

**Микросхема интегральная 1892ХД5Т**  
**Техническое описание**

**Версия 1.3**

**3.02.2015**

## **АННОТАЦИЯ**

Данный документ является техническим описанием микросхемы 1892ХД5Т - контроллера удаленных периферийных устройств с последовательным каналом SpaceWire (КПУ).

## СОДЕРЖАНИЕ

Аннотация .....	2
Содержание .....	3
1. Основные особенности .....	4
2. Назначение выводов .....	5
3. Функциональное описание .....	7
3.1. Управление тактовыми частотами и мощностью потребления .....	7
3.2. Последовательный порт управления .....	8
3.3. Интерфейс SpaceWire .....	11
3.3.1 Установление соединения .....	12
3.3.2 Настройка скорости передачи .....	12
3.3.3 Просмотр текущего состояния .....	13
3.3.4 Использование регистров маршрутизации .....	15
3.3.5 Пример настройки таблицы маршрутизации с путевой адресацией .....	15
3.3.6 Пример настройки таблицы маршрутизации без путевой адресации .....	17
3.4. Порт периферийных устройств .....	20
3.4.1. Режим GPIO/MSPI .....	20
3.4.2. Режим PP .....	21
3.4.3. Режим MBOX .....	23
3.5. Микропроцессорный интерфейс $\mu P$ .....	25
4. Регистры управления .....	28
4.1. Внутреннее адресное пространство .....	28
4.2. SWRST .....	29
4.3. DEVID .....	29
4.4. INT .....	29
4.4. INTE .....	30
4.5. INTR .....	30
4.6. CFG .....	30
4.7. CLKO .....	30
4.9. GPIO0_DIR, GPIO1_DIR .....	31
4.10 GPIO0_DAT, GPIO1_DAT .....	31
4.11. PORT0_CFG1, PORT1_CFG1 .....	31
4.12. PORT0_CFG2, PORT1_CFG2 .....	32
4.13. PORT0_CFG3, PORT1_CFG3 .....	32
4.14. FIFO0_TH, FIFO1_TH .....	32
4.15. FIFO0_STAT, FIFO1_STAT .....	32
4.16. MSPI0_DAT, MSPI1_DAT .....	32
4. Корпус .....	40

## 1 ОСНОВНЫЕ ОСОБЕННОСТИ

Интерфейсы управления контроллером:

- SpaceWire
- SPI (вспомогательный)

Два интерфейса управления периферийными устройствами со следующими функциями:

- 16 сигналов ввода/вывода общего назначения и SPI-мастер
- 16-разрядный двунаправленный параллельный порт
- 16-разрядный двунаправленный почтовый ящик

Режим двунаправленной микропроцессорной шины:

- Ширина шины данных: 16 бит
- Ширина шины адреса: 16 бит
- Поддерживаемые режимы: Intel, Motorola, Multicore

Напряжения питания:

- Питание ядра: 1.8 В
- Питание периферии: 3.3 В
- Питание приемопередатчиков SpaceWire: 3.3 В

## 2 НАЗНАЧЕНИЕ ВЫВОДОВ

Назначение выводов микросхемы приведено в таблице ниже.

**Таблица 2.1 Назначение выводов микросхемы.**

Имя	Тип	Описание	КОЛ-ВО ВЫВОДОВ
<b>Питание</b>			<b>43</b>
VDD	PWR	питание ядра 1.8 В	7
DVDD	PWR	питание периферии 3.3 В	10
SVDD	PWR	питание приемопередатчиков SpaceWire W 3.3В	4
GND	GND	земля ядра	7
DGND	GND	земля периферии	11
SGND	GND	земля приемопередатчиков SpaceWire	4
<b>SpW</b>			<b>16</b>
SINp[0]	I (LVDS)	вход приемника SpW канала А: строб данных, прямой	1
SINn[0]	I (LVDS)	вход приемника SpW канала А: строб данных, инверсный	1
DINp[0]	I (LVDS)	вход приемника SpW канала А: данные, прямой	1
DINn[0]	I (LVDS)	вход приемника SpW канала А: данные, инверсный	1
SOUTp[0]	O (LVDS)	выход передатчика SpW канала А: строб данных, прямой	1
SOUTn[0]	O (LVDS)	выход передатчика SpW канала А: строб данных, инверсный	1
DOUp[0]	O (LVDS)	выход передатчика SpW канала А: данные, прямой	1
DOUn[0]	O (LVDS)	выход передатчика SpW канала А: данные, инверсный	1
SINp[1]	I (LVDS)	вход приемника SpW канала В: строб данных, прямой	1
SINn[1]	I (LVDS)	вход приемника SpW канала В: строб данных, инверсный	1
DINp[1]	I (LVDS)	вход приемника SpW канала В: данные, прямой	1
DINn[1]	I (LVDS)	вход приемника SpW канала В: данные, инверсный	1
SOUTp[1]	O (LVDS)	выход передатчика SpW канала В: строб данных, прямой	1
SOUTn[1]	O (LVDS)	выход передатчика SpW канала В: строб данных, инверсный	1
DOUp[1]	O (LVDS)	выход передатчика SpW канала В: данные, прямой	1
DOUn[1]	O (LVDS)	выход передатчика SpW канала В: данные, инверсный	1
<b>SPI</b>			<b>4</b>
SCSn	I	вход “выбор кристалла”	1
SCK	I	вход тактовой частоты	1
MOSI	I	входные данные	1
MISO	O	выходные данные	1
<b>GPIO/MSPI/PP/mP</b>			<b>40 (20)</b>
GPIO1_ADDR[15:0]	IO	в режиме GPIO/MSPI: сигналы общего назначения нулевого периферийного порта в режиме PP/MBOX: двунаправленная шина данных нулевого периферийного порта в режиме $\mu$ P: выходная шина адреса	1
OCTRA0	IO	в режиме GPIO/MSPI: выход тактовой частоты SPI нулевого периферийного порта в режиме PP: выход строб записи нулевого периферийного порта в режиме $\mu$ P Intel, Multicore: выход строб записи в режиме $\mu$ P Motorola: выход строб данных в режиме MBOX: выход готовности почтового ящика для записи нулевого периферийного порта	1
OCTRB0	O	в режиме GPIO/MSPI: выходные данные SPI нулевого периферийного порта в режиме PP: выход строб чтения нулевого периферийного порта в режиме $\mu$ P Intel, Multicore: выход строб чтения в режиме $\mu$ P Motorola: выход запись(1)/чтение(0) в режиме MBOX: выход готовности почтового ящика для чтения нулевого периферийного порта	1

Имя	Тип	Описание	КОЛ-ВО ВЫВОДОВ
ICTRA0	I	в режиме GPIO/MSPI: входные данные SPI нулевого периферийного порта в режиме PP: вход готовности нулевого периферийного порта в режиме $\mu$ P Intel: вход готовности нулевого адресного пространства в режиме $\mu$ P Multicore, Motorola: вход подтверждения нулевого адресного пространства в режиме MBOX: строб записи почтового ящика нулевого периферийного порта	1
ICTRB0	I	в режиме GPIO/MSPI/PP/ $\mu$ P: запрос нулевого прерывания в режиме MBOX: строб чтения почтового ящика нулевого периферийного порта	1
GPIO2_DATA[15:0]	IO	в режиме GPIO/MSPI: сигналы общего назначения первого периферийного порта в режиме $\mu$ P/PP/MBOX: двунаправленная шина данных первого периферийного порта	1
OCTRA1	O	в режиме GPIO/MSPI: выход тактовой частоты SPI первого периферийного порта в режиме PP: выход строб записи первого периферийного порта в режиме $\mu$ P: сигнал выбора первого адресного пространства в режиме MBOX: выход готовности почтового ящика для записи первого периферийного порта	1
OCTRB1	O	в режиме GPIO/MSPI: выходные данные SPI первого периферийного порта в режиме PP: выход строб чтения первого периферийного порта в режиме $\mu$ P: сигнал выбора первого адресного пространства в режиме MBOX: выход готовности почтового ящика для чтения первого периферийного порта	1
ICTRA1	I	в режиме GPIO/MSPI: входные данные SPI первого периферийного порта в режиме PP: вход готовности первого периферийного порта в режиме $\mu$ P Intel: вход готовности первого адресного пространства в режиме $\mu$ P Multicore, Motorola: вход подтверждения первого адресного пространства в режиме MBOX: строб записи почтового ящика первого периферийного порта	1
ICTRB1	I	в режиме GPIO/MSPI/PP/ $\mu$ P: запрос первого прерывания в режиме MBOX: строб чтения почтового ящика первого периферийного порта	1
<b>Прочие</b>			<b>8</b>
RSTn	I	сигнал начальной установки.	1
MODE	I	выбор интерфейса: 0 – SPI, 1 – SpW	1
PLL_EN	I	включение/выключение внутренних PLL 1 – PLL включена 0 – PLL выключена	1
IRQ	O	сигнал прерывания	1
XTALI	I	вход для подключения кварцевого резонатора или внешнего генератора тактовой частоты	1
XTALO	O	выход для подключения кварцевого резонатора	1
RSTO	O	сигнала начальной установки периферийных устройств	1
CLKO	O	тактовая частота периферийных устройств	1

### 3 ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ

Функциональная диаграмма микросхемы приведена на рисунке 3.1.

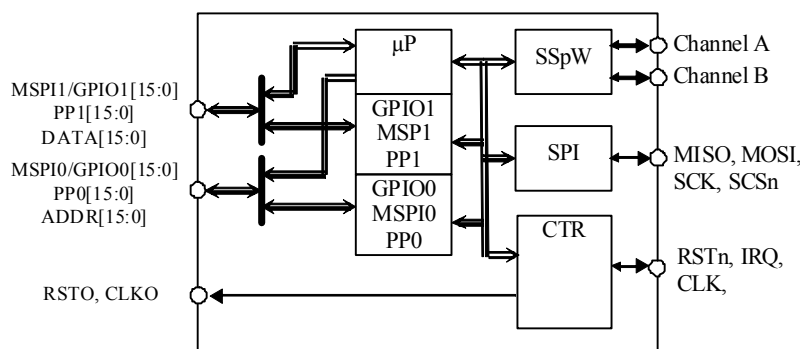


Рисунок 3.1 Функциональная диаграмма микросхемы.

SSpW – двухканальный Slave-контроллер SpaceWire. Обеспечивает управление микросхемой по протоколу RMAP.

SPI – контроллер последовательного порта SPI-slave. Предназначен для локального управления микросхемой.

CTR – блок конфигурации микросхемы. Содержит регистры управления и блоки формирования программного сброса, прерываний, а также выходных сигналов начальной установки и тактовой частоты периферийных устройств.

GPIO/MSP/PP – универсальный контроллер сигналов общего назначения, последовательного (SPI-master) порта и параллельного порта для подключения периферийных устройств.

μP – контроллер 16-разрядной микропроцессорной шины для подключения периферийных устройств.

#### 3.1 УПРАВЛЕНИЕ ТАКТОВЫМИ ЧАСТОТАМИ И МОЩНОСТЬЮ ПОТРЕБЛЕНИЯ.

Тактирование внутренних блоков осуществляется с помощью сигналов тактовых частот, названия и назначения которых представлены в таблице 3.1

Таблица 3.1

Название тактового сигнала	Назначение
LCLK	локальная тактовая частота порта SpaceWire
HCLK	тактовая частота шины АНВ
TX_0_CLK	тактовая частота нулевого канала передачи SpaceWire
TX_1_CLK	тактовая частота первого канала передачи SpaceWire

Имеется возможность выбрать один из нескольких режимов тактирования микросхемы. Выбор режима тактирования осуществляется с помощью внешних сигналов MODE, PLL\_EN, SDI и SCK. Доступные режимы тактирования представлены в таблице 3.2

Таблица 3.2 Режимы тактирования.

Входные сигналы					Внутренние тактовые частоты		
MODE	PLL_EN	SCK	SDI	XTAL1, МГц	HCLK, МГц	LCLK, МГц	TX_CLK1/2 (max), МГц
1	1	0	0	10	50	100	200
1	1	0	1	20	50	100	200
1	1	1	0	10	20	40	80
1	1	1	1	20	20	40	80
1	0	0	0	100	50	100	5
0	1	–	–	5	25	–	0
0	0	–	–	$f^*$	$f$	–	0
–	–	–	–	–	–	–	–

\*–  $f \leq 50$  МГц

### 3.2 ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ УПРАВЛЕНИЯ

Последовательный порт управления позволяет обращаться к регистрам управления микросхемы. Порт работает в режиме SPI-slave.

Обмен по последовательному порту управления осуществляется с помощью 40-разрядных команд. Команды управления считываются с входа MOSI по переднему фронту тактового сигнала SCK. Выходные данные выдаются на выход MISO по заднему фронту тактового сигнала SCK. Команды и выходные данные передаются старшим значимым битом вперед. Если длина команды управления меньше 40 бит, то команда не выполняется. Если длина команды управления больше 40 бит, то команда выполняется в соответствии с первыми 40 битами, а остальные биты игнорируются.

Обмен с регистрами управления двухуровневый: команды управления непосредственно обращаются к внутренним регистрам адреса и данных последовательного порта, доступ к регистрам управления микросхемы осуществляется посредством этих регистров адреса и данных.

Команды последовательного порта состоят из семибитного кода операции и 33-битного параметра. Поддерживаемые команды управления последовательного порта приведены в таблице 3.3. Значениями параметров являются 33-битный адрес, 32-битные данные или 16-битные данные.

Таблица 3.3

Команда	Код	Параметр	Назначение
1	2	3	4
SETA	0000000	addr	установка адреса в регистре адреса порта
GETA	0000001	xxxxxx	чтение адреса из регистра адреса порта
GETD	0000010	xxxxxx	чтение данных из регистра данных порта
FETCH	0000100	addr	установка адреса в регистре адреса порта с выборкой значения регистра управления по этому адресу в регистр данных порта
WR	0000101	data	запись данных в регистр управления по адресу в регистре адреса порта
WRI	0000110	data	запись данных в регистр управления по адресу в регистре адреса порта с последующей инкрементацией регистра адреса порта
RMW	0000111	mask, data	запись данных по маске в регистр управления по адресу в регистре адреса порта

Коды команд, не перечисленные в таблице, игнорируются.



В таблицах 3.3, 3.4 и 3.5 соответственно представлены форматы команд последовательного интерфейса, в случаях, когда значением параметра является 33-разрядный адрес, 32-разрядные данные и 16-разрядные данные. Примерами таких команд являются SETA, FETCH, FETCHI, WR, и WRI.

**Таблица 3.3**

Команда			Параметр						
39	...	33	32	31	30	...	2	1	0
cmd[6]	...	cmd[0]	addr[32]	addr[31]	addr[30]	...	addr[2]	addr[1]	addr[0]

**Таблица 3.4**

Команда			Параметр						
39	...	33	32	31	30	...	2	1	0
cmd[6]	...	cmd[0]	0	data[31]	data[30]	...	data[2]	data[1]	data[0]

**Таблица 3.5**

Команда			Параметр								
39	...	33	32	31	...	16	15	...	2	1	0
cmd[6]	...	cmd[0]	0	0	...	0	data[15]	...	data[2]	data[1]	data[0]

Команда RMW в качестве значения параметра использует 16-разрядные данные и их маску. Данная команда поддерживается только для 16-разрядных регистров управления. Разряды маски, установленные в состояние логической единицы, указывают разряды регистров управления, которые будут переписаны в соответствии с передаваемыми данными. Разряды маски, установленные в состояние логического нуля, не изменяют соответствующие разряды регистров управления. Результатом выполнения команды RMW является изменение нужных разрядов в регистре управления. Выборка значения этого регистра в регистр данных порта не производится. Формат команды RMW представлен в таблице 3.6.

**Таблица 3.6**

Команда			Параметр								
39	...	33	32	31	...	17	16	15	...	1	0
cmd[6]	...	cmd[0]	0	mask[15]	...	mask[1]	mask[0]	data[15]	...	data[1]	data[0]

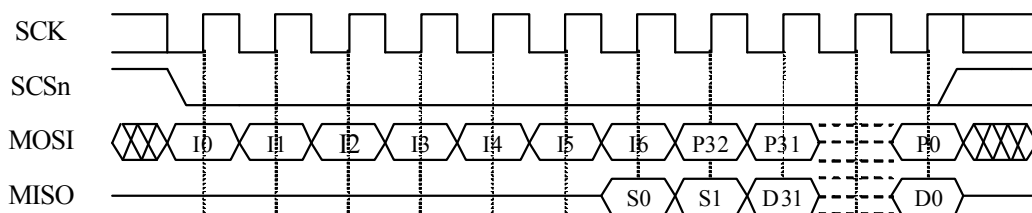
Ответ на текущую команду представляет собой 34-разрядное слово. Выдаваться оно начинает с седьмого такта SCK, а первые шесть тактов вывод MISO находится в высокоимпедансном состоянии. Далее, в зависимости от команды следуют один или два разряда статуса и выходные данные. Выходные данные размещаются в младших разрядах 34-разрядного ответного слова. Назначение разрядов статуса представлено в таблице 3.7.

**Таблица 3.7 Назначение разрядов статуса ответного слова**

Имя разряда статуса	Назначение
S0	Признак действительности содержимого регистра данных порта 1 – регистр данных содержит корректные данные 0 – регистр данных содержит некорректные данные *
S1	Признак пропуска текущей команды 1 – текущая команда будет проигнорирована ** 0 – текущая команда будет выполнена
Примечания * Считается, что регистр данных содержит некорректные данные, если после приема 40 разряда команды FETCH чтение новых данных еще не завершено или оно завершилось с ошибкой ** Игнорируются только текущие команды FETCH, WR, WRI или RMW, если предыдущей командой также была одна из названных и к моменту передачи последнего разряда кода текущей команды, выполнение предыдущей еще не завершено. Команды SETA, GETA и GETD выполняются всегда	

Последовательный порт совместим с интерфейсом SPI нулевой модификации в ведомом режиме. Команды управления и выходные данные передаются в одном цикле. При этом выходные данные являются ответом на передаваемую команду.

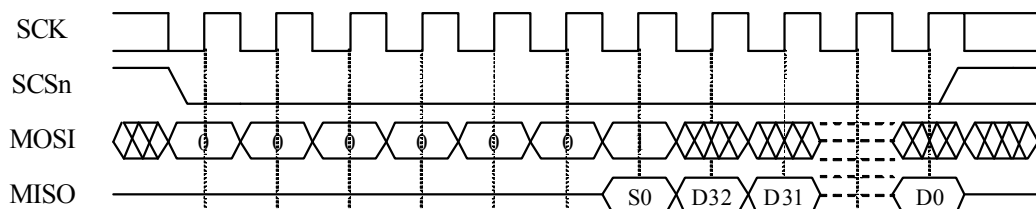
Временная диаграмма работы последовательного интерфейса при выполнении команды SETA и FETCH показана на рисунке 3.2.



I – код команды, P – параметр команды, S – значение разрядов статуса, D – значение выходных данных.

**Рисунок 3.2**

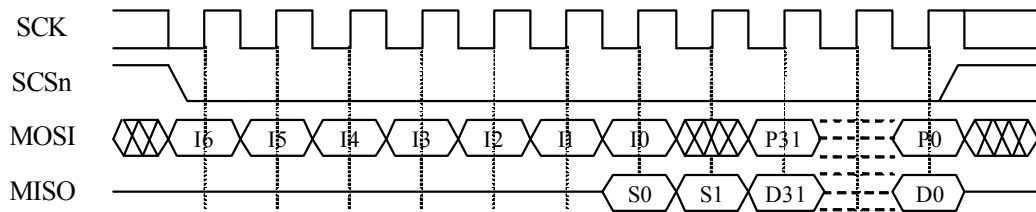
Временная диаграмма работы последовательного интерфейса при выполнении команды GETA показана на рисунке 3.3.



P – параметр команды, S – значение разрядов статуса, D – значение выходных данных.

**Рисунок 3.3**

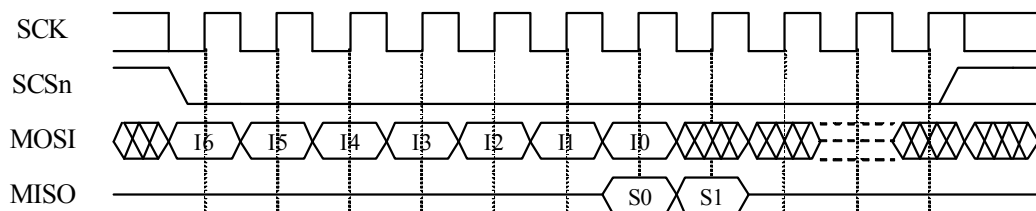
Временная диаграмма работы последовательного интерфейса в этом режиме при выполнении команд WR, WRI, RMW и GETD показана на рисунке 3.4.



I – код команды, P – параметр команды, S – значение разрядов статуса, D – значение выходных данных.

Рисунок 3.4

Временная диаграмма работы последовательного порта в случае подачи некорректного кода команды представлена на рисунке 3.5.



I – код команды, P – параметр команды, S – значение разрядов статуса.

Рисунок 3.5

Тактовый сигнал SCK формируется внешним устройством и может быть асинхронным по отношению к другим тактовым сигналам микросхемы. Период сигнала SCK не должен быть меньше периода внутреннего тактового сигнала HCLK.

Длительность сигнала выборки SCSn должна составлять 40 такта SCK. Если длительность SCSn меньше, чем 40 такта SCK, то выполнение команды прекращается. Если длительность SCSn больше, чем 40 такта SCK, то лишние такты игнорируются.

При выполнении команд SETA, FETCH, WR и WRI на выход MISO выдается текущее значение регистра данных последовательного интерфейса (значение регистра управления, считанного с помощью команды FETCH). Это позволяет лишний раз не использовать команду GETD в цепочках команд FETCH/SETA/WR/WRI.

### 3.3 ИНТЕРФЕЙС SPACEWIRE

Микросхема содержит двухканальный контроллер SpaceWire, обеспечивающий управление микросхемой по протоколу RMAP (Remote Memory Access Protocol).

Протокол RMAP обеспечивает удаленное обращение к регистрам управления через адресное пространство шины АНВ, при этом 16-разрядные регистры управления отображаются в младшие биты 32-разрядной шины АНВ. Соответственно, адрес регистра управления на шине АНВ формируется добавлением двух младших разрядов к адресу регистра, указанному в таблице 4.1.

Контроллер поддерживает пакетные обращения RMAP с инкрементом и без инкремента адреса. Пакетные обращения RMAP отображаются на один или несколько пакетов на шине АНВ, в зависимости от размера передаваемых/запрашиваемых данных в пакете

RMAP и максимального размера пакета на шине АНВ. Максимальный размер пакета на шине АНВ задается программно в регистрах управления контроллера SpaceWire.

Реализация протокола RMAP в микросхеме 1892ХД5Т соответствует стандарту ECSS-E-ST-50-52С, но имеет следующие особенности:

- длина поля данных в пакете RMAP должна быть кратной четырем байтам;
- выполнение записи с проверкой CRC возможно только для пакетов RMAP общей длиной не более 28 байт;
- не реализована команда «чтение-модификация-запись».

### 3.3.1 Установление соединения

По умолчанию в Slave-контроллере включен режим автоматической установки соединения на скорости 10 Мбит/с. Соединение будет автоматически установлено при разрешенной автоматической установке соединения во втором устройстве.

Для настройки установки соединения в slave-контроллере существуют регистры MODE\_CR1 и MODE\_CR2 (*регистры блока управления Slave-контроллера*) расположенные по адресам 0x448 и 0x44C для 1 и 2 портов соответственно.

Взведение 0-го бита LinkDisabled регистра MODE\_CR запрещает установку соединения.

Взведение 24-го бита AUTO\_SPEED регистра MODE\_CR разрешает автоматическую установку соединения (значение после сброса – 1).

При установленном 1-м бите AutoStart в регистре MODE\_CR Slave-контроллер будет находиться в состоянии Ready до получения NULL-кода от второго устройства.

При установленном 2-м бите LinkStart в регистре MODE\_CR Slave-контроллер перейдет в состояние Started и начнет отправлять NULL-коды в канал.

Значения по умолчанию AutoStart = 0, LinkStart = 1.

Для установления соединения необходимо чтобы одно из устройств находилось в состоянии LinkStart, а другое в AutoStart или LinkStart.

### 3.3.2 Настройка скорости передачи

Для настройки скорости передачи в Slave-контроллере существуют два регистра TX\_SPEED1 и TX\_SPEED2 (*регистры блока управления Slave-контроллера*) расположенные по адресам 0x450 и 0x454 для 1 и 2 портов соответственно.

Соединение устанавливается на скорости 10 Мбит/с, далее происходит переход на базовую скорость передачи. Значение коэффициента скорости передачи задается в регистре TX\_SPEED в разрядах [7:0] (поле TX\_SPEED.SEL). Значение коэффициента скорости передачи, необходимое для настройки желаемой скорости, зависит от используемой PLL.

### 3.3.3 Просмотр текущего состояния

Для просмотра текущего состояния 1 и 2 портов в Slave-контроллере существуют два регистра Status\_1 и Status\_2 (*регистры блока управления Slave-контроллера*) расположенные по адресам 0x440 и 0x444 соответственно.

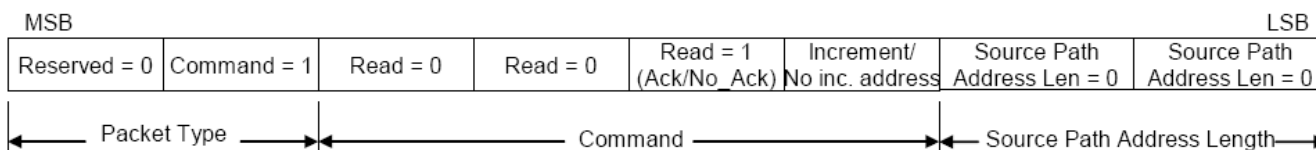
*Пример RMAP команды чтения регистра Status\_1:*

#### Формат пакета RMAP для команды чтения

*First byte transmitted*

Destination Logical Address	Protocol Identifier	Packet Type, Command Source Path Addr Len	Destination Key
Source Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Extended Read Address
Read Address (MS)	Read Address	Read Address	Read Address (LS)
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
EOP	<i>Last byte transmitted</i>		

#### Формат поля Packet Type пакета RMAP для команды чтения



#### Пример пакета RMAP для команды чтения регистра Status\_1

Логический адрес получателя	Идентификатор протокола	Тип команды	Ключ получателя
0x00	0x01	0x48	<b>0xA4</b>
Лог. адрес отправителя	Идентификатор транзакции	Идентификатор транзакции	Расширенный адрес чтения
0x01	0x00	0x00	0x00
Адрес чтения	Адрес чтения	Адрес чтения	Адрес чтения
0x00	0x00	0x04	0x40
Длина пакета данных	Длина пакета данных	Длина пакета данных	CRC заголовка
0x00	0x00	0x04	<b>CRC</b>
<b>EOP</b>			

- Данные RMAP принимаются и передаются во внутренние регистры блока управления, при логическом адресе, указанном в пакете, равном 0 (CONFIGURATION\_ADDR = 0x00);
- Значение поля Protocol ID – идентификатор протокола (должно быть равным 0x01);
- Поле ключ получателя в данном примере имеет случайное значение.

### Формат ответного пакета RMAP для команды чтения

*First byte transmitted*

Source Logical Address	Protocol Identifier	Packet Type, Command, Source Path Addr Len	Status
Destination Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Reserved = 0
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data CRC	EOP	

*Last byte transmitted*

### Формат поля Packet Type ответного пакета RMAP для команды чтения

MSB				LSB			
Reserved = 0	Response = 0	Read = 0	Read = 0	Read = 1 (Ack/No Ack)	Increment/ No inc. address	Source Path Address Length	Source Path Address Length
← Packet Type →		← Command →			← Source Path Address Length →		

### Пример ответного пакета на RMAP пакет чтения регистра Status\_1

Лог. адрес отправителя	Идентификатор протокола	Тип команды	Статус (код ошибки)
0x01	0x01	0x08	0x00
Логический адрес получателя	Идентификатор транзакции	Идентификатор транзакции	-
0x00	0x00	0x00	0x00
Длина пакета данных	Длина пакета данных	Длина пакета данных	CRC заголовка
0x00	0x00	0x04	<b>CRC</b>
Данные	Данные	Данные	Данные
0xA0	0x10	0x00	0x00
CRC данных	<b>EOP</b>		
<b>CRC</b>			

В данном примере полученный регистр статус информирует:

- 1) ошибки отсутствуют;
- 2) номер состояния машины состояний DS-макрочейки – 101 – Run;
- 3) принят первый бит при установке соединения;

### 3.3.4 Использование регистров маршрутизации

Таблица маршрутизации (см. *отдельную спецификацию*) Slave-контроллера состоит из регистра адресов ADDRESS1 и четырех регистров строк маршрутизации STR1\_1, STR1\_2, STR1\_3, STR1\_4, расположенных по адресам 0x000 и 0x004, 0x008, 0x00C, 0x010 соответственно.

ADDRESS1 – 32-разрядный регистр адресов (логически разделен на 4 поля адреса), содержит значения адресов, которым соответствуют строки таблицы маршрутизации. В каждое поле может быть записан логический адрес, с доступными значениями в диапазоне от 32 до 255.

В строке маршрутизации настраиваются номера портов, в которые передается пакет с логическим адресом соответствующим данной строке, а также уровень приоритета, признак адаптивной групповой маршрутизации и признак действительности строки.

Для настройки регистра адресов и регистров строк маршрутизации необходимо отправить RMAP команду записи во внутренний конфигурационный порт Slave-контроллера (см. *отдельную спецификацию*).

### 3.3.5 Пример настройки таблицы маршрутизации с путевой адресацией

*Рассмотрим следующий вариант настройки:*

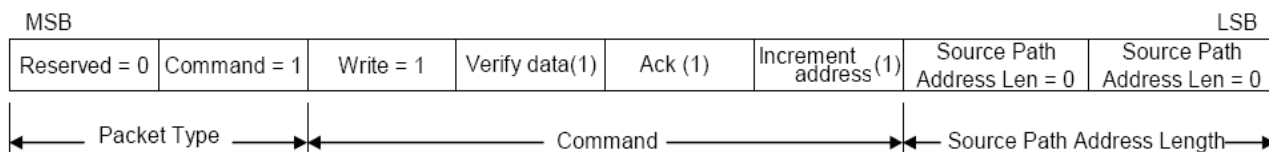
Нужно настроить таблицу маршрутизации таким образом, чтобы приходящие пакеты с полем Destination Logical Adress равным 100 отправлялись в 1 порт Slave-контроллера.

Для этого необходимо в разряды регистра ADDRESS1, соответствующие выбранной строке маршрутизации, записать нужное значение адреса, так же установить значение 1-го разряда строки маршрутизации равным 1 (признак отправки в первый порт).

- 1) Будем использовать первую строку маршрутизации;
- 2) В 0..7 разряды регистра ADDRESS1 RMAP командой запишем желаемое значение адреса 100:
  - Поле *Write Adress* - 0 (адрес регистра ADDRESS1 в адресном пространстве slave-контроллера)
  - Поле *Data* должно иметь значение 0x00000064 (значение настраиваемого DLA = 100)
  - Поле *Data Length* должно иметь значение 4 (размер регистра ADDRESS1 в байтах)
  - Поле *SLA* при значении 1 или 2 будет использовано как *путевой* адрес.

**Пример пакета RMAP для команды записи в регистр ADDRESS1**

Логический адрес получателя	Идентификатор протокола	Тип команды	Ключ получателя
0x00	0x01	0x7D	0xAA
Путевой адрес отправителя	Путевой адрес отправителя	Путевой адрес отправителя	Путевой адрес отправителя
0x00	0x00	0x00	0x01
Лог. адрес отправителя	Идентификатор транзакции	Идентификатор транзакции	Расширенный адрес записи
0x01	0x00	0x00	0x00
Адрес записи	Адрес записи	Адрес записи	Адрес записи
0x00	0x00	0x00	0x00
Длина пакета данных	Длина пакета данных	Длина пакета данных	CRC заголовка
0x00	0x00	0x04	CRC
Данные	Данные	Данные	Данные
0x64	0x00	0x00	0x00
CRC данных	EOP		
CRC			

**Формат поля Packet Type пакета RMAP для команды записи**


3) В регистре первой строки маршрутизации необходимо установить первый разряд равным 1;

- *Write Adress* - 4 (адрес первой строки маршрутизации в адр. пространстве Slave-контроллера)
- В поле *Data* - желаемое значение строки маршрутизации.



### Пример пакета RMAP для команды записи в регистр STR1\_1

Логический адрес получателя	Идентификатор протокола	Тип команды	Ключ получателя
0x00	0x01	0x7D	0xA4
Путевой адрес отправителя	Путевой адрес отправителя	Путевой адрес отправителя	Путевой адрес отправителя
0x00	0x00	0x00	0x01
Лог. адрес отправителя	Идентификатор транзакции	Идентификатор транзакции	Расширенный адрес записи
0x01	0x00	0x00	0x00
Адрес записи	Адрес записи	Адрес записи	Адрес записи
0x00	0x00	0x00	0x04
Длина пакета данных	Длина пакета данных	Длина пакета данных	CRC заголовка
0x00	0x00	0x04	CRC
Данные	Данные	Данные	Данные
0x02	0x00	0x00	0x00
CRC данных	EOP		
CRC			

После отправки двух указанных RMAP команд записи в регистры ADDRESS1 и STR1\_1, пакеты с полем логического адреса получателя равным 100 будут отправляться в первый порт Slave-контроллера.

### 3.3.6 Пример настройки таблицы маршрутизации без путевой адресации

Рассмотрим следующий вариант настройки:

Нужно настроить таблицу маршрутизации таким образом, чтобы входящие пакеты с полем Destination Logical Address отправлялись в конфигурационный порт Slave-контроллера, а ответный пакет возвращался из нужного порта.

Для того, чтобы произошла транзакция на шине АНВ нужно входящий пакет должен поступить в RMAP-контроллер. Для этого необходимо в разряды регистра ADDRESS1, соответствующие выбранной строке маршрутизации, записать нужное значение адреса (Destination Logical Address), так же установить значение 0-го разряда строки маршрутизации равным 1 (признак отправки в конфигурационный порт). Затем значение Destination Logical Address записать в поле rmap\_logical\_addr регистра RMAP\_CONTR2.

Для того, чтобы ответный пакет возвращался из нужного порта необходимо в разряды регистра ADDRESS1, соответствующие выбранной строке маршрутизации, записать значение Source Logical Address, указываемое в входящем пакете. Также установить значение 1-го разряда строки маршрутизации равным 1, если ответный пакет должен возвращаться из порта 1 или установить значение 2-го разряда строки маршрутизации равным 1, если ответный пакет должен возвращаться из порта 2.

1) Будем использовать первую строку маршрутизации для задания адреса, который необходим для передачи входящего пакета в конфигурационный порт (например, 33); вторую строку маршрутизации для записи значения Source Logical Address (например, 77).

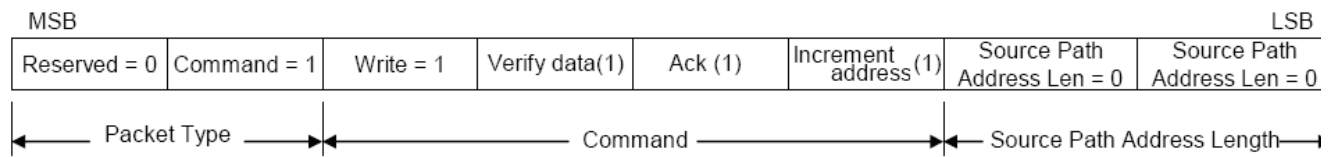
2) В 0...7 разряды регистра ADDRESS1 RMAP командой запишем желаемое значение 33; В 7...15 разряды регистра ADDRESS1 RMAP командой запишем желаемое значение 77;

- Поле *Write Adress* - 0 (адрес регистра ADDRESS1 в адресном пространстве Slave-контроллера)
- Поле *Data* должно иметь значение 0x00004D21

#### Пример пакета RMAP для команды записи в регистр ADDRESS1

Логический адрес получателя	Идентификатор протокола	Тип команды	Ключ получателя
0x00	0x01	0x70	0xAA
Лог. адрес отправителя	Идентификатор транзакции	Идентификатор транзакции	Расширенный адрес записи
0x4D	0x00	0x00	0x00
Адрес записи	Адрес записи	Адрес записи	Адрес записи
0x00	0x00	0x00	0x00
Длина пакета данных	Длина пакета данных	Длина пакета данных	CRC заголовка
0x00	0x00	0x04	CRC
Данные	Данные	Данные	Данные
0x21	0x4D	0x00	0x00
CRC данных	<b>EOP</b>		
CRC			

#### Формат поля Packet Type пакета RMAP для команды записи



3) В регистре первой строки маршрутизации необходимо установить нулевой разряд равным 1;

- *Write Adress* - 4 (адрес первой строки маршрутизации в адр. пространстве Slave-контроллера)
- В поле *Data* - желаемое значение строки маршрутизации.

#### Пример пакета RMAP для команды записи в регистр STR1\_1

Логический адрес получателя	Идентификатор протокола	Тип команды	Ключ получателя
0x00	0x01	0x70	0xAA
Лог. адрес отправителя	Идентификатор транзакции	Идентификатор транзакции	Расширенный адрес записи
0x4D	0x00	0x00	0x00
Адрес записи	Адрес записи	Адрес записи	Адрес записи
0x00	0x00	0x00	0x04
Длина пакета данных	Длина пакета данных	Длина пакета данных	CRC заголовка
0x00	0x00	0x04	CRC
Данные	Данные	Данные	Данные
0x01	0x00	0x00	0x00
CRC данных	<b>EOP</b>		
CRC			

4) В регистре второй строки маршрутизации необходимо установить первый или второй разряд равным 1 в зависимости номера порта из которого необходимо получить ответный пакет.

- *Write Address* - 8 (адрес второй строки маршрутизации в адр. пространстве Slave-контроллера)
- В поле *Data* - желаемое значение строки маршрутизации.

#### Пример пакета RMAP для команды записи в регистр STR1\_2

Логический адрес получателя	Идентификатор протокола	Тип команды	Ключ получателя
0x00	0x01	0x70	0xAA
Лог. адрес отправителя	Идентификатор транзакции	Идентификатор транзакции	Расширенный адрес записи
0x4D	0x00	0x00	0x00
Адрес записи	Адрес записи	Адрес записи	Адрес записи
0x00	0x00	0x00	0x08
Длина пакета данных	Длина пакета данных	Длина пакета данных	CRC заголовка
0x00	0x00	0x04	CRC
Данные	Данные	Данные	Данные
0x02 или 0x04	0x00	0x00	0x00
CRC данных	<b>EOP</b>		
CRC			

Чтобы использовать RMAP запишем в поле *rmap\_logical\_addr* регистра RMAP\_CONTR2 значение 33, *rmap\_dest\_key* зададим равным 170.

- *Write Address* - 4B0 (адрес регистра RMAP\_CONTR2)
- В поле *Data* - 0x0000AA21.

#### Пример пакета RMAP для команды записи в регистр RMAP\_CONTR2

Логический адрес получателя	Идентификатор протокола	Тип команды	Ключ получателя
0x00	0x01	0x70	0xAA
Лог. адрес отправителя	Идентификатор транзакции	Идентификатор транзакции	Расширенный адрес записи
0x4D	0x00	0x00	0x00
Адрес записи	Адрес записи	Адрес записи	Адрес записи
0x00	0x00	0x04	0xB0
Длина пакета данных	Длина пакета данных	Длина пакета данных	CRC заголовка
0x00	0x00	0x04	CRC
Данные	Данные	Данные	Данные
0x21	0xAA	0x00	0x00
CRC данных	<b>EOP</b>		
CRC			

Теперь мы имеем возможность обмениваться данными с устройством, подключенным к Slave-контроллеру через АНВ-интерфейс(RMAP-протокол).

В RMAP-пакете для этого необходимо установить:

Destination Logical Address = 33;

Source Logical Address = 77;

Destination Key = 170.

Значение адреса устанавливается равным адресу регистра RMR-02P умноженному на 4. Например, регистр DEVID имеет адрес 0x1, в пакете устанавливается значение 0x4.

### 3.4 ПОРТ ПЕРИФЕРИЙНЫХ УСТРОЙСТВ.

Микросхема содержит два универсальных порта для управления периферийными устройствами. Каждый порт может быть запрограммирован в одном из следующих режимов:

- GPIO/MSPI: 16 двунаправленных сигналов общего назначения и порт SPI-мастер.
- PP: 16-разрядный двунаправленный параллельный порт.
- MBOX: 16-разрядный двунаправленный почтовый ящик.

#### 3.4.1 Режим GPIO/MSPI.

В режиме GPIO/MSPI порт управления периферийными устройствами используется как 16 двунаправленных сигналов ввода-вывода общего назначения, совмещенный с портом SPI-мастер.

Каждый сигнал ввода-вывода общего назначения может быть индивидуально установлен как вход или выход. Кроме того, каждый сигнал ввода-вывода может быть использован как сигнал выбора при выполнении операции по порту MSPI (таким образом может быть использовано до 16 сигналов выбора). Сигналы ввода-вывода содержат логику, которая формирует сигнал прерывания при изменении состояния сигналов ввода-вывода, установленных как входы.

Порт SPI-мастер позволяет аппаратно выполнять обмен по последовательной шине длиной до 32 бит. При этом, как отмечено выше, в качестве сигналов выбора могут быть использованы сигналы ввода-вывода общего назначения. Сигналы ввода-вывода общего назначения, используемые в качестве сигналов выбора порта MSPI, должны быть сконфигурированы как выходы и установлены в 1. После завершения процессов обмена формируется соответствующее прерывание.

Порт может быть использован для формирования более длинных посылок. В этом случае сигналы выборки должны формироваться “программно”.

Временные диаграммы работы порта в режиме SPI для нулевой, первой, второй и третьей модификаций представлены на рисунках 3.6 – 3.9 соответственно.

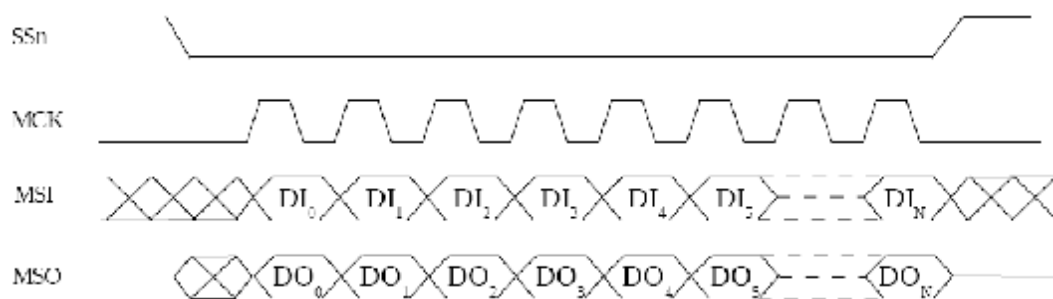


Рисунок 3.6

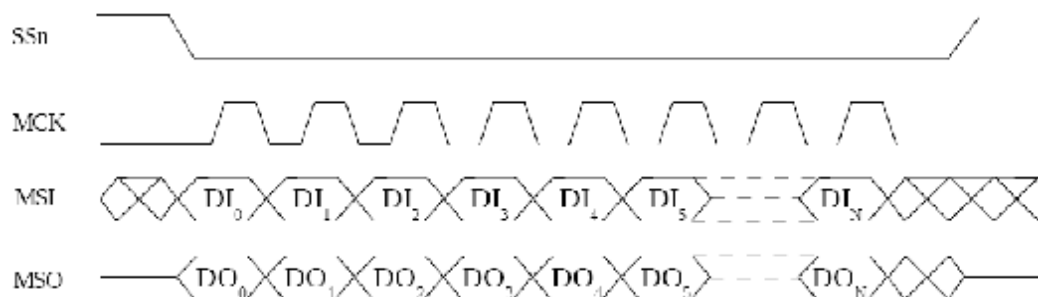


Рисунок 3.7

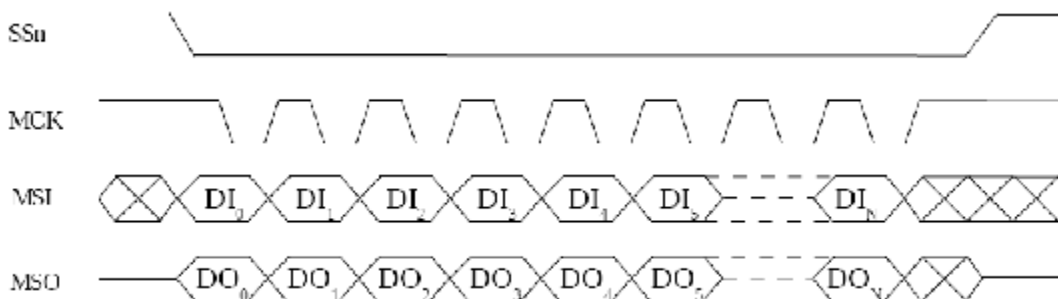


Рисунок 3.8

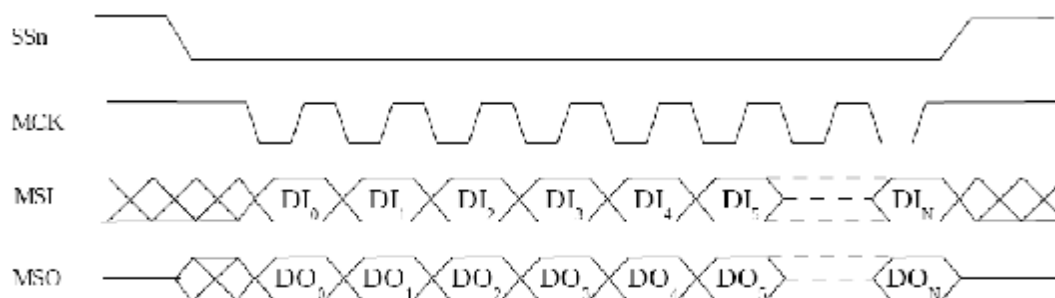


Рисунок 3.9

В этом режиме также может использоваться сигнал запроса прерывания от периферийных устройств.

### 3.4.2 Режим PP.

В этом режиме периферийный порт используется в качестве двунаправленного 16-разрядного порта с отдельными stroбами чтения и записи и сигналом готовности периферийных устройств. Порт может выполнять операции чтения и записи периферийных устройств.

Длительность stroбов чтения/записи и паузы между ними программируется. Временная диаграмма работы режима параллельного порта для этого случая показаны на рисунке 3.10.

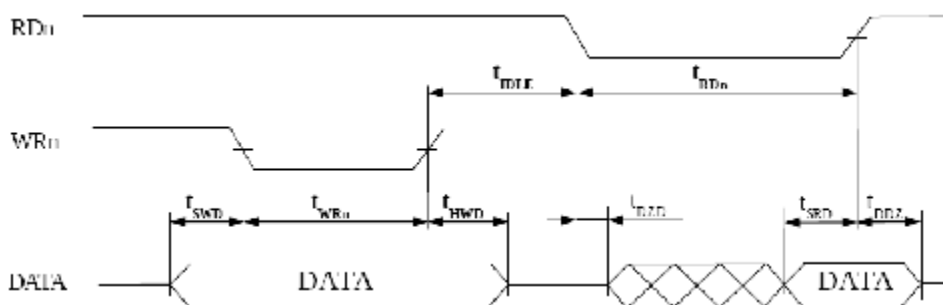


Рисунок 3.10

Параллельный порт является асинхронным интерфейсом. На диаграмме, представленной на рисунке 3.10, показаны основные временные соотношения. Часть из этих параметров обеспечивается непосредственно логической структурой микросхемы. Их значения представлены в таблице 3.8.

Таблица 3.8

Параметр	Обозначение	Значение
Длительность stroba записи	$t_{WR}$	$(1-16) \cdot t_{HCLK}$
Длительность stroba чтения	$t_{RDn}$	$(1-16) \cdot t_{HCLK}$
Длительность паузы между обращениями	$t_{IDLE}$	$(1-16) \cdot t_{HCLK}$
Время установки данных относительно сигнала WRn	$t_{SAD}$	$t_{HCLK}$
Время удерживания данных относительно сигнала WRn	$t_{HAD}$	$t_{HCLK}$
Примечания		
1 $t_{WRn}$ , $t_{RDn}$ , $t_{IDLE}$ – запрограммированная длительность stroбов записи, чтения и паузы между ними		
2 $t_{HCLK}$ – период частоты внутри кристалльной шины АНВ		

Параллельный порт может обмениваться данными с периферийными устройствами, которые имеют сигнал готовности. Сигнал готовности в этом случае используется для идентификации завершения выполнения операции. Временная диаграмма работы параллельного порта в этом случае представлена на рисунке 3.11.

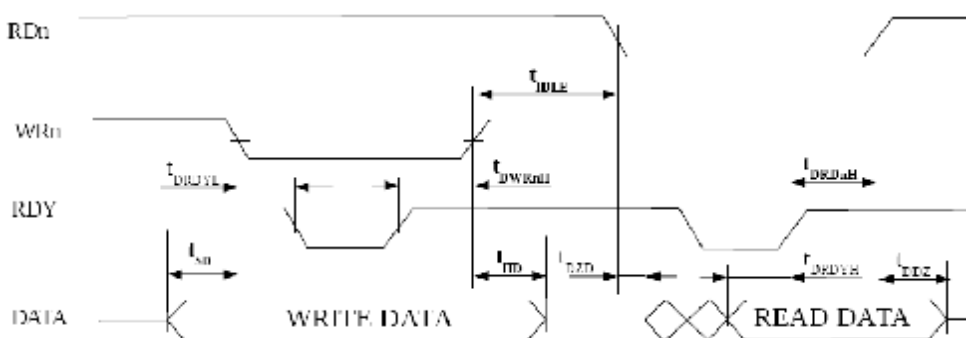


Рисунок 3.11

Для того, чтобы предотвратить зависание системы, предусмотрен механизм принудительного завершения операции. Он срабатывает в том случае, если периферийное устройство удерживает сигнал RDY в нулевом состоянии дольше максимально допустимого времени. После истечения этого времени с момента снятия сигнала RDY, соответствующая операция будет принудительно завершена, и в сеть SpaceWire будет отправлено прерывание. Максимальное время ожидания завершения операции также программируется.

С помощью параллельного порта можно осуществлять не только одиночные, но и пакетные обращения к периферийным устройствам. В режиме параллельного порта может использоваться сигнал запроса прерывания от периферийных устройств.

Обмен данными с периферийными устройствами в режиме параллельного порта осуществляется через регистр GPIO\_DAT.

### 3.4.3 Режим MBOX.

В этом режиме порт используется в качестве двунаправленного 16-разрядного почтового ящика со стробами чтения и записи от периферийного устройства и сигналами готовности почтового ящика по записи и чтению. В данном режиме периферийное устройство может писать в почтовый ящик или читать из него.

На рисунке 3.12 показана временная диаграмма работы почтового ящика.

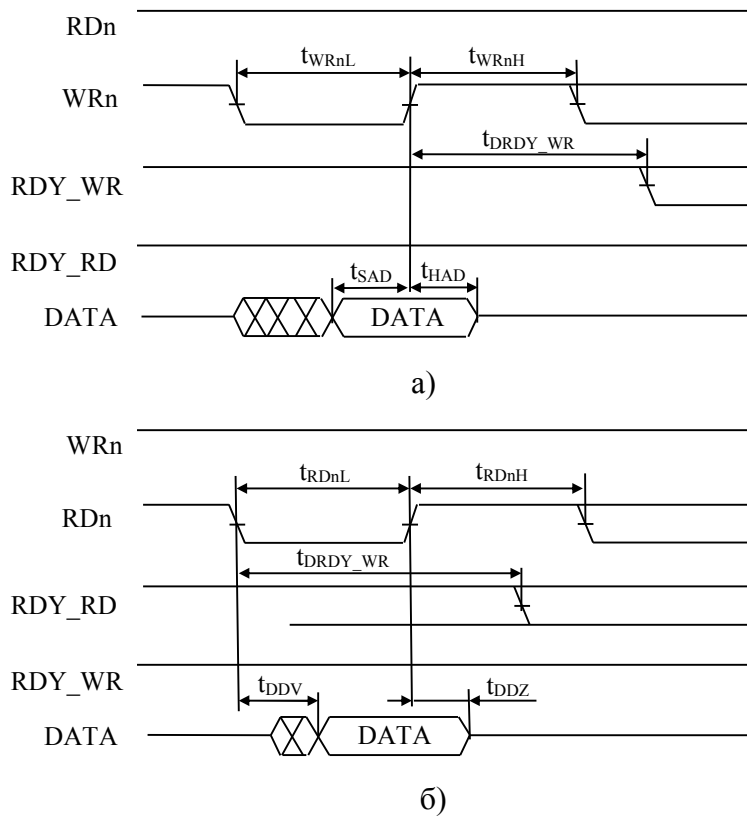


Рисунок 3.12 а) – запись данных в ящик; б) – чтение данных из ящика

Временные параметры тактовой диаграммы представлены в таблице 3.9

Таблица 3.9

Параметр	Обозначение	Значение, нс	
		не менее	не более
1	2	3	4
Длительность низкого уровня stroba записи	$t_{WRnL}$	40	–
Длительность высокого уровня stroba записи	$t_{WRnH}$	40	–
Длительность низкого уровня stroba чтения	$t_{RDnL}$	40	–
Длительность высокого уровня stroba чтения	$t_{RDnH}$	40	–
Время установки данных при записи относительно переднего фронта сигнала WRn	$t_{SAD}$	10	–
Время удерживания данных при записи относительно переднего фронта сигнала WRn	$t_{HAD}$	10	–
Задержка появления данных относительно заднего фронта сигнала RDn	$t_{DDV}$	–	10
Время удерживания данных относительно переднего фронта сигнала RDn	$t_{DDZ}$	–	10
Задержка снятия сигнала готовности почтового ящика по записи относительно переднего фронта сигнала WRn	$t_{DRDY\_WR}$	–	65
Задержка снятия сигнала готовности почтового ящика по чтению относительно заднего фронта сигнала RDn	$t_{DRDY\_RD}$	–	65

Порт в своем составе имеет два буфера размером 128 16-разрядных слов. Один буфер предназначен для передачи данных периферийному устройству, другой для приема данных от периферийного устройства.

При записи почтового ящика периферийным устройством информация помещается в буфер приема. Из буфера приема эта информация может считываться контроллером SSpW или контроллера SPI-порта через регистр GPIO\_DAT.

Буфер приема имеет порог наполненности. Если во время процесса записи периферийным устройством количество слов в почтовом ящике будет больше или равно порогу наполненности, в сеть SpaceWire будет отправлено прерывание. Значение порога наполненности буфера приема устанавливается с помощью регистра FIFO\_TH.

По мере заполнения буфера приема данными, почтовый ящик может перейти в состояние неготовности для записи. Сигнал готовности по записи RDY\_WR, показывающий периферийному устройству, можно ли в него записать новые данные, будет снят. Снятый сигнал RDY\_WR дает пессимистическую оценку наполненности буфера приема почтового ящика. Физически в нем могут быть пустые места. Уровень пессимизма задается с помощью регистра FIFO\_TH.

Если периферийное устройство попытается записать данные в полный буфер приема ящика, то записываемые данные будут проигнорированы, а в сеть SpaceWire будет отправлено прерывание.

При чтении почтового ящика со стороны периферийного устройства информация выдается из буфера передачи. Этот буфер может записываться со стороны контроллера



SSpW или SPI через регистр GPIO\_DAT. Данные, записанные в регистр GPIO\_DAT, помещаются в буфер передачи. Когда в буфере передачи окажется хотя бы одно слово, почтовый ящик перейдет в режим готовности по чтению - будет установлен сигнал готовности по чтению. Буфер передачи имеет порог наполненности, который задается с помощью регистра FIFO\_TH. Если в процессе работы количество слов в буфере передачи окажется меньше порога наполненности, в сеть SpaceWire будет отправлено соответствующее прерывание.

Если буфер передачи окажется пустым, почтовый ящик перейдет в состояние неготовности для чтения, и сигнал готовности по чтению будет снят. Если периферийное устройство попытается прочитать данные из пустого почтового ящика, в сеть SpaceWire будет отправлено прерывание.

В регистре FIFO\_STAT содержится информация о количестве слов, находящихся в буфере приема, и о количестве свободных слов в буфере передачи. Регистр доступен только для чтения.

В случае необходимости имеется возможность сбросить буферы приема и передачи почтового ящика. Осуществляется это с помощью регистра SWRST.

### 3.5 МИКРОПРОЦЕССОРНЫЙ ИНТЕРФЕЙС $\mu P$

В микросхеме предусмотрен 16-разрядный двунаправленный микропроцессорный интерфейс для управления периферийными устройствами. Данный интерфейс позволяет эмулировать следующие микропроцессорные шины:

- Intel (раздельные стробы записи-чтения и сигнал готовности);
- Motorola (общий строб записи-чтения и сигнал подтверждения);
- Multicore (раздельные стробы записи-чтения и сигнал подтверждения).

Микропроцессорный интерфейс имеет два адресных пространства размером 64К слов и сигналы управления, связанные с этими пространствами. Для каждого адресного пространства может быть независимо установлена минимальная длительность стробов чтения-записи и паузы между ними. Временные диаграммы работы интерфейса показаны на рисунках 3.14 – 3.19.

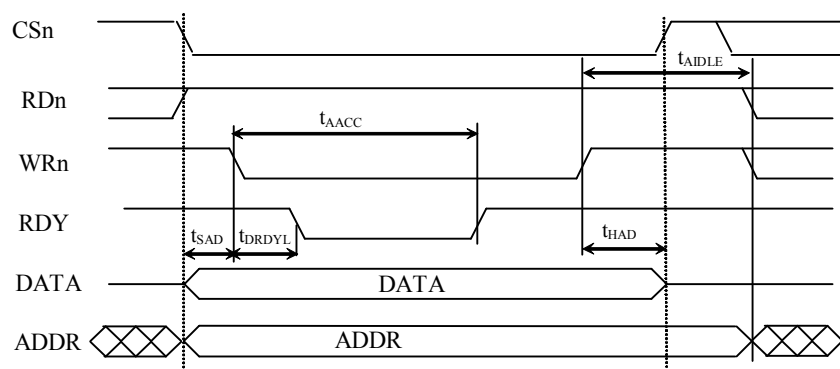


Рисунок 3.14 Обмен по микропроцессорному интерфейсу: запись в режиме Intel.

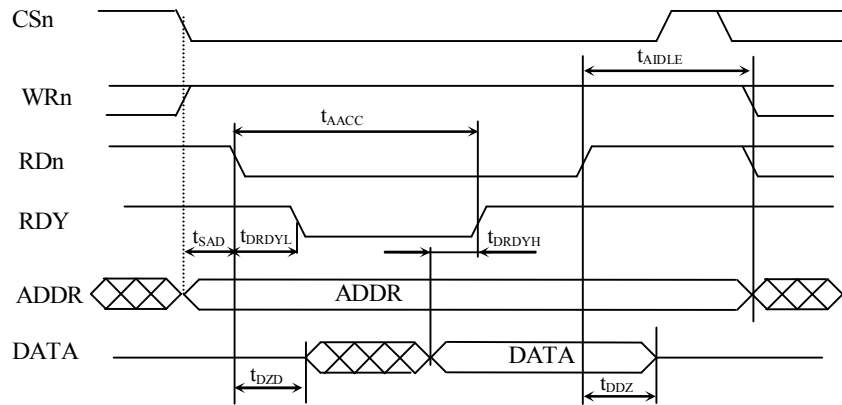


Рисунок 3.15 Обмен по микропроцессорному интерфейсу: чтение в режиме Intel.

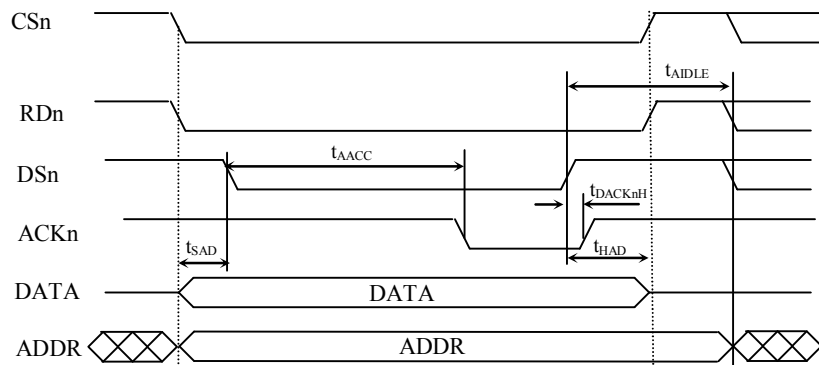


Рисунок 3.16 Обмен по микропроцессорному интерфейсу: запись в режиме Motorola.

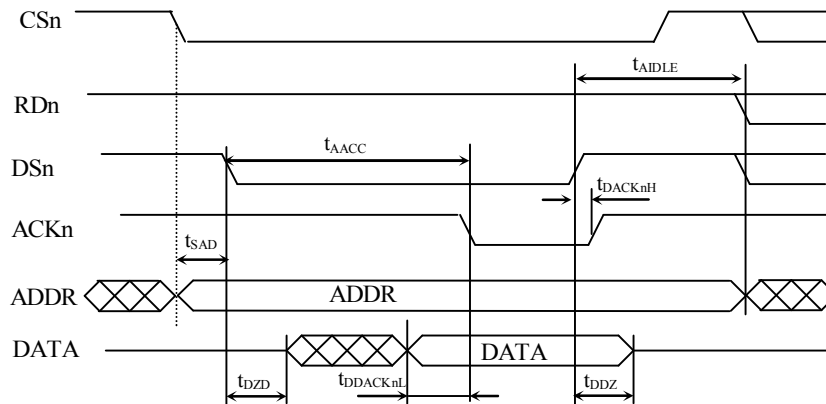


Рисунок 3.17 Обмен по микропроцессорному интерфейсу: чтение в режиме Motorola.

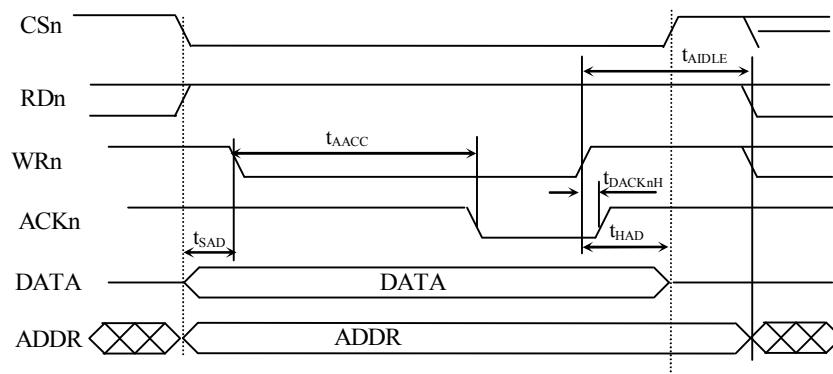


Рисунок 3.18 Обмен по микропроцессорному интерфейсу: запись в режиме Multicore.

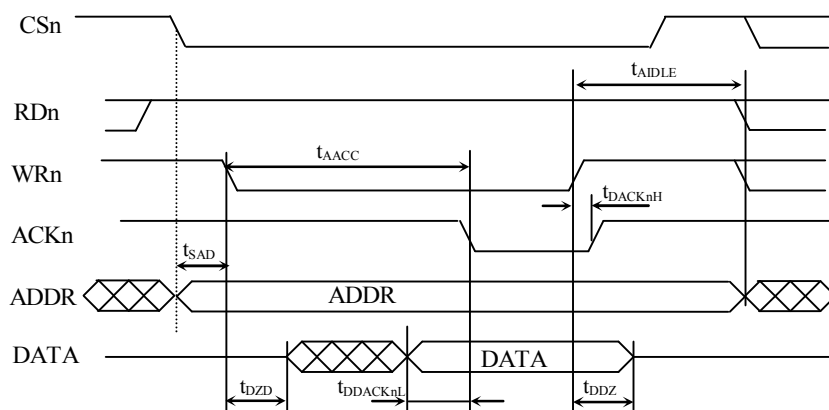


Рисунок 3.19 Обмен по микропроцессорному интерфейсу: чтение в режиме Multicore.

Сигналы микропроцессорного интерфейса мультиплексированы с сигналами нулевого и первого портов управления периферийными устройствам. Поэтому в режиме  $\mu P$  порты GPIO1/MSP11/PP1/MBOX1 и GPIO2/MSP12/PP2/MBOX2 не могут быть использованы.

В режиме  $\mu P$  периферийные устройства может использовать два сигнала запроса прерывания ICTRB0 и ICTRB1.

## 4 РЕГИСТРЫ УПРАВЛЕНИЯ

### 4.1 ВНУТРЕННЕЕ АДРЕСНОЕ ПРОСТРАНСТВО

Внутренне адресное пространство содержит управляющие и статусные 16-разрядные регистры.

Таблица 4.1 Адресное пространство регистров управления

Адрес.	Сброс.	Тип.	Имя.	Назначение.
00000	0000	wo	SWRST	Регистр программного сброса
00001	0501	ro	DEVID	Идентификатор устройства, только чтение
00002	0000	rw	INT	Флаги прерываний
00003	0000	rw	INTE	Маска флагов прерываний
00004	0000	rw	INTR	Регистр сброса прерываний
00005	0000	rw	CFG	Регистр конфигурации
00006	0000	rw	CLKO	Регистр конфигурации выходной тактовой частоты
00007	0000	rw	RSTO	Регистр конфигурации выходного сигнала инициализации
00008	0000	rw	GPIO0_DIR	Регистр направления сигналов GPIO0
00009	0000	rw	GPIO0_DAT	Регистр данных GPIO0/PP0/MBOX0/USART0
0000A	0000	rw	PORT0_CFG1	Регистр конфигурации 1 порта 0
0000B	0000	rw	PORT0_CFG2	Регистр конфигурации 2 порта 0
0000C	0000	rw	PORT0_CFG3	Регистр конфигурации 3 порта 0
0000D	0000	rw	FIFO0_TH	Регистр настройки буферов приемников/передатчиков порта 0
0000E	0000	ro	FIFO0_STAT	Регистр статуса буферов приемников/передатчиков порта 0
0000F	0000	rw	MSPI0_DAT	Регистр данных порта MSPI0. Младшие биты.
00010	0000	rw	GPIO1_DIR	Регистр направления сигналов GPIO1
00011	0000	rw	GPIO1_DAT	Регистр данных GPIO0/PP0/MBOX1/USART1
00012	0000	rw	PORT1_CFG1	Регистр конфигурации 1 порта 1
00013	0000	rw	PORT1_CFG2	Регистр конфигурации 2 порта 1
00014	0000	rw	PORT0_CFG3	Регистр конфигурации 3 порта 1
00015	0000	rw	FIFO0_TH	Регистр настройки буферов приемников/передатчиков порта 0
00016	0000	ro	FIFO0_STAT	Регистр статуса буферов приемников/передатчиков порта 0
00017	0000	rw	MSPI1_DAT	Регистр данных порта MSPI1. Младшие биты.
10000.. 1FFFF	xxxx		μP1	Адресное пространство 1 μP
20000.. 2FFFF	xxxx		μP2	Адресное пространство 2 μP

Указанные адреса регистров соответствуют номеру регистра в командах последовательного интерфейса. При использовании контроллера SpaceWire протокола RMAP должны применяться адреса регистров на шине АНВ, которые получаются из указанных адресов путем добавления двух нулевых битов со стороны младших разрядов (т.е. адрес на АНВ в 4 раза больше адреса, указанного в данной таблице). Зарезервированные поля и регистры читаются нулями. Запись в них игнорируется.

### 4.1.1 SWRST

Регистр программного сброса. При записи в регистр определенных числе возможны следующие варианты сброса:

0x0078 - программный сброс всей микросхемы, полностью аналогичный аппаратному сбросу.

0x0178 - сброс буфера приемника порта 0.

0x0278 - сброс буфера передатчика порта 0.

0x0378 - сброс буфера приемника порта 1.

0x0478 - сброс буфера передатчика порта 0.

При чтении возвращается ‘0’.

### 4.1.2 DEVID

Регистр 16-разрядного идентификатора типа устройства. Доступен только по чтению. Значение идентификатора – 0x501.

### 4.1.3 INT

Регистр INT содержит флаги ошибок и важных событий. Наличие установленного флага при соответствующем разрешении в регистре INTE приводит к установке сигнала IRQ. Регистр доступен для записи, что может быть использовано для тестирования ПО. Запись ‘1’ вызывает установку флага. Запись ‘0’ игнорируется. Для сброса прерываний необходимо использовать регистр INTR.

Бит	Имя поля	Назначение
15:14	res	зарезервировано
13	irq0	запрос нулевого прерывания
12	irq1	запрос первого прерывания
11	mb1_tx_ovr	в режиме MBOX: записаны данные в заполненный почтовый ящик первого периферийного порта со стороны управляющего устройства
10	mb1_tx_und	в режиме MBOX: из пустого почтового ящика первого периферийного порта прочитаны данные со стороны периферийного устройства
9	spi1_tx_rd mb1_tx_rd	в режиме MBOX: прочитаны данные из почтового ящика первого периферийного порта со стороны периферийного устройства
8	spi1_rx_ovr mb1_rx_ovr	в режиме MBOX: записаны данные в заполненный почтовый ящик первого периферийного порта со стороны периферийного устройства
7	spi1_rx_wr mb1_rx_und	в режиме MBOX: из пустого почтового ящика первого периферийного порта прочитаны данные со стороны управляющего устройства
6	gpio1 mb1_rx_wr pp1_tout	в режиме GPIO/SPI: состояние входов GPIO первого периферийного порта изменилось в режиме MBOX: записаны данные в почтовый ящик первого периферийного порта со стороны периферийного устройства в режиме PP: превышение максимальной длительности операции чтения/записи через параллельный порт первого периферийного порта
5	mb0_tx_ovr	в режиме GPIO/SPI: буфер приема нулевого SPI-порта переполнен в режиме MBOX: записаны данные в заполненный почтовый ящик нулевого периферийного порта со стороны управляющего устройства
4	mb0_tx_und	в режиме GPIO/SPI: записаны данные в буфер приема первого SPI-порта в режиме MBOX: из пустого почтового ящика нулевого периферийного порта прочитаны данные со стороны периферийного устройства
3	spi0_tx_rd mb0_tx_rd	в режиме GPIO/SPI: прочитаны данные из буфера передачи нулевого SPI-порта в режиме MBOX: прочитаны данные из почтового ящика нулевого периферийного порта со стороны периферийного устройства
2	spi0_rx_ovr mb0_rx_ovr	в режиме GPIO/SPI: буфер приема нулевого SPI-порта переполнен в режиме MBOX: записаны данные в заполненный почтовый ящик нулевого периферийного порта со стороны периферийного устройства

Бит	Имя поля	Назначение
1	spi0_rx_wr mb0_rx_wr	в режиме GPIO/SPI: записаны данные в буфер приема нулевого SPI-порта в режиме MBOX: из пустого почтового ящика нулевого периферийного порта прочитаны данные со стороны управляющего устройства
0	gpio0 mb0_rx_und pp0_tout	в режиме GPIO/SPI: состояние входов GPIO нулевого периферийного порта изменилось в режиме MBOX: записаны данные в почтовый ящик нулевого периферийного порта со стороны периферийного устройства в режиме PP: превышение максимальной длительности операции чтения/записи через параллельный порт нулевого периферийного порта

Сигнал прерывания IRQ формируется в соответствии со следующим уравнением:  
 $IRQ = CFG.inte \& | (INT \& INTE)$

#### 4.1.4 INTE

Регистр содержит маски флагов ошибок и важных событий в полном соответствии с регистром INT. Установленный бит INTE разрешает формирование сигнала IRQ по соответствующему флагу в регистре INT.

#### 4.1.5 INTR

Регистр INTR предназначен для сброса флагов прерываний, установленных в регистре INT: запись единицы в какой-либо бит INTR сбрасывает соответствующий бит регистра INT, если он был установлен. Запись нуля не вызывает никаких действий. При чтении возвращается ноль.

#### 4.1.6 CFG

Бит	Имя поля	Назначение
15	inte	Разрешение прерывания
14	intp	Полярность прерывания
13	res	зарезервировано
12:8	spw_int_addr	Адрес источника прерываний в сети SpW
7:6	irq1_mode	Режим сигнала прерывания IRQn1: 0= по заднему фронту 1= по переднему фронту 2= по нижнему уровню 3= по высокому уровню
5:4	irq0_mode	Режим сигнала прерывания IRQn0: 0= по заднему фронту 1= по переднему фронту 2= по нижнему уровню 3= по высокому уровню
3	res	зарезервировано
2:1	µp_mode	Режим работы µP: 0= Intel 1= Motorola 2= Multicore
0	µp_en	1= разрешение работы µP (GPIO/MSPI/PP/MBOX1 и GPIO/MSPI/PP/MBOX2 не должны использоваться)

#### 4.1.7 CLKO

Бит	Имя поля	Назначение
15:8	res	Зарезервировано
7	clko_en	0=Сигнал CLKO не формируется 1=Сигнал CLKO формируется
6	clko_pol	Поолярность сигнала CLKO

Бит	Имя поля	Назначение
5:0	clk_div	Коэффициент деления CLK для формирования сигнала CLKO 0 - 128 1...63 - (2·clk_div)

#### 4.1.8 RSTO

Бит	Имя поля	Назначение
15:8	Res	Зарезервировано
7	Rst	1=формируется сигнал RSTOn 0=не формируется сигнал RSTOn
6	rst_pol	Полярность сигнала RSTOn
5:0	rst_len	Длительность сигнала RSTO в тактах сигнала HCLK 0 - 128 1...63 - (2·rst_len)

#### 4.1.9 GPIO0\_DIR, GPIO1\_DIR

Бит	Имя поля	Назначение
15:0	Dir	Определяет состояние соответствующего вывода GPIO 1= выход 0= вход В режимах PP и MBOX не используется.

#### 4.1.10 GPIO0\_DAT, GPIO1\_DAT

Бит	Имя поля	Назначение
15:0	Dat	Регистр состояния внешних сигналов GPIO[15:0]. Если вывод GPIO сконфигурирован как выход, то при записи он устанавливается в записываемое состояние. При чтении возвращается состояние соответствующих выводов GPIO[15:0]. Регистр данных в режимах PP и MBOX.

При использовании сигналов GPIO в качестве сигнала выбора канала MSPI, направление соответствующего сигнала GPIO должно быть установлено как “выход”, а его состояние в “1”.

#### 4.1.11 PORT0\_CFG1, PORT1\_CFG1

Бит	Имя поля	Назначение
15:13	mode	Режим работы: 0xx= выключен 100= GPIO/SPI 101= PP 110= MBOX 111= ADV_PP
11:6	mck_div	Коэффициент деления HCLK для формирования сигнала MCK $f_{MCK} = f_{HCLK} / (2 * (mck\_div + 1))$ 1= /4 2= /6 7= /256
5:4	spi_mode	Режим работы SPI порта
3:2	ssn_sck_setup	Время установки сигналаа SSn перед первым перепадом SCK: 0: $T_{hclk}$ 1: $2 T_{hclk}$ 2: $3 T_{hclk}$ 3: $4 T_{hclk}$
1:0	ssn_sck_hold	Время удерживания сигналаа SSn после последнего перепада SCK: 0: $T_{hclk}$ 1: $2 T_{hclk}$ 2: $3 T_{hclk}$ 3: $4 T_{hclk}$

#### 4.1.12 PORT0\_CFG2, PORT1\_CFG2

Бит	Имя поля	Назначение
15:12	mcs	Номер канала SPI (определяет номер бита сигнала GPIO, используемого в качестве сигнала выбора канала SPI)
11:5	pause	Пауза между двумя посылками по SPI-порту: 0: 255 T <sub>hclk</sub>
4:0	len	Длина передаваемых данных по SPI: 0= 1 бит ... 31= 32 бит

#### 4.1.13 PORT0\_CFG3, PORT1\_CFG3

Бит	Имя поля	Назначение
15:12	stb_len	PP: минимальная длительность стробов записи/чтения в тактах сигнала HCLK: 0= 16 тактов В режимах работы GPIO/SPI, MBOX не используется
11:8	gap_len	PP: минимальная длительность паузы между стробами записи/чтения в тактах сигнала HCLK; 0= 16 тактов В режимах работы GPIO/SPI, MBOX не используется
7:0	timeout	Время ожидания завершения операции периферийным устройством в тактах сигнала HCLK .

Регистры PORT1\_CFG3 и PORT2\_CFG3 в режиме  $\mu$ P используются как регистры управления длительностями стробов и временем ожидания для адресных пространств 1 и 2.

#### 4.1.14 FIFO0\_TH, FIFO1\_TH

Бит	Имя поля	Назначение
15:9	tx_fifo_th	Количество слов в буфере передачи, начиная с которого формируется прерывание, сообщающее о том, что в буфере осталось меньше слов чем задано этим полем либо равно ему
8:2	rx_fifo_th	Количество слов в буфере приема, превышая которое формируется прерывание, сообщающее о том, что в буфере имеется больше слов чем задано этим полем либо равно ему
1:0	rx_afull_th	Порог снятия сигнала готовности почтового ящика по записи. Если в почтовом ящике окажется меньше или равно свободных слов чем задано в данном поле, то почтовый ящик перейдет в состояние не готовности по записи. Сигнал готовности почтового ящика по записи будет снят.

#### 4.1.15 FIFO0\_STAT, FIFO1\_STAT

Бит	Имя поля	Назначение
15:8	tx_word_cnt	Количество свободных слов в буфере передачи
7:0	rx_word_cnt	Количество принятых слов в буфере приема

#### 4.1.16 MSPI0\_DAT, MSPI1\_DAT

Бит	Имя поля	Назначение
31:0	dat	Регистр данных порта SPI-master. В режимах PP и MBOX не используется.



## 4.2 РЕГИСТРЫ КОНТРОЛЛЕРА SPACEWIRE

Список программно-доступных регистров контроллера SpaceWire представлен в таблице 4.2.

Таблица 4.2. Перечень программно-доступных регистров контроллера SpaceWire

Условное обозначение	Описание	Тип доступа	Исходное состояние	Адрес
<u>Регистры маршрутизации</u>				
ADDRESS1	Регистр адресов	RW	0	000
STR1_1	Регистр строки маршрутизации 1	RW	0	004
STR1_2	Регистр строки маршрутизации 2	RW	0	008
STR1_3	Регистр строки маршрутизации 3	RW	0	00C
STR1_4	Регистр строки маршрутизации 4	RW	0	010
<u>Регистры управления (базовая часть)</u>				
ID_VER	Регистр версии SSpW	RO	0	400
<u>Регистры портов SpaceWire</u>				
Status_1	Регистр статуса первого порта SpaceWire	WR/RD	0	440
Status_2	Регистр статуса второго порта SpaceWire	WR/RD	0	444
MODE_CR1	Регистр режима работы первого порта SpaceWire	WR/RD	–	448
MODE_CR2	Регистр режима работы второго порта SpaceWire	WR/RD	–	44C
TX_SPEED1	Регистр коэффициента скорости передачи первого порта SpaceWire	WR/RD	0	450
TX_SPEED2	Регистр коэффициента скорости передачи второго порта SpaceWire	WR/RD	0	454
RX_SPEED1	Регистр коэффициента скорости приема первого порта SpaceWire	WR/RD	0	458
RX_SPEED2	Регистр коэффициента скорости приема второго порта SpaceWire	WR/RD	0	45C
RMAP_CONTR2	Регистр режима функционирования 2 RMAP контроллера	WR/RD	254	4B0

### 4.2.1 Регистр адресов ADDRESS1

Данный регистр содержит значения адресов, которым соответствуют строки таблицы маршрутизации, расположенные в регистрах STR1\_1 - STR1\_4. Он реализован с доступом по чтению и по записи. Формат регистра ADDRESS1 представлен в таблице 4.3.

Таблица 4.3. Формат регистра ADDRESS1

Бит	Имя поля	Назначение
31:24	Addr4	Адрес, соответствующий строке 4
23:16	Addr3	Адрес, соответствующий строке 3
15:8	Addr2	Адрес, соответствующий строке 2
7:0	Addr1	Адрес, соответствующий строке 1

После выхода контроллера из состояния сброса значение этого регистра –“0”.

#### 4.2.2 Регистр строки маршрутизации 1 STR1\_1

Данный регистр содержит значение строки маршрутизации, соответствующее адресу Addr1 из регистра ADDRESS1. Он реализован с доступом по чтению и по записи. Формат регистра STR1\_1 представлен в таблице 4.4.

**Таблица 4.4. Формат регистра STR1\_1**

Бит	Имя поля	Назначение
31..0	str	Значение строки маршрутизации

После выхода контроллера из состояния сброса значение этого регистра – “0”.  
Формат строки таблицы маршрутизации представлен в таблице 4.5.

**Таблица 4.5. Формат строки таблицы маршрутизации**

Бит	Назначение
31	Признак действительности строки. Если он установлен в значение “0”, то строка считается действительной. Если он установлен в значение “1”, то строка считается недействительной, все пакеты, приходящие на данный адрес, изымаются из сети и уничтожаются
30	Признак адаптивной групповой маршрутизации. Если этот разряд установлен в “0”, то адаптивная групповая маршрутизация выполняется по регистрам адаптивной групповой маршрутизации. При этом, если в первом и во втором разрядах стоит “1”, то пакет широковещательно отправляется в оба порта. Если этот разряд установлен в “1”, то если в первом и во втором разрядах стоит “1”, то пакет отправляется в любой из портов
29	Зарезервирован для служебного применения. При программной записи в этот разряд должен записываться “0”, при чтении из него читается “0”
28:19	Не используется
18	Признак отделения заголовка. Если этот разряд установлен в “1”, то происходит отделение первого байта заголовка
17	Уровень приоритета (“1” - высокий приоритет, “0”, – низкий)
16:3	В данной реализации не используется
2	Если в этом разряде установлена “1”, то это указывает, что пакет должен быть передан во второй порт
1	Если в этом разряде установлена “1”, то это указывает, что пакет должен быть передан в первый порт
0	Если в этом разряде установлена “1”, то это указывает, что пакет должен быть передан в конфигурационный порт SSpW (в RMAP-контроллер)

#### 4.2.3 Регистр строки маршрутизации 1 STR1\_2

Данный регистр содержит значение строки маршрутизации, соответствующее адресу Addr2 из регистра ADDRESS1. Он реализован с доступом по чтению и по записи. Формат регистра STR1\_2 представлен в таблице 4.6.

**Таблица 4.7. Формат регистра STR1\_2**

Бит	Имя поля	Назначение
31..0	str	Значение строки маршрутизации

После выхода контроллера из состояния сброса значение этого регистра – “0”.

#### 4.2.4 Регистр строки маршрутизации 1 STR1\_3

Данный регистр содержит значение строки маршрутизации, соответствующее адресу Addr3 из регистра ADDRESS1. Он реализован с доступом по чтению и по записи. Формат регистра STR1\_3 представлен в таблице 4.7.

Таблица 4.7. Формат регистра STR1\_3

Бит	Имя поля	Назначение
31..0	str	Значение строки маршрутизации

После выхода контроллера из состояния сброса значение этого регистра – “0”.

#### 4.2.5 Регистр строки маршрутизации 1 STR1\_4

Данный регистр содержит значение строки маршрутизации, соответствующее адресу Addr4 из регистра ADDRESS1. Он реализован с доступом по чтению и по записи. Формат регистра STR1\_4 представлен в таблице 4.8.

Таблица 4.8. Формат регистра STR1\_4.

Бит	Имя поля	Назначение
31..0	str	Значение строки маршрутизации

После выхода контроллера из состояния сброса значение этого регистра – “0”.

#### 4.2.6 Регистр версии контроллера ID\_VER

Данный регистр содержит номер версии контроллера SpaceWire. Значение регистра равно 0x000C. Регистр доступен только по чтению.

#### 4.2.7 Регистр RMAP\_CONTR2

Регистр RMAP\_CONTR2 доступен хосту по чтению и записи. В данном регистре задается значение логического адреса узла и значение destination key для RMAP контроллера. Формат регистра RMAP\_CONTR2 представлен в таблице 4.9.

Таблица 4.9. Формат регистра RMAP\_CONTR2

Бит	Имя поля	Назначение
15:8	Rmap_dest_key	Значение Destination key для данного RMAP узла. Значение по умолчанию – “0”.
7:0	rmap_logical_addr	Значение логического адреса для данного узла (используется RMAP-контроллером). Значение по умолчанию - 254

#### 4.2.8 Регистр статуса – Status

Регистр Status предназначен для оперативного контроля состояния фаз работы порта SpaceWire. Регистр доступен по чтению и записи. Сброс ряда разрядов регистра может осуществляться хостом путем записи в них “1”.

Формат регистра STATUS представлен в таблице 4.10.

**Таблица 4.10. Формат регистра STATUS**

Бит	Имя поля	Назначение
31	D_LVDS	Текущее значение S_LVDS (если в регистре режима установлен режим тестирования LVDS)
30	S_LVDS	Текущее значение S_LVDS (если в регистре режима установлен режим тестирования LVDS)
29:13	–	Не используется
12	CONNECTED	Устанавливается в “1” при принятии первого бита при установке соединения. После выхода Slave контроллера или DS-макроячейки из состояния сброса этот разряд установлен в “0”
11	BUFF_EMPTY	Устанавливается в “1”, если буфер порта SpaceWire пуст. После выхода контроллера из состояния сброса этот разряд установлен в “0”
10:9	–	Не используется
8	BUFF_FULL	Устанавливается в “1”, если буфер порта SpaceWire полон. После выхода Slave контроллера или из состояния сброса этот разряд установлен в “0”
7:5	DS_STATE	Номер состояния, в котором в данный момент находится машина состояний DS-макроячейки: “000” – ErrorReset – начальное состояние (состояние сброса); “001” – ErrorWait – ожидание возникновения ошибки; “010” – Ready – состояние готовности; “011” – Started – начало передачи; “100” – Connecting – ожидание кредитования; “101” – Run – передача данных. После выхода контроллера или DS-макроячейки из состояния сброса эти разряды установлены в “0”
4	–	Не используется
3	CREDIT_ERR	Признак ошибки кредитования: “1” – ошибка произошла; “0” – нет ошибки (после сигнала сброса). Запись “1” в этот разряд сбрасывает этот разряд в “0”. После выхода контроллера или DS-макроячейки из состояния сброса этот разряд установлен в “0”
2	ESC_ERR	Признак ошибки в ESC последовательности: “1” – ошибка произошла; “0” – нет ошибки (после сигнала сброса). Запись “1” в этот разряд сбрасывает этот разряд в '0'. После выхода контроллера или DS-макроячейки из состояния сброса этот разряд установлен в “0”
1	P_ERR	Признак ошибки четности: “1” – ошибка произошла; “0” – нет ошибки (после сигнала сброса). Запись “1” в этот разряд сбрасывает этот разряд в '0'. После выхода контроллера или DS-макроячейки из состояния сброса этот разряд установлен в “0”

Бит	Имя поля	Назначение
0	DC_ERR	Признак ошибки рассоединения (DisconnectError): “1” – ошибка произошла; “0” – нет ошибки (после сигнала сброса). Запись “1” в этот разряд сбрасывает этот разряд в “0”. После выхода контроллера или DS-макроячейки из состояния сброса этот разряд установлен в “0”

#### 4.2.9 Регистр режима работы – MODE\_CR

Регистр режима работы доступен по чтению и записи. Формат регистра MODE\_CR представлен в таблице 4.11.

Таблица 4.11. Формат регистра MODE\_CR

Бит	Имя поля	Назначение
31	D_LVDS	Значение, которое будет подано на вход D LVDS в режиме тестирования LVDS
30	S_LVDS	Значение, которое будет подано на вход S LVDS в режиме тестирования LVDS
29	LVDS_regime	Если данный разряд установлен в “1”, то включается режим тестирования LVDS. Значение по умолчанию – “0”
28:27	–	Не используется
26	Link_NR	Данный разряд рекомендуется устанавливать в значение “1” если планируется выполнить программный подбор скорости передачи по данному порту. Если этот разряд установлен в “1”, то передача всех пакетов, адресованных в данный порт будет приостановлена (они будут накапливаться в буферах). Значение по умолчанию – “0”
25	CODEC_TIMER_RE D	Установка этого разряда в “1” используется для тестирования кодеров/декодеров SpaceWire с сокращенными временами таймаутов установки соединения. Значение по умолчанию – “0”
24	AUTO_SPEED	Если данный разряд установлен в “1”, то разрешена автоматическая установка скорости передачи. Значение по умолчанию – “1”
23:21	–	Не используется
20:18	ACK_CODE	Значение разрядов с пятого по седьмой управляющего кода, который будет интерпретироваться как код подтверждения (используется только в режиме пятиразрядных кодов распределенных прерываний). Значение по умолчанию “101”
17:15	INT_CODE	Значение разрядов с пятого по седьмой управляющего кода, который будет интерпретироваться как код распределенного прерывания (используется только в режиме пятиразрядных кодов распределенных прерываний). Значение по умолчанию “100”
14	INT_ACK_REGIME	Если данный разряд установлен в “0”, то по этому порту используются шестиразрядные коды распределенных прерываний, если в “1” – то пятиразрядные коды распределенных прерываний. После выхода контроллера из состояния сброса этот разряд установлен в “1”
13	BUF_MODE	Тип буферизации порта SpaceWire (“0” – запрос канала на передачу, если в буфере есть хотя бы один символ. “1” – запрос канала на передачу, если в буфере есть хотя бы один пакет или буфер полон). После выхода контроллера из состояния сброса этот разряд установлен в “0”
12	CODEC_LOOPBACK	При установке в “1” включается режим Codec LoopBack. После выхода контроллера из состояния сброса этот разряд установлен в “0”
11	LVDS_LOOPBACK	При установке в “1” включается режим LVDS LoopBack. После выхода контроллера из состояния сброса этот разряд установлен в “0”
10:3	–	Не используется

Бит	Имя поля	Назначение
2	LinkStart	Установка LinkStart для блока DS-кодирования, при записи в этот разряд “1” управляющий сигнал LinkStart устанавливается в “1”, при записи “0” – сбрасывается. После выхода контроллера из состояния сброса этот разряд установлен в “1”
1	AutoStart	Установка Autostart для блока DS-кодирования, при записи в этот разряд “1” управляющий сигнал Autostart устанавливается в “1”, при записи “0” – сбрасывается. После выхода контроллера из состояния сброса этот разряд установлен в “0”
0	LinkDisabled	Установка LinkDisabled для блока DS-кодирования. При записи в этот разряд “1” управляющий сигнал LinkDisabled устанавливается в “1”, при записи “0” – сбрасывается. После выхода контроллера из состояния сброса этот разряд установлен в “0”

В начале работы и по сигналу сброса бит LinkDisabled устанавливается в “0”, бит AutoStart=0 и LinkStart=1. В результате сразу по включении питания порты SpaceWire начнут осуществлять попытки установки соединения

#### 4.2.10 Регистр коэффициента скорости передачи – TX\_SPEED

Регистр коэффициента скорости передачи доступен по чтению и по записи. Данный регистр предназначен для задания коэффициента скорости передачи, который поступает на TX\_PLL. Формат регистра показан в таблице 4.12.

Таблица 4.12. Формат регистра TX\_SPEED

Бит	Имя поля	Назначение
28:20	KOEFF_10	Значение коэффициента для подсчета таймаутов установки соединения
19:10	TX_SPEED10	Определяет скорость передачи данных в ходе установки соединения, если задан режим автоматической установки соединения
	TX_SPEED10.SEL= TX_SPEED10(7..0)	Значение коэффициента скорости передачи
	TX_SPEED10.PIEN= TX_SPEED10(8)	Разрешение работы TX_PLL, для разрешения работы должно быть установлено в “1”
	TX_SPEED10.lvdsEN= TX_SPEED10(9)	Разрешение работы LVDS, для разрешения работы должно быть установлено в “1”
9:0	TX_SPEED	Определяет скорость передачи данных (скорость передачи данных после установки соединения, если задан режим автоматической установки соединения)
	TX_SPEED.SEL= TX_SPEED(7..0)	Значение коэффициента скорости передачи
	TX_SPEED.PIEN= TX_SPEED(8)	Разрешение работы TX_PLL, для разрешения работы должно быть установлено в “1”
	TX_SPEED10.lvdsEN= TX_SPEED(9)	Разрешение работы LVDS, для разрешения работы должно быть установлено в “1”

#### ***4.2.11 Регистр коэффициента скорости приема – RX\_SPEED***

Десятиразрядный регистр коэффициента скорости приема доступен хосту по чтению и по записи.

Значению 0x3FF этого регистра соответствует значение 800Мбит/с в канале. Значению 0x00 этого регистра соответствует значение 0 Мбит/с в канале. Остальные значения скорости в канале отражает прямая зависимость между значением регистра и реальной скорости в канале.

Значение регистра обновляется каждые 2048 тактов локальной частоты контроллера (100 МГц) в соответствии с оценкой текущей скорости приема. При изменении скорости во время этого периода в тактов будет соответствовать некоторое смежное значение скорости между скоростью на которой шла передача данных в канале и на которую сделано переключение в канале.

## 5 КОРПУС

Микросхема 1892ХД5Т изготовлена в металлокерамическом корпусе CQFP-112. Расположение выводов корпуса и их названия показаны на рисунке 4.1 и в таблице 4.1 соответственно.

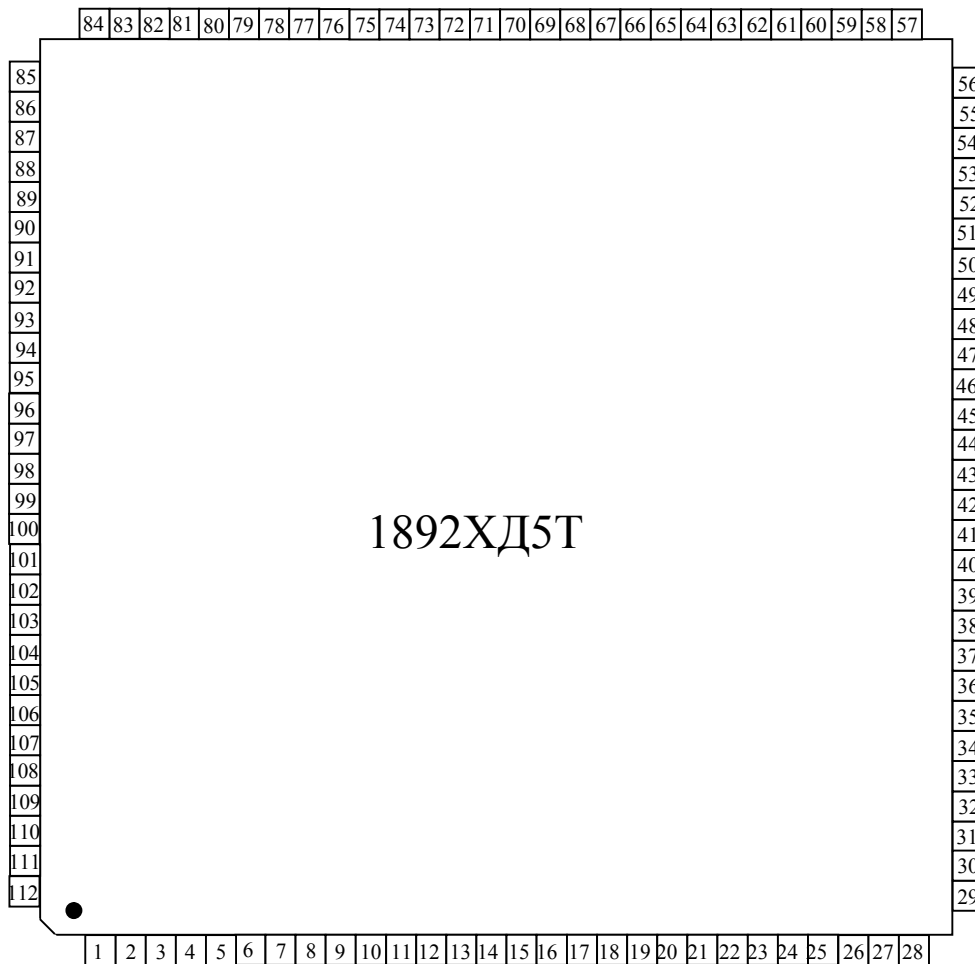


Рисунок 4.1 Расположение выводов корпуса CQFP-112.



Таблица 4.1 Нумерация выводов в корпусе CQFP-112.

Номер вывода	Название вывода	Номер вывода	Название вывода	Номер вывода	Название вывода	Номер вывода	Название вывода
1	GPIO_ADDR[14]	29	GPIO_DATA[15]	57	SCK	85	DVDD
2	GPIO_ADDR[15]	30	DVDD	58	SCSn	86	OCTRA0
3	GPIO_DATA[0]	31	DGND	59	DINM_A	87	OCTRB0
4	DVDD	32	OCTRA1	60	DINP_A	88	ICTRA0
5	DGND	33	OCTRB1	61	SGND	89	ICTRB0
6	GPIO_DATA[1]	34	ICTRA1	62	SINM_A	90	GPIO_ADDR[0]
7	GPIO_DATA[2]	35	ICTRB1	63	SINP_A	91	GPIO_ADDR[1]
8	VDD	36	VDD	64	SVDD	92	VDD
9	GND	37	GND	65	SOUTM_A	93	GND
10	GPIO_DATA[3]	38	MODE	66	SOUTP_A	94	GPIO_ADDR[2]
11	GPIO_DATA[4]	39	PLL_EN	67	SGND	95	GPIO_ADDR[3]
12	GPIO_DATA[5]	40	RSTn	68	DOUTM_A	96	GPIO_ADDR[4]
13	DVDD	41	DVDD	69	DOUTP_A	97	GPIO_ADDR[5]
14	DGND	42	DGND	70	SVDD	98	DVDD
15	GPIO_DATA[6]	43	DGND	71	DINM_B	99	DGND
16	GPIO_DATA[7]	44	RSTO	72	DINP_B	100	GPIO_ADDR[6]
17	GPIO_DATA[8]	45	CLKO	73	SGND	101	GPIO_ADDR[7]
18	GPIO_DATA[9]	46	IRQ	74	SINM_B	102	VDD
19	GPIO_DATA[10]	47	DVDD	75	SINP_B	103	VDD
20	VDD	48	XTALI	76	SVDD	104	GND
21	GND	49	XTALO	77	SOUTM_B	105	GPIO_ADDR[8]
22	GND	50	DGND	78	SOUTP_B	106	GPIO_ADDR[9]
23	DVDD	51	DVDD	79	SGND	107	GPIO_ADDR[10]
24	DGND	52	DGND	80	DOUTM_B	108	GPIO_ADDR[11]
25	GPIO_DATA[11]	53	MISO	81	DOUTP_B	109	DVDD
26	GPIO_DATA[12]	54	MOSI	82	SVDD	110	DGND
27	GPIO_DATA[13]	55	VDD	83	DGND	111	GPIO_ADDR[12]
28	GPIO_DATA[14]	56	GND	84	-	112	GPIO_ADDR[13]

## **6 ИСТОРИЯ ИЗМЕНЕНИЙ**

### **6.1 24 НОЯБРЯ 2014**

- В разделе 3.3 уточнены особенности реализации протокола RMAP.
- В разделе 4.1.2 уточнено значение регистра DEVID.

### **6.2 3 ФЕВРАЛЯ 2014**

- Добавлен раздел 4.2 с описанием регистров контроллера SpaceWire.