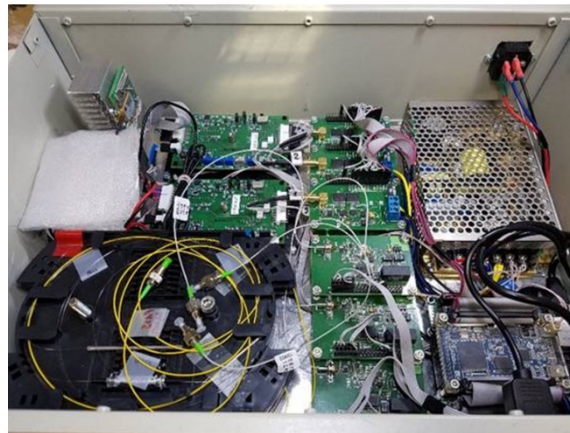
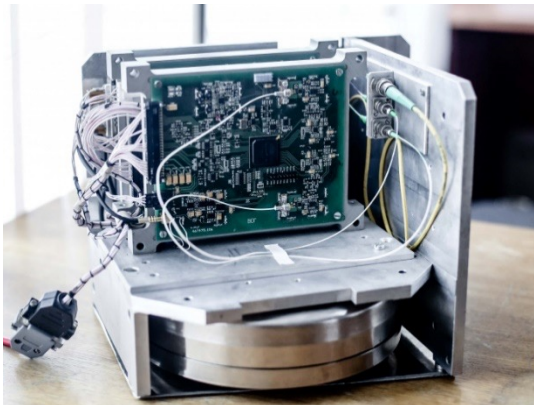


# Топ 10 ошибок FPGA-шников - байки, кейсы, занимательные истории

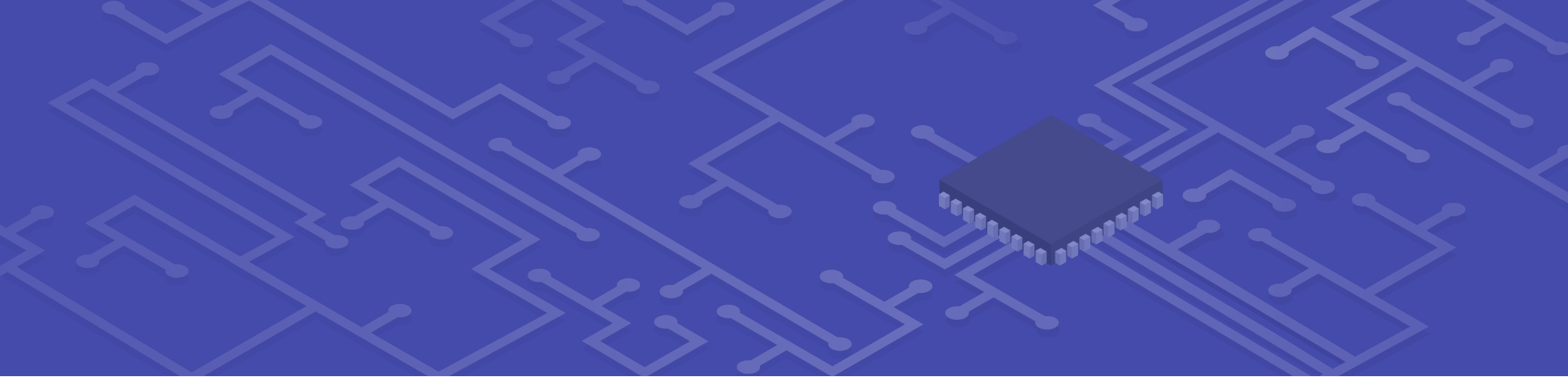
## Университет ИТМО

- Лаборатория Моделирования и Программирования (МиП)
  - Исследования
  - Разработка
  - Обучение
    - Инженерия Цифровых Систем (Магистратура)
    - Центр Программируемой Электроники (курсы ДПО)



# Структура доклада

1. Недостаточное внимание к архитектуре
  2. Недостаточное внимание к топологии проекта (этап проектирования платы)
  3. Опять о метастабильности
  4. Учёт аналоговых эффектов
  5. НЕПРАВИЛЬНОЕ использование констрейнтов
  6. НЕПРАВИЛЬНОЕ восприятие констрейнтов
  7. Внешние интерфейсы - констрейнты
  8. ПЛИС МОЖЕТ ВСЁ
  9. Учёт архитектуры тактовых сигналов
  10. Разное...
-



# ВВЕДЕНИЕ

# Доклад - набор полезных баек, объединённых общей идеей

- Было сложно выбрать топ-10
- Возможных ошибок полная авоська



- Выбрал то, что пришло в голову, если формат понравится аудитории и будет полезен - продолжим
- Некоторые отсылки к предыдущему выступлению на FPGA-systems для логической связки

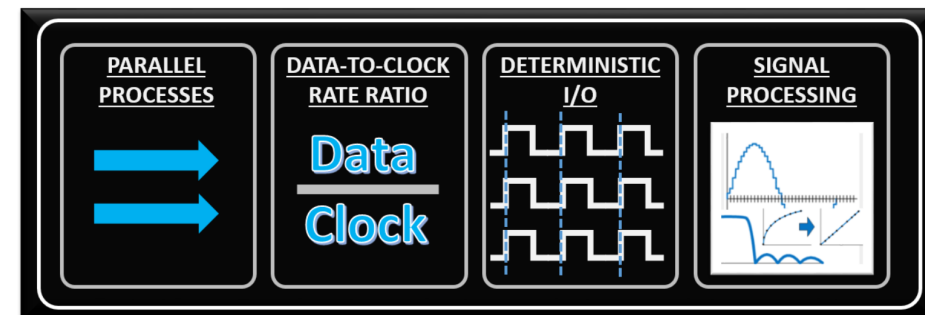
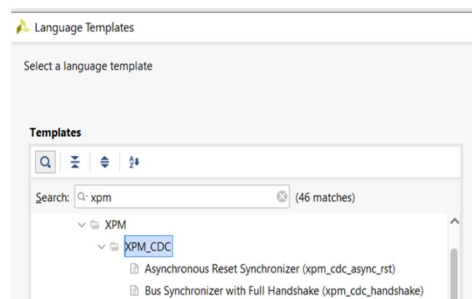


# Альтернативные подборочки

1. Having Asynchronous Logic
2. Not Using Primitives Provided on Device
3. Under-utilizing DSP Slices
4. Not Checking for Inferred Latches in your Design
5. Not Running DRC Checks at Each Stage
6. Not following Proper Naming and Coding Conventions
7. Unnecessary Usage if Reset Signal
8. Completely utilize the all available resources
9. Using Block Designs for Large Designs
10. Not Checking Log all Signals Option in Simulation Settings

1. Thinking that you're writing software
2. Assuming the bulk of your algorithm is the bulk of development time
3. Not paying enough attention to timing stability
4. Not worrying about cyber security
5. Not understanding the risks for safety-related applications
6. Not having at least one driving Algorithmic reason to use an FPGA
7. Not doing design reviews
8. Jumping right into coding
9. Trying to use all of the FPGA!
10. Not planning for Enough Bugs

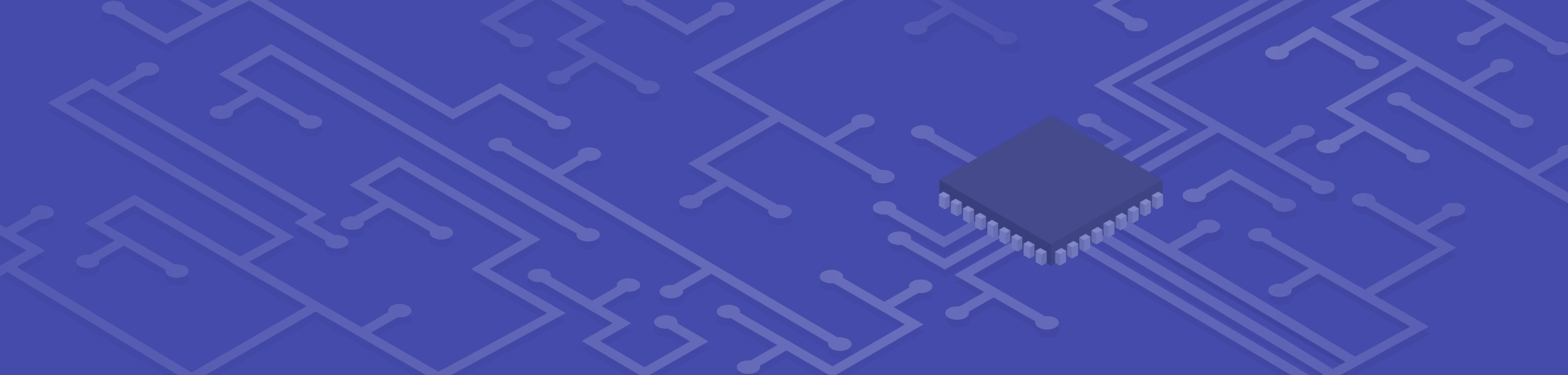
- Adding i\_ for all inputs
- Adding o\_ for all outputs
- Adding s\_ for all signals
- Adding c\_ for all constants
- Adding g\_ for all generic





Часть первая

# НЕДОСТАТОЧНОЕ ВНИМАНИЕ К АРХИТЕКТУРЕ



ИСТОРИЯ ПРО БАГУ В  
М/СХ GOWIN





# Бага - spi-slave, м/сх - GW2A

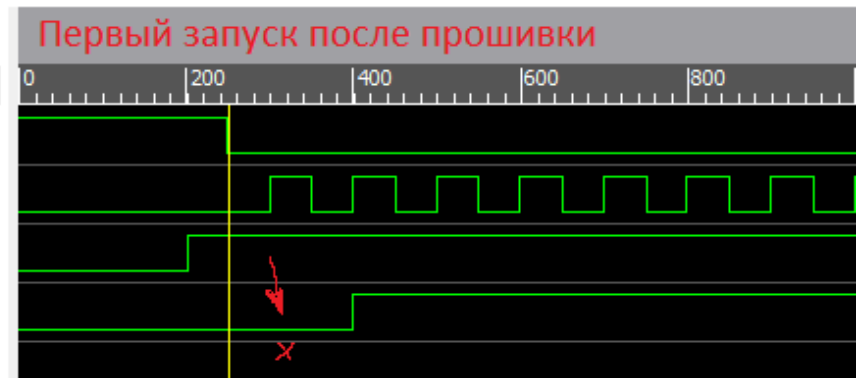
## Описание проблемы:

После перезагрузки (power on/off) платы (или после перепрошивки ПЛИС) **данные**, которые приходятся на **первый** защелкивающий **фронт** триггера, **не поступают на его выход**. Все **последующие** защелкивания происходят **корректно**

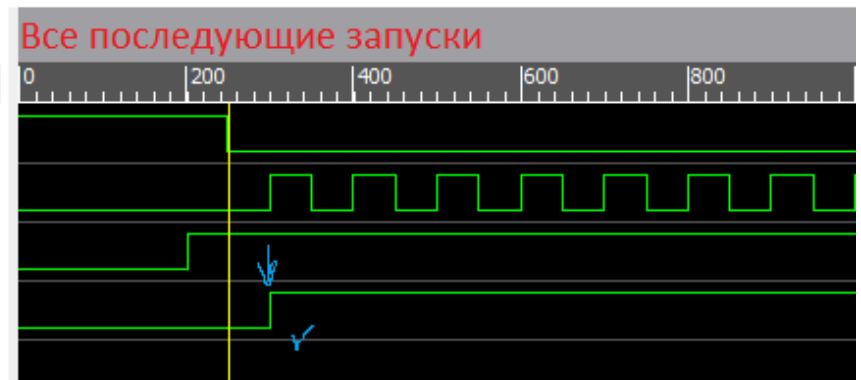
```
begin
  -- тестовый триггер
  process (RST, CLK)
  begin
    if RST = '1' then
      DOUT <= '0';
    elsif rising_edge(CLK) then
      DOUT <= DIN;
    end if;
  end process;

  --clk_la <= CLK; --!!!!!!
  dout_la <= DOUT;
end architecture;
```

Name	Value
test_in_rst	0
test_in_clk	0
test_in_data	1
LED	0



Name	Value
test_in_rst	0
test_in_clk	0
test_in_data	1
LED	0



# UPDATE!

Бага протестирована 21.05.24 на  
Gowin\_Ver 1.9.9.03\_x64 - проблема осталась.

Из хорошего, на новом семействе Arora V такой  
проблемы нет (проверено на плате от Sipeed).



Часть вторая

# НЕДОСТАТОЧНОЕ ВНИМАНИЕ К ТОПОЛОГИИ ПРОЕКТА (ЭТАП ПРОЕКТИРОВАНИЯ ПЛАТЫ)

# Далее слайды от дяди Серёжи



# ПЛИС и недокументированные возможности

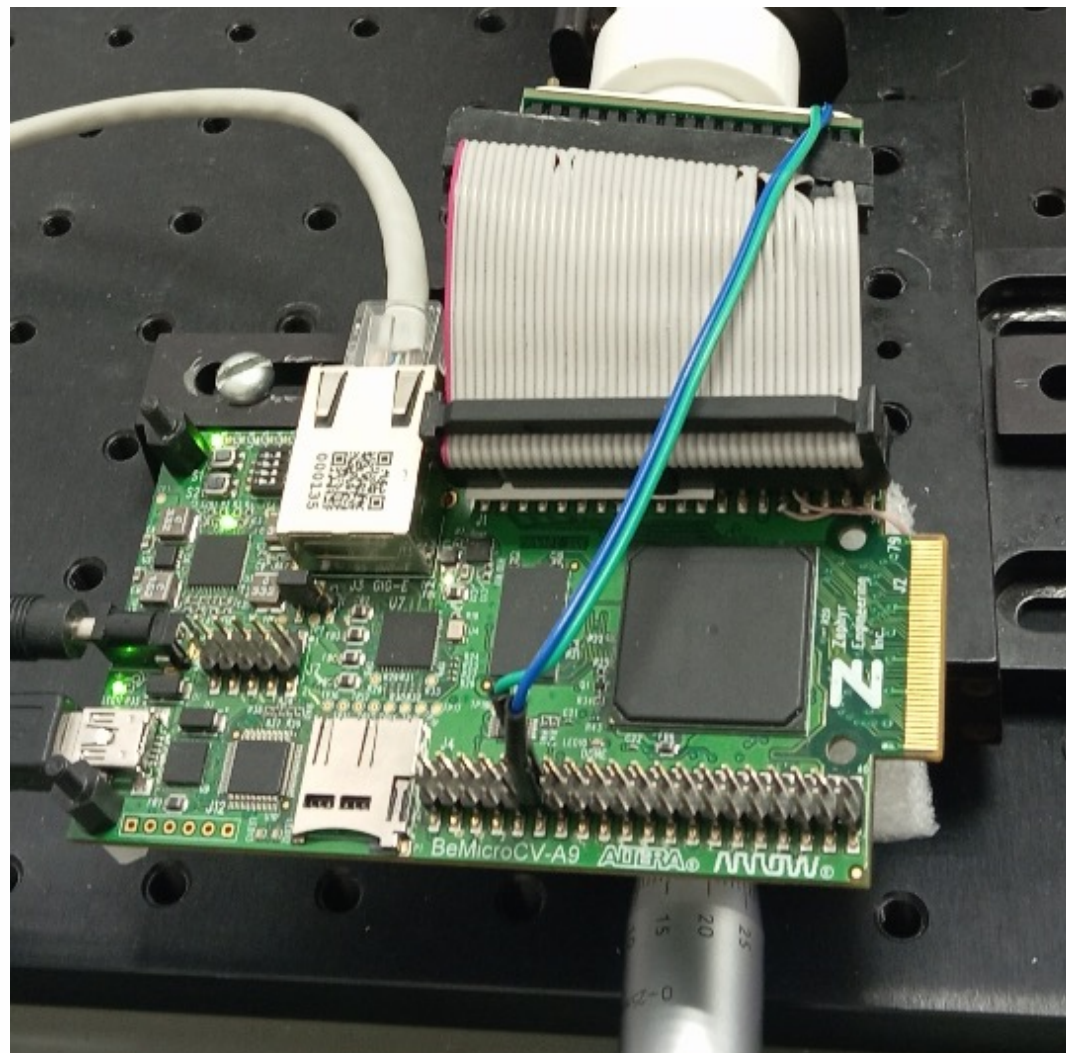


Подключили к плате VeriMicro CV A9 кастомный модуль со скоростной фотоматрицей

При передаче данных по LVDS падает лок на PLL, ситуация улучшается, если плату с ПЛИС принудительно охлаждать

Проблему решает переключение банка, куда приходят LVDS-сигналы с матрицы, на питание 3.3 В вместо 2.5 В по умолчанию

Документация на ПЛИС запрещает питать банки с LVDS от 3.3 В, Quartus тоже ругается, **но всё работает, данные летят, лок не падает**



# ПЛИС и стройматериалы

іТМО

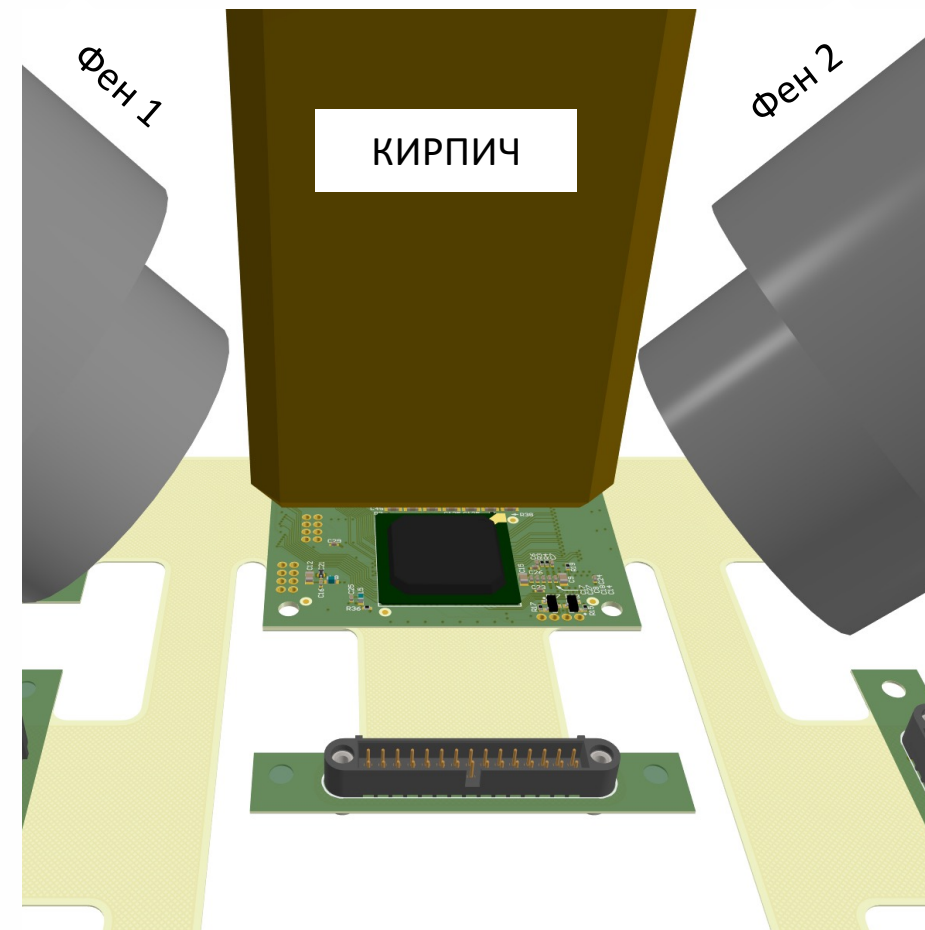
Разработали гигантских размеров ГЖПП для Cyclone IV, размеры платы 210x160 мм, 2-6 слоёв, 5 класс точности

Платы обошлись в приличную сумму (около 100к/шт., 2014 год) при этом жёсткую часть **покоробило**

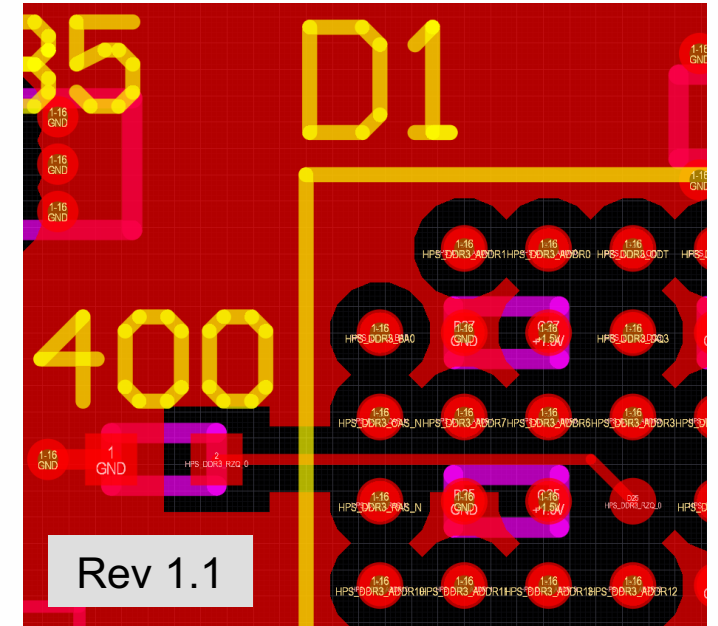
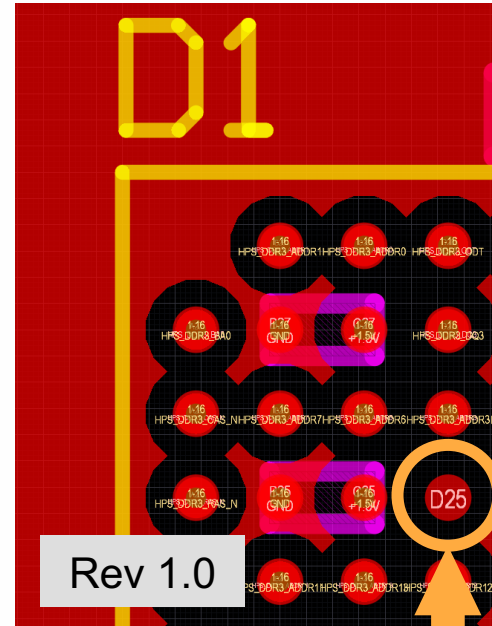
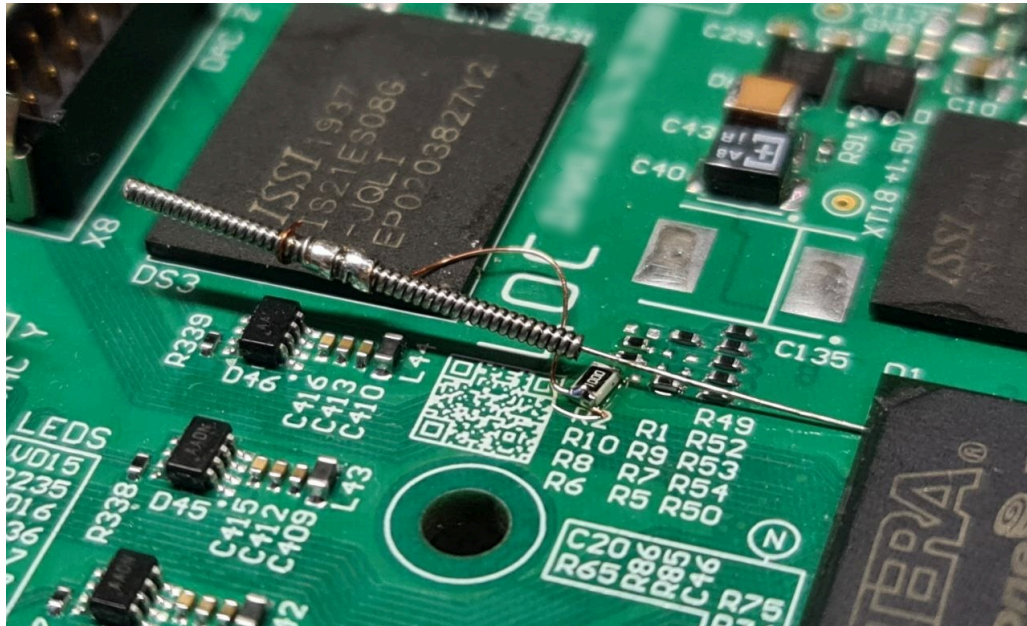
Выяснилось, что отклонения геометрии платы «на тоненького» укладываются в ГОСТ 23752-79, но корпус FBGA-484 всё равно не касается платы частью шариков

Разогрели плату двумя фенами и распрямили её с помощью **КИРПИЧА**, поставили BGA, плавно остудили

Всё запустилось, отладили прошивку на этой плате, впоследствии проблему с короблением решили сменой производителя плат



Примерная реконструкция процедуры  
(на тот момент было не до фото!)

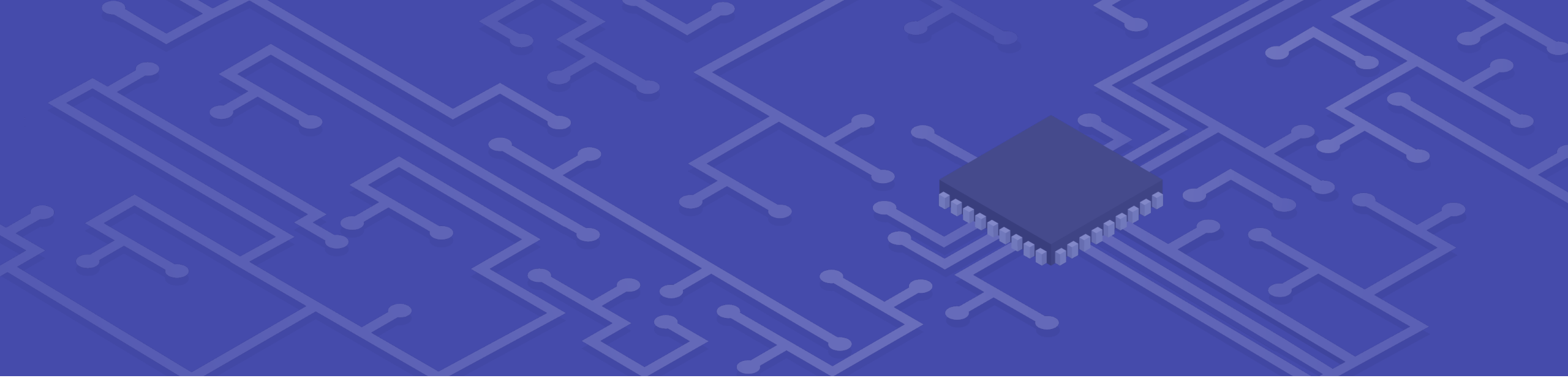


пин D25

В первой ревизии платы с Cyclone V SoC забыли подключить пин RZQ на землю через 100 Ом, DDR3 не стартовала

Покрывали акупунктурную иглу уретановым лаком (чтобы не замкнуть лишнего), воткнули вслепую в пин RZQ (D25)

Всё запустилось, отлаживали прошивку на этой плате несколько месяцев, заодно подоспела ревизия 1.1



Часть пятая

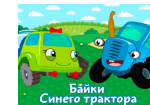
# НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ КОНСТРЕЙНТОВ



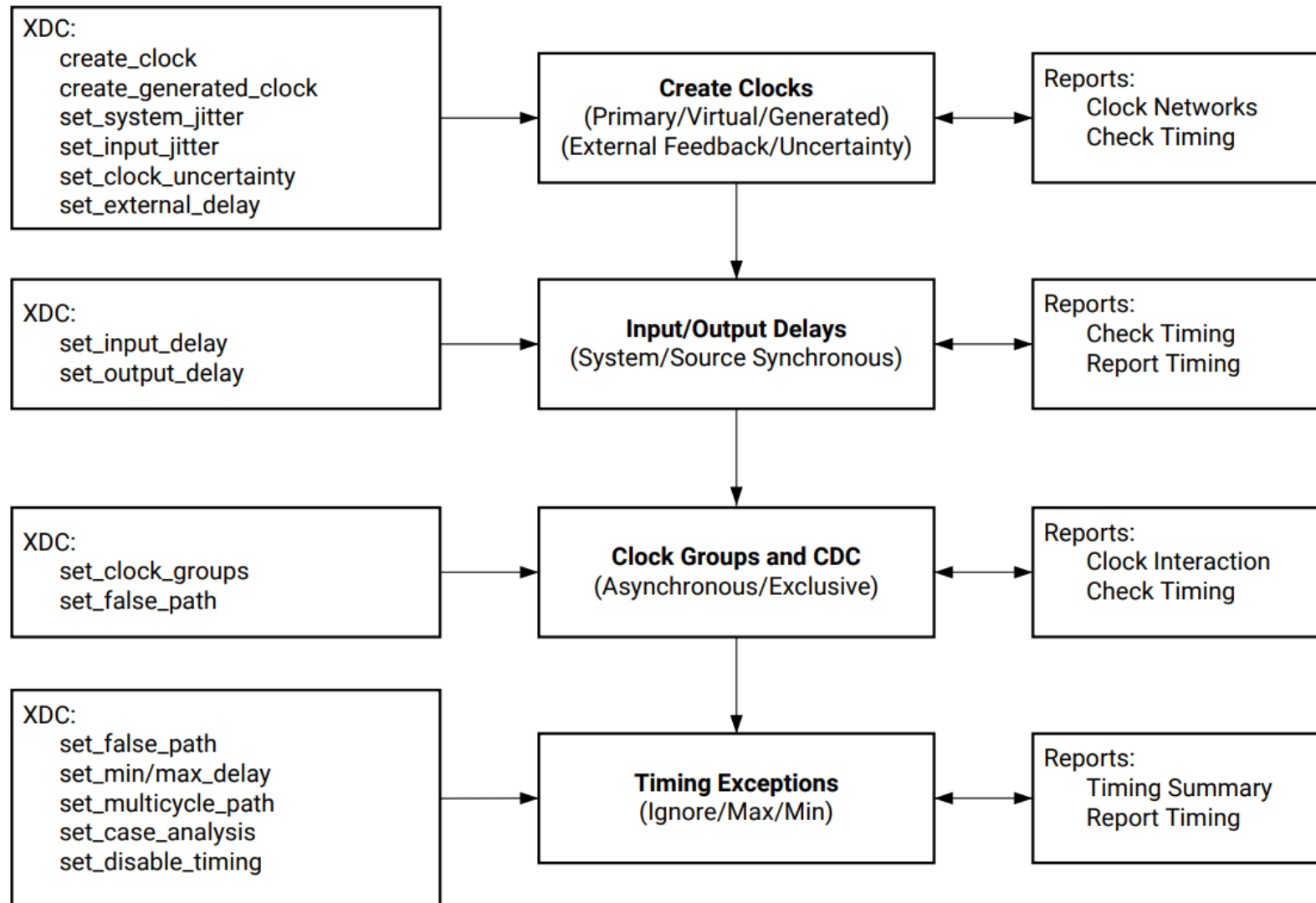
# Многообразие терминов

removal Latch  
skew false multicycle  
path slack  
Domain-crossing  
setup required hold  
Launch  
arrival groups  
level

Большинство инженеров, работающих с ПЛИС, НЕ УДЕЛЯЮТ ВРЕМЕННЫМ ОГРАНИЧЕНИЯМ  
ДОСТАТОЧНО ВНИМАНИЯ !!!



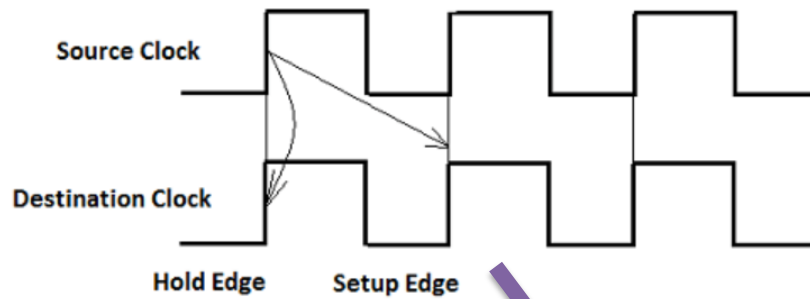
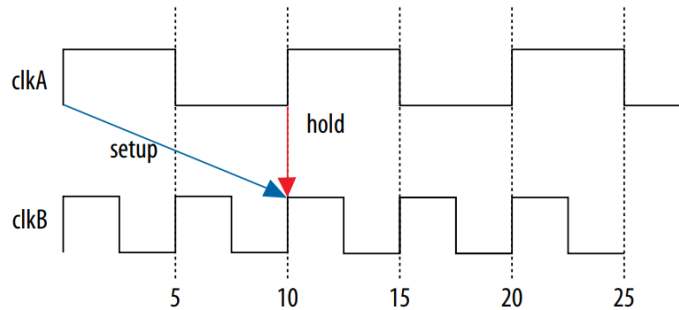
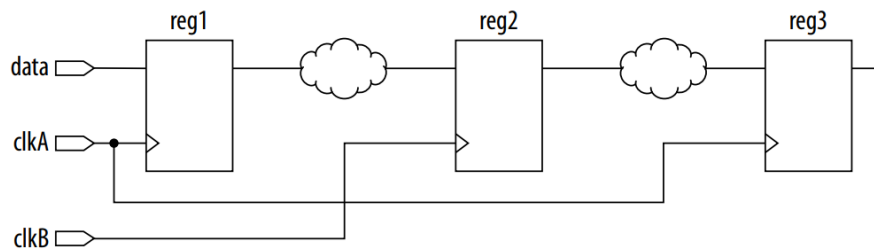
# Steps for Developing Timing Constraints



X13445-122019

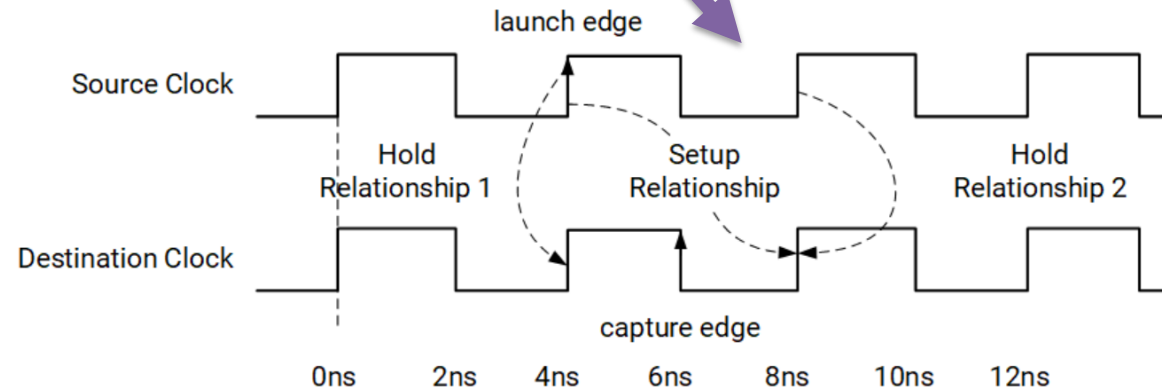
# Вид констрейнтов – мультициклы - сложность

Intel Quartus Prime Timing Analyzer Cookbook



[mathworks.com/help/hdlcoder/](https://www.mathworks.com/help/hdlcoder/)

Xilinx UG903



# Байка про то, как лечили времянку с помощью мультициклов

“-Не используй мультициклы, тебя на этом пути жду одни неприятности!  
-Ну как же не использовать, они же ЖДУТ!”



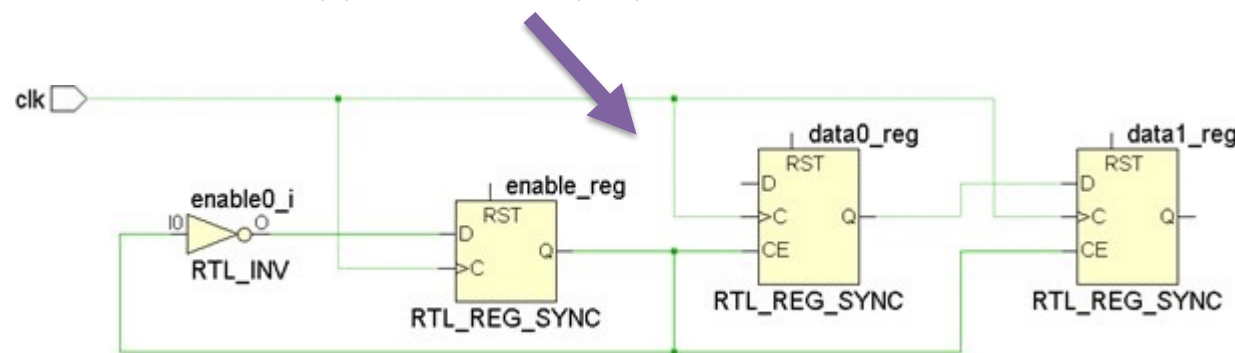
Последовательность действий - увидели непроходящие пути ->

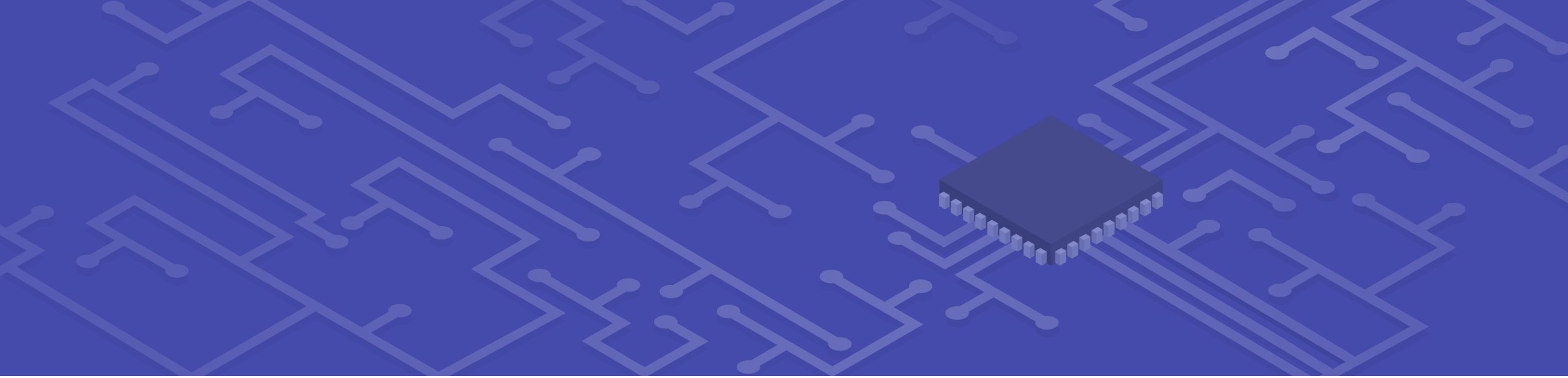
Slow 1200mV 85C Model Setup Summary			
«Filter»			
	Clock	Slack	End Point TNS
1	inst1 altpll_component auto_generated pll1 clk[1]	-1.182	-13.216
2	inst1 altpll_component auto_generated pll1 clk[0]	-0.523	-7.906
3	altera_reserved_tck	44.130	0.000

написали мультициклы -> ???  
Profit

Top Failing Paths  
No failing paths found.

Использование мультициклов должно соответствовать функционалу проекта!!!!





Часть шестая

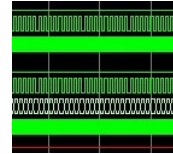
# НЕПРАВИЛЬНОЕ ВОСПРИЯТИЕ ВРЕМЯНКИ/ КОНСТРЕЙНТОВ

# Способы анализа проекта по производительности



## STA

- Быстрый и эффективный метод анализа
- Не учитывает логику работы



## Timing simulation

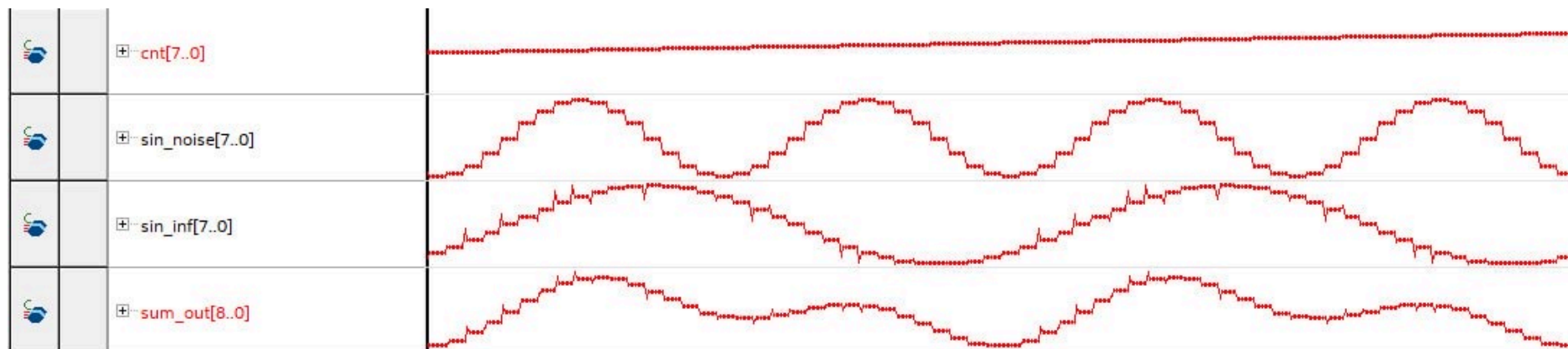
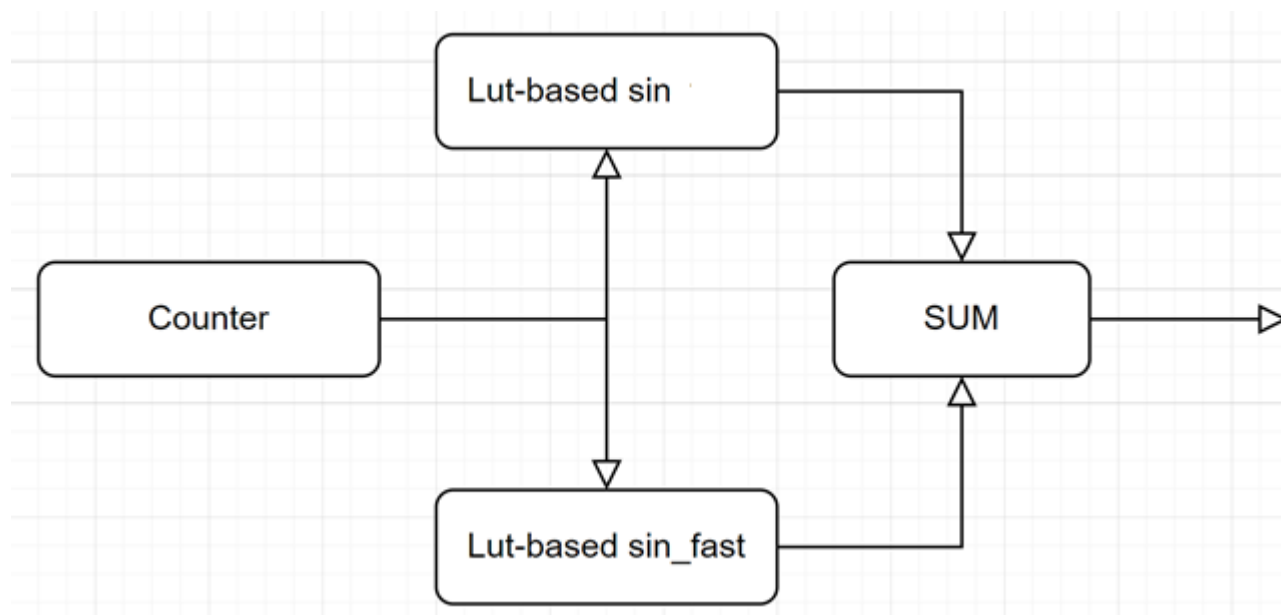
- Наиболее близкий к верификации метод моделирования
- Долгий и громоздкий



## On-board verification

- Самый любимый 😊 способ анализа
- Может не проходить по таймингам сам отладчик!

# Демо с восприятием на отладчике



# Демо с восприятием на отладчике

Slack	From Node	To Node	Launch Clock	Latch Clock
-0.990	sld_signaltap:auto_signaltap_0/sld_signaltap_ld_signaltap_body/status_data_pipe_reg[0][19]	sld_signaltap:auto_signaltap_0/sld_signaltap_o_generated/ram_block1a1~porta_address_reg0	inst altpll_component auto_generated pll1 clk[0]	inst altpll_component auto_generated pll1
-0.990	sld_signaltap:auto_signaltap_0/sld_signaltap_d_buffer_manager_inst/modified_post_count[4]	sld_signaltap:auto_signaltap_0/sld_signaltap_r_manager_inst/buffer_manager_next_address[1]	inst altpll_component auto_generated pll1 clk[0]	inst altpll_component auto_generated pll1
-0.988	counter_inst5/counter_sig[4]	sin_table_inst4/o_data[5]	inst altpll_component auto_generated pll1 clk[0]	inst altpll_component auto_generated pll1
-0.987	sld_signaltap:auto_signaltap_0/sld_signaltap_d_buffer_manager_inst/modified_post_count[6]	sld_signaltap:auto_signaltap_0/sld_signaltap_r_manager:sld_buffer_manager_inst/counter[0]	inst altpll_component auto_generated pll1 clk[0]	inst altpll_component auto_generated pll1
-0.987	sld_signaltap:auto_signaltap_0/sld_signaltap_r_manager:sld_buffer_manager_inst/counter[6]	sld_signaltap:auto_signaltap_0/sld_signaltap_r_manager:sld_buffer_manager_inst/counter[3]	inst altpll_component auto_generated pll1 clk[0]	inst altpll_component auto_generated pll1
-0.987	sld_signaltap:auto_signaltap_0/sld_signaltap_r_manager:sld_buffer_manager_inst/counter[3]	sld_signaltap:auto_signaltap_0/sld_signaltap_r_manager_inst/last_buffer_write_address_sig[7]	inst altpll_component auto_generated pll1 clk[0]	inst altpll_component auto_generated pll1
-0.987	sld_signaltap:auto_signaltap_0/sld_signaltap_d_buffer_manager_inst/modified_post_count[3]	sld_signaltap:auto_signaltap_0/sld_signaltap_r_manager_inst/buffer_manager_next_address[10]	inst altpll_component auto_generated pll1 clk[0]	inst altpll_component auto_generated pll1
-0.987	sld_signaltap:auto_signaltap_0/sld_signaltap_r_manager:sld_buffer_manager_inst/counter[6]	sld_signaltap:auto_signaltap_0/sld_signaltap_r_manager:sld_buffer_manager_inst/counter[6]	inst altpll_component auto_generated pll1 clk[0]	inst altpll_component auto_generated pll1
-0.987	sld_signaltap:auto_signaltap_0/sld_signaltap_g_pipeline_less_than_two.sm0:4.sm1/regoutff	sld_signaltap:auto_signaltap_0/sld_signaltap_i_ela_trigger_flow_mgr_entity/last_level_delayed	inst altpll_component auto_generated pll1 clk[0]	inst altpll_component auto_generated pll1

#923: Setup slack is -0.988 (VIOLATED)

Summary Statistics Data Path Waveform

Arrival Path

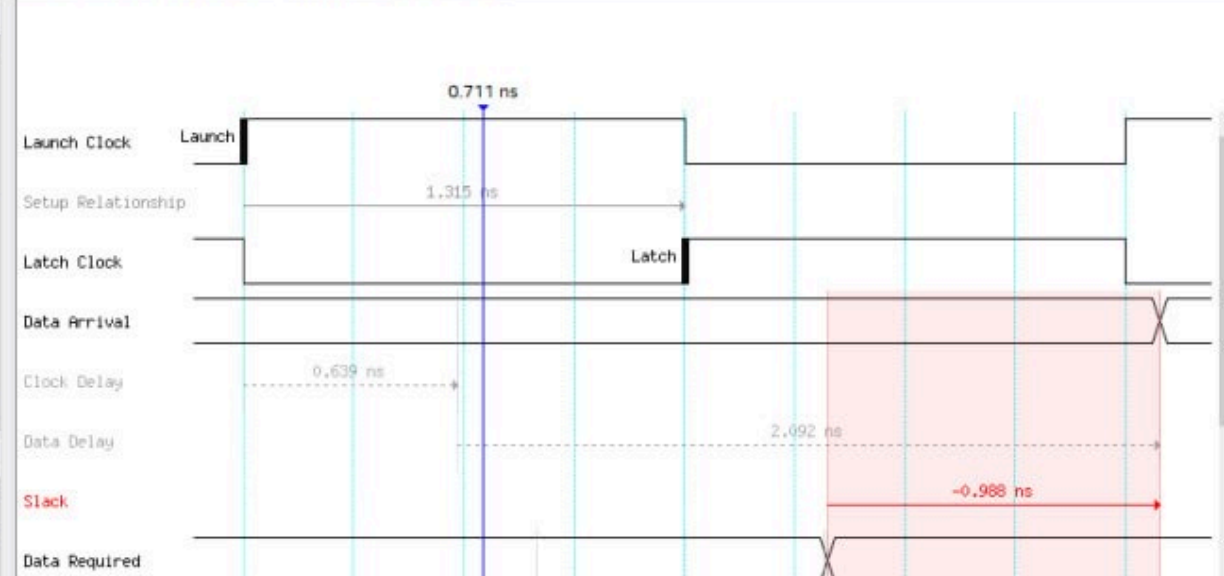
Total	Incr	RF	Type	Fanout	Location	Element
0.000	0.000					launch edge time
0.639	0.639					clock path
0.000	0.000					source latency
0.000	0.000			1	PIN_P11	CLOCK_50
0.000	0.000	RR	IC	1	IOIBUF_X34_Y0_N29	CLOCK_50~input i
0.793	0.793	RR	CELL	1	IOIBUF_X34_Y0_N29	CLOCK_50~input o
4.279	3.486	RR	IC	1	PLL_1	inst altpll_component auto_generated pll1 inclk[0]
-4.179	-8.458	RR	COMP	2	PLL_1	inst altpll_component auto_generated pll1 observablev
-4.179	0.000	RR	CELL	1	PLL_1	inst altpll_component auto_generated pll1 clk[0]

Required Path

Total	Incr	RF	Type	Fanout	Location	Element
1.315	1.315					latch edge time

Path #923: Setup slack is -0.988 (VIOLATED)

Path Summary Statistics Data Path Waveform

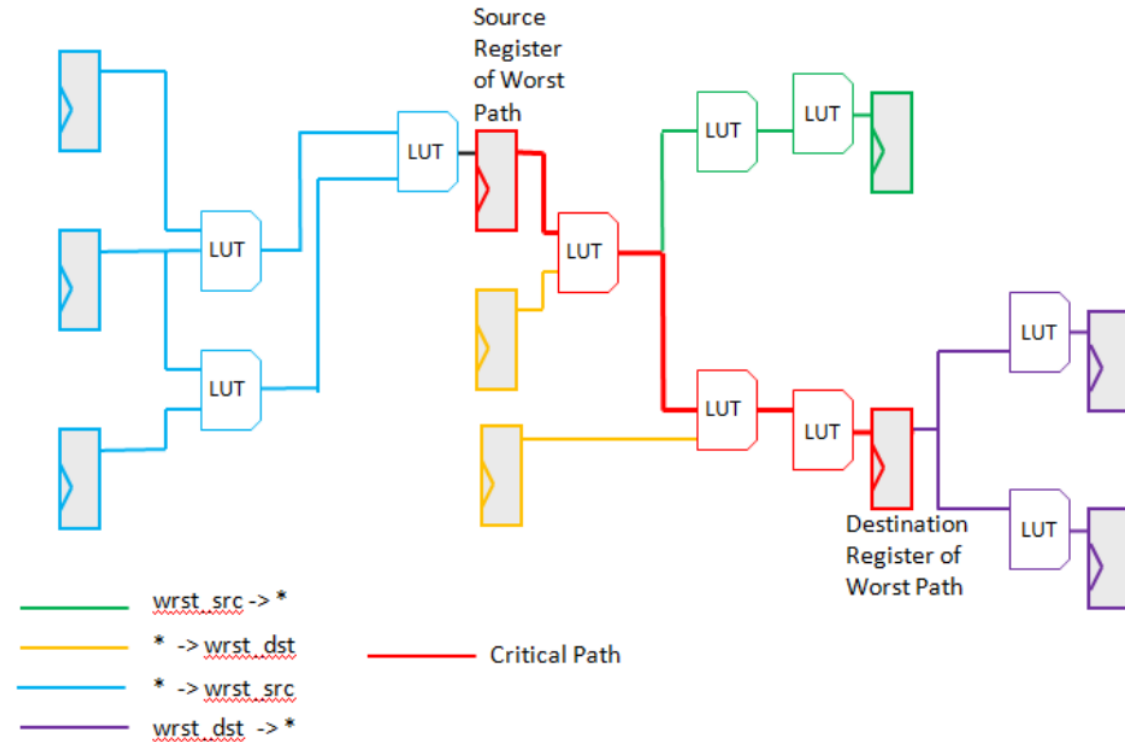




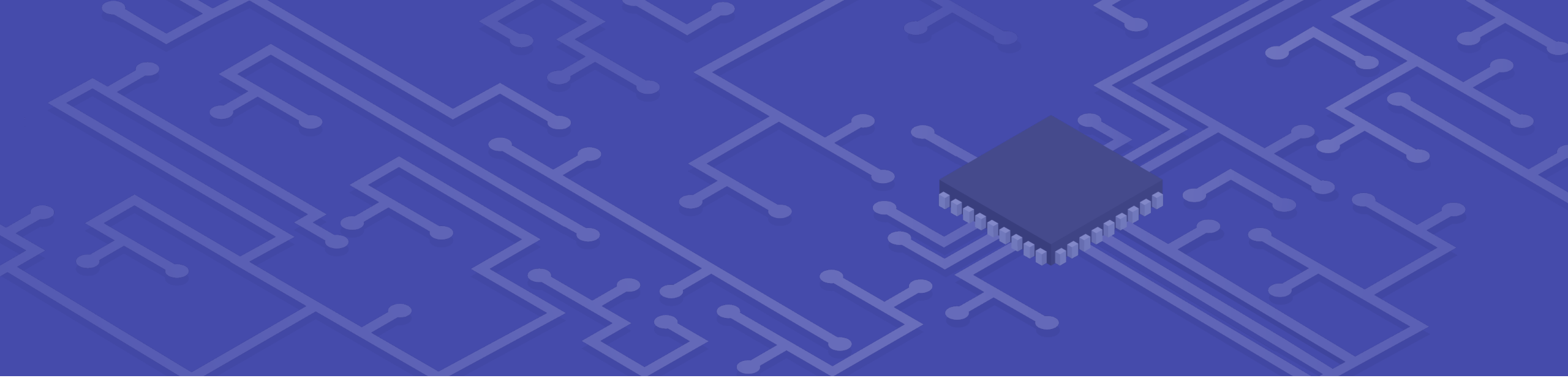
# Полезные советы



1. С точки зрения производительности (временки), полезно проанализировать что растаскивает элементы в критическом пути



2. WNS – критические пути могут меняться по мере прохождения этапов компиляции!!



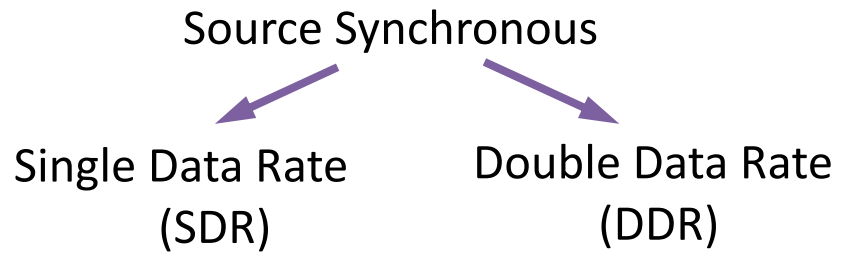
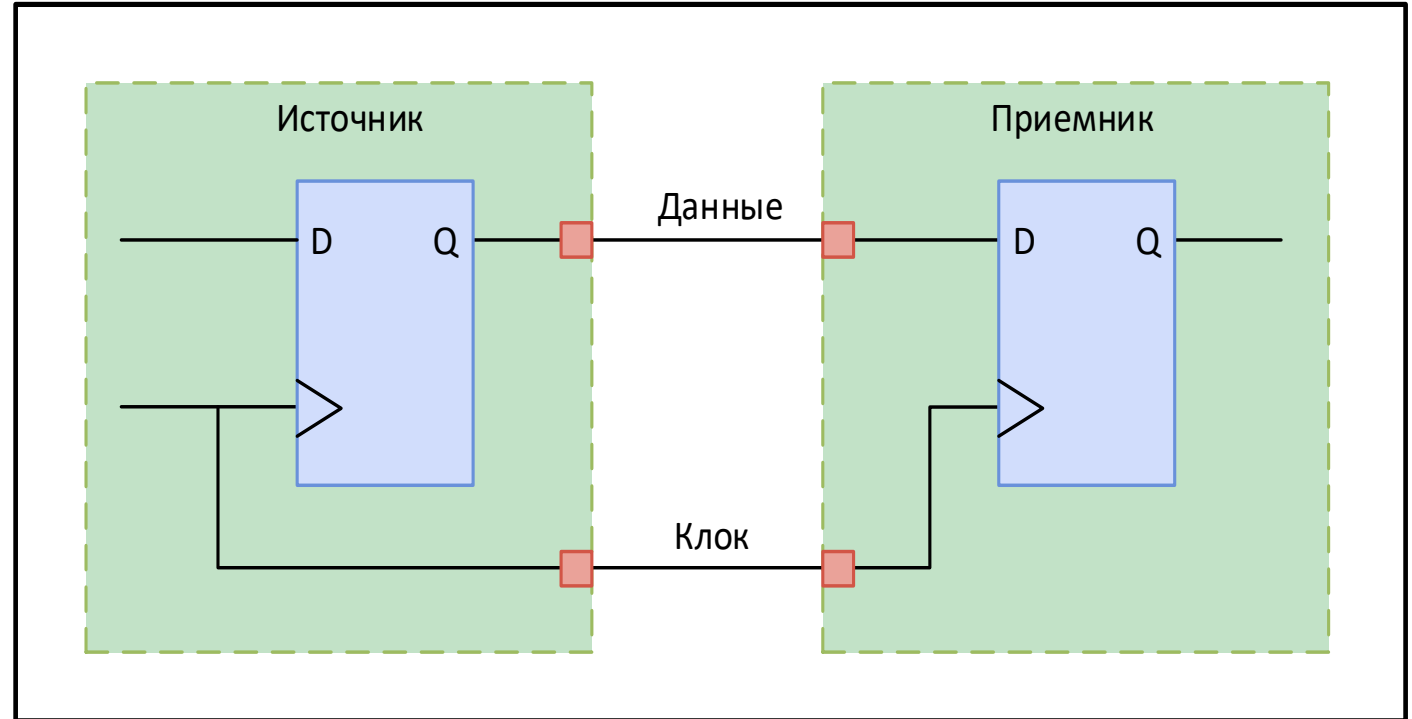
Часть седьмая

# ВНЕШНИЕ ИНТЕРФЕЙСЫ - КОНСТРЕЙНТЫ

# Что такое Source Synchronous интерфейсы

## Source Synchronous интерфейс

- Клок и данные передаются по одинаковым путям, напрямую от источника к приемнику.
- Внешние помехи (изменение напряжения, температуры и т.д.) оказывают одинаковое влияние как на тракт передачи данных, так и на тракт клона.
- Основное преимущество - увеличение скорости передачи данных.



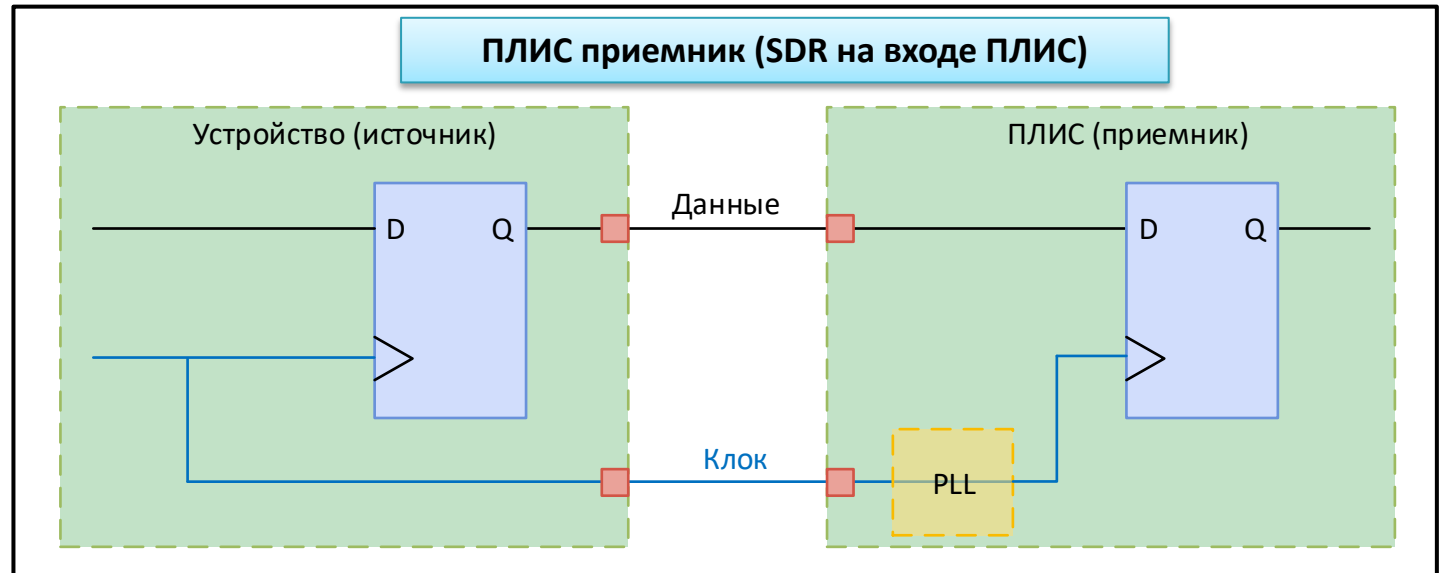
## Высокоскоростная передача данных

- DDR память
- HyperTransport шины (например, HyperBus)
- PCI Express
- Кастомные интерфейсы (например, АЦП и ЦАП)

# Временной анализ SDR интерфейсов в ПЛИС

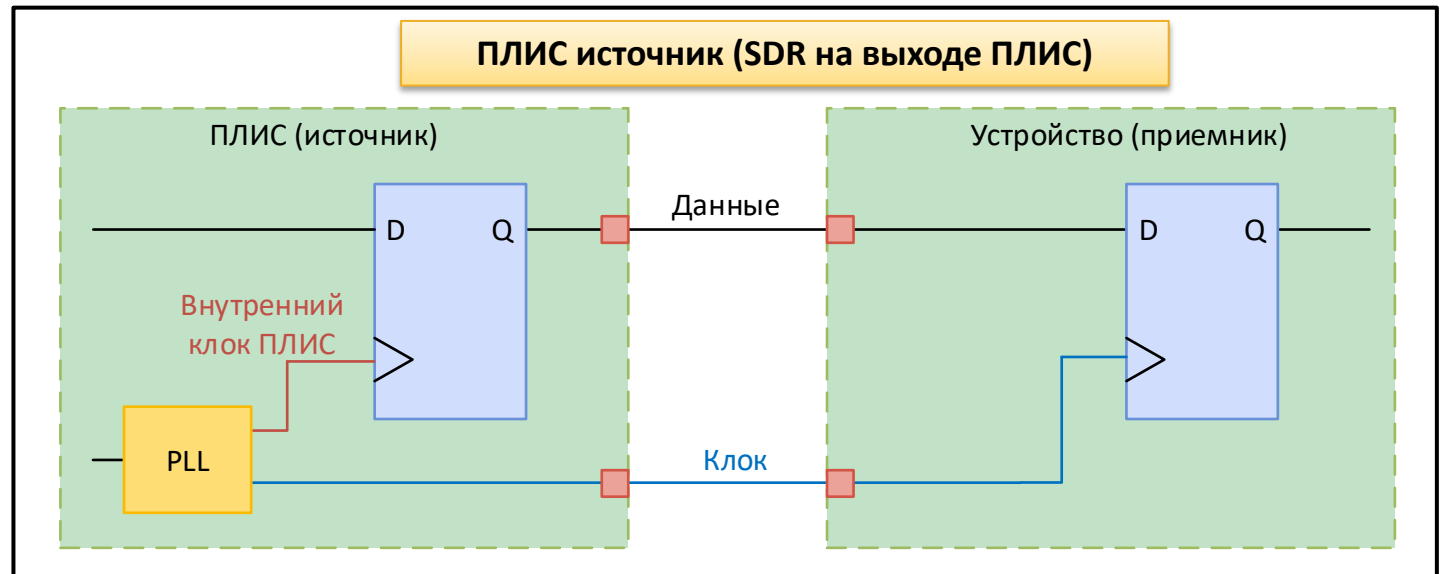
## Случай, когда ПЛИС приемник

1. Создать клок, приходящий на входной пин ПЛИС (base clock).
2. Задать минимальную и максимальную задержки для данных на входе ПЛИС (input delay) относительно приходящего клона.
3. Добавить дополнительные констрейты (исключения), если требуется.



## Случай, когда ПЛИС источник

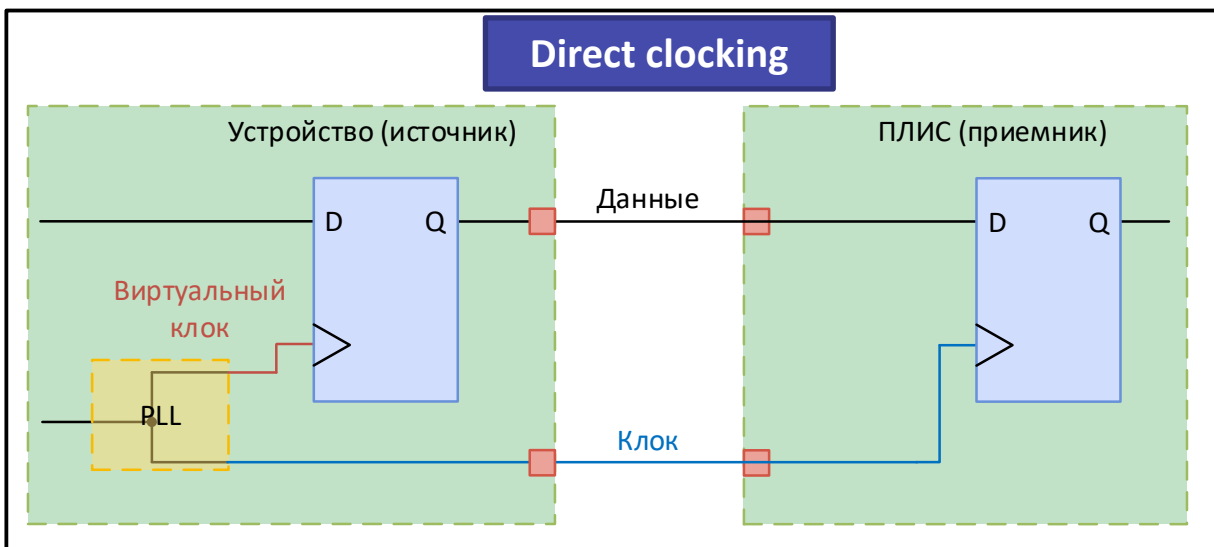
1. Создать клок, исходящий из выходного пина ПЛИС (generated clock).
2. Задать минимальную и максимальную задержки для данных на выходе ПЛИС относительно исходящего клона (output delay).
3. Добавить исключения, если требуется.



# Базовые схемы подключения SDR интерфейса ко входу ПЛИС

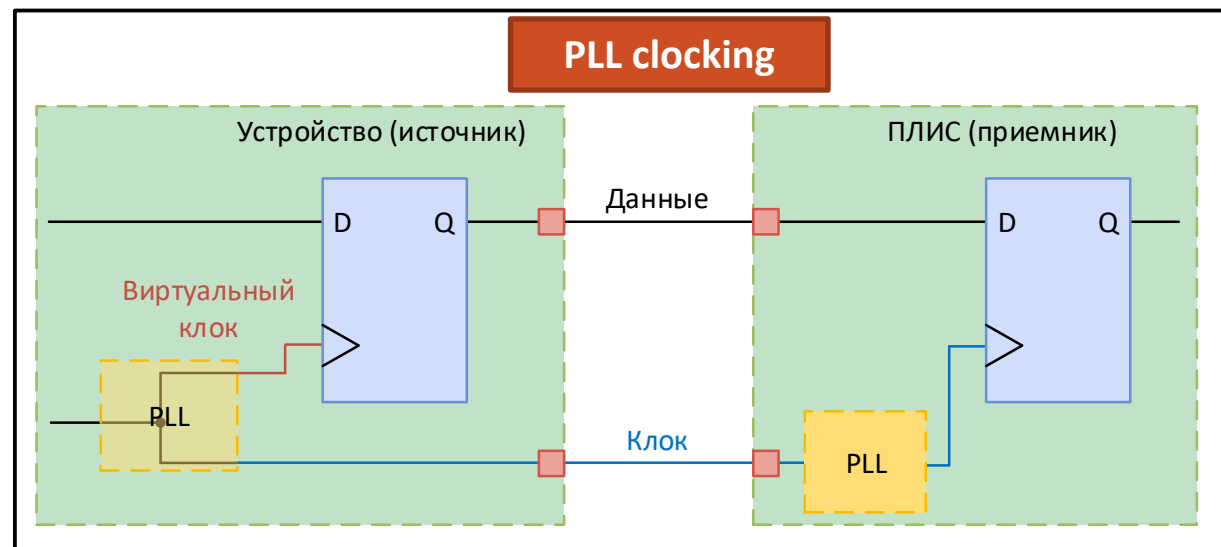
## Подключение входящего блока напрямую к регистрам ПЛИС (Direct clocking)

1. Только для center-aligned передач (однако, в некоторых случаях можно использовать и при edge-aligned)
2. Низкие скорости передачи
3. Входящий клок может прерываться, т.е. быть не постоянным (not free-running)

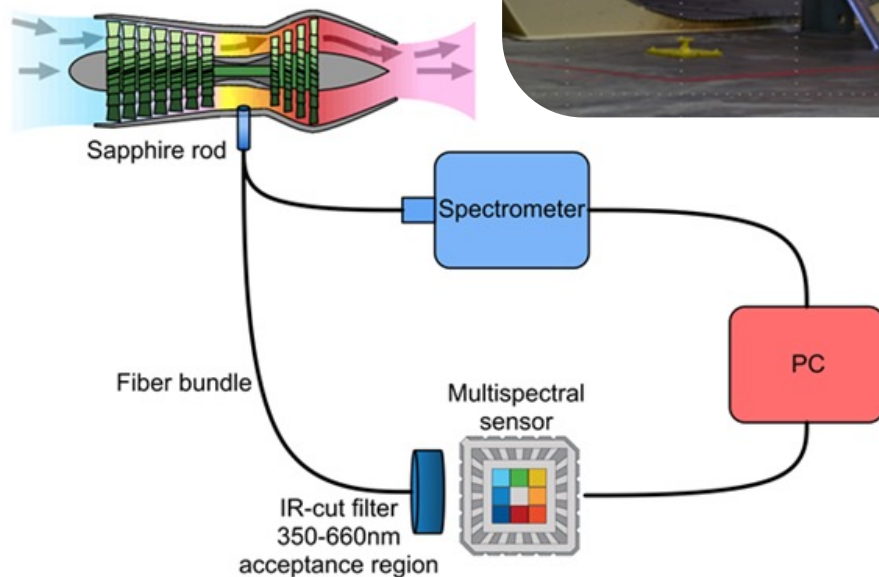
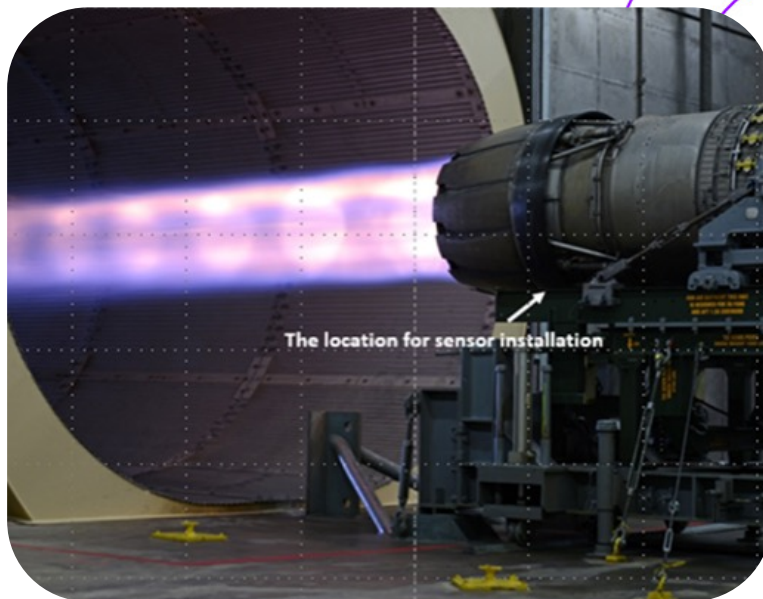


## Подключение входящего блока к регистрам через MMCM или PLL (PLL clocking)

1. Подходит для обоих режимов передачи
2. Высокие скорости передачи
3. Точная настройка фазового соотношения между клокком и данными
4. Входящий клок не может прерываться, т.е. должен быть постоянным (free-running)



# История - мониторинг температуры в ГТД



Традиционные методы контроля температуры (например, термопары) непригодны для точных и долговременных измерений

Волоконная оптика позволяет использовать спектрометрические методы контроля температуры, при этом основной блок обработки находится вне зоны высоких температур.

Анализ соотношения воздух/топливо и химического состава горючих газов внутри камеры сгорания позволяет достичь диапазона измерений от 800°C до 2600°C.

# Мониторинг температуры в ГТД

ИТМО

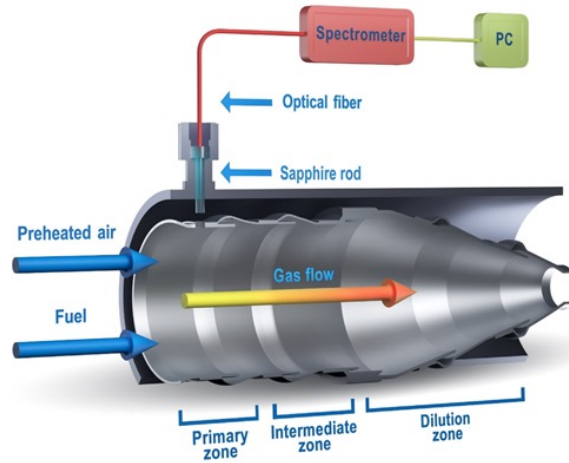
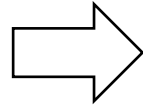
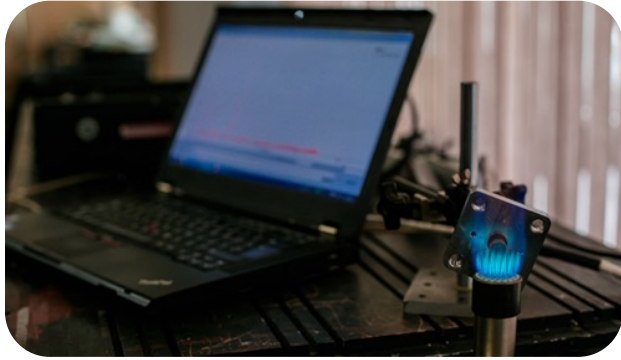
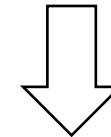
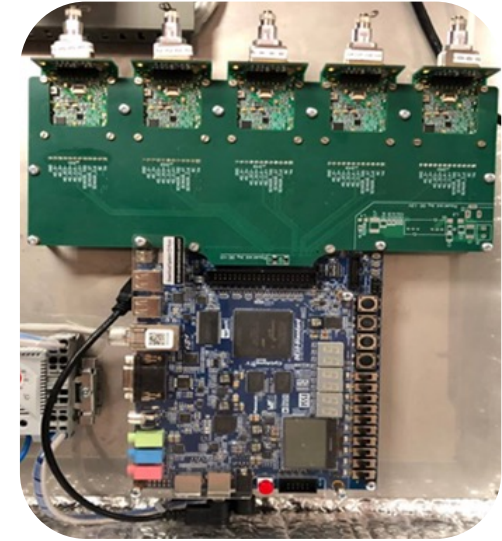
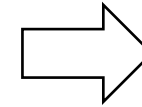
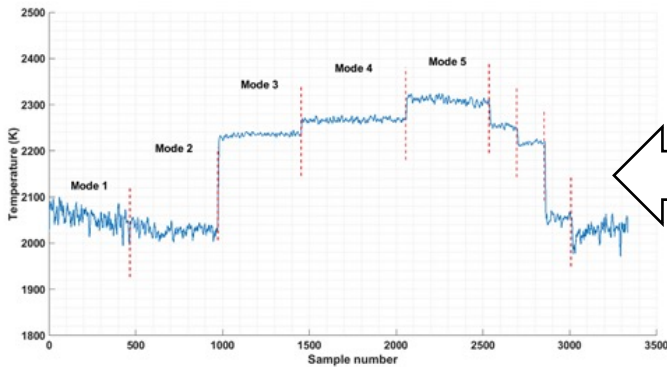
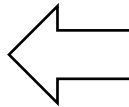


Схема эксперимента

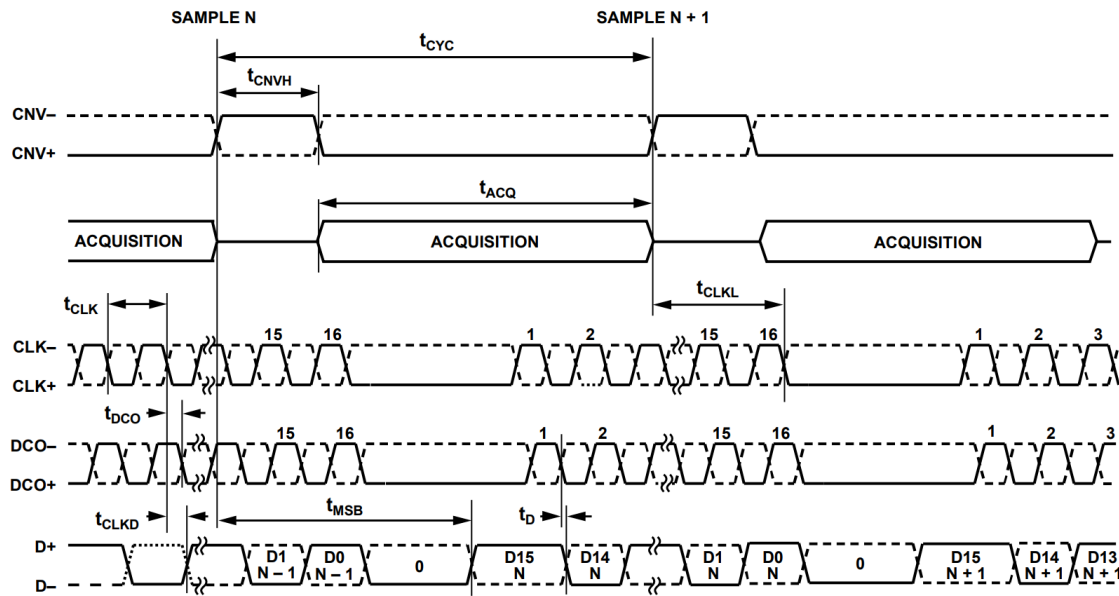


Спектр на логическом анализаторе

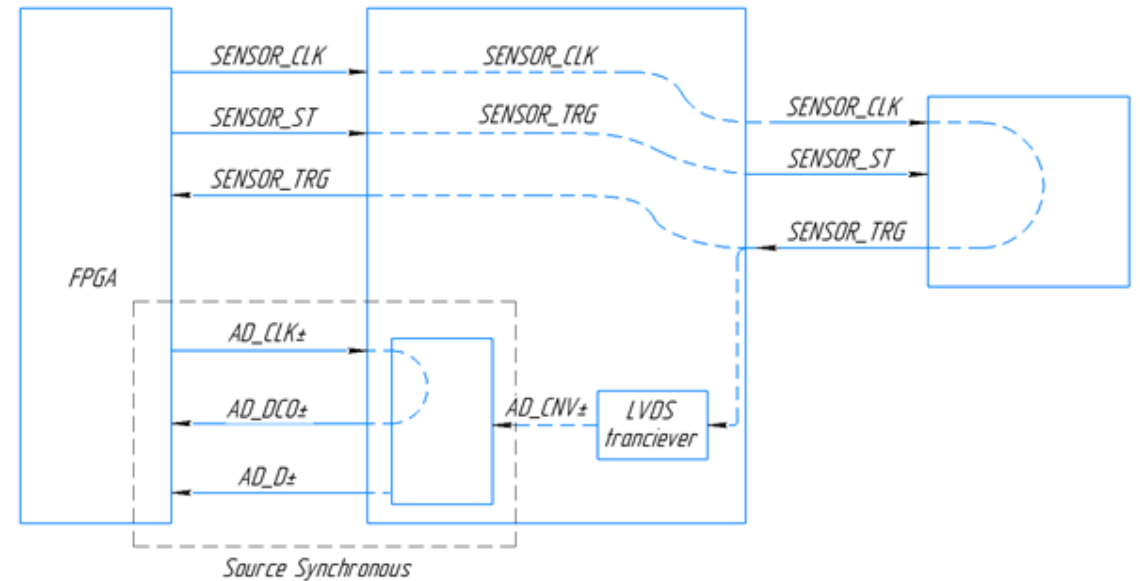


Температура газового потока внутри камеры сгорания определяется с помощью комбинации температуры сажи и вибрационной температуры C2.

# Особенности интерфейса взаимодействия со спектрометром



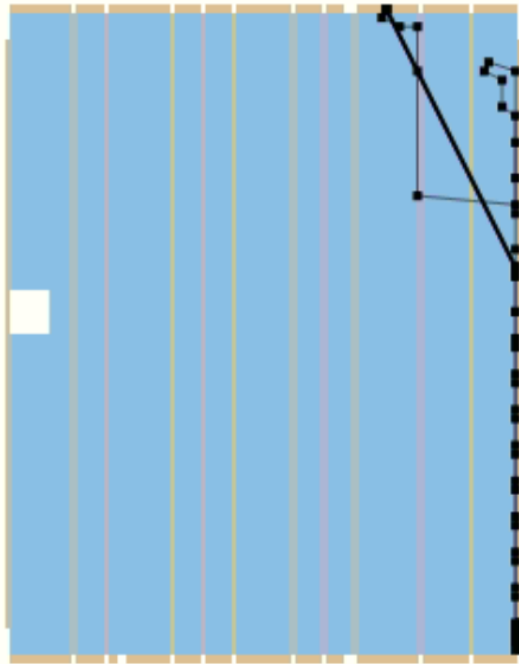
Временная диаграмма АЦП в режиме интерфейса с эхо-синхронизацией



Блок-схема взаимодействия сигналов



## На заметку!



Бывает, что данные прибегают на ПЛИС **сильно быстрее** блока, их защёлкивающего



Нужно заранее подумать об этом при проектировании, иначе придётся изощряться



Генеральный партнёр конференции FPGA-Systems 2024.1



Первая современная отечественная САПР,  
реализующая сквозной цикл проектирования печатных плат



[www.aremex.ru](http://www.aremex.ru)

# Где найти FPGA / RTL / Verification комьюнити?

[FPGA-Systems.ru](http://FPGA-Systems.ru)

Сайт комьюнити

[FPGA-Systems Magazine \(FSM\)](#)

Первый журнал о программируемой логике

[@fpgasystems](#)

Телеграм чат

[admin@fpga-systems.ru](mailto:admin@fpga-systems.ru)

Электронная почта

[Youtube.com/c/fpgasystems](https://Youtube.com/c/fpgasystems)

Youtube канал

