

# **СТРУКТУРНЫЙ СИНТЕЗ И ПЛАНИРОВАНИЕ ВЫЧИСЛЕНИЙ В УПРАВЛЯЮЩИХ СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ**

Кафедра АСВК,  
Балашов В.В.



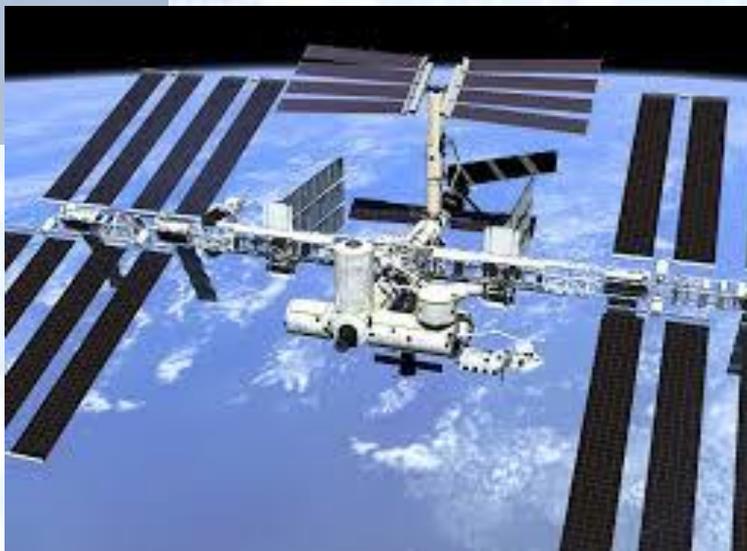
# Аннотация

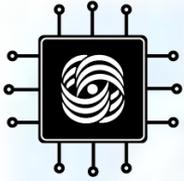
- Управляющие системы реального времени – что это такое, в чем специфика, примеры
- Задачи планирования вычислений в системах реального времени
- Чуть поподробнее:
  - примеры систем
  - примеры задач
- Что предлагает кафедра АСВК по данной тематике
  - как устроена курсовая работа
  - направления исследований
  - поддержка в учебных курсах
- Профессиональные перспективы



# Системы реального времени

Сложные технические объекты управляются  
распределёнными компьютерными  
системами



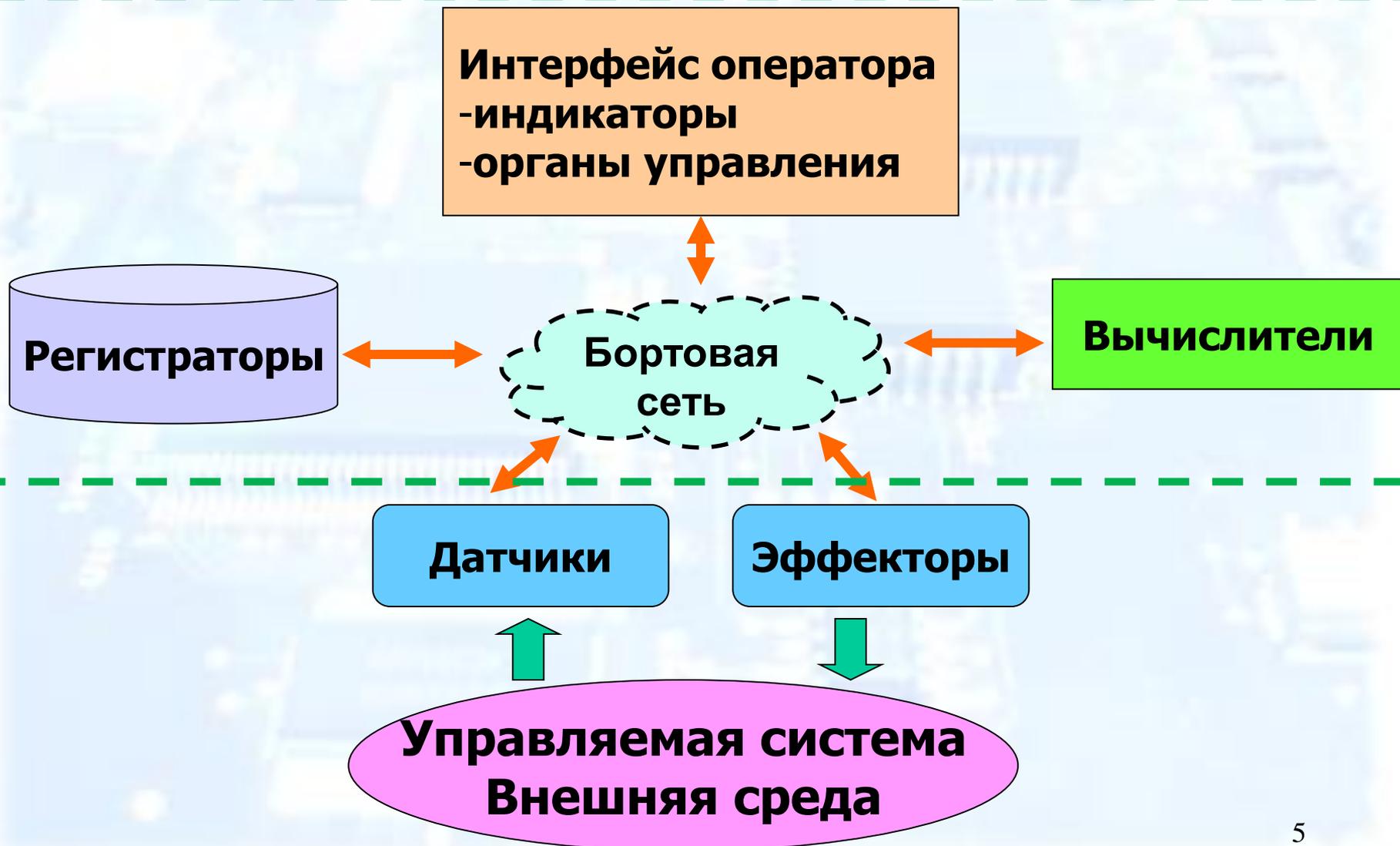


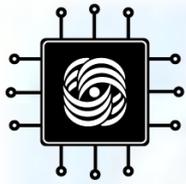
# ИУС РВ

- Информационно-управляющая система – вычислительная система в составе объекта, обеспечивающая:
  - управление функционированием объекта и мониторинг его состояния
  - взаимодействие между объектом и оператором
- Функционирует в реальном времени
  - рассчитать результат *правильно и вовремя*
- Где применяются
  - автоматизация производства, энергетика, наземный транспорт, авиация/космос, «умный» дом

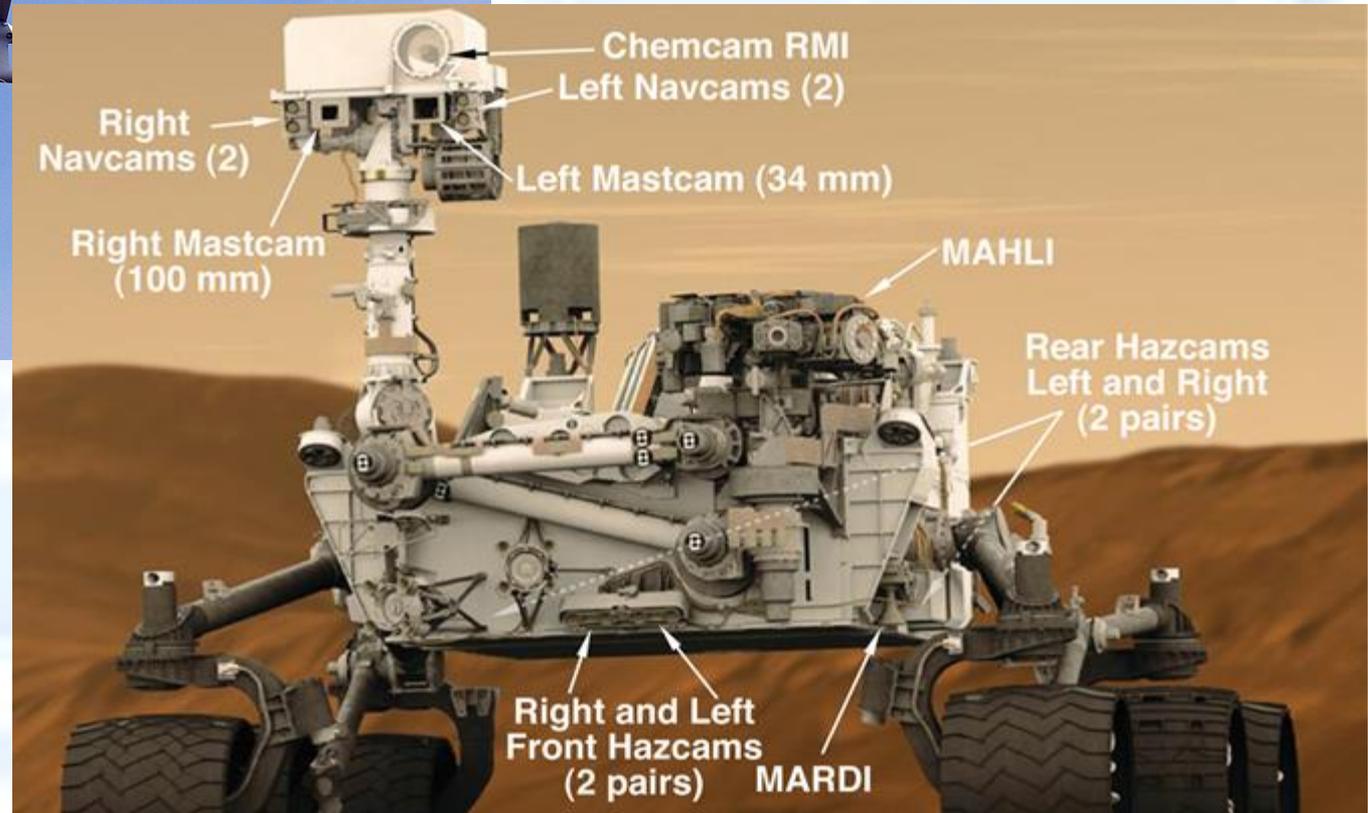
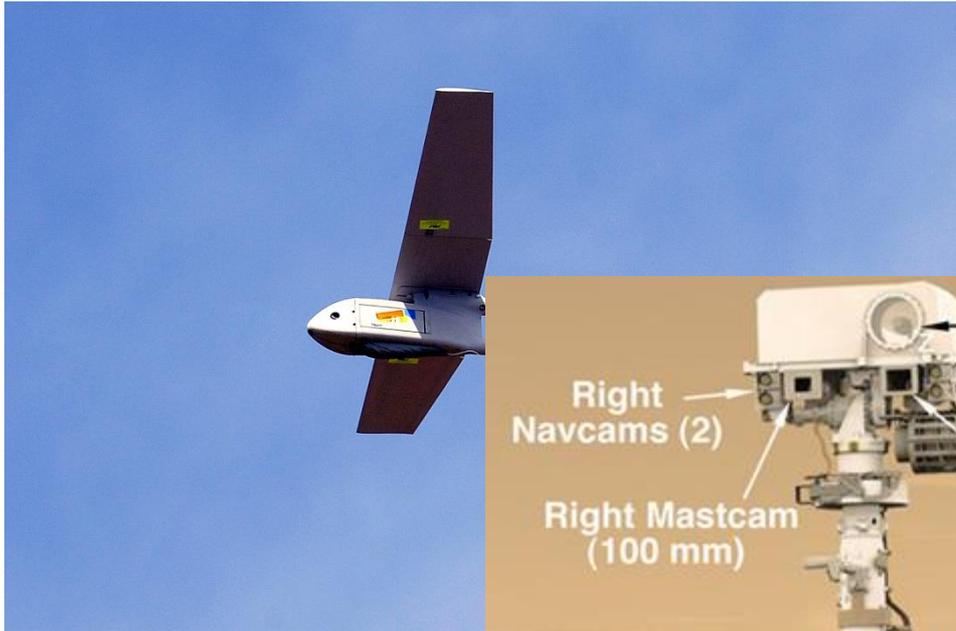


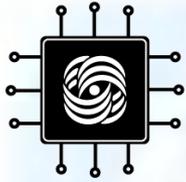
# Состав управляющей системы





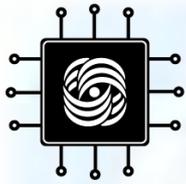
# Прикладная нагрузка





# Специфика управляющих систем

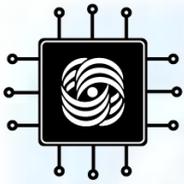
- Работа в реальном времени
  - ориентация на «наихудшие» случаи
- Непрерывное функционирование
- Параллелизм
  - управлять одновременно многим
- Интеграция с управляемой системой
- Критичность для управляемой системы
  - высокая цена ошибки
- Устойчивость к сбоям
- Ограниченное участие оператора
- Предсказуемое поведение
- «Экстремальные» условия работы
- Ограничения по ресурсам
- Координация между ИУС взаимодействующих объектов



# Работа в реальном времени

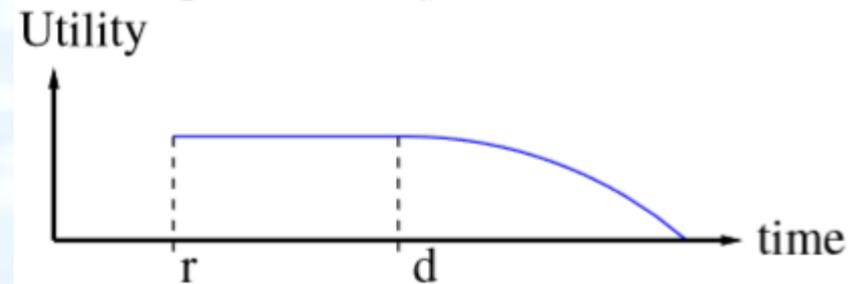
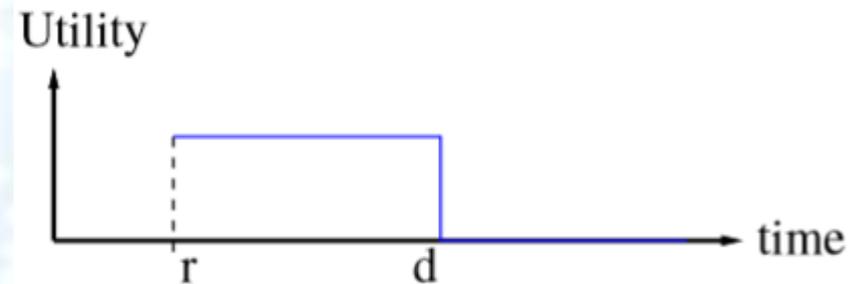
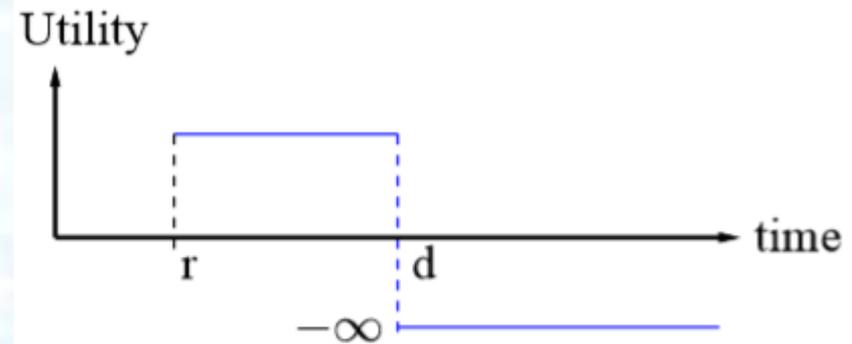
- Реагирующая система (reactive system) – ВС, функционирующая в постоянном взаимодействии с внешней средой и отвечающая на внешние воздействия в темпе, определяемом внешней средой
- Реакция на каждое воздействие должна укладываться в *директивный срок*

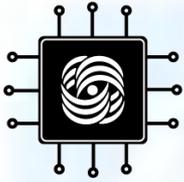




# Градации требований реального времени

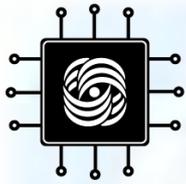
- Жёсткое (hard) РВ: нарушение ДС приводит к фатальным последствиям для управляемой системы (управление полётом)
- Промежуточный вариант (firm): нарушение ДС приводит к бесполезности результата, без фатальных последствий
- Мягкое (soft) РВ: нарушение ДС приводит к постепенному снижению ценности результата (автомобильный навигатор)





# Распространённые заблуждения

- Работа в реальном времени = быстрая работа
  - своевременность важнее быстродействия
  - предсказуемость и надёжность важнее быстродействия
- Рост производительности процессоров решит все проблемы с реальным временем
  - современные высокопроизводительные процессоры быстры в «среднем» случае, а для РВ критичен наихудший случай
  - тонкая технология производства => ненадёжность в экстремальных условиях
  - источники быстродействия современных процессоров слишком непредсказуемы
- Бессмысленно говорить о работе в реальном времени, если аппаратура может дать сбой
  - постепенная деградация функциональности
  - реконфигурируемость, «сбойные» режимы
- Разработка систем реального времени – чистая инженерия, здесь нет науки
  - об этом чуть позже...

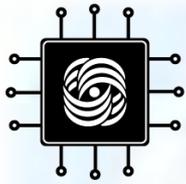


# Примеры управляющих систем

Functions by embedded processing:

- ABS: Anti-lock braking systems
  - ESP: Electronic stability control
  - Airbags
  - Efficient automatic gearboxes
  - Theft prevention with smart keys
  - Blind-angle alert systems
  - ... etc ...
- 
- Multiple networks
  - Multiple networked processors

© P. Marvedel, 2011

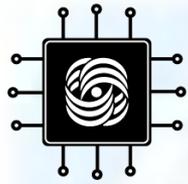


# Примеры управляющих систем

- Flight control systems,
- anti-collision systems,
- pilot information systems,
- power supply system,
- flap control system,
- entertainment system,
- ...



© P. Marvedel, 2011



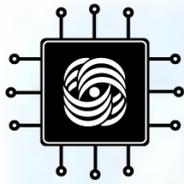
# Примеры управляющих систем



## Networked computer system

- Controlling arms & tools
- Navigating the forest
- Recording the trees harvested
- Crucial to efficient work

“Tough enough to be out in the woods”



# 産業用ASU TP

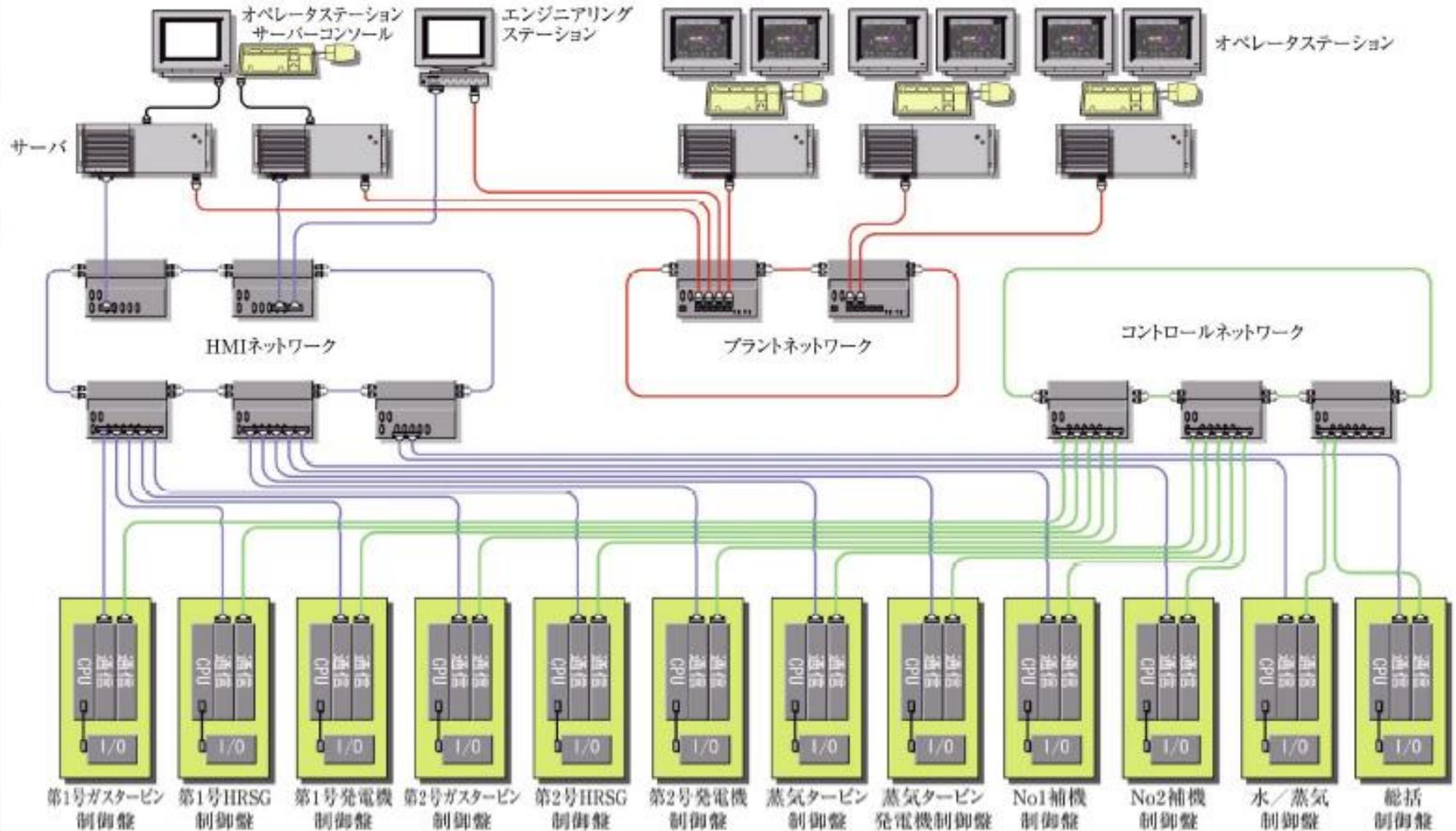
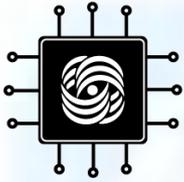


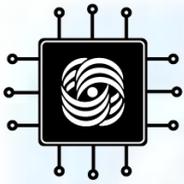
図2 中型発電制御装置のシステム構成例

Fig.2 System configuration of medium class gas turbine power plant



# Математические задачи планирования вычислений

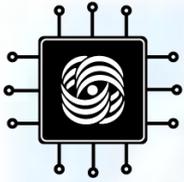
- Выбор (синтез) оптимальной конфигурации ИУС РВ
  - требования реального времени
  - требования надёжности
  - ограничения по ресурсам
- Распределение вычислительной нагрузки по процессорам
- Построение расписания вычислений
- Построение расписания обмена данными
- Конфигурирование коммутируемой среды обмена данными
- Динамическое перераспределение потоков данных и вычислительной нагрузки при сбоях системы и при изменениях состава нагрузки



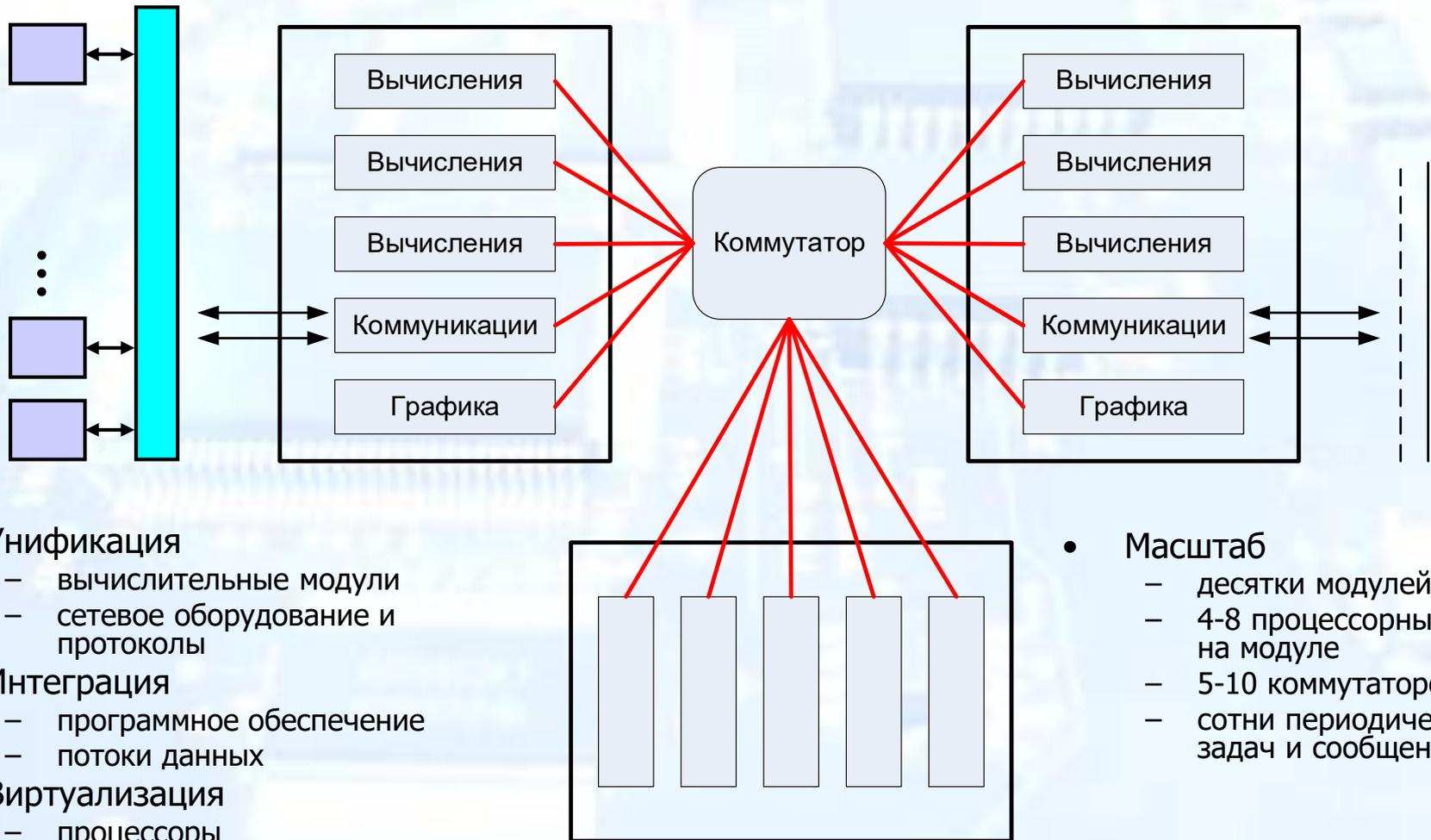
# Сочетание математики и программирования

## *Курсовая работа:*

- Практическая задача, возникающая при разработке реальных систем
- Математическая формулировка задачи
- Обзор известных методов решения похожих задач
- Построение алгоритма решения задачи
- Исследование алгоритма
- Реализация программной системы

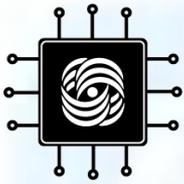


# ИУС РВ с модульной архитектурой



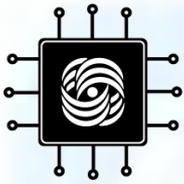
- Унификация
  - вычислительные модули
  - сетевое оборудование и протоколы
- Интеграция
  - программное обеспечение
  - потоки данных
- Виртуализация
  - процессоры
  - память
  - сеть

- Масштаб
  - десятки модулей
  - 4-8 процессорных ядер на модуле
  - 5-10 коммутаторов
  - сотни периодических задач и сообщений



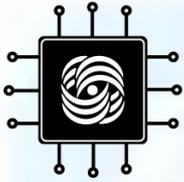
# Организация вычислений в модульных ИУС РВ

- Логически единый распределенный вычислитель
  - унифицированные модули нескольких типов
  - единая архитектура процессоров
  - унифицированный программный интерфейс
- Разделение вычислительных ресурсов между ПО различных подсистем
- Проблемы:
  - изоляция по памяти
  - разделение процессорного времени
- Решение:
  - каждой подсистеме → раздел
  - окна выполнения разделов



# Задача планирования: ВХОДНЫЕ ДАННЫЕ

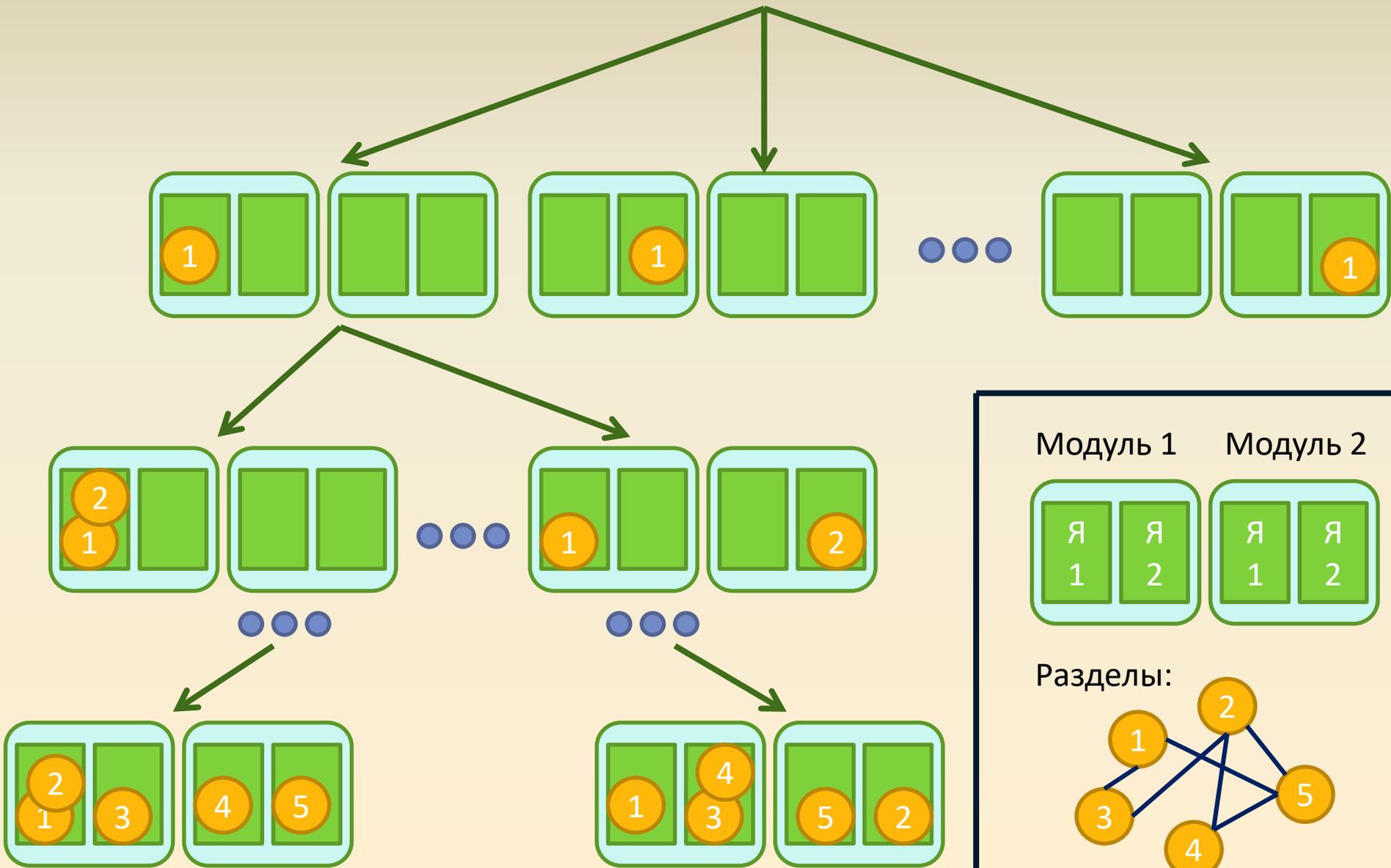
- Описание системы
  - набор модулей
  - модуль → набор и типы процессорных ядер
  - ядро → верхняя граница загрузки
- Описание рабочей нагрузки
  - наборы задач, сообщений, разделов
  - задача → период, приоритет, макс. время выполнения (для типа ядра)
  - раздел → задачи, допустимые ядра
  - сообщение → отправитель, получатель, размер, длительность передачи (через память, через сеть)



# Задача планирования: цели

- Привязать разделы к ядрам
  - Трафик между модулями  $\rightarrow \min$   
(минимизация загрузки сети)
  - Загрузка ядра  $\leq U_{\max}$ (ядро)
  - Привязка к допустимым ядрам
  - Инкрементальная привязка (расширение прежней)
- Синтезировать конфигурацию коммутируемой межмодульной сети
  - Маршруты передачи данных
  - Разделение пропускной способности
  - Расписания выдачи данных
- Построить расписание окон для каждого ядра
  - Корректность расписания проверяется моделированием работы динамического планировщика
  - Расписание считается корректным, если все задачи выполняются в пределах директивных сроков при максимальных длительностях выполнения

# Распределение разделов по модулям и ядрам методом ветвей и границ



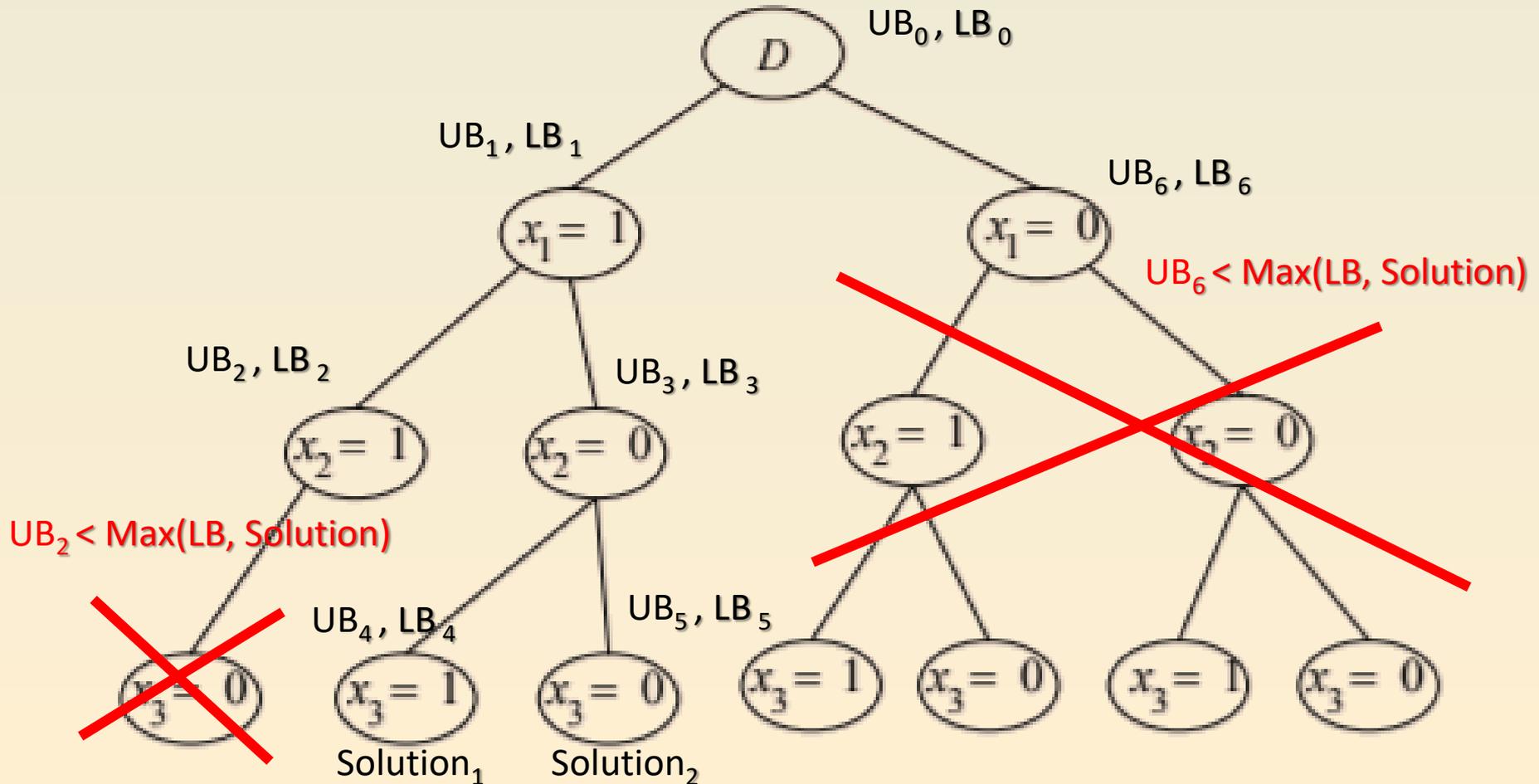
# Метод ветвей и границ: отсечение

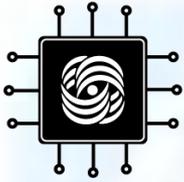
UB – верхняя граница;

LB – нижняя граница;

Solution – решение, полученное проходом из корня до листа дерева поиска;

Max(LB, Solution) – максимальное значение целевой функции среди построенных ранее на нижних границах или полученных достижением листьев дерева поиска .

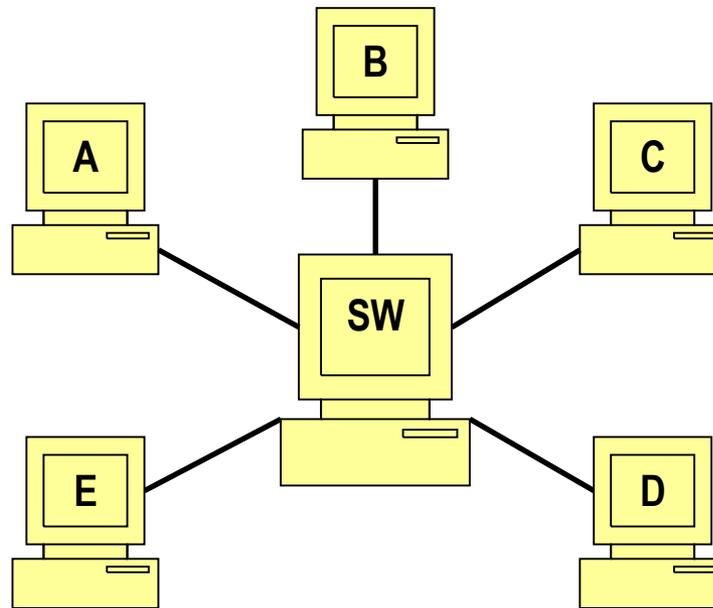




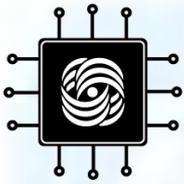
# Другие виды базовых алгоритмов

- Генетический/эволюционный алгоритм
  - решение = генетический код
  - популяция решений
  - решения скрещиваются и эволюционируют
  - от поколения к поколению популяция «улучшается»
- Муравьиный алгоритм
  - решение = путь в графе
  - путь ведет от муравейника к пище
  - короткий путь – хороший
  - по графу ходят «муравьи», оставляющие на хороших путях «феромонный след»
  - последовательно пускаем много муравьёв => обнаруживаем лучший путь
- Роевые, кукушечные, ...

# Коммутатор: параллелизм обмена



- Дуплексные (двунаправленные) каналы
- Частичный параллелизм обмена
  - $A \rightarrow B \parallel B \rightarrow C \parallel C \rightarrow D$  – нет конфликта
  - $A \rightarrow B, C \rightarrow B$  – конфликт на линии  $SW \rightarrow B$ ; как делить линию?
- Проблема мультиплексирования потоков данных при отправке
- Оставшиеся конфликты – как лечить?
  - синхронизация доступа (нужно единое время)
  - централизованное управление (накладные расходы...)
  - верхние оценки задержек (а если между A и B большой поток?)

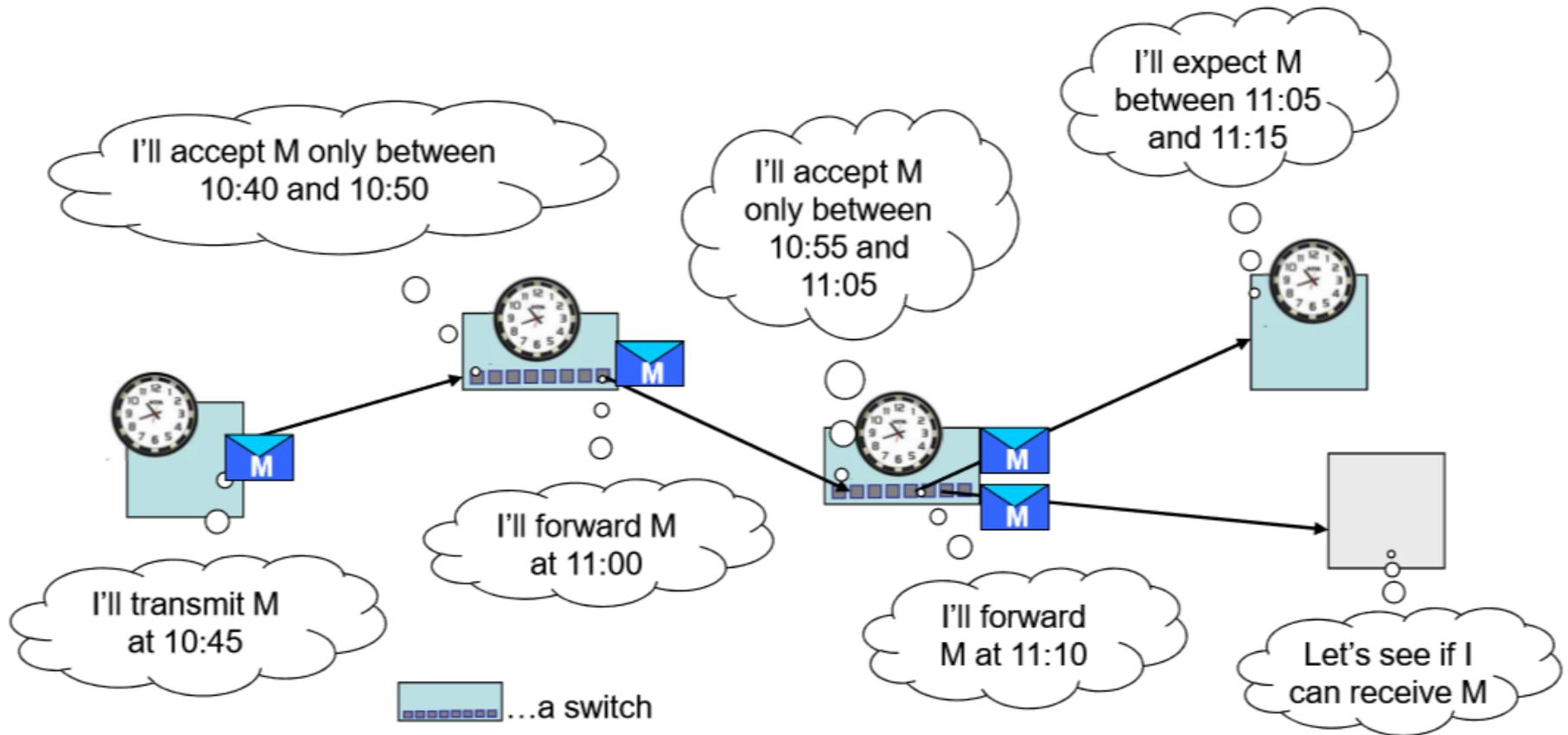


# Ethernet с временной синхронизацией

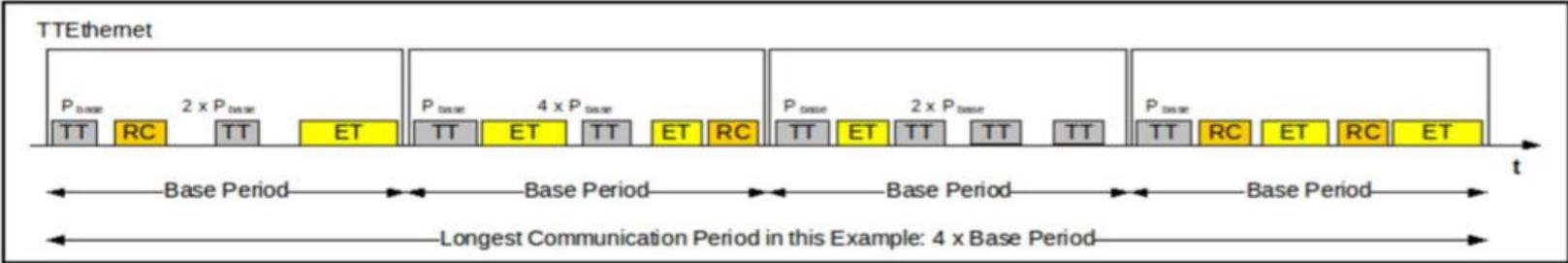
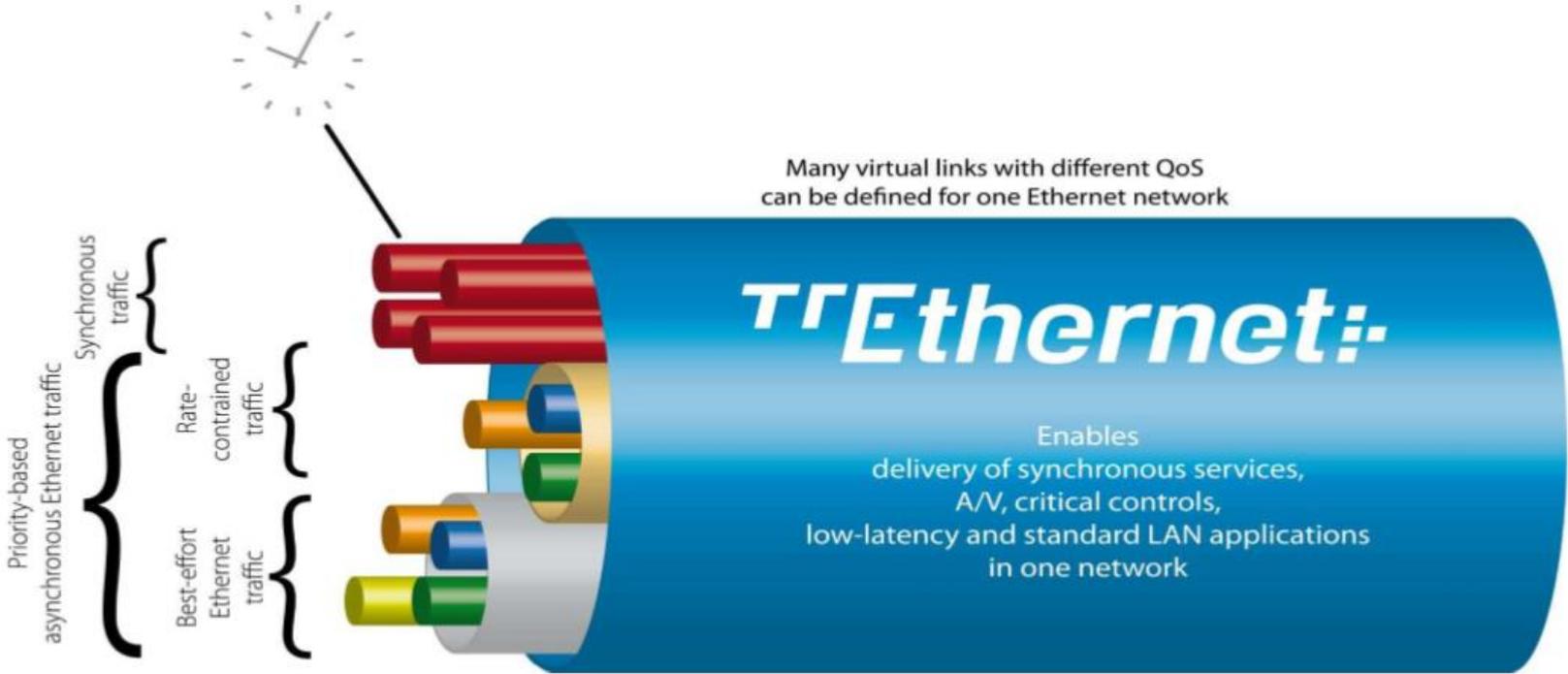
- Особенности «обычного» Ethernet:
  - «+»: использование отработанной технологии
  - «-»: коллизии, непредсказуемость задержек
- Стандарт бортовых сетей на базе Ethernet:
  - единое время у всех абонентов
  - глобальное расписание использования сети
  - максимальная предсказуемость задержек (детерминизм)

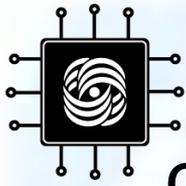
# Time-triggered Traffic Timing

- Full control of timings in the system
- Defined latency and sub-microsecond jitter
- Minimum memory needs



# TTEthernet Traffic Partitioning



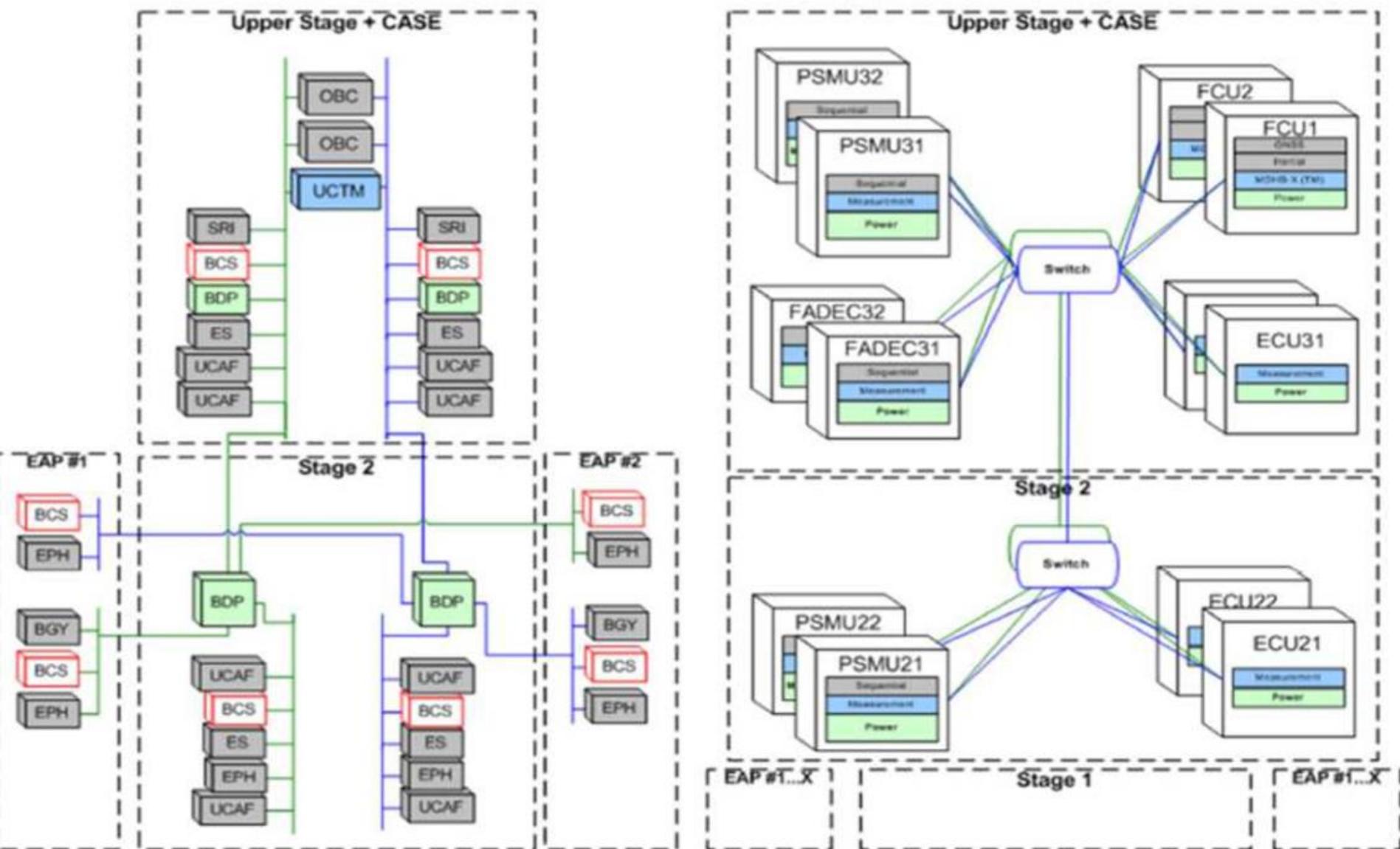


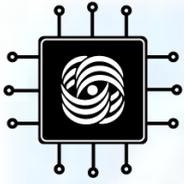
# Ariane 5 → Ariane 6:

общая шина MIL STD-1553B → сеть TTEthernet

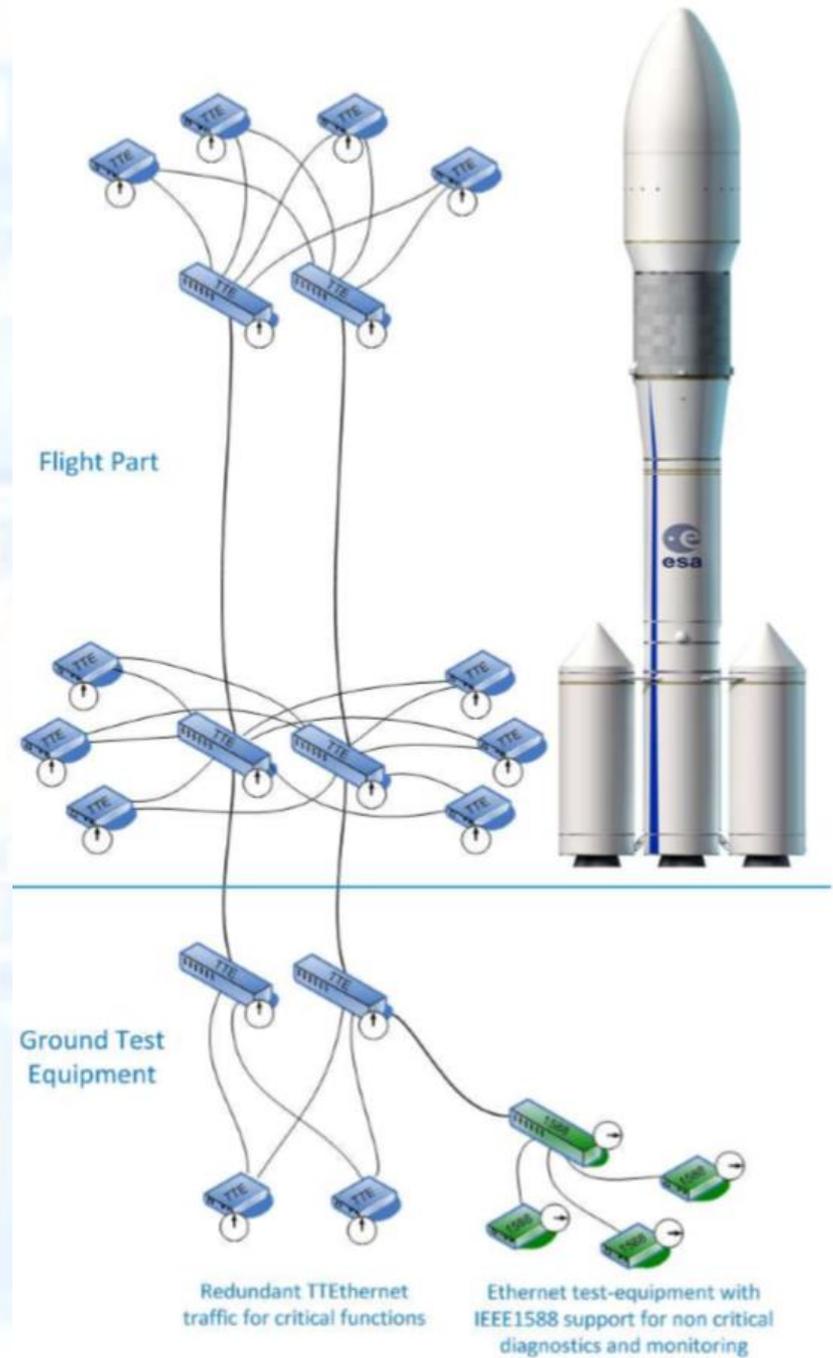


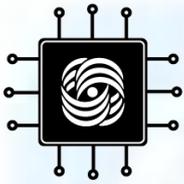
# Launcher Application – Ariane 6





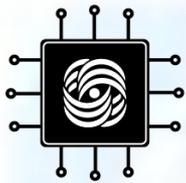
# Интеграция с наземными системами



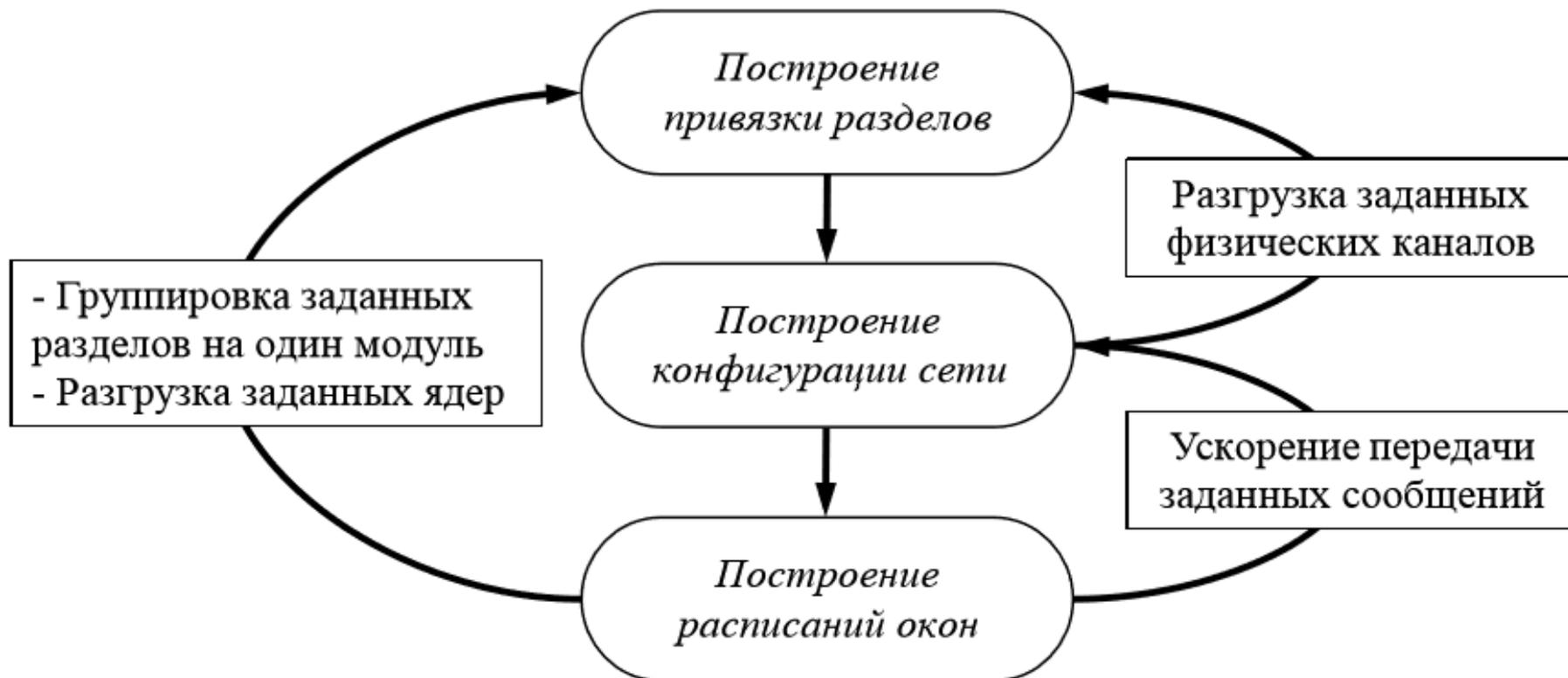


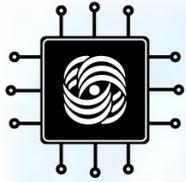
# ТТ-Ethernet: планирование передачи данных по сети

- Вычисление интервалов отправки:
  - Совокупность ограничений передается в решатель систем линейных ограничений
  - Любое найденное решение – корректное расписание выдачи
- Вычисление интервалов ожидания на коммутаторах
  - Интервал ожидания =  
Интервал отправки +  
фиксир. задержка передачи до коммутатора



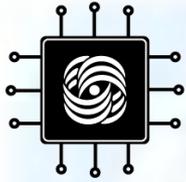
# Планирование вычислений с обратной связью





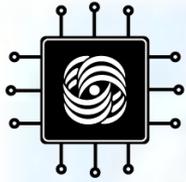
# Направления исследований

- Совместное планирование задач и конфигурирование сети
- Планирование с «обратной связью»
- Динамическое перераспределение нагрузки и потоков данных в системах с поддержкой миграции задач
- Планирование вычислений и информационного обмена в системах мягкого реального времени
  - центры обработки данных
  - телекоммуникационные системы
- Планирование вычислительных ресурсов в сетевых коммутаторах



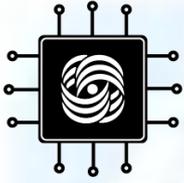
# Поддержка в учебных курсах

- Архитектура современных компьютеров
- Планирование вычислений в распределённых системах
- Архитектура управляющих систем реального времени
- Имитационное моделирование в исследовании и разработке информационных систем



# Профессиональные перспективы

- Разработка управляющих систем для
  - транспорта (наземного, водного, воздушного...), в т.ч. беспилотного
  - технологических процессов (производственные линии, добыча полезных ископаемых, ...)
  - энергетики, «умного» дома, ...
- Разработка систем планирования ресурсов для ЦОД, облачных платформ, телекоммуникационных систем
- Разработка средств автоматизации проектирования управляющих систем
- Профессиональные роли:
  - разработчик
  - алгоритмист
  - аналитик
  - архитектор



**Спасибо за внимание!**