

МИКРОСХЕМА ИНТЕГРАЛЬНАЯ

K1891BM068

Руководство по эксплуатации

Часть первая

ТВГИ.431281.027РЭ

Литера «--- »

Перв. примен.	ТВГИ.431281.027
---------------	-----------------

РЭ содержит основные сведения об устройстве и работе микросхемы интегральной К1891ВМ068 (далее по тексту - микропроцессор «Эльбрус-2С3», микропроцессор), который является системой на кристалле и содержит два 64-разрядных процессорных ядра общего назначения с архитектурой широкого командного слова «Эльбрус», графический контроллер с поддержкой аппаратного ускорения 2D/3D графики, два контроллера оперативной памяти DDR4 и контроллеры канала ввода-вывода, PCI Express 3.0, Ethernet 1G(SGMII/1000Base-X) / 2.5G(SGMII), SATA 3.0, USB 3.0, audio, видео.

В РЭ представлены также сведения об интерфейсах микросхемы, электрических параметрах входных и выходных сигналов, электропитании, потребляемой мощности, конструкции, даны рекомендации по использованию и условиям эксплуатации. Микросхема реализована с топологическими нормами 16 нм.

Микропроцессор «Эльбрус-2С3» предназначен для создания на его основе встраиваемых и мобильных систем, функционирующих под управлением общего программного обеспечения «Эльбрус».

РЭ предназначается для разработчиков и производителей вычислительных систем на основе микропроцессора «Эльбрус-2С3», разработчиков программного обеспечения для таких вычислительных систем и обслуживающего персонала.

РЭ состоит из трех частей.

Часть первая – ТВГИ.431281.027РЭ. Содержит структурную схему микропроцессора, описание составляющих его устройств и внешних интерфейсов, эксплуатационные ограничения.

Часть вторая – ТВГИ.431281.027РЭ1. Содержит описание архитектурных регистров центральной части микропроцессора.

Часть третья – ТВГИ.431281.027РЭ2. Содержит описание регистров контроллера периферийных интерфейсов ЕІОН.

При изучении РЭ следует ознакомиться со следующими документами:

- ТВГИ.431281.027ТБ. Таблица назначения выводов;
- ТВГИ.431281.027Д67. Система команд. Часть 1. Описание архитектуры;
- ТВГИ.431281.02767.1. Система команд. Часть 2. Организация широкой команды и адреса архитектурных регистров процессорного ядра;
- ТВГИ.431281.02767.2 Система команд. Часть 3. Приложение С. Книга 1. Описание операций;
- ТВГИ.431281.02767.3. Система команд. Часть 4. Приложение С. Описание операций. Книга 2;
- ТВГИ.431281.02767.4. Система команд. Часть 5. Приложения D, E, F, H, M.

Содержание

1 Структурная схема микропроцессора	10
2 Функциональные узлы микропроцессора.....	17
2.1 Процессорное ядро Elbrus Core.....	17
2.1.1 Структурная схема	17
2.1.2 Обзор архитектуры широкого командного слова	19
2.1.3 Конвейер выполнения команд	44
2.2 Специализированные графические и видео ядра.....	50
2.3 Коммутационная сеть на кристалле OCI (On-Chip Interconnect)	54
2.4 Устройство доступа к оперативной памяти НМУ	56
2.5 Контроллер памяти МС	61
2.6 Устройство доступа к внешней памяти ХМУ	68
2.6.1 Структурная схема устройства ХМУ.....	68
2.6.2 Контроллер доступа в пространство ввода-вывода НС	69
2.6.3 Контроллер прерываний EPIC	76
2.7 Встроенный контроллер периферийных интерфейсов EIOH.....	78
2.7.1 Структурная схема	78
2.7.2 Программная модель контроллера EIOH.....	79
2.7.3 Системный коммутатор	81
2.7.4 Мультиконтроллер высокоскоростных линков WLINK/PCIe.....	83
2.7.5 Контроллер USB 3.0.....	92
2.7.6 Мультиконтроллер SATA/Ethernet.....	95
2.7.7 Legacy контроллеры	100
2.8 Графическая подсистема	115
2.8.2 Ускоритель трехмерной графики 3D GPU.....	115
2.8.3 Ускоритель двухмерной графики 2D GPU	117
2.8.4 Ускорители декодирования и кодирования Google.....	123
2.8.5 Ускорители декодирования и кодирования видео VXD и VXE.....	135
2.9 Система синхронизации	144

3 Встроенные средства энергосбережения, контроля и восстановления работоспособности.....	148
3.1 Средства энергосбережения	148
3.2 Контроллер управления энергосбережением PCS.....	148
3.3 Тестирование и разбраковка	154
3.4 Контроль работоспособности блоков кэш-памяти перед началом работы.....	155
3.5 Исключение дефектных ячеек блоков кэш-памяти при сохранении общей работоспособности кэш-памяти	155
3.6 Обнаружение и исправление в критических местах сбоев кэш-памяти и блоков оперативной памяти в процессе штатной работы	156
3.7 Логические анализаторы	156
4 Резерв.....	158
5 Описание интерфейсов микропроцессора.....	159
5.1 Напряжения питания периферийных элементов и условные обозначения типов выводов.....	159
5.2 Системные сигналы.....	160
5.3 Каналы оперативной памяти DDR4	167
5.4 Канал SPI.....	170
5.5 Каналы I2C master	170
5.6 Канал HDA.....	170
5.7 Каналы WLINK/PCIE.....	171
5.8 Каналы SATA/ETH.....	173
5.9 Каналы RS-232.....	175
5.10 Каналы USB	176
5.11 Канал RGB.....	177
5.12 Каналы HDMI	178
5.13 Каналы LVDS (видеовыходы).....	179
5.14 Канал I2C slave	179
5.15 Сигналы предупреждения о нештатных ситуациях	180
5.16 Каналы управления вентиляторами	181

5.17 Канал калибровки и диагностики РVT-сенсоров.....	182
5.18 Сигналы питания	182
6 Распределение пространства физических адресов	187
6.1 Аппаратно различимые области в пространстве физических адресов системы	187
6.2 Карта физической памяти.....	189
6.3 Правила отображения оперативной памяти в пространство физических адресов.....	192
6.3.2 Мэппирование оперативной памяти нижнего диапазона	192
7 Эксплуатационные ограничения	194
7.1 Электрические параметры и режимы эксплуатации	194
7.2 Рекомендации по проектированию модулей на базе микропроцессора.....	205
7.3 Стойкость к внешним воздействиям	230
7.4 Указания по применению и эксплуатации	231
7.5 Справочные данные	235
8 Хранение	244
9 Транспортирование	245
10 Утилизация.....	246

ТВГИ.431281.027РЭ1 Часть 2 Регистры состояния и управления

ТВГИ.431281.027РЭ2 Часть 3 Регистры встроенного контроллера периферийных
интерфейсов ЕЮН

Листов 236

Условные обозначения, термины и сокращения

В настоящем РЭ применяют следующие обозначения, термины и сокращения с соответствующими определениями.

Условный элемент	Описание
#	номер
глобалы	глобальные данные
квадро	формат данных 128 бит
линк	синоним канала обмена (межпроцессорный и ввода-вывода)
МП	микропроцессор
ОП	оперативная память
ПЛИС	программируемая логическая интегральная схема
ПО	программное обеспечение
ресет	сброс
РЭ	руководство по эксплуатации
СПО	специальное программное обеспечение
узел	синоним микропроцессора, когда он используется в качестве элемента в сети микропроцессоров (элемент в многопроцессорной системе)
AAU	Array Access Unit - устройство обращения к массивам
АС	Address Controller - адресный контроллер
ALC	Arithmetic Logic Channel - арифметико-логический канал (синоним ALU)
ALU	Arithmetic Logic Unit - арифметико-логическое устройство (синоним ALC)
АРВ	Array Prefetch Buffer - буфер предварительной подкачки массивов
CF	Chain File - файл стека связующей информации
CPU	Central Processor Unit - центральный процессор
СТРР	Control Transfer Preparation Register - регистр подготовки передачи управления
CU	Control Unit - устройство управления
DAM	Disambiguation Memory - устройство для мониторинга конфликтов

Условный элемент	Описание
	обращений в память по чтению и записи
DDR	Double Data Rate - передача данных с двойной скоростью
DW	Double Word – двойное слово (8 байтов)
hard_reset	аппаратный сброс
host	центральная часть устройства
IB	Instruction Buffer - буфер команд (устройство)
IO	Input Output - ввод - вывод
ITAG	Instruction Tag - память тегов кэша команд
ITLB	Instruction Table Look-Aside Buffer - кэш таблицы страниц для программного кода
I\$	Instruction Cache - кэш команд (первого уровня)
L1D\$	Level 1 Data Cache - кэш данных первого уровня
L2\$	Level 2 Cache - кэш второго уровня
L3\$	Level 3 Cache - кэш третьего уровня
MAU	Memory Access Unit - устройство обращения в память
MC	Memory Controller – контроллер памяти
MLT	Memory Locks Table - таблица блокировок памяти
MMX	Multi Media Extension - расширение системы команд для обработки мультимедийных данных
MMU	Memory Management Unit - устройство управления памятью
NA	Not Assign – не назначено, отсутствует
offset	смещение
PA	Physical Address – физический адрес
PF	Predicate File - предикатный файл
PIO	Processor Input Output – обмен с вводом-выводом по командам от процессорного ядра
PLU	Predicate Logic Unit - устройство логических предикатов (синоним PU - Predicate Unit)
PT	Page Table - таблица страниц (соответствия страниц виртуальной и физической памяти)
PTE	Page Table Entry - строка таблицы страниц

Условный элемент	Описание
PU	Predicate Unit - устройство предикатов (синоним PLU)
RF	Register File - регистровый файл
SLT	Semaphore Locks Table - таблица блокировок семафора
soft_reset	программный сброс
src	source - источник
SRU	SubRoutine Unit - устройство подпрограмм
TLB	Table Look-Aside Buffer - буфер таблицы страниц
TLU	Table Look-Aside Unit - устройство обращения в таблицу страниц
TU	Trap Unit - устройство обработки прерываний
VA	Virtual Address – виртуальный адрес

1 Структурная схема микропроцессора

1.1 Микропроцессор предназначен для создания мобильных и встраиваемых систем. В микропроцессоре реализованы вычислительные средства, графический контроллер с поддержкой аппаратного ускорения 2D/3D графики и контроллеры наиболее употребительных периферийных интерфейсов.

Структурная схема микропроцессора Эльбрус-2С3 представлена на рисунке 1.1.

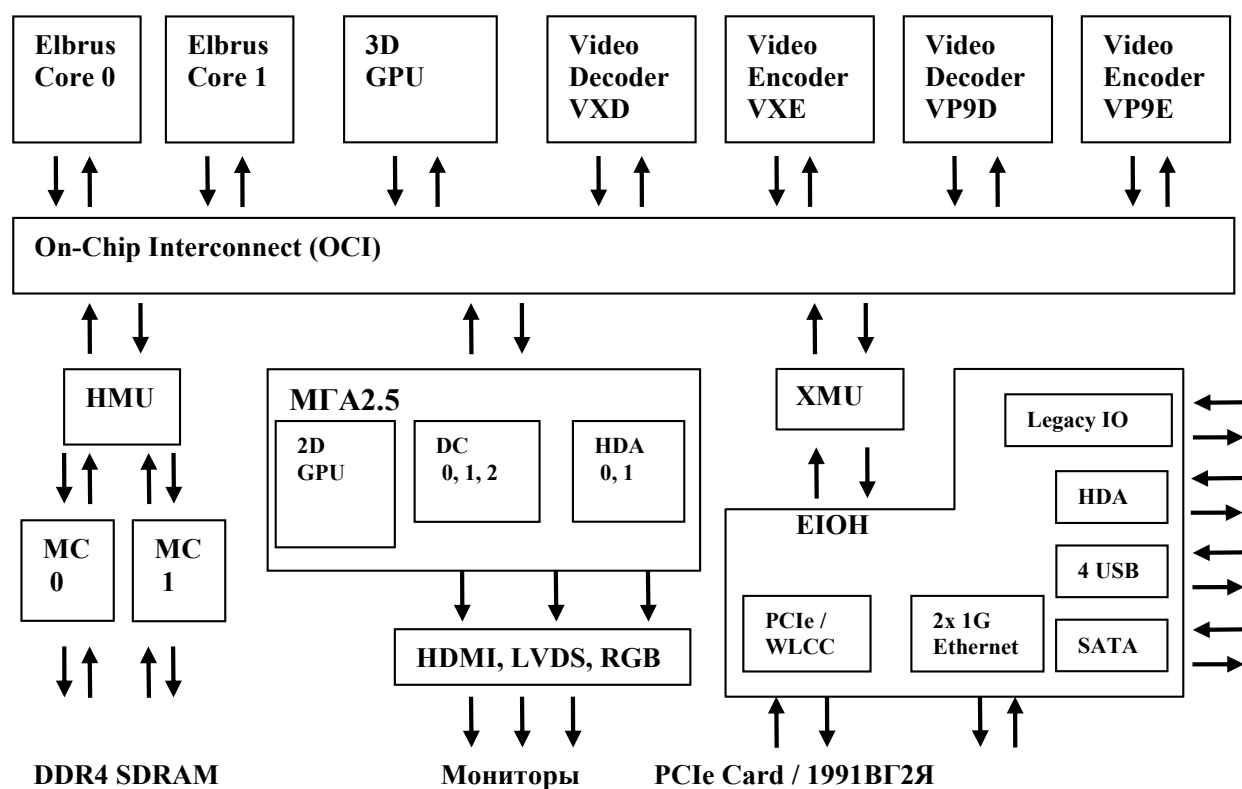


Рисунок 1.1 – Микропроцессор Эльбрус-2С3

Микропроцессор Эльбрус-2С3 содержит следующие основные блоки:

- Elbrus Core 0, 1 – процессорные ядра, реализующие архитектуру «Эльбрус»;
- 3D GPU – специализированное ядро, ускоритель трехмерной графики;

- Video Decoder VXD – специализированное ядро, ускоритель декодирования видео H.265, H.264, MJPEG и других;
- Video Encoder VXE – специализированное ядро, ускоритель кодирования видео H.265, H.264, MJPEG и других;
- Video Decoder VP9D – специализированное ядро, ускоритель декодирования видео в формате VP9;
- Video Encoder VP9E – специализированное ядро, ускоритель кодирования видео в формате VP9;
- OCI – коммутационная сеть на кристалле (On-Chip Interconnect);
- HMU – устройство доступа к оперативной памяти (Home Memory Unit);
- MC 0, 1 – контроллеры (Memory Controller) доступа к оперативной памяти;
- 2D GPU – специализированное ядро, ускоритель двухмерной графики;
- DC 0, 1, 2 – блок дисплейных контроллеров, предназначенный для периодического вывода изображений из кадровых буферов;
- HDA 0, 1 – звуковые (High Definition Audio) контроллеры;
- HDMI, LVDS, RGB – контроллеры интерфейсов, предназначенные для вывода изображений из кадровых буферов на мониторы; через HDMI интерфейс также может выводиться звук, формируемый двумя HDA контроллерами;
- MGA2.5 – устройство, объединяющее 2D GPU, блок дисплейных контроллеров и два звуковых контроллера;
- XMU – устройство доступа к периферийным интерфейсам (eXternal Memory Unit);
- EIOH – встроенный контроллер периферийных интерфейсов (Embedded Input-Output Hub), включающий в себя следующие контроллеры:
 - PCIe / WLCC – контроллер PCIe 3.0, способный работать в режиме 1x16, или 2x8, или 4x4 с общей шириной 16 линий, а также контроллер ввода/вывода WLCC, обеспечивающий подключение микросхемы 1991ВГ2Я. Контроллеры объединены через мультиплексор, позволяющий подключить к одним физическим линиям контроллер ввода/вывода WLCC или два контроллера

PCIЕ 3.0;

- 2x 1G Ethernet – два контроллера 1G(SGMII/1000Base-X) / 2.5G(SGMII);

- SATA – контроллер SATA 3.0;

- HDA – звуковой (High Definition Audio) контроллер;

- 4 USB – контроллер USB 3.0 с 4 портами выхода;

- Legacy IO – набор контроллеров низкоскоростных и служебных интерфейсов: RS-232, SPI, I2C, IPMB, GPIO.

1.2 Микропроцессор Эльбрус-2С3 является многоядерной системой на кристалле, состоящей из двух процессорных ядер общего назначения с архитектурой «Эльбрус» и шести специализированных ядер, предназначенных для обработки графических (2D и 3D графики) и видео (сжатия и распаковки сжатых видео изображений) потоков данных. Доступ в оперативную память обеспечивается с помощью устройства НМУ (Home Memory Unit), к которому подключены два контроллера оперативной памяти DDR4 (MC, Memory Controller). Функцию подсистемы ввода-вывода выполняет устройство ХМУ (eXternal Memory Unit) и встроенный контроллер периферийных интерфейсов ЕИОН (Embedded IO Hub). Процессорные ядра, как универсальные, так и специализированные, а также устройства НМУ и ХМУ соединены между собой сетью на кристалле ОСИ (On-Chip Interconnect).

1.3 Основные технические характеристики микропроцессора Эльбрус-2С3 представлены в таблице 1.1.

Таблица 1.1 - Основные технические характеристики микропроцессора Эльбрус-2С3

Наименование параметра	Значение параметра
Условное обозначение микросхем	K1891BM06A8, K1891BM06B8, K1891BM06C8
Основное функциональное назначение	Система на кристалле с двумя процессорными ядрами с 64-разрядной архитектурой «Эльбрус», графическим контроллером, двумя контроллерами оперативной памяти

Наименование параметра	Значение параметра
	ти, контроллером канала ввода-вывода и набором контроллеров периферийных интерфейсов для создания мобильных и встраиваемых систем
Количество процессорных ядер общего назначения	2
Рабочая частота процессорного ядра общего назначения, ГГц, не менее: - K1891BM06A8 - K1891BM06B8 - K1891BM06C8	2,00 1,8 1,60
Пиковая производительность на операциях с двойной точностью (FP64), GFLOPS, не менее: - K1891BM06A8 - K1891BM06B8 - K1891BM06C8	190 170 150
Пиковая производительность процессорного ядра, GFLOPS, не менее: - K1891BM06A8 - K1891BM06B8 - K1891BM06C8	95 85 75
Пиковая производительность графического контроллера на операциях с одинарной точностью (FP32) OpenCL , GFLOPS, не менее	300
Рабочая частота видеоядра, МГц, не менее	800
Разрядность данных: — целые — с плавающей запятой	8, 16, 32, 64 32, 64
Универсальное процессорное ядро: - ёмкость кэша данных, Кбайт - ёмкость кэша команд, Кбайт - ёмкость кэша второго уровня, Мбайт	64 128 2

Наименование параметра	Значение параметра
<p>Количество контроллеров оперативной памяти:</p> <ul style="list-style-type: none"> - доступный объем памяти, Гбайт - канал обмена - пиковая пропускная способность одного канала, Гбайт/с, не менее - пиковая пропускная способность двух каналов, Гбайт/с, не менее 	<p>2</p> <p>256</p> <p>DDR4-3200 с ECC</p> <p>20,8</p> <p>41,6</p>
<p>Графический контроллер:</p> <ul style="list-style-type: none"> - отображение - интерфейсы дисплея - разрешение дисплея 	<p>на 3 монитора</p> <p>RGB*, HDMI (2 канала), LVDS</p> <p>RGB – FULLHD</p> <p>HDMI – 4 К</p> <p>LVDS – 2,5 К</p>
<p>Мультиконтроллер WLINK/PCIe:</p> <ul style="list-style-type: none"> - конфигурации каналов - пиковая пропускная способность линии передачи PCIe/WLCC, Гбит/с - пиковая пропускная способность канала обмена из 16 линий передачи PCIe/WLCC, Гбайт/с 	<p>PCIЕ x16</p> <p>PCIЕ 2x8</p> <p>PCIЕ x8 + WLCC x8</p> <p>PCIЕ 2x4 + WLCCx8</p> <p>PCIЕ 4x4</p> <p>WLCC x16</p> <p>16</p> <p>32</p>
<p>Мультиконтроллер SATA/Ethernet:</p> <ul style="list-style-type: none"> - конфигурации каналов 	<p>4xSATA</p> <p>2xSATA + 1x1G(SGMII/1000Base-X) + 1x2.5G(SGMII)</p> <p>2xSATA + 2x1G(SGMII/1000Base-X)</p> <p>2xSATA + 2x2.5G(SGMII)</p>

Наименование параметра	Значение параметра
Контроллер SATA: - количество портов - количество контроллеров интерфейса Ethernet 1G/2.5G	Версия 3.0, 6 GT/s 4 2
Контроллер интерфейса USB 3.0 - количество портов - пропускная способность линии передачи USB 3.0, Гбит/с - пропускная способность линии передачи USB 2.0, Мбит/с	4 5 480
Контроллеры интерфейса RS-232	2
Контроллер интерфейса SPI	4 устройства
Контроллер интерфейса I2C	5 каналов
Контроллер аудио интерфейса HDA	1
Контроллер интерфейса GPIO/MPV	16 линий
Контроллер интерфейса тестирования JTAG	1 канал
Контроллер управления режимами энергосбережения микросхемы PMC	1
Контроллер управление питанием и режимами энергосбережения на уровне вычислительного комплекса SPMC	1
Контроллер EPIC	1
Контроллер IOEPIC	1
Системный таймер	1
Сторожевой (watchdog) таймер	1
Напряжения питания, В	0,8; 0,82; 0,85; 1,2; 1,5; 1,8; 3,3
Готребляемая мощность**, Вт, не более: - K1891BM06A8 - K1891BM06B8 - K1891BM06C8	30,0 (20,0) 27,0 (18,0) 24,0 (15,0)
Условное обозначение корпуса	1903 HFCEGA
Количество элементов в схеме электрической	4300000000

Наименование параметра	Значение параметра
<p>* В текущей итерации интерфейс не работает.</p> <p>** Первое значение является максимальной потребляемой мощностью, второе (в скобках) - средняя потребляемая мощность</p>	

2 Функциональные узлы микропроцессора

2.1 Процессорное ядро Elbrus Core

2.1.1 Структурная схема

Система команд вычислительного (процессорного) ядра построена по принципам архитектуры широкого командного слова. Она однозначно определяет набор устройств вычислительного ядра и состав связей между ними, но не ограничивает их внутреннюю организацию и пропускную способность. Набор операций, предоставляемый системой команд, не имеет сколько-нибудь выраженной специализации, что позволяет использовать вычислительное ядро для создания универсальных микропроцессоров. Он включает скалярные целочисленные операции над данными форматов 4 и 8 байт и скалярные вещественные операции над данными форматов 4 (single precision), 8 (double precision) и 10 (extended precision) байт. Для массивных вычислений также предусмотрены SIMD-операции над 8-байтовыми и 16-байтовыми «упаковками» из целочисленных данных форматов 1, 2, 4 и 8 байт или вещественных данных форматов 4 и 8 байт.

Структура процессорного ядра полностью соответствует спецификации архитектуры в части обеспечения параллельного выполнения программ, а также поддержки защищенных вычислений и двоичной компиляции кодов платформы Intel x86.

Устройства процессорного ядра полностью конвейеризованы, имеют многопортовую и многоканальную организацию, что обеспечивает параллельное бесконфликтное выполнение любых, определенных архитектурой, сочетаний операций. Процессорное ядро содержит ряд устройств, позволяющих в ходе выполнения программы контролировать правильность компиляторных оптимизаций кода и, в случае необходимости, корректировать выполнение программы. Процессорное ядро имеет 32-байтовую внешнюю шину данных для записи в память и чтения из памяти.

Параллельное и в большинстве случаев бесконфликтное выполнение операций обеспечивается многоконвейерной, многопортовой и многоканальной организацией устройств микропроцессора, большой пропускной способностью внутренних шин передачи данных и канала доступа в память. Компилятор несет ответственность за эффективное использование аппаратуры процессорного ядра на основе планирования выполнения программы. Аппаратура, в свою очередь, обеспечивает разрешение конфликтов по причинам неоптимального планирования или динамически возникших ситуаций при выполнении программы.

На рисунке 2.1 представлена структурная схема вычислительного ядра.

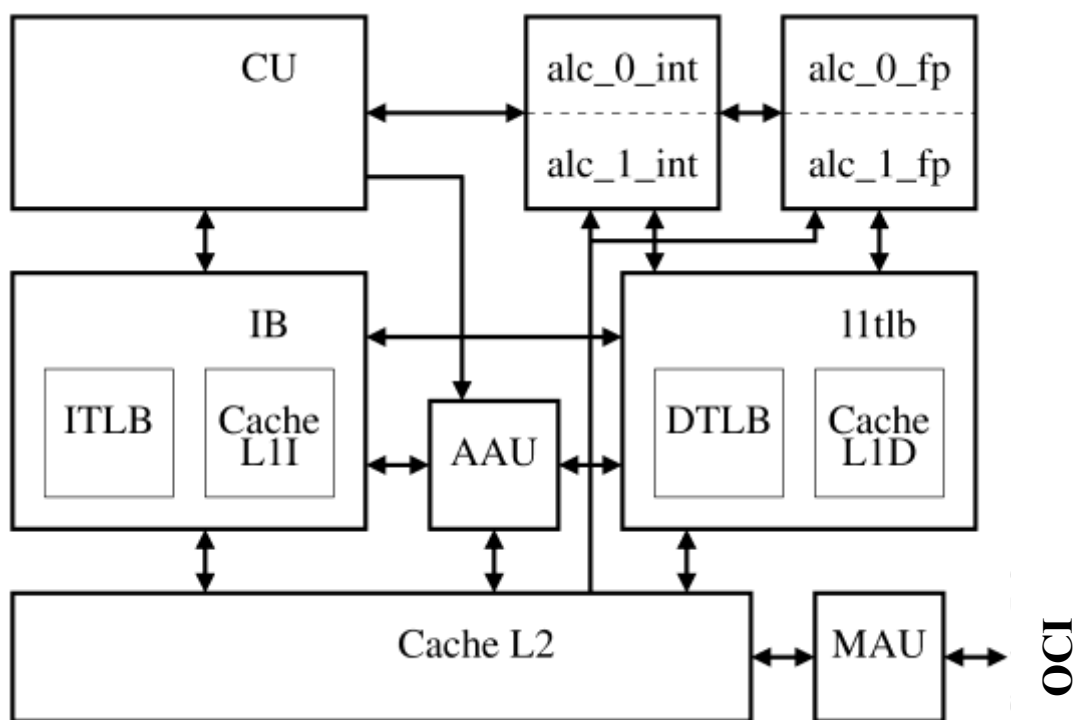


Рисунок 2.1 — Вычислительное ядро

Вычислительное ядро содержит следующие устройства:

- устройство управления CU;
- буфер команд IB, включающий:
 - кэш-память команд первого уровня Cache L1I;

- кэш-память таблицы страниц ITLB для трансляции виртуальных адресов команд в физические;
- устройства целочисленной арифметики alc_0/1_int, организованные в виде двух кластеров (0-й и 1-й) по три канала в каждом;
- устройства вещественной арифметики alc_0/1_fr, организованные в виде двух кластеров (0-й и 1-й) по три канала в каждом; также эти устройства выполняют целочисленные и вещественные SIMD-операции;
- блок - l1tlb, выполняющий первичную обработку заявок в память. Блок включает:
 - кэш-память данных первого уровня Cache L1D;
 - кэш-память таблицы страниц DTLB для трансляции виртуальных адресов данных в физические;
 - блок AAU, оптимизирующий обращения к элементам массивов;
 - унифицированная кэш-память команд и данных второго уровня Cache L2;
 - блок MAU, предназначенный для реализации системного протокола обращения в следующие уровни подсистемы памяти;
 - контроллер JTAG;
 - контроллер BIST;
 - распределенный логический анализатор.

2.1.2 Обзор архитектуры широкого командного слова

2.1.2.1 В процессорном ядре микропроцессора реализована архитектура широкого командного слова, которая предоставляет компилятору явные средства для использования и управления работой параллельной аппаратуры микропроцессора. Компилятор анализирует внутренний программный параллелизм и генерирует параллельный код с учетом информационных и ресурсных зависимостей. В том случае, когда информационная зависимость определяется динамически, это явным образом отображается в коде программы.

Архитектура процессорного ядра позволяет в различных сочетаниях выполнять в одном такте 12 арифметико-логических операций, четыре обращения в память, три операции обработки булевских значений, условную передачу управления, приращение счетчика цикла, две групповые операции подкачки элементов массива в буфер предварительной подкачки с продвижением адресов, четыре считывания элементов массива из буфера предварительной подкачки в регистровый файл, вычисление шести условий, управляющих условным выполнением команд. Ограничением является максимальный размер широкой команды – до 64 байт. В результате, в одной широкой команде может содержаться не более 25 операций с данными формата 64 и не более 61 операции с упакованными данными формата 32.

2.1.2.2 Явное выражение параллелизма в языке машины. В архитектуре широкого командного слова параллельное выполнение представлено явно в коде программы. Широкая команда содержит набор операций, которые одновременно дешифрируются и параллельно выполняются, каждая в своем отдельном конвейере с известным временем выполнения. Это дает возможность компилятору выполнять практически все планирование статически, во время трансляции, определяя все информационные зависимости и учет занятости ресурсов, управлять каждым тактом. Благодаря параллельному коду (представлению программы в виде последовательности широких команд, содержащих наборы параллельно выполняемых операций), компилятор может достаточно ясно и точно (с точностью до такта) представить результаты своего планирования для случая параллельного выполнения программы.

В процессе работы аппаратура с помощью широкой команды осуществляет только запуск исполнительных устройств. Кроме того, вся организация процессорного ядра выполнена таким образом, что вся дальнейшая после запуска устройств их работа предсказуема с точностью до такта процессора. Это делает возможным освободить аппаратуру от заботы о взаимных блокировках устройств и возложить ее на компилятор.

С точки зрения программы процессор состоит из исполнительных устройств: арифметико-логических устройств, адресных регистров с аппаратурой их продвижения и устройства запуска обращений в память как вне, так и внутри цикла. Все эти устройства работают на четырех видах регистровой памяти: регистровом файле, адресных регистрах, буфере предварительной подкачки массивов и регистре предикатов. Различные устройства, работающие на одном канале, могут иметь статически известные, но различающиеся времена исполнения.

С учетом конвейерной работы всех устройств, компилятор должен следить, чтобы отсутствовали конфликты по ресурсам. В частности, например, две разные по времени операции, запущенные в одном канале не должны заканчиваться в одном и том же такте и т.д. Ошибки компилятора в планировании не нарушают корректности, но приводят к снижению эффективности. Результаты выполнения с выходов устройств на их входы поступают:

- через регистровый файл;
- через байпасы с выходов на входы непосредственно.

Последний вариант самый быстрый. Это свойство в сочетании с универсальным набором устройств в кластерах позволяет компилятору размещать большинство критических путей на этом наиболее быстродействующем пути.

2.1.2.3 Реализация целочисленных и вещественных операций. Процессорное ядро микропроцессора имеет шесть каналов для выполнения арифметико-логических операций. Каждый канал содержит набор исполнительных устройств для выполнения целочисленных и вещественных операций. Регистровый файл является общим для хранения результатов целых и вещественных операций. В связи с этим, каждый канал имеет три выделенных порта чтения операндов и один порт записи результатов операций. Каждый такт канал может принимать одну целочисленную или вещественную операцию. Организация конвейера канала обеспечивает его постоянную готовность к приему очередной операции. Исключение составляет операция деления, для которой имеются ограничения на темп обработки.

2.1.2.4 Ускорение выполнения вычислений. Архитектура широкого командного слова, благодаря параллельной работе аппаратуры и глубокому анализу программы и тщательному планированию ее выполнения транслятором на стадии компиляции, позволяет значительно повысить скорость счета как скалярных, так и векторных задач. Для скалярных вычислений основным фактором, ограничивающим скорость счета таких задач, является информационная зависимость между различными операциями. Для векторных вычислений, как правило, ограничивающим фактором является не столько информационная зависимость, сколько общий объем вычислений и недостаток вычислительных ресурсов (в основном исполнительных устройств).

Спекулятивный и условный режимы выполнения команд, а также выполнение передач управления с использованием предварительной подкачки кода в направлении передачи управления способствуют ускорению выполнения скалярного кода.

Спекулятивный режим выполнения операций позволяет компилятору перемещать выше по коду отдельные операции и фрагменты программы и выполнять их параллельно на фоне текущих вычислений, что сокращает общее время выполнения программы. Например, спекулятивное чтение из памяти скрывает задержку доступа в память на фоне выполнения текущих вычислений. Спекулятивный режим выполнения арифметических операций позволяет производить параллельно вычисления сразу по нескольким альтернативным ветвям программы на фоне вычисления условия ветвления. Спекулятивный режим не вызывает прерываний, а результаты выполнения спекулятивных операций могут использоваться после вычисления условий их выполнения.

Условный режим выполнения операций часто заменяет условную передачу управления. Условный режим выполнения операций позволяет управлять выполнением операции по заданному и ранее вычисленному условию (аналогично использованию условия в операции условной передачи управления). В частности, так может быть выбрано для продолжения вычислений одна из спекулятивно выполненных ветвей программы.

Предварительная подкачка кода в направлении ветвления (на фоне выполнения основной ветви) скрывает задержку по доступу к коду при передачах управления и, тем самым, позволяет выполнить передачу управления без остановки конвейера выполнения, когда уже известно условие ветвления.

Файл предикатов дает возможность параллельного вычисления нескольких условий. Это важно как для скалярных вычислений, обеспечивая условное выполнение операций, так и для работы в циклах, содержащих условные ветви.

Выполнение циклов методом программного конвейера является действенным методом ускорения векторных вычислений. Последовательные итерации цикла выполняются с некоторым наложением друг на друга. Шаг, с каким итерации накладываются, определяет темп выполнения итераций, который может быть существенно выше строго последовательного выполнения итераций. В каждом шаге программного конвейера выполняются операции из разных наложенных итераций, и их количество может быть большим. Для их выполнения может потребоваться одна или несколько широких команд. Как отмечалось выше, широкая команда «вмещает» в себя до 25 различных операций с 64-разрядными данными и до 61 операции с 32-разрядными данными (за счет использования операций с 32-разрядными данными, упакованными в формате 128), что эффективно используется при конвейерном выполнении циклов.

Обращения к массивам в памяти является важным моментом в векторных вычислениях. Здесь играют роль большое и переменное время доступа и необходимость на длительное время резервировать ячейки регистрового файла под вызываемые элементы массивов, что снижает эффективность использования регистрового файла. В микропроцессоре введен буфер предварительной подкачки, в который данные из массивов вызываются отдельной ветвью программы, выполняемой параллельно на фоне вычислений. Имеются специальные средства управления, позволяющие подстраивать опережение вызова элементов массивов по отношению к их использованию в вычислениях.

Вызванные элементы массивов хранятся в буфере подкачки и передаются в регистровый файл из основной ветви программы непосредственно перед исполь-

зованием. В регистровом файле определена область для загрузки элементов массивов в цикле с базой, определяющей начало данных для текущей итерации цикла. Это позволяет обращаться к данным различных итераций цикла путем соответствующей индексации, поскольку в широкой команде могут присутствовать операции, относящиеся этим различным итерациям.

2.1.2.5 Организация сверхоперативной памяти. Сверхоперативная память имеет многоуровневую организацию. Она включает в себя регистровый файл RF, буфер предварительной подкачки массивов APB, кэш команд первого уровня I\$, кэш данных первого уровня L1D\$, кэш второго и уровня L2\$, общий для команд и данных.

Регистровый файл RF состоит из 256 128-разрядных регистров. Время доступа – один такт. 20 портов чтения и 10 портов записи. Регистровый файл RF предназначен для хранения глобальных данных (32 регистра) и областей данных последовательно запущенных процедур (верхней части стека процедур – 224 регистра). Глобальные данные и область данных текущей процедуры доступны для выполнения вычислений в текущей процедуре. Текущей процедуре может быть выделено окно размером до 64 регистров для локальных данных процедуры и окно размером до 128 регистров с продвигаемой в циклах базой для хранения подкачиваемых элементов массивов. Переполнение регистрового файла вызывает откачку его части в продолжение стека процедур в памяти, опустошение - подкачку из продолжения стека процедур в памяти.

Регистровый файл RF обеспечивает в каждом такте для каждого из шести арифметических каналов чтение трех аргументов, чтение двух аргументов для записи в память, запись шести результатов (по одному из каждого арифметического канала) и запись четырех чисел, считанных из памяти.

Буфер предварительной подкачки массивов APB, объем 4 Кбайт, состоит из двух блоков по 2 Кбайт, время доступа четыре такта. Имеет два канала подкачки массивов буфер и четыре канала пересылки элементов массивов из буфера APB в регистровый файл RF. Можно определить до 64 массивов подкачки. Подкачка порциями по 8, 16 или 32 байта, чтение в формате элементов массивов. Для каж-

дого из подкачиваемых массивов в APB организуется FIFO буфер. Канал подкачки массивов наполняет такой FIFO буфер, а канал пересылки передает данные из FIFO буфера в регистровый файл RF (опустошает FIFO буфер).

Двухканальный по чтению и однопортовый по записи кэш команд первого уровня I\$. Объем 128 Кбайт, время доступа три такта. Чтение программного кода блоками по 64 байта. Одновременно может быть считан код для основной ветви и в направлении подготавливаемой передачи управления (в пределах 256 байт в сумме). Подкачка программного кода блоками по 256 байт (восемь запросов в память по 32 байта). Обращение по виртуальному и физическому адресам.

Четырехпортовый кэш данных первого уровня L1D\$, объем 64 Кбайт, время доступа три такта, частичная ассоциативность четыре колонки, размер блока данных 32 байта, заведение при чтении, запись в память насквозь (write-through). Обращение по виртуальному и физическому адресу.

Четырехканальный кэш второго уровня L2\$, объем 1 Мбайт, время доступа девять тактов, сет-ассоциативность четыре колонки, размер блока данных 64 байта, заведение при чтении и записи, запись в память при замещении (write-back). Обращение по физическому адресу.

Все три кэша поддерживают согласованное (когерентное) состояние в многопроцессорной системе.

2.1.2.6 Виртуальная память. В микропроцессоре реализована общая для команд и данных виртуальная память объемом 256 Тбайт (48 разрядов виртуального адреса). Физическая память также имеет объем 256 Тбайт (48 разрядов физического адреса). Поддерживаются три размера страниц - 4 Кбайта, 2 Мбайта и 1 Гбайт.

Архитектура предоставляет два различных виртуальных пространства - первичное виртуальное пространство и вторичное виртуальное пространство.

Первичное (primary) пространство - это основное рабочее пространство архитектуры Эльбрус. В нем разворачиваются коды и данные ОС и запускаемых ею пользователей. Вторичное виртуальное пространство, главным образом, предназначено для эмуляционного исполнения приложений, скомпилированных для

иных архитектур. Существенным отличием от первичного пространства является применение альтернативного алгоритма трансляции виртуальных адресов в физические, что предполагает использование независимой таблицы страниц.

Тип адресуемого виртуального пространства определяется кодом операции обращения в память или разметкой адресных переменных. Для каждого пользователя организуется свое адресное пространство, маркируемое целым числом - идентификатором процесса. Аппаратно поддержан 12-разрядный идентификатор процесса (pid), который дополняет виртуальный адрес в разнообразных кэширующих структурах, что позволяет одновременно хранить в них записи, относящиеся к разным пользователям.

Таблица страниц РТ устанавливает соответствие виртуальных и физических адресов страниц. В первичном виртуальном пространстве использован «линейный» вариант организации таблицы страниц с загрузкой ее в виртуальную память. Четыре уровня таблицы страниц загружены в ту же виртуальную память задачи, которую они же сами описывают. Трансляция виртуальных адресов таблицы выполняется в точности по тому же правилу, что и любого другого адреса задачи. Адресация к элементам таблицы осуществляется через указатель, индексированный виртуальным номером страницы. Формат строки таблицы страниц РТЕ для всех уровней - 64 разряда. Таким образом, последний третий уровень таблицы, описывающий исходное 48-разрядное виртуальное пространство, занимает $2^{27} \cdot 4$ Кбайт страниц. Второй уровень таблицы, содержащий РТЕ, описывающие собственно этот последний уровень таблицы, занимает уже $2^{18} \cdot 4$ Кбайт страниц. Следующий первый уровень – $2^9 \cdot 4$ Кбайт страниц. Наконец последний нулевой уровень занимает одну 4 Кбайт страницу и адресация к нему осуществляется через указатель по физическому адресу.

Во вторичном виртуальном пространстве реализованы таблицы страниц для различных сочетаний 32-разрядных и 48-разрядных виртуальных и физических адресов.

2.1.2.7 Организация работы виртуальной машины. Понятие «виртуальная машина» обозначает режим исполнения программ (включая и системное ПО), в

котором истинные аппаратные ресурсы изолированы от них дополнительным программным слоем - гипервизором. Для исполняемого системного ПО эта изоляция может быть невидимой, и оно считает, что распоряжается непосредственно аппаратными ресурсами (оперативной памятью, устройствами ввода-вывода и т.д).

Однако в действительности оно распоряжается виртуальными ресурсами, предоставляемыми ему гипервизором. Гипервизор формирует и запускает виртуальную машину (также используется термин «гость») с заказанной конфигурацией и далее отображает ее виртуальные ресурсы на реальные аппаратные. Такой подход позволяет запустить несколько независимых виртуальных машин одновременно или в режиме разделения времени.

Архитектура определяет набор средств, обеспечивающих изоляцию виртуальной машины от аппаратных ресурсов:

- дополнительный режим исполнения, отличающий гостя от гипервизора. Этот режим дополняет привычные привилегированный/непривилегированный режимы и ортогонален им, то есть программные компоненты гипервизора и гостя могут исполняться как в непривилегированном, так и в привилегированном режиме;

- операция запуска гостя/виртуальной машины - GLAUNCH. Операция автоматически сохраняет оперативную часть контекста гипервизора в стеках и теневых регистрах, загружает управляющие регистры значениями из гостевого контекста, подкачивает гостевые стеки из памяти и передает управление на гостя. Одна и та же операция служит как для первоначального старта гостя/виртуальной машины, так и для возврата в гостя/виртуальной машины после перехвата (см. ниже);

- изоляция гостя от аппаратных ресурсов реализуется так называемым механизмом «перехвата» избранных гостевых событий, то есть автоматической передачи управления гипервизору в случае попытки гостя выполнить некое действие. Этот механизм похож на вход в прерывание, но отличается полным переключением контекста и режима исполнения. Среди перехватываемых собы-

тий, прежде всего, нужно назвать обращение к физической оперативной памяти, к контроллеру прерываний и к устройствам ввода-вывода. Также бывает необходимо перехватывать обращения к некоторым управляющим регистрам - главным образом задающим режимы работы аппаратуры. Наконец, для имитации непрерывности и монотонности астрономического времени необходимо корректировать значения, считываемые гостем с регистров, имитирующих часы или подсчитывающих некоторые события.

Механизм перехвата обеспечивает автоматическое полное переключение контекста гостя на контекст гипервизора и соответствующую смену режима исполнения;

- для создания у гостя полной иллюзии монопольного владения оборудованием, перехват, после обработки гипервизором завершается возвратом в гостя. При этом часто необходимо подменить некоторые элементы его контекста, например, значение, записанное гостем в управляющий регистр;

- в аппаратуру введен механизм двухуровневой трансляции виртуальных адресов гостя: сначала гостевой виртуальный адрес VA транслируется в гостевой физический адрес GPA, а затем GPA транслируется в физический адрес супервизора PA. Первая трансляция осуществляется типовым образом по таблице страниц PT, которую формирует и поддерживает гостевое СПО. Вторая трансляция осуществляется по таблице страниц, которую формирует и поддерживает гипервизор.

Второй уровень трансляции выполняется «невидимо» для гостя, поддерживая стратегию изоляции гостя от реальных аппаратных ресурсов.

2.1.2.8 Защищенные вычисления. Архитектура микропроцессора

Эльбрус-2С3 включает аппаратные средства для организации защищенных вычислений. Использует метод «контекстной защиты», заключающийся в том, что каждый активный агент (это может быть пользователь, программа, процедура и т.п.) имеет перечень известных ему объектов со спецификацией прав доступа. Физически это выражается в использовании набора некоторых служебных слов - дескрипторов, каждый из которых описывает местоположение своего объекта, а

также содержит необходимые признаки. Доступ к объектам возможен только посредством дескрипторов.

Дескрипторы являются системной информацией и не могут быть модифицированы агентом иначе, как в сторону сокращения прав. Равным образом, агент не может расширить список (свой контекст) иначе, как получив доступ к объекту в качестве параметра, либо запросив новый объект у системы. Агент может передавать свои дескрипторы другому агенту (возможно, с сокращением прав доступа), тем самым, образуя нужную конфигурацию доступности. Для этого в архитектуре адресные данные отличаются от числовых данных, то есть типизированы.

В архитектуре используется динамическая типизация или тегирование. Она реализуется добавлением к информационным разрядам дополнительного поля - тега, содержащего тип. Минимальный нечисловой тип занимает 4 байта, поэтому тег добавляется к 4-байтовому слову. Слово в памяти, на регистрах и в шинах сопровождается 2-разрядным тегом. В архитектуре определено три базовых формата нечисловых данных: одинарное слово, двойное слово, квадратное слово, причем все слова, образующие составные форматы должны иметь одинаковый тип.

Таким образом, тег слова кодирует следующее:

- 0 - слово содержит числовую информацию, либо само по себе, либо являясь фрагментом составного формата;
- 1 - слово содержит нечисловую информацию формата одинарное слово;
- 2 - слово содержит фрагмент нечисловой информации формата двойное слово;
- 3 - слово содержит фрагмент нечисловой информации формата квадратное.

Длина и тип числовых данных архитектурно не различимы. Слово числового типа может содержать как переменную типа `char`, так и фрагмент переменной типа `longint` или `longdouble`. Семантическое наполнение числовой переменной отслеживается компилятором и проявляется, когда она становится операндом какой-либо операции.

В отличие от числовых данных, нечисловые данные строго типизируются. Прежде всего, гарантируется целостность составных форматов (двойное слово и квадрат), что предполагает, что:

- все фрагменты должны иметь тип, соответствующий формату;
- при любых манипуляциях фрагменты должны сохранять свой порядок.

Второе качество реализуется выровненным расположением данных, как в памяти, так и в оперативных регистрах процессора. Так, младшее слово адресной переменной формата двойное слово всегда помещается по адресу, кратному 8-ми байтам, а старшее - следом за ним, смежным образом. Пересылки отдельных фрагментов приводят к разрушению их типов (превращению в неспециализированные данные). Операции, требующие специализированных операндов (например, адресных данных), строго контролируют их тип и целостность.

В рамках каждой из форматных групп конкретный тип специализированных данных определяется значением выделенной группы информационных разрядов - внутреннего тега.

2.1.2.9 Стеки. Данные, доступ к которым организован по методу стека, подразделяются на три группы по своим функциональным характеристикам и уровню доступности для пользователя:

- параметры, локальные данные и промежуточные значения процедуры, размещенные в оперативных регистрах;
- параметры и локальные данные процедуры, размещенные в памяти;
- «связующая информация», описывающая предыдущую (запустившую) процедуру в стеке процедур.

Для каждой из групп выделяется отдельный стек (они перечислены в том же порядке):

- стек процедур;
- стек пользователя;
- стек связующей информации.

Стек процедур предназначен для данных, вынесенных на оперативные регистры. Это могут быть параметры, локальные данные и промежуточные значения.

Каждая процедура работает только в своем окне, которое может пересекаться с предыдущим окном областью параметров (она же является областью возвращаемых значений). Верхняя часть стека (окно текущей процедуры - обязательно, данные предыдущих процедур - возможно) расположена на регистрах, оставшаяся часть продолжается в памяти.

Обращение за данными (для пользователя) возможно только в текущее окно, всегда расположенное на оперативных регистрах. Регистры адресуются с помощью индекса относительно начала окна. Индекс может быть как абсолютный, так и вычисляемый относительно дополнительной базы, предназначенной для обработки циклов.

Стек пользователя предназначен для данных, которые пользователь считает нужным разместить в памяти. На регистр процессора загружен дескриптор, описывающий свободную память под стек пользователя. Введена специальная операция, которая резервирует память требуемого размера (фрейм) и возвращает в качестве результата дескриптор на зарезервированную для текущей процедуры область, а также уменьшает свободную часть стека пользователя. Освобождение памяти выполняется автоматически при возврате из процедуры.

Поскольку в стеке виртуальная память переиспользуется, встает проблема защиты данных. Она имеет два аспекта: 1) переиспользование памяти (выделение ранее освобожденного пространства), 2) "зависшие" указатели. Первая проблема решается автоматической очисткой переиспользуемой памяти. Принцип решения второй проблемы следующий. Указатели на текущий фрейм процедуры можно сохранять только в текущем фрейме, либо передавать в качестве параметра в вызываемую процедуру (передавать вверх по стеку). Соответственно, указатель не может быть ни записан в глобалы, ни передан в качестве возвращаемого значения, ни записан в глубину стека. Это касается как стека пользователя, так и стека процедур.

Стек связующей информации предназначен для размещения информации о предыдущей (вызвавшей) процедуре и используемой при возврате. При защищенном программировании пользователь не имеет возможности изменять эту

информацию, поэтому для нее выделен специальный стек, доступный только операционной системе и аппаратуре. Стек связующей информации устроен так же, как стек процедур, то есть верхняя часть его размещена на специальных регистрах, а продолжение - в памяти. Область памяти, зарезервированная под стек связующей информации, описывается указателем стека связующей информации, загруженным на регистр процессора. Указатель стека связующей информации состоит из дескриптора и индекса. Дескриптор описывает всю зарезервированную память, а индекс указывает на верхнюю границу той части стека, которая расположена в памяти. Регистровая часть стека описывается указателем аппаратной вершины стека связующей информации, также загруженным на регистр процессора. Перекачка данных между памятью и регистрами осуществляется автоматически.

2.1.2.10 Блокировка и мониторинг фрагментов памяти. Фрагменты памяти длиной байт, полуслово, одинарное слово, двойное слово или quadro слово могут быть заблокированы, чтобы регистрировать неожиданные обращения к этому фрагменту, как от своего микропроцессора, так и от других микропроцессоров многопроцессорной системы (термин «свой» означает процессор, который заблокировал фрагмент). Операция является конфликтующей с блокировкой, если адресуемый фрагмент пересекается с заблокированным фрагментом. Блокировки памяти реализуются посредством ассоциативных таблиц DAM, SLT, MLT и MPT, в которых каждая строка соответствует защищаемому фрагменту памяти.

Таблица DAM используется для проверки корректности перестановок операций считывания по коду выше предшествующих операций записи. Операция считывания со специальным признаком блокировки «lock» выполняет обращение в память, а также заносит в DAM физический адрес и номер регистра результата (устанавливает блокировку). Последующие операции записи сравнивают свои физические адреса с адресами, хранимыми в DAM и, в случае сравнения, сбрасывают признак блокировки. Повторная контролирующая операция считывания с таким же регистром результата и специальным признаком проверки выполняет ассоциативный поиск по заданному номеру регистра результата и ана-

лизирует состояние блокировки. В случае сохранности блокировки операция считывания не выполняется, так как заблокированный фрагмент не был модифицирован со времени установки блокировки. В случае сброса блокировки контролирующая операция считывания выполняется повторно, так как после установки блокировки фрагмент был модифицирован и требуется его повторное считывание. Это же условие сброса блокировки может использоваться и для передачи управления на некий компенсирующий код, если по алгоритму программы некорректный результат считывания был использован в вычислениях и требуется провести коррекцию вычислений с использованием уже модифицированного значения.

Таблица SLT используется при программировании атомарных модификаций семафора. Модификация семафора заключается в считывании из памяти, изменении его состояния и обратной записи в память. Таблица SLT блокирует обращения к ячейке семафора в памяти от других микропроцессоров на период времени между двумя этими операциями. Операция считывания с признаком блокировки "lock wait" выполняет чтение значения (семафора) из памяти, заносит физический адрес в строку таблицы SLT и устанавливает признак блокировки (захватывает адрес семафора). Признак блокировки, установленный в одном процессоре первым, запрещает последующим операциям считывания с признаком блокировки «lock wait» в других процессорах устанавливать блокировку в своих таблицах SLT в своих процессорах.

Операция записи с совпадающим физическим адресом в случае наличия блокировки выполняет запись модифицированного семафора в память и сбрасывает блокировку. Операция записи при отсутствии блокировки не выполняется, так как предшествующая попытка захватить адрес семафора была неудачной. Такая операция записи устанавливает предикат «lock_cond», по которому может быть выполнена передача управления на повторную попытку захватить семафор (выполнить чтение семафора с признаком блокировки «wait»).

Таблица MLT позволяет заблокировать обращение по записи или считыванию/записи в заданный фрагмент памяти. Это свойство используется для

некоторых оптимизаций компилятора. Например, в системе битовой компиляции, когда определенные переменные передаются на регистры процессора в целях ускорения вычислений и, поэтому, необходимо запретить доступ в память к фрагментам, где они ранее находились. Нарушение блокировки вызывает выработку особой ситуации «exc_mem_lock».

Кэш таблицы защиты памяти MPT (Memory Protection Table) содержит восемь 64-разрядных слов для хранения флагов доступа к 4 Кбайт областям памяти при обращении во вторичное виртуальное пространство. Каждое слово хранит флаги доступа по записи MPT_WR и считыванию MPT_RD для 32-х 4 Кбайт областей памяти.

2.1.2.11 Средства поддержки совместимости с платформой «x86». Архитектура определяет средства поддержки выполнения программного обеспечения платформы «x86» и «x86-64», перетранслированного в коды «Эльбрус». К ним относятся:

- совместимое представление данных, реализация набора операций, позволяющего эффективно имитировать операции платформы «x86» и «x86-64», включая моделирование стека для вещественных операций;

- два пространства виртуальной памяти;

- совместимый формат таблиц страниц;

- поддержка сегментной адресации;

- поддержка совместимости на уровне управления доступом в память (соблюдение порядка выполнения операций записи и разрешение невыровненных считываний).

2.1.2.12 Поддержка согласованного состояния памяти. Трехуровневая организация памяти (кэши команд и данных первого уровня, кэш второго уровня и оперативная память), а также многопроцессорный режим работы с общей оперативной памятью требуют специальной организации для обеспечения согласованного (когерентного) состояния памяти. Кэш первого уровня использует стратегию заведение при считывании и запись насквозь в кэш второго уровня. Это обеспечивает согласованное состояние кэшей всех уровней. Кэш второго уровня

использует стратегию заведение при считывании и записи, и запись в память только в случае замещения. То есть операции считывания и записи при промахах в кэше второго уровня всегда вызывают данные из памяти в кэш второго уровня, где и выполняется их модификация.

Когерентность доступа в общую память поддерживается посредством запросов проверки когерентности (когерентных запросов), рассылаемых в кэши системы в соответствии с информацией из кэша L2, в которой представлены состояние каждого 64-байтового используемого блока данных.

2.1.2.13 Организация данных и принципы доступа к ним. Архитектура микропроцессора Эльбрус-2С3 позволяет образовывать два независимых виртуальных пространства: первичное (эльбрусовское) и вторичное. Обращения в то или иное пространство определяется кодом операции. В первичном пространстве могут размещаться данные двух типов: незащищенные и защищенные. Незащищенные данные адресуются произвольно, посредством абсолютного виртуального адреса, который представляется целым числом.

Для обращения к защищенным данным используется относительная адресация, посредством специальных адресных данных - дескрипторов и индексов. Данные обоих типов могут быть перемешаны в одном виртуальном пространстве, однако во избежание незащищенных обращений к защищенным данным используется постраничная защита. Страницы в таблице страниц помечаются как содержащие либо защищенные, либо незащищенные данные. Механизмы выделения памяти различаются для защищенных и незащищенных программ и оперируют каждый только своими страницами. При обращении к данным контролируется, что тип адреса соответствует типу страницы.

Область памяти представляется как связанная последовательность байтов и описывается двумя параметрами: адрес первого байта (как правило, имеет название «база» или его сокращения) и количество байтов (как правило, имеет название «размер» или его сокращения). Байты нумеруются от нуля.

2.1.2.14 Организация программного кода и доступ к нему. Различаются защищенные и незащищенные коды.

Защищенные коды организованы в виде модулей компиляции (compilation unit), представляющих собой смежную область памяти и содержащих одну или несколько статически слинкованных процедур со своими константами. Модуль занимает целое количество страниц и описывается дескриптором модуля компиляции CUD (Compilation Unit Descriptor).

Все модули задачи пронумерованы. Номер присваивается модулю на все время исполнения задачи и называется индексом модуля компиляции (Compilation Unit Index - CUI). Дескриптор модуля размещен в таблице модулей компиляции (Compilation Unit Table - CUT), в строку с номером, равным CUI. CUT описывается дескриптором CUTD, загруженным на регистр процессора.

При передаче управления модулю, нужная строка CUT считывается, и дескриптор помещаются на регистр процессора для текущего использования. CUI, присвоенный модулю, записан во всех строках таблицы страниц всех страниц кода, принадлежащих модулю. Таким образом, передача управления процедуре модуля предполагает следующую цепочку действий:

```
target address -> page table -> PTE -> code
      |
      +-> CUI -+
            |
            +-> CUT -> CUD -> регистр процессора
            |
            CUTD -+
```

Поскольку форматы строки таблицы страниц PTE для защищенного и незащищенного кода не отличаются, то для определенности нулевой CUI зарезервирован для незащищенных страниц. Защищенные же модули (страницы) нумеруются, начиная с единицы.

Незащищенные коды размещаются в произвольном месте виртуальной памяти за вычетом областей, отведенных под защищенные коды и данные.

Обращение к незащищенным кодам осуществляется с помощью целого числа, значение которого воспринимается как адрес.

2.1.2.15 Представление данных. С точки зрения разрядности данные могут иметь форматы, приведенные таблице 2.1.

Таблица 2.1 - Форматы данных

Формат	Наименование	Значение
8 бит	байт	целое
2 байта	полуслово	целое
4 байта	одинарное слово	целое, вещественное
8 байтов	двойное слово	целое, вещественное, адресная информация
10 байтов	расширенное	вещественное
16 байтов	квадро слово	адресная информация, упакованные данные

2.1.2.16 Структура широкой команды. Архитектура определяет два типа команд и, соответственно, программ:

- синхронные, или "просто" команды (программы);
- асинхронные команды (программы), предназначенные для предварительной подкачки элементов массивов в циклах.

Асинхронная программа представляет собой несвязанные фрагменты кода, каждый из которых логически связан с некоторым циклом основной программы и предназначен только для циклического исполнения. Очередной фрагмент асинхронной программы запускается и останавливается из основной программы. Программы разного типа образуют разные командные потоки и предназначены для исполнения на разном оборудовании параллельно и асинхронно по отношению друг к другу. Программы разного типа могут содержаться в одном программном сегменте в произвольном порядке, но программист должен гарантировать, что командные потоки не пересекаются.

Команда имеет длину от 8 до 64 байт и выровнена до двойного слова. Она состоит из слогов и полуслогов - нетегированных слов (4 байта) и полуслов (2 байта). Каждый слог, один или в комбинации с другими (полу) слогами, кодирует отдельную операцию.

Последовательность полуслогов упаковывается в слова. Перечень типов (полу) слогов представлен в таблице 2.2.

Таблица 2.2 - Перечень типов слогов и их возможное количество в составе широкой команды

Обозначение	Наименование	Количество в команде
HS	Слог Заголовок	1
SS	Слог коротких операций	1
ALS	Слоги арифметико-логических каналов	до 6
ALES	Полуслоги расширения для арифметико-логических каналов	до 6
CS	Слоги команд управления	до 2
CDS	Слоги условного исполнения	до 3
PLS	Слоги каналов обработки логических предикатов	до 3
LTS	Слоги литералов	до 4
AAS	Полуслоги каналов обращения к элементам массивов	до 6

Слог Заголовок HS присутствует обязательно, прочие слог - по мере необходимости. Определен следующий порядок размещения слогов в широкой команде: HS, ALS0, ALS1, ALS2, ALS3, ALS4, ALS5, CS0, CS1, SS, ALES0, ALES1, ALES3, ALES4, AAS0, AAS1, AAS2, AAS3, AAS4, AAS5, LTS3, LTS2, LTS1, LTS0, PLS2, PLS1, PLS0, CDS2, CDS1, CDS0.

Слог Заголовок HS является обязательной составной частью любой широкой команды, поскольку он содержит информацию о длине и структуре широкой команды, используемую при ее распаковке и дешифрации.

Слог SS предназначен для операций, кодировка которых занимает всего несколько разрядов, например, передача управления или продвижение вращающейся базы регистрового файла.

Каждый слог типа ALS целиком содержит операцию одноименного арифметико-логического канала. Арифметико-логические каналы выполняют арифметические и логические операции, формируют адрес обращения в память, выполняют запись и считыванием регистров процессора. Формат операции традиционный. Слог содержит код операции, адреса операндов и результата, как правило, в регистровом файле. В качестве адреса назначения могут быть указаны также некоторые регистры процессора. Слог содержит специальный разряд, определяющий режим неспекулятивного или спекулятивного выполнения операции. В полях операндов может содержаться или кодироваться литеральное значение.

Полуслоги ALES являются расширением соответствующих слогов ALS и содержат либо просто расширение кода обычной операции, либо код для второго этажа комбинированной двухэтажной операции.

В слогах CS кодируются операции, выполняющиеся в рамках устройства управления или специальные операции. В типовом случае слог CS0 содержит операции подготовки передачи управления со статическим смещением, а слог CS1 - операции быстрой литеральной загрузки некоторых специальных регистров, а также операции синхронизации.

Слоги CDS содержат указания на то, какие из операций в обрабатываемых каналах выполняются условно и под управлением каких предикатов.

Слоги PLS содержат операции обработки логических величин - предикатов, считываемых из файла предикатов. Результат также помещается в файл предикатов.

Слоги LTS содержат литеральные значения, которые могут использоваться арифметико-логическими каналами в качестве операндов. Каждый слог может содержать либо 32-разрядное знаковое значение, либо любую из половин 64-разрядного значения.

Полуслоги AAS кодируют операции, которые пересылают элементы массивов, предварительно подкаченные в буфер, в регистровый файл для обработки.

Кодировка операции может занимать часть слога, слог или состоять из нескольких слогов. Так слог ALS определяет операцию для арифметико-логического канала над операндами из регистрового файла или с использованием короткого литерала в качестве операнда. Добавление слога (слов) LTS позволяет задать литералы длиной 2, 4 или 8 байт. Добавление полуслога ALES позволяет определить комбинированную (составную) операцию или расширить код операции. Добавление слога CDS позволяет задать условный режим выполнения операции, представленной в слоге ALS, и предикат, определяющий возможность выполнения заданной операции.

Операции скалярного обращения в память кодируются также в слогах ALS. Добавление слога CS определяет режимы выполнения обращения в память: без поиска в кэше, без трансляции виртуального адреса в физический адрес, спекулятивный режим выполнения, специальные операции в кэше и таблице страниц типа откачки и чистки, обращения к регистрам подсистемы памяти и т.д.

Слог коротких операций SS, наоборот, позволяет задать сразу несколько коротких операций. В их число входят операция передачи управления (задаются номер регистра подготовки передачи управления и адрес предиката для условной передачи управления), операции продвижения текущих баз во вращающихся областях стека процедур и предикатном файле, а также операции начала и конца выполнения асинхронной программы в устройстве обращения к массивам AAU.

Операция пересылки элементов массивов из буфера предварительной подкачки APB в регистровый файл RF всегда занимает полтора полуслога AAS. Целый слог определяет формат элемента массива и область в APB, а полуслог - адрес назначения в регистровом файле RF.

2.1.2.17 Структура асинхронной команды. Асинхронная команда имеет фиксированную длину - 8 байт и выровнена по границе двойного слова. Аналогично синхронной команде, асинхронная команда состоит из слогов словной длины (4 байта), которые кодируют операции. Различаются два типа слогов, которые должны присутствовать в команде в порядке, приведенном в таблице 2.3.

Таблица 2.3 – Перечень и порядок следования слогов в асинхронной команде

Название	Обозначение
Слог подкачки массива	APS0
Слог литерала для подкачки массива	APLS0
Слог подкачки массива	APS1
Слог литерала для подкачки массива	APLS1

Асинхронная команда может содержать до двух операций подкачки (для левого и правого каналов предварительной подкачки), а также операцию циклического перехода. Слоги APS0 и APLS0 задают операцию предварительной подкачки для левого канала, а APS1 и APLS1 - для правого. В слоге APS задается дескриптор массива и размер кванта подкачки, в слоге APLS - константа для индексации адреса обращения в память (постоянное смещение). Операция циклического перехода передает управление на начало асинхронной программы.

2.1.2.18 Операции широкого командного слова. Целочисленные и вещественные арифметические операции, логические операции и операции сдвига используют, в зависимости от типа операции один, два или три операнда и формируют результат в виде числового значения, набора флагов или предиката (операции сравнения). В качестве операндов могут быть числовые значения форматов слово или двойное слово из регистрового файла или константы, представленные в коде операции. Константы могут кодироваться в поле адреса операнда или как ссылка на дополнительные один или два слога литерала для задания констант форматов 32 и 64 соответственно. Результаты в виде числового

значения или флагов помещаются в регистровый файл, в виде предикатов – предикатный файл.

Определены также вещественные арифметические операции с расширенным вещественным 80-разрядным форматом.

Целочисленные и вещественные операции могут использоваться в комбинированных трехадресных операциях. Комбинированная операция состоит из двух последовательно выполняемых операций. Первая операция выполняет заданную операцию с операндами 1 и 2, вторая – с результатом первой операции и третьим операндом. Результат второй операции заносится по адресу назначения в регистровый файл.

Операции над упакованными значениями используют операнды и формируют результат в формате двойное слово. В упакованном формате могут быть представлены байты и полуслова (только для операций над целыми значениями), одинарные слова и двойные слова. Операции над упакованными значениями определены для целых и вещественных данных.

Операции над предикатами позволяют считать предикаты из предикатного файла, выполнить функцию "логическое И" над двумя предикатами или их инверсиями (что реализует наиболее употребительный набор бинарных логических функций), записать вычисленный предикат в предикатный файл или направить его на управление условным выполнением арифметических операций или операций передачи управления.

Операции обращения в память выполняют считывание и запись значений в память. Определены форматы значений байт, полуслово, одинарное слово, двойное слово и quadro слово.

Обращения в память сопровождаются спецификатором, задаваемым явно или формируемым по умолчанию и определяющим режим обращения в память. Спецификатор по умолчанию задает обращение в память с размещением данных в кэшах процессора. Спецификатор, явно определенный в дополнительном слого, позволяет задать обращения в память минуя кэши процессорного ядра, без транс-

ляции виртуального адреса в физический, с проверкой блокировки фрагментов памяти в таблицах DAM, SLT и MLT, в область ввода-вывода.

Адрес обращения в память формируется как результат индексации дескриптора. Операнд 1 задает дескриптор – это значение формата quadro, операнд 2 задает индекс – это значение формата одинарное слово. Операнд 3 в операциях записи – записываемое в память значение.

Обращения в память к глобалам и коду выполняется с использованием аппаратных регистров, хранящих соответствующие дескрипторы. В этих случаях операнды 1 и 2 – индексы формата одинарное слово.

В операциях обращения в память к массиву адреса дескриптора и индекса задаются в устройстве обращения к массивам AAU, а литеральное смещение задается в дополнительном литеральном слоге.

В операциях обращения в память в незащищенное пространство дескриптор и индекс – это операнды формата двойное слово.

Операции преобразования адресных объектов выполняют индексацию заданного дескриптора и помещают результат в регистровый файл.

Операции с тегом выполняют считывание и установку внешнего тега для заданного значения.

Операции обращения к управляющим регистрам выполняют чтение и запись заданных управляющих регистров.

Операции подготовки передачи управления формируют значение указателя команды для предстоящей передачи управления и выполняют подкачку кода в заданный регистр подготовки передачи управления.

Операции передачи управления выполняют переключение дешифрации команд на заданное подготовленное направление и формируют новое регистровое окно в стеке процедур и стеке связующей информации (для процедурных передач управления).

Операции поддержки наложений цикла выполняют установку и продвижение управляющих регистров выполнения цикла.

Операции над кэшами позволяют очистить заданный кэш или его строку, а также прочитать значение заданной строки кэша.

2.1.2.19 Аппаратные команды. В определенных динамических ситуациях процессорное ядро выполняет некоторые действия, не описанные в программном коде (откачка / подкачка регистрового файла, вход в прерывание). В этих случаях формируются соответствующие аппаратные команды.

2.1.2.20 Прерывания. Прерывания могут быть программными и аппаратными. Программное прерывание инициируется специальной операцией и подобно вызову процедуры. Аппаратное прерывание инициируется аппаратурой и может принадлежать к одной из следующих категорий:

- точное - команда, которая вызвала прерывание, не произвела видимого эффекта и может быть повторно выполнена после возврата из обработчика прерывания;

- отложенное - команда, которая вызвала прерывание, произвела некоторый видимый эффект и не может быть повторно выполнена после возврата из обработчика прерывания. Однако, сохранена информация о конкретной операции, которая вызвала прерывание. Используя эту информацию, причина прерывания может быть устранена;

- неточное - команда, которая вызвала прерывание, произвела некоторый видимый эффект. Отсутствует информация (аргументы, окружение, т.п.) о конкретной операции, которая вызвала прерывание, поэтому она не может быть повторена или смоделирована программными средствами;

- асинхронное - прерывание, которое случается асинхронно по отношению к командному потоку, например, внешнее прерывание.

2.1.3 Конвейер выполнения команд

2.1.3.1 Широкое командное слово содержит набор параллельно выполняемых различных операций. В связи с этим конвейер выполнения широких команд

имеет сложную структуру, позволяющую выполнять параллельно различные составляющие операции разной длительности.

Начальная часть конвейера является общей для всех операций широкой команды и включает 8 восемь регистровых станций в соответствии с рисунком 2.2 и таблицей 2.4. После их прохождения каждая из операций широкой команды поступает в свою параллельную ветвь для продолжения и завершения выполнения.

Такт	1	2	3	4	5	6	7	8
Станция конвейера	L/C	A	F0	F1	S	D	B	R

Рисунок 2.2 – Начальная часть конвейера

Таблица 2.4 - Начальная часть конвейера

Станция конвейера	Описание
L/C	Регистры указателей команд для прямой (Linear) ветви Flat_IP 0, 1 и для передач управления (control transfer) CTPR 1, 2, 3
A	Associative search in I\$. Обращение в память тегов команд ITAG и формирование адреса строки по памяти данных буфера команд IDATA. Обращение в память тегов ITLB и формирование адреса по памяти физических адресов
F0	Fetch 0. Регистры адресов четной и нечетной строк в памяти команд IDATA, регистр адреса памяти тегов ITLB. Формирование адресов обращения в каждый банк памяти команд IDATA. Обращение в память физических адресов ITLB и формирование физического адреса блока программного кода
F1	Fetch 1. Входные регистры в банках памяти команд IDATA. Выборка строки до 256 байтов из памяти команд IDATA
S	Scattering. 4 регистра распаковки S. Распаковка широких команд основного и подготавливаемых потоков на составляющие операции

Станция конвейера	Описание
D	Decode. 4 регистра декодирования D. Выборка широкой команды из 4 потоков для выполнения. Дешифрация операций, вычисление адресов операндов
B	Branch. Регистр B. Передача адресов операндов на регистры адресов стека операндов. Проверка условий передачи управления и коммутация широких команд на стадии D
R	Read. Регистр адреса регистрового файла RF. Чтение операндов из RF

Конвейер выполнения операции загрузки Load для случая чтения из кэша первого уровня L1\$ представлен на рисунке 2.3 и в таблице 2.5.

Такт	1-8	9	10	11	12	13
Станция конвейера	L-R	E0	M1	M2	M3	M4

Рисунок 2.3 – Временная диаграмма операции загрузки из кэша первого уровня

Таблица 2.5 – Временная диаграмма операции загрузки из кэша первого уровня

Станция конвейера	Описание
L-R	Начальная часть конвейера
E0	Входной регистр AU. Вычисление адреса
M1	Memory 1. Входной регистр адреса L1D\$. Обращение в память тегов и данных L1D\$
M2	Memory 2. Выходной регистр данных L1D\$. Hit в L1D\$.
M3	Memory 3. Входной регистр RF. Запись данных в RF
M4	Memory 4. Данные в RF. Считывание данных из RF

Конвейер выполнения операции загрузки Load для случая чтения из кэша второго уровня L2\$ представлен на рисунке 2.4 и в таблице 2.6.

Такт	1-8	9	10	11	12	13	14	15	16	17
Станция конвейера	L-R	E0	M1	M2	M3	M4	M5	M6	M7	M8

Рисунок 2.4 – Временная диаграмма операции загрузки из кэша второго уровня

Таблица 2.6 – Временная диаграмма операции загрузки из кэша второго уровня

Станция конвейера	Описание
L-R	Начальная часть конвейера
E0	Входной регистр АУ. Вычисление адреса
M1	Мемору 1. Входные регистры адреса L1D\$ и DTLB. Обращение в память тегов и данных кэша первого уровня. Обращение в память тегов и строк TLB, формирование физического адреса
M2	Miss в L1D\$. Выходной регистр физического адреса ADDR_OUT TLB. Передача физического адреса по байпасу на входной регистр арбитра L2\$
M3	Выходной регистр арбитра L2\$. Обращение в память тегов и данных L2\$
M4	Выходной регистр L2\$, hit в L2\$
M5	Регистр сборки с данными из буфера записи STB
M6	Регистр сборки с данными из MAU
M7	Регистр сборки с данными из APB (входной регистр RF). Запись в RF
M8	Данные в RF. Считывание данных из RF

Временная диаграмма операции записи Store представлена на рисунке 2.5 и в таблице 2.7.

Такт	1-8	9	10	11	12	13	14	15	16	17
Станция конвейера	L-R	E0	M1	M2	M3	M4	M5	M6	M7	M8

Рисунок 2.5 - Временная диаграмма операции записи

Таблица 2.7 – Временная диаграмма операции записи

Станция конвейера	Описание
L-R	Начальная часть конвейера
E0	Входные регистры целочисленного АУ. Вычисление адреса и его передача в L1D\$ и TLB. Чтение записываемого значения из стека операндов
M1	Адрес на входных регистрах L1D\$ и TLB. Записываемые данные на буферном регистре st_align. Обращение память тегов кэша первого уровня и TLB
M2	Hit в L1D\$. Записываемые данные на нижнем регистре входного буфера DBF в L1D\$ Физический адрес на выходном регистре ADDR_OUT в TLB. Записываемые данные на выходном регистре DATA в TLB
M3	Записываемые данные на верхнем регистре входного буфера DBF в L1D\$. Запись значения в память данных L1D\$ Физический адрес на выходном регистре арбитра L2\$. Записываемые данные в буфере арбитра L2\$. Передача адреса и значения в буфер записи
M4	Hit в L2\$. Адрес записи и данные в буферном регистре STB
M5	Адрес записи и данные в STB
M6	Записываемое значение на выходном регистре STB
M7	Записываемое значение на входном регистре памяти данных L2\$. Запись значения в память данных L2\$
M8	Записываемое значение в L2\$

Временная диаграмма выполнения целочисленной операции представлена на рисунке 2.6 и в таблице 2.7.

Такт	1-8	9	10	11	12	13
Станция конвейера	L-R	E0	E1	E2	E3	E4

Рисунок 2.6 – Временная диаграмма целочисленной операции

Таблица 2.8 – Временная диаграмма целочисленной операции

Станция конвейера	Описание
L-R	Начальная часть конвейера
E0	Входной регистр целочисленного АУ первого уровня. Выполнение операции в исполнительном устройстве первого уровня
E1	Входной регистр целочисленного АУ второго уровня. Выполнение операции в исполнительном устройстве второго уровня
E2	Выходной регистр целочисленного АУ
E3	Входной регистр RF. Запись в RF
E4	Данные в RF. Считывание данных из RF

Временная диаграмма выполнения операции вещественного сложения представлена на рисунке 2.7 и в таблице 2.9.

Такт	1-8	9	10	11	12	13	14	15
Станция конвейера	L-R	E0	E1	E2	E3	E4	E5	E6

Рисунок 2.7 – Временная диаграмма вещественной операции сложения

Таблица 2.9 – Временная диаграмма вещественной операции сложения

Станция конвейера	Описание
L-R	Начальная часть конвейера
E0	Входные регистры целочисленного АУ. Передача операндов на входные регистры вещественного АУ
E1	Входные регистры вещественного АУ. Выравнивание порядков
E2	Входные регистры сумматора. Выполнение сложения
E3	Выходной регистр сумматора. Выбор результата без нормализации или с нормализацией
E4	Выходной буфер вещественного АУ. Сборка результатов целого и вещественного АУ
E5	Входной регистр RF. Запись в RF
E6	Данные в RF. Считывание данных из RF

2.2 Специализированные графические и видео ядра

2.2.1 Микропроцессор Эльбрус-2С3 содержит шесть специализированных блоков (ядер), предназначенных для ускорения обработки и вывода на мониторы видеoinформации: ускоритель трехмерной графики 3D GPU, ускоритель двухмерной графики 2D GPU, ускорители декодирования видео VXD и VP9D, ускорители кодирования видео VXE, VP9E. Некоторые блоки — например, гра-

фический ускоритель 3D GPU можно также использовать для обычных массивных вычислений.

Ускоритель трехмерной графики 3D GPU предназначен для создания растрового (двухмерного) изображения трехмерной сцены. Трехмерная сцена задается в виде координат и свойств вершин, составляющих объекты в трехмерном пространстве, исполнения программ для обработки этих вершин, текстур (растровых изображений, наносимых на трехмерные и двухмерные объекты), исполнения программ для обработки пикселей, получаемых во время растеризации, а также обработки прочих параметров. Как исходные данные трехмерной сцены, так и результирующее изображение, хранятся в памяти, непосредственный доступ к которой имеет устройство 3D GPU.

Низкоуровневого программирования устройства 3D GPU не требуется (и такая возможность не предоставляется) - всё взаимодействие с ним выполняется через программный драйвер посредством использования общепризнанных открытых стандартов для интерфейсов программирования (API), таких как OpenGL и OpenGL ES.

Несмотря на то, что устройство 3D GPU предназначено для обработки трехмерных изображений (благодаря наличию специализированных блоков по обработке вершин, пикселей, текстурных блоков и т.п.), оно может быть использовано для выполнения обычных расчетов на его универсальных вычислительных блоках. Устройство 3D GPU содержит многокластерный SIMD вычислитель, каждое АЛУ которого может эффективно выполнять вещественные операции типа умножение-сложение в 16-битном и 32-битном форматах. Для этого также используется открытый стандарт API OpenCL.

Подключение к коммуникационной сети микропроцессора выполняется высокоскоростными AXI шинами, через которые устройство 3D GPU самостоятельно работает с физической памятью, и отдельной управляющей шиной через которую процессорные ядра «Эльбрус» получают доступ к управляющим регистрам.

Ускоритель двухмерной графики (далее «2D ускоритель»), предназначен для решения ряда задач, возникающих при работе оконной системы: заливка и копирование прямоугольных областей (как в видеопамяти, так и между видеопамятью и памятью процессора с изменением форматов пикселей), наложение полупрозрачных изображений, преобразование цветовых пространств, требуемых при выводе видео, и прочего.

В силу специализации, 2D ускоритель может работать быстрее и потреблять меньше энергии по сравнению с тем, когда та же задача решается универсальным процессором. Для его использования поставляется DDX драйвер для X Windows, но для некоторых применений возможно его непосредственное программирование.

Как и в случае 3D ускорителя, подключение 2D ускорителя выполняется отдельными шинами для доступа к памяти и для доступа к управляющим регистрам. Работа с управляющими регистрами может быть сведена к минимуму за счет использования специального режима работы, когда 2D ускоритель самостоятельно считывает посредством прямого доступа к памяти пакеты с заданиями из памяти процессора.

Блок дисплейных контроллеров предназначен для периодического вывода изображений из кадровых буферов, размещенных в оперативной памяти, на мониторы через различные интерфейсы, такие как LVDS, HDMI, RGB. Поддерживаются различные форматы пикселей, может быть выполнено наложение аппаратных курсоров, обеспечивается получение информации посредством DDC каналов о возможных и предпочтительном разрешениях у подключенных мониторов. Возможно управление яркостью подсветки жидкокристаллических индикаторных панелей. Один из дисплейных контроллеров имеет набор VGA совместимых регистров с целью поддержки режима двоичной трансляции.

Для предотвращения срыва регенерации изображений дисплейный контроллеры имеют максимальный приоритет при обращении к памяти.

Ускоритель декодирования видео VXD предназначен для распаковки сжатого видеоизображений в форматах H.265, H.264, MJPEG и других с разрешением

от 64x64 до 4096x2160 пикселей. Предусмотрена возможность уменьшение финальных изображений до 1/4 от исходного формата.

Ускоритель VXD применяется для отображения и обработки видео, получаемых из сети Интернет и видеокamer. Также может быть использоваться в задачах доступа к удаленному рабочему столу (VNC). Кроме того, поддерживается распаковка статического изображения в формате JPEG разрешением до 16384x16384 пикселей.

Подключение к коммуникационной сети микропроцессора выполняется высокоскоростной AXI шиной, через которую ускоритель VXD самостоятельно работает с физической памятью, забирая оттуда сжатый поток видео, там же сохраняя промежуточные кадры и записывая туда финальные кадры распакованного видео. Отдельная управляющая шина используется для доступа к управляющим регистрам со стороны процессорных ядер «Эльбрус».

Ускоритель кодирования видео VXE позволяет упаковывать видеоизображения в форматы H.265 и H.264. Применяется для организации видеоконференцсвязи, архивного хранения видео и доступа к удаленному рабочему столу (VNC). Поддерживаются разрешения от 128x64 до 4096x2160 пикселей.

Ускоритель VXE подключается к коммуникационной сети микропроцессора несколькими AXI шинами; в силу большей сложности задачи упаковки видео данный блок создаёт большую нагрузку на память по сравнению с декодером VXD.

Ускорители декодирования и кодирования видео Google VP9 G2 и VP9 BigEv2, соответственно предназначены для распаковки сжатого видеоизображений и сжатия видеоизображения в формате VP9. Данный видео формат используется в видео-конференц связи, в подавляющем большинстве интернет браузеров для видео трансляций и видео связи, в охранных системах видео наблюдения, а также для хранения видео архивов, встроен во многие широко распространенные микропроцессоры. Ускорители подключаются к коммуникационной сети микропроцессора несколькими AXI шинами.

2.3 Коммутационная сеть на кристалле OCI (On-Chip Interconnect)

2.3.1 Для соединения различных устройств на верхнем уровне микропроцессора Эльбрус-2С3 — процессорных ядер, НМУ, ХМУ, GPU, кодеков — требуется сложная сеть, умеющая передавать пакеты различных размеров и типов. При этом она должна обладать достаточной пропускной способностью, низкими задержками и уметь распределять пропускную способность узких мест системы таким образом, чтобы соблюдались приоритеты между потоками, а между потоками одного приоритета распределение было оптимальным и справедливым (fairness), т.е. обеспечивающим одинаковое качество обслуживания.

С учётом особенностей микропроцессора Эльбрус-2С3 в качестве таковой выбрана сеть топологии 1D mesh (рисунок 2.8), разделённая на несколько слоёв для разных размеров пакетов, поддерживающая произвольное число виртуальных каналов (здесь они называются типами), два уровня приоритета и по логическому устройству соответствующая распределённому коммутатору с отдельными очередями для пакетов разных типов.

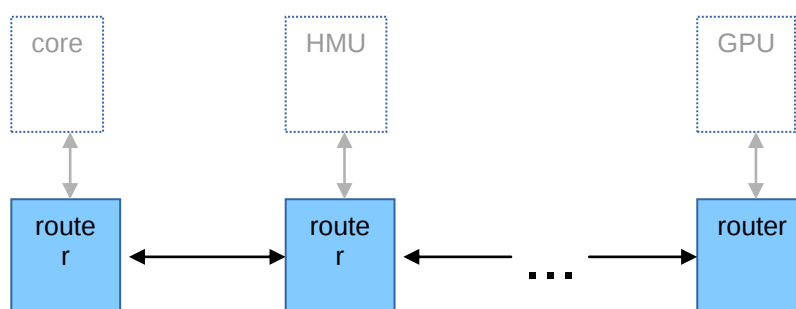


Рисунок 2.8 - Сеть топологии 1D mesh с подключенными устройствами

К локальным интерфейсам могут подключаться сами устройства-абоненты, переходники на их различные интерфейсы, пересинхронизаторы, обычные коммутаторы. Интерфейсы самой сети с учётом этого сделаны достаточно универсальными, не привязанными к конкретному протоколу.

Верхний модуль сети — роутер (рисунок 2.9) — включает в себя роутеры разных слоёв сети, интерфейсы которых одинаковы с точностью до значений параметров. Между роутерами могут быть проходные регистры.

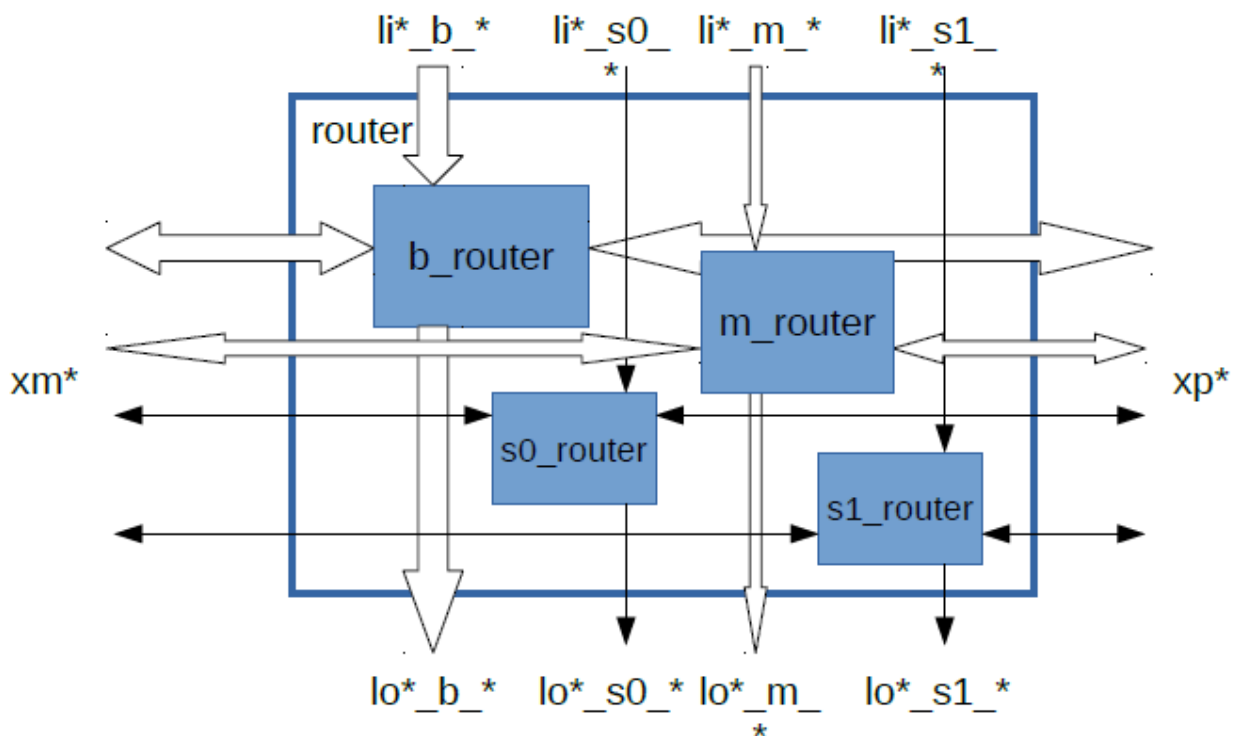


Рисунок 2.9 – Верхний модуль, 4 роутера внутри, внешние интерфейсы

Внутреннее устройство роутеров выбрано для каждого слоя отдельно в соответствии с особенностями слоёв:

- **b**(ig) – пакеты данных; не требуют сохранения порядка (см. ниже) и приоритетов;

- **m**(edium) – пакеты первичных и снуп-запросов; порядок и приоритеты соблюдается;

- s*(mall) – все прочие, короткие пакеты; не требуют порядка, приоритетов или справедливости (fairness).

Сохранять порядок пакетов требуется первичным запросам. Это значит, что любые два пакета с одинаковыми начальным и конечным адресом и номером типа выходят из сети в том же порядке, в котором вошли в неё.

Приоритеты и последовательность предполагаются необходимыми только первичным запросам, поскольку остальной трафик является производным от них.

Разные типы должны назначаться пакетам разных виртуальных каналов принятого протокола обменов (например, данным по чтению и по записи, первичным и снуп-запросам) для предотвращения дедлоков и, плюс к этому, пакетам разного приоритета — для строгого их соблюдения. На вход сети пакеты разных типов должны подаваться из разных буферов.

2.4 Устройство доступа к оперативной памяти НМУ

2.4.1 Подсистема памяти микропроцессора Эльбрус-2С3 содержит устройство НМУ (Home Memory Unit – устройство доступа в свою память), к которому подключены два контроллера оперативной памяти МС. Основным функциональным назначением устройства НМУ является обеспечение когерентного доступа в оперативную память от различных абонентов.

Основные функции устройства НМУ:

- отвечает за сериализацию всех запросов в свою память;
- отвечает за рассылку когерентных запросов процессорным ядрам;
- собирает короткие когерентные ответы и когерентные ответы с данными для операций записи в память. Для операций по чтению из памяти когерентные ответы собирает сам запросчик (DMA или процессные ядра).

Структурная схема устройства НМУ представлена на рисунке 2.10.

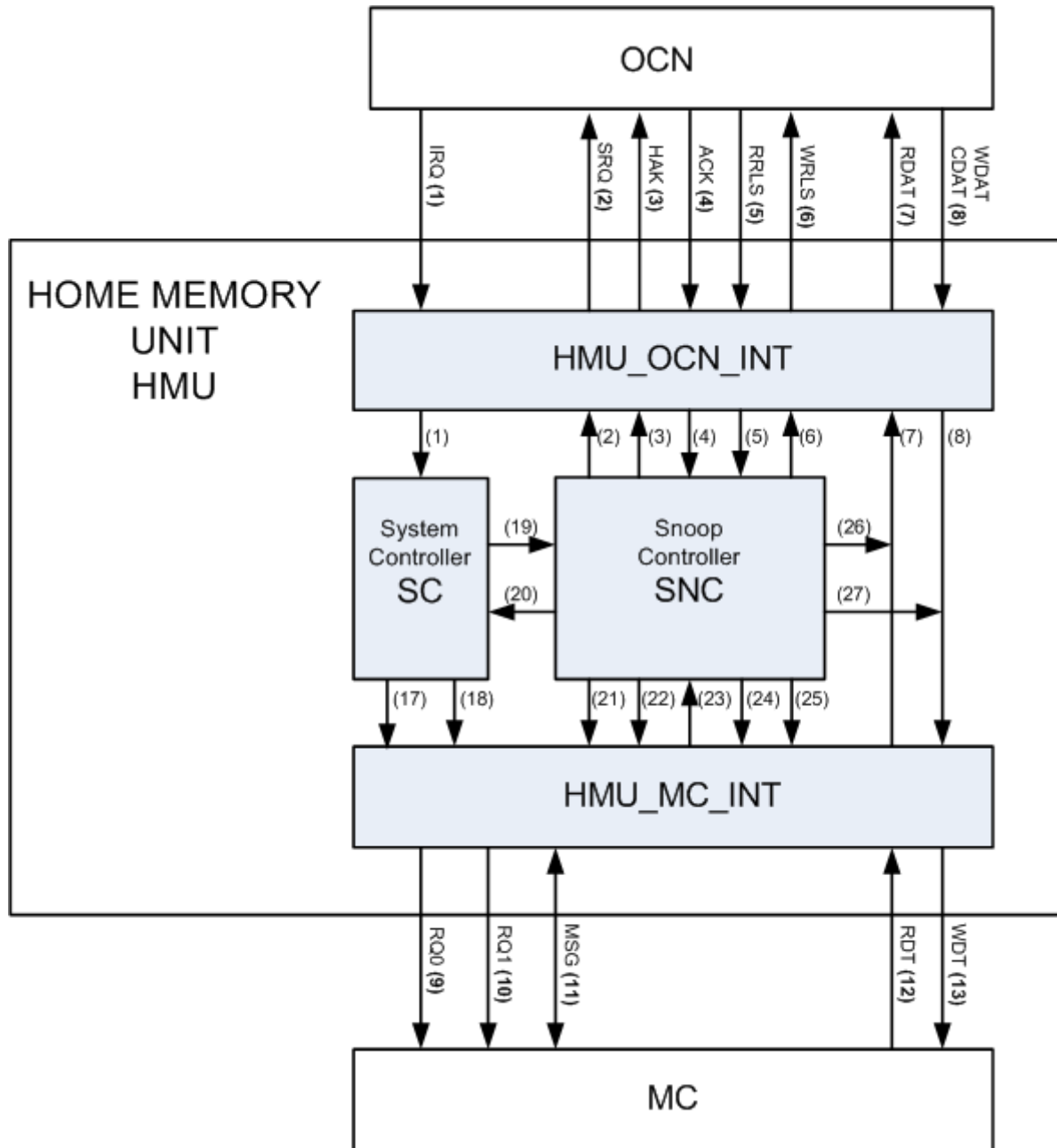


Рисунок 2.10 - Устройство НМУ

Основные внешние интерфейсы устройства НМУ:

- (1) – IRQ (Initial Request) – первичные запросы от процессорных ядер или DMA;
- (2) – SRQ (Snoop Request) – когерентные запросы от НМУ в процессорные ядра или DMA;

- (3) – НАК (Home Acknowledgment) – короткий ответ от НМУ запросчику (в процессорные ядра, DMA); При записях – запрос за данными DRQ/DRC (Data Request/ Data Request & Complete), при чтениях – сообщение с количеством ожидаемых ответов, которые должен собрать запросчик;

- (4) – АСК (Acknowledgment) - короткий когерентный ответ без данных от процессорных ядер, DMA;

- (5, 6) – RLS (Release) – сообщение о завершении чтения или записи. При записях WRLS – сообщение отправляет НМУ запросчику (в процессорные ядра, DMA). При чтениях (RRLS) сообщение отправляется в НМУ запросчиком;

- (7) – RDAT (Read Data) – считанные данные из оперативной памяти в процессорные ядра или DMA;

- (8) – WDAT (Write Data), CDAT (Coherent Data) – данные записи в оперативную память, а также когерентные ответы с данными для записи в память. Данные записи приходят в ответ на запрос за данными DRQ/DRC, когерентные ответы с данными – в ответ на когерентные запросы;

- (9, 10) – RQ0, RQ1 – 2 порта запросов в контроллер памяти МС (интерфейс функционирует на частоте контроллера памяти);

- (11) – MSG – интерфейс обмена сообщениями с контроллером памяти (отмена запросов, резервирование ячеек для данных записи). Интерфейс работает на частоте контроллера памяти;

- (12) – RDT – интерфейс приема данных чтения из контроллера памяти (интерфейс функционирует на частоте контроллера памяти, ширина интерфейса 32 байта);

- (13) – WDT – интерфейс выдачи данных в МС для записи в оперативную память памяти (интерфейс функционирует на частоте контроллера памяти, ширина интерфейса 32 байта);

Устройство НМУ состоит из следующих основных модулей (On-Chip Interconnect (OCI) обозначен как OCN):

- НМУ_OCN_INT (НМУ OCN Interface) – интерфейсный модуль между устройствами НМУ и OCI;

- HMU_MC_INT (HMU MC Interface) – интерфейсный модуль между HMU и контроллером памяти MC. В данном модуле происходит пересинхронизация интерфейса на частоту контроллера памяти. Модуль отвечает за выдачу номеров свободных ячеек для приема данных записи;

- SC (System Controller) – системный контроллер. Устройство, отвечающее за сериализацию запросов в память. Системный контроллер содержит конвейер, в котором определяется возможность выполнения запроса (проверяются адресные коллизии и наличие ресурсов для выполнения запроса). Если ничего не мешает выполнению запроса, запрос регистрируется в буфере запросов SC (Release Buffer, RLB) и передается на дальнейшее исполнение в Контроллер когерентных запросов SNC. В буфере RLB (64 ячейки) хранятся все запросы в оперативную память, выполняющиеся в данный момент в системе. Также Системный контроллер занимается отправкой команды чтения в контроллер памяти байпасом;

- SNC (Snoop Controller) – Контроллер когерентных запросов. Осуществляет контроль выполнения всех операций для запросов. Контроллер SNC рассылает все необходимые сообщения для выполнения запроса: когерентные запросы, сообщения HAK и RLS, запросы чтения и записи в оперативную память. Контроллер SNC собирает короткие когерентные ответы и отправляет сообщения о завершении операции в буфер запросов SC.

Для обращения в пространство памяти абонент (процессорное ядро, устройство ввода-вывода DMA) отправляет первичный запрос IRQ (1) в HMU в соответствии с адресом. Каждый запрос имеет уникальную метку Label, которая однозначно идентифицирует запросчика. Запрос попадает в системный контроллер SC (System Controller), буферизируется и отправляется в конвейер SC. При прохождении запросом конвейера, производится адресная проверка запроса. Если предыдущий запрос с таким же адресом не завершился, запрос будет заблокирован в конвейере системного контроллера. Также, при прохождении запросом конвейера проверяется наличие ресурсов для выполнения данного запроса (наличие свободных ячеек в контроллере памяти, в очередях когерентных запросов и.т.д). Если запрос прошел проверку на адресные коллизии и есть все необходи-

мые ресурсы для его выполнения, запрос записывается в Release Buffer (RLB) системного контроллера, где хранится до своего завершения.

Прошедший конвейер запрос (19) передается в контроллер когерентных запросов (Snoop Controller, SNC). Отправляется запрос в контроллер памяти (21, 22). Запрос чтения может быть отправлен байпасом в конвейере системного контроллера (17) и отменен (18). При отправке запроса записи в контроллер памяти резервируется ячейка для приема данных записи (и когерентных ответов с данными) в контроллере памяти (23). Запросчику отправляется сообщение НАК (3). Для чтений сообщение НАК содержит информацию о количестве ожидаемых когерентных ответов, которые должен собрать запросчик. Для записей сообщение НАК – это запрос за данными записи. В случае, когда нет необходимости дожидаться сбора когерентных ответов, сообщение НАК – это запрос за данными записи и сообщение о завершении операции записи. В ответ на данное сообщение запросчик выдает данные записи в НМУ.

Для запросов по чтению когерентные ответы собирает запросчик. Данные поступают запросчику либо из оперативной памяти (12, 7), либо от владельца кэш строки (кэшей процессорных ядер или DMA кэша процессора). Информация о запросчике не передается в запросах в память, а сохраняется в контроллере SNC и восстанавливается по номеру ячейки RLB (26). После сбора всех ответов, запросчик отправляет сообщение RRLS о завершении чтения в НМУ (5).

Для запросов по записи когерентные ответы собираются в НМУ. Короткие когерентные ответы собираются в контроллере когерентных запросов (4). Когерентные ответы с данными CDAT (8) передаются в НМУ_MC_INT. Номер ячейки в контроллере памяти и номер контроллера памяти восстанавливается из контроллера когерентных запросов SNC по номеру ячейки RLB (27). После сбора когерентных ответов в контроллер памяти отправляется сообщение об отмене/подтверждении запроса записи (25). Данные записи WDAT (8) передаются в НМУ_MC_INT и не требуют восстановления номера ячейки контроллера памяти. После сбора всех когерентных ответов, не дожидаясь данных записи (для коге-

рентных непочтовых записей необходимо дожидаться данных записи), НМУ отправляет запросчику сообщение о завершении операции записи (6).

После завершения обработки запроса SNC отправляет сообщение о завершении выполнения запроса в RLB Системного контроллера (20).

2.5 Контроллер памяти МС

2.5.1 Подсистема памяти микропроцессора Эльбрус-2С3 состоит из двух каналов, организованных на двух независимых контроллерах памяти DDR4 SDRAM. Разделение адресного пространства между контроллерами МС производится по заданному адресному разряду. Номер адресного разряда задаётся на этапе инициализации системы и хранится в регистре конфигурации чипсета.

Общие технические характеристики:

- соответствие стандарту «Jedec Standard. DDR4 SDRAM Specification» (JESD79- 4A);
- максимальный поддерживаемый объем памяти - 256 Гбайт;
- DDR4 SDRAM интерфейс ECC & TAGs: 72 бита = 64 Data-бит + 4 Tag-бита + 4 ECC-бита noECC & noTAGs: 64 бита = 64 Data-бит;
- поддержка модулей памяти Registered (RDIMM), Unbuffered (EUDIMM), Load Reduced (LRDIMM);
- поддержка 3D Stacked чипов памяти (3DS) до глубины 8Н;
- поддержка чипов памяти: 512 Мб x 4, 1 Gb x 4, 2 Gb x 4, 4 Gb x 4 256 Мб x 8, 512 Мб x 8, 1 Gb x 8, 2 Gb x 8 128 Мб x 16, 256 Мб x 16, 512 Мб x 16, 1 Gb x 16;
- поддержка длины пакета 32 и 64 байта (BL=4 и BL=8);
- поддержка до 64 одновременно открытых страниц (2 слота X 2 физических банка X 16 логических банков);
- поддержка режима 2Т (режим расширения фазы команды/адреса на интерфейсе DDR4 SDRAM до 2-х тактов);
- возможность распределения адресов (интерливинг) по физическим банкам;

- оптимизация выполнения последовательности запросов путём планирования очередности обращения в память (изменение порядка следования операций).

Функциональные характеристики:

- частота функционирования ядра (mc_clk) – не более 800 МГц;
- частота функционирования внешнего интерфейса DDR4 (ddr_clk) – не более 3200 Мбит/с;
- размер буфера запросов - 48;
- темп приёма запросов - 2 запроса/такт;
- задержка формирования команды на внешнем интерфейсе: 2 такта ядра + 2 такта DDR4 интерфейса;
- задержка по считыванию данных: 2 такта ядра + 2 такта DDR4 интерфейса.

Внешние интерфейсы контроллера памяти МС представлены на рисунке 2.11.

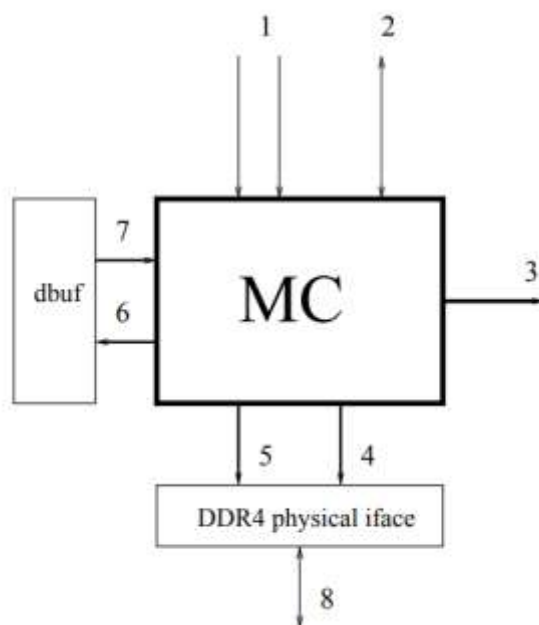


Рисунок 2.11 - Внешние интерфейсы контроллера памяти МС

Внешние интерфейсы контроллера МС обозначены на рисунке 2.11:

- 1 - два порта приёма запросов в память от системы;
- 2 - интерфейс обмена сообщениями с системным коммутатором. Приём сообщений производится по клоку МС (mc_clk);
- 3 - выдача параметров считанных данных в систему;
- 4 - интерфейс взаимодействия с физическим уровнем DDR4 - шина команд;
- 5 - интерфейс **взаимодействия** с физическим уровнем DDR4 - шина данных (чтение и запись);
- 6 - выдача номера регистра с данными, запись которых производится в память, в буфер данных;
- 7 - получение данных для записи.

Структурная схема контроллера памяти МС представлена на рисунке 2.12.

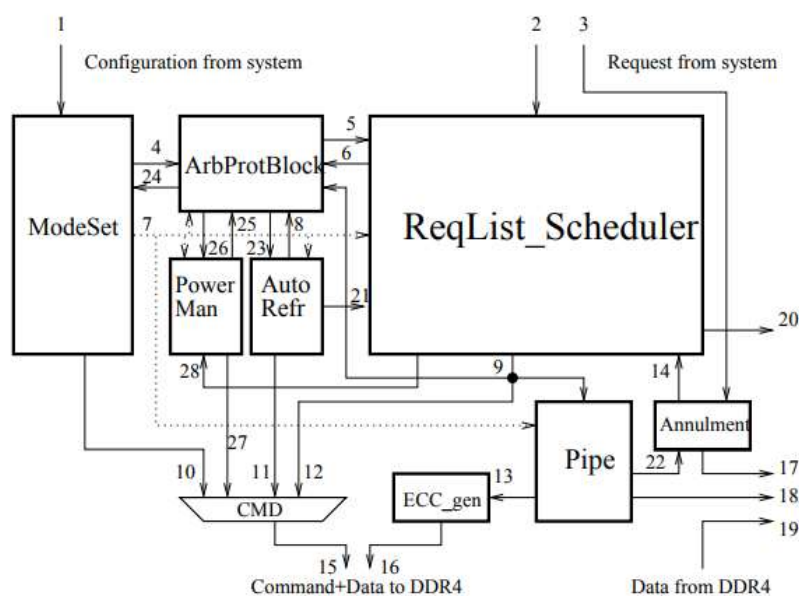


Рисунок 2.12 - Контроллер памяти МС

Контроллер памяти МС имеет в своем составе модули:

- RequestList_Scheduler - Реестр заявок, объединённый с планировщиком. Представляет собой буфер из **48** ячеек и набор фильтров. Каждая ячейка хранит в себе следующие параметры заявки:

1) VAL - Значимость. При помещении заявки в ячейку данный признак устанавливается в активное состояние, при завершении обработки заявки признак сбрасывается;

2) Составляющие адреса запроса в терминах интерфейса DDR4: ADDR_ar, ADDR_ac, ADDR_lb, ADDR_bg и ADDR_pb;

3) COP - Тип операции;

4) COPext - Расширение кода операции;

5) DRDY - Признак готовности данных (для операций записи);

6) STAGE - Шкала состояния процесса обработки заявки. Данная информация используется для принятия решения о дальнейших действиях по обработке заявки;

7) AGE - Возраст заявки. Данный параметр нужен для соблюдения корректной последовательности выполнения заявок с совпадающими адресами доступа (перемежение внутри пар rd→wr, wr→wr, wr→rd с одинаковыми адресами доступа запрещено);

8) DST - Локальный идентификатор запросчика;

9) ATTR - Системные параметры запроса (номер регистра процессора, идентификатор запросчика и т.д). Данный параметр возвращается в систему вместе со считанными из памяти данными без изменения, имеет параметризуемый размер;

10) DCELL - номер ячейки буфера данных чипсета, хранящего данные для записи;

11) SIZE - Размер запроса. Параметр используется для терминации burst-посылки (оптимизация) и определения достаточного места для приёма данных по чтению в коммутаторе данных;

12) SEALED - Признак "залипания" заявки чтения (заявка находится в реестре в течение продолжительного периода из-за того, что она не "вписывается" в стратегию оптимизации планировщика в данный период).

Приём запросов из системы производится по интерфейсу 2, организованному в виде двух независимых портов. При одновременном поступлении запросов

по двум портам более старым полагается запрос, пришедший с порта 1. Слежение за занятостью входного буфера контроллера возложена на систему. Все заявки, принятые от системы, передаются в блок планирования выполнения запросов.

- Scheduler - Планировщик. Осуществляет оптимизацию очередности выполнения заявок с целью увеличения эффективной пропускной способности канала памяти. При переупорядочивании учитываются возможные зависимости по адресам между заявками, попадания и промахи в открытые строки модулей памяти, тип последней выполненной операции, наличие "забытых" заявок (заявки чтения, которые находятся в очереди в режиме ожидания по причине нарушения ими стратегии оптимизации планировщика при попытке их выполнения в течение продолжительного времени) и т.п. В планировщике запрос может быть разбит на несколько субопераций, каждая из которых является зависимой или независимой с точки зрения перемежения с субоперациями других запросов. Например, операция записи с маской разбивается на интерфейсе DDR4 на две операции - чтение подложки и запись модифицированных данных;

- Pipe - конвейер управляемой длины для передачи сигналов управления буфером данных по записи, шиной DDR4 SDRAM и выдачей считанных данных в систему. Т.к. между выдачей команды записи / приёмом данных по чтению и выдачей / приёмом соответствующих этой команде данных на шину DDR4 SDRAM необходимо выдерживать промежуток, определяемый временными параметрами и настройками контроллера и модуля памяти (CAS Latency, Additive Latency, Parity Latency, mT - режим расширения фазы адреса, RM - Registered или Unbuffered модуль памяти), необходимо осуществлять задержку сигналов управления (выдачей / приёмом) данных и управления шиной на соответствующую этим параметрам величину. Сигналы управления длиной конвейера приходят из блока ModeSet (линия 7). Конвейер организован в виде последовательно соединённых линий задержки двух типов: с постоянной и переменной задержками. Фиксированная задержка определяется, исходя из внутреннего устройства контроллера. Переменная задержка определяется в блоке ModeSet в зависимости от установленных модулей памяти. При записи задержанный сигнал управления вы-

дацией данных передаётся по линии 13 в модуль генерации ECC ECC_gen. После выполнения операции Pipe формирует сообщение в чипсет о выполнении запроса освобождения ячейки буфера данных по записи. При адаптации контроллера памяти к физическому уровню DDR4 требуется подстройка глубины очереди параметров для операций чтения под задержку поступления считанных данных;

- ECC Gen - Генератор битов ECC. Для каждого (256 Data + 16 Tag = 272)-битового блока данных формируются 10 бит ECC. По линии 16 в записываемые данные с ECC передаются в физический уровень DDR4. Вспомогательные функции: генерация битов CRC при включённом режиме Write CRC(включение CRC при записи); инверсия байтов шины данных при включённом режиме Write DBI (инвертирование шины данных при записи);

- ECC Check - Блок обнаружения и коррекции ошибок ECC. Данный модуль вынесен за пределы контроллера памяти;

- AutoRefr - Блок планирования циклов регенерации памяти. По линии 7 в модуль передаются значение периода регенерации и признак расширения адресной фазы на шине DDR4 SDRAM до двух тактов. С соответствующей периодичностью по линии 11 из модуля поступает запрос в арбитр, и после получения разрешения и при отсутствии блокировок (линия 23) на шине DDR4 SDRAM выполняется цикл регенерации. Блок планирования поддерживает режим упреждающей регенерации;

- ModeSet - Блок установки режимов контроллера памяти, программирования режимов модулей памяти и начальной инициализации памяти. При изменении значений параметров (линия 1), относящихся к модулям памяти, из модуля в арбитр поступает запрос (линия 10), и после получения разрешения и при отсутствии блокировок (линия 24) на шину DDR4 SDRAM посылаются комплекс команд типа "Mode Register Set" (команды программирования регистров управления и режима работы модулей памяти). После окончания выполнения команд на линиях режима и параметров работы контроллера памяти (линии 7) устанавливаются значения, соответствующие новому режиму работы модулей. При изменении значений параметров, относящихся только к режиму работы кон-

троллера памяти (т.е. не требующих перепрограммирования регистров управления модулей памяти) и требующих остановки выдачи команд на шину DDR4 SDRAM, модуль также выдает запрос в арбитр (получение разрешения от арбитра означает, что в данный момент никакие другие команды на шину не выдаются) и при получении разрешения выполняет установку новых значений на линиях параметров и управления контроллера памяти;

- PowerMan - Блок управления режимами энергосбережения DDR4 SDRAM. По линии 7 в модуль передаются параметры для выбранного режима энергосбережения DDR4 SDRAM. По линии 25 из модуля в арбитр поступает запрос, и после получения разрешения и при отсутствии блокировок (линия 26) DDR4 SDRAM переводится в заданный режим энергосбережения (линия 27) по истечении таймера перевода DDR4 SDRAM в режим энергосбережения при отсутствии заявок в Scheduler (линия 28);

- ArbProtBlock - Блок арбитража и протокольных блокировок. Подсистема арбитража определяет приоритеты запросчиков на выдачу команды на интерфейс DDR4 SDRAM. До завершения выполнения начальной инициализации модулей памяти подсистемой ModeSet арбитр заблокирован, и никто из абонентов не может получить разрешения на доступ к шине DDR4 SDRAM. После завершения начальной инициализации арбитр начинает работу в нормальном режиме.

Приоритет обслуживания абонентов на доступ к шине DDR4:

- 1) AutoRefr;
- 2) ModeSet;
- 3) PowerMan;
- 4) ReqList_Scheduler.

Подсистема протокольных блокировок обеспечивает соблюдение временных пауз между последовательно выдаваемыми на шину DDR4 SDRAM командами согласно параметрам работы модулей и контроллера памяти и требованиям стандарта DDR4 SDRAM. Параметры для работы данной подсистемы передаются в ArbProtBlock из ModeSet по линии 7;

- ErrRec - Блок исправления ошибок чётности адресно-командной шины интерфейса DDR, ошибок CRC на шине данных интерфейса DDR при записи и ошибок ECC на шине данных интерфейса DDR при чтении. Ошибки исправляются путём повтора серии запросов, включающей тот, на котором произошла ошибка.

2.6 Устройство доступа к внешней памяти XMU

2.6.1 Структурная схема устройства XMU

2.6.1.1 Связь процессорных ядер с пространством ввода-вывода осуществляется посредством устройства XMU (eXternal Memory Unit).

Основными функциями XMU являются:

- выполнение запросов от ядер в пространство ввода-вывода;
- выполнение DMA-запросов от устройств ввода-вывода в память;
- пересылка сообщений между контроллером внешних прерываний (IOEPIC) и локальным контроллером прерываний (EPIC).

Структурная схема устройства XMU представлена на рисунке 2.13.

В состав устройства XMU входят следующие устройства:

- HC (Host Controller) – контроллер доступа в пространство ввода-вывода;
- CR (Configuration Registers) – блок конфигурационных регистров процессорного узла;
- EPIC – контроллер прерываний;
- XMU Crossbar – коммутатор связей.

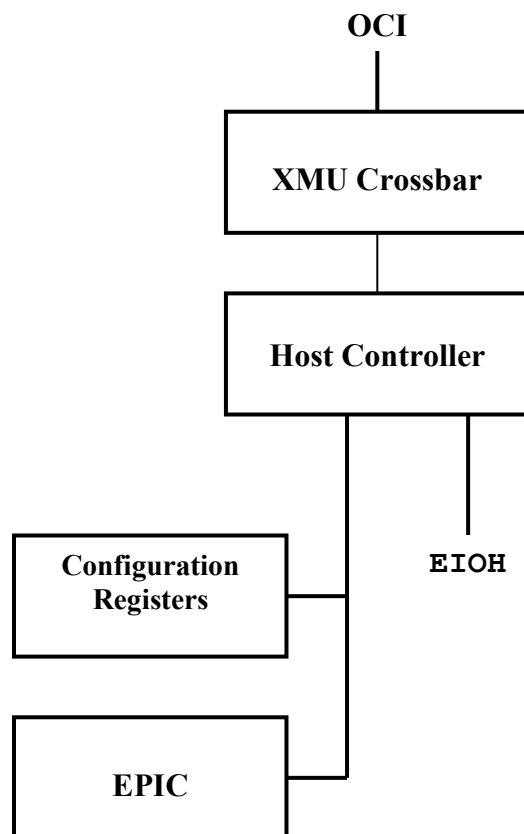


Рисунок 2.13 – Устройство HMU

2.6.2 Контроллер доступа в пространство ввода-вывода НС

2.6.2.1 Структурная схема.

Host-контроллер НС обеспечивает взаимодействие процессорных ядер с каналом ввода-вывода (контроллером EIOH), а также со встроенным программируемым контроллером прерываний EPIC и блоком конфигурационных регистров. Host-контроллер НС выполняет следующие функции:

- маршрутизация и обработка запросов от ядер в пространство ввода-вывода (IO-операции);
- осуществление функции DMA-канала для обеспечения доступа внешних устройств к оперативной памяти;

- маршрутизация сообщений о прерываниях между контроллером внешних прерываний IOERIC и EPIC.

Структурная схема Host-контроллера НС представлена на рисунке 2.14.

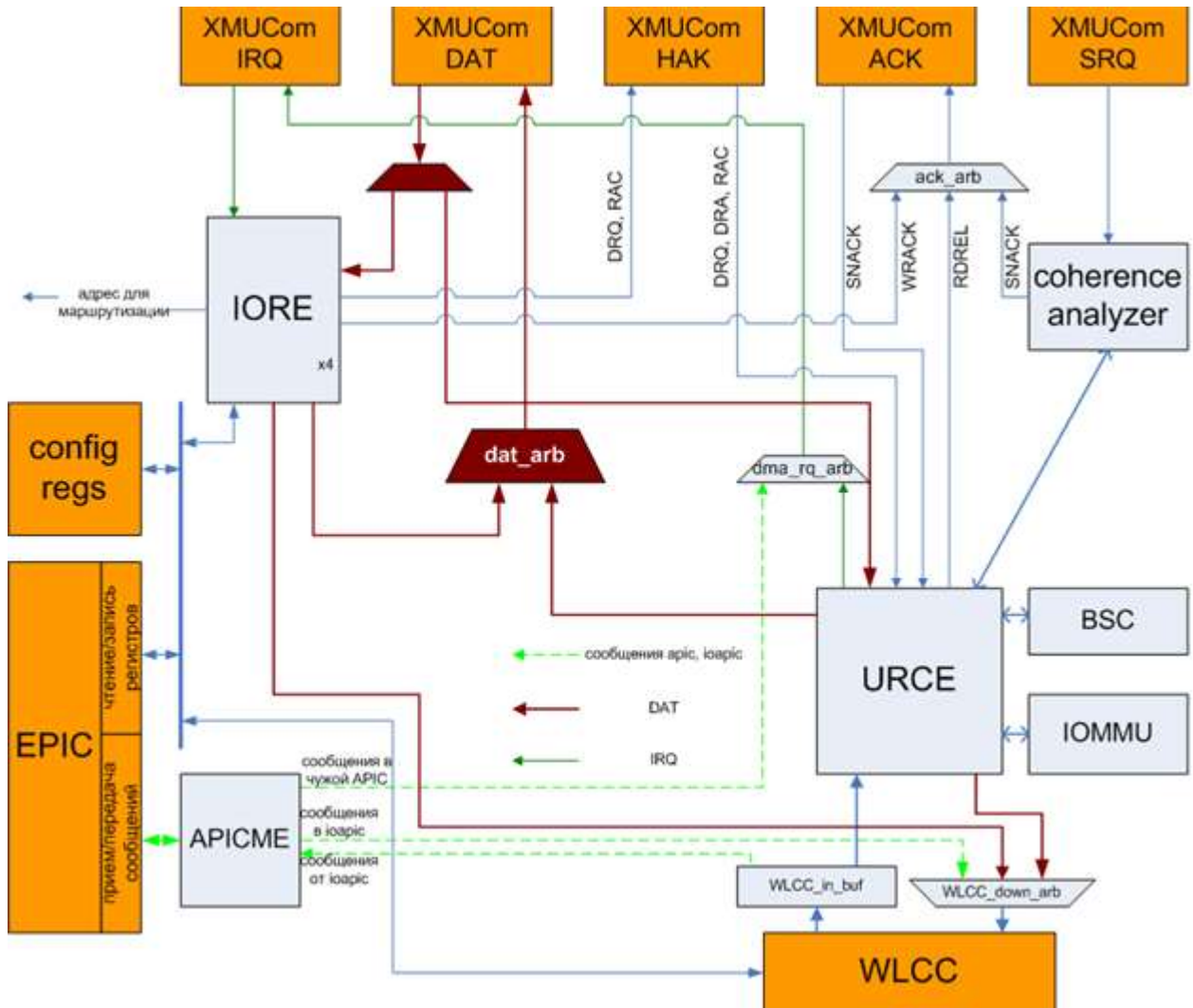


Рисунок 2.14 - Контроллер НС в составе устройства NMU

Host-контроллер НС взаимодействует со следующими устройствами:

- распределенный коммутатор XMU Xbar; интерфейсы: канал первичных запросов от процессора в пространство ввода-вывода и конфигурационные регистры, канал снуп-запросов, канал коротких сообщений о ходе выполнения на

запись, канал когерентных сообщений (ответов без данных), канал DMA-запросов, канал данных;

- контроллер канала ввода-вывода (WLCC); интерфейсы: двунаправленный канал запросов и ответов;

- программируемый контроллер прерываний EPIC; интерфейсы сообщений о прерывании и доступе к регистрам EPIC;

- блок конфигурационных регистров, блоки конфигурационных регистров EPIC, WLCC, контроллера RDMA (встроен в WLCC) взаимодействуют с НС посредством шины чтения и записи регистров.

Все операции, выполняемые НС, можно разделить на следующие классы:

- IO-запросы – запросы от процессора через НС в пространство памяти внешних устройств или конфигурационные регистры;

- DMA-запросы – запросы от внешних устройств в память;

- пересылка сообщений EPIC;

- PIO-запросы.

К IO-запросам относятся все запросы чтения и записи от процессорных ядер системы, выдаваемые в НС.

К PIO-запросам относятся:

- запросы в конфигурационные регистры, расположенные в IO Box, конфигурационные регистры устройств IO Box (EPIC, WLCC) и конфигурационные регистры НС;

- запросы в конфигурационное пространство КПИ и внешних устройств;

- запросы в пространство памяти внешних устройств;

- запросы в IO-пространство внешних устройств.

Запросы первого типа выполняются внутри НС, запросы второго, третьего и четвертого типа отправляются в IO-канал.

IO-запросы поступают из коммутатора XMU Xbar в НС в общую очередь. Сообщения EPIC относятся к другому типу запросов.

В состав устройства НС входят следующие устройства:

- IORE (IO Requests Executor) – устройство выполнения процессорных запросов (IO-запросов);
- URCE (Upward Requests and Completions Executor) – устройство обработки запросов от внешних устройств, ответов на процессорные запросы, а также снуп-запросов;
- APICME (APIC Messages Executor) – устройство обработки сообщений APIC, IOAPIC0 и IOAPIC1;
- coherence analyzer – устройство обработки snoop-запросов;
- IOMMU – устройство трансляции виртуальных адресов для запросов от внешних устройств;
- BSC (Bit Scale Controller) – контроллер битовой шкалы, реализует механизмы защиты памяти по записи;
- WLCC_in_buf – приемный буфер входных транзакций из WLCC.

2.6.2.2 Обработчик запросов и ответов из IO-линка (URCE).

Обработчик запросов и ответов из IO-линка (Upward Requests and Completions Executor, URCE) предназначен для приема всех запросов и ответов из одного виртуального канала IO-линка. Основной задачей URCE является контроль порядка выполнения операций.

Структурная схема устройства URCE представлена на рисунке 2.15.

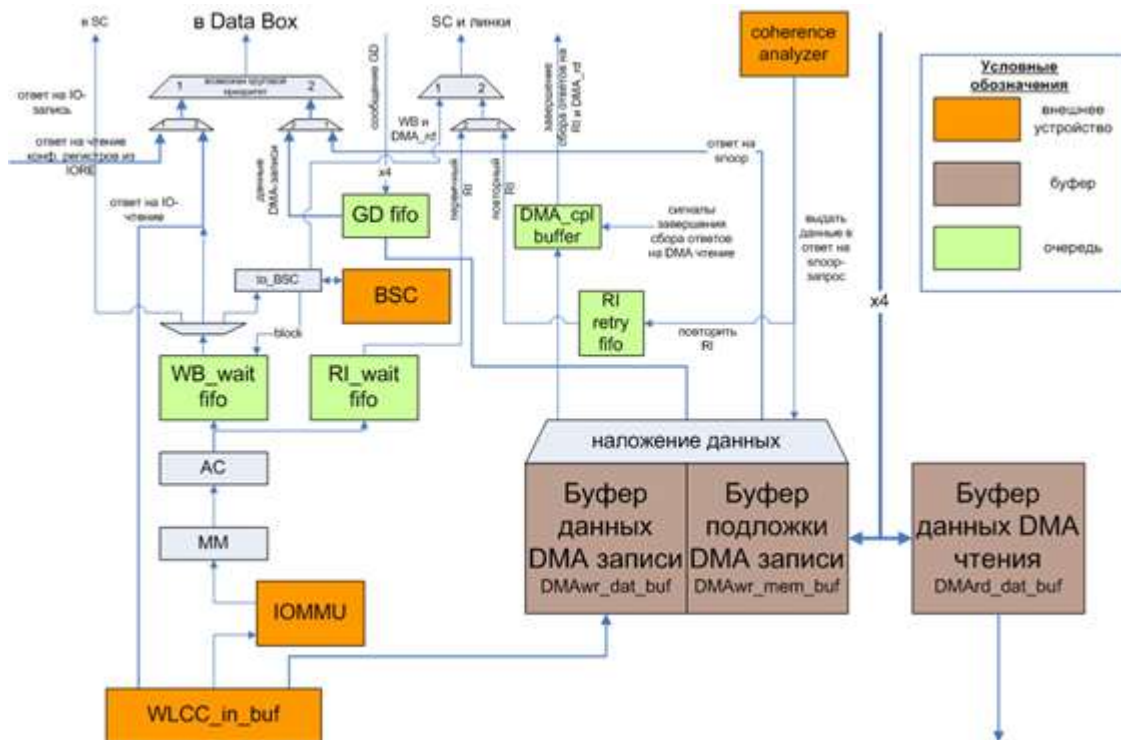


Рисунок 2.15 - Обработчик запросов и ответов из Ю-линка URCE

В состав URCE входят:

- буфер данных DMA-записи;
- буфер подложки DMA-записи;
- буфер данных DMA-чтения;
- блок вычисления адресных конфликтов (Address Conflicts, AC);
- блок защиты памяти маскированием (Memory Masking, MM);
- очередь первичных запросов RI(I) RI_wait fifo и очередь повторных запросов RI RI retry fifo;
- очередь отправки запросов/ответов WB_wait fifo;
- очередь сообщений о завершении RI(I) и DMA read (DMA_cpl buffer);
- очередь сообщений «дай данные» GD fifo.

2.6.2.3 Обработчик запросов в пространство ввода-вывода IORE.

Основной задачей обработчика запросов в пространство ввода-вывода (Input/Output Requests Executor, IORE) является прием и исполнение всех процессорных запросов в пространство ввода-вывода и обращений к системным регистрам, регистрам WLCC, регистрам IOMMU, регистрам НС и регистрам EPIC.

Структурная схема устройства IORE представлена на рисунке 2.16.

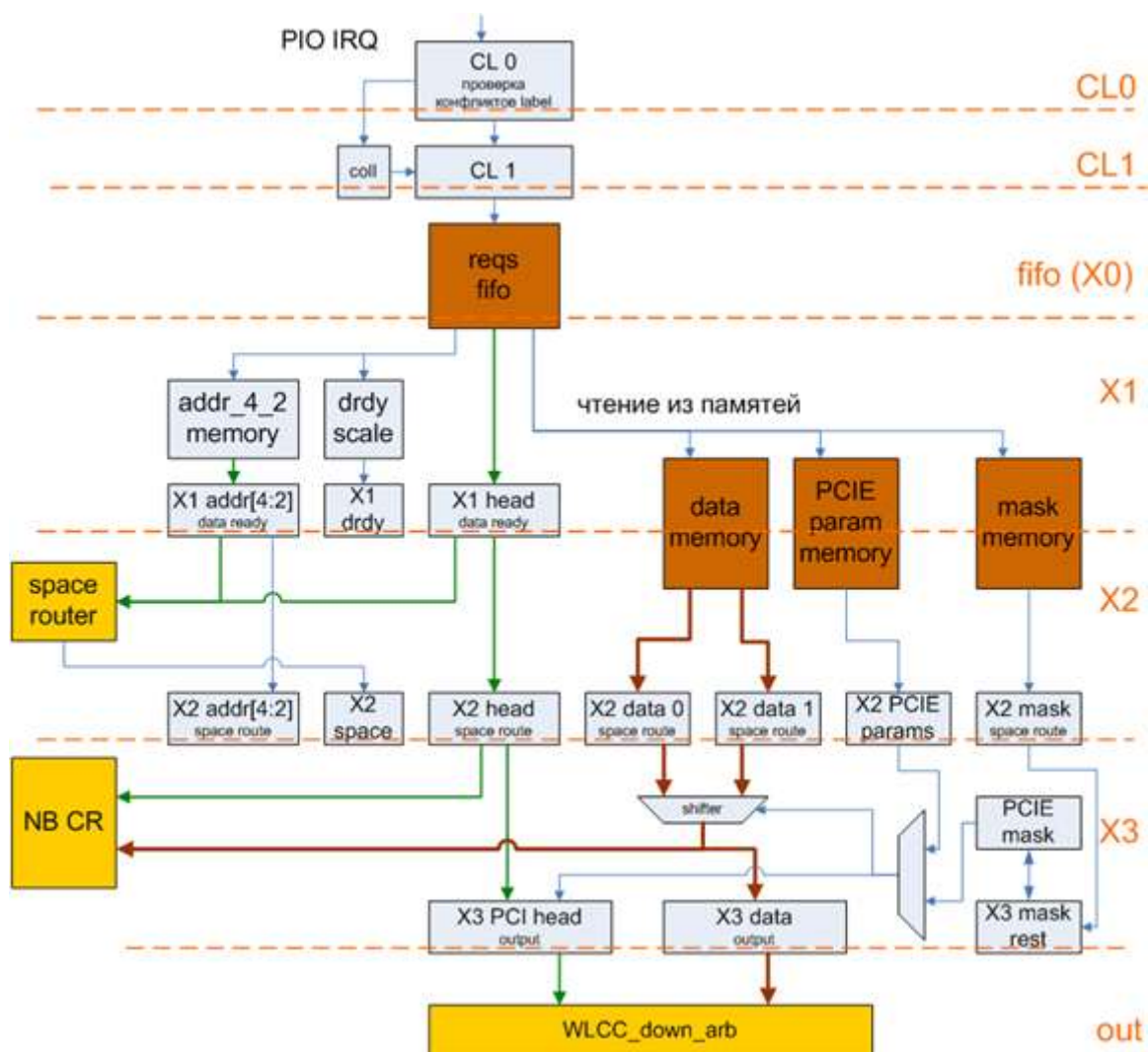


Рисунок 2.16 - Обработчик запросов в пространство ввода-вывода IORE

В состав устройства IORE входят:

- входная очередь запросов (reqs fifo);
- буферы данных, масок и параметров PIO-записей (data memory, PCIE param memory, mask memory);
- буфер параметров исполняемых запросов по записи Write_buf;
- буфер параметров исполняемых запросов по чтению Read_buf;
- очередь свободных идентификаторов запросов по чтению Read_nreg_fifo;
- очередь свободных идентификаторов запросов по записи Write_nreg_fifo;
- очередь коротких сообщений о ходе процессорных записей Write_cpl_fifo.

2.6.2.4 Устройство трансляции виртуальных адресов для запросов от внешних устройств IOMMU.

Устройство IOMMU выполняет трансляцию виртуального адреса (VA) DMA-запроса в физический (PA). Трансляция выполняется в соответствии с таблицей трансляции (Таблица страниц, Page Table, PT), которая расположена в оперативной памяти.

Виртуальные адреса транслируются в физические постранично, единственный поддерживаемый размер страницы – 4 Кбайт. Каждой виртуальной странице соответствует 4-байтный элемент таблицы трансляции (Page Table Element, PTE).

Структурная схема устройства IOMMU представлено на рисунке 2.17.

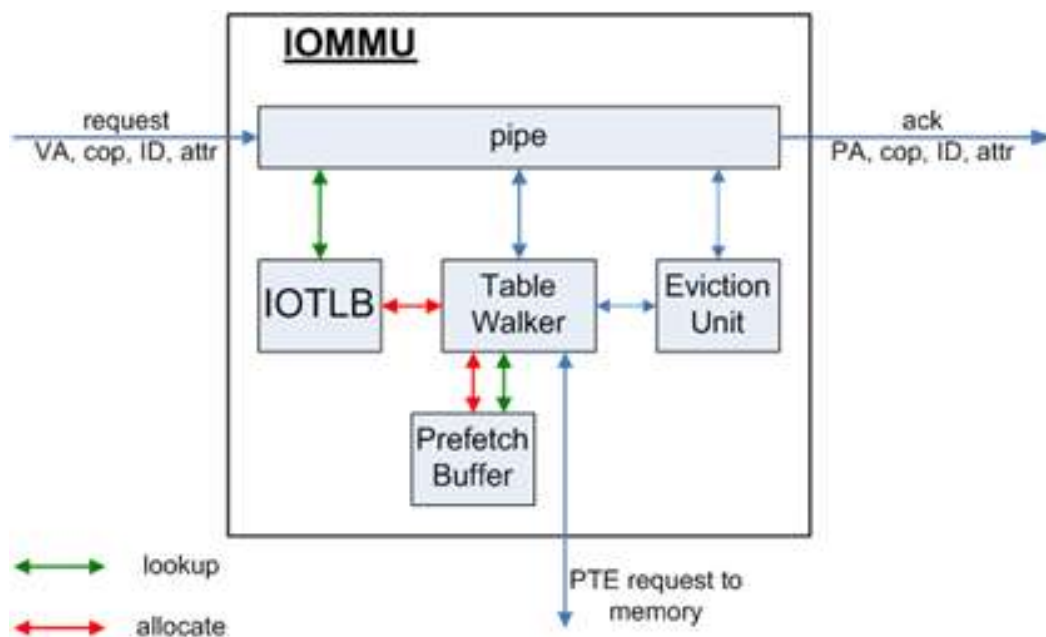


Рисунок. 2.17 - Устройство IOMMU

2.6.3 Контроллер прерываний EPIC

2.6.3.1 Контроллер прерываний EPIC (Elbrus Programmable Interrupt Controller) - программируемое устройство, предназначенное для фиксирования, хранения и доставки внешних прерываний к процессорным ядрам вычислительной системы.

Структурная схема устройства EPIC представлено на рисунке 2.18.

Контроллер EPIC состоит из следующих элементов:

- PREPIC (Processor Elbrus Programmable Interrupt Controller) - один на микропроцессор;
- CEPIC (Core Elbrus Programmable Interrupt Controller) - по одному на каждое процессорное ядро микропроцессора;
- IOEPIC (Input Output Elbrus Programmable Interrupt Controller) – расположен в составе контроллера EIOH.

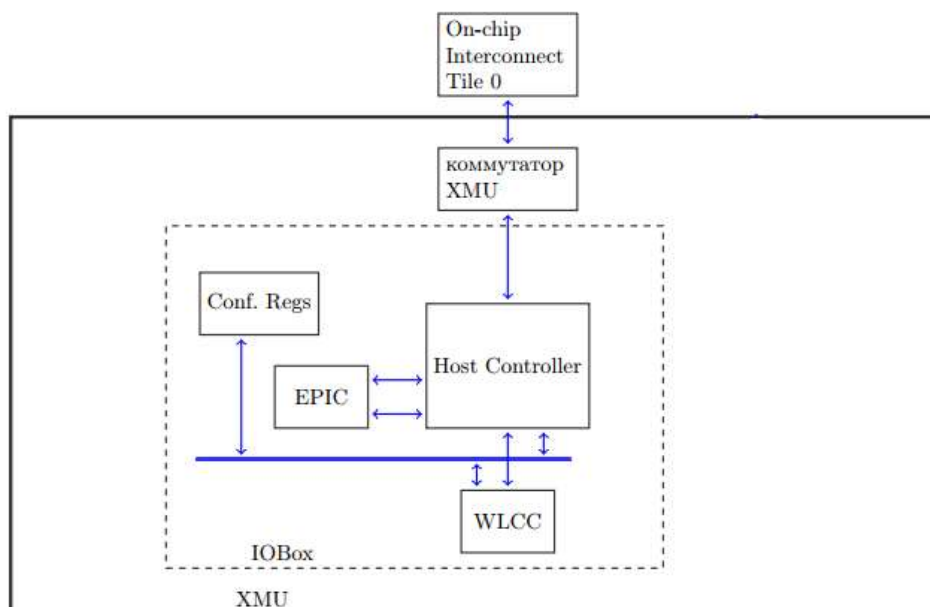


Рисунок. 2.18 - Контроллер прерываний EPIC

Управление контроллером осуществляется через набор программно доступных регистров EPIC в выделенной области пространства физических адресов. Элементы системы обработки прерываний (SEPIC, PREPIC, IOEPIC) обмениваются сообщениями. Сообщения формируются аппаратно. Контроллер реализует аппаратную поддержку виртуализации системы прерываний.

Для аппаратной поддержки виртуализации введены гиперпривилегированные регистры, доступные только гипервизору, и в каждом SEPIC реализован дополнительный набор регистров

Разделение между запросами хоста и гостя происходит по признаку host (1 - обращения хоста, 0 - обращение гостя) в запросе.

Хост может обращаться к гостевому набору регистров через выделенную область адресов.

Функции EPIC:

- принимает, обрабатывает и выдает ответы на запросы к регистрам системы прерываний;

- формирует, фиксирует, хранит и доставляет прерывания процессорным ядрам вычислительной системы;

- реализует аппаратную поддержку виртуализации системы прерываний.

Запросы от процессорных ядер к регистрам EPIC приходят строго по одному. Запросы буферизуются в хост-контроллере; перед отправкой в EPIC очередного запроса хост-контроллер ожидает ответа на предыдущий.

Запросы от процессорных ядер к некоторым регистрам (CEPIC_PMIRR, CEPIC_ICR, CEPIC_DAT) могут быть 4-байтными и 8-байтными (к остальным регистрам - только 4-байтные).

Доставка запроса конкретному CEPIC'у производится по физическому номеру процессорного ядра.

Прерывания в микропроцессоре могут формироваться:

- устройствами CEPIC, посредством регистров CEPIC_ICR (межядерные прерывания), CEPIC_EPIC_INT (служебные прерывания) и CEPIC_TIMER, CEPIC_NM_TIMER (прерывания по таймерам);

- устройством PREPIC, посредством регистров PREPIC_LINP (локальные системные прерывания) и PREPIC_EPIC_INT (служебные прерывания);

- внешними устройствами (внешние прерывания).

2.7 Встроенный контроллер периферийных интерфейсов EIOH

2.7.1 Структурная схема

2.7.1.1 Встроенный контроллер периферийных интерфейсов EIOH (Extended Input Output Hub) содержит набор контроллеров наиболее употребительных интерфейсов. Структурная схема контроллера EIOH представлена на рисунке 2.19.

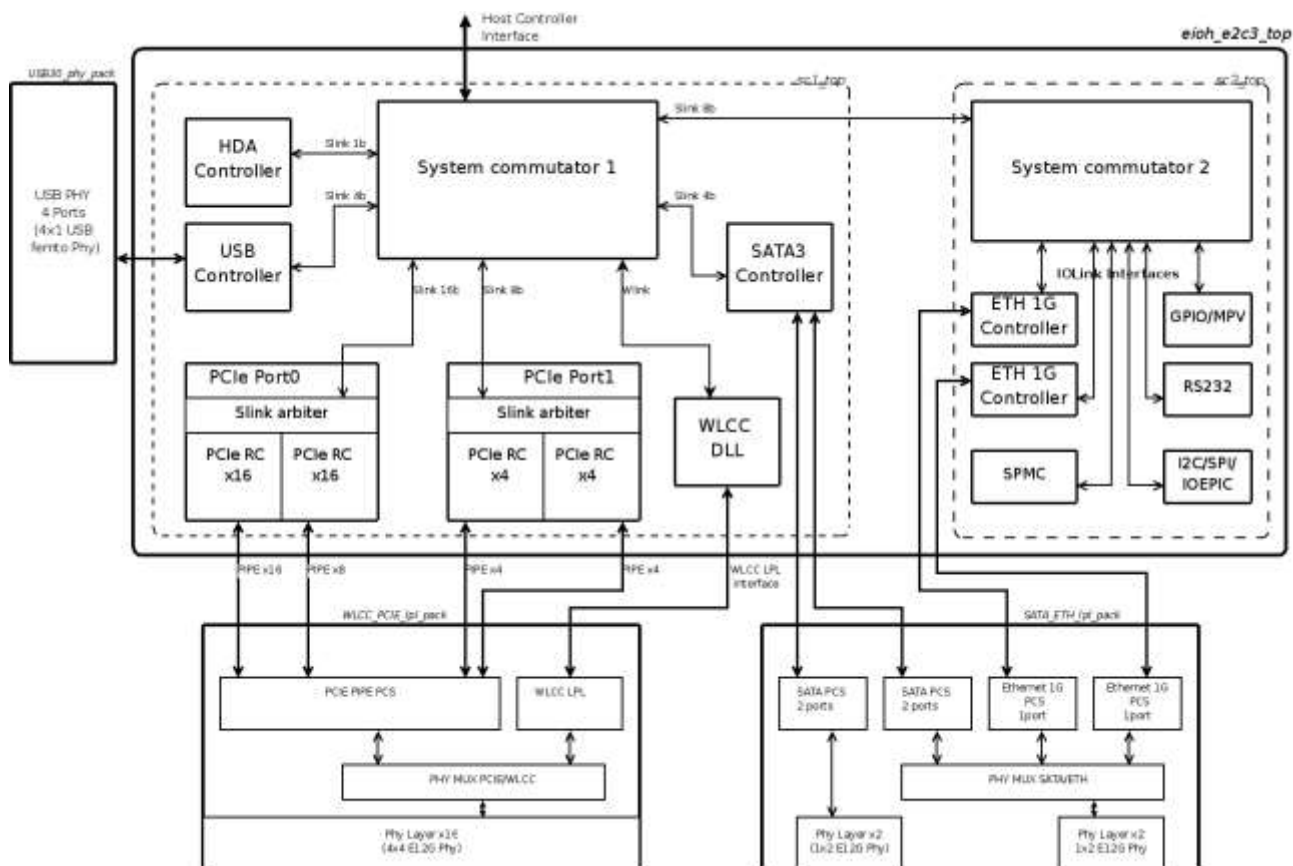


Рисунок 2.19 - Встроенный контроллер периферийных интерфейсов ЕЮН

2.7.2 Программная модель контроллера ЕЮН

2.7.2.1 С программной точки зрения контроллер периферийных интерфейсов ЕЮН представляет собой дерево PCI-устройств. Самым верхним в данной структуре является мост PCI-to-PCI (устройство номер N), подключенный к виртуальной шине номер 0. Номер N задается программно с помощью регистров <ИОУБ_DevNum.N> и <ИОУБ_DevNum.val> по следующему правилу:

- если <ИОУБ_DevNum.val>==1, то N берется из <ИОУБ_DevNum.N>;

- если $\langle \text{IOHUB_DevNum.val} \rangle == 0$, то мост PCI-to-PCI отвечает на все конфигурационные запросы с полем $\text{DevNum} = k$, где k может принимать любое значение от 0 до 31.

В таблице 2.10 и на рисунке 2.20 показано представление программной модели контроллера EIOH.

Таблица 2.10 - Табличное представление программной модели контроллера EIOH

Bus:Device:Function	Описание функции
Bus 0: Dev N:Func 0	Мост PCI-to-PCI
Bus m: Dev 0:Func 0	USB 3.0
Bus m: Dev 1:Func 0	Ethernet1G_0
Bus m: Dev 1:Func 1	Ethernet1G_1
Bus m: Dev 2:Func 0	GPIO/MPV
Bus m: Dev 2:Func 1	I2C/SPI
Bus m: Dev 2:Func 2	Serial Port
Bus m: Dev 2:Func 3	HDA
Bus m: Dev 3:Func 0	SATA 3.0
Bus m: Dev 5:Func 0	PCIe 3.0 x16(x8/x4)
Bus m: Dev 6:Func 0	PCIe 3.0 x8(x4)
Bus m: Dev 7:Func 0	PCIe 3.0 x4
Bus m: Dev 8:Func 0	PCIe 3.0 x4
Bus m: Dev 9:Func 0	SPMC
Bus m: Dev 10:Func 0	КПИ-2 (IOHUB2)(*)
<p>(*) - в случае наличия в системе микросхемы КПИ-2 её корневой мост будет находиться по адресу «Bus m: Dev 10:Func 0»</p>	

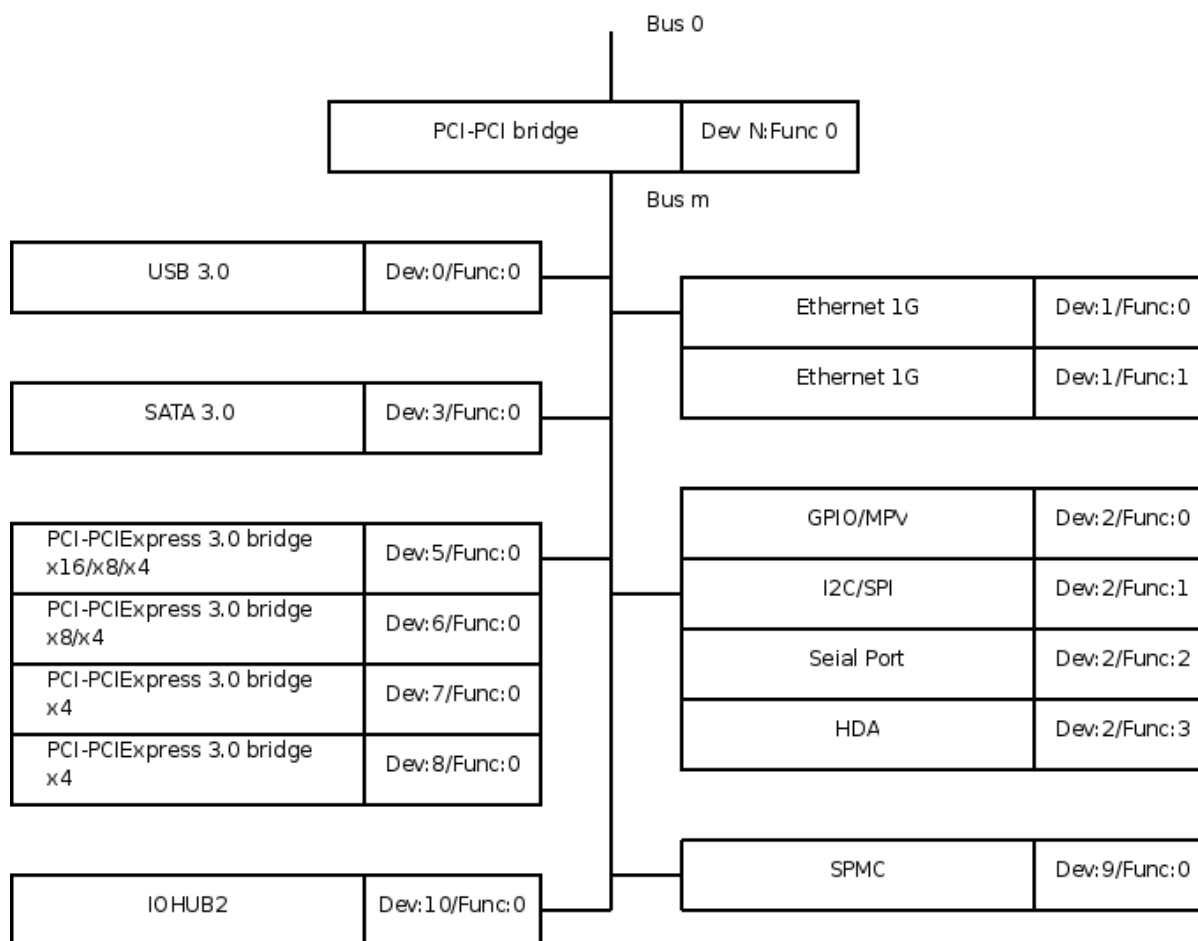


Рисунок 2.20 - Программная модель в контроллера EION

2.7.3 Системный коммутатор

2.7.3.1 Системный коммутатор в устройстве EION связывает между собой все контроллеры. Он состоит из двух частей System Commutator 1 и System Commutator 2.

2.7.3.2 Системный коммутатор первого уровня System Commutator 1 объединяет контроллеры с Slink интерфейсом, системный коммутатор второго уровня (System Commutator 2) и высокоскоростной канал ввода-вывода Wlink (WLCC_DLL/WLCC_LPL) для связи с микросхемой КПИ-2, с системным интерфейсом хост контроллера микропроцессора.

При организации множества каналов обмена между однотипными модулями на одном кристалле, как правило, используется единый для всех модулей интерфейс, обладающий фиксированной максимальной пропускной способностью. Пропускная способность интерфейса выбирается такой, чтобы покрыть потребности самого требовательного из модулей. При этом для каналов, требования которых на пропускную способность ниже, используется тот же интерфейс и то же количество аппаратных ресурсов. В случае больших межмодульных расстояний и широкого диапазона требований на пропускную способность используемых каналов это может приводить к неоправданному избыточному использованию аппаратных ресурсов и усложнению трассировки проводников на кристалле, особенно в случае, когда потребности в пропускной способности самого требовательного из каналов велики. Одним из способов решения указанной проблемы является использование масштабируемого унифицированного интерфейса SLink в соответствии с рисунком 2.21.

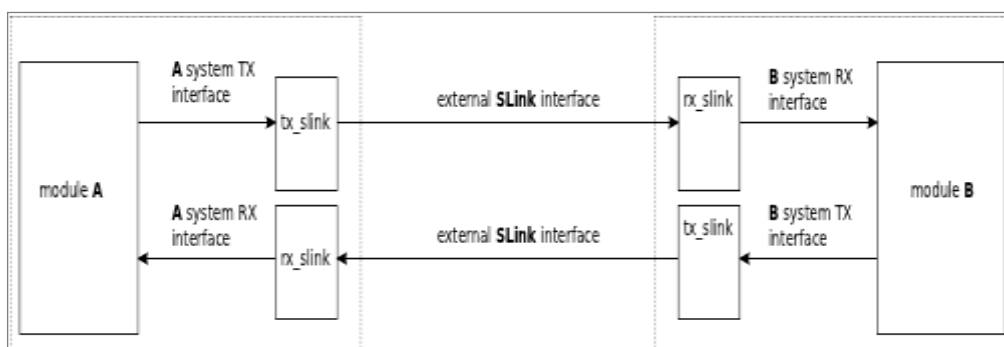


Рисунок 2.21 – Использование интерфейса SLink

Интерфейс SLink является коммутируемой (одни и те же линии используются для передачи как параметров (адрес, команда и т.п.) так и данных пакета) однонаправленной шиной. Ширина линии данных/параметров является задаваемым в соответствии с требуемой пропускной способностью параметром, а сам протокол обмена и набор управляющих сигналов является фиксированным. Параметр ширины линии данных/параметров задается в байтах и может быть задан с шагом в 1 байт в диапазоне от 1 байта и выше. Для организации двунаправленно-

го обмена используются отдельные интерфейсы SLink для каждого из направлений. Таким образом, использование интерфейса SLink позволяет организовать межмодульные каналы обмена с пропускной способностью, сбалансированной с параметрами потоков обмена между указанными модулями в обоих направлениях. Для подключения контроллеров SLink к модулям используются унифицированные системные интерфейсы.

2.7.3.3 Системный коммутатор второго уровня (System Commutator 2) объединяет контроллеры с системным интерфейсом IOLink, который использовался в микросхеме КПИ (1991ВГ1Я) в системном коммутаторе первого уровня (System Commutator 1).

Интерфейс IOLink представляет собой коммутируемую (одни и те же линии используются для передачи как параметров (адрес, команда и т.п.) так и данных пакета) однонаправленную шину фиксированной ширины 4 байта. IO-link интерфейс предназначен для передачи запросов и ответов на запросы. Запросы и ответы передаются в виде пакетов, имеющих различный формат в зависимости от их параметров. Передаваемый пакет состоит либо только из заголовка длительностью от двух до четырех тактов, либо из заголовка длительностью от двух до четырех тактов и блока данных длительностью от одного до 16 тактов. Для организации двунаправленного обмена используется два канала интерфейса.

2.7.4 Мультиконтроллер высокоскоростных линков WLINK/PCIe

2.7.4.1 Мультиконтроллер WLINK/PCIe реализует линк шириной 16 линий, в котором с помощью настроек реализуются интерфейсы — PCI-EXPRESS Gen 3 версии стандарта 3.1 и интерфейс канала ввода-вывода WLCC. 16 линий могут работать как несколько линков, реализуя интерфейсы PCI-EXPRESS и WLCC. Поддерживаемые режимы работы:

- PCI-EXPRESS Gen3 x16;
- PCI-EXPRESS Gen3 2x8;
- PCI-EXPRESS Gen3 x8 + WLCC x8;

- PCI-EXPRESS Gen3 2x4 + WLCCx8;
- PCI-EXPRESS Gen3 4x4;
- WLCC x16.

Структурная схема мультиконтроллера приведена на рисунке 2.22.

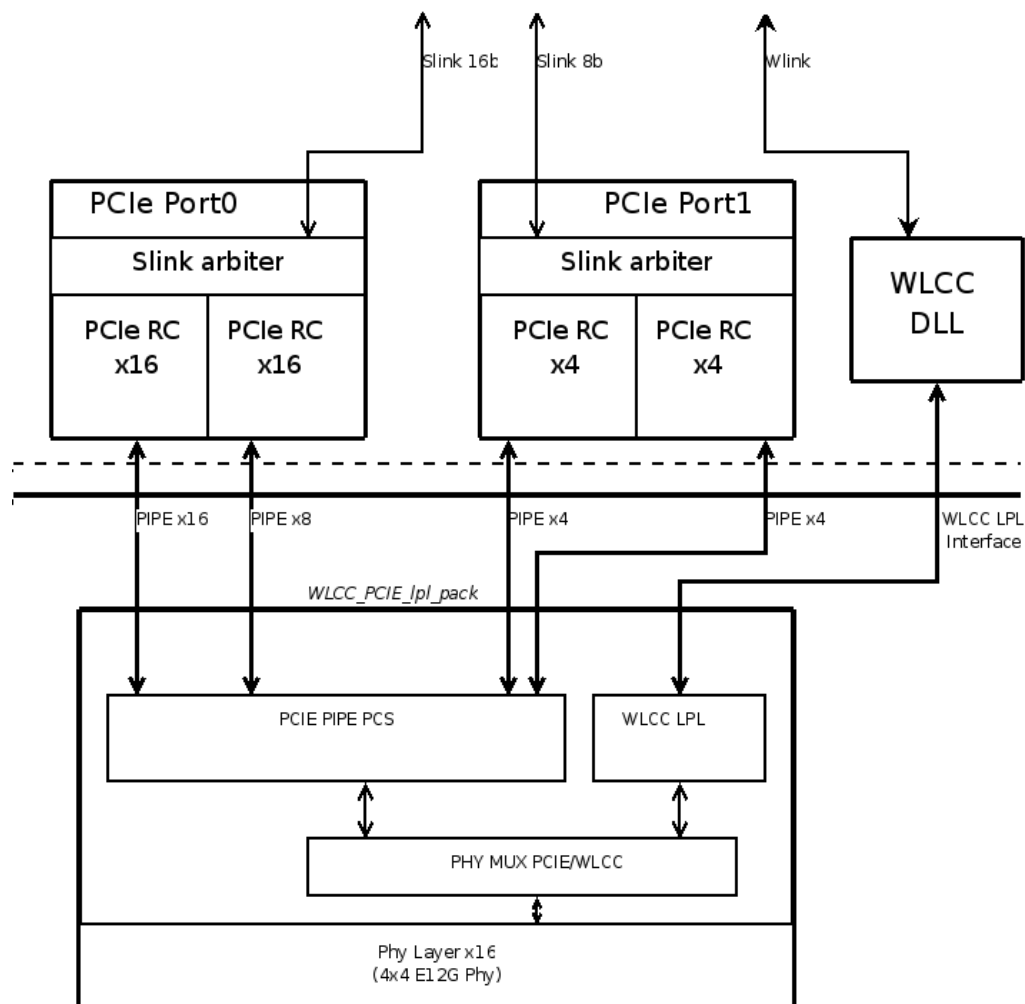


Рисунок 2.22 – Мультиконтроллер WLINK/PCIe

Upstream-интерфейсами контроллера являются Slink и WLCCsys. Slink — масштабируемый интерфейс, предназначенный для передачи данных на большие расстояния внутри кристалла. WLCCsys — интерфейс, предназначенный для реализации системного интерфейса чипсета. Наружу выходят 16 высокоскоростных линий, способных передавать данные на частоте до 12 Гбит/с. За реализацию физического уровня отвечает четыре блока «Enterprise 12G PHY x4». Для реализации стандарта PCI-EXPRESS Gen3 физический уровень настраивается на

передачу данных со скоростью 8 Гбит/с и электрическими параметрами сигнала, соответствующими PCI-EXPRESS 3.1. В случае реализации интерфейса WLCC физический уровень настраивается в зависимости от подключаемого к интерфейсу устройства. К примеру, в случае подключения микросхемы КПИ-2 физический уровень должен быть настроен на передачу данных до 5 Гбит/с. Для будущих чипсетов возможно увеличение скорости передачи данных до 12 Гбит/с.

Мультиконтроллер включает в себя четыре контроллера интерфейса PCI-EXPRESS (контроллер PCIe RC x16), каждый из которых виден в системе как независимый мост PCI-to-PCI. Каждый контроллер PCIe RC x16 состоит из двух блоков, первый — собственно сам контроллер PCIe RC Controller, второй — блок сопряжения с интерфейсом Slink, выходы которого подаются на блок Slink Arbiter. Блок Phy Layer x16 реализует управление физическим уровнем, а также расщепление физического уровня между контроллерами.

Краткие характеристики PCI-EXPRESS контроллеров:

- интерфейс с системой – Slink;
- поддержка трёх режимов работы: x16, 2x8 или 4x4 PCIe интерфейсов, в соответствии со спецификацией PCI Express 3.1;
- поддержка режимов x4, x8, x16 на интерфейсе PCIe;
- максимальная пропускная способность физического уровня PCIe Gen3 — 8 Гбит/с на линию (lane) в каждом направлении;
- четыре PIPE (максимум x16) интерфейса с элементами PHY;
- поддержка 32/64-битной адресации;
- поддержка механизма MSI (Message Signal Interrupt);
- максимальный размер передаваемых в пакете данных - 256 байт;
- поддержка до 64-х незавершенных запросов;
- поддержка атомарных операций;
- поддержка TPN/Steering Tag.

Также мультиконтроллер содержит в себе контроллер WLCC для подключения внешнего чипсета, который реализован контроллерами физического уровня WLCC_PHY и канального уровня WLCC_DLL.

Контроллер физического уровня WLCC_PHL (Wide Link Communication Controller Physical Layer) предназначен для преобразования потоков данных между контроллером канального уровня WLCC_DLL (Wide Link Communication Controller Data Link Layer) и физической средой передачи контроллера WLCC .

Модуль имеет в своем составе следующие интерфейсы:

- внешний линк - динамически масштабируемая (x4, x8, x16) последовательная шина (full duplex 5 Гбит/с + 5 Гбит/с per lane);

- интерфейс с контроллером WLCC_DLL, тракт передачи - 128 бит, 500 МГц, 8 Гбайт/с;

- интерфейс с контроллером WLCC_DLL, тракт приема - 128 бит, 500 МГц, 8 Гбайт/с.

Контроллер физического уровня WLCC_PHL состоит из двух блоков:

- электрический блок физического уровня - EPL (Electrical Physical Layer) - Synopsys PHY;

- логический блок физического уровня – MCST LPL (Logical Physical Layer).

Структурная схема контроллера физического уровня WLCC_PHL представлена на рисунке 2.23.

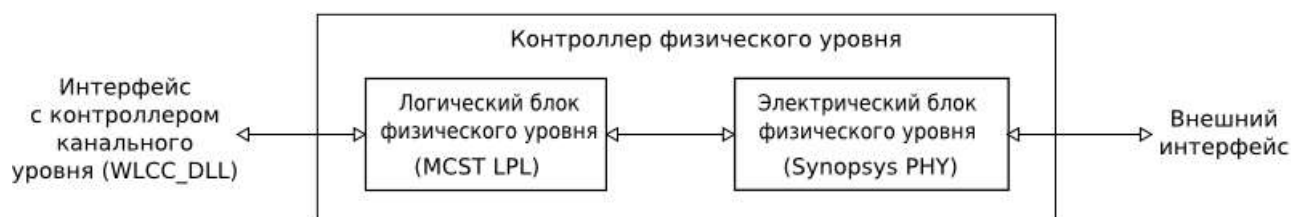


Рисунок 2.23 – Контроллер физического уровня WLCC_PHL

Электрический блок физического уровня EPL осуществляет преобразование входящего битового потока межпроцессорного линка в символьный поток (1 символ = 10 бит), направляемый в приемный канал модуля LPL, а также преобразование символьного потока из передающего канала модуля LPL в исходящий битовый поток межпроцессорного линка. Кроме того блок EPL формирует сигнала-

лы тактовой синхронизации для функционирования модуля LPL, а также осуществляет детектирование абонента на удаленном конце линка по команде от модуля LPL и передает результаты детектирования в LPL.

Логический блок физического уровня LPL контроллера WLCC_PHY взаимодействует с одной стороны с блоком EPL, с другой стороны с контроллером канального уровня WLCC_DLL.

Модуль LPL осуществляет:

- начальную инициализацию модуля EPL;

- исполнение команд от системы на:

- 1) инициализацию линка;

- 2) переинициализацию линка;

- 3) динамическое изменение ширины линка без остановки передачи данных;

- 4) восстановление линка при нарушении его функционирования;

- 5) перевод линка в одно из состояний: "Sleep", "Disable", "PowerOff";

- исполнение команд от удаленного абонента на:

- 1) инициализацию линка;

- 2) переинициализацию линка;

- 3) динамическое изменение ширины линка без остановки передачи

данных;

- 4) восстановление линка при нарушении его функционирования;

- 5) перевод линка в состояние "Sleep";

- передачу в контроллер WLCC_DLL информации о текущем состоянии линка;

- передачу в контроллер WLCC_DLL информации о результате выполнения команды от системы;

- прием потока данных из WLCC_DLL и его кодирование для выдачи на внешний линк посредством модуля PCIe2 PHY;

- прием потока данных с внешнего линка посредством модуля PCIe2 PHY и его декодирование для передачи в WLCC_DLL.

WLCC_DLL - контроллер канального уровня с некоторыми функциями уровня транзакций в составе высокоскоростного канала ввода/вывода. Предназначен для обеспечения связи северного моста с контроллером периферийных интерфейсов (КПИ-2) и выполняет следующие основные функции:

- обеспечение обмена с системным уровнем и с контроллером физического уровня в соответствии с протоколом;
- контроль целостности передаваемых данных;
- исправление ошибок, возникших при передаче;
- контроль состояния буферов.

Контроллер взаимодействует с системным уровнем посредством квантов информации, далее называемых транзакциями, обеспечивая сервис по надежному высокоскоростному обмену с системным уровнем внешнего абонента. Транзакции являются составляющими элементами системных операций, таких как чтение и запись, и передаются пакетами. Все операции включают отправку транзакции запроса (Request). Операции делятся на почтовые, то есть, не предполагающие транзакцию ответа (Completion), и на непочтовые, завершающиеся получением транзакции ответа (Completion).

Взаимодействие с контроллером физического уровня осуществляется посредством пакетов TLP. Передача пакета транзакции (TLP) на линке состоит из следующих фаз:

- Header - фаза заголовка;
- Payload - фаза данных транзакции (если есть);
- Tail - фаза завершения (или хвост) транзакции.

Для обеспечения контроля целостности данных при передаче по линку используется механизм нумерации TLP в последовательности и CRC-кодирование.

Протокол обмена по линку использует механизм группового подтверждения принятия TLPs. В случае сбоя, на линке инициируется процесс повтора последовательности TLPs, начиная с соответствующего номера. Повтор не может возникнуть по причине нехватки ресурсов приемника: это гарантируется использованием кредитного механизма управления потоком Flow Control Mechanism.

Для подтверждения приема, а также для передачи информации об освобождении ресурсов приемника, используются служебные пакеты DLLP.

Прием/передача пакетов транзакций по линку осуществляется в автоматическом режиме: то есть, для организации обмена не требуются обращения к внутренним регистрам. Доступ к внутренним регистрам контроллера используется только для задания режима работы, обработки прерываний и контроля состояния линка.

Контроллер поддерживает использование двух независимых виртуальных каналов. Под независимыми виртуальными каналами (VC) понимается способ организации однонаправленных потоков запросов и ответов между устройствами, находящимися в системных уровнях, исключающий взаимные конфликты по ресурсам системного уровня по приему (приемного буфера). Каждому виртуальному каналу на приемной стороне должен соответствовать выделенный для него ресурс буфера и кредитный механизм контроля его состояния (FC-механизм). Запросы, передающиеся внутри виртуального канала, сохраняют свой порядок, то есть, исключены обгоны одних запросов другими.

Ответы, передающиеся внутри виртуального канала, имеют право обгонять другие ответы, а также все непочтовые запросы, и почтовые запросы от других источников. Ответ не может обгонять почтовый запрос от того же источника. Ответы на непочтовые запросы для всех VC имеют бесконечный ресурс, что исключает конфликты по ресурсам приемной стороны и, следовательно, не учитываются FC-механизмами.

Структурная схема контроллера WLCC_DLL представлена на рисунке 2.24.

Контроллер WLCC_DLL содержит следующие блоки:

- TTQB (Transmit Transaction Queues Block) - Прием и распределение транзакций от системного уровня по VCs. Абитрация между VC. Обеспечение работы протокола управления FC-механизмами VCs по передаче (master part VC FC Management);

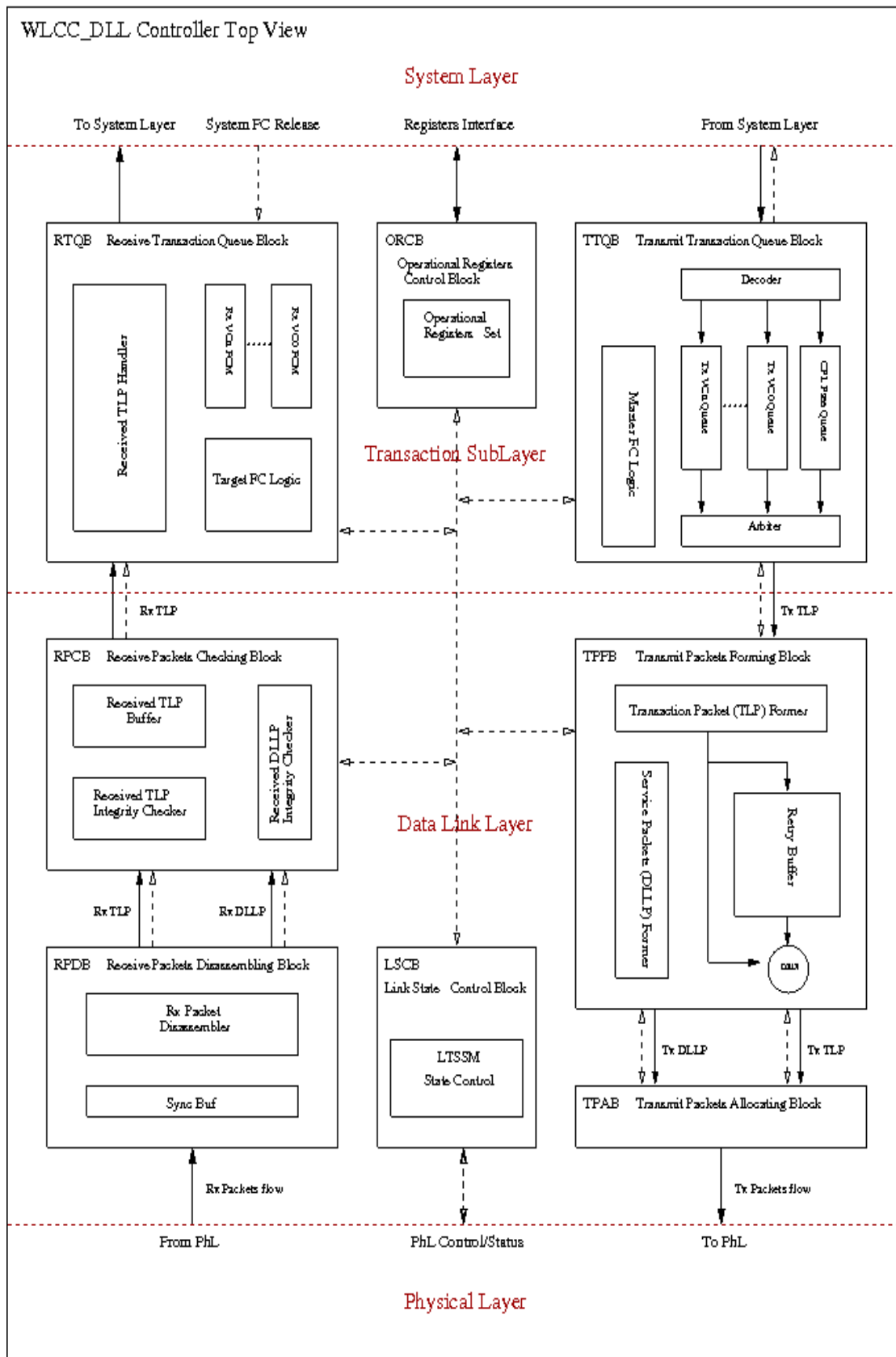


Рисунок 2.24 - Структурная схема контроллера канального уровня WLCC_DLL

- TPFB (Transmit Packet Forming Block) - Формирование пакетов транзакций (TLP) и служебных (DLLP) для передачи. Организация повторов последовательности TLPs;

- TPAB (Transmit Packet Allocating Block) - Подготовка потока передаваемых пакетов для оптимального использования пропускной способности линка. Выделение места под символы пакетной синхронизации;

- RPDB (Receive Packet Disassembling Block) - Прием и пересинхронизации потока принимаемых пакетов. Очистка от символов пакетной синхронизации. Разделение потока пакетов на TLP и DLLP потоки. Выделение в пакетах полей CRC;

- RPCB (Receive Packet Checking Block) - Разуплотнение потока TLPs, проверка целостности и генерация запросов на отправку подтверждений.

Разуплотнение потока DLLPs, проверка целостности, декодирование и передача служебной информации в TPFB, TTQB, RTQB;

- RTQB (Receive Transaction Queue Block) - Анализ принятых транзакций и передача их в системный уровень. Обеспечение работы протокола управления FC-механизмами VCs по приему (target part VC FC Management);

- LSCB (Link State Control Block) - Управление состоянием физического уровня;

- ORCB (Operational Registers Control Block) - Обеспечение возможности управления режимами и контроля параметров работы линка посредством доступа к операционным регистрам контроллера.

Ниже приведены основные характеристики контроллера WLCC_DLL:

- полнодуплексный канал обмена. Системная частота работы WLCC_DLL — от 250 до 500 МГц. Ширина интерфейса по данным между уровнями – 16 байт;

- режим автоматического определения деградации линка и запуск процедуры подстройки параметров линка;

- набор операционных регистров для управления режимами работы контроллера. Отдельный интерфейс для доступа к ним;

- поддержка двух независимых виртуальных каналов. Настраиваемая весовая схема арбитража виртуальных каналов при передаче;
- поддержка режимов 32-разрядной и 56-разрядной адресации при запросах в пространство памяти и операций скрытого канала;
- кредитный механизм контроля состояния буферов для каждого виртуального канала;
- учет кредитных ресурсов виртуального канала отдельно по заголовкам и по данным позволяет более рационально использовать буферы системного уровня;
- пакетная передача. Пакеты транзакций выровнены по границе 32 бит. Максимальный размер пакета - 80 байт (64 байта - данные), минимальный – 8 байт;
- низкие «накладные расходы» при передаче пакета. Существенно меньше, чем в пакетах PCIe;
- сервисные пакеты для передачи служебной информации размером 8 байт;
- механизмы контроля целостности пакетов транзакций - 16-разрядный CRC-код, нумерация пакетов в последовательности;
- информирование системы прерыванием в случае возникновения сбоев или нештатных ситуаций, требующих реакции системы;
- преодоление сбоев на линке путем повтора последовательности пакетов;
- буфер повторов размером ~2 Кбайт.

2.7.5 Контроллер USB 3.0

2.7.5.1 Контроллер универсальной последовательной шины USB содержит четыре порта USB 3.0.

Логическая часть хост-контроллера является универсальным интерфейсом XHCI (eXtensible Host Controller Interface), который поддерживает USB 1.x, USB 2.0, и USB 3.0 устройства. Структурная схема хост-контроллера USB представлена на рисунке 2.25.

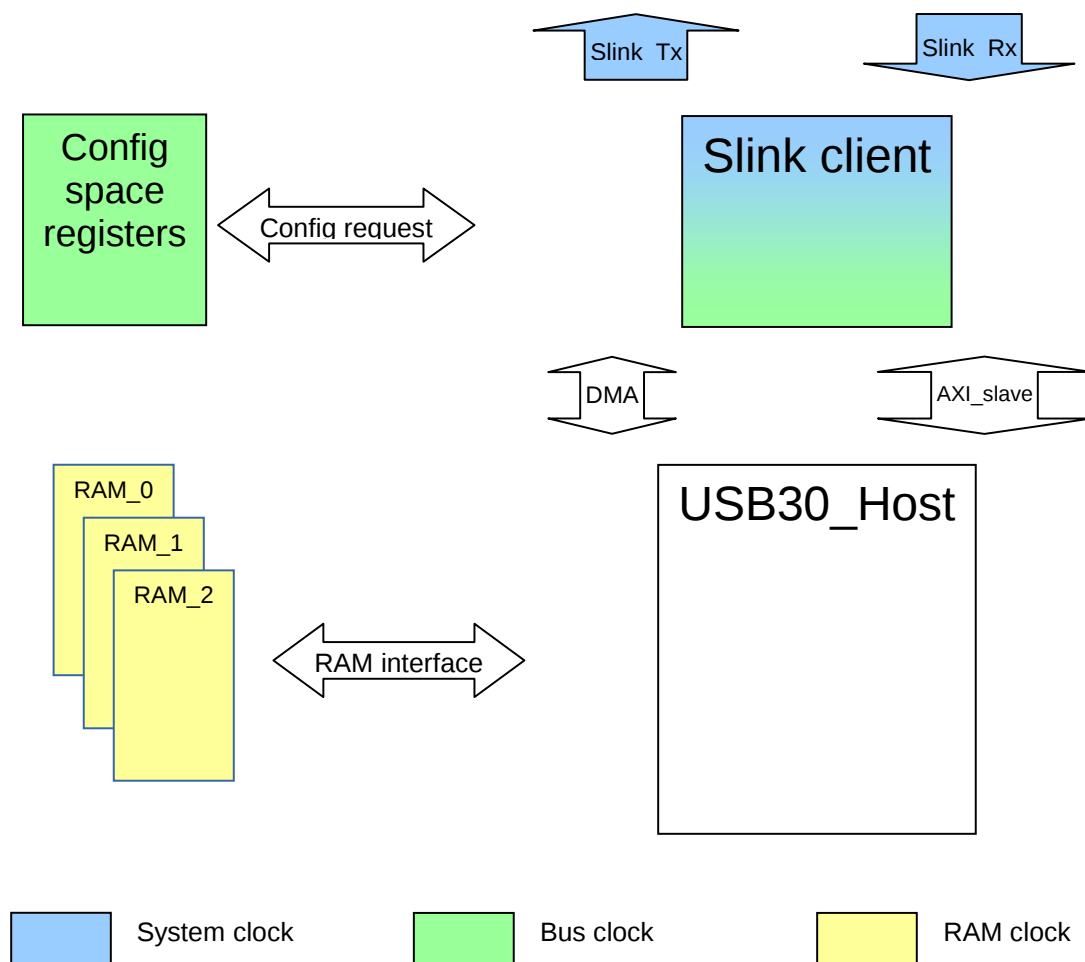


Рисунок 2.25 - Хост-контроллер USB

Характеристики контроллера:

- поддержка до 63 USB устройств;
- поддержка 32/64-битной адресации;
- скорость каждого USB 2.0 порта – 480 Мбит/с (полудуплекс);
- скорость каждого USB 3.0 порта – 5 Гбит/с (в обоих направлениях);
- интерфейс с системой – Slink.

Контроллер способен взаимодействовать с оперативной памятью в режиме прямого доступа к памяти (DMA).

Доступ к регистрам ХНСІ и внутренним регистрам хост-контроллера НС осуществляется через интерфейс AXI (Advanced eXtensible Interface). Регистры адресуются в пространстве памяти.

Для хранения пакетов необходимо подключить три памяти с произвольным доступом (RAM):

- RAM_0 – кэш дескрипторов;
- RAM_1 – пакеты на передачу;
- RAM_2 – принятые пакеты.

Имеется возможность включить в этих памяти режим с коррекцией ошибок (ECC), исправляющий изменения одного бита в одном машинном слове и вызывающий прерывание при ошибках в более чем одном бите.

Связь с системным коммутатором происходит через интерфейс Slink.

Для преобразования DMA запросов в транзакции интерфейса Slink реализован Slink клиент, который осуществляет разбиение на транзакции DMA запроса и пересинхронизацию на системную частоту. Также клиент возвращает ответы на DMA запросы. Внеочередные ответы на транзакции хранятся в отдельной памяти.

Реализовано конфигурационное пространство PCI. Клиент осуществляет доступ к регистрам конфигурационного пространства через принятые запросы configuration requests type 0. Если запрос в пространство памяти, принятый из системного коммутатора, попадает в регистр BAR0 конфигурационного пространства, происходит доступ к внутренним регистрам хост-контроллера НС.

Обмен с физическим уровнем происходит через интерфейсы UTMI+ для USB 2.0, PIPE для USB 3.0.

Для каждого USB порта используется отдельный блок физического уровня.

Сигналы UTMI+ портов 1- 3 пересинхронизируются на частоту utmi_clk[0]. Подача 0 на сигнал COMMONONN предотвращает выключение модуля фазовой автоподстройки частоты (PLL) UTMI в режиме низкого потребления энергии. Таким образом, синхросигнал на выходе FREECLK всегда присутствует.

Физический блок имеет повторитель входного опорного синхросигнала. Все блоки, кроме самого первого, получают на входе буферизированный синхросигнал от предыдущего блока.

2.7.6 Мультиконтроллер SATA/Ethernet

2.7.6.1 Структурная схема.

Структурная схема мультиконтроллера SATA/Ethernet представлена на рисунке 2.26.

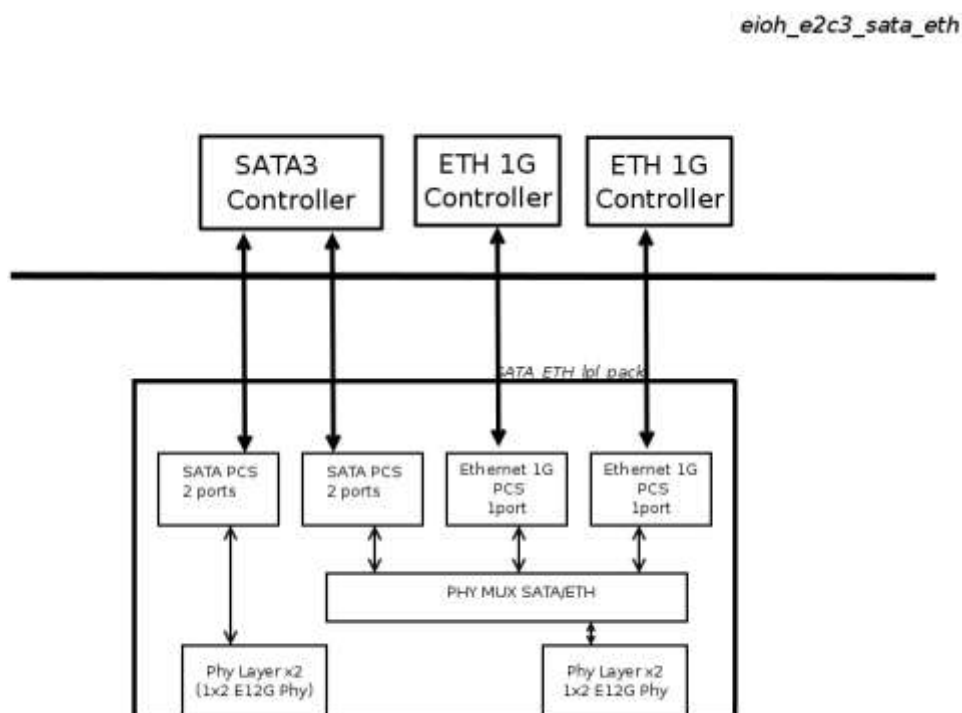


Рисунок 2.26 - Мультиконтроллер SATA/Ethernet

Мультиконтроллер содержит:

- контроллер интерфейса SATA 3.0 на четыре порта;
- два двухпортовых блока подключения интерфейса SATA к мультистандартному физическому уровню PHY 12G x2;
- два контроллера интерфейса 1 Gb Ethernet (1G(SGMII/1000Base-X) /

2.5G (SGMII);

- два блока подключения контроллеров интерфейса 1 Gb Ethernet к мульти-стандартному физическому уровню PHY 12G x2;
- мультиплексор интерфейсов SATA и Ethernet (позволяет использовать одни и те же физические линии для SATA или для Ethernet);
- два блока физического уровня PHY 12G x2.

Поддерживаемые конфигурации:

- 4 SATA;
- 2 SATA + 2 ETHERNET 1G.

Конфигурация определяется внешним сигналом во время сброса системы. Любой из контроллеров ETHERNET 1G способен работать в режимах 1G или 2.5G. Режим работы настраивается программно во время конфигурации оборудования.

2.7.6.2 Контроллер SATA 3.0.

Контроллер SATA3.0 состоит из логической части и блоков физического уровня.

Контроллер поддерживает четыре порта, каждый из которых способен работать в режимах SATA Gen3/2/1. Логический уровень контроллера реализует программную модель AHCI, а так же имеет возможность работать в режиме «legacy» (наследственный программный интерфейс контроллера IDE).

Структурная схема контроллера SATA представлена на рисунке 2.27.

В режиме AHCI контроллер поддерживает работу с 32 командными слотами для каждого порта, а так же поддерживает NCQ (zero buffer offset и non-zero buffer offset guaranteed-in-order). Системным интерфейсом контроллера SATA является межмодульный масштабируемый интерфейс Slink шириной 4 байта. Системный интерфейс реализуется с помощью Slink Client'a, через который осуществляется доступ к регистрам конфигурационного пространства, регистрам AHCI и legacy регистрам, а так же, с помощью внутреннего арбитра, клиент передает запросы чтения и записи системной памяти от DMA контроллеров портов.

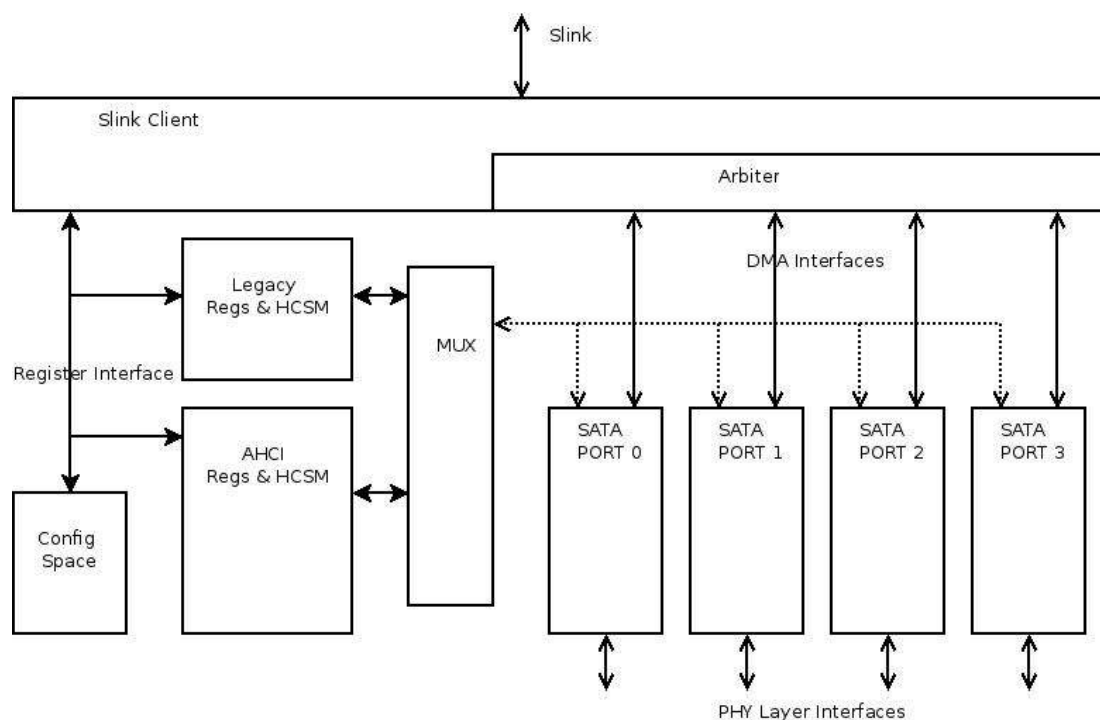


Рисунок 2.27 - Контроллер SATA

Интерфейсы для связи с физическим уровнем для контроллера SATA разделены на две группы по два порта. Обе группы для связи с физическими уровнями используют модуль настройки и управления физическим уровнем SATA PCS. Но при этом первая пара портов всегда подключена к физическому уровню. А вторая пара - через мультиплексор интерфейсов SATA и Ethernet, который может отключать эти порты контроллера SATA от физического уровня.

Краткие технические характеристики контроллера SATA 3.0:

- четыре порта;
- программируемые регистры базовых адресов;
- буфера размером 1024 байт для передачи данных, для каждого порта, в каждом направлении;
- встроенный DMA-контроллер, для каждого порта;
- системный интерфейс Slink;

- интерфейс для связи с SATA phy уровнем первого, второго и третьего поколения SATA;

- поддержка режима AHCI;

- поддержка NCQ (zero buffer offset и non-zero buffer offset guaranteed-in-order).

Контроллер соответствует спецификациям:

- Serial ATA specification, Revision 3.1;

- Programming Interface for Bus Master IDE Controller, Revision 1.0;

- Serial ATA Advanced Host Controller Interface (AHCI), Revision 1.3.

2.7.6.3 Контроллер Ethernet 1Gb ((1G(SGMII/1000Base-X) / 2.5G (SGMII)).

Микропроцессор Эльбрус-2С3 содержит два контроллера Ethernet 1G.

Структурная схема контроллера Ethernet 1Gb представлена на рисунке 2.28.

Контроллер имеет интерфейс IOLINK с системным коммутатором, и интерфейс GMII с блоком ETHERNET_1G_PCS, который используется для связи с блоком физического уровня E12G.

Краткие технические характеристики:

- поддерживаемые режимы:

- 1) 1G - поддерживает передачу данных на скоростях 10/100/1000 Мбит в полнодуплексном и полудуплексном режимах;

- 2) 2.5G - поддерживает передачу данных на скоростях 25/250/2500 Мбит в полнодуплексном и полудуплексном режимах;

- количество очередей: две на передачу и две на приём — в каждой очереди может быть до 512 дескрипторов, у каждой очереди есть независимый DMA канал (канал прямого доступа в память);

- системный интерфейс Iolink (частота до 500 МГц);

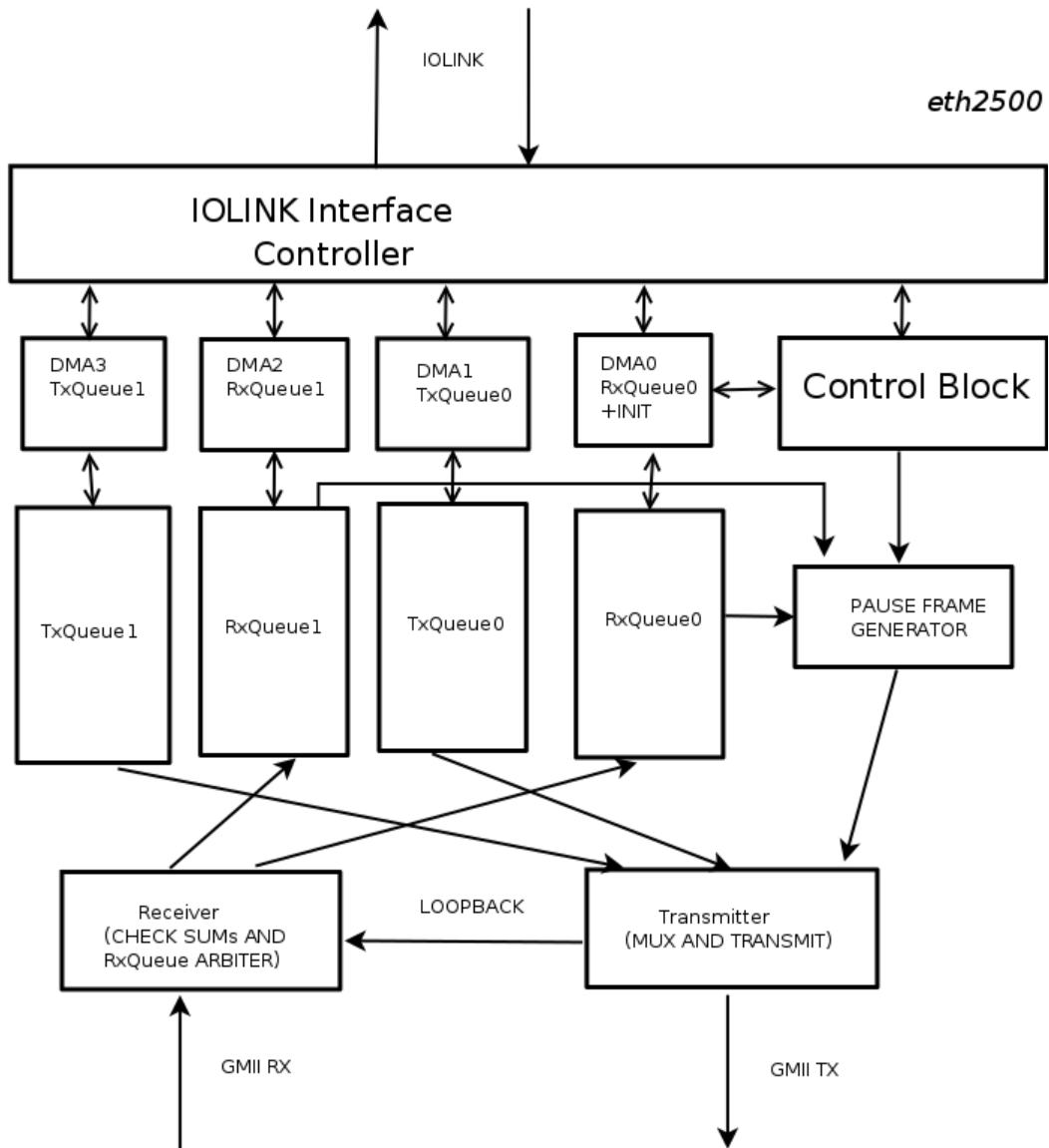


Рисунок 2.28 - Контроллер Ethernet 1Gb

- внешний интерфейс (контроллера) - GMII (частота 125 МГц для режима 1Gbit и 312,5 МГц для режима 2.5Gbit);
- поддержка управления потоком (pause frame);
- буфера для принимаемых и передаваемых пакетов в каждой очереди по

8 Кбайт;

- поддержка ieee-1588;
- автоматический подсчет контрольных сумм IPv4 пакетов (IPv4 header, TCP, UDP);

- автоматический подсчет CRC для Ethernet фреймов;
- интерфейс mdio для доступа к регистрам PCS блока и внешнего PHY;
- внешний интерфейс (микросхемы) - SGMII (1Gbit/2.5Gbit).

Режим работы 1G или 2.5G задается частотой синхронизации - 125 или 312,5 МГц. Контрольный бит, управляющий этой функциональностью, находится в блоке ETHERNET_1G_PCS. Регистры блока доступны через mdio интерфейс. Также с помощью регистров этого блока программируется блок физического уровня E12G для работы в режимах 1G или 2.5G.

2.7.7 Legacy контроллеры

2.7.7.1 Состав.

Микропроцессор Эльбрус-2С3 содержит набор legacy (служебных) контроллеров:

- I2C-SPI – включает в себя контроллеры интерфейсов I2C, IPMB, SPI, контроллер прерываний IOEPIC, истемный и Watchdog таймера, управление сбросом системы;

- HDA - аудиоконтроллер высокого разрешения;
- RS232 - контроллер последовательного порта (в дальнейшем, EIOH_SP);
- SPMC - контроллер управлением питания и режимами энергосбережения на уровне вычислительного комплекса;

- GPIO/MPV - контроллер программируемых входов-выходов со встроенным функциональным блоком приема синхросигналов от систем реального времени или от генераторов меток точного времени.

2.7.7.2 Контроллер I2C-SPI.

Контроллер I2C-SPI обеспечивает работу с интерфейсами I2C, IPMB, SPI; дополнительно включает в себя контроллер прерываний ввода-вывода IOEPIC с поддержкой виртуализации, системный и Watchdog таймеры, управление сбросом системы.

Структурная схема контроллера I2C-SPI представлена на рисунке 2.29.

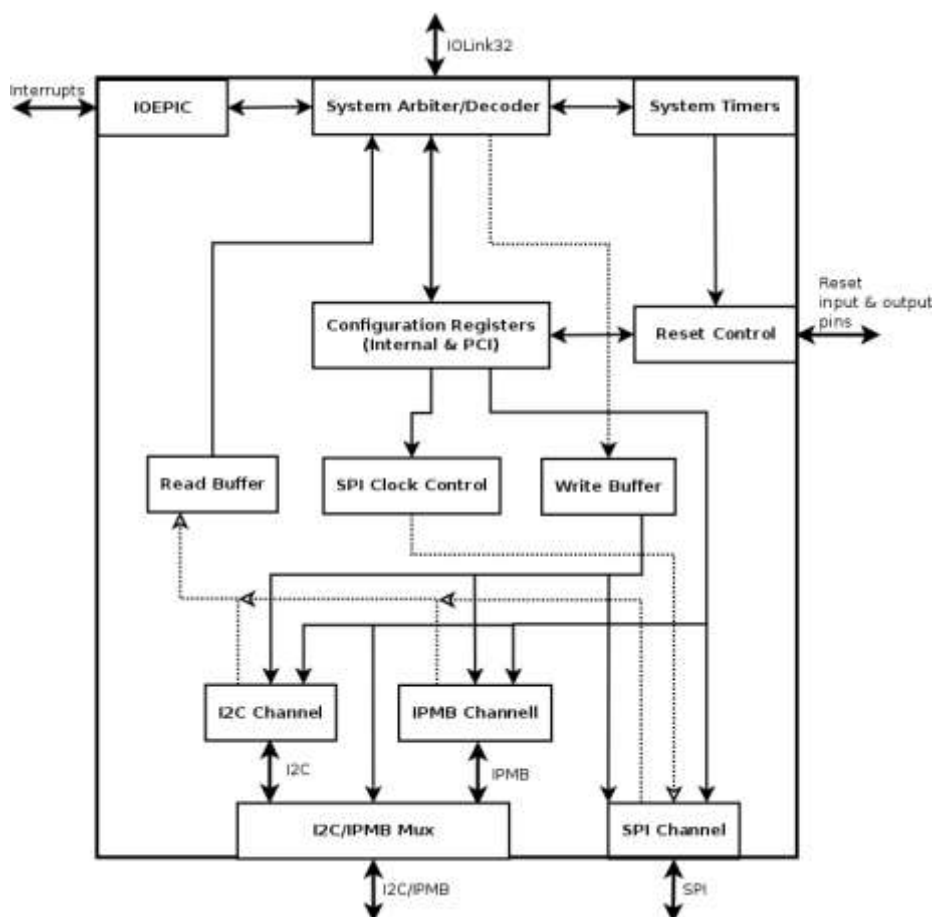


Рисунок 2.29 - Контроллер I2C-SPI

Контроллер состоит из следующих блоков:

- System Arbiter/Decoder - отвечает за запросы/ответы, приходящие с IOLink.

Формирует команды к IOLink по полученным прерываниям с контроллера;

- IOEPIC - контроллер прерываний ввода-вывода с поддержкой виртуализации, обеспечивает управление и отправку прерываний через Arbiter/Decoder;
- Reset Control - обеспечивает управление сбросом системы;
- System Timers. Системный и Watchdog таймеры обеспечивают работу по отправке прерываний через заданные промежутки времени и сброс системы;
- I2C, IPMB, SPI блоки отвечают за передачу сообщений по соответствующим интерфейсам;
- Read Buffer и Write Buffer. Буферы чтения и записи являются промежуточным звеном по передачи данных между Arbiter/Decoder и блоками I2C, IPMB, SPI.

Технические характеристики:

- системная частота (Iolink_clk) — до 500 МГц;
- опорная частота интерфейса SPI (spi_ref_clk) - 100 МГц;
- рабочая частота системного и Watchdog таймеров (tmr_clk) - 10 МГц.

SPI-Master:

- подключение до четырех устройств;
- поддержка SPI Mode 0,3;
- частоты SPI: 50, 25, 12,5, 6,25 МГц.

I2C/IPMB - подключение до пяти различных устройств I2C/IPMB.

I2C Master:

- поддержка I2C Standard Mode, Fast Mode, Fast Mode Plus;
- поддержка 7- битной и 10-битной адресации (I2C).

IPMB Master/Slave.

IOEPIC:

- поддержка до 64 прерываний;
- поддержка фронтовых и уровневых прерываний;
- отправка IOEPIC сообщений;
- программирование источника для поддержки виртуализации.

Таймеры и сброс:

- системный таймер;
- Watchdog таймер;

- управление сбросом системы.

Функционирование I2C-SPI:

Для доступа к конфигурационным регистрам используются конфигурационные обращения нулевого типа. Размер доступа - 1DW. При доступе к несуществующим конфигурационным регистрам, ошибка не возникает, запись не имеет эффекта. Доступ к операционным регистрам осуществляется с помощью следующих базовых адресов, представленных в таблице 2.11.

Таблица 2.11 - Базовые адреса контроллера I2C-SPI

Номер BAR	Назначение	Адрес
Base Address Register 0	I2C/SPI	10h
Base Address Register 1	Read/Write Buffers	14h
Base Address Register 2	System timers	18h
Base Address Register 3	IOEPIC	1Ch
Base Address Register 4	IPMB	20h

Для доступа к различным блокам I2C-SPI используется устройство, обеспечивающее управление по обмену данными через интерфейс IOLink - System Arbiter/Decoder. После получения команды через IOlink и ее дешифрации происходит чтение/запись данных во внутренние регистры блоков (контроллеров).

Для доступа к регистрам управления I2C, IPMB, SPI портов используется обращение в пространство памяти с 32-битным адресом. Регистры размещены в пространстве памяти с помощью BAR0 - I2C и SPI и BAR4 - IPMB. При обращении к несуществующим регистрам управления выдается ошибка. Размер доступа - 1 DW. В зависимости от настроек, может быть подключено пять устройств I2C или IPMB.

Для доступа к буферам записи и чтения используются обращения в пространство памяти. Размер доступа - 1, 2, 4, 8, 16 DW. Для обращения к буферу записи используются операции записи, буферу чтения - операции чтения.

Для обращения к System и WatchDog таймерам используются обращения в пространство памяти с 32-битный адресом. Регистры размещены в пространстве памяти с использованием BAR2. System timer может использоваться для жесткого, мягкого и системного сброса. WatchDog timer может работать в двух режимах. В первом режиме по достижению предела WatchDog таймера, генерируется сигнал сброса. Во втором случае, отправляется прерывание.

Для обработки внутренних и внешних прерываний используется контроллер IOEPIC. Регистры размещены в пространстве памяти с помощью BAR3. После получения прерываний, по алгоритму RoundRobin выбирается незамаскированное прерывание, проверяется Arbiter на выполнение задачи и, если Arbiter свободен, формируется команда, которая отправляется через IOLink. Сообщение состоит из четырех или DW, в зависимости от размера адреса. В IOEPIC используется таблица прерываний, в которой могут храниться данные по 64 внутренним и внешним прерываниям. Прерывания могут быть как фронтовыми, так и уровневыми. Для блокировки прерывания используется маска.

Список прерываний IOEPIC представлен в таблице 2.12.

Таблица 2.12 - Список прерываний IOEPIC контроллера

Номер входа IOEPIC	Источник прерывания
0	IPMB (внутри контроллера I2C-SPI)
1	SCI (power management, из SPMC контроллера)
2	System Timer (внутри контроллера I2C-SPI)
3	Ethernet0_tx0 – прерывание от передатчика нулевой очереди нулевого Ethernet контроллера
4	Ethernet0_tx1 – прерывание от передатчика первой очереди нулевого Ethernet контроллера
5	Ethernet0_rx0 – прерывание от приёмника нулевой очереди нулевого Ethernet контроллера
6	Ethernet0_rx1 – прерывание от приёмника первой очереди нулевого Ethernet

Номер входа ЮЕРИС	Источник прерывания
	контроллера
7	Ethernet0_sys – системное прерывание от нулевого Ethernet контроллера
8	HDA (eioh)
9	Mpv_timers0
10	Mpv_timers1
11	Mpv_timers2
12	GPIO0
13	GPIO1
14	Serial Port
15	I2C/SPI (внутри контроллера I2C-SPI)
16	PCI IRQ A – legacy PCIE
17	PCI IRQ B – legacy PCIE
18	PCI IRQ C – legacy PCIE
19	PCI IRQ D – legacy PCIE
20	WD Timer (внутри контроллера I2C-SPI)
21	SATA
22	SERR (от всех устройств)
23	Ethernet1_tx0 – прерывание от передатчика нулевой очереди первого Ethernet контроллера
24	Ethernet1_tx1 – прерывание от передатчика первой очереди первого Ethernet контроллера
25	Ethernet1_rx0 – прерывание от приёмника нулевой очереди первого Ethernet контроллера
26	Ethernet1_rx1 – прерывание от приёмника первой очереди первого Ethernet контроллера
27	Ethernet1_sys – системное прерывание от первого Ethernet контроллера

Номер входа ЮЕРІС	Источник прерывания
28	USB
29	WLCC

2.7.7.3 Аудиоконтроллер высокого разрешения HDA (high definition audio).

Краткие технические характеристики аудиоконтроллера HDA представлены в таблице 2.13.

Таблица 2.13 - Краткие технические характеристики аудиоконтроллера HDA

Параметр	Значение
Скорость передачи данных, Мбит/с	48
Скорость приёма данных, Мбит/с	24
Число потоков воспроизведения	1
Число потоков записи	1
Число каналов в потоке	до 6
Поддерживаемые частоты дискретизации аудиопотока, КГц	от 6 до 192
Поддерживаемые разрядности квантов (замеров), бит	8, 16, 20, 24 и 32

Структурная схема аудиоконтроллера HDA представлена на рисунке 2.30. Аудиоконтроллер состоит из пяти основных частей: контроллера SLink, блока буферов (буфера команд, буфера ответов, буфера дескрипторов и буфера данных), блока операционных регистров, обработчика аудио потока, контроллера интерфейса с кодеком (High Definition Audio Interface).

Контроллер работает в двух режимах: «slave» и «DMA».

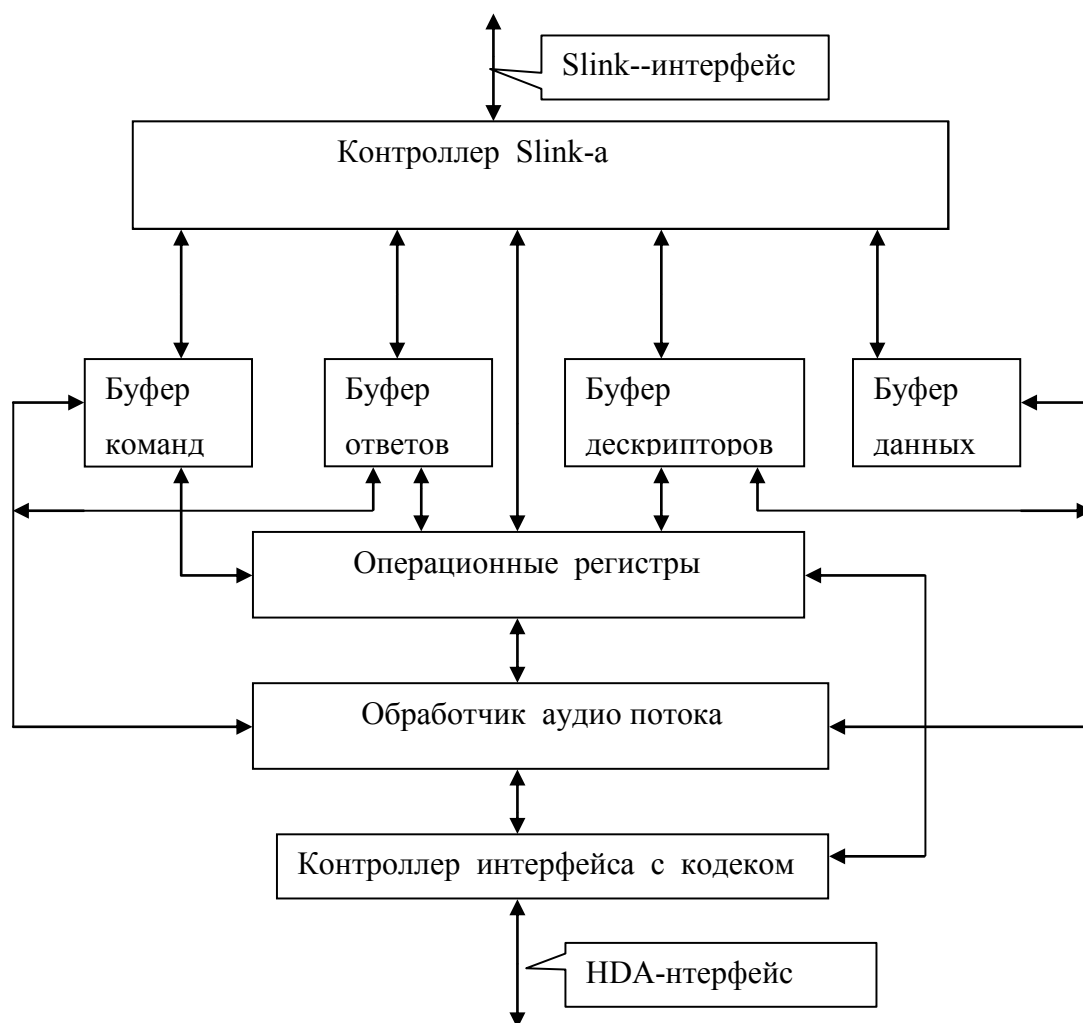


Рисунок 2.30 - Аудиоконтроллер HDA

В «slave» режиме запросы записи/чтения регистров, приходящие от коммутатора контроллера EIOH, помещаются во внутренний буфер клиента системного интерфейса, затем по очереди выполняются. Выполнение следующего запроса невозможно до завершения текущего. Пакеты завершения непочтовых «slave» операций складываются во внутреннее FIFO «завершения slave операций/DMA запросов».

В «DMA» режиме сформированный в блоке буферов пакет запроса поступает в клиент и (так же, как пакеты завершения «slave» операций) помещается в FIFO. При наличии доступа к системной шине клиент считывает очередной пакет из FIFO и выдаёт его на шину. Если клиент одновременно поступают сигнал завершения не почтовой «slave» операции и «DMA» запрос, то пакет завершения «slave» операции будет записан в FIFO первым.

Буферы команд, ответов и дескрипторов являются кольцевыми и содержат соответствующие структуры данных. Буфер данных – линейный, содержит сэмплы аудиоданных.

2.7.7.4 Контроллер последовательного порта для реализации интерфейса RS-232.

Контроллер последовательного порта EIOH_SP обеспечивает микропроцессор интерфейсами RS-232/RS-485 подключением к внешним микросхемам физического уровня (конкретный тип интерфейса определяется соответствующей микросхемой физического уровня). В микропроцессоре реализовано два контроллера EIOH_SP.

Контроллер EIOH_SP подключается к коммутатору EIOH через интерфейс Iolink. Контроллер состоит из контроллера последовательных каналов SCC, а также, блока реализующего подключение к интерфейсу Iolink и содержащего регистры конфигурационного пространства. Структурная схема контроллера EIOH_SP показана на рисунке 2.31.

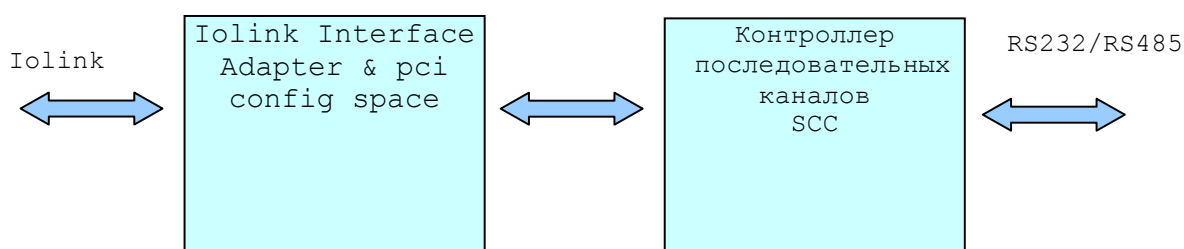


Рисунок 2.31 - Контроллер EIOH_SP

Контроллер EION_SP виден системе как функция 2 устройства 2 на внутренней шине EION. Взаимодействие с контроллером осуществляется через операционные регистры. Для доступа к операционным регистрам контроллера последовательного канала в пространстве памяти выделены регистры, представленные в таблице 2.14.

Таблица 2.14 – Операционные регистры контроллера SCC

Параметр	Возможные значения	Значение по умолчанию	Регистр
Скорость, бит/с* _____ * при pclk = 4,9152 МГц	150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 76800, 115200	---	WR11 WR12 WR13 WR6 WR7
Кодировка	NRZ, NRZI, FM0, FM1	NRZ	WR10
Количество информационных битов на принимаемый символ	5,6,7,8	8	WR3
Количество информационных битов на передаваемый символ	5,6,7,8	8	WR5
Бит чётности	есть/нет, если есть, валиден 0 или 1	нет	WR4
Количество стоп-битов на символ	1, 1.5, 2	1	WR4
Источник сигнала синхронизации приёма	внешние сигналы - RTxC, TRxC, генератор частоты в бодах - BRG, схема цифровой АПЧ - DPPL	---	WR11
Источник сигнала синхронизации передачи	внешние сигналы - RTxC, TRxC, генератор частоты в бодах - BRG, схема цифровой АПЧ – DPPL	---	WR11

Контроллер последовательных каналов SCC включает два полностью независимых полнодуплексных последовательных канала передачи.

Структурная схема контроллера SCC приведена на рисунке 2.32.

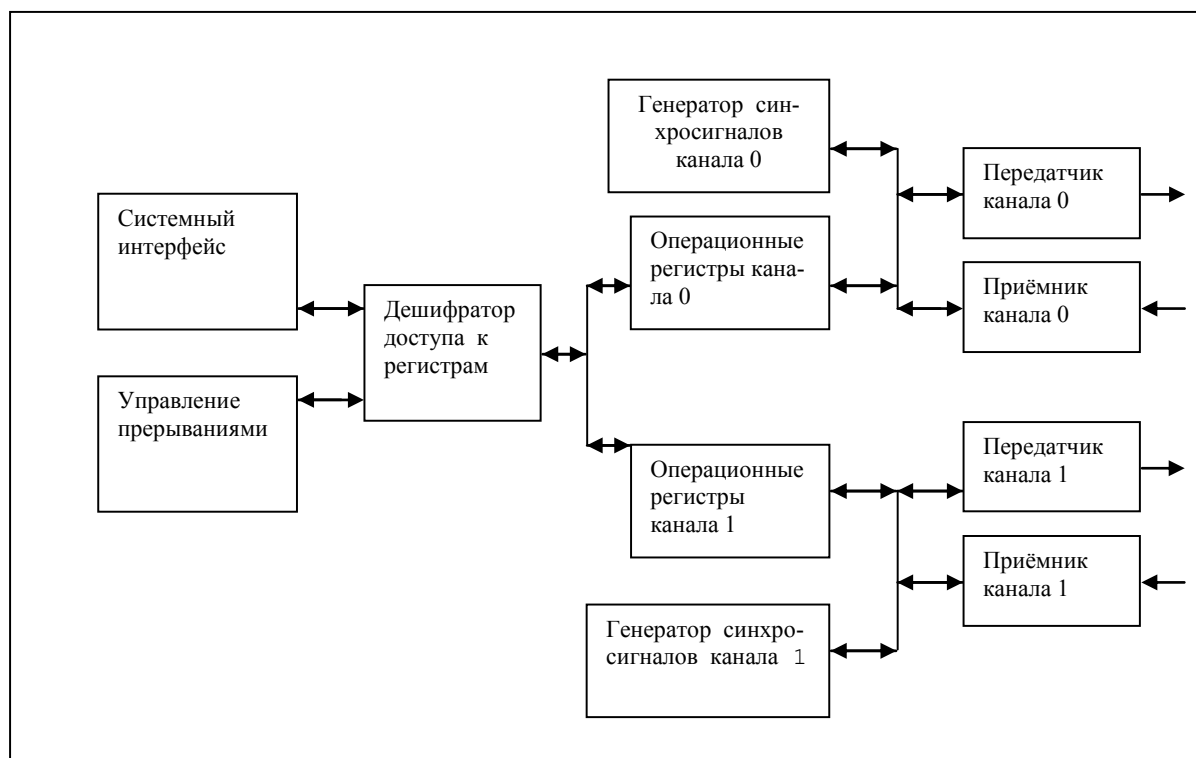


Рисунок 2.32 – Контроллер последовательного канала SCC

2.7.7.5 Контроллер управления питанием и энергосбережением SPMC.

Аппаратная поддержка управления энергосбережением на уровне вычислительного комплекса реализована в контроллере SPMC, который находится в домене неотключаемого питания. Управление энергосбережением производится программно из операционной системы путем записи/считывания программно-доступных регистров контроллера SPMC и обработки прерываний обусловленных событиями энергосбережения, источником которых является SPMC.

Программно-доступные регистры SPMC позволяют реализовать следующие функции энергосбережения:

- отслеживание и обработка событий от кнопки питания;
- отслеживание и обработка событий от источника питания (переключение с питания от сети на питание от батареи);

- отслеживание и обработка событий от батареи (низкий уровень заряда батареи);
- отслеживание и обработка событий от таймера, отслеживающего состояния простоя (генерация прерывания при длительном состоянии простоя);
- перевод системы в состояние сна (пониженного энергопотребления) при длительном состоянии простоя (отсутствия задач);
- отслеживание и обработка событий от USB.

Контроллер SPMC поддерживает следующие состояния энергосбережения на уровне вычислительного комплекса:

- S0 - active state;
- S3- suspend-to-ram;
- S4 – suspend-to-disk;
- S5 – soft off.

За отключение питания на плате отвечают выходные сигналы sleep_s3_, sleep_s4_, sleep_s5. Сигнал atn_sus может быть использован в качестве retention для режима S3 (suspend-to-ram).

В системах, где отсутствует «suspend- питание неотключаемое в режимах энергосбережения», сигналы sus_pwrok и sys_pwrok должны быть объединены.

Сигнал sus_pwrok является сбросом для всего контроллера.

Если во время перехода sus_pwrok из 0 в 1 сигнал ac_power_psnt равен 1, то сигналы sleep_s3_, sleep_s4_, sleep_s5_ переходят в неактивное состояние ($\rightarrow 1$), включая блок питания без дополнительного нажатия кнопки питания (power button).

Если во время перехода sus_pwrok из 0 в 1 сигнал ac_power_psnt равен 0, то сигналы sleep_s3_, sleep_s4_, sleep_s5_ останутся в активном состоянии ($== 0$), и для включения блока питания потребуется дополнительное нажатие кнопки питания (power button).

Дополнительное требование на сигнал ac_power_psnt — он должен быть стабилен (1мкс до и 1 мкс после) относительно переднего фронта (из 0 в 1) сиг-

нала `sus_rwrok`. В противном случае неизвестно потребуется или нет дополнительное нажатие кнопки питания для включения блока питания.

Вход в состояние энергосбережения может осуществляться либо в автоматическом (без вмешательства ПО), либо в программном режиме.

В автоматическом режиме возможен только переход в состояние `s5` при нажатии и удержании кнопки «POWER» более 4 сек. В остальных случаях используется программный переход, инициатором которого может быть `sci`-прерывание.

Источники прерывания в `s0` состоянии:

- а) РМ таймер;
- б) нажатие кнопки «POWER»;
- в) изменение сигнала `ac_power_psnt` (подключение - отключение от сети переменного тока).

Последовательность перехода в состояние энергосбережения:

1) при переходе в состояния `s3`, `s4`, `s5` сначала выставляется сигнал `atn_sus` на время определяемое значением регистра `ATNSUS_CNT`, ноль в регистре означает, что этот сигнал не должен выставляться;

2) затем этот сигнал переходит в неактивное состояние с одновременным выставлением сигналов `sleep_s3_`, `sleep_s4_`, `sleep_s5_` в значения соответствующее состоянию перехода:

`s3` — `sleep_s3_` → 0, `sleep_s4_` → 1, `sleep_s5_` → 1

`s4` — `sleep_s3_` → 0, `sleep_s4_` → 0, `sleep_s5_` → 1

`s5` — `sleep_s3_` → 0, `sleep_s4_` → 0, `sleep_s5_` → 0

При переходе в состояние энергосбережения как в программном режиме, так и в автоматическом, сигнал `ant_sus` должен быть выставлен в активный уровень.

Выход из состояний `s3`, `s4`, `s5` энергосбережения не возможен, если `ac_power_psnt` == 0 и `batlow_` == 0 (система работает от батареи и её заряд недостаточен).

Инициаторами выхода из состояния энергосбережения могут быть следующие события:

- а) нажатие кнопки «POWER» (инициирует выход из состояний s3, s4, s5);
- б) получение запроса на выход из состояния энергосбережения wake_up_ (инициирует выход из состояний s3, s4).

Примечание:

1. Минимальное время нахождения в состояниях s3, s4, s5 равно 1 сек, и достигается путем маскирования сигналов выхода из состояний энергосбережения (wake_up_ и pwr_btn_) на это время.
2. Минимальное время гарантирующее срабатывание сигналов wake_up_ и pwr_btn_ - 5 тактов (более 4 тактов) входного сигнала sus_clk, при любых меньших длительностях существует только вероятность срабатывания (в зависимости от попадания на фронт сигнала sus_clk)
3. Нельзя постоянно держать низкий уровень у сигнала pwr_btn_. Для включения машины обязательно должно произойти переключение этого сигнала.

Последовательность выхода из состояния энергосбережения:

- 1 - сигналы sleep_s3_, sleep_s4_, sleep_s5_ переходят в неактивное состояние (→ 1);
- 2 - ожидание установления всех системных питаний sys_pwrok = 1;
- 3 - ожидание окончания ресетов (процессор может выполнять команды);
- 4 – ожидание перехода atn_sus в неактивное состояние (процессор может работать с системной памятью).

2.7.7.6 Контроллер GPIO/MPV.

Контроллер GPIO/MPV является контроллером программируемых входов-выходов и имеет встроенный функциональный блок приема синхросигналов от систем реального времени или от генераторов меток точного времени.

Контроллер GPIO/MPV позволяет программным способом управлять своими 16 линиями. Контроллер имеет следующие параметры:

- 16 линий ввода/вывода;
- системный интерфейс IOLink;
- настраиваемое направление линий (ввод или вывод);
- две группы прерываний (любую линию можно приписать к одному из двух прерываний);
- настраиваемый режим формирования прерываний (по фронту или уровню сигнала для каждой линии);
- программируемый уровень срабатывания прерывания для каждой линии.

Блок приема синхросигналов от систем реального времени или от генераторов меток точного времени (МТВ) обеспечивает микросекундную точность определения прихода этих сигналов относительно друг друга и позволяет:

- принимать до трёх линий сигналов (2:0) внешних прерываний в соответствии с программно задаваемой маской и полярностью;
- формировать сигналы прерывания системы, а также измерять время от получения сигнала прерывания до его восприятия системой (путём считывания значения счётчика текущего времени от момента получения прерывания);
- генерировать прерывания для каждой линии (2:0), выдавать генерируемый сигнал прерывания в систему через внешний вывод;
- выдавать в систему прерывания по трём линиям – по одной сигналы от каждой внешней линии;
- передавать в систему внешний сигнал прерывания для Ethernet контроллера напрямую для поддержки стандарта ieee-1588.

Работой блока управляют с помощью операционных регистров, располагающихся в пространстве памяти.

2.8 Графическая подсистема

2.8.1 Состав

2.8.1.1 Графическая подсистема микропроцессора Эльбрус-2С3 состоит из шести специализированных блоков (ядер), предназначенных для ускорения обработки и вывода на мониторы видеоинформации: ускоритель трехмерной графики 3D GPU, ускоритель двумерной графики 2D GPU, ускорители декодирования видео VXD и VP9D, ускорители кодирования видео VXE и VP9E.

2.8.2 Ускоритель трехмерной графики 3D GPU

2.8.2.1 Ускоритель трехмерной графики 3D GPU предназначен для создания растрового (двухмерного) изображения трехмерной сцены. Трехмерная сцена задается в виде координат и свойств вершин, составляющих объекты в трехмерном пространстве, и исполнением программ для обработки этих вершин, текстур (растровых изображений, наносимых на трехмерные и двумерные объекты), исполнением программ для обработки пикселей, получаемых во время растеризации, а также обработки прочих параметров. Как исходные данные трехмерной сцены, так и результирующее изображение, хранятся в памяти, непосредственный доступ к которой имеет устройство 3D GPU.

Ускоритель 3D GPU выполняет следующие типы работ:

- 3D графика, которая заключается в обработке вершин и пикселей для воспроизведения 3D изображений;
- 2D графика, которая используется для загрузки фоновых изображений и выполнения основных 2D операций, таких как копирование и наложение фрагментов изображений в памяти, и другие растровые операции;
- вычислительные задачи для обработки данных общего назначения.

Архитектура ускорителя 3D GPU полностью соответствует графическим спецификациям DirectX 10, OpenGL 3.2, OpenGL ES 3, OpenCL 1.2 EP, Vulkan 1.0, Open VX 1.x and Renderscript.

Ускоритель 3D GPU содержит многопоточковые унифицированные Shading кластеры (Multi-threaded Unified Shading Cluster (USC)), архитектура АЛУ которого имеет высокую эффективность SIMD обработки и поддерживает задержанный рендеринг с параллельной обработкой множества прямоугольников. Shading кластер включает в себя пиксельный shader, вертекс shader и вычислительный shader pixel shader, vertex shader и вычислительный shader.

Структурная схема ускорителя 3D GPU представлена на рисунке 2.33.

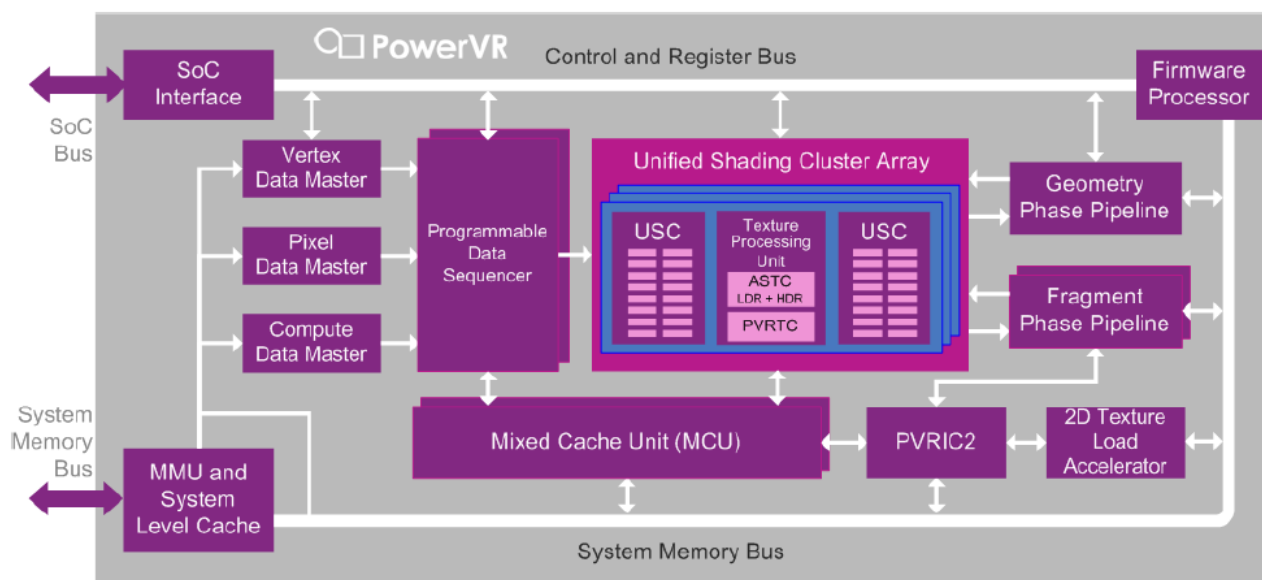


Рисунок 2.33 - Ускоритель трехмерной графики 3D GPU

Краткая характеристика ускорителя 3D GPU:

- выполнение каждый такт 384 операций с плавающей запятой формата 32 или 768 операций с плавающей запятой формата 16;
- шесть Shading кластеров;
- 16 АЛУ с четырьмя конвейерами каждое в каждом кластере;
- АЛУ с выполнением 16-разрядных операций сложения с умножением с плавающей запятой;

- кэши локальных данных, текстур и команд;
- команды переменной длины;
- скалярная и векторная SIMD модели выполнения команд.

2.8.3 Ускоритель двумерной графики 2D GPU

2.8.3.1 Структурная схема.

Ускоритель двумерной графики 2D GPU, предназначен для решения ряда задач, возникающих при работе оконной системы: заливка и копирование прямоугольных областей (как в видеопамяти, так и между видеопамтью и памятью процессора с изменением форматов пикселей), наложение полупрозрачных изображений, преобразование цветовых пространств, требуемых при выводе видео, и прочего.

Ускоритель 2D GPU имеет четыре внешних интерфейсов: 2 HDMI, LVDS и RGB, поддерживает видеорежимы вплоть до 4096x2160x60Hz и работу трех мониторов.

Структурная схема ускорителя 2D GPU представлена на рисунке 2.34. Ускоритель 2D GPU содержит:

- три дисплейных контроллера DC0, DC1 и DC2;
- регистровый блок;
- два контроллера блиттеров BB0 и BB1;
- устройство автоматического управления блиттерами;
- мультиплексор AXI-интерфейса в видеопамть для дисплейных контроллеров;
- мультиплексор AXI-интерфейса в системную память;
- мультиплексор AXI-интерфейса в видео память для блиттеров;

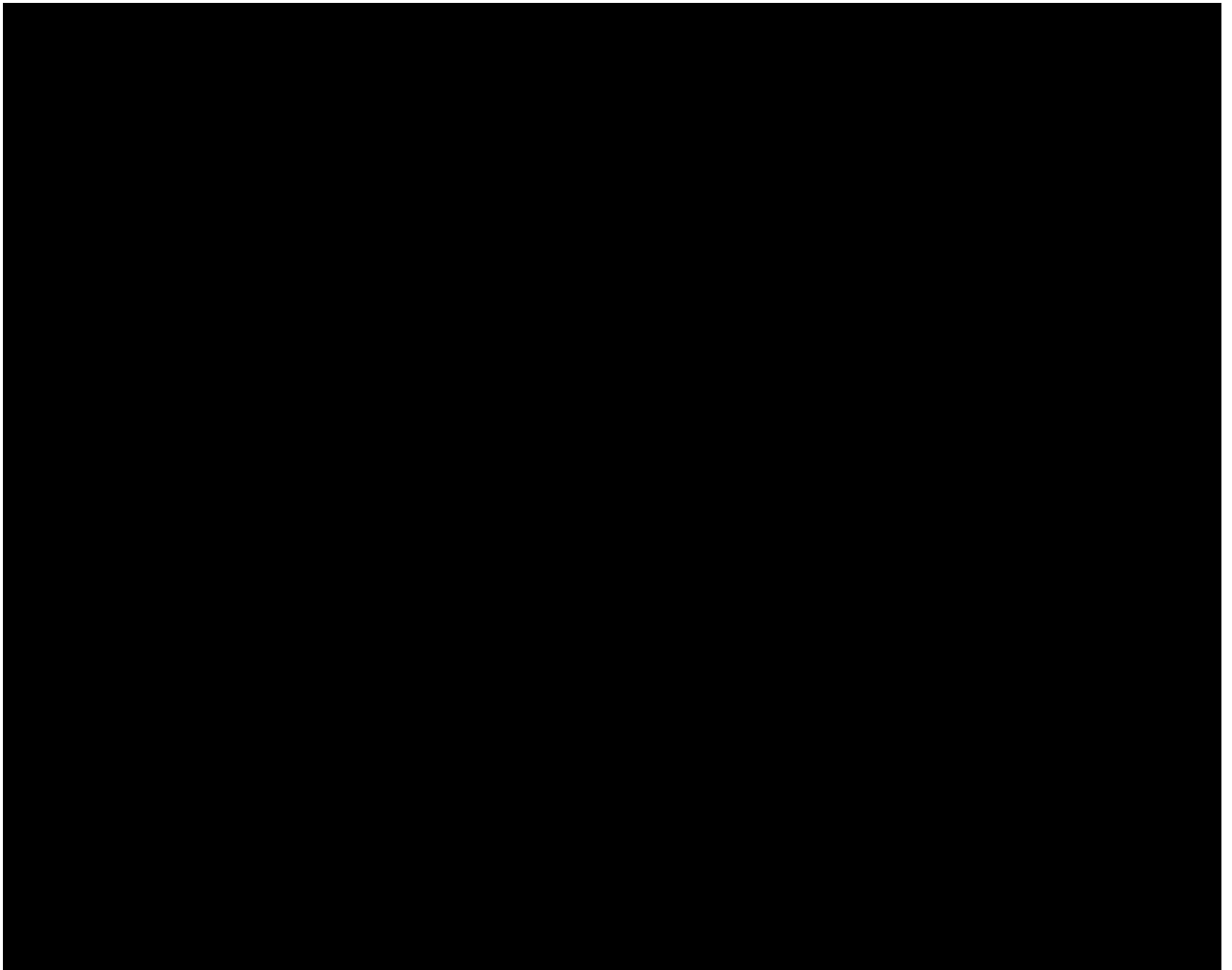


Рисунок 2.34 - Ускоритель 2D GPU

- мультиплексор видео выходов;
- контроллеры внешних интерфейсов HDMI0, HDMI1, LVDS, RGB.

2.8.3.2 Дисплейный контроллер.

Структурная схема дисплейного контроллера представлена на рисунке 2.35.

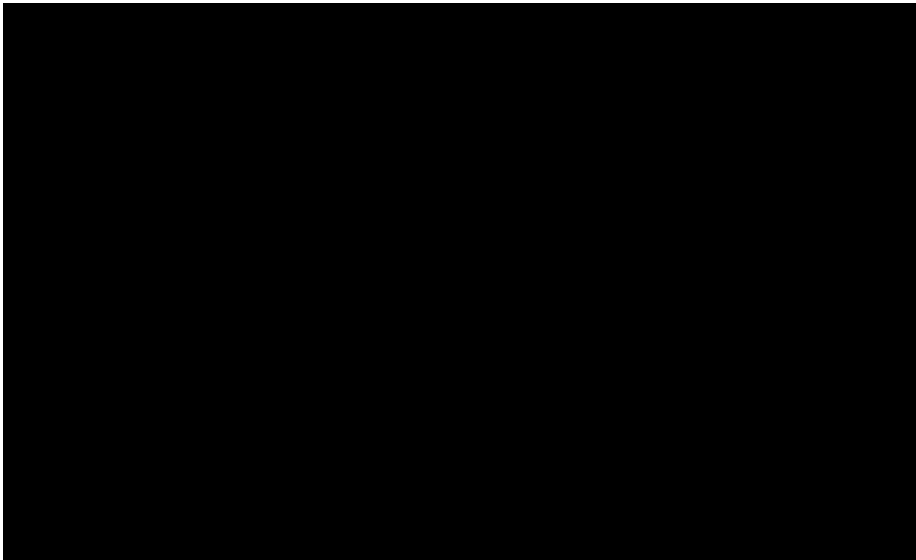


Рисунок 2.35 - Дисплейный контроллер

Дисплейный контроллер обеспечивает:

- видеоинтерфейсы:

а) HDMI0 до 4096x2160x60Hz;

б) HDMI1 до 4096x2160x60Hz;

в) LVDS:

1) до 4096x2160x30Hz;

2) до 2560x1600x60Hz;

г) RGB до 2560x1600x60Hz;

- цветовые разрешения: 8bpp+палитра, 15bpp, 16bpp, 24bpp, 32bpp;

- наличие таблиц гамма-коррекции и дизеринга: гамма коррекция (3x256 8bit LUT), dither LUT;

- полная совместимость с VGA в соответствии с IBM VGA functional description;

- поддержка аппаратного курсора размером 64x64 пикселей: ARGB8 src-over;

- поддержка оверлея.

Ядро 2D-ускорений состоит из двух ROP2-блиттеров и блока автоматического управления блиттерами без вмешательства процессора.

Задача блиттера состоит в быстром перемещении изображений (прямоугольных областей памяти) в видеопамяти, в системной памяти, а также между ними, с одновременным выполнением какого-либо вида обработки с перемещаемым изображением. Каждая операция блиттера производится над двумя независимыми областями в памяти, которые могут соответствовать каким-либо изображениям или их частям. Одна область служит только источником данных, другая – как источником, так и приёмником. При этом любой источник может быть заблокирован, а вместо него будет производиться подстановка константы (это даёт возможность, например, не допускать холостых чтений памяти при операциях копирования или заполнения константой).

Операции, выполняемые блиттером, подразделяются на два независимых класса: класс битовых операций и класс альфа-операций.

Операции первого класса рассматривают данные из обоих источников, как две последовательности битов, и производят вычисление произвольной логической функции двух переменных над ними. Результат таких вычислений образует данные, записываемые в приёмник блиттера. Всего существует 16 различных функций двух логических переменных. Путём выбора подходящей функции получают такие операции над изображениями, как копирование, наложение по XOR, OR, AND, инверсия и т.д.

Операции второго класса оперируют с 4-компонентными пикселями, представленными тремя компонентами цветов и одной компонентой альфа-канала. Каждая компонента представлена максимум 8-битным числом.

Альфа-компоненту можно приближённо рассматривать, как степень прозрачности изображения в данном пикселе. Вычисление производится путём арифметических (умножения, сложения, вычитания и деления) преобразований альфа- и цветовых компонент. Конкретный вид преобразования задаётся требуемой альфа-операцией.

Структурная схема ROP2-блиттера изображена на рисунке 2.36.

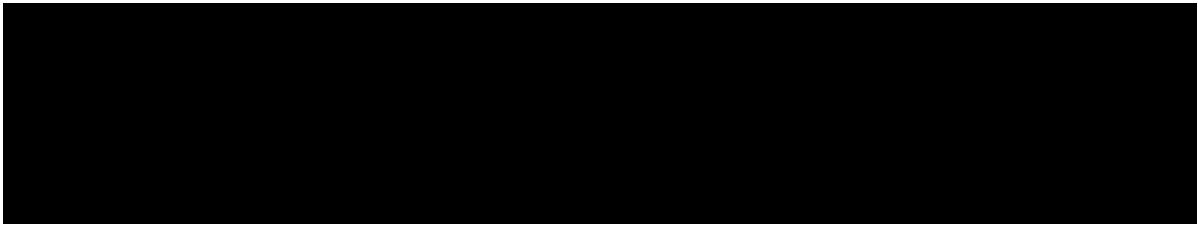


Рисунок 2.36 - ROP2-блиттер

Контроллер источника генерирует запросы чтения данных на шине, либо видеопамяти, либо системной памяти через интерфейс источника, затем в зависимости от класса операции производит предварительную обработку/форматирование, например такую как расширение цветности из монохромного изображения, преобразование YUV цветового пространства в RGB, или подготовка четырех компонент пикселя для альфа-операции. Это обеспечивает входной поток данных источника в процессор данных.

Процессор данных производит окончательную обработку данных, как это задано в параметрах операции, над данными из контроллера источника и контроллера приемника.

Контроллер приёмника генерирует запросы записи и, в зависимости от операции запросы чтения данных на шине, либо видеопамяти, либо системной памяти через интерфейс приёмника, и обеспечивает запись результата в область памяти приёмника. Данный модуль всегда работает с RGB данными (записывает/считывает). Поэтому для операций, которые требуют данные из области приёмника, данные передаются в процессор данных как есть, т.е. без предварительной обработки.

Контроллер потока данных следит за адресом приёмника, помогает генерировать адрес источника, и обеспечивает запись в память приёмника нужного количества байт в строке и количества строк.

Основные возможности и характеристики ROP2-блиттера:

- 64-битная внутренняя архитектура;
- поддерживается 8-ми, 16-ти, 24-х и 32-х битная глубина цветности;
- расширение цветности из монохромного изображения (Color Expansion);
- размножение образа размером (Pattern Copy);
- 16 растровых операций, выраженные формулой:

$$\begin{aligned} & \text{ROP [0]} * (\text{DST} \& \text{ SRC}) \mid \\ \text{DST} = & \text{ROP [1]} * (\text{DST} \& ! \text{ SRC}) \mid \\ & \text{ROP [2]} * (! \text{ DST} \& \text{ SRC}) \mid \\ & \text{ROP [3]} * (! \text{ DST} \& ! \text{ SRC}) \mid \end{aligned}$$

- поддержка реверсивного изменения адреса;
- конвейерный запуск операций;
- режим доступа в системную память для источника и/или приёмника;
- поддержка alpha-операций (Clear, Src, Dst, Over, OverReverse, In, InReverse, Out, OutReverse, Atop, AtopReverse, Xor, Add, Saturate, Disjoint*, Conjoint*).

Эти возможности позволяют обеспечить полную поддержку функциональности X.Org EXA/XAA.

Задача блока автоматического управления – осуществлять пакетную обработку изображений путём выставления последующих задач для блиттера по мере выполнения им предыдущих. Блок работает по программе, представленной на рисунке 2.37, которая состоит из дескрипторов (один дескриптор – одно задание блиттера), организованных в виде односвязного списка и располагающихся в системной памяти. Процесс установки задачи для блиттера состоит в последовательной записи определённых значений в определённые регистры блиттера (конкретные номера регистров и записываемые значения заданы в дескрипторе) и последующее ожидание готовности блиттера для приёма нового задания.

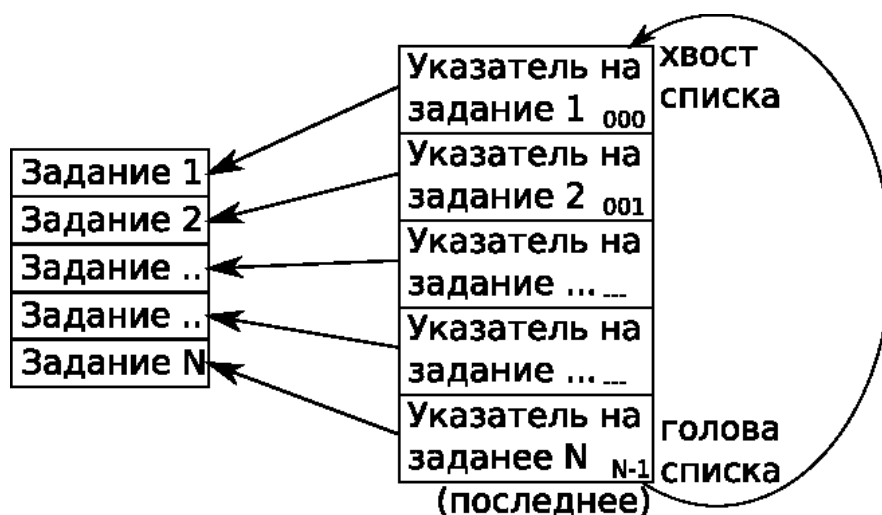


Рисунок 2.37 - Программа блока автоматического управления

В процессе пакетной обработки блок автоматического управления поддерживает в системной памяти ячейку, индицирующую состояние выполненности (сколько дескрипторов поданы на исполнение и т.д.). Такая организация позволяет избежать постоянных времязатратных обращений процессора во внешние Ю-регистры блока.

Ускоритель двухмерной графики 2D GPU имеет возможность воспроизведения звука через видеointерфейсы HDMI. Это достигается за счет того, что ускоритель 2D GPU имеет в своем составе два независимых HDA-аудиоконтроллера HDA0 и HDA1, что позволяет воспроизводить два независимых звуковых потока с частотой дискретизации 48 КГц.

2.8.4 Ускорители декодирования и кодирования Google

2.8.4.1 Ускоритель декодирования и кодирования видео данных Google декодер VP9D и кодер VP9E предназначены для распаковки и сжатия видеоизображения соответственно в формате VP9. Данный видео формат используется в видео-конференц связи, в подавляющем большинстве интернет-браузеров для видео трансляций и видео связи, в охранных системах видео наблюдения, а также для хранения видео архивов.

2.8.4.2 Ускоритель декодирования видео декодер VP9D.

Мультиформатный декодер VP9D совместим с интерфейсами шин AXI и АНВ. Декодер поддерживает 128-разрядный интерфейс AXI master. Поддерживаемые slave интерфейсы – 32-разрядные AXI и АНВ

Поддерживаемые стандарты, профили и уровни представлены в таблице 2.15.

Таблица 2.15 – Поддерживаемые стандарты, профили и уровни

Стандарт	Поддержка декодера
ITU Rec. (04/2013) H.265 или ISO/IEC 230082	Main Profile, levels 1 – 5.1, High Tier
VP9	VP9, Feature Set 0

Структурная схема декодера G2 представлена на рисунке 2.38.

Входные параметры декодирования представлены в таблице 2.16.

Таблица 2.16 - Входные параметры декодирования

Параметр	Значение параметра
Формат входных данных	HEVC: HEVC NAL unit stream
	VP9: VP9 Bitstream
Схема декодирования	Кадр за кадром
Формат выходных данных	YCbCr 4:2:0 semiplanar 4x4 tiled
Поддерживаемый размер изображения	От 180x180 до 4096x2304
	Шаг 16 пикселей
Максимальная фреймовая скорость	60 фреймов/с при 3840 x 2160
Максимальная битовая скорость	HEVC: Как определено HEVC MP level 5.1
	VP9: Как определено спецификацией VP9 bitstream
Обнаружение и исправление ошибки	Поддерживается

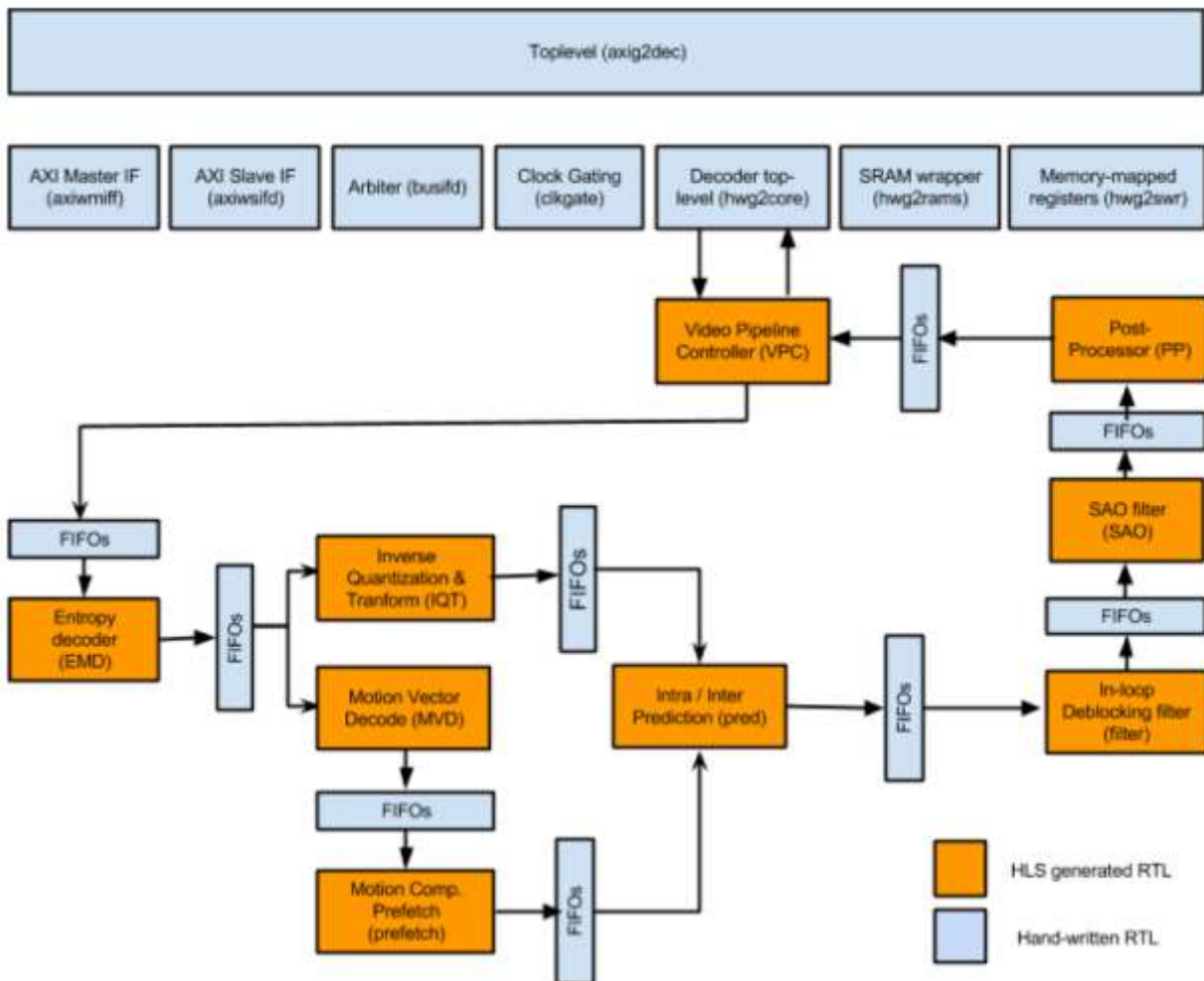


Рисунок 2.38 – Декодер G2

Выходные параметры декодирования представлены в таблице 2.17.

Таблица 2.17 - Выходные параметры декодирования

Параметр	Значение параметра
Формат входных данных	YCbCr 4:2:0 tiled 4x4
Источник входного изображения	Вход декодера
Формат выходных данных	YCbCr 4:2:0 semiplanar
	Big или little endian
Размер выходного изображения	От 16x16 до 4096x23044
	Шаг 16 пикселей

Параметры обмена декодера представлены в таблице 2.18. Отметим, что режимы порядка следования байтов могут быть установлены отдельно для входных и выходных данных.

Таблица 2.18 – Параметры обмена

Параметр	Значение
AXI Single Command Multiple Data	Да
Одновременная работа AXI каналов чтения и записи	Да
Групповые передачи AXI	Да. Максимальный размер для каналов чтения и записи может быть установлен отдельно перед синтезом декодера (по умолчанию значение 50 для обоих)
Групповые передачи AXI с обгоном	Нет
Адрес памяти	128- или 64-разрядный выровненный адрес, без байтовой адресации
Ограничение максимальной длины выдачи AXI групповой передачи	Да, любое значение до 1256
Метод опроса прерывания или уровень, основанный на прерывании по нарушению разрешенного порядка байтов	Да; 8/16/32/64-разрядный обмен, разрешаемый/отключаемый Google

Опции конфигурации представлены в таблице 2.19.

Таблица 2.19 – Опции конфигурации

Параметр	Значение
Одна команда много данных AXI	Да
Одновременная работа AXI каналов чтения и записи	Да
Групповые передачи AXI	Да. Максимальный размер для каналов чтения и записи может быть установлен отдельно перед синтезом декодера (по умолчанию значение 50 для обоих)
Групповые передачи AXI с обгоном	Нет
Адрес памяти	128- или 64-разрядный выровненный адрес, без байтовой адресации

Параметр	Значение
Ограничение максимальной длины выдаваемого блока AXI	Да, любое значение до 1256
Метод прерывания	Опрос или базированный уровень прерывания
Конфигурируемый порядок следования байтов	Да; 8/16/32/64/128-разрядная перестановка разрешена/запрещена

Работа декодера показана на рисунке 2.39.

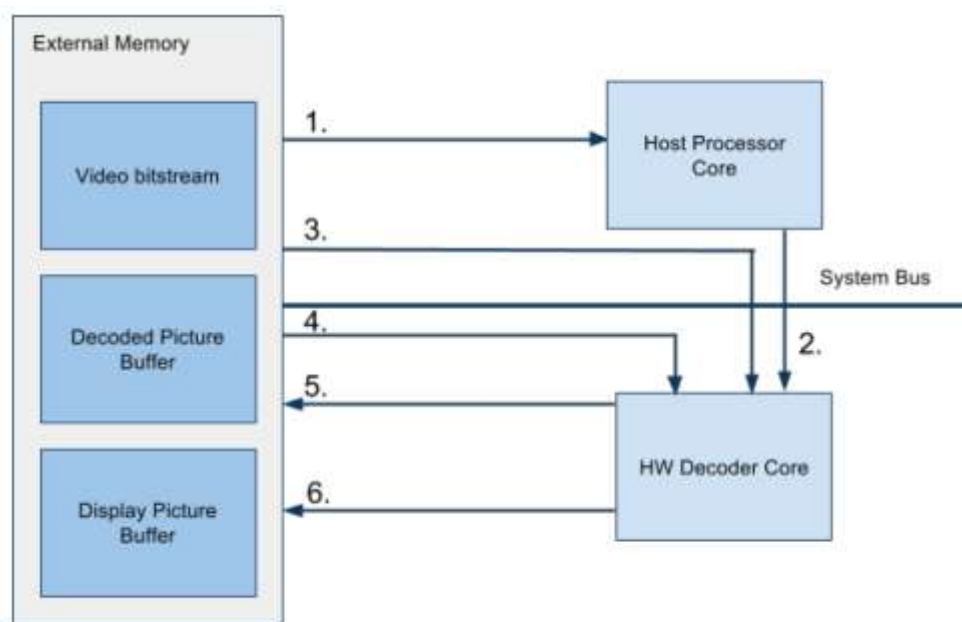


Рисунок 2.39 – Декодер G2 и поток обращений во внешнюю память

Программное обеспечение (ПО) декодера начинает декодирование анализом заголовков потока (1). Затем ПО устанавливает управляющие регистры (размер изображения, начальный адрес потока и т.д.) и разрешает работу аппаратуры (2).

Аппаратура декодирует изображение путем чтения потока (3), таблиц САВАС и эталонных изображений (4) (необходимо для межкадрового декодирования) из внешней памяти. Если видео поток содержит V изображения, Декодер будет дополнительно записывать в *direct mode* векторы движения в память во время P изображений и считывать их в время V изображений.

Аппаратура записывает декодированное изображение в память (5). Когда изображение будет полностью декодировано или аппаратура исчерпала поток данных, она выдает прерывание с соответствующим статусным флагом, выдает адрес конца потока для ПО для продолжения и возвращается в исходное состояние.

Если аппаратура обнаруживает поток с ошибкой при декодировании, она выдает прерывание ошибки, и ПО начинает удаление ошибки из текущего изображения.

Декодер может также записать дополнительный кадр для процесса дисплея (6). Этот кадр является растроканированной версией эталонного кадра.

Декодер выполняет обращение во внешнюю память с одним контекстом данных (одним типом данных) в данное время, например, транзакции входного потока чтения и чтение эталонного изображения всегда выполняются отдельно. Каждое такое обращение может содержать различное количество выданных групповых передач.

Групповые передачи одного контекста выдаются как незаконченные групповые передачи. Декодер может начинать выдачу групповых передач следующего контекста до того, как данные из предыдущего контекста считаны декодером или приняты slave. Каналы чтения и записи независимы и работают одновременно.

Порядок, в котором эти контексты данных передаются, не может быть полностью детерминированным, так как порядок зависит от схемы кодирования каждого макроблока.

Выходное эталонное изображение записывается в порядке растрового сканирования tile за tile и суперблок за суперблоком. Tiles конфигурируемы по размеру, а суперблоки всегда состоят из 64x64 пикселей. Обычно, блоки яркости 32x32 для каждого суперблока выдаются в порядке растрового сканирования. Исключениями из этого порядка является суперблок на правом краю, нижнем краю, нижнем правом углу или, если вертикальное разбиение применяется декодером к суперблоку, на первом слое разбиения.

Декодер имеет два интерфейса шин - master и slave.

Интерфейс master позволяет аппаратуре декодера обращаться к внешней памяти для чтения, например, входного потока данных и данных эталонного изображения, а также для записи данных выходного изображения.

Интерфейс slave позволяет CPU обращаться к регистрам аппаратуры декодера для записи и чтения управляющих и конфигурационных данных, включения и сброса аппаратуры, чтения прерывания и флагов состояния.

Декодер соединен с системной шиной 128-разрядными AXI шинами. Ширина шины данных master задается установкой в регистре декодера. Ширина шины данных slave всегда 32 бит.

2.8.4.3 Ускоритель кодирования VP9E.

Ускоритель кодирования VP9E является кодером VP9 профиль 0. Поддерживаемые параметры представлены в таблице 2.20. Кодер совместим с 128-разрядными AXI master, и 32-разрядными AXI и AHB slave.

Таблица 2.20 – Поддерживаемые параметры VP9

Типы кадров	I, P, Golden и altref кадры
Основные параметры кодирования	10 режимов внутреннего предсказания
	4x4, 8x8, 16x16, 32x32 DCT и/или ADST преобразования
	Loop filter
	Устойчивость к ошибкам: потоковый режим без накопительной вероятностной корректировки
Оценки движения	Область поиска: горизонтальная [128, 127]; вертикальная [64, 63]
	Пиксельная точность: 1/8 пикселя
	SB и subSB разбиения: от 8x8 до 64x64 предсказанных блоков
	Алгоритм: Индексированная оценка движения
Количество эталонных кадров	До 8 (3 одновременно)
Tiles	Автоматическое разбиение потоков на вертикальные tiles шириной до 512 пикселей
	Поддержка до 16 tiles
Масштабирование эталонных кадров	Не поддержан

Типы кадров	I, P, Golden и altref кадры
Сложный прогноз	Поддержан
Вероятностная корректировка вперед и назад	Поддержан
Сегменты	Поддержка двух сегментов
Временная фильтрация	Использование 3/6 входных кадров для создания временного фильтрованного альтернативного эталонного кадра

Структурная схема кодера VP9E представлена на рисунке 2.40.

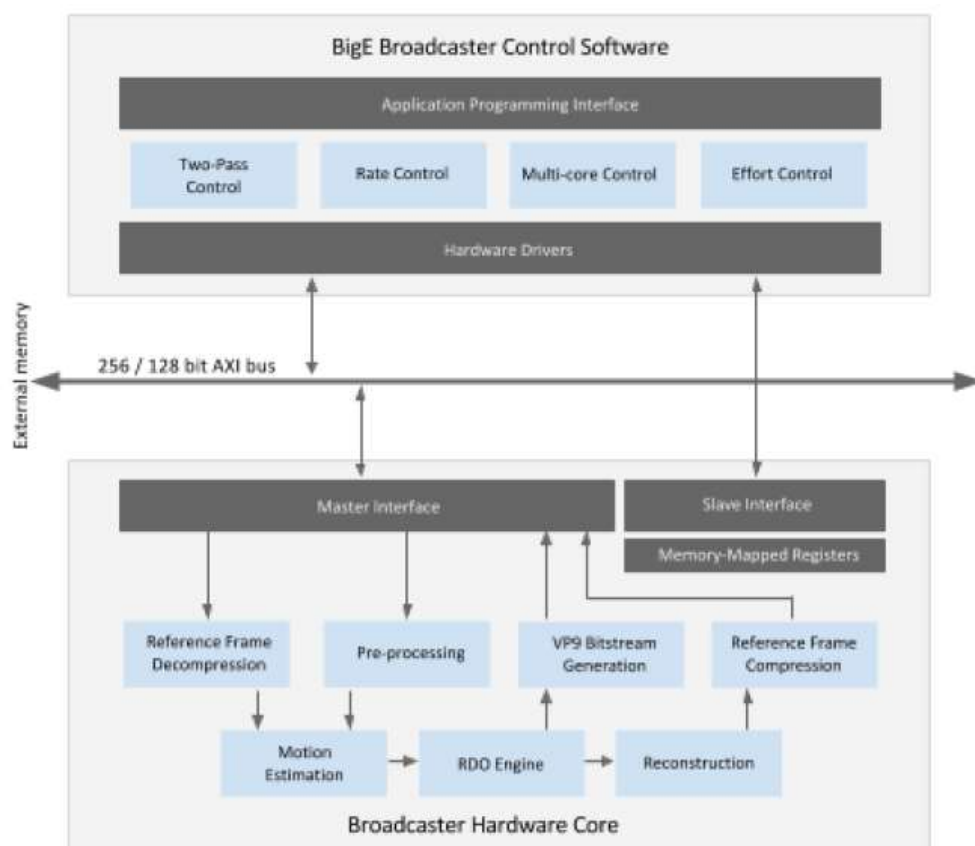


Рисунок 2.40 – Кодер VP9E

Параметры кодера VP9E представлены в таблице 2.21.

Таблица 2.21 – Параметры кодера VP9E

Параметр	Описание
Формат входных данных	YCbCr 4:2:0, 8 разрядов на sample
	Набор форматов RGB и YCbCr поддержан через преобразование цвета в препроцессоре. Независимо от кодировки исходного формата всегда

Параметр	Описание
	переводится в формат YCbCr 4:2:0
Формат выходных данных	VP9 битовый поток
Поддержанный размер изображения	От 132 x 96 до 8128 x 4352
Максимальная кадровая скорость	Не ограничена в аппаратуре *
	30 fps при 3840 x 2160
Битовая скорость	Не ограничена в аппаратуре
	Высокая битовая скорость может влиять на максимальную достижимую производительность
* Реальная максимальная фреймовая скорость зависит от частоты логики, производительности системной шины и использованных установок кодера	

Препроцессорная обработка выполняется в конвейере кодера. Препроцессорная обработка может использоваться только тогда, когда имеются кодирование и входное изображение без препроцессорной обработки. Параметры препроцессорной обработки представлены в таблице 2.22.

Таблица 2.22 - Параметры препроцессорной обработки

Параметр	Описание
От RGB до YCbCr 4:2:0 преобразование цветового пространства	BT.601 или BT.709 преобразование для RGBA (32bit). Поддержаны оба full и studio swing
От YCbCr до YCbCr 4:2:0 цветное пространство	Форматы YCbCr: - YCbCr 4:2:0 planar; - YCbCr 4:2:0 semiplanar (NV12); - YCrCb 4:2:0 semiplanar (NV21); - YCbCr 4:2:2 interleaved (YUYV); - YCrCb 4:2:2 interleaved (UYVY); - YCbCr 4:2:0 semiplanar tiled 4x4 (совместимо с декодером G1); - YCbCr 4:2:0 semiplanar tiled 8x4 (совместимо с декодером G2)

Параметр	Описание
Группирование	Поддержаны размеры кодирования от 8192 x 8192 до любого
Поворот	0, -90, 90 и 180 градусов
	Вертикальный переворот
Уменьшение размера	Максимальное уменьшение размера 1/16 на размерность*
	Масштабирование поддержано только для следующих входных форматов:
	YCbCr 4:2:0 planar
	YCbCr 4:2:0 semiplanar tiled 4x4
	YCbCr 4:2:0 semiplanar tiled 8x4
	Не может использоваться с переворотом и вращением
* Наименьшая размерность может быть ((исходная/16) + 1)	

Параметры обмена представлены в таблице 2.23.

Таблица 2.23 – Параметры обмена кодера VP9E

Параметр	Значение
Одна команда много данных AXI	Да
Одновременная работа каналов чтения и записи AXI	Да
Групповые передачи AXI	Да
Групповые передачи AXI с обгоном	Нет
Адрес памяти	256-разрядный выровненный адрес, без байтовой адресации
Ограничение максимальной длины выдаваемого блока AXI	Да, любое значение до 1256
Метод прерывания	Опрос или базируемый уровень прерывания
Конфигурируемый порядок следования байтов	Да; 8/16/32/64/128-разрядная перестановка разрешена/запрещена

Поток обращений в память показан на рисунке 2.41.

Числа на рисунке 2.41 представляют следующие транзакции:

- 1 - инициализация аппаратуры кодера программным обеспечением через регистры в памяти;
- 2 - чтение входного изображения для кодирования;
- 3 - чтение эталонных изображений для кодирования текущего изображения;

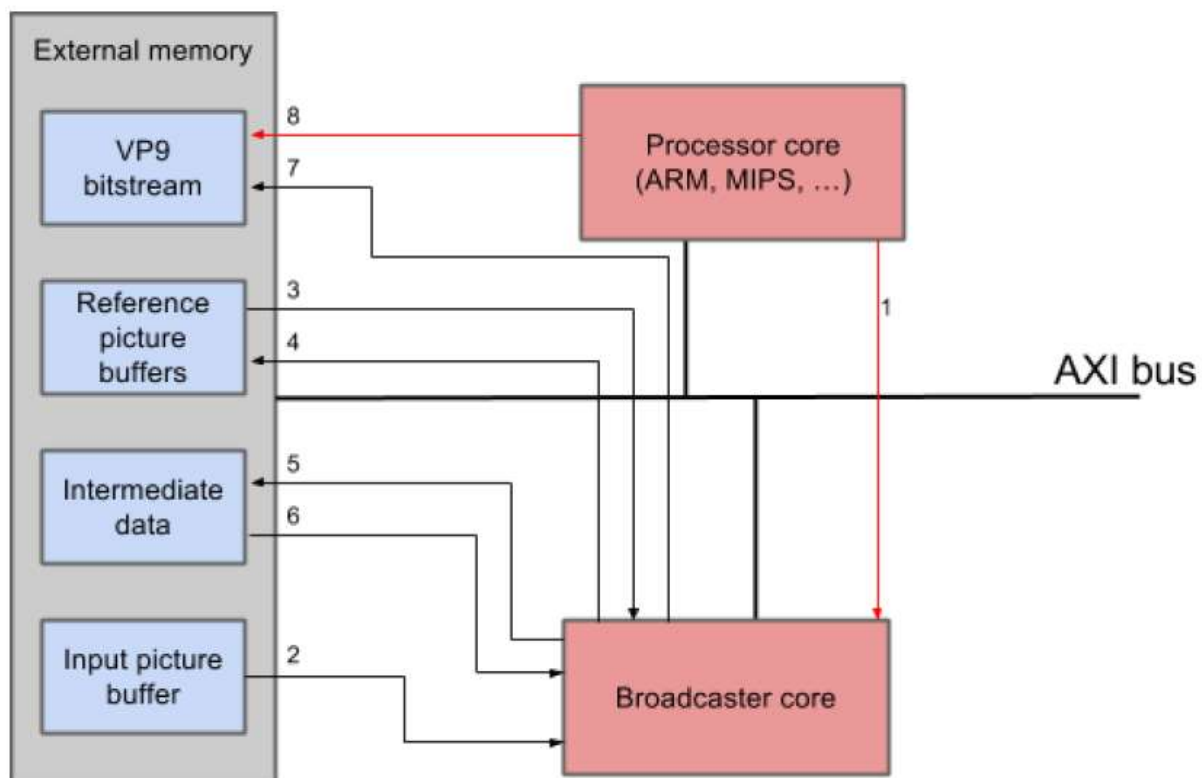


Рисунок 2.41 – Поток обращений в память кодера VP9E

- 4 - запись эталонного изображения для кодирования следующего изображения;
- 5 - запись промежуточных данных (концы tile,..);
- 6 - чтение промежуточных данных;
- 7 - запись выходного битового потока;
- 8 - запись программным обеспечением битового потока заголовков.

Программное обеспечение кодера начинает кодирование первого изображения инициализацией аппаратуры и записью заголовков. После того, как аппаратура закодировала изображение, программное обеспечение вычисляет новые значения квантования, корректирует вероятности для аппаратуры и инициализирует аппаратуру для следующего кадра.

Каналы чтения и записи кодера независимы, чтения и записи могут выполняться одновременно. По каналам чтения и записи кодер обращается во внешнюю память с одним контекстом данных (одним типом данных) в данное время, например, транзакции записи выходного потока и записи эталонного изображения всегда выполняются отдельно. Каждое такое обращение может содержать переменное количество выдаваемых групповых передач.

Групповые передачи, выдаваемые в одном контексте, выдаются как незаконченные групповые передачи. Кодер VP9E может начинать выдачу групповых передач из следующего контекста до того, как данные из предыдущего контекста прочитаны декодером или приняты slave.

Порядок, в котором эти контексты данных передаются, не может быть полностью детерминированным, так как порядок зависит от схемы кодирования каждого суперблока.

Кодер VP9E использует схему сжатия (компрессии) эталонного кадра без потерь для уменьшения нагрузки на DRAM. Эффективность алгоритма сжатия зависит от используемого контента и уровня квантования.

Кодер VP9E имеет два интерфейса шин - master и slave.

Интерфейс master позволяет аппаратуре кодера обращаться во внешнюю память, например, для чтения данных входного изображения и данных эталонного изображения и для записи выходного потока данных.

Интерфейс slave позволяет CPU обращаться к регистрам кодера для записи и чтения данных управления и конфигурации, сброса и разрешения работы кодера, чтения прерывания и флагов состояния.

2.8.5 Ускорители декодирования и кодирования видео VXD и VXE

2.8.5.1 Ускоритель декодирования видео VXD.

Ускоритель декодирования видео VXD предназначен для распаковки сжатого видеоизображений в форматах H.265, H.264, MJPEG и других с разрешением от 64x64 пикселей до 4096x2160 пикселей. Предусмотрена возможность уменьшение финальных изображений до 1/4 от исходного формата.

Ускоритель декодирования VXD применяется для отображения и обработки видео, получаемых из сети Интернет и видеокамер. Также может быть использован в задачах доступа к удаленному рабочему столу (VNC). Кроме того, поддерживается распаковка статического изображения в формате JPEG разрешением до 16384x16384 пикселей.

Подключение к коммуникационной сети микропроцессора выполняется высокоскоростной AXI шиной, через которую ускоритель VXD самостоятельно работает с физической памятью, забирая оттуда сжатый поток видео, там же сохраняя промежуточные кадры и записывая туда финальные кадры распакованного видео. Отдельная управляющая шина используется для доступа к управляющим регистрам со стороны процессорных ядер «Эльбрус». Основные параметры декодера VXD представлены в таблице 2.24.

Таблица 2.24 – Основные параметры декодера VXD

Параметр	Значение параметра
Количество конвейеров обработки пикселей	3
Масштабирование	Поддержано уменьшение масштаба (применимо к видео, но исключено для JPEG)
Вращение	Поддержано для 4:2:0
Сжатие Frame Buffer	Не поддерживается
SOCIF – шина связи процессорами	Шина AXI3
Разрядность шины SOCIF	32 разряда
MEMIF – шина связи с памятью	AXI3
Разрядность шины MEMIF	256 разрядов

Параметр	Значение параметра
Разрядность адреса обращения в память	40 разрядов
Максимальная задержка	384 такта
Встроенный CPU	Микроконтроллер с 128 Кбайт памяти RAM

Структурная схема декодера VXD представлена на рисунке 2.42.

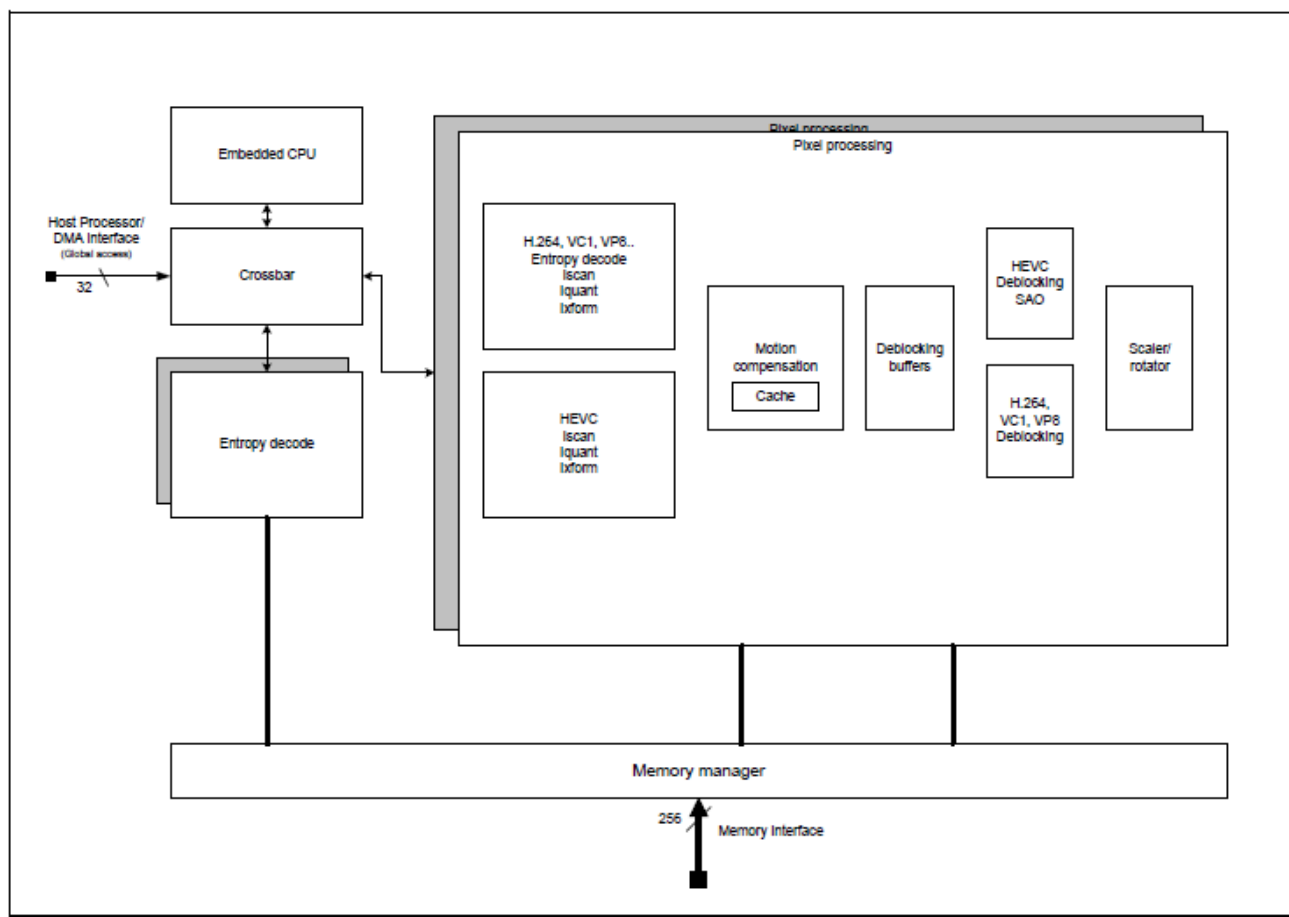


Рисунок 2.42 - Декодер VXD

Производительность одного конвейера:

- 0,6 бита за один такт;
- разрешение 55 тактов на блок 8x8.
- декодер имеет внутренние буферы и кэши, что позволяет иметь задержку обращения в память 384 такта декодера.

Встроенный CPU является микроконтроллером, используемым для декодирования высших уровней, которые включают заголовки на уровне

разбиения и выше. Микроконтроллер использует внутреннюю память SRAM, которая загружается фирменным программным обеспечением перед декодированием.

Декодер поддерживает вращение выходного изображения на 90°, 180° и 270° для 4:0:0 (монохромного) и 4:2:0 8 бит / 10 бит распакованных форматов.

Декодер поддерживает уменьшение размера выходного изображения для видео до $\frac{1}{4}$ оригинального размера в вертикальном и горизонтальном направлениях. Главной целью этого является уменьшение системной пропускной способности при сохранении высокого качества масштабирования. Масштабирование не возможно для JPEG изображений.

Поддерживаются следующие форматы видео память / пиксель:

- 8-битовые форматы:

1) 420PL12YCbCr8 & 420PL12YCrCb8 (этот формат также используется для неподвижной картинки HEVC);

2) 422PL12YCbCr8 & 422PL12YCrCb8 (монохромное изображение поддерживается как часть 422 профилей и использует одну часть этого формата);

- 10-разрядные форматы:

1) 10bit per pel packing:

- 420PL12YCbCr10 & 420PL12YCrCb10

- 422PL12YCbCr10 & 422PL12YCrCb10 (монохромное изображение поддерживается как часть 422 профилей и использует одну часть этого формата);

2) 16bit per pel packing, для 10-битовых данных:

- 420PL12YCbCr16 & 420PL12YCrCb16

- 422PL12YCbCr16 & 422PL12YCrCb16 (монохромное изображение поддерживается как часть 422 профилей и использует одну часть этого формата).

Цветной пространственный формат JPEG проходит через декодер без изменений.

Декодер преобразует пиксельные адреса из нормального растрового адреса в адреса прямоугольников, чтобы уменьшить разрывы страниц.

2.8.5.2 Ускоритель кодирования видео VXE.

Ускоритель кодирования видео VXE позволяет упаковывать видеоизображения в форматы H.265 и H.264. Применяется для организации видеоконференцсвязи, архивного хранения видео и доступа к удаленному рабочему столу (VNC). Поддерживаются разрешения от 128x64 до 4096x2160 пикселей.

Кодер VXE подключается к коммуникационной сети микропроцессора несколькими AXI шинами; в силу большей сложности задачи упаковки видео данный блок создаёт большую нагрузку на память по сравнению с декодером VXD.

Кодер VXE может кодировать AVC (Advanced Video Codec), также называемый как H.264), и HEVC (High Efficiency Video Codec), также называемый как H.265. Кодер VXE может кодировать Ultra-High Definition video (4096x2160). Кодер VXE может считывать серию фрэймов и кодировать их для создания элементарного потока битов. Поддержанные стандарты профилей представлены в таблице 2.25.

Таблица 2.25 – Поддержанные стандарты профилей

Стандарт	Профиль
AVC	Baseline
AVC	Constrained BP
AVC	Main
AVC	High
AVC	MVC
AVC	High 10 P
HEVC	Main (High Tier)
HEVC	Main10 (High Tier)
HEVC	Main Still Picture

Структурная схема кодера VXE представлена на рисунке 2.43. Кодер VXE содержит встроенный процессор, устройство DMA для передачи данных между

Встроенным процессором и хост-памятью, модуль интерфейса памяти и три модуля конвейера кодирования.

В каждом конвейере кодирования фрейм для кодирования может быть считан из хост-памяти или получен через аппаратный интерфейс (Host System Interface). Фрейм может быть уменьшен или изменено его цветовое пространство

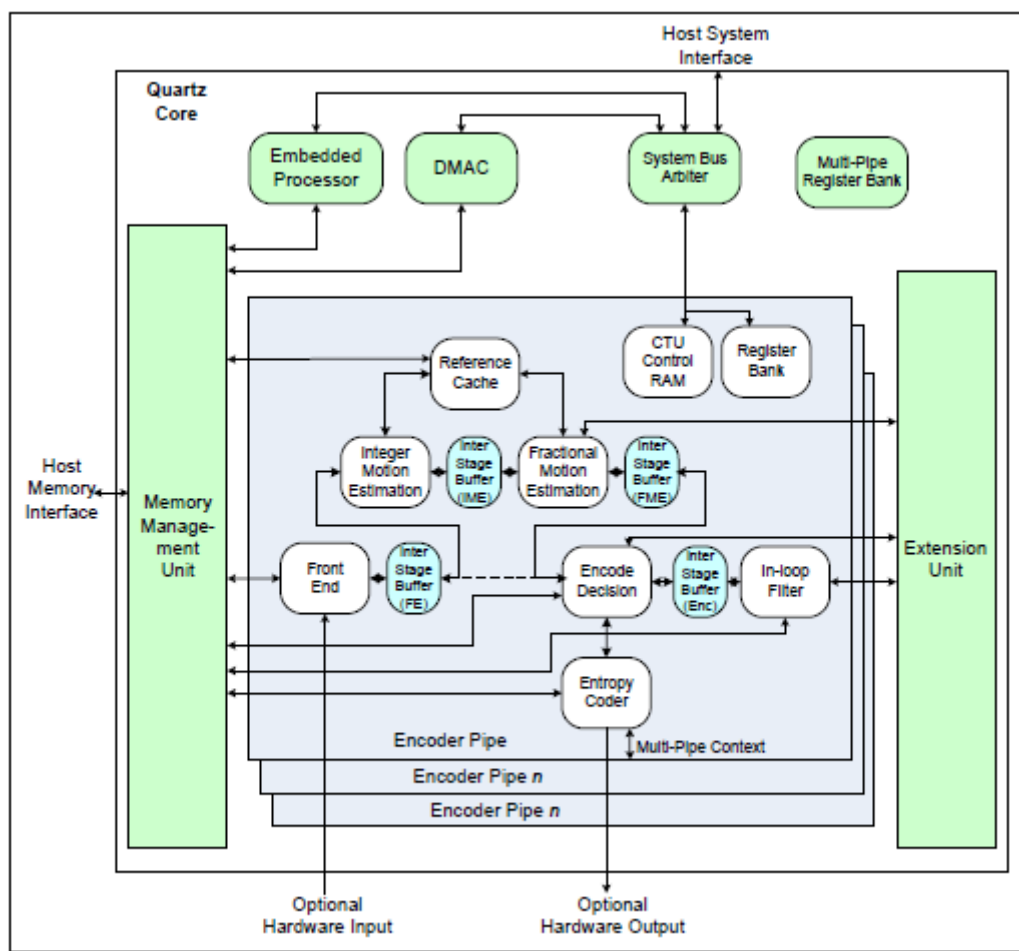


Рисунок 2.43 – Кодер VXE

из входного формата в кодированном формате. В блоке Intra frame кодируемые данные посылаются непосредственно на стадию конвейера Encode Decision. В блоке Inter frame модули Integer Motion Estimation and Fractional Motion Estimation могут определить используемые операции.

Часть конвейера по оценке движения (Reference Cache, Integer Motion Estimation and Fractional Motion Estimation) может работать над двумя картинками.

Encode Pipe (конвейер кодирования) содержит основное оборудование логики кодера. Оно разделено на несколько модулей, которые в большинстве случаев обрабатывают различные СТU (Coded Tree Unit – блоки данных и яркости) на различных стадиях конвейера.

Модуль Front End (входной модуль) может считывать фрейм для кодирования из памяти или получать его через аппаратный интерфейс. Имеется два типа аппаратного интерфейса. Один из них получает пиксели в растровом порядке и затем буферизует данные так, чтобы выдавать блоки размером СТU. Другой интерфейс получает блоки пикселей в порядке, который готов непосредственно для выдачи в конвейер кодирования.

Модуль Front End содержит масштабирующее устройство, которое может уменьшать входной фрейм. Модуль Front End может также выполнять преобразование цветового пространства и выполнять операции с понижением цветовой дискретизацией для изменения входного формата в закодированный формат (например, он может считывать данные из хранилища фреймов ABC (обычно RGB) и выполнить цветовое преобразование в YCbCr перед понижением цветовой дискретизацией к 4:2:0). Модуль Front End может считывать данные из хранилища фреймов 4:4:4 и масштабировать цветность до 4:2:2 или 4:2:0, а также может читать из хранилища фреймов 4:2:2 и масштабировать цветность до 4:2:0.

Модуль Front End может также получать параметры управления уровнем СТU из СТU Control RAM, микропрограммное обеспечение которого можно использовать для управления обработкой поблочно.

Модуль Reference Cache (эталонный кэш) предварительно выбирает части ранее закодированных эталонных кадров и кэширует данные для использования в операциях оценки движения. Reference Cache может кэшировать два эталонных изображения, а затем выводить массив пикселей 16x16 (при любом горизонтальном или вертикальном выравнивании) из обоих эталонных

изображений в каждом такте. Модули Integer Motion Estimation и Fractional Motion Estimation (целочисленная оценка движения и дробная оценка движения) запрашивают пиксели, необходимые им для операций оценки движения (большая часть полосы пропускания кэша используется для оценки целочисленного движения).

The Reference Cache имеет механизмы для ограничения объема используемой полосы пропускания памяти хоста (Reference Cache обычно отвечает за большую часть полосы пропускания памяти хоста).

Модуль Integer Motion Estimation (целочисленная оценка движения IME) выполняет операции оценки движения с разрешением пикселей, чтобы попытаться найти наилучшее совпадение между блоком для кодирования с пикселями из ранее закодированных опорных кадров. Модуль содержит два блока, где каждый блок выполняет поиск другого опорного кадра. Алгоритм поиска использует соседние пространственные и временные векторы движения в качестве отправной точки, а затем выполняет дальнейший поиск вокруг лучших векторов-кандидатов.

Модуль Fractional Motion Estimation (FME - дробная оценка движения) использует лучшие векторы движения, найденные модулем целочисленной оценки движения, а затем выполняет операции оценки движения с субпиксельным разрешением, чтобы попытаться уточнить лучший вектор движения. Модуль FME может корректировать вектор движения IME на $\pm 3/4$ пикселя, а также проверяет, является ли оценка двунаправленного или однонаправленного движения наилучшей. Помимо уточнения векторов движения из алгоритма поиска, модуль FME также оценивает векторы движения с пропуском и объединением, чтобы выбрать лучший режим межкадрового предсказания.

Модуль Encode Decision (решение кодирования) отвечает за выбор наилучшего режима для кодирования макроблока (или CTU). Это включает в себя оценку количества режимов прогнозирования Intra, преобразование остатка из двух лучших режимов Intra плюс соответствующее прогнозирование Inter из FME,

а затем выбор наилучшего режима для текущего размера блока. Четыре отдельных конвейера выполняют эти операции параллельно с четырьмя разными размерами преобразования, а затем выбирается лучший размер блока кодирования. Затем модуль Encode Decision генерирует последовательность параметров кодирования для выбранного наилучшего режима, которые отправляются в Entropy Coder (энтропийный кодер), который реализует кодирование CABAC и CAVLC для генерации битового потока. Для HEVC модуль Encode Decision также отвечает за оценку наилучшего режима адаптивного смещения выборки (SAO) для использования в In-loop Filter (фильтр цикла).

Модуль Entropy Coder получает поток параметров кодирования, которые необходимо преобразовать в сжатый битовый поток. Для HEVC необходимо использовать CABAC, но для AVC это может быть CABAC или CAVLC. Модуль Entropy Coder включает логику для динамической вставки заголовков слайсов и логику для управления буферами, в которые записывается битовый поток.

Entropy Coder также выполняет вывод битового потока через 32-битный Hardware Stream Output Interface (аппаратный интерфейс выходного потока).

Если текущий кодируемый кадр должен быть эталонным кадром, то модуль In-loop Filter (фильтр цикла) может выполнить раскрутку цикла и Sample Adaptive Offset операции фильтра цикла для HEVC. Затем модуль In-loop Filter запишет восстановленный кадр в память хоста (которую Reference Cache может затем прочитать при обработке будущих кадров).

Inter Stage Buffer (буфер между стадиями конвейера) - это настраиваемый модуль, который можно использовать для буферизации данных между стадиями конвейера. Типы хранимых данных различаются в зависимости от стадии конвейера. Буферизация может помочь компенсировать задержку памяти в случае операций оценки движения, ожидающих, пока пиксели будут доступны в Reference Cache, и может помочь сгладить производительность в случае модулей, которым требуется переменное количество времени для обработки CTU. Каждый Inter Stage Buffer может буферизовать от двух до четырех CTU данных. Даже

когда оборудование сконфигурировано для поддержки 64x64 CTU, но в настоящее время обрабатывает 16x16 CTU, в каждом буфере хранится максимум четыре CTU, чтобы ограничить максимальную задержку конвейера от модуля Front-End до модуля In-loop Filter.

Inter Stage Buffer в модуле Encoder Pipe выполняет следующее:

- Inter Stage Buffer (FE) хранит текущие пиксели для кодирования, а также структуру управления CTU, которая определяет положение блока и ряд параметров, которые могут быть установлены с помощью управления скоростью для управления решениями кодирования;

- Inter Stage Buffer (IME) содержит все поля из Inter Stage Buffer (FE), а также структуру вектора движения, которая отражает результат операции целочисленной оценки движения;

- Inter Stage Buffer (FME) содержит в себе большинство полей из Inter Stage Buffer (IME), но заменяет структуру вектора движения IME другой структурой вектора движения FME, которая отражает результат операции дробной оценки движения. В дополнение к этому, в буфере также хранятся пиксели прогнозирования из лучшего режима межкадрового прогнозирования, выбранного FME;

- Inter Stage Buffer (Enc) заменяет текущие пиксели для кодирования реконструированными пикселями до In-loop Filter, которые рассчитал модуль Encode Decision на основе выбранного наилучшего режима кодирования. Кроме того, структура вектора движения FME заменяется закодированной структурой, которая содержит поля из решения о режиме кодирования, которые требуются In-loop Filter, плюс приведенная выше структура, которая требуется In-loop Filter для завершения фильтрации нижнего края ряда CTU выше.

CTU Control RAM (управляющая память CTU) имеет три интерфейса. System Bus Arbiter (арбитр системной шины) может читать и писать в RAM, где микропрограммное обеспечение может записывать структуру каждого блока CTU для управления некоторыми решениями кодирования и может считывать информацию об окончательных решениях кодирования. Модуль Front-End имеет

интерфейс с этой RAM, где он может считывать структуру каждого STU и добавлять информацию в структуру управления STU, отправляемую по конвейеру кодирования. Модуль Encode Decision имеет интерфейс с этой RAM, где он может записывать структуру для каждого STU с информацией о выбранном режиме кодирования.

Encoder Pipe Register Bank (банк регистров конвейера кодирования) содержит все регистры управления и состояния для Encoder Pipe. Большинство регистров управления предназначены для однократной настройки перед началом обработки кадра / слайса, а затем остаются статичными для всего кадра / слайса. Регистр управления 'kick' (пинок) определяет, сколько последовательных STU будет передавать вниз по конвейеру кодирования модуль Front End, где некоторые регистры управления скоростью могут быть настроены для каждого 'kick'. Некоторые регистры состояния накапливаются внутри STU в 'kick' а затем регистр статуса, видимый для микропрограммного обеспечения (или хоста), обновляется в начале следующего 'kick' (или при установке программного обеспечения для сброса регистра).

2.9 Система синхронизации

2.9.1 Система синхронизации позволяет устанавливать оптимальную рабочую частоту устройств микропроцессора с учетом качества его изготовления, условий эксплуатации и текущей рабочей нагрузки. Большинство основных блоков микропроцессора имеют отдельные схемы формирования рабочих синхросигналов для установки оптимальных рабочих частот блоков при аттестации микропроцессора после его изготовления и их оперативного изменения в целях поддержания микросхемы в заданных диапазонах температуры и энергопотребления.

Микропроцессор Эльбрус-2С3 имеет пять входных синхросигналов:

- два синхросигнала CLK_REF_100M_TOP и CLK_REF_100M_BOT с частотой $(100 \pm 0,01)$ МГц используются в качестве опорных синхросигналов для

формирования рабочих частот почти всех основных блоков микропроцессора. Синхросигнал CLK_REF_100M_BOT используется для формирования рабочих частот контроллера EIOH, синхросигнал CLK_REF_100M_TOP формирует рабочие частоты остального оборудования микропроцессора;

- синхросигнал CLK_REF_156M с частотой $(156,25 \pm 0,015625)$ МГц используется в качестве опорного синхросигнала для одного блока E12G PHY x2 в том случае, если он работает в режиме Ethernet;

- синхросигнал SUS_CLK_14M с частотой $(14,31818 \pm 0,001431818)$ МГц используется в качестве рабочего синхросигнала контроллера блока Efuse, регистров распределенной системы распространения и применения настроек из блока Efuse и таймеров SPMC-контроллера (System Power Management Controller), который управляет включением основных напряжений питания микропроцессора;

- синхросигнал TCK контроллера JTAG может иметь частоту до 50 МГц.

В таблице 2.26 приведен перечень основных внутренних сигналов синхронизации микропроцессора Эльбрус-2С3. Структурная схема системы синхронизации микропроцессора Эльбрус-2С3 представлена на рисунке 2.44.

Таблица 2.26 - Используемые синхросигналы

Имя сигнала	Диапазон частот, значение по умолчанию (МГц)	Динамическое изменение частоты
clk_ref_100m_top	100	-
clk_ref_100m_bot	100	-
clk_ref_156m	156,25	-
sus_clk_14m	14,31818	-
TCK	Не более 50	
clock_core[0]	250-2000, 2000	Настройки меняются на ходу через pcs
clock_gpu	250-800, 800	Настройки меняются на ходу через pcs
clock_gpu (bfs_bypass)	2000/3=667 (2000/2=1000)	Настройкой делителя на 2 или 3 управляет pcs
clock_dec	250-667, 667	Настройки меняются на ходу через pcs

Имя сигнала	Диапазон частот, значение по умолчанию (МГц)	Динамическое изменение частоты
clock_dec (bfs_bypass)	2000/3=667, 2000/4=500	Настройкой делителя на 3 или 4 управляет pcs
clock_core[1]	250-2000, 2000	Настройки меняются на ходу через pcs
clock_mga	500-1000, 1000	Настройки меняются на ходу через pcs
clock_mga (bfs_bypass)	2000/2=1000, 2000/3=667	Настройкой делителя на 2 или 3 управляет pcs
clock_enc	250-667, 667	Настройки ccs_bfs_c[1] меняются на ходу через pcs
clock_enc (bfs_bypass)	2000/3=667, 2000/4=500	Настройкой делителя на 3 или 4 управляет pcs
clock_mc[0]	200-800, 800	Настройки устанавливаются при включении питания
clock_mc[1]	200-800, 800	Настройки устанавливаются при включении питания
clock_oci	400-1600, 1600	Настройки меняются на ходу через pcs
clock_ec1	800-1600, 1600	-
clock_ec2	400-800, 800	-
clock_usb_ge	480	-
clock_48m	48	-
clock_sp	4,897959184	-
clock_dc[3:0]	25-1900	Управление PLL программное из hdmi_build

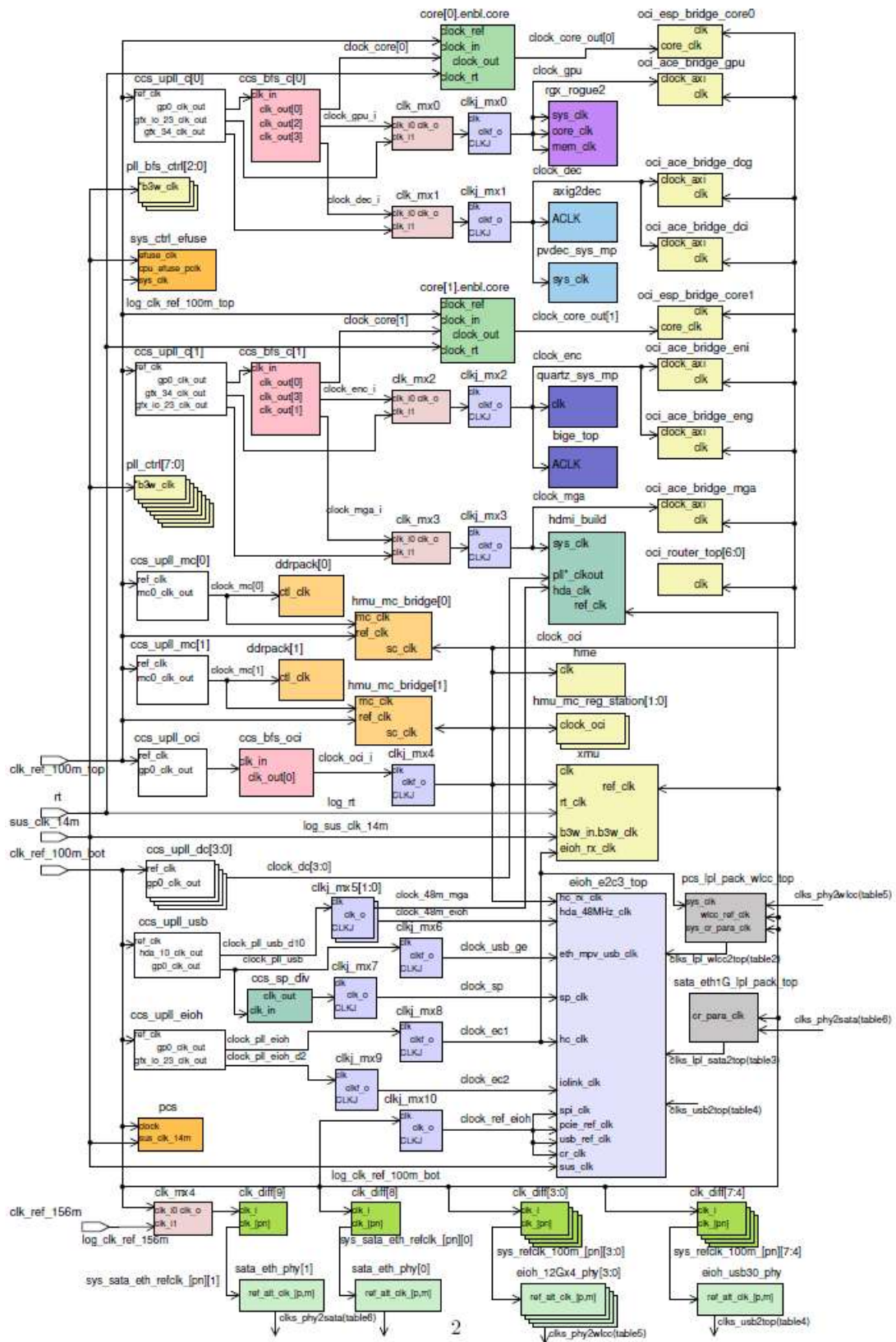


Рисунок 2.44 – Система синхронизации микропроцессора Эльбрус-2С3

3 Встроенные средства энергосбережения, контроля и восстановления работоспособности

3.1 Средства энергосбережения

3.1.1 В микропроцессоре Эльбрус-2С3 реализованы средства экономии потребляемой мощности в случае постоянной или временной невостробованности аппаратных ресурсов. Эти средства включают:

- для процессорных ядер Core: регулировка частоты и полное отключение синхронизации;
- для коммутационной сети ОСІ: регулировка частоты синхронизации;
- для каналов PCI Express и ввода-вывода: регулировка частоты синхронизации, перевод в малопотребляющий режим и полное отключение синхронизации, аппаратное отключение синхронизации, если канал не подключен;
- для каналов обращения в оперативную память: регулировка частоты и полное отключение синхронизации, перевод в малопотребляющий режим.

3.2 Контроллер управления энергосбережением PCS

3.2.1 Общие сведения.

Контроллер управления энергосбережением PCS (Power Control System) содержит средства управления энергопотреблением наиболее энергоемкой центральной части микропроцессора – процессорных ядер Core и коммутационной сети ОСІ. Эти средства используются в целях экономии энергопотребления при неполной рабочей нагрузке и для защиты от перегрева при чрезмерной рабочей нагрузке микропроцессора. В каждом процессорном ядре можно остановить дешифрацию команд, снизить частоту синхронизации или отключить синхрони-

зацию полностью. В коммутационной сети ОСИ можно снизить частоту синхронизации.

Устройство PCS содержит также средства контроля и управления температурным режимом кристалла микросхемы с помощью встроенных термодатчиков и вентиляторов охлаждения.

В состав устройства PCS входят:

- контроллер управления частотой синхронизации и общей политикой энергосбережения PMC (Power Manager Controller);
- блок датчиков PVT температуры, напряжения питания и качества изготовления (TS, VM, PD);
- контроллер датчиков PVT;
- два контроллера канала вентиляторов охлаждения FAN;
- блок системных событий;
- контроллер интерфейса I2C-slave, предоставляющий "внешнему system/power менеджеру" доступ к контроллерам FAN и PMC.

Контроллер PMC содержит регистры настройки частот системы синхронизации центральной части микропроцессора и реализует восемь режимов управления частотами синхронизации с разной степенью автоматизации отдельно для каждого устройства процессора.

Контроллер датчиков PVT:

- опрашивает датчики температуры TS, получая их показания во внутреннем представлении АЦП в режимах автоматического опроса показаний раз в 1,3 мс или однократного опроса;
- преобразует показания датчиков TS в градусы Цельсия;
- вычисляет максимум среди показаний всех термосенсоров;
- опрашивает датчики VM и PD под программным управлением.

Контроллер канала вентиляторов FAN обеспечивает:

- режим прямого программного управления (драйвер ОС или внешний менеджер задают темп вращения вентиляторов);

- режим автоматического управления по таблице с температурными порогами (драйвер ОС или внешний менеджер настраивают таблицу, а аппаратура изменяет скорость вращения вентиляторов в зависимости от текущей температуры).

Блок системных событий:

- регистрирует системные события;
- выдает прерывания при появлении интересующего события;
- по заданным событиям выполняет автоматическую аппаратную реакцию (снижение энергопотребления, перевод вентиляторов охлаждения на максимальные обороты и т.д.).

Контроллер интерфейса I2C-slave предоставляет внешний доступ к управлению частотой синхронизации и контроллерам каналов вентиляторов.

Краткие характеристики контроллера:

- рабочая частота 14,3 МГц;
- адрес устройства на шине - 7'b1001100;
- скорость передачи данных до 1 Мбит/с.

3.2.2 Структурная схема.

Система управления энергосбережением PCS, представленная на рисунке 3.1, состоит из следующих модулей:

- модуль PCS_TOP – включает в себя несколько других подмодулей;
- модуль PCS_CFG_PUB – коммутатор доступа к управляющим регистрам PMS и PVT контроллеров;
- модуль I2C DEV (SLAVE) – служебный блок для внешнего управления конфигурацией вентиляторов и регистрами PMS;
- модуль PCS_JTAG_REG – служебный блок для обращения к управляющим регистрам через JTAG;
- два модуля PWM Cotrloller – контроллеры вентиляторов охлаждения;

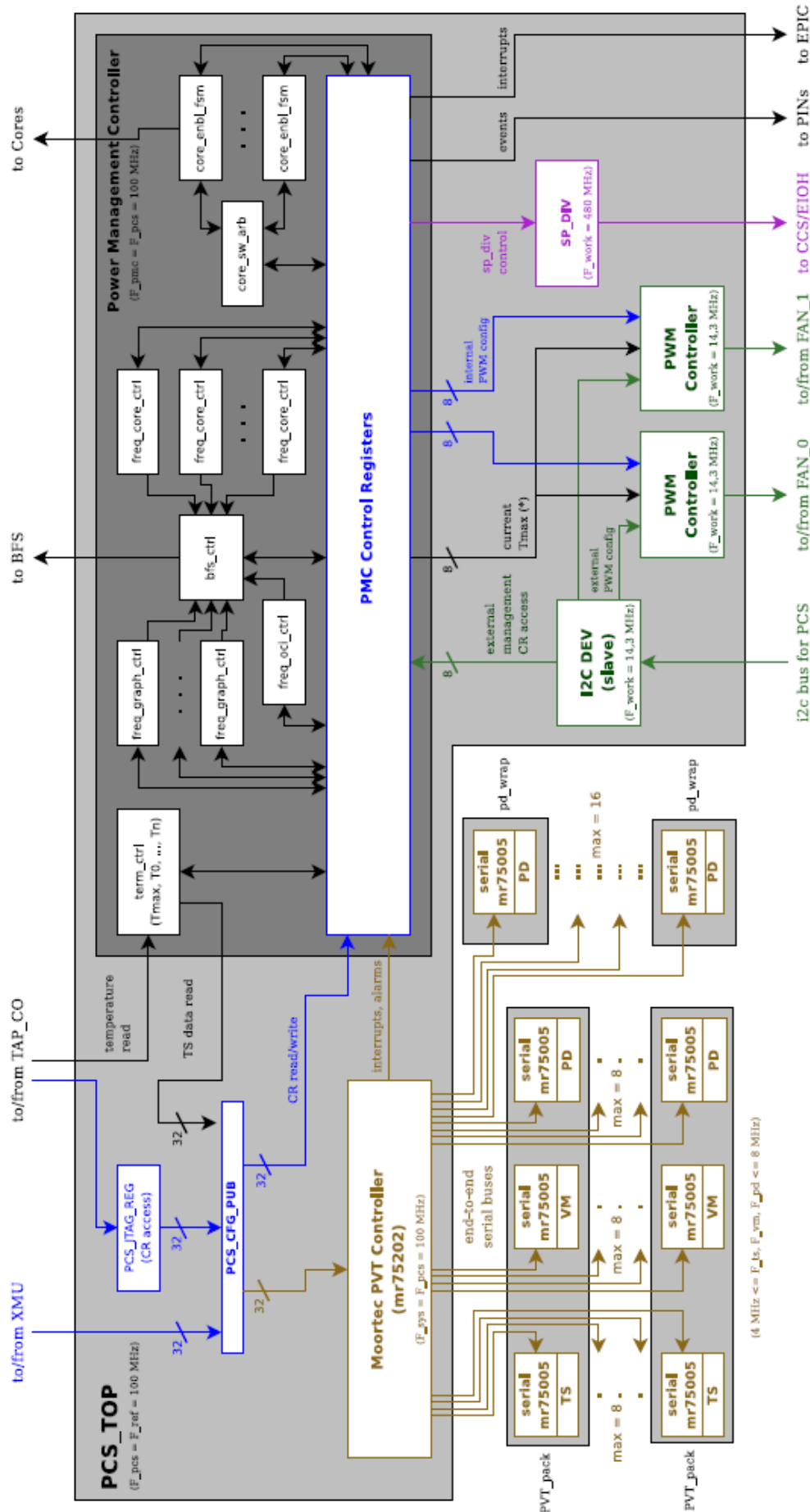


Рисунок 3.1 - Система управления энергосбережением PCS

- модуль SP_DIV – служебный блок для управления делителем частоты Serial Port (для нужд EIOH);
- модуль Power Management Controller – управление энергосбережением/энергопотреблением микропроцессора (PMC);
- модуль PVT Controller – контроллер датчиков PVT;
- два модуля ccs_clk_mux_scan – служебные мультиплексоры для подключения сигнала синхронизации JTAG TCK вместо рабочего синхросигнала;
- восемь модулей PVT_PACK – обертка для PVT-датчиков, включает несколько других подмодулей:
 - три модуля serial mr75005 – адаптеры последовательного интерфейса для каждого PVT-датчика;
 - модуль TS – термосенсор;
 - модуль VM – монитор напряжения;
 - модуль PD – детектор техпроцесса;
- 16 модулей pd_wgap – обертка для PD-датчиков, включает несколько других подмодулей:
 - модуль serial mr75005 – адаптер последовательного интерфейса для каждого PD-датчика;
 - модуль PD – детектор техпроцесса.

3.2.3 Контроллер и датчики PVT.

Контроллер датчиков PVT подключен к шести температурным сенсорам TS, шести мониторам напряжения VM и 22 детекторам процесса PD. Контроллер PVT имеет параллельный интерфейс для обмена с системой и содержит адаптеры последовательной шины для подключения всех датчиков. Структурная схема контроллера PVT и его датчиков представлена на рисунке 3.2.

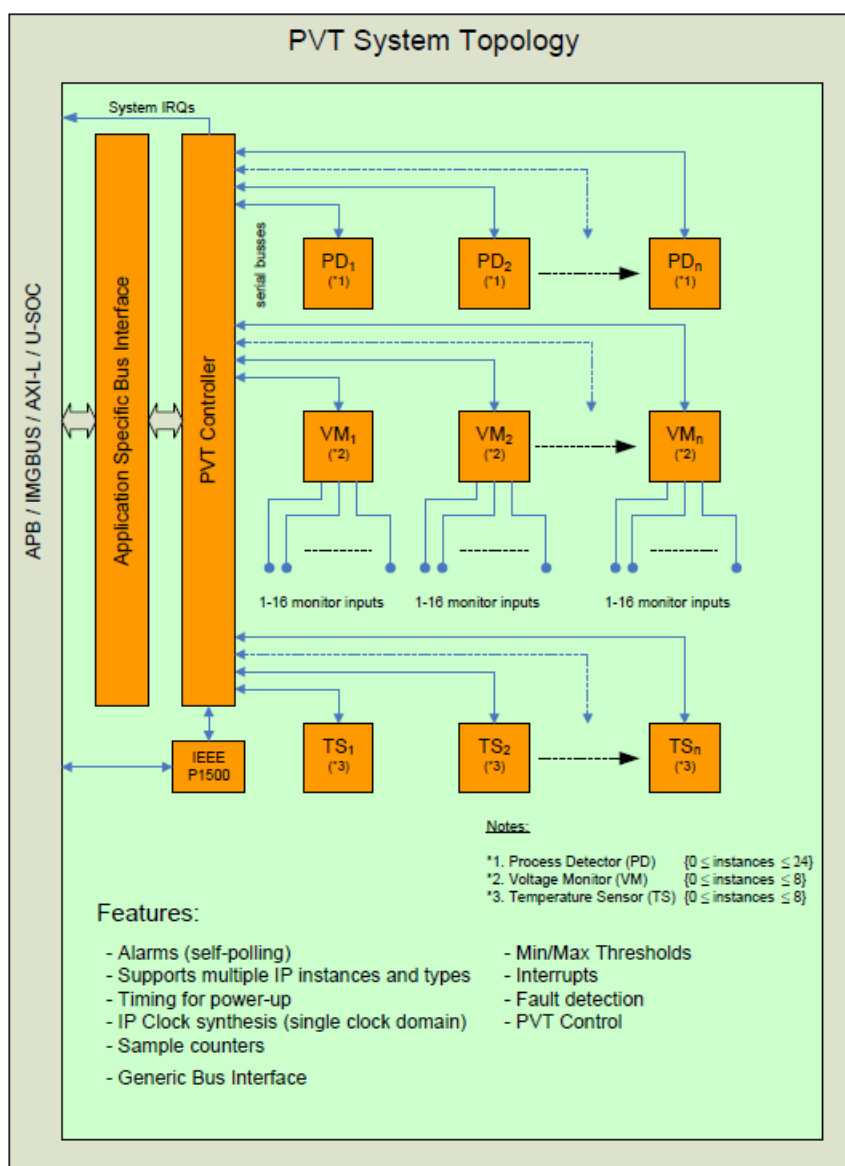


Рисунок 3.2 – Контроллер и датчики PVT

Температурный сенсор TS измеряет с высокой точностью температуру кристалла в заданной точке. Для увеличения точности измерений сенсор TS может быть откалиброван, причем для этого не требуется знать температуру кристалла. В типовое применение сенсора TS входит оптимизация частоты синхронизации, управление питанием, управление температурой и характеристика микросхем.

Основные характеристики:

- точность без калибровки.....±3,0 °C;
- точность с калибровкой.....±1,0 °C;
- код результата измерения12-разрядный (10 или 8 разрядов при меньшей точности).

Монитор напряжения VM предназначен для измерения напряжений питания 0,8 В в ядре кристалла. Для увеличения точности монитор VM может быть откалиброван, причем для этого не требуется знать температуру кристалла. В типовое применение монитора VM входит измерение напряжения, измерение падения напряжения между различными точками измерений, управление питанием.

Основные характеристики:

- измерения.....10 точек;
- точность без калибровки..... ± 1 %;
- точность с калибровкой..... $\pm 0,6$ %;
- код результата измерения.....14-разрядный.

Детектор процесса PD позволяет разработчикам интегральных схем обнаруживать изменения процесса, вызванные изменчивостью процесса изготовления, и отличия в улучшенных узлах цифровых КМОП устройств. Детектор PD измеряет частоту генерации сигнала в замкнутой цепи из набора инверторов.

Детектор PD может использоваться для непрерывной оптимизации управления напряжением питания и частотой, мониторинга вариаций производства, измерения задержки вентиля, анализа критических путей, анализа критических напряжений и мониторинга старения кристалла.

Основные характеристики:

- пять встроенных кольцевых цепей задержки разной длины;
- возможность подключения внешней кольцевой цепи задержки;
- 12-разрядный код результата измерения.

3.3 Тестирование и разбраковка

3.3.1 Для отбраковки дефектных экземпляров микропроцессора предусмотрен режим скан-тестирования, в котором большинство триггеров микропроцессора логически объединяется в цепочку, что позволяет через канал JTAG загружать произвольные тестовые последовательности, выполнять такт параллельного срабатывания и выдавать результат. Этот метод предназначен для

проверки триггеров и комбинационной логики. Контроллеры канала JTAG размещены в каждом процессорном ядре и еще один общий контроллер канала JTAG предназначен для доступа к цепям канирования остальных блоков микропроцессора и контроллерам канала JTAG процессорных ядер.

Канал JTAG также используется для отладки микропроцессора. Через него управляется встроенная тестовая логика, позволяющая считывать внутреннее состояние процессорных ядер и внеядерных устройств, выполнять произвольные команды, обращаться в оперативную память и т.д.

3.4 Контроль работоспособности блоков кэш-памяти перед началом работы

3.4.1 Большинство блоков встроенной памяти процессорных ядер и внеядерных устройств снабжено схемами автоматической самопроверки BIST, которые запускаются либо аппаратно при включении процессора, либо командой через диагностический канал JTAG.

3.5 Исключение дефектных ячеек блоков кэш-памяти при сохранении общей работоспособности кэш-памяти

3.5.1 Для крупных блоков памяти, где вероятность дефекта повышена, предусмотрены средства исключения или замены дефектных ячеек на запасные — это позволяет повысить процент выхода годных микросхем.

Так в кэше L1 могут быть исключены 32-байтовые блоки с неисправными ячейками. После сигнала сброса выполняется BIST тестирование блоков памяти кэша и для неисправных блоков устанавливается бит запрета использования dontuse. Этот бит не позволяет механизму назначения использовать неисправный блок.

В кэше L2 предусмотрены дополнительные два разряда в каждом блоке 8 Кбайт, которыми можно заменить неисправные разряды. После сигнала сброса

выполняется BIST тестирование памяти кэша и в неисправных блоках неисправные разряды автоматически заменяются на резервные.

3.6 Обнаружение и исправление в критических местах сбоев кэш-памяти и блоков оперативной памяти в процессе штатной работы

3.6.1 Помимо «постоянных» фабричных дефектов, блоки кэш-памяти и оперативной памяти могут быть подвержены временным сбоям, обусловленным воздействием реликтового излучения. Для обнаружения таких сбоев информация во всех блоках кэш-памятей и оперативной памяти сопровождается дополнительными контрольными разрядами.

В кэше L1 теги и каждый байт данных имеют дополнительные разряды для контроля по четности.

В кэше L2 теги и разряды механизма старения имеют дополнительные разряды для контроля по четности, а каждое 8-байтовое слово данных снабжено корректирующим кодом ЕСС, позволяющим корректировать одиночную ошибку и обнаруживать двойную.

Оперативная память снабжена средствами контроля корректности хранимой информации: каждые 32 байта данных сопровождаются дополнительными разрядами, содержащими код ЕСС, что позволяет корректировать одиночную ошибку и обнаруживать двойную.

Обнаружение ошибки при контроле по четности и двойной ошибки в коде ЕСС вызывает прерывание и обращение к операционной системе с целью устранения последствий ошибки.

3.7 Логические анализаторы

3.7.1 Для исследования функционирования микропроцессора в реальном времени предусмотрены встроенные логические анализаторы, к которым подключено большое количество внутренних сигналов различных устройств.

Логические анализаторы имеют встроенную память для записи трассы значений сигналов микропроцессора, подключенных к анализатору. Запись может осуществляться как в режиме реального времени, так и с паузами - это определяется статическими (программируемыми) режимами и динамическими состояниями анализатора. В логических анализаторах предусмотрены программируемые средства анализа входного потока и выработки событий, которые используются либо для изменения состояния анализатора, либо для выдачи ввне сигнала для остановки других анализаторов (включая себя) и микропроцессора. Логический анализатор программируется и запускается через процессорный канал JTAG.

4 Резерв

5 Описание интерфейсов микропроцессора

5.1 Напряжения питания периферийных элементов и условные обозначения типов выводов

5.1.1 Напряжения питания периферийных элементов приведены в таблице 5.1, условные обозначения, условные обозначения типов выводов – в таблице 5.2.

Таблица 5.1 - Напряжения питания периферийных элементов

Название	Значение, В
Напряжение питания периферийных элементов типа SSTL_12	1,2
Напряжение питания периферийных элементов всех остальных типов, кроме USB	1,8
Напряжение питания периферийных элементов типа USB	3,3

Таблица 5.2 – Условные обозначения типов выводов

Обозначение	Описание
in	вход
out	выход
inout	двухнаправленный вход-выход
pullup	внутренняя утяжка на питание
pulldn	внутренняя утяжка на землю
analog	аналоговый сигнал
opendrain	выход с открытым коллектором
Rterm	внешняя (на корпусе) утяжка на землю

5.2 Системные сигналы

5.2.1 Системные сигналы приведены в таблице 5.3

Таблица 5.3 - Системные сигналы

Название	Вход/ выход	Тип	Назначение
AC_POWER_PSNT	in	CMOS_18	Показывает, какого типа источник питания: 0 – аккумулятор; 1 – сеть переменного тока. Домен питания- «suspend» Асинхронный сигнал чувствительный к помехам более 100 нс. Для логики есть внутренняя защита от глитчей ~290нс.
ATE_MODE	in	CMOS_18 pulldn	Управление отладочным оборудованием: признак установки на тестирующий стенд и возможное отсутствие штатного входного синхросигнала. Активный уровень - высокий
ATN_SUS_	out	CMOS_18	Сигнал о том, что скоро будут отключены питания; может быть использован для управления режимом retention. Активный уровень - низкий. Домен питания - "suspend".
BATLOW_	in	CMOS_18 pullup	Сигнал о низком уровне заряда аккумулятора. Активный уровень - низкий. Домен питания - "suspend" Асинхронный сигнал чувствительный к помехам более 100 нс. Для логики есть внутренняя защита от глитчей ~290нс.
CLK_REF_100M_BOT_P CLK_REF_100M_BOT_N	in	LVDS_18 Rterm	Опорный синхросигнал. Подается парафазной линией.

Название	Вход/ выход	Тип	Назначение
			Рабочая частота 100 МГц. Для этого синхросигнала у процессора имеется встроенный (установленный на подложке корпуса) согласующий резистор с номиналом 100 Ом и мощностью рассеяния 0,063 Вт
CLK_REF_100M_TOP_P CLK_REF_100M_TOP_N	in	LVDS_18 Rterm	Опорный синхросигнал. Подается парафазной линией. Рабочая частота 100 МГц. Для этого синхросигнала у процессора имеется встроенный (установленный на подложке корпуса) согласующий резистор с номиналом 100 Ом и мощностью рассеяния 0,063 Вт
CLK_REF_156M_P CLK_REF_156M_N	in	LVDS_18 Rterm	Опорный синхросигнал. Подается парафазной линией. Рабочая частота 156,25 МГц. Для этого синхросигнала у процессора имеется встроенный (установленный на подложке корпуса) согласующий резистор с номиналом 100 Ом и мощностью рассеяния 0,063 Вт. Этот опорный синхросигнал используется только при условии, что на вход процессора SATAETH_CONFIG подается константа 1. Иначе допускается не подавать этот синхросигнал, и тогда эти контакты следует соединить с GND
CORE_ENBL[1:0]	in	CMOS_18 pullup	Битовая маска включения ядер процессора: высокий уровень - ядро включено низкий уровень - ядро выключено Номера разрядов соответствуют

Название	Вход/ выход	Тип	Назначение
			уомерам ядер
CPU_DISABLE_SOFT_RST_	in	CMOS_18 pullup	Запрет выработки программного ресета; может использоваться в случае наличия КПИ-2. Активный уровень - низкий
CPU_HRST_IN_	in	CMOS_18 pullup	Сигнал "жесткого" обнуления процессо- ров, кроме bsp. Активный уровень - низкий. Домен питания - "suspend"
CPU_SRST_IN_	in	CMOS_18 pullup	Сигнал "мягкого" обнуления процессоров, кроме bsp. Активный уровень - низкий. Домен питания - "suspend"
DBG_RST_DSBL	in	CMOS_18 pulldn	Управление отладочным оборудованием: запрет reset встроенных логических анали- заторов. Активный уровень - высокий
DBG_STOP	in	CMOS_18 pulldn	Управление отладочным оборудованием: остановка процессора после reset. Активный уровень - высокий
DDR_PWROK	in	CMOS_18	Признак нахождения DDRPHY в состоя- нии холодного сброса при включении питания/признак входа DDRPHY в режим энергосбережения S3. Соединяется на пе- чатной плате с сигналом ATN_SUS. Активный уровень - низкий. Домен питания - "suspend"
EFUSE_MODE[1:0]	in	CMOS_18 pulldn	Признак тестового режима для оборудования, связанного с eFUSE. Задаёт один из 4 режимов запуска процес- сора: 0 - полевой режим;

Название	Вход/ выход	Тип	Назначение
			1 - обход efuse; 2 - проверка efuse; 3 - программирование efuse
FREQ_MODE[1:0]	in	CMOS_18 pulldn	Ограничение частот процессорных ядер и общей кэш-памяти третьего уровня (предназначено для машин, в которых требуется ограничить потребляемую процессором мощность): 0 (00) - ограничения частот нет; 1 (01) - самое слабое ограничение; 2 (10) - более сильное ограничение; 3 (11) - самое сильное ограничение
PWR_BTN_	in	CMOS_18 pulldn	Сигнал о нажатии кнопки включения питания. Активный уровень - низкий. Домен питания - "suspend" Асинхронный сигнал чувствительный к помехам. Есть внутренняя пересинхронизация.
RFU[5:0]	inout	CMOS_18	Резервные контакты
RT	in	CMOS_18 pulldn	Синхросигнал точного времени 1 Гц. Используется для подстройки счетчика тактов опорной частоты
SLEEP_S3_	out	CMOS_18	Управление питанием в режиме энергосбережения - уровни S3 - S5. Активный уровень - низкий. Домен питания - "suspend"
SLEEP_S4_	out	CMOS_18	Управление питанием в режиме энергосбережения - уровни S4 - S5. Активный уровень - низкий. Домен питания - "suspend"

Название	Вход/ выход	Тип	Назначение
SLEEP_S5_	out	CMOS_18	Управление питанием в режиме энерго- сбережения - уровень S5. Активный уровень - низкий. Домен питания - "suspend"
SUS_CLK_14M	in	CMOS_18	Синхросигнал логики домена "неотключа- емого" ("suspend") питания; рабочая частота 14,31818 МГц. Домен питания - "suspend"
SUS_PWROK	in	CMOS_18	Сигнал о завершении включения на плате "неотключаемого" ("suspend") питания. Активный уровень - высокий. Домен питания - "suspend" Асинхронный сигнал чувствительный к помехам. Примечание - При отключении штатного питания (sys_pwrok) кнопкой на плате должно также отключаться suspend- питание (не относится к перезагрузке); наличие suspend-питания должно отобра- жаться на плате светодиодом.
SYS_GPIO[15:0]	inout	CMOS_18 pullup	Системные сигналы GPIO. [12:0] – GPIO[12:0] [15:13] –GPIO[15:13] или MPV[2:0], В зависимости от флага в контроллере GPIO/MPV
SYS_KPI2BOOT_ENA	in	CMOS_18 pulldn	Определяет направление считывания про- грамма начальной загрузки (boot); 0 - доступ осуществляется через встроенный канал процессора; 1 - доступ осуществляется через микросхему КПИ-2, подключенную к процессору

Название	Вход/ выход	Тип	Назначение
SYS_PLTRST_	out	CMOS_18	Сигнал обнуления всех элементов вычислительной системы, за исключением процессоров. Активный уровень - низкий. Домен питания - "suspend"
SYS_PWROK	in	CMOS_18	Сигнал о завершении включения на плате штатного питания. Активный уровень - высокий. Домен питания - "suspend" Асинхронный сигнал чувствительный к помехам. Для логики есть внутренняя защита от глитчей ~290нс, так же есть управление изоляторами srmc блока, которое идет напрямую.
TCK	in	CMOS_18 pulldn	Интерфейс JTAG. Сигнал синхронизации. На сигнале TCK установлена внутренняя утяжка к низкому уровню
TDI	in	CMOS_18 pullup	Интерфейс JTAG. Вход цепи сканирования
TDO	out	CMOS_18	Интерфейс JTAG. Выход цепи сканирования
TMS	in	CMOS_18 pullup	Интерфейс JTAG. Включение режима JTAG. Примечание - Перед подключением JTAG-контроллера необходимо убедиться, что он работает в таком режиме, когда уровни сигналов не превышают 1,8 В
TRIG_IN	in	CMOS_18 pulldn	Сигнал остановки от внешнего логического анализатора. Активный уровень - высокий
TRIG_OUT	out	CMOS_18	Сигнал остановки для внешнего

Название	Вход/ выход	Тип	Назначение																																				
		pulldn	логического анализатора. Активный уровень - высокий																																				
WAKE_UP_	in	CMOS_18 pullup	Сигнал о приходе события для выхода из энергосберегающего режима. Активный уровень - низкий. Домен питания - "suspend"																																				
WLCC_PRESETS[2:0]	in	CMOS_18 pulldn	Выбор настроек для канала WLCC. Одновременно выбирается битовая скорость канала BR (bitrate) и настройка коррекции АЧХ передатчиков DE (TX de-emphasis). При неудачном установлении соединения возможно автоматическое переключение на резервную скорость RBR (reserve BR), $RBR = BR/2$ для всех настроек: <table border="1" data-bbox="874 1093 1318 1653"> <thead> <tr> <th>Код [2:0]</th> <th>BR, Гбит/с</th> <th>RBR, Гбит/с</th> <th>DE, дБ</th> </tr> </thead> <tbody> <tr> <td>0 (000)</td> <td>5</td> <td>2,5</td> <td>3,7</td> </tr> <tr> <td>1 (001)</td> <td>6</td> <td>3</td> <td>3,7</td> </tr> <tr> <td>2 (010)</td> <td>5</td> <td>2,5</td> <td>6,0</td> </tr> <tr> <td>3 (011)</td> <td>6</td> <td>3</td> <td>6,0</td> </tr> <tr> <td>4 (100)</td> <td>2,5</td> <td>1,25</td> <td>0</td> </tr> <tr> <td>5 (101)</td> <td>4</td> <td>2</td> <td>3,7</td> </tr> <tr> <td>6 (110)</td> <td>4</td> <td>2</td> <td>6,0</td> </tr> <tr> <td>7 (111)</td> <td>8</td> <td>4</td> <td>3,7</td> </tr> </tbody> </table>	Код [2:0]	BR, Гбит/с	RBR, Гбит/с	DE, дБ	0 (000)	5	2,5	3,7	1 (001)	6	3	3,7	2 (010)	5	2,5	6,0	3 (011)	6	3	6,0	4 (100)	2,5	1,25	0	5 (101)	4	2	3,7	6 (110)	4	2	6,0	7 (111)	8	4	3,7
Код [2:0]	BR, Гбит/с	RBR, Гбит/с	DE, дБ																																				
0 (000)	5	2,5	3,7																																				
1 (001)	6	3	3,7																																				
2 (010)	5	2,5	6,0																																				
3 (011)	6	3	6,0																																				
4 (100)	2,5	1,25	0																																				
5 (101)	4	2	3,7																																				
6 (110)	4	2	6,0																																				
7 (111)	8	4	3,7																																				

5.3 Каналы оперативной памяти DDR4

5.3.1 Микропроцессор имеет два канала памяти: 0, 1. Они имеют одинаковые интерфейсы, и в таблице 5.4 символ j представляет одну из этих цифр.

Таблица 5.4 – Сигналы каналов памяти

Название	Вход/ выход	Тип	Назначение
MCj_A[17:0]	out	SSTL_12	Адрес строки / Адрес столбца / команда
MCj_ACT_N	out	SSTL_12	Признак команды Activate
MCj_ALERT_N	inout	SSTL_12	Ошибка четности адресной шины или CRC данных, либо дополнительный тестовый вывод цифровых сигналов PHY. Указания по подключению: - при размещении микросхем памяти на материнской плате утягивающий резистор 50 Ом к верхнему уровню (pull-up) должен быть установлен в конце линии Alert_n после последней микросхемы SDRAM; - в случае использования модулей DIMM утягивающий резистор 50 Ом к верхнему уровню (pull-up) должен быть установлен на линии Alert_n в непосредственной близости к микропроцессору
MCj_BA[1:0]	out	SSTL_12	Адрес логического банка в группе банков
MCj_BG[1:0]	out	SSTL_12	Номер группы банков
MCj_CID[2:0]	out	SSTL_12	Номер логического ранка (стека) в модулях памяти типа 3DS. Примечание - К выводу CS2_n/C0 модуля памяти должен быть подведён вывод MCj_CID[0] процессора, к выводу CS3_n/C1 – вывод MCj_CID[1] процессора
MCj_CK_P[3:0] MCj_CK_N[3:0]	out	SSTL_12	Синхросигналы. Передаются парафазными линиями

Название	Вход/ выход	Тип	Назначение
MCj_CKE[3:0]	out	SSTL_12	Clock enable. Высокий уровень активирует ИО-буферы и синхроимпульс в модуле памяти. Сигнал используется при инициализации модулей памяти и их переводе в ждущее состояние
MCj_CS_N[3:0]	out	SSTL_12	Позиционный адрес физического банка (Chip select). Сигнал выбирает один из четырёх физических банков при отсылке команды в память. Активный уровень - низкий. Примечание - К выводу CS2_n/C0 модуля памяти должен быть подведён вывод MCj_CID[0] процессора, к CS3_n/C1 - MCj_CID[1]
MCj_DQ[71:0]	inout	SSTL_12	Двунаправленная шина данных
MCj_DQS_P[17:0] MCj_DQS_N[17:0]	inout	SSTL_12	Дифференциальные линии строба данных. Единственно допустимая карта соединений на материнской плате: - пины MCj_DQS_P[8:0] - пины DQS[8:0] модуля памяти; - пины MCj_DQS_P[17:9] - пины DQS[17:9]/TDQS[17:9]/DM[8:0]/DBI[8:0] модуля памяти; - пины MCj_DQS_N[8:0] - пины DQS#[8:0] модуля памяти; - пины MCj_DQS_N[17:9] - пины DQS#[17:9]/TDQS#[17:9] модуля памяти
MCj_MTEST	out	SSTL_12	Основной тестовый вывод цифровых сигналов PHY. Сигналы MCj_MTEST необходимо вывести на тестовые точки. Эти сигналы требуют согласования (100 Ом на GND) (100 Ом на 1,2 В). Рядом с каждой тестовой точкой установить тестовую точку GND. При необходимости можно эти сигналы оставить не

Название	Вход/ выход	Тип	Назначение
			<p>подключенными.</p> <p>Проводники, идущие к тестовым точкам, должны быть защищены от помех (экранированные GND).</p> <p>Не допускать прохождения над разрезами полигонов</p>
MCj_ODT[3:0]	out	SSTL_12	<p>Управление терминацией в модулях памяти.</p> <p>Активный уровень - высокий</p>
MCj_PARITY	out	SSTL_12	Четность адресно-командной шины
MCj_RESET_N	out	SSTL_12	<p>Сброс.</p> <p>Активный уровень - низкий</p>
MCj_VREF_TEST	analog inout	SSTL_12	<p>Тестовый вывод аналоговых сигналов PLL, либо опорное напряжение с MASTER_PHY.</p> <p>Нельзя использовать в качестве источника сигнала VREFCA для модулей памяти.</p> <p>Сигналы MCj_VREF_TEST необходимо вывести на тестовые точки. Рядом с каждой тестовой точкой установить тестовую точку GND.</p> <p>При необходимости можно эти сигналы оставить не подключенными.</p> <p>Проводники, идущие к тестовым точкам, должны быть защищены от помех (экранированные GND).</p> <p>Не допускать прохождения над разрезами полигонов</p>
MCj_ZN	analog out	SSTL_12	<p>Подключение внешнего калибровочного сопротивления $240 \pm 1\%$ Ом.</p> <p>Калибровочный резистор должен располагаться в непосредственной близости к микропроцессору.</p> <p>Проводники MCj_ZN идущие к калибровочным резисторам должны быть защищены от помех (экранированные GND).</p> <p>Не допускать прохождения над разрезами полигонов</p>

5.4 Канал SPI

5.4.1 Сигналы канала SPI приведены в таблице 5.5

Таблица 5.5 - Сигналы канала SPI

Название	Вход/ выход	Тип	Назначение
SPI_CS[3:0]_N	out	CMOS_18	Сигналы chip select
SPI_MISO	in	CMOS_18	Выходные данные
SPI_MOSI	out	CMOS_18	Входные данные
SPI_SCK	out	CMOS_18	Синхросигнал

5.5 Каналы I2C master

5.5.1 Микропроцессор имеет пять каналов I2C master: 0 - 4. Они имеют одинаковые интерфейсы, и в таблице 5.6 символ j представляет одну из этих цифр.

Таблица 5.6 – Сигналы каналов I2C

Название	Вход/ выход	Тип	Назначение
I2C_SCLj	inout	CMOS_18 opendrain	Синхросигнал
I2C_SDAj	inout	CMOS_18 opendrain	Адрес/данные

5.6 Канал HDA

5.6.1 Сигналы канала HDA приведены в таблице 5.7.

Таблица 5.7 - Сигналы канала HDA

Название	Вход/ выход	Тип	Назначение
HDA_CLK	out	CMOS_18	Синхросигнал
HDA_RST_	out	CMOS_18	Сигнал начальной установки.

Название	Вход/ выход	Тип	Назначение
			Активный уровень - низкий
HDA_SDI	inout	CMOS_18	Линия приема ответов на команды и потока записи от кодека. Выходная линия в процессе присвоения кодекам адресов
HDA_SDO	out	CMOS_18	Линия передачи команд и потока воспроизведения в кодек
HDA_SYNC	out	CMOS_18	Маркер фреймов и потоков воспроизведения (Frame Synchronization). Активный уровень - высокий

5.7 Каналы WLINK/PCIE

5.7.1 Возможны следующие конфигурации каналов WLINK/PCIE:

- PCI-EXPRESS Gen3 x16;
- PCI-EXPRESS Gen3 2x8;
- PCI-EXPRESS Gen3 x8 + WLINK x8;
- PCI-EXPRESS Gen3 2x4 + WLINCx8;
- PCI-EXPRESS Gen3 4x4;
- WLINC x16.

Сигналы каналов WLINK/PCIE приведены в таблице 5.8.

5.7.2 Назначения сигналов в конфигурациях:

- PCIE0[15:0]: IOWL_PE_{R, T} LANE_{P, N}[15:0];
- PCIE0[15:8] + PCIE0[7:0]: IOWL_PE_{R, T} LANE_{P, N}[15:8] + IOWL_PE_{R, T} LANE_{P, N}[7:0];
- PCIE0[7:0] + WLCC[7:0]: IOWL_PE_{R, T} LANE_{P, N}[7:0] + IOWL_PE_{R, T} LANE_{P, N}[15:8];
- PCIE0[7:4] + PCIE0[3:0] + WLCC[7:0]: IOWL_PE_{R, T} LANE_{P, N}[7:4] + IOWL_PE_{R, T} LANE_{P, N}[3:0] + IOWL_PE_{R, T} LANE_{P, N}[15:8];

- WLCC[15:0]: IOWL_PE_{R, T} LANE_{P, N}[15:0].

Таблица 5.8 - Сигналы каналов WLINK/PCIE

Название	Вход/ выход	Тип	Назначение
PE_RESREF	inout	analog	Подключение калибровочного сопротивления ($200 \pm 1\%$) Ом, 100-ppm/С
PEWL_CONFIG[1:0]	in	CMOS_18 pulldn	<p>Конфигурация канала.</p> <p>PEWL_CONFIG:</p> <p>0 - весь канал имеет функциональность PCIE (при этом режим <code>gx_cdr_ssc_en</code> выключен); разбивка на подканалы PCIE задается контактами PE_PRE_DET#;</p> <p>1 - весь канал имеет функциональность WLCC;</p> <p>2 - весь канал имеет функциональность PCIE (при этом режим <code>gx_cdr_ssc_en</code> включен); разбивка на подканалы PCIE задается контактами PE_PRE_DET#;</p> <p>3 - лэйны [15:8] имеют функциональность WLCC; лэйны [7:0] имеют функциональность PCIE; разбивка на подканалы PCIE задается контактами PE_PRE_DET#</p>
PE_PRE_DET#[3:0]	in	CMOS_18 pullup	<p>Сигналы Presence Detect, соответствующие подключенному PCIE устройству.</p> <p>Активный уровень – низкий.</p> <p>Расшифровка зависит от значения контактов PEWL_CONFIG. Значения PE_PRE_DET# представлены в двоичном виде; символ 'x' означает безразличное состояние.</p> <p>PEWL_CONFIG==0:</p> <p>1110 - lanes[15:0] - PCIEx16(Dev5);</p> <p>110x - lanes[7:0] - PCIEx8(Dev5); lanes[15:8] - PCIEx8(Dev6);</p> <p>x0xx - lanes[3:0] - PCIEx4(Dev5); lanes[7:4] - PCIEx4(Dev6);</p>

Название	Вход/ выход	Тип	Назначение
			lanes[11:8] - PCIex4(Dev7); lanes[15:12] - PCIex4(Dev8); 0xxx - как x0xx; PEWL_CONFIG==3: 1110 - lanes[7:0] - PCIex8(Dev5); xx0x - lanes[3:0] - PCIex4(Dev5); lanes[7:4] - PCIex4(Dev6); x0xx - как xx0x; 0xxx - как xx0x;
PE_RLANE_P[15:0] PE_RLANE_N[15:0]	in	PCIe	Входные данные. Передаются парафазной линией
PE_TLANE_P[15:0] PE_TLANE_N[15:0]	out	PCIe	Выходные данные. Передаются парафазной линией

5.8 Каналы SATA/ETH

5.8.1 Возможны следующие конфигурации каналов SATA/ETH:

- 4xSATA;
- 2xSATA + 1x1G(SGMII/1000Base-X) + 1x2.5G(SGMII);
- 2xSATA + 2x1G(SGMII/1000Base-X);
- 2xSATA + 2x2.5G(SGMII).

Сигналы каналов SATA/ETH приведены в таблице 5.9.

5.8.2 В конфигурации с 4xSATA назначение сигналов следующее:

- SATA0: SATA0_{R,T}}_{P,N};
- SATA1: SATA1_{R,T}}_{P,N};
- SATA2: SATA2_ETH0_{R,T}}_{P,N};
- SATA3: SATA3_ETH1_{R,T}}_{P,N}.

Во всех остальных конфигурациях, с 2xSATA и 2xETH, назначение сигналов следующее:

- SATA0: SATA0_{R,T}}_{P,N};

- SATA1: SATA1_{R,T}}_{P,N};
- eth1G 0 или eth1G 0: SATA2_ETH0_{R,T}}_{P,N};
- eth1G 1: SATA3_ETH1_{R, T}}_{P,N}.

Микропроцессор имеет два канала ETH: 0-1. Они имеют одинаковые интерфейсы (помимо разделяемых с SATA), и в таблице 5.9 символ j представляет одну из этих цифр.

Таблица 5.9 - Сигналы каналов SATA/ETH

Название	Вход/ выход	Тип	Назначение
ETHj_FDUP	in	CMOS_18 pulldn	ETHj full duplex
ETHj_FETH	in	CMOS_18 pulldn	ETHj fast ethernet
ETHj_GETH	in	CMOS_18 pulldn	ETHj gigabit ethernet
ETHj_LINKUP_LED	out	CMOS_18	ETHj link status LED. Активный уровень - низкий.
ETHj_LSTA	in	CMOS_18 pulldn	ETHj link status
ETHj_MDC	out	CMOS_18	ETHj management clock
ETHj_MDIO	inout	CMOS_18 pullup	ETHj management data io
ETHj_MOD_ABS	in	CMOS_18	ETHj module absent
ETHj_PHYRST_	out	CMOS_18	ETHj phy reset
ETHj_RX_LOSS	in	CMOS_18	ETHj receiver loss
ETHj_RXACT_LED	out	CMOS_18	ETHj receiver activity LED. Активный уровень - низкий.
ETHj_TX_DISABLE	out	CMOS_18	ETHj transmitter disable
ETHj_TX_FAULT	in	CMOS_18	ETHj transmitter fault
ETHj_TXACT_LED	out	CMOS_18	ETHj transmitter activity LED. Активный уровень - низкий.
SATA_LED	out	CMOS_18	Контакт для подключения LED SATA.

Название	Вход/ выход	Тип	Назначение
			Активный уровень - высокий
SATAETH_CONFIG	in	CMOS_18 pullup	Конфигурация канала: 0 - 4x SATA; 1 - одна из оставшихся (задается программно)
SATAETH_RESREF	inout	analog	Подключение калибровочного сопротивления $200 \pm 1\%$ Ом, 100-ppm/°C
SATAETH_RLANE_P[3:0] SATAETH_RLANE_N[3:0]	in	PCIe	Входные данные. Передаются парафазной линией
SATAETH_TLANE_P[3:0] SATAETH_TLANE_N[3:0]	out	PCIe	Выходные данные. Передаются парафазной линией

5.9 Каналы RS-232

5.9.1 Микропроцессор имеет два канала последовательного интерфейса RS-232: А, В. Они имеют одинаковые интерфейсы, и в таблице 5.9 символ j представляет одну из этих букв.

Таблица 5.9 – Сигналы каналов RS-232

Название	Вход/ выход	Тип	Назначение
SP_CTSj_	in	CMOS_18 pullup	Сигнал clear to send. Активный уровень - низкий
SP_DCDj_	in	CMOS_18 pullup	Сигнал data carrier detect. Обнаружено другое устройство на линии. Активный уровень - низкий
SP_DTRj_	out	CMOS_18	Сигнал data terminal ready. Запрос на прием данных. Активный уровень - низкий. (он же SP_RCQj_ "gp output")
SP_RTSj_	out	CMOS_18	Сигнал request to send.

Название	Вход/ выход	Тип	Назначение
			Запрос на выдачу данных. Активный уровень - низкий
SP_RXDj	in	CMOS_18 pullup	Входные данные. Активный уровень - высокий
SP_SYNCj_	in	CMOS_18 pullup	Синхросигнал
SP_TXDj	out	CMOS_18	Выходные данные. Активный уровень - высокий

5.10 Каналы USB

5.10.1 Микропроцессор имеет четыре порта USB 3.0 (0 – 3) . Они имеют одинаковые интерфейсы, и в таблице 5.11 сигналов символ j представляет одну из этих цифр.

Таблица 5.11 – Сигналы каналов USB

Название	Вход/ выход	Тип	Назначение
USB_PORT_OVR_CUR_[7:0]	in	CMOS_18 pullup	Сигнал перегрузки по току порта usb. Разряды [3:0] соответствуют портам 3-0 USB 2.0. Разряды [7:4] соответствуют портам 3-0 USB 3.0. Активный уровень - низкий
USBj_D_P USBj_D_N	inout	USB2.0	Порт данных USB 2.0. Передаются парафазной линией
USBj_ID	in inout	USB2.0 analog	Признак соединителя mini-usb или контакт для аналогового тестирования
USBj_RESREF	inout	analog	Подключение калибровочного сопротивления (200 ± 1%) Ом, 100-ppm/С
USBj_RX_P	in	USB3.0	Входные данные USB 3.0.

Название	Вход/ выход	Тип	Назначение
USBj_RX_N			Передаются парафазной линией
USBj_TX_P USBj_TX_N	out	USB3.0	Выходные данные USB 3.0. Передаются парафазной линией
USBj_VBUS	inout	analog	Наличие питания 5,0 В. Подается через сопротивление $30 \pm 1\%$ кОм

5.11 Канал RGB

5.11.1 Сигналы канала RGB представлены в таблице 5.12.

Таблица 5.12 - Сигналы канала RGB

Название	Вход/ выход	Тип	Назначение
GFX0_RGB_CLK_P GFX0_RGB_CLK_N	out	CMOS_18	Синхросигнал (pixel clock) Передаются парафазной линией
GFX0_RGB_DATA[23:0]	out	CMOS_18	Выходные данные. Активный уровень - высокий
GFX0_RGB_HSYNC	out	CMOS_18	horizontal sync Активный уровень - высокий
GFX0_RGB_VSYNC	out	CMOS_18	vertical sync Активный уровень - высокий
GFX0_RGB_CSYNC	out	CMOS_18	composit sync Активный уровень - высокий
GFX0_RGB_DE	out	CMOS_18	display enable Активный уровень - высокий
GFX0_DDC_SCL	inout	CMOS_18 opendrain	DDC-канал монитора: I2C - синхросигнал
GFX0_DDC_SDA	inout	CMOS_18 opendrain	DDC-канал монитора: I2C - адрес/данные
GFX0_GPIO[5:2]	inout	CMOS_18 pullup	Контакты GPIO. Для GFX0_GPIO[5:2] установлена внутренняя утяжка к высокому уровню
GFX0_GPIO[1:0]	inout	CMOS_18	Контакты GFX0_GPIO[1:0] могут использо-

Название	Вход/ выход	Тип	Назначение
			ваться в качестве управление DVI PHY, в этом случае GFX0_GPIO[1] соответствует TX_SCL, I2C - синхросигнал, GFX0_GPIO[0] соответствует TX_SDA, I2C - адрес/данные

5.12 Каналы HDMI

5.12.1 Микропроцессор имеет два канала HDMI: 1, 2. Они имеют одинаковые интерфейсы, и в таблице 5.13 символ j представляет одну из этих цифр.

Таблица 5.13 – Сигналы каналов HDMI

Название	Вход/ выход	Тип	Назначение
GFXj_HDMI_TMDS_CLK_P GFXj_HDMI_TMDS_CLK_N	out	HDMI	Синхросигнал. Передается парафазной линией
GFXj_HDMI_TMDS_DAT_P[2:0] GFXj_HDMI_TMDS_DAT_N[2:0]	out	HDMI	Выходные данные. Передаются парафазной линией
GFXj_HDMI_RESREF	inout	analog.	Контакты для подключения калибровочных сопротивлений
GFXj_HDMI_VP	inout	analog.	Контакты аналогового питания PHY 0.8V
GFXj_HDMI_VPH	inout	analog.	Контакты аналогового питания PHY 1.8V
GFXj_HDMI_GD	inout	analog.	Контакты аналоговой земли PHY
FXj_HDMI_HPD	inout	analog.	Сигнал hot plug detect
GFXj_HDMI_DDCCEC_GND	inout	analog.	Контакты земли
GFXj_HDMI_DDC_SCL	inout	CMOS_18 opendrain	DDC-канал монитора: I2C - синхросигнал
GFXj_HDMI_DDC_SDA	inout	CMOS_18 opendrain	DDC-канал монитора: I2C - адрес/данные
GFXj_HDMI_GPIO	inout	CMOS_18	Контакты GPIO

5.13 Каналы LVDS (видеовыходы)

5.13.1 Микропроцессор Эльбрус-2С3 имеет четыре канала LVDS: 0 - 3. Они имеют одинаковые интерфейсы, и в таблице 5.14 символ j представляет одну из этих цифр.

Таблица 5.14 – Сигналы каналов LVDS

Название	Вход/ выход	Тип	Назначение
GFX3_LVDS_CLK_j_P	out	LVDS_18	Синхросигнал
GFX3_LVDS_CLK_j_N			Передаются парафазной линией
GFX3_LVDS_DAT_j_P[4:0]	out	LVDS_18	Выходные данные.
GFX3_LVDS_DAT_j_N[4:0]			Передаются парафазной линией
GFX3_LVDS_PWM[1:0]	inout	CMOS_18	Управление подсветкой и контрастностью монитора

5.14 Канал I2C slave

5.14.1 Канал I2C slave используется внешним контроллером системного менеджмента (IPMI/BMC) для доступа к встроенным fan-контроллерам и rtc-контроллеру процессора, которые видны как единое устройство с адресом `i2c_device_addr[6:0]==7'b1001100` (как у lm96163).

Сигналы канала I2C slave приведены в таблице 5.15.

Таблица 5.15 - Сигналы канала I2C slave

Название	Вход/ выход	Тип	Назначение
MNTR_SCL	in	CMOS_18	Входной синхросигнал. Частота - не более 1 МГц (1 Мбит/с)

Название	Вход/ выход	Тип	Назначение
MNTR_SDA	inout	CMOS_18 opendrain	Адрес/данные

5.15 Сигналы предупреждения о нештатных ситуациях

5.15.1 Сигналы предупреждения о нештатных ситуациях приведены в таблице 5.16.

Таблица 5.16 - Сигналы предупреждения о нештатных ситуациях

Название	Вход/ выход	Тип	Назначение
CPU_FAULT_N	out	CMOS_18 opendrain	Аварийный сигнал, сообщающий о критических внутренних ошибках микропроцессора. Активный уровень - низкий
CPU_PWR_ALERT_N	in	CMOS_18 pullup	Сигнал предупреждения о нештатной ситуации от источников питания микропроцессора (за исключением источников питания 1,2 В для каналов памяти). Активный уровень - низкий
MACHINE_GEN_ALERT_N	in	CMOS_18 pullup	Сигнал предупреждения о нештатной ситуации от прочих компонентов машины (система охлаждения и т.д.). Активный уровень - низкий
MACHINE_PWR_ALERT_N	in	CMOS_18 pullup	Сигнал предупреждения о нештатной ситуации от AC/DC блока питания машины. Активный уровень - низкий
MC_DIMM_EVENT_N	in	CMOS_18 pullup	Сигнал предупреждения о нештатной ситуации от DIMM-модулей памяти каналов

Название	Вход/ выход	Тип	Назначение
			памяти MC0, MC1. Активный уровень - низкий
MC_PWR_ALERT_N	in	CMOS_18 pullup	Сигнал предупреждения о нештатной ситуации от источника питания 1,2. Активный уровень - низкий
MC0_FAULT_N	out	CMOS_18 opendrain	Аварийные сигналы, сообщающие о критических ошибках внутри контроллеров и модулей памяти. Активный уровень - низкий
MNTR_ALERTN_	out	CMOS_18 opendrain	Сигнал предупреждения о вынужденном снижении системной частоты из-за повышенной температуры микропроцессора. Активный уровень - низкий
MNTR_TTRIPN_	out	CMOS_18 opendrain	Аварийный сигнал, сообщающий о повышении температуры микропроцессора до предельно допустимой. Активный уровень - низкий

5.16 Каналы управления вентиляторами

5.16.1 Микропроцессор имеет два канала для управления вентиляторами: 0, 1. Они имеют одинаковые интерфейсы, и в таблице 5.17 символ j представляет одну из этих цифр.

Таблица 5.17 – Сигналы каналов управления вентиляторами

Название	Вход/ выход	Тип	Назначение
FAN_ALERTN[j]_	out	CMOS_18 opendrain	Нештатная ситуация в FAN-контроллере или в вентиляторе. Активный уровень - низкий
FAN_PWM[j]	out	CMOS_18	ШИМ-сигнал управления скоростью вращения вентилятора.

Название	Вход/ выход	Тип	Назначение
			(Обычный push-pull без z-состояния)
FAN_TACH[j]	in	CMOS_18 pulldn	Сигнал обратной связи от вентилятора. Должен подключаться к вентилятору через level shifter

5.17 Канал калибровки и диагностики PVT-сенсоров

5.17.1 Сигналы канала калибровки и диагностики PVT-сенсоров приведены в таблице 5.18.

Таблица 5.18 - Сигналы канала калибровки и диагностики PVT-сенсоров

Название	Вход/ выход	Тип	Назначение
AN_IO_TS[1:0]	inout	CMOS_18	Сигнал калибровки и диагностики встроенных термосенсоров
AN_IO_VM[1:0]	inout	CMOS_18	Сигнал диагностики встроенных мониторов питания (вольтметров)

5.18 Сигналы питания

5.18.1 Сигналы питания приведены в таблице 5.19.

Таблица 5.19 - Сигналы питания

Наименование цепи питания	Описание	Номинальное значение, В	Опорная земля	Обозначение группы

Наименование цепи питания	Описание	Номинальное значение, В	Опорная земля	Обозначение группы
PWR_0V8_CORE	Основная цепь питания МП	0,85	GND_CORE	U _{CC1}
PWR_0V8_BIO	Цепь питания низкоскоростных интерфейсов	0,82	GND_12G	U _{CC2}
PWR_A0V8_PE0	Цепь аналогового низковольтного питания PHY блока части интерфейса WLCC/PCIe [0:3]	0,82	GND_12G	
PWR_A0V8_PE1	Цепь аналогового низковольтного питания PHY блока части интерфейса WLCC/PCIe [4:7]	0,82	GND_12G	
PWR_A0V8_PE2	Цепь аналогового низковольтного питания PHY блока части интерфейса WLCC/PCIe [8:11]	0,82	GND_12G	
PWR_A0V8_PE3	Цепь аналогового низковольтного питания PHY блока части интерфейса WLCC/PCIe [12:15]	0,82	GND_12G	
PWR_A0V8_SATAETH0	Цепь аналогового питания физических уровней SATA	0,82	GND_12G	
PWR_A0V8_SATAETH1	Цепь аналогового низковольтного питания физических уровней SATA/Ethernet	0,82	GND_12G	
PWR_A0V8_USB0	Цепь аналогового низковольтного питания физических уровней USB	0,82	GND_12G	
PWR_A0V8_USB1	Цепь аналогового низковольтного питания физических уровней USB	0,82	GND_12G	
PWR_A0V8_USB2	Цепь аналогового низковольтного питания физических уровней USB	0,82	GND_12G	
PWR_A0V8_USB3	Цепь аналогового низковольтного питания физических уровней USB	0,82	GND_12G	
PWR_A0V8_GFX1	Цепь аналогового низковольтного питания канала GFX1_HDMI	0,82	GND_12G	

Наименование цепи питания	Описание	Номинальное значение, В	Опорная земля	Обозначение группы
PWR_A0V8_GFX2	Цепь аналогового низковольтного питания канала GFX2_HDMI	0,82	GND_12G	
PWR_MC	Цепь питания выходных каскадов каналов DDR4 0, 1	1,2	GND_MC	U _{CC3}
PWR_1V8_BIO	Цепь питания выходных каскадов низкоскоростных интерфейсов	1,8	GND_BIO	U _{CC4}
PWR_A1V8_PE	Цепь аналогового высоковольтного питания PHY блока части интерфейса IOWLink/PCIe0 [15:0]	1,8	GND_12G	
PWR_A1V8_MC0_PLL	Цепь аналогового питания блоков синхронизации канала памяти DDR4 0	1,8	GND_MC	
PWR_A1V8_MC1_PLL	Цепь аналогового питания блоков синхронизации канала памяти DDR4 1	1,8	GND_MC	
PWR_A1V8_SATAETH	Цепь аналогового высоковольтного питания физических уровней SATA и SATA/Ethernet	1,8	GND_12G	
PWR_A1V8_SYNC_TOP	Цепь аналогового высоковольтного питания блоков синхронизации	1,8	GND_SYNC_TOP	
PWR_A1V8_SYNC_BOTTOM	Цепь аналогового высоковольтного питания блоков синхронизации	1,8	GND_SYNC_BOTTOM	
PWR_A1V8_GFX1	Цепь аналогового высоковольтного питания канала GFX1_HDMI	1,8	GND_HDMI	
PWR_A1V8_GFX2	Цепь аналогового высоковольтного питания канала GFX2_HDMI	1,8	GND_HDMI	
PWR_A3V3_USB	Цепь аналогового высоковольтного питания физических уровней USB	3,3	GND_HDMI	U _{CC5}
PWR_0V8_SUS	Цепь питания логики блока Suspend. Источником питания является неот-	0,8	GND_HDMI	U _{CC6}

Наименование цепи питания	Описание	Номинальное значение, В	Опорная земля	Обозначение группы
	ключаемое питание в соответствии со спецификацией ATX			
PWR_1V8_SUS	Цепь питания выходных каскадов блока Suspend. Источником питания является неотключаемое питание в соответствии со спецификацией ATX	1,8	GND_HDMI	U _{C7}
PWR_0V8_MC	Цепь питания логики каналов DDR4 0, 1. Допускается подавать повышенное напряжение от 0.855 - 0.9 - 0.945 В. Допускается подключение к группе U _{CC2}	0,8	GND_MC	U _{CC8}
PWR_1V5_HDA	Цепь питания HDA. Допускает работу от номинала 1,8 В и объединение с группой U _{CC4} .	1,5	GND_BIO	U _{CC9}
PWR_1V8_EFUSE	Цепь питания, которое используется для программирования блока однократно программируемого ПЗУ на кристалле типа eFuse. Для программирования необходимо подключить к питанию 1,8 В. Для штатной работы подключить к земле	1,8	GND_BIO	U _{CC10}
PWR_A0V8_SYNC_TOP	Цепь аналогового низковольтного питания блоков синхронизации. Допускается подключение с группе U _{CC2}	0,82	GND_SYNC_TOP	U _{CC11}
PWR_A0V8_SYNC_BOT	Цепь аналогового низковольтного питания блоков синхронизации.	0,82	GND_SYNC_BOT	

Наименование цепи питания	Описание	Номинальное значение, В	Опорная земля	Обозначение группы
	Допускается подключение с группе Uсс2			
GND_12G	Цепь земли высокоскоростных каналов и низкоскоростных интерфейсов. Должна быть подключена на общую землю МПП			GND
GND_CORE	Цепь земли основного домена питания. Должна быть подключена на общую землю МПП			GND
GND_MC	Цепь земли каналов DDR4 0, 1. Должна быть подключена на общую землю МПП			GND
GND_USB	Цепь земли каналов USB. Должна быть подключена на общую землю МПП			GND
GND_BIO	Цепь земли низкоскоростных интерфейсов блока Suspend и каналов LVDS. Должна быть подключена на общую землю МПП			GND
GND_HDMI	Цепь земли каналов HDMI. Должна быть подключена на общую землю МПП			GND
GND_SYNC_TOP	Цепь земли блоков синхронизации. Должна быть подключена на общую землю МПП			GND
GND_SYNC_BOT	Цепь земли блоков синхронизации. Должна быть подключена на общую землю МПП			GND

6 Распределение пространства физических адресов

6.1 Аппаратно различимые области в пространстве физических адресов системы

6.1.1 В пространстве физических адресов системы может быть выделено несколько семантически различных типов областей. В аппаратуре микропроцессора и межкластерных коммутаторов предусмотрены средства для определения принадлежности адреса запроса той или иной области и направления запроса в тот ресурс системы, который реализует данный адрес.

Аппаратно различаются следующие типы областей:

- а) область программы начальной загрузки - BOOT. Эта область является единственной в системе и располагается по фиксированным адресам (см. 6.2);
- б) системные и служебные регистры узлов - NBSR_j (j - номер узла). Пространство для размещения этих областей начинается с фиксированного адреса (см. 6. 2) и имеет переменный размер, определяемый количеством процессоров в системе. Адреса начала областей NBSR_j кратны их размеру, поэтому одноименные адреса разных областей отличаются только значениями определенных разрядов;
- в) оперативная память. Вводятся понятия оперативной памяти верхнего диапазона и нижнего диапазона.

Верхний диапазон описывает всю установленную в системе оперативную память и располагается за совокупным пространством системных регистров вплоть до конца физпамяти. Этот диапазон заполняется непересекающимися областями МНІ_j (j – номер узла); каждая область описывает память, установленную в j-м узле. В общем случае соседние области МНІ_j могут лежать несмежно, а каждая из областей может быть не сплошной - это зависит от конфигурации установленных модулей памяти. Если память узла содержит "дыры", это должно учитываться системным ПО с помощью карты доступной памяти. Отображение установленных модулей памяти узла в пространство

физических адресов должно следовать определенным правилам, описанным в 6.3.1.

Нижний диапазон может располагаться от 0-го адреса до начала области PCI, но не выше 4 GB. Он заполняется произвольным количеством непересекающихся областей MLO_j (j - номер узла), каждая из которых должна целиком принадлежать одному узлу. Границы областей MLO_j кратны 128 MB. Поскольку вся установленная память изначально принадлежит верхнему диапазону, для наполнения областей MLO_j реальной памятью в системе предусмотрены средства перемещения фрагментов памяти из верхнего диапазона в нижний; при этом "дыры", образующиеся в верхнем диапазоне, должны учитываться системным ПО с помощью карты доступной памяти. Переназначение памяти из верхнего диапазона в нижний выполняется по правилам, описанным в 6.3.2;

г) области MEM на шинах PCI - PCIM_j (j - номер узла). Они располагаются по динамически назначаемым непересекающимся адресам в пределах 4 GB (см. 6.2).

Границы областей кратны 128 MB. Если к узлу не подключено периферийное оборудование, его область PCIM_j имеет нулевой размер;

д) области Prefetchable MEM на шинах PCI - PCIMP_j (j - номер узла). Они располагаются по динамически назначаемым непересекающимся адресам (см. 6.2).

Границы областей кратны 128 MB. Если к узлу не подключено периферийное оборудование, его область PCIMP_j имеет нулевой размер;

е) области IO на шинах PCI - PCIIO_j (j - номер узла). Они располагаются по динамически назначаемым непересекающимся адресам в определенном диапазоне (см. 6.2). Границы областей кратны 128 MB; размеры равны 4 KB;

ж) области конфигурации на шинах PCI - PCICFG_j (j - номер узла). Пространство для размещения этих областей начинается с динамически назначаемого адреса и имеет переменный размер, определяемый количеством КПИ в системе. Адреса начала областей PCICFG_j кратны их размеру, поэтому

одноименные адреса разных областей отличаются только значениями определенных разрядов;

з) программно доступные регистры IOAPIC - IOAPICREG_j (j - номер узла). Они располагаются по динамически назначаемым непересекающимся адресам в определенном диапазоне (см. 6.2) в пределах 4 GB. Границы областей кратны 4 KB; размеры равны 4 KB;

и) программно доступные регистры EPIIC - EPIICREG. Программные регистры EPIIC конкретного ядра доступны только своему ядру, поэтому для всех EPIIC системы выделяется одинаковый диапазон адресов - EPIICREG. Он располагается по динамически назначаемым адресам в определенном диапазоне (см. 6.2) в пределах 4 GB. Границы областей EPIICREG кратны 16 KB; размеры равны 16 KB;

к) память видеокарты - VGAMEM. Эта область является единственной в системе и располагается по фиксированным адресам (см. 2). Эта область может пересекаться с областями MLO_j - в этом случае пересекающиеся фрагменты MLO_j теряются;

л) служебные адреса EPIIC, используемые для передачи сообщений из IOAPIC (КПИ-2) в EPIIC;

м) служебные адреса IOAPIC (КПИ-2), используемые для передачи сообщений из EPIIC в IOAPIC (КПИ-2);

н) область для передачи сообщений типа MSI из IOEPIIC в SEPIIC.

6.2 Карта физической памяти

6.2.1 Карта физической памяти приведена в таблице 6.1.

Таблица 6.1 - Карта физической памяти

Адреса	Наименование областей
00_0000_0000 00_ххFF_FFFF	Нижний диапазон оперативной памяти - MLO; верхняя граница диапазона MLO назначается динамически системным ПО и не превышает нижней границы областей PCIM _j ; границы областей MLO _j кратны 128 Мбайт

Адреса	Наименование областей
00_000A_0000 00_000B_FFFF	Память видеокарты - VGAMEM; может пересекаться с диапазоном MLO - в этом случае пересекающиеся фрагменты MLO теряются
00_хх00_0000 00_FEBF_FFFF	Области MEM на шинах PCI - PCIMj; нижняя граница областей PCIMj назначается динамически системным ПО и не может быть меньше верхней границы диапазона MLO; границы областей PCIMj кратны 128 Мбайт
00_FEC0_0000 00_FED0_FFFF	Программно доступные регистры IOAPIC - IOAPICREGj; границы областей IOAPICREGj кратны 4 Кбайт; размеры равны 4 Кбайт.
00_FED1_0000 00_FFDF_FFFF	Резерв
00_FEE0_0000 00_FEF0_FFFF	Программно доступные регистры EPIC - EPICREG; границы областей EPICREG кратны 4 Кбайт; размеры равны 4 Кбайт.
00_FEE0_0000 00_FEE0_0FFF	Программно доступные регистры CEPIC; обращения в режиме (GM==0) попадают в комплект регистров хоста; обращения в режиме (GM==1) попадают в комплект регистров гостя
00_FEE0_1000 00_FEE0_1FFF	Адреса для обращения гипервизора (GM==0) к комплекту регистров CEPIC гостя;
00_FEE0_2000 00_FEE0_2FFF	Программно доступные регистры PREPIC - PREPICREG
00_FEE0_4000 00_FFFF_FFFF	Резерв
01_0000_0000 01_00FF_FFFF	Область программы начального старта - BOOT
01_0100_0000 01_0100_FFFF	Области IO на шинах PCI - PCIOj; границы областей PCIOj кратны 4 Кбайт
01_1000_0000 01_100F_FFFF	Программно доступные регистры узла #0 - NBSR0. Адреса начала и конца области j вычисляется как: $Abgn = 0x01_1000_0000 + j*0x10_0000;$ $Aend = Abgn + 1MB - 1;$
-----	-----
01_1хх0_0000 01_1ххF_FFFF	Программно доступные регистры узла #j - NSRj. Адреса начала и конца области j вычисляется как

Адреса	Наименование областей
	$Abgn = 0x01_1000_0000 + j*0x10_0000;$ $Aend = Abgn + 1 \text{ Мбайт} - 1;$
01_2000_0000 01_200F_FFFF	Область для передачи сообщений типа MSI из IOEPIC в CEPIC
01_2100_0000 01_21FF_FFFF	<p>Область служебных адресов EPIC, используемых для передачи сообщений из IOAPIC (КПИ-2) в EPIC.</p> <p>Примечание - По умолчанию адресный регистр IOAPIC принимает нулевое значение, поэтому ПО обязано его инициализировать.</p>
01_2200_0000 01_22FF_FFFF	Области служебных адресов IOAPIC, используемых для сообщений в с- Область служебных адресов IOAPIC, используемых для передачи сообщений из EPIC в IOAPIC (КПИ-2)
01_2300_0000 01_FFFF_FFFF	Резерв
02_0000_0000 0x_xFFF_FFFF	<p>- Области конфигурационных регистров PCI - PCICFG. Для каждого кластера выделяется смежная область, включающая подобласти для всех КПИ кластера. Начало области определяется регистром RT_PCICFGB.</p> <p>Адреса начала и конца области для КПИ #j вычисляется как:</p> $Abgn = RT_PCICFGB + j*0x1000_0000;$ $Aend = Abgn + j*0x100_0000;$ $Aend = Abgn + 256 \text{ Мбайт} - 1$
0x_x000_0000 yy_yyFF_FFFF	Области MEMP на шинах PCI - PCIMPj; нижняя граница областей PCIMj назначается динамически системным ПО; границы областей PCIMPj кратны 128 MB
z0_0000_0000 FF_FFFF_FFFF	- Верхний диапазон оперативной памяти - МНІ. Адрес начала диапазона зависит от размера диапазона PCIMP и назначается системным ПО; границы областей МНІj кратны 4 GB
Примечание - Многие из перечисленных в 6.1 областей могут быть размещены в произвольном месте физпамяти, поэтому приведенное ниже распределение является лишь примером	

6.3 Правила отображения оперативной памяти в пространство физических адресов

6.3.1 Мэппирование оперативной памяти верхнего диапазона.

Подсистема оперативной памяти узла имеет K каналов (для "Эльбрус-2С3" $K=2$), каждый из которых обслуживает по два разъема ($m=0, 1$) для установки модулей памяти емкостью от 2 GB до 128 GB.

Вариант интерливинга по разрядам адреса:

6 - для "Эльбрус-2С3" и режим полного чередования адресов.

В подсистеме памяти предусмотрен управляющий регистр, позволяющий системному ПО задавать распределение (интерливинг) физических адресов по каналам. Поле регистра `HMU_MIS.mci1_bit0` задает номер разряда физадреса, значение которого номер канала.

Указанный разряд изымается из адреса, а более старшие разряды сдвигаются на его место.

Размер и расположение суммарного диапазона физадресов вычисляется следующим образом.

Пусть суммарная емкость памяти в k -м канале равна $S_{ch}[k]$. Пусть наибольшее из них равно s :

$$s = \max(S_{ch}[k]), \text{ для всех } k;$$

Пусть ближайшее число, равное или большее s и являющееся степенью двойки, равно S , а подходящая степень двойки равна N .

Тогда суммарный диапазон адресов, занимаемых узлом j , имеет размер

$$S_{node}[j] = K * S;$$

Область памяти узла размещается в пространстве физических адресов так, чтобы ее базовый адрес был кратен как минимум 4 GB; если $S_{node}[j]$ превышает 4 GB, базовый адрес должен быть кратен $S_{node}[j]$. Если ($S_{ch}[k] < S$), системное ПО должно гарантировать неиспользование отсутствующих физических адресов.

6.3.2 Мэппирование оперативной памяти нижнего диапазона

Определено, что вся установленная в узле память изначально принадлежит верхнему диапазону, поэтому для наполнения областей MLO_j реальной памятью в системе предусмотрены средства перемещения фрагментов памяти из верхнего диапазона в нижний. Перемещение выполняется по следующим правилам.

Пусть размер назначенной для узла *j* области верхнего диапазона равен $S_node[j]$, и базовый адрес кратен как минимум 4 GB (см. 6.3.1). Тогда разрешенными кандидатами на перемещение в нижний диапазон являются адреса, принадлежащие первому 4-гигабайтному участку. Порождаемый адрес нижнего диапазона (A_{hi2lo}) равен смещению (*offset*) адреса в верхнем диапазоне (A_{hi}) относительно его начала (A_{bgn}):

$$A_{hi} = A_{bgn} + \text{offset};$$

$$A_{hi2lo} = \text{offset};$$

Будучи перенесен в нижний диапазон, адрес A_{hi} должен быть изъят из карты памяти верхнего диапазона.

Примечания:

1 Системное ПО должно учитывать, что при определенном расположении начала области MLO_j адреса A_{hi2lo} могут "вырезать дыру" в верхнем диапазоне таким образом, что останется несмежный фрагмент MН_{ij}. Например, пусть требуемый MLO_j имеет размер 384 MB (128 MB*3) и начинается с адреса 128 MB. Тогда в первых 4-х гигабайтах области MН_{ij} будут "вырезаны" 1-й, 2-й и 3-й сегменты по 128 MB, а 0-й сегмент останется в верхнем диапазоне отдельно от остальной памяти. Карта памяти узла становится "дырявой".

2 Системное ПО должно учитывать, что, при размере MLO_j сравнимом с объемом памяти узла, не всякое расположение области MLO_j корректно. Например, пусть в узле установлена память емкостью 1 GB и она "прижата" к началу MН_{ij}. Пусть требуемый MLO_j имеет размер 0,5 GB, но начинается с адреса 1 GB. Тогда адреса-жертвы A_{hi} (и A_{hi2lo}) попадают в участок, который выходит за границы наличной памяти.

7 Эксплуатационные ограничения

7.1 Электрические параметры и режимы эксплуатации

7.1.1 Электрические параметры микросхем приведены в таблице 7.1.

Таблица 7.1 - Электрические параметры микросхем

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Темпе- ратура среды, °С	Номер пункта примечания
		не менее	не более		
1 Выходное напряжение низкого уровня интерфейсов типа CMOS_18, В (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,70$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,70$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,85$ В, $I_{O1L} = 2$ мА)	U_{OL1}	—	0,45	от ми- нус 40 до плюс 90*	
2 Выходное напряжение высокого уровня интерфейсов типа CMOS_18, В (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,70$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,70$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,85$ В, $I_{O1L} = 2$ мА)	U_{OH1}	1,35	—	от ми- нус 40 до плюс 90*	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Темпе- ратура среды, °С	Номер пункта примечания
		не менее	не более		
3 Ток утечки на входе интерфейсов типа CMOS_18, мкА (при $U_{CC1} = 0,85 \text{ В}$, $U_{CC2} = 0,85 \text{ В}$, $U_{CC3} = 1,20 \text{ В}$, $U_{CC4} = 1,90 \text{ В}$, $U_{CC5} = 3,30 \text{ В}$, $U_{CC6} = 0,85 \text{ В}$, $U_{CC7} = 1,90 \text{ В}$, $U_{CC8} = 0,85 \text{ В}$, $U_{CC9} = 1,50 \text{ В}$, $U_{CC11} = 0,85 \text{ В}$, и $U_{IH1} = 1,90 \text{ В}$, $U_{IL1} = 0 \text{ В}$)	I_{IL}	- 10,0	10,0	от ми- нус 40	1
		- 300	300	до плюс 90*	2
4 Выходное напряжение низкого уровня интерфейса данных и адреса каналов памяти (тип SSTL2), В (при $U_{CC1} = 0,85 \text{ В}$, $U_{CC2} = 0,85 \text{ В}$, $U_{CC3} = 1,14 \text{ В}$, $U_{CC4} = 1,80 \text{ В}$, $U_{CC5} = 3,30 \text{ В}$, $U_{CC6} = 0,85 \text{ В}$, $U_{CC7} = 1,80 \text{ В}$, $U_{CC8} = 0,85 \text{ В}$, $U_{CC9} = 1,50 \text{ В}$, $U_{CC11} = 0,85 \text{ В}$)	U_{OL2}	—	0,23	От минус 40 до плюс 90*	
5 Выходное напряжение высокого уровня интерфейса данных и адреса каналов памяти (тип SSTL2), В (при $U_{CC1} = 0,85 \text{ В}$, $U_{CC2} = 0,85 \text{ В}$, $U_{CC3} = 1,14 \text{ В}$, $U_{CC4} = 1,80 \text{ В}$, $U_{CC5} = 3,30 \text{ В}$, $U_{CC6} = 0,85 \text{ В}$, $U_{CC7} = 1,80 \text{ В}$, $U_{CC8} = 0,85 \text{ В}$, $U_{CC9} = 1,50 \text{ В}$, $U_{CC11} = 0,85 \text{ В}$)	U_{OH2}	0,91	—	от ми- нус 40 до плюс 90*	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Темпе- ратура среды, °С	Номер пункта примечания
		не менее	не более		
6 Выходное напряжение низкого уровня интерфейса LVDS, В (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,70$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,70$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,85$ В, $R_{L1} = 100$ Ом)	U_{OL3}	0,92	—	от ми- нус 40 до плюс 90*	
7 Выходное напряжение высокого уровня интерфейса LVDS, В (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,70$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,70$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,85$ В, $R_{L1} = 100$ Ом)	U_{HL3}	—	1,48	от ми- нус 40 до плюс 90*	
8 Выходное напряжение низкого уровня интерфейса HDA, В (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,8$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,80$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,40$ В, $U_{CC11} = 0,85$ В, $I_{OL2} = 1,5$ мА)	U_{OL4}	—	0,45	от ми- нус 40 до плюс 90*	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Темпе- ратура среды, °С	Номер пункта примечания
		не менее	не более		
9 Выходное напряжение высокого уровня интерфейса HDA, В (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,857$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,80$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,80$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,40$ В, $U_{CC11} = 0,85$ В, $I_{OH2} = 0,5$ мА)	U_{OH4}	1,35	—	от ми- нус 40 до плюс 90*	
10 Ток потребления статический от источников питания U_{CC1} , U_{CC2} , U_{CC6} , U_{CC8} и U_{CC11} , А (при $U_{CC1} = 0,90$ В, $U_{CC2} = 0,90$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,80$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,90$ В, $U_{CC7} = 1,80$ В, $U_{CC8} = 0,90$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,90$ В)	$I_{CC1}+I_{CC2}+$ $+I_{CC6}+I_{CC8}+$ $+I_{CC11}$	—	2,0	от ми- нус 40 до плюс 90*	
11 Ток потребления статический от источника питания U_{CC3} , А (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,26$ В, $U_{CC4} = 1,80$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,80$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,85$ В)	I_{CC3}	—	0,5	от минус 40 до плюс 90*	
12 Ток потребления статический от источника питания U_{CC4} , U_{CC7} и U_{CC9} , А (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,90$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,90$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,85$ В)	$I_{CC4}+I_{CC7}+$ $+ I_{CC9}$	—	0,4	от ми- нус 40 до плюс 90*	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Темпе- ратура среды, °С	Номер пункта примечания
		не менее	не более		
13 Ток потребления статический от источников питания U_{CC5} , А (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,80$ В, $U_{CC5} = 3,60$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,80$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,85$ В)	I_{CC5}	—	0,5	от ми- нус 40 до плюс 90*	
14 Ток потребления при проверке коротких замыканий от источни- ков питания U_{CC1} , U_{CC2} , U_{CC6} , U_{CC8} , U_{CC9} , и U_{CC11} , мА (при нормальных климатических условиях, $U_{CC1} = 0,3$ В, $U_{CC2} = 0,3$ В, $U_{CC3} = 0,3$ В, $U_{CC4} = 0,3$ В, $U_{CC5} = 0,3$ В, $U_{CC6} = 0,3$ В, $U_{CC7} = 0,3$ В, $U_{CC8} = 0,3$ В, $U_{CC9} = 0,3$ В, $U_{CC11} = 0,3$ В)	I_{CCS1+} $+I_{CCS2+}$ $+I_{CCS6+}$ $+I_{CCS8+}$ $+I_{CCS11}$	—	300	25	
15 Ток потребления при проверке коротких замыканий от источников питания U_{CC3} , мА (при нормальных климатических условиях, $U_{CC1} = 0,3$ В, $U_{CC2} = 0,3$ В, $U_{CC3} = 0,3$ В, $U_{CC4} = 0,3$ В, $U_{CC5} = 0,3$ В, $U_{CC6} = 0,3$ В, $U_{CC7} = 0,3$ В, $U_{CC8} = 0,3$ В, $U_{CC9} = 0,3$ В, $U_{CC11} = 0,3$ В)	I_{CCS3}	—	10,0	25	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Темпе- ратура среды, °С	Номер пункта примечания
		не менее	не более		
16 Ток потребления при проверке коротких замыканий от источника питания U_{CC4} , U_{CC7} и U_{CC9} , мА (при нормальных климатических условиях, $U_{CC1} = 0,3$ В, $U_{CC2} = 0,3$ В, $U_{CC3} = 0,3$ В, $U_{CC4} = 0,3$ В, $U_{CC5} = 0,3$ В, $U_{CC6} = 0,3$ В, $U_{CC7} = 0,3$ В, $U_{CC8} = 0,3$ В, $U_{CC9} = 0,3$ В, $U_{CC11} = 0,3$ В)	I_{CCS4+} $+I_{CCS7+}$ $+I_{CCS9}$	—	10,0	25	
17 Ток потребления при проверке коротких замыканий от источника питания U_{CC5} , мА (при нормальных климатических условиях, $U_{CC1} = 0,3$ В, $U_{CC2} = 0,3$ В, $U_{CC3} = 0,3$ В, $U_{CC4} = 0,3$ В, $U_{CC5} = 0,3$ В, $U_{CC6} = 0,3$ В, $U_{CC7} = 0,3$ В, $U_{CC8} = 0,3$ В, $U_{CC9} = 0,3$ В, $U_{CC11} = 0,3$ В)	I_{CCS5}	—	10,0	25	
18 Выходное дифференциальное напряжение интерфейсов PCI Express, Ethernet, SATA, HDMI, В (при $U_{CC1} = 0,83$ В, $U_{CC2} = 0,83$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,80$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,83$ В, $U_{CC7} = 1,80$ В, $U_{CC8} = 0,83$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,83$ В и $R_{L2} = 100$ Ом)	U_{OD1}	0,40	0,60	от ми- нус 40 до плюс 90*	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Темпе- ратура среды, °С	Номер пункта примечания
		не менее	не более		
19 Выходное дифференциальное напряжение интерфейса LVDS, В (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,70$ В, $U_{CC5} = 3,30$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,80$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,85$ В и $R_{L3} = 100$ Ом)	U_{OD2}	0,25	0,40	от ми- нус 40 до плюс 90*	
20 Выходное напряжение низкого уровня интерфейса USB (full speed режим), В (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,80$ В, $U_{CC5} = 3,07$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,80$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,85$ В и $R_{L4} = 1,5$ КОм)	U_{OL4}	—	0,30	от минус 40 до плюс 90*	
21 Выходное напряжение высокого уровня интерфейса USB (full speed режим), В, (при $U_{CC1} = 0,85$ В, $U_{CC2} = 0,85$ В, $U_{CC3} = 1,20$ В, $U_{CC4} = 1,80$ В, $U_{CC5} = 3,07$ В, $U_{CC6} = 0,85$ В, $U_{CC7} = 1,80$ В, $U_{CC8} = 0,85$ В, $U_{CC9} = 1,50$ В, $U_{CC11} = 0,85$ В и $R_{L4} = 1,5$ КОм)	U_{OH4}	2,80	—	от ми- нус 40 до плюс 90*	
<p>Примечания</p> <p>1 В случаях, когда уровень входного сигнала $U_{IL1} = 0$ В или $U_{IH1} = 1,90$ В совпадает с уровнем подключения сопротивления на входе к 0 В или 1,90 В соответственно.</p> <p>2 В случаях, когда уровень входного сигнала $U_{IH1} = 1,90$ В или $U_{IL1} = 0$ В не сов-</p>					

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Темпе- ратура среды, °С	Номер пункта примечания
		не менее	не более		
падает с уровнем подключения сопротивления на входе к 0 В или 1,90 В соответствен- но (см. ТВГИ.431281.027ТБ).					
* Температура корпуса					

Номинальное значение напряжения питания микросхем:

- $U_{CC1} = 0,85$ В для домена CORE;
- $U_{CC2} = 0,82$ В для домена UNCORE и низковольтного питания PHY блоков периферии;
- $U_{CC3} = 1,20$ В для IO ячеек каналов памяти MC0, MC1;
- $U_{CC4} = 1,80$ В для интерфейсов управления и диагностики, LVDS и высоковольтного питания PHY блоков периферии (кроме USB);
- $U_{CC5} = 3,30$ В для высоковольтного питания PHY блоков интерфейса USB;
- $U_{CC6} = 0,80$ В для питания внутренних схем блока Suspend;
- $U_{CC7} = 1,80$ В для питания выходных каскадов блока Suspend;
- $U_{CC8} = 0,80$ В для питания внутренних схем каналов памяти MC0 – MC1;
- $U_{CC9} = 1,80$ В для интерфейса HDA;
- $U_{CC10} = 1,80$ В для питания при программировании блока ПЗУ eFuse;
- $U_{CC11} = 0,82$ В для питания блоков синхронизации.

Допустимые отклонения напряжений питания от номинальных не должны быть более:

U_{CC1}	от 0,8 до 0,9 В
$U_{CC2}, U_{CC6}, U_{CC8}$	от 0,78 до 0,90 В
U_{CC11}	$\pm 0,08$ В
U_{CC3}	$\pm 0,06$ В
U_{CC4}, U_{CC7} и U_{CC10}	$\pm 0,09$ В

U_{CC9} от 1,425 до 1,89 В

U_{CC5} от 3,07 до 3,63 В

7.1.2 Значения предельно допустимых и предельных режимов эксплуатации в диапазоне рабочих температур среды соответствуют нормам, приведенным в таблице 7.2.

Таблица 7.2 — Предельно допустимые и предельные режимы эксплуатации

Наименование параметра, режима, единица измерения	Буквенное обозначение параметра	Предельно допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
1 Напряжение питания домена CORE, В	U_{CC1}	0,80	0,90	-0,2	1,1
2 Напряжение питания домена UNCORE, В	U_{CC2}	0,78	0,90	-0,2	1,1
3 Напряжение питания интерфейса каналов памяти, В	U_{CC3}	1,14	1,26	-0,3	1,5
4 Напряжение питания интерфейсов типа CMOS_1.8, LVDS и PHY блоков периферии (кроме USB), В	U_{CC4}	1,71	1,89	-0,3	2,0
5 Напряжение питания PHY блоков интерфейса USB, В	U_{CC5}	3,07	3,63	-0,3	4,2
6 Напряжение питания внутренних схем блока Suspend, В	U_{CC6}	0,78	0,90	-0,2	1,1
7 Напряжение питания выходных каскадов блока Suspend, В	U_{CC7}	1,71	1,89	-0,3	2,0
8 Напряжение питания внутренних схем каналов памяти, В	U_{CC8}	0,78	0,90	-0,2	1,1
9 Напряжение питания интерфейса HDA*, В	U_{CC9}	1,425	1,89	-0,30	2,00
10 Напряжение питания при программировании блока ПЗУ eFuse, В	U_{CC10}	1,71	1,89	-0,3	2,0
11 Напряжение питания блоков синхро-	U_{CC11}				

Наименование параметра, режима, единица измерения	Буквенное обозначение параметра	Предельно допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
низации, В		0,72	0,88	-0,2	1,1
12 Входное напряжение высокого уровня интерфейса типа CMOS_1.8, В	U_{IH1}	$0,65 U_{CC4}$	1,98	-0,3	$U_{CC4} + 0,3$
13 Входное напряжение низкого уровня интерфейса типа CMOS_1.8, В	U_{IL1}	-0,3	$0,35 U_{CC4}$	-0,3	$U_{CC4} + 0,3$
14 Диапазон входного дифференциального напряжения интерфейса LVDS, В	U_{ID1}	0,1	1,8	-0,3	$U_{CC4} + 0,3$
15 Входное дифференциальное напряжение интерфейсов PCIe, Ethernet, SATA, RDMI, В	U_{ID3}	0,15	0,60	-0,3	$U_{CC4} + 0,3$
16 Рабочая частота системного синхросигнала, МГц	f_C	99,98	100,02	—	—
17 Емкость нагрузки каждого выхода интерфейсов, кроме JTAG, пФ	C_{L1}	—	5	—	—
18 Емкость нагрузки каждого выхода интерфейса JTAG, пФ	C_{L2}	—	35	—	—
* Допускается использование напряжения питания 1,8 В.					

7.1.3 Предельно допустимые и предельные режимы источников питания представлены в таблице 7.3.

Таблица 7.3 - Предельно допустимые и предельные режимы источников питания

Наименование цепи питания микросхемы	Не менее, В	Номинал, В	Не более, В	Максимальный ток в цепи, мА	Максимальный ток по группе, А
1 PWR_0V8_CORE	0,80	0,85	0,90	38000	38,00
2 PWR_0V8_BIO	0,78	0,82	0,90	300	2,63
3 PWR_A0V8_PE0	0,78	0,82	0,90	435,43	

Наименование цепи питания микросхемы	Не менее, В	Номинал, В	Не более, В	Максимальный ток в цепи, мА	Максимальный ток по группе, А	
4 PWR_A0V8_PE1	0,78	0,82	0,90	435,43		
5 PWR_A0V8_PE2	0,78	0,82	0,90	435,43		
6 PWR_A0V8_PE3	0,78	0,82	0,90	435,43		
7 PWR_A0V8_SATAETH0	0,78	0,82	0,90	136,48		
8 PWR_A0V8_SATAETH1	0,78	0,82	0,90	136,48		
9 PWR_A0V8_USB0	0,78	0,82	0,90	58,1		
10 PWR_A0V8_USB1	0,78	0,82	0,90	58,1		
11 PWR_A0V8_USB2	0,78	0,82	0,90	58,1		
12 PWR_A0V8_USB3	0,78	0,82	0,90	58,1		
13 PWR_A0V8_GFX1	0,78	0,82	0,90	40		
14 PWR_A0V8_GFX2	0,78	0,82	0,90	40		
15 PWR_MC	1,14	1,2	1,26	7800		7,80
16 PWR_1V8_BIO	1,71	1,8	1,89	400		1,30
17 PWR_A1V8_PE	1,71	1,8	1,89	544,56		
18 PWR_A1V8_MC0_PLL	1,71	1,8	1,89	30		
19 PWR_A1V8_MC1_PLL	1,71	1,8	1,89	30		
20 PWR_A1V8_SATAETH	1,71	1,8	1,89	98		
21 PWR_A1V8_SYNC_TOP	1,71	1,8	1,89	72		
22 PWR_A1V8_SYNC_BOT	1,71	1,8	1,89	95		
23 PWR_A1V8_GFX1	1,71	1,8	1,89	15		
24 PWR_A1V8_GFX2	1,71	1,8	1,89	15		
25 PWR_A3V3_USB	3,07	3,3	3,63	230,864	0,23	
26 PWR_0V8_SUS	0,70	0,8	0,90	200	0,20	
27 PWR_1V8_SUS	1,62	1,8	1,98	100	0,10	
28 PWR_0V8_MC	0,78	0,8	0,90	4200	4,20	
29 PWR_1V5_HDA	1,425	1,5	1,89	30	0,03	
30 PWR_1V8_EFUSE	1,71	1,8	1,89	140	0,14	
31 PWR_A0V8_SYNC_TOP	0,72	0,82	0,88	179	0,20	
32 PWR_A0V8_SYNC_BOT	0,72	0,82	0,88	15		

7.1.4 Подключение цепей земли микросхемы представлено в таблице 7.4.

Таблица 7.4 – Подключение цепей земли микросхемы

Цепи земли микросхемы	Цепь земли МПП
GND_12G	GND
GND_CORE	GND
GND_MC	GND

Цепи земли микросхемы	Цепь земли МПП
GND_USB	GND
GND_BIO	GND
GND_HDMI	GND
GND_SYNC_TOP	GND
GND_SYNC_BOT	GND

7.2 Рекомендации по проектированию модулей на базе микропроцессора

7.2.1 Последовательность включения и выключения питания.

7.2.1.1 Последовательность включения питания.

Включение питания выполняется в 5 этапов. Распределение цепей питания по группам включения приведено в таблице 7.5. Номера групп включения определяют последовательность включения питания. Внутри группы 3 включения напряжения в цепях питания могут нарастать в любом порядке. Внутри группы 4 включения напряжения в цепях питания должны нарастать одновременно. Каждая последующая группа должна запускаться при достижении всех напряжений внутри предыдущей группы минимального порога в 90 % от номинальных значений.

Таблица 7.5 – Распределение цепей питания по группам включения

Группа включения	Обозначение цепей питания	Наименование цепи питания	Примечание
1	U _{CC7}	PWR_1V8_SUS	Скорость нарастания напряжения не должна превышать 18 мВ/мкс. Должны включаться первыми и отключаться последними. Напряжения для этих цепей обычно формируются из напряжения плюс 5 В дежурного питания, который подаётся
2	U _{CC6}	PWR_0V8_SUS	

Группа включения	Обозначение цепей питания	Наименование цепи питания	Примечание
			из блока питания
3	U _{CC3}	PWR_MC	Скорость нарастания напряжения не должна превышать 5 мВ/мкс
	U _{CC9}	PWR_1V5_HDA	Скорость нарастания напряжения не должна превышать 18 мВ/мкс
	U _{CC4}	PWR_1V8_BIO	
		PWR_A1V8_PE	Скорость нарастания напряжения не должна превышать 180 мВ/мкс
		PWR_A1V8_SATAETH	
		PWR_A1V8_MC	Скорость нарастания напряжения не должна превышать 5 мВ/мкс
		PWR_A1V8_SYNC_TOP	Скорость нарастания напряжения не должна превышать 18 мВ/мкс
		PWR_A1V8_SYNC_BOT	
		PWR_A1V8_GFX1	
		PWR_A1V8_GFX2	
U _{CC5}	PWR_A3V3_USB	Скорость нарастания напряжения не должна превышать 100 мВ/мкс	
4	U _{CC1}	PWR_0V8_CORE	Скорость нарастания напряжения во всех цепях питания должна быть одинаковой и не должна превышать 18 мВ/мкс Время между включением питаний в группах 3 и 4 должно быть меньше десятков миллисекунд
	U _{CC2}	PWR_0V8_BIO	
		PWR_A0V8_PE0	
		PWR_A0V8_PE1	
		PWR_A0V8_PE2	
		PWR_A0V8_PE3	
		PWR_A0V8_SATA	
		PWR_A0V8_SATAETH	
		PWR_A0V8_USB0	
	PWR_A0V8_USB1		

Группа включения	Обозначение цепей питания	Наименование цепи питания	Примечание
		PWR_A0V8_USB2	
		PWR_A0V8_USB3	
		PWR_A0V8_GFX1	
		PWR_A0V8_GFX2	
	U _{CC8}	PWR_0V8_MC	
	U _{CC11}	PWR_A0V8_SYNC_TOP	
PWR_A0V8_SYNC_BOT			
5	U _{CC10}	PWR_1V8_EFUSE	Скорость нарастания напряжения не должна превышать 18 мВ/мкс. Питание группы 5 включается только в случае записи в EFUSE

Диаграмма последовательности включения питания показана на рисунке 7.1.

Диаграмма последовательности включения/выключения питания

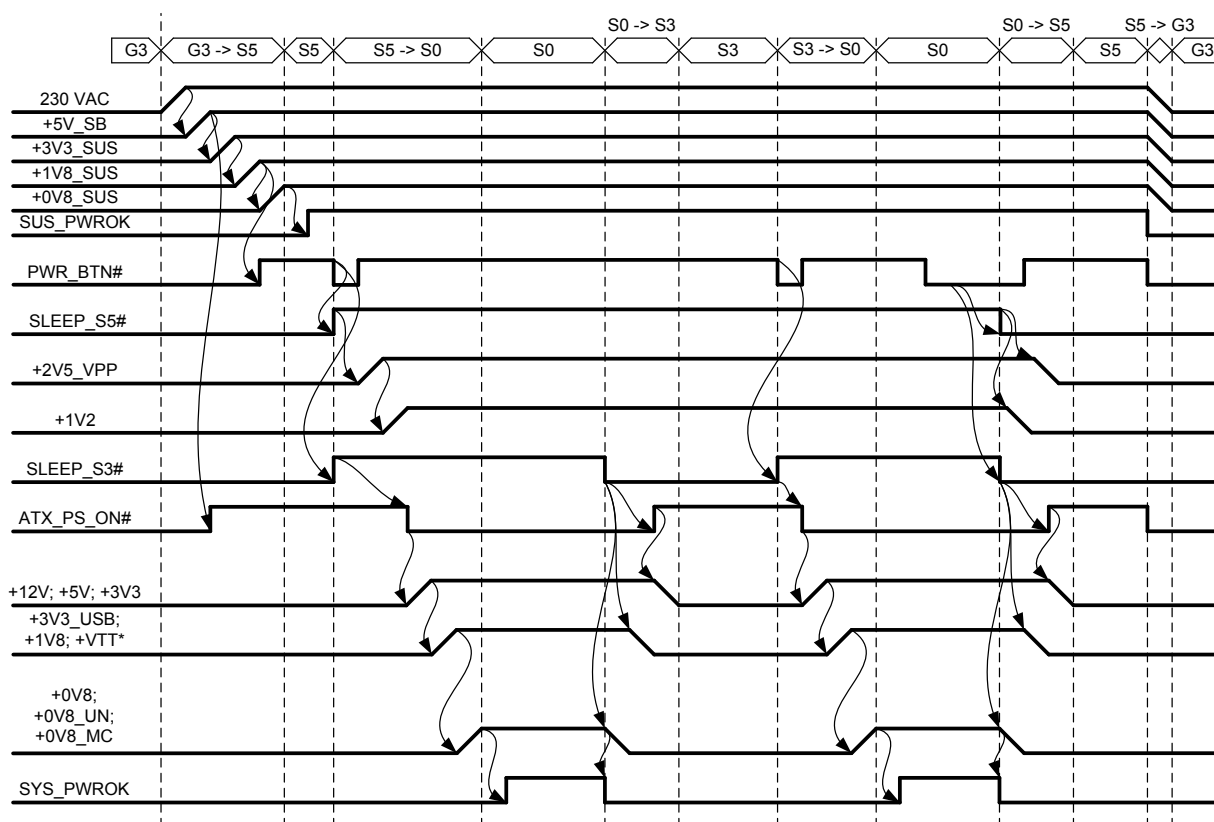


Рисунок 7.1 - Диаграмма последовательности включения/выключения питания

7.2.2 Последовательность выключения питания.

Последовательность выключения питания обратна последовательности включения питания. Внутри группы 3 напряжения в цепях питания могут спадать в любом порядке. Внутри группы 4 напряжения в цепях питания должны спадать одновременно. Каждая последующая группа должна выключаться при достижении всех напряжений внутри предыдущей группы максимального порога в 10 % от номинальных значений.

Диаграмма последовательности выключения питания показана на рисунке 7.1.

7.2.3 Особенности включения/выключения PWR_1V8_EFUSE.

Цепь питания PWR_1V8_EFUSE всегда должна включаться самой последней из всех цепей питания непосредственно перед программированием блока однократно программируемого ПЗУ на кристалле типа eFuse. Сразу после программирования это питание должно отключаться. Программирование блока eFuse происходит в стенде тестирования и разбраковки микропроцессора, в серийных платах это питание должно быть подключено к GND.

7.2.4 Поддержка питания микропроцессора в состояниях энергосбережения.

Поддержка питания микропроцессора и состояние сигналов завершения включения питания на его входах SUS_PWROK и SYS_PWROK в зависимости от состояния энергосбережения на уровне вычислительного комплекса приведена в таблице 7.6. Сигнал SUS_PWROK должен переключаться в высокий уровень только после того, как в норму придут напряжения из 1 и 2 групп включения. Сигнал SYS_PWROK должен переключаться в высокий уровень только после того, как в норму придут напряжения из 3 и 4 групп включения.

Таблица 7.6 – Поддержка питания микропроцессора в зависимости от состояния энергосбережения

Состояние вычислительного комплекса	Поддержка питания	Состояние входов
Режим конфигурации блока eFuse	Все группы	SUS_PWROK = 1, SYS_PWROK = 1
Нормальная работа (S0)	Все группы, кроме группы 5	SUS_PWROK = 1, SYS_PWROK = 1
Ждущий режим (S3)	Группы 1, 2, цепь U _{CC3} из группы 3	SUS_PWROK = 1, SYS_PWROK = 0
Спящий режим (S4)	Группы 1, 2	SUS_PWROK = 1, SYS_PWROK = 0
Программное выключение (S5)	Группы 1, 2	SUS_PWROK = 1, SYS_PWROK = 0
Механическое выключение (G3)	Нет	SUS_PWROK = 0, SYS_PWROK = 0

7.2.5 Проектирование система питания.

7.2.5.1 На рисунке 7.2 показан пример построения системы питания.

FBC – фильтр на входах аналогового питания, состоящий из:

- индуктивности ферритовой BLM15AX100SN1D или аналогичной с импедансом не менее 10 Ом и сопротивлением не более 25 мОм в корпусе 0402;
- конденсатора керамического 0402ZD105KAT или аналогичного с параметрами не хуже: 1 мкФ 6.3В X5R в корпусе 0402;
- конденсаторов керамических GRM033R60J224ME15D или аналогичных с параметрами не хуже: 0,22 мкФ 6.3В X5R в корпусе 0201, по одному на каждый вход питания.

Система питания

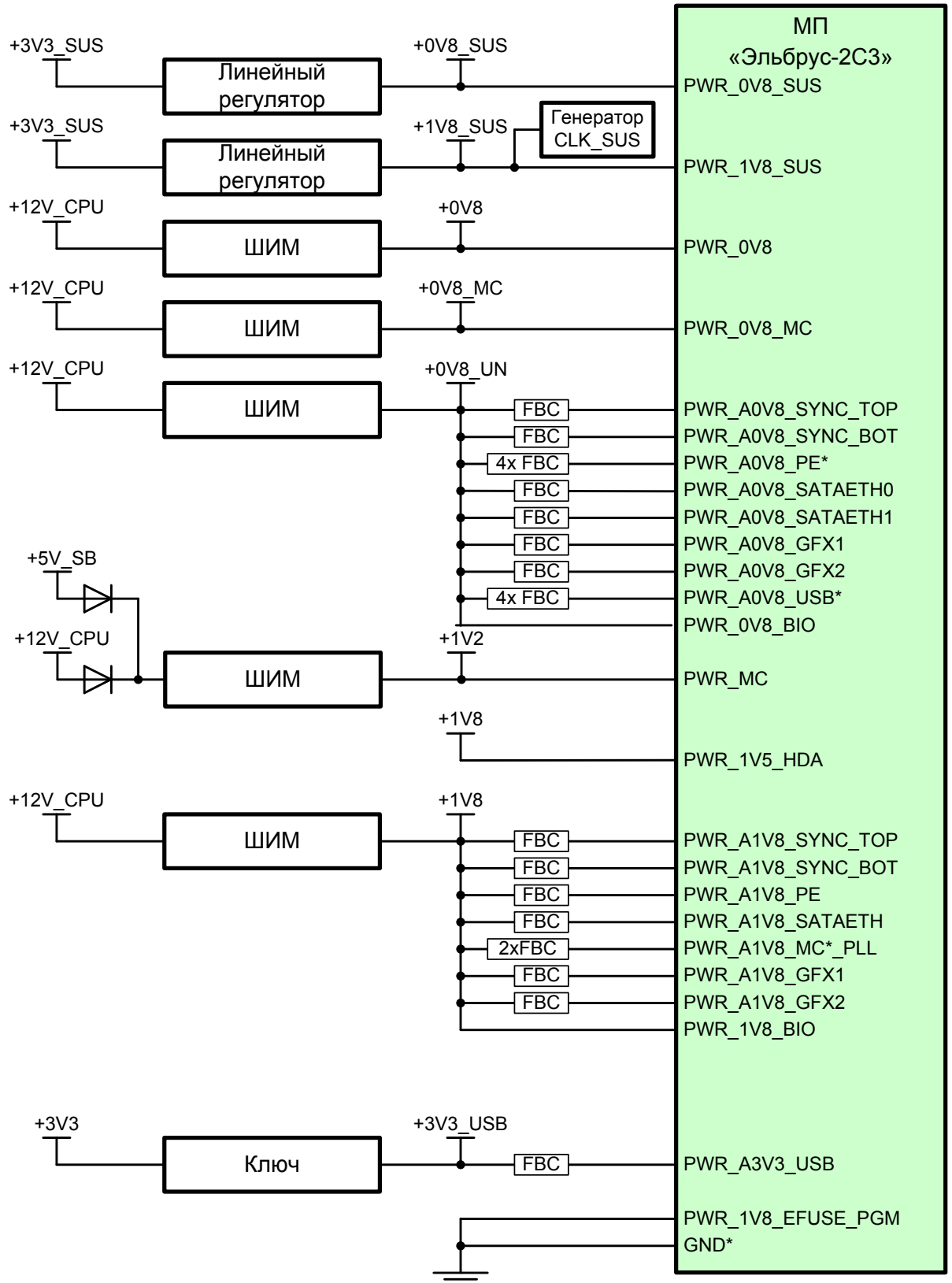


Рисунок 7.2 - Пример построения системы питания

7.2.5.2 Реализация обратной связи источников питания и контроль номинальных значений напряжения питания микропроцессора.

Для цепей питания +0V8, +0V8_MC, +0V8_UN, +1V2 необходимо применять источники питания с поддержкой дистанционного измерения напряжения на нагрузке. Значения номиналов питаний, указанных в таблице 7.3, источник питания должен поддерживать на выделенных точках, описанных ниже.

Для цепи питания +0V8 необходимо для реализации обратной связи и контроля значения напряжения использовать пару выводов микросхемы AU12 – GND_0V8 и AR12 – PWR_0V8.

Эти выводы на материнской плате не подключать к полигонам питания, а использовать непосредственно для подключения к дифференциальным входам источника.

Для цепи питания +0V8_MC для реализации обратной связи и контроля значения напряжения использовать пару выводов микросхемы AL24 –PWR_0V8_MC и AM23 – GND_MC.

Эти выводы на материнской плате не подключать к полигонам питания, а использовать непосредственно для подключения к дифференциальным входам источника. При неиспользовании этих выводов под обратную связь их необходимо подключить к полигону земли и питания.

Для цепи питания +0V8_UN для реализации обратной связи выбрать FBC, подключенный к одной из шин питания PWR_A0V8_PE*, PWR_A0V8_SATAETH*. При выборе FBC необходимо, чтобы данный интерфейс выбранной шины питания был использован на МПП. Для удобства разработчиков материнских плат можно выбрать любой из используемых FBC. Подключение дифференциальной связи необходимо осуществить следующим образом. Один проводник подключается к питанию ферритовой индуктивности со стороны источника питания, второй проводник подключается к земляному выводу рядом стоящего конденсатора, данные контакты необходимо использовать непосредственно для подключения к дифференциальным входам источника. Контроль значения напряжения питания необходимо осуществлять на паре выводов земли и

питания микропроцессора, подключенных к FBC на которой реализовано подключение обратной связи.

Для цепи питания +1V2 для реализации обратной связи выбрать один из контактов питания ближайшего к микропроцессору слота канала оперативной памяти. Один проводник подключается к выводу питания, второй к выводу земли, располагаемый в непосредственной близости. Эти контакты необходимо использовать непосредственно для подключения к дифференциальным входам источника. Контроль значения напряжения питания осуществлять на двух парах выводов V39 и W38, T9 и U10.

Для всех остальных шин питания использовать локальное измерение, и обратную связь брать на выходных конденсаторах источника питания, расположенных в непосредственной близости к ИП.

7.2.6 Проектирование системы сбросов.

7.2.6.1 Пример схемы системы сбросов показан на рисунке 7.3.



Рисунок 7.3 – Пример схемы системы сбросов

Сигнал SUS_PWROK должен переключаться в высокий уровень минимум через 10 мс после того, как в норму придут напряжения из групп включения 1 и 2 и синхросигнал SUS_CLK. Сигнал SYS_PWROK должен переключаться в высокий уровень минимум через 10 мс после того, как в норму придут напряжения из групп включения 3 и 4 и все синхросигналы микропроцессора.

При нажатии кнопки сброса требуется сбрасывать сигнал SYS_PWROK, чтобы плата (ее основное питание) не выключалась.

7.2.7 Порядок включения и выключения синхросигналов.

7.2.7.1 В схеме подключения МП всегда должен присутствовать источник синхросигнала SUS_CLK_14М. Требования 1) – 5), связанные с синхросигналом SUS_CLK_14М, должны выполняться безусловно:

1) запрещается включать любое питание во всей схеме, если может быть нарушено хотя бы одно из требований 2) – 5). В частности, запрещается включать любое питание во всей схеме, если источник синхросигнала SUS_CLK_14М отсутствует или неисправен. Нарушение этого запрета может привести к отказу МП;

2) при выключенном питании PWR_1V8_SUS, источник синхросигнала SUS_CLK_14М должен быть выключен. Источник синхросигнала SUS_CLK_14М считается выключенным только при выполнении одного из двух следующих условий. Источник синхросигнала SUS_CLK_14М считается выключенным, если выключено его питание. Источник синхросигнала SUS_CLK_14М также считается выключенным, если его выход находится в высокоимпедансном состоянии.

3) включение питания PWR_1V8_SUS должно приводить к включению источника синхросигнала SUS_CLK_14М. Источник синхросигнала SUS_CLK_14М считается включенным только после того, как все параметры этого синхросигнала пришли в норму. Питание PWR_0V8_SUS должно включаться только после того, как включился источник синхросигнала SUS_CLK_14М;

4) Пока питание PWR_1V8_SUS остается включенным, все параметры синхросигнала SUS_CLK_14М должны оставаться в норме. Для этого необходимо,

чтобы источник синхросигнала SUS_CLK_14M оставался включенным, пока включено питание PWR_1V8_SUS.

5) выключение питания PWR_1V8_SUS должно приводить к выключению источника синхросигнала SUS_CLK_14M.

7.2.7.2 В схеме подключения МП всегда должен присутствовать источник синхросигнала CLK_REF_100M_TOP_P/N. Требования 1) – 5), связанные с синхросигналом CLK_REF_100M_TOP_P/N, должны выполняться безусловно:

1) запрещается включать любое питание во всей схеме, если может быть нарушено хотя бы одно из требований 2) – 5). В частности, запрещается включать любое питание во всей схеме, если источник синхросигнала CLK_REF_100M_TOP_P/N отсутствует или неисправен. Нарушение этого запрета может привести к отказу МП;

2) при выключенном питании PWR_A1V8_SYNC_TOP, источник синхросигнала CLK_REF_100M_TOP_P/N должен быть выключен. Источник синхросигнала CLK_REF_100M_TOP_P/N считается выключенным только при выполнении одного из двух следующих условий. Источник синхросигнала CLK_REF_100M_TOP_P/N считается выключенным, если выключено его питание. Источник синхросигнала CLK_REF_100M_TOP_P/N также считается выключенным, если его выход находится в высокоимпедансном состоянии;

3) источник синхросигнала CLK_REF_100M_TOP_P/N и питание PWR_A0V8_SYNC_TOP должны включаться после появления питания PWR_A1V8_SYNC_TOP и до переключения в высокий уровень сигнала SYS_PWROK. Источник синхросигнала CLK_REF_100M_TOP_P/N считается включенным только после того, как все параметры этого синхросигнала пришли в норму;

4) пока питание PWR_A1V8_SYNC_TOP остается включенным, все параметры синхросигнала CLK_REF_100M_TOP_P/N должны оставаться в норме. Для этого необходимо, чтобы источник синхросигнала CLK_REF_100M_TOP_P/N оставался включенным, пока включено питание PWR_A1V8_SYNC_TOP;

5) выключение питания PWR_A1V8_SYNC_TOP должно приводить к выключению источника синхросигнала CLK_REF_100M_TOP_P/N.

7.2.7.3 В схеме подключения МП всегда должен присутствовать источник синхросигнала CLK_REF_100M_BOT_P/N. Требования 1) – 5), связанные с синхросигналом CLK_REF_100M_BOT_P/N, должны выполняться безусловно:

1) запрещается включать любое питание во всей схеме, если может быть нарушено хотя бы одно из требований 2) – 5). В частности, запрещается включать любое питание во всей схеме, если источник синхросигнала CLK_REF_100M_BOT_P/N отсутствует или неисправен. Нарушение этого запрета может привести к отказу МП;

2) при выключенном питании PWR_A1V8_SYNC_BOT, источник синхросигнала CLK_REF_100M_TOP_P/N должен быть выключен. Источник синхросигнала CLK_REF_100M_TOP_P/N считается выключенным только при выполнении одного из двух следующих условий. Источник синхросигнала CLK_REF_100M_TOP_P/N считается выключенным, если выключено его питание. Источник синхросигнала CLK_REF_100M_TOP_P/N также считается выключенным, если его выход находится в высокоимпедансном состоянии;

3) источник синхросигнала CLK_REF_100M_TOP_P/N и питание PWR_A0V8_SYNC_BOT должны включаться после появления питания PWR_A1V8_SYNC_BOT и до переключения в высокий уровень сигнала SYS_PWROK. Источник синхросигнала CLK_REF_100M_TOP_P/N считается включенным только после того, как все параметры этого синхросигнала пришли в норму;

4) пока питание PWR_A1V8_SYNC_BOT остается включенным, все параметры синхросигнала CLK_REF_100M_TOP_P/N должны оставаться в норме. Для этого необходимо, чтобы источник синхросигнала CLK_REF_100M_TOP_P/N оставался включенным, пока включено питание PWR_A1V8_SYNC_BOT;

5) Выключение питания PWR_A1V8_SYNC_BOT должно приводить к выключению источника синхросигнала CLK_REF_100M_TOP_P/N.

7.2.7.4 Синхросигнал CLK_REF_156M_P/N требуется для работы МП во всех конфигурациях с каналами ETH. Этот синхросигнал используется тогда, когда вход SATAETH_CONFIG находится в состоянии лог. «1».

Если в схеме подключения МП не предусмотрена работа МП в конфигурации с каналами ETH, и вход SATAETH_CONFIG подключен к проводнику GND печатной платы, то источник синхросигнала CLK_REF_156M_P/N может быть исключен из схемы подключения МП.

Если источник синхросигнала CLK_REF_156M_P/N исключен из схемы подключения МП, то входы CLK_REF_156M_P/N должны быть подключены к тому проводнику GND печатной платы, к которому подключены выводы GND_SYNC_BOT. Запрещается включать любые питания МП, если источник синхросигнала CLK_REF_156M_P/N исключен из схемы подключения МП, и хотя бы один из входов CLK_REF_156M_P/N не подключен к указанному проводнику GND.

Требования 1) – 5), связанные с синхросигналом CLK_REF_156M_P/N, должны выполняться при условии, что источник синхросигнала CLK_REF_156M_P/N не исключен из схемы подключения микропроцессора.

1) запрещается включать любое питание во всей схеме, если может быть нарушено хотя бы одно из требований 2) – 5). В частности, запрещается включать любое питание во всей схеме, если источник синхросигнала CLK_REF_156M_P/N отсутствует или неисправен. Нарушение этого запрета может привести к отказу микропроцессора;

2) при выключенном питании PWR_A1V8_SYNC_BOT, источник синхросигнала CLK_REF_156M_P/N должен быть выключен. Источник синхросигнала CLK_REF_156M_P/N считается выключенным только при выполнении одного из двух следующих условий. Источник синхросигнала CLK_REF_156M_P/N считается выключенным, если выключено его питание. Источник синхросигнала CLK_REF_156M_P/N также считается выключенным, если его выход находится в высокоимпедансном состоянии;

3) источник синхросигнала CLK_REF_156M_P/N и питание PWR_A0V8_SYNC_BOT должны включаться после появления питания PWR_A1V8_SYNC_BOT и до переключения в высокий уровень сигнала SYS_PWROK. Источник синхросигнала CLK_REF_156M_P/N считается включенным только после того, как все параметры этого синхросигнала пришли в норму;

4) пока питание PWR_A1V8_SYNC_BOT остается включенным, все параметры синхросигнала CLK_REF_156M_P/N должны оставаться в норме. Для этого необходимо, чтобы источник синхросигнала CLK_REF_156M_P/N оставался включенным, пока включено питание PWR_A1V8_SYNC_BOT;

5) выключение питания PWR_A1V8_SYNC_BOT должно приводить к выключению источника синхросигнала CLK_REF_156M_P/N.

7.2.8 Проектирование конфигураций памяти.

7.2.8.1 Микропроцессор Эльбрус-2С3 имеет два канала памяти MC0 и MC1, каждый канал поддерживает работу с одно-, двух- и четыреххранковыми модулями памяти ECC RDIMM, UDIMM, LRDIMM, 3DS в одно- и двухслотовом режиме.

Примечание – Двухслотовый режим поддерживается для всех типов модулей памяти, кроме четыреххранковых.

Допускается установка модулей памяти в любые соединители любых каналов памяти микропроцессора.

Выводы каналов памяти расположены по двум углам микропроцессора.

Для каждого канала используется отдельный блок формирования частоты, поэтому для каналов настройка частоты работы независимая.

На рисунке 7.4 изображена схема взаимного расположения областей выводов каналов памяти кристалла и корпуса микропроцессора.

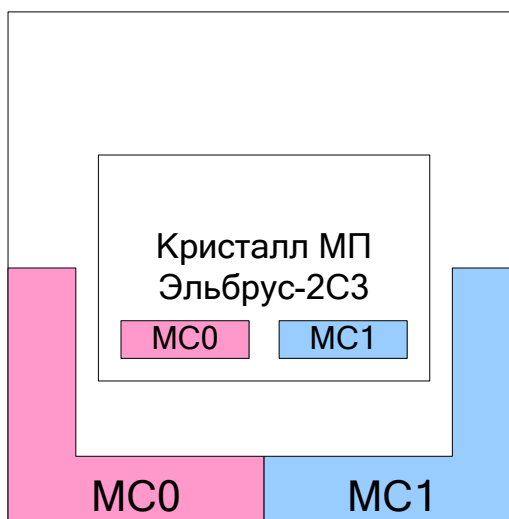


Рисунок 7.4 – Схема взаимного расположения областей выводов каналов памяти кристалла и корпуса микропроцессора

Кроме того, топологический дизайн микропроцессора в части трассировки каналов памяти имеет ряд особенностей, которые могут быть использованы для получения оптимальных характеристик каналов памяти изделий на основе микропроцессора.

На рисунках 7.5 и 7.6 изображены рекомендуемые схемы подключения соединителей (слотов) для модулей памяти к каналам памяти микропроцессора.

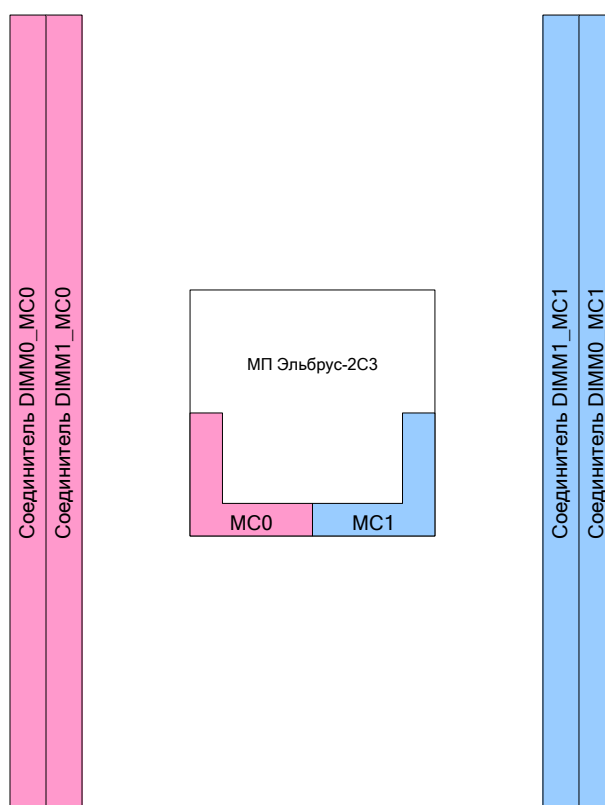


Рисунок 7.5 – Рекомендуемая схема № 1 подключения соединителей для модулей памяти к микропроцессору

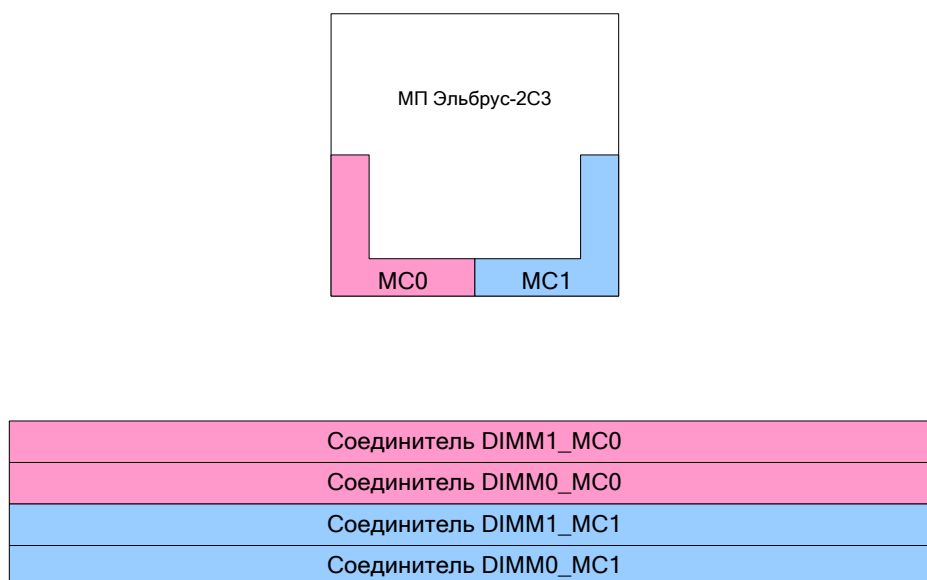


Рисунок 7.6 – Рекомендуемая схема № 2 подключения соединителей для модулей памяти к микропроцессору

Рекомендуемые характеристики топологических связей каналов памяти печатной платы на базе микропроцессора приведены в таблице 7.7.

Таблица 7.7 – Рекомендуемые характеристики топологических связей каналов памяти печатной платы на основе микропроцессора

Описание характеристики	Величина характеристики
Номинальное волновое сопротивление одиночного проводника, Z_{single} , Ом	от 45 до 50
Номинальное волновое сопротивление дифференциальной пары проводников, $Z_{different}$, Ом	от 90 до 95
Ширина проводника, W , мкм, не менее	100
Зазор между проводниками сигналов DQ, не менее*	$(2,5-3,0) \cdot H^{**}$
Зазор между проводниками сигналов DQS/CLK и другими сигналами, не менее*	$4 \cdot H$
Максимальная длина связи, мм, не более	100
Допустимые отклонения длин проводников, мм, не более:	
DQ к DQ внутри одного байта (полубайта) канала памяти	1,5
DQ к DQS	1,0
DQS к CLK	30,0
АС (сигналы адресной шины) к CLK	2,0
Длина свободной части переходного отверстия (STUB) для каждого из двух отверстий, мм, не более	1,5
* Допускается уменьшение зазоров между проводниками в ограниченных по длине узких местах.	
** Где H – высота диэлектрика до опорного полигона.	

Для обеспечения максимальной производительности оперативной памяти в конечном изделии рекомендуется проектировать сбалансированные конфигурации памяти, такие, в которых трассировка одного канала памяти ведется в двух слоях (минимизации помех и неоднородностей), а в части размещения модулей памяти применяются следующие правила:

- количество задействованных каналов памяти: один или два;

- все задействованные каналы памяти должны иметь одинаковое количество установленных модулей памяти (один модуль на канал или два модуля на канал), и эти модули должны быть одинаковыми.

Примечание – Несбалансированные в части размещения модулей памяти конфигурации могут существенно уменьшить производительность конечного изделия.

Таблица 7.8 иллюстрирует применение правил проектирования сбалансированной в части производительности конфигурации.

DIMM0, DIMM1 – обозначения расположения установленных модулей памяти, причем DIMM0 устанавливается в соединитель, находящийся на большем расстоянии от микропроцессора. Взаимное расположение соединителей для модулей памяти соответствует схеме, приведенной на рисунке 7.5.

Таблица 7.8 – Рекомендуемые конфигурации памяти на базе микропроцессора

Количество задействованных модулей памяти	Обозначение канала памяти процессора	
	MC0	MC1
1	DIMM0	
2	DIMM0	DIMM0
4	DIMM0, DIMM1	DIMM0, DIMM1

Для обеспечения возможности подключения четырехканальных модулей памяти в однослотовом режиме рекомендуется использовать схемы, приведенные на рисунках 7.7 и 7.8.

Примечание – Схема подключения, приведенная на рисунке 7.5, может быть использована для подключения всех типов модулей памяти, за исключением 3DS.

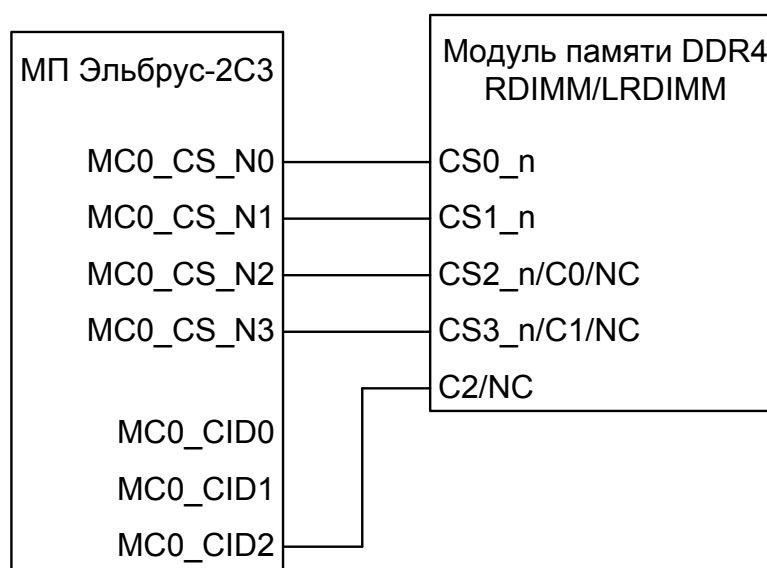


Рисунок 7.7 – Рекомендуемая схема № 1 подключения четырехканальных модулей памяти к микропроцессору в однослотовом режиме (за исключением модулей памяти 3DS)

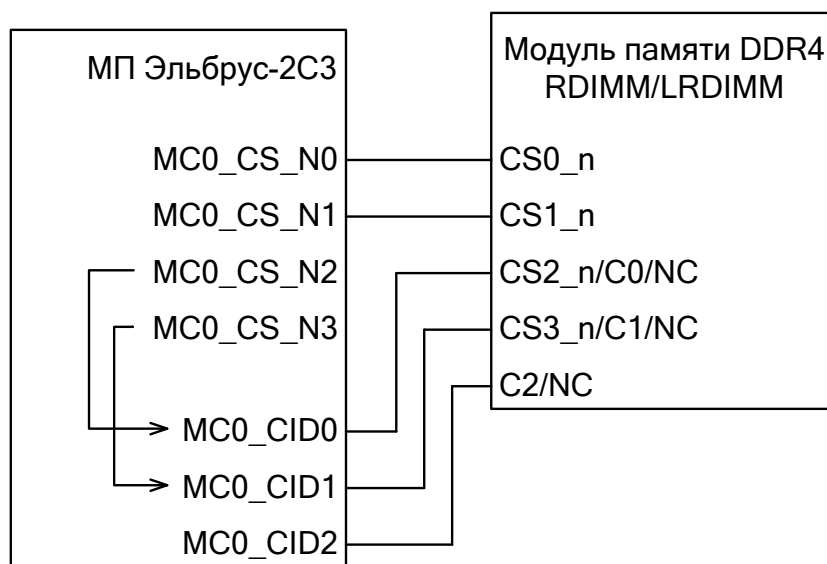


Рисунок 7.8 – Рекомендуемая схема № 2 подключения четырехранковых модулей памяти к микропроцессору в однослотовом режиме

Для обеспечения возможности подключения четырехранковых модулей памяти в двухслотовом режиме рекомендуется использовать схему, приведенную на рисунке 7.9.

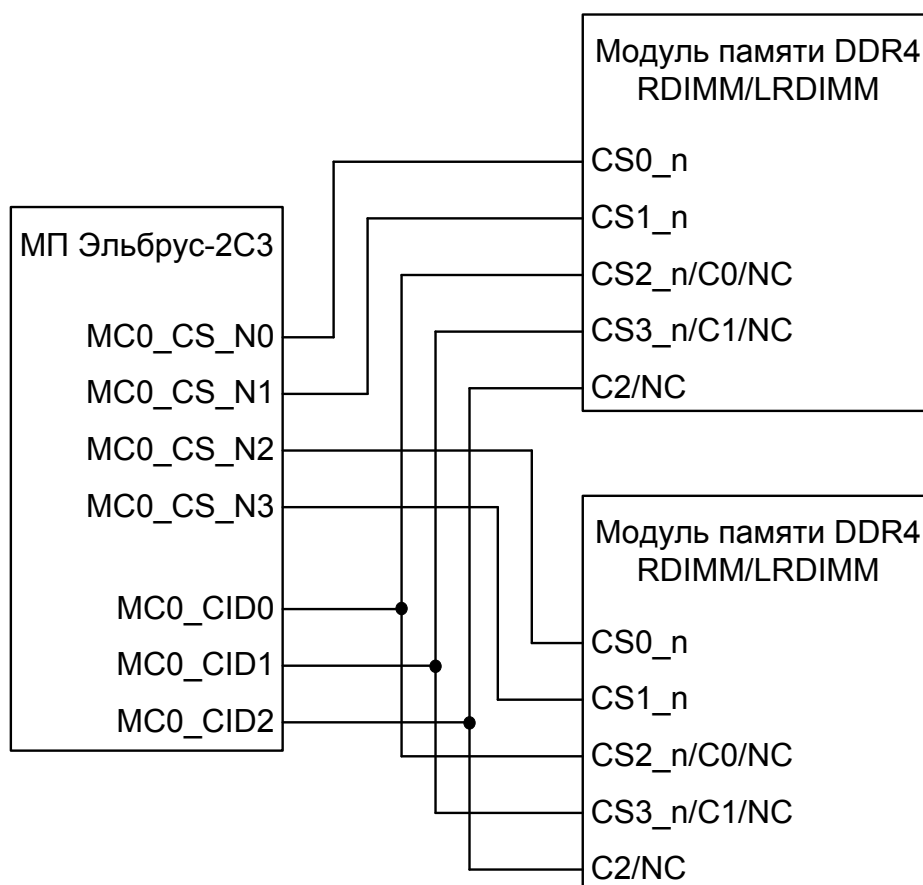


Рисунок 7.9 – Рекомендуемая схема подключения четырехранковых модулей памяти к микропроцессору в двухслотовом режиме

При проектировании следует учитывать, что суммарная пиковая скорость работы памяти конечного изделия на основе микропроцессора зависит от примененных типов модулей памяти и оптимальности дизайна, а также от количества модулей памяти, подключенных к каналу. Суммарная пиковая скорость работы памяти в зависимости от типа модулей и конфигурации памяти указана в таблице 7.9.

Таблица 7.9 – Суммарная пиковая скорость работы памяти в зависимости от типа модулей и конфигурации памяти

Тип модулей памяти	Количество ранков на модуль	Частота модуля памяти, МГц, не менее	Конфигурация памяти	Частота канала памяти, МГц		Суммарная пиковая скорость работы памяти, Гбайт/с
				Обозначение канала памяти		
				МС0	МС1	
1	2	3	4	5	6	7
RDIMM	1R	3200	Двухслотовый режим, один модуль памяти	3200	3200	51,2
			Двухслотовый режим, два модуля памяти	2600	2600	41,6
	2R	3200	Двухслотовый режим, один модуль памяти	3200	3200	51,2
			Двухслотовый режим, два модуля памяти	2600	2600	41,6

7.2.8.1 Особенности подключения сигналов МСj_ALERT_N.

Размещение микросхем памяти на материнской плате.

Утягивающий резистор 50 Ом к верхнему уровню (pull-up) должен быть установлен в конце линии Alert_N, после последней микросхемы SDRAM, как показано на рисунке 7.10.



Рисунок 7.10 - Схема подключения сигналов MCj_ALERT_N в случае размещения микросхем памяти на материнской плате

Использование модулей DIMM.

Утягивающий резистор 50 Ом к верхнему уровню (pull-up) должен быть установлен на линии ALERT_N в непосредственной близости к микропроцессору, как показано на рисунке 7.11.

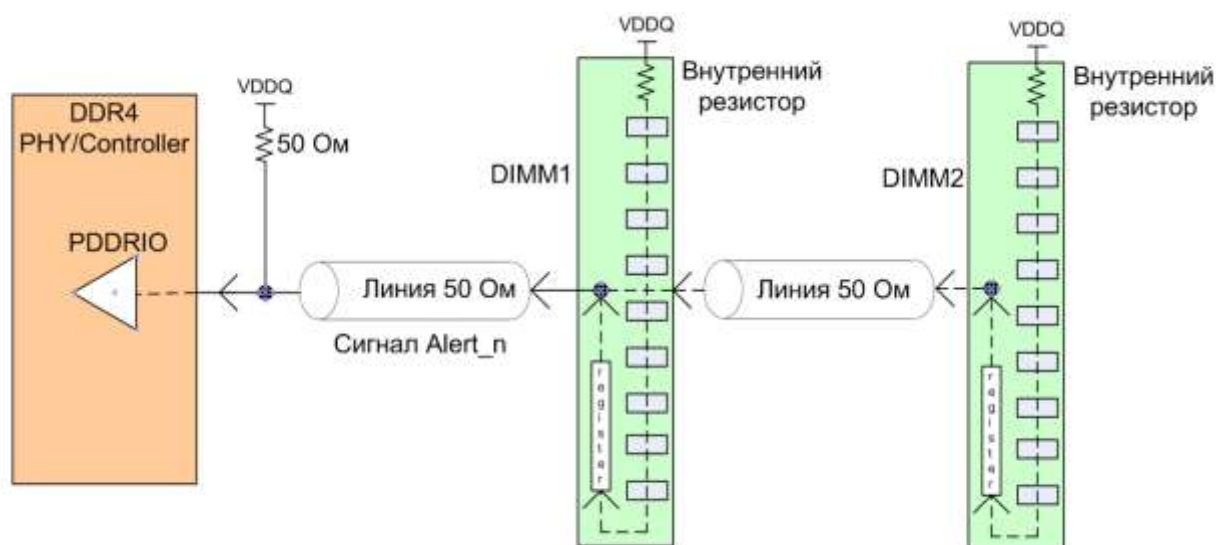


Рисунок 7.11 - Схема подключения сигналов MCj_ALERT_N в случае использования модулей DIMM

7.2.9 Проектирование синхронизации микропроцессора.

7.2.9.1 На рисунке 7.12 показан пример построения синхронизации микропроцессора.

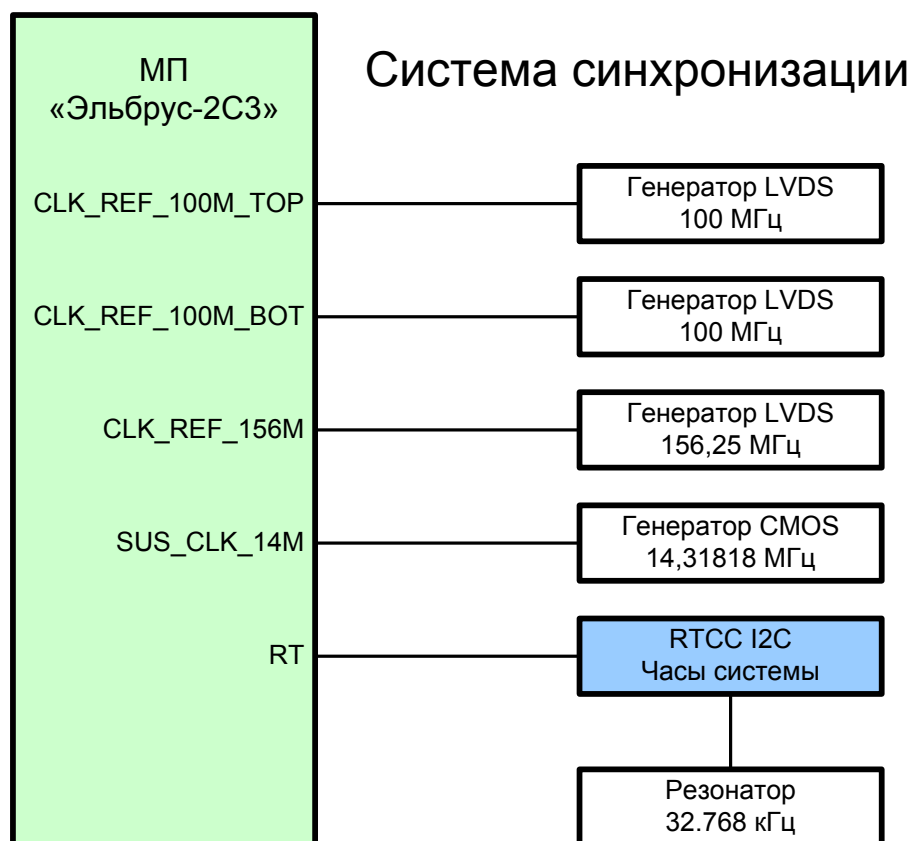


Рисунок 7.12 – Пример построения синхронизация микропроцессора

Требования к источнику синхросигнала CLK_REF_100M:

- частота 100 МГц;
- общая нестабильность частоты ± 50 ppm (с учетом диапазона температур, диапазона напряжений и эксплуатации в течение 10 лет);
- интегральный случайный джиттер (RJ) в полосе частот от 12 КГц до 20 МГц – 0,7 ps rms.

Требования к источнику синхросигнала CLK_REF_156M:

- частота 156,25 МГц;
- общая нестабильность частоты ± 50 ppm (с учетом диапазона температур, диапазона напряжений и эксплуатации в течение 10 лет);
- интегральный случайный джиттер (RJ) в полосе частот от 12 КГц до 20 МГц – 0,7 ps rms.

7.2.10 Рекомендованное посадочное место микропроцессора.

7.3 Стойкость к внешним воздействиям

7.3.1 Стойкость к воздействию механических факторов.

7.3.1.1 Стойкость к воздействию механических факторов приведена в таблице 7.10.

Таблица 7.10 - Стойкость к воздействию механических факторов

Параметры воздействующего фактора, единица измерения	Значение воздействующего фактора
1 Синусоидальная вибрация:	
- диапазон частот, Гц	1- 5000
- амплитуда ускорения, м/с ² (g)	400 (40)
2 Удары одиночного действия в любом направлении	
- амплитуда пикового ударного ускорения, м/с ² (g)	15000 (1500)
- длительность действия ударного ускорения, мс	0,1 – 2,0
3 Атмосферное пониженное рабочее давление	6,65*10 ² Па (5 мм рт. ст.)

7.3.2 Стойкость к воздействию климатических факторов.

7.3.2.1 Стойкость к воздействию климатических факторов приведена в таблице 7.11.

Таблица 7.11 - Стойкость к воздействию климатических факторов

Параметры воздействующего фактора, единица измерения	Значение воздействующего фактора
1 Атмосферное пониженное рабочее давление, Па	6,65 × 10 ²
2 Повышенное рабочее давление, кПа	294
3 Повышенная температура, °С:	
- рабочая температура корпуса	90
- предельная температура среды	125

Параметры воздействующего фактора, единица измерения	Значение воздействующего фактора
4 Пониженная температура среды, °С: - рабочая - предельная	минус 40 минус 60
5 Смена температур, °С: - от пониженной предельной температуры среды - до повышенной предельной температуры среды	минус 60 125
6 Повышенная относительная влажность при 35 °С, %	98

7.4 Указания по применению и эксплуатации

7.4.1 Устанавливать и извлекать микросхемы из контактных приспособлений, а также производить их замену необходимо только после снятия напряжений со всех выводов микросхем.

7.4.2 При эксплуатации и испытаниях микросхем, в том числе в моменты включения, выключения, при переходных процессах, при изменении режимов работы должны быть приняты меры по исключению замыканий (даже кратковременных) цепей питания или нагрузочных элементов на выводах микросхем с другими выводами микросхем или другими элементами печатной платы.

7.4.3 После раскрытия упаковки обращаться с микросхемами в следующих производственных условиях: температуре от плюс 20 до плюс 30 °С и относительной влажности не более 60 %.

Микросхемы рекомендуется устанавливать в аппаратуру в течение 168 часов после вскрытия пакета. Если в аппаратуру устанавливается только часть партии, остальные микросхемы рекомендуется повторно упаковать или поместить в шкаф сухого хранения в течение часа с момента вскрытия пакета.

Упаковка должна включать осушитель и индикатор влажности, помещенные вместе с микросхемами во влагонепроницаемый пакет. Не допускается

помещать в пакет поглощающие влагу материалы (например, поддоны, паллеты) без их предварительной сушки.

Пакет должен допускать герметизацию температурным воздействием. Рекомендуется использовать пакеты из влагонепроницаемого материала с толщиной пленки не менее 90 мкм, обеспечивающие механическую прочность, гибкость, защиту от статического электричества, стойкость к проколам.

Осушитель рекомендуется использовать не коррозионный, беспыльный, упакованный во влагонепроницаемый пакет и обладающий абсорбирующими свойствами. Количество вещества в осушителе, поглощающего влагу, должно обеспечивать внутри упаковки с микросхемами относительную влажность менее 20% при температуре плюс 25 °С.

Индикатор влажности, как правило, представляет собой полоску из впитывающей влагу бумаги, снабженную цветовыми точками – индикаторами уровней относительной влажности от 5 до 60 %. Уровень влажности определяется по изменению цвета соответствующих точек согласно маркировке индикатора.

Не упакованные микросхемы могут храниться в шкафу сухого хранения при температуре от плюс 20 до плюс 35 °С и относительной влажности не более 10 %. Шкаф должен восстанавливать внутренний уровень влажности в течение часа после открывания / закрывания двери.

Упакованные микросхемы рекомендуется хранить при температуре от плюс 20 до плюс 40 °С и относительной влажности не более 90 % в условиях отсутствия конденсации. При этом контролировать показания вложенного индикатора влажности не менее чем раз в 12 месяцев с даты упаковки.

Если при открытии пакета показания вложенного индикатора влажности превышают 20 % или не соблюдены условия, представленные выше, то перед установкой в аппаратуру или перед упаковкой микросхемы должны быть подвергнуты горячей сушке (термообработке):

- в течение 48 ч при температуре 125 (+10; -0) °С и относительной влажности не более 5 %;

- в течение 240 ч при температуре 90 (+8; -0) °С и относительной влажности не более 5 % при условии, что время нахождения микросхем в неупакованном виде или вне шкафа сухого хранения не превышало 240 ч;

- в течение 1896 ч при температуре 40 (+5; -0) °С и относительной влажности не более 5 % при условии, что время нахождения микросхем в неупакованном виде или вне шкафа сухого хранения не превышало 240 ч.

Примечания

1 Перед упаковкой допускается проводить горячую сушку (термообработку) в течение 24 ч при температуре 150 (+10; -0) °С и относительной влажности не более 5 %.

2 Не допускается проводить сушку микросхем в носителях (поддонах, паллетах) с предельно-допустимой температурой ниже, чем установленная температура сушки.

Не допускается проводить сушку микросхем в носителях, не имеющих маркировки предельно-допустимой для них рабочей температуры.

7.4.4 Монтаж (пайку) микросхемы и всех электрорадиоизделий, размещаемых совместно с микросхемой на одной печатной плате, производить в соответствии с типовым технологическим процессом монтажа, согласованным с АО «МЦСТ»:

- в процессе подготовки производства, монтаж микросхем проводить с учетом рекомендаций по пайке компонентов типа BGA с безсвинцовыми шариковыми выводами, приведенных в ГОСТ Р 56427 и JEDEC-STD;

- рекомендуемый профиль пайки приведен на рисунке 7.13.

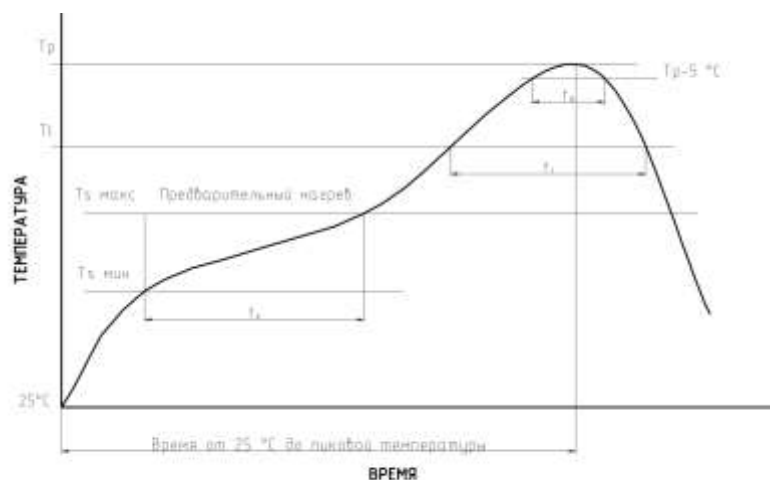


Рисунок 7.13 - Профиль пайки

В таблице 7.12 приведены параметры профиля пайки.

Таблица 7.12 - Параметры профиля пайки

Параметр	Значение
Предварительный нагрев	
Минимальная температура нагрева T_s мин., °C	150
Максимальная температура нагрева T_s макс., °C	200
Время достижения от T_s мин до T_s макс, t_s , с	60 – 120
Скорость нарастания температуры от T_l до T_r , °C/с	не более 3
Температура плавления T_l , °C	217
Время нахождения припоя в расплавленном состоянии, t_l , с	60 – 150
Пиковая температура корпуса микросхемы T_r , °C	260
Время нахождения в пределах 5 °C от пиковой температуры корпуса T_r , t_r , с	не более 30
Скорость снижения температуры от T_r до T_l , °C/с	не более 6
Время нарастания температуры от 25 °C до T_r , мин	не более 8
Примечания	
1 Температура снимается с центра крышки корпуса микросхемы.	
2 Указанные параметры являются справочными, приведены для подготовки производства и разработки технологического процесса изготовления конкретного изделия в части монтажа микросхемы.	
3 Монтаж (пайку) микросхемы и всех электрорадиоизделий, размещаемых совместно с микросхемой на одной печатной плате, производить в соответствии с типовым технологическим процессом монтажа, согласованным с АО «МЦСТ».	

7.4.6 Микросхемы необходимо покрывать защитным лаком в случае использования в условиях повышенной влажности.

7.4.7 Питание микросхем должно осуществляться от стабилизированных источников напряжений питания с отклонением выходного напряжения в пределах $\pm 1,5\%$.

7.4.8 Микросхемы после снятия с эксплуатации подлежат утилизации в порядке и методами, устанавливаемыми в контракте на поставку, в соответствии с действующими нормативными документами.

7.5 Справочные данные

7.5.1 Гамма-процентная наработка T_γ при $\gamma = 97,5\%$ в режимах и условиях эксплуатации, допускаемых ТУ, при температуре окружающей среды не более $(65 + 5)^\circ\text{C}$, составляет 200000 ч.

Прогнозируемая зависимость надежности от температуры кристалла приведена на рисунке 7.14.

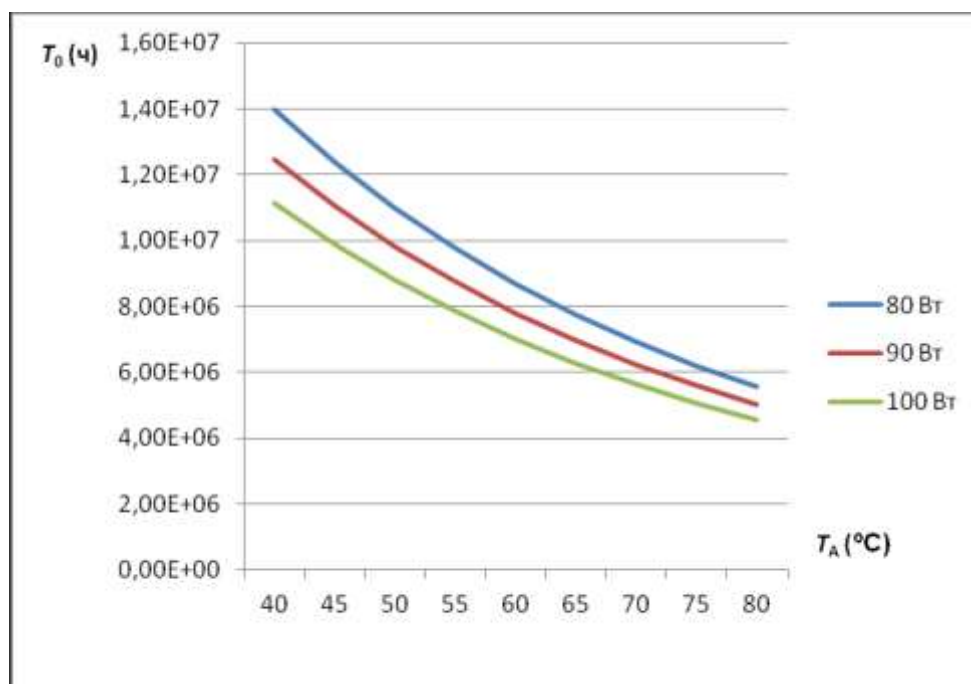


Рисунок 7.14 - Прогнозируемая зависимость надежности микросхемы от температуры окружающей среды и рассеиваемой мощности

7.5.2 Зависимости основных электрических параметров микросхем от режимов и условий эксплуатации приведены ниже.

Рассеиваемая мощность не более 30 Вт.

Динамическая потребляемая мощность определяется, в основном, мощностью потребляемой ядром микросхемы (внутренними схемами). Потребляемую мощность ядра P_{U_x} , Вт, вычисляют по формуле

$$P_{U_x} = P_{\max} \times \left(\frac{U_x}{U_{\max}} \right)^2, \quad (1)$$

где P_{\max} - потребляемая мощность при номинальной частоте и максимальном напряжении питания U_{\max} (с учетом допуска);

U_x – рабочее напряжение.

Потребляемую мощность ядра $P_{f_{\text{раб}}}$, Вт, вычисляют по формуле

$$P_{f_{\text{раб}}} = 0,8P_{\text{ном}} \times \frac{f_{\text{раб}}}{f_{\text{ном}}} + 0,2P_{\text{ном}}, \quad (2)$$

где $f_{\text{ном}}$ – номинальная частота, Гц;

$f_{\text{раб}}$ – рабочая (пониженная) частота, Гц;

$P_{\text{ном}}$ – мощность, потребляемая при номинальных частоте и напряжении питания.

Допустимое (номинальное) напряжение питания ядра может быть уменьшено до 0,9 $U_{\text{ном}}$ пропорционально уменьшению частоты синхронизации. Например, на частоте 0,9 $f_{\text{ном}}$ напряжение питания и потребляемая мощность могут иметь значения: $U_{\text{ядра}} = 0,9 U_{\text{ном}}$ и $P_{\text{потр}} \approx 0,75 P_{\text{ном}}$. Наиболее экономичным является одновременное снижение напряжения питания и частоты.

Микросхема может использоваться при пониженном значении частоты следования импульсов тактовых сигналов ядра микросхемы не менее 500 МГц.

Токи периферийных усилителей практически не зависят от температуры.

Статический ток ядра микросхемы имеет слабую зависимость от температуры - увеличивается в 5 раз при изменении температуры окружающей среды от минус 40 °С до плюс 90 °С на крышке корпуса.

Динамический ток потребления микросхем изменяется линейно от изменения питающего напряжения и от изменения частоты.

Токи потребления внешних интерфейсов, использующих дифференциальные сигналы, не меняются при изменении частоты.

Задержки элементов ядра $t_{d_эл.}$ и максимально-допустимая частота синхронизации f_C вычисляются по формулам

$$t_{d_эл.} = t_{d_эл. \text{ ном.}} \times (U_{\text{ядра ном.}} / U_{\text{ядра}}) \quad (3)$$

$$f_C = f_{C \text{ ном.}} \times (U_{\text{ядра}} / U_{\text{ядра ном.}}), \quad (4)$$

где $t_{d_эл. \text{ ном.}}$ - задержка элементов ядра при номинальном напряжении питания, с;

$f_{C \text{ ном.}}$ - номинальное значение синхронизации, Гц;

$U_{\text{ядра ном.}}$ - номинальное значение напряжения питания ядра, В.

Зная зависимости частоты синхронизации и мощности от напряжения питания ядра микросхемы можно оптимизировать режимы использования микропроцессора.

7.5.3 Предельная температура р-п-перехода кристалла – 150 °С.

7.5.5 Тепловые характеристики микросхемы представлены в таблице 7.13.

Таблица 7.13 - Тепловые характеристики микросхемы

Наименование параметра	Обозначение параметра	Значение параметра, °С/Вт
Сопротивление кристалл – внешняя среда	θ_{JA}	0,30
Сопротивление кристалл - центр теплораспределителя	ψ_{JT}	0,09
Сопротивление кристалл - внешний радиатор	θ_{JC}	0,09

Температура крышки корпуса микросхемы не должна превышать 90 °С.

7.5.6 Емкость входа, выхода и входа/выхода – 5 пФ.

7.5.7 Шарики выводов сформированы оловянно-медным припоем, содержащим 98 % олова и 2 % меди.

7.5.8 Размеры корпуса микросхемы 42,5 x 42,5 x 3,7 мм.

7.5.9 Количество выводов корпуса 1903.

7.5.10 Технологический размер элементов 16 нм. Размеры кристалла 21,85 x 10,18 мм (222,43 мм²).

7.5.11 Микросхема не имеет собственных резонансных частот ниже 100 Гц.

7.5.12 Микросхема не содержит горючих материалов. Пожаробезопасный режим под электрической нагрузкой – не более 50 Вт потребляемой мощности в течение 30 минут в условиях отсутствия системы охлаждения.

7.5.13 Условное графическое обозначение микросхем приведено на рисунке 7.15.

	Интерфейс синхронизации		Интерфейс синхронизации	
—	CLK_REF_100M_BOT_P			
—	CLK_REF_100M_BOT_N			
—	CLK_REF_100M_TOP_P			
—	CLK_REF_100M_TOP_N			
—	SUS_CLK_14M			
—	CLK_REF_156M_P			
—	CLK_REF_156M_N			
—	TCK			

		Интерфейс управления и диагностики		Интерфейс управления и диагностики	
—		AC_POWER_PSNT		RFU[5:0]	—
—o		ATE_MODE		SLEEP_S3_	o—
—		BATLOW_		SLEEP_S4_	o—
—		CORE_ENBL[1:0]		SLEEP_S5_	o—
—o		CPU_DISABLE_SOFT_RST_		SYS_GPIO[15:0]	—
—o		CPU_HRST_IN_		TDO	—
—o		CPU_SRST_IN_		TRIG_OUT	—
—		DBG_RST_DSBL			
—		DBG_STOP			
—		DDR_PWROK[1:0]			
—		EFUSE_MODE[1:0]			
—		FREQ_MODE[1:0]			
—o		PWR_BTN_			
—		RT			
—		SUS_PWROK			
—		SYS_KPI2BOOT_ENA			
—		SYS_PWROK			
—		TDI			
—		TMS			
—		TRIG_IN			
		Интерфейс каналов памяти, j = 0 - 1		Интерфейс каналов памяти, j = 0 - 1	
—	Z↔	MCj_DQS_P [17:0]		MCj_CK_P[3:0], MCj_CK_N [3:0]	—
—	Z↔	MCj_DQS_N [17:0]		MCJ_MTEST, MCj_BG[1:0], MCj_CKE [3:0]	—
—	Z↔	MCj_DQ [71:0]		MCj_CS_N [3:0]	o—
—o	ZA	MCj_ALERT_N		MCj_ACT_N, MCj_RESET_N	o—
—	↔	MCj_VREF_TEST		MCj_BA [1:0], MCj_A [17:0]	—
				MCj_CID[2:0], MCj_PARITY,	—
				MCj_MTEST, MCj_ODT [3:0]	—
				MCj_ZN	—

Рисунок 7.15 – Условное графическое обозначение микросхем

(лист 1 из 4)

—		Интерфейс канала SPI SPI_MISO		Интерфейс канала SPI SPI_SCK, SPI_MOSI SPI_CS0_N, SPI_CS1_N, SPI_CS2_N SPI_CS3_N		— o— o—
—	OD↔	Интерфейс каналов I2C, J=0-4 I2C_SCLJ		Интерфейс каналов I2C, J=0-4 I2C_SDAJ	OD ↔	—
—	←	Интерфейс канала HDA HDA_SDI		Интерфейс канала HDA HDA_SDO, HDA_SYNC, HDA_CLK HDA_RST_N		— o—
— — — —		Интерфейс канала WLINK/PCle PE_RLANE[15:0]_P PE_RLANE[15:0]_N PE_PRE_DET#[3:0] PEWL_CONFIG[1:0]		Интерфейс канала WLINK/PCle PE_TLANE[15:0]_P PE_TLANE[15:0]_N PE_RESREF	↔A	— — —
— — — — — — — — — —	↔	Интерфейс каналов SATA/ETH ETH0(1)_MDIO ETH0(1)_LSTA, ETH0(1)_FETH ETH0_GETH, ETH0(1)_FDUP ETH0(1)_TX_FAULT ETH0(1)_RX_LOSS ETH0(1)_MOD_ABS SATAETH_RLANE_P[3:0] SATAETH_RLANE_N[3:0] SATAETH_CONFIG		Интерфейс каналов SATA/ETH ETH0(1)_PHYRST_N ETH0(1)_MDC, ETH0(1)_LINKUP_LED ETH0(1)_RXACT_LED ETH0(1)_TXACT_LED ETH0(1)_TX_DISABLE SATA_LED SATAETH_RESREF SATAETH_TLANE_P[3:0] SATAETH_TLANE_N[3:0]		o— — — — — — — — — —
—o —o — —o		Интерфейс каналов RS-232, j=A, B SP_CTSj_ SP_DCDj_ SP_RXDj SP_SYNCj_		Интерфейс каналов RS-232, j= A, B SP_DTRj_ SP_RTsj_ SP_TXDj		o— o— —
— —	A ↔	Интерфейс каналов USB 3.0, j=0-3 USBj_RX_P, USBj_RX_N USB_PORT_OVRCUR_[7:0]		Интерфейс каналов USB 3.0, j=0-3 USBj_D_P, USBj_D_N USBj_ID USBj_RESREF USBj_TX_P, USBj_TX_N USBj_VBUS	A↔ A↔ — A↔	— — — —

Рисунок 7.15 (Лист 2 из 4)

—	OD ↔	Интерфейс графического порта 0 (RGB) GFX0_DDC_SCL		Интерфейс графического порта 0 (RGB) GFX0_RGB_CLK_P	—
—	OD ↔	GFX0_DDC_SDA		GFX0_RGB_CLK_N	—
—	↔	GFX0_GPIO[5:0]		GFX0_RGB_DE	—
				GFX0_RGB_HSYNC	—
				GFX0_RGB_VSYNC	—
				GFX0_RGB_CS[0]	—
				GFX0_RGB_DATA[23:0]	—
—	↔	Интерфейс графических портов 1 и 2 HDMI, j=1, 2 GFXj_HDMI_GPIO[1:0]		Интерфейс графических портов 1 и 2 HDMI, j=1, 2 GFX j_HDMI_TMDS_CLK_P	—
—	A ↔	GFXj_HDMI_RESREF		GFX j_HDMI_TMDS_CLK_N	—
—	A ↔	FXj_HDMI_HPD		GFX1 j_HDMI_TMDS_DAT_P[2:0]	—
—	OD ↔	GFXj_HDMI_DDC_SCL		GFX j_HDMI_TMDS_DAT_N[2:0]	—
—	OD ↔	GFXj_HDMI_DDC_SDA		GFXj_HDMI_GD	A ↔
—	A ↔	GFXj_HDMI_VP		GFXj_HDMI_DDCCEC_GND	A ↔
—	A ↔	GFXj_HDMI_VPH			—
		Интерфейс графического порта 3 (LVDS, j=0-3)		Интерфейс графического порта 3 (LVDS, j=0-3) GFX3_LVDS_PWM[1:0]	↔
				GFX3_LVDS_CLK_j_P	—
				GFX3_LVDS_CLK_j_N	—
				GFX3_LVDS_DAT_j_P[4:0]	—
				GFX3_LVDS_DAT_j_N[4:0]	—
—		Интерфейс канала I2C Slave MNTR_SCL		Интерфейс канала I2C Slave MNTR_SDA	↔

Рисунок 7.15 (Лист 3 из 4)

	Питание	Общий U _{SS}
— U _{CC1}	PWR_0V8	GND_0V8
— U _{CC2}	PWR_0V8_BIO PWR_A0V8_GFX1 PWR_A0V8_GFX2 PWR_A0V8_PE0 PWR_A0V8_PE1 PWR_A0V8_PE2 PWR_A0V8_PE3 PWR_A0V8_SATAETH0 PWR_A0V8_SATAETH1 PWR_A0V8_USB0 PWR_A0V8_USB1 PWR_A0V8_USB2 PWR_A0V8_USB3	GND_12G
— U _{CC3}	PWR_1V2_MC	GND_1V2_MC
— U _{CC4}	PWR_1V8_BIO PWR_A1V8_GFX1 PWR_A1V8_GFX2 pwr_a1v8_mc0_pll pwr_a1v8_mc1_pll PWR_A1V8_PE PWR_A1V8_SATAETH PWR_A1V8_SYNC_BOT PWR_A1V8_SYNC_TOP	GND_12G
— U _{CC5}	PWR_A3V3_USB	GND_12G
— U _{CC6}	PWR_0V8_SUS	GND_12G
— U _{CC7}	PWR_1V8_SUS	GND_12G
— U _{CC8}	PWR_0V8_MC	GND_12G
— U _{CC9}	PWR_1V5_HDA	GND_12G
— U _{CC10}	PWR_1V8_EFUSE	GND_12G
— U _{CC11}	PWR_A0V8_SYNC_BOT PWR_A0V8_SYNC_TOP	GND_SYNC_BOT GND_SYNC_TOP

Рисунок 7.15 (Лист 4 из 4)

Примечание – Назначение сигналов и координаты выводов приведены в разделе 5 и ТВГИ.431281.027ГБ соответственно.

7.5.14 Динамические токи потребления микросхемы составляют:

- I_{OCC1} – не более 35,0 А (от источника питания U_{CC1} домена CORE);
- I_{OCC2} – не более 1,00 А (от источника питания U_{CC2} домена UNCORE и низковольтного питания PHY периферии);
- I_{OCC3} – не более 3,70 А (от источников питания U_{CC3} интерфейса каналов памяти);
- $I_{OCC4} + I_{OCC7} + I_{OCC9}$ – не более 2,00 А (от источника питания U_{CC4} , U_{CC7} , U_{CC9} интерфейсов BIO, LVDS, HDA и высоковольтного питания PHY блоков периферии (за исключением USB 3.0));
- I_{OCC5} – не более 0,30 А (от источника питания U_{CC5} высоковольтного питания PHY блоков USB 3.0);
- I_{OCC6} – не более 0,10 А (от источника питания U_{CC6} внутренних схем блока Suspend);
- I_{OCC7} – не более 0,15 А (от источника питания U_{CC7} выходных каскадов блока Suspend);
- I_{OCC8} – не более 2,00 А (от источника питания U_{CC8} внутренних схем каналов памяти MC0 – MC1);
- I_{OCC10} – не более 0,14 А (от источника питания U_{CC10} при программировании блока ПЗУ eFuse);
- I_{OCC11} – не более 0,20 А (от источника питания U_{CC11} блоков синхронизации).

7.5.15 Микросхема содержит цветные металлы:

- медь 12,24 г;
- алюминий 0,0007 г;
- олово 1,59 г;
- никель 0,27 г.

8 Хранение

8.1 Гамма-процентный срок сохраняемости T_{γ} при $\gamma = 99\%$ при хранении микросхем в упаковке изготовителя в условиях отапливаемых хранилищ, хранилищ с кондиционированием воздуха по ГОСТ В 9.003-84, а также вмонтированных в защищенную аппаратуру или находящихся в защищенном комплекте ЗИП – не мене 25 лет.

8.2 Значения T_{γ} для всех климатических районах по ГОСТ В 9.003-84 (кроме районов с тропическим климатом) в условиях, отличных от указанных в 8.1, в зависимости от мест хранения равен значениям, указанным в таблице 8.1.

Таблица 8.1 – Значения гамма-процентного срока сохраняемости

Место хранения	Значения T_{γ} , лет, при хранении	
	в упаковке изготовителя	в составе незащищенных аппаратуры и комплекта ЗИП
Неотапливаемое хранилище	16,5	16,5
Под навесом	12,5	12,5
На открытой площадке	хранение не допускается	12,5

9 Транспортирование

Транспортирование микросхем производится только в упаковке предприятия – изготовителя. При этом транспортирование допускается осуществлять транспортом любого вида на любые расстояния по правилам перевозок грузов, действующим на транспорте данного вида, согласно ГОСТ РВ 20.39.412-97.

10 Утилизация

Микросхемы не содержат драгоценных материалов и, после списания, подлежат передаче в установленном порядке в соответствующее специализированное предприятие по переработке.

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Входящий номер сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					