СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ

Центр параллельных вычислительных технологий

Семинар 2

Карта памяти и межпроцессорные взаимодействия SoC NM6408

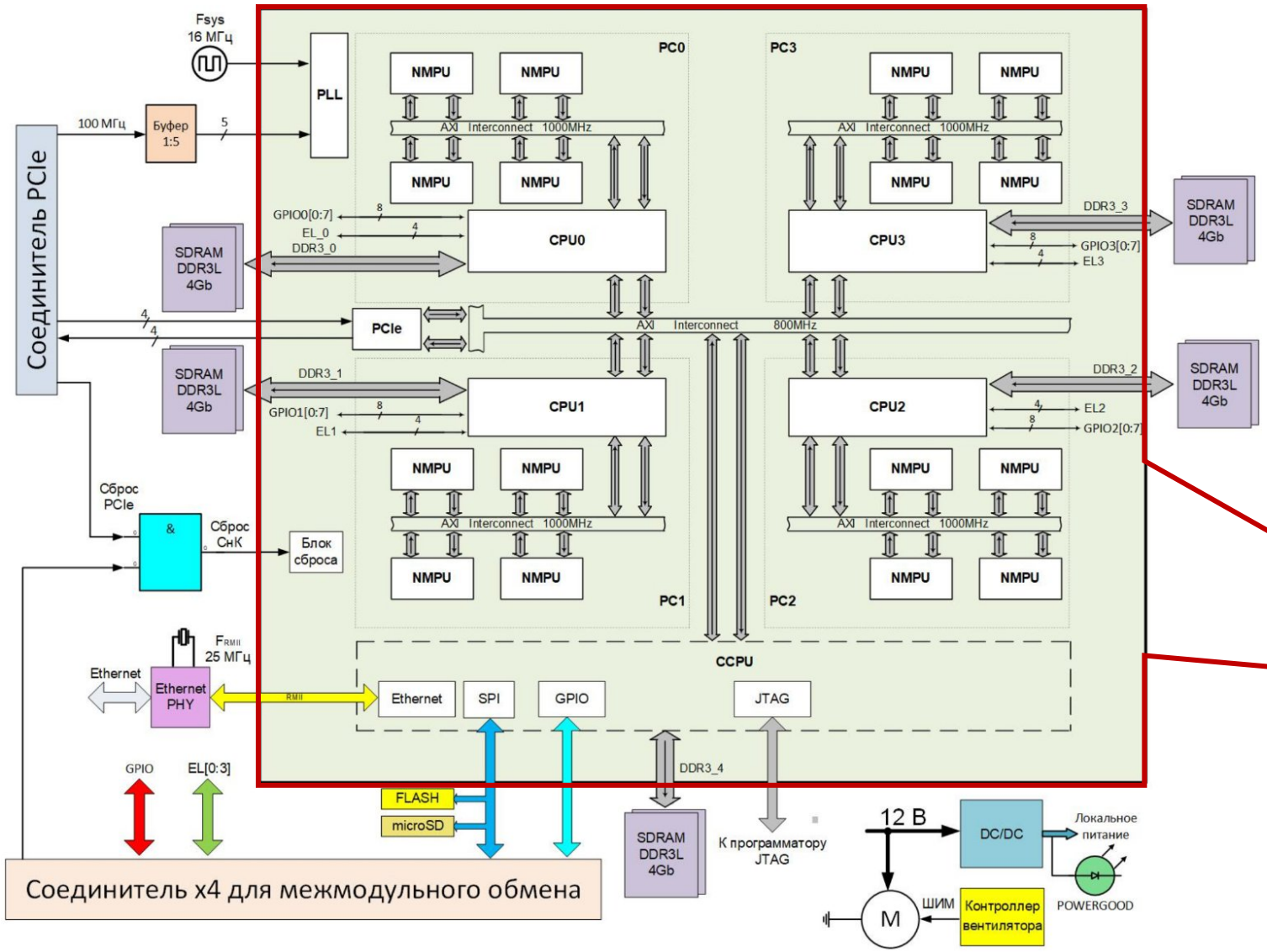
Михаил Георгиевич Курносов

Цикл семинаров по архитектуре процессоров цифровой обработки сигналов НТЦ Модуль NeuroMatrix

Содержание

- Структурная организация SoC NM6408 (СБИС К1879ВМ8Я)
- Иерархия памяти SoC NM6408
- Карта памяти
- Статическое выделение памяти – загрузка сегментов исполняемого файла
- Динамическое выделение памяти (DDR3, локальные банки)
- Синхронизация выполнения процессов
- Обмен информацией host – SoC

Модуль NM Card Mini



- **СБИС К1879ВМ8Я (NM6408)**
  - 16 ядер NMC4
  - 5 ядер ARM Cortex-A5
- **SDRAM**: 5GB DDR3L
- **NIC**: Ethernet 100 Мб/с
- **PCIe** 2.0 x4
- **4 коммуникационных порта** E-Link для межмодульного обмена (до 16 ГБ/с)

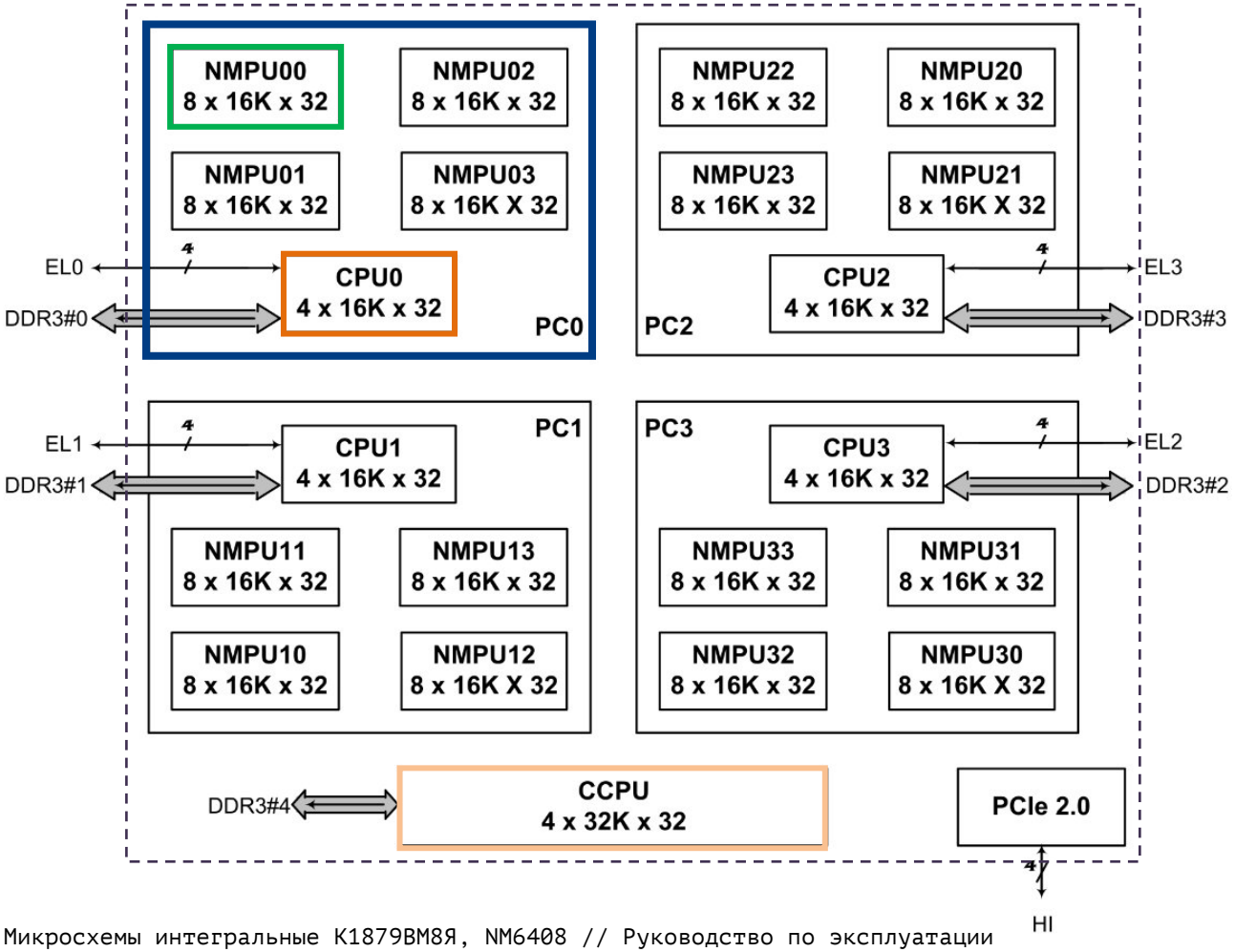


Функциональная схема Модуля ЮФКВ.469535.006 // Руководство по эксплуатации NM Card Mini

СБИС К1879ВМ8Я (SoC NM6408)



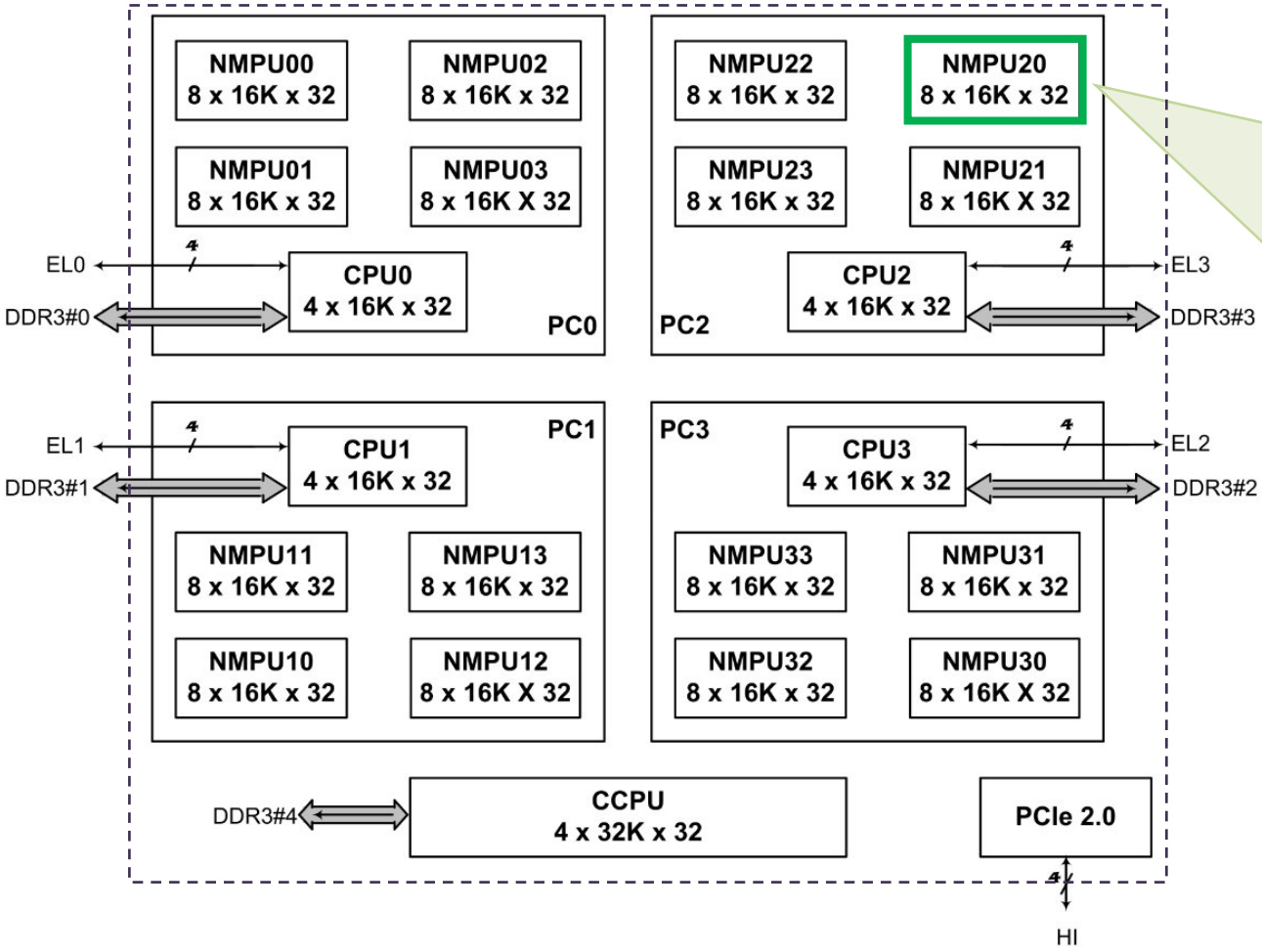
- **Гетерогенная многопроцессорная система на кристалле (СнК, SoC), 512 GFLOPS**
  - https://www.module.ru/products/1/26-18798
  - 16 процессорных ядер NeuroMatrix Core 4 (DSP)
  - 5 ядер ARM Cortex-A5 (управление системой)

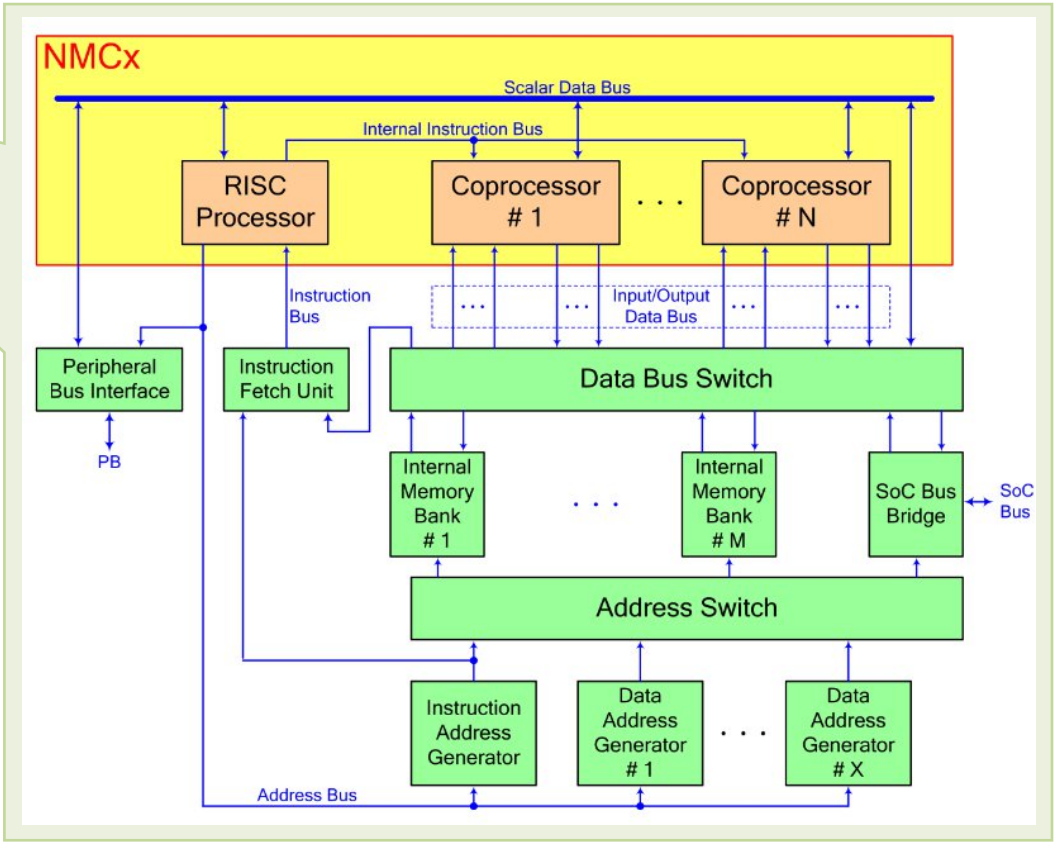


- **Процессорный узел (ПУ)** – ядро ARM/NMC и 4 или 8 банков памяти (16Kx32 бита)
- **Процессорный кластер** (PC – Processing Cluster) – 4 ПУ на базе NMC, 1 ПУ ARM, контроллер DDR3, 128 GFLOPS
- **NMPU** (NeuroMatrix Processing Unit) – ПУ на базе ядра NMC4 (1 GHz), 8 банков памяти, 32 GFLOPS
- **CPU** (Cortex Processing Unit) – управляющий ПУ на базе ядра ARM (800 MHz), 4 банка памяти, доступ к DDR3, доступ к другим модулям через каналы EL
- **CCPU** (Central Cortex Processing Unit) - центральный управляющий ПУ на базе ARM (600 MHz), 4 банка внутренней памяти, управление системой и доступ к памяти DDR3 по 32-разрядной шине данных
- **Host** – системная плата с процессором x86-64 (Elbrus, ARM) к шине PCIe, которого подключен модуль
- **ARM-часть** – ПУ на ядре ARM SoC NM6408
- **NMC-часть** – ПУ на ядре NMC SoC NM6408
- **Board** – SoC NM6408 + его ресурсы

Ядро NMC4 (32 GFLOPS)

- Управляющее RISC-ядро (декодирование VLIW инструкций)
- Сопроцессоры цифровой обработки данных (векторно-матричные)



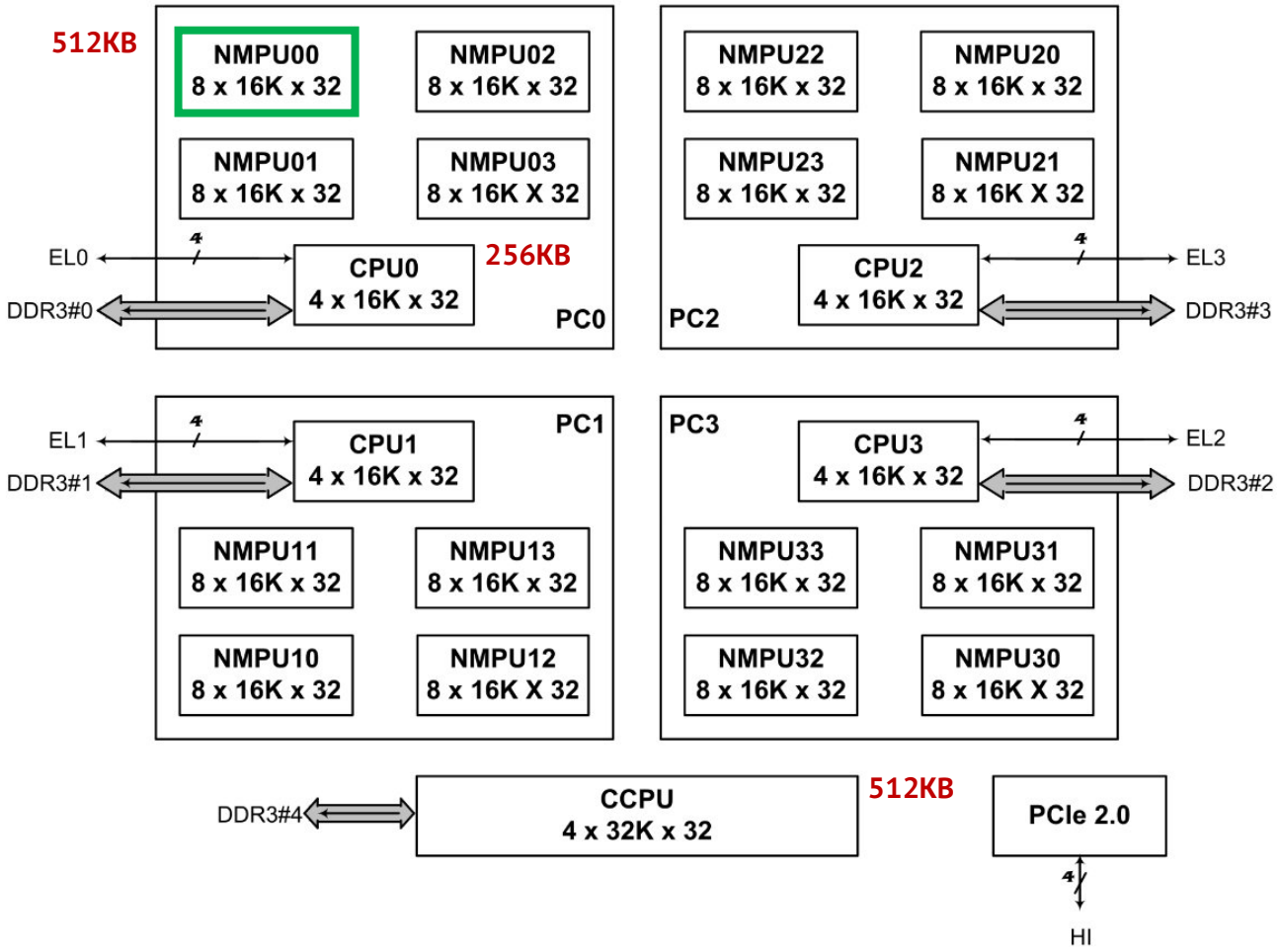


- Управляющее RISC-ядро (декодирование инст.)
- Сопроцессоры цифровой обработки данных

Микросхемы интегральные К1879ВМ8Я, NM6408 // Руководство по эксплуатации

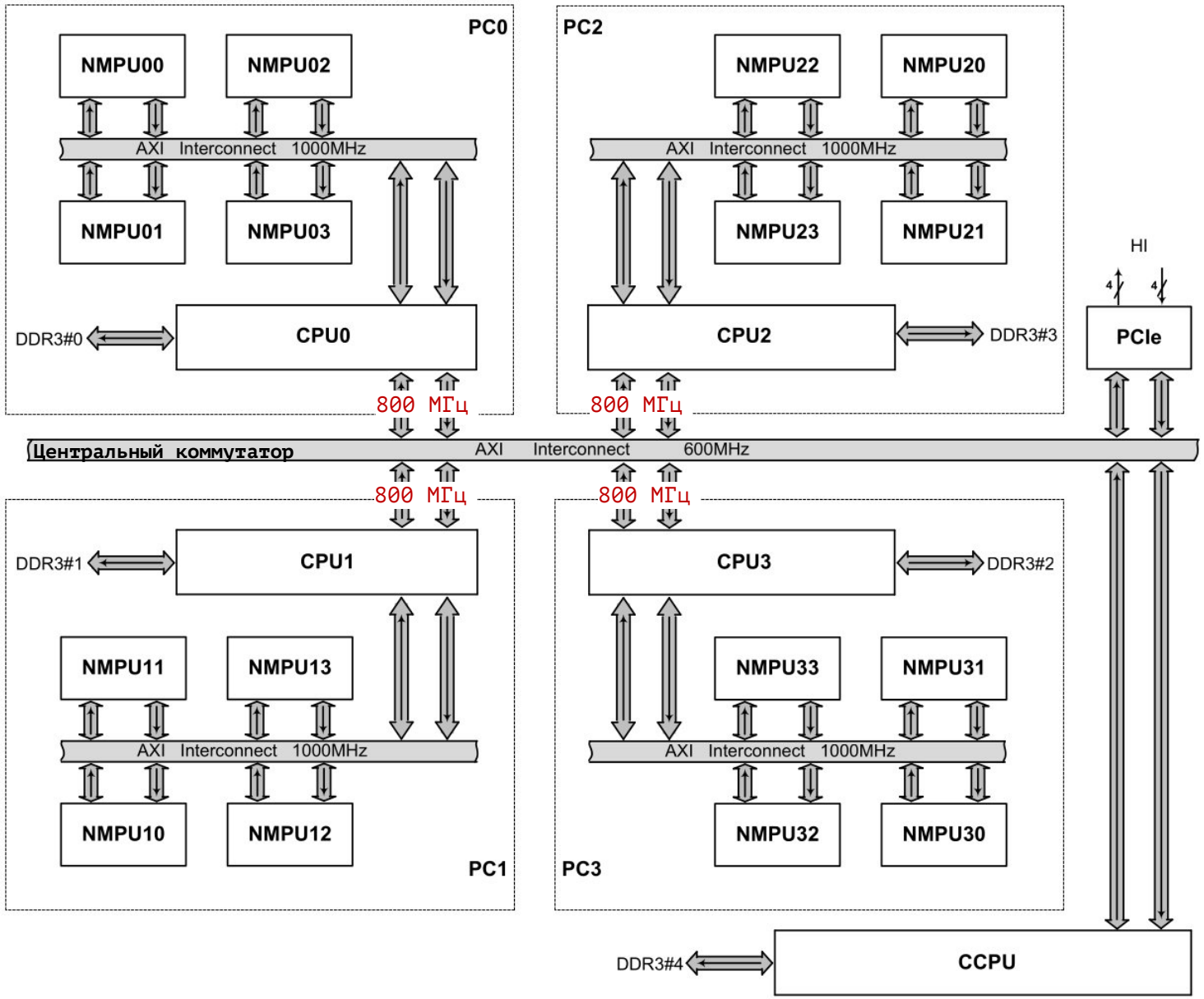
Иерархия памяти SoC NM6408

- **Каждый NMC** имеет 8 банков внутренней памяти SRAM (16K x 32 бита, 512КБ), адресация блоками по 32 бита
- **Каждый CPU** имеет 4 банка внутренней памяти (16K x 32 бита, 256КБ), адресация байтовая
- **CCPU** имеет 4 банка внутренней памяти (32K x 32 бита, 512КБ), адресация байтовая
- Внутренняя память: 76 Мбит (9.5 MB)
- Внешняя память DDR3: 40 Гбит (5 ГБ)
- Внутренняя память и DDR3 включена в единое адресное пространство, доступно всем ядрам ARM и NMC
- Единое адресное пространство в SoC охватывает 4ГБ, используются физические адреса
- Все NMPU имеют доступ к внутренней памяти друг друга и DDR3
- NMC и ARM формируют 32-разрядные адреса



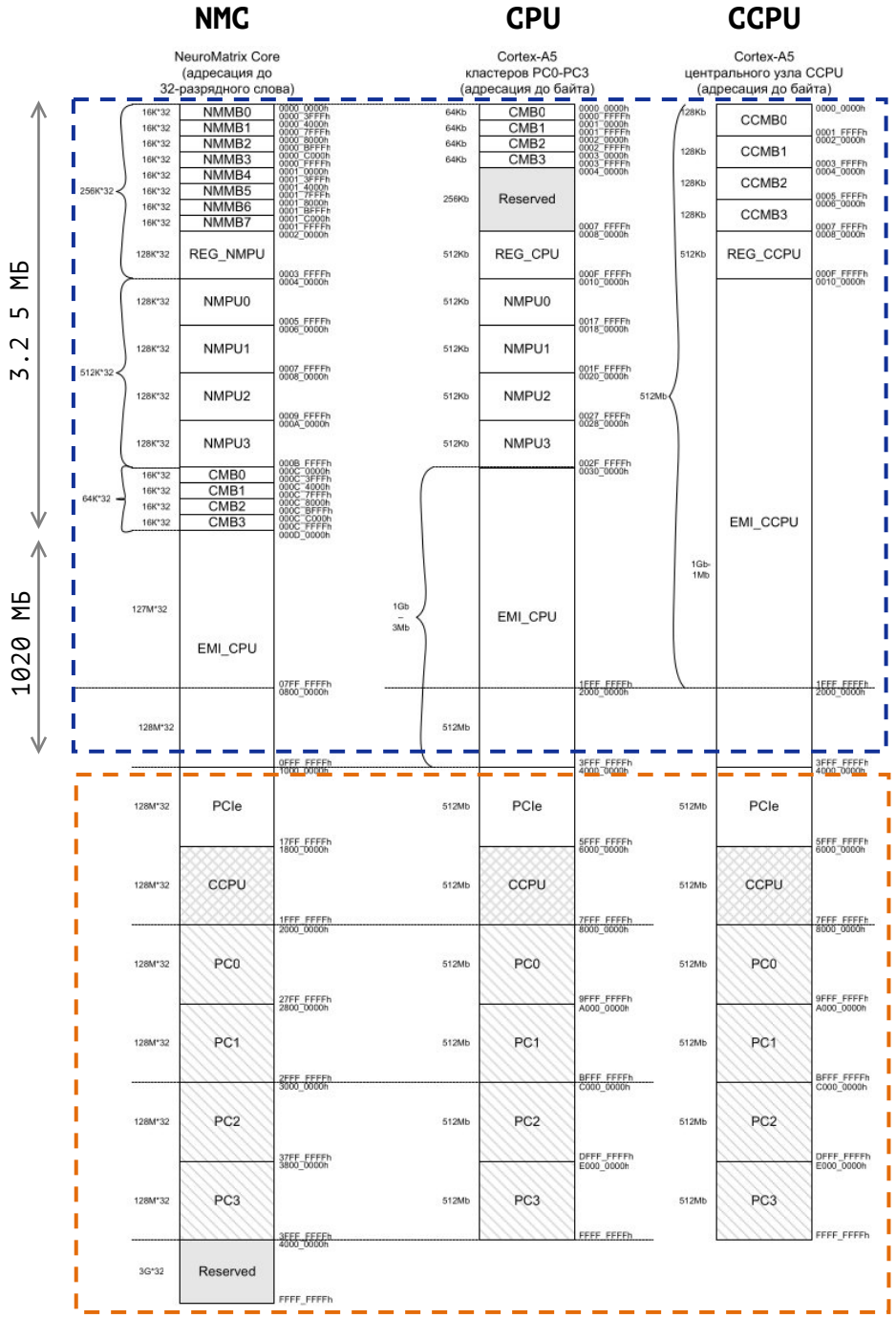
Система межпроцессорных шин SoC NM6408

- Доступ NMC к своей внутренней памяти NMC (NMMB)
- Доступ к внутренней памяти NMC соседа в кластере (AXI 1GHz)
- Доступ к внутренней памяти CPU в кластере (AXI 1GHz)
- Доступ к внешней DDR3 памяти своего кластера (AXI 1GHz)
- Доступ к внутренней памяти ПУ другого кластера (AXI 800MHz, AXI 600 MHz)
- Доступ к внутренней памяти CCPU (AXI 800MHz, AXI 600 MHz)
- Доступ к DDR3 памяти другого кластера (AXI 800MHz, AXI 600 MHz)

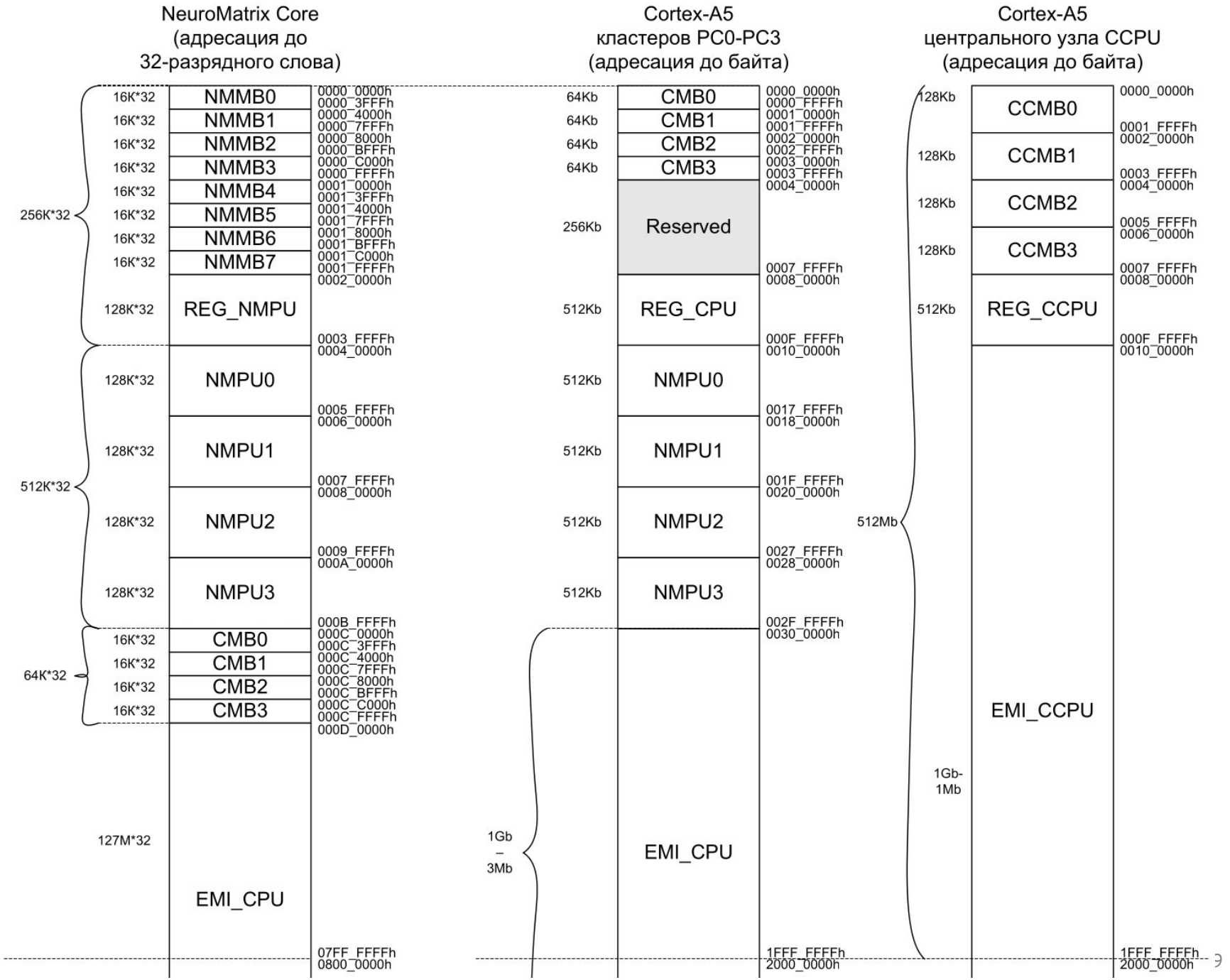


Карта памяти К1879ВМ8Я

- 7 областей памяти 0..4ГБ
- **Область младших адресов в 1 Гбайт (ПУ одного кластера):** адресация ячеек памяти разная для NMC, CPU, CCPU
  - **NMMB**i – банк i внутренней памяти NMPU (64KB), в банке NMMB0 хранятся программы обработки прерываний и начальный загрузчик (i=0,7)
  - **REG_NMPU** – регистры внешних устройств NMPU
  - **NMPU**j – внутренняя память NMPU j=0,3 кластера (512KB)
  - **CMB**j – банк j=0,3 внутренней памяти CPU (256KB)
  - **EMI_CPU** – банк внешней памяти DDR3 своего кластера (1ГБ), со стороны NMPU недоступны 1М младших ячеек
- **Адресное пространство в области 3 старших ГБ** одинаковое для любого типа процессора NMC, CPU, CCPU (6 областей):
  - **PCIe** – 512 Мбайт
  - **CCPU** – 512 Мбайт CCPU (младшая половина области 1 CCPU)
  - **PC0** – 512 Мбайт PC0 (младшая половина области 1 CPU 0)
  - **PC1** – 512 Мбайт PC1 (младшая половина области 1 CPU 1)
  - **PC2** – 512 Мбайт PC2 (младшая половина области 1 CPU 2)
  - **PC3** – 512 Мбайт PC3 (младшая половина области 1 CPU 3)
- Любому процессору при обращении к DDR-памяти другого кластера будет доступно не более 127M её ячеек (512MB)

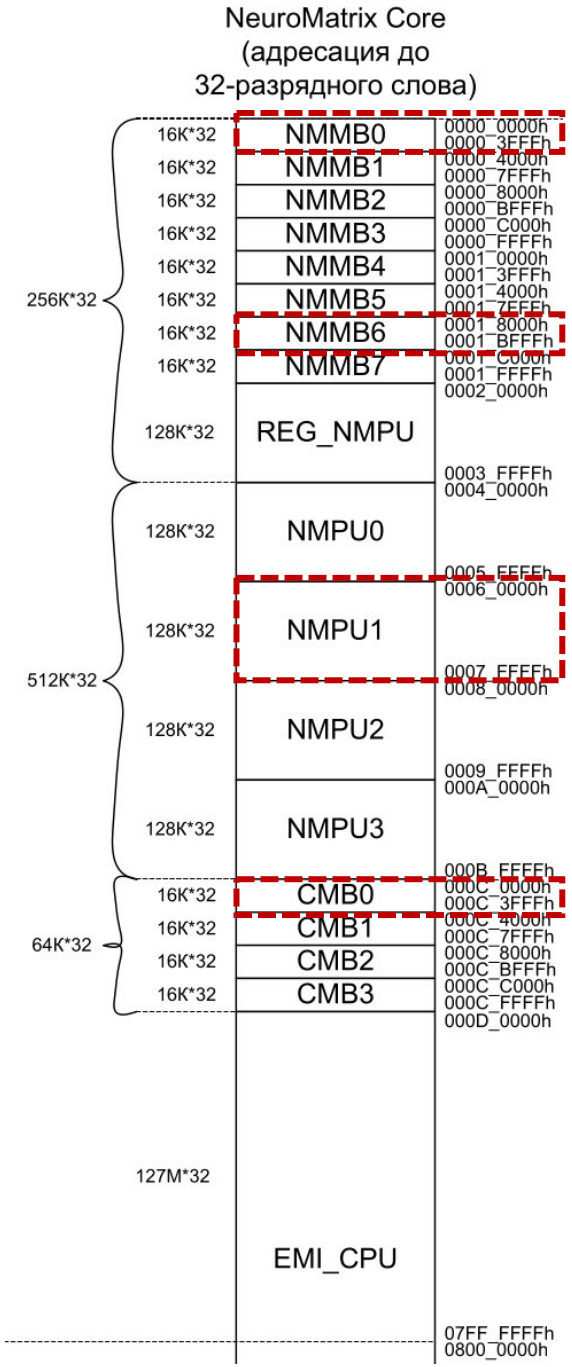


Карта памяти область младших адресов



Карта памяти NMC: область <= 1ГБ

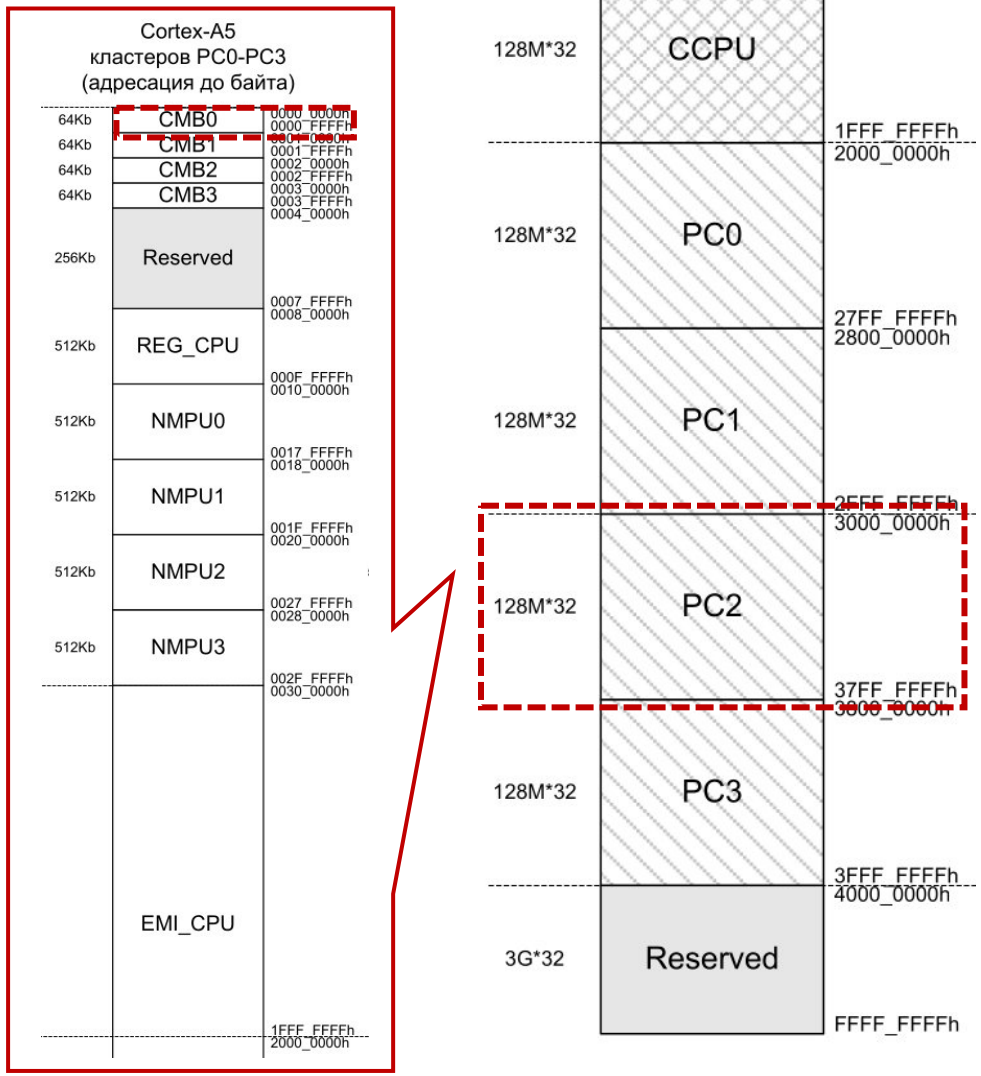
- **Область младших адресов в 1 Гбайт (ПУ одного кластера):**
  - **NMMB**i – банк i внутренней памяти NMPU (64KB), в банке NMMB0 хранятся программы обработки прерываний и начальный загрузчик (i=0,7)
  - **REG_NMPU** – регистры внешних устройств NMPU
  - **NMPU**j – внутренняя память NMPU j=0,3 кластера (512KB)
  - **CMB**j – банк j=0,3 внутренней памяти CPU (256KB)
  - **EMI_CPU** – банк внешней памяти DDR3 своего кластера (1ГБ), со стороны NMPU недоступны 1М младших ячеек
- Если NMC обращается по адресу 0x0000_0000 – маршрутизатор направляет запрос в банк NMMB0
- Если NMC обращается по адресу 0x0001_8000 – маршрутизатор направляет запрос в банк NMMB6
- Если NMC обращается по адресу 0x0006_0000 – это регион NMPU1, маршрутизатор направляет запрос к внутренней памяти NMPU1 текущего кластера, банк NMMB0
- Если NMC обращается по адресу 0x000C_0000 – это регион CMB0, маршрутизатор направляет запрос к внутренней памяти CPU текущего кластера, банк CMB0
- Если NMC обращается по адресу 0x000D_0000 – это регион EMI_CPU, маршрутизатор направляет запрос к DDR3 текущего кластера



Карта памяти NMC: область > 1ГБ

- **Адресное пространство в области 3 старших ГБ** одинаковое для любого типа процессора NMC, CPU, CCPU (6 областей):
  - **PCIe** – 512 Мбайт
  - **CCPU** – 512 Мбайт CCPU (младшая половина области 1 CCPU)
  - **PC0** – 512 Мбайт PC0 (младшая половина CPU 0)
  - **PC1** – 512 Мбайт PC1 (младшая половина CPU 1)
  - **PC2** – 512 Мбайт PC2 (младшая половина CPU 2)
  - **PC3** – 512 Мбайт PC3 (младшая половина CPU 3)

- Если NMC обращается по адресу 0x3000_0000 – это регион PC2, маршрутизатор направляет запрос в область памяти кластера 1, его CPU (младшая половина первого 1 ГБ), банк памяти CMB0
- Если обращение к PC2 выполнятся с NMC кластера 0, то происходит выход за пределы кластера на центральный коммутатор AXI 600MHz



Статические выделение памяти

- Размещение секций кода .text ELF-файла в областях памяти при запуске программы
- Размещение секций данных ELF-файла в областях памяти
  - .data – секция инициализированных данных
  - .rodata – секция инициализированных данных read-only
  - .bss – секция не инициализированных данных

- Модифицированный пример из nmc-devkit/share/nmc-first-steps/steps-sdk/step07_sections
  - Размещение массивов и функций в различных областях памяти (NMMB, EMI_CPU)
  - Описание размещения секций в памяти – скрипт .lds компоновщика

- Пример sections

Статические выделение памяти

- Пример `5_prog_emi_pc0`: код в DDR3

Динамическое выделение памяти

- Функция malloc динамически выделяет память в секции .heap
- Функция malloc принимает размер массива в 32-разрядных словах, не байтах

- Для возможности выделения памяти в разных банках, предусмотрена работа с несколькими кучами:
  - .heap
  - .heap1, .heap2, ..., .heap15

- Функция выбора активной кучи nmc_malloc_set_heap(int heap_num), heap_num - номер кучи для выделения памяти на следующем вызове malloc()

  - heap_num=0 - активна куча .heap,
  - heap_num=1 - .heap1
  - heap_num=2 - .heap2
  - ...

- По умолчанию, если не было вызова nmc_malloc_set_heap(), функция malloc() ищет требуемый объем памяти в куче .heap, затем в .heap1, .heap2, ..., .heap15

- Пример 6_malloc

Что дальше?

- Запуск кода на 16 NMC
- Копирование данных их памяти хоста в память SoC
- Организация параллельных вычислений
- Барьерная синхронизация
- Копирование памяти SoC -> Host