



### КРАТКИЙ ОБЗОР ПРОЦЕССОРА

Микропроцессор МСр0411100101 имеет в своем составе мультиклеточное процессорное ядро – первое процессорное ядро с принципиально новой (пост-неймановской) мультиклеточной архитектурой российской разработки. Мультиклеточный процессор предназначен для решения широкого круга задач управления и цифровой обработки сигналов в приложениях, требующих минимального энергопотребления и высокой производительности.

Данный мультиклеточный процессор состоит из 4 клеток (когерентных процессорных блоков), объединенных интеллектуальной коммутационной средой.

#### Особенности:

- Число клеток — 4
- Разрядность процессора — 32/64 бита
- ПД — 128Кб (4\*4К\*64)
- ПП — 128Кб (4\*4К\*64)
- ПЗУ — 256Кб<sup>1</sup>
- блок операций над числами с плавающей запятой (в каждой клетке)
- Тактовая частота — 100 МГц
- Производительность процессора — 2,4 Gflops;

#### Общие характеристики:

- Корпус — QFP-208
- Условия эксплуатации – (0...+70)
- Максим. потребл. мощность:
  - ядра — 45мВт
  - периферии — 80мВт
- Напряжение питания (раздельное):
  - ядра — 1,8В
  - периферии — 3,3В

#### Периферийные устройства:

- 2 интерфейса SPI с селектором “ведомых” устройств (в режиме “ведущий”)
- 4 универсальных асинхронных приёмопередатчика UART с FIFO на прием/передачу
- 2 интерфейса I2C (один “master” и один “slave”)
- интерфейс I2S
- Ethernet контроллер 10/100Мб/с
- USB 1.1 FS (device) контроллер с последовательным внешним интерфейсом для подключения приемо-передатчика
- часы реального времени с календарем
- 7 таймеров общего назначения
- 4 порта ввода-вывода, общее количество вводов-выводов – 104
- 4-х канальный контроллер ШИМ
- сторожевой таймер

<sup>1</sup> — ПЗУ в исполнении «1» отсутствует, для хранения исполняемого кода должно применяться внешнее последовательное FLASH ПЗУ XCF04S.



### Преимущества процессора:

- увеличение производительности в 4-5 раз при одновременном снижении энергопотребления (при одинаковых тактовых частотах и топологических нормах исполнения):
  - в 2-4 раза по сравнению с аудиопроцессорами;
  - в 10-15 раз по сравнению с процессорными ядрами со сверхнизкими энергопотреблением;
- «естественная» реализация параллелизма (без решения задачи распараллеливания);
- уменьшение площади кристалла;
- эффективная реализация любого класса задач (коммутационная среда не вносит каких-либо ограничений в межклеточный обмен данными);
- выполнение программы без перекомпиляции на любом количестве клеток.

### Области применения:

- системы промавтоматики от интеллектуальных датчиков до систем управления двигателями;
- универсальные навигационные приемники ГЛОНАСС/GPS/Galileo/COMPASS(Китай)/ IRNSS(Индия)/QZSS(Япония);
- мобильные телефоны;
- видеотехника 3D;
- автомобильная электроника для «интеллектуальных» бортовых систем, контролирующих дорожную обстановку и предупреждающих водителей об опасностях и пробках;
- системы безопасности, автоматически распознающих «своих» и «чужих».



### ОБЗОР ДОКУМЕНТА

В документе приведено краткое описание особенностей и возможностей микропроцессора МСр0411100101, его внутренней организации, адресного пространства запоминающих устройств, а также сделан обзор периферийных устройств МП.



## ОГЛАВЛЕНИЕ

Принятые сокращения.....	5
1. Структура МП.....	6
1.1. Ядро МП.....	7
1.2. Адресное пространство.....	8
1.2.1. Поле регистров.....	9
1.2.2. Память программ.....	14
1.2.3. Память данных и шина периферийных устройств.....	15
1.2.4. Процесс инициализации МП.....	16
1.3. Системный сброс и система тактирования.....	18
1.4. Контроллер прерываний.....	19
1.5. Системный таймер.....	22
2. Периферийные устройства.....	24
2.1. Универсальный асинхронный приёмопередатчик (UART0..UART3 ).....	24
2.2. Последовательный интерфейс I2C.....	25
2.2.1. I2C master (I2C0).....	25
2.2.2. I2C slave (I2C1).....	26
2.3. Последовательный интерфейс SPI (SPIx).....	27
2.4. Сторожевой таймер (WDT).....	28
2.5. Часы реального времени (RTC).....	29
2.6. Контроллер ШИМ (PWM0).....	30
2.7. Порт ввода-вывода (GPIOx).....	31
2.8. Ethernet контроллер (Ethernet0).....	32
2.9. Таймер общего назначения (GPTIMx).....	33
2.10. Последовательный интерфейс I2S (I2S0).....	34
2.11. USB (USB0).....	35
3. Назначение выводов процессора.....	36
4. Маркировка мультиклеточных процессоров.....	37
5. Таблица ревизий.....	38



### Принятые сокращения:

МП — микропроцессор;

ПО – программное обеспечение;

ПК – персональный компьютер;

ПЗУ – постоянное запоминающее устройство;

ОЗУ – оперативное запоминающее устройство;

СОЗУ — статическое оперативное запоминающее устройство;

ПБ – процессорный блок (клетка);

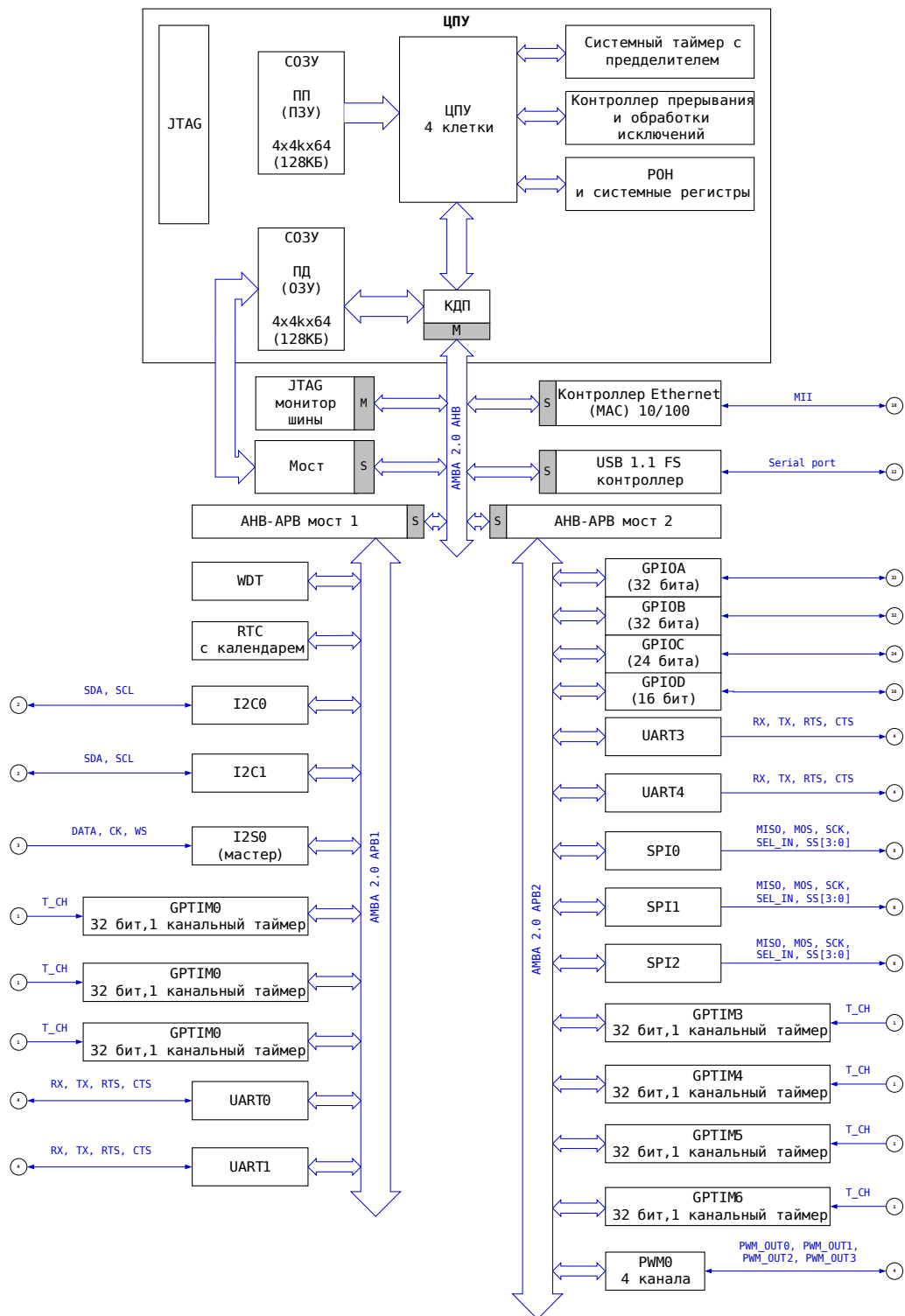
ПП – память программ;

ПД – память данных;

ЦПУ – центральное процессорное устройство;

РОН — регистр(ы) общего назначения.

## 1. Структура МП



*Рис. 1 - Общая структура процессора*

## ЦПУ:

- контроллер прерываний с аппаратом обработки исключительных ситуаций;
- ядро, предназначенное для вычислений и осуществления функций управления;
- системный таймер;
- модули внутрисхемной отладки;
- интерфейсы доступа к памяти и системной шине;
- системные регистры и регистры общего назначения.

## Шина периферийных устройств:

Описание подключенных к шине периферийных устройств смотрите в п. 2.

## 1.1. Ядро МП

Ядро в своем составе имеет:

- 4 ПБ, имеющих нумерацию [0, 3];
- поле системных регистров и регистров общего назначения (РОН);

Структурная схема ядра микропроцессора представлена на Рис. 2

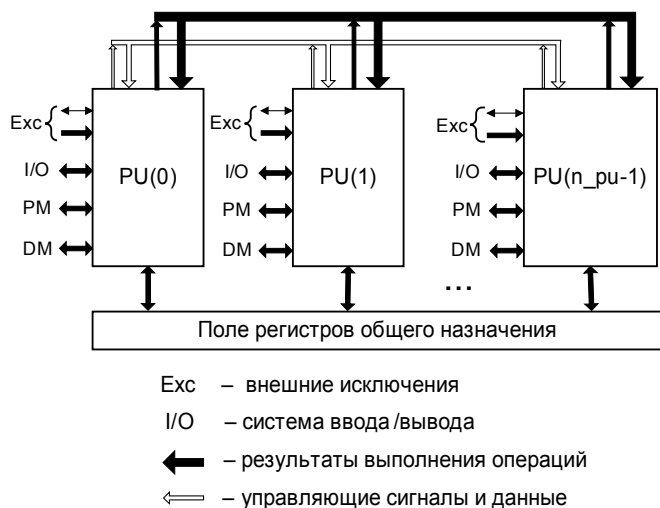


Рис. 2 - Структурная схема ядра

ПБ представляет собой набор управляющих и исполнительных блоков, имеет развитую систему команд. На Рис. 2 имеются следующие обозначения: PM — шина памяти программ, DM — шина памяти данных, I/O — шина ввода/вывода, Exc — шина связи с контроллером прерываний и обработчиком исключительных ситуаций.

## 1.2. Адресное пространство МП

В своем адресном пространстве МП имеет:

- память программ(ПП);
- память данных (ПД);
- шину периферийных устройств.

Ко всем этим компонентам пользователь может адресоваться в своем программном алгоритме, исключение составляет ПП, она работает только в режиме чтения и не доступна пользователю для записи и произвольного чтения. ПП имеет отдельную шину подсоединения к ядру МП.

ПД и шина периферийных устройств находятся в одном адресном пространстве и имеют отдельную шину подсоединения к ядру МП.

Поле регистров доступно по выделенной шине и физически является частью ядра МП.

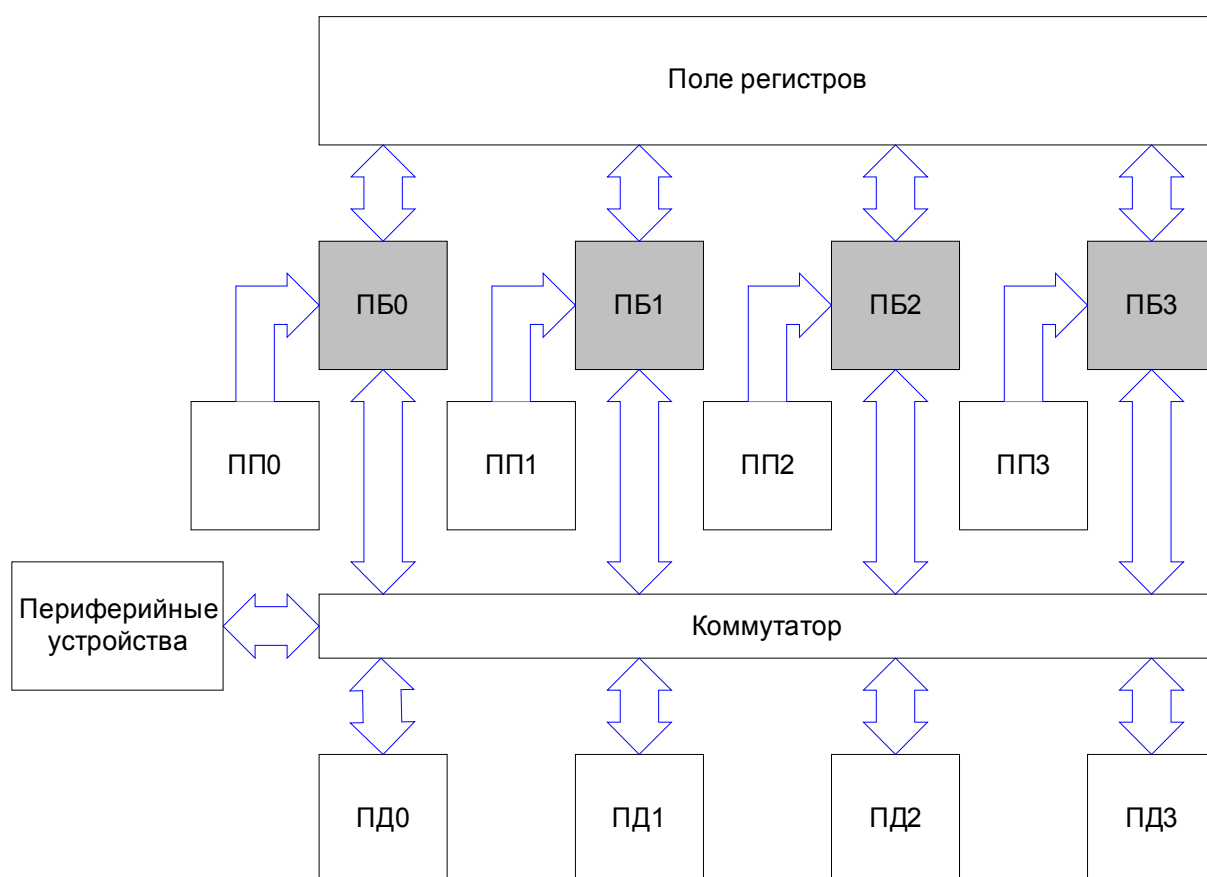


Рис. 3 - Организация памяти



### 1.2.1. Поле регистров

Все регистры, кроме управляющих, имеют ширину 64 бита.

Чтение/запись регистров происходит с помощью специализированных команд.

Тип регистров	Номер
Регистры общего назначения	0-7
<i>Резерв</i>	<i>8-31</i>
Индексные регистры	32-47
Управляющие регистры:	
PSW	48
INTR	49
MSKR	50
ER	51
IRETADDR	52
STVAL	53
STCR	54
IHOOKADDR	55
INTNUMR	56
MODR	57

#### Индексные регистры:

Используются для косвенной адресации и имеют следующую логическую структуру:

Номера битов	63..48	47..32	31..0
	Индекс (Index)	Маска (Mask)	База (Base)

В общем случае, при использовании регистра данного типа в качестве аргумента операции, значение этого аргумента формируется согласно следующему алгоритму:

- вычисление исполнительного адреса, согласно следующей формуле:

$$Address = Index + Base$$

- обращение к памяти данных по исполнительному адресу Address для чтения значения аргумента согласно типу используемой операции.

Модификация значения индексного регистра осуществляется аппаратно по завершению параграфа в том случае, если установлен соответствующий бит регистра MODR маски изменения индексных регистров в регистре PSW, согласно следующей



формуле:

$$Index = ((Index \mid \sim Mask) + 1) \& Mask,$$

где  $\mid$  - операция побитового «ИЛИ»,  $\&$  - операция побитового «И»,  $\sim$  - операция побитового инвертирования.

## Управляющие регистры:

Процессор имеет в своем составе следующие управляющие регистры:

PSW	30h	RW	Регистр управления
INTR	31h	RW	Регистр прерываний
MSKR	32h	RW	Регистр маски прерываний
ER	33h	RC	Регистр ошибок
IRETADDR	34h	R	Регистр адреса возврата
STVAL	35h	RW	Период счетчика
STCR	36h	RW	Регистр управления счетчиком
IHOOKADDR	37h	RW	Регистр адреса первичного обработчика прерываний
INTNUMR	38h	R	Номер выработанного прерывания
MODR	39h	RW	Регистр модификации индексных регистров

## Описание управляющих регистров:

### PSW

Номер бита  
Операции  
Начальное состояние  
Описание

Регистр управления PSW																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R=0																							RW	RW	RW	R	RW	R	RW		
0																							0	0	0	0	0	0	0		
Зарезервировано																							RW	SWR	STOP	Reserved	SUSPEND	Reserved	OUTROS		

#### Назначение разрядов регистра:

- 0 Разрешение обработки немаскируемых прерываний:  
0 – запрещено  
1 – разрешено  
Признак не блокирует прием прерываний регистром INTR, а блокирует только их дальнейшую обработку. Устанавливается всегда только программно, снимается программно и аппаратно (при переходе на программу обработки прерываний).
- 1 Зарезервировано
- 2 Переход в режим ожидания:  
По окончании параграфа, в котором устанавливается данный бит, ядро переходит в режим ожидания. В этом состоянии оно находится до прихода прерывания. По приходу прерывания данный признак снимается аппаратно и управление передается на программу обслуживания прерываний. По окончании обслуживания прерывания, если программистом не заданы какие-либо действия, продолжается выполнения приостановленной программы.
- 3 Зарезервировано
- 4 Остановка ядра:  
При установке в 1, по завершению текущего параграфа, ядро прекращает выборку команд. Возобновление работы возможно только извне, путем аппаратного сброса или подачи на вход "wake\_up" процессора напряжения уровня логической "1".
- 5 Программный сброс:  
При установке в 1, по завершению параграфа процессор проходит полную инициализацию, ядро автоматически начинает работу с исходного состояния, регистры сбрасываются в исходное состояние. Содержимое оперативной памяти не сохраняется.
- 6 Очередность исполнения команд чтения/записи:  
0 – команды записи исполняются только после выполнения всех команд чтения данного параграфа  
1 – контроль очередности не выполняется
- 7-31 Зарезервировано

### MSKR

Номер бита  
Операции  
Начальное состояние  
Описание

Регистр маски прерываний MSKR																																		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
R=0																RW																		
0																0																		
Зарезервировано																Маскируемые прерывания																		

#### Назначение разрядов регистра:

- 0-27 Маска для 4-31 битов регистра INTR  
0 – запрет обработки запроса прерывания  
1 – разрешение обработки прерывания
- 27-31 Зарезервировано





## Краткое описание процессора МСр0411100101

Начальное состояние  
Описание

0	0
Зарезервировано	IRETADDR

### Назначение разрядов регистра:

- 0-7 Адрес возврата, формируемый только при прерывании  
8-31 Зарезервировано

### STVAL

Номер бита  
Операции  
Начальное состояние  
Описание

Регистр периода счетчика STVAL																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0																															
CNTVAL																															

### Назначение разрядов регистра:

- 0-31 Значение периода счетчика в периодах системной частоты после предделителя системного таймера (см. STCR).

### STCR

Номер бита  
Операции  
Начальное состояние  
Описание

Регистр управления счетчиком STCR																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R=0																RW															
0																0															
Зарезервировано																PREDIV															
																Зарезервировано								CNTOP		ENCNT		EN			

### Назначение разрядов регистра:

- 0 Разрешение работы счетчика:  
0 – запрещено  
1 – разрешено
- 1 Разрешение циклической работы таймера (если запрещена циклическая работа, то таймер после одного периода выключится, бит 0 данного регистра будет установлен в "0"):  
0 – разрешено  
1 – запрещено
- 2 Признак завершения счета периода, заданного в STVAL (по данному признаку вырабатывается запрос на обработку прерывания от таймера):  
0 – счетчик не отсчитал заданный период  
1 – счетчик отсчитал период (в циклическом режиме работы признак обновляется после каждого периода, программист должен самостоятельно очищать данный признак)
- 3-7 Зарезервировано
- 8-15 Значение предделителя счетчика. Системная частота делится на значение, заданное в данных битах. Поделенная частота является тактовой для счетчика таймера.
- 16-31 Зарезервировано



## Краткое описание процессора МСр0411100101

Начальное состояние  
Описание

0	0
Зарезервировано	INTHOOKADDR

### Назначение разрядов регистра:

0-7 Значение адреса первичного обработчика прерываний  
8-31 Зарезервировано

### INTNUMR

Номер бита  
Операции  
Начальное состояние  
Описание

Регистр номера выработанного прерывания INTNUMR																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R=0																R															
0																0															
Зарезервировано																INT_NUM															

### Назначение разрядов регистра:

0-5 Номер самого приоритетного разрешенного прерывания на данный момент  
6-31 Зарезервировано

### MODR

Номер бита  
Операции  
Начальное состояние  
Описание

Регистр модификации индексных регистров MODR																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R=0																RW				RW				RW				RW			
0																0				0				0				0			
Зарезервировано																															

### Назначение разрядов регистра:

0 Разрешение модификации индексного регистра номер 0:  
0 – запрещено  
1 – разрешено  
1 Разрешение модификации индексного регистра номер 1:  
0 – запрещено  
1 – разрешено  
2 Разрешение модификации индексного регистра номер 2:  
0 – запрещено  
1 – разрешено  
3 Разрешение модификации индексного регистра номер 3:  
0 – запрещено  
1 – разрешено  
4 Разрешение модификации индексного регистра номер 4:  
0 – запрещено  
1 – разрешено  
5 Разрешение модификации индексного регистра номер 5:  
0 – запрещено  
1 – разрешено  
6 Разрешение модификации индексного регистра номер 6:  
0 – запрещено  
1 – разрешено  
7 Разрешение модификации индексного регистра номер 7:  
0 – запрещено  
1 – разрешено  
8-31 Зарезервировано

### 1.2.2. Память программ

Память программ представляет собой независимые блоки СОЗУ (PM0 – PM3), число которых равняется числу ПБ. Соответственно, каждый ПБ имеет свою собственную ПП. Указанные блоки памяти не связаны между собой и функционируют независимо.

Для пользователя ПП работает только в режиме чтения и используется только для хранения программного алгоритма. Для констант используется выделенная для этого область в ПД.

Все блоки ПП имеют организацию — 4Кх64 (32кБ), адресация происходит к 64-х разрядному двойному слову.

пп0	пп1	пп2	пп3
0x0000	0x0000	0x0000	0x0000
0x0001	0x0001	0x0001	0x0001
...	...	...	...
0x0fff	0x0fff	0x0fff	0x0fff

*Рис. 4 - Структура ПП*

## 1.2.3. Память данных и шина периферийных устройств

ПД и шина периферийных устройств находятся в одном адресном пространстве.

Блок	Начальный адрес	Конечный адрес	Размер, байт	Описание
<b>Блок 8</b>	0xFFFF F000	0xFFFF FFFF		Конфигурационная информация AMBA АНВ (только чтение)
<b>АМБА АНВ, периферия ввода-вывода</b>	0xFFFF 5100	0xFFFF EFFF		Зарезервировано
	0xFFFF 5000	0xFFFF 50FF	256	Ethernet0
	0xFFFF 4100	0xFFFF 4FFF		Зарезервировано
	0xFFFF 4000	0xFFFF 40FF	256	USB0
	0xFFFF 0000	0xFFFF 3FFF		Зарезервировано
	0xE021 F400	0xFFEF FFFF		Зарезервировано
<b>Блок 7</b>	0xE020 0000	0xE021 F3FF		ПДП канал к ПД (128 КБ)
	0xE000 0000	0xE01F FFFF		Зарезервировано
	0xC020 0000	0xDFFF FFFF		Зарезервировано
	0xC01F F000	0xC01F FFFF		Конфигурационная информация AMBA APB1 (только чтение)
	0xC01F 0400	0xC01F EFFF		Зарезервировано
	0xC01F 0300	0xC01F 03FF	256	GPIOD
<b>АМБА APB1</b>	0xC01F 0200	0xC01F 02FF	256	GPIOC
	0xC01F 0100	0xC01F 01FF	256	GPIOB
	0xC01F 0000	0xC01F 00FF	256	GPIOA
	0xC011 4F00	0xC01E FFFF		Зарезервировано
	0xC011 0300	0xC011 03FF	256	Timer6
	0xC011 0200	0xC011 02FF	256	Timer5
	0xC011 0100	0xC011 01FF	256	Timer4
	0xC011 0000	0xC011 00FF	256	Timer3
	0xC010 2300	0xC010 FFFF	256	Зарезервировано
	0xC010 2200	0xC010 22FF	256	SPI2
	0xC010 2100	0xC010 21FF	256	SPI1
	0xC010 2000	0xC010 20FF	256	SPI0
	0xC010 0300	0xC010 1FFF		Зарезервировано
	0xC010 0200	0xC010 02FF	256	UART3
	0xC010 0100	0xC010 01FF	256	UART2
	0xC010 0000	0xC010 00FF	256	Зарезервировано
	0xC00F F000	0xC00F FFFF		Конфигурационная информация AMBA APB0 (только чтение)
	0xC00E 0100	0xC00E FFFF		Зарезервировано
	0xC00E 0000	0xC00E 00FF	256	WDT
	0xC001 F100	0xC001 FFFF		Зарезервировано
<b>АМБА APB0</b>	0xC001 F000	0xC001 F0FF	256	RTC
	0xC001 0300	0xC001 EFFF		Зарезервировано
	0xC001 0200	0xC000 02FF	256	Timer2
	0xC001 0100	0xC000 01FF	256	Timer1
	0xC001 0000	0xC000 00FF	256	Timer0
	0xC000 5100	0xC000 FFFF		Зарезервировано
	0xC000 5000	0xC000 50FF	256	Ethernet0 (регистры управления)
	0xC000 2100	0xC000 4FFF		Зарезервировано
	0xC010 2000	0xC010 20FF	256	I2S0
	0xC000 1200	0xC010 1FFF		
	0xC000 1100	0xC000 11FF	256	I2C1
	0xC000 1000	0xC000 00FF	256	I2C0
	0xC000 0300	0xC000 0FFF		Зарезервировано
	0xC000 0200	0xC000 02FF	256	UART1
	0xC000 0100	0xC000 01FF	256	UART0
	0xC000 0000	0xC000 00FF		Зарезервировано
<b>Блок 6</b>		0xBFFF FFFF		Зарезервировано
<b>Блок 5</b>	0xA000 0000	0x9FFF FFFF		Зарезервировано
<b>Блок 4</b>	0x8000 0000	0x7FFF FFFF		Зарезервировано
<b>Блок 3</b>	0x6000 0000	0x5FFF FFFF		Зарезервировано
<b>Блок 2</b>	0x4000 0000	0x3FFF FFFF		Зарезервировано
<b>Блок 1</b>	0x2000 0000	0x1FFF FFFF		Зарезервировано
<b>ПД</b>	0x0002 0000			
	0x0000 0000	0x0001 FFFF	128k	Пользовательская ПД (адресация к каждому байту)

Рис. 5 - Адресное пространство периферийных устройств и ПД

## Память данных:

ПД представляет собой СОЗУ, имеет организацию — 4х4Кх64 (128кБ), адресация происходит побайтно. Особенностью организации ПД является то, что ячейки со смежными адресами находятся в разных блоках ПД. Для сокращения времени доступа к памяти рекомендуется данные выравнивать на 8 байт.

Адрес (базовый)	пд0			пд1			пд2			пд3		
	смещение адреса											
0x00000	0x00000	...	0x00007	0x00008	...	0x0000f	0x00010	...	0x00017	0x00018	...	0x0001f
0x00020	0x00000	...	0x00007	0x00008	...	0x0000f	0x00010	...	0x00017	0x00018	...	0x0001f
...	...	...	...	...	...	...	...	...	...	...	...	...
0x1FFE0	0x00000	...	0x00007	0x00008	...	0x0000f	0x00010	...	0x00017	0x00018	...	0x0001f

Рис. 6 - Структура ПД

Адрес (базовый)	смещение адреса							
	64бита							
	байт 0	байт 1	байт 2	байт 3	байт 4	байт 5	байт 6	байт 7
0x00000	0x00000	0x00001	0x00002	0x00003	0x00004	0x00005	0x00006	0x00007
0x00020	0x00000	0x00001	0x00002	0x00003	0x00004	0x00005	0x00006	0x00007
...	...	...	...	...	...	...	...	...
0x1FFE0	0x00000	0x00001	0x00002	0x00003	0x00004	0x00005	0x00006	0x00007

Рис. 7 - Структура блока ПД на примере ПД0

Для данного МП нет понятия констант, хранящихся в ПЗУ, вместо них используются инициализируемые переменные, которые при старте МП помещаются в выделенную область ПД. Аппаратной защиты от записи во время исполнения программных алгоритмов пользователя на данный момент нет.

## 1.2.4. Процесс инициализации МП

В виду своих архитектурных особенностей, процесс инициализации МП так же имеет некоторые особенности.

Как упоминалось ранее, ПП и ПД – это СОЗУ. В связи с этим для хранения программного алгоритма и инициализируемых переменных (констант) применяется ПЗУ (FLASH).

### Процесс загрузки представляет собой следующее:

- После снятия внешнего сигнала сброса, начинает работать загрузчик программ, который переписывает данные из ПЗУ в ПП и ПД. В ПП переписывается программный алгоритм, в ПД записываются инициализируемые переменные (размер области инициализируемых переменных зависит от модели МП, на данный момент инициализируется вся ПД).
- После завершения процесса загрузки памяти снимается внутренний сигнал сброса и МП начинает функционировать в рабочем режиме.





В данном варианте МП применяется внешняя последовательная память. Объем считываемой информации – 256 кБ, т.е.  $8 \cdot 256 = 2\,048$  Бит. Максимальная скорость обмена данными с внешним ПЗУ – 30 Мбит/с. Из этого:

$T = 2/30 \sim 67$  мс, - время выхода МП на рабочий режим после снятия внешнего сигнала сброса.

### 1.3. Системный сброс и система тактирования

Система тактирования использует в качестве опорного сигнал внешнего кварцевого генератора. На данный момент ядро и периферийные устройства МП тактируются одной частотой от одного источника. Возможность управлять частотами отсутствует.

#### Управление сигналом системного сброса:

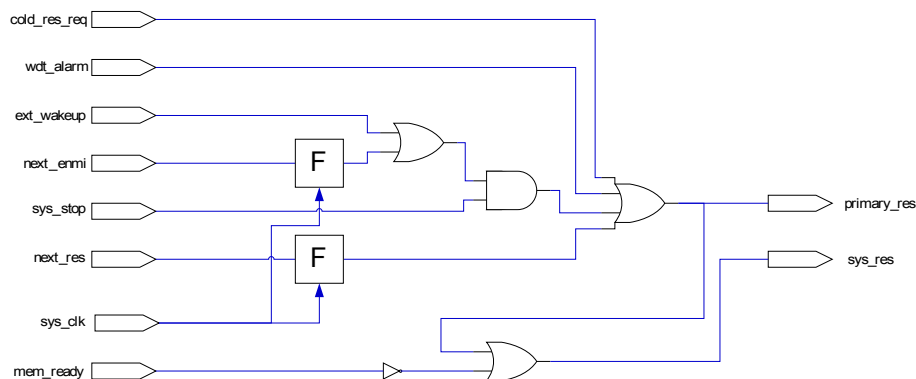


Рис. 8 - Схема управления системным сбросом

#### Внешний сигнал сброса:

- после снятия сигнала «nreset» снимается сигнал «primary\_res» и начинает работу загрузчик памяти;
- после завершения загрузки памяти выставляется сигнал «mem\_ready», по наличию которого, снимается сигнал «sys\_res», ядро начинает выбирать команды из памяти, начиная с адреса 0.

#### Сигнал «Стоп»:

- если ядро МП выполнит команду «Стоп», то активизируется сигнал «sys\_stop», по которому будет активизирован сигнал «primary\_res»;
- для выхода из режима «Стоп», необходимо возникновение события: «penmi», «ext\_wakeup» или любого запроса прерывания;
- альтернативный способ выхода из режима «Стоп» — подача внешнего сигнала сброса «nreset».

#### Запрос сброса:

- ядро системы может запросить выполнить сброс системы. Выставление сигнала запроса является результатом работы программного алгоритма (записана единица в бит 5 регистра PSW);
- после возникновения сигнала «cold\_res\_req» выставляется сигнал «primary\_res», будут выполнены такие же действия, как при снятии сигнала «nreset».

## 1.4. Контроллер прерываний

Система прерываний МП допускает обработку 32 прерываний. Источник с номером «0» имеет наивысший приоритет при обработке прерываний.

0	Немаскируемое внутреннее прерывание (INMI)
1	Немаскируемое внешнее прерывание (ENMI)
2	Немаскируемое исключение в аппаратной части (PERE)
3	Немаскируемое программное исключение (PPGE)
4	Маскируемое программное исключение (MPRGE)
5	Прерывание от системного таймера (SWT)
6	Программное прерывание (SWI)
7	Маскируемое прерывание от UART0
8	Маскируемое прерывание от UART1
9	Маскируемое прерывание от UART2
10	Маскируемое прерывание от UART3
11	Маскируемое прерывание от I2C0
12	Маскируемое прерывание от I2C1
13	Маскируемое прерывание от SPI0
14	Маскируемое прерывание от SPI1
15	Маскируемое прерывание от SPI2
16	Маскируемое прерывание от I2S0
17	Маскируемое прерывание от GPTIM0
18	Маскируемое прерывание от GPTIM1
19	Маскируемое прерывание от GPTIM2
20	Маскируемое прерывание от GPTIM3
21	Маскируемое прерывание от GPTIM4
22	Маскируемое прерывание от GPTIM5
23	Маскируемое прерывание от GPTIM6
24	Маскируемое прерывание от PWM0
25	Маскируемое прерывание от RTC
26	Маскируемое прерывание от GPIOA
27	Маскируемое прерывание от GPIOB
28	Маскируемое прерывание от GPIOC
29	Маскируемое прерывание от GPIOD
30	Маскируемое прерывание от ETHERNET0
31	Маскируемое прерывание от USB0

Рис. 9 - Список прерываний

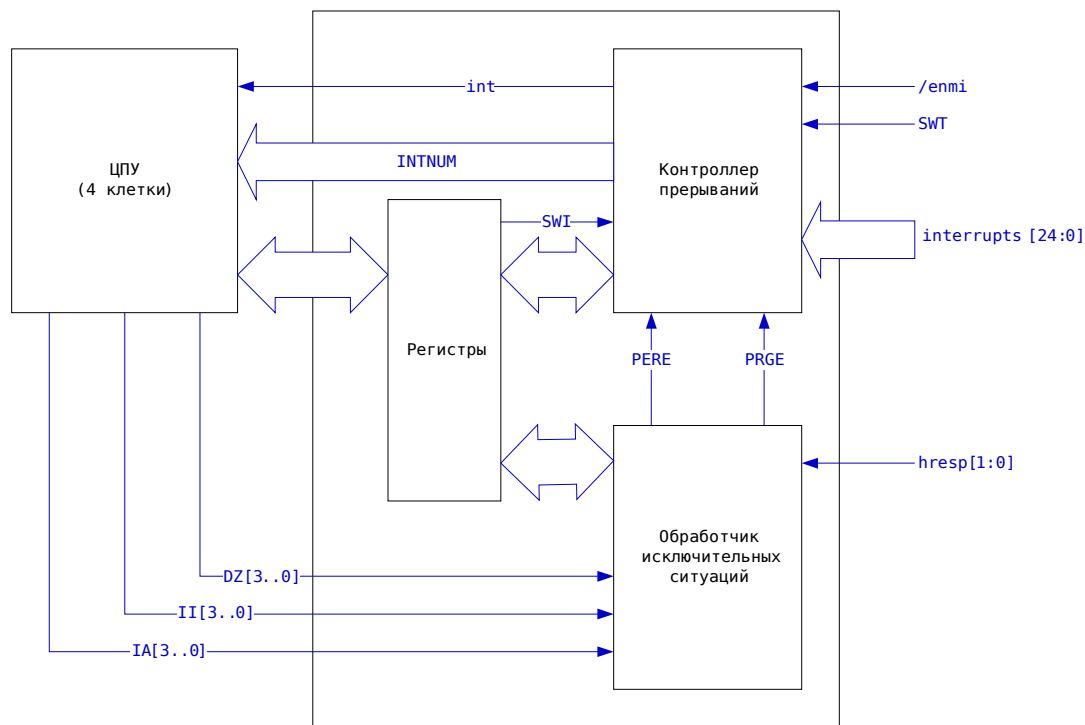
### Немаскируемые прерывания:

В контроллере предусмотрено 4 немаскируемых прерывания с номерами 0-3. Немаскируемые прерывания приводят к немедленному переходу на программу обработки прерываний. Их нельзя запретить, они разрешены сразу после начала работы ядра.

### Маскируемые прерывания:

Прерывания с номерами 4-31 являются маскируемыми, они глобально разрешаются битом ONIRQS в регистре PSW. Индивидуальное разрешение задается регистром MSKR.

## Устройство:



**Рис. 10 - Схема соединений процессорного ядра и контроллера прерываний**

Контроллер прерываний состоит из 2 функциональных модулей:

- контроллер прерываний - определяет самое приоритетное прерывание на каждом такте и формирует его номер на шине INTNUM;
- обработчик исключительных ситуаций - обрабатывает информацию об аппаратных и программных сбоях, передает сигнал об их возникновении в контроллер прерываний, что приводит к формированию запроса прерывания.

Из ЦПУ на контроллер прерываний приходят 3 группы сигналов об ошибках в ходе выполнения программы, по которым в обработчике исключений формируется сигнал PRGE:

- DZ – попытка выполнения деления на ноль;
- II – выбрана несуществующая инструкция;
- IA – сформирован несуществующий адрес.

Так же ЦПУ формирует сигнал SWI – программно-формируемое прерывание.

С шины периферийных устройств на обработчик исключений поступают сигналы hresp, состояние линий которых говорит об ошибках или их отсутствии при обращении по шине. При наличии ошибки формируется сигнал PERE.



Также с шины периферийных устройств приходят сигналы (interrupts[24:0]) запроса прерывания от каждого периферийного устройства.

С системного таймера поступает сигнал SWT, а с ввода МП приходит сигнал/enmi – внешнее немаскируемое прерывание.

### Регистры контроллера прерываний:

Для работы системы прерывания и функционирования программ обработки прерываний имеются следующие регистры:

INTR	Регистр прерываний
MSKR	Регистр маски прерываний
ER	Регистр ошибок
IRETADDR	Регистр адреса возврата
IHOOKADDR	Регистр адреса первичного обработчика прерываний
INTNUMR	Номер выработанного прерывания

Регистры INTR, MSKR, ER, INTNUM — относятся к управлению контроллером прерываний.

Регистры IRETADDR, IHOOKADDR — используются программным алгоритмом.

В памяти МП не выделено какой-либо фиксированной зоны для размещения обработчика прерываний. Программист может разместить обработчики прерываний в любом месте адресного пространства ПП. При возникновении прерывания ядро перейдет по адресу, записанному в регистре, IHOOKADDR. По данному адресу программист может расположить первичный обработчик прерываний, который может осуществить диспетчеризацию и перенаправить программу на необходимый адрес, где располагается обработчик для конкретного прерывания. Адрес возврата автоматически запоминается в регистре IRETADDR, программист имеет к нему полный доступ.

### Порядок обработки прерываний:

При возникновении прерывания, контроллер прерываний определяет самое приоритетное прерывание и формирует сигнал int.

Вся программа разбита на «параграфы» - группы команд, аналогом которых можно считать линейные участки. На время выполнения «параграфа» все прерывания запрещены, кроме немаскируемых, которые могут прервать работу процессорного ядра в любое время.

После завершения «параграфа» возможен переход на адрес первичного обработчика прерываний, если был запрос прерывания, прерывание не маскировано и нет глобального запрета прерываний (бит ONIRQS в регистре PSW). В регистр IRETADDR автоматически записывается адрес следующего «параграфа» и устанавливается глобальный запрет прерываний. Ядро переходит на выполнение алгоритма, размещенного по адресу, записанному в IHOOKADDR. После завершения работы обработчиков прерываний программный алгоритм должен осуществить выход из прерывания, адрес возврата записан в IRETADDR. Глобальное разрешение прерываний программа должна выставить сама.

## 1.5. Системный таймер

Системный таймер предназначен для формирования заданных периодических или однократных временных интервалов.

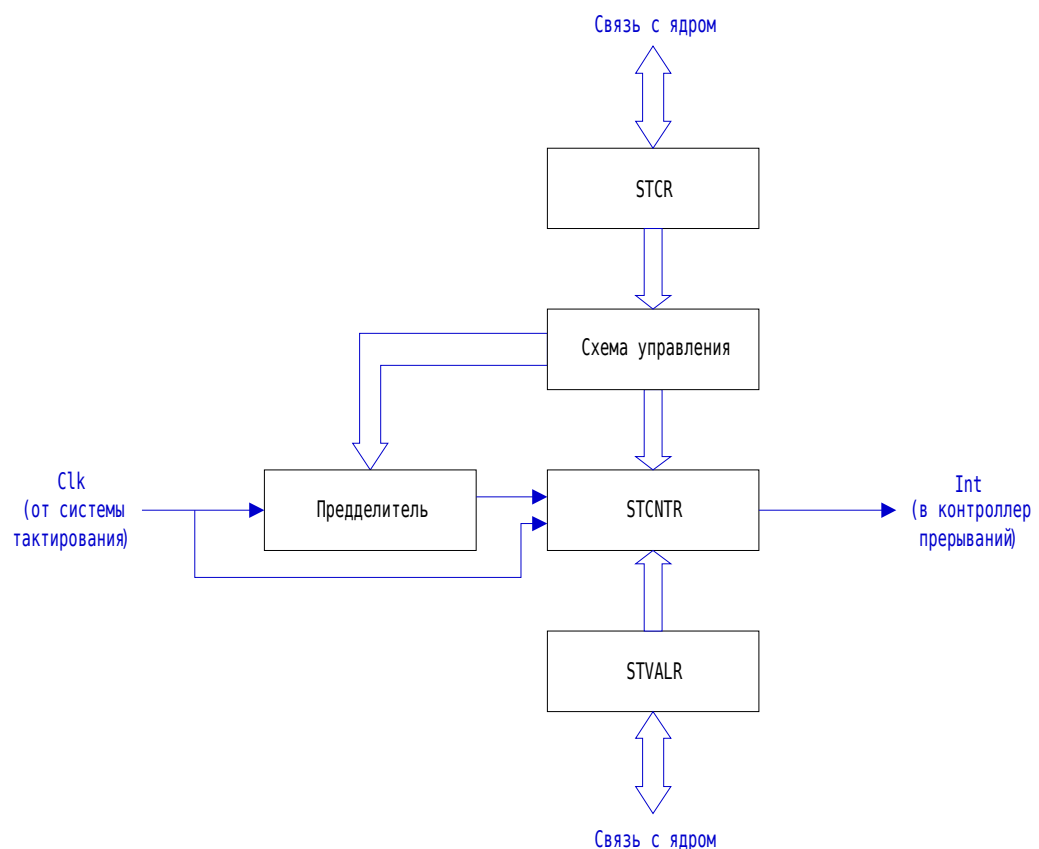
Таймер представляет собой декрементирующий счетчик с делителем тактового сигнала на входе. Начальное значения счетчика записывается в регистр STVALR, управление осуществляется через регистр STCR. По истечении заданного временного интервала формируется запрос на обработку прерывания. На Рис. 11 изображена блок-схема таймера.

Ниже приведены формулы для расчета частоты и периода интервалов, формируемых системным таймером. Значения PREDIV и CNTVAL задаются в регистрах STVALR и STCR соответственно:

$$T = T_{clk} \cdot PREDIV \cdot CNTVAL$$
 - период формируемого интервала

$$F = \frac{F_{clk}}{PREDIV \cdot CNTVAL} \quad \text{- частота следования временных интервалов}$$

Внимание! При значениях  $PREDIV = [0,1]$  тактовый сигнал передается на счетчик в обход делителя. Следовательно в формуле вместо  $PREDIV$  надо подставлять 1.



**Рис. 11 - Блок-схема системного таймера**



### Режимы работы таймера:

- Однократное формирование временного интервала. Таймер запускается пользователем (в бит EN регистра STCR записывается «1»), а по достижении счетчиком таймера значения «0», таймером выдается запрос на обработку прерывания, после этого в бит EN регистра STCR записывается «0» и таймер останавливается до следующей записи значения «1» в бит EN регистра STCR;
- Периодическая генерация временных интервалов. Таймер запускается и останавливается пользователем (в бит EN регистра STCR записывается соответствующее значение). По достижении счетчиком таймера значения «0», таймером выдается запрос обработки прерывания, счетчик перезагружается значением заданным пользователем в регистре STVALR и работа таймера продолжается до того, как пользователь запишет значение «0» в бит EN регистра STCR.

### Особенности работы:

При записи в регистр STVALR новое значение будет передано в счетчик при его очередной перезагрузке, когда он достигнет значения «0».

Если во время работы изменится регистр STCR, то таймер немедленно остановится и запустится с новыми параметрами.

Настоятельно рекомендуется перед изменением режима работы таймера сначала остановить его, записав в бит EN регистра STCR «0», потом задать новые значения в регистре STCR.

Запрещается менять во время работы таймера значение регистра STVALR на 0 и после этого изменять STCR. Это может привести к возникновению запроса обработки прерывания.

## 2. Периферийные устройства

В данном разделе кратко описаны периферийные устройства.

### 2.1. Универсальный асинхронный приёмопередатчик (UART0..*UART3*)

- полнодуплексный режим связи;
- независимые FIFO буферы приемника и передатчика по 32 байта;
- устанавливаемый пользователем тип проверки чётности.

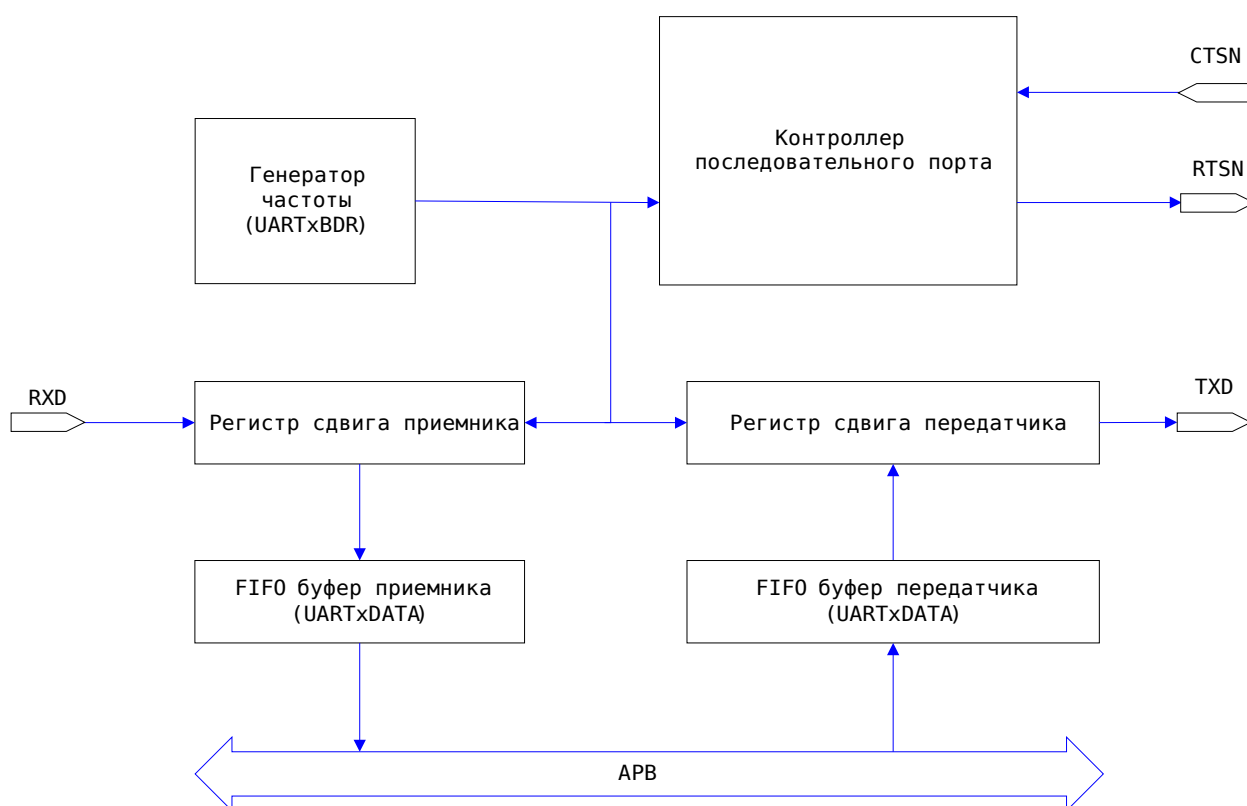


Рис. 12 - Блок-схема UARTx



## 2.2. Последовательный интерфейс I2C

### 2.2.1. I2C master (I2C0)

- работает в режиме «ведущий»;
- поддерживает 7 и 10-ти битную адресацию;
- поддерживаются стандартный (100kb/s) и скоростной (400kb/s) режимы;
- устанавливаемые пользователем величины длительности тактового импульса в состоянии «0» и «1», а также длительности периода предустановки данных;
- устанавливаемое пользователем число адресных байт при обращении к ведомому устройству на шине I2C.

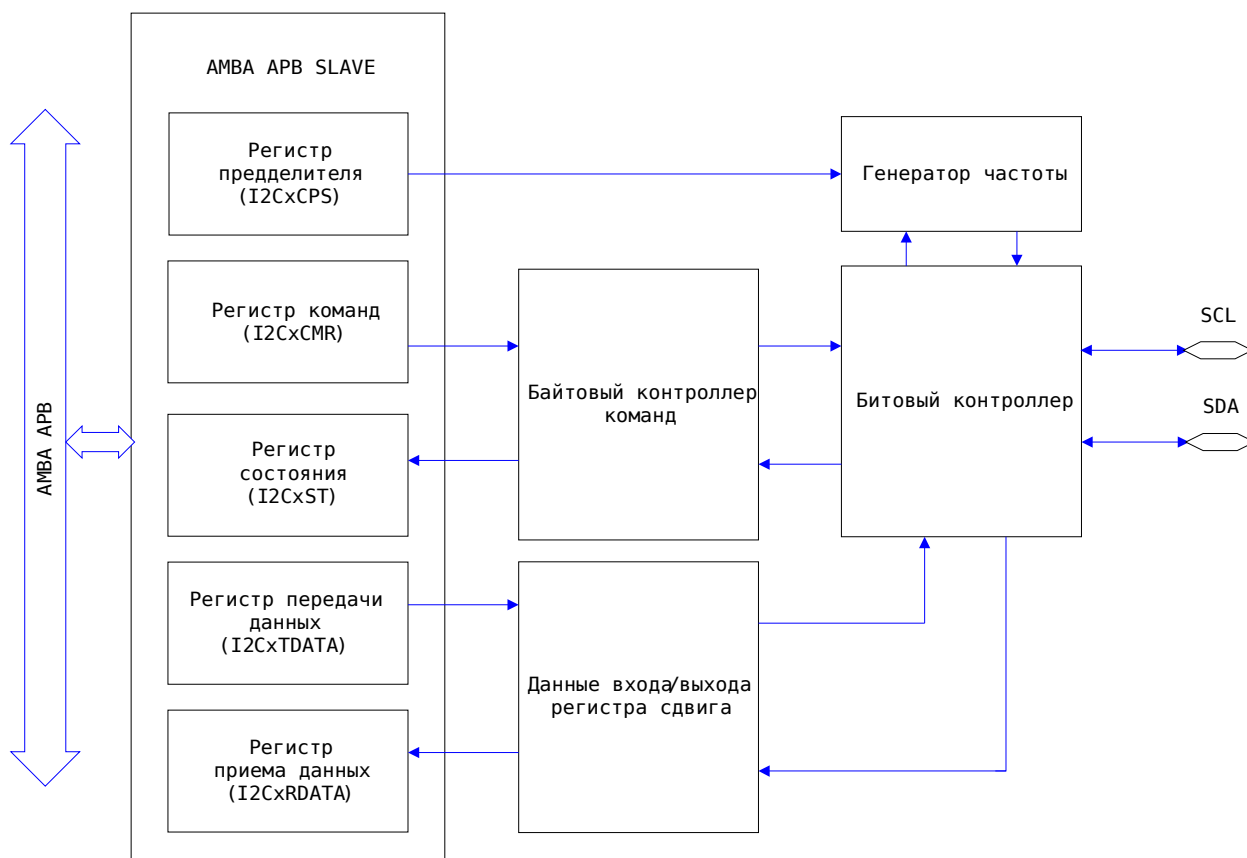


Рис. 13 - Блок-схема I2C0

### 2.2.2. I2C slave (I2C1)

- работает в режиме «ведомый»;
- поддерживает 7 и 10-ти битную адресацию с дополнительным программно задаваемым адресом;
- полностью поддерживаются стандартный (100kb/s) и скоростной (400kb/s) режимы.

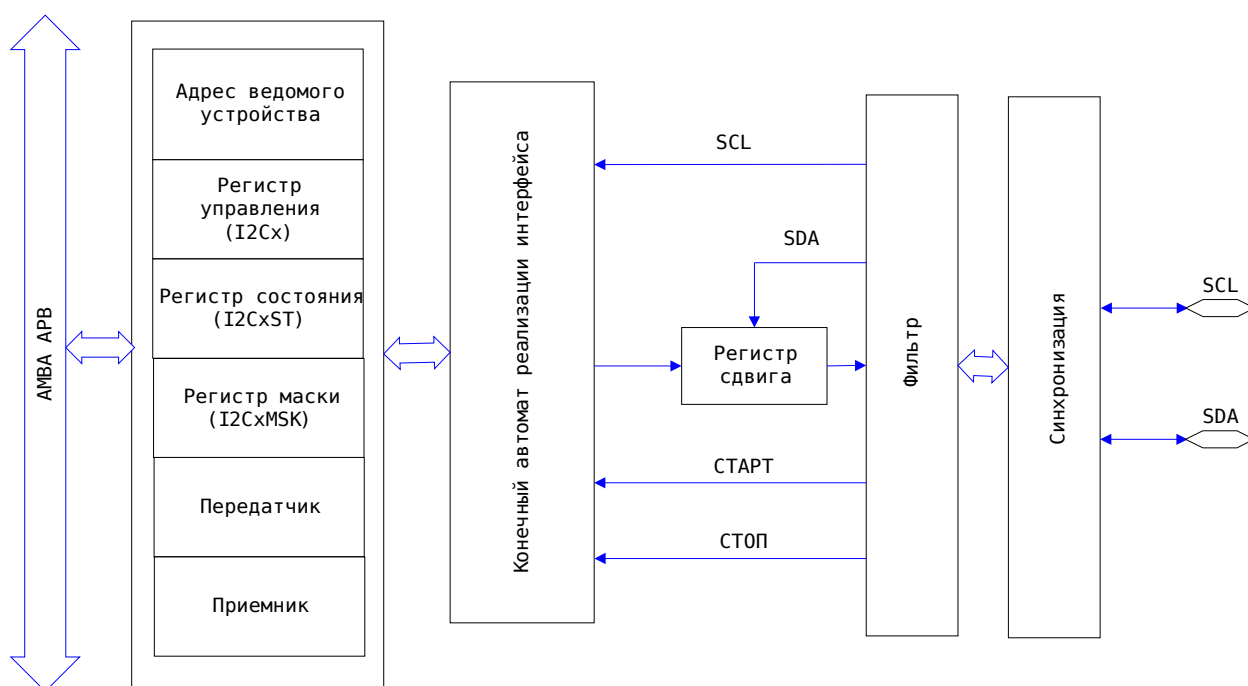


Рис. 14 - Блок-схема I2C1

## 2.3. Последовательный интерфейс SPI (SPIx)

- может работать в режимах "ведущий" или "ведомый";
- поддерживаются все режимы SPI, а также трехпроводной режим, в котором используется одна двунаправленная линия данных;
- настраиваемая длина посылки;
- селектор на 4 ведомых устройства;
- устанавливаемый пользователем формат кадра данных — LSB или MSB;
- устанавливаемые пользователем полярность CPOL (0 или 1) и фаза CPHA (0 или 1) тактового сигнала;
- устанавливаемая пользователем скорость обмена данными.

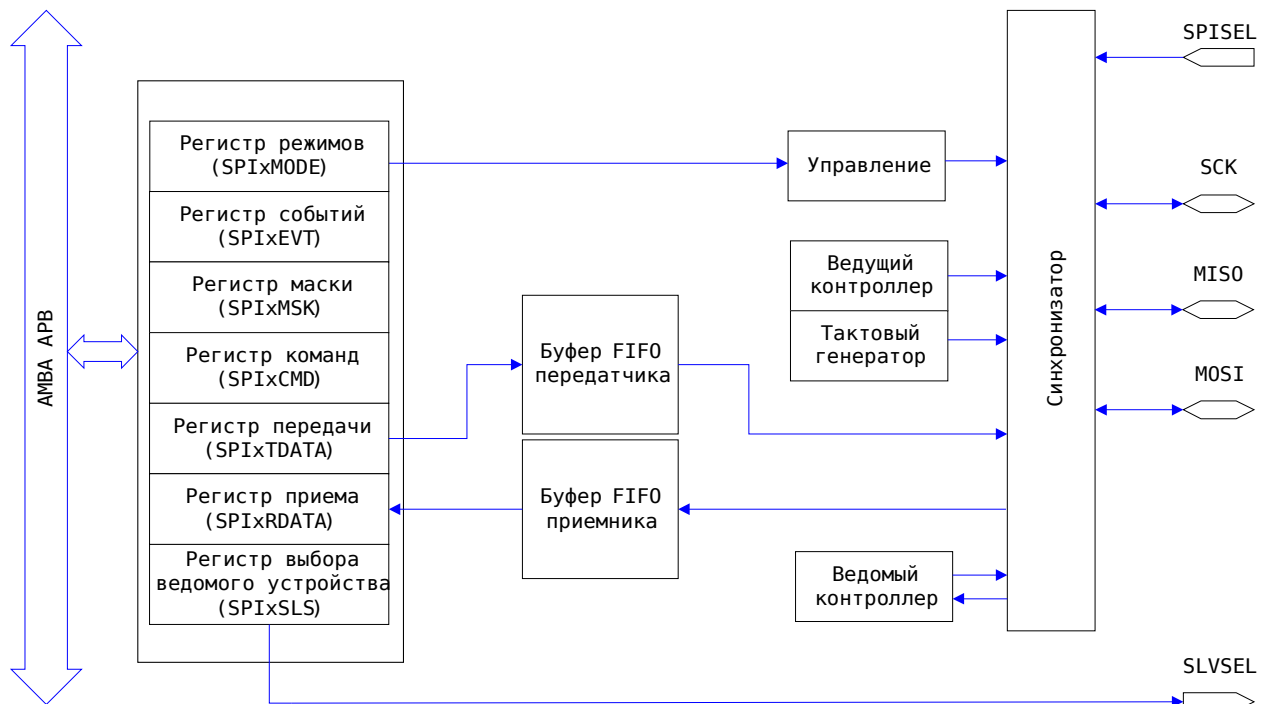


Рис. 15 - Блок-схема SPIx

## 2.4. Сторожевой таймер (WDT)

- предназначен для формирования сигнала глобального сброса в случае отсутствия сигнала сброса, генерируемого ядром МП, в течение заданного времени/интервала.

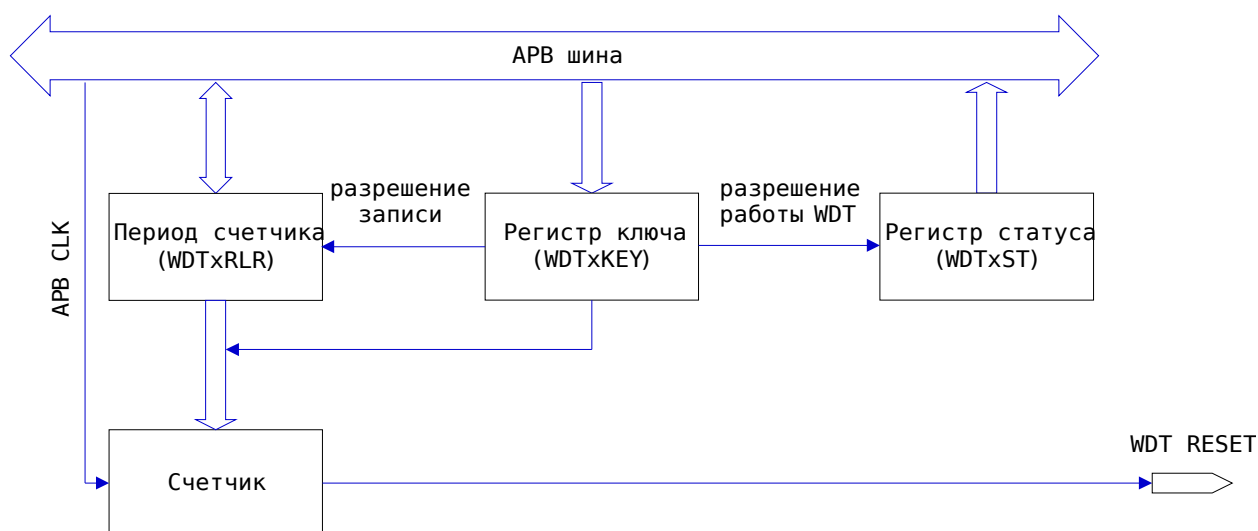


Рис. 16 - Блок-схема сторожевого таймера WDT

## 2.5. Часы реального времени (RTC)

- календарь с секундами, минутами, часами, днем недели, месяцем и годом;
- два программируемых сигнализатора (будильника), срабатывающих при совпадении значений времени.

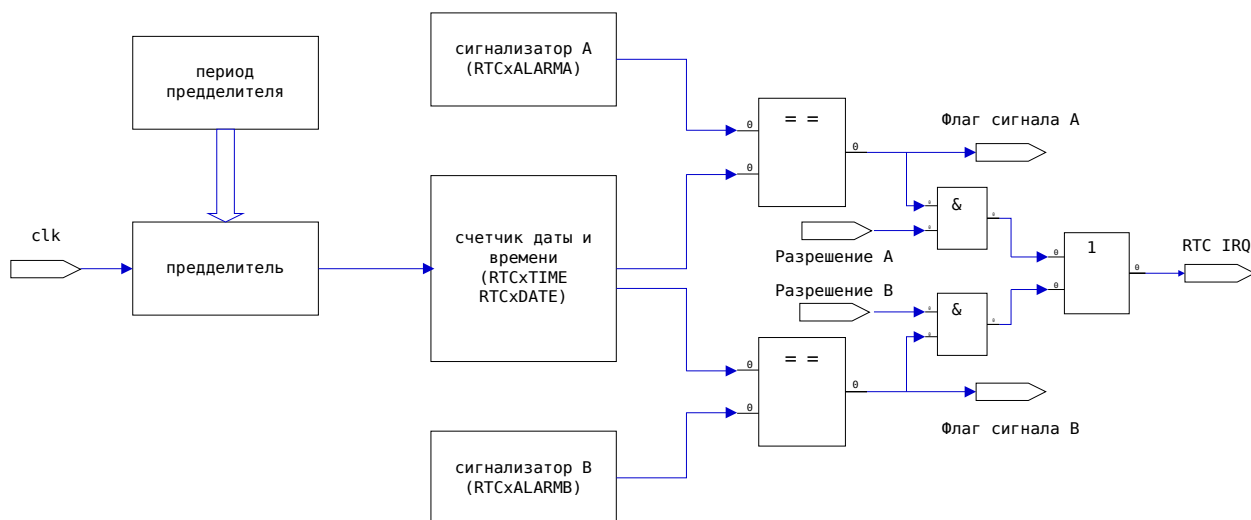
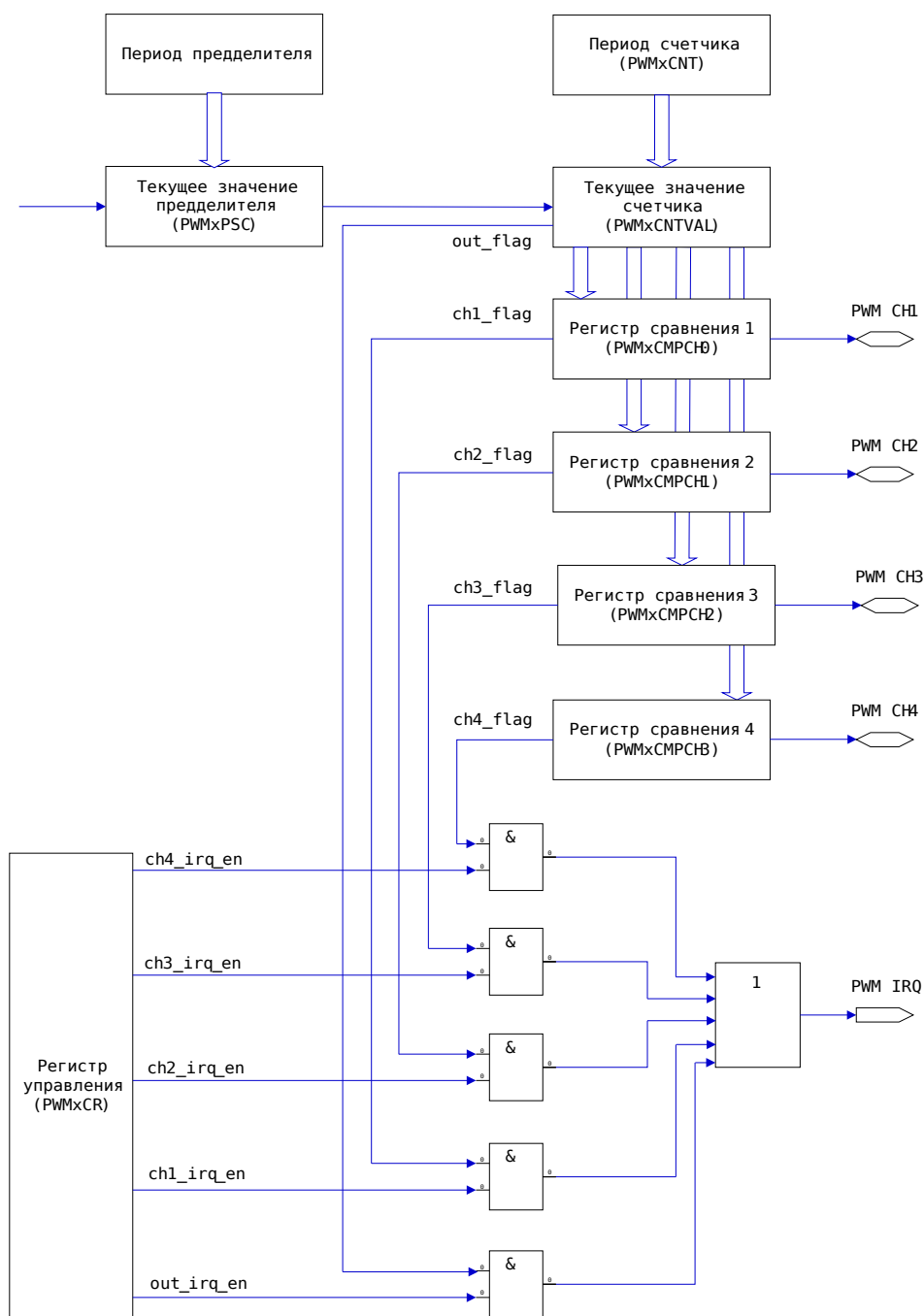


Рис. 17 - Блок-схема RTC

## 2.6. Контроллер ШИМ (PWM0)

- режим генерации одиночного импульса;
- возможность изменения периода счетчика во время его работы (при определенных условиях).



**Рис. 18 - Блок-схема контроллера PWM0**

## 2.7. Порт ввода-вывода (GPIOx)

- каждый бит порта ввода-вывода может быть индивидуально настроен на ввод или вывод, а также дополнительно может являться источником прерывания;
- для формирования прерывания вывод может быть настроен на срабатывание по уровню и по переднему/заднему фронту;
- порт так же предназначен для вывода альтернативных функций, например линий приема и передачи UART.

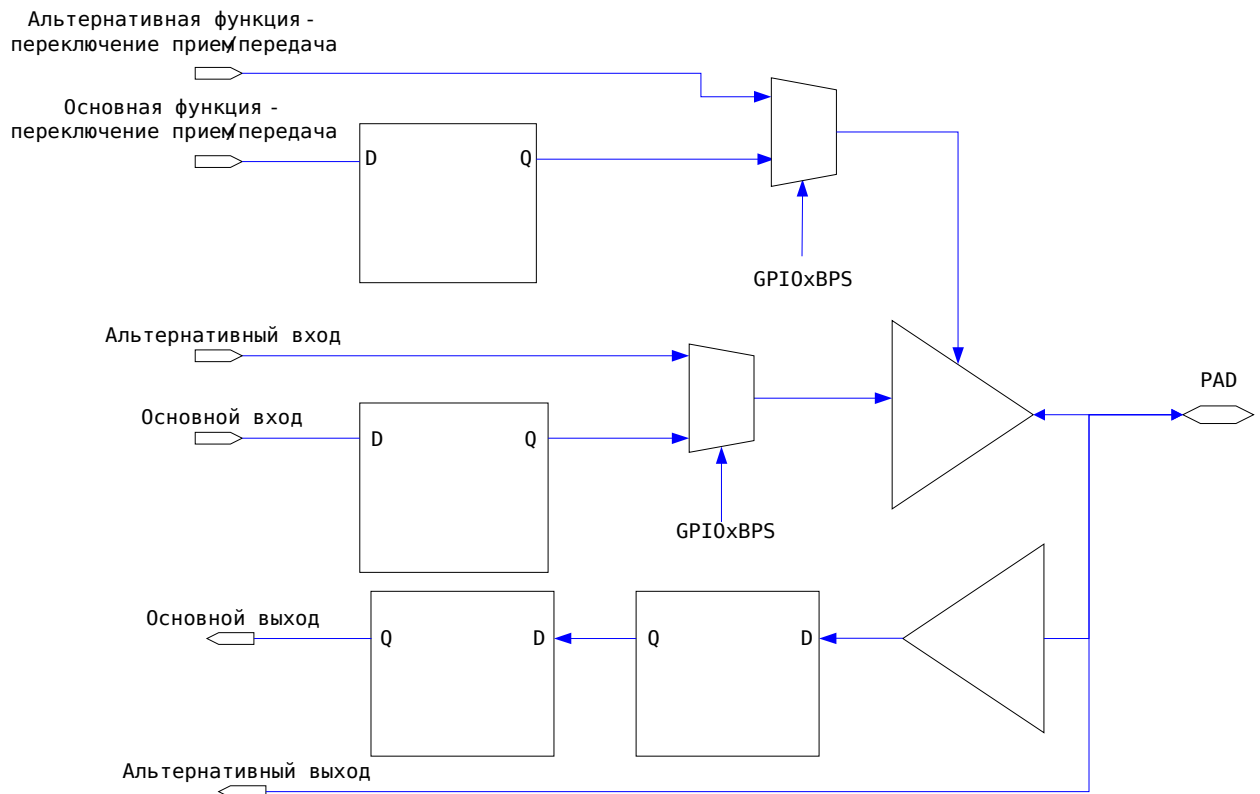


Рис. 19 - Блок-схема одного бита GPIOx

## 2.8. Ethernet контроллер (Ethernet0)

- контроллер поддерживает скорость 10/100 Мбит в полно- и полудуплексном режиме;
- DMA для передатчика и для приемника;
- поддерживает МП и RMP интерфейсы.

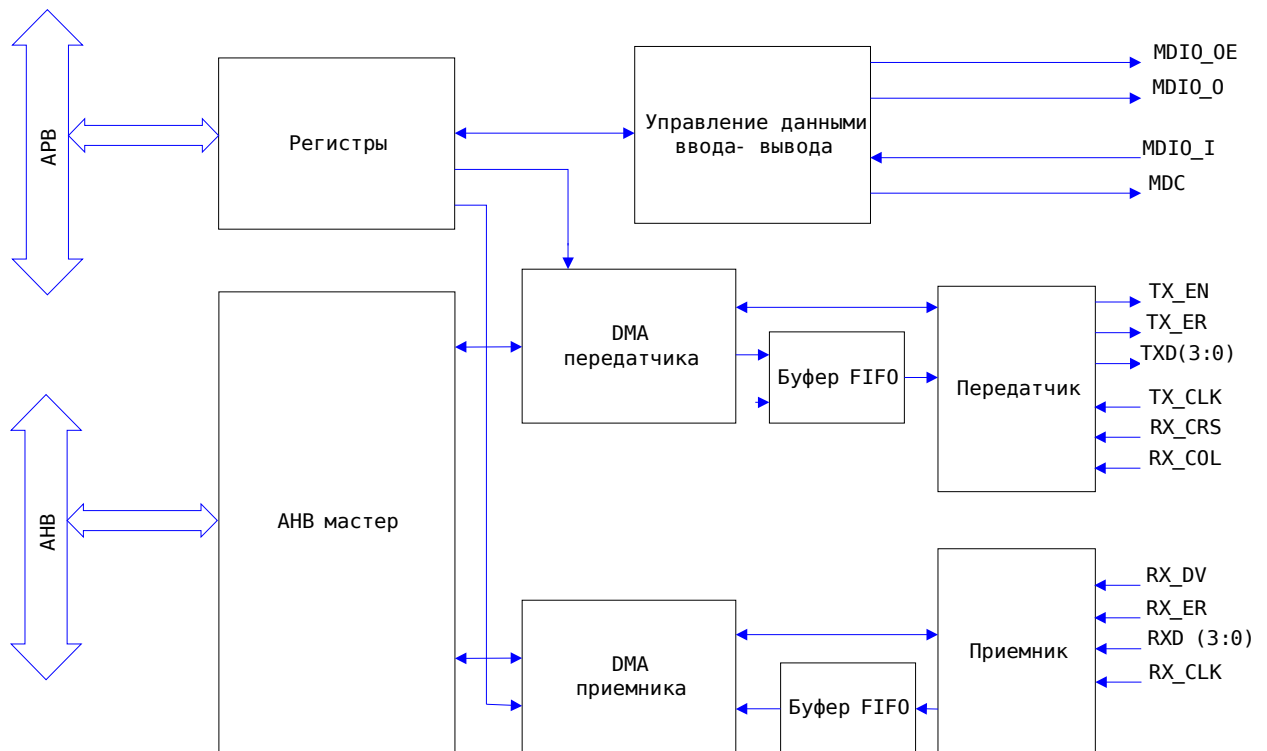


Рис. 20 - Блок-схема контроллера Ethernet0



## 2.9. Таймер общего назначения (GPTIMx)

- можно использовать как внешний так и системный тактовый сигнал;
- представляет собой декрементирующий счетчик с предделителем.

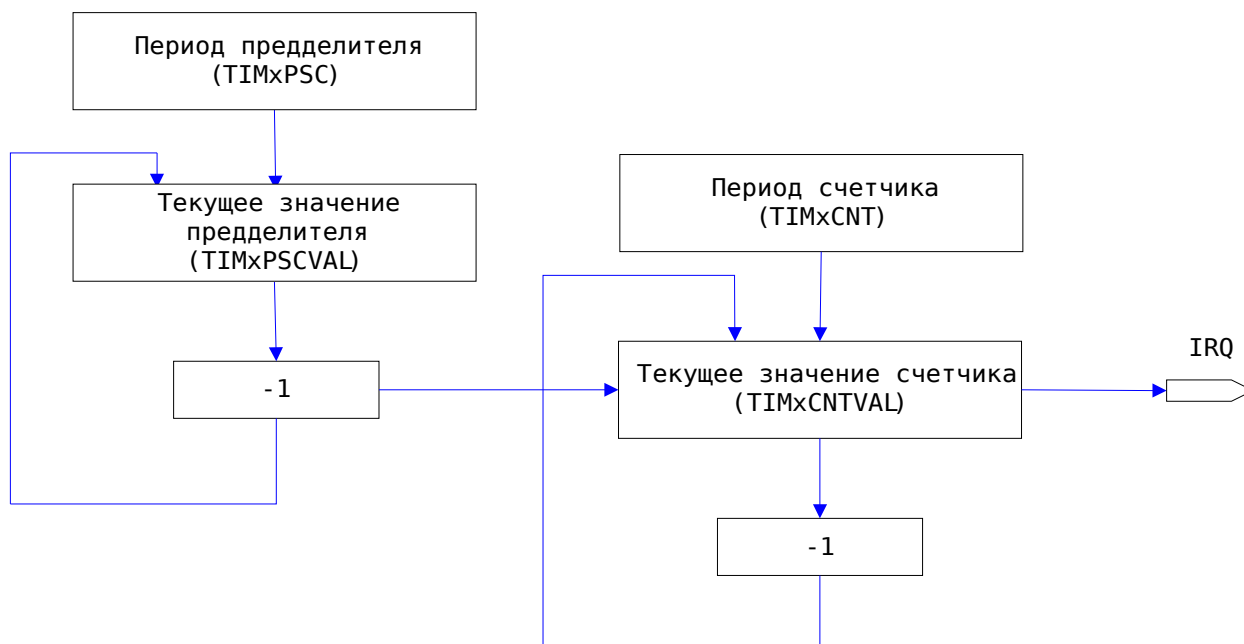


Рис. 21 - Блок-схема GPTIMx

## 2.10. Последовательный интерфейс I2S (I2S0)

- работает в режиме «ведущий»;
- осуществляет только прием данных от «ведомых» устройств.

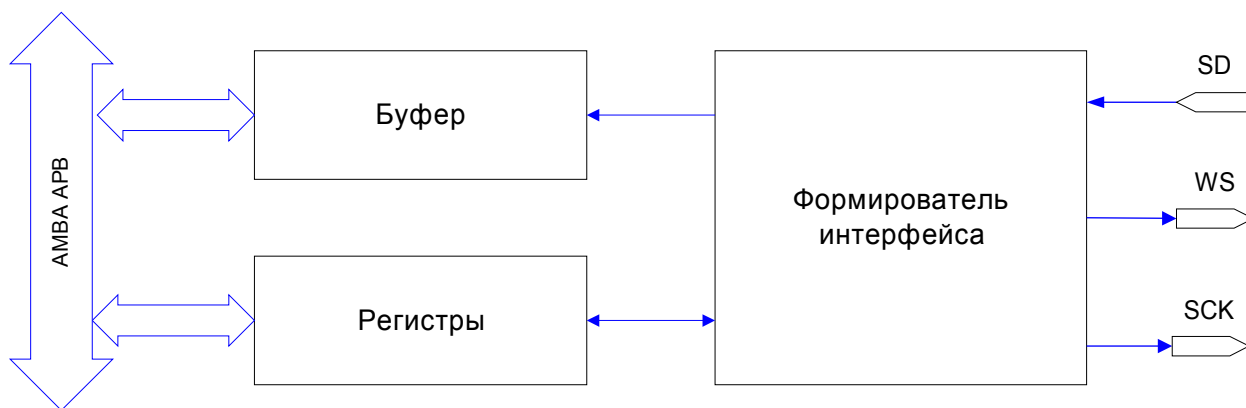


Рис. 22 - Блок-схема I2S0

## 2.11. USB (USB0)

- реализует USB 1.1 FS, совместим с микросхемой Fairchild USB1T11A;
- работает только в режиме “device”;
- поддержка 4 классов каналов: поточный, управляющий, изохронный, прерывание;
- поддерживает LS (1,5 Мбит/с) и FS (12 Мбит/с) режим;
- включает 4-х канальный буфер обмена (FIFO);
- для работы необходим внешний тактовый генератор 48МГц.

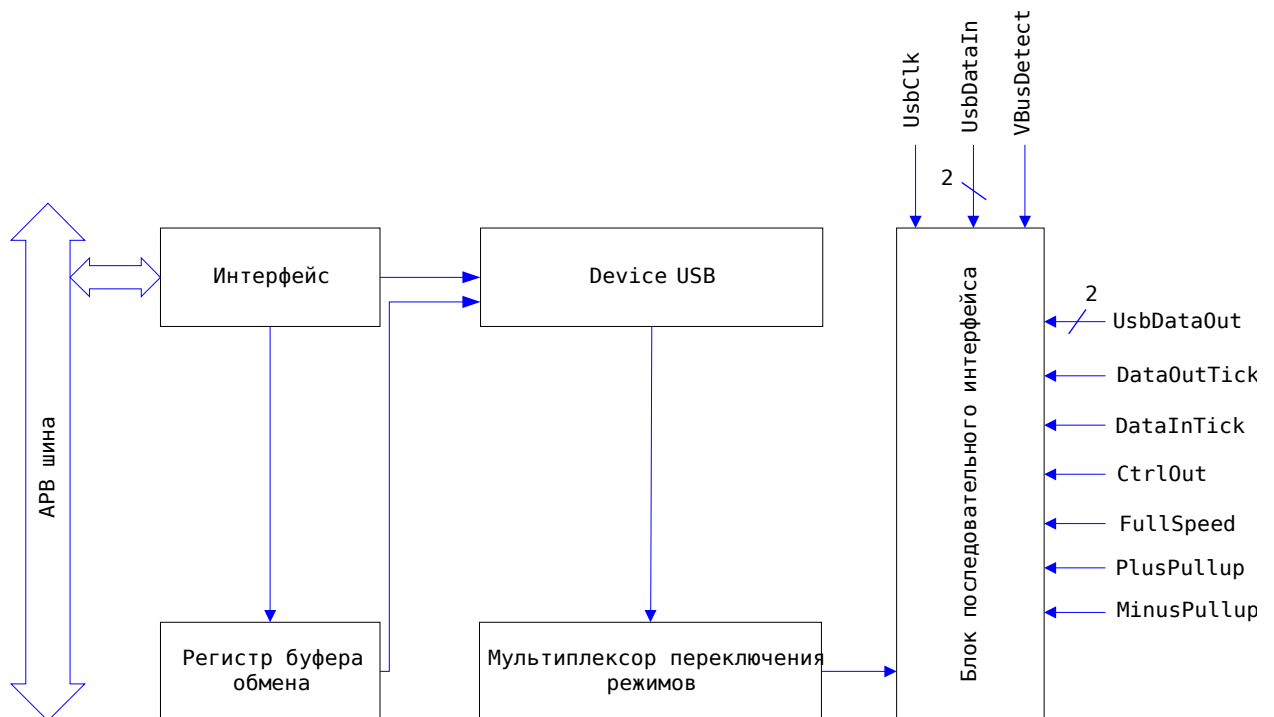


Рис. 23 - Блок-схема USB0



## 3. Назначение выводов процессора





## 4. Маркировка мультиклеточных процессоров

	МС	р	04	1	11	001	01	— Q	208	С
<b>Тип продукта</b>	_____									
р – процессор										
с – ядро										
<b>Число клеток ядра</b>	_____									
04, 08, 16 – четыре, восемь, шестнадцать клеток										
<b>Тип ядра</b>	_____									
0 – с фиксированной запятой										
1 – с плавающей запятой										
<b>Версия ядра</b>	_____									
11 – максимальная производительность при низком энергопотреблении										
21 – сверхнизкое энергопотреблении при высокой производительности										
<b>Номер программной комплектации</b>	_____									
<b>Исполнение</b>	_____									
01 – общепромышленное назначение										
<b>Тип корпуса</b>	_____									
Q – QFP										
LQ – LQFP										
TQ – TQFP										
PQ – PQFP										
CQ – CQFP										
B – BGA										
CB – CBGA										
PB – PBGA										
<b>Количество выводов</b>	_____									
<b>Температурный диапазон</b>	_____									
С – 0° - +70°C										
I – -40° - +85°C										