

### Основные особенности

- Напряжение питания 5,0 В;
- Ток потребления 8,0 мА;
- Микроконтроллерное ядро архитектуры 8051:
  - машинный цикл 1 такт.
- Встроенные модули:
  - 12 разрядный АЦП;
  - 12 разрядный ЦАП;
  - источник опорного напряжения с масштабирующим ОУ;
  - RC-генератор;
  - супервизор питания;
  - блок ФАПЧ;
  - термодатчик;
  - регуляторы напряжения.
- Режим энергосбережения («SLEEP»);
- Температурный диапазон
  - группа А: от –60°C до +85°C;
  - группа Б: от –45°C до +85°C;
  - группа В: от –25°C до +85°C.

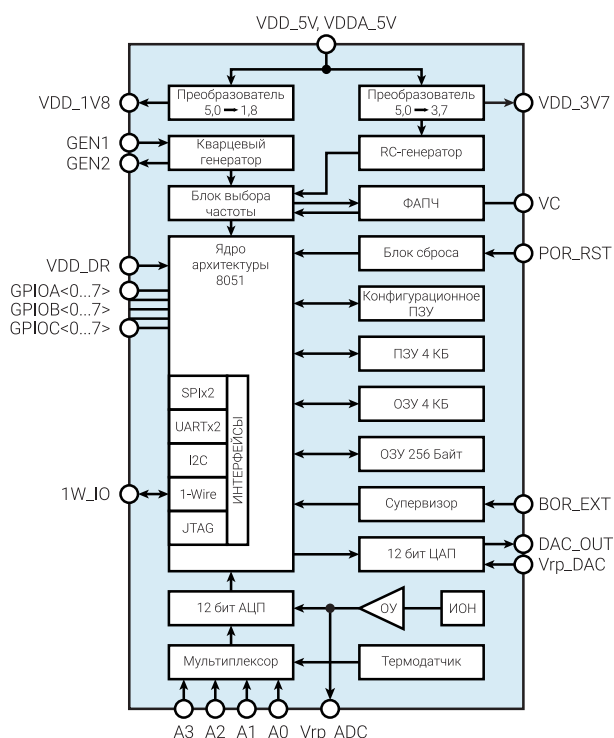


Рисунок 1. Структурная схема



ГГ – год выпуска  
НН – неделя выпуска

Рисунок 2. Внешний вид  
микросхемы 5400TP105-003

### Общее описание

Микросхема 5400TP105-003 – 8-ми разрядный контроллер с архитектурой ядра 8051 для бортовых систем управления с диапазоном рабочих частот до 8,0 МГц (тактирование системы осуществляется несколькими способами: кварцевый резонатор, RC-генератор, внешний источник, а также путем умножения этих частот с помощью встроенного блока ФАПЧ). Микросхема выполнена на базе радиационно-стойкого аналого-цифрового БМК 5400TP10 по технологии КНИ.

Микросхема работает как в отладочном режиме с возможностью многократного перезаписывания исполняемой программы (до 4 КБ), так и в основном режиме после записи информации в однократно программируемую память (antifuse, до 4 КБ). После программирования микросхема готова к работе при включении питания, времени на загрузку не требуется.

Микросхема выполнена в 48-ми выводном металлокерамическом корпусе 5142.48-А.

**Вычислительное ядро**

- система команд 8051;
- тактовая частота до 8 МГц;
- машинный цикл 1 такт;
- возможность выбора источника тактирования: кварцевый генератор, RC-генератор, внешний источник;
- настраиваемые прерывания по внешним событиям.

**Встроенные периферийные модули**

- два интерфейса SPI;
- два интерфейса UART;
- интерфейс I2C;
- интерфейс 1-Wire;
- интерфейс JTAG (программирование и отладка);
- 24 универсальные линии ввода/вывода GPIO с индивидуальной настройкой направления;
- три 24-разрядных таймера/счетчика;
- сторожевой таймер;
- модуль перевода системы в режим пониженного энергопотребления (SLEEP).
- 4-канальный 12-разрядный 200 кВыб/с АЦП;
- 12-разрядный ЦАП;
- источник опорного напряжения с масштабирующим ОУ (с возможностью программирования коэффициента усиления);
- RC-генератор с возможностью настройки частоты;
- блок ФАПЧ с возможностью настройки коэффициента умножения;
- супервизор питания;
- термодатчик;
- регуляторы напряжения электропитания.

**Память***Режим «HARD»:*

- память программ 4 КБ (ОППЗУ);
- память данных 4352 байт внешней (большая и малая ОЗУ) и 256 внутренней ОЗУ (ядро 8051).

*Режим «SOFT»:*

- память программ 4 КБ (большая ОЗУ);
- память данных 256 байт внешней (малая ОЗУ) и 256 внутренней ОЗУ (ядро 8051).

**Функциональное назначение микросхемы**

- прием, усиление и обработка аналоговых и цифровых сигналов;
- реализация бортовых систем управления;
- аналого-цифровое и цифро-аналоговое преобразование сигнала.

## Электрические параметры микросхемы

Таблица 1. Электрические характеристики

| Параметр, единица измерения                                                                                                                                                                                               | Норма параметра |         |          |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---------|----------|
|                                                                                                                                                                                                                           | не менее        | типовое | не более |
| Выходное напряжение линейных регуляторов <sup>(1)</sup> , В                                                                                                                                                               |                 |         |          |
| 5,0 В → 3,7 В (вывод VDD_3V7)                                                                                                                                                                                             | 3,5             | 3,7     | 4,2      |
| 5,0 В → 1,8 В (вывод VDD_1V8)                                                                                                                                                                                             | 1,62            | 1,8     | 1,98     |
| Ток потребления, мА                                                                                                                                                                                                       |                 | 8,0     | 18       |
| Дифференциальная нелинейность ЦАП, МЗР                                                                                                                                                                                    | –0,99           |         | 3,0      |
| Интегральная нелинейность ЦАП, МЗР                                                                                                                                                                                        | –8,0            |         | 8,0      |
| Дифференциальная нелинейность АЦП <sup>(2)</sup> , МЗР                                                                                                                                                                    | –0,99           |         | 1,0      |
| Интегральная нелинейность АЦП <sup>(2)</sup> , МЗР                                                                                                                                                                        | –4,0            |         | 4,0      |
| Нижняя граница диапазона настройки частоты RC-генератора, кГц                                                                                                                                                             |                 |         | 90       |
| Верхняя граница диапазона настройки частоты RC-генератора, кГц                                                                                                                                                            | 400             |         |          |
| Ток утечки портов ввода/вывода (GPIO), мкА                                                                                                                                                                                |                 | 0,01    | 20       |
| Напряжение высокого уровня выходных цифровых сигналов, В                                                                                                                                                                  |                 |         |          |
| при VDD_DR = 2,5 В                                                                                                                                                                                                        | VDD_DR – 0,3    | 2,5     |          |
| при VDD_DR = 5,0 В                                                                                                                                                                                                        | VDD_DR – 0,4    | 5,0     |          |
| Напряжение низкого уровня выходных цифровых сигналов, В                                                                                                                                                                   |                 |         |          |
| при VDD_DR = 2,5 В                                                                                                                                                                                                        |                 | 0       | 0,3      |
| при VDD_DR = 5,0 В                                                                                                                                                                                                        |                 | 0       | 0,4      |
| <b>Справочные данные</b>                                                                                                                                                                                                  |                 |         |          |
| Ток потребления микросхемы в режиме низкого энергопотребления («SLEEP»), мкА                                                                                                                                              |                 |         |          |
| Задействованы блоки: ИОН, регуляторы напряжения, RC-генератор на низкой частоте, цифровая часть в режиме таймера                                                                                                          |                 |         |          |
| при частоте тактирования ~ 300 кГц                                                                                                                                                                                        |                 | 110     |          |
| при частоте тактирования ~ 50 кГц                                                                                                                                                                                         |                 | 80      |          |
| <b>Примечание:</b><br>1) Линейные регуляторы предназначены для формирования внутреннего питания, нагрузка выводов VDD_3V7, VDD_1V8 недопустима.<br>2) Расчет производится для преобразования с разрядностью равной 10 бит |                 |         |          |

## Электростатическая защита

Микросхема имеет встроенную защиту от электростатического разряда до 1000 В по модели человеческого тела. Требуется мер предосторожности.

## Предельно-допустимые и предельные режимы эксплуатации

Таблица 2. Предельно-допустимые и предельные режимы эксплуатации

| Параметр, единица измерения                                                                                                                                                                                                                                                                                                | Предельно-допустимый режим                                     |                               | Предельный режим |                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|-------------------------------|------------------|-------------------------------|
|                                                                                                                                                                                                                                                                                                                            | не менее                                                       | не более                      | не менее         | не более                      |
| Аналоговое напряжение питания (VDDA_5V), В                                                                                                                                                                                                                                                                                 | 4,5                                                            | 5,25                          | –0,1             | 5,35                          |
| Цифровое напряжение питания (VDD_5V), В                                                                                                                                                                                                                                                                                    | 4,5                                                            | 5,25                          | –0,1             | 5,35                          |
| Напряжение питания интерфейсной части (VDD_DR), В                                                                                                                                                                                                                                                                          | 2,25                                                           | 5,25                          | –0,1             | 5,35                          |
| Напряжение программирования ПЗУ (VPP_9V), В                                                                                                                                                                                                                                                                                | 8,5                                                            | 9,0                           | –0,1             | 9,5                           |
| Напряжение внешнего опорного уровня АЦП (V <sub>rp</sub> _ADC), В                                                                                                                                                                                                                                                          | 2,25                                                           | 5,0                           | –0,1             | VDDA_5V                       |
| Напряжение внешнего опорного уровня ЦАП (V <sub>rp</sub> _DAC), В                                                                                                                                                                                                                                                          | 2,25                                                           | 5,0                           | –0,1             | VDDA_5V                       |
| Напряжение высокого уровня входных цифровых сигналов (GPIO в режиме входа, BOR_EXT/PGM, TM, DBG, GEN1), В                                                                                                                                                                                                                  | VDD_DR<br>–0,4                                                 | VDD_DR<br>+0,3 <sup>(1)</sup> | –0,1             | VDD_DR<br>+0,5 <sup>(2)</sup> |
| Напряжение низкого уровня входных цифровых сигналов (GPIO в режиме входа, BOR_EXT/PGM, TM, DBG, GEN1), В                                                                                                                                                                                                                   | 0                                                              | 0,4                           | –0,1             | VDD_DR<br>+0,5 <sup>(2)</sup> |
| Нагрузочная способность портов ввода/вывода (GPIO), мА                                                                                                                                                                                                                                                                     |                                                                |                               |                  |                               |
| при VDD_DR = 2,5 В                                                                                                                                                                                                                                                                                                         | –                                                              | 1,5                           | –                | 3,0                           |
| при VDD_DR = 5,0 В                                                                                                                                                                                                                                                                                                         | –                                                              | 3,0                           | –                | 5,0                           |
| Нагрузочная способность вывода DAC_OUT <sup>(6)</sup> , мА                                                                                                                                                                                                                                                                 | –                                                              | 1,0                           | –                | 1,5                           |
| Рабочая частота тактирования, МГц                                                                                                                                                                                                                                                                                          | –                                                              | 8,0                           | –                | 8,0                           |
| Рабочая частота тактирования в режиме отладки <sup>(7)</sup> , МГц                                                                                                                                                                                                                                                         | –                                                              | 16                            | –                | 16                            |
| Температура эксплуатации, °С                                                                                                                                                                                                                                                                                               | –25 <sup>(3)</sup><br>–45 <sup>(4)</sup><br>–60 <sup>(5)</sup> | +85                           | –60              | +150                          |
| Примечание:<br>1) не более 5,25 В<br>2) не более 5,35 В<br>3) для группы В<br>4) для группы Б<br>5) для группы А<br>6) при включенном выходном буфере (бит DAC_BUF_EN = «1», регистр ANALOG_O_BUF)<br>7) режим отладки – программирование и настройка микроконтроллера с использованием программы из встроенного ОЗУ в НКУ |                                                                |                               |                  |                               |

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

## Конфигурация и функциональное описание выводов

Таблица 3. Функциональное описание выводов

| № вывода | Тип вывода  | Наименование вывода | Назначение вывода                                                                                                                                                                                                                              |
|----------|-------------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | DI/DO       | GPIOB<5>            | Порт ввода-вывода микроконтроллера, разряд №5 группы В                                                                                                                                                                                         |
| 2        | DI/DO       | GPIOB<6>            | Порт ввода-вывода микроконтроллера, разряд №6 группы В                                                                                                                                                                                         |
| 3        | DI/DO       | GPIOB<7>            | Порт ввода-вывода микроконтроллера, разряд №7 группы В                                                                                                                                                                                         |
| 4        | PWR         | VDD_5V              | Вывод цифрового положительного напряжения питания                                                                                                                                                                                              |
| 5        | AO          | VC                  | Вывод блока ФАПЧ для подключения RC-фильтра                                                                                                                                                                                                    |
| 6        | DI          | GEN1                | Вход подключения кварцевого резонатора/<br>вход для подачи внешней тактовой частоты                                                                                                                                                            |
| 7        | DO          | GEN2                | Вход подключения кварцевого резонатора                                                                                                                                                                                                         |
| 8        | PWR         | VSSD                | Вывод цифрового отрицательного напряжения питания                                                                                                                                                                                              |
| 9        | DI/DO       | GPIOC<0>/<br>TCK    | Порт ввода-вывода микроконтроллера, разряд №0 группы С/<br>вход TCK интерфейса JTAG в тестовом режиме (TM = 1)                                                                                                                                 |
| 10       | DI/DO       | GPIOC<1>/<br>TMS    | Порт ввода-вывода микроконтроллера, разряд №1 группы С/<br>вход TMS интерфейса JTAG в тестовом режиме (TM = 1)                                                                                                                                 |
| 11       | DI/DO       | GPIOC<2>/<br>TDI    | Порт ввода-вывода микроконтроллера, разряд №2 группы С/<br>вход TDI интерфейса JTAG в тестовом режиме (TM = 1)                                                                                                                                 |
| 12       | DI/DO       | GPIOC<3>/<br>TDO    | Порт ввода-вывода микроконтроллера, разряд №3 группы С/<br>выход TDO интерфейса JTAG в тестовом режиме (TM = 1)                                                                                                                                |
| 13       | DI/DO       | GPIOC<4>            | Порт ввода-вывода микроконтроллера, разряд №4 группы С                                                                                                                                                                                         |
| 14       | DI/DO       | GPIOC<5>            | Порт ввода-вывода микроконтроллера, разряд №5 группы С                                                                                                                                                                                         |
| 15       | DI/DO       | GPIOC<6>            | Порт ввода-вывода микроконтроллера, разряд №6 группы С                                                                                                                                                                                         |
| 16       | DI/DO       | GPIOC<7>            | Порт ввода-вывода микроконтроллера, разряд №7 группы С                                                                                                                                                                                         |
| 17       | AO /<br>PWR | VDD_1V8             | Вывод напряжения питания ядра<br>(выходное напряжение LDO-регулятора 5,0 В → 1,8 В)                                                                                                                                                            |
| 18       | PWR         | VDD_DR              | Вывод положительного напряжения питания универсальных<br>портов ввода-вывода микроконтроллера 2,5 В – 5,0 В                                                                                                                                    |
| 19       | DI/DO       | 1W_IO               | Вывод интерфейса 1-Wire (тип вывода – открытый сток)                                                                                                                                                                                           |
| 20       | DI/DO       | GPIOA<0>            | Порт ввода-вывода микроконтроллера, разряд №0 группы А                                                                                                                                                                                         |
| 21       | DI/DO       | GPIOA<1>            | Порт ввода-вывода микроконтроллера, разряд №1 группы А                                                                                                                                                                                         |
| 22       | DI/DO       | GPIOA<2>            | Порт ввода-вывода микроконтроллера, разряд №2 группы А                                                                                                                                                                                         |
| 23       | DI/DO       | GPIOA<3>            | Порт ввода-вывода микроконтроллера, разряд №3 группы А                                                                                                                                                                                         |
| 24       | DI/DO       | GPIOA<4>            | Порт ввода-вывода микроконтроллера, разряд №4 группы А                                                                                                                                                                                         |
| 25       | DI/DO       | GPIOA<5>            | Порт ввода-вывода микроконтроллера, разряд №5 группы А                                                                                                                                                                                         |
| 26       | DI/DO       | GPIOA<6>            | Порт ввода-вывода микроконтроллера, разряд №6 группы А                                                                                                                                                                                         |
| 27       | DI/DO       | GPIOA<7>            | Порт ввода-вывода микроконтроллера, разряд №7 группы А                                                                                                                                                                                         |
| 28       | DI          | BOR_EXT/<br>PGM     | Вход внешнего супервизора питания/<br>вход выбора режима программирования ПЗУ в тестовом<br>режиме (TM = 1):<br>лог. «1» – перевод в режим программирования,<br>от шины отключается CPU и подключается JTAG;<br>лог. «0» – стандартная работа. |

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

| № вывода                                                                                                                                    | Тип вывода | Наименование вывода    | Назначение вывода                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------|------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29                                                                                                                                          | DI         | TM                     | Вход для выбора режима работы микроконтроллера:<br>лог. «1» – тестовый режим;<br>лог. «0» – стандартная работа.                                                                                                                                                              |
| 30                                                                                                                                          | DI         | DBG                    | Вход для выбора блока памяти, выполняющего роль ПЗУ:<br>лог. «1» – память ПЗУ (ROM_RAM_4KB) 4 КБ, ОЗУ 256 Б;<br>лог. «0» – память ПЗУ (ROM_OTP_4KB) 4 КБ, ОЗУ 4 КБ + 256 Б.                                                                                                  |
| 31                                                                                                                                          | DI         | POR_RST                | Вход для подключения внешнего конденсатора блока сброса                                                                                                                                                                                                                      |
| 32                                                                                                                                          | AI         | Vrp_DAC                | Вывод положительного опорного напряжения ЦАП                                                                                                                                                                                                                                 |
| 33                                                                                                                                          | AO         | DAC_OUT                | Выход ЦАП                                                                                                                                                                                                                                                                    |
| 34                                                                                                                                          | AI         | VPP_9V                 | Вывод для программирования ПЗУ и конфигурационной памяти                                                                                                                                                                                                                     |
| 35                                                                                                                                          | PWR        | VDDA_5V                | Вывод аналогового положительного напряжения питания                                                                                                                                                                                                                          |
| 36                                                                                                                                          | AO / PWR   | VDD_3V7                | Вывод положительного напряжения питания RC-генератора (выходное напряжение LDO-регулятора 5,0 В → 3,7 В)                                                                                                                                                                     |
| 37                                                                                                                                          | PWR        | VSSA                   | Вывод аналогового отрицательного напряжения питания                                                                                                                                                                                                                          |
| 38                                                                                                                                          | AI/AO      | Vrp_ADC                | Вывод положительного опорного напряжения АЦП/<br>выход масштабирующего ОУ (если ОУ включен в ходе настройки конфигурационной памяти)                                                                                                                                         |
| 39                                                                                                                                          | AI         | A0                     | Вход 0-го канала АЦП                                                                                                                                                                                                                                                         |
| 40                                                                                                                                          | AI         | A1                     | Вход 1-го канала АЦП                                                                                                                                                                                                                                                         |
| 41                                                                                                                                          | AI         | A2                     | Вход 2-го канала АЦП                                                                                                                                                                                                                                                         |
| 42                                                                                                                                          | AI         | A3                     | Вход 3-го канала АЦП                                                                                                                                                                                                                                                         |
| 43                                                                                                                                          | PWR        | VDD_tech               | Технологический вывод (объединить с VDD_5V и/или VDDA_5V)                                                                                                                                                                                                                    |
| 44                                                                                                                                          | DI/DO      | GPIOB<0>/<br>H_S       | Порт ввода-вывода микроконтроллера, разряд №0 группы В/<br>выбор режима конфигурационной памяти в тестовом режиме (TM = 1):<br>лог. «1» – источником данных конфигурационной памяти являются регистры;<br>лог. «0» – источником данных конфигурационной памяти является ПЗУ. |
| 45                                                                                                                                          | DI/DO      | GPIOB<1>/<br>RC_CLKOUT | Порт ввода-вывода микроконтроллера, разряд №1 группы В/<br>выходная частота RC-генератора в тестовом режиме (TM = 1)                                                                                                                                                         |
| 46                                                                                                                                          | DI/DO      | GPIOB<2>               | Порт ввода-вывода микроконтроллера, разряд №2 группы В                                                                                                                                                                                                                       |
| 47                                                                                                                                          | DI/DO      | GPIOB<3>               | Порт ввода-вывода микроконтроллера, разряд №3 группы В                                                                                                                                                                                                                       |
| 48                                                                                                                                          | DI/DO      | GPIOB<4>               | Порт ввода-вывода микроконтроллера, разряд №4 группы В                                                                                                                                                                                                                       |
| Примечание:<br>DI – цифровой вход<br>DO – цифровой выход<br>AI – аналоговый вход<br>AO – аналоговый выход<br>PWR – вывод напряжения питания |            |                        |                                                                                                                                                                                                                                                                              |

## Эквивалентные схемы

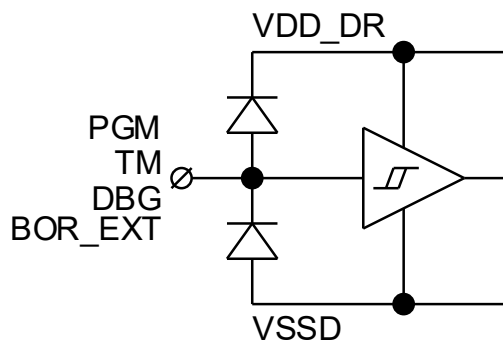


Рисунок 3. Цифровые входы  
PGM, TM, DBG, BOR\_EXT

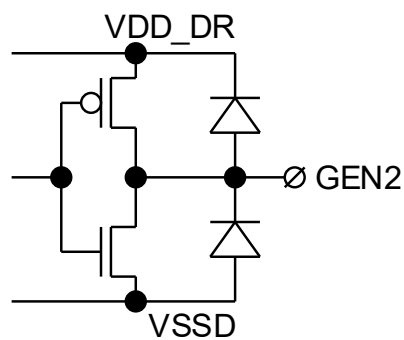


Рисунок 4. Цифровой выход GEN2

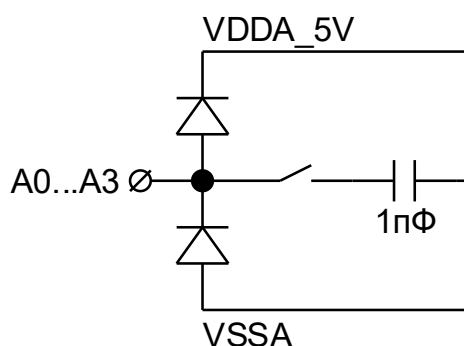


Рисунок 5. Аналоговые входы  
A0, A1, A2, A3

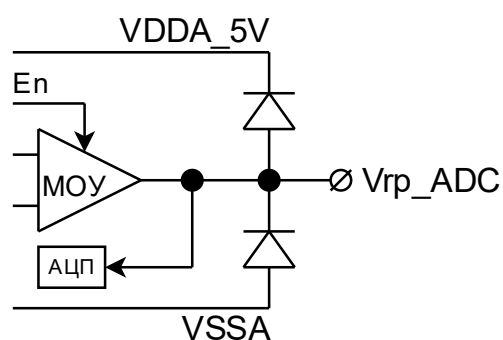


Рисунок 6. Вывод положительного опорного  
напряжения АЦП Vrp\_ADC

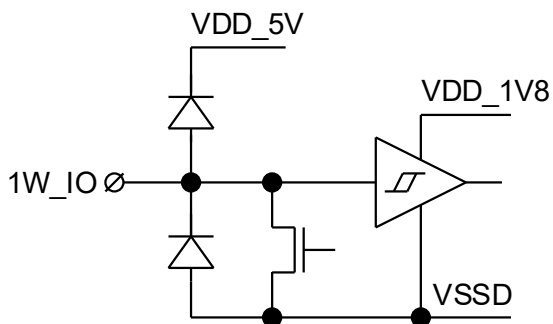


Рисунок 7. Цифровой вход/выход 1W\_IO

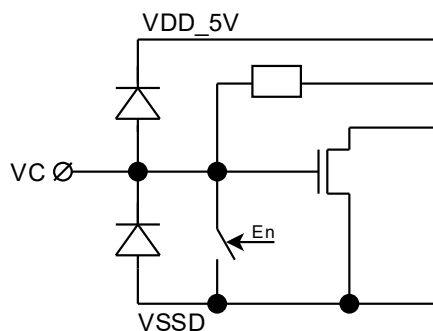


Рисунок 8. Вывод блока ФАПЧ  
для подключения RC-фильтра VC

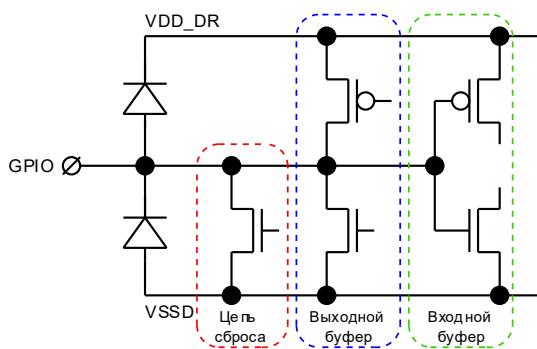


Рисунок 9. Порты ввода-вывода GPIO

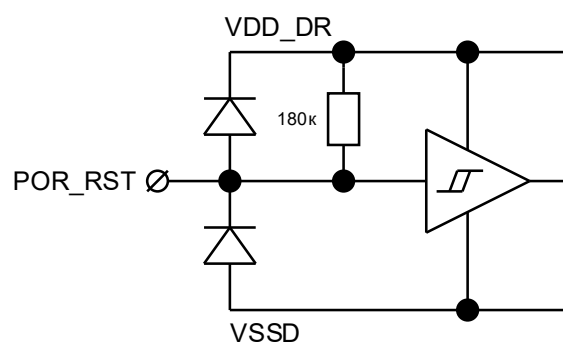
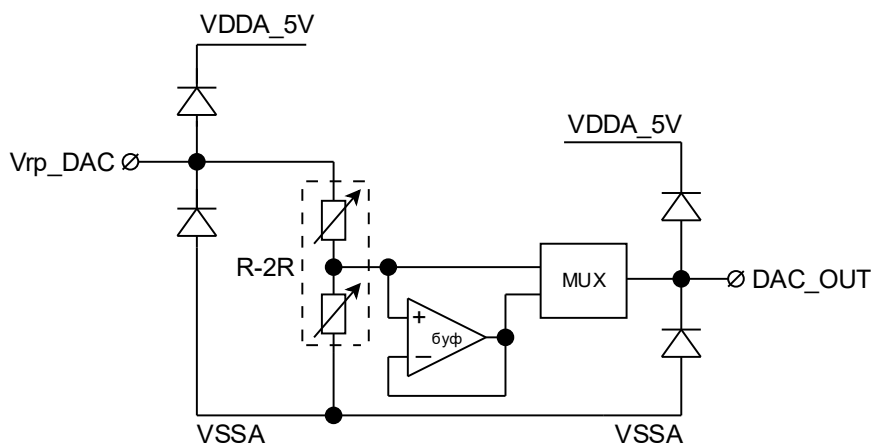


Рисунок 10. Вывод POR\_RST

Рисунок 11. Вывод ЦАП DAC\_OUT  
и положительного опорного напряжения ЦАП Vrp\_DAC

Рекомендуемая схема применения

Таблица 4. Таблица внешних компонентов

| Компонент                                    | Номинал    | Компонент | Номинал      |
|----------------------------------------------|------------|-----------|--------------|
| R1                                           | 1 – 2 МОм  | C1, C2    | 16 – 64 пФ   |
| R2                                           | 2 – 20 кОм | C3        | 10 нФ        |
| R3                                           | 100 Ом     | C4        | 1 нФ         |
| R4                                           | 4,7 кОм    | C5        | 20 нФ        |
| ZQ – кварцевый резонатор с частотой до 8 МГц |            | C6 – C13  | 100 – 200 нФ |

Конденсаторы высокочастотные керамические, либо сдвоенные. В случае сдвоенных конденсаторов, один из них обязательно должен быть высокочастотный керамический емкостью не менее 10 нФ. Шунтирующие конденсаторы должны располагаться на плате в непосредственной близости к соответствующим выводам микросхемы.

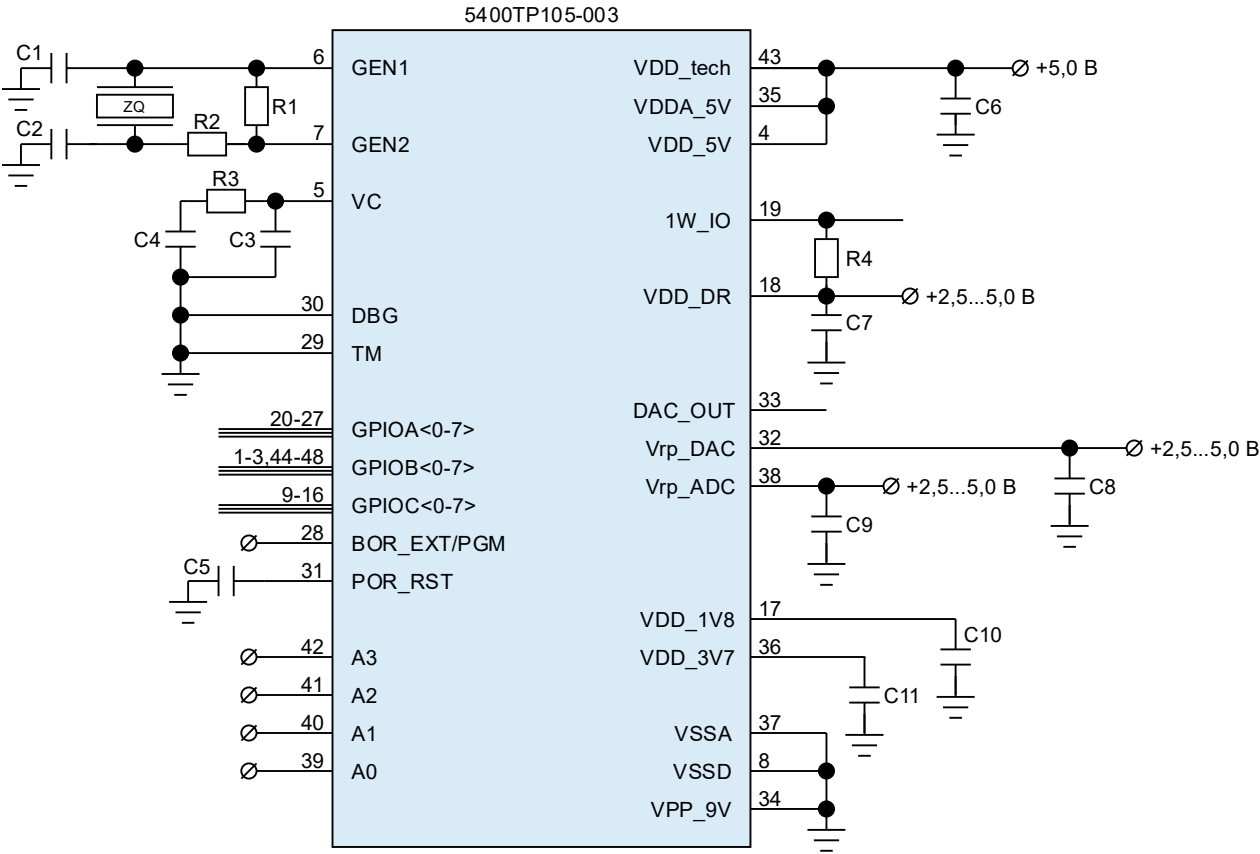


Рисунок 12. Рекомендуемая схема применения после программирования микросхемы в ПЗУ

## Важные замечания при работе с микросхемой

Если 1-Wire интерфейс не используется, то вывод 1W\_IO (19) необходимо оставить в обрыве.

При использовании I2C интерфейса необходимо выводы SDA и SCL подтянуть резисторами 4,7 кОм к питанию.

Если ФАПЧ не используется, то вывод VC (5) подключить к VSSA (37).

При использовании внешнего генератора, вывод GEN2 (7) необходимо оставить в обрыве.

При использовании внутреннего RC-генератора вывод GEN1 (6) необходимо объединить с VSSA, вывод GEN2 (7) оставить в обрыве.

Если размах цифровых уровней генератора 5,0 В, а напряжение питания портов ввода-вывода отлично от 5,0 В (например, 3,3 В или 2,5 В), то в конфигурационной памяти необходимо выбрать источник тактирования «Кварцевый резонатор» (GEN1\_QV или GEN1\_QV+PLL) путем записи соответствующих бит в регистр ANALOG\_O\_PLL (см. Таблица 6).

При использовании внутреннего супервизора питания (бит BOR = «0», регистр ANALOG\_O\_RC\_R), вывод BOR\_EXT/PGM необходимо подключить к VSSD.

Если ЦАП не используется, то вывод Vrp\_DAC (32) необходимо подключить к VSSA (37).

Если АЦП не используется, то выводы A3 – A0 необходимо подключить к VSSA (37).

На выводы Vrp\_DAC и Vrp\_ADC задается внешнее опорное напряжение для блоков АЦП и ЦАП. Есть возможность формирования внутреннего опорного напряжения с помощью масштабирующего операционного усилителя (МОУ). Для этого необходимо включить МОУ (бит OA\_EN = «1», регистр ANALOG\_O\_BUF) и настроить коэффициент масштабирования (биты OA\_GAIN\_N, OA\_GAIN\_M, регистр ANALOG\_O\_BUF). Данное напряжение можно проконтролировать на выводе Vrp\_ADC. Это же напряжение можно подать и на вход опорного уровня ЦАП с помощью внешнего соединения выводов Vrp\_ADC и Vrp\_DAC.

**Важно!** При TM = «1» микросхема работает в тестовом режиме (подробнее см. Конфигурационная память (ANALOG\_CFG)), H\_S = «1» (источником данных конфигурационной памяти являются регистры) и в бит REF\_OUT\_DISABLE регистра ANALOG\_RC\_R записан лог. «0», выход 39 работает как тестовый вывод опорного напряжения. В данном режиме нельзя подключать вывод 39 к VSSA (37), т.к. данный опорный уровень вместе с линейными регуляторами формирует питание аналоговой и цифровой частей микросхемы.

В случае перехода в тестовый режим и когда источником данных конфигурационной памяти являются регистры (H\_S = «1») рекомендуется всегда записывать в бит REF\_OUT\_DISABLE лог. «1».

В случае перехода в тестовый режим и когда источником данных конфигурационной памяти является ПЗУ (H\_S = «0») в бите REF\_OUT\_DISABLE по умолчанию уже записана лог. «1».

Также биты V\_REF в регистре ANALOG\_O\_REF являются технологическими и уже настроены и прошиты в ПЗУ на этапе производства. При прожиге конфигурационной памяти следует убедиться, что данные биты не содержат лог «1».

## Конфигурационная память (ANALOG\_CFG)

В микросхеме реализована конфигурационная память (ANALOG\_CFG), которая может работать как в «SOFT», так и в «HARD» режиме.

Регистры модуля ANALOG\_CFG подключены к конфигурационному однократно программируемому ПЗУ.

Вывод TM (Test Mode) определяет режим работы следующих выводов микросхемы 9–12, 28, 39, 44, 45. При TM = «1» выводы работают следующим образом: выводы 9-12 – JTAG; вывод 28 – PGM; вывод 39 – OUT\_REF; вывод 44 – H\_S; вывод 45 – RC.

В зависимости от вывода 44 – H\_S источником данных конфигурационной памяти могут быть, либо регистры, либо ПЗУ («1» – «SOFT» режим, «0» – «HARD» режим).

В «SOFT» режиме при подаче 9,0 В на вывод VPP\_9V конфигурационное ПЗУ прожигается.

При TM = 0 конфигурационная память будет работать в режиме «HARD», а вывод GPIOB<0>/H\_S как порт ввода-вывода.

### Регистры модуля ANALOG\_CFG

| №     | Аббревиатура  | Доступ | Описание                                                                                                                                                                              |
|-------|---------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3300h | ANALOG_O_BUF  | W      | Регистр управления выходным буфером ЦАП, управления выходным масштабирующим операционным усилителем (МОУ) после ИОН (вывод V <sub>gr_ADC</sub> ), настройки коэффициента усиления МОУ |
| 3301h | ANALOG_O_PLL  | W      | Регистр настройки источника тактовой частоты                                                                                                                                          |
| 3302h | ANALOG_O_RC   | W      | Регистр настройки емкости конденсатора RC-генератора                                                                                                                                  |
| 3303h | ANALOG_O_REF  | W      | Регистр управления конденсаторами частотной коррекции МОУ и настройки выходного напряжения ИОН (вывод A0)                                                                             |
| 3306h | ANALOG_O_RC_R | W      | Регистр настройки сопротивления резистора RC-генератора, выбора источника BOR, управления выводами GPIOB<1>, A0 и источниками тактовой частоты в тестовом режиме                      |

### ANALOG\_O\_BUF

| Бит                | 7          | 6     | 5         | 4 | 3 | 2         | 1 | 0 |
|--------------------|------------|-------|-----------|---|---|-----------|---|---|
| Назначение         | DAC_BUF_EN | OA_EN | OA_GAIN_N |   |   | OA_GAIN_M |   |   |
| Начальное значение | 0          |       |           |   |   |           |   |   |

**DAC\_BUF\_EN** – включение выходного буфера ЦАП:

- 1 – выходной буфер включен;
- 0 – выходной буфер выключен.

**OA\_EN** – включение МОУ (вывод V<sub>gr\_ADC</sub>):

- 1 – МОУ включен;
- 0 – МОУ выключен (Z-состояние).

**OA\_GAIN\_N** – коэффициент масштабирования N для настройки коэффициента усиления МОУ.

**OA\_GAIN\_M** – коэффициент масштабирования M для настройки коэффициента усиления МОУ.

Блок МОУ состоит из ОУ общего назначения и потенциометра, используемого для подбора коэффициентов масштабирования. На вход блока подается выходное напряжение ИОН, которое усиливается в зависимости от масштабирующих коэффициентов. Если блок выключен, то он находится в режиме ожидания и не потребляет ток. Имеется возможность отключения частотной коррекции.

Таблица 5. Настройка коэффициентов масштабирования

| Коэффициент усиления МОУ             |      |      |          |      |      |          |          |       |
|--------------------------------------|------|------|----------|------|------|----------|----------|-------|
| $\begin{matrix} M \\ N \end{matrix}$ | 000b | 001b | 010b     | 011b | 100b | 101b     | 110b     | 111b  |
| 000b                                 | 2    | 1,5  | 1,333333 | 1,25 | 1,2  | 1,166667 | 1,142857 | 1,125 |
| 001b                                 | 3    | 2    | 1,666667 | 1,5  | 1,4  | 1,333333 | 1,285714 | 1,25  |
| 010b                                 | 4    | 2,5  | 2        | 1,75 | 1,6  | 1,5      | 1,428571 | 1,375 |
| 011b                                 | 5    | 3    | 2,333333 | 2    | 1,8  | 1,666667 | 1,571429 | 1,5   |
| 100b                                 | 6    | 3,5  | 2,666667 | 2,25 | 2    | 1,833333 | 1,714286 | 1,625 |
| 101b                                 | 7    | 4    | 3        | 2,5  | 2,2  | 2        | 1,857143 | 1,75  |
| 110b                                 | 8    | 4,5  | 3,333333 | 2,75 | 2,4  | 2,166667 | 2        | 1,875 |
| 111b                                 | 9    | 5    | 3,666667 | 3    | 2,6  | 2,333333 | 2,142857 | 2     |

**ANALOG\_O\_PLL**

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | PLL |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

**PLL** – совместно с TM и TM\_CLK\_TYPE значения в данном регистре определяют и настраивают источник тактового сигнала микроконтроллера XTAL\_CLK, а также совместно с RC\_OUT\_DISABLE определяют источник частоты в тестовом режиме на выводе GPIOB<1>/RC\_CLK\_OUT.

Таблица 6. Определение и настройка источника тактовой частоты микроконтроллера

| TM | TM_CLK_TYPE | PLL |   |   |   |   |    |    |    | RC_CLK | XTAL_CLK     |
|----|-------------|-----|---|---|---|---|----|----|----|--------|--------------|
|    |             | 7   | 6 | 5 | 4 | 3 | 2  | 1  | 0  |        |              |
| 1  | 0           | x   | x | x | x | x | x  | x  | x  | RC     | GEN1_QV      |
| 1  | 1           | x   | 0 | 0 | x | 0 | x  | x  | x  | RC     | GEN1_EXT     |
| 1  | 1           | x   | 0 | 1 | x | 0 | x  | x  | x  | RC     | GEN1_QV      |
| 1  | 1           | x   | 1 | x | x | 0 | x  | x  | x  | RC     | RC           |
| 1  | 1           | x   | x | 0 | 0 | 1 | K2 | K1 | K0 | RC     | GEN1_EXT+PLL |
| 1  | 1           | x   | x | 1 | 0 | 1 | K2 | K1 | K0 | RC     | GEN1_QV+PLL  |
| 1  | 1           | x   | x | x | 1 | 1 | K2 | K1 | K0 | RC     | RC+PLL       |
| 0  | x           | x   | 0 | 0 | x | 0 | x  | x  | x  | RC     | GEN1_EXT     |
| 0  | x           | x   | 0 | 1 | x | 0 | x  | x  | x  | RC     | GEN1_QV      |
| 0  | x           | x   | 1 | x | x | 0 | x  | x  | x  | RC     | RC           |
| 0  | x           | x   | x | 0 | 0 | 1 | K2 | K1 | K0 | RC     | GEN1_EXT+PLL |
| 0  | x           | x   | x | 1 | 0 | 1 | K2 | K1 | K0 | RC     | GEN1_QV+PLL  |
| 0  | x           | x   | x | x | 1 | 1 | K2 | K1 | K0 | RC     | RC+PLL       |

Таблица 7. Определение источника GPIOB&lt;1&gt;/RC\_CLK\_OUT

| TM | TM_CLK_TYPE | RC_OUT_DISABLE | PLL |   |   |   |   |    |    |    | GPIOB<1>/RC_CLK_OUT |
|----|-------------|----------------|-----|---|---|---|---|----|----|----|---------------------|
|    |             |                | 7   | 6 | 5 | 4 | 3 | 2  | 1  | 0  |                     |
| 1  | x           | 0              | 0   | x | x | x | x | x  | x  | x  | RC                  |
| 1  | x           | 0              | 1   | x | x | x | 0 | x  | x  | x  | 1                   |
| 1  | 1           | 0              | 1   | x | 0 | 0 | 1 | K2 | K1 | K0 | GEN1_EXT+PLL        |
| 1  | 1           | 0              | 1   | x | 1 | 0 | 1 | K2 | K1 | K0 | GEN1_QV+PLL         |
| 1  | 0           | 0              | 1   | x | x | 0 | 1 | K2 | K1 | K0 | GEN1_QV+PLL         |
| 1  | x           | 0              | 1   | x | x | 1 | 1 | K2 | K1 | K0 | RC+PLL              |
| 1  | x           | 1              | x   | x | x | x | x | x  | x  | x  | RC                  |
| 0  | x           | x              | x   | x | x | x | x | x  | x  | x  | GPIOB<1>            |

Используемые в таблицах 6, 7 обозначения:

**x** – биты, которые не имеют значения;

**GEN1\_EXT** – внешняя тактовая частота с вывода GEN1 (источник – внешний генератор);

**GEN1\_QV** – тактовая частота с вывода GEN1 (источник – встроенный генератор на основе внешнего кварцевого резонатора). Возможно тактирование от внешнего генератора с размахом цифровых уровней от 0 В до 5,0 В. При использовании внешнего генератора, вывод GEN2 (7) необходимо оставить в обрыве. Частота задается на вывод GEN1 (6) без внешних компонентов.

**GEN1\_EXT+PLL** – внешняя тактовая частота, пропущенная через ФАПЧ;

**GEN1\_QV+PLL** – частота с кварцевого генератора, пропущенная через ФАПЧ;

**RC+PLL** – частота с RC-генератора, пропущенная через ФАПЧ;

**RC** – тактовая частота с выхода встроенного RC-генератора;

**K2, K1, K0** – коэффициент умножения ФАПЧ.

| K2 | K1 | K0 | Коэффициент умножения блока PLL |
|----|----|----|---------------------------------|
| 0  | 0  | 0  | 2                               |
| 0  | 0  | 1  | 4                               |
| 0  | 1  | 0  | 8                               |
| 0  | 1  | 1  | 16                              |
| 1  | 0  | 0  | 32                              |
| 1  | 0  | 1  | 64                              |

При умножении частоты с помощью блока PLL стоит учитывать, что итоговая частота не должна превышать 8 МГц.

При записи в биты RC\_C регистра ANALOG\_O\_RC максимального значения 7Fh, а в биты RC\_R регистра ANALOG\_O\_RC\_R минимального значения 0 – частота RC-генератора будет не более 90 кГц.

При записи в биты RC\_C регистра ANALOG\_O\_RC минимального значения 0, а в биты RC\_R регистра ANALOG\_O\_RC\_R максимального значения 7 – частота RC-генератора будет не менее 400 кГц.

**ANALOG\_O\_RC**

| Бит                | 7      | 6    | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|--------|------|---|---|---|---|---|---|
| Назначение         | Резерв | RC_C |   |   |   |   |   |   |
| Начальное значение | 0      |      |   |   |   |   |   |   |

**RC\_C** – настройка емкости конденсатора RC-генератора (совместно с регистром RC\_R настраивается частота внутреннего RC-генератора, Таблица 8)

**ANALOG\_O\_REF**

| Бит                | 7      | 6 | 5       | 4     | 3 | 2 | 1 | 0 |
|--------------------|--------|---|---------|-------|---|---|---|---|
| Назначение         | Резерв |   | OA_C_EN | V_REF |   |   |   |   |
| Начальное значение | 0      |   |         |       |   |   |   |   |

**OA\_C\_EN** – включение конденсаторов частотной коррекции МОУ:

1 – конденсаторы частотной коррекции включены;

0 – конденсаторы частотной коррекции выключены.

**V\_REF** – настройка выходного напряжения ИОН (вывод A0).

Перед работой с данным регистром рекомендуется ознакомиться с пунктом «Важные замечания при работе с микросхемой».

**ANALOG\_O\_RC\_R**

| Бит                | 7   | 6              | 5               | 4           | 3      | 2    | 1 | 0 |
|--------------------|-----|----------------|-----------------|-------------|--------|------|---|---|
| Назначение         | BOR | RC_OUT_DISABLE | REF_OUT_DISABLE | TM_CLK_TYPE | Резерв | RC_R |   |   |
| Начальное значение | 0   |                |                 |             |        |      |   |   |

**BOR** – супервизор питания:

1 – внешний (вывод BOR\_EXT/PGM при TM = 0);

0 – внутренний.

**RC\_OUT\_DISABLE** – определение источника тактового сигнала на выводе GPIOB<1>/RC\_CLK\_OUT в тестовом режиме (при TM = 1).

**REF\_OUT\_DISABLE** – отключение подачи напряжения с ИОН на вывод A0 в тестовом режиме (при TM = 1):

1 – напряжение не подается на вывод;

0 – напряжение подается на вывод.

**TM\_CLK\_TYPE** – определение источника тактового сигнала в тестовом режиме (при TM = 1, Таблица 6, Таблица 7).

**RC\_R** – настройка сопротивления резистора RC-генератора (совместно с регистром RC\_C настраивается частота внутреннего RC-генератора, Таблица 8)

Перед работой с данным регистром рекомендуется ознакомиться с пунктом «Важные замечания при работе с микросхемой».

Таблица 8. Типовое значение частоты генератора.

| Биты<br>RC_R* | Биты RC_C** |     |          |
|---------------|-------------|-----|----------|
|               | 0000000     | ... | 1111111  |
| 000           | ~330 кГц    | ... | ~35 кГц  |
| 001           | ~500 кГц    | ... | ~65 кГц  |
| 010           | ~600 кГц    | ... | ~95 кГц  |
| 011           | ~680 кГц    | ... | ~120 кГц |
| 100           | ~750 кГц    | ... | ~150 кГц |
| 101           | ~800 кГц    | ... | ~170 кГц |
| 110           | ~850 кГц    | ... | ~190 кГц |
| 111           | ~900 кГц    | ... | ~210 кГц |

\* – Биты RC\_R регистра ANALOG\_O\_RC\_R.

\*\* – Биты RC\_C регистра ANALOG\_O\_RC.

Указанные диапазоны типовые и могут отличаться в конкретной микросхеме и климатических условиях. При необходимости, если требуемая частота не подобрана в текущем диапазоне настройки R, рекомендуется подбирать максимально близкую к требуемой в других диапазонах.

## Тактирование микроконтроллера

Микросхема содержит встроенный RC-генератор. Поддерживается конфигурация, при которой в качестве частоты по умолчанию возможно установить частоту от внутреннего RC-генератора. Данная частота будет использоваться системой при включении питания. Возможность подключения внешней RC-цепи отсутствует.

Выбор источника тактирования осуществляется в конфигурационной памяти (см. пункт Конфигурационная память (ANALOG\_CFG).

При использовании внутреннего RC-генератора вывод GEN1 (6) необходимо объединить с VSSA, вывод GEN2 (7) оставить в обрыве.

При использовании внешнего генератора, вывод GEN2 (7) необходимо оставить в обрыве. Частота задается на вывод GEN1 (6) без внешних компонентов. Максимальная частота тактирования – 8 МГц.

Размах цифровых сигналов генератора от VSSD до VDD\_DR. Если размах цифровых уровней генератора 5,0 В, а напряжение питания портов ввода-вывода отлично от 5,0 В (например, 3,3 В или 2,5 В), то необходимо выбрать источник тактирования «Кварцевый резонатор» (Таблица 6) и частоту подавать на вывод GEN1 (6), вывод GEN2 (7) оставить в обрыве.

Настройка RC-генератора может осуществляться от ~35 кГц до ~900 кГц с расчетной точностью от 0,1...0,3 кГц в нижней области диапазона и до 2,0...20 кГц в верхней области диапазона.

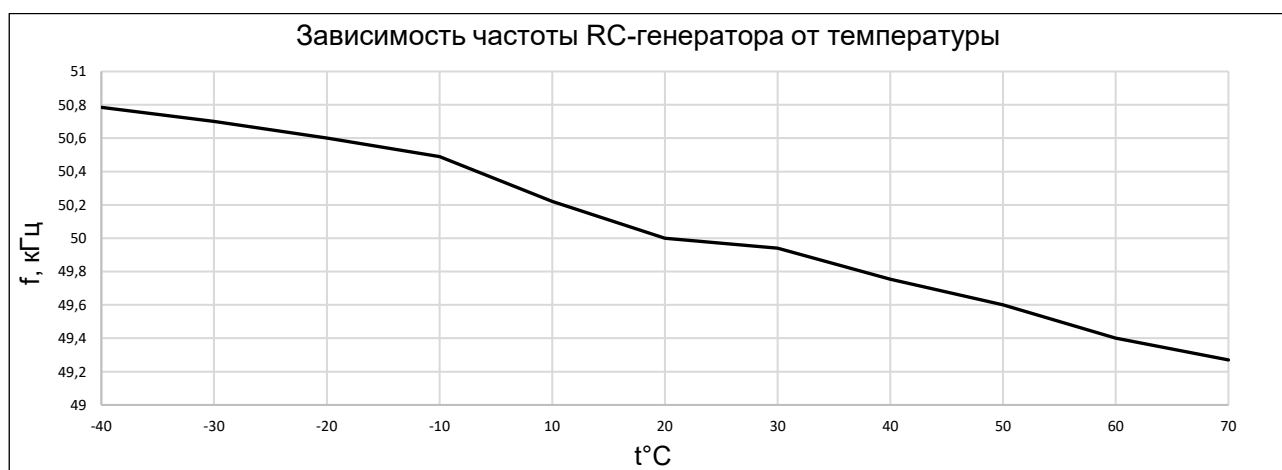


Рисунок 13. График зависимости частоты RC-генератора от температуры

На графике представлена экспериментальная температурная зависимость для генератора, настроенного на 50 кГц. Температурная зависимость при измерениях составила 13,6 Гц/°C, если нормировать на 50 кГц, то 0,027 %/°C.

Сброс микроконтроллера

Процесс сброса микроконтроллера состоит из аппаратного и программного сброса. Влияние помехи по питанию на сброс представлено на рисунке ниже (Рисунок 14).

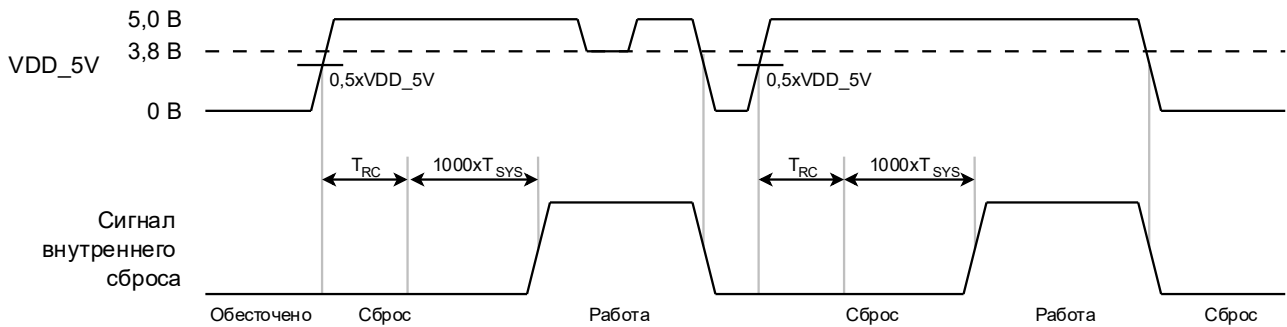


Рисунок 14. Временная диаграмма при включении и выключении питания

Длительность времени заряда RC-цепи рассчитывается по формуле:

$T_{RC} \approx 0,16 \times C5,$

где C5 – емкость внешнего конденсатора (Рисунок 12). Расчетная длительность аналогового сигнала первоначального сброса относительно «резкого» (1,0 мкс) включения питания на выводе VDD\_5V = 5,0 В представлена в таблице ниже. Замедление включения питания будет соответственно затягивать сброс.

| Емкость внешнего конденсатора | Длительность аналогового сигнала первоначального сброса (POR_RST), мс |
|-------------------------------|-----------------------------------------------------------------------|
| для конденсатора C5 = 1,0 нФ  | 0,15 – 0,175                                                          |
| для конденсатора C5 = 10 нФ   | 1,5 – 1,75                                                            |
| для конденсатора C5 = 100 нФ  | 15 – 17,5                                                             |

После срабатывания аналогового сброса добавляется еще цифровая фильтр-задержка (T<sub>SYS</sub>) в течение 1000 периодов частоты, установленной в качестве системной.

Выводы POR\_RST и BOR\_EXT имеют активный уровень «0». Вывод BOR\_EXT работает, только когда вывод ТМ находится в «0».

Программный сброс микросхемы описан в «Руководство пользователя 5400TP105-003 приложение А» в разделе «Программный сброс».

## Настройка GPIO после сброса

Для избежания помех после сброса порты GPIO необходимо настроить в определенное состояние. Порт GPIO\_B всегда предустанавливается в лог. «0». Для GPIO\_A и GPIO\_C начальное состояние зависит от требуемого логического уровня: если порт настроен как выход и должен выдавать лог. «1», значение устанавливается в 0xFF, в противном случае – в 0x00.

```
__sfr __at (0x80) P0 = 0x00; // GPIO_A
__sfr __at (0xA0) P2 = 0x00; // GPIO_B
__sfr __at (0x80) P3 = 0x00; // GPIO_C
```

Рисунок 15. Предустановка GPIO.

При условии выдачи логического нуля на все GPIO.

```
__sfr __at (0x80) P0 = 0xFF; // GPIO_A
__sfr __at (0xA0) P2 = 0x00; // GPIO_B
__sfr __at (0x80) P3 = 0xFF; // GPIO_C
```

Рисунок 16. Предустановка GPIO.

При условии выдачи логической единицы на все GPIO.

На рисунках выше (Рисунок 15, Рисунок 16) показана конфигурация, где порты GPIO\_A и GPIO\_C настроены как выход и выдают лог. «0» (Рисунок 15) и лог. «1» (Рисунок 16). Порт GPIO\_B всегда предустанавливается в 0x00. Начальные значения портов задаются перед выполнением основной части программы.

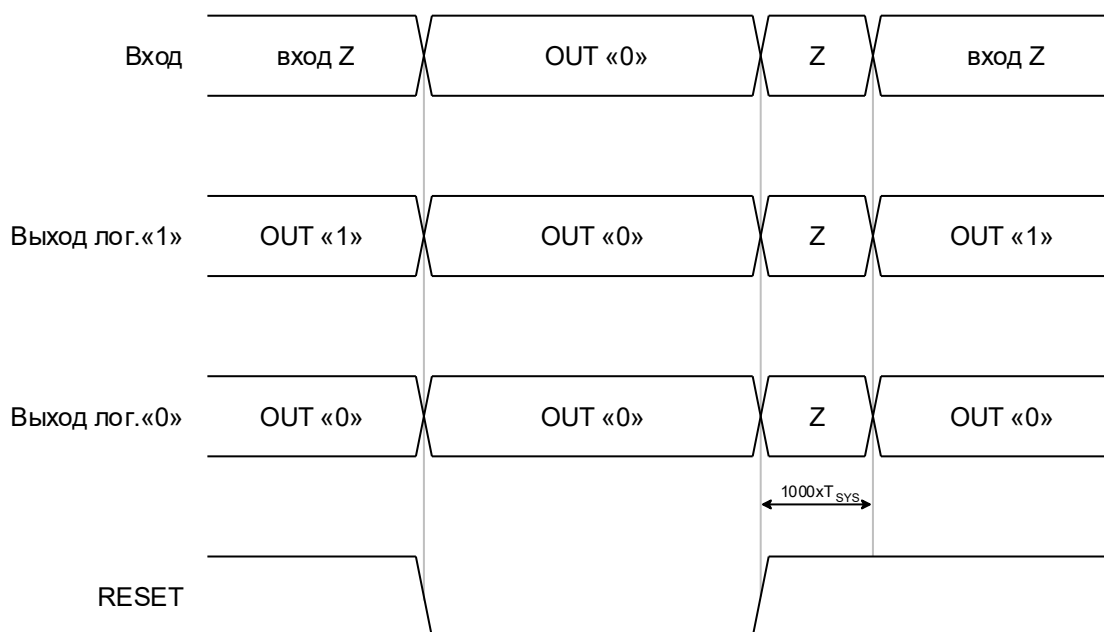


Рисунок 17. Состояние выводов GPIO в режиме входа и выхода при сбросе микросхемы

## Габаритный чертёж

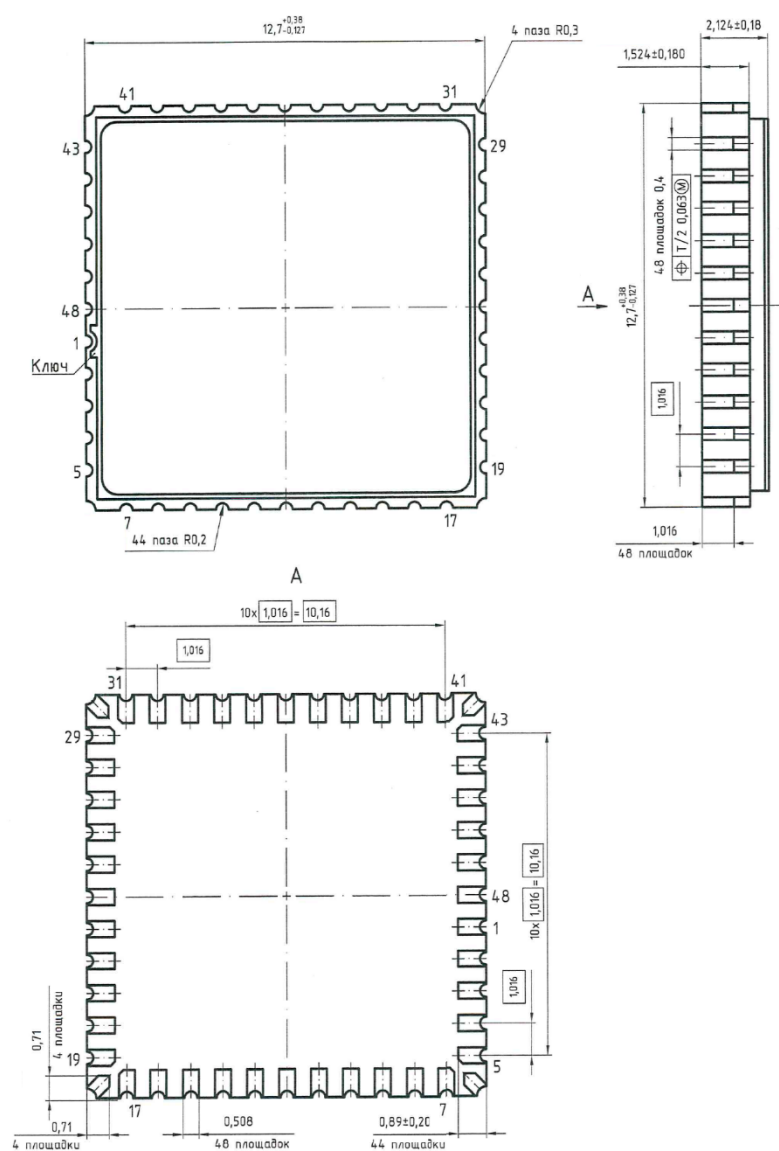


Рисунок 18. Габаритный чертёж корпуса 5142.48-A (размеры в мм)

## Информация для заказа

| Обозначение                                                                            | Маркировка       | Корпус    | Температурный диапазон |
|----------------------------------------------------------------------------------------|------------------|-----------|------------------------|
| Категория качества «ВП»                                                                |                  |           |                        |
| 5400TP105-003 группа А<br>АЕНВ.431260.544ТУ<br>карта заказа КФЦС.431260.012-003Д16     | 5400TP105-003    | 5142.48-A | –60°С...+85°С          |
| 5400TP105-003 группа Б<br>АЕНВ.431260.544ТУ<br>карта заказа КФЦС.431260.012-003Д16     | 5400TP105-003I   | 5142.48-A | –45°С...+85°С          |
| 5400TP105-003 группа В<br>АЕНВ.431260.544ТУ<br>карта заказа КФЦС.431260.012-003Д16     | 5400TP105-003II  | 5142.48-A | –25°С...+85°С          |
| Категория качества «ОТК»                                                               |                  |           |                        |
| К5400TP105-003 группа А<br>КФЦС.431000.001ТУ<br>карта заказа КФЦС 431260.012-003.01Д16 | К5400TP105-003   | 5142.48-A | –60°С...+85°С          |
| К5400TP105-003 группа Б<br>КФЦС.431000.001ТУ<br>карта заказа КФЦС 431260.012-003.01Д16 | К5400TP105-003I  | 5142.48-A | –45°С...+85°С          |
| К5400TP105-003 группа В<br>КФЦС.431000.001ТУ<br>карта заказа КФЦС 431260.012-003.01Д16 | К5400TP105-003II | 5142.48-A | –25°С...+85°С          |

Микросхемы категории качества «ВП» маркируются ромбом.

Микросхемы категории качества «ОТК» маркируются буквой «К».

**Оглавление**

|                                                             |    |
|-------------------------------------------------------------|----|
| Основные особенности .....                                  | 1  |
| Общее описание .....                                        | 1  |
| Вычислительное ядро .....                                   | 2  |
| Встроенные периферийные модули .....                        | 2  |
| Память .....                                                | 2  |
| Функциональное назначение микросхемы .....                  | 2  |
| Электрические параметры микросхемы .....                    | 3  |
| Электростатическая защита .....                             | 4  |
| Предельно-допустимые и предельные режимы эксплуатации ..... | 4  |
| Конфигурация и функциональное описание выводов .....        | 5  |
| Эквивалентные схемы .....                                   | 7  |
| Рекомендуемая схема применения .....                        | 9  |
| Важные замечания при работе с микросхемой .....             | 10 |
| Конфигурационная память (ANALOG_CFG) .....                  | 11 |
| Регистры модуля ANALOG_CFG .....                            | 11 |
| ANALOG_O_BUF .....                                          | 11 |
| ANALOG_O_PLL .....                                          | 12 |
| ANALOG_O_RC .....                                           | 14 |
| ANALOG_O_REF .....                                          | 14 |
| ANALOG_O_RC_R .....                                         | 14 |
| Тактирование микроконтроллера .....                         | 16 |
| Сброс микроконтроллера .....                                | 17 |
| Настройка GPIO после сброса .....                           | 18 |
| Габаритный чертеж .....                                     | 19 |
| Информация для заказа .....                                 | 20 |
| Техническое описание .....                                  | 29 |
| Блок схема .....                                            | 29 |
| Состав изделия .....                                        | 30 |
| Конфигурация выводов, альтернативные функции .....          | 31 |
| Карта памяти .....                                          | 34 |
| Распределение адресного пространства .....                  | 35 |
| Процессор 8051 .....                                        | 36 |
| Общие сведения .....                                        | 36 |
| Поддержка IDE .....                                         | 36 |
| Скорость выполнения команд .....                            | 36 |

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

|                                                                        |    |
|------------------------------------------------------------------------|----|
| Прерывания.....                                                        | 38 |
| Подприоритеты прерываний.....                                          | 38 |
| Регистры управления прерываниями .....                                 | 39 |
| IE .....                                                               | 39 |
| IP .....                                                               | 39 |
| SCON .....                                                             | 39 |
| Ручной вызов прерываний .....                                          | 40 |
| P1 .....                                                               | 40 |
| Таймер-счетчик .....                                                   | 40 |
| Блок управления внешними регистрами .....                              | 41 |
| Обращение к внешним регистрам.....                                     | 41 |
| Типы статусов и прерывания .....                                       | 43 |
| Система управления сбросом и питанием (PMM) .....                      | 44 |
| Структурная схема.....                                                 | 44 |
| Формирование сброса .....                                              | 44 |
| Регистры модуля системы управления сбросом и питанием .....            | 45 |
| PMM_CTRL .....                                                         | 45 |
| Модуль управления источниками тактовых сигналов системы (CMM).....     | 46 |
| Общая информация .....                                                 | 46 |
| Структурная схема.....                                                 | 46 |
| Выбор источника тактирования системы .....                             | 46 |
| Деление частоты тактирования.....                                      | 47 |
| Регистры модуля управления источниками тактовых сигналов системы ..... | 47 |
| CMM_CTRL.....                                                          | 47 |
| CMM_DIV .....                                                          | 47 |
| CMM_MSK .....                                                          | 47 |
| CMM_ST.....                                                            | 48 |
| Сторожевой таймер (WDT) .....                                          | 49 |
| Общая информация .....                                                 | 49 |
| Регистры сторожевого таймера.....                                      | 49 |
| WDT_LOAD0.....                                                         | 49 |
| WDT_LOAD1.....                                                         | 50 |
| WDT_LOAD2.....                                                         | 50 |
| WDT_LOAD3.....                                                         | 50 |
| WDT_VAL0.....                                                          | 50 |
| WDT_VAL1.....                                                          | 50 |

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| WDT_VAL2 .....                                                          | 50 |
| WDT_VAL3 .....                                                          | 50 |
| WDT_CTRL .....                                                          | 51 |
| WDT_CLR .....                                                           | 51 |
| WDT_INTRAW .....                                                        | 51 |
| WDT_INT .....                                                           | 51 |
| WDT_LOCK .....                                                          | 52 |
| WDT_TCR .....                                                           | 52 |
| WDT_TOP .....                                                           | 52 |
| GPIO .....                                                              | 53 |
| Общая информация .....                                                  | 53 |
| Структурная схема .....                                                 | 53 |
| Статусы и прерывания .....                                              | 54 |
| Регистры GPIO .....                                                     | 54 |
| GPIOx_DIR_SET/CLR .....                                                 | 55 |
| GPIOx_ALTF1 .....                                                       | 56 |
| GPIOx_INTEN_SET/CLR .....                                               | 56 |
| GPIOx_INTTYPE_SET/CLR .....                                             | 57 |
| GPIOx_INTPOL_SET/CLR .....                                              | 57 |
| GPIOx_INT .....                                                         | 57 |
| SPI .....                                                               | 58 |
| Общая информация .....                                                  | 58 |
| Структурная схема .....                                                 | 58 |
| Алгоритмы работы .....                                                  | 59 |
| Дуплексный и симплексный режимы .....                                   | 60 |
| Передача данных .....                                                   | 60 |
| Прием данных .....                                                      | 60 |
| Работа с входом выбора микросхемы .....                                 | 61 |
| Особенности работы в режиме «ведомый». Синхронизация и фильтрация ..... | 61 |
| Регистры SPI .....                                                      | 62 |
| SPIx_CTRL .....                                                         | 63 |
| SPIx_CFG0 .....                                                         | 63 |
| SPIx_CFG1 .....                                                         | 64 |
| SPIx_CFG2 .....                                                         | 64 |
| SPIx_CFG3 .....                                                         | 65 |
| SPIx_MSK .....                                                          | 65 |

|                                               |    |
|-----------------------------------------------|----|
| SPIx_ST.....                                  | 65 |
| SPIx_TX0.....                                 | 66 |
| SPIx_TX1.....                                 | 66 |
| SPIx_RX0 .....                                | 66 |
| SPIx_RX1 .....                                | 66 |
| UART .....                                    | 67 |
| Общая информация .....                        | 67 |
| Структурная схема.....                        | 67 |
| Делитель частоты .....                        | 68 |
| Высокоскоростной режим .....                  | 68 |
| Приемник .....                                | 69 |
| Буфер приемника.....                          | 69 |
| Передатчик .....                              | 70 |
| Буфер передатчика .....                       | 70 |
| Прерывания.....                               | 70 |
| Таймер тайм-аута .....                        | 70 |
| Генерация и распознавание сигнала break ..... | 70 |
| Особые режимы работы.....                     | 71 |
| Аппаратный контроль обмена .....              | 72 |
| Регистры UART .....                           | 72 |
| UARTx_CFG0.....                               | 73 |
| UARTx_CFG1.....                               | 74 |
| UARTx_BDR0.....                               | 74 |
| UARTx_BDR1.....                               | 74 |
| UARTx_TXFIFOLVL0.....                         | 75 |
| UARTx_TXFIFOLVL1 .....                        | 75 |
| UARTx_RXFIFOLVL0 .....                        | 75 |
| UARTx_RXFIFOLVL1 .....                        | 75 |
| UARTx_NBMSK .....                             | 75 |
| UARTx_NBADDR0.....                            | 76 |
| UARTx_NBADDR1 .....                           | 76 |
| UARTx_MSK0 .....                              | 76 |
| UARTx_MSK1 .....                              | 76 |
| UARTx_CTRL.....                               | 76 |
| UARTx_TX.....                                 | 77 |
| UARTx_RX0 .....                               | 77 |

|                                                      |    |
|------------------------------------------------------|----|
| UARTx_RX1 .....                                      | 78 |
| UARTx_ST0.....                                       | 78 |
| UARTx_ST1.....                                       | 79 |
| I2C .....                                            | 80 |
| Общая информация .....                               | 80 |
| Структурная схема.....                               | 80 |
| Алгоритмы работы .....                               | 81 |
| Статусы и прерывания .....                           | 83 |
| Регистры I2C .....                                   | 83 |
| I2C_CFG.....                                         | 84 |
| I2C_CTRL.....                                        | 84 |
| I2C_ST0.....                                         | 85 |
| I2C_ST1.....                                         | 85 |
| I2C_ST2.....                                         | 86 |
| I2C_ADDR0 .....                                      | 86 |
| I2C_ADDR1 .....                                      | 86 |
| I2C_PRSC0 .....                                      | 86 |
| I2C_PRSC1 .....                                      | 87 |
| I2C_PRSC2 .....                                      | 87 |
| I2C_PRSC3 .....                                      | 88 |
| I2C_MSK0 .....                                       | 88 |
| I2C_MSK1 .....                                       | 88 |
| I2C_MSK2 .....                                       | 89 |
| I2C_TXFIFO .....                                     | 89 |
| I2C_RXFIFO .....                                     | 89 |
| I2C_TXWORDS.....                                     | 89 |
| I2C_RXTHRESHOLD .....                                | 89 |
| 1-Wire .....                                         | 90 |
| Общая информация .....                               | 90 |
| Структурная схема.....                               | 90 |
| Сигнализация шины 1-Wire.....                        | 91 |
| Предварительная настройка .....                      | 93 |
| Генерация цикла сброса/обнаружения присутствия ..... | 93 |
| Передача байта .....                                 | 93 |
| Прием байта .....                                    | 93 |
| Битовый режим .....                                  | 94 |

|                                                             |     |
|-------------------------------------------------------------|-----|
| Ускоренная передача .....                                   | 94  |
| Режим ускоренного поиска ПЗУ (Search ROM Accelerator) ..... | 94  |
| Пример использования режима ускоренного поиска ПЗУ .....    | 95  |
| Статусы и прерывания .....                                  | 96  |
| Регистры OWI .....                                          | 96  |
| OWI_CFG .....                                               | 96  |
| OWI_BUF .....                                               | 97  |
| OWI_ST .....                                                | 97  |
| OWI_MSK .....                                               | 98  |
| OWI_PRCR .....                                              | 98  |
| OWI_CTRL .....                                              | 99  |
| Рабочий автомат (WORK_FSM) .....                            | 100 |
| Общая информация .....                                      | 100 |
| Особенности работы .....                                    | 100 |
| Регистры «Рабочего автомата» .....                          | 100 |
| FSM_CTRL .....                                              | 100 |
| FSM_PRD2 .....                                              | 101 |
| FSM_PRD1 .....                                              | 101 |
| FSM_PRD0 .....                                              | 101 |
| Таймер .....                                                | 102 |
| Общая информация .....                                      | 102 |
| Структурная схема .....                                     | 102 |
| Работа в режиме «Простой таймер» .....                      | 102 |
| Принцип работы .....                                        | 102 |
| Статусы и прерывания .....                                  | 103 |
| Алгоритм работы .....                                       | 103 |
| Работа в режиме «Таймер с внешней остановкой» .....         | 103 |
| Принцип работы .....                                        | 103 |
| Статусы и прерывания .....                                  | 103 |
| Алгоритм работы .....                                       | 103 |
| Работа в режиме «Межсобытийный таймер» .....                | 104 |
| Принцип работы .....                                        | 104 |
| Статусы и прерывания .....                                  | 104 |
| Алгоритм работы .....                                       | 104 |
| Работа в режиме «Таймер-счетчик» .....                      | 104 |
| Принцип работы .....                                        | 104 |

|                                         |     |
|-----------------------------------------|-----|
| Статусы и прерывания .....              | 105 |
| Алгоритм работы .....                   | 105 |
| Регистры таймеров .....                 | 106 |
| TMRx_CTRL .....                         | 106 |
| TMRx_CFG .....                          | 107 |
| TMRx_PRDH .....                         | 107 |
| TMRx_PRDL .....                         | 108 |
| TMRx_VALH .....                         | 108 |
| TMRx_VALM .....                         | 108 |
| TMRx_VALL .....                         | 108 |
| TMRx_MSK .....                          | 108 |
| TMRx_ST .....                           | 109 |
| АЦП .....                               | 110 |
| Общая информация .....                  | 110 |
| Одиночный режим .....                   | 110 |
| Автоматический режим .....              | 110 |
| Алгоритмы работы .....                  | 111 |
| Регистры модуля «АЦП» .....             | 112 |
| ADC_CTRL .....                          | 112 |
| ADC_CFG .....                           | 113 |
| ADC_ST .....                            | 113 |
| ADC_MSK .....                           | 113 |
| ADC_RES0 .....                          | 114 |
| ADC_RES1 .....                          | 114 |
| ЦАП .....                               | 115 |
| Общая информация .....                  | 115 |
| Алгоритм работы .....                   | 115 |
| Регистры модуля «ЦАП» .....             | 115 |
| DAC_CTRL .....                          | 115 |
| DAC_VALUE0 .....                        | 115 |
| DAC_VALUE1 .....                        | 116 |
| Контроллер прерываний (INT_CTRL) .....  | 117 |
| Регистры «Контроллера прерываний» ..... | 117 |
| INT_FIX_CLR0 .....                      | 117 |
| INT_FIX_CLR1 .....                      | 117 |
| INT_FIX_CLR2 .....                      | 118 |

|                                                                    |     |
|--------------------------------------------------------------------|-----|
| INT_FIX_CLR3 .....                                                 | 118 |
| Режим «SLEEP» .....                                                | 119 |
| Общая информация .....                                             | 119 |
| Сон процессора .....                                               | 119 |
| Глубокий сон .....                                                 | 119 |
| JTAG .....                                                         | 120 |
| Общее описание .....                                               | 120 |
| Структурная схема .....                                            | 120 |
| Инструкции «TEST_MODE», «EXTEST», «BYPASS», «SAMPLE/PRELOAD» ..... | 121 |
| Инструкция «JTAG_RAM_ACCESS» .....                                 | 121 |
| Инструкция «JTAG_ROM_ACCESS» .....                                 | 122 |
| Инструкция «IF_ACCESS» .....                                       | 123 |
| Инструкция «DEBUGGER» .....                                        | 123 |
| Лист регистрации изменений .....                                   | 124 |

## Техническое описание

## Блок схема

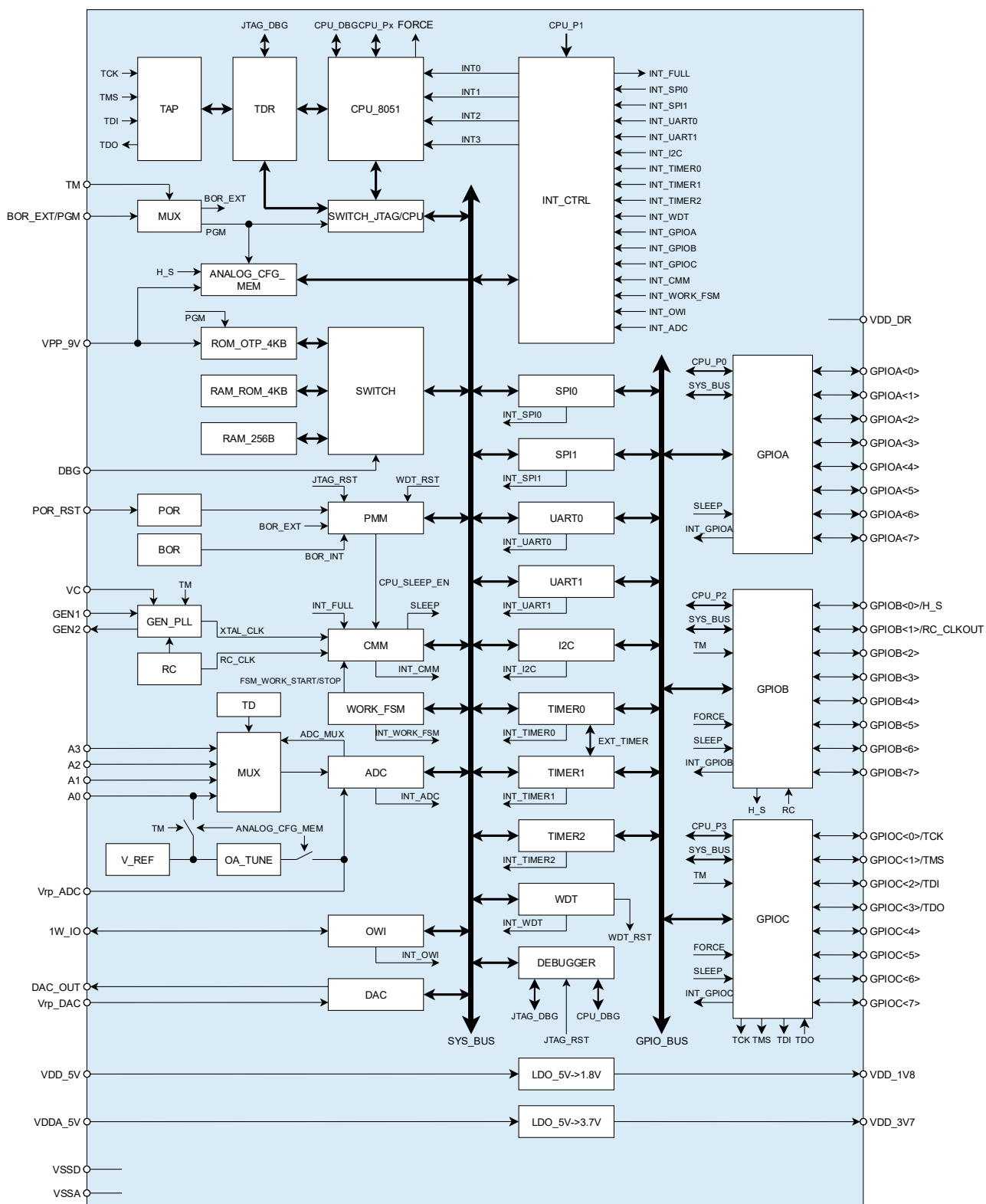


Рисунок 19. Блок схема 5400TP105-003

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

**Состав изделия**

Системные устройства:

- CPU\_8051 – процессорное ядро;
- TAP (Test Access Port) – автомат JTAG;
- TDR (Test Data Register) – регистры данных JTAG;
- MUX – мультиплексор;
- ANALOG\_CFG\_MEM – конфигурационная память;
- ROM\_OTP\_4KB (One-Time Programmable) – однократно программируемое постоянное запоминающее устройство емкостью 4 КБ, память программ устройства;
- RAM\_ROM\_4KB – оперативное запоминающее устройство емкостью 4 КБ, микроконтроллер работает с данным модулем, как с памятью программ или памятью данных в зависимости от значения вывода DBG;
- RAM\_256B – оперативное запоминающее устройство емкостью 256 В, для хранения промежуточных данных, память данных устройства;
- SWITCH – модуль, разграничивающий доступ к RAM\_OTP\_4KB, RAM\_ROM\_4KB, RAM\_256B;
- SWITCH JTAG/CPU – модуль, разграничивающий доступ к шине для JTAG и CPU;
- POR (Power-On Reset) – модуль сброса при включении;
- BOR (Brown-Out Reset) – модуль сброса при снижении напряжения питания;
- PMM (Power Management Module) – модуль системы управления сбросом и питанием;
- GEN\_PLL – модуль ввода-вывода и фазовой автоподстройки частоты (ФАПЧ) тактирования;
- RC – RC-генератор;
- CMM (Clock Management Module) – модуль управления источниками тактовых сигналов системы;
- DEBUGGER – отладчик;
- OA\_TUNE – масштабирующий операционный усилитель (МОУ);
- V\_REF – источник опорного напряжения (ИОН);
- LDO\_5V->3V7, LDO\_5V->1V8 – линейные регуляторы напряжения.

Периферийные устройства:

- INT\_CTRL – контроллер прерываний;
- SPI0, SPI1 – модули SPI;
- UART0, UART1 – модули UART;
- I2C – модуль I2C;
- GPIOA, GPIOB, GPIOC – контроллеры портов ввода-вывода;
- TIMER0, TIMER1, TIMER2 – универсальные таймеры;
- WDT – сторожевой таймер;
- WORK\_FSM – рабочий автомат для режима «SLEEP»;
- OWI – модуль 1-Wire;
- ADC – аналого-цифровой преобразователь;
- TD – термодатчик;
- DAC – цифро-аналоговый преобразователь.

## Конфигурация выводов, альтернативные функции

Таблица 9. Альтернативные функции выводов GPIO

| № вывода | Наименование вывода    | Альтернативная функция |           |              | Пояснение                                                                                                                                                                                                                            |
|----------|------------------------|------------------------|-----------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          |                        | АФ0                    | АФ1       | АФ2          |                                                                                                                                                                                                                                      |
| Порт А   |                        |                        |           |              |                                                                                                                                                                                                                                      |
| 20       | GPIOA<0>               | SPI0_MOSI              | SPI1_MOSI | I_TIMER0_EXT | Порт ввода-вывода микроконтроллера, разряд №0 группы А:<br>• SPI0/1 – MOSI (направление определяется режимом работы «ведущий»/«ведомый»);<br>• TIMER0 – I_TIMER0_EXT (вход).                                                         |
| 21       | GPIOA<1>               | SPI0_MISO              | SPI1_MISO | I_TIMER1_EXT | Порт ввода-вывода микроконтроллера, разряд №1 группы А:<br>• SPI0/1 – MISO (направление определяется режимом работы «ведущий»/«ведомый»);<br>• TIMER1 – I_TIMER1_EXT (вход).                                                         |
| 22       | GPIOA<2>               | SPI0_SCK               | SPI1_SCK  | O_SLEEP      | Порт ввода-вывода микроконтроллера, разряд №2 группы А:<br>• SPI0/1 – SCK (направление определяется режимом работы «ведущий»/«ведомый»);<br>• режим «Глубокий сон» – O_SLEEP (выход).                                                |
| 23       | GPIOA<3>               | SPI0_I_CS              | SPI1_I_CS | SPI0_O_CS    | Порт ввода-вывода микроконтроллера, разряд №3 группы А:<br>• SPI0/1 – I_CS (вход);<br>• SPI0 – O_CS (выход).                                                                                                                         |
| 24       | GPIOA<4>               | UART0_TX               | UART1_TX  | SPI1_O_CS    | Порт ввода-вывода микроконтроллера, разряд №4 группы А:<br>• UART0/1 – TX (выход);<br>• SPI1 – O_CS (выход).                                                                                                                         |
| 25       | GPIOA<5>               | UART0_RX               | UART1_RX  | «0»          | Порт ввода-вывода микроконтроллера, разряд №5 группы А:<br>• UART0/1 – RX (вход);<br>• лог. «0» (выход).                                                                                                                             |
| 26       | GPIOA<6>               | I2C_SCL                | UART0_CTS | UART1_CTS    | Порт ввода-вывода микроконтроллера, разряд №6 группы А:<br>• I2C – SCL (направление определяется режимом работы «ведущий»/«ведомый»);<br>• UART0/1 – CTS (вход).                                                                     |
| 27       | GPIOA<7>               | I2C_SDA                | UART0_RTS | UART1_RTS    | Порт ввода-вывода микроконтроллера, разряд №7 группы А:<br>• I2C – SDA (направление определяется протоколом);<br>• UART0/1 – RTS (выход).                                                                                            |
| Порт В   |                        |                        |           |              |                                                                                                                                                                                                                                      |
| 44       | GPIOB<0>/<br>H_S       | SPI0_MOSI              | SPI1_MOSI | I_TIMER0_EXT | Порт ввода-вывода микроконтроллера, разряд №0 группы В;<br>при TM = 1 вывод принудительно работает как H/S (вход); во время обращения к внешним регистрам этот вывод работает как DATA<0> (направление определяется командой)        |
| 45       | GPIOB<1>/<br>RC_CLKOUT | SPI0_MISO              | SPI1_MISO | I_TIMER2_EXT | Порт ввода-вывода микроконтроллера, разряд №1 группы В;<br>при TM = 1 вывод принудительно работает как RC_CLKOUT (выход); во время обращения к внешним регистрам этот вывод работает как DATA<1> (направление определяется командой) |
| 46       | GPIOB<2>               | SPI0_SCK               | SPI1_SCK  | O_SLEEP      | Порт ввода-вывода микроконтроллера, разряд №2 группы В;<br>во время обращения к внешним регистрам этот вывод работает как DATA<2> (направление определяется командой)                                                                |

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

| № вывода      | Наименование вывода | Альтернативная функция |           |              | Пояснение                                                                                                                                                                                                      |
|---------------|---------------------|------------------------|-----------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               |                     | АФ0                    | АФ1       | АФ2          |                                                                                                                                                                                                                |
| 47            | GPIOB<3>            | SPI0_I_CS              | SPI1_I_CS | SPI0_O_CS    | Порт ввода-вывода микроконтроллера, разряд №3 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<3> (направление определяется командой)                                             |
| 48            | GPIOB<4>            | UART0_TX               | UART1_TX  | SPI1_O_CS    | Порт ввода-вывода микроконтроллера, разряд №4 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<4> (направление определяется командой)                                             |
| 1             | GPIOB<5>            | UART0_RX               | UART1_RX  | «0»          | Порт ввода-вывода микроконтроллера, разряд №5 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<5> (направление определяется командой)                                             |
| 2             | GPIOB<6>            | I2C_SCL                | UART0_CTS | UART1_CTS    | Порт ввода-вывода микроконтроллера, разряд №6 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<6> (направление определяется командой)                                             |
| 3             | GPIOB<7>            | I2C_SDA                | UART0_RTS | UART1_RTS    | Порт ввода-вывода микроконтроллера, разряд №7 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<7> (направление определяется командой)                                             |
| <b>Порт С</b> |                     |                        |           |              |                                                                                                                                                                                                                |
| 9             | GPIOC<0>/TCK        | SPI0_MOSI              | SPI1_MOSI | I_TIMER1_EXT | Порт ввода-вывода микроконтроллера, разряд №0 группы С; при ТМ = 1 вывод принудительно работает как TCK (вход) интерфейса JTAG; во время обращения к внешним регистрам этот вывод работает как WR/RD (выход)   |
| 10            | GPIOC<1>/TMS        | SPI0_MISO              | SPI1_MISO | I_TIMER2_EXT | Порт ввода-вывода микроконтроллера, разряд №1 группы С; при ТМ = 1 вывод принудительно работает как TMS (вход) интерфейса JTAG; во время обращения к внешним регистрам этот вывод работает как EN (выход)      |
| 11            | GPIOC<2>/TDI        | SPI0_SCK               | SPI1_SCK  | O_SLEEP      | Порт ввода-вывода микроконтроллера, разряд №2 группы С; при ТМ = 1 вывод принудительно работает как TDI (вход) интерфейса JTAG; во время обращения к внешним регистрам этот вывод работает как SEL<0> (выход)  |
| 12            | GPIOC<3>/TDO        | SPI0_I_CS              | SPI1_I_CS | SPI0_O_CS    | Порт ввода-вывода микроконтроллера, разряд №3 группы С; при ТМ = 1 вывод принудительно работает как TDO (выход) интерфейса JTAG; во время обращения к внешним регистрам этот вывод работает как SEL<1> (выход) |
| 13            | GPIOC<4>            | UART0_TX               | UART1_TX  | SPI1_O_CS    | Порт ввода-вывода микроконтроллера, разряд №4 группы С                                                                                                                                                         |
| 14            | GPIOC<5>            | UART0_RX               | UART1_RX  | «0»          | Порт ввода-вывода микроконтроллера, разряд №5 группы С                                                                                                                                                         |
| 15            | GPIOC<6>            | I2C_SCL                | UART0_CTS | UART1_CTS    | Порт ввода-вывода микроконтроллера, разряд №6 группы С; во время обращения к внешним регистрам этот вывод работает как SEL<2> (выход)                                                                          |

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

| № вывода | Наименование вывода | Альтернативная функция |           |           | Пояснение                                                                                                                                |
|----------|---------------------|------------------------|-----------|-----------|------------------------------------------------------------------------------------------------------------------------------------------|
|          |                     | АФ0                    | АФ1       | АФ2       |                                                                                                                                          |
| 16       | GPIOC<7>            | I2C_SDA                | UART0_RTS | UART1_RTS | Порт ввода-вывода микроконтроллера, разряд №7 группы C;<br>во время обращения к внешним регистрам этот вывод работает как SEL<3> (выход) |

Порты JTAG (TCK, TDI, TMS, TDO) мультиплексированы с портами GPIOC. Выбор назначения выводов (GPIOC или JTAG) осуществляется выводом TM.

Порты интерфейсов (SPI, UART и т.д.) также мультиплексированы с портами GPIO. Выбор назначения выводов осуществляется с помощью альтернативных функций во время работы микроконтроллера.

## Карта памяти

Адресное пространство памяти программ и данных разделено. В данное адресное пространство отображаются различные модули памяти и периферии.

Таблица 10. Адресное пространство памяти программ и данных

| №     | Выделенное пространство | Зона | Аббревиатура | Описание                                  |
|-------|-------------------------|------|--------------|-------------------------------------------|
| 0000h | 4 КБ                    | ПЗУ  | ROM          | ПЗУ                                       |
| 0000h | 4352 Б                  | ОЗУ  | RAM          | ОЗУ                                       |
| 2000h | 256 Б                   |      | PMM          | Подсистема управления электропитанием     |
| 2100h | 256 Б                   |      | CMM          | Подсистема управления тактированием       |
| 2200h | 256 Б                   |      | WDT          | Сторожевой таймер                         |
| 2300h | 256 Б                   |      | GPIOA        | Универсальные порты ввода-вывода          |
| 2400h | 256 Б                   |      | GPIOB        |                                           |
| 2500h | 256 Б                   |      | GPIOC        |                                           |
| 2600h | 256 Б                   |      | SPI0         | Интерфейсы SPI                            |
| 2700h | 256 Б                   |      | SPI1         |                                           |
| 2800h | 256 Б                   |      | UART0        | Интерфейсы UART                           |
| 2900h | 256 Б                   |      | UART1        |                                           |
| 2A00h | 256 Б                   |      | I2C          | Интерфейс I2C                             |
| 2B00h | 256 Б                   |      | OWI          | Интерфейс 1-WIRE                          |
| 2C00h | 256 Б                   |      | WORK_FSM     | Рабочий автомат                           |
| 2D00h | 32 Б                    |      | TIMER0       | Таймеры                                   |
| 2D20h | 32 Б                    |      | TIMER1       |                                           |
| 2D40h | 32 Б                    |      | TIMER2       |                                           |
| 2F00h | 256 Б                   |      | ADC          | Аналого-цифровой преобразователь          |
| 3000h | 256 Б                   |      | DAC          | Цифро-аналоговый преобразователь          |
| 3100h | 256 Б                   |      | INT_CTRL     | Контроллер прерываний                     |
| 3200h | 64 Б                    |      | DEBUGGER     | Отладчик                                  |
| 3300h | 8 Б                     |      | ANALOG_CFG   | Конфигурационная память аналоговых блоков |

Если CPU обратится по адресу вне выделенного пространства, то при чтении будут приняты нулевые данные, запись не будет иметь никакого эффекта.

## Распределение адресного пространства

Микроконтроллеры 8051 оперируют двумя типами памяти: памятью программ и памятью данных. Память программ и память данных физически и логически разделены, имеют различные механизмы адресации, работают под управлением различных сигналов и выполняют разные функции.

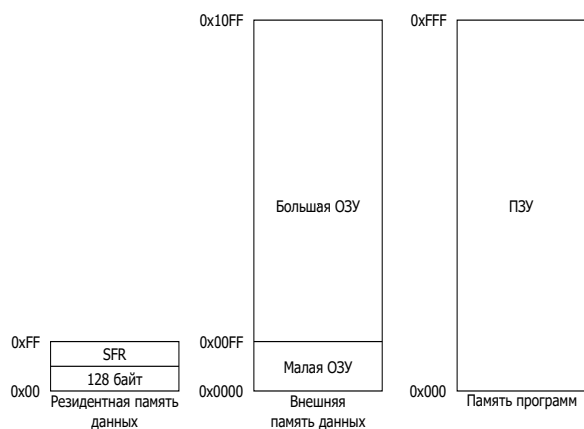


Рисунок 20. Структура памяти микроконтроллера в режиме «HARD»

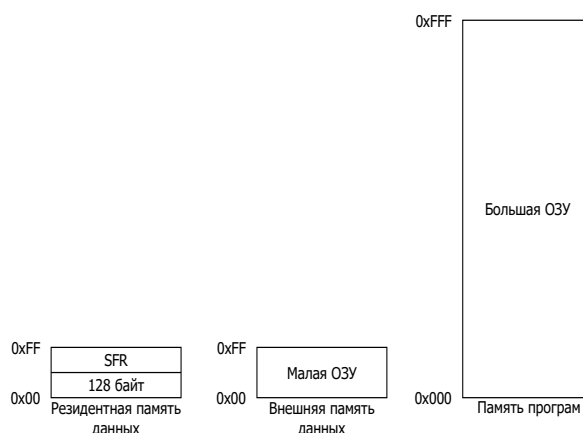


Рисунок 21. Структура памяти микроконтроллера в режиме «SOFT»

Обращение к внешней памяти данных осуществляется по команде «movx», а обращение к памяти программ по команде «movc».

Структура резидентной памяти данных представлена на рисунке 8. Первые 32 байта сгруппированы в 4 банка по 8 регистров R0 – R7. Команды микроконтроллера могут обращаться к ним как к регистрам общего назначения R0 – R7 и как к ячейкам ОЗУ с прямой адресацией от 00h до 1Fh. Ячейки резидентного ОЗУ с адреса 20h до адреса 2Fh обеспечивают кроме прямой адресации, адресацию 128 битов (0-127).

| Адрес байта    |                       | Адреса битов |    |    |    |    |    |    |                                    | Область памяти с битовой адресаций | Адрес байта                      |    | Адреса битов |    |    |    |    |    |                                    |  | Область памяти с битовой адресаций |
|----------------|-----------------------|--------------|----|----|----|----|----|----|------------------------------------|------------------------------------|----------------------------------|----|--------------|----|----|----|----|----|------------------------------------|--|------------------------------------|
|                |                       |              |    |    |    |    |    |    |                                    |                                    |                                  |    |              |    |    |    |    |    |                                    |  |                                    |
| 27             | 3F                    | 3E           | 3D | 3C | 3B | 3A | 39 | 38 | Область памяти с битовой адресаций | 7F                                 | Область памяти общего назначения |    |              |    |    |    |    |    |                                    |  |                                    |
| 26             | 37                    | 36           | 35 | 34 | 33 | 32 | 31 | 30 |                                    |                                    |                                  |    |              |    |    |    |    |    |                                    |  |                                    |
| 25             | 2F                    | 2E           | 2D | 2C | 2B | 2A | 29 | 28 |                                    |                                    |                                  |    |              |    |    |    |    |    |                                    |  |                                    |
| 24             | 27                    | 26           | 25 | 24 | 23 | 22 | 21 | 20 |                                    |                                    |                                  |    |              |    |    |    |    |    |                                    |  |                                    |
| 23             | 1F                    | 1E           | 1D | 1C | 1B | 1A | 19 | 18 |                                    |                                    |                                  |    |              |    |    |    |    |    |                                    |  |                                    |
| 22             | 17                    | 16           | 15 | 14 | 13 | 12 | 11 | 10 |                                    |                                    |                                  |    |              |    |    |    |    |    |                                    |  |                                    |
| 21             | 0F                    | 0E           | 0D | 0C | 0B | 0A | 09 | 08 |                                    |                                    |                                  |    |              |    |    |    |    |    |                                    |  |                                    |
| 20             | 07                    | 06           | 05 | 04 | 03 | 02 | 01 | 00 |                                    |                                    |                                  |    |              |    |    |    |    |    |                                    |  |                                    |
| 1F             | Банк 3                |              |    |    |    |    |    |    | 30                                 | 2F                                 | 7F                               | 7E | 7D           | 7C | 7B | 7A | 79 | 78 | Область памяти с битовой адресаций |  |                                    |
| 18             | Банк 2                |              |    |    |    |    |    |    | 2E                                 | 77                                 | 76                               | 75 | 74           | 73 | 72 | 71 | 70 |    |                                    |  |                                    |
| 17             |                       |              |    |    |    |    |    |    | 2D                                 | 6F                                 | 6E                               | 6D | 6C           | 6B | 6A | 69 | 68 |    |                                    |  |                                    |
| 10             |                       |              |    |    |    |    |    |    | 2C                                 | 67                                 | 66                               | 65 | 64           | 63 | 62 | 61 | 60 |    |                                    |  |                                    |
| 0F             | Банк 1                |              |    |    |    |    |    |    | 2B                                 | 5F                                 | 5E                               | 5D | 5C           | 5B | 5A | 59 | 58 |    |                                    |  |                                    |
| 08             | Банк 0 (по умолчанию) |              |    |    |    |    |    |    | 2A                                 | 57                                 | 56                               | 55 | 54           | 53 | 52 | 51 | 50 |    |                                    |  |                                    |
| 07             |                       |              |    |    |    |    |    |    | 29                                 | 4F                                 | 4E                               | 4D | 4C           | 4B | 4A | 49 | 48 |    |                                    |  |                                    |
| 00             |                       |              |    |    |    |    |    |    | 28                                 | 47                                 | 46                               | 45 | 44           | 43 | 42 | 41 | 40 |    |                                    |  |                                    |
| Регистры R0-R7 |                       |              |    |    |    |    |    |    |                                    |                                    |                                  |    |              |    |    |    |    |    |                                    |  |                                    |

Рисунок 22. Структура резидентной памяти данных

Резидентная память данных имеет 8-битную шину адреса, через которую обеспечивается доступ к памяти из регистра адреса – RAR или из регистра-указателя стека – SP. Регистр-указатель стека используется микроконтроллером при работе со стеком. Через регистр адреса осуществляется прямая и косвенная адресация памяти данных.

## Процессор 8051

### Общие сведения

Текущая версия процессора 8051 соответствует стандартному ядру, за исключением некоторых особенностей, которые будут описаны в данной главе.

### Поддержка IDE

IDE – Keil. Потактовая симуляционная модель – Atmel AT89LP52.

### Скорость выполнения команд

Машинный цикл равен 1 такту. Уход на обработчика прерывания занимает 5 машинных циклов.

Таблица 11. Скорость выполнения команд

| Команда              | Кол-во машинных циклов | Команда              | Кол-во машинных циклов |
|----------------------|------------------------|----------------------|------------------------|
| ACALL                | 3                      | MOV_ATRI_DIRECT      | 2                      |
| ADD_A_ATRI           | 2                      | MOV_ATRI_IMMEDIATE   | 2                      |
| ADD_A_DIRECT         | 2                      | MOV_BIT_C            | 2                      |
| ADD_A_IMMEDIATE      | 2                      | MOV_C_BIT            | 2                      |
| ADD_A_RN             | 1                      | MOV_DIRECT_A         | 2                      |
| ADDC_A_ATRI          | 2                      | MOV_DIRECT_ATRI      | 2                      |
| ADDC_A_DIRECT        | 2                      | MOV_DIRECT_DIRECT    | 3                      |
| ADDC_A_IMMEDIATE     | 2                      | MOV_DIRECT_IMMEDIATE | 3                      |
| ADDC_A_RN            | 1                      | MOV_DIRECT_RN        | 2                      |
| AJMP                 | 3                      | MOV_DPTR_IMMEDIATE:  | 3                      |
| ANL_A_ATRI           | 2                      | MOV_RN_A             | 1                      |
| ANL_A_DIRECT         | 2                      | MOV_RN_DIRECT        | 2                      |
| ANL_A_IMMEDIATE      | 2                      | MOV_RN_IMMEDIATE     | 2                      |
| ANL_A_RN             | 1                      | MOVC_A_ATDPTR        | 3                      |
| ANL_C_BIT            | 2                      | MOVC_A_ATPC          | 3                      |
| ANL_C_NBIT           | 2                      | MOVC_A_ATDPTR        | 3                      |
| ANL_DIRECT_A         | 2                      | MOVC_A_ATRI          | 3                      |
| ANL_DIRECT_IMMEDIATE | 3                      | MOVC_ATDPTR_A        | 3                      |
| CJNE_A_DIRECT        | 4                      | MOVC_ATRI_A          | 3                      |
| CJNE_A_IMMEDIATE     | 4                      | MUL_AB               | 2                      |
| CJNE_ATRI_IMMEDIATE  | 4                      | NOP                  | 1                      |
| CJNE_RN_IMMEDIATE    | 4                      | ORL_A_ATRI           | 2                      |
| CLR_A                | 1                      | ORL_A_DIRECT         | 2                      |
| CLR_BIT              | 2                      | ORL_A_IMMEDIATE      | 2                      |
| CLR_C                | 1                      | ORL_A_RN             | 1                      |
| CPL_A                | 1                      | ORL_C_BIT            | 2                      |
| CPL_BIT              | 2                      | ORL_C_NBIT           | 2                      |
| CPL_C                | 1                      | ORL_DIRECT_A         | 2                      |
| DA_A                 | 1                      | ORL_DIRECT_IMMEDIATE | 3                      |

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

| Команда         | Кол-во машинных циклов | Команда              | Кол-во машинных циклов |
|-----------------|------------------------|----------------------|------------------------|
| DEC_A           | 1                      | POP                  | 2                      |
| DEC_ATRI        | 2                      | PUSH                 | 2                      |
| DEC_DIRECT      | 2                      | RET                  | 4                      |
| DEC_RN          | 1                      | RETI                 | 4                      |
| DIV_AB          | 4                      | RL_A                 | 2                      |
| DJNZ_DIRECT     | 4                      | RLC_A                | 1                      |
| DJNZ_RN         | 3                      | RR_A                 | 1                      |
| INC_A           | 1                      | RRC_A                | 1                      |
| INC_ATRI        | 2                      | SETB_BIT             | 2                      |
| INC_DIRECT      | 2                      | SETB_C               | 1                      |
| INC_DPTR        | 2                      | SJMP                 | 3                      |
| INC_RN          | 1                      | SUBB_A_ATRI          | 2                      |
| JB              | 4                      | SUBB_A_DIRECT        | 2                      |
| JBC             | 4                      | SUBB_A_IMMEDIATE     | 2                      |
| JC              | 3                      | SUBB_A_RN            | 1                      |
| JMP_A_DPTR      | 2                      | SWAP_A               | 1                      |
| JNB             | 4                      | XCH_A_ATRI:          | 2                      |
| JNC             | 3                      | XCH_A_DIRECT         | 2                      |
| JNZ             | 3                      | XCH_A_RN             | 1                      |
| JZ              | 3                      | XCHD_A_ATRI          | 2                      |
| LCALL           | 4                      | XRL_A_ATRI           | 2                      |
| LJMP            | 4                      | XRL_A_DIRECT         | 2                      |
| MOV_A_ATRI      | 2                      | XRL_A_IMMEDIATE      | 2                      |
| MOV_A_DIRECT    | 2                      | XRL_A_RN             | 1                      |
| MOV_A_IMMEDIATE | 3                      | XRL_DIRECT_A         | 2                      |
| MOV_A_RN        | 1                      | XRL_DIRECT_IMMEDIATE | 3                      |
| MOV_ATRI_A      | 1                      |                      |                        |

**Прерывания**

В текущей версии процессора 8051 предусмотрено 6 векторов прерываний.

Таблица 12. Описание прерываний и адрес перехода

| Номер прерывания | Описание                     | Адрес перехода | Внутреннее название прерываний |
|------------------|------------------------------|----------------|--------------------------------|
| 0                | WDT                          | 0003h          | IE0                            |
|                  | WORK_FSM                     |                |                                |
|                  | CMM                          |                |                                |
| 1                | Внутренний таймер/счетчик T0 | 000Bh          | TF0                            |
| 2                | I2C                          | 0013h          | IE1                            |
|                  | UART0                        |                |                                |
|                  | SPI0                         |                |                                |
|                  | TIMER0                       |                |                                |
|                  | GPIOA                        |                |                                |
| 3                | Внутренний таймер/счетчик T1 | 001Bh          | TF1                            |
| 4                | OWI                          | 0023h          | IS0                            |
|                  | UART1                        |                |                                |
|                  | SPI1                         |                |                                |
|                  | TIMER1                       |                |                                |
|                  | GPIOB                        |                |                                |
| 5                | GPIOC                        | 002Bh          | IS1                            |
|                  | TIMER2                       |                |                                |

Прерывания IS0 и IS1 всегда работают по уровню.

**Подприоритеты прерываний**

При одинаковом приоритете обработка прерываний выполняется в следующей последовательности

Таблица 13. Описание подприоритетов прерываний

| Подприоритет | Название прерывания |
|--------------|---------------------|
| 1 – высший   | IE0                 |
| 2            | TF0                 |
| 3            | IS0                 |
| 4            | IE1                 |
| 5            | TF1                 |
| 6 – низший   | IS1                 |

## Регистры управления прерываниями

При добавлении новых прерываний, в регистре приоритетов прерываний и регистре разрешения прерываний, были внесены изменения относительно стандартного ядра 8051.

### IE

Измененный регистр разрешения прерываний представлен ниже.

| Бит                | 7  | 6      | 5   | 4   | 3   | 2   | 1   | 0   |
|--------------------|----|--------|-----|-----|-----|-----|-----|-----|
| Назначение         | EA | Резерв | ES1 | ES0 | ET1 | EX1 | ET0 | EX0 |
| Начальное значение | 0  |        |     |     |     |     |     |     |

**EA** – включение системы прерываний.

Изменение заключается в добавлении битов ES0, ES1, отвечающих за разрешение прерываний IS0, IS1 (биты ET0, ET1, EX0, EX1 отвечают за разрешение прерываний TF0, TF1, IE0, IE1 соответственно).

### IP

Измененный регистр приоритетов прерываний представлен ниже.

| Бит                | 7      | 6 | 5   | 4   | 3   | 2   | 1   | 0   |
|--------------------|--------|---|-----|-----|-----|-----|-----|-----|
| Назначение         | Резерв |   | PS1 | PS0 | PT1 | PX1 | PT0 | PX0 |
| Начальное значение | 0      |   |     |     |     |     |     |     |

Изменение заключается в добавлении битов PS0, PS1, отвечающих за установку приоритетов прерываний IS0, IS1. (биты PT0, PT1, PX0, PX1 отвечают за установку приоритетов прерываний TF0, TF1, IE0, IE1 соответственно).

### SCON

Измененный регистр SCON представлен ниже.

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1   | 0   |
|--------------------|--------|---|---|---|---|---|-----|-----|
| Назначение         | Резерв |   |   |   |   |   | IS1 | IS0 |
| Начальное значение | 0      |   |   |   |   |   |     |     |

Изменение заключается в удалении битов, отвечающих за управление работой последовательного порта и добавление флагов прихода прерываний IS0, IS1.

### Ручной вызов прерываний

Пользователь может самостоятельно вызвать прерывания IE0, IE1, IS0, IS1. Вызвать прерывания можно через регистры управления портом P1.

#### P1

| Бит                | 7      | 6 | 5 | 4    | 3   | 2   | 1   | 0   |
|--------------------|--------|---|---|------|-----|-----|-----|-----|
| Назначение         | Резерв |   |   | EN_N | IS1 | IS0 | IE1 | IE0 |
| Начальное значение | 255    |   |   |      |     |     |     |     |

**EN\_N** – разрешение прерывания на основе битов текущего регистра:

1 – прерывания будут вызваны в соответствии с таблицей прерываний;

0 – разрешения вызова прерываний битами текущего регистра.

При разрешении вызова прерываний битами текущего регистра, прерывания от периферийных модулей будут отключены от входных линий прерываний IE0, IE1, IS0, IS1.

**IS1** – прерывание IS1:

1 – передать на вход линии прерывания IS1 высокий логический уровень;

0 – передать на вход линии прерывания IS1 низкий логический уровень.

**IS0** – прерывание IS0:

1 – передать на вход линии прерывания IS0 высокий логический уровень;

0 – передать на вход линии прерывания IS0 низкий логический уровень.

**IE1** – прерывание IE1:

1 – передать на вход линии прерывания IE1 высокий логический уровень;

0 – передать на вход линии прерывания IE1 низкий логический уровень.

**IE0** – прерывание IE0:

1 – передать на вход линии прерывания IE0 высокий логический уровень;

0 – передать на вход линии прерывания IE0 низкий логический уровень.

### Таймер-счетчик

В текущей версии внешнее событие для таймера-счетчика всегда равно нулю.

В стандартной реализации таймера-счетчик инкрементируется каждый машинный цикл. В данном случае машинный цикл равен 1 такту системной частоты.

# Блок управления внешними регистрами

Блок управления внешними регистрами предназначен для быстрого доступа к 16 внешним 8-разрядным регистрам. Для доступа к этим регистрам в процессорное ядро 8051 модифицировано поведение команды MOV.

При обращении командой MOV в диапазоны памяти E8h-EFh и D8h-DFh резидентного ОЗУ выполняется остановка большей части блоков процессора и задействуется блок управления внешними регистрами. Этот блок выполняет запись/чтение внешнего регистра, разворачивая диаграмму записи/чтения на линиях ввода-вывода. Блок управления внешними регистрами управляет линиями ввода-вывода напрямую, минуя системную шину, что позволяет прочитать или записать данные в регистр за минимальное количество тактов.

Во время остановки процессора продолжает работать таймер и блок обращения к памяти программ, осуществляющий предварительную вычитку данных в FIFO.

## Обращение к внешним регистрам

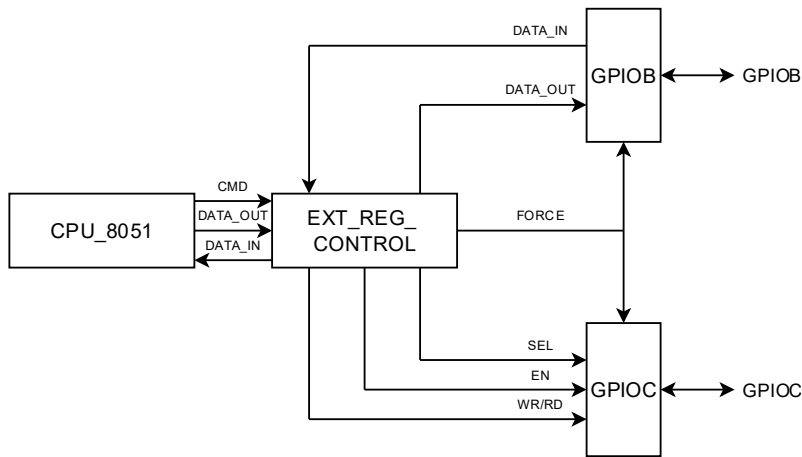


Рисунок 23. Принцип работы блока управления внешними регистрами

Когда ядро 8051 декодирует одну из команд обращения к внешним регистрам, оно передает эту команду (и, в случае записи, данные для записи) в блок управления внешними регистрами (EXT\_REG\_CONTROL). Блок, в свою очередь, выдает сигнал FORCE для GPIOB и GPIOC. Поэтому сигналы на выходы GPIOC начинают выдаваться сигналы SEL, EN и WR/RD для внешних регистров, а на выходы GPIOB выдаются данные для записи в регистр (при записи), либо же эти выходы принимают данные из внешнего регистра (при чтении). Включение/выключение выходного буфера GPIO происходит автоматически, вне зависимости от текущего значения регистра GPIO\_DIR. Вывод GPIO должен быть настроен как вывод общего назначения (соединен с портом P процессора) в регистрах GPIO\_ALTF0 и GPIO\_ALTF1.

Таблица 14. Описание сигналов внешних регистров

| Сигнал    | Вывод GPIO      | Направление                       | Описание                                               |
|-----------|-----------------|-----------------------------------|--------------------------------------------------------|
| DATA<7:0> | GPIOB<7:0>      | Выход при записи, вход при чтении | 8-битная шина данных                                   |
| SELL<3:0> | GPIOC<7:6; 3:2> | Выход                             | Выбор внешнего регистра                                |
| EN        | GPIOC<1>        | Выход                             | Сигнал разрешения доступа к внешнему регистру          |
| WR/RD     | GPIOC<0>        | Выход                             | Сигнал записи/чтения:<br>«1» – запись,<br>«0» – чтение |

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

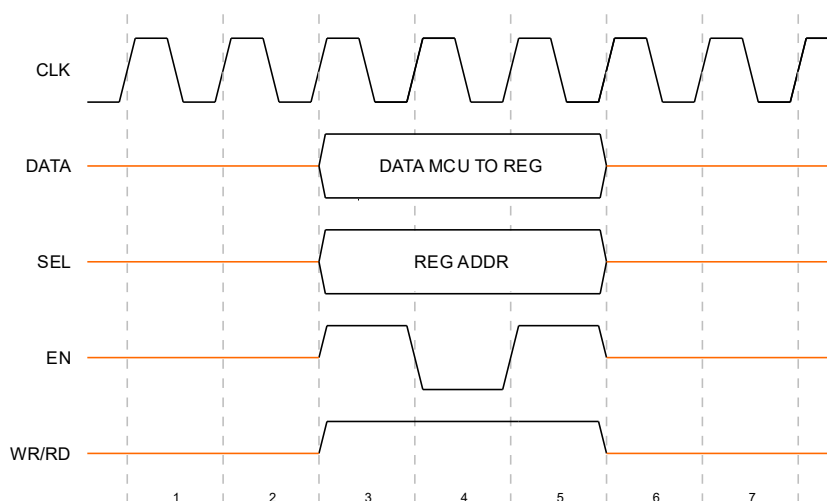


Рисунок 24. Временная диаграмма записи во внешний регистр

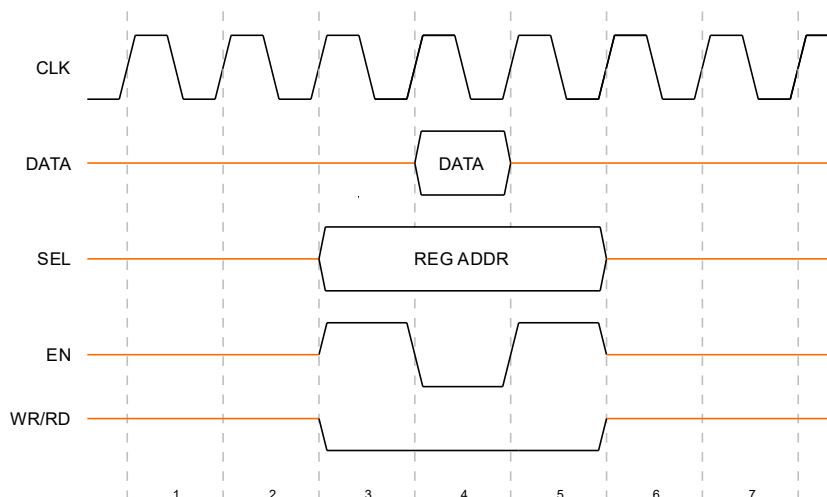


Рисунок 25. Временная диаграмма чтения из внешнего регистра

До начала диаграммы GPIOB и GPIOC были настроены на вход, поэтому все сигналы (DATA, SEL, EN, WR/RD) находятся в состоянии высокого импеданса.

Сигнал CLK, показанный на диаграммах – это тактовый сигнал процессора 8051. Это внутренний сигнал системы, он не выдается на внешние регистры, однако именно к нему привязаны обе формируемые диаграммы.

Диаграмма начинается в момент, когда процессор начинает выполнение команды MOV. На такте 1 CPU декодирует полученную из памяти программ команду. На такте 2 он передает команду в блок управления регистрами. В этот момент блок управления внешними регистрами захватывает контроль над GPIOB и GPIOC. Следующие 3 такта (такты 3, 4, 5) блок выдает диаграмму записи или чтения. После этого все линии GPIOB и GPIOC возвращаются в состояние высокого импеданса.

На тактах 6 и 7 несмотря на то, что чтение или запись закончены, CPU удерживается в состоянии остановки. Это сделано для того, чтобы если следующей командой CPU считал значение со своих портов P, он не считал неверные данные. Данные от GPIO к портам P попадают через синхронизатор, который задерживает данные на 2 такта. Соответственно, эти 2 такта, CPU должен удерживаться, чтобы данные успели пройти через синхронизатор. Таким образом, после окончания команды MOV CPU увидит на своих портах точно те же данные, что и до выполнения команды MOV.

На 8 такте CPU начинает выполнение следующей команды.

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

## Типы статусов и прерывания

Биты в регистрах статусов периферийных блоков подразделяются на флаги (FLAG) и события (EVENT). Разница между ними в том, что флаги сбрасываются, когда указанное условие перестает выполняться, а биты, соответствующие событиям, сбрасываются по чтению. При этом фиксация события имеет приоритет над сбросом.

Прерывание по биту-флагу формируется в момент перехода бита статуса из лог. «0» в лог. «1» (но не наоборот). Если бит статуса уже находился в лог. «1», и в этот момент разрешается прерывание по этому биту, то прерывание тут же сработает.

Прерывание по биту-событию формируется в момент возникновения данного события. Следует иметь в виду, что бит в регистре статуса установится в лог. «1» от первого события, и, если его не сбросить чтением, то он останется в лог. «1». Прерывания же формируются и от последующих событий и неважно в лог. «0» или в лог. «1» статусный бит.

## Система управления сбросом и питанием (PMM)

### Структурная схема

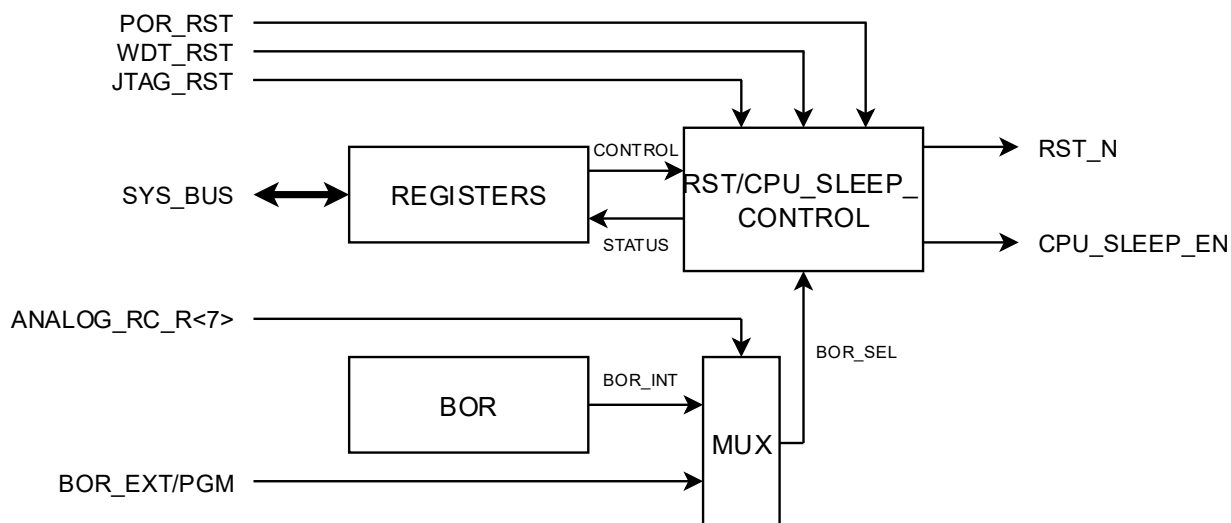


Рисунок 26. Структурная схема системы управления сбросом и питанием

Система управления сбросом и питанием состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- BOR – супервизор питания, формирует сигнал BOR при падении напряжения питания ниже  $3,8\text{ В} \pm 0,2\text{ В}$ ;
- MUX – мультиплексор, позволяет выбрать источник BOR битом ANALOG\_RC\_R<7> конфигурационной памяти;
- RST/CPU\_SLEEP\_CONTROL – блок управления сбросом и сном процессора, формирует системный сброс и разрешение на переход в режим «Сон процессора».

### Формирование сброса

Выводы POR\_RST и BOR\_EXT/PGM имеют активный уровень «0». Вывод BOR\_EXT/PGM работает, только когда вывод TM находится в «0». После того как на выводе POR\_RST и внутреннем сигнале BOR\_SEL сформируется неактивный уровень, еще в течение 1000 тактов системной частоты блок RST/CPU\_SLEEP\_CONTROL подает сигнал сброса системы RST\_N. Микроконтроллер может быть сброшен программными методами от сторожевого таймера (сигнал WDT\_RST) или через JTAG (сигнал JTAG\_RST). Следует иметь в виду, что JTAG сбрасывается только по сигналу POR\_RST, а модуль DEBUGGER по POR\_RST и BOR\_SEL. В модуле CMM регистры, связанные с выбором и настройкой текущей частоты сбрасываются только по POR\_RST и BOR\_SEL, однако регистры CMM\_ST, CMM\_MSK и регистры, связанные с переходами в какой-либо режим «SLEEP», сбрасываются еще и по WDT\_RST и JTAG\_RST. Таким образом, после сброса от сторожевого таймера или JTAG система продолжит работу на прежней частоте или продолжит переход на другую частоту, а если находится в режиме «SLEEP», то выйдет из него.

Также блок RST/CPU\_SLEEP\_CONTROL на основании данных блока REGISTERS формирует сигнал разрешения перехода в режим «Сон процессора» для модуля CMM (сигнал CPU\_SLEEP\_EN).

## Регистры модуля системы управления сбросом и питанием

| №     | Аббревиатура | Доступ | Описание           |
|-------|--------------|--------|--------------------|
| 2000h | PMM_CTRL     | W      | Регистр управления |
| 2001h | PMM_ST       | RC     | Регистр статусов   |

## PMM\_CTRL

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0            |
|--------------------|--------|---|---|---|---|---|---|--------------|
| Назначение         | Резерв |   |   |   |   |   |   | CPU_SLEEP_EN |
| Начальное значение | 0      |   |   |   |   |   |   |              |

**CPU\_SLEEP\_EN** – переход в режим «Сон процессора»;

1 – перейти в режим «Сон процессора»;

0 – режим «Сон процессора» выключен.

## PMM\_ST

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1        | 0       |
|--------------------|--------|---|---|---|---|---|----------|---------|
| Назначение         | Резерв |   |   |   |   |   | JTAG_RST | WDT_RST |
| Тип статуса        |        |   |   |   |   |   | EVENT    |         |
| Начальное значение | 0      |   |   |   |   |   |          |         |

**JTAG\_RST** – зафиксирован сброс по JTAG.

**WDT\_RST** – зафиксирован сброс по сторожевому таймеру.

## Модуль управления источниками тактовых сигналов системы (СММ)

### Общая информация

Модуль управления источниками тактовых сигналов системы предназначен для переключения между источниками тактирования, изменения периода синхросигнала и для перехода в режим «SLEEP».

Микроконтроллер в процессе работы может переключаться между двумя источниками тактового сигнала:

- XTAL\_CLK;
- RC\_CLK.

В свою очередь, источником синхросигнала XTAL\_CLK в зависимости от настроек конфигурационной памяти (модуль ANALOG\_CFG) может быть как внешний кварцевый генератор, так и встроенный RC-генератор, а источником RC\_CLK всегда является встроенный RC-генератор.

### Структурная схема

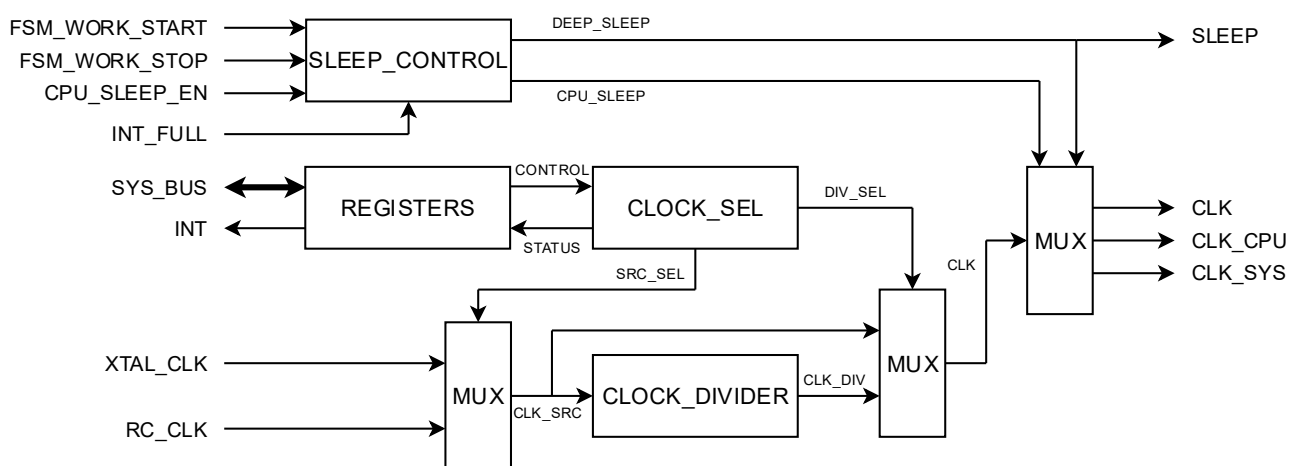


Рисунок 27. Структурная схема модуля управления источниками тактовых сигналов системы

Модуль управления источниками тактовых сигналов системы состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- SLEEP\_CONTROL – блок управления режимом «SLEEP», предназначен для запуска режимов «Сон процессора» и «Глубокий сон»;
- CLOCK\_SEL – блок выбора частоты тактирования, предназначен для выбора источника частоты тактирования системы;
- MUX – мультиплексор, предназначен для управления частотами, которые будут переданы в систему;
- CLOCK\_DIVIDER – делитель частоты, предназначен для деления частоты выбранного источника тактирования.

### Выбор источника тактирования системы

При подаче питания тактирование системы осуществляется от XTAL\_CLK. Пользователь может изменить источник тактирования путем записи необходимого значения в бит RC\_XTAL регистра CMM\_CTRL. На время операции переключения в регистре CMM\_ST в активном уровне будет находиться бит SWITCH. Когда операция переключения будет завершена, бит SWITCH примет значение «0», а NSWITCH «1». При большой разнице частот, процесс переключения займет порядка 12 тактов частоты, на которую происходит переключение, и 3 тактов текущей частоты.

Пользователь имеет возможность не переключаться между источниками XTAL\_CLK и RC\_CLK при переходе в режим «Глубокий сон», если источником XTAL\_CLK уже является RC-генератор без ФАПЧ. Переход системы в режим «Глубокий сон» при тактировании от источника XTAL\_CLK на какой-либо другой частоте приведет к полной остановке системы без возможности программного восстановления работоспособности.

### Деление частоты тактирования

Текущая частота тактирования системы может быть поделена. Для этого необходимо записать коэффициент в регистр CMM\_DIV.

### Регистры модуля управления источниками тактовых сигналов системы

| №     | Аббревиатура | Доступ | Описание                 |
|-------|--------------|--------|--------------------------|
| 2100h | CMM_CTRL     | RW     | Регистр управления       |
| 2101h | CMM_DIV      | RW     | Регистр делителя частоты |
| 2102h | CMM_MSK      | RW     | Регистр маски            |
| 2103h | CMM_ST       | R      | Регистр статуса          |

#### CMM\_CTRL

| Бит                | 7       | 6 | 5 | 4 | 3 | 2       | 1 | 0       |
|--------------------|---------|---|---|---|---|---------|---|---------|
| Назначение         | Резерв* |   |   |   |   | Резерв* |   | RC_XTAL |
| Начальное значение | 0       |   |   |   |   | 1       |   | 0       |

\* – Не изменять начальное значение.

**RC\_XTAL** – бит управления источником тактирования системы  $f_{src}$ :

1 – тактирование от RC\_CLK;

0 – тактирование от XTAL\_CLK.

#### CMM\_DIV

| Бит                | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|------|---|---|---|---|---|---|---|
| Назначение         | COEF |   |   |   |   |   |   |   |
| Начальное значение | 0    |   |   |   |   |   |   |   |

**COEF** – коэффициент деления исходной частоты тактирования.

При COEF>0 частота тактирования системы рассчитывается по формуле:

$$f_{clk} = \frac{f_{src}}{2 * COEF},$$

где  $f_{src}$  – исходная частота от источника XTAL\_CLK или RC\_CLK.

При COEF=0:  $f_{clk} = f_{src}$

#### CMM\_MSK

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
|--------------------|--------|---|---|---|---|---|---|---------|
| Назначение         | Резерв |   |   |   |   |   |   | NSWITCH |
| Начальное значение | 0      |   |   |   |   |   |   |         |

**NSWITCH** – разрешить прерывание по статусу NSWITCH:

1 – данное прерывание формируется;

0 – данное прерывание не формируется.

## CMM\_ST

| Бит                | 7             | 6 | 5 | 4 | 3 | 2 | 1      | 0       |
|--------------------|---------------|---|---|---|---|---|--------|---------|
| Назначение         | <i>Резерв</i> |   |   |   |   |   | SWITCH | NSWITCH |
| Тип статуса        |               |   |   |   |   |   | FLAG   | EVENT   |
| Начальное значение | 0             |   |   |   |   |   | 0      | 0       |

**SWITCH** – переключение между частотой XTAL\_CLK и RC\_CLK:

1 – идет переключение на выбранную частоту;

0 – система работает на выбранной частоте.

**NSWITCH** – процесс переключения частоты тактирования окончен.

## Сторожевой таймер (WDT)

### Общая информация

Сторожевой таймер предназначен для принудительной перезагрузки системы в случае ее зависания. В основе таймера лежит счетчик разрядностью 32 бита со счетом «вниз». Модуль формирует регулярное прерывание INT\_WDT в зависимости от запрограммированного значения. Каждый такт синхросигнала значение счетчика уменьшается на единицу. Когда значение счетчика достигает 0, формируется сигнал прерывания. Затем счетчик перезагружается и заново начинает отсчет к нулю. Если к моменту, когда счетчик достиг заново значения 0, прерывание не очищено, то в систему формируется сигнал сброса WDT\_RST. Таким образом, сторожевой таймер предоставляет возможность восстановления системы после сбоя программного обеспечения. При необходимости модуль может быть выключен.

### Регистры сторожевого таймера

| №     | Аббревиатура | Доступ | Описание                                         |
|-------|--------------|--------|--------------------------------------------------|
| 2200h | WDT_LOAD0    | RW     | Период счета сторожевого таймера 0               |
| 2201h | WDT_LOAD1    | RW     | Период счета сторожевого таймера 1               |
| 2202h | WDT_LOAD2    | RW     | Период счета сторожевого таймера 2               |
| 2203h | WDT_LOAD3    | RW     | Период счета сторожевого таймера 3               |
| 2204h | WDT_VAL0     | R      | Текущее значение сторожевого таймера 0           |
| 2205h | WDT_VAL1     | R      | Текущее значение сторожевого таймера 1           |
| 2206h | WDT_VAL2     | R      | Текущее значение сторожевого таймера 2           |
| 2207h | WDT_VAL3     | R      | Текущее значение сторожевого таймера 3           |
| 2208h | WDT_CTRL     | RW     | Регистр управления                               |
| 220Ch | WDT_CLR      | W      | Регистр сброса прерывания                        |
| 2210h | WDT_INTRAW   | R      | Регистр исходного прерывания                     |
| 2214h | WDT_INT      | R      | Регистр маскируемого прерывания                  |
| 2218h | WDT_LOCK     | RW     | Регистр блокировки доступа к сторожевому таймеру |
| 221Ch | WDT_TCR      | RW     | Регистр перехода в тестовый режим                |
| 2220h | WDT_TOP      | W      | Регистр управления в тестовом режиме             |

### WDT\_LOAD0

Регистр содержит значение, с которого счетчик начнет уменьшаться. При перезаписи данного регистра таймер немедленно стартует с записанного значения. Минимальное допустимое значение для WDT\_LOAD равно 1. 32-х разрядное значение поместится в регистр сторожевого таймера в момент записи в данный регистр.

| Бит                | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-------|---|---|---|---|---|---|---|
| Назначение         | VALUE |   |   |   |   |   |   |   |
| Начальное значение | 0     |   |   |   |   |   |   |   |

**VALUE** – период счета сторожевого таймера, младшая часть.

**WDT\_LOAD1**

| Бит                | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-------|---|---|---|---|---|---|---|
| Назначение         | VALUE |   |   |   |   |   |   |   |
| Начальное значение | 0     |   |   |   |   |   |   |   |

**VALUE** – период счета сторожевого таймера, биты <15:8>.

**WDT\_LOAD2**

| Бит                | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-------|---|---|---|---|---|---|---|
| Назначение         | VALUE |   |   |   |   |   |   |   |
| Начальное значение | 0     |   |   |   |   |   |   |   |

**VALUE** – период счета сторожевого таймера, биты <23:16>.

**WDT\_LOAD3**

| Бит                | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-------|---|---|---|---|---|---|---|
| Назначение         | VALUE |   |   |   |   |   |   |   |
| Начальное значение | 0     |   |   |   |   |   |   |   |

**VALUE** – период счета сторожевого таймера, старшая часть.

**WDT\_VAL0**

| Бит                | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------------|---|---|---|---|---|---|---|
| Назначение         | CURRENT_VALUE |   |   |   |   |   |   |   |
| Начальное значение | 0             |   |   |   |   |   |   |   |

**CURRENT\_VALUE** – текущее значение таймера, младшая часть.

**WDT\_VAL1**

| Бит                | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------------|---|---|---|---|---|---|---|
| Назначение         | CURRENT_VALUE |   |   |   |   |   |   |   |
| Начальное значение | 0             |   |   |   |   |   |   |   |

**CURRENT\_VALUE** – текущее значение таймера, биты <15:8>.

**WDT\_VAL2**

| Бит                | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------------|---|---|---|---|---|---|---|
| Назначение         | CURRENT_VALUE |   |   |   |   |   |   |   |
| Начальное значение | 0             |   |   |   |   |   |   |   |

**CURRENT\_VALUE** – текущее значение таймера, биты <23:16>.

**WDT\_VAL3**

| Бит                | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------------|---|---|---|---|---|---|---|
| Назначение         | CURRENT_VALUE |   |   |   |   |   |   |   |
| Начальное значение | 0             |   |   |   |   |   |   |   |

**CURRENT\_VALUE** – текущее значение таймера, старшая часть.

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

**WDT\_CTRL**

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1      | 0      |
|--------------------|--------|---|---|---|---|---|--------|--------|
| Назначение         | Резерв |   |   |   |   |   | RST_EN | INT_EN |
| Начальное значение | 0      |   |   |   |   |   |        |        |

**RST\_EN** – разрешение на формирование сигнала сброса по завершению счета таймера:

1 – сброс разрешен;

0 – сброс запрещен.

**INT\_EN** – разрешение на формирование прерывания по окончании счета таймера:

1 – прерывание разрешено;

0 – прерывание запрещено.

Сторожевой таймер начинает счет, если установлен бит INT\_EN. Сторожевой таймер перестает считать, если бит INT\_EN сброшен в ноль. После разрешения прерывания, если пользователь до этого его запрещал, счетчик автоматически перезагрузится со значения из регистра WDT\_LOAD.

**WDT\_CLR**

Запись любого значения в данный регистр очищает прерывание сторожевого таймера и перезагружает счетчик значением из регистра WDT\_LOAD.

| Бит                | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-------|---|---|---|---|---|---|---|
| Назначение         | CLEAR |   |   |   |   |   |   |   |
| Начальное значение | 0     |   |   |   |   |   |   |   |

**WDT\_INTRAW**

Данный регистр указывает на необработанное прерывание от счетчика. На основании данного сигнала формируется маскируемое прерывание.

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
|--------------------|--------|---|---|---|---|---|---|---------|
| Назначение         | Резерв |   |   |   |   |   |   | RAW_WDT |
| Начальное значение | 0      |   |   |   |   |   |   |         |

**RAW\_WDT** – исходное прерывание таймера:

1 – прерывание произошло;

0 – прерывание отсутствует.

**WDT\_INT**

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
|--------------------|--------|---|---|---|---|---|---|---------|
| Назначение         | Резерв |   |   |   |   |   |   | INT_WDT |
| Начальное значение | 0      |   |   |   |   |   |   |         |

**INT\_WDT** – маскируемое прерывание таймера:

1 – прерывание произошло;

0 – прерывание отсутствует.

Данное прерывание формируется на основании битов RAW\_WDT и INT\_EN и передается в систему.

**WDT\_LOCK**

Данный регистр блокирует доступ на запись во все остальные регистры сторожевого таймера. Запись значения 55h обеспечивает доступ на запись ко всем регистрам. Запись любого другого значения блокирует доступ. По умолчанию запись во все регистры сторожевого таймера заблокирована.

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0        |
|--------------------|--------|---|---|---|---|---|---|----------|
| Назначение         | Резерв |   |   |   |   |   |   | LOCK_WDT |
| Начальное значение | 0      |   |   |   |   |   |   | 1        |

**LOCK\_WDT** – блокирование записи в регистры:

1 – запись во все регистры сторожевого таймера заблокирована;

0 – запись во все регистры сторожевого таймера разрешена.

**WDT\_TCR**

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
|--------------------|--------|---|---|---|---|---|---|---------|
| Назначение         | Резерв |   |   |   |   |   |   | TEST_EN |
| Начальное значение | 0      |   |   |   |   |   |   |         |

**TEST\_EN** – тестовый режим:

1 – сторожевой таймер в тестовом режиме;

0 – сторожевой таймер в рабочем режиме.

В тестовом режиме пользователь непосредственно управляет маскируемым прерыванием и сбросом от сторожевого таймера через регистр WDT\_TOP.

**WDT\_TOP**

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1       | 0       |
|--------------------|--------|---|---|---|---|---|---------|---------|
| Назначение         | Резерв |   |   |   |   |   | TST_INT | TST_RES |
| Начальное значение | 0      |   |   |   |   |   |         |         |

**TST\_INT** – формирование сигнала прерывания от таймера в тестовом режиме:

1 – прерывание активно;

0 – прерывание не активно.

**TST\_RES** – формирование сигнала сброса от таймера в тестовом режиме:

1 – сброс активен;

0 – сброс не активен.



Если вывод соединен с альтернативной функцией, то направление (вход/выход) определяется этой альтернативной функцией (периферийным устройством DEVICEх).

В блоке GPIO присутствуют два детектора, способных формировать прерывания - синхронный (SYNCH\_EDGE\_DETECTOR) и асинхронный (ASYNCH\_EDGE\_DETECTOR). Работа детекторов управляется регистрами INTEN, INTPOL и INTTYPE. Статус прерываний сохраняется в регистре INT.

### Статусы и прерывания

GPIO поддерживает два режима регистрации событий - синхронный и асинхронный. Синхронный детектор работает в рабочем режиме и в режиме «Сон процессора», но не работает в режиме «Глубокий сон». Асинхронный детектор, наоборот работает только в режиме «Глубокий сон» и предназначен для вывода системы из него по внешнему сигналу.

Прерывание для каждого из выводов разрешается и запрещается записью в регистры GPIO\_INTEN\_SET/ GPIO\_INTEN\_CLR.

Синхронное прерывание может быть сформировано как по фронту сигнала, так и по уровню, выбор типа прерывания осуществляется записью в регистры GPIO\_INTTYPE\_SET/GPIO\_INTTYPE\_CLR. Регистры GPIO\_INTPOL\_SET/GPIO\_INTPOL\_CLR определяют, какой уровень (низкий/высокий) или какой фронт (возрастающий/спадающий) вызовет прерывание.

Асинхронный детектор не использует системную частоту, поэтому может работать в режиме «Глубокого сна» микроконтроллера. Асинхронное прерывание выводит микроконтроллер из режима «SLEEP», таким образом блок GPIO можно использовать, чтобы выйти из режима «Глубокого сна» по внешнему событию. Асинхронный детектор фронта работает с несинхронизированным на системную частоту входным сигналом, поэтому даже короткий глитч входного сигнала будет гарантированно зарегистрирован как фронт.

Асинхронное прерывание может быть сформировано только по фронту. При переходе в режим «Глубокого сна» (где работает асинхронный детектор) значение в регистрах GPIO\_INTTYPE\_SET/ GPIO\_INTTYPE\_CLR игнорируется, прерывание срабатывает по фронту сигнала (возрастающему или спадающему, в зависимости от GPIO\_INTPOL\_SET/GPIO\_INTPOL\_CLR).

Какой именно вывод вызвал прерывание можно выяснить, прочитав регистр статуса прерываний GPIO\_INT. Соответствующий бит в регистре GPIO\_INT выставляется в «1», только если прерывание по этому выводу разрешено.

### Регистры GPIO

| №            | Аббревиатура      | Доступ | Описание                                                              |
|--------------|-------------------|--------|-----------------------------------------------------------------------|
| <b>GPIOA</b> |                   |        |                                                                       |
| 2300h        | GPIOA_DIR_SET     | RW     | Установка режима работы выходного буфера                              |
| 2301h        | GPIOA_DIR_CLR     | RW     |                                                                       |
| 2304h        | GPIOA_ALTF0       | RW     | Выбор альтернативной функции                                          |
| 2305h        | GPIOA_ALTF1       | RW     |                                                                       |
| 2306h        | GPIOA_INTEN_SET   | RW     | Разрешение прерываний                                                 |
| 2307h        | GPIOA_INTEN_CLR   | RW     |                                                                       |
| 2308h        | GPIOA_INTTYPE_SET | RW     | Выбор типа прерывания (фронт/уровень)                                 |
| 2309h        | GPIOA_INTTYPE_CLR | RW     |                                                                       |
| 230Ah        | GPIOA_INTPOL_SET  | RW     | Выбор полярности входного сигнала, при которой формируются прерывания |
| 230Bh        | GPIOA_INTPOL_CLR  | RW     |                                                                       |
| 230Ch        | GPIOA_INT         | R      | Статус прерываний                                                     |

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

| №            | Аббревиатура      | Доступ | Описание                                                              |
|--------------|-------------------|--------|-----------------------------------------------------------------------|
| <b>GPIOB</b> |                   |        |                                                                       |
| 2400h        | GPIOB_DIR_SET     | RW     | Установка режима работы выходного буфера                              |
| 2401h        | GPIOB_DIR_CLR     | RW     |                                                                       |
| 2404h        | GPIOB_ALTF0       | RW     | Выбор альтернативной функции                                          |
| 2405h        | GPIOB_ALTF1       | RW     |                                                                       |
| 2406h        | GPIOB_INTEN_SET   | RW     | Разрешение прерываний                                                 |
| 2407h        | GPIOB_INTEN_CLR   | RW     |                                                                       |
| 2408h        | GPIOB_INTTYPE_SET | RW     | Выбор типа прерывания (фронт/уровень)                                 |
| 2409h        | GPIOB_INTTYPE_CLR | RW     |                                                                       |
| 240Ah        | GPIOB_INTPOL_SET  | RW     | Выбор полярности входного сигнала, при которой формируются прерывания |
| 240Bh        | GPIOB_INTPOL_CLR  | RW     |                                                                       |
| 240Ch        | GPIOB_INT         | R      | Статус прерываний                                                     |
| <b>GPIOC</b> |                   |        |                                                                       |
| 2500h        | GPIOC_DIR_SET     | RW     | Установка режима работы выходного буфера                              |
| 2501h        | GPIOC_DIR_CLR     | RW     |                                                                       |
| 2504h        | GPIOC_ALTF0       | RW     | Выбор альтернативной функции                                          |
| 2505h        | GPIOC_ALTF1       | RW     |                                                                       |
| 2506h        | GPIOC_INTEN_SET   | RW     | Разрешение прерываний                                                 |
| 2507h        | GPIOC_INTEN_CLR   | RW     |                                                                       |
| 2508h        | GPIOC_INTTYPE_SET | RW     | Выбор типа прерывания (фронт/уровень)                                 |
| 2509h        | GPIOC_INTTYPE_CLR | RW     |                                                                       |
| 250Ah        | GPIOC_INTPOL_SET  | RW     | Выбор полярности входного сигнала, при которой формируются прерывания |
| 250Bh        | GPIOC_INTPOL_CLR  | RW     |                                                                       |
| 250Ch        | GPIOC_INT         | R      | Статус прерываний                                                     |

**GPIOx\_DIR\_SET/CLR**

GPIO\_DIR\_SET/GPIO\_DIR\_CLR – парные регистры управления режимом работы выходных буферов порта.

| Бит                | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
|--------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| Назначение         | IO7_DIR | IO6_DIR | IO5_DIR | IO4_DIR | IO3_DIR | IO2_DIR | IO1_DIR | IO0_DIR |
| Начальное значение | 0       |         |         |         |         |         |         |         |

Запись в **IOx\_DIR** регистра GPIO\_DIR\_SET:

- 1 – включить выходной буфер на передачу;
- 0 – не меняет текущую настройку.

Запись в **IOx\_DIR** регистра GPIO\_DIR\_CLR:

- 1 – выключить выходной буфер;
- 0 – не меняет текущую настройку.

Чтение **IOx\_DIR** регистров GPIO\_DIR\_SET/GPIO\_DIR\_CLR:

- 1 – выходной буфер включен на передачу;
- 0 – выходной буфер выключен.

**GPIOx\_ALTF0**

| Бит                | 7        | 6 | 5        | 4 | 3        | 2 | 1        | 0 |
|--------------------|----------|---|----------|---|----------|---|----------|---|
| Назначение         | IO3_ALTF |   | IO2_ALTF |   | IO1_ALTF |   | IO0_ALTF |   |
| Начальное значение | 0        |   |          |   |          |   |          |   |

**IOx\_ALTF** – альтернативная функция:

11b – включена альтернативная функция ALTF2;

10b – включена альтернативная функция ALTF1;

01b – включена альтернативная функция ALTF0;

00b – альтернативные функции выключены, выходным буфером управляет GPIO.

**GPIOx\_ALTF1**

| Бит                | 7        | 6 | 5        | 4 | 3        | 2 | 1        | 0 |
|--------------------|----------|---|----------|---|----------|---|----------|---|
| Назначение         | IO7_ALTF |   | IO6_ALTF |   | IO5_ALTF |   | IO4_ALTF |   |
| Начальное значение | 0        |   |          |   |          |   |          |   |

**IOx\_ALTF** – альтернативная функция:

11b – включена альтернативная функция ALTF2;

10b – включена альтернативная функция ALTF1;

01b – включена альтернативная функция ALTF0;

00b – альтернативные функции выключены, выходным буфером управляет GPIO.

**GPIOx\_INTEN\_SET/CLR**

GPIO\_INTEN\_SET/GPIO\_INTEN\_CLR – парные регистры установки разрешения генерации прерываний по событиям на входах GPIO.

| Бит                | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|--------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Назначение         | IO7_IE | IO6_IE | IO5_IE | IO4_IE | IO3_IE | IO2_IE | IO1_IE | IO0_IE |
| Начальное значение | 0      |        |        |        |        |        |        |        |

Запись в **IOx\_IE** регистра GPIO\_INTEN\_SET:

1 – разрешить генерацию прерывания по событиям на данном входе;

0 – не меняет текущую настройку.

Запись в **IOx\_IE** регистра GPIO\_INTEN\_CLR:

1 – запретить генерацию прерывания по событиям на данном входе;

0 – не меняет текущую настройку.

Чтение **IOx\_IE** регистров GPIO\_INTEN\_SET/GPIO\_INTEN\_CLR:

1 – разрешена генерация прерывания по событиям на данном входе;

0 – запрещена генерация прерывания по событиям на данном входе.

**GPIOx\_INTTYPE\_SET/CLR**

GPIO\_INTTYPE\_SET/GPIO\_INTTYPE\_CLR – парные регистры установки типа прерывания (по фронту/уровню) генерируемого GPIO.

| Бит                | 7         | 6         | 5         | 4         | 3         | 2         | 1         | 0         |
|--------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Назначение         | IO7_ITYPE | IO6_ITYPE | IO5_ITYPE | IO4_ITYPE | IO3_ITYPE | IO2_ITYPE | IO1_ITYPE | IO0_ITYPE |
| Начальное значение | 0         |           |           |           |           |           |           |           |

Запись в **IOx\_ITYPE** регистра GPIO\_INTTYPE\_SET:

- 1 – установить генерацию прерывания по фронту;
- 0 – не меняет текущую настройку.

Запись в **IOx\_ITYPE** регистра GPIO\_INTTYPE\_CLR:

- 1 – установить генерацию прерывания по уровню;
- 0 – не меняет текущую настройку.

Чтение **IOx\_ITYPE** регистров GPIO\_INTTYPE\_SET/GPIO\_INTTYPE\_CLR:

- 1 – генерация прерывания осуществляется по фронту;
- 0 – генерация прерывания осуществляется по уровню.

**GPIOx\_INTPOL\_SET/CLR**

GPIO\_INTPOL\_SET/GPIO\_INTPOL\_CLR – парные регистры установки полярности события GPIO, по которому генерируется прерывание.

| Бит                | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|--------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Назначение         | IO7_IPOL | IO6_IPOL | IO5_IPOL | IO4_IPOL | IO3_IPOL | IO2_IPOL | IO1_IPOL | IO0_IPOL |
| Начальное значение | 0        |          |          |          |          |          |          |          |

Запись в **IOx\_IPOL** регистра GPIO\_INTPOL\_SET:

- 1 – установить генерацию прерывания по положительному фронту или высокому уровню (зависит от GPIO\_INTTYPE\_SET/CLR);
- 0 – не меняет текущую настройку.

Запись в **IOx\_IPOL** регистра GPIO\_INTPOL\_CLR:

- 1 – установить генерацию прерывания по отрицательному фронту или низкому уровню (зависит от GPIO\_INTTYPE\_SET/CLR);
- 0 – не меняет текущую настройку.

Чтение **IOx\_IPOL** регистров GPIO\_INTPOL\_SET/GPIO\_INTPOL\_CLR:

- 1 – генерация прерывания осуществляется по положительному фронту или высокому уровню;
- 0 – генерация прерывания осуществляется по отрицательному фронту или низкому уровню.

**GPIOx\_INT**

| Бит                | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
|--------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| Назначение         | IO7_INT | IO6_INT | IO5_INT | IO4_INT | IO3_INT | IO2_INT | IO1_INT | IO0_INT |
| Тип статуса        | EVENT   |         |         |         |         |         |         |         |
| Начальное значение | 0       |         |         |         |         |         |         |         |

**IOx\_INT** – статус прерывания соответствующего вывода GPIO:

- 1 – был зафиксирован фронт или уровень (согласно заданным в регистрах GPIO\_INTPOL\_SET/CLR и GPIO\_INTTYPE\_SET/CLR условиям) на данном выводе;
- 0 – фронт или уровень не был зафиксирован.

## SPI

### Общая информация

SPI – последовательный синхронный стандарт передачи данных в режиме дуплекса, предназначенный для обеспечения простого высокоскоростного сопряжения микроконтроллера и периферии.

Основные характеристики:

- работа в режиме «ведомого» и «ведущего»;
- работа в дуплексном и симплексном режимах;
- формат кадра 8 или 16 бит;
- максимальная скорость передачи соответствует системной частоте;
- отдельные буферы на прием и передачу глубиной 8 слов;
- фильтр по линии синхросигнала.

### Структурная схема

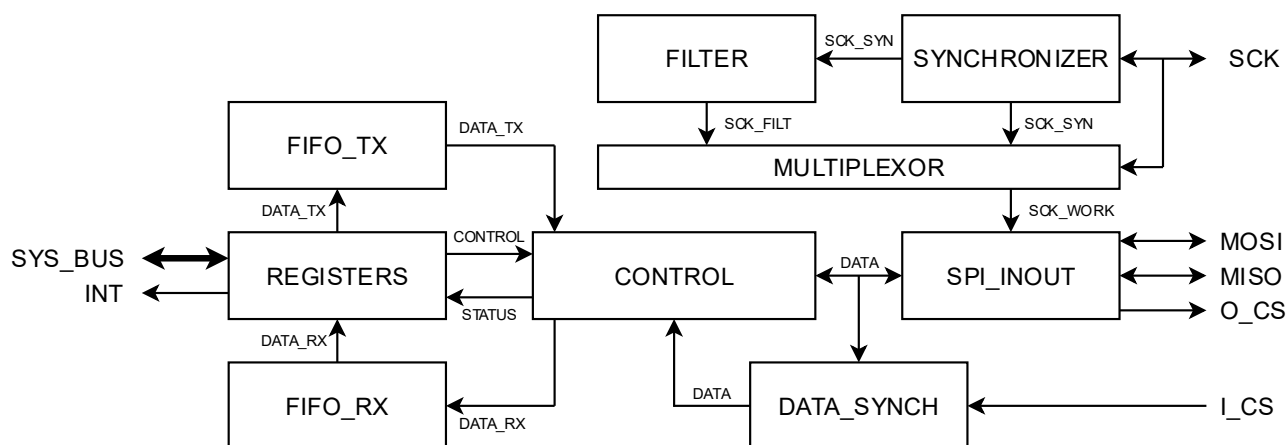


Рисунок 29. Структурная схема SPI интерфейса

SPI состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- FIFO\_TX – буфер передатчика;
- FIFO\_RX – буфер приемника;
- CONTROL – управляющий автомат модуля SPI;
- SPI\_INOUT – блок управления линиями интерфейса;
- DATA\_SYNCH – синхронизатор данных, производящий синхронизацию данных с системной частоты на частоту SCK и в обратном направлении;
- SYNCHRONIZER – синхронизатор линии SCK, работающий по требованию пользователя;
- FILTER – фильтр линии SCK, избавляющий от помех;
- MULTIPLEXOR – мультиплексор, выбирающий частоту тактирования блока SPI\_INOUT в зависимости от входных настроек.

## Алгоритмы работы

В режиме «ведущий» модуль автоматически генерирует синхросигнал для ведомого устройства по линии SCK. Процедура работы с ведущим:

- задействовать один из свободных выводов GPIO как CS, настроить его на выход и перевести в «1»;
- в регистре SPI\_MSK разрешить необходимые прерывания;
- в регистре SPI\_CFG2 установить биты BR для определения скорости обмена;
- в регистре SPI\_CFG0:
  - определить формат данных с помощью бита DWW;
  - настроить формат кадра при помощи бита BO;
  - установить биты POL и PHA для определения соответствия между данными и синхросигналом (настройка должна быть одинакова для ведущего и для ведомого);
  - установить бит MOD для перехода в режим «ведущий».
- для режимов, связанных с передачей данных необходимо записать отправляемую информацию в регистры SPI\_TX1 и SPI\_TX0 (необходимо записать столько слов данных, сколько необходимо отправить за данную передачу; имеется возможность отправить более 8 слов в режимах с PHA в «1», дописывая слова по опустошению буфера передатчика);
- перевести CS в «0», используя ранее настроенный GPIO, таким образом разрешив работу ведомому устройству;
- в регистре SPI\_CTRL разрешить работу модуля битом EN;
- на основании данных регистра SPI\_ST установить момент завершения передачи, перевести CS в «1» и выключить модуль битом EN в регистре SPI\_CTRL;
- принимаемые данные можно вычитать из регистров SPI\_RX1 и SPI\_RX0;
- при необходимости продолжить работу с процедуры записи отправляемой информации или следующей за ней процедуры.

В данном формате MOSI работает как выход данных, а MISO как вход данных. Так как CS задается через вывод GPIO, пользователю необходимо самостоятельно выдерживать задержки между фронтами SCK и CS в начале и конце передачи данных. Минимальная задержка перед первым фронтом SCK определяется временем записи бита EN в регистр SPI\_CTRL.

В режиме «ведомый» синхросигнал поступает от ведущего устройства. Значение бит BR не влияет на работу модуля. Рекомендуется включить модуль до того, как начнет передаваться синхросигнал, иначе будут приняты сдвинутые данные. Буфер передатчика должен содержать данные **до первого изменения синхросигнала**, иначе отправляемые данные могут быть искажены при перезаписи в сдвиговой регистр.

Процедура работы с ведомым:

- в регистре SPI\_MSK разрешить необходимые прерывания;
- в регистре SPI\_CFG0:
  - определить формат данных с помощью бита DWW;
  - настроить формат кадра при помощи бита BO;
  - установить биты POL и PHA для определения соответствия между данными и синхросигналом (настройка должна быть одинакова для ведущего и для ведомого);
  - установить бит MOD для перехода в режим «ведомый».
- в регистре SPI\_CTRL разрешить работу модуля битом EN;
- для режимов, связанных с передачей данных необходимо записать отправляемую информацию в регистры SPI\_TX1 и SPI\_TX0;
- на основании данных регистра SPI\_ST установить момент завершения передачи;
- принимаемые данные можно вычитать из регистров SPI\_RX1 и SPI\_RX0;
- при необходимости продолжить передачу данных с процедуры записи отправляемой информации или следующей за ней процедуры.

В данном формате MOSI работает как вход данных, MISO как выход данных.

## Дуплексный и симплексный режимы

SPI может работать в двух конфигурациях:

- две однонаправленные линии данных;
- одна двунаправленная линия данных (режим «только прием» или «только передача»).

Две однонаправленные линии данных (стандартный режим). Активируется, когда бит TR\_MOD регистра SPI\_CFG0 равен «0». Бит RXO регистра SPI\_CFG1 задает режим передачи данных. Если RXO неактивен, то модуль работает как на прием, так и на передачу. Если RXO активен, то модуль работает только на прием. Ведущий постоянно генерирует синхросигнал для ведомого. Линия передачи остается незадействованной. У ведущего это линия MOSI, у ведомого – MISO.

Одна двунаправленная линия данных. Режим активируется, когда бит TR\_MOD регистра SPI\_CFG0 равен «1». В данном режиме модуль SPI работает только на прием, либо только на передачу, что определяется битом EN\_TX. У ведущего для приема и передачи используется линия MOSI, у ведомого MISO. MISO у ведущего или MOSI у ведомого не используется. В режиме «только прием» ведущий постоянно генерирует синхросигнал для ведомого. В режиме «только передача» необходимо игнорировать состояние принимающего буфера.

## Передача данных

В соответствии с требованиями протокола SPI, в зависимости от настройки модуля, данные передаются на линию либо по фронту, либо по спаду синхросигнала. Передача начинается при наличии данных в буфере передатчика. Данные поступают в сдвиговый регистр и далее биты слова передаются последовательно старшим или младшим вперед в зависимости от настройки BO. Установка бита TXB\_E, сообщает о том, что буфер передатчика опустел, но данные еще могут находиться в сдвиговом регистре, следовательно, будет продолжаться их передача. Об окончании передачи последнего слова данных сообщает бит NULL\_BUF. С этого момента генерация синхросигнала будет остановлена, до новой записи в буфер передатчика.

Буфер рассчитан на 8 слов. При переполнении буфера передатчика в регистре SPI\_ST устанавливается бит TXB\_OV. Записываемое слово будет утеряно. Пользователь имеет возможность очистить буфер передатчика путем записи в бит TXBS\_RST единицы.

По битам TXB\_E, NULL\_BUF, TXB\_OV может быть настроено прерывание в регистре SPI\_MSK.

## Прием данных

В соответствии с требованиями протокола SPI, в зависимости от настройки модуля, данные фиксируются по фронту, либо по спаду синхросигнала. Биты последовательно заполняют сдвиговый регистр. Затем из сдвигового регистра принятые данные поступают в буфер приемника. Когда в буфер передано хотя бы одно слово данных, в регистре SPI\_ST выставляется бит RX\_NE.

Буфер рассчитан на 8 слов данных. Пользователь может последовательно вычитать принятые данные или очистить буфер путем записи бита RXBS\_RST в регистр SPI\_CTRL во избежание переполнения. В случае, когда буфер заполнен, приход нового слова данных вызовет установку бита RXB\_OV регистра SPI\_ST. Само слово записано не будет.

В режиме «только передача» данные в буфере приемника не фиксируются.

По битам RX\_NE, RXB\_OV может быть настроено прерывание в регистре SPI\_MSK.

## Работа с входом выбора микросхемы

Модуль SPI имеет в своем составе вход выбора микросхемы –  $I_{CS}$ .

В режиме «ведомый» необходимо держать низкий логический уровень на входе  $I_{CS}$  на протяжении всей передачи. При появлении высокого логического уровня на данном входе, регистр приемника будет очищен, а работа модуля остановлена до появления требуемого значения на  $I_{CS}$ .

Пользователь может перейти в программный режим управлением входа  $I_{CS}$  с помощью бита  $SS\_CTRL$  регистра  $SPI\_CFG0$ . Далее пользователю достаточно имитировать необходимый логический уровень с помощью бита  $SSS$  регистра  $SPI\_CTRL$ . В остальном принцип работы модуля полностью соответствует работе в аппаратном режиме.

## Особенности работы в режиме «ведомый». Синхронизация и фильтрация

Модуль SPI может вести прием и передачу данных на системной частоте и близкой к ней. Однако это накладывает ограничение на работу с буфером передатчика в режиме «ведомый». Данные в буфер должны быть гарантированно записаны до выдачи последнего бита данных из сдвигового регистра передатчика в режимах с РНА «0» или до первого изменения  $SCK$  перед началом передачи нового слова в режимах с РНА «1», либо же после полного окончания передачи (актуально если отправляется несколько слов в одной посылке, часть которых дописывается в процессе отправки данных). В противном случае, слово, которое было записано в нарушении этих требований, может быть искажено. Данное ограничение можно избежать, если разрешить работу синхронизатора. В таком случае в сдвиговый регистр будут записаны новые данные или нули.

Модуль SPI оснащен синхронизатором синхросигнала. Для включения синхронизатора необходимо записать «1» в бит  $SYN$  регистра  $SPI\_CFG1$ . Синхронизатор вносит задержку в два такта системной частоты. Значит для корректного приема и передачи данных синхрочастота должна быть уменьшена и соотносится с системной как  $f_{clk}/7$ .

Модуль SPI оснащен фильтром синхрочастоты. Для включения фильтра необходимо записать «1» в бит  $FIL$  регистра  $SPI\_CFG1$ . Не рекомендуется включать фильтр независимо от синхронизатора. Фильтр может сгладить один сбой на промежутке времени, соответствующем трем тактам системной частоты. При этом частота синхросигнала также должна быть понижена.

С учетом включенного синхронизатора и фильтра частота синхросигнала должна быть меньше системной частоты минимум в 13 раз.

Также при работе на системной частоте или близкой к ней необходимо учитывать, что синхронизация  $CS$  занимает 1-2 такта частоты микроконтроллера. Поэтому после перехода  $CS$  в «0» необходимо сделать паузу 3-4 такта частоты микроконтроллера перед началом подачи  $SCK$ . В случае РНА равной «0» первый бит данных поступит на  $MISO$  по синхронизированному  $CS$  с буфера передатчика. Захват данных с буфера в сдвиговый регистр происходит на первом такте  $SCK$  по фронту или срезу в зависимости от  $POL$  (аналогично в режимах с РНА равной «1»). После приема последнего бита сообщения данные поступят в буфер приемника через 2-3 такта частоты микроконтроллера. Останется ли  $CS$  в «0» или перейдет в «1» не имеет значения.  $MISO$  перейдет в Z-состояния через 1-2 такта после перехода  $CS$  в «0». Ниже на рисунке представлен лучший вариант передачи в режиме «00», так как не потребовалось дополнительных тактов на синхронизацию.

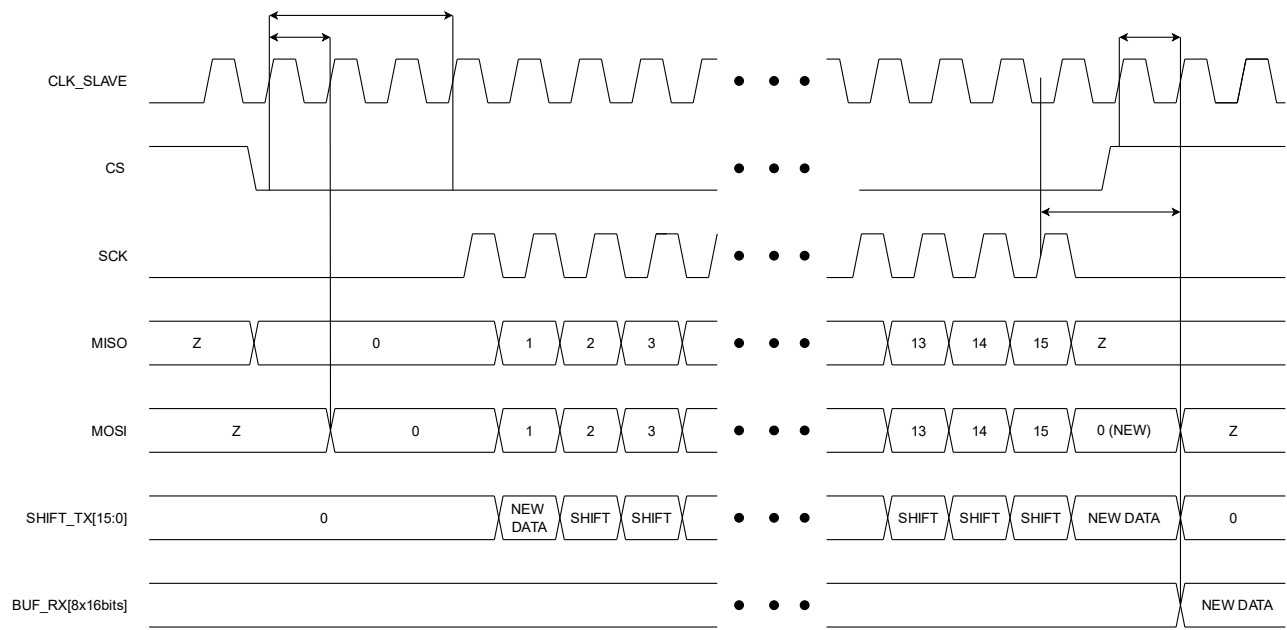


Рисунок 30. Передача данных в режиме «ведомый» на частоте микроконтроллера

Как видно на рисунке новые данные поступают в сдвиговый регистр по последнему срезу (или фронту в зависимости от POL) SCK в режимах с PHA «0». Если буфер пуст, то это будут нули. В режимах с PHA «1» новые данные поступают в сдвиговый регистр по первому фронту (или срезу) первого бита нового слова. Из этого следует, что в режимах с PHA «1» пользователь может делать паузы между передачами слов данных в рамках одной передачи SPI, во время которых ведомый микроконтроллер может обработать только что полученные данные и записать результат в буфер передатчика.

Таким образом, данные в буфере передатчика не должны меняться рядом с моментом захвата в сдвиговый регистр передатчика или в сдвиговый регистр приемника ведущего.

Регистры SPI

| №     | Аббревиатура | Доступ | Описание                                        |
|-------|--------------|--------|-------------------------------------------------|
| SPI0  |              |        |                                                 |
| 2600h | SPI0_CTRL    | RW     | Регистр управления                              |
| 2604h | SPI0_CFG0    | RW     | Регистр конфигурации 0                          |
| 2605h | SPI0_CFG1    | RW     | Регистр конфигурации 1                          |
| 2606h | SPI0_CFG2    | RW     | Регистр конфигурации 2                          |
| 2607h | SPI0_CFG3    | RW     | Регистр конфигурации 3                          |
| 2608h | SPI0_MSK     | RW     | Регистр маски                                   |
| 260Ch | SPI0_ST      | R      | Регистр статуса                                 |
| 2610h | SPI0_TX0     | W      | Регистр для записи данных в буфер передатчика 0 |
| 2611h | SPI0_TX1     | W      | Регистр для записи данных в буфер передатчика 1 |
| 2610h | SPI0_RX0     | R      | Регистр для чтения принятых данных 0            |
| 2611h | SPI0_RX1     | R      | Регистр для чтения принятых данных 1            |
| SPI1  |              |        |                                                 |
| 2700h | SPI1_CTRL    | RW     | Регистр управления                              |
| 2704h | SPI1_CFG0    | RW     | Регистр конфигурации 0                          |
| 2705h | SPI1_CFG1    | RW     | Регистр конфигурации 1                          |

Обращаем внимание, документация носит ознакомительный характер.  
При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

| №     | Аббревиатура | Доступ | Описание                                        |
|-------|--------------|--------|-------------------------------------------------|
| 2706h | SPI1_CFG2    | RW     | Регистр конфигурации 2                          |
| 2707h | SPI1_CFG3    | RW     | Регистр конфигурации 3                          |
| 2708h | SPI01_MSK    | RW     | Регистр маски                                   |
| 270Ch | SPI1_ST      | R      | Регистр статуса                                 |
| 2710h | SPI1_TX0     | W      | Регистр для записи данных в буфер передатчика 0 |
| 2711h | SPI1_TX1     | W      | Регистр для записи данных в буфер передатчика 1 |
| 2710h | SPI1_RX0     | R      | Регистр для чтения принятых данных 0            |
| 2711h | SPI1_RX1     | R      | Регистр для чтения принятых данных 1            |

**SPIx\_CTRL**

| Бит                | 7      | 6 | 5 | 4 | 3   | 2        | 1        | 0  |
|--------------------|--------|---|---|---|-----|----------|----------|----|
| Назначение         | Резерв |   |   |   | SSS | TXBS_RST | RXBS_RST | EN |
| Начальное значение | 0      |   |   |   |     |          |          |    |

**SSS (Slave Select Signal)** – состояние этого бита воздействует на устройство только при установленном бите SS\_CTRL (при программном управлении I\_CS):

1 – имитировать высокий логический уровень;

0 – имитировать низкий логический уровень.

Значение этого бита принудительно заменяет состояние входа I\_CS, которое при этом игнорируется.

**TXBS\_RST (Transmitter Buffer Set Reset)** – сброс состояния буфера передатчика:

1 – сброс состояния буфера;

0 – сброс неактивен.

Сброс записанной «1» происходит аппаратно через один такт.

**RXBS\_RST (Receiver Buffer Set Reset)** – сброс состояния буфера приемника:

1 – сброс состояния буфера;

0 – сброс неактивен.

Сброс записанной «1» происходит аппаратно через один такт.

**EN** – включение модуля SPI:

1 – модуль включен;

0 – модуль выключен.

**SPIx\_CFG0**

| Бит                | 7     | 6      | 5       | 4   | 3  | 2   | 1   | 0   |
|--------------------|-------|--------|---------|-----|----|-----|-----|-----|
| Назначение         | EN_TX | TR_MOD | SS_CTRL | DWW | BO | POL | PHA | MOD |
| Начальное значение | 0     |        |         |     |    |     |     |     |

**EN\_TX**– в режиме двунаправленного обмена данными по одной линии, разрешает или запрещает передачу данных:

1 – выход активен (только передача данных);

0 – выход неактивен (только прием данных).

**TR\_MOD**– разрешает или запрещает использование двунаправленного режима обмена данными по одной линии:

1 – режим 1-ой двунаправленной линии данных;

0 – режим 2-х однонаправленных линий данных.

**SS\_CTRL (Slave Signal Control)** – программное управление выбором устройства:

1 – программный контроль включен;

0 – программный контроль выключен.

Когда этот бит установлен, вместо уровня на входе I\_CS учитывается состояние бита SSS.

**DWW (Data Word Width)** – формат данных:

1 – 16 бит;

0 – 8 бит.

**BO (Bit Order)** – порядок передачи данных:

1 – первым передается младший значащий разряд – LSB;

0 – первым передается старший значащий разряд – MSB.

**POL** – полярность тактового сигнала:

1 – высокий уровень в режиме ожидания на выводе SCK;

0 – низкий уровень в режиме ожидания на выводе SCK.

**PHA** – фаза тактового сигнала:

1 – строб данных происходит по второму перепаду тактового сигнала;

0 – строб данных происходит по первому перепаду тактового сигнала.

**MOD** – выбор режима:

1 – режим «ведущий»;

0 – режим «ведомый».

#### SPIx\_CFG1

| Бит                | 7      | 6 | 5 | 4 | 3 | 2   | 1   | 0   |
|--------------------|--------|---|---|---|---|-----|-----|-----|
| Назначение         | Резерв |   |   |   |   | FIL | SYN | RXO |
| Начальное значение | 0      |   |   |   |   |     |     |     |

**FIL** – разрешение работы фильтра I\_SCK:

1 – I\_SCK передается через фильтр;

0 – I\_SCK передается без фильтрации.

Фильтр необходимо включать вместе с синхронизатором. Скорость сигнала I\_SCK при включенном синхронизаторе и фильтре должна быть не более  $f_{clk}/13$ .

**SYN** – разрешение работы синхронизаторов I\_SCK:

1 – I\_SCK передается через два триггера;

0 – I\_SCK передается напрямую.

Скорость сигнала I\_SCK при включенном синхронизаторе должна быть не более  $f_{clk}/7$ :

**RXO (Receive Only)** – только прием:

1 – выход отключен (только прием данных);

0 – полнодуплексный режим (передача и прием данных).

В комбинации с битом TR\_MOD задает направление передачи данных.

#### SPIx\_CFG2

| Бит                | 7  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|----|---|---|---|---|---|---|---|
| Назначение         | BR |   |   |   |   |   |   |   |
| Начальное значение | 0  |   |   |   |   |   |   |   |

**BR (Bit Rate)** – выбор скорости обмена:

$$255 - f_{clk}/510;$$

...

$$3 - f_{clk}/6;$$

$$2 - f_{clk}/4;$$

$$1 - f_{clk}/2;$$

$$0 - f_{clk}.$$

### SPIx\_CFG3

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
|--------------------|--------|---|---|---|---|---|---|------|
| Назначение         | Резерв |   |   |   |   |   |   | SOCS |
| Начальное значение | 0      |   |   |   |   |   |   |      |

**SOCS (Set Output Chip Select)** – выбор ведомого устройства:

1 – ведомое устройство выбрано (сигнал O\_CS переходит в «0»);

0 – ведомое устройство не выбрано (сигнал O\_CS в «1»).

### SPIx\_MSK

Прерывания формируется, только если модуль включен.

| Бит                | 7      | 6 | 5 | 4        | 3     | 2        | 1     | 0      |
|--------------------|--------|---|---|----------|-------|----------|-------|--------|
| Назначение         | Резерв |   |   | NULL_BUF | RX_NE | MOD_FAIL | TXB_E | RXB_OV |
| Начальное значение | 0      |   |   |          |       |          |       |        |

Для каждого из битов справедливо:

1 – данное прерывание формируется;

0 – данное прерывание не формируется.

### SPIx\_ST

| Бит                | 7      | 6        | 5    | 4      | 3     | 2        | 1     | 0      |
|--------------------|--------|----------|------|--------|-------|----------|-------|--------|
| Назначение         | Резерв | NULL_BUF | BUSY | TXB_OV | RX_NE | MOD_FAIL | TXB_E | RXB_OV |
| Тип статуса        |        | EVENT    | FLAG | EVENT  | FLAG  | EVENT    | FLAG  | EVENT  |
| Начальное значение | 0      |          |      |        |       |          | 1     | 0      |

**NULL\_BUF** – зафиксирован пустой буфер передатчика на момент окончания выдачи слова.

**BUSY** – флаг занятости. В режиме «ведомый» данный статус выставляется по первому принятому биту и сбрасывается только при выключении модуля. В режиме «ведущий» статус переходит в активный уровень только при наличии данных в сдвиговом регистре передатчика или при обмене данными. Прерывание по данному биту отсутствует.

**TXB\_OV (Transmitter Buffer Overwrite)** – переполнение буфера передатчика. Сообщает о том, что пользователь пытается записать слово данных в уже заполненный буфер передатчика. Записываемое слово будет утеряно. Прерывание по данному биту отсутствует.

**RX\_NE (Receiver Not Empty)** – в буфере приемника содержится хотя бы одно слово данных. Если пользователь сбросит буфер или вычитает все слова, то бит перейдет в неактивное состояние.

**MOD\_FAIL** – вход I\_CS был переведен в низкий логический уровень в режиме «ведущий». Данная ситуация возможна при включенном модуле только, если выбрана соответствующая альтернативная функция GPIO или выбран программный контроль линии I\_CS битом SS\_CTRL регистра SPI\_CFG0 и бит SSS регистра SPI\_CTRL в «0».

**TXB\_E (Transmitter Buffer Empty)** – буфер передатчика пуст. Статус переходит в неактивный уровень, когда в буфер будет записано хотя бы одно слово данных. Стоит учитывать, что записанное слово переместится в сдвиговый регистр, как только тот опустеет и буфер опять окажется пустым.

**RXB\_OV (Receiver Buffer Overwrite)** – буфер приемника переполнен, а значит он уже потерял одно из принимаемых слов данных. Для того, чтобы такой ситуации не происходило необходимо вычитывать принимаемые данные из буфера через регистр SPI\_RX.

#### SPIx\_TX0

При записи в данный регистр в буфер передатчика записываются старшая и младшая часть данных.

| Бит                | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------|---|---|---|---|---|---|---|
| Назначение         | DATA_TX |   |   |   |   |   |   |   |
| Начальное значение | 0       |   |   |   |   |   |   |   |

**DATA\_TX** – данные на передачу, младшая часть.

#### SPIx\_TX1

| Бит                | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------|---|---|---|---|---|---|---|
| Назначение         | DATA_TX |   |   |   |   |   |   |   |
| Начальное значение | 0       |   |   |   |   |   |   |   |

**DATA\_TX** – данные на передачу, старшая часть.

#### SPIx\_RX0

Данный регистр предназначен для чтения принятых данных из буфера приемника. Первое принятое слово данных сразу же выставляется на выходе данного регистра и меняется на следующее при чтении из данного регистра. При чтении из пустого буфера будут вычитаны нули.

| Бит                | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------|---|---|---|---|---|---|---|
| Назначение         | DATA_RX |   |   |   |   |   |   |   |
| Начальное значение | 0       |   |   |   |   |   |   |   |

**DATA\_RX** – принятые данные, младшая часть.

#### SPIx\_RX1

| Бит                | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------|---|---|---|---|---|---|---|
| Назначение         | DATA_RX |   |   |   |   |   |   |   |
| Начальное значение | 0       |   |   |   |   |   |   |   |

**DATA\_RX** – принятые данные, старшая часть.

## UART

### Общая информация

UART (универсальный асинхронный приемо-передатчик) осуществляет асинхронный полнодуплексный обмен данными по последовательным линиям RX и TX с другими устройствами UART.

Основные характеристики:

- изменение скорости передачи заданием коэффициента делителя частоты;
- изменение формата посылки. От 1 до 8 бит в слове данных, 1 или 2 стоп-бита, бит контроля четности (4 режима: odd, parity, space, mark);
- входной и выходной FIFO буферы позволяют снизить количество прерываний от UART. Количество слов в буфере (глубина буфера), при которой формируется прерывание, задается программно. Глубина буферов 8 слов;
- тестовые режимы:
  - эхо-режим;
  - режим внутренней петли;
  - режим внешней петли.
- 9-битный режим с автоматической сверкой адреса для систем из нескольких UART;
- высокоскоростной режим (четыре семпла на бит вместо шестнадцати);
- аппаратный контроль обмена через сигналы RTS и CTS;
- детектирование и формирование break-сигнала;
- тайм-аут программируемой длительности;
- возможность инвертирования логических уровней передачи сигнала.

### Структурная схема

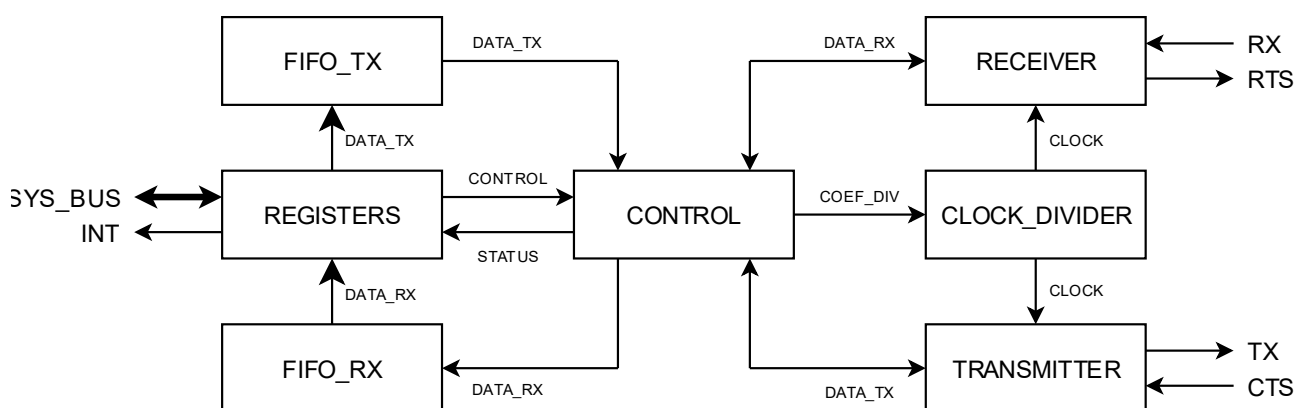


Рисунок 31. Структурная схема UART

UART состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- FIFO\_TX – буфер передатчика;
- FIFO\_RX – буфер приемника;
- CONTROL – управляющий автомат модуля UART;
- RECEIVER – приемник, реализующий функцию приема интерфейса;
- TRANSMITTER – передатчик, реализующий функцию передачи интерфейса;
- CLOCK\_DIVIDER – делитель частоты.

## Делитель частоты

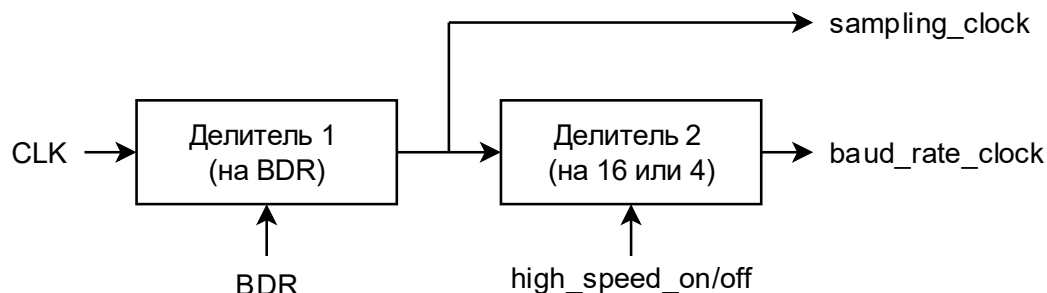


Рисунок 32. Структурная схема делителя частоты

Делитель частоты модуля UART состоит из двух делителей. Первый делитель осуществляет деление системной частоты на BDR – содержимое специальных регистров UART\_BDR0 и UART\_BDR1 (от 0 до 65535). Этот делитель определяет скорость обмена. Полученная частота (*sampling\_clock*) – это частота, с которой приемник сканирует линию RX до приема валидного стартового бита.

Второй делитель делит частоту *sampling\_clock* на 16 или на 4 в высокоскоростном режиме. С полученной частотой (*baud\_rate\_clock*) передатчик выдает биты на линию TX, а приемник сканирует линию RX в процессе приема посылки. На время передачи одного бита приходится несколько (16 или 4) семпла линии RX, для того чтобы приемник мог синхронизироваться по изменению сигнала RX в процессе работы. Частота работы взаимодействующих UART пересинхронизируется (счетчик-делитель сбрасывается) по стартовому биту и каждый раз, когда меняется уровень на линии RX в процессе приема посылки. Это позволяет справиться с «уходом» тактовых частот UART. При BDR = 0 делитель частоты не работает, соответственно не работают приемник и передатчик.

Для установления необходимой скорости обмена необходимо записать в регистры UART\_BDR0 и UART\_BDR1 соответствующий коэффициент деления, который рассчитывается по формуле:

$$BDR = \frac{f_{clk}}{16 * desired\_baud\_rate},$$

где  $f_{clk}$  – частота тактирования системы, *desired\_baud\_rate* – желаемая скорость передачи (в бодах или в бит/с).

При этом если по формуле получилось не целое число, то в результате округления реальная частота обмена будет несколько отличаться от желаемой. Реальная частота обмена (*baud\_rate\_clock*) в бодах рассчитывается по формуле:

$$baud\_rate\_clock = \frac{f_{clk}}{16 * BDR}.$$

Для самого длинного из возможных форматов посылки (12 бит), если ресинхронизации в процессе происходить не будет (то есть посылка состоит из одних единиц/нулей кроме стартового/стоп-бита), максимально допустимая разница частот, взаимодействующих UART – 3.6% (2% в высокоскоростном режиме). При меньшем формате посылки максимально допустимая разница частот соответственно увеличивается.

## Высокоскоростной режим

При выставлении бита «Высокоскоростной режим» регистра UART\_CFG0 в «1» меняется коэффициент второго делителя частоты с 16 на 4. Таким образом, в высокоскоростном режиме делитель частоты осуществляет деление на 4\*BDR, скорость передачи увеличивается в 4 раза. Формулы выше остаются справедливыми, если заменить число 16 на число 4. При этом синхронизация происходит менее точно. Этот режим рекомендуется использовать только при полном отсутствии помех на линии.

## Приемник

После сброса устройства приемник выключен, для включения необходимо записать «1» в бит RE регистра UART\_CTRL. После этого приемник начинает сканировать линию RX с частотой *sampling\_clock*, ожидая старт-бита (уровня логического нуля). При выключении приемника записью в RE «0», UART перестает сканировать линию, однако если в этот момент шла операция приема данных, она будет закончена, и данные будут помещены в буфер приемника.

UART считает старт-битом логический «0» длительностью больше, чем половина времени передачи одного бита ( $9/16$  *baud\_rate\_clock*, то есть 9 семплов *sampling\_clock*). Если уровень логического «0» держится меньшее время, это считается помехой, и UART продолжает сканировать линию на наличие старт-бита. Если логический «0» держится 9 семплов, то приемник начинает сканировать линию с меньшей в 16 раз частотой *baud\_rate\_clock* (таким образом, что сканирование на временной диаграмме происходит в предполагаемой середине каждого бита), помещая в регистр сдвига значение на линии RX. Когда вся посылка принята (длина посылки определяется форматом посылки, задающимся в регистрах UART\_CFG0 и UART\_CFG1), приемник формирует статус наличия break, ошибки четности/совпадения адреса и стоп-бита (а также соответствующие прерывания, если их формирование разрешено в регистре UART\_MSK0) и помещает их и принятое слово данных в буфер приемника. Если битов в слове данных выставлено меньше восьми в регистре UART\_CFG1, то старшие разряды данных заполняются нулями. Если буфер заполнен, формируется прерывание переполнения буфера приемника. Временная диаграмма работы приемника представлена на рисунке ниже.

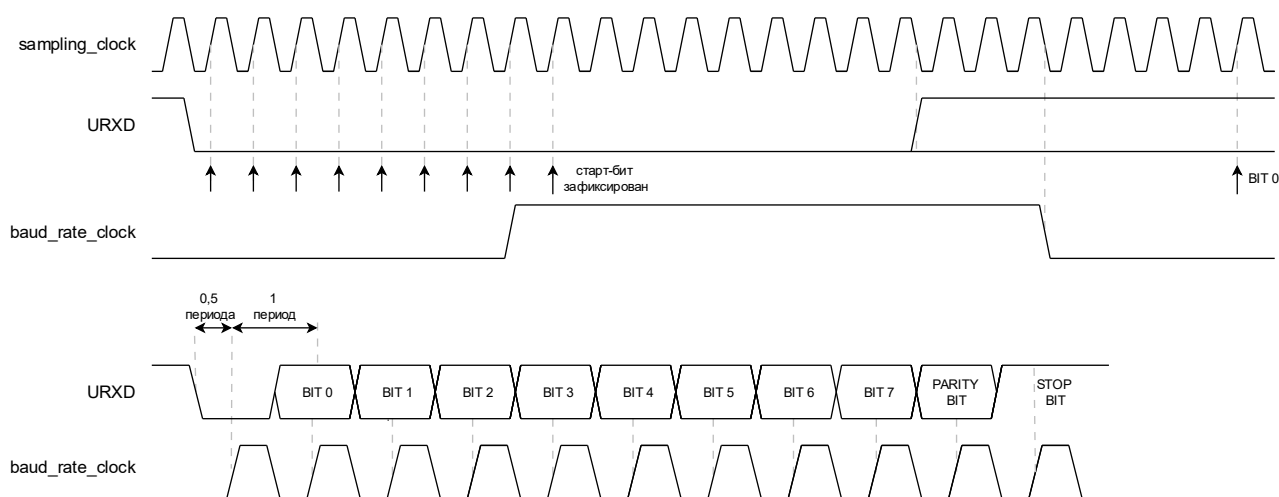


Рисунок 33. Временная диаграмма работы приемника

Между приемником и выводом RX находится простейший фильтр нижних частот: три последних значения сигнала RX (взятых с частотой *sampling\_clock*) попадают в мажоритар 2 из 3, затем результат поступает в приемник. Таким образом, одиночные помехи линии длиной меньше периода *sampling\_clock* не поступают в приемник. Помехи длиной больше периода *sampling\_clock* могут привести к ошибочной ресинхронизации и ошибке приема.

## Буфер приемника

Принятые данные попадают в буфер приемника, откуда могут быть вычитаны программно через регистр UART\_RX0. Буфер имеет 11 разрядов, 8 бит данных и 3 разряда под статус наличия break, ошибки четности/совпадения адреса и стоп-бита. Биты RBRPL, RBF, RBNE регистров UART\_ST0 и UART\_ST1 позволяют контролировать заполненность буфера. Все эти биты равны «0», если приемник выключен.

## Передатчик

После сброса устройства передатчик выключен, для включения необходимо записать «1» в бит TE в UART\_CTRL. При выключении передатчика записью в TE «0» UART сначала завершает передачу текущего слова данных.

Передатчик выдает биты слова данных, полученного из буфера передатчика, с частотой *baud\_rate\_clock* в соответствии с форматом посылки, указанным в UART\_CFG0 и UART\_CFG1. Примечание: если передатчик и буфер передатчика пусты, от момента записи в UART\_TX до начала передачи, может пройти время меньше или равное периоду *baud\_rate\_clock* (время передачи одного бита). Это происходит из-за того, что передатчик ждет следующего импульса *baud\_rate\_clock*, чтобы начать передачу посылки.

## Буфер передатчика

Передатчик берет данные для отправки из буфера передатчика, куда их можно поместить записью в регистр UART\_TX. Буфер передатчика имеет 9 разрядов, 8 из них для передаваемого слова данных. 9-ый бит используется только в 9-битном режиме как идентификатор адреса и передается вместо бита четности. Биты TBNF, TBRPL, TBE, TI регистра UART\_ST1 позволяют контролировать заполненность буфера. Все эти биты равны 0, если передатчик выключен. Если после начальной конфигурации и задания маски прерываний включить передатчик, сработают все эти прерывания.

## Прерывания

По любому из битов UART\_ST0 или UART\_ST1 можно разрешить формировать прерывание, записав «1» в соответствующий бит регистра макси прерываний UART\_MSK0 или UART\_MSK1 (расположение битов в регистрах одинаково).

Прерывание возникает, как только соответствующее событие было зафиксировано. Из-за наличия буфера может быть не очевидно, при приеме какого именно слова возникла ошибка, поэтому статус наличия break, ошибки четности/совпадения адреса и ошибки стоп-бита данного слова доступны также через младшие разряды регистра UART\_RX1.

## Таймер тайм-аута

В режиме сканирования линии приемником каждый период *baud\_rate\_clock* инкрементируется счетчик-таймера тайм-аута. Он не работает, если приемник выключен. Таймер сбрасывается в «0», как только зафиксирован валидный старт-бит и остается в «0» во время приема данных. После приема стоп-бита, он начинает считать снова. Таймер сбрасывается в «0», при записи «1» в бит RTT регистра UART\_CTRL и остается в «0» до момента записи «0» в RTT. Максимальное значение, до которого досчитывает таймер, определяется в разрядах TV регистра UART\_CFG0. Возможные значения: 2, 4, 8, 16, 32, 64, 128, 256 периодов *baud\_rate\_clock*. Досчитав до этого значения, таймер останавливается, статусный бит RTO переходит в «1» и генерируется соответствующее прерывание, если его формирование разрешено в регистре UART\_MSK0.

## Генерация и распознавание сигнала break

Сигнал break, то есть уровень логического «0» на линии на время большее времени передачи посылки, используется как индикатор серьезного сбоя в работе. Даже если никакая информация не может быть передана из-за слишком большой разности в частотах двух UART, долгий логический «0» на линии может быть правильно принят и интерпретирован приемником. Также в большинстве физических реализаций при разрыве линии сигнал RX также принимает значение логического «0».

Приемник засчитывает за сигнал break посылку, в которой старт-бит, все биты данных, бит четности (если, конечно, бит четности включен в регистре UART\_CFG1) и стоп-бит равны «0» (ошибка стоп-бита при этом не возникает). В момент приема последнего из битов бит BD переходит в «1» и генерируется соответствующее прерывание, если его формирование включено в регистре UART\_MSK0. В буфер приемника при этом записывается одно слово из восьми нулей с выставленным в «1» битом BD. Break сигнал может длиться и дольше, приемник ждет конца сигнала, то есть перехода линии RX в «1», и только потом продолжает работу.

Передатчик выдает сигнал break при записи «1» в бит SB регистра UART\_CTRL. Если запись произошла в момент передачи слова, то передача будет закончена перед выдачей сигнала break. Линия TX удерживается в состоянии логического «0» до момента записи «0» в разряд SB, таким образом продолжительность сигнала определяется программно. После окончания сигнала линия TX удерживается в «1» на 12 периодов *baud\_rate\_clock*, чтобы удаленный приемник смог корректно определить конец сигнала, предпринять необходимые действия и начать прием следующей посылки.

### Особые режимы работы

Выбор режима работы происходит записью в разряды MODE регистра UART\_CFG1.

**Эхо-режим.** В этом режиме все данные принятые UART с линии RX побитово ретранслируются на линию TX. Передатчик при переходе в этот режим работы автоматически выключается, программно включить его в регистре UART\_CTRL нельзя.

**Режим внутренней петли.** Пины UART TX и RX замыкаются внутри устройства, таким образом, выдаваемая передатчиком информация может быть принята приемником. При этом на линию TX выдается высокий уровень. Этот режим применяется для тестирования работы UART без использования линии и другого UART.

**Режим внешней петли.** Линии TX и RX замыкаются внутри устройства, таким образом, вся информация, принятая по линии RX, попадает, на линию TX. Отличие от эхо-режима в том, что эта информация не фиксируется приемником UART. Приемник и передатчик при переходе в этот режим работы автоматически выключаются, программно включить их записью в UART\_CTRL нельзя.

**9-битный режим.** 9-бит режим, или режим с аппаратным детектированием и сверкой адреса, необходим для соединения по UART одного ведущего устройства с несколькими ведомыми. Этот режим работы может быть полезен при реализации сетевых протоколов, например Modbus RTU. При работе в этом режиме формат посылки тот же, что и в обычном режиме, но вместо бита четности ведущий UART передает бит, определяющий тип данных в посылке: 1 - в посылке содержится адрес ведомого устройства, 0 - в посылке обычное слово данных (настройки бита четности PE и PT в регистрах конфигурации игнорируются в 9-бит режиме). Ведомое устройство распознает свой адрес, принимает последующие данные и/или передает ведущему устройству свои данные.

По умолчанию после включения 9-битного режима приемник не принимает слова данных, а адреса (слова данных с «1» в девятом бите) сравнивает с заданными в регистрах UART\_NBADDR0 и UART\_NBADDR1 (обычно один адрес индивидуальный для ведомого, а второй – широковещательный, общий для всех ведомых). При этом приемник сравнивает только биты адреса незамаскированные регистром UART\_NBMASK. Если адрес совпадает, то приемник посылает его в буфер вместе с 3-мя статусными битами (при чтении этого слова бит PE/AM регистра UART\_RX1 установлен в «1», показывая, что это слово – адрес), устанавливает соответствующий статус (PE/AM) в регистре UART\_ST0 и формирует прерывание по совпадению адреса, если его формирование разрешено в регистре UART\_MSK0. После этого приемник начинает принимать последующие слова данных (с «0» вместо бита четности) и продолжает делать это, пока не получит новую посылку с адресом («1» вместо бита четности). Если в формате посылки выставлено меньше восьми битов в слове, то перед сверкой идет дополнение нулями до восьми разрядов (в 9-битном режиме рекомендуется все же выставить 8 бит в слове, хотя это не обязательно). Передатчик вместо бита четности отправляет значение бита SADDR регистра UART\_CTRL.

### Аппаратный контроль обмена

Для установления аппаратного контроля обмена необходимо соединить вывод RTS (Request To Send, запрос на отправку данных, выходной сигнал) с выводом CTS того UART, с которым происходит обмен, и соединить вывод CTS (Clear To Send, запрос на получение данных, входной сигнал) с выводом RTS того UART, с которым происходит обмен. При работе в этом режиме передатчик начинает отправку посылки, только если сигнал CTS равен «0».

При работе в этом режиме при приеме стартового бита оценивается состояние буфера приемника. Если в результате приема данного слова буфер приемника будет полностью заполнен, то сигнал RTS переходит из «0» в «1», запрещая удаленному передатчику отправлять следующие данные. Чтение регистра RX переводит RTS обратно из «1» в «0», разрешая удаленному передатчику дальнейшую отправку. Если приемник выключен, RTS равно «1».

В других режимах работы сигнал RTS управляется битом RTS регистра UART\_CTRL. Этот сигнал можно использовать для управления внешним приемопередатчиком физического уровня, например RS-485.

### Регистры UART

| №            | Аббревиатура     | Доступ | Описание                                                |
|--------------|------------------|--------|---------------------------------------------------------|
| <b>UART0</b> |                  |        |                                                         |
| 2800h        | UART0_CFG0       | RW*    | Регистр конфигурации 0                                  |
| 2801h        | UART0_CFG1       | RW*    | Регистр конфигурации 1                                  |
| 2804h        | UART0_BDR0       | RW*    | Регистр настройки скорости обмена 0                     |
| 2805h        | UART0_BDR1       | RW*    | Регистр настройки скорости обмена 1                     |
| 2808h        | UART0_TXFIFOLVL0 | RW     | Регистр контроля уровня заполнения буфера передатчика 0 |
| 2809h        | UART0_TXFIFOLVL1 | RW     | Регистр контроля уровня заполнения буфера передатчика 1 |
| 280Ch        | UART0_RXFIFOLVL0 | RW     | Регистр контроля уровня заполнения буфера приемника 0   |
| 280Dh        | UART0_RXFIFOLVL1 | RW     | Регистр контроля уровня заполнения буфера приемника 1   |
| 2810h        | UART0_NBMSK      | RW     | Регистр маски адреса 9-битного режима                   |
| 2814h        | UART0_NBADDR0    | RW     | Регистр фильтра адреса в 9-битном режиме 0              |
| 2815h        | UART0_NBADDR1    | RW     | Регистр фильтра адреса в 9-битном режиме 1              |
| 2818h        | UART0_MSK0       | RW     | Регистр маски прерываний 0                              |
| 2819h        | UART0_MSK1       | RW     | Регистр маски прерываний 1                              |
| 281Ch        | UART0_CTRL       | RW     | Регистр управления                                      |
| 2820h        | UART0_TX         | W      | Регистр буфера передатчика                              |
| 2824h        | UART0_RX0        | R      | Регистр буфера приемника 0                              |
| 2825h        | UART0_RX1        | R      | Регистр буфера приемника 1                              |
| 2828h        | UART0_ST0        | R      | Регистр статусов 0                                      |
| 2829h        | UART0_ST1        | R      | Регистр статусов 1                                      |
| <b>UART1</b> |                  |        |                                                         |
| 2900h        | UART1_CFG0       | RW*    | Регистр конфигурации 0                                  |
| 2901h        | UART1_CFG1       | RW*    | Регистр конфигурации 1                                  |
| 2904h        | UART1_BDR0       | RW*    | Регистр настройки скорости обмена 0                     |
| 2905h        | UART1_BDR1       | RW*    | Регистр настройки скорости обмена 1                     |
| 2908h        | UART1_TXFIFOLVL0 | RW     | Регистр контроля уровня заполнения буфера передатчика 0 |
| 2909h        | UART1_TXFIFOLVL1 | RW     | Регистр контроля уровня заполнения буфера передатчика 1 |

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

| №     | Аббревиатура     | Доступ | Описание                                              |
|-------|------------------|--------|-------------------------------------------------------|
| 290Ch | UART1_RXFIFOLVL0 | RW     | Регистр контроля уровня заполнения буфера приемника 0 |
| 290Dh | UART1_RXFIFOLVL1 | RW     | Регистр контроля уровня заполнения буфера приемника 1 |
| 2910h | UART1_NBMSK      | RW     | Регистр маски адреса 9-битного режима                 |
| 2914h | UART1_NBADDR0    | RW     | Регистр фильтра адреса в 9-битном режиме 0            |
| 2915h | UART1_NBADDR1    | RW     | Регистр фильтра адреса в 9-битном режиме 1            |
| 2918h | UART1_MSK0       | RW     | Регистр маски прерываний 0                            |
| 2919h | UART1_MSK1       | RW     | Регистр маски прерываний 1                            |
| 291Ch | UART1_CTRL       | RW     | Регистр управления                                    |
| 2920h | UART1_TX         | W      | Регистр буфера передатчика                            |
| 2924h | UART1_RX0        | R      | Регистр буфера приемника 0                            |
| 2925h | UART1_RX1        | R      | Регистр буфера приемника 1                            |
| 2928h | UART1_ST0        | R      | Регистр статусов 0                                    |
| 2929h | UART1_ST1        | R      | Регистр статусов 1                                    |

\* – Запись в регистры UART\_CFG0, UART\_CFG1, UART\_BDR0, UART\_BDR1 **запрещена аппаратно** во время работы UART, т.к. изменение содержимого этих регистров во время работы может привести к ошибкам передачи или приема. Поэтому запись в регистры производится один раз в начале работы. Если необходимо изменить содержимое регистров, то приемник и передатчик должны быть выключены, и текущие операции приема/передачи должны закончиться. Когда эти условия выполнены, бит CRWE регистра UART\_ST1 переходит в «1», показывая, что запись в эти регистры разрешена.

#### UARTx\_CFG0

| Бит                | 7  | 6 | 5   | 4   | 3      | 2  | 1 | 0 |
|--------------------|----|---|-----|-----|--------|----|---|---|
| Назначение         | PT |   | NSB | HSE | Резерв | TV |   |   |
| Начальное значение | 0  |   |     |     |        | 4  |   |   |

**PT (Parity Type)** – тип контроля четности:

11b – mark, бит четности всегда равен «1»;

10b – space, бит четности всегда равен «0»;

01b – odd, бит четности формируется как XNOR всех битов данных в слове;

00b – even, бит четности формируется как XOR всех битов данных в слове.

**NSB (Number of Stop Bits)** – количество стоп-битов в посылке:

1 – два стоп-бита. Приемник не проверяет наличие второго стоп-бита, эта настройка влияет только на передатчик;

0 – один стоп-бит.

**HSE (High Speed Enabled)** – высокоскоростной режим:

1 – высокоскоростной режим включен. Коэффициент делителя частоты равен 4;

0 – высокоскоростной режим выключен. Коэффициент делителя частоты 16.

**TV (Time-out Value)** – длительность тайм-аута в периодах *baud\_rate\_clock*:

111b – 256;

110b – 128;

101b – 64;

100b – 32;

011b – 16;

010b – 8;

001b – 4;

000b – 2.

## UARTx\_CFG1

| Бит                | 7    | 6    | 5 | 4 | 3    | 2 | 1 | 0  |
|--------------------|------|------|---|---|------|---|---|----|
| Назначение         | INVE | MODE |   |   | CHRL |   |   | PE |
| Начальное значение | 0    |      |   |   | 7    |   |   | 0  |

**INVE (Inversion Enabled)** – инверсия сигналов линии:

1 – сигналы линии обмена (RX, TX, RTS, CTS) инвертированы. Активный уровень для RX и TX – «0», для RTS и CTS – «1»;

0 – сигналы линии обмена не инвертированы.

**MODE** – режим работы:

101b – аппаратный контроль обмена;

100b – 9-битный режим;

011b – режим внешней петли;

010b – режим внутренней петли;

001b – эхо-режим;

000b – обычный режим работы.

**CHRL (Character Length)** – размер слова данных:

111b – 8 бит;

110b – 7 бит;

101b – 6 бит;

100b – 5 бит;

011b – 4 бита;

010b – 3 бита;

001b – 2 бита;

000b – 1 бит.

**PE (Parity Enabled)** – контроль четности:

1 – контроль четности включен, бит четности передается после слова данных;

0 – контроль четности выключен, бит четности не формируется.

## UARTx\_BDR0

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | BDR |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

**BDR** – коэффициент деления, младшая часть.

## UARTx\_BDR1

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | BDR |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

**BDR** – коэффициент деления, старшая часть.

**UARTx\_TXFIFOLVL0**

| Бит                | 7       | 6 | 5 | 4 | 3         | 2 | 1 | 0 |
|--------------------|---------|---|---|---|-----------|---|---|---|
| Назначение         | Резерв* |   |   |   | TXFIFOLVL |   |   |   |
| Начальное значение | 0       |   |   |   |           |   |   |   |

\* – Не изменять начальное значение.

**TXFIFOLVL (Transmitter FIFO Buffer Level)** – заданный уровень буфера передатчика. Когда количество слов в буфере передатчика меньше или равно заданному уровню, разряд TBRPL регистра UART\_ST1 равен «1». Таким образом, в момент опустошения буфера до заданного уровня, бит TBRPL регистра UART\_ST1 переходит из «0» в «1».

**UARTx\_TXFIFOLVL1**

| Бит                | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------|---|---|---|---|---|---|---|
| Назначение         | Резерв* |   |   |   |   |   |   |   |
| Начальное значение | 0       |   |   |   |   |   |   |   |

\* – Не изменять начальное значение.

**UARTx\_RXFIFOLVL0**

| Бит                | 7       | 6 | 5 | 4 | 3         | 2 | 1 | 0 |
|--------------------|---------|---|---|---|-----------|---|---|---|
| Назначение         | Резерв* |   |   |   | RXFIFOLVL |   |   |   |
| Начальное значение | 0       |   |   |   |           |   |   |   |

\* – Не изменять начальное значение.

**RXFIFOLVL (Receiver FIFO Buffer Level)** – заданный уровень буфера приемника. Когда количество слов в буфере приемника больше или равно заданному уровню, разряд RBRPL регистра UART\_ST0 равен «1». Таким образом, в момент заполнения буфера до заданного уровня, бит RBRPL регистра UART\_ST0 переходит из «0» в «1».

**UARTx\_RXFIFOLVL1**

| Бит                | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------|---|---|---|---|---|---|---|
| Назначение         | Резерв* |   |   |   |   |   |   |   |
| Начальное значение | 0       |   |   |   |   |   |   |   |

\* – Не изменять начальное значение.

**UARTx\_NBMSK**

Содержимое этого регистра имеет эффект только в 9-битном режиме.

| Бит                | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-------|---|---|---|---|---|---|---|
| Назначение         | NBMSK |   |   |   |   |   |   |   |
| Начальное значение | 255   |   |   |   |   |   |   |   |

**NBMSK (Nine Bit Mode Address Mask)** – маска адреса в 9-битном режиме:

1 – этот разряд сравнивается при приеме адреса в 9-битном режиме;

0 – этот разряд не сравнивается при приеме адреса в 9-битном режиме.

**UARTx\_NBADDR0**

Содержимое этого регистра имеет эффект только в 9-битном режиме.

| Бит                | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|------|---|---|---|---|---|---|---|
| Назначение         | NBA2 |   |   |   |   |   |   |   |
| Начальное значение | 0    |   |   |   |   |   |   |   |

**NBA2 (Nine Bit Mode Address 2)** – адрес устройства в 9-битном режиме №2.

**UARTx\_NBADDR1**

Содержимое этого регистра имеет эффект только в 9-битном режиме.

| Бит                | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|------|---|---|---|---|---|---|---|
| Назначение         | NBA1 |   |   |   |   |   |   |   |
| Начальное значение | 0    |   |   |   |   |   |   |   |

**NBA1 (Nine Bit Mode Address 1)** – адрес устройства в 9-битном режиме №1.

**UARTx\_MSK0**

Возможно формирование прерывания по любому биту статусного регистра UART\_ST0. Расположение битов в UART\_ST0 и UART\_MSK0 аналогично.

| Бит                | 7     | 6   | 5  | 4   | 3  | 2     | 1  | 0     |
|--------------------|-------|-----|----|-----|----|-------|----|-------|
| Назначение         | RBRPL | RBF | OE | RTO | BD | PE/AM | FE | CTSIC |
| Начальное значение | 0     |     |    |     |    |       |    |       |

Для каждого из битов справедливо:

- 1 – данное прерывание формируется;
- 0 – данное прерывание не формируется.

**UARTx\_MSK1**

Возможно формирование прерывания по любому биту статусного регистра UART\_ST1. Расположение битов в UART\_ST1 и UART\_MSK1 аналогично.

| Бит                | 7      | 6    | 5    | 4    | 3     | 2   | 1  | 0    |
|--------------------|--------|------|------|------|-------|-----|----|------|
| Назначение         | Резерв | CTSI | CRWE | TBNF | TBRPL | TBE | TI | RBNE |
| Начальное значение | 0      |      |      |      |       |     |    |      |

Для каждого из битов справедливо:

- 1 – данное прерывание формируется;
- 0 – данное прерывание не формируется.

**UARTx\_CTRL**

| Бит                | 7      | 6 | 5  | 4  | 3  | 2   | 1   | 0     |
|--------------------|--------|---|----|----|----|-----|-----|-------|
| Назначение         | Резерв |   | RE | TE | SB | RTS | RTT | SADDR |
| Начальное значение | 0      |   |    |    |    |     |     |       |

**RE (Receiver Enabled):**

1 – приемник включен;

0 – приемник выключен, если в момент выключения происходил прием слова, он будет завершен перед выключением.

**TE (Transmitter Enabled):**

1 – передатчик включен;

0 – передатчик выключен, если в момент выключения происходила передача слова, она будет завершена перед выключением.

**SB (Send Break)** – устанавливает линию TX в «0». Если в этот момент происходила передача посылки, она будет завершена перед выдачей break сигнала. Сигнал заканчивается при записи «0» в этот бит, таким образом длительность сигнала break задается программно. После конца сигнала break линия TX устанавливается в «1» на 12 периодов *baud\_rate\_clock*. После этого передатчик продолжает работу.

**RTS (Request To Send)** – управляет сигналом RTS:

1 – пассивный уровень сигнала RTS (RTS установлен в «1», если инверсия сигналов линии не включена битом INVE регистра UART\_CFG1);

0 – активный уровень сигнала RTS (RTS установлен в «0», если инверсия сигналов линии не включена битом INVE регистра UART\_CFG1).

Значение этого бита не имеет эффекта в режиме работы «Аппаратный контроль обмена», так как в этом режиме UART сам управляет сигналом RTS.

**RTT (Reset Timeout Timer)** – сбросить таймер тайм-аута:

1 – таймер тайм-аута сброшен в «0» и не работает;

0 – таймер тайм-аута работает.

**SADDR (Send Address)** – отправить адрес, этот бит имеет эффект только в 9-битном режиме:

1 – все записываемые в TX слова отправляются как адреса (с «1» вместо бита четности);

0 – все записываемые в TX слова отправляются как слова данных (с «0» вместо бита четности).

**UARTx\_TX**

Запись в этот регистр запрещена, если буфер передатчика полон (бит TBNF регистра ST равен «0»). Стоит избегать записи в заполненный буфер.

| Бит                | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-------|---|---|---|---|---|---|---|
| Назначение         | CHTBT |   |   |   |   |   |   |   |
| Начальное значение | 0     |   |   |   |   |   |   |   |

**CHTBT (Character To Be Transmitted)** – слово данных для передачи. Если количество битов в слове (биты CHRL в регистре UART\_CFG1) выбрано меньше 8, то передаваемые биты – младшие. Например, если выбрано 5 битов в слове, а в CHTBT записано 00001111b, то передатчик отправит на линию TX 01111b (младшими битами вперед).

**UARTx\_RX0**

Чтение буфера приемника UART\_RX0, если тот пуст (бит RBNE регистра UART\_ST1 равен «0») приводит к выдаче «0» во всех разрядах. Стоит избегать чтения пустого буфера.

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | RCH |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

**RCH (Received Character)** – принятое слово данных. Если количество битов в слове (биты CHRL в регистре UART\_CFG1) выбрано меньше 8, то принятые биты дополняются нулями слева до 8 перед записью в RCH. Например, если выбрано 5 битов данных в слове и приемник принял 10110b, тогда RCH = 00010110b.

### UARTx\_RX1

Если необходимо считать UART\_RX1 и UART\_RX0, то сначала нужно считать данные регистра UART\_RX1, а затем UART\_RX0. Если нужно считать только принятое слово данных, то в чтение регистра UART\_RX1 необходимости нет.

| Бит                | 7      | 6 | 5 | 4 | 3 | 2  | 1     | 0  |
|--------------------|--------|---|---|---|---|----|-------|----|
| Назначение         | Резерв |   |   |   |   | BD | PE/AM | FE |
| Начальное значение | 0      |   |   |   |   |    |       |    |

**BD (Break Detected)** – это слово данных состоит только из «0», даже вместо стоп-бита, и поэтому было интерпретировано как сигнал break.

**PE/AM (Parity Error/Address Match)** – значение этого бита зависит от режима работы:

1 – обнаружена ошибка четности или это слово – адрес.

0 – ошибка четности не обнаружена или это слово – слово данные.

Во всех режимах кроме 9-битного значение бита PE/AM – ошибка четности при приеме данного слова. В 9-битном режиме, где бит четности не используется, этот бит показывает содержимое девятого бита принятого слова. Слово вместе с битами статуса отправляются в буфер приемника, только если адрес совпал при сверке.

**FE (Frame Error)** – при приеме этого слова произошла ошибка стоп-бита, то есть вместо «1» на месте старт бита принят «0».

### UARTx\_ST0

Биты BD, PE/AM, FE при приеме слова выставляются в «1» даже в случае, если в буфере приемника не было места, чтобы сохранить принятое слово.

| Бит                | 7     | 6    | 5     | 4   | 3  | 2     | 1  | 0     |
|--------------------|-------|------|-------|-----|----|-------|----|-------|
| Назначение         | RBRPL | RBF  | OE    | RTO | BD | PE/AM | FE | CTSIC |
| Тип статуса        | FLAG  | FLAG | EVENT |     |    |       |    |       |
| Начальное значение | 0     |      |       |     |    |       |    |       |

**RBRPL (Receiver Buffer Reached Preprogrammed Level)** – в буфере приемника количество слов больше или равно заданному в регистре RXFIFOLVL уровню (и приемник включен).

**RBF (Receiver Buffer Full)** – буфер приемника полон (и приемник включен). Если приемник закончит прием еще одного слова данных, это приведет к ошибке Overrun Error.

**OE (Overrun Error)** – произошла ошибка переполнения приемника, то есть стоп-бит был принят в момент, когда буфер приемника был полон, в результате чего принятое слово было утеряно.

**RTO (Receiver Timeout)** – произошел тайм-аут приемника, то есть приемник не зафиксировал старт-бита после приема последнего слова по истечению количества периодов baud\_rate\_clock, заданного в разрядах TV регистра UART\_CFG0.

**BD (Break Detected)** – обнаружен сигнал break, то есть линия RX удерживалась в «0» в течение времени, большего времени передачи посылки (иными словами, принята посылка, состоящая только из нулей, даже в стоп-бите). При этом приемник записывает в FIFO-буфер одно слово из восьми нулей, вне зависимости от того, как долго держится «0» на RX. Прием следующей посылки начнется только после перехода линии в «1», а затем приема старт-бита. Сигнал break не считается ошибкой стоп-бита.

**PE/AM (Parity Error/Address Match)** – зафиксирована ошибка четности/совпадение адреса в 9-битном режиме. Значение этого бита зависит от режима работы UART. Во всех режимах работы, кроме 9-битного режима, значение этого бита – ошибка четности, то есть содержимое бита четности принятой посылки не соответствовало биту, сформированному приемником для проверки в соответствии с выбранным типом контроля. В 9-битном режиме значение этого бита – совпадение адреса, то есть приемник принял адрес (посылка с «1» вместо бита четности), совпадающий с одним из адресов, заданных в регистрах UART\_NBADDR0 и UART\_NBADDR1. При этом сравниваются только разряды адреса немаскированные в UART\_NBMSK.

**FE (Frame Error)** – зафиксирована ошибка стоп-бита, то есть в принятой посылке значение стоп-бита было равно «0». Этот бит не устанавливается в «1», если все остальные биты слова также равны «0», так как это воспринимается как сигнал break.

**CTSIC (Clear To Send Input Change)** – произошло изменение сигнала CTS. Это событие не возникает, если передача запрещена, чтобы избежать ложной генерации этого события после сброса.

#### UARTx\_ST1

| Бит                | 7      | 6    | 5    | 4    | 3     | 2   | 1  | 0    |
|--------------------|--------|------|------|------|-------|-----|----|------|
| Назначение         | Резерв | CTSI | CRWE | TBNF | TBRPL | TBE | TI | RBNE |
| Тип статуса        | FLAG   |      |      |      |       |     |    |      |
| Начальное значение | 0      |      | 1    | 0    |       |     |    |      |

**CTSI (Clear To Send Image)** – текущее значение сигнала CTS:

1 – удаленный приемник готов принимать данные;

0 – удаленный приемник не готов принимать данные.

Значение этого бита берется из входного сигнала CTS. Если инверсия сигналов линии (бит INVE регистра UART\_CFG1) выключена, то он равен инверсии CTS, если включена, то самому сигналу CTS.

**CRWE (Config Registers Write Enable)** – запись в регистры UART\_CFG0, UART\_CFG1, UART\_BDR0, UART\_BDR1 разрешена. Этот бит равен «1» при условиях:

- приемник и передатчик выключены;
- приемник и передатчик закончили свои операции.

**TBNF (Transmitter Buffer Not Full)** – буфер передатчика не полон, в нем есть место еще как минимум для одного слова данных (и передатчик включен).

**TBRPL (Transmitter Buffer Reached Preprogrammed Level)** – в буфере передатчика осталось количество слов меньше или равно заданному в регистрах UART\_TXFIFOLVL0 и UART\_TXFIFOLVL1 уровню (и передатчик включен).

**TBE (Transmitter Buffer Empty)** – буфер передатчика пуст (и передатчик включен).

**TI (Transmitter Idle)** – буфер передатчика пуст и у него нет больше слов для отправки (и передатчик включен). Формируется до завершения последнего стоп-бита. Таким образом, если новое слово будет записано до завершения последнего стоп-бита, то новое слово начнет отправляться сразу после его завершения (при CTS в активном уровне в режиме аппаратного контроля обмена).

**RBNE (Receiver Buffer Not Empty)** – в буфере приемника есть как минимум одно неп прочитанное слово данных (и приемник включен).

## I2C

### Общая информация

Интерфейс I2C применяется для связи между собой однокристальных микроконтроллеров, ЖКИ-индикаторов, портов ввода-вывода, микросхем памяти, аналого-цифровых и цифро-аналоговых преобразователей и т.д.

Основные характеристики:

- только две линии – последовательная линия данных (SDA) и последовательная линия синхронизации (SCL);
- возможностью работы в multi-master среде;
- последовательная передача данных по 8 бит;
- скорости передачи данных: 100 кбит/с, 400 кбит/с;
- фильтрация сигналов на линиях передачи данных (SDA, SCL) от помех;
- глубина буфера приемника и передатчика 8 слов.

Все операции на шине I2C осуществляются при помощи двух проводов SDA и SCL. Как SDA, так и SCL являются двунаправленными линиями, которые необходимо подсоединить к положительному источнику питания через подтягивающий резистор 4,7 кОм. Когда шина свободна, обе линии за счет подтягивающих резисторов принимают высокий логический уровень. Выходные каскады устройств, подключенных к шине, должны иметь открытый сток или открытый коллектор.

### Структурная схема

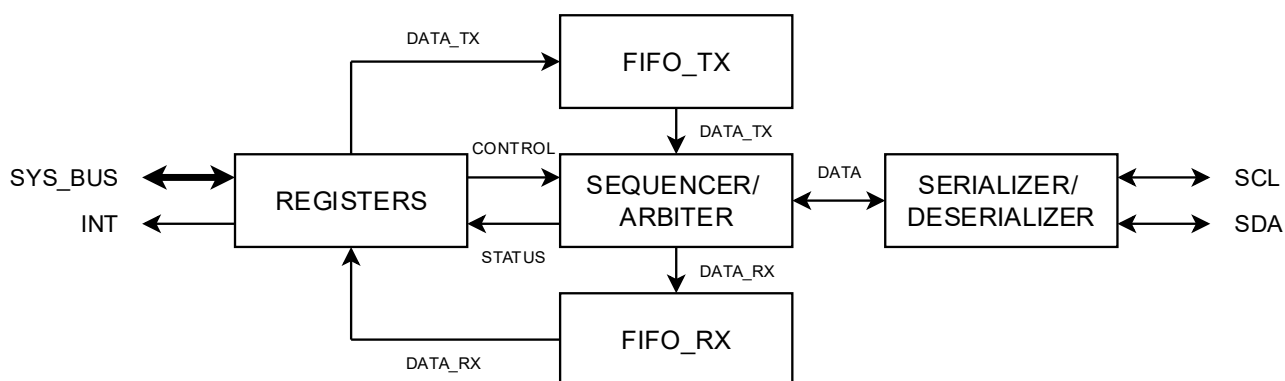


Рисунок 34. Структурная схема I2C

I2C состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- FIFO\_TX – буфер передатчика;
- FIFO\_RX – буфер приемника;
- SEQUENCER/ARBITER – управляющий автомат интерфейса;
- SERIALIZER/DESERIALIZER – блок предназначен для преобразования параллельного потока данных от управляющего автомата в последовательный, а также для преобразования последовательно потока с внешней шины, в параллельный – для управляющего автомата.

## Алгоритмы работы

Процедура передачи данных ведущим:

- в регистрах I2C\_MSK0, I2C\_MSK1, I2C\_MSK2 разрешить необходимые прерывания;
- в регистре I2C\_CFG установить биты FILT\_DEPTH, определяющие глубину фильтрации;
- в регистре I2C\_PRSC0 определить длительность высокого и низкого уровня SCL, установив младшие 8 бит PRSC;
- в регистре I2C\_PRSC1:
  - определить режим работы битом F/S, если выбран Fast-mode, то задать коэффициент заполнения битом DUTY;
  - определить длительность высокого и низкого уровня SCL, установив старшие 4 бита PRSC.
- в регистре I2C\_PRSC3 установить максимальную длительность петли обратной связи линии SCL битами TRISE;
- записать адрес с «0» в младшем бите и отправляемые данные в регистр I2C\_TXFIFO (данный этап можно провести после включения или подачи START последовательности; также данные можно дописывать как во время передачи, контролируя состояние буфера посредством регистра I2C\_TXWORDS, статуса BTF регистра I2C\_ST0 или статуса FIFO\_EMPTY\_TX регистра I2C\_ST1 и соответствующего прерывания, так и после когда бит статуса TX\_END\_EMPTY\_FIFO регистра I2C\_ST2 уже выставлен);
- в регистре I2C\_CTRL:
  - определить тип адресации битом ADDR\_MOD;
  - установить бит START;
  - включить модуль битом EN.
- по статусному биту FIFO\_EMPTY\_TX регистра I2C\_ST1 или TX\_END\_EMPTY\_FIFO регистра I2C\_ST2 установить бит STOP регистра I2C\_CTRL (если бит STOP устанавливается по флагу FIFO\_EMPTY\_TX, т.е. во время передачи последнего слова, то STOP последовательность будет развернута незамедлительно после отправки последнего байта данных и считывания его бита подтверждения, в противном случае между последним  $T_{low}/2$  и началом STOP последовательности возникает задержка, связанная с уходом на обработчик прерываний, чтением и записью регистров);
- на основании статусного бита STOP регистра I2C\_ST0 установить момент завершения подачи STOP последовательности, выключить модуль или продолжить работу.

При потере арбитража во время отправки данных в случае, если адрес записан в первом бите FIFO\_TX, пользователь имеет возможность предпринять вторую попытку после освобождения шины без повторного заполнения буфера, сбросив указатель чтения битом RST\_TX\_RDP регистра I2C\_CTRL.

Процедура приема данных ведущим:

- в регистрах I2C\_MSK0, I2C\_MSK1, I2C\_MSK2 разрешить необходимые прерывания;
- в регистре I2C\_CFG:
  - установить биты FILT\_DEPTH, определяющие глубину фильтрации;
  - определить режим работы FIFO\_RX битом EN\_OV.
- в регистре I2C\_PRSC0 определить длительность высокого и низкого уровня SCL, установив младшие 8 бит PRSC;
- в регистре I2C\_PRSC1:
  - определить режим работы битом F/S, если выбран Fast-mode, то задать коэффициент заполнения битом DUTY;
  - определить длительность высокого и низкого уровня SCL, установив старшие 4 бита PRSC.
- в регистре I2C\_PRSC3 установить максимальную длительность петли обратной связи линии SCL битами TRISE;
- записать адрес с «1» в младшем бите в регистр I2C\_TXFIFO;

- если необходимо принять более одного байта, то необходимо установить количество слов, при котором формируется соответствующий статус в регистре I2C\_RXTHRESHOLD (для завершения приема необходимо отправить NACK во время передачи последнего байта, поэтому установленное значение должно быть как минимум на одно меньше, чем принимаемых байт);
- в регистре I2C\_CTRL:
  - определить тип адресации битом ADDR\_MOD;
  - если необходимо принять более одного байта установить бит ACK;
  - установить бит START;
  - включить модуль битом EN.
- в регистре I2C\_CTRL по статусному биту RX\_THRESHOLD\_PASS регистра I2C\_ST1 или FIFO\_RX\_NOT\_EMPTY, или TX\_END\_EMPTY\_FIFO регистра I2C\_ST2 (в зависимости от количества принимаемых байт; если бит STOP устанавливается по FIFO\_RX\_NOT\_EMPTY, то между последним  $T_{low}/2$  и началом STOP последовательности возникает задержка, связанная с уходом на обработчик прерываний, чтением и записью регистров):
  - если принимается более одного байта записать «0» в бит ACK;
  - установить бит STOP.
- на основании статусного бита STOP регистра I2C\_ST0 установить момент завершения подачи STOP последовательности, считать полученные данные из регистра I2C\_RXFIFO, выключить модуль или продолжить работу.

Пользователь имеет возможность считывать буфер принятых данных и изменять регистр I2C\_RXTHRESHOLD во время передачи данных ведомым, что позволяет принимать более 8 байт.

Процедура передачи данных ведомым:

- в регистрах I2C\_MSK0, I2C\_MSK1, I2C\_MSK2 разрешить необходимые прерывания;
- в регистре I2C\_CFG:
  - установить биты FILT\_DEPTH, определяющие глубину фильтрации;
  - определить режим работы FIFO\_RX битом EN\_OV.
- задать адрес устройства в регистрах I2C\_ADDR1 и I2C\_ADDR0;
- в регистре I2C\_PRSC2 определить длительность удержания уровня SDA после заднего фронта SCL;
- записать отправляемые данные в регистр I2C\_TXFIFO;
- в регистре I2C\_CTRL:
  - определить тип адресации битом ADDR\_MOD;
  - установить бит ACK;
  - включить модуль битом EN.
- дописывать данные в регистр I2C\_TXFIFO по мере опустошения выходного буфера, отслеживая его состояние с помощью бит BTF, FIFO\_EMPTY\_TX или TX\_END\_EMPTY\_FIFO статусных регистров, а также регистра I2C\_TXWORDS.

Если на момент чтения нового слова из выходного буфера он пуст, то будут отправлены нули. Мастер должен отправить NACK после приема последнего байта данных, иначе ведомый предпримет попытку продолжить передачу.

Процедура приема данных ведомым:

- в регистрах I2C\_MSK0, I2C\_MSK1, I2C\_MSK2 разрешить необходимые прерывания;
- в регистре I2C\_CFG:
  - установить биты FILT\_DEPTH, определяющие глубину фильтрации;
  - определить режим работы FIFO\_RX битом EN\_OV.
- задать адрес устройства в регистрах I2C\_ADDR1 и I2C\_ADDR0;

- в регистре I2C\_PRSC2 определить длительность удержания уровня SDA после заднего фронта SCL;
- в регистре I2C\_RXTHRESHOLD установить количество слов, при котором формируется соответствующий статус;
- в регистре I2C\_CTRL:
  - определить тип адресации битом ADDR\_MOD;
  - установить бит ACK;
  - включить модуль битом EN.
- считывать данные и управлять битом ACK регистра I2C\_CTRL по мере формирования соответствующих статусов.

### Статусы и прерывания

В отличие от остальных модулей в I2C:

- прерывание по биту-флагу не формируется, если бит уже находится в «1», и в этот момент разрешается прерывание;
- прерывание по биту-событию не формируется от последующих событий после перехода бита в «1», пока он не будет сброшен чтением обратно в «0».

Остальные принципы работы, описанные в разделе «Типы статусов и прерывания», остаются неизменными.

### Регистры I2C

| №     | Аббревиатура    | Доступ | Описание                                                                                                               |
|-------|-----------------|--------|------------------------------------------------------------------------------------------------------------------------|
| 2A00h | I2C_CFG         | RW     | Регистр конфигурации                                                                                                   |
| 2A04h | I2C_CTRL        | RW     | Регистр управления                                                                                                     |
| 2A08h | I2C_ST0         | R      | Регистр статуса 0                                                                                                      |
| 2A09h | I2C_ST1         | R      | Регистр статуса 1                                                                                                      |
| 2A0Ah | I2C_ST2         | R      | Регистр статуса 2                                                                                                      |
| 2A0Ch | I2C_ADDR0       | RW     | Регистр адреса приемника 0                                                                                             |
| 2A0Dh | I2C_ADDR1       | RW     | Регистр адреса приемника 1                                                                                             |
| 2A10h | I2C_PRSC0       | RW     | Предделитель 0                                                                                                         |
| 2A11h | I2C_PRSC1       | RW     | Предделитель 1                                                                                                         |
| 2A12h | I2C_PRSC2       | RW     | Предделитель 2                                                                                                         |
| 2A13h | I2C_PRSC3       | RW     | Предделитель 3                                                                                                         |
| 2A14h | I2C_MSK0        | RW     | Регистр маски прерываний 0                                                                                             |
| 2A15h | I2C_MSK1        | RW     | Регистр маски прерываний 1                                                                                             |
| 2A16h | I2C_MSK2        | RW     | Регистр маски прерываний 2                                                                                             |
| 2A18h | I2C_TXFIFO      | RW     | Буфер данных на передачу                                                                                               |
| 2A1Ch | I2C_RXFIFO      | R      | Буфер принятых данных                                                                                                  |
| 2A20h | I2C_TXWORDS     | R      | Количество непрочитанных слов в буфере передачи                                                                        |
| 2A24h | I2C_RXTHRESHOLD | RW     | Указывает количество непрочитанных слов в буфере приема, при котором формируется соответствующий признак в регистре ST |

**I2C\_CFG**

| Бит                | 7      | 6 | 5          | 4 | 3 | 2 | 1 | 0 |       |
|--------------------|--------|---|------------|---|---|---|---|---|-------|
| Назначение         | Резерв |   | FILT_DEPTH |   |   |   |   |   | EN_OV |
| Начальное значение | 0      |   |            |   |   |   |   |   |       |

**FILT\_DEPTH** – настройка глубины фильтров для входных сигналов SDA, SCL. Выбирается следующим образом:

$$FILT\_DEPTH > \frac{T_{noise}}{T_{clk}},$$

где  $T_{noise}$  – максимальная длительность помехи, которую необходимо отфильтровать,  $T_{clk}$  – период частоты тактирования системы. Соответственно при максимальном значении битов FILT\_DEPTH может быть отфильтрована помеха длительностью менее  $31 T_{clk}$ . При значении «0» фильтр выключен.

**EN\_OV** – при переполнении буфера FIFO приемника разрешает или запрещает перезаписывать данные:

- 1 – перезапись данных разрешена;
- 0 – перезапись данных запрещена.

При приеме в режиме «ведущий», когда переполнение разрешено, контроллер может запрашивать от ведомого новые слова и записывать их поверх старых. При возникновении переполнения поднимается признак FIFO\_RX\_OV. Если переполнение запрещено ведущий остановит прием после появления признака FIFO\_RX\_FULL.

При приеме в режиме «ведомый», когда переполнение разрешено, можно записывать новые слова поверх старых. При возникновении переполнения установится признак FIFO\_RX\_OV. Если переполнение запрещено, ведомый не будет записывать новые слова поверх старых, однако при приеме нового слова и наличии признака FIFO\_RX\_FULL установится признак FIFO\_RX\_OV.

**I2C\_CTRL**

| Бит                | 7            | 6          | 5            | 4        | 3   | 2    | 1     | 0  |
|--------------------|--------------|------------|--------------|----------|-----|------|-------|----|
| Назначение         | RST_RX_PNTRS | RST_TX_RDP | RST_TX_PNTRS | ADDR_MOD | ACK | STOP | START | EN |
| Начальное значение | 0            |            |              |          |     |      |       |    |

**RST\_RX\_PNTRS** – запись единицы приводит к сбросу указателей FIFO\_RX.

**RST\_TX\_RDP (Reset Transmitter Read Pointer)** – запись единицы приводит к сбросу указателя чтения FIFO\_TX.

**RST\_TX\_PNTRS** – запись единицы приводит к сбросу указателей FIFO\_TX.

**ADDR\_MOD** – тип адресации:

- 1 – 10-ти битная адресация;
- 0 – 7-ми битная адресация.

**ACK** – разрешение подтверждение после приема байта (адреса или данных):

- 1 – отправлять подтверждение;
- 0 – не отправлять подтверждение.

При переводе бита в «0» модуль запишет последний байт в буфер приема и подаст на линию NACK.

**STOP** – сформировать STOP последовательность:

- 1 – в режиме «ведущий» формируется STOP;
- 0 – STOP не формируется.

Бит переводится в «0» устройством после STOP.

**START** – сформировать START последовательность.:

1 – в режиме «ведущий» формируется START;

0 – START не формируется.

Бит переводится в «0» устройством после START.

**EN** – разрешение работы модуля:

1 – модуль включен;

0 – модуль выключен.

## I2C\_ST0

| Бит                | 7          | 6            | 5        | 4         | 3           | 2     | 1     | 0     |
|--------------------|------------|--------------|----------|-----------|-------------|-------|-------|-------|
| Назначение         | FIFO_RX_OV | FIFO_RX_FULL | ARB_LOST | BUS_ERROR | ACK_FAILURE | BTF   | STOP  | START |
| Тип статуса        | FLAG       |              | EVENT    | EVENT     | EVENT       | EVENT | EVENT | EVENT |
| Начальное значение | 0          |              |          |           |             |       |       |       |

**FIFO\_RX\_OV** – FIFO\_RX переполнен.

**FIFO\_RX\_FULL** – FIFO\_RX полностью заполнен. При наличии FIFO\_RX\_OV обнуляется, контролироваться не должен.

**ARB\_LOST** – признак потери арбитража (только для режима «ведущий»). При потере арбитража в процессе передачи слова данных ведущий продолжает формировать импульсы SCL для всех 8 бит данных, затем освобождает линию SCL, и после этого формирует статус ARB\_LOST.

**BUS\_ERROR** – признак несвоевременного события START или STOP. Признак появляется, если зафиксирован перепад линии SDA на фоне единичного уровня SCL во время передачи байта данных или бита подтверждения.

**ACK\_FAILURE** – не было подтверждения после передачи байта данных/адреса.

**BTF (Byte Transfer Completed)** – обмен байтом завершен:

1 – обмен байтом завершился успешно;

0 – обмена байтом не было/завершился не успешно.

Переводится в единицу, если:

- при приеме, когда принят новый байт и отправлен ACK/NACK;
- при передаче, если байт был передан и получен ACK.

Примечание: если при передаче байта принят NACK, BTF не переводится в единицу.

**STOP** – в режиме «ведущий» сформирован STOP, в режиме «ведомый» обнаружен STOP.

**START** – в режиме «ведущий» сформирован START, в режиме «ведомый» обнаружен START.

## I2C\_ST1

| Бит                | 7                 | 6    | 5         | 4      | 3      | 2             | 1            | 0                 |
|--------------------|-------------------|------|-----------|--------|--------|---------------|--------------|-------------------|
| Назначение         | RX_THRESHOLD_PASS | MODE | BUS_CLEAR | SLV_RX | SLV_TX | FIFO_EMPTY_TX | FIFO_FULL_TX | FIFO_RX_NOT_EMPTY |
| Тип статуса        | FLAG              |      |           |        |        |               |              |                   |
| Начальное значение | 1                 | 0    | 1         | 0      | 0      | 1             | 0            | 0                 |

**RX\_THRESHOLD\_PASS** – количество непрочитанных слов в буфере приема больше либо равно значению регистра I2C\_RX\_THRESHOLD.

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

**MODE** – режим:

1 – блок находится в режиме «ведущий»;

0 – блок находится в режиме «ведомый».

**BUS\_CLEAR** – признак того, что на линии не идет обмен. Линия будет считаться занятой, если зафиксирован низкий уровень на линиях SDA или SCL, и свободной – если произошло событие STOP.

**SLV\_RX** – признак, что в режиме «ведомый» устройство адресовано на запись.

**SLV\_TX** – признак, что в режиме «ведомый» устройство адресовано на чтение.

**FIFO\_EMPTY\_TX** – FIFO\_TX пуст. Возникает перед передачей последнего слова данных в момент чтения из буфера (за такт системной частоты до выдачи первого бита на линии SDA в момент времени  $T_{low}/2$ ).

**FIFO\_FULL\_TX** – FIFO\_TX полностью заполнен.

**FIFO\_RX\_NOT\_EMPTY** – FIFO\_RX не пуст. При наличии FIFO\_RX\_OV обнуляется, контролироваться не должен.

#### I2C\_ST2

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0                 |
|--------------------|--------|---|---|---|---|---|---|-------------------|
| Назначение         | Резерв |   |   |   |   |   |   | TX_END_EMPTY_FIFO |
| Тип статуса        |        |   |   |   |   |   |   | EVENT             |
| Начальное значение | 0      |   |   |   |   |   |   |                   |

**TX\_END\_EMPTY\_FIFO** – признак того, что в момент завершения передачи слова буфер передачи был пуст. Возникает в момент завершения приема ACK либо NACK после передачи слова (за такт системной частоты до  $T_{low}/2$ ).

#### I2C\_ADDR0

| Бит                | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|------|---|---|---|---|---|---|---|
| Назначение         | ADDR |   |   |   |   |   |   |   |
| Начальное значение | 0    |   |   |   |   |   |   |   |

**ADDR** – адрес в режиме «ведомый», младшая часть.

#### I2C\_ADDR1

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1    | 0 |
|--------------------|--------|---|---|---|---|---|------|---|
| Назначение         | Резерв |   |   |   |   |   | ADDR |   |
| Начальное значение | 0      |   |   |   |   |   |      |   |

**ADDR** – адрес в режиме «ведомый», старшая часть.

#### I2C\_PRSC0

| Бит                | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|------|---|---|---|---|---|---|---|
| Назначение         | PRSC |   |   |   |   |   |   |   |
| Начальное значение | 0    |   |   |   |   |   |   |   |

**PRSC** – предделитель, младшая часть.

## I2C\_PRSC1

| Бит                | 7   | 6    | 5      | 4 | 3    | 2 | 1 | 0 |
|--------------------|-----|------|--------|---|------|---|---|---|
| Назначение         | F/S | DUTY | Резерв |   | PRSC |   |   |   |
| Начальное значение | 0   |      |        |   |      |   |   |   |

**F/S** – режим:

1 – Fast-mode;

0 – Standard-mode.

**DUTY** – коэффициент заполнения для Fast-mode:

$$1 - \frac{T_{high}}{T_{low}} = \frac{9}{16};$$

$$0 - \frac{T_{high}}{T_{low}} = \frac{1}{2}.$$

**PRSC** – предделитель, определяет длительность высокого ( $T_{high}$ ) и низкого ( $T_{low}$ ) уровня SCL в режиме «ведущий», старшая часть.

При Fast-mode и DUTY = 1:

$$T_{high} = 9 * PRSC * T_{clk};$$

$$T_{low} = 16 * PRSC * T_{clk}.$$

При Fast-mode и DUTY = 0:

$$T_{high} = PRSC * T_{clk};$$

$$T_{low} = 2 * PRSC * T_{clk}.$$

При Standard-mode:

$$T_{high} = 2 * PRSC * T_{clk};$$

$$T_{low} = 2 * PRSC * T_{clk}.$$

При установке битов PRSC необходимо, чтобы общее время, затраченное на обратную связь при спаде SCL, не превышало  $T_{low}/2$ . Для этого достаточно выполнения следующего неравенства:

$$\frac{T_{low}}{2 * T_{clk}} > \left\lceil \frac{T_{fall}}{T_{clk}} \right\rceil + FILT\_DEPTH + 2,$$

где  $T_{fall}$  – максимальная длительность заднего фронта SCL на входе устройства. Из этого следует, что при Standard-mode или Fast-mode с DUTY в «0» и  $T_{fall} < T_{clk}$  минимальное значение битов PRSC равно 3 при выключенном фильтре (биты FILT\_DEPTH в «0»), однако рекомендуется округлять отношение длительности заднего фронта и периода синхросигнала в большую сторону.

Смена значения на линии SDA в режиме «ведущий», если он передает, происходит в момент времени  $T_{low}/2$ .

## I2C\_PRSC2

| Бит                | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------|---|---|---|---|---|---|---|
| Назначение         | SLV_HLD |   |   |   |   |   |   |   |
| Начальное значение | 0       |   |   |   |   |   |   |   |

**SLV\_HLD** – длительность удержания уровня SDA, младшая часть.

**I2C\_PRSC3**

| Бит                | 7      | 6     | 5 | 4 | 3 | 2 | 1       | 0 |
|--------------------|--------|-------|---|---|---|---|---------|---|
| Назначение         | Резерв | TRISE |   |   |   |   | SLV_HLD |   |
| Начальное значение | 0      |       |   |   |   |   |         |   |

**TRISE** – максимальная длительность петли обратной связи линии SCL в режиме «ведущий». Используется с целью установить корректную скважность на линии SCL, которая не будет зависеть от длительности переднего фронта SCL на входе устройства и глубины фильтрации. Значение должно быть установлено исходя из максимального времени нарастания переднего фронта SCL, выраженного в периодах частоты тактирования системы:

$$TRISE = \left\lceil \frac{T_{rise}}{T_{clk}} \right\rceil + FILT\_DEPTH + 3,$$

где  $T_{rise}$  – максимальная длительность переднего фронта SCL на входе устройства. Также следует учитывать:

$$TRISE \leq \frac{T_{high}}{T_{clk}}.$$

**SLV\_HLD** – длительность удержания уровня SDA, старшая часть. В режиме «ведомый» через поле SLV\_HLD определяется длительность удержания уровня SDA от заднего фронта SCL до выдачи бита ведомым I2C:

$$T_{hold} = T_{fall} + (FILT\_DEPTH + SLV\_HLD + 6) * T_{clk}.$$

**I2C\_MSK0**

Возможно формирование прерывания по любому биту статусного регистра I2C\_ST0. Расположение битов в I2C\_ST0 и I2C\_MSK0 аналогично.

| Бит                | 7          | 6            | 5        | 4         | 3           | 2   | 1    | 0     |
|--------------------|------------|--------------|----------|-----------|-------------|-----|------|-------|
| Назначение         | FIFO_RX_OV | FIFO_RX_FULL | ARB_LOST | BUS_ERROR | ACK_FAILURE | BTF | STOP | START |
| Начальное значение | 0          |              |          |           |             |     |      |       |

Для каждого из битов справедливо:

- 1 – данное прерывание формируется;
- 0 – данное прерывание не формируется.

**I2C\_MSK1**

Возможно формирование прерывания по любому биту статусного регистра I2C\_ST1. Расположение битов в I2C\_ST1 и I2C\_MSK1 аналогично.

| Бит                | 7                 | 6    | 5         | 4      | 3      | 2             | 1            | 0                 |
|--------------------|-------------------|------|-----------|--------|--------|---------------|--------------|-------------------|
| Назначение         | RX_THRESHOLD_PASS | MODE | BUS_CLEAR | SLV_RX | SLV_TX | FIFO_EMPTY_TX | FIFO_FULL_TX | FIFO_RX_NOT_EMPTY |
| Начальное значение | 0                 |      |           |        |        |               |              |                   |

Для каждого из битов справедливо:

- 1 – данное прерывание формируется;
- 0 – данное прерывание не формируется.

**I2C\_MSK2**

Расположение статусного бита аналогично I2C\_ST2.

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0                 |
|--------------------|--------|---|---|---|---|---|---|-------------------|
| Назначение         | Резерв |   |   |   |   |   |   | TX_END_EMPTY_FIFO |
| Начальное значение | 0      |   |   |   |   |   |   |                   |

**TX\_END\_EMPTY\_FIFO** - разрешить прерывание по статусу TX\_END\_EMPTY\_FIFO:

1 – данное прерывание формируется;

0 – данное прерывание не формируется.

**I2C\_TXFIFO**

| Бит                | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------|---|---|---|---|---|---|---|
| Назначение         | TX_FIFO |   |   |   |   |   |   |   |
| Начальное значение | 0       |   |   |   |   |   |   |   |

**TX\_FIFO** – буфер передаваемых данных. При передаче адреса в режиме «ведущий» пользователь должен самостоятельно задать R/W битом 0.

**I2C\_RXFIFO**

| Бит                | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---------|---|---|---|---|---|---|---|
| Назначение         | RX_FIFO |   |   |   |   |   |   |   |
| Начальное значение | 0       |   |   |   |   |   |   |   |

**RX\_FIFO** – буфер принятых данных.

**I2C\_TXWORDS**

| Бит                | 7      | 6 | 5 | 4 | 3        | 2 | 1 | 0 |
|--------------------|--------|---|---|---|----------|---|---|---|
| Назначение         | Резерв |   |   |   | TX_WORDS |   |   |   |
| Начальное значение | 0      |   |   |   |          |   |   |   |

**TX\_WORDS** – количество непрочитанных слов в буфере передачи (от 0 до 8).

**I2C\_RXTHRESHOLD**

| Бит                | 7      | 6 | 5 | 4 | 3        | 2 | 1 | 0 |
|--------------------|--------|---|---|---|----------|---|---|---|
| Назначение         | Резерв |   |   |   | RX_THRES |   |   |   |
| Начальное значение | 0      |   |   |   |          |   |   |   |

**RX\_THRES** – количество непрочитанных слов в буфере приема (от 0 до 8), при котором формируется соответствующий признак в регистре статуса.

## 1-Wire

### Общая информация

1-Wire или OWI – это интерфейс, который представляет собой двунаправленную шину связи для устройств с низкоскоростной передачей данных, в которой данные передаются по цепи питания. Используется всего два провода, один общий, а второй для питания и данных.

Основные характеристики модуля:

- работа в режиме «ведущего»;
- минимальная частота тактирования 1 МГц;
- поддержка стандартной и повышенной скоростей обмена;
- поддержка набора стандартных команд;
- расширенный набор статусов и прерываний.

### Структурная схема

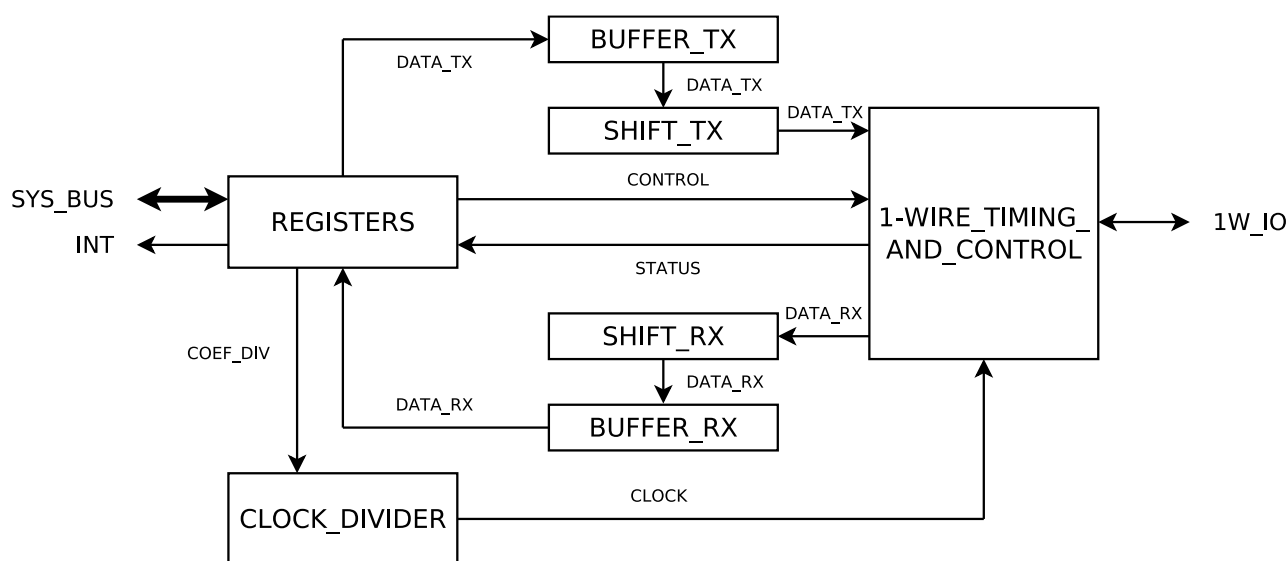


Рисунок 35. Структурная схема 1-Wire

OWI состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- BUFFER\_TX – буфер передатчика;
- SHIFT\_TX – сдвиговый регистр передатчика;
- BUFFER\_RX – буфер приемника;
- SHIFT\_RX – сдвиговый регистр приемника;
- CLOCK\_DIVIDER – делитель частоты;
- 1-WIRE\_TIMING\_AND\_CONTROL – основной блок интерфейса, содержащий управляющие автоматы и отвечающий за корректное функционирование.

## Сигнализация шины 1-Wire

Модуль OWI поддерживает следующие формы сигналов: цикл сброса/обнаружения присутствия, тайм-слот записи нуля, тайм-слот записи единицы и тайм-слот чтения.

Цикл сброса/обнаружения присутствия состоит из двух этапов: в течение времени  $t_{RSTL}$  ведущий подтягивает линию к низкому уровню, далее начинается окно  $t_{RSTH}$ , в течение которого после задержки  $t_{PDH}$  ведомое устройство переводит линию к низкому уровню на промежуток времени  $t_{PDL}$ , чтобы сообщить о своем присутствии. Для проверки импульса присутствия модуль OWI сначала ожидает  $t_{PDW}$ , а затем в течение  $t_{PDSW}$  производит выборку напряжения вывода 1W\_IO.

Модуль OWI позволяет включить маскирование импульса присутствия, генерируемого ведомым устройством, установив бит PPM регистра OWI\_CTRL. При установленном бите PPM модуль OWI генерирует импульс присутствия (пунктирная линия), который начинается с конца  $t_{PPMS}$  и длится до конца  $t_{PPME}$ .

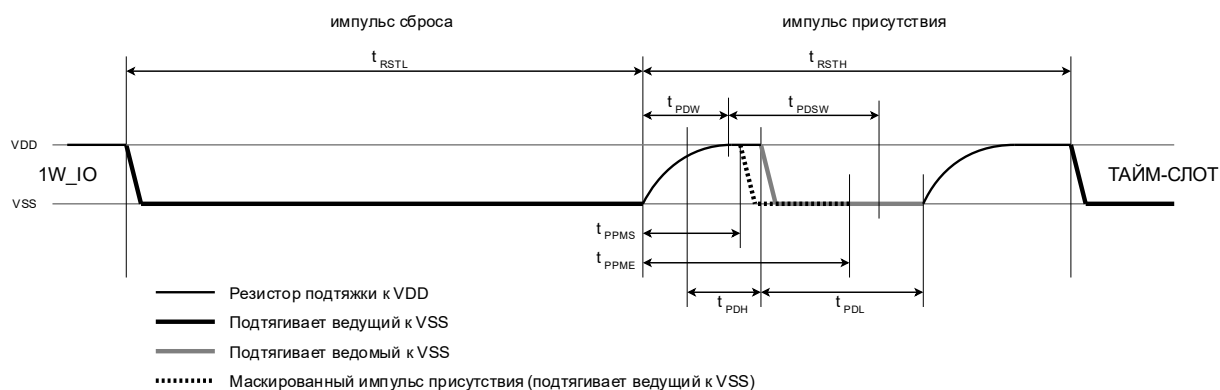


Рисунок 36. Цикл сброса/обнаружения присутствия

Тайм-слот записи нуля состоит из двух этапов: ведущий подтягивает линию в течение  $t_{W0L}$ , далее следует время восстановления  $t_{REC0}$ . Сумма времени  $t_{W0L}$  и  $t_{REC0}$  равна длительности тайм-слота  $t_{SLOT}$ .

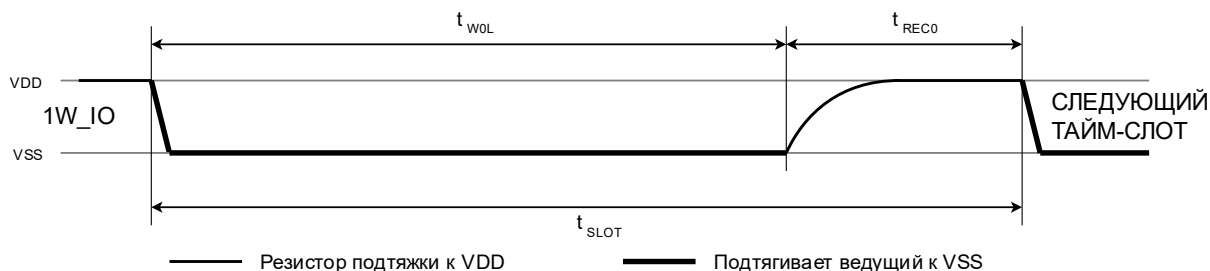


Рисунок 37. Тайм-слот записи нуля

Тайм-слот записи единицы также состоит из двух этапов: ведущий подтягивает линию в течение  $t_{W1L}$ , затем ожидается завершение тайм-слота.

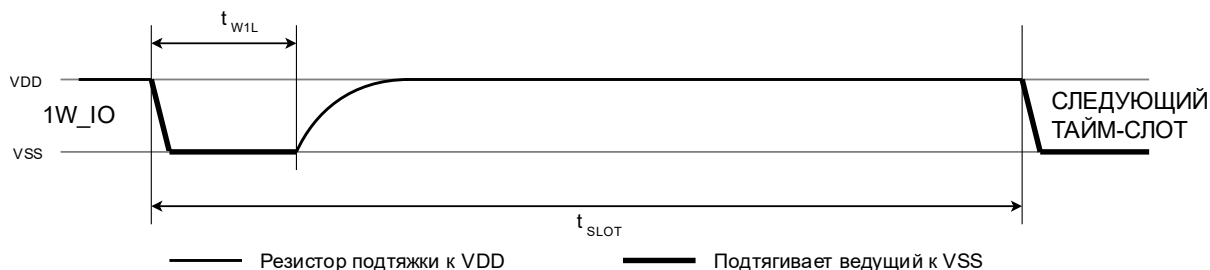


Рисунок 38. Тайм-слот записи единицы

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФС.431260.012-003Д16

Тайм-слот чтения состоит из трех этапов: ведущий подтягивает линию к низкому уровню в течение  $t_{W1L}$ , ведомый устанавливает уровень на линии, после чего ожидается завершение тайм-слота. В течение  $t_{MRS}$  модуль OWI осуществляет выборку напряжения вывода  $1W\_IO$ . Следует иметь в виду, что ведомое устройство, отвечая нулем, начинает устанавливать низкий уровень на линии  $1W\_IO$  до того, как истечет время  $t_{W1L}$ .

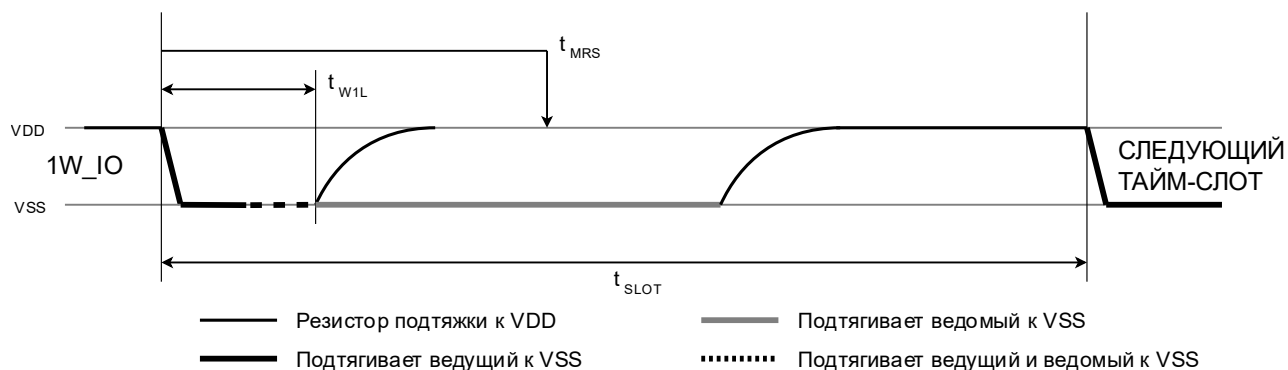


Рисунок 39. Тайм-слот чтения

Таблица 15. Временные характеристики вывода  $1W\_IO^*$ 

| Обозначение | Параметр, единица измерения                                                          | Условия                                                                                                  | Мин. | Макс. |
|-------------|--------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|------|-------|
| $T_{OWI}$   | Внутренний период синхросигнала модуля, мкс                                          | Для достижения данного периода делитель входной системной частоты настроить в соответствии с таблицей 13 | 0,8  | 1     |
| $t_{SLOT}$  | Длительность тайм-слота, мкс                                                         | Стандартная скорость передачи, LLM = 0                                                                   | 56   | 70    |
|             |                                                                                      | Стандартная скорость передачи, LLM = 1                                                                   | 64   | 80    |
|             |                                                                                      | Ускоренная передача                                                                                      | 8    | 10    |
| $t_{WOL}$   | Время низкого уровня тайм-слота записи нуля (Write-zero Low time), мкс               | Стандартная скорость передачи                                                                            | 48   | 60    |
|             |                                                                                      | Ускоренная передача                                                                                      | 6,4  | 8     |
| $t_{W1L}$   | Время низкого уровня тайм-слота записи единицы/чтения (Write-one/read Low time), мкс | Стандартная скорость передачи, LLM = 0                                                                   | 4,8  | 6     |
|             |                                                                                      | Стандартная скорость передачи, LLM = 1                                                                   | 6,4  | 8     |
|             |                                                                                      | Ускоренная передача                                                                                      | 0,8  | 1     |
| $t_{MRS}$   | Время выборки тайм-слота чтения (Master Read Sample time), мкс                       | Стандартная скорость передачи, LLM = 0                                                                   | 12   | 15    |
|             |                                                                                      | Стандартная скорость передачи, LLM = 1                                                                   | 19,2 | 24    |
|             |                                                                                      | Ускоренная передача                                                                                      | 1,6  | 2     |
| $t_{RECO}$  | Время восстановления уровня тайм-слота записи нуля (Write-zero Recovery time), мкс   | Стандартная скорость передачи, LLM = 0                                                                   | 8    | 10    |
|             |                                                                                      | Стандартная скорость передачи, LLM = 1                                                                   | 16   | 20    |
|             |                                                                                      | Ускоренная передача                                                                                      | 1,6  | 2     |
| $t_{RSTL}$  | Время низкого уровня цикла сброса (Reset Low time), мкс                              | Стандартная скорость передачи                                                                            | 480  | 600   |
|             |                                                                                      | Ускоренная передача                                                                                      | 56   | 70    |
| $t_{RSTH}$  | Время высокого уровня цикла сброса (Reset High time), мкс                            | Стандартная скорость передачи                                                                            | 384  | 480   |
|             |                                                                                      | Ускоренная передача                                                                                      | 46,4 | 58    |

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

| Обозначение | Параметр, единица измерения                                                      | Условия                                | Мин. | Макс. |
|-------------|----------------------------------------------------------------------------------|----------------------------------------|------|-------|
| $t_{PDW}$   | Время ожидания цикла присутствия (Presence-Detect Wait time), мкс                | Стандартная скорость передачи          | 8    | 10    |
|             |                                                                                  | Ускоренная передача                    | 1,6  | 2     |
| $t_{PDSW}$  | Время выборки цикла присутствия (Presence-Detect Sample Window), мкс             | Стандартная скорость передачи, LLM = 0 | 48,8 | 61    |
|             |                                                                                  | Стандартная скорость передачи, LLM = 1 | 60,8 | 76    |
|             |                                                                                  | Ускоренная передача                    | 6,4  | 8     |
| $t_{PPMS}$  | Время начала маскированного цикла присутствия (Presence-Pulse Mask Start), мкс   | Стандартная скорость передачи          | 16   | 20    |
| $t_{PPME}$  | Время завершения маскированного цикла присутствия (Presence-Pulse Mask End), мкс | Стандартная скорость передачи          | 72   | 90    |

\* – Временные характеристики зависят от задержек на универсальных портах ввода-вывода. Если порты работают медленно, то эти характеристики изменятся соответствующим образом.

### Предварительная настройка

Перед началом работы с модулем OWI настроить предделитель и делитель частоты и включить систему тактирования в регистре OWI\_PRCR. При необходимости использовать порывания задать соответствующие значение в регистр маски OWI\_MSK. После этого модуль готов к работе на стандартной скорости и в байтовом режиме.

### Генерация цикла сброса/обнаружения присутствия

Для генерации цикла сброса/обнаружения присутствия записать код 01h в регистр OWI\_CFG. Если включено прерывание по биту PD в регистре OWI\_MSK, то микроконтроллер может продолжить работу до завершения цикла. После завершения прочитать бит PDR регистра OWI\_ST, чтобы установить, был ли обнаружен импульс присутствия.

### Передача байта

Чтобы передать байт по шине 1-Wire необходимо записать его по адресу OWI\_BUF в буфер передатчика. Затем данные переходят в сдвиговый регистр передатчика, где они последовательно отправляются на шину младшим битом вперед (LSB). Соответственно новый байт данных может быть записан в буфер передатчика. Как только сдвиговый регистр передатчика станет пустым, данные будут снова переданы из буфера, и процесс повторится. Каждый из этих регистров имеет флаг (регистр OWI\_ST), на основе которого можно вызвать прерывание (регистр OWI\_MSK). Бит TBE статусного регистра устанавливается, когда буфер передатчика пуст и готов принять новые данные. Как только происходит запись байта в буфер передатчика, TBE очищается. Бит TEMT статусного регистра устанавливается, когда сдвиговый регистр пуст и готов принять новые данные. Как только байт передается из буфера в сдвиговый регистр, бит TEMT очищается, а TBE устанавливается.

### Прием байта

Для считывания данных с шины 1-Wire необходимо подготовить ведомое устройство (команды, ранее полученные от ведущего). Процесс приема данных аналогичен передаче. Ведущий начинает прием, записывая FFh по адресу OWI\_BUF в буфер передатчика. Данные, которые затем передаются в сдвиговый регистр приемника – это побитовое «И» переданных и принятых данных.

После того, как сдвиговый регистр приемника заполняется, данные передаются в буфер приемника, откуда они могут быть прочитаны по адресу OWI\_BUF. Следующие байты от ведомого устройства могут быть прочитаны путем повторной записи FFh в буфер приемника. Если ведомый не был готов передать данные, то полученные данные будут идентичны переданным. На основе состояния буфера приемника и сдвигового регистра приемника также могут генерироваться прерывания. Бит RBF статусного регистра устанавливается, когда данные из сдвигового регистра переданы в буфер, а очищается после чтения буфера. Пока флаг RBF установлен, дальнейшие передачи по шине 1-Wire выполняться не должны, иначе данные могут быть потеряны, так как произойдет перезапись буфера приемника следующим принятым байтом.

### Битовый режим

Чтобы активировать битовый режим необходимо записать в бит BIT\_CTL регистра OWI\_CTRL «1». Последующая работа такая же, как и в байтовом режиме, однако важен только младший бит в буфере приемника/передатчика.

### Ускоренная передача

Переключение между скоростями происходит посредством бита OD регистра OWI\_CTRL. Любые операции на шине 1-Wire после обновления бита OD происходят на новой скорости. Режим длинной линии относится только к стандартной скорости.

### Режим ускоренного поиска ПЗУ (Search ROM Accelerator)

Прежде чем перейти в режим ускоренного поиска ПЗУ необходимо выполнить цикл сброса/обнаружения присутствия и отправить команду поиска ПЗУ (Search ROM – F0h) в байтовом режиме. Буфер приемника не должен содержать данных. Далее можно переходить в режим ускоренного поиска ПЗУ, переведя бит SRA регистра OWI\_CFG в «1».

Для завершения одной процедуры поиска ведущий должен передать 16 байт, построенных согласно таблице 12. Каждый байт, записанный в регистр передатчика, содержит полубайт данных решений несоответствий. В конце каждой процедуры ведущий узнает идентификатор одного ведомого устройства.

Таблица 16. Последовательность переданных/принятых байт в режиме ускоренного поиска ПЗУ

| № байта | Буфер      | Бит 7            | Бит 6           | Бит 5            | Бит 4           | Бит 3            | Бит 2           | Бит 1            | Бит 0           |
|---------|------------|------------------|-----------------|------------------|-----------------|------------------|-----------------|------------------|-----------------|
| 1       | Передатчик | r <sub>3</sub>   | x               | r <sub>2</sub>   | x               | r <sub>1</sub>   | x               | r <sub>0</sub>   | x               |
|         | Приемник   | ID <sub>3</sub>  | d <sub>3</sub>  | ID <sub>2</sub>  | d <sub>2</sub>  | ID <sub>1</sub>  | d <sub>1</sub>  | ID <sub>0</sub>  | d <sub>0</sub>  |
| 2       | Передатчик | r <sub>7</sub>   | x               | r <sub>6</sub>   | x               | r <sub>5</sub>   | x               | r <sub>4</sub>   | x               |
|         | Приемник   | ID <sub>7</sub>  | d <sub>7</sub>  | ID <sub>6</sub>  | d <sub>6</sub>  | ID <sub>5</sub>  | d <sub>5</sub>  | ID <sub>4</sub>  | d <sub>4</sub>  |
| ...     |            |                  |                 |                  |                 |                  |                 |                  |                 |
| 16      | Передатчик | r <sub>63</sub>  | x               | r <sub>62</sub>  | x               | r <sub>61</sub>  | x               | r <sub>60</sub>  | x               |
|         | Приемник   | ID <sub>63</sub> | d <sub>63</sub> | ID <sub>62</sub> | d <sub>62</sub> | ID <sub>61</sub> | d <sub>61</sub> | ID <sub>60</sub> | d <sub>60</sub> |

x – биты, которые не имеют значения;

r<sub>n</sub> – биты, которые выдаст ведущий при несоответствии идентификаторов;

d<sub>n</sub> – флаги несоответствия (равен «1», если есть несоответствие или нет ответа);

ID<sub>n</sub> – биты фактически выбранного идентификатора.

Для каждого бита (с 0 по 63) ведущий генерирует следующие три тайм-слота:

b1 – прием данных (истинный бит);

b2 – прием данных (инверсный бит);

b3 – передача данных.

В третьем временном интервале  $b_3$  ведущий передает «1» и «0» следующим образом:

$b_3 = r_n$ , если у двух и более устройств разные значения  $n$ -ого бита;

$b_3 = b_1$ , если у всех устройств одинаковый  $n$ -ый бит;

$b_3 = \text{«1»}$ , в случае если есть ошибка (нет ответа).

Бит, переданный в тайм-слоте  $b_3$ , идентичен  $ID_n$ .

Для начала выполнения поиска ПЗУ во все  $r_n$  биты пишутся нули. В случае возникновения ошибки все последующие биты  $ID_n$  будут равны «1» до тех пор, пока бит SRA не будет деактивирован. Таким образом, если биты  $ID_{63}$  и  $d_{63}$  равны «1», то произошла ошибка во время процедуры поиска и необходимо повторить последнюю последовательность. В противном случае, все  $ID_n$  (с 0 до 63) – это идентификатор ПЗУ устройства, которое было найдено и адресовано. После завершения поиска ПЗУ нужно очистить бит SRA для вывода ведущего из режима Search ROM Accelerator. Для поиска следующего устройства повторно используется предыдущий набор  $r_n$  ( $n = 0 \dots 63$ ), но  $r_m$  устанавливается в «1», где  $m$  – наивысший порядок флага несоответствия. Все  $r_i$ , где  $i > m$ , устанавливаются в «0». Этот процесс повторяется до тех пор, пока не произойдет расхождение на одной и той же битовой позиции в течение двух проходов.

### Пример использования режима ускоренного поиска ПЗУ

В данном примере рассматривается процесс поиска четырех различных устройств на шине 1-Wire. Они имеют следующие идентификаторы (представлены только младшие значащие биты от 0 до 8):

ROM1 = 00110101...

ROM2 = 10101010...

ROM3 = 11110101...

ROM4 = 00010001...

1) Микроконтроллер инициализирует импульс сброса, записывая 01h в регистр OWI\_CFG. Все подчиненные устройства отвечают импульсом обнаружением присутствия.

2) Микроконтроллер отправляет команду поиска ПЗУ (Search ROM), записывая F0h в буфер передатчика. Необходимо дождаться установления статуса RBF и считать (освободить) буфер приемника.

3) Микроконтроллер переводит модуль в режим Search ROM Accelerator, записывая 02h в регистр OWI\_CFG.

4) Микроконтроллер записывает 00h в буфер передатчика и считывает возвращаемые данные из буфера приемника. Этот процесс повторяется для всех 16 байт. Считанные данные будут содержать в  $ID_n$  ROM4, а «1» установлена в  $d_0$  ( $r_n$  содержит «0», следовательно, ROM2 и ROM3 переходят в состояние ожидания) и  $d_2$  (аналогично первому случаю ROM1 переходит в состояние ожидания). Принятые данные выглядят следующим образом ( $ID_n$  подчеркнуты, наиболее значащий флаг несоответствия **выделен**):

полученные данные (1) = 1000100100000001...

5) Микроконтроллер преобразует полученные данные в идентификатор устройства путем устранения лишних битов.

6) Микроконтроллер записывает 00h в регистр OWI\_CFG для выхода из режима ускоренного поиска ПЗУ. После этого ведущий может продолжить работу с адресованным ведомым.

7) Для нахождения следующего устройства шаги с 1-ого по 6-ой повторяются, только на этот раз в передаваемой последовательности наиболее значащий бит решения несоответствия ( $r_2$  в данном случае) инвертируется, и все данные, следующие за ним, устанавливаются в ноль. Полученные данные содержат идентификатор ROM1, а  $d_0$  и  $d_2$  снова установлены в «1»:

отправленные данные (2) = 0000010000000000...

полученные данные (2) = 1000110100010001...

8) Так как наиболее значащий флаг несоответствия остался прежним ( $d_2$ ), то нужно использовать следующий наиболее значащий ( $d_0$ ). Процесс повторяется. Следующие итерации выглядят следующим образом:

отправленные данные (3) = 010000000000000...

полученные данные (3) = 1110010001000100...

отправленные данные (4) = 010100000000000...

полученные данные (4) = 1111010100010001...

9) Так как наиболее значащий флаг несоответствия ( $d_2$ ) не изменился, то переходим к следующему ( $d_0$ ). Однако,  $d_0$  обработан дважды, а менее значимых флагов несоответствия больше нет. Следовательно поиск завершен, найдены идентификаторы всех четырех устройств.

## Статусы и прерывания

В отличие от остальных модулей в OWI прерывание по бит-событию не формируется от последующих событий после перехода бита в «1», пока он не будет сброшен чтением обратно в «0». Остальные принципы работы, описанные в разделе «Типы статусов и прерывания», остаются неизменными.

## Регистры OWI

| №     | Аббревиатура | Доступ | Описание                             |
|-------|--------------|--------|--------------------------------------|
| 2B00h | OWI_CFG      | RW*    | Регистр конфигурации                 |
| 2B04h | OWI_BUF      | RW     | Регистр буфера приемника/передатчика |
| 2B08h | OWI_ST       | R      | Регистр статуса                      |
| 2B0Ch | OWI_MSK      | RW     | Регистр маски                        |
| 2B10h | OWI_PRCR     | RW     | Регистр делителя частоты             |
| 2B14h | OWI_CTRL     | RW     | Регистр управления                   |

\* – Не все регистры доступны для записи.

## OWI\_CFG

Данный регистр используется для генерации цикла сброса/обнаружения присутствия 1-Wire, а также для активации или деактивации режима ускоренного поиска ПЗУ (Search ROM Accelerator), который позволяет избежать использования одноканальной операции с шиной (бит FOW) во время процедуры поиска. В дополнение к этим двум функциям регистр содержит 2 бита для непосредственного управления выводом 1W\_IO.

| Бит                | 7      | 6 | 5 | 4 | 3     | 2   | 1   | 0   |
|--------------------|--------|---|---|---|-------|-----|-----|-----|
| Назначение         | Резерв |   |   |   | OW_IN | FOW | SRA | 1WR |
| Начальное значение | 0      |   |   |   | 1W_IO | 0   |     |     |

**OW\_IN (1-Wire Input)** – текущее значение на линии 1-Wire (только чтение):

1 – значение на линии соответствует логической единице;

0 – значение на линии соответствует логическому нулю.

Используется в сочетании с битом FOW для выполнения пользовательских функций. Например, для чтения необходимо изменить FOW на «0», затем подождать необходимое количество времени и установить состояние OW\_IN.

**FOW (Force 1-Wire)** – прямое управление линией 1-Wire:

1 – переводит линию в состояние логического нуля;

0 – оставляет линию в состоянии высокого импеданса.

Данную функцию необходимо разрешить путем записи бита EN\_FOW регистра OWI\_CTRL. В противном случае изменение FOW не влияет на состояние шины. Если EN\_FOW и FOW равны «1», то генерация импульса сброса и функция передачи/приема (регистр OWI\_BUF) не работают. Изменение EN\_FOW на «0» имеет тот же эффект, что и запись в FOW «0».

**SRA (Search ROM Accelerator)** – включение режима ускоренного поиска ПЗУ:

1 – модуль работает в режиме ускоренного поиска ПЗУ;

0 – модуль работает в нормальном режиме.

**1WR (1-Wire Reset)** – управление генерацией последовательности сброса/обнаружения присутствия на линии 1-Wire:

1 – инициализировать генерацию цикла сброса/обнаружения присутствия;

0 – цикл сброса/обнаружения присутствия не генерируется.

Бит переводится в «0» после завершения генерации цикла сброса/обнаружения присутствия. Цикл можно прервать, записав «0».

## OWI\_BUF

Все передаваемые и принимаемые данные проходят соответственно через буфер передатчика и приемника. Запись по данному адресу подключает к шине данных буфер передатчика, а считывание подключает к шине данных буфер приемника.

| Бит                | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|------|---|---|---|---|---|---|---|
| Назначение         | DATA |   |   |   |   |   |   |   |
| Начальное значение | 0    |   |   |   |   |   |   |   |

**DATA** – данные на передачу или принятые данные. Данные могут передаваться в байтовом или битовом режиме в зависимости от бита BIT\_CTL регистра OWI\_CTRL. В битовом режиме имеет значение только младший бит.

## OWI\_ST

| Бит                | 7      | 6        | 5    | 4   | 3    | 2   | 1   | 0     |
|--------------------|--------|----------|------|-----|------|-----|-----|-------|
| Назначение         | OW_LOW | OW_SHORT | RSRF | RBF | TEMT | TBE | PDR | PD    |
| Тип статуса        | EVENT  | EVENT    | FLAG |     |      |     |     | EVENT |
| Начальное значение | 0      |          |      |     | 1    | 1   | 1   | 0     |

**OW\_LOW** – принят импульс присутствия от ведомого устройства. Ведомое устройство при подключении к незанятой шине генерирует спонтанный импульс присутствия, который устанавливает этот бит. Любой переход в «0» на незанятой шине также устанавливает этот бит.

**OW\_SHORT** – короткое замыкание на шине. Перед началом импульса сброса или тайм-слота ведущий проверяет, находится ли шина в режиме ожидания на высоком уровне. Если в этот момент уровень шины низкий, то связь невозможна.

**RSRF (Receive Shift Register Full)** – сдвиговый регистр приемника заполнен. Когда этот бит равен «0», то это указывает на то, что сдвиговый регистр приемника не содержит данных или осуществляет их прием. После передачи данных в буфер приемника этот бит сбрасывается аппаратно.

**RBF (Receive Buffer Full)** – буфер приемника заполнен. Флаг переходит в «1», когда есть новые данные, полученные из сдвигового регистра приемника. Если бит равен «0», то это указывает, что буфер приемника не содержит новых данных. Этот бит сбрасывается аппаратно после чтения буфера приемника.

**TEMT (Transmit Shift Register Empty)** – сдвиговый регистр передатчика пуст и готов принять новые данные. Флаг устанавливается в «1», когда регистр сдвига передатчика не содержит данных. После

передачи данных из буфера передатчика в сдвиговый регистр передатчика этот бит переходит в «0». Если бит равен «0», то это указывает на то, что регистр сдвига передатчика занят передачей данных.

**TBE (Transmit Buffer Empty)** – буфер передатчика пуст и готов принять новые данные. Флаг устанавливается в «1», когда буфер передатчика не содержит данных. После записи данных в буфер передатчика этот бит переходит в «0». Если бит равен «0», то это указывает на то, что буфер передатчика ждет окончания сдвига данных из сдвигового регистра передатчика.

**PDR (Presence Detect Result)** – результат цикла обнаружения присутствия:

1 – импульс присутствия не обнаружен;

0 – обнаружен импульс присутствия.

**PD (Presence Detect)** – цикл сброса/обнаружения присутствия завершен, а результат находится в бите PDR. При использовании прерывания по данному биту оно должно быть включено в OWI\_MSK до инициализации цикла сброса.

### OWI\_MSK

| Бит                | 7      | 6        | 5    | 4   | 3    | 2   | 1      | 0  |
|--------------------|--------|----------|------|-----|------|-----|--------|----|
| Назначение         | OW_LOW | OW_SHORT | RSRF | RBF | TEMT | TBE | Резерв | PD |
| Начальное значение | 0      |          |      |     |      |     |        |    |

Для каждого из битов справедливо: 1 – данное прерывание формируется; 0 – данное прерывание не формируется.

### OWI\_PRCR

Модуль OWI требует внутренней тактовой частоты 1 МГц. Эта частота генерируется от тактовой частоты системы  $f_{clk}$ . Регистр OWI\_PRCR позволяет управлять делителем и передделителем тактовой частоты с целью максимально приблизить  $f_{clk}$  к 1 МГц. Систему тактирования модуля необходимо настроить и включить до начала передачи по шине 1-Wire.

| Бит                | 7      | 6      | 5 | 4 | 3   | 2 | 1   | 0 |
|--------------------|--------|--------|---|---|-----|---|-----|---|
| Назначение         | CLK_EN | Резерв |   |   | DIV |   | PRE |   |
| Начальное значение | 0      |        |   |   |     |   |     |   |

**CLK\_EN** – включение системы тактирования OWI:

1 – система тактирования включена;

0 – система тактирования выключена.

**DIV** – выбор коэффициента делителя частоты:

111b –  $f_{clk}/128$ ;

110b –  $f_{clk}/64$ ;

101b –  $f_{clk}/32$ ;

100b –  $f_{clk}/16$ ;

011b –  $f_{clk}/8$ ;

010b –  $f_{clk}/4$ ;

001b –  $f_{clk}/2$ ;

000b –  $f_{clk}$ .

**PRE** – выбор коэффициента предделителя частоты:

11b –  $f_{clk}/7$ ;

10b –  $f_{clk}/5$ ;

01b –  $f_{clk}/3$ ;

00b –  $f_{clk}$ .

Делитель и предделитель работают совместно.

Таблица 17. Настройка коэффициентов деления частоты

| Мин. $f_{clk}$ , МГц | Макс. $f_{clk}$ , МГц | Макс. погрешность $f_{clk}$ , % | Коэф. деления | DIV |   |   | PRE |   |
|----------------------|-----------------------|---------------------------------|---------------|-----|---|---|-----|---|
|                      |                       |                                 |               | 2   | 1 | 0 | 1   | 0 |
| 1                    | 1,25                  | 25                              | 1             | 0   | 0 | 0 | 0   | 0 |
| 2                    | 2,5                   | 25                              | 2             | 0   | 0 | 1 | 0   | 0 |
| 3                    | 3,75                  | 25                              | 3             | 0   | 0 | 0 | 0   | 1 |
| 4                    | < 5                   | 25                              | 4             | 0   | 1 | 0 | 0   | 0 |
| 5                    | < 6                   | 20                              | 5             | 0   | 0 | 0 | 1   | 0 |
| 6                    | < 7                   | 17                              | 6             | 0   | 0 | 1 | 0   | 1 |
| 7                    | < 8                   | 14                              | 7             | 0   | 0 | 0 | 1   | 1 |
| 8                    | < 10                  | 25                              | 8             | 0   | 1 | 1 | 0   | 0 |

**OWI\_CTRL**

| Бит                | 7      | 6  | 5       | 4       | 3 | 2      | 1   | 0   |
|--------------------|--------|----|---------|---------|---|--------|-----|-----|
| Назначение         | Резерв | OD | BIT_CTL | Резерв* |   | EN_FOW | PPM | LLM |
| Начальное значение | 0      |    |         |         |   |        |     |     |

\* – Не изменять начальное значение.

**OD (Overdrive)** – режимом ускоренной передачи:

- 1 – режим ускоренной передачи включен;
- 0 – режим ускоренной передачи выключен.

**BIT\_CTL** – выбор битового или байтового режима:

- 1 – режим битовой передачи включен;
- 0 – режим байтовой передачи выключен.

Следовательно, в битовом режиме соответствующие биты в статусном регистре устанавливаются после 1 тайм-слота, а не после 8 тайм-слотов, как в байтовом режиме.

**EN\_FOW (Enable Force 1-Wire)** – разрешение прямого управления линией 1-Wire:

- 1 – разрешено управление линией 1-Wire посредством бита FOW регистра OWI\_CFG;
- 0 – прямое управление линией 1-Wire запрещено.

**PPM (Presence Pulse Masking)** – режим маскирования импульса присутствия:

- 1 – маскирование импульса присутствия включено;
- 0 – маскирование импульса присутствия выключено.

В условиях длинной линии импульсы присутствия, генерируемые ведомыми устройствами, могут вызывать помехи на линии. Маскирование импульса присутствия может предотвратить неблагоприятные эффекты, вызванные импульсом присутствия. Если PPM равен «1», то результат цикла обнаружения присутствия (бит PDR регистра OWI\_ST) всегда равен «0», даже если на шине нет ведомого устройства.

**LLM (Long Line Mode)** – режим длинной линии:

- 1 – режим длинной линии включен;
- 0 – режим длинной линии выключен.

Включение режима длинной линии изменяет стандартные временные характеристики, чтобы они лучше подходили для связи по длинной линии.

## Рабочий автомат (WORK\_FSM)

### Общая информация

Модуль «Рабочий автомат» представляет собой 24-разрядный таймер. Данный таймер считает «вверх» на частоте системы. Начало работы данного модуля автоматически означает переход системы в режим «Глубокий сон» (более подробно смотрите пункт «Режим «SLEEP»»). По окончании счета будет сформировано немаскируемое прерывание, которое выведет систему из режима «Глубокий сон».

### Особенности работы

Актуальное значение периода счета обязательно должно быть записано перед включением модуля «Рабочий автомат». Таймер модуля «Рабочий автомат» работает на частоте системы. Это значит, что перед запуском данного модуля для снижения энергопотребления пользователь обязан переключить систему на частоту тактирования от RC-генератора в регистре CMM\_CTRL.

Существует два способа запуска модуля «Рабочий автомат»:

- классический запуск;
- запуск с ожиданием.

При классическом запуске пользователь переключает частоту тактирования (модуль CMM). Пользователю необходимо узнать о завершении данного процесса по прерыванию, либо по чтению из статусного регистра CMM\_ST (бит SWITCH). И только после этого пользователь может запустить модуль «Рабочий автомат». По записи сигнала EN в регистр FSM\_CTRL модуль начнет счет «вверх» на частоте системы и переведет микроконтроллер в режим «Глубокий сон».

Для запуска с ожиданием пользователь должен записать «1» в бит EN\_WSW регистра FSM\_CTRL. Модуль «Рабочий автомат» будет ожидать события переключения частоты (модуль CMM). Пользователю необходимо записать команду на переключение частоты в регистр CMM\_CTRL. По окончании процесса переключения модуль «Рабочий автомат» автоматически начнет счет и инициирует процесс перехода в режим «Глубокий сон».

По окончании счета «Рабочий автомат» сформирует прерывание, по которому микроконтроллер начнет выход из режима «Глубокий сон».

### Регистры «Рабочего автомата»

| №     | Аббревиатура | Доступ | Описание                |
|-------|--------------|--------|-------------------------|
| 2C00h | FSM_CTRL     | W      | Регистр управления      |
| 2C01h | FSM_PRD2     | RW     | Регистр периода счета 2 |
| 2C02h | FSM_PRD1     | RW     | Регистр периода счета 1 |
| 2C03h | FSM_PRD0     | RW     | Регистр периода счета 0 |

### FSM\_CTRL

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1      | 0  |
|--------------------|--------|---|---|---|---|---|--------|----|
| Назначение         | Резерв |   |   |   |   |   | EN_WSW | EN |
| Начальное значение | 0      |   |   |   |   |   |        |    |

**EN\_WSW (Enable Wait Switch and Work)** – разрешение работы модуля по переключению частоты:

- 1 – ожидать переключения;
- 0 – модуль выключен.

**EN** – разрешение работы модуля:

- 1 – модуль включен;
- 0 – модуль выключен.

## FSM\_PRD2

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | PRD |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

PRD – период счета таймера, старшая часть.

## FSM\_PRD1

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | PRD |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

PRD – период счета таймера, средняя часть.

## FSM\_PRD0

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | PRD |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

PRD – период счета таймера, младшая часть.

Значение, до которого будет считать модуль определяется как:

$$N_{clk} = PRD + 1,$$

где  $N_{clk}$  – количество тактов системной частоты.

## Таймер

### Общая информация

В системе присутствует 3 модуля таймера. Каждый таймер поддерживает 3 различных независимых режима работы. Кроме того, TIMER0 и TIMER1 дополнительно поддерживают 4-ый режим работы, в котором они взаимодействуют между собой. Каждый модуль имеет внешний вход, который, в зависимости от режима, управляет таймером.

### Структурная схема

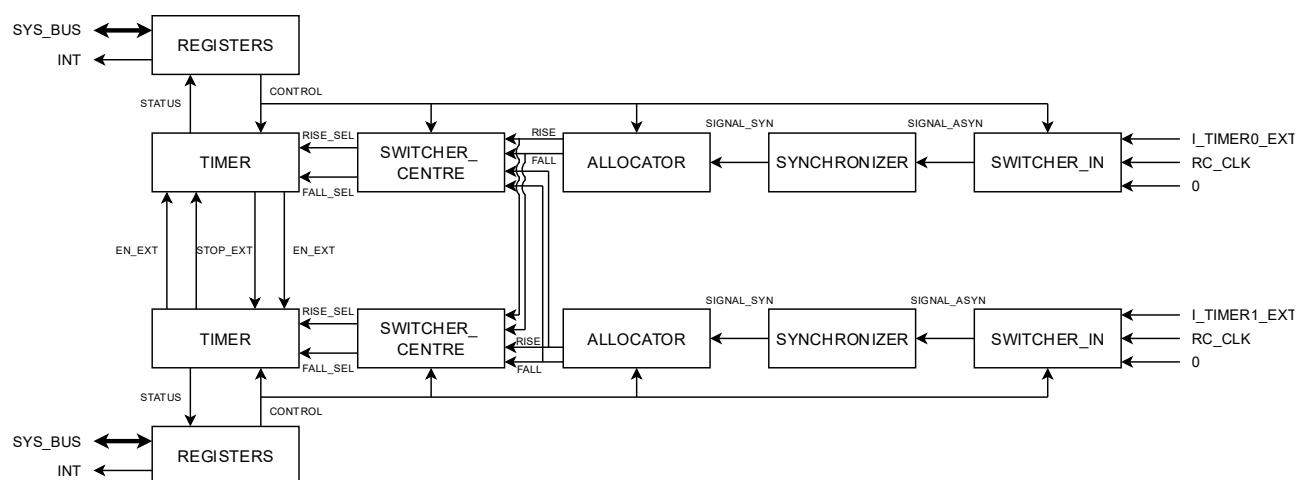


Рисунок 40. Структурная схема двух соединенных модулей таймера

Таймер состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- TIMER – блок управления счетом;
- SWITCHER\_CENTRE – переключатель, определяет будут ли использованы внутренние сигналы или сигналы с соседнего таймера;
- ALLOCATOR – выделитель, выделяет фронт и спад внешнего сигнала;
- SYNCHRONIZER – синхронизатор, синхронизирует внешний сигнал к системной частоте;
- SWITCHER\_IN – переключатель, определяет источник внешнего сигнала.

### Работа в режиме «Простой таймер»

#### Принцип работы

В режиме «Простой таймер» модуль представляет собой 24-разрядный таймер с инкрементацией каждый такт системной частоты. Таймер считает до значения, записанного в регистры периода счета таймера TMR\_PRDH, TMR\_PRDM и TMR\_PRDL. По достижению заданного значения таймер либо останавливается, либо начинает счет с нуля. Данная функция определяется битом CYCLES регистра TMR\_CTRL. Также во время работы допускается перезапись текущего периода счета таймера. В результате перезаписи текущее значение таймера будет сброшено в 0 и счет начнется заново. Текущее значение таймера находится в регистрах TMR\_VALH, TMR\_VALM, TMR\_VALL.

Значение, до которого будет считать модуль, определяется как:

$$N_{clk} = PRD + 1,$$

где  $N_{clk}$  – количество тактов системной частоты.

## Статусы и прерывания

Статусы содержатся в регистре TMR\_ST. В данном режиме вырабатывается статус окончания счета периода – бит END\_PRD. На основании данного статуса возможно возникновение прерывания. Прерывание разрешается путем записи «1» в соответствующий бит регистра маски прерываний TMR\_MSK.

## Алгоритм работы

Процедура настройки режима «Простой таймер»:

- в регистры TMR\_PRDH, TMR\_PRDM и TMR\_PRDL записать требуемое значение периода счета таймера;
- в регистре TMR\_MSK при необходимости разрешить прерывание;
- в регистре TMR\_CTRL:
  - установить требуемое значение бита CYCLES;
  - установить бит T/C в «0»;
  - в биты MODE необходимо записать значение 00b;
  - установить бит EN\_EXT в «0»;
  - установить бит EN в «1».
- текущее значение таймера можно вычитать из регистров TMR\_VALH, TMR\_VALM и TMR\_VALL.

## Работа в режиме «Таймер с внешней остановкой»

### Принцип работы

В режиме «Таймер с внешней остановкой» модуль представляет собой 24-разрядный таймер с инкрементацией каждый такт системной частоты. При запуске таймер считает с нуля до момента возникновения заданного события на соответствующем выводе GPIO (альтернативная функция I\_TIMER\_EXT). Тип события остановки определяется битом STOP\_TYPE регистра TMR\_CFG. Текущее значение таймера находится в регистрах TMR\_VALH, TMR\_VALM и TMR\_VALL.

## Статусы и прерывания

Статусы содержатся в регистре TMR\_ST. В данном режиме вырабатывается статус переполнения таймера – бит OVW и статус остановки таймера по внешнему событию – бит STOP\_EVENT. На основании данных статусов возможно возникновение прерываний. Прерывание разрешается путем записи «1» в соответствующий бит регистра маски прерываний TMR\_MSK.

## Алгоритм работы

Процедура настройки режима «Таймер с внешней остановкой»:

- в регистре TMR\_CFG задать тип события остановки счета таймера путем записи требуемого значения в бит STOP\_TYPE;
- в регистре TMR\_MSK разрешить необходимые прерывания;
- в регистре TMR\_CTRL:
  - установить бит CYCLES в «0»;
  - установить бит T/C в «0»;
  - в биты MODE необходимо записать значение 01b;
  - установить бит EN\_EXT в «0»;
  - установить бит EN в «1».
- текущие значения таймера можно вычитать из регистров TMR\_VALH, TMR\_VALM и TMR\_VALL.

## Работа в режиме «Межсобытийный таймер»

### Принцип работы

В режиме «Межсобытийный таймер» модуль представляет собой 24-разрядный таймер с инкрементацией каждый такт системной частоты. После разрешения работы таймер ожидает событие старта на соответствующем выводе GPIO (альтернативная функция I\_TIMER\_EXT). Тип события старта определяется битом START\_TYPE регистра TMR\_CFG. Окончанием счета является событие остановки счета таймера на выводе GPIO. Тип события остановки определяется битом STOP\_TYPE регистра TMR\_CFG. Текущее значение таймера находится в регистрах TMR\_VALH, TMR\_VALM и TMR\_VALL.

### Статусы и прерывания

Статусы содержатся в регистре TMR\_ST. В данном режиме вырабатывается статус переполнения таймера – бит OVW, статус запуска таймера по внешнему событию – бит START\_EVENT и статус остановки таймера по внешнему событию – бит STOP\_EVENT. На основании данных статусов возможно возникновение прерываний. Прерывание разрешается путем записи «1» в соответствующий бит регистра маски прерываний TMR\_MSK.

### Алгоритм работы

Процедура настройки режима «Межсобытийный таймер»:

- в регистре TMR\_CFG задать тип события запуска и остановки счета таймера путем записи требуемого значения в биты START\_TYPE и STOP\_TYPE;
- в регистре TMR\_MSK разрешить необходимые прерывания;
- в регистре TMR\_CTRL:
  - установить бит CYCLES в «0»;
  - установить бит T/C в «0»;
  - в биты MODE необходимо записать значение 10b;
  - установить бит EN\_EXT в «0»;
  - установить бит EN в «1».
- текущие значения таймера можно вычитать из регистров TMR\_VALH, TMR\_VALM и TMR\_VALL.

## Работа в режиме «Таймер-счетчик»

**Примечание:** данный режим работы поддерживают модули TIMER0 и TIMER1. Модуль TIMER2 данный режим работы не поддерживает. При запуске TIMER2 в режиме «Таймер-счетчик» модуль уйдет в состояние ожидания, выход из которого возможен только по отключению модуля.

### Принцип работы

В режиме «Таймер-счетчик» модули взаимодействуют между собой. Один из модулей необходимо настроить в режим таймера, а второй в режим счетчика. На вход модуля, который работает в режиме счетчика может быть подана либо частота внутреннего RC-генератора, либо сигнал с соответствующего вывода GPIO (альтернативная функция I\_TIMER\_EXT). Источник сигнала, события на котором будут отслеживаться, определяется битом EVENT\_TYPE регистра TMR\_CFG модуля счетчика. Тип события старта задается в обоих модулях битом START\_TYPE регистра TMR\_CFG.

После настройки модуль в режиме таймера и модуль в режиме счетчика ожидают события старта от ранее выбранного источника. Затем модуль, работающий в режиме таймера, считает до момента остановки его модулем в режиме счетчика. Модуль в режиме счетчика считает до значения, записанного в регистры периода счета TMR\_PRDH, TMR\_PRDM и TMR\_PRDL. Счетчик увеличивается на 1 каждый раз, когда фиксирует заданное событие. Тип события определяется битом FIX\_TYPE регистра TMR\_CFG. По завершению счета модуль в режиме счетчика формирует событие окончания

счета для модуля, работающего в режиме таймера. В результате оба модуля прекращают счет. Перезаписывать значение TMR\_PRDH, TMR\_PRDM и TMR\_PRDL для модуля в режиме счетчика в ходе работы запрещено.

При START\_TYPE = FIX\_TYPE значение, до которого будет считать счетчик, определяется как:

$$N_{event} = PRD + 1,$$

где  $N_{event}$  – количество событий.

При START\_TYPE != FIX\_TYPE:

$$N_{event} = PRD.$$

### Статусы и прерывания

Статусы содержатся в регистре TMR\_ST. В данном режиме, для модуля, работающего в режиме таймера, вырабатывается статус переполнения таймера – бит OVW, статус запуска таймера по внешнему событию – бит START\_EVENT и статус остановки таймера по внешнему событию – бит STOP\_EVENT. Для модуля, работающего в режиме счетчика, вырабатывается статус запуска таймера по внешнему событию – бит START\_EVENT. На основании данных статусов возможно возникновение прерываний. Прерывание разрешается путем записи «1» в соответствующий бит регистра маски прерываний TMR\_MSK.

### Алгоритм работы

Процедура настройки режима «Таймер-счетчик» для модуля в режиме таймера:

- в регистре TMR\_CFG задать тип события запуска счета таймера путем записи требуемого значения в бит START\_TYPE;
- в регистре TMR\_MSK разрешить необходимые прерывания;
- в регистре TMR\_CTRL:
  - установить бит CYCLES в «0»;
  - установить бит T/C в «0»;
  - в биты MODE необходимо записать значение 11b;
  - установить бит EN\_EXT в «1»;
  - установить бит EN в «1».
- текущее значения таймера можно вычитать из регистров TMR\_VALH, TMR\_VALM и TMR\_VALL.

Бит EN\_EXT разрешает соседнему модулю начать работу, таким образом синхронизируя запуск обоих модулей.

Процедура настройки режима «Таймер-счетчик» для модуля в режиме счетчика:

- в регистре TMR\_CFG:
  - задать источник сигнала битом EVENT\_TYPE;
  - задать тип события счетчика битом FIX\_TYPE;
  - задать тип события запуска счета битом START\_TYPE.
- в регистры TMR\_PRDH, TMR\_PRDM и TMR\_PRDL записать требуемое значение периода счета таймера;
- в регистре TMR\_MSK при необходимости разрешить прерывание;
- в регистре TMR\_CTRL:
  - установить бит CYCLES в «0»;
  - установить бит T/C в «1»;
  - в биты MODE необходимо записать значение 11b;
  - установить бит EN\_EXT в «1»;
  - установить бит EN в «1».
- текущее значения таймера можно вычитать из регистров TMR\_VALH, TMR\_VALM и TMR\_VALL.

Бит EN\_EXT разрешает соседнему модулю начать работу, таким образом синхронизируя запуск обоих модулей.

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

## Регистры таймеров

| №             | Аббревиатура | Доступ | Описание                                |
|---------------|--------------|--------|-----------------------------------------|
| <b>TIMER0</b> |              |        |                                         |
| 2D00h         | TMR0_CTRL    | RW     | Регистр управления                      |
| 2D01h         | TMR0_CFG     | RW     | Регистр конфигурации                    |
| 2D02h         | TMR0_PRDH    | RW     | Период счета таймера, старшая часть     |
| 2D03h         | TMR0_PRDM    | RW     | Период счета таймера, средняя часть     |
| 2D04h         | TMR0_PRDL    | RW     | Период счета таймера, младшая часть     |
| 2D05h         | TMR0_VALH    | R      | Текущее значение таймера, старшая часть |
| 2D06h         | TMR0_VALM    | R      | Текущее значение таймера, средняя часть |
| 2D07h         | TMR0_VALL    | R      | Текущее значение таймера, младшая часть |
| 2D08h         | TMR0_MSK     | RW     | Регистр маски прерываний                |
| 2D09h         | TMR0_ST      | R      | Регистр статусов                        |
| <b>TIMER1</b> |              |        |                                         |
| 2D20h         | TMR1_CTRL    | RW     | Регистр управления                      |
| 2D21h         | TMR1_CFG     | RW     | Регистр конфигурации                    |
| 2D22h         | TMR1_PRDH    | RW     | Период счета таймера, старшая часть     |
| 2D23h         | TMR1_PRDM    | RW     | Период счета таймера, средняя часть     |
| 2D24h         | TMR1_PRDL    | RW     | Период счета таймера, младшая часть     |
| 2D25h         | TMR1_VALH    | R      | Текущее значение таймера, старшая часть |
| 2D26h         | TMR1_VALM    | R      | Текущее значение таймера, средняя часть |
| 2D27h         | TMR1_VALL    | R      | Текущее значение таймера, младшая часть |
| 2D28h         | TMR1_MSK     | RW     | Регистр маски прерываний                |
| 2D29h         | TMR1_ST      | R      | Регистр статусов                        |
| <b>TIMER2</b> |              |        |                                         |
| 2D40h         | TMR2_CTRL    | RW     | Регистр управления                      |
| 2D41h         | TMR2_CFG     | RW     | Регистр конфигурации                    |
| 2D42h         | TMR2_PRDH    | RW     | Период счета таймера, старшая часть     |
| 2D43h         | TMR2_PRDM    | RW     | Период счета таймера, средняя часть     |
| 2D44h         | TMR2_PRDL    | RW     | Период счета таймера, младшая часть     |
| 2D45h         | TMR2_VALH    | R      | Текущее значение таймера, старшая часть |
| 2D46h         | TMR2_VALM    | R      | Текущее значение таймера, средняя часть |
| 2D47h         | TMR2_VALM    | R      | Текущее значение таймера, младшая часть |
| 2D48h         | TMR2_MSK     | RW     | Регистр маски прерываний                |
| 2D49h         | TMR2_ST      | R      | Регистр статусов                        |

## TMRx\_CTRL

| Бит                | 7      | 6 | 5      | 4   | 3    | 2 | 1      | 0  |
|--------------------|--------|---|--------|-----|------|---|--------|----|
| Назначение         | Резерв |   | CYCLES | T/C | MODE |   | EN_EXT | EN |
| Начальное значение | 0      |   |        |     |      |   |        |    |

**CYCLES** – работа таймера по достижению значения периода:

1 – повторный счет с нуля;

0 – остановка счета.

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

**T/C** – работа в режиме таймера или счетчика, имеет значение только при MODE «Таймер-счетчик»:

1 – режим счетчика;

0 – режим таймера.

**MODE** – режимы работы модуля:

11b – «Таймер-счетчик»;

10b – «Межсобытийный таймер»;

01b – «Таймер с внешней остановкой»;

00b – «Простой таймер».

**EN\_EXT** – разрешение работы соседнего модуля таймера, имеет значение только при MODE «Таймер-счетчик»:

1 – работа соседнего таймера разрешена;

0 – работа соседнего таймера запрещена.

При работе с регистрами модуля TIMER0 бит EN\_EXT управляет модулем TIMER1. При работе с регистрами модуля TIMER1 бит EN\_EXT управляет модулем TIMER0.

**EN** – разрешение работы таймера:

1 – таймер включен;

0 – таймер выключен.

#### TMRx\_CFG

| Бит                | 7      | 6 | 5 | 4 | 3          | 2        | 1         | 0          |
|--------------------|--------|---|---|---|------------|----------|-----------|------------|
| Назначение         | Резерв |   |   |   | EVENT_TYPE | FIX_TYPE | STOP_TYPE | START_TYPE |
| Начальное значение | 0      |   |   |   |            |          |           |            |

**EVENT\_TYPE** – источник сигнала, на котором отслеживаются события в режиме «Таймер-счетчик»:

1 – вывод GPIO;

0 – внутренний RC-генератор.

**FIX\_TYPE** – тип события, которое считается счетчиком в режиме «Таймер-счетчик»:

1 – событие заднего фронта;

0 – событие переднего фронта.

**STOP\_TYPE** – тип события остановки счета таймера:

1 – остановка по заднему фронту;

0 – остановка по переднему фронту.

**START\_TYPE** – тип события старта счета таймера:

1 – старт по заднему фронту;

0 – старт по переднему фронту.

#### TMRx\_PRDH

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | PRD |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

**PRD** – период счета таймера, старшая часть.

**TMRx\_PRDM**

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | PRD |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

PRD – период счета таймера, средняя часть.

**TMRx\_PRDL**

Период счета таймера обновляется при записи в данный регистр.

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | PRD |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

PRD – период счета таймера, младшая часть.

**TMRx\_VALH**

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | VAL |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

VAL – текущее значение таймера, старшая часть.

**TMRx\_VALM**

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | VAL |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

VAL – текущее значение таймера, средняя часть.

**TMRx\_VALL**

| Бит                | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-----|---|---|---|---|---|---|---|
| Назначение         | VAL |   |   |   |   |   |   |   |
| Начальное значение | 0   |   |   |   |   |   |   |   |

VAL – текущее значение таймера, младшая часть.

**TMRx\_MSK**

Возможно формирование прерывания по любому биту статусного регистра TMR\_ST. Расположение битов в TMR\_ST и TMR\_MSK аналогично.

| Бит                | 7      | 6 | 5 | 4 | 3          | 2           | 1   | 0       |
|--------------------|--------|---|---|---|------------|-------------|-----|---------|
| Назначение         | Резерв |   |   |   | STOP_EVENT | START_EVENT | OVW | END_PRD |
| Начальное значение | 0      |   |   |   |            |             |     |         |

Для каждого из битов справедливо:

1 – данное прерывание формируется;

0 – данное прерывание не формируется.

## TMRx\_ST

| Бит                | 7      | 6 | 5 | 4 | 3          | 2           | 1   | 0       |
|--------------------|--------|---|---|---|------------|-------------|-----|---------|
| Назначение         | Резерв |   |   |   | STOP_EVENT | START_EVENT | OVW | END_PRD |
| Тип статуса        |        |   |   |   | EVENT      |             |     |         |
| Начальное значение | 0      |   |   |   |            |             |     |         |

**STOP\_EVENT** – зафиксировано событие остановки таймера;

**START\_EVENT** – зафиксировано событие старта таймера;

**OVW** – зафиксировано переполнение таймера;

**END\_PRD** – зафиксирован конец счета периода.

## АЦП

### Общая информация

12-разрядный АЦП последовательного приближения. Минимальное время преобразования – 16 тактов.

Блок управления АЦП имеет два режима работы:

- одиночный режим;
- автоматический режим.

### Одиночный режим

После подачи сигнала START микроконтроллер ожидает READY на протяжении 32 периодов. Если READY не пришел, то в статусном регистре ADC\_ST будет записана «1» в соответствующий бит. Можно разрешить прерывание по приходу READY. Результаты преобразования находятся в регистрах ADC\_RES0, ADC\_RES1. Далее START можно подать снова, записав бит START\_ADC в регистре ADC\_CTRL.

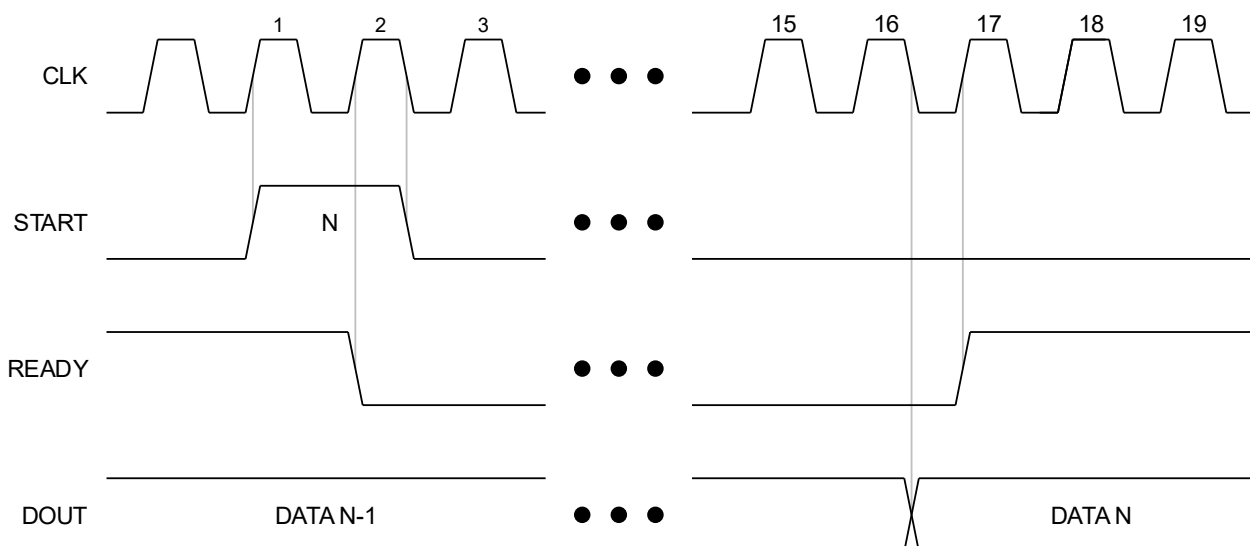


Рисунок 41. Временная диаграмма работы в одиночном режиме

Вне зависимости от режима, после включения АЦП записью «1» в бит EN регистра ADC\_CTRL, начинается отсчет 16 тактов частоты АЦП, по завершению которых формируется сигнал START, если записан бит START\_ADC. Сигнал START формируется по фронту частоты АЦП и снимается по срезу, длится 1,5 периода.

### Автоматический режим

В этом режиме START подается автоматически, пока записан бит START\_ADC в регистре ADC\_CTRL. Микроконтроллер ожидает READY на протяжении 32 периодов, если READY не придет START будет отправлен снова. Микроконтроллер должен успевать считывать данные с АЦП. В том случае, если микроконтроллер будет читать данные настолько долго, что АЦП успеет завершить новое преобразование, то в регистр ADC\_ST будет записан соответствующий бит. Сигнал READY формируется и снимается по фронту частоты АЦП, длится 2 периода.

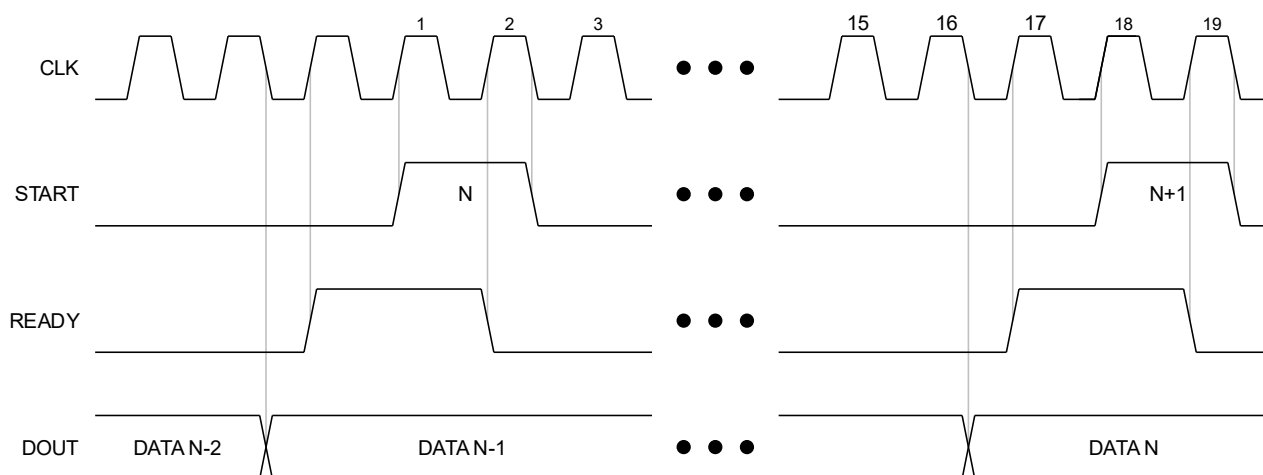


Рисунок 42. Временная диаграмма максимального быстродействия в автоматическом режиме

### Алгоритмы работы

Процедура работы с АЦП в одиночном режиме:

- в регистре ADC\_MSK разрешить необходимые прерывания;
- в регистре ADC\_CFG:
  - установить выравнивание данных в регистрах ADC\_RES1 и ADC\_RES0 битом SHIFT;
  - установить биты DIV для определения частоты работы АЦП;
  - бит MODE должен быть в «0».
- в регистре ADC\_CTRL:
  - выбрать необходимый вывод мультиплексора с помощью бит MUX;
  - подать START битом START\_ADC;
  - включить АЦП битом EN.
- по завершению преобразования вычитать результат из регистров ADC\_RES1 и ADC\_RES0;
- выключить АЦП битом EN или продолжить работу снова записав бит START\_ADC (в этом режиме после подачи START бит START\_ADC автоматически перейдет в «0», снова записать «1» возможно только после завершения преобразования или по истечению времени ожидания READY).

Процедура работы с АЦП в автоматическом режиме:

- в регистре ADC\_MSK разрешить необходимые прерывания;
- в регистре ADC\_CFG:
  - установить выравнивание данных в регистрах ADC\_RES1 и ADC\_RES0 битом SHIFT;
  - установить биты DIV для определения частоты работы АЦП;
  - выбрать режим, в котором будет происходить чтение READ\_MODE;
  - установить бит MODE в «1».
- в регистре ADC\_CTRL:
  - выбрать необходимый вывод мультиплексора с помощью бит MUX;
  - подать START битом START\_ADC;
  - включить АЦП битом EN.
- по завершению преобразования вычитать результат из регистров ADC\_RES1 и ADC\_RES0;
- для завершения преобразования достаточно записать «0» в бит EN или START\_ADC, при этом, если во время записи уже идет преобразование, то оно будет прервано.

Данный режим предполагает оперативное считывание данных по прерыванию RDY.

## Регистры модуля «АЦП»

| №     | Аббревиатура | Доступ | Описание                  |
|-------|--------------|--------|---------------------------|
| 2F00h | ADC_CTRL     | RW     | Регистр управления        |
| 2F04h | ADC_CFG      | RW     | Регистр конфигурации      |
| 2F08h | ADC_MSK      | W      | Регистр маски прерываний  |
| 2F0Ch | ADC_ST       | R      | Регистр статусов          |
| 2F10h | ADC_RES0     | R      | Регистр принятых данных 0 |
| 2F11h | ADC_RES1     | R      | Регистр принятых данных 1 |

## ADC\_CTRL

| Бит                | 7       | 6 | 5 | 4   | 3 | 2 | 1         | 0  |
|--------------------|---------|---|---|-----|---|---|-----------|----|
| Назначение         | Резерв* |   |   | MUX |   |   | START_ADC | EN |
| Начальное значение | 0       |   |   |     |   |   |           |    |

\* – Не изменять начальное значение.

**MUX** – выбор пина аналогового мультиплексора:

111b – TD (термодатчик);

011b – A3;

010b – A2;

001b – A1;

000b – A0.

Для работы термодатчика используется регистр ADC\_CTRL модуля АЦП.

Таблица 18. Типовые значения температурного датчика

| Температура, °C | Выходной код |            |            |
|-----------------|--------------|------------|------------|
|                 | Образец №1   | Образец №2 | Образец №3 |
| –60             | 1690         | 1666       | 1683       |
| –45             | 1744         | 1721       | 1729       |
| –25             | 1815         | 1793       | 1811       |
| –10             | 1888         | 1851       | 1855       |
| +5              | 1931         | 1908       | 1922       |
| +25             | 1996         | 1966       | 1981       |
| +60             | 2093         | 2065       | 2074       |
| +85             | 2154         | 2131       | 2144       |

**Важно!** Настройка каждого температурного датчика уникальна.

**START\_ADC** – разрешает подачу старта:

1 – подать старт;

0 – START не подается.

**EN** – подача частоты и включение АЦП:

1 – подана тактовая частота и АЦП включен в рабочий режим;

0 – режим энергосбережения.

**ADC\_CFG**

| Бит                | 7      | 6 | 5     | 4   | 3 | 2 | 1         | 0    |
|--------------------|--------|---|-------|-----|---|---|-----------|------|
| Назначение         | Резерв |   | SHIFT | DIV |   |   | READ_MODE | MODE |
| Начальное значение | 0      |   |       |     |   |   |           |      |

**SHIFT** – выравнивание 12-бит данных в регистрах ADC\_RES1 и ADC\_RES0 по левому или правому краю (соответственно младшие или старшие 4 бита будут заполнены нулями):

- 1 – выравнивание по правому краю;
- 0 – выравнивание по левому краю.

**DIV** – выбор частоты АЦП:

- 110b –  $f_{clk}/2$ ;
- 101b –  $f_{clk}/4$ ;
- 100b –  $f_{clk}/8$ ;
- 011b –  $f_{clk}/16$ ;
- 010b –  $f_{clk}/32$ ;
- 001b –  $f_{clk}/64$ ;
- 000b –  $f_{clk}/128$ .

**READ\_MODE** – выбор режима чтения, имеет значение только в автоматическом режиме:

- 1 – режим чтения только старшего байта;
- 0 – режим чтения старшего и младшего байта.

**MODE** – выбор режима работы:

- 1 – автоматический режим;
- 0 – одиночный режим.

**ADC\_ST**

| Бит                | 7      | 6 | 5 | 4 | 3 | 2     | 1       | 0       |
|--------------------|--------|---|---|---|---|-------|---------|---------|
| Назначение         | Резерв |   |   |   |   | RDY   | ERR_RDY | ERR_SPD |
| Тип статуса        |        |   |   |   |   | EVENT |         |         |
| Начальное значение | 0      |   |   |   |   |       |         |         |

**RDY** – новые данные готовы для считывания.

**ERR\_RDY** – START был отправлен, но после 32 периодов частоты АЦП сигнал готовности данных не пришел.

**ERR\_SPD** – данные читались настолько долго, что АЦП успел завершить еще одно преобразование и данные обновились, т.е. микроконтроллер не успевает читать данные. Процесс чтения должен быть завершен до момента прихода нового READY.

**ADC\_MSK**

Возможно формирование прерывания по любому биту статусного регистра ADC\_ST. Расположение битов в ADC\_ST и ADC\_MSK аналогично.

| Бит                | 7      | 6 | 5 | 4 | 3 | 2   | 1       | 0       |
|--------------------|--------|---|---|---|---|-----|---------|---------|
| Назначение         | Резерв |   |   |   |   | RDY | ERR_RDY | ERR_SPD |
| Начальное значение | 0      |   |   |   |   |     |         |         |

Для каждого из битов справедливо:

- 1 – данное прерывание формируется;
- 0 – данное прерывание не формируется.

**ADC\_RES0**

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|--------|---|---|---|---|---|---|---|
| Назначение         | RESULT |   |   |   |   |   |   |   |
| Начальное значение | 0      |   |   |   |   |   |   |   |

**RESULT** – результат преобразования, младшая часть.

В зависимости от 5 бита (SHIFT) в регистре ADC\_CFG младший бит выходного кода (D0) находится в бите 4 (SHIFT=0), либо в бите 0 (SHIFT=1).

**ADC\_RES1**

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|--------|---|---|---|---|---|---|---|
| Назначение         | RESULT |   |   |   |   |   |   |   |
| Начальное значение | 0      |   |   |   |   |   |   |   |

**RESULT** – результат преобразования, старшая часть.

В зависимости от 5 бита (SHIFT) в регистре ADC\_CFG старший бит выходного кода (D11) находится в бите 7 (SHIFT = 0), либо в бите 3 (SHIFT = 1).

## ЦАП

### Общая информация

12-разрядный R-2R ЦАП.

### Алгоритм работы

Процедура работы с ЦАП:

- в регистре DAC\_CFG установить выравнивание данных в регистрах DAC\_VALUE1 и DAC\_VALUE0 битом SHIFT;
- в регистре DAC\_CTRL разрешить подачу синхросигнала на ЦАП битом EN;
- отправлять данные на преобразование в регистры DAC\_VALUE1 и DAC\_VALUE0 (12 бит данных обновятся на входе ЦАП после записи в регистр DAC\_VALUE1);
- выключить ЦАП битом EN или продолжить работу записав новые данные.

### Регистры модуля «ЦАП»

| №     | Аббревиатура | Доступ | Описание                   |
|-------|--------------|--------|----------------------------|
| 3000h | DAC_CTRL     | RW     | Регистр управления         |
| 3004h | DAC_CFG      | RW     | Регистр конфигурации       |
| 3008h | DAC_VALUE0   | RW     | Данные на преобразование 0 |
| 3009h | DAC_VALUE1   | RW     | Данные на преобразование 1 |

#### DAC\_CTRL

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0  |
|--------------------|--------|---|---|---|---|---|---|----|
| Назначение         | Резерв |   |   |   |   |   |   | EN |
| Начальное значение | 0      |   |   |   |   |   |   |    |

**EN** – включение ЦАП:

1 – ЦАП включен;

0 – ЦАП выключен.

#### DAC\_CFG

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
|--------------------|--------|---|---|---|---|---|---|-------|
| Назначение         | Резерв |   |   |   |   |   |   | SHIFT |
| Начальное значение | 0      |   |   |   |   |   |   |       |

**SHIFT** – выравнивание 12-бит данных в регистрах DAC\_VALUE1 и DAC\_VALUE0 по левому или правому краю (соответственно младшие или старшие 4 бита не будут иметь значения):

1 – выравнивание по правому краю;

0 – выравнивание по левому краю.

#### DAC\_VALUE0

| Бит                | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-------|---|---|---|---|---|---|---|
| Назначение         | VALUE |   |   |   |   |   |   |   |
| Начальное значение | 0     |   |   |   |   |   |   |   |

**VALUE** – данные на преобразование, младшая часть.

В зависимости от 0 бита (SHIFT) в регистре DAC\_CFG младший бит входного кода (D0) находится в бите 4 (SHIFT = 0), либо в бите 0 (SHIFT = 1).

**DAC\_VALUE1**

Данные обновятся на входе ЦАП после записи в регистр DAC\_VALUE1.

| Бит                | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|-------|---|---|---|---|---|---|---|
| Назначение         | VALUE |   |   |   |   |   |   |   |
| Начальное значение | 0     |   |   |   |   |   |   |   |

**VALUE** – данные на преобразование, старшая часть.

В зависимости от 0 бита (SHIFT) в регистре DAC\_CFG старший бит входного кода (D11) находится в бите 7 (SHIFT = 0), либо в бите 3 (SHIFT = 1).

## Контроллер прерываний (INT\_CTRL)

При выполнении функции, вызванной прерыванием, необходимо произвести сброс данного прерывания.

### Регистры «Контроллера прерываний»

| №     | Аббревиатура | Доступ | Описание                                     |
|-------|--------------|--------|----------------------------------------------|
| 3100h | INT_FIX_CLR0 | RW     | Регистр зафиксированных прерываний, группа 0 |
| 3101h | INT_FIX_CLR1 | RW     | Регистр зафиксированных прерываний, группа 1 |
| 3102h | INT_FIX_CLR2 | RW     | Регистр зафиксированных прерываний, группа 2 |
| 3103h | INT_FIX_CLR3 | RW     | Регистр зафиксированных прерываний, группа 3 |

#### INT\_FIX\_CLR0

| Бит                | 7      | 6 | 5 | 4 | 3      | 2      | 1      | 0      |
|--------------------|--------|---|---|---|--------|--------|--------|--------|
| Назначение         | Резерв |   |   |   | I3_FIX | I2_FIX | I1_FIX | I0_FIX |
| Начальное значение | 0      |   |   |   |        |        |        |        |

**I3\_FIX** – прерывание CMM.

**I2\_FIX** – прерывание WDT.

**I1\_FIX** – прерывание ADC.

**I0\_FIX** – прерывание WORK\_FSM.

Запись в **Ix\_FIX** регистра INT\_FIX\_CLR:

1 – сбросить прерывание;

0 – не менять текущую настройку.

Чтение **Ix\_FIX** регистра INT\_FIX\_CLR:

1 – зафиксировано прерывание;

0 – прерывание отсутствует.

#### INT\_FIX\_CLR1

| Бит                | 7      | 6 | 5 | 4      | 3      | 2      | 1      | 0      |
|--------------------|--------|---|---|--------|--------|--------|--------|--------|
| Назначение         | Резерв |   |   | I4_FIX | I3_FIX | I2_FIX | I1_FIX | I0_FIX |
| Начальное значение | 0      |   |   |        |        |        |        |        |

**I4\_FIX** – прерывание I2C.

**I3\_FIX** – прерывание SPI0.

**I2\_FIX** – прерывание UART0.

**I1\_FIX** – прерывание TIMER0.

**I0\_FIX** – прерывание GPIOA.

Запись в **Ix\_FIX** регистра INT\_FIX\_CLR:

1 – сбросить прерывание;

0 – не меняет текущую настройку.

Чтение **Ix\_FIX** регистра INT\_FIX\_CLR:

1 – зафиксировано прерывание;

0 – прерывание отсутствует.

**INT\_FIX\_CLR2**

| Бит                | 7      | 6 | 5 | 4      | 3      | 2      | 1      | 0      |
|--------------------|--------|---|---|--------|--------|--------|--------|--------|
| Назначение         | Резерв |   |   | I4_FIX | I3_FIX | I2_FIX | I1_FIX | I0_FIX |
| Начальное значение | 0      |   |   |        |        |        |        |        |

**I4\_FIX** – прерывание OWI.

**I3\_FIX** – прерывание SPI1.

**I2\_FIX** – прерывание UART1.

**I1\_FIX** – прерывание TIMER1.

**I0\_FIX** – прерывание GPIOB.

Запись в **Ix\_FIX** регистра INT\_FIX\_CLR:

1 – сбросить прерывание;

0 – не меняет текущую настройку.

Чтение **Ix\_FIX** регистра INT\_FIX\_CLR:

1 – зафиксировано прерывание;

0 – прерывание отсутствует.

**INT\_FIX\_CLR3**

| Бит                | 7      | 6 | 5 | 4 | 3 | 2 | 1      | 0      |
|--------------------|--------|---|---|---|---|---|--------|--------|
| Назначение         | Резерв |   |   |   |   |   | I1_FIX | I0_FIX |
| Начальное значение | 0      |   |   |   |   |   |        |        |

**I1\_FIX** – прерывание TIMER2.

**I0\_FIX** – прерывание GPIOC.

Запись в **Ix\_FIX** регистра INT\_FIX\_CLR:

1 – сбросить прерывание;

0 – не меняет текущую настройку.

Чтение **Ix\_FIX** регистра INT\_FIX\_CLR:

1 – зафиксировано прерывание;

0 – прерывание отсутствует.

## Режим «SLEEP»

### Общая информация

Микроконтроллер имеет возможность переходить в режим пониженного энергопотребления. Существует два различных режима «SLEEP»:

- «Сон процессора»;
- «Глубокий сон».

### Сон процессора

Для перевода системы в режим «Сон процессора» необходимо записать бит CPU\_SLEEP\_EN в регистр PMM\_CTRL модуля PMM при отсутствии активных прерываний в регистре INT\_FIX модуля «Контроллер прерываний». При этом с процессора будет снята синхрочастота, до момента, пока контроллер прерываний не зафиксирует одно из прерываний системы. При возникновении прерывания подача синхросигнала на процессор 8051 будет возобновлена.

### Глубокий сон

Для перевода системы в режим «Глубокий сон» необходимо записать бит EN в регистр FSM\_CTRL модуля «Рабочий автомат». При этом произойдет запуск модуля «Рабочий автомат», и по началу счета система начнет переход в режим «Глубокий сон». Со всех модулей системы, кроме модулей «Рабочий автомат», СММ, PMM будет снята синхрочастота, а также ПЗУ, ФАПЧ и МОУ перейдут в малопотребляющий режим. Выход из данного режима произойдет, когда таймер модуля «Рабочий автомат» окончит счет. При этом будет сформировано немаскируемое прерывание, и подача синхрочастоты возобновится на все модули системы. Также выход из данного режима может быть осуществлен по фронту сигнала на выводе заранее настроенного GPIO или по любому сигналу сброса.

## JTAG

### Общее описание

Микроконтроллер содержит модуль JTAG, который поддерживает обязательные и ряд дополнительных пользовательских инструкции.

| Инструкция      | Код инструкции | Разрядность регистра данных |
|-----------------|----------------|-----------------------------|
| EXTEST          | 000b           | –                           |
| TEST_MODE       | 001b           | –                           |
| SAMPLE/PRELOAD  | 010b           | –                           |
| JTAG_RAM_ACCESS | 011b           | 24                          |
| JTAG_ROM_ACCESS | 100b           | 24                          |
| IF_ACCESS       | 101b           | 18                          |
| DEBUGGER        | 110b           | 5                           |
| BYPASS          | 111b           | 1                           |

### Структурная схема

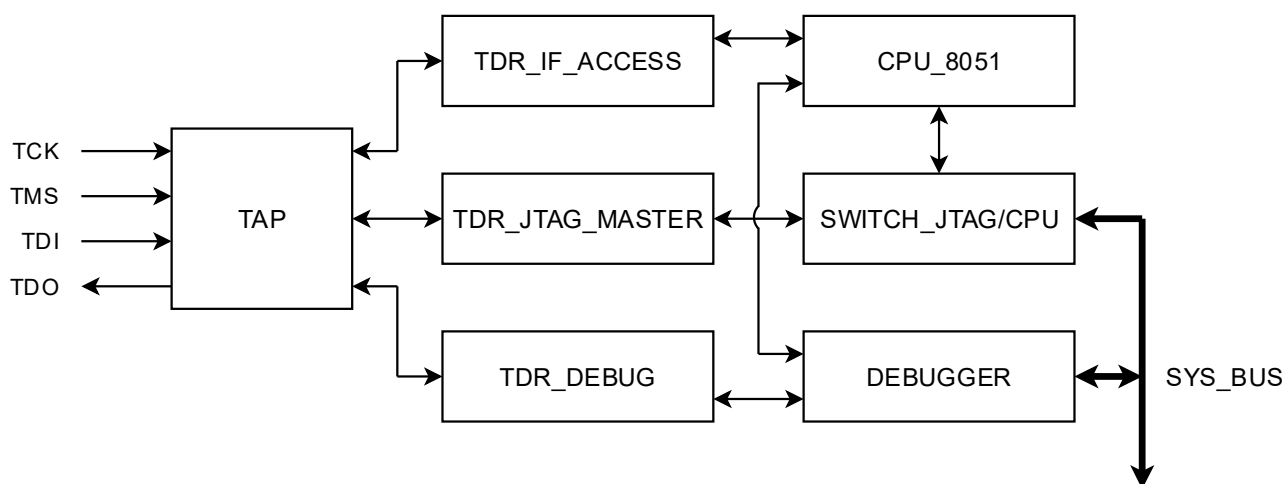


Рисунок 43. Структурная схема модуля JTAG с подключенными к нему блоками

На представленном выше рисунке изображены следующие блоки:

- TAP – управляющий рабочий автомат JTAG;
- TDR\_IF\_ACCESS – блок управления тестовым пользовательским регистром данных инструкции «IF\_ACCESS»;
- TDR\_JTAG\_MASTER – блок управления тестовым пользовательским регистром данных инструкций «JTAG\_RAM\_ACCESS» и «JTAG\_ROM\_ACCESS»;
- TDR\_DEBUG – блок управления тестовым пользовательским регистром данных инструкции «DEBUGGER»;
- CPU\_8051 – процессорное ядро;
- SWITCH\_CPU/JTAG – блок, дающий доступ к системной шине CPU или JTAG;
- DEBUGGER – отладчик.

**Инструкции «TEST\_MODE», «EXTEST», «BYPASS», «SAMPLE/PRELOAD»**

Инструкции «EXTEST» и «SAMPLE/PRELOAD» декодируются TAP-контроллером, но не имеют регистровых цепочек.

Инструкция «TEST\_MODE» имеет трехбитовый код 001b и не предполагает использование пользователем.

Инструкция «BYPASS» имеет трехбитовый код 111b и функционирует в соответствии со стандартом IEEE Std 1149.1-2001.

**Инструкция «JTAG\_RAM\_ACCESS»**

Инструкция «JTAG\_RAM\_ACCESS» представляет собой дополнительную пользовательскую инструкцию с кодом 011b, в результате выбора которой CPU 8051 будет отключен от системной шины, а в место него будет подключен тестовый пользовательский регистр данных разрядностью 24 бит:

|                    |      |        |      |    |    |    |    |    |
|--------------------|------|--------|------|----|----|----|----|----|
| Бит                | 23   | 22     | 21   | 20 | 19 | 18 | 17 | 16 |
| Назначение         | RW   | CE_RAM | ADDR |    |    |    |    |    |
| Начальное значение | 0    |        |      |    |    |    |    |    |
| Бит                | 15   | 14     | 13   | 12 | 11 | 10 | 9  | 8  |
| Назначение         | ADDR |        |      |    |    |    |    |    |
| Начальное значение | 0    |        |      |    |    |    |    |    |
| Бит                | 7    | 6      | 5    | 4  | 3  | 2  | 1  | 0  |
| Назначение         | DATA |        |      |    |    |    |    |    |
| Начальное значение | 0    |        |      |    |    |    |    |    |

**RW** – операция чтения или записи:

1 – прочитать данные;

0 – записать данные.

**CE\_RAM** – обращение в область памяти RAM:

1 – обращение в RAM-память или к ведомым устройствам системы;

0 – область RAM не задействуется.

**ADDR** – адрес обращения.

**DATA** – данные для записи. Младший бит данных находится в бите 0.

При чтении пользовательского тестового регистра данных, считанные с системной шины данные будут переданы со следующим за командой чтения пакетом данных в формате, представленном ниже:

| Бит                | 23     | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--------------------|--------|----|----|----|----|----|----|----|
| Назначение         | Резерв |    |    |    |    |    |    |    |
| Начальное значение | x      |    |    |    |    |    |    |    |
| Бит                | 15     | 14 | 13 | 12 | 11 | 10 | 9  | 8  |
| Назначение         | Резерв |    |    |    |    |    |    |    |
| Начальное значение | x      |    |    |    |    |    |    |    |
| Бит                | 7      | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| Назначение         | DATA   |    |    |    |    |    |    |    |
| Начальное значение | 0      |    |    |    |    |    |    |    |

При этом во время чтения должна отправляться команда с CE\_RAM равным «0». Передача прочитанных данных осуществляется в соответствии со стандартом IEEE Std 1149.1-2001. Примечание: выход из инструкции не обнуляет пользовательский тестовый регистр данных.

Обращаем внимание, документация носит ознакомительный характер.

При разработке аппаратуры необходимо руководствоваться КД: технические условия АЕНВ.431260.544ТУ, карта заказа КФЦС.431260.012-003Д16

**Инструкция «JTAG\_ROM\_ACCESS»**

Инструкция «JTAG\_ROM\_ACCESS» представляет собой дополнительную пользовательскую инструкцию с кодом 100b, в результате выбора которой CPU 8051 будет отключен от системной шины, а в место него будет подключен тестовый пользовательский регистр данных разрядностью 24 бит:

|                    |      |        |      |    |    |    |    |    |
|--------------------|------|--------|------|----|----|----|----|----|
| Бит                | 23   | 22     | 21   | 20 | 19 | 18 | 17 | 16 |
| Назначение         | RW   | CE_ROM | ADDR |    |    |    |    |    |
| Начальное значение | 0    |        |      |    |    |    |    |    |
| Бит                | 15   | 14     | 13   | 12 | 11 | 10 | 9  | 8  |
| Назначение         | ADDR |        |      |    |    |    |    |    |
| Начальное значение | 0    |        |      |    |    |    |    |    |
| Бит                | 7    | 6      | 5    | 4  | 3  | 2  | 1  | 0  |
| Назначение         | DATA |        |      |    |    |    |    |    |
| Начальное значение | 0    |        |      |    |    |    |    |    |

**RW** – операция чтения или записи;

1 – прочитать данные;

0 – записать данные.

**CE\_ROM** – обращение в область памяти ROM:

1 – обращение в ROM-память;

0 – область ROM не задействуется.

**ADDR** – адрес обращения.

**DATA** – данные для записи. Младший бит данных находится в бите 0.

При чтении пользовательского тестового регистра данных, считанные с системной шины данные будут переданы со следующим за командой чтения пакетом данных в формате, представленном ниже:

| Бит                | 23     | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--------------------|--------|----|----|----|----|----|----|----|
| Назначение         | Резерв |    |    |    |    |    |    |    |
| Начальное значение | x      |    |    |    |    |    |    |    |
| Бит                | 15     | 14 | 13 | 12 | 11 | 10 | 9  | 8  |
| Назначение         | Резерв |    |    |    |    |    |    |    |
| Начальное значение | x      |    |    |    |    |    |    |    |
| Бит                | 7      | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| Назначение         | DATA   |    |    |    |    |    |    |    |
| Начальное значение | 0      |    |    |    |    |    |    |    |

При этом во время чтения должна отправляться команда с CE\_ROM равным «0». Передача прочитанных данных осуществляется в соответствии со стандартом IEEE Std 1149.1-2001.

Примечание: выход из инструкции не обнуляет пользовательский тестовый регистр данных.

**Инструкция «IF\_ACCESS»**

Инструкция «IF\_ACCESS» («Интерфейс доступа») представляет собой дополнительную пользовательскую инструкцию с кодом 101b и пользовательским тестовым регистром данных разрядностью 19 бит.

Примечание: выход из инструкции не обнуляет пользовательский тестовый регистр данных.

**Инструкция «DEBUGGER»**

Инструкция «DEBUGGER» («Отладчик») представляет собой дополнительную пользовательскую инструкцию с кодом 110b и пользовательским тестовым регистром данных разрядностью 5 бит.

Примечание: выход из инструкции не обнуляет пользовательский тестовый регистр данных.

## Лист регистрации изменений

| Дата       | Версия | Изменения                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24.11.2021 | 1.0    | Исходная версия                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 10.03.2022 | 1.1    | Обновлены регистры                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23.08.2022 | 1.2    | Добавлено примечание к схеме применения (стр. 6);<br>Обновлена таблица 5;<br>Обновлены регистры SPI (стр. 44-48);<br>Обновлены регистры ЦАП (стр. 93)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 08.08.2023 | 1.3    | Обновлен пункт «Основные особенности»;<br>– обновлена структурная схема.<br>Обновлен пункт «Общее описание»;<br>Обновлен пункт «Электрические параметры микросхемы»:<br>– добавлен ток потребления линейных регуляторов.<br>– добавлена нагрузочная способность выводов GPIO.<br>Добавлен пункт «Справочные данные»<br>Обновлен пункт «Предельно-допустимые и предельные режимы эксплуатации»:<br>– обновлена таблица 2.<br>Обновлен пункт «Конфигурация и функциональное описание выводов»;<br>– обновлена таблица 3.<br>Обновлен пункт «Рекомендуемая схема применения»:<br>– обновлен рисунок 3;<br>– добавлена схема применения при питании 5,0 В (рисунок 4).<br>Обновлен пункт «Техническое описание»:<br>– обновлен рисунок 5;<br>– обновлена таблица 6.<br>Обновлен пункт «Тактирование микроконтроллера»:<br>– расширено описание.<br>– добавлен график зависимости частоты RC-генератора от температуры (рисунок 5)<br>Добавлен пункт «АЦП»;<br>Обновлен пункт «ANALOG_CFG». |
| 02.10.2023 | 1.4    | Изменена структура документа.<br>Обновлен пункт «Основные особенности»;<br>– обновлена структурная схема.<br>Обновлен пункт «Электрические параметры микросхемы»:<br>– обновлена таблица 1.<br>Обновлен пункт «Предельно-допустимые и предельные режимы эксплуатации»:<br>– обновлена таблица 2.<br>Добавлен пункт «Важные замечания при работе с микросхемой».<br>Скорректированы схемы применения (Рисунки 3, 4);<br>Обновлен пункт «ANALOG_CFG»;<br>Обновлен пункт «Техническое описание»:<br>– обновлен рисунок 7.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 15.11.2023 | 1.5    | Обновлен пункт «Электрические параметры микросхемы»:<br>– обновлена таблица 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

[illegible]