

# Проектирование в Quartus II

Верификация проектов

# Верификация проекта

- Большая часть времени, затрачиваемого на проектирование ПЛИС, уходит на верификацию
- Методы верификации
  - Временной анализ
    - Проверка на соответствие проекта временным требованиям
  - Моделирование (отдельных модулей и системы в целом)
    - Проверка на корректность функционирования путем моделирования
  - Формальную верификацию
    - Доказательство корректности реализации
  - Анализ энергопотребления
    - Проверка на соответствие требованиям по энергопотреблению
  - Анализ целостности сигналов
    - Проверка соответствия требованиям внешних интерфейсов
  - Внутрисхемное тестирование
    - Отладка и контроль в процессе работы устройства

# Возможности Quartus II

- Quartus® II предоставляет следующие возможности для верификации:
  - Временной анализатор TimeQuest
  - Пакет ModelSim-Altera [Starter] Edition и поддержка других симуляторов
  - Анализатор энергопотребления PowerPlay Power Analyzer
  - Анализ шумов при одновременных переключениях SSN Analyzer
  - Средства отладки интерфейсов External Memory Interface Toolkit и Transceiver Toolkit
  - Средства внутрисхемной отладки

# Проектирование в Quartus II

Временной анализ

# Цели

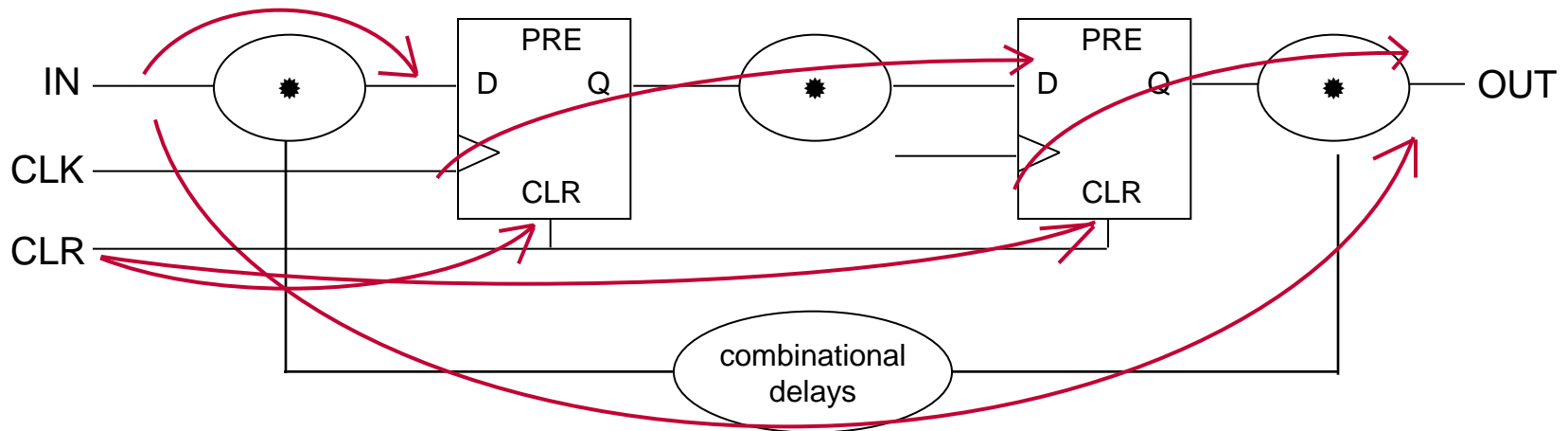
- Базовые понятия временного анализа
- Интерфейс TimeQuest
- Разработка файлов временных требований SDC
- Проверка временных характеристик простых проектов в анализаторе TimeQuest

# Принципы временной верификации

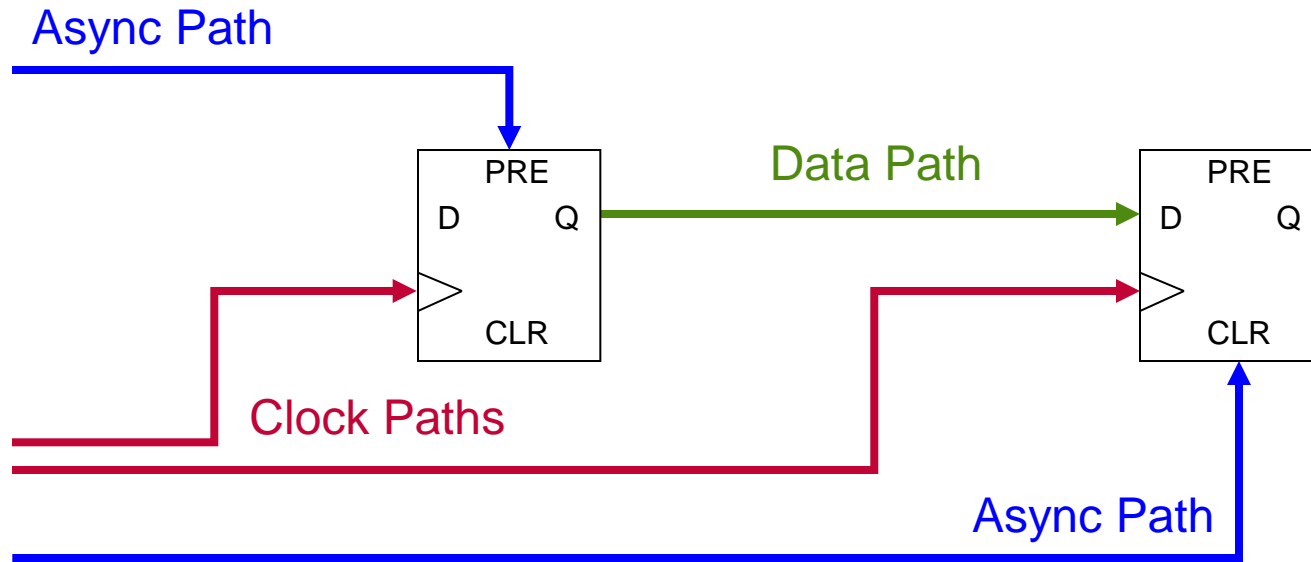
- Должен быть проведен анализ всех путей распространения сигналов на соответствие временным требованиям и ограничениям
  - Позволяет обнаружить ошибки, связанные с нарушениями временных требований, быстрее и надежнее, чем временное моделирование и тестирование устройства
- Требуется ввод всех требований и исключений
  - Используется при синтезе и разводке для оптимизации проектов
  - Используется при временном анализе после компиляции для контроля выполнения временных требований и анализа временных характеристик

# Принципы временной верификации

- Каждый временной путь имеет начальную и конечную точки
- Например, путь от источника тактовых импульсов до входа тактовых импульсов триггера



# Пути распространения сигналов



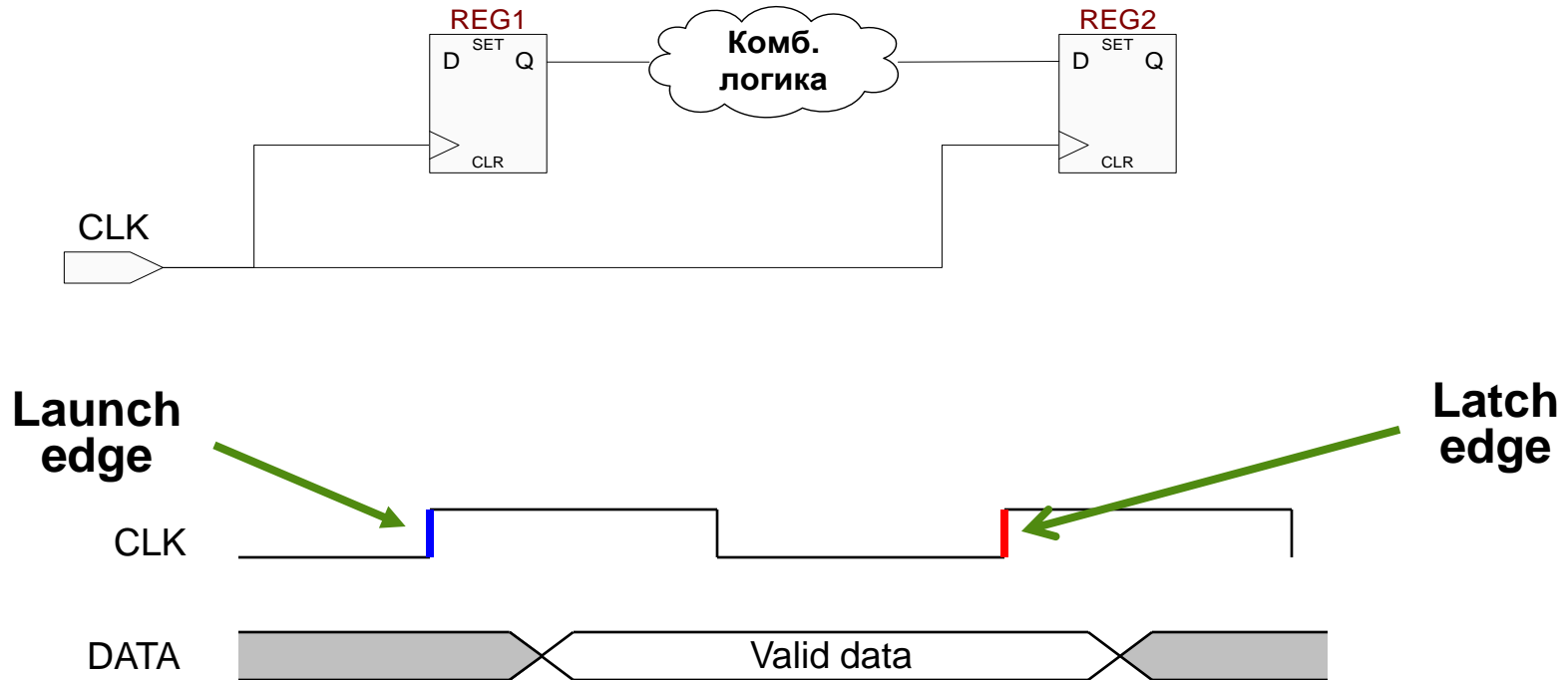
Три типа путей

1. Пути тактовых импульсов
2. Пути данных
3. Асинхронные пути\*

*\*Пути распространения сигналов, поступающих на асинхронные входы триггеров*

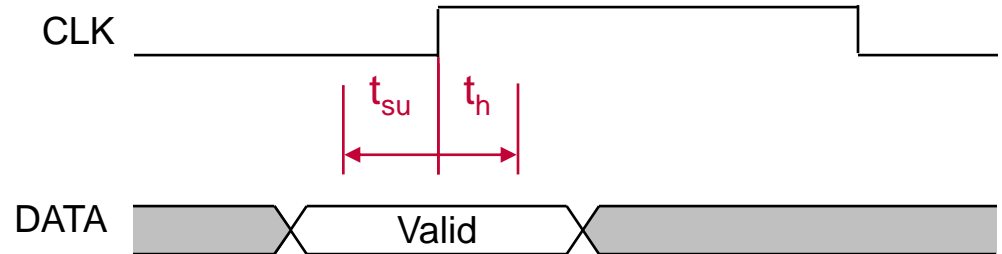


# Запускающий и защелкивающий фронт



- Запускающий фронт (Launch edge) – фронт, по которому срабатывает регистр-источник
- Защелкивающий фронт (Latch Edge) – фронт, по которому защелкиваются данные в регистре-приемнике

# Setup & Hold



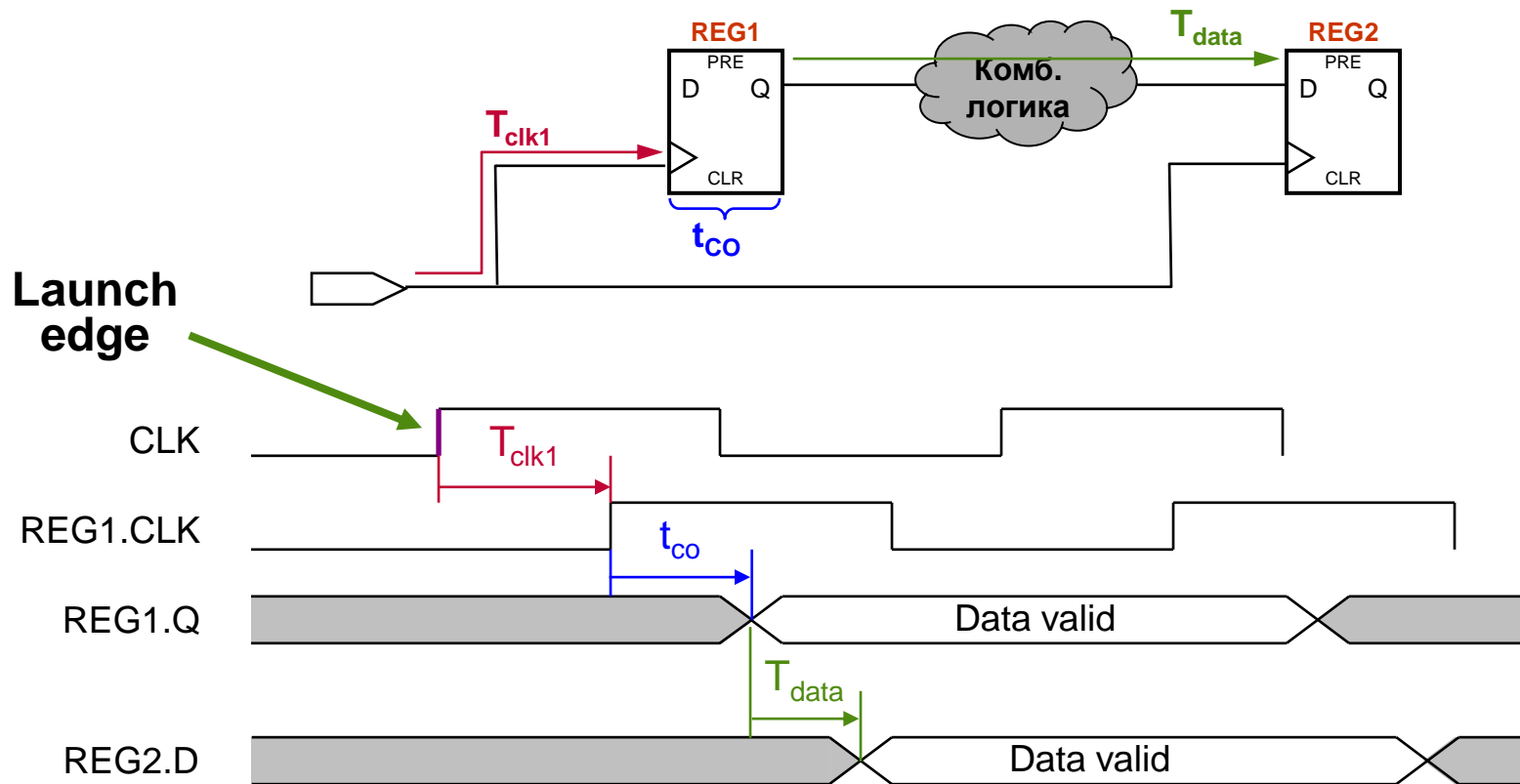
Setup (время предустановки) – минимальное время, в течение которого сигнал данных не должен изменяться **перед** приходом тактового импульса

Hold (время удержания) - минимальное время, в течение которого сигнал данных не должен изменяться **после** прихода тактового импульса

**Время установки и удержания формируют временное окно относительно фронта тактового импульса (Data Required Window), в котором данные не должны изменяться**

# Data Arrival Time

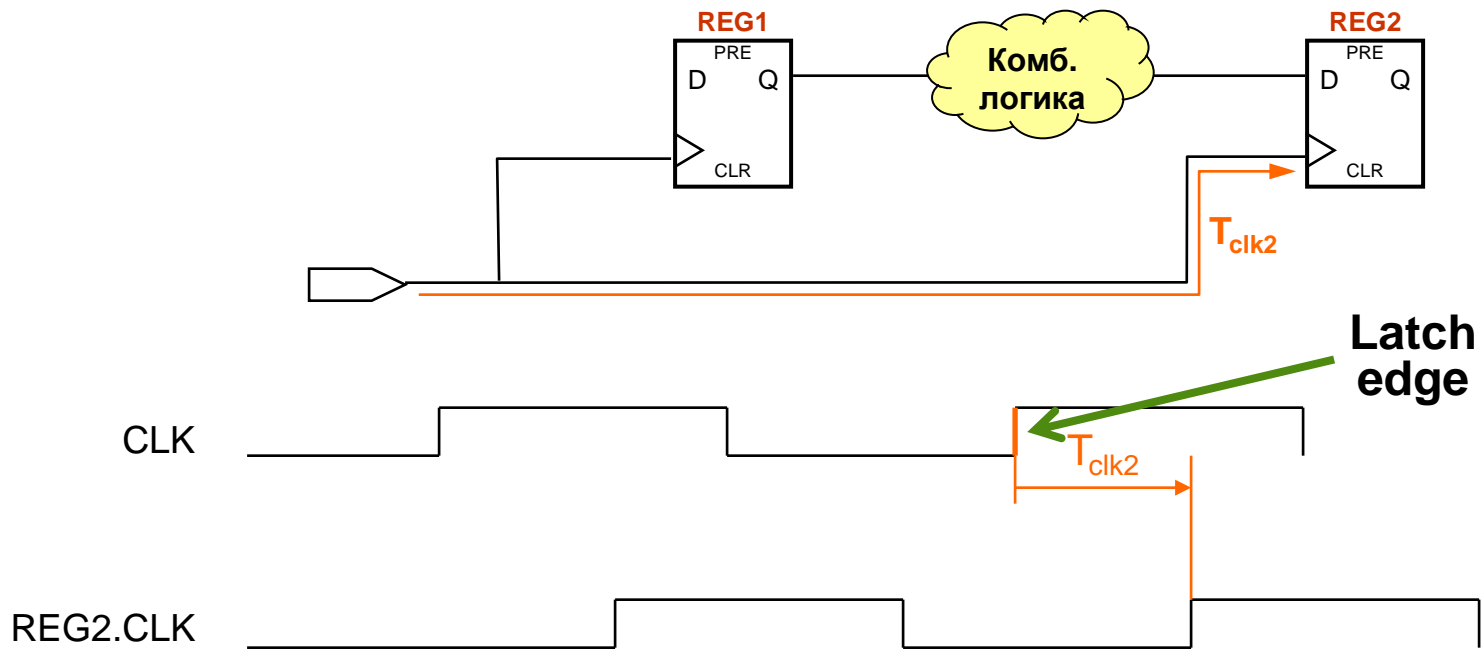
- Время прибытия данных - время поступления данных на синхронный вход регистра-приемника



$$\text{Data Arrival Time} = \text{launch edge} + T_{clk1} + t_{co} + T_{data}$$

# Clock Arrival Time

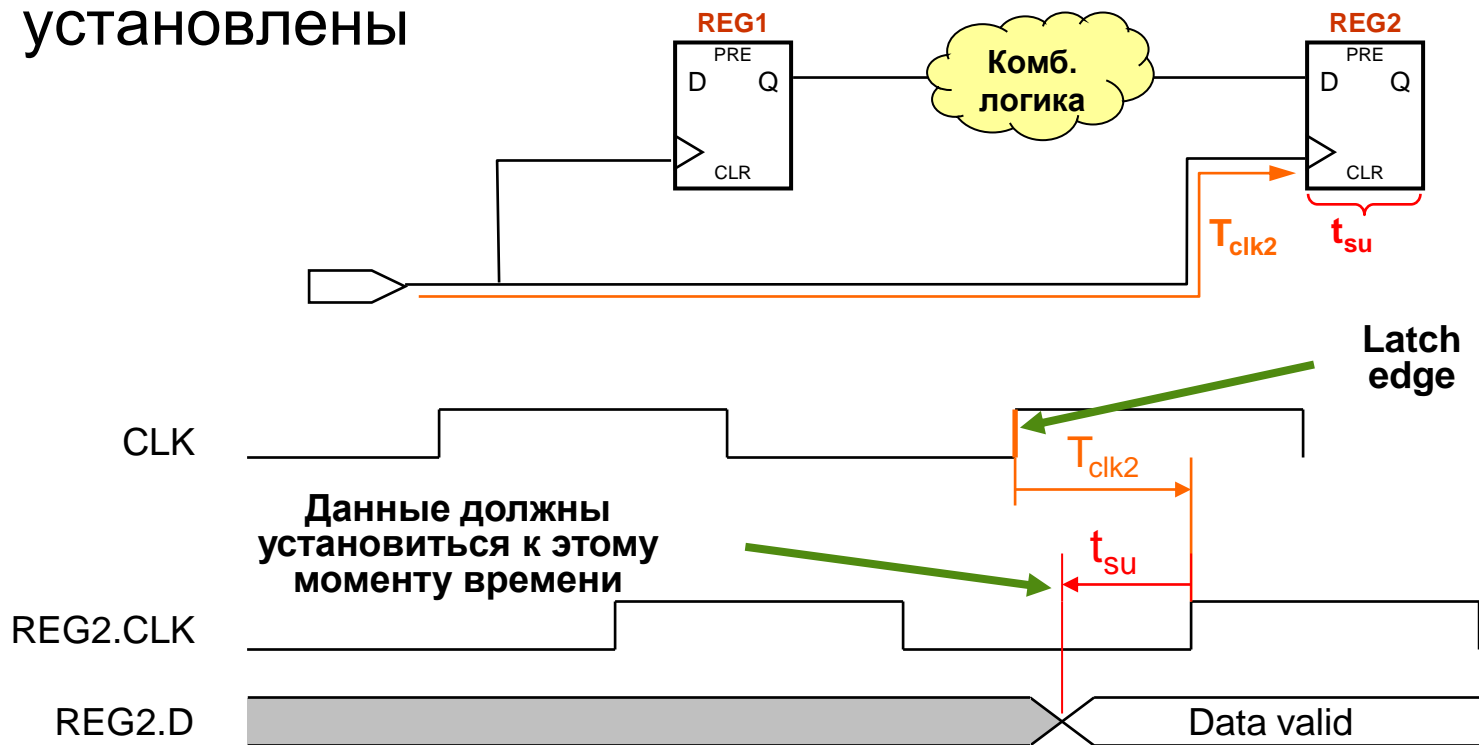
- Время прибытия тактового импульса - время поступления защелкивающего фронта на вход тактового импульса регистра-приемника



$$\text{Clock Arrival Time} = \text{latch edge} + T_{clk2}$$

# Data Required Time (Setup)

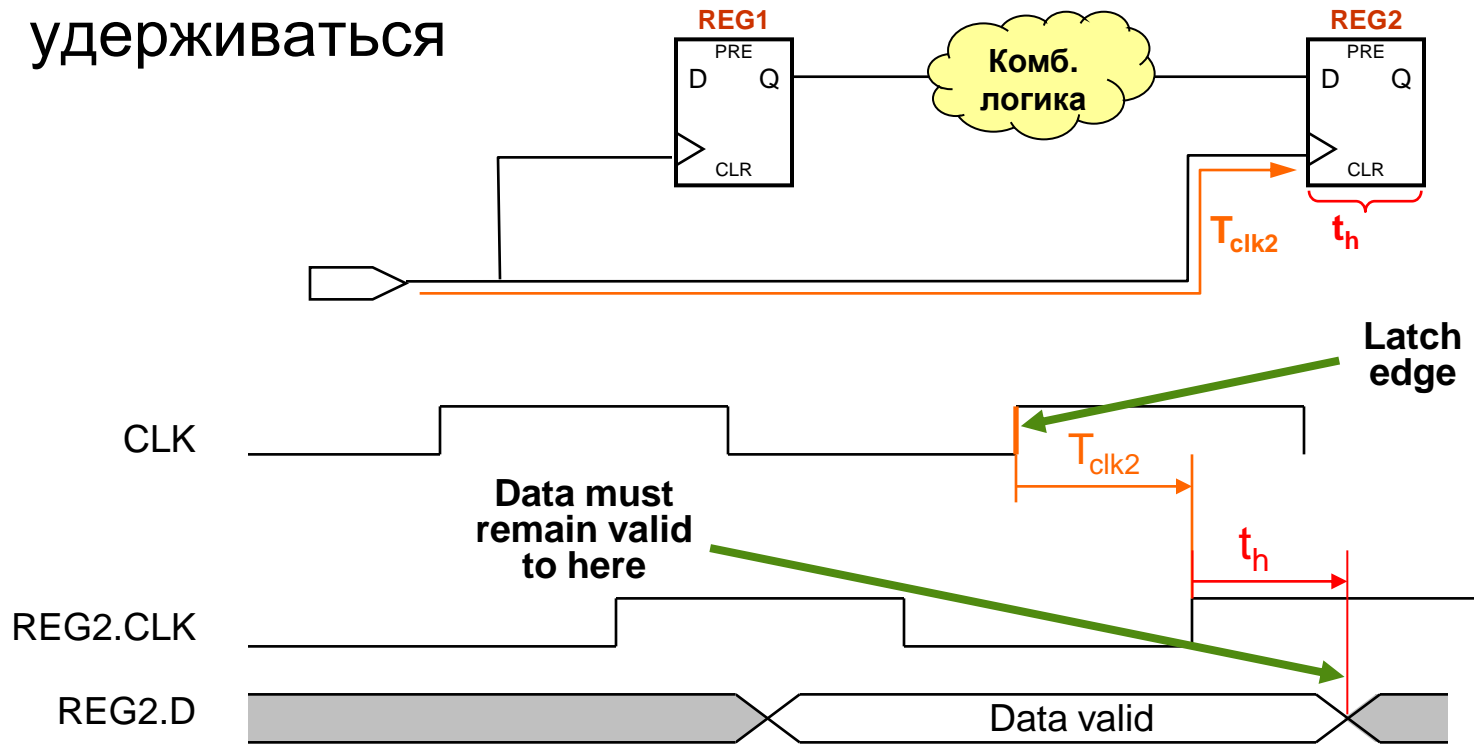
- Требуемое время предустановки данных - момент времени перед приходом защелкивающего фронта, к которому данные на входе триггеры должны быть установлены



$$\text{Data Required Time (Setup)} = \text{Clock Arrival Time} - t_{su} - \text{Setup Uncertainty}$$

# Data Required Time (Hold)

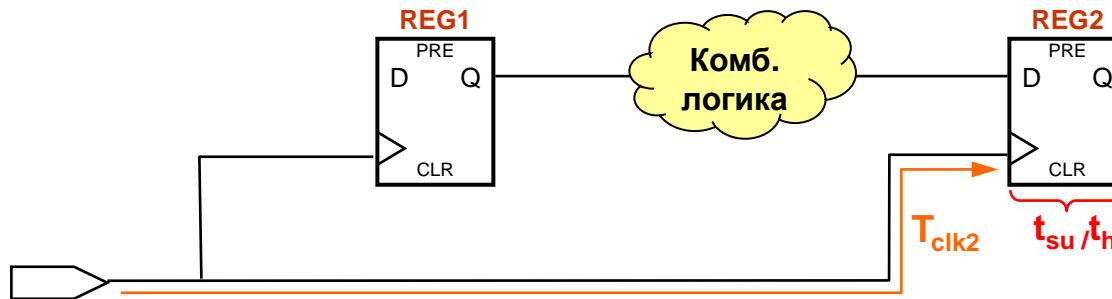
- Требуемое время удержания данных - момент времени после прихода защелкивающего фронта, до которого данные на входе триггера должны удерживаться



$$\text{Data Required Time (Hold)} = \text{Clock Arrival Time} + t_h + \text{Hold Uncertainty}$$

# Data Required Time и тактовый импульс

- Требуемые времена предустановки и удержания данных в основном определяются временем распространения тактового импульса от источника



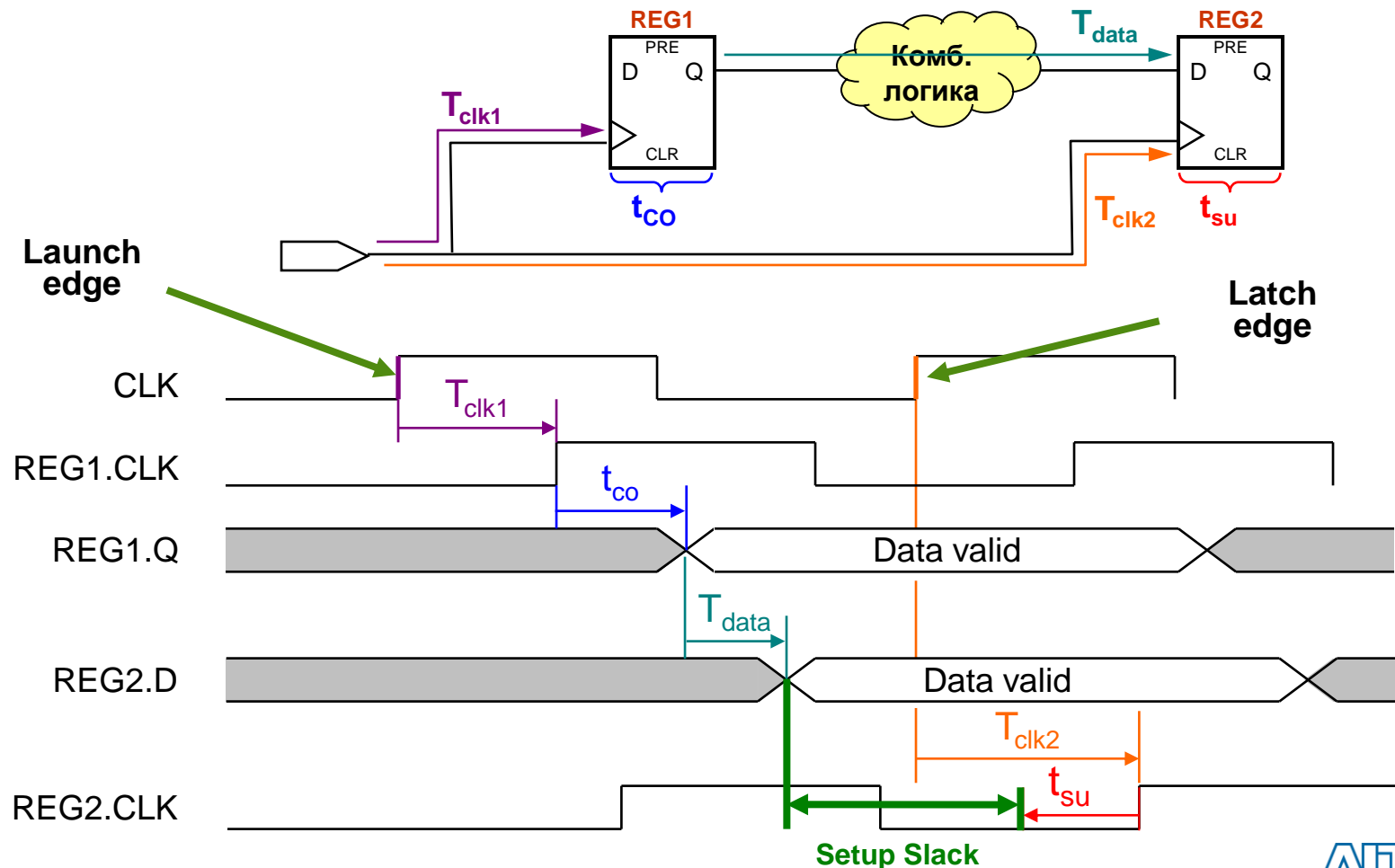
Data Required Time (Setup) = Clock Arrival Time -  $t_{su}$  - Setup Uncertainty

Data Required Time (Hold) = Clock Arrival Time +  $t_h$  + Hold Uncertainty

Clock Arrival Time  $\gg t_h, t_{su}$

# Setup Slack

- Запас предустановки – запас, с которым выполняется требование предустановки





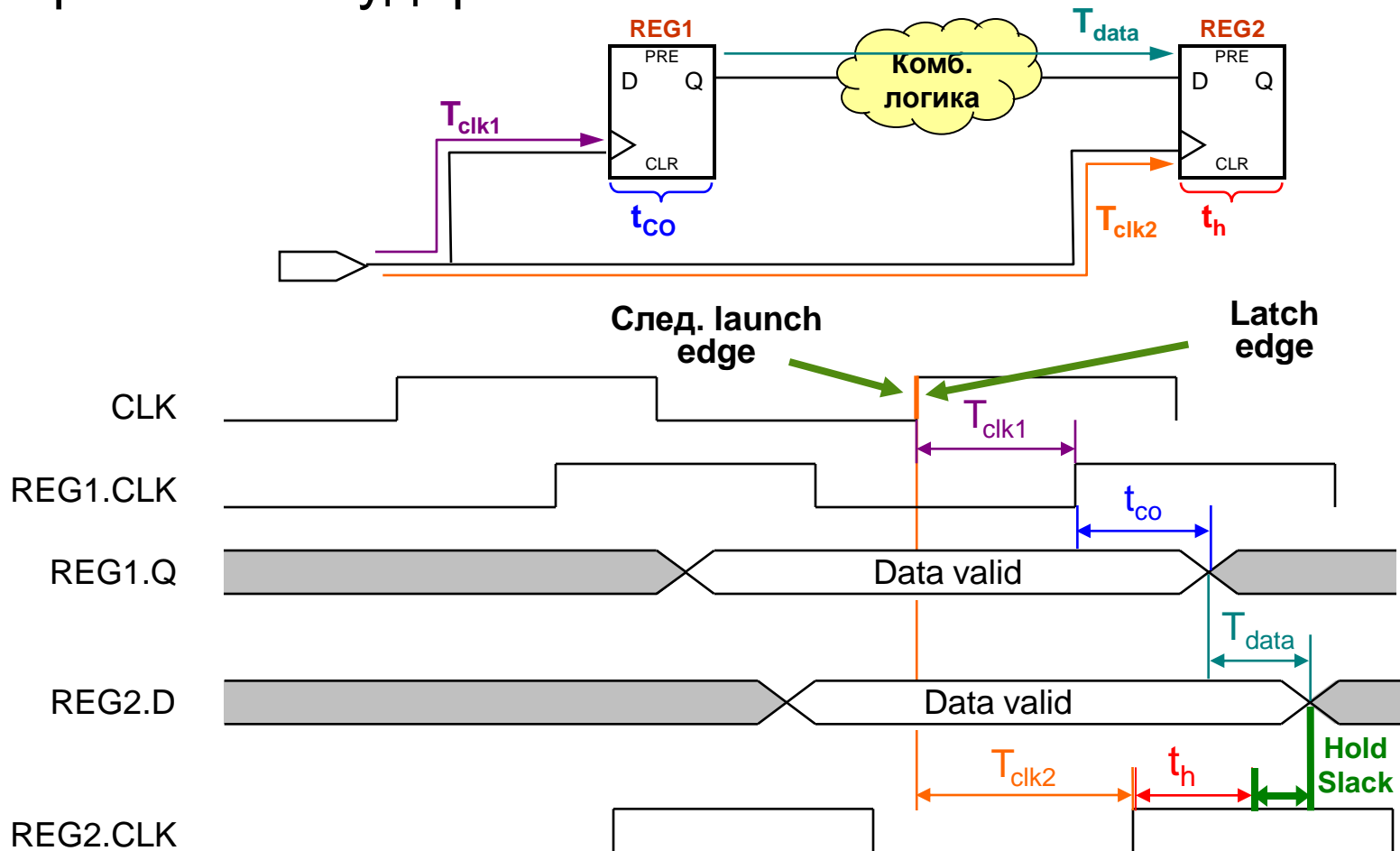
# Setup Slack

- Равен разнице между минимальным требуемым временем предустановки и максимальным временем прибытия данных.
- Positive slack – положительный запас, временные требования выполняются
- **Negative slack** – отрицательный запас, временные требования не выполняются

$$\text{Setup Slack} = \textit{Minimum Data Required Time (Setup)} \\ - \textit{Maximum Data Arrival Time}$$

# Hold Slack

- Запас удержания – запас, с которым выполняется требование удержания.



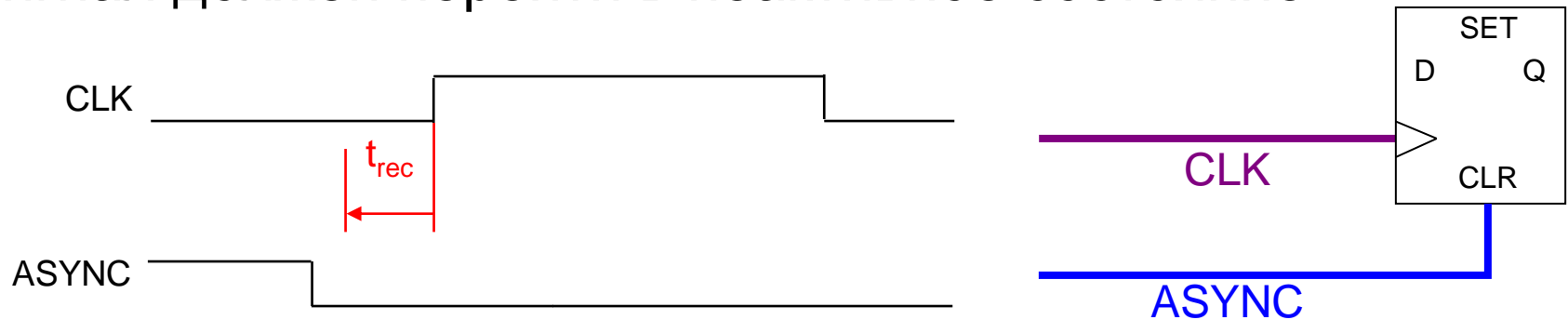
# Hold Slack

- Равен разнице между минимальным временем прибытия данных и максимальным требуемым временем удержания.
- Positive slack – положительный запас, временные требования выполняются
- **Negative slack** – отрицательный запас, временные требования не выполняются

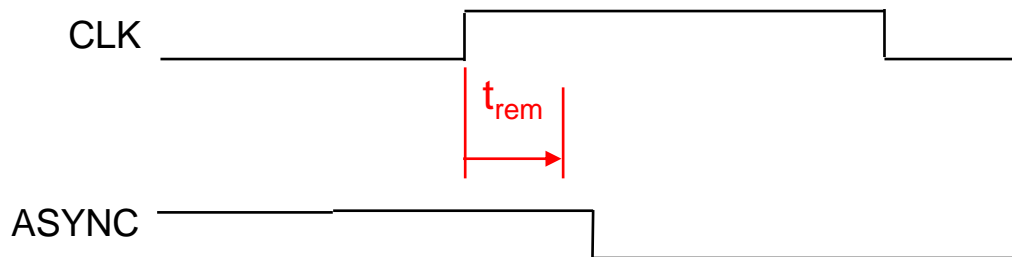
$$\text{Hold Slack} = \text{Minimum Data Arrival Time} - \text{Maximum Data Required Time (Hold)}$$

# Recovery & Removal

Recovery (время восстановления) - минимальное время до прихода тактового импульса, за которое асинхронный сигнал должен перейти в неактивное состояние

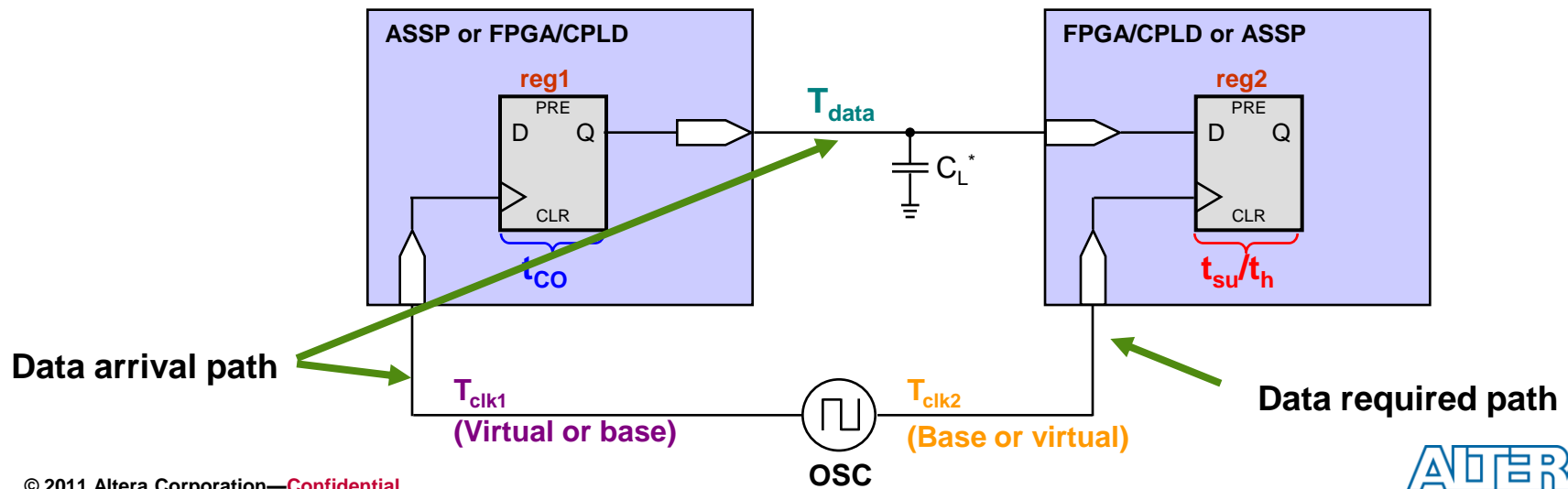


Removal (время снятия) - минимальное время после прихода тактового импульса, по истечении которого асинхронный сигнал может перейти в неактивное состояние



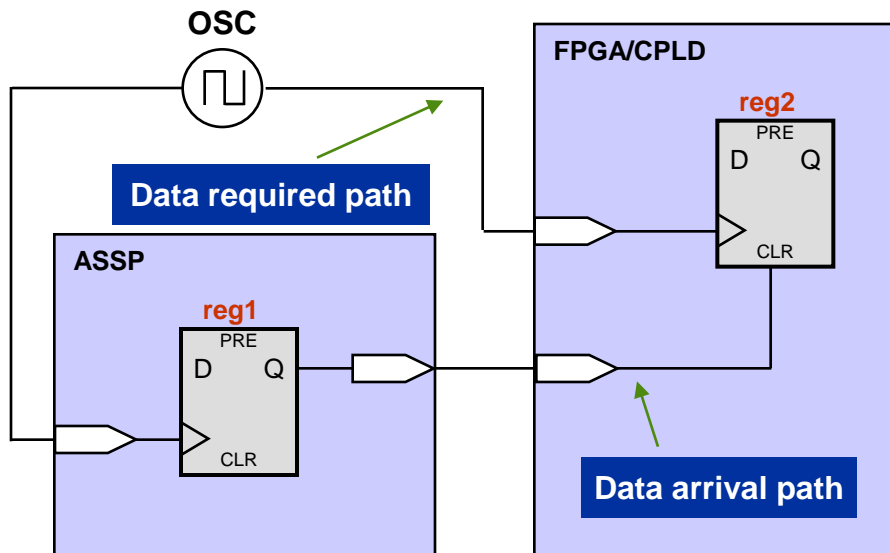
# Анализ ввода/вывода с общим тактовым импульсом

- Анализ временных характеристик по портам ввода/выводы в синхронном проекте использует те же уравнения для расчета запаса
  - Требуется учесть временные характеристики внешнего устройства и задержки на печатной плате
  - Рекомендуется определять виртуальные тактовые импульсы для задания внешних задержек по входам/выходам

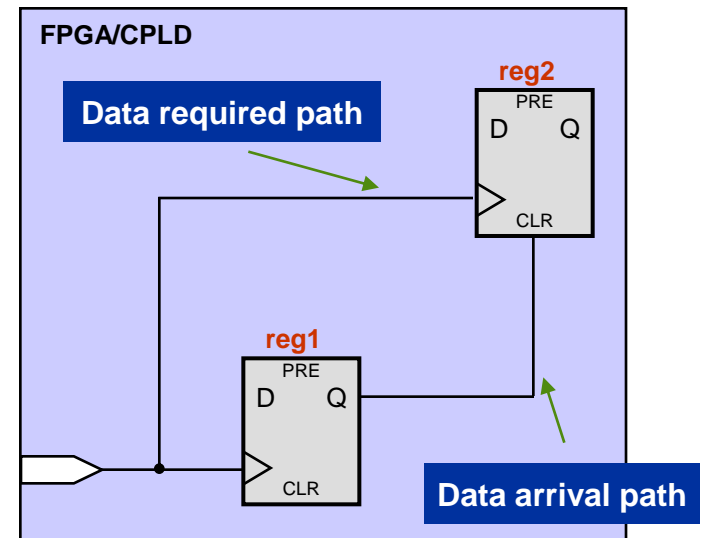


# Асинхронный и синхронные сигналы

- Асинхронный управляющий сигнал рассматривается как синхронный
  - Уравнения расчета запаса
    - Асинхронный путь рассматривается как путь прибытия данных
    - Recovery, аналогично предустановке, removal - удержанию



Пример 1



Пример 2

# Временные модели в Quartus II

- Quartus II по умолчанию оценивает временные характеристики устройства для двух точек PVT
  - Медленная модель (Slow Corner )
    - Медленное устройство при максимально допустимой температуре и минимальном напряжении питания
  - Быстрая модель (Fast Corner)
    - Быстрое устройство при минимально допустимой температуре и максимальном напряжении питания
- Времена предустановки должны выполняться в медленной модели
- Времена удержания должны выполняться в быстрой модели
- Третья модель (медленная при минимальной температуре) используется для устройств на основе технологии 65 нм и меньше (феномен температурной инверсии)

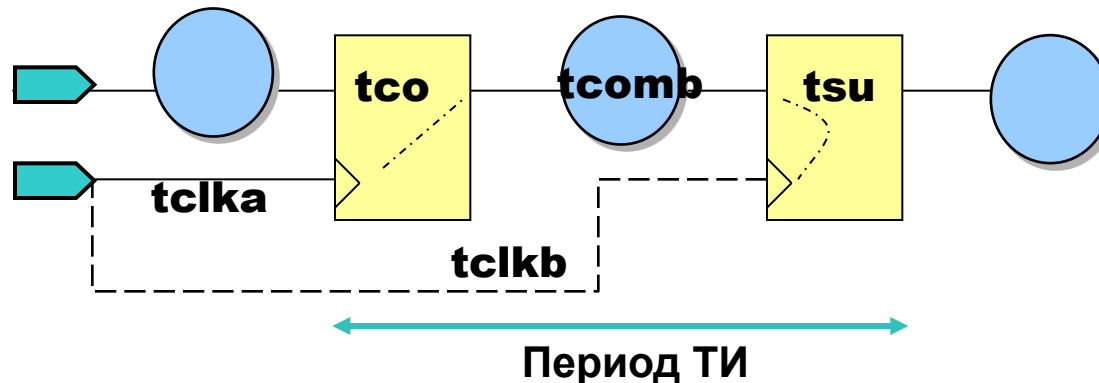
# Проектирование в Quartus II

Стандартные временные параметры



# Установка ТИ (fmax)

- Максимальная частота, с которой можно тактировать схему
  - Ограничивается цепью с наибольшей задержкой (критической цепью)

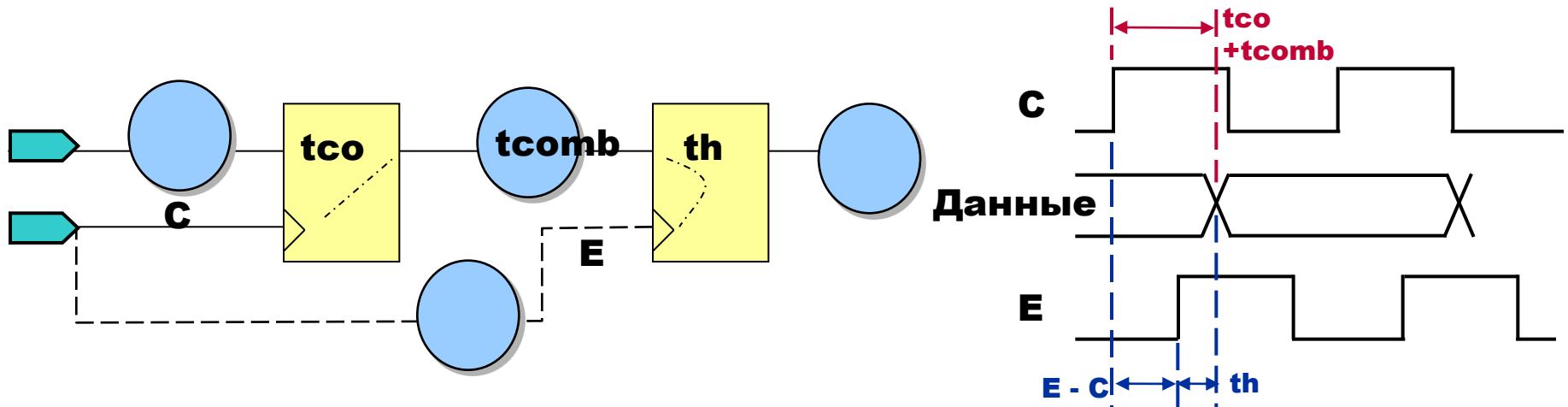


$$\text{Период ТИ} = tco + tcomb + tsu - (tclkb - tclka)$$

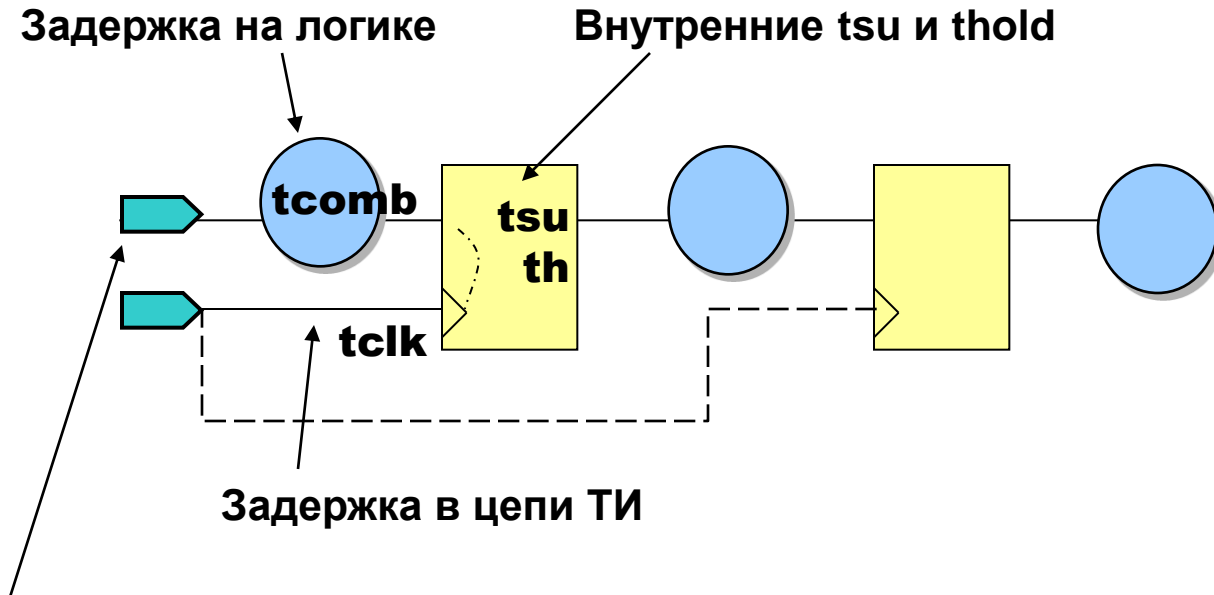
$$f_{max} = 1/\text{Период ТИ}$$

# Анализ удержания ТИ

- Проверяет выполнение внутреннего времени удержания регистра приемника
  - Серьезная ошибка - **схема не работоспособна**
- Проблема возникает, когда задержка в цепи данных меньше перекоса тактовых импульсов
  - Результат передачи ТИ не по глобальным шинам
  - Применение логики для формирования ТИ



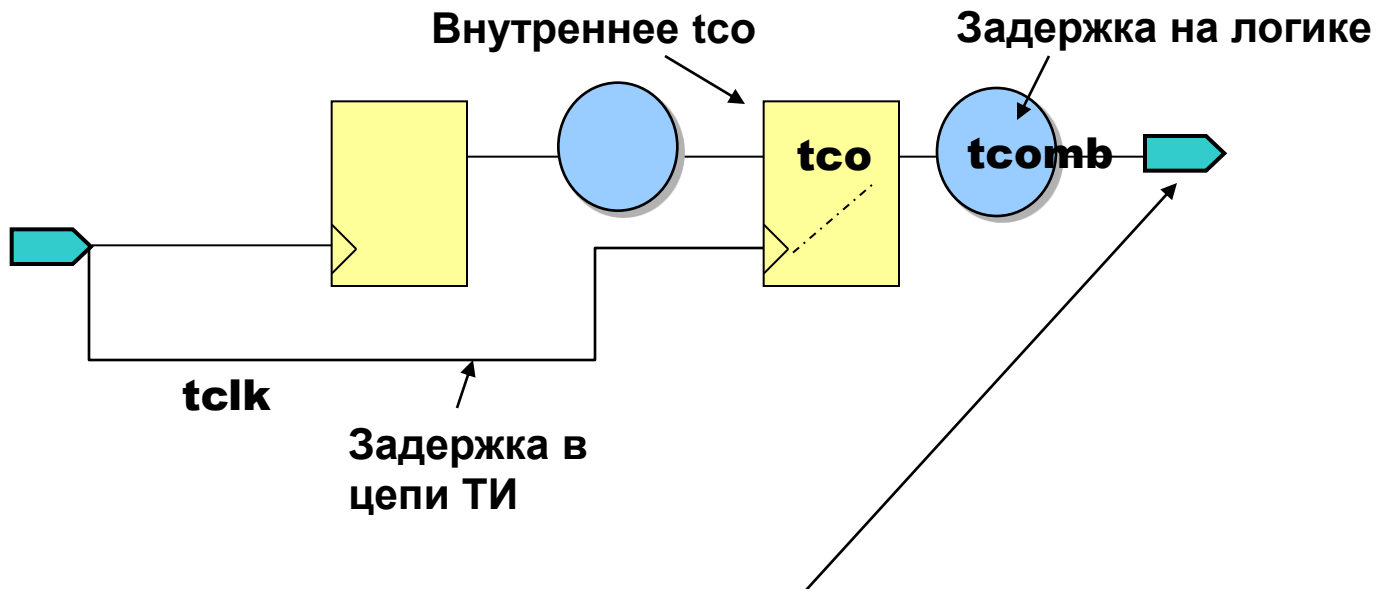
# Анализ времен установки и удержания



$$tsu = tcomb - tclk + tsu_{ВН}$$

$$th = tclk - tcomb + th_{ВН}$$

# Анализ задержки выхода по ТИ



$$tco = tclk + \text{внутр. tco} + tcomb$$