

Потактовая симуляция

Григорий Речистов

grigory.rechistov@phystech.edu

Вопросы

- Свойства «исполняющей» модели (процессор)
- Свойства «неисполняющей» модели
- Может ли процессор быть промоделирован «неисполняющим» образом?

На предыдущих лекциях

- Функциональная модель процессора
 - Один агент, события на каждом шаге
- DES
 - Много агентов, события «асинхронны», не на каждом шаге

Внутреннее устройство

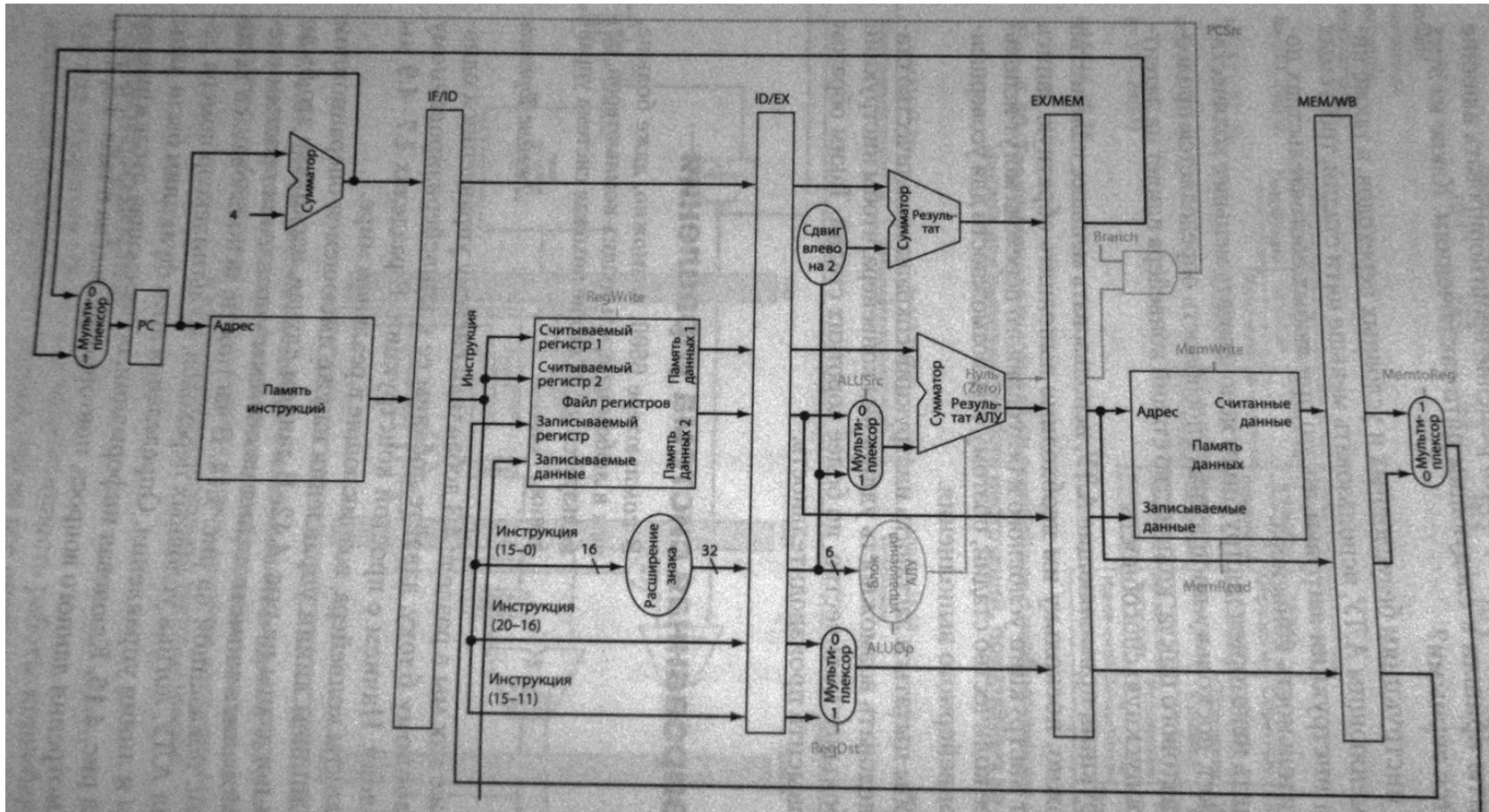
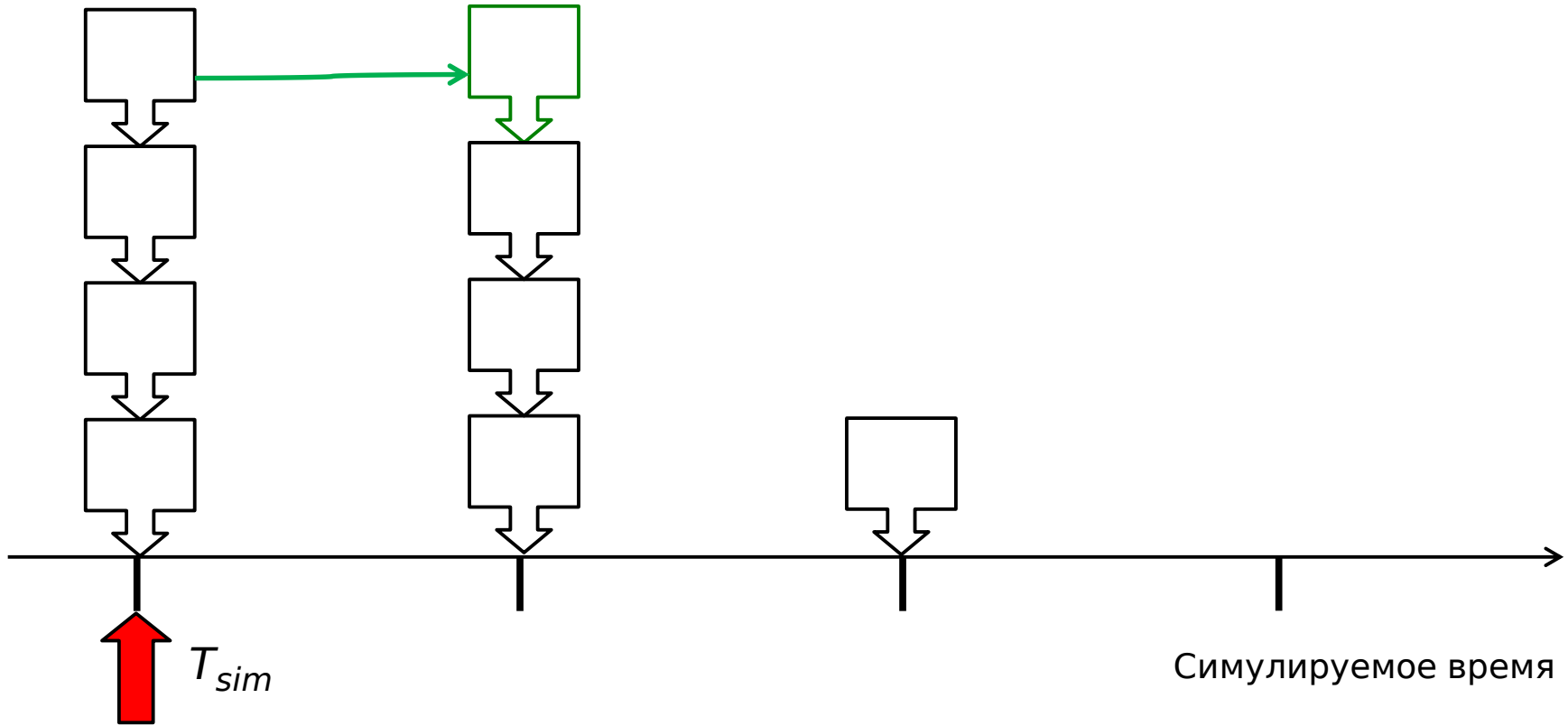


Иллюстрация взята из [1], стр. 398

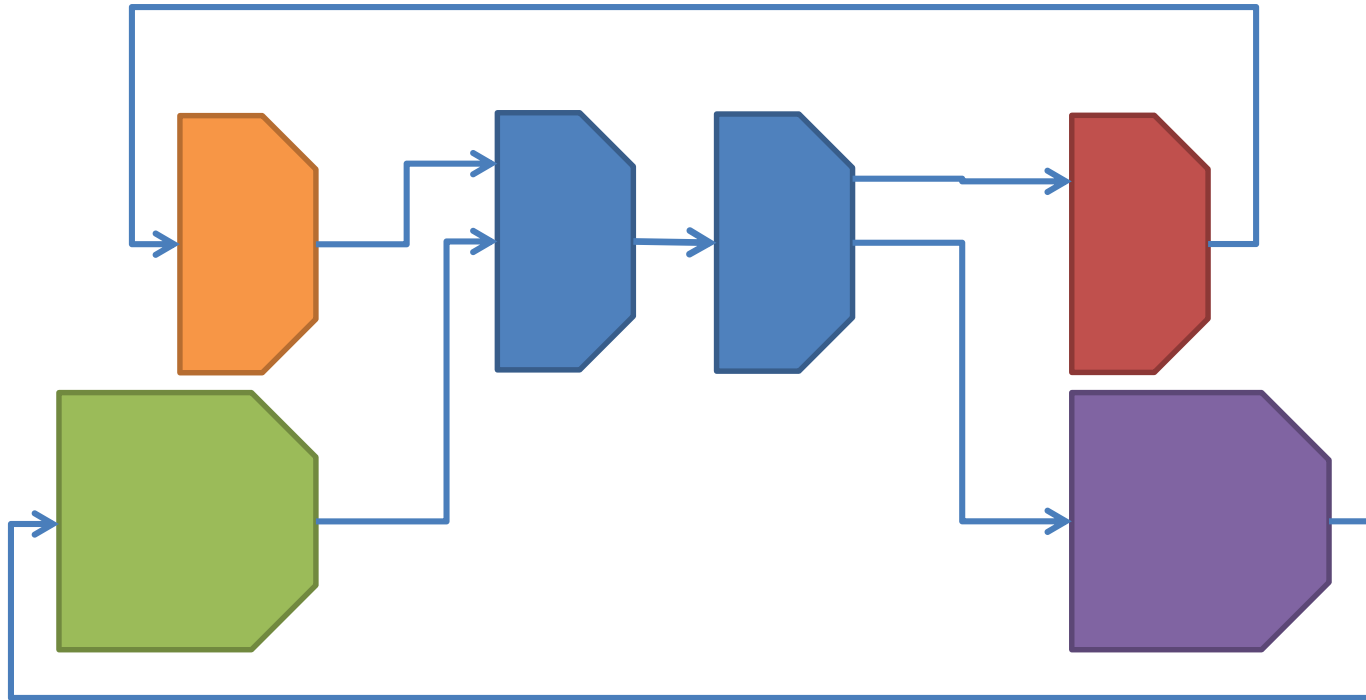
DES для потактовых моделей



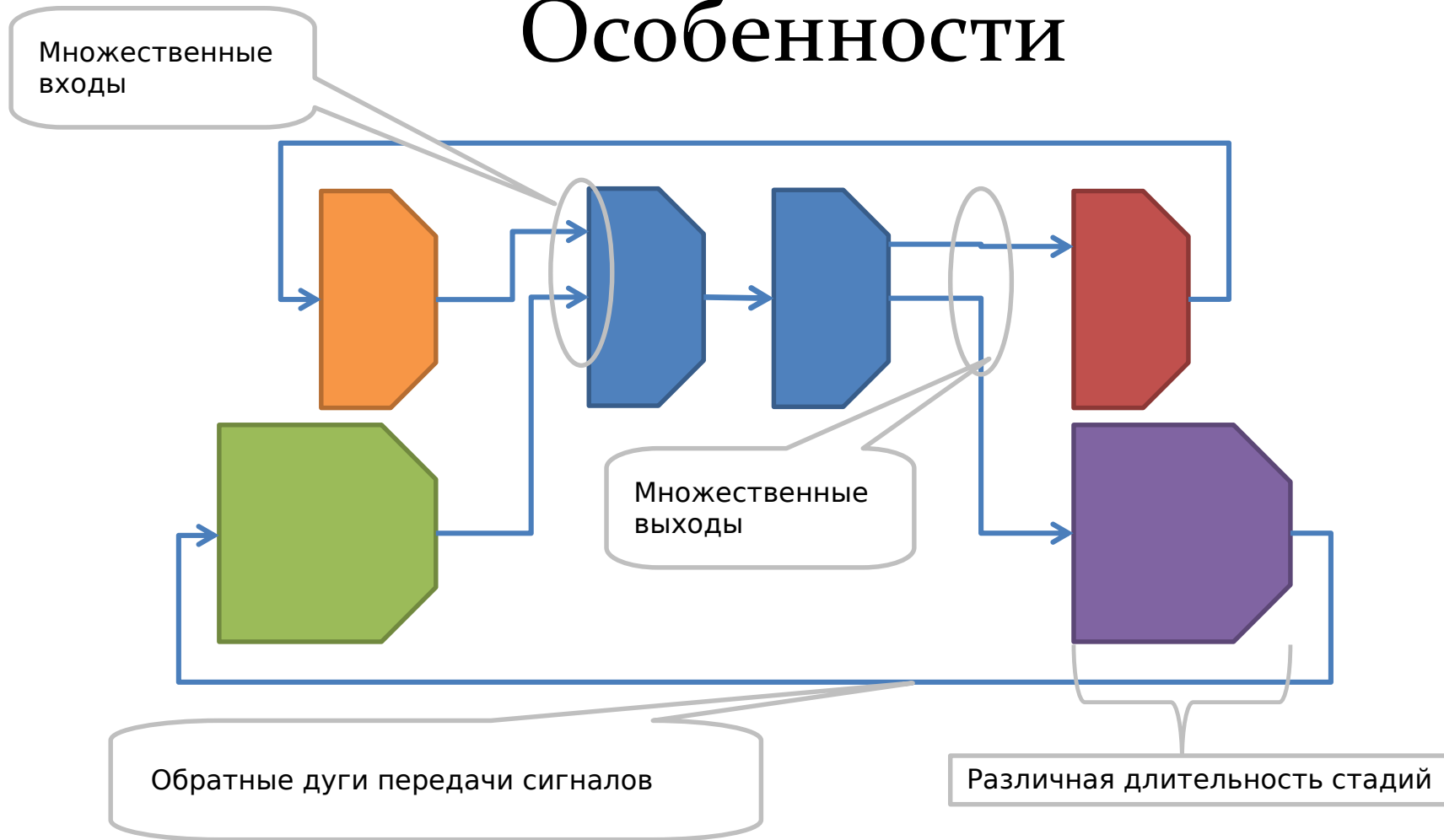
Проблемы

- DES – работает, но предоставляет не ту абстракцию
 - Хочется видеть «течение» данных между узлами
- Мы имеем систему, в которой
 - Много устройств
 - Много событий – на каждом такте (почти) все устройства меняют состояние и обмениваются данными

Моделируемая система



Особенности



Проблемы

- Длительность одной операции у разных узлов могут быть различными
 - Как проверять готовность «медленных» узлов?
- Результаты обработки данных должны появляться не ранее, чем на такте, следующим за текущим
 - Нельзя в произвольном порядке обновлять состояние блоков

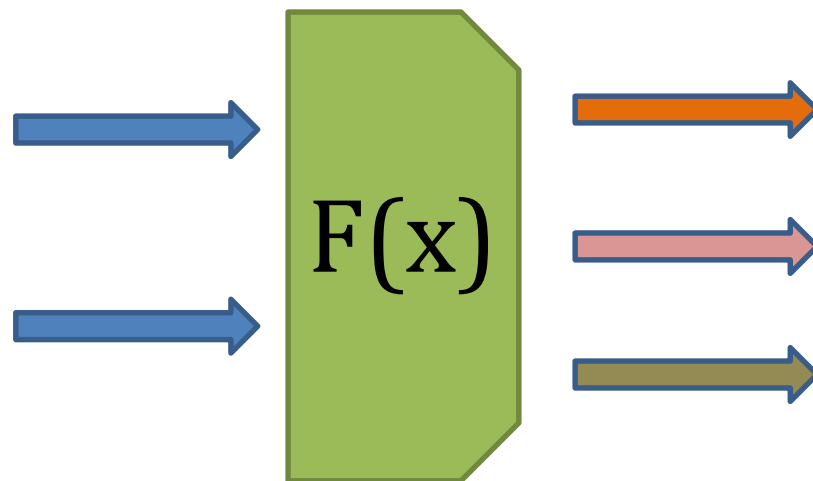
Абстрагируемся

Отделим:

- Функции узлов
- Время, затрачиваемое на их выполнение
- Внутреннее состояние узлов

Функциональный элемент

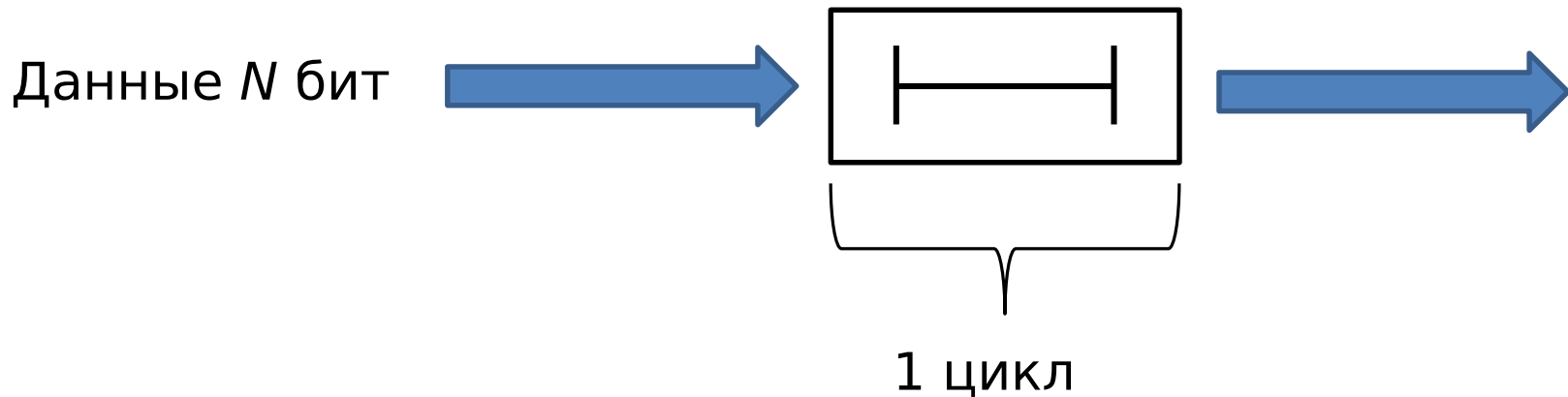
Результат готов «мгновенно» при наличии
ВХОДНЫХ ДАННЫХ



Порт

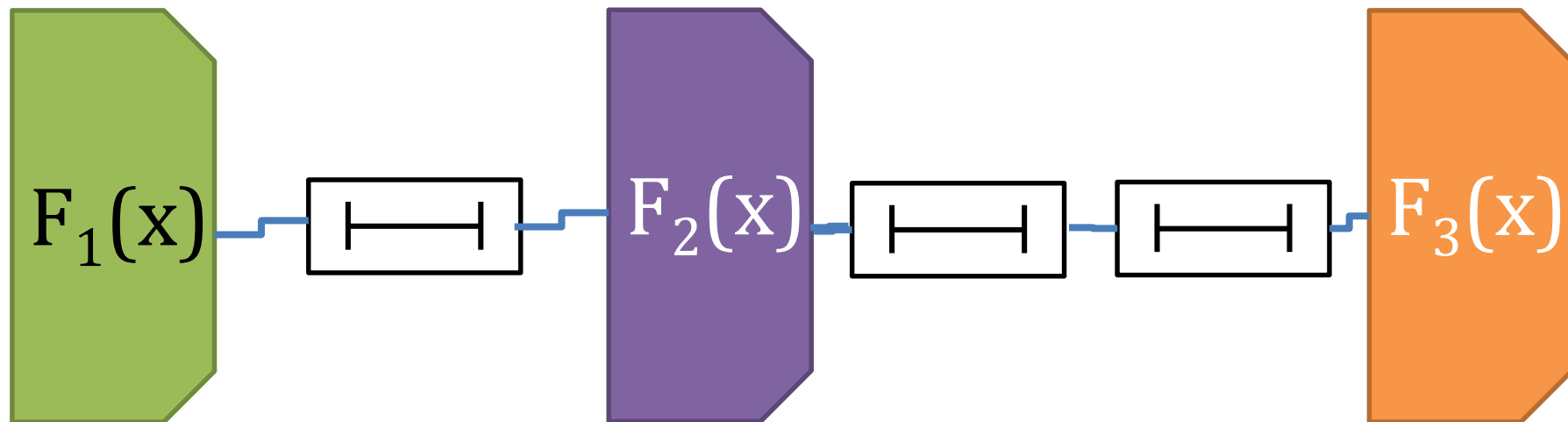
Очередь фиксированной задержки

Ширина N бит, задержка 1 такт

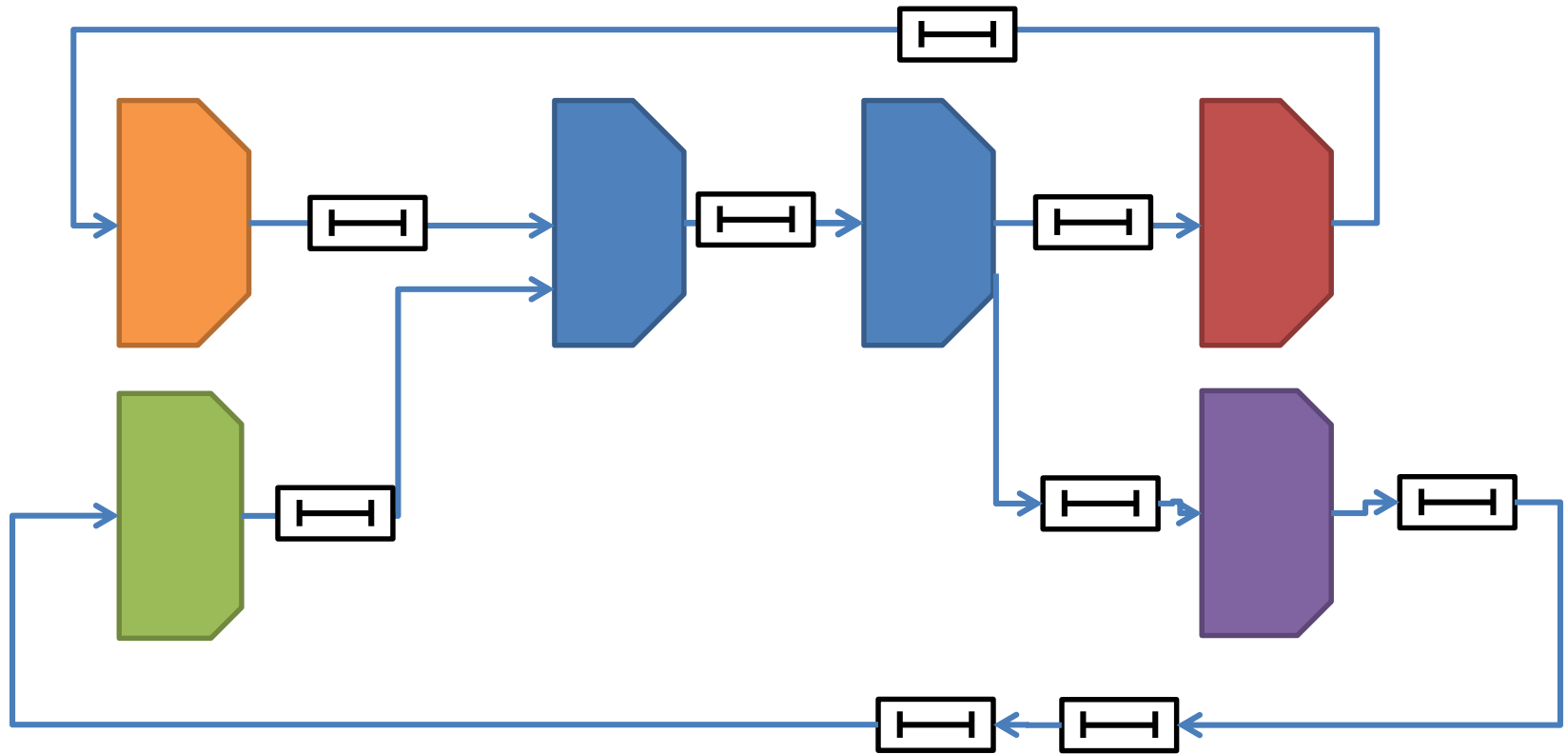


Правило

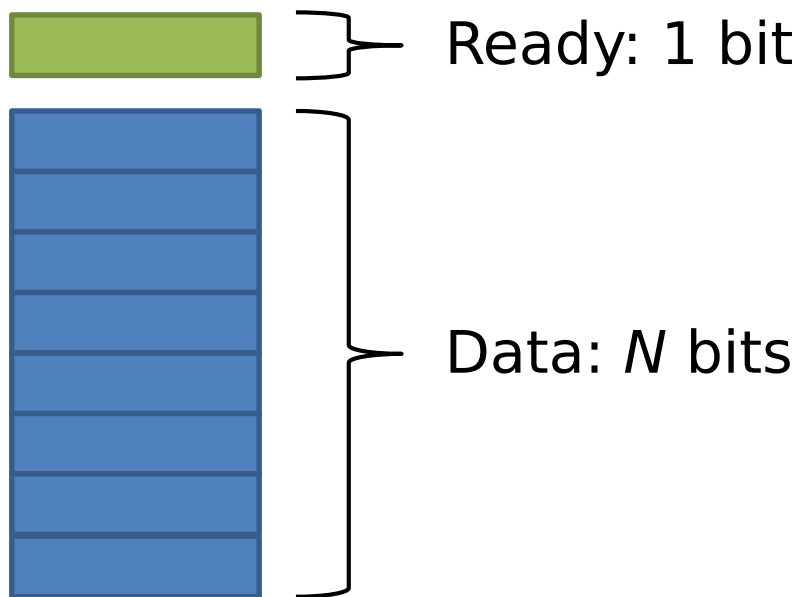
Функции не могут соединяться непосредственно друг с другом



Симуляция с портами



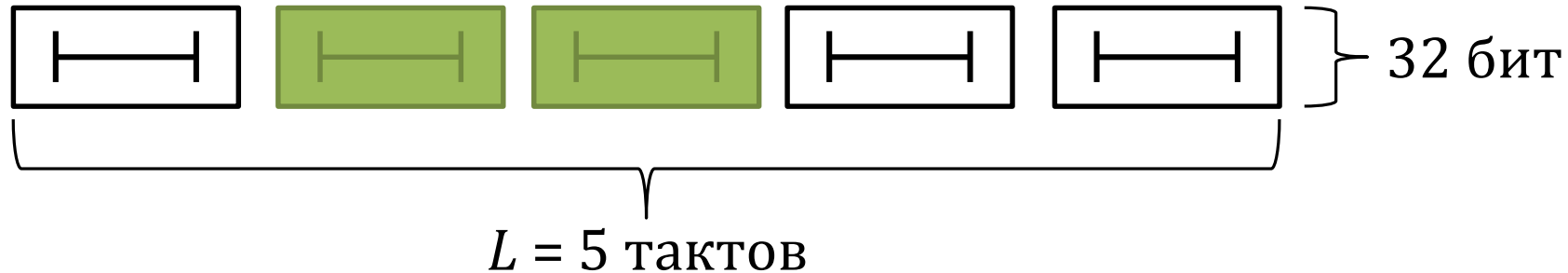
Как понять, готовы ли входные данные?



Полоса пропускания и латентность в терминах портов

В линии может быть больше одной транзакции в пути

$$Bw = 32 \text{ бит/такт}$$

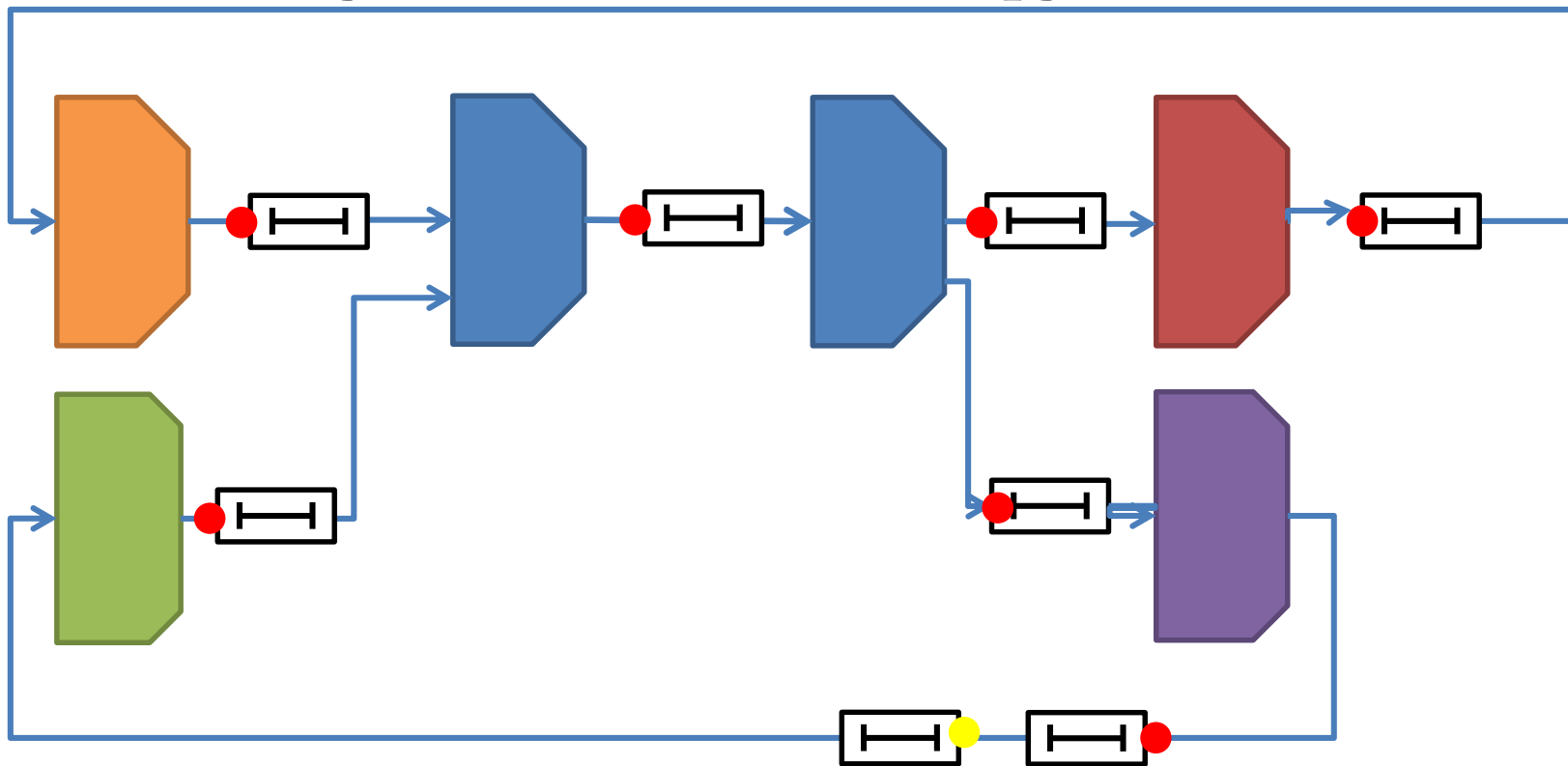


Как это работает?

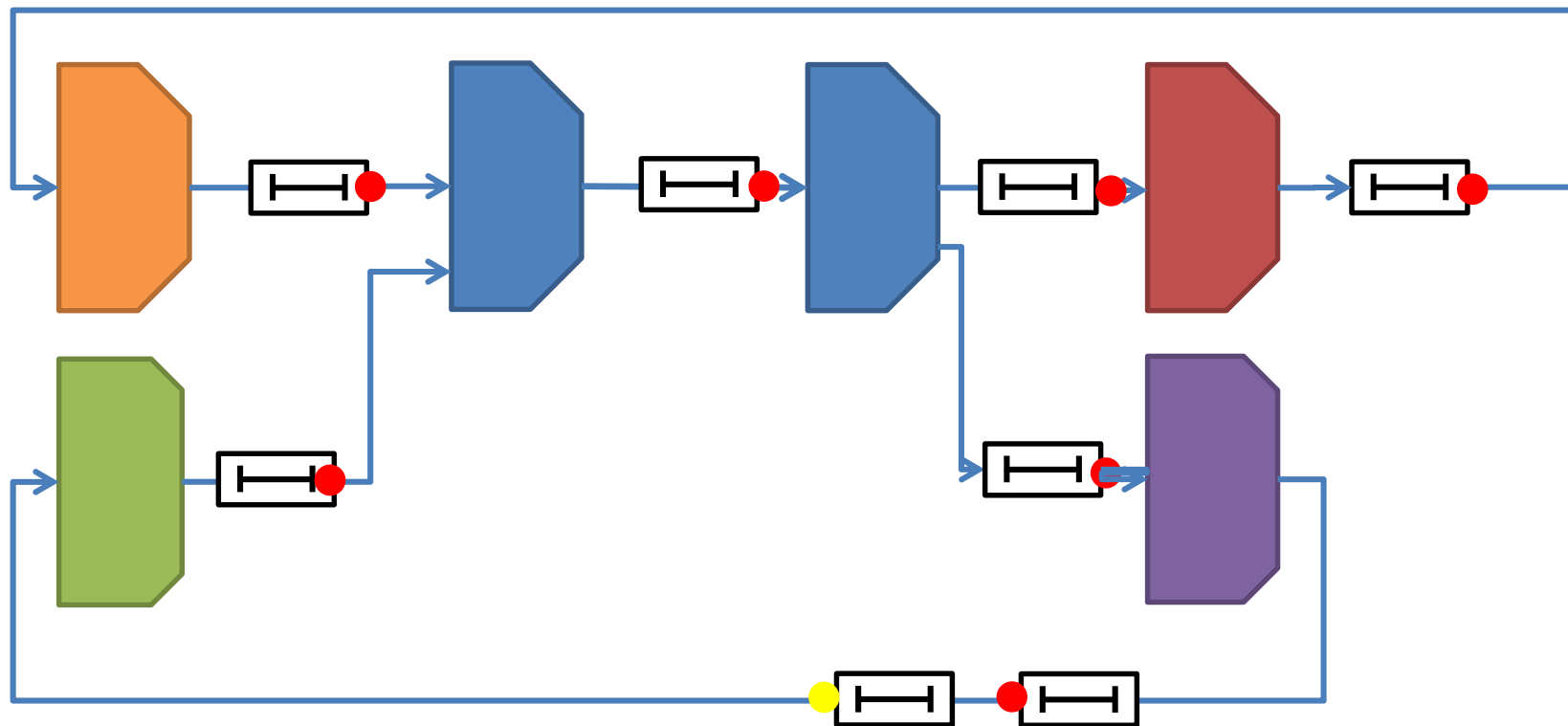
Раздельные стадии

- симуляция функций
- симуляция передачи результатов

Симуляция. Шаг № 1 – функции



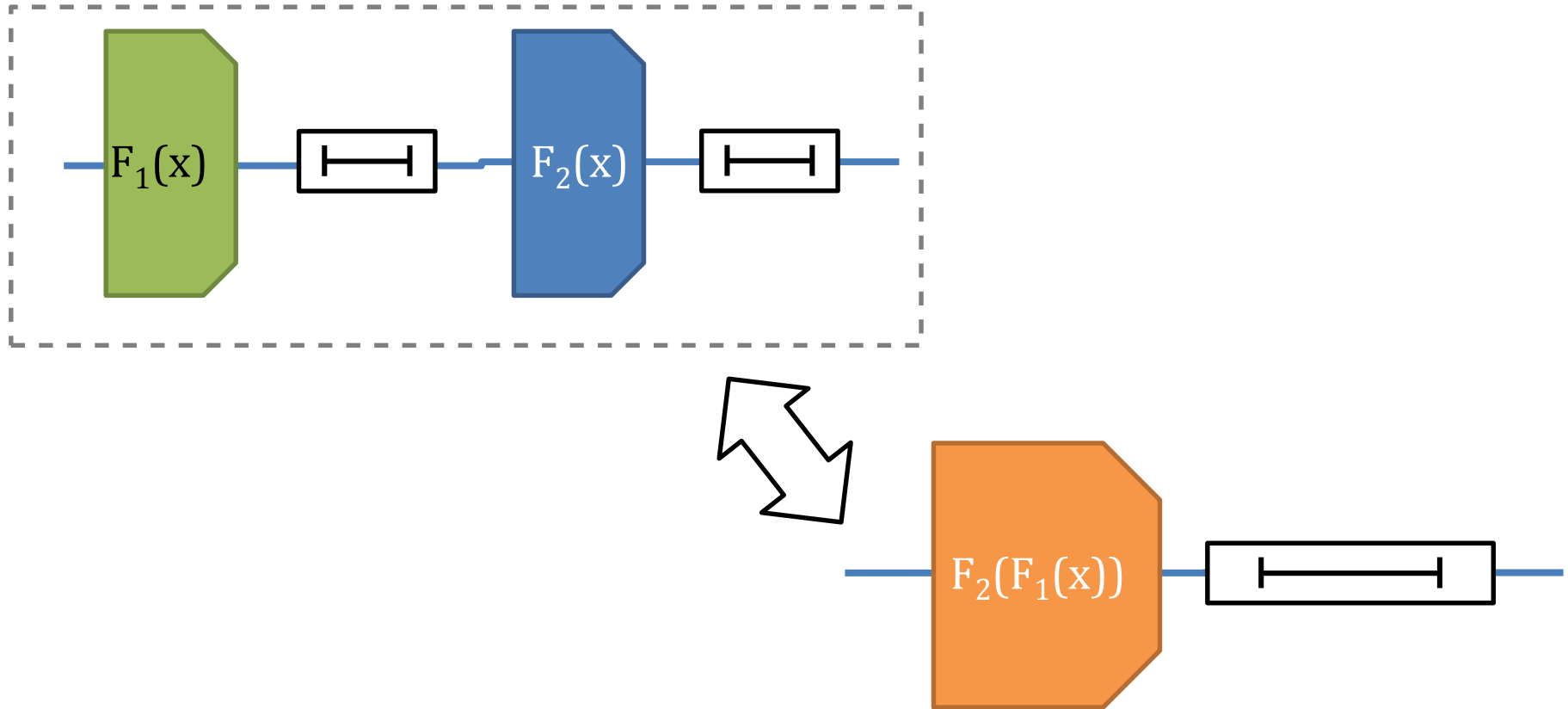
Симуляция. Шаг № 2 – передача данных



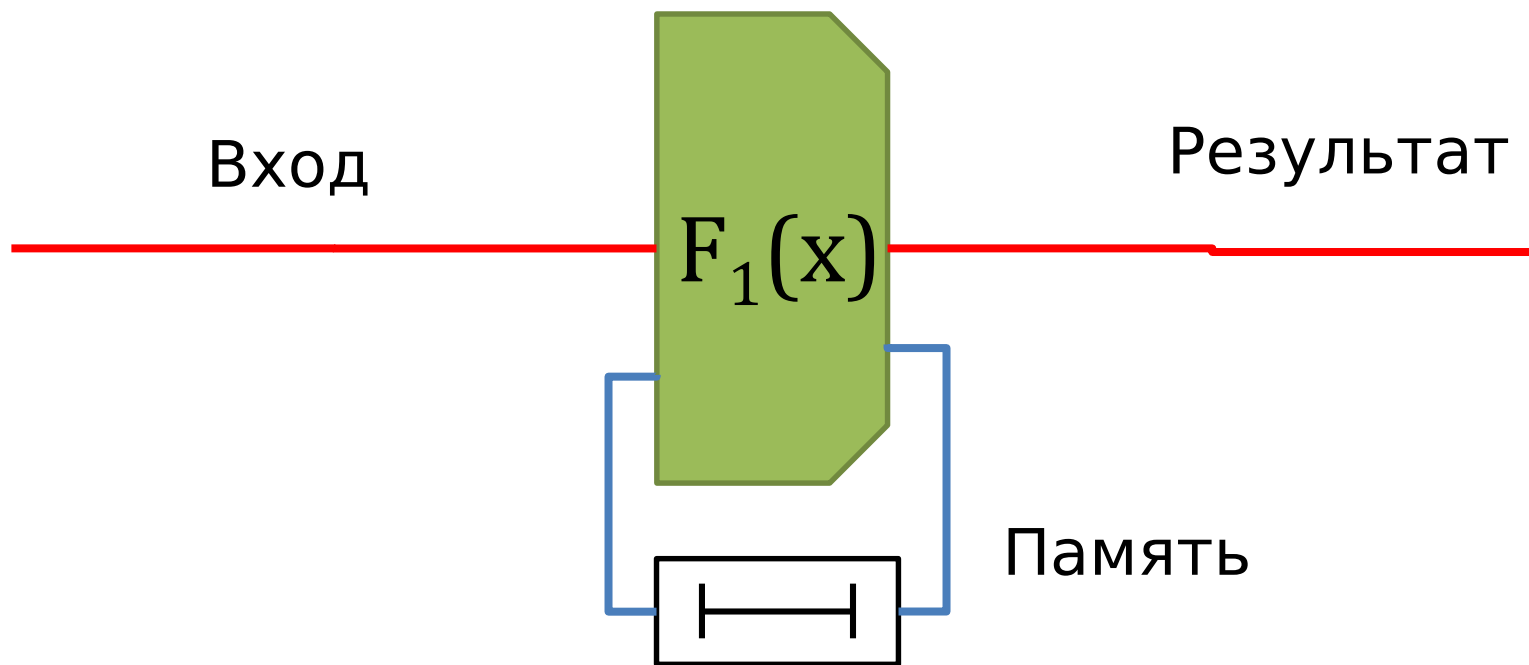
Преимущества

- Внутри одной стадии порядок вызова устройств неважен
- Гибкая расширяемость
- Изолированность узлов позволяет иметь параллельные реализации симулятора

Вложенность абстракции

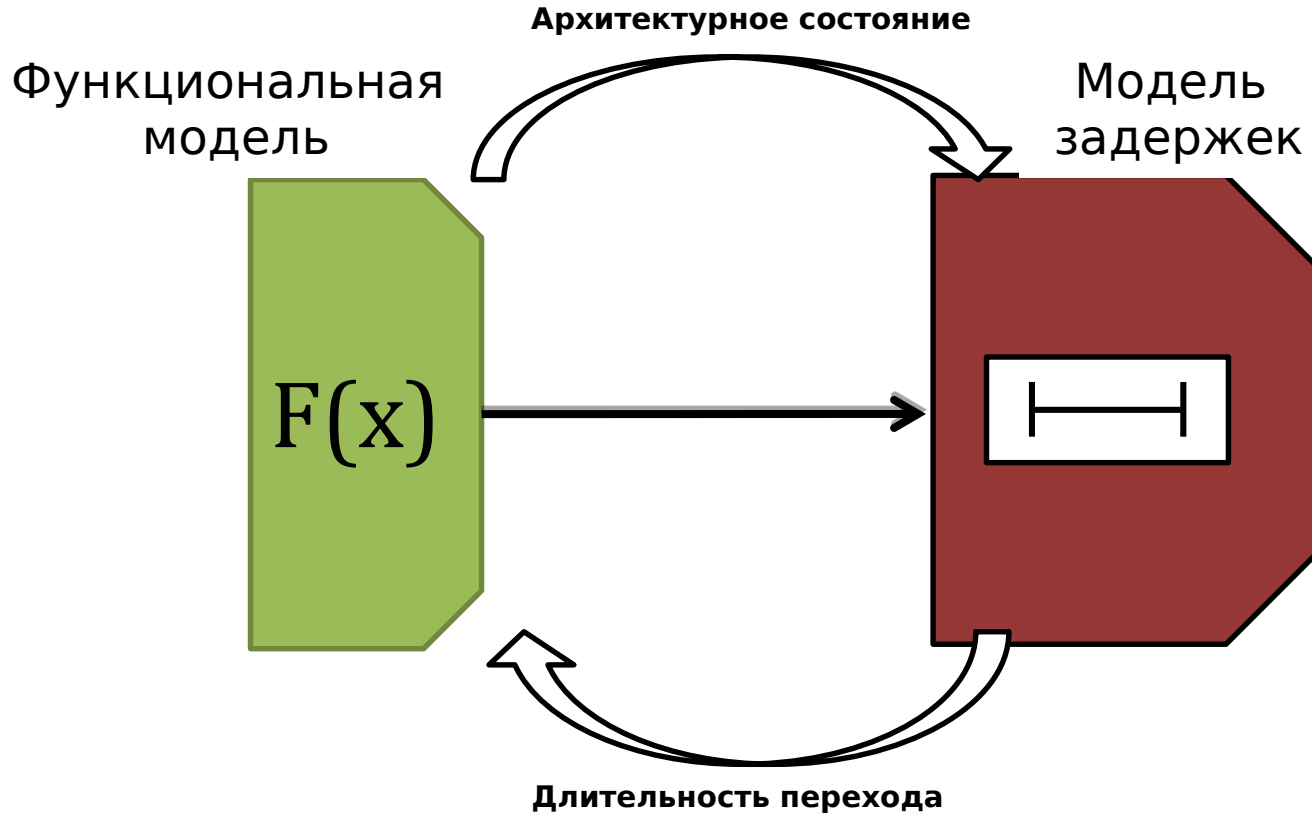


Могут ли функциональные элементы иметь память?



Комбинационные и последовательностные логические устройства

Модель задержек



Реализации на FPGA

- Схожая структура хозяина и моделируемой системы
- Возможность сделать параллельную реализацию модели
- Однако добиться соотношения «1 такт FPGA == 1 такт модели» затруднительно
- Модель может и не влезть на чип

Схема № 1

- На работу каждого узла отводится N тактов – по скорости самого медленного узла
- Фиксируем-ограничиваем скорость модели сверху

Схема № 1

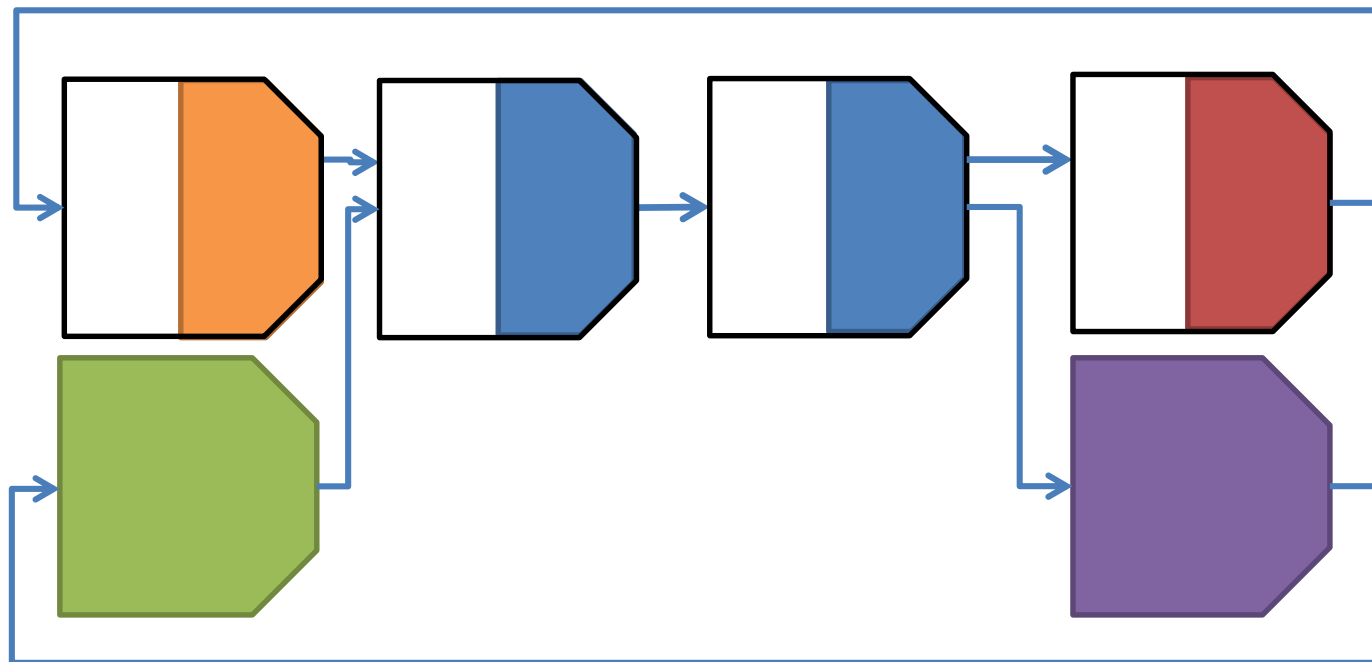


Схема № 2

- Каждый узел сообщает о готовности результатов центру синхронизации
- По готовности всех узлов продвигается симулируемое время
- Дополнительная сложность в синхронизации

Схема № 2

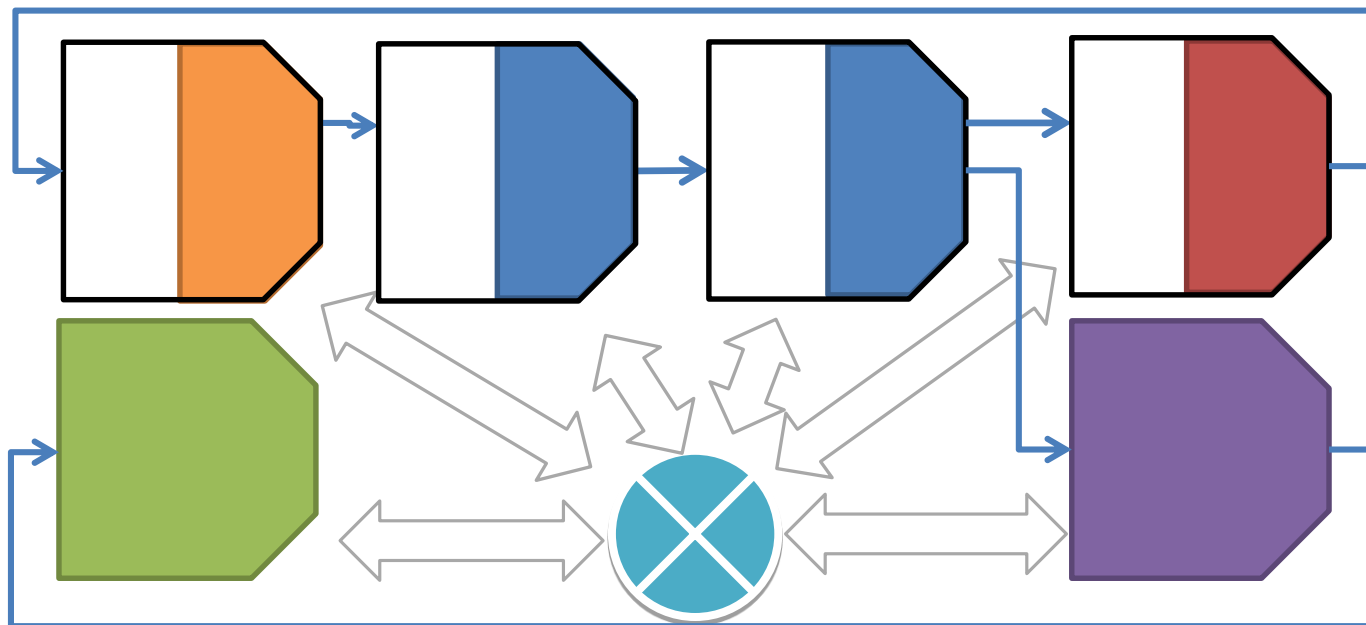


Схема № 3

- Используем порты
- Каждое устройство независимо от остальных
- Текущий такт передаётся как часть информации по портам

Рекомендуемая литература

1. Дэвид Паттерсон и Джон Хэннесси. **Архитектура компьютера и проектирование компьютерных систем**. 4-е изд. Питер, 2012.
2. Joel Emer, Pritpal Ahuja, Eric Borch, Artur Klauser, Chi-Keung Luk, Srilatha Manne, Shubhendu S. Mukherjee, Harish Patil, Steven Wallace, Nathan Binkert, Roger Espasa, Toni Juan. **Asim: A Performance Model Framework** // Computer 35 (2002), p. 68–76.
3. Ю. В. Байда. Методы разработки и тестирования аппаратных потактовых моделей микропроцессоров на программируемых логических интегральных схемах. Дисс. к.т.н. 2013

На следующей лекции:

Параллельная симуляция

Консервативные схемы

Спасибо за внимание!

Все материалы курса выкладываются на сайте лаборатории:
http://iscalare.mipt.ru/material/course_materials/

Замечание: все торговые марки и логотипы, использованные в данном материале, являются собственностью их владельцев. Представленная здесь точка зрения отражает личное мнение автора, не выступающего от лица какой-либо организации.