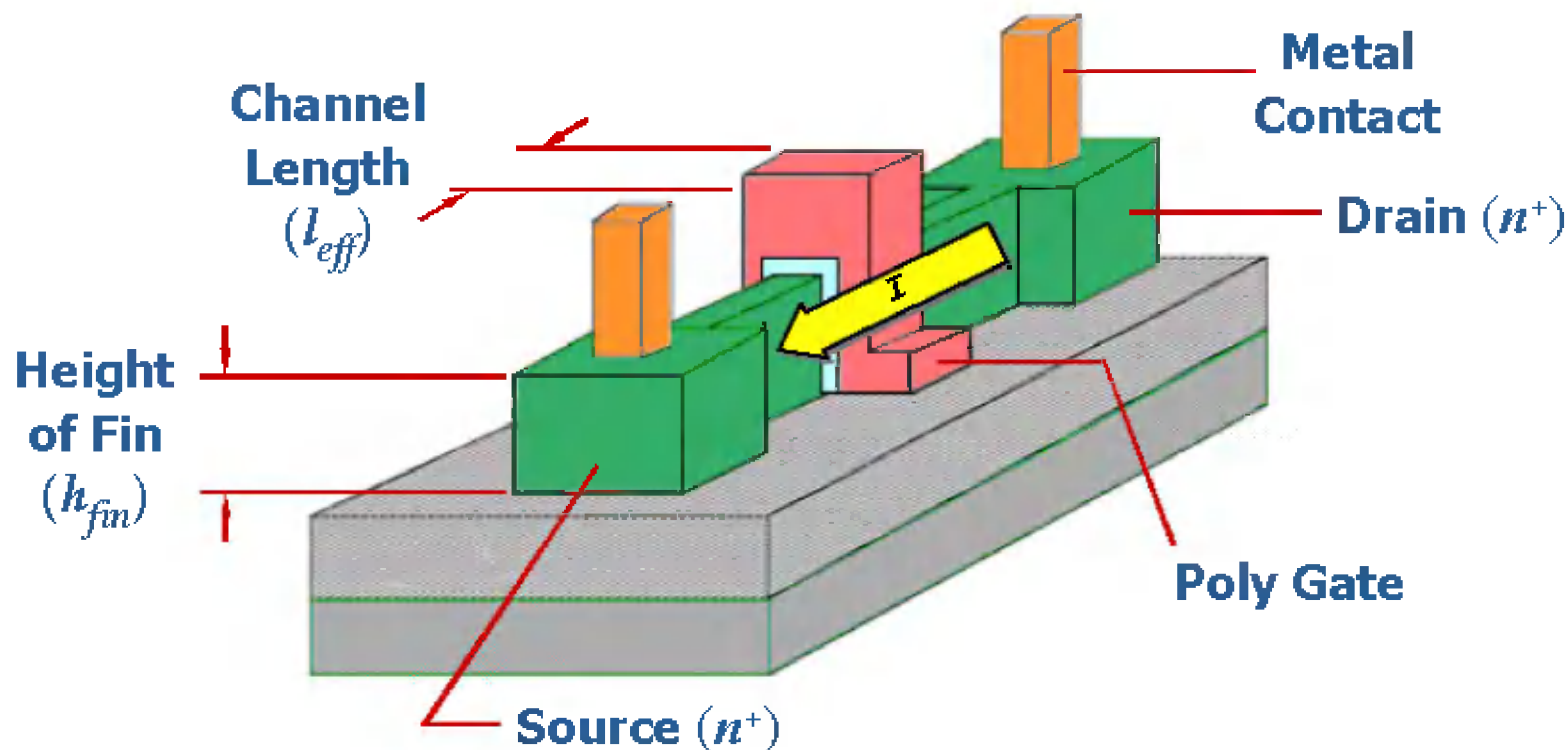


- **Исток** и **сток** формируются путем **имплантации** на незащищенных концах «плавника».
- Область оставшаяся под поликремниевым затвором является **нелегированным каналом**.
- Таким образом, транзистор определяется пересечением «плавника» с поликремниевой областью.
- Изготовление комплементарного  **$p$  МОП** плавникового транзистора аналогично.



# Одиночный $n$ МОП плавниковый транзистор (FinFET) (4/4)

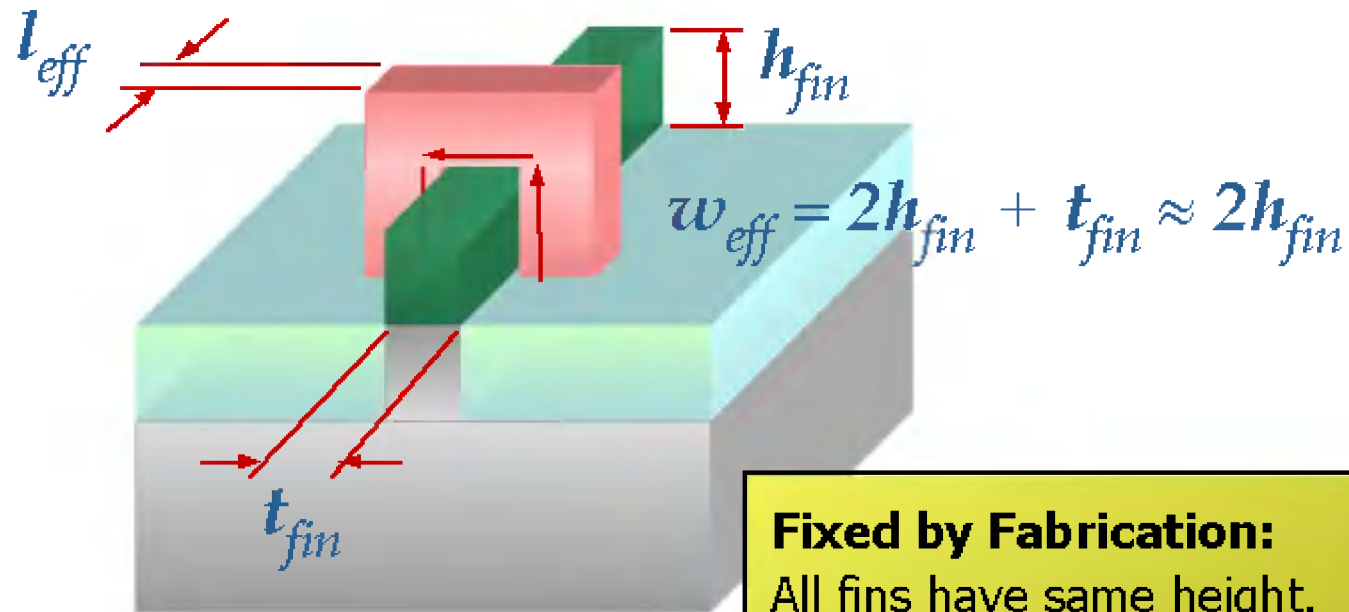
# A-12



- Как обычно, добавляются контакты для истока, стока и затвора.
- Зеленые концы «плавника» ведут себя подобно **нанопроволоке**.
- Свойства полученного плавникового транзистора приближается к свойствам идеального транзистора с длинным каналом.
- Структура должна хорошо **масштабироваться** от технологической нормы 22 нм к 14 нм и 10 нм.



SEM  
(20 nm)



**Fixed by Fabrication:**  
All fins have same height.  
The non-planar dimension  $h_{fin}$  is a *process parameter*.

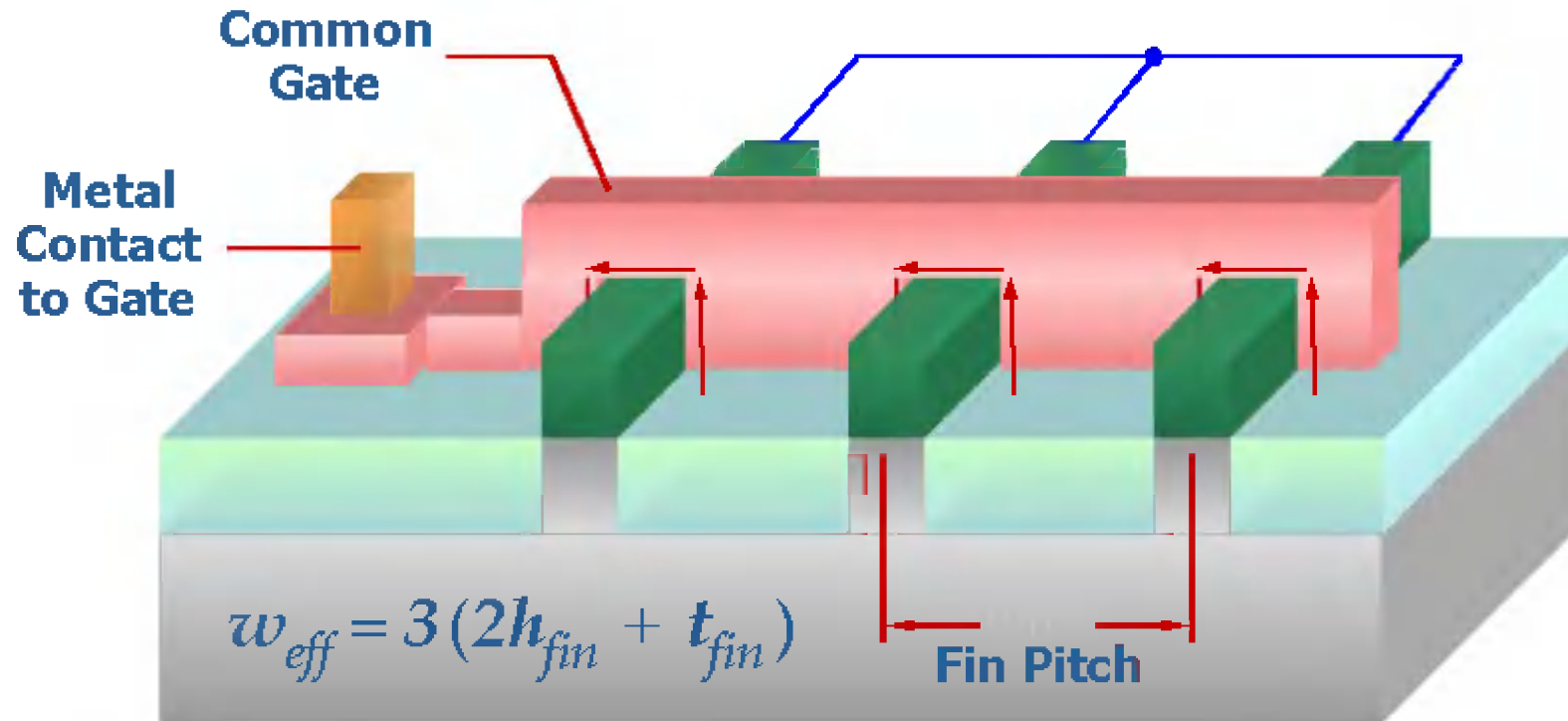
- **Ширина** канала плавникового транзистора больше не определяется 2-D геометрией.
- Его свойства зависят от **высоты** «плавника», которая измеряется в вертикальном направлении.
- Ширину нельзя больше легко изменять для увеличения **силы драйвера**.

## **Плавниковый транзистор с несколькими «плавниками»:**

- **Типичный транзистор с несколькими «плавниками»**
- **Подстройка силы драйвера**
- **Общий исток и сток**

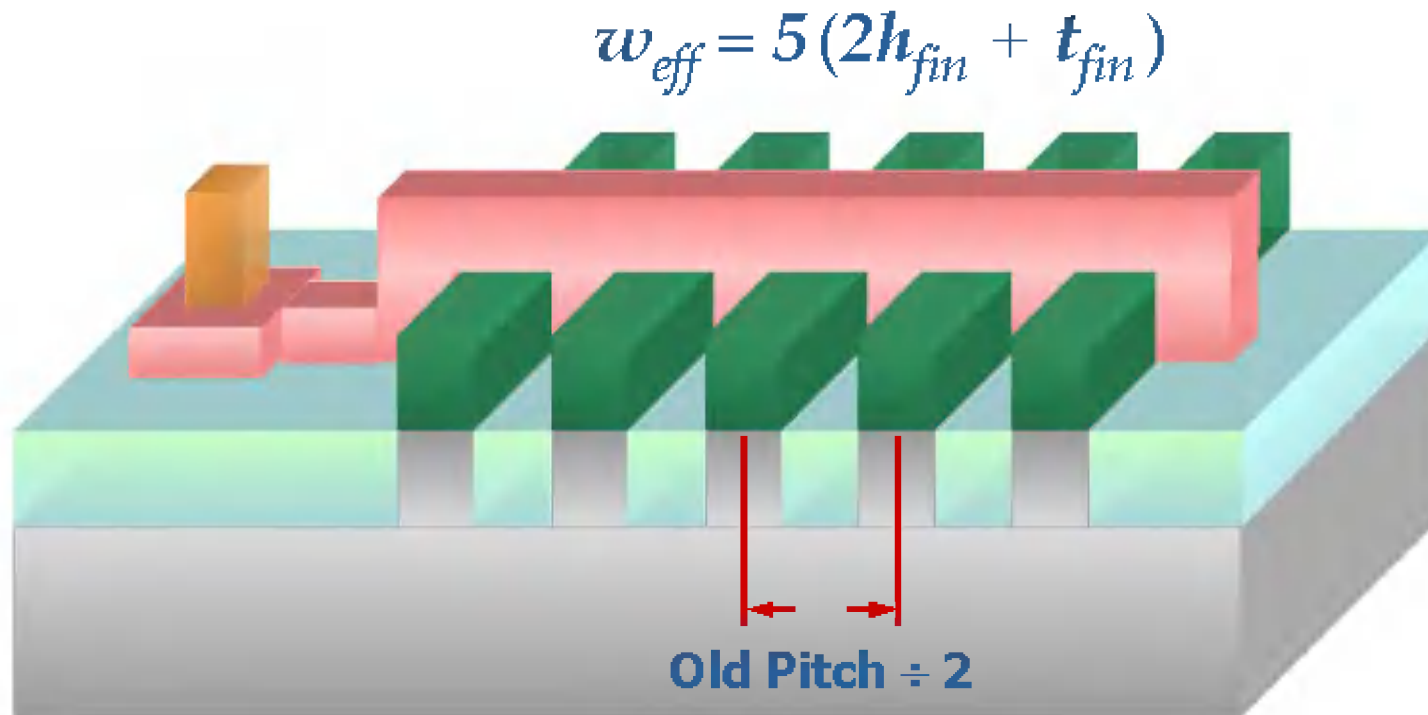
# Типичный транзистор с несколькими «плавниками»

A-15



- Чтобы подстроить силу драйвера, **несколько «плавников»** соединяются параллельно.
- Три параллельно соединенные транзисторы **утраивают** эффективную ширину канала.
- Общая площадь большой ячейки может быть уменьшена минимизацией **шага плавник-плавник**.

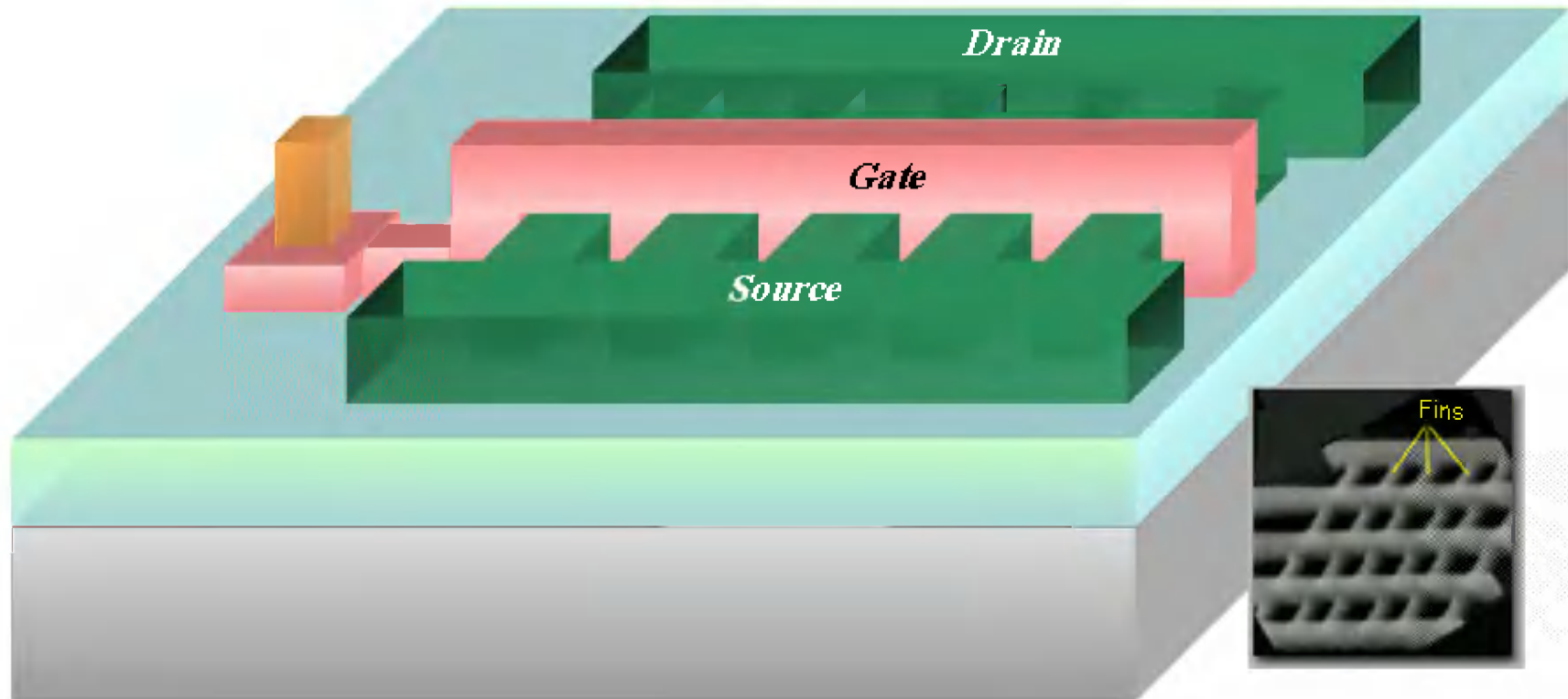
# Увеличение силы драйвера A-16



- **Уменьшение** шага плавников позволяет разместить мощный драйвер на той же самой площади.
- Но ширина **параметризована**—в отличие от планарной технологии.
- Изготовление близко расположенных «плавников» требует двойного литографирования.

# Общий исток и сток

A-17



- **Исток/сток** травятся подобно «плавнику», нет необходимости в высоком разрешении.
- «Плавники» соединяются с истоком/стоком на уровне **базового слоя**, проводники не нужны.
- Полученный многоплавниковый прибор является **типичным** плавниковым транзистором (finFET).

## **Особенности технологии плавниковых транзисторов:**

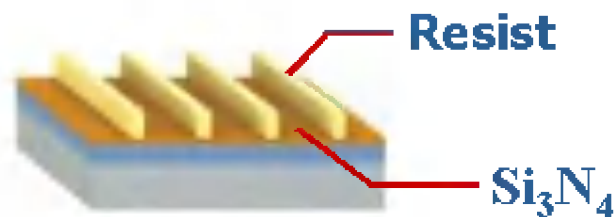
- **Технология двойной литографии**
- **Новый правила проектирования (DRC) для плавниковых транзисторов**
- **Аналоговые и ВЧ эксплуатационные свойства**



# Двойная литография

A-19

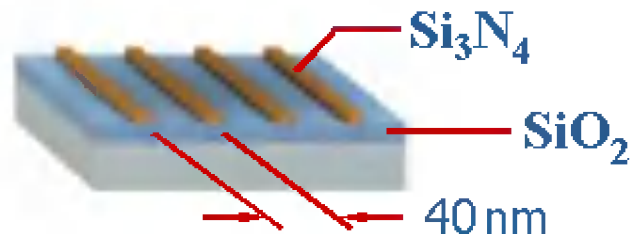
① **First mask transfers line pattern to resist:**



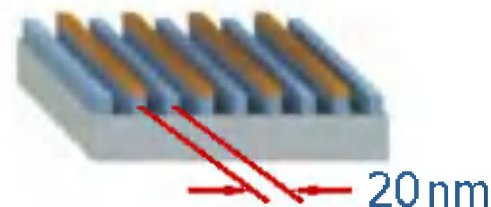
③ **Second mask transfers its lines to new resist:**



② **Nitride is etched, down to oxide:**



④ **Oxide is etched, down to silicon:**



⑤ **Lines in oxide can now mask 9 20-nm fins.**

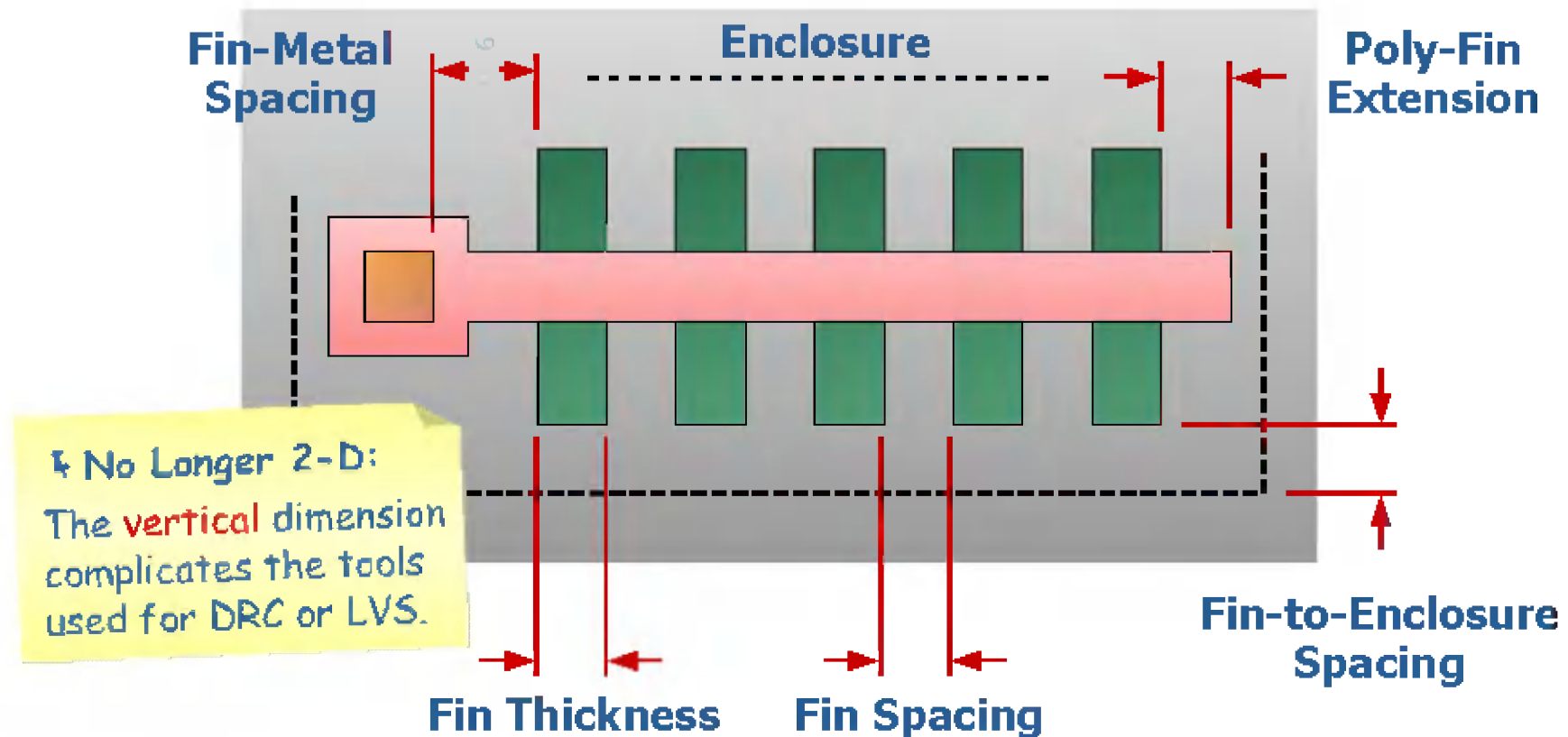


- Обычно для изготовления одного КМОП слоя использовалась только **одна маска**.
- Двойная литография с использованием **двух масок** позволяет удвоить разрешение.
- Требуется **очень точное** наложение—в пределах нескольких нанометров.

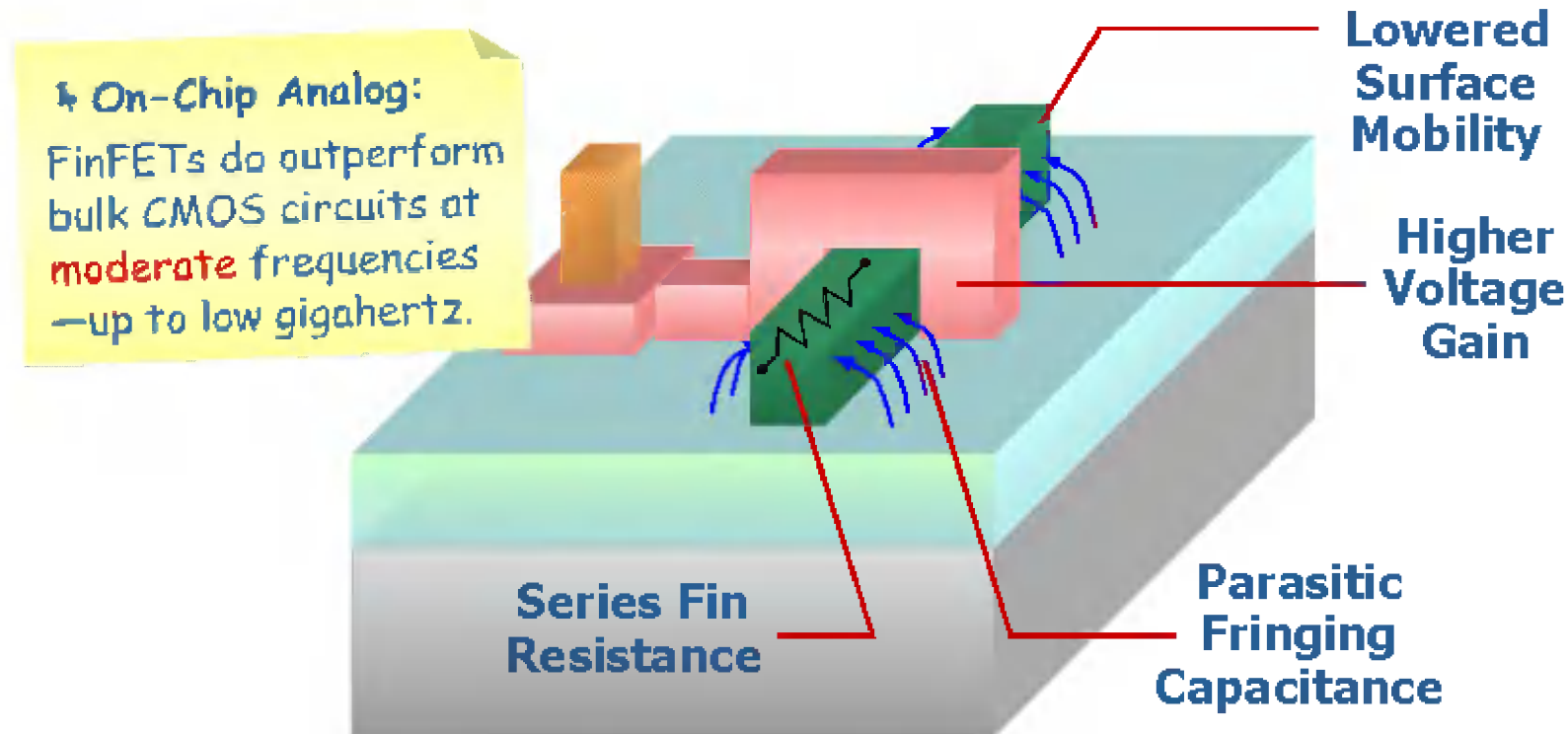


# Новые правила проектирования (DRC) для плавниковых транзисторов

# A-20

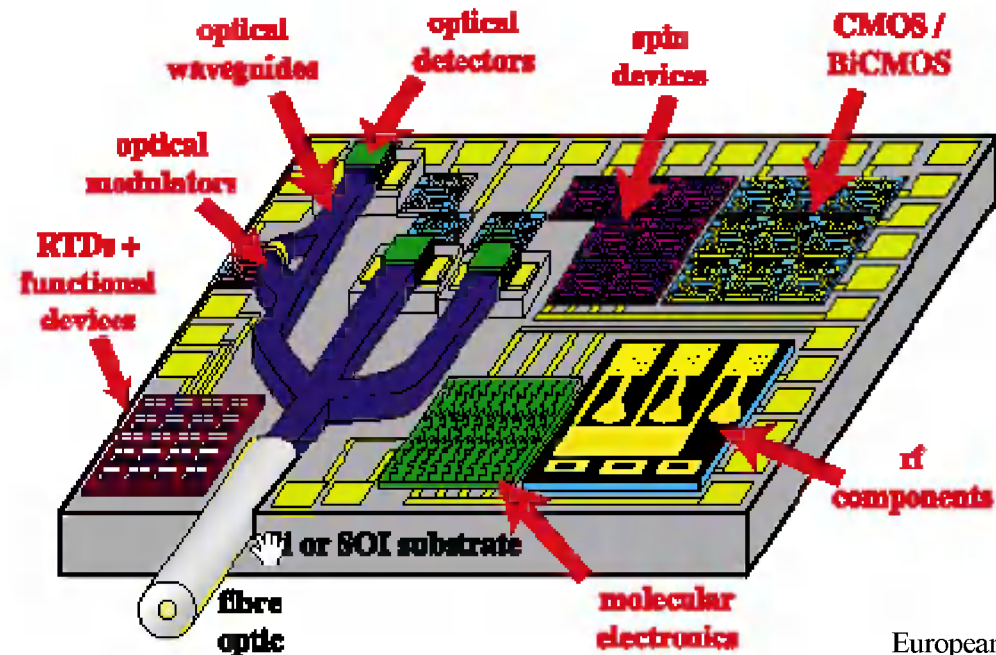


- Обычно для изготовления одного КМОП слоя использовалась только **одна маска**.
- Двойная литография с использованием **двух масок** позволяет удвоить разрешение.
- Требуется **очень точное** наложение—в пределах нескольких нанометров.



- Плавниковые транзисторы имеют большее **собственное усиление** из-за лучшего управления затвором.
- Свойства на высоких частотах хуже из-за сопротивления «плавников» и краевых эффектов.
- Если канал **нелегирован**, малая изменчивость улучшает согласование.
- Точные пассивные **резисторы** легко изготовить из легированных «плавников».

# За пределами технологии КМОП микросхем **A-22**



Гордон Мур,  
Intel ('65)

European Nanotechnology Roadmap

- Ожидается, что закон Мура будет выполняться еще **два десятилетия**.
- Потребуется новые **приборы** и **соединения**.
- Но они будут полностью использовать базу **КМОП технологии**.