

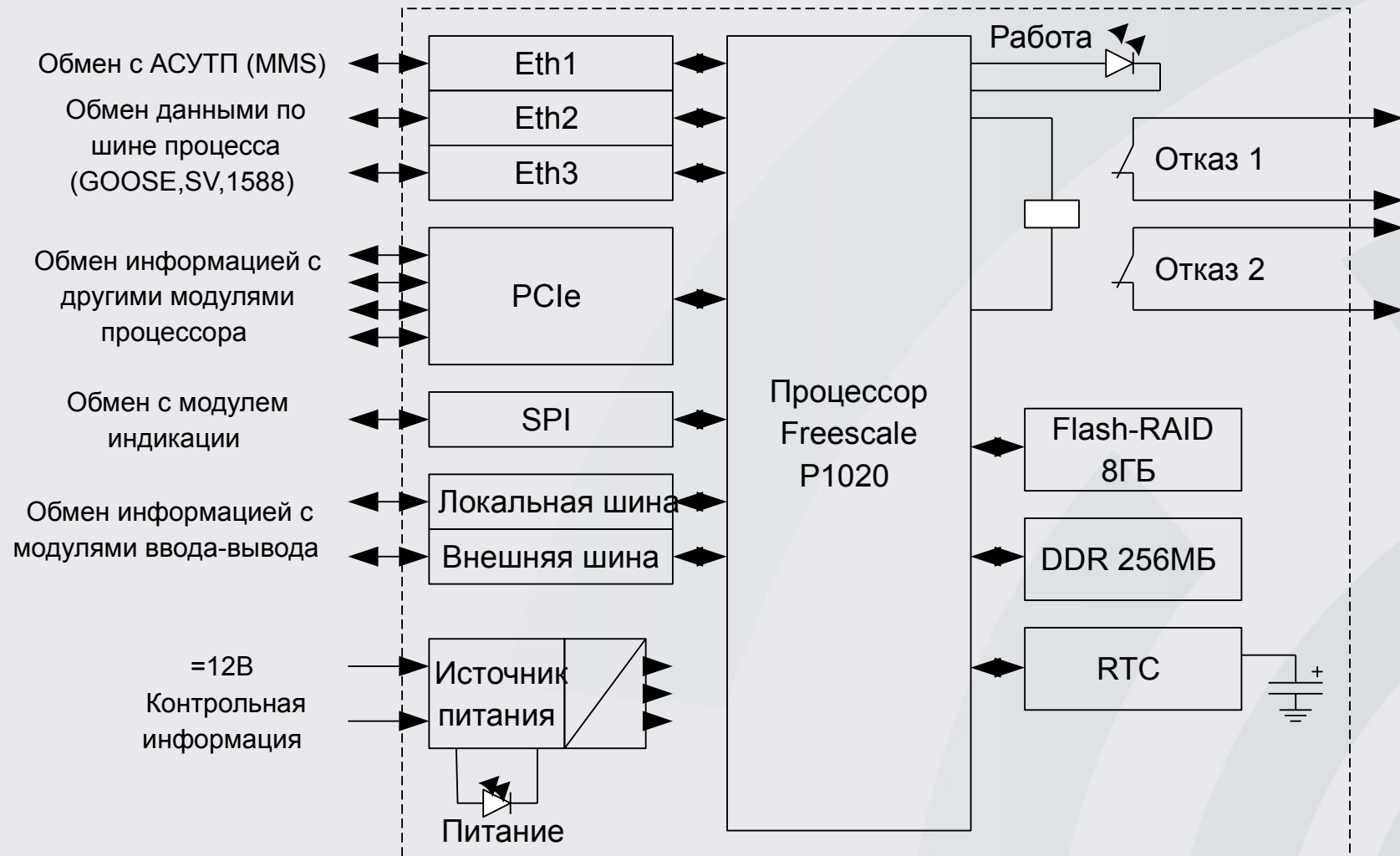


Сосуществование Linux и RTOS на
единой аппаратной платформе

Требования к разрабатываемому устройству

- SCADA для электрических подстанций
- Опрос десятков датчиков
- Полное резервирование всех узлов и линий связи
- Синхронизация времени между всеми территориально разнесёнными узлами с точностью не хуже 1 мкс
- Ведение журнала о неисправностях и срабатываний автоматики, запись журнала на встроенный носитель
- Десятки программных потоков, ответственных за первичную обработку получаемых данных
- Время реагирования и обработки внешнего события не более 1 мс
- Поддержка сетевых протоколов: IEC 61850 (MMS, GOOSE, SV), IEC 62531, IEEE 1588
- Графический интерфейс оператора

Структурная схема блока сбора и первичной обработки данных



Проблема выбора ОС

➤ Проблемы Linux:

- Не отвечает требованиям жёсткого реального времени (несмотря на наличие RT патчей ядра)
- ОС общего назначения => сложная в реализации => потенциально менее надёжная

➤ Проблемы RTOS (eCos, FreeRTOS и т.п.):

- Неполюценный или отсутствующий сетевой стек
- Ограниченность в выборе инструментария и вспомогательных библиотек при реализации приложений верхнего уровня

Выбранная архитектура: асимметричная мультипроцессорная обработка данных

Первое ядро процессора

Task 1: получение данных с датчиков

Task 2: ведение и запись журнала

Task 3: диагностика, оперативное переключение на резервные модули

Task 4: управление исполнительными устройствами

Task 5: Обмен данными с вспомогательным ядром

Task 6..40: первичная обработка полученных данных

RTOS

Второе ядро процессора

Графический интерфейс с пользователем

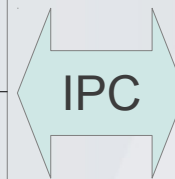
Диагностика

Протоколы прикладного уровня

Сетевая подсистема

Взаимодействие с основным ядром

Linux



Проблемы АМР архитектуры

- Разделение аппаратных ресурсов между двумя ОС: память, процессор, устройства ввода-вывода

Варианты решения:

- Виртуализация разделяемых устройств
- Жёсткая привязка устройств к заданному ядру
- Кооперативное использование ресурса

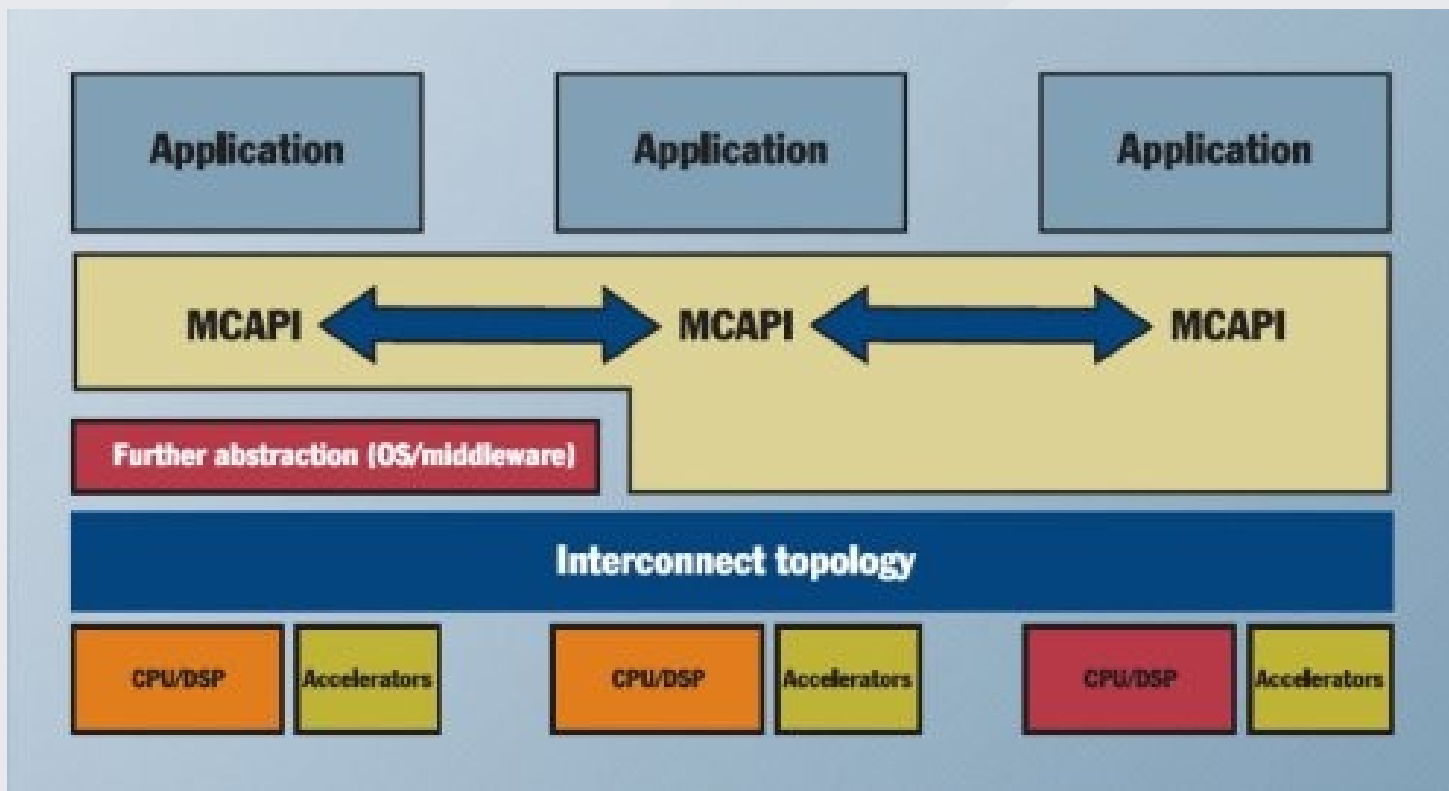
- Межпроцессорное взаимодействие

Варианты решения:

- Разделяемая область памяти
- Аппаратные средства (mailbox, SRIO и т.п.)
- Готовые решения от вендоров: MIPC, DSPLINK
- Свободные универсальные решения: MCAPI, RPMsg + virtio

МСАPI: АРІ для межпроцессорного взаємодія

- Розроблений і підтримується The Multicore Association
- Обеспечує стандартний АРІ для взаємодія користувальських програм, запущених на різних ядрах АМР системи



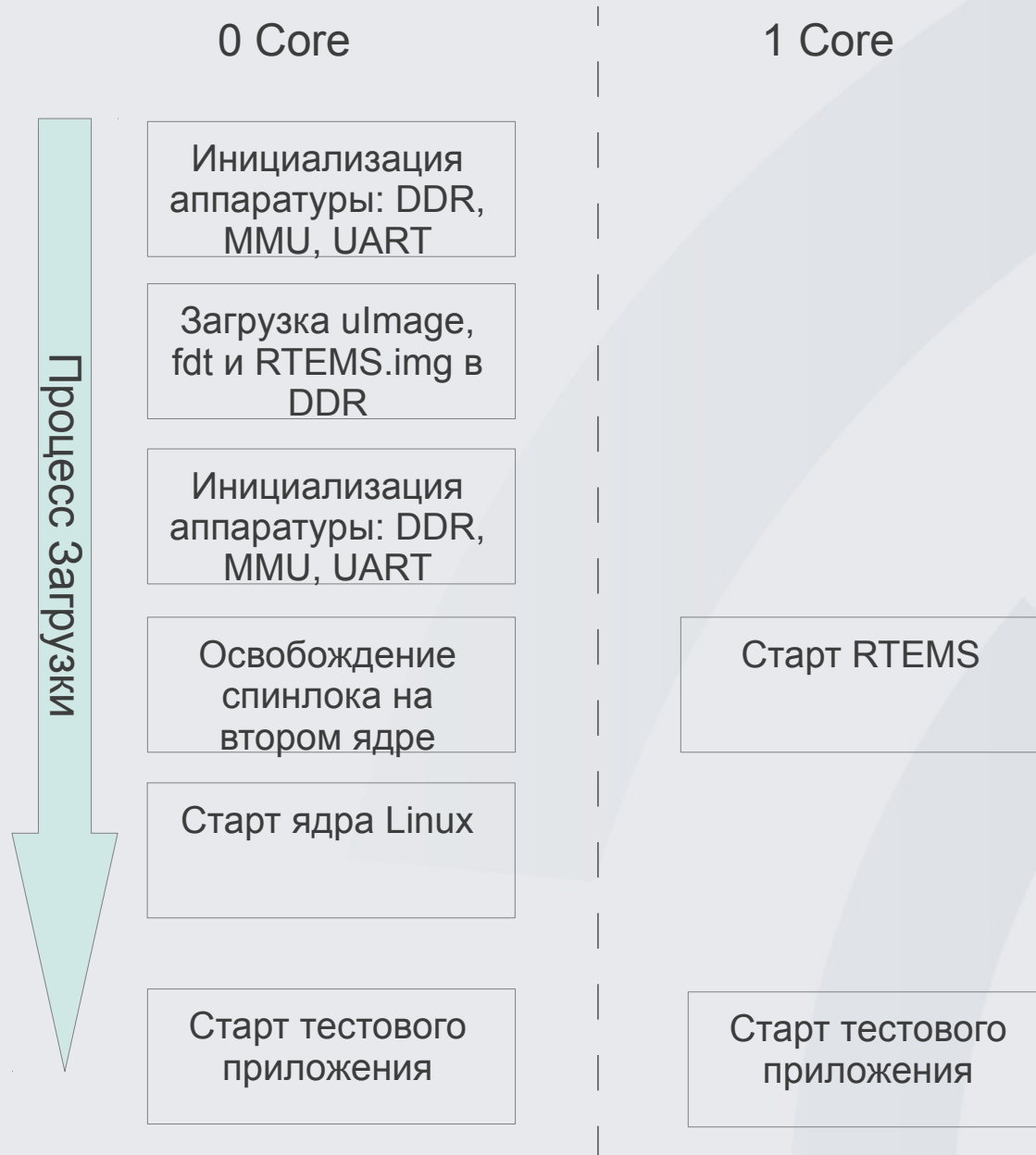
MSAPI: детали реализации и возможности

- API разработан для взаимодействия как процессорных ядер в одном чипе, так и для отдельных процессоров
- Ключевые цели при реализации:
 - Производительность
 - Низкая латентность
 - Платформонезависимость
- Может работать как поверх “голого железа”, так и поверх гипервизора
- Позволяет работать с сотнями ядер, обеспечивая эффективную маршрутизацию сообщений между ними
- Топология маршрутизации задаётся на этапе компиляции
- Поддерживает передачу как сообщений, так и непрерывных байтовых потоков

Практическая реализация

- SoC: Freescale P1020 (dual core, 64K L1, 256K L2, 1Gb DDR)
- RTOS: RTEMS 4.11 for PowerPC
- Linux: 2.6.35
- U-Boot: v2010.12
- Измерение временных интервалов при помощи встроенного в SoC таймера, работающего на 10МГц
- Тестовые приложения запускаются на обоих ядрах и замеряют:
 - Максимальный интервал между генерацией аппаратного прерывания и входом в процедуру обработчика
 - Время на переключение контекста
 - Время передачи сообщения из Linux в RTOS и обратно

Диаграмма загрузки тестового приложения



Результаты измерений

- Время реакции на прерывание: 0,3мкс
- Время переключения контекста: 0,8мкс
- Максимальное время передачи сообщения из Linux в RTEMS: 3мкс
- Максимальное время передачи сообщения из RTEMS в Linux: 18мкс

Выводы

- Был продемонстрирован ещё один способ построения hard real time системы на GNU/Linux и других свободных программных компонентах
- Кроме AMP архитектуры существуют и другие варианты реализации системы, близкой к жёсткой многозадачности реального времени:
 - Linux + PREEMPT_RT планировщик:
задачи реального времени в контексте ядра
 - Linux + Xenomai/RTAI фреймворк:
задачи реального времени в юзер спейсе
 - RTLinux:
задачи реального времени в контексте RTOS,
Linux запускается как idle задача
- Для совместного доступа к аппаратуре в AMP системе можно применить гипервизор, как коммерческий (MontaVista, WindRiver), так и свободный (KVM, XEN)

Спасибо!

Dmitriy Gorokh

R&D Dept, Promwad

dmitriy.gorokh@promwad.com