



# Table of contents:

## Общая информация

- Графическое представление
- Общее описание

## Основные характеристики

- Электрические параметры микросхемы
  - Таблица 1. Электрические характеристики (температурный диапазон от  $-60^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$ )
- Электростатическая защита
- Предельно-допустимые и предельные режимы эксплуатации

## Порты

- Распиновка
  - Таблица выводов

## Рекомендуемая схема применения

## Выбор схемы питания микросхемы

## Типы датчиков

- Датчики типа СКВТ (Resolver)
- Датчики типа сельсин (Synchro)
- Датчики типа ЛРДТ (LVDT)

## Архитектура и принцип работы

- Принцип преобразования сигналов СКВТ в код угла
- Архитектура микросхемы

## Программируемый генератор

- Программируемый генератор
- Генератор синусоидальных сигналов
- Генерация импульсного сигнала
- Вывод постоянного уровня

## Входные группы преобразователя

- Включение
- Компенсатор смещения

## Блок масштабирования и преобразования координат

## Блок восстановления сигнала опорной частоты

## Следящий контур

## Блок модели датчика

- Модель датчика в режиме СКВТ/сельсин
- Модель ЛРДТ по 5-проводной схеме

## Полоса пропускания контура

- Полоса пропускания преобразователя

## Блок обработки результатов

## Эмуляция квадратурного энкодера

- Эмуляция квадратурного энкодера

## Блок определения ошибок подключения

## Интерфейс SPI

- Описание структуры SPI
- Коды операций
- Временные параметры последовательного интерфейса
- Дополнительные режимы SPI
  - Фиксация значения выходной ячейки для множественного чтения.
  - Чтение ранее переданной транзакции
  - Блокировка записи слов
  - Программный адрес
  - Управление внешним приемо-передатчиком с помощью вывода VC
- Режим параллельной передачи результата
- Назначение выводов в режиме параллельной передачи результата

## Тактирование

- Выбор источника тактирования
- Настройка умножителя частоты в ФАПЧ
- Настройка частоты FINT
- Настройка частоты АЦП
- Выключение тактирования

## Постоянное запоминающее устройство

- Структура
  - ПЗУ с параллельным доступом
  - ПЗУ с последовательным доступом

## Подсистема сброса

- Таблица этапов распространения сигнала сброс

## Микровычислители

- Архитектура конвейера
- Регистры общего назначения
- Структура памяти
  - Ячейки обмена данными CPU
  - Ячейки обмена с интерфейсом SPI
  - Буферы
  - Адресация памяти

- Форматы данных
  - 28-битный формат с плавающей запятой
    - Структура формата
    - Кодирование (Float32 → Float28)
    - Float28 → Float32
    - Float28 → Integer
- Система команд
  - Арифметико-логические команды
  - Команды сдвига
  - Команды умножения и преобразования форматов
  - Команды работы с памятью
  - Индексные команды работы с памятью
  - Команды условного перехода
  - CORDIC-команды
  - Канальные регистры
  - Команды управления HAND-интерфейсом
  - Команды управления и синхронизации
- Взаимодействие с преобразователем
  - Запись данных в преобразователь
  - Запись Coord/Vel
  - Запись Pole\_addi
- Синхронизация двух CPU
- Отладчик
  - Управление выполнением
  - Чтение регистров
- Пример программы
  - CPU1: Коррекция угла по 1-й гармонике
- Сводная таблица команд

#### Регистры конфигурации

- Таблица адресов регистров
  - C1KampS/C2KampS
  - C1KampC/C2KampC
  - C1KbiasS/C2KbiasS
  - C1KbiasC/C2KbiasC
  - C1fbias/C2fbias
  - C1ExPhShft/C2ExPhShft
  - C1ExoStngs/C2ExoStngs

- Поля регистра
- C1EXInc/C2EXInc
- C1Amp\_th/C2Amp\_th
  - Поля регистра
- C1InputStngs/C2InputStngs
  - Поля регистра
- C1Lock\_th/C2Lock\_th
- C1Zero/C2Zero
- C1Mask/C2Mask
  - Поля регистра
- C1KonturStngs/C2KonturStngs
  - Поля регистра
- C1ResCntrl/C2ResCntrl
  - Поля регистра
- C1Vcnt\_bound/C2Vcnt\_bound
- C1Coord/C2Coord
- C1CoordHB/C2CoordHB
- C1AdcS/C2AdcS
  - Поля регистра
- C1AdcC/C2AdcC
  - Поля регистра
- C1OutS/C2OutS
  - Поля регистра
- C1VirtualS/C2VirtualS
  - Поля регистра
- C1Err\_metric/C2Err\_metric
- C1Amp\_metric/C2Amp\_metric
- C1Vel/C2Vel
- C1VelHB/C2VelHB
- C1PhiS/C2PhiS
  - Поля регистра
- C1PhiC/C2PhiC
  - Поля регистра
- C1OutC/C2OutC
  - Поля регистра
- C1VirtualC/C2VirtualC
  - Поля регистра

- C1Stat/C2Stat
  - Поля регистра
- C1Pole\_addi/C2Pole\_addi
- IC\_addr
- ADC\_config
  - Поля регистра
- Mask\_Stat
  - Поля регистра
- Flags\_delay
- WR\_lock
- CMP\_lth
- AFE\_config
  - Поля регистра
- Mode\_config
  - Поля регистра
- NOCLK\_stat
  - Поля регистра
- SPI\_req
- alive\_cnt
- Stat\_main
  - Поля регистра
- Dcpu1LB
- Dcpu1HB
- Dcpu2LB
- Dcpu2HB
- PLL\_config
  - Поля регистра
- INIT\_conf
  - Поля регистра
- UOTP\_ctrl
  - Поля регистра
- BUS\_addr
- BOTP\_addr
- BOTP\_data
- BOTP\_ctrl
  - Поля регистра
- BOTP\_out

- P1BG\_ctrl/P2BG\_ctrl
  - Поля регистра
- P1BG\_data/P2BG\_data
  - Поля регистра

#### Методика подключения и настройки

- 1. Проверка подключения
- 2. Настройка тактовой частоты микросхемы
- 3. Настройка режима работы аналоговой части
- 4. Настройка сигнала возбуждения датчика (на примере канала 1)
- 5. Настройка входных каскадов (на примере канала 1)
- 5. Диагностика подключения (на примере канала 1)
- 6. Калибровка усиления (на примере канала 1)
- 7. Настройка флагов состояния (на примере канала 1)
- 8. Пример настройки для подключения СКВТ
- 9. Пример настройки для подключения сельсин
- 10. Пример настройки для подключения ЛРДТ
- 11. Блокировка записи и инициализация из ПЗУ

#### Описание ПО

# Общая информация

5400TP065A-022 – Микросхема преобразователя сигналов с датчиков типа сельсин, СКВТ и ЛРДТ в цифровой код угла

## ЗАМЕТКА

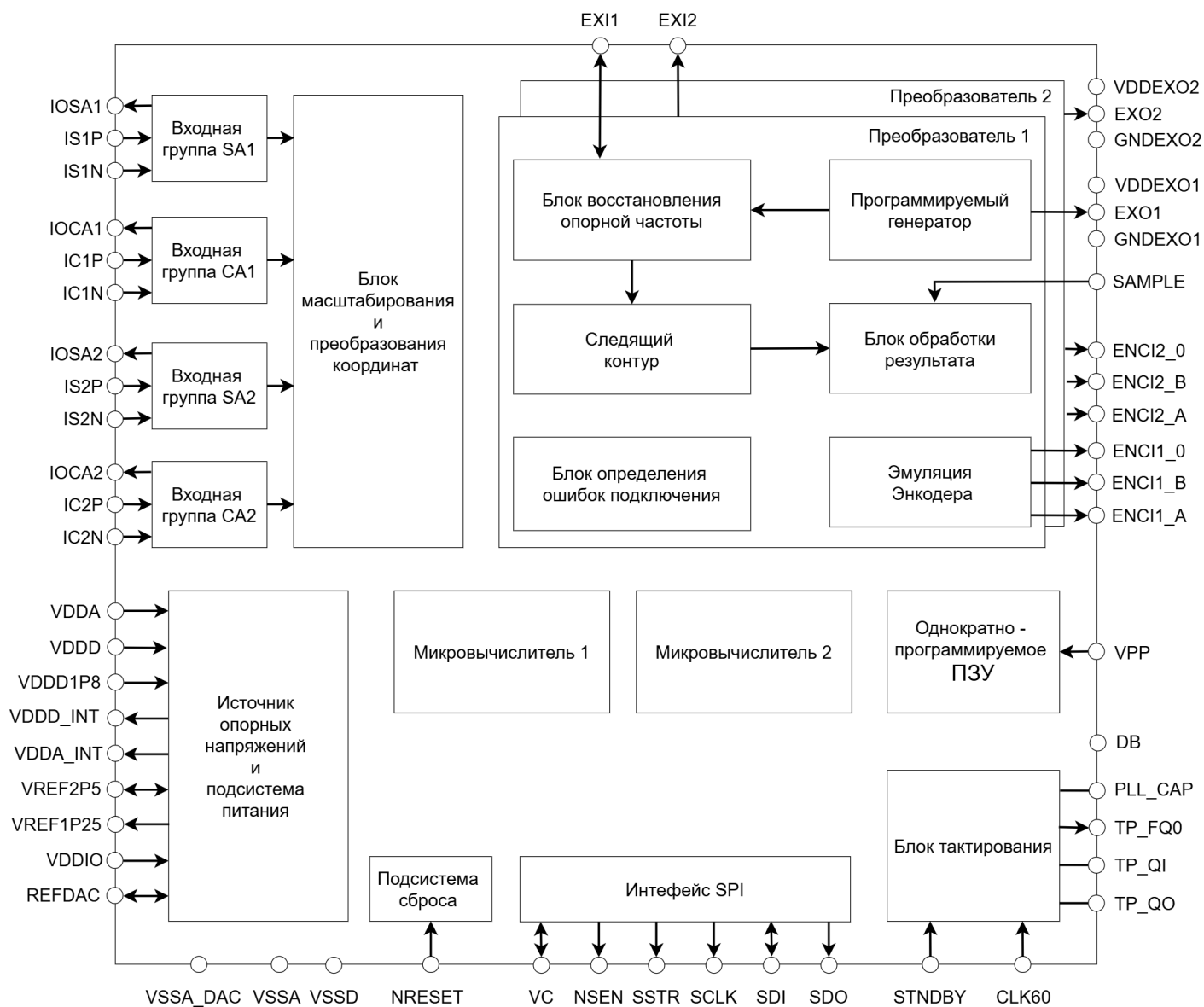
Дата последнего обновления документации 04.05.2026

## ПОДСКАЗКА

Скачать pdf версию документации [5400TP065A-022.pdf](#)

- Напряжения питания 5,0 В;
- Период обновления информации не более 2 мкс;
- Частота возбуждения датчиков от 0 до 30 кГц;
- Разрядность выходной информации от 8 бит до 16 бит;
- Два независимых преобразователя;
- Два независимых генератора опорных сигналов с частотой от 20 Гц до 30 кГц;
- Эмуляция квадратурного энкодера;
- Последовательный (SPI) интерфейс;
- Температурный диапазон от  $-60^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$ .

## Графическое представление



Блок схема

## Общее описание

Микросхема предназначена для преобразования сигналов с датчиков типа сельсин, СКВТ (синусно-косинусный вращающийся трансформатор) и датчиков линейного перемещения – ЛРДТ (линейный регулируемый дифференциальный трансформатор) (см. [раздел типы датчиков](#)). Микросхема в своем составе содержит программируемый генератор возбуждающего напряжения и два следящих контура, производящих вычисление угла поворота вала или перемещения сердечника ЛРДТ. Микросхема выполнена в 64-выводном металлокерамическом корпусе МК 5153.64-3.



# Основные характеристики

## Электрические параметры микросхемы

Таблица 1. Электрические характеристики  
(температурный диапазон от  $-60^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$ )

| Параметр, единица измерения  | Букв. обозн.         | Не менее | Не более | Номер выв.           |
|--|----------------------|----------|----------|----------------------|
| Статический ток потребления в режиме покоя, мА                           | $I_{\text{ПОТ1}}$    | —        | 10       | 33, 63               |
| Ток утечки высокого уровня по цифровым выводам <sup>2</sup> , мкА        | $I_{1\text{УТ}}$     | —        | 10       | 52, 53, 55-59, 62    |
| Ток утечки низкого уровня по цифровым выводам <sup>2</sup> , мкА         | $I_{0\text{УТ}}$     | —        | 10       | 52, 53, 55-59, 62    |
| Динамический ток потребления, мА   | $I_{\text{ПОТ2}}$    | —        | 200      | 33, 63               |
| Время задержки, нс   | $t_{\text{ЗД}}$      | —        | 80       | 51, 55               |
| Выходное напряжение высокого уровня по цифровым выводам <sup>3</sup> , В | $U_{1\text{ВЫХ}}$    | 2,4      | —        | 41-44, 46-51, 53     |
| Выходное напряжение низкого уровня по цифровым выводам <sup>4</sup> , В  | $U_{0\text{ВЫХ}}$    | —        | 0,4      | 41-44, 46-51, 53, 62 |
| Выходное напряжение ИОН для формирования опорного напряжения АЦП, В      | $U_{\text{ВЫХ\_ОП}}$ | 2,25     | 2,75     | 11                   |

| Параметр, единица измерения   | Букв. обозн.         | Не менее                    | Не более | Номер выв. |
|---|----------------------|-----------------------------|----------|------------|
| Выходное напряжение ИОН схемы смещения постоянной составляющей входных сигналов, В                      | $U_{\text{ВЫХ\_СМ}}$ | 1,125                       | 1,375    | 12         |
| Выходное напряжение встроенного линейного регулятора 4 В питания цифровой части, В                      | $U_{\text{ЛР\_ЦИФ}}$ | 3,7                         | 4,3      | 31         |
| Выходное напряжение встроенного линейного регулятора 4 В питания аналоговой части, В                    | $U_{\text{ЛР\_АН}}$  | 3,7                         | 4,3      | 30         |
| Выходное напряжение встроенного линейного регулятора 1,8 В питания цифрового ядра, В                    | $U_{\text{ЛР\_1.8}}$ | 1,62                        | 1,98     | 36, 61     |
| Максимальное выходное напряжение по выводу EXO1 <sup>5</sup> , В  | $U_{\text{МАКС1}}$   | $U_{\text{ВХ\_БУФ1}} - 0,7$ | —        | 6          |
| Максимальное выходное напряжение по выводу EXO2 <sup>5</sup> , В  | $U_{\text{МАКС2}}$   | $U_{\text{ВХ\_БУФ2}} - 0,7$ | —        | 3          |
| Минимальная частота выходного сигнала (сигнала опорной частоты), основная гармоника <sup>5</sup> , Гц   | $f_{\text{МИН}}$     | —                           | 20       | 3, 6       |
| Минимальное выходное напряжение по выводам EXO1, EXO2 <sup>5</sup> , В                                  | $U_{\text{МИН}}$     | —                           | 0,7      | 3, 6       |
| Максимальная частота выходного сигнала (сигнала опорной частоты), основная гармоника <sup>5</sup> , кГц | $f_{\text{МАКС}}$    | 30                          | —        | 3, 6       |

| Параметр, единица измерения  | Букв. обозн. | Не менее | Не более | Номер выв.             |
|--|--------------|----------|----------|------------------------|
| Входной ток на выводах VREF2P5 в режиме подключения внешнего источника напряжения 2,55 В, мА         | $I_{ВХ}$     | —        | 1        | 11                     |
| Полная погрешность определения угла при преобразовании сигналов датчика типа СКВТ при F=1,5 кГц, ЕМР | $E_A$        | -20      | 20       | 21, 24, 50, 18, 15, 49 |

<sup>1</sup> Допускается уточнять нормы на параметры до проведения испытаний. <sup>2</sup> Выводы: CLK60, EXI2, EXI1, SDI, VC, SCLK, nSEN, SSTR, STNDBY, SAMPLE. <sup>3</sup> Выводы: EXI1, EXI2, SDO, VC, ENC1\_0, ENC1\_A, ENC1\_B, ENC2\_0, ENC2\_A, ENC2\_B, DB, SDI. <sup>4</sup> Выводы: EXI1, EXI2, SDO, VC, ENC1\_0, ENC1\_A, ENC1\_B, ENC2\_0, ENC2\_A, ENC2\_B, DB, SDI, NRESET. <sup>5</sup> В режиме выдачи синусоидального напряжения.

| Наименование параметра, единица измерения, режим измерения                              | Буквенное обозначение параметра | не менее | типовое значение | не более |
|---|---------------------------------|----------|------------------|----------|
| <b>Параметры генератора сигнала возбуждения датчика</b>                                 |                                 |          |                  |          |
| Шаг изменения частоты генератора, Гц, при $f_{CLK} = 500$ кГц                           | $f_{STP}$                       | —        | 0,477            | —        |
| Динамический диапазон выходного сигнала без шума и искажения, дБ, на выходах EXO1, EXO2 | $SFDR_O$                        | 20       | —                | —        |

| Наименование параметра, единица измерения, режим измерения  | Буквенное обозначение параметра | не менее | типовое значение | не более |
|---|---------------------------------|----------|------------------|----------|
| Дифференциальная нелинейность, ЕМР, ЦАП генератора  | $E_{LD\_DAC}$                   | -1       | —                | 1        |
| Интегральная нелинейность, ЕМР, ЦАП генератора  | $E_{L(adj)\_DAC}$               | -1       | —                | 1        |
| <b>Параметры АЦП и буферного усилителя</b>  |                                 |          |                  |          |
| Входная емкость буферного усилителя, пФ, на парах входов ISP1-ISN1, ICP1-ICN1, ISP2-ISN2, ICP2-ICN2 | $C_I$                           | —        | —                | —        |
| Частота дискретизации АЦП, МГц  | $f_{CLK}$                       | —        | 0,5              | 0,5      |
| Число разрядов АЦП, бит   | $N_{ADC}$                       | 12       | —                | —        |
| Смещение АЦП, мВ  | $E_{0\_ADC}$                    | -10      | —                | 10       |
| Смещение буферного усилителя и АЦП, мВ  | $E_{0\_ADC}$                    | -10      | —                | 10       |
| Погрешность коэффициента преобразования АЦП, %  | $E_{G\_ADC}$                    | -1       | —                | 1        |
| Погрешность коэффициента преобразования, %, буферного усилителя и АЦП                               | $E_{G\_ADC}$                    | —        | —                | 1        |
| Дифференциальная нелинейность АЦП, ЕМР  | $E_{LD\_ADC}$                   | -1       | —                | 1        |

| Наименование параметра, единица измерения, режим измерения  | Буквенное обозначение параметра | не менее | типичное значение | не более |
|---|---------------------------------|----------|-------------------|----------|
| Интегральная нелинейность АЦП, ЕМР  | $E_{L\_ADC}$                    | -2       | —                 | 2        |
| Время преобразования АЦП, мкс   | $t_{c\_ADC}$                    | —        | $2/f_{CLK}$       | —        |
| <b>Параметры преобразования</b>   |                                 |          |                   |          |
| Погрешность, вносимая в измерения цифровым контуром, угловых минут  | $E_{DIG}$                       | -0,164   | —                 | 0,164    |
| Погрешность преобразования координаты из трехфазной в декартову (сельсин - СКВТ), угловых минут   | $E_{sync}$                      | -0.1     | —                 | 0,1      |
| Дифференциальная нелинейность, ЕМР, скорость и координата, - при $U_{CC} < 4,5 \text{ В}$ , $0,4 \text{ В} \leq U_{ЮА} \leq 1,4 + 0,4(U_{CC}^* - 3)$ ; - при $U_{CC} \geq 4,5 \text{ В}$ $0,2 \text{ В} \leq U_{ЮА} \leq U_{REF} - 0,5$ | $E_{LD}$                        | -0.9     | —                 | 0,9      |
| Диапазон настройки сдвига фазы напряжений в датчике, градусов   | $\varphi_{RANGE}$               | 0        | —                 | 360      |
| Шаг настройки сдвига фазы напряжений в датчике, градусов  | $\varphi_{STEP}$                | 0.004    | -                 | 10.8     |
| Диапазон допустимого изменения сдвига фазы напряжений в датчике в   | $\varphi_{TRACT}$               | 0        | —                 | $\pm 40$ |

| Наименование параметра, единица измерения, режим измерения   | Буквенное обозначение параметра | не менее | типичное значение          | не более |
|--|---------------------------------|----------|----------------------------|----------|
| процессе работы преобразователя, градусов  |                                 |          |                            |          |
| Время нахождения угла после снятия сигнала сброс, мкс, LBW = 5   | $T_{START}$                     | —        | $100 + 2/f_{IN} + T_{SR5}$ | —        |
| Время достижения заявленных параметров преобразования после снятия сигнала сброс, мс, $f_{IN} \geq 380$ Гц | $T_{DONE}$                      | —        | 300                        | -        |
| Разрядность преобразователя (координата), бит  | $N_C$                           | 16       | —                          | —        |
| Разрядность преобразователя (скорость), бит  | $N_V$                           | 15       | —                          | —        |

## 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 3. Предельно-допустимые и предельные режимы эксплуатации микросхем

| Параметр, единица измерения | Предельно-допустимый режим | Предельный режим |
|-----------------------------|----------------------------|------------------|
|-----------------------------|----------------------------|------------------|

|  | <b>не менее</b>             | <b>не более</b>           | <b>не менее</b> | <b>не более</b> |
|--|-----------------------------|---------------------------|-----------------|-----------------|
| Напряжение питания аналоговой части (Uпит1), В                 | 4,5                         | 5,25                      | -0,3            | 5,35            |
| Напряжение питания цифровой части (Uпит2), В                   | 4,5                         | 5,25                      | -0,3            | 5,35            |
| Напряжение питания интерфейсной части (Uпит3), В               | 3                           | 3,6                       | -0,3            | 4               |
| Входное опорное напряжение АЦП (Uвх_оп), В                     | 2,125                       | 2,875                     | 0               | 5,35            |
| Входное напряжение высокого уровня цифровых выводов, В         | $0,8 \cdot U_{\text{пит3}}$ | $U_{\text{пит3}} + 0,3^2$ | -0,3            | 4               |
| Входное напряжение низкого уровня цифровых выводов, В          | 0                           | 0,6                       | -0,3            | 4               |
| Входное напряжение питания выходного буфера EXO1 (Uвх_буф1), В | 3                           | 5,25                      | -0,3            | 5,35            |
| Входное напряжение питания выходного буфера EXO2 (Uвх_буф2), В | 3                           | 5,25                      | -0,3            | 5,35            |
| Температура эксплуатации, °С                                   | -60                         | +125                      | -60             | +150            |

<sup>1</sup> Допускается уточнять электрические режимы после проведения испытаний. <sup>2</sup> Не более 5,25 В.



| <b>№ вывода</b>    | <b>Наименование вывода</b> | <b>Тип вывода</b> | <b>Описание</b>  |
|--------------------|----------------------------|-------------------|--|
| 1                  | VSSA_DAC                   | PWR               | Общий вывод опорного напряжения генератора   |
| 2                  | GNDEXO2                    | PWR               | Общий вывод генератора опорной частоты 2   |
| 3                  | EXO2                       | AO                | Выход генератора опорной частоты 2   |
| 4                  | VDDEXO2                    | PWR               | Питание генератора опорной частоты 2   |
| 5                  | VDDEXO1                    | PWR               | Питание генератора опорной частоты 1   |
| 6                  | EXO1                       | AO                | Выход генератора опорной частоты 1   |
| 7                  | GNDEXO1                    | PWR               | Общий вывод генератора опорной частоты 1   |
| 8, 9               | NC                         | -                 | Не используется  |
| 10, 29, 34, 54, 60 | VSSD                       | PWR               | Общий вывод питания цифровой части   |
| 11                 | VREF2P5                    | AI/AO             | Вход-выход источника опорного напряжения АЦП. Подключен к шине «общий» через конденсатор |
| 12                 | VREF1P25                   | AO                | Выход источника опорного напряжения АЦП. Подключен к шине «общий» через конденсатор      |
| 13                 | IC2P                       | AI                | Прямой вход усилителя cos2   |
| 14                 | IC2N                       | AI                | Инверсный вход усилителя cos2  |

| <b>№ вывода</b> | <b>Наименование вывода</b> | <b>Тип вывода</b> | <b>Описание</b>                                  |
|-----------------|----------------------------|-------------------|--|
| 15              | IOCA2                      | AO                | Выход буферного усилителя cos2 (вход АЦП cos2)   |
| 16              | IS2P                       | AI                | Прямой вход усилителя sin2                       |
| 17              | IS2N                       | AI                | Инверсный вход усилителя sin2                    |
| 18              | IOSA2                      | AO                | Выход буферного усилителя sin2 (вход АЦП sin2)   |
| 19              | IC1P                       | AI                | Прямой вход усилителя cos1                       |
| 20              | IC1N                       | AI                | Инверсный вход усилителя cos1                    |
| 21              | IOCA1                      | AO                | Выход буферного усилителя cos1 (вход АЦП cos1)   |
| 22              | IS1P                       | AI                | Прямой вход усилителя sin1                       |
| 23              | IS1N                       | AI                | Инверсный вход усилителя sin1                    |
| 24              | IOSA1                      | AO                | Выход буферного усилителя sin1 (вход АЦП sin1)   |
| 25              | VDDIO2                     | PWR               | Питание генератора на кварцевом резонаторе       |
| 26              | TP_FQO                     | DO                | Выход частоты генератора на кварцевом резонаторе |
| 27              | TP_QO                      | AO                | Вывод для подключения кварцевого резонатора      |

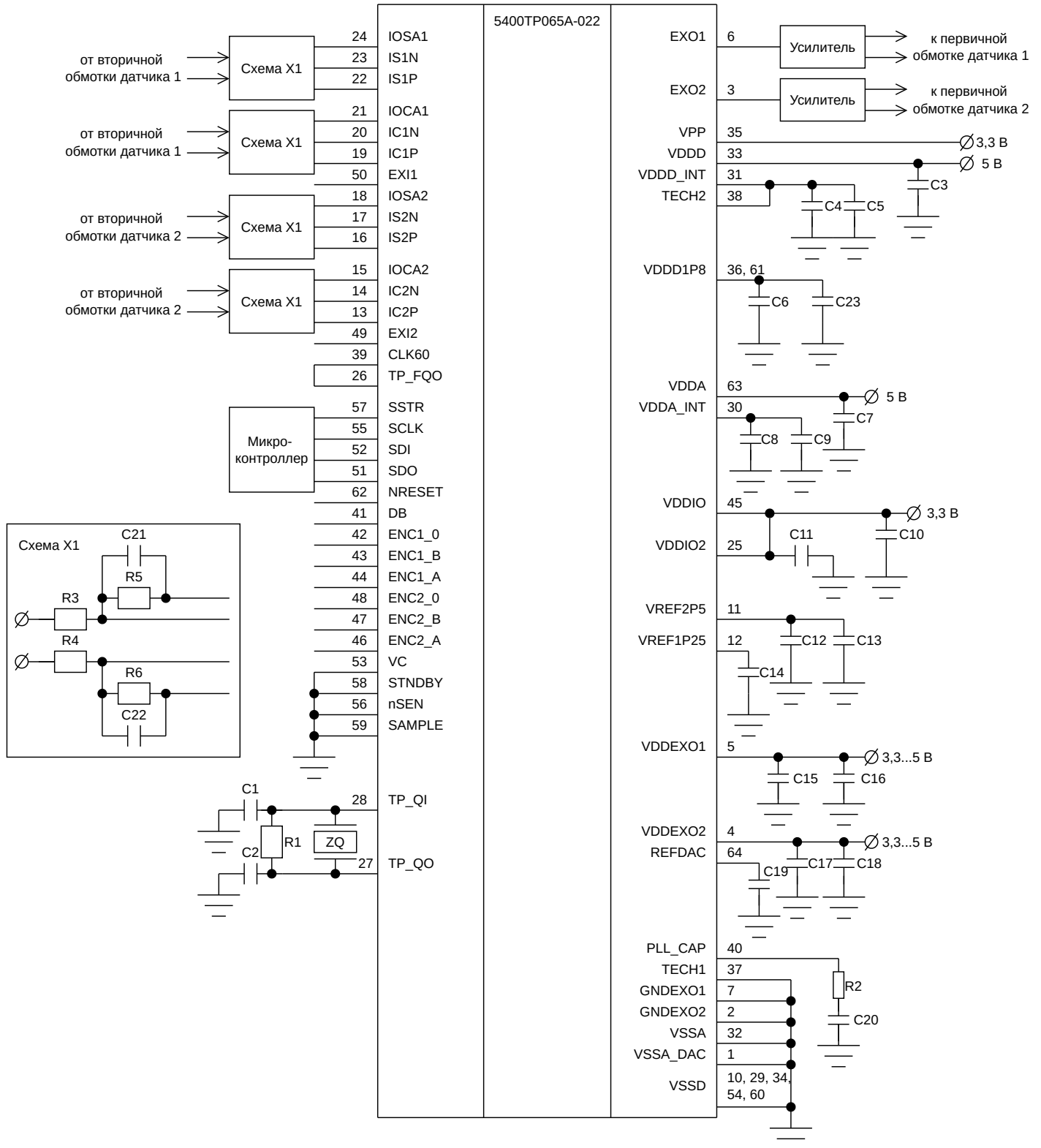
| <b>№ вывода</b> | <b>Наименование вывода</b> | <b>Тип вывода</b> | <b>Описание</b>  |
|-----------------|----------------------------|-------------------|--|
| 28              | TP_QI                      | AI                | Вывод для подключения кварцевого резонатора  |
| 30              | VDDA_INT                   | PWR               | Вывод питания аналоговой части   |
| 31              | VDDD_INT                   | PWR               | Вывод питания цифровой части   |
| 32              | VSSA                       | PWR               | Общий вывод питания аналоговой части   |
| 33              | VDDD                       | PWR               | Вход линейного регулятора питания цифровой части   |
| 35              | VPP                        | AI                | Вывод напряжения программирования  |
| 36, 61          | VDDD1P8                    | PWR               | Вывод питания ядра 1,8 В   |
| 37              | TECH1                      | AO                | Служебный вывод  |
| 38              | TECH2                      | AI                | Служебный вывод  |
| 39              | CLK60                      | AI                | Вход тактового сигнала   |
| 40              | PLL_CAP                    | AO                | Вывод петлевого фильтра умножителя частот  |
| 41              | DB                         | DO                | Вывод отладки режимов и программ   |
| 42              | ENC1_0                     | DO                | Выход эмуляции квадратурного энкодера, датчик 1, канал 0. Выход дополнительного канала SPI |
| 43              | ENC1_B                     | DO                | Выход эмуляции квадратурного энкодера, датчик 1, канал B. Выход дополнительного канала SPI |

| <b>№ вывода</b> | <b>Наименование вывода</b> | <b>Тип вывода</b> | <b>Описание</b>   |
|-----------------|----------------------------|-------------------|---|
| 44              | ENC1_A                     | DO                | Выход эмуляции квадратурного энкодера, датчик 1, канал А. Выход дополнительного канала SPI  |
| 45              | VDDIO                      | PWR               | Питание портов ввода-вывода   |
| 46              | ENC2_A                     | DO                | Выход эмуляции квадратурного энкодера, датчик 2, канал А. Выход дополнительного канала SPI  |
| 47              | ENC2_B                     | DO                | Выход эмуляции квадратурного энкодера, датчик 2, канал В. Выход дополнительного канала SPI  |
| 48              | ENC2_0                     | DO                | Выход эмуляции квадратурного энкодера, датчик 2, канал 0. Выход дополнительного канала SPI  |
| 49              | EXI2                       | DI/DO             | Выход опорного напряжения, датчик 2.  |
| 50              | EXI1                       | DI/DO             | Выход опорного напряжения, датчик 1.<br>Вход опорного напряжения для датчика 1 и 2  |
| 51              | SDO                        | DO                | Выход данных SPI  |
| 52              | SDI                        | DI/DO             | Вход/выход данных SPI   |
| 53              | VC                         | DI/DO             | Вход/выход. В режиме прямой/параллельной передачи угла выбирает угол или скорость. В остальных режимах управляет внешними приемопередатчиками |

| <b>№ вывода</b> | <b>Наименование вывода</b> | <b>Тип вывода</b> | <b>Описание</b>  |
|-----------------|----------------------------|-------------------|--|
| 55              | SCLK                       | DI                | Тактовый сигнал SPI  |
| 56              | NSEN                       | DI                | Выбор микросхемы. Активный уровень – лог. «0»                    |
| 57              | SSTR                       | DI                | Строб SPI  |
| 58              | STNDBY                     | DI                | Режим пониженного энергопотребления: Активный уровень – лог. «1» |
| 59              | SAMPLE                     | DI                | Строб сэмплирования координаты и скорости для SPI                |
| 62              | NRESET                     | DI                | Сброс микросхемы. Активный уровень – лог. «0»                    |
| 63              | VDDA                       | PWR               | Вход регулятора питания аналоговой части                         |
| 64              | REFDAC                     | AI/AO             | Вход/выход опорного напряжения генератора                        |
| EP              | VSSA                       | PWR               | Общий вывод  |

Последнее обновление **17 июн. 2026 г.** от **Ilya Slepchenkov**

# Рекомендуемая схема применения



## Рекомендуемая схема применения

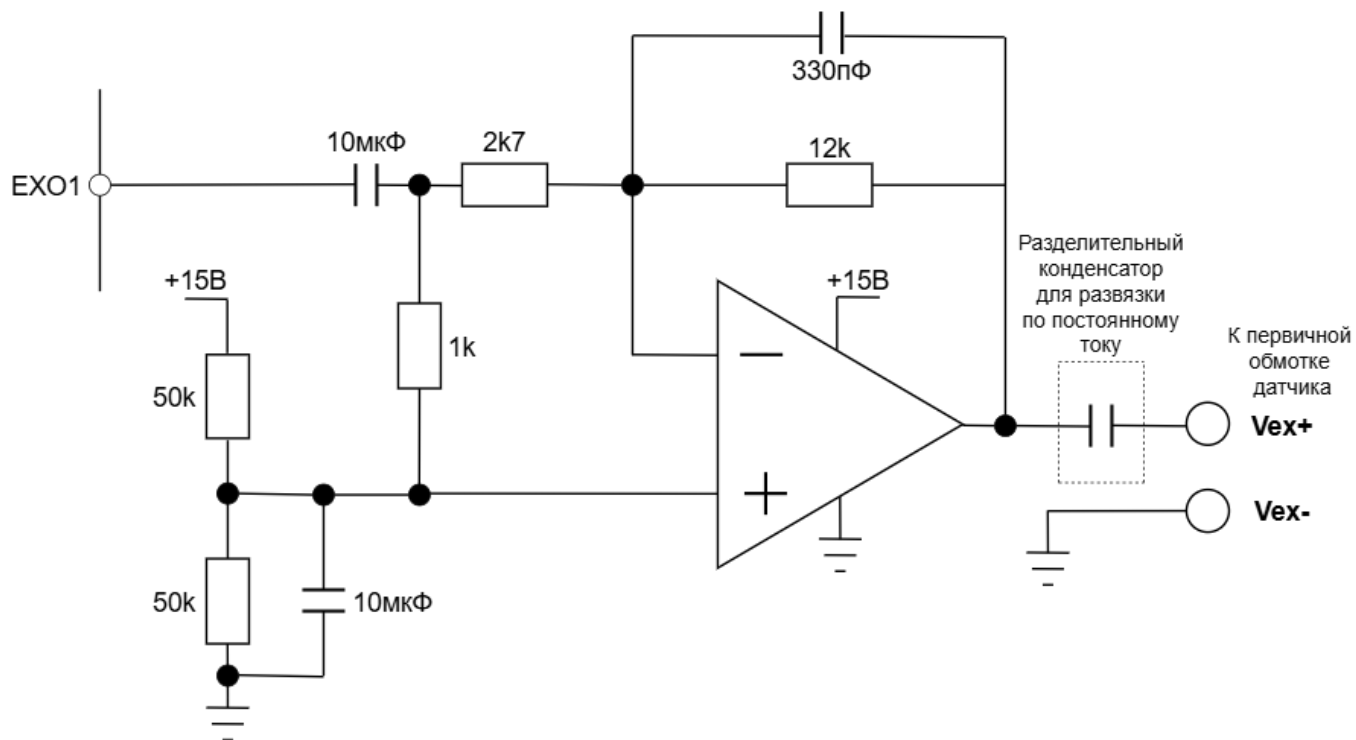
Таблица 4. Таблица внешних компонентов

| Обозначение  | Номинал | Примечание  |
|--|---------|---|
| R1   | 510 кОм |   |
| R2   | 20 Ом   |   |
| R3, R4   | 27 кОм  |   |
| R5, R6   | 10 кОм  |   |
| C1, C2   | 12 пФ   |   |
| C3, C4, C6, C7, C8, C10, C11, C12, C14, C16, C18, C19, C20 | 0,1 мкФ |   |
| C5, C9, C13, C15, C17                                      | 4,7 мкФ | Не обязательны, рекомендуются для выравнивания заряда |
| C21, C22   | 330 пФ  |   |
| ZQ   | 10 МГц  | Кварцевый резонатор                                   |

Конденсаторы либо 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.

**Примечания:** Допускается подключение **VDDIO** (45) к питанию **VDDD** (33) или **VDDD\_INT** (31). Для программирования ячейки OTP памяти вывод **VPP** (35) запитывается  $9,8 \text{ В} \pm 0,3 \text{ В}$ , для чтения ячеек вывод **VPP** (35) запитывается 3,3 В. При **VREF\_en** = 0 (регистр **AFE\_config**) вывод **VREF2P5** (11) должен быть запитан опорным уровнем 2,5 В. При использовании

внешнего тактового генератора подключенного к выводу CLK60 (39) вывод TP\_QI (28) подключить к VDDD\_INT, вывод TP\_QO (27) подключить к GND.



Типовая схема усиления сигнала возбуждения

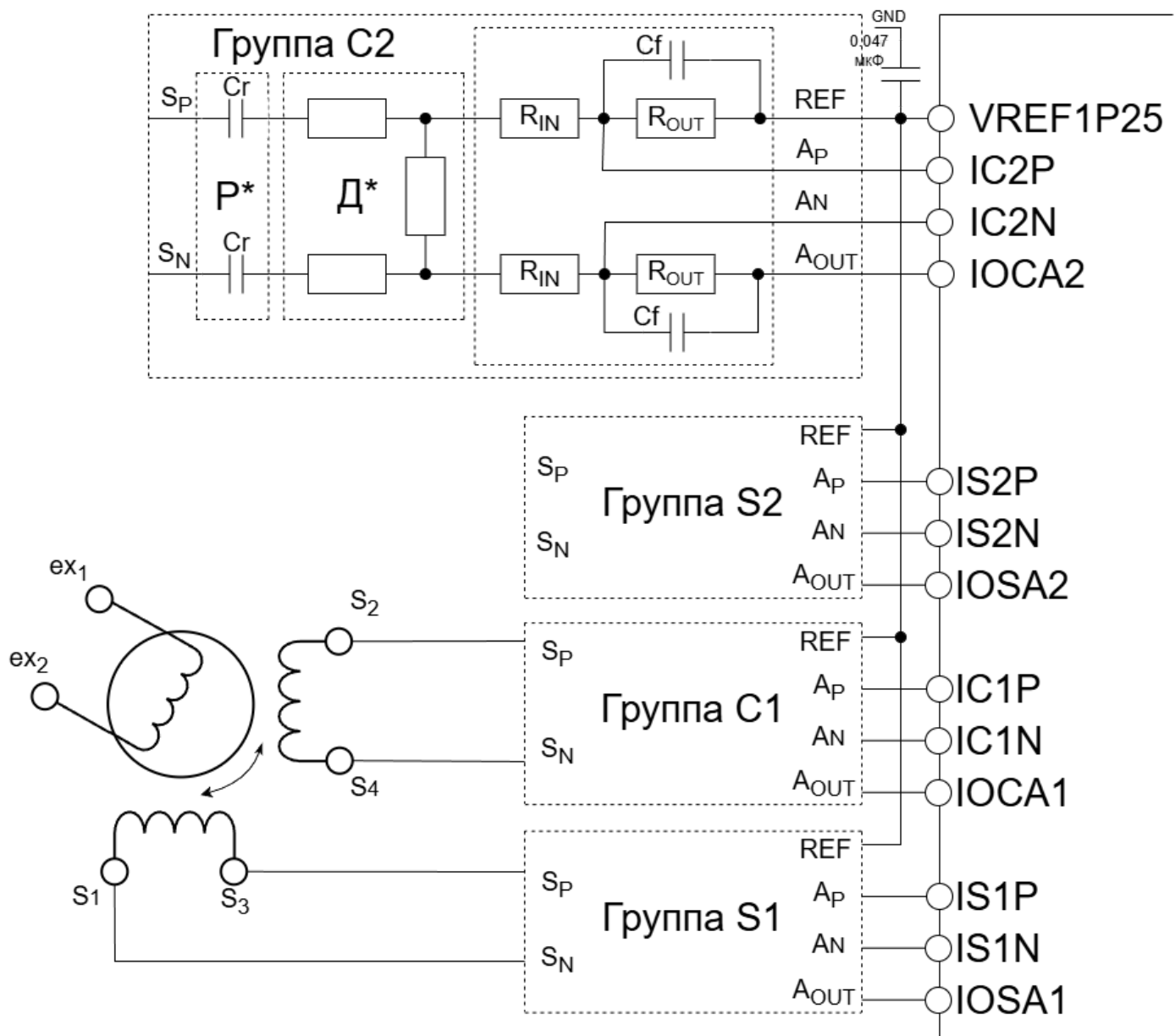


Схема подключения входов преобразователя для датчиков типа СКВТ с развязкой по постоянному напряжению и дополнительным делителем на входах

**Д** - Делитель напряжения для выравнивания амплитуды (опционально)

**Р** - Разделительные конденсаторы C1, C2 для развязки по постоянному току (опционально)

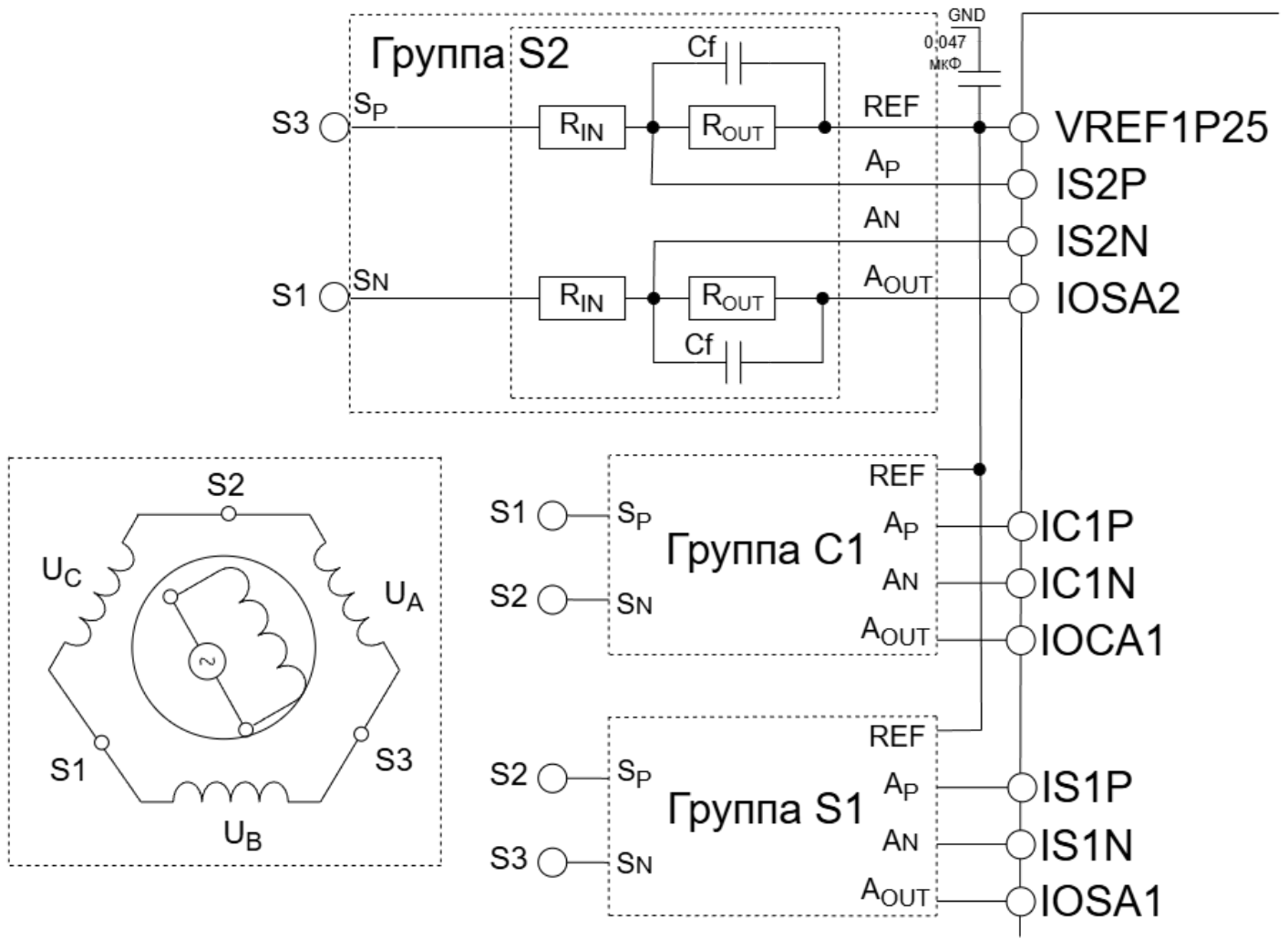
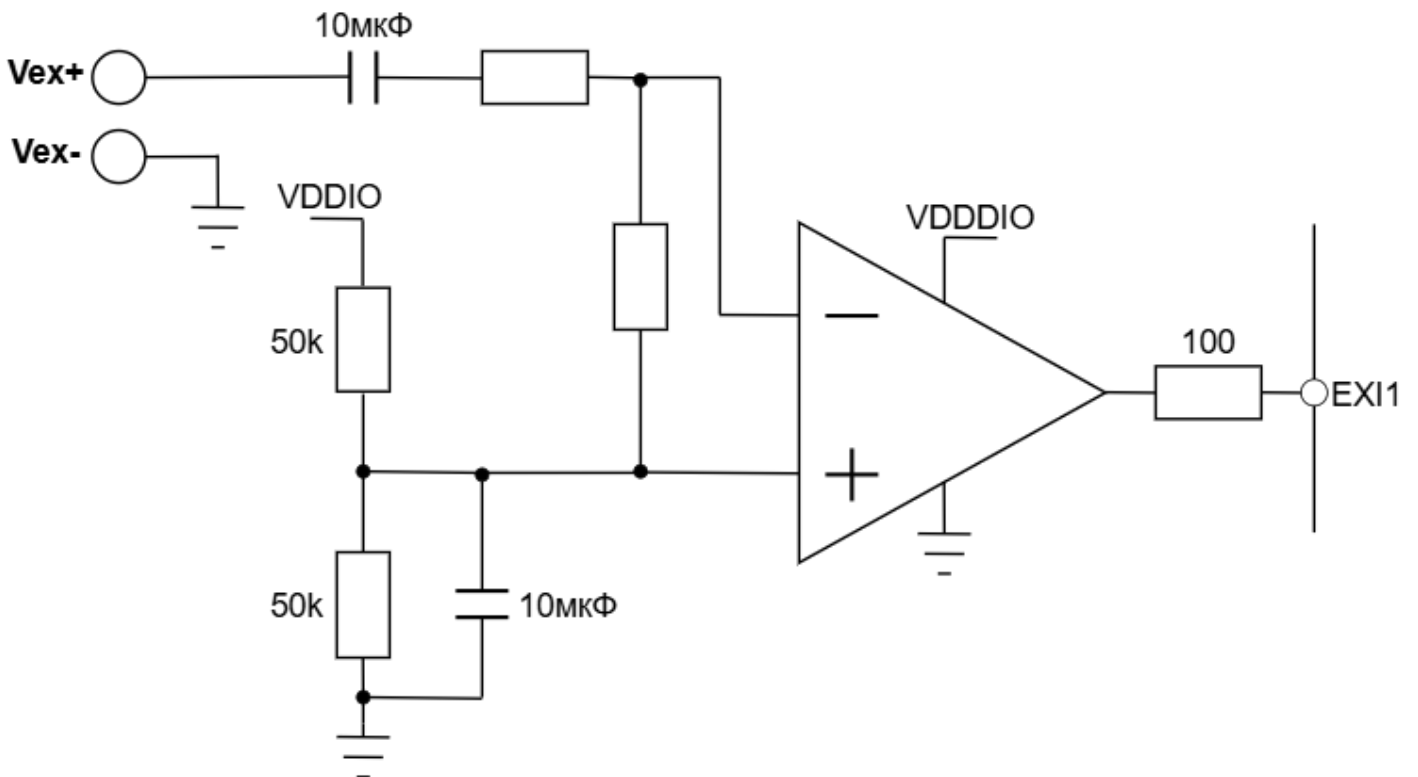


Схема подключения входов преобразователя для датчиков типа сельсин



Типовая схема подключения входа [EXI1](/docs/5400TP065A-022/ad3s-pins#EXI1) при Ex\_source=1

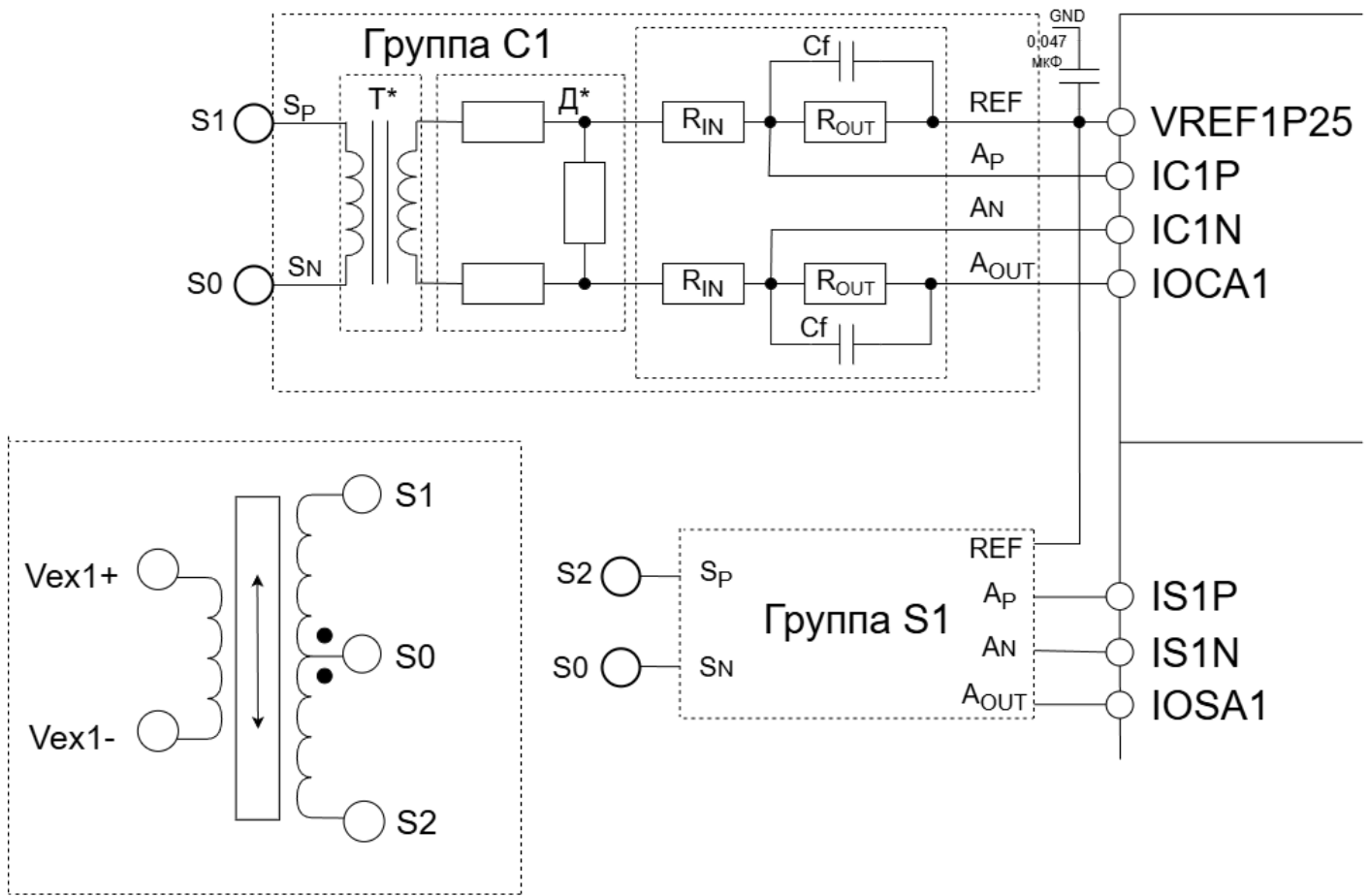


Схема подключения входов преобразователя для датчиков типа ЛРДТ, подключаемых по 5-ти проводной схеме, с дополнительным делителем и трансформаторной развязкой по постоянному напряжению при задании sensor\_mode=1

**Т** - Трансформаторная развязка по постоянному напряжению

**Д** - Делитель напряжения для выравнивания амплитуды (опционально)

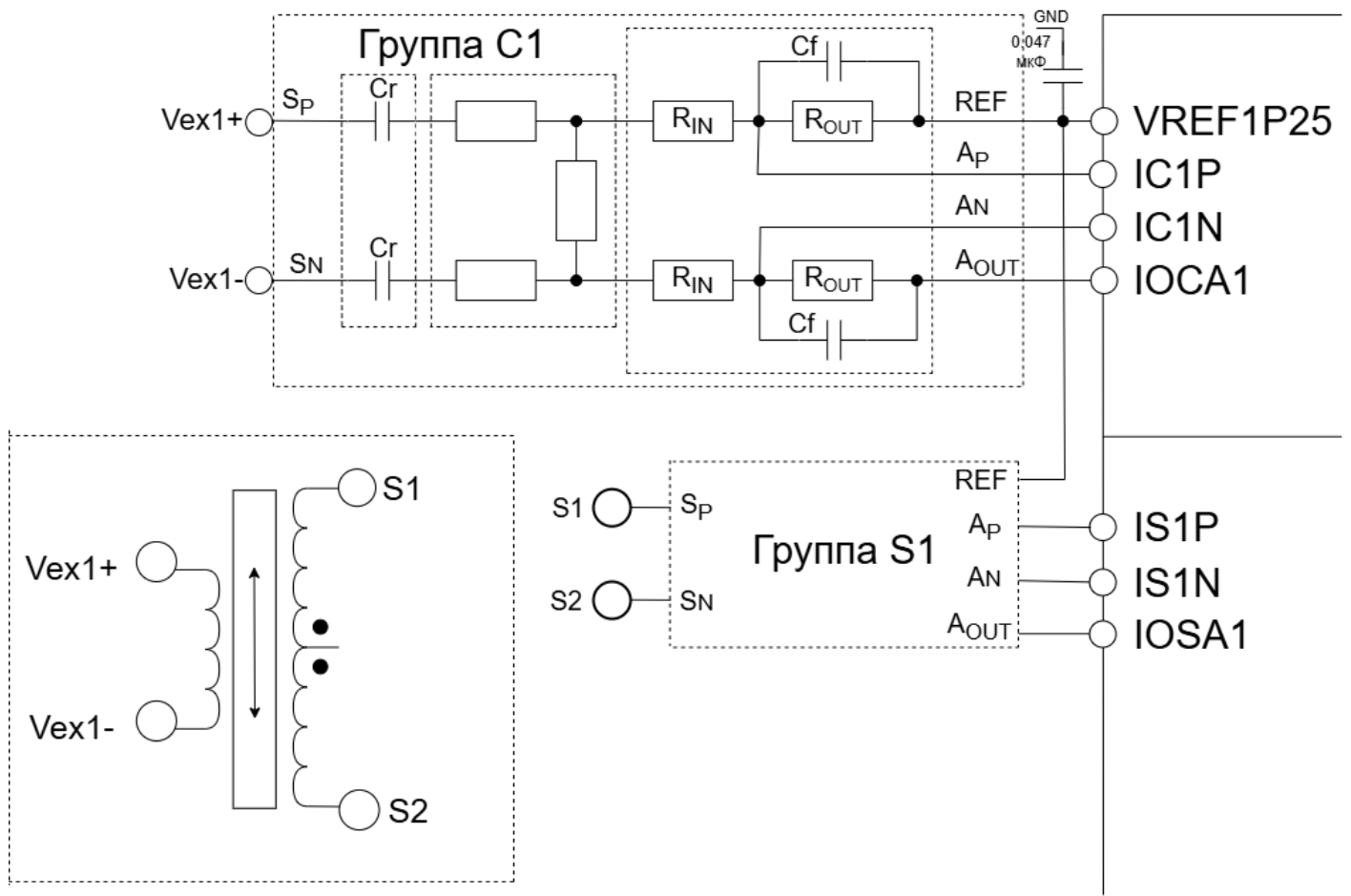
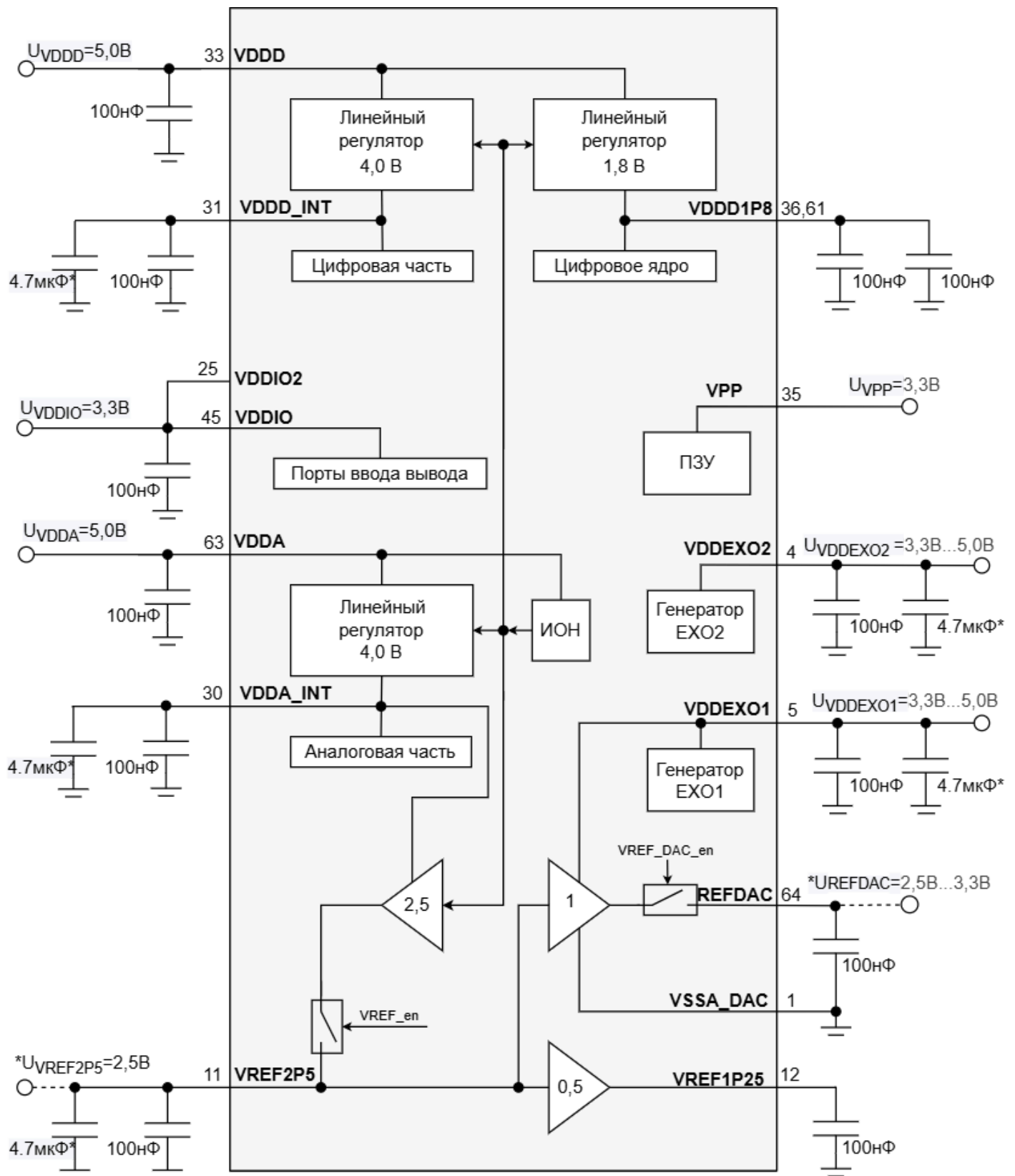


Схема подключения входов преобразователя для датчиков типа ЛРДТ с последовательно соединенными обмотками при задании `Sensor_mode=2`

# **Выбор схемы питания микросхемы**



Структура питания микросхемы

- Конденсаторы 4.7мкФ не обязательны, но рекомендуются - для выравнивания заряда и снижения пульсаций. Чем меньше размер корпуса конденсаторов 100нФ, тем лучше он справляется с пульсациями.

## Организация питания микросхемы:

- **Аналоговая и цифровая части** питаются от встроенных в микросхему линейных регуляторов, выходное напряжение которых настроено на  $4,0\text{ В} \pm 10\%$ .
- **Цифровое ядро** получает питание от отдельного встроенного линейного регулятора на  $1,8\text{ В}$ .
- **Порты ввода-вывода** запитываются через выводы **VDDIO** и **VDDIO2**, на которые пользователь должен подать внешнее напряжение  $3,3\text{ В} \pm 10\%$ .
- **Блок ПЗУ** запитываются через выводы **VPP** и **VDDIO2**, на которые пользователь должен подать внешнее напряжение  $3,3\text{ В} \pm 10\%$ .

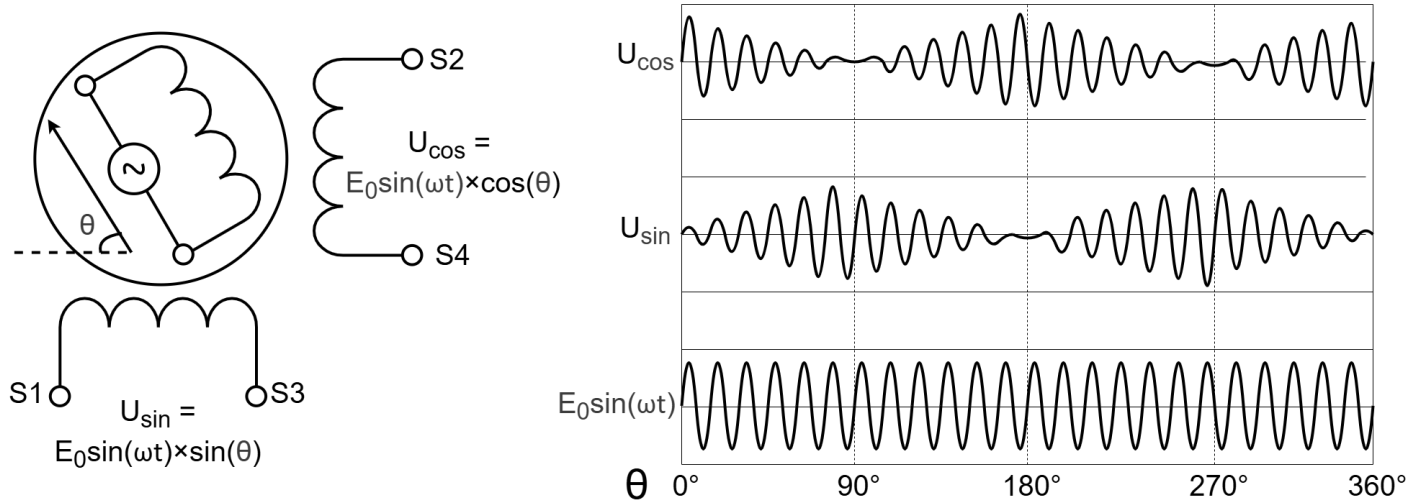
## Источники опорного напряжения (ИОН):

- **Опорное напряжение для АЦП (2,5 В)** микросхема может обеспечивать сама при установке бита **VREF\_en** регистра **AFE\_config**. Также допускается его подача на вывод **VREF2P5** с внешнего прецизионного регулятора (по умолчанию данный бит равен 0).
- **Опорное напряжение для ЦАП** микросхема может формировать самостоятельно при включении бита **DAC\_en** регистра **AFE\_config**. Кроме того, оно может быть подано от внешнего источника на вывод **REFDAC** (в этом случае бит **VREF\_DAC\_en** должен быть равен 0). Конкретное значение выбирается пользователем в диапазоне  $[2,5\text{ В}...3,3\text{ В}]$ .

Последнее обновление **17 июн. 2026 г.** от **Ilya Slepchenkov**

# Типы датчиков

## Датчики типа СКВТ (Resolver)



Датчик типа СКВТ

Для классического СКВТ датчика статорные обмотки механически расположены под углом  $90^\circ$ . Первичная обмотка возбуждается переменным опорным (эталонным) напряжением. Индуктивная связь наводит ЭДС на вторичные обмотки в соответствии с функцией положения ротора (вала) относительно статора. Таким образом, датчик вырабатывает два выходных напряжения  $U_{\cos}$ ,  $U_{\sin}$ , модулированных синусом и косинусом угла поворота вала. Выходные напряжения имеют схожие уравнения:

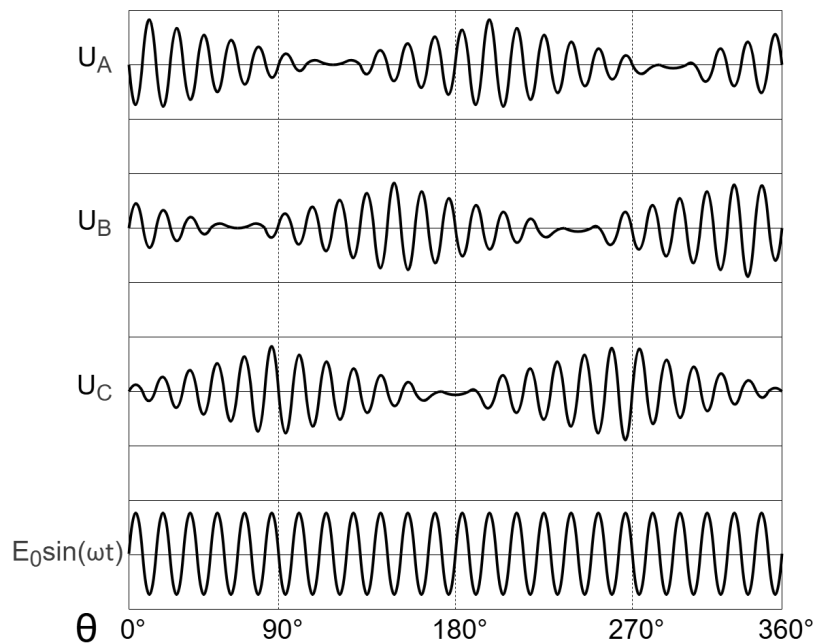
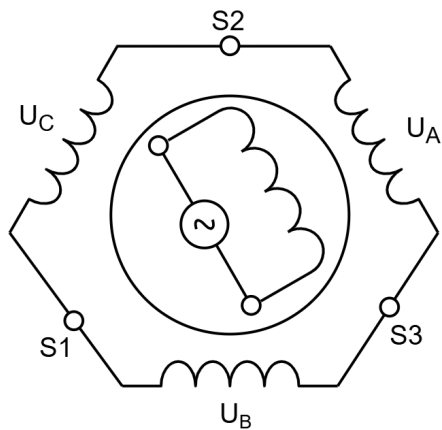
$$S3 - S1 = E_0 \sin \omega t \times \sin \theta = U_{\sin}$$

$$S2 - S4 = E_0 \sin \omega t \times \cos \theta = U_{\cos}$$

где:

- $\theta$  — угол ротора резольвера
- $\sin \omega t$  — частота сигнала возбуждения
- $E_0$  — амплитуда сигнала возбуждения

## Датчики типа сельсин (Synchro)



### Датчик типа Сельсин

Сельсин – вращающийся трансформатор, состоящий из одной первичной обмотки (ротор) и тремя вторичными обмотками (статор). Три обмотки статора расположены под углом  $120^\circ$  друг к другу.

При подаче переменного напряжения возбуждения  $U_{Bx}$  на первичную обмотку сельсина в машине создается электромагнитное поле, ориентированное перпендикулярно плоскости ротора. Данное поле вызывает индуцирование электродвижущей силы во вторичных обмотках сельсина. Амплитуда напряжения, возникающего во вторичных обмотках, определяется углом поворота ротора  $\theta$  относительно соответствующей вторичной обмотки и рассчитывается согласно выражениям:

$$U_A = k \cdot U_{Bx} \cdot \sin(\theta),$$

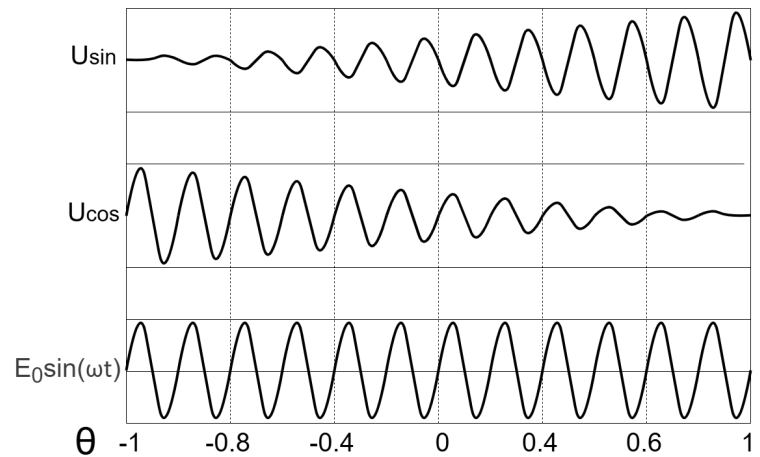
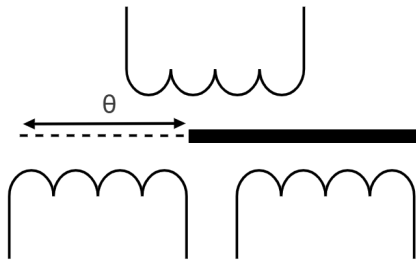
$$U_B = k \cdot U_{Bx} \cdot \sin(\theta + 120^\circ),$$

$$U_C = k \cdot U_{Bx} \cdot \sin(\theta - 120^\circ).$$

При подключении сельсина к микросхеме производится преобразование координат из трехфазной в двухфазную (преобразование alpha-beta-zero). Результирующие сигналы после преобразования соответствуют формату сигналов, поддерживаемых преобразователем в режиме СКВТ. Использование встроенного преобразователя координат позволяет подключить к микросхеме только один сельсин. В случае необходимости одновременного подключения двух сельсинов преобразование

координат должно быть реализовано схмотехническими средствами на печатной плате, а микросхема должна работать в режиме СКВТ.

## Датчики типа ЛРДТ (LVDT)



### Датчик типа ЛРДТ

Линейный регулируемый дифференциальный трансформатор – электрическая машина обычно с одной первичной обмоткой и двумя вторичными обмотками, расположенными на одной линии. ЛРДТ используются в качестве датчиков перемещений.

При подаче сигнала переменного напряжения  $U_{вх}$  на первичную обмотку ЛРДТ в машине создается электромагнитное поле, которое наводит ЭДС во вторичных обмотках ЛРДТ. Амплитуда напряжения во вторичных обмотках зависит от смещения сердечника  $\theta$  относительно вторичных обмоток и определяется по формулам

$$U_S = k \cdot U_{Bx} \cdot (1 + \theta),$$

$$U_C = k \cdot U_{Bx} \cdot (1 - \theta).$$

Часто используется последовательное соединение вторичных обмоток ЛРДТ. При этом происходит сложение сигналов UA и UB в противофазе. При таком включении сигналы на входе микросхемы имеют зависимость, которая задается по формулам

$$U_S = k \cdot U_{Bx} \cdot \theta,$$

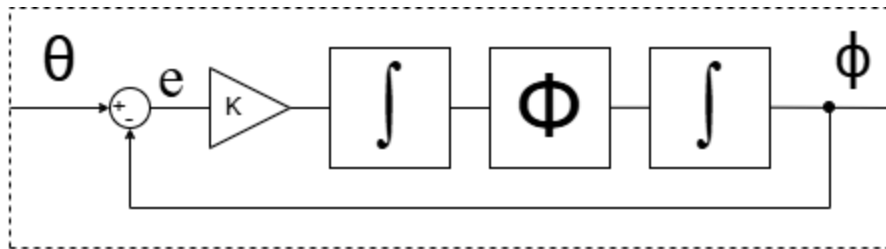
$$U_C = k \cdot U_{Bx} \cdot 1.$$



# Архитектура и принцип работы

## Принцип преобразования сигналов СКВТ в код угла

В основе преобразователя 5400TP065A-022 лежит система регулирования с замкнутой обратной отрицательной связью по углу поворота.

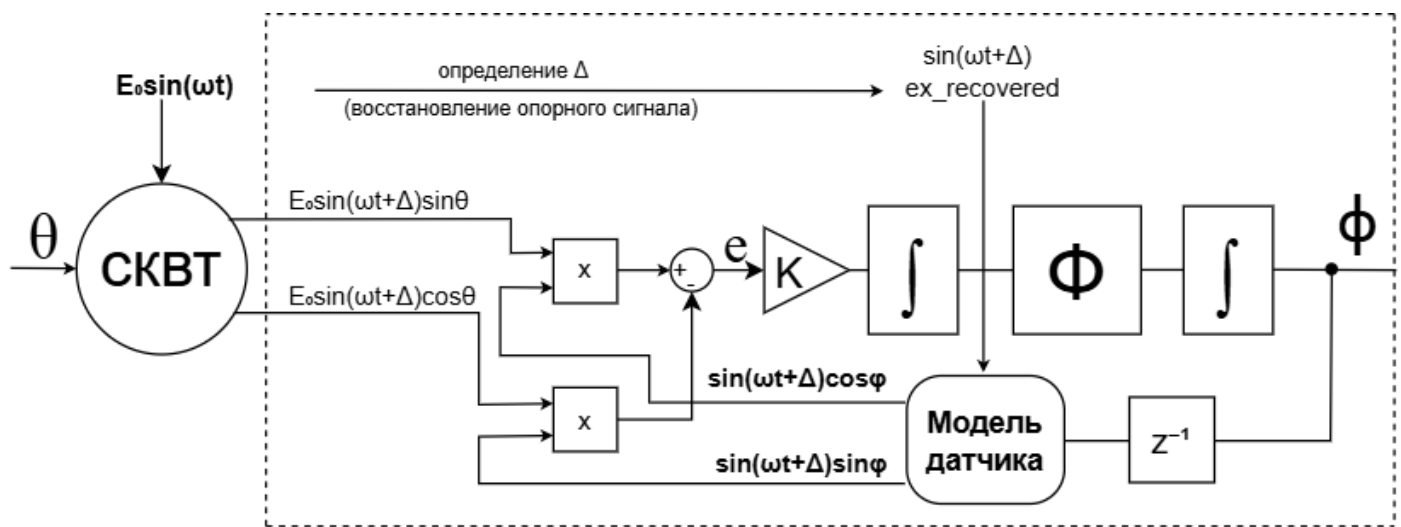


Контур 1

Следящий контур стремится свести разницу выходного и входного значения к нулю, делая свой выход равным входной координате. Контур содержит следующие элементы:

- вычислитель ошибки
- усилитель
- интегратор для демодуляции сигнала ошибки
- петлевой фильтр и интегратор для компенсации запаздывания и получения нулевой ошибки при равномерном движении датчика.

Для получения ошибки  $e$  контур в обратной связи содержит модель датчика, которая осуществляет преобразование вычисленной на предыдущем шаге координаты в виртуальные сигналы датчика  $\sin \phi$ ,  $\cos \phi$ .



Контур 2

Виртуальные сигналы умножаются на входные сигналы с датчика:

$$E_0 \sin(\omega t + \Delta) \sin \theta \times \sin(\omega t + \Delta) \cos \phi$$

$$E_0 \sin(\omega t + \Delta) \cos \theta \times \sin(\omega t + \Delta) \sin \phi$$

Вычисляется разность сигналов:

$$E_0 \sin^2(\omega t + \Delta) \times (\sin \theta \cos \phi - \cos \theta \sin \phi) \quad (1)$$

Высокочастотная составляющая  $\sin^2(\omega t + \Delta) = \frac{1}{2} - \frac{1}{2} \cos(2\omega t + 2\Delta)$  подавляется на первом интегрирующем звене контура, выполняющим роль ФНЧ.

Оставшееся выражение можно преобразовать с использованием тригонометрического тождества:

$$E_0 (\sin \theta \cos \phi - \cos \theta \sin \phi) = E_0 \sin(\theta - \phi) \quad (2)$$

При корректной работе контура разность  $(\theta - \phi)$  становится малым числом, для которого верно следующее выражение:

$$E_0 \sin(\theta - \phi) \approx E_0 (\theta - \phi) \quad (3)$$

Полученное после первого интегрирования значение  $E_0 (\theta - \phi)$  является разницей между истинным положением ротора и вычисленной следящим контуром координатой, и это значение получено без проведения вычислений, обратных к функции СКВТ. Модель функции СКВТ в обратной связи может быть усложнена параметрами и естественными

искажениями порождаемых сигналов в целях увеличения соответствия виртуальных к реальным сигналам с датчика.

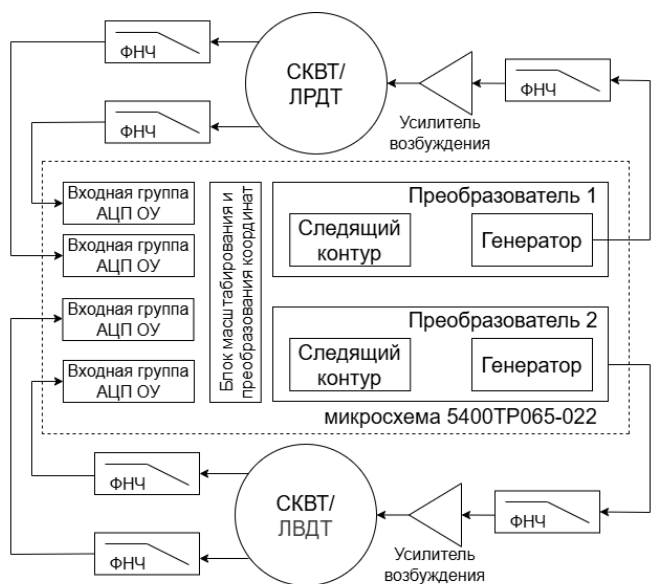
Взятие разницы двух последовательных по времени координат эквивалентно дифференцированию угла, а результат является угловой скоростью, из которой в контуре восстанавливается координата с помощью второго интегратора.

$$\omega[n] = \theta[n] - \phi[n - 1]$$

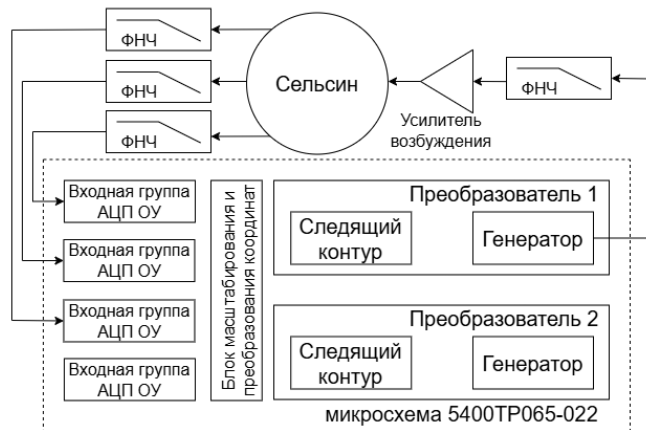
$$\phi[n] = \int \omega[n]$$
(4)

## Архитектура микросхемы

Микросхема содержит два независимых блока преобразователя координата/код (далее “преобразователь 1” и “преобразователь 2”), которые могут использоваться как независимо, так и совместно для повышения точности или надежности преобразования. В обозначении регистров, принадлежащих каждому преобразователю, используется префикс “С1” и “С2” соответственно. Каждый блок преобразователя реализует следящий принцип измерения. Для подачи возбуждающего напряжения на датчик микросхема содержит два генератора сигналов возбуждения датчиков. Сигналы с вторичных обмоток датчика преобразуются в цифровую форму и подаются на вход преобразователя, который осуществляет нахождение координаты, используя виртуальную модель датчика и контур с обратной связью.



А



Б

Варианты использования микросхемы: А - два независимых СКВТ датчика, Б - подключение датчика типа Сельсин

Наличие 4 входных групп и 2 преобразователей позволяет использовать микросхему в различных включениях:

- один или два датчика СКВТ (с синхронным или независимым возбуждением)
- СКВТ с 4-мя выходными обмотками и одной обмоткой возбуждения
- Сельсин (3 выходных обмотки)
- один или два датчика ЛРДТ (с синхронным или независимым возбуждением)

Микросхема позволяет использовать внешний независимый генератор возбуждения.

Микросхема имеет возможность обрабатывать демодулированные сигналы.

*Последнее обновление **22 мая 2026 г.** от **islepchenkov***

# Программируемый генератор

## Программируемый генератор

Микросхема содержит два независимых программируемых генератора, формирующих опорный сигнал на выводах [EXO1/EXO2](#). Питание выходного буфера генератора подводится непосредственно с выводов [VDDEXO1/VDDEXO2](#), [GNDEXO1/GNDEXO2](#). Напряжение питания генератора не зависит от напряжения [VDDD](#), [VDDA](#) и выбирается исходя из конкретной схемы усилителя возбуждения в диапазоне от 3,3 В до 5,0 В.

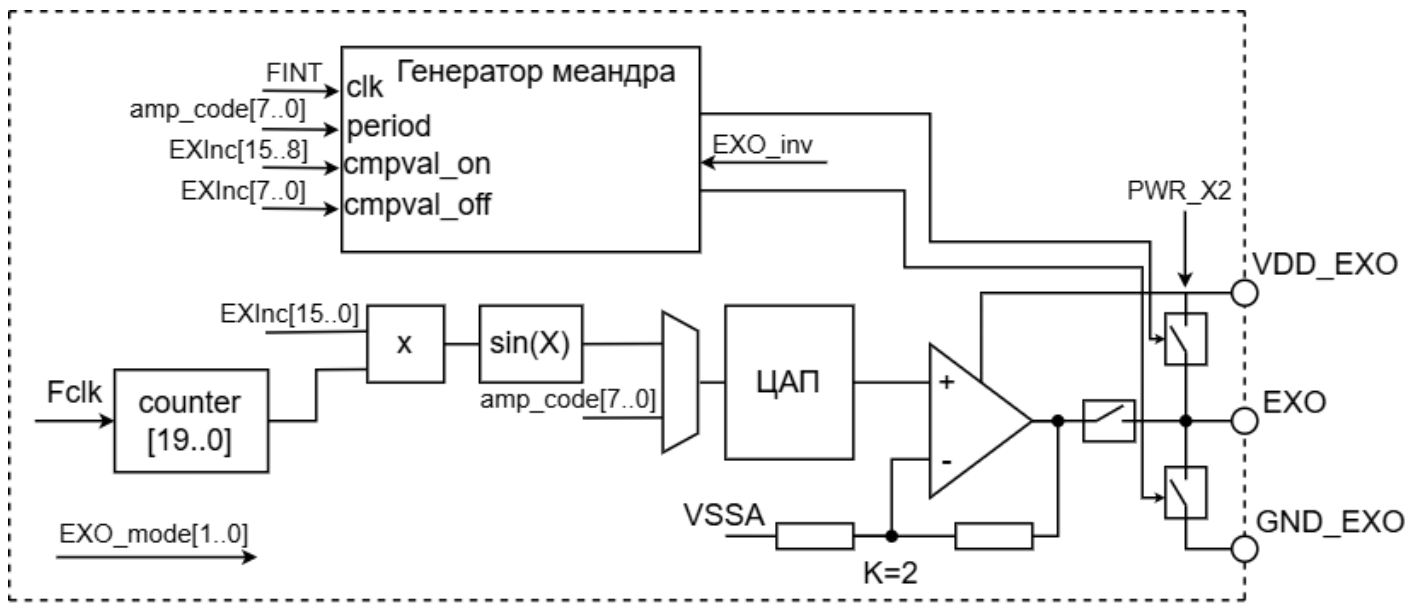
С помощью настройки [EXO\\_mode](#) выбирается режим работы генератора:

- [EXO\\_mode = 00](#) — генератор выключен
- [EXO\\_mode = 01](#) — вывод импульсного сигнала (меандр)
- [EXO\\_mode = 10](#) — вывод синусоидального сигнала
- [EXO\\_mode = 11](#) — вывод постоянного уровня

Для включения одного из режимов необходимо выбрать соответствующий режим и выставить биты [EXO1\\_en/EXO2\\_en](#) регистра [Mode\\_config](#) в единицу.

## Генератор синусоидальных сигналов

На рисунке ниже представлена структурная схема генератора сигналов.



Структурная схема генератора синусоидальных сигналов

В цифровом формирователе кода на каждом такте  $F_{clk}$  инкрементируется 20-ти разрядный счетчик, значение которого перемножается на 16-битный коэффициент  $C1EXInc, C2EXInc$ . Далее для [19:12] разрядов произведения ставится в соответствие знаковое 8-ми разрядное значение единичного синуса, затем перемножается на беззнаковый коэффициент  $(176 + AMP\_CODE) / 1024$  и для обеспечения линейности усиления на выходном буфере добавляется значение  $63 - AMP\_CODE / 4$ . Полученное 8-ми разрядное значение переводится в напряжение с опорным сигналом 2,5 В и усиливается выходным буфером с коэффициентом 2 относительно 0 В.

По формуле (5) определяется частота сигналов  $f_{ex}$ , по формуле (6) задается амплитуда сигналов  $V_{exo}$ :

$$f_{ex} = \frac{EXInc}{1048576} f_{clk} \quad (5)$$

$$V_{EXO} = 2 \cdot V_{REF2P5} \cdot \left( \frac{\sin\left(\frac{2\pi}{f_{ex}}\right) \cdot \left(\frac{176+AMP\_CODE}{4}\right) + 63 + \frac{AMP\_CODE}{4}}{256} \right) \quad (6)$$

Частоты  $f_{ex1}, f_{ex2}$  выбираются исходя из спецификации датчика. Пользователь осуществляет восстановление (фильтрацию) и усиление сигналов генераторов перед подачей на датчики.

Для исключения нелинейности усиления коэффициент  $AMP\_Code$  должен быть выбран таким образом, чтобы значение  $V_{amp}$  было ниже напряжения питания выходного буфера

VDDEXO1/VDDEXO2 как минимум на 0,4 В.

Опорное напряжение для ЦАП может быть в диапазоне [2.5В ... 3.3В], и должно быть подано на вывод REFDAC. Имеется возможность использовать внутренний ИОН, для чего требуется установить бит VREF\_DAC\_en в единицу. И в том и другом случае требуется шунтирование вывода REFDAC конденсатором 0,1 мкФ.

## Генерация импульсного сигнала

Генерация импульсного сигнала выполняется 8-битным счетчиком тактируемым системной частотой FINT. Частота импульсов определяется значением Amp\_code[7:0] в соответствии с формулой (7):

$$F_{imp} = \frac{FINT}{Amp\_code} \quad (7)$$

Скважность меандра определяется 8-битными значениями EXInc[15:8], EXInc[7:0] регистров C1EXInc/C2EXInc. Когда значение 8-битного счетчика равно EXInc[15:8], на выводе EXO1/EXO2 устанавливается уровень VDDEXO1/VDDEXO2, когда значение счетчика равно EXInc[7:0], выход подключается к GNDEXO1/GNDEXO2. Если задать бит EXO\_inv регистра C1ExoStngs/C2ExoStngs, то выход инвертируется. При выставлении бита PWR\_X2 регистра C1ExoStngs/C2ExoStngs ток коммутации вывода EXO1/EXO2 на VDDEXO1/VDDEXO2 и GNDEXO1/GNDEXO2 будет увеличен в 2 раза. Для синхронизации двух 8-битных счетчиков в импульсном режиме требуется установить бит EXO\_sync в регистре AFE\_config в единицу.

## Вывод постоянного уровня

В режиме постоянного уровня, на выходе EXO1/EXO2 будет формироваться напряжение в соответствии с формулой (8):

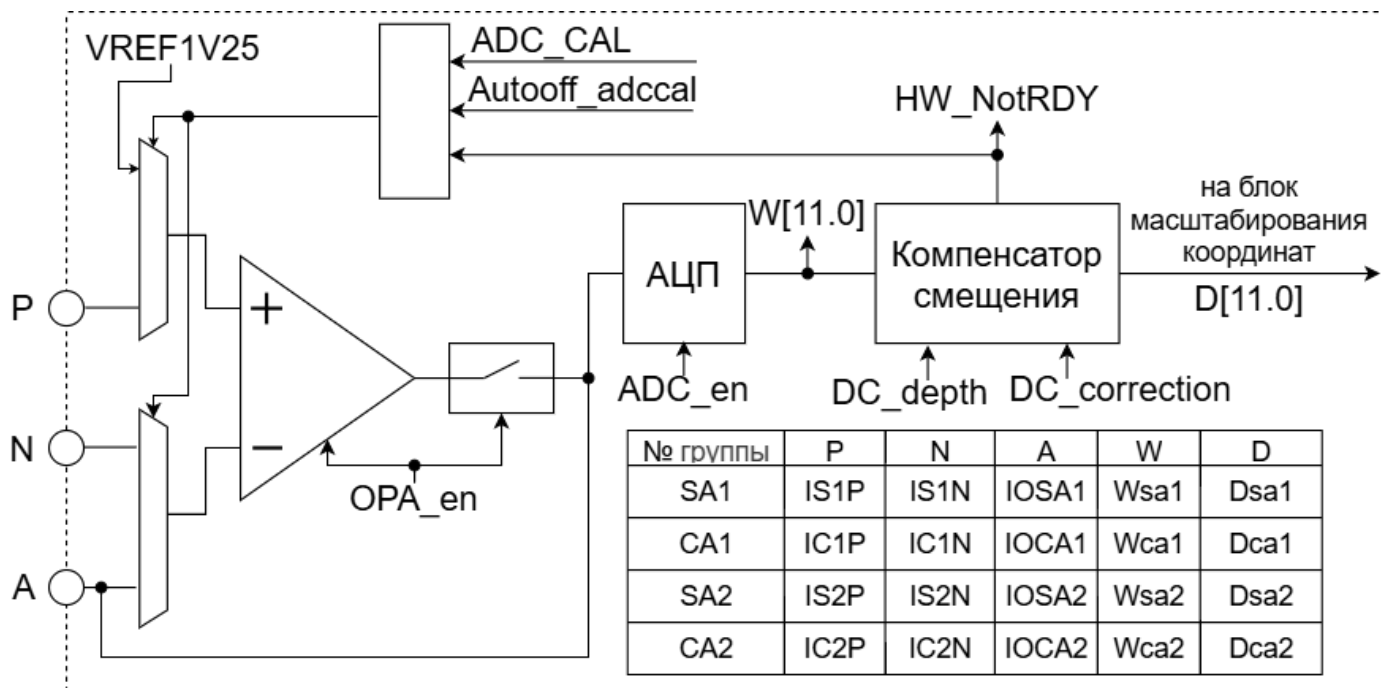
$$V_{EXO} = 2 \cdot V_{REF2P5} \cdot \left( \frac{AMP\_CODE}{255} \right) \quad (8)$$

Пользователь должен убедиться, что напряжение питания VDDEXO1/VDDEXO2 превышает расчетное значение Vexo.

Последнее обновление **22 мая 2026 г.** от *islerchenkov*

# Входные группы преобразователя

В микросхеме реализовано 4 идентичные группы (SA1, CA1, SA2, CA2) преобразования аналогового сигнала в цифровой код.



Входная группа преобразователя

## Включение

По умолчанию буферные ОУ выключены, в этом случае подача сигналов на АЦП осуществляется через выходы [IOSA1](#), [IOCA1](#), [IOSA2](#), [IOCA2](#). Для использования встроенных ОУ требуется установить бит [OPA\\_en](#). Для работы АЦП требуется установить бит [ADC\\_en](#) в единицу.

## Компенсатор смещения

В преобразователе реализовано 4 компенсатора смещений среднего напряжения. Компенсатор по заданному объему данных вычисляет постоянную составляющую смещения, затем вычитает её из последующих выборок АЦП. Объем выборки определения смещения задается битами [DC\\_depth](#). Чем больше объем выборки и время

калибровки, тем более точными будут определены коэффициенты смещения. Компенсатор настоятельно рекомендуется использовать для модулированных сигналов, т.к. смещения вызывают шум и погрешности в работе блока восстановления частоты и в определении выходной координаты следящим контуром.

Для преобразования сигналов без модуляции необходимо установить бит [DC\\_carrier](#) в единицу. В этом случае рекомендуется включить режим автокалибровки АЦП (установить бит [ADC\\_CAL](#) и [Wait\\_1offset](#)), который в своей работе используется компенсатор смещений. Автокалибровка заключается в определении и компенсации смещения нулевого сигнала входных трактов преобразования (ОУ и АЦП) при включении питания. В режиме калибровки входы ОУ отключаются от выводов микросхемы, и на вход ОУ подается напряжение [VREF1p25](#). Коды, получаемые с АЦП, подаются на компенсатор смещения, пока бит [ADC\\_CAL](#) установлен в 1. Завершение определения коэффициентов смещения возможно узнать с помощью чтения флага [HW\\_NotRDY](#). Для автоматического выключения калибровки требуется установить бит [Autooff\\_adccal](#). В режимах с модулированным сигналом датчика калибровка не требуется (т.к. компенсатор смещения будет работать постоянно). Для удержания в сбросе блока восстановления частоты и следящего контура до расчета компенсатором смещения требуется установить бит [Wait\\_1offset](#).

*Последнее обновление 22 мая 2026 г. от islepchenkov*

# Блок масштабирования и преобразования координат

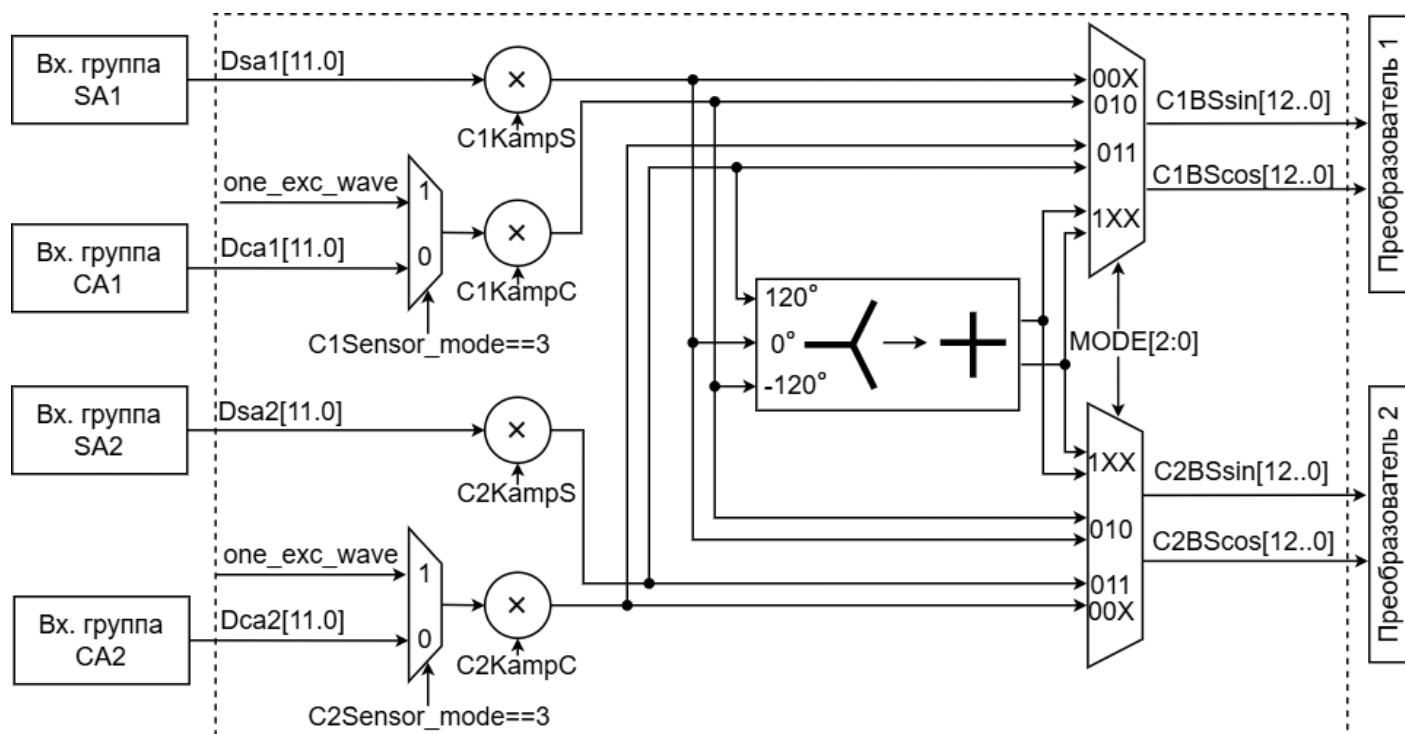


Схема блока масштабирования и преобразования координат

Блок масштабирования позволяет изменять масштаб сигналов после аналого-цифрового преобразования. Масштабирование предназначено для коррекции ошибки усиления ALIT и аналоговых цепей на плате, а также погрешностей датчика. Масштаб сигналов задается в регистрах **C1KampS** (вход *IOSA1*), **C1KampC** (вход *IOCA1*), **C2KampS** (вход *IOSA2*), **C2KampC** (вход *IOCA2*). Значения по умолчанию в этих регистрах предполагают сигнал на входе микросхемы амплитудой 2,5 В (пик-пик). Блок производит преобразование координат из 3-х фазной системы координат в декартову систему координат для режима Сельсин. Выбор режима Сельсин производится установкой **Mode[2:0]=4** (регистр **AFE\_config**). В данном режиме используются входы, приведенные в таблице 5.

Таблица 5 – Использование входов микросхемы в зависимости от режима преобразования

| Вход         | Режим Сельсин                                   | Режим СКВТ                       |
|--------------|---|----------------------------------|
| <i>IOSA1</i> | Вход преобразователя $\sin(\Theta + 0^\circ)$   | Вход 1-го преобразователя $\sin$ |
| <i>IOCA1</i> | Вход преобразователя $\sin(\Theta + 120^\circ)$ | Вход 1-го преобразователя $\cos$ |
| <i>IOSA2</i> | Вход преобразователя $\sin(\Theta - 120^\circ)$ | Вход 2-го преобразователя $\sin$ |
| <i>IOCA2</i> | Не используется                                 | Вход 2-го преобразователя $\cos$ |

Значения битов [Mode\[2:0\]](#) регистра [AFE\\_config](#) определяют режимы совместной работы двух преобразователей (таблица ниже).

**Таблица 6 – Режимы совместной работы двух преобразователей**

| Регистр <a href="#">Mode[2:0]</a> | Режим   |
|-----------------------------------|---|
| 000                               | Оба преобразователя включены и работают независимо. Первый преобразователь подключен к выводам <a href="#">IOSA1</a> , <a href="#">IOCA1</a> , <a href="#">EXI1</a> . Второй преобразователь подключен к выводам <a href="#">IOSA2</a> , <a href="#">IOCA2</a> , <a href="#">EXI2</a> . |
| 010                               | Оба преобразователя включены и работают параллельно. Преобразователь 2 использует входы преобразователя 1 ( <a href="#">IOSA1</a> , <a href="#">IOCA1</a> , <a href="#">EXI1</a> ).   |

| Регистр <b>Mode[2:0]</b>  | Режим   |
|---|---|
| 011   | Оба преобразователя включены и работают параллельно. Преобразователь 1 использует входы преобразователя 2 ( <b>IOSA2</b> , <b>IOCA2</b> , <b>EXI2</b> ).  |
| 100   | Режим Сельсин. Оба преобразователя включены и работают параллельно. На входы преобразователей подается сигнал со входов <b>IOSA1</b> , <b>IOCA1</b> , <b>IOSA2</b> , <b>EXI1</b> . Включено преобразование координат из 3-х фазной в декартову. |
| Примечание. В режиме подачи внешнего опорного сигнала <b>Ex_source==01</b> преобразователь 2 использует опорный сигнал со вывода <b>EXI1</b> .  |   |
| При использовании двух преобразователей параллельно пользователь может задать в регистре <b>CMP_lth</b> максимально допустимое различие между каналами. Если результаты двух преобразователей отличаются на величину большую, чем записана в регистр <b>CMP_lth</b> , выставляется бит <b>Not_Equal</b> в регистре <b>Stat_main</b> . В зависимости от настройки в регистре <b>Mask_Stat</b> это также может установить в состояние логического нуля выход <b>Ready</b> микросхемы. |   |

Для экономии энергии любой из преобразователей может быть отключен установкой битов **CONV1\_en**, **CONV2\_en** в регистре **Mode\_config** в состояние логического нуля. При этом для корректной работы выхода **Ready** необходимо бит **MSK\_Not\_Equal** также установить в состояние логического нуля.



# Блок восстановления сигнала опорной частоты

Точность определения координаты в следящем контуре базируется на соответствии виртуальных сигналов входным сигналам датчика. Индуктивная природа кодирования угла поворота в СКВТ подразумевает использование переменных сигналов. В типовой схеме фаза сигнала сформированная генератором проходит ряд преобразований - фильтр в усилителе, механический сдвиг фазы в СКВТ, фильтры и усилители входных каскадов, АЦП и задержки в цифровых вычислениях, что приводит к значительному изменению фазы принимаемых сигналов *IOSA1/IOSA2* и *IOCA1/IOCA2* по сравнению с *EXO1/EXO2*. Изменение фазы сигналов имеет температурную зависимость и при изменении температуры вносит значительную погрешность в результат преобразования (при отключенном блоке восстановления сигнала опорной частоты).

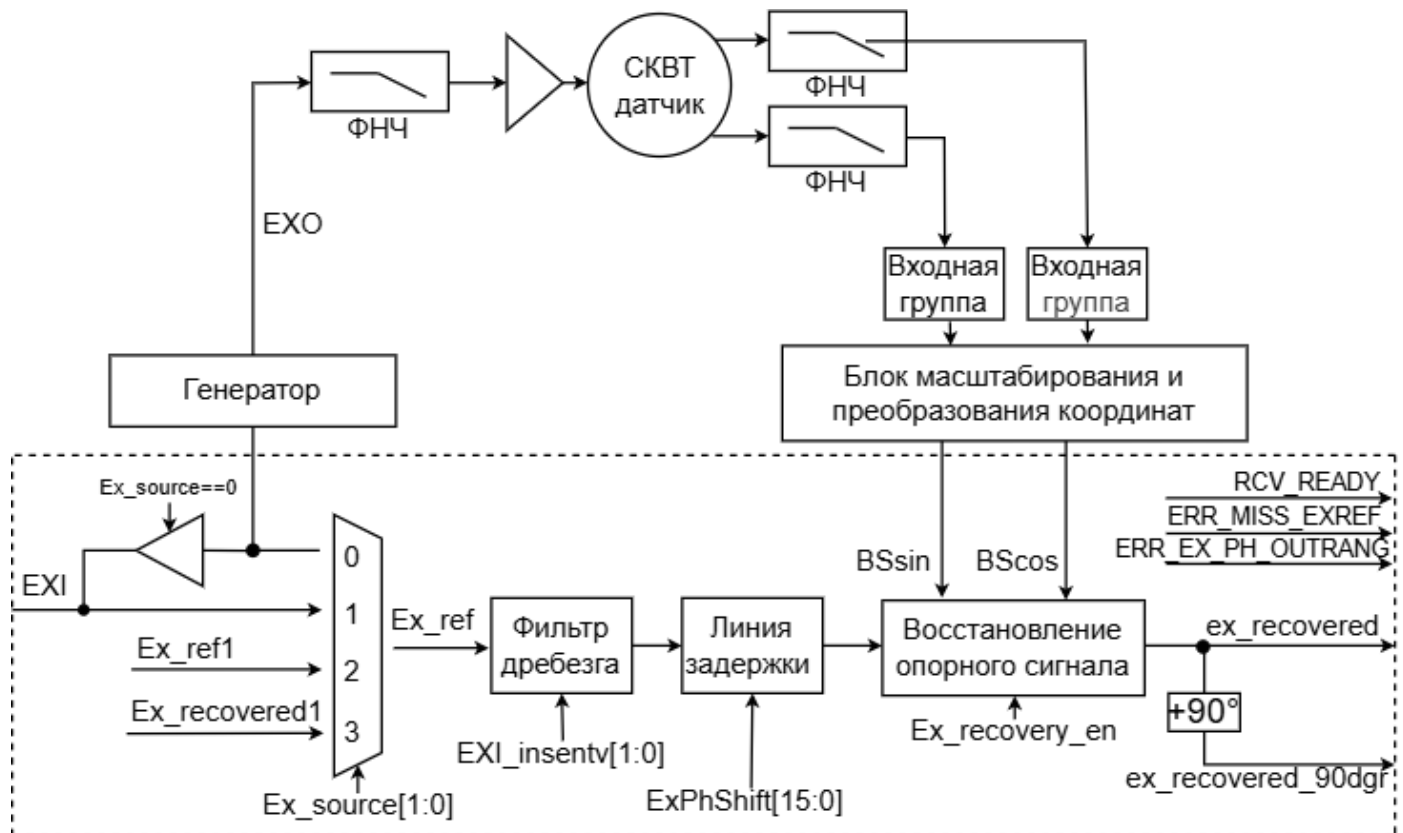


Схема блока восстановления опорного сигнала в следящем контуре

Чтобы избавиться от погрешности, вызванной этой задержкой, а также подавить квадратурные составляющие сигналов, возникающие при движении вала датчика, вместо

сигнала возбуждения микросхема использует в модели датчика восстановленный сигнал опорной частоты **ex\_recovered**. Сигнал **ex\_recovered** вычисляется из принимаемых сигналов **IOSA1/IOSA2**, **IOCA1/IOCA2** и вспомогательного сигнала **Ex\_ref**, который необходим для определения квадранта, в котором находится СКВТ. Суть восстановления частоты сводится к автоматическому сдвигу **Ex\_ref** к моментам перехода через нуль большего из сигналов **IOSA1/IOSA2**, **IOCA1/IOCA2**. Поэтому для корректной работы блока сдвиг фазы между сигналом **Ex\_ref** и сигналами **IOSA1/IOSA2**, **IOCA1/IOCA2** должен находиться в пределах одного квадранта ( $\pm 45^\circ$ ). При сильной зависимости сдвига фазы сигналов от температуры, рекомендуется устанавливать в пределах  $\pm 20^\circ$ . Для проведения настройки в таблице 7 указано соответствие фазы сигнала **ex\_ref** к входным с СКВТ сигналам, чтобы выходная координата следящего контура совпадала с положением ротора СКВТ.

**Таблица 7 – Полярность сигналов СКВТ в зависимости от угла поворота вала**

| Квадрант, в котором находится СКВТ/сельсин | Знак сигнала IOSA1/IOSA2  | Знак сигнала IOCA1/IOCA2  |
|--|---------------------------|---------------------------|
| I (0 – 90°)                                | Совпадает с <b>Ex_ref</b> | Совпадает с <b>Ex_ref</b> |
| II (90° – 180°)                            | Совпадает с <b>Ex_ref</b> | Инверсный к <b>Ex_ref</b> |
| III (180° – 270°)                          | Инверсный к <b>Ex_ref</b> | Инверсный к <b>Ex_ref</b> |
| IV (270° – 360°)                           | Инверсный к <b>Ex_ref</b> | Совпадает с <b>Ex_ref</b> |

Контроль разницы фаз между **ex\_ref** и сигналами **IOSA1/IOSA2**, **IOCA1/IOCA2** лучше осуществлять программным способом с помощью буферизированного считывания ячейки **C1AdcS/C2AdcS**, в которой содержится старшие разряды входных и опорных сигналов поступающих непосредственно на преобразователь.

Для точной корректировки сдвига фазы **Ex\_ref** должен использоваться регистр **C1ExPhShft/C2ExPhShft**, который в тактах **f\_clk** задает задержку **Ex\_ref**. Выбор источника **Ex\_ref** осуществляется с помощью битов **Ex\_source** в регистре **C1InputStngs/C2InputStngs** (таблица 8):

**Таблица 8 – Источники опорного сигнала **Ex\_ref****

| № преобразователя | Ex_Source | Источник ex_ref                  |
|-------------------|-----------|----------------------------------|
| 1                 | 0         | с генератора опоры 1             |
| 1                 | 1         | со входа EXI1                    |
| 1                 | 2         | с генератора опоры 1             |
| 1                 | 3         | со входа EXI1                    |
| 2                 | 0         | с генератора опоры 2             |
| 2                 | 1         | со входа EXI1                    |
| 2                 | 2         | ex_ref с преобразователя 1       |
| 2                 | 3         | ex_recovered с преобразователя 1 |

Примечание. Преобразователь 2 не может принять внешний опорный сигнал с вывода EXI2, при выборе режима Ex\_source=01 для преобразователя 2, сигнал опоры будет браться с вывода EXI1. При использовании внутренних источников Ex\_ref (Ex\_source!=1) вывод EXI1/EXI2 становится выходом сигнала Ex\_ref. Этот выход также может использоваться для работы нескольких микросхем от одного источника опорной частоты.

Для введения нечувствительности к ложным импульсам ex\_ref задать в регистр EXI\_insentv[1:0] значение отличное от нуля.

Если опорный сигнал ex\_ref расходится с входным сигналом более чем 40°, то сформируется флаг неисправности EX\_PH\_OUTRANG. При отсутствии сигнала ex\_ref сформируется флаг ошибки MISS\_EXREF.

При наличии большого смещения уровня входных сигналов рекомендуется отложить включение блока восстановления опорной частоты до момента, когда будут рассчитаны первые коэффициенты смещений. Для этого требуется записать в бит Wait\_1offset значение "1". В этом случае блок восстановления частоты начнет расчет с первого полного периода входного сигнала с момента наступления флага HW\_NotRDY. При включении преобразователя и до расчета восстановленного сигнала флаг готовности

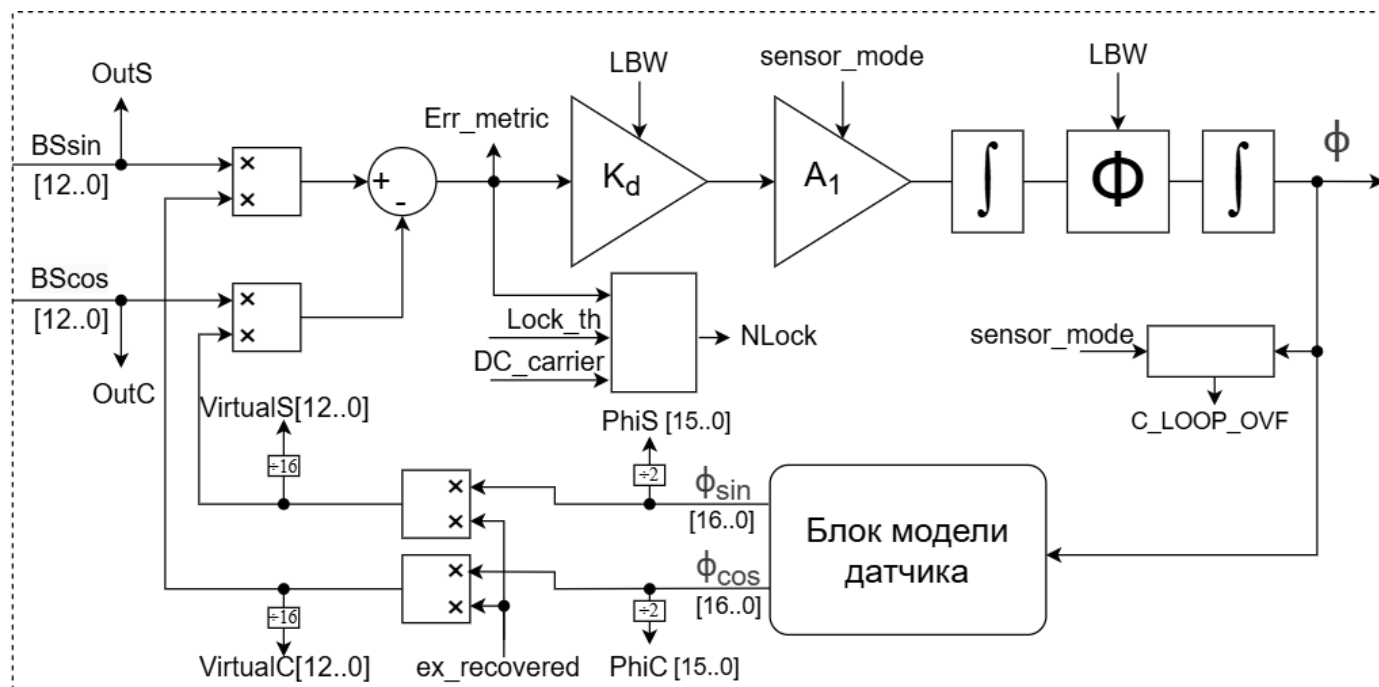
**RCV\_notRDY** удерживается в единице. При включенном блоке восстановления частоты и отсутствии его готовности следящий контур удерживается в сбросе.

При установке бита **DC\_carrier** сигнал **ex\_recovered** всегда равен **1**. Если частота вращения датчика больше  $\frac{1}{8}$  от частоты опорного сигнала, схема восстановления должна быть отключена (**Ex\_recovery\_en=0**), и использован внешний или внутренний сигнал опорной частоты. При этом сдвиг фазы этого сигнала должен быть точно подогнан к фазе **IOSA1/IOSA2** и **IOCA1/IOCA2**. Остаточный сдвиг фазы влияет на шум преобразования, а также погрешность при движении датчика. Данный режим работы не рекомендуется использовать, т.к. возникает зависимость результата преобразования от температуры.

**Внимание!** При неправильной настройке блока восстановления опорного сигнала преобразователь может давать ошибку  $180^\circ$ , при этом флаги ошибок будут сброшены.

*Последнее обновление 22 мая 2026 г. от islepchenkov*

# Следящий контур



Следящий контур

Для анализа качества внесенных настроек и точности работы следящего контура выведено множество контрольных точек, доступных для чтения по интерфейсу SPI:

- Регистры **OutC/OutS** - вход следящего контура, выход блока масштабирования координат. В этой точке размах изменения амплитуды должен быть равен двум единичным амплитудам, т.е. значению 4096 (настраивается коэффициентами **KampC/KampS**). Для анализа амплитуды модулированных сигналов рекомендуется использовать микропрограмму детектора амплитуды на CPU и натуральное вращение датчика. В контур поступают 13 битные значения, поэтому наличие смещений не переполняет разрядную сетку.
- Регистры **PhiC/PhiS** - выход модели датчика. В этой точке смещение среднего уровня должно быть равно смещению **OutC/OutS** (настраивается коэффициентами).
- Регистры **VirtualC/VirtualS** - выход модели датчика, модулированные сигналом восстановленной частоты. В этой точке сигналы должны быть очень близки к входным сигналам по фазе и амплитуде. Для анализа использовать буферизированную запись пар первичных сигналов `[OutS, VirtualS]` или `[OutC, VirtualC]`.

- Регистр `Err_metric` - ошибка рассогласования виртуальных и реальных датчиков. Может быть использован для анализа уровня соответствия коэффициентов реальным сигналам датчика, а также для анализа уровня ошибки следящего контура при угловых ускорениях вращения датчика. Сигнал `Err_metric` используется для автоматического определения неисправности и формирования флага `NLock`. `NLock` формируется если `Err_metric` превышает заданный порог `Lock_th` (для немодулированных сигналов). Для модулированных сигналов для сравнения берется модуль максимальной амплитуды `Err_metric` за каждый период.

Для датчиков с конечным диапазоном угла в контуре отслеживается переполнение интегратора введен `C_LOOP_OVF` который сигнализирует о переполнении суммы второго интегрирующего звена. Возможна некорректная работа микросхемы.

*Последнее обновление **22 мая 2026 г.** от **islepchenkov***

# Блок модели датчика

В микросхему заложено несколько моделей вычисления виртуальных сигналов в зависимости от типа датчика. Выбор и настройка параметров модели дает возможность приблизить виртуальные сигналы к реальным, сокращая разницу между ними, что снижает нелинейности и шум координаты на выходе следящего контура. Если погрешности датчика и схемы не известны, регистры должны оставаться в значениях по умолчанию. Выбор типа модели осуществляется битами `Sensor_mode`.

## Модель датчика в режиме СКВТ/сельсин

При подключении СКВТ или сельсина используется модель СКВТ/сельсин (`Sensor_mode==0`). Для сельсина сигналы преобразуются в декартовую систему координат (аналогичную СКВТ) блоком масштабирования и преобразования координат.

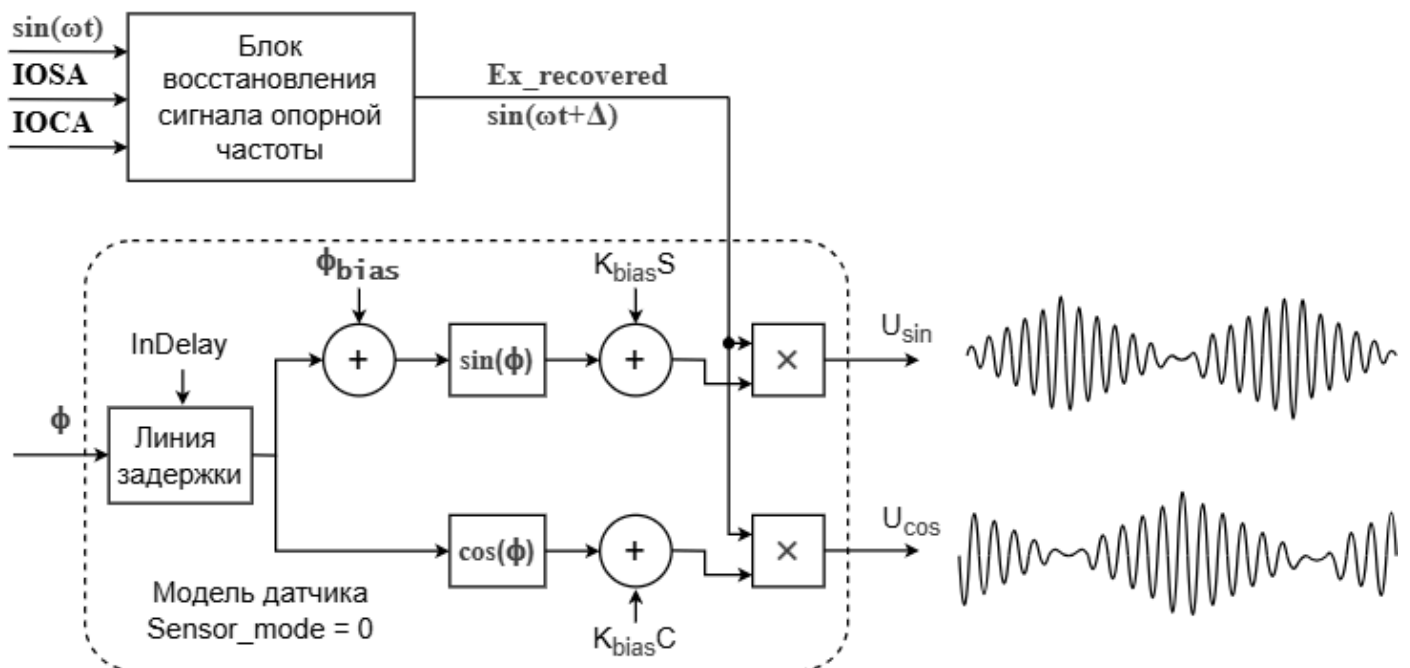


Схема модели СКВТ/сельсин (`Sensor_mode==0`)

Модель датчика в режиме СКВТ/сельсин использует восстановленный сигнал опорной частоты **`Ex_recovered`** (см. подраздел 5.6 «Блок восстановления сигнала опорной частоты») и вычисленный угол  $\phi$  для расчета сигналов по формулам:

$$U_S = Ex\_recovered \cdot \left( \sin \left( \phi' + \phi_{bias} \cdot \frac{\pi}{65536} \right) + K_{bias}S \cdot \frac{0,25}{32768} \right), \quad (9)$$

$$U_C = Ex\_recovered \cdot \left( \cos(\phi') + K_{bias}C \cdot \frac{0,25}{32768} \right), \quad (10)$$

где:

- **InDelay** – фиксированная дополнительная погрешность к групповой задержке сигналов в фильтрах на плате и в датчике. Основная часть групповой задержки сигнала  $U_{ex}$  в фильтрах и датчике имитируется с помощью операции восстановления до  $Ex\_recovered$  в блоке восстановления сигнала опорной частоты.
- $\phi'$  – угол в диапазоне  $[0 \div 2\pi]$ , вычисленный на предыдущем шаге, задержанный на **InDelay** тактов  $f_{clk}$ .
- $K_{bias}S, K_{bias}C$  – коэффициенты смещения нуля по каналам  $\sin$  и  $\cos$  (знаковое число в дополнительном коде);
- $\phi_{bias}$  – неортогональность осей X и Y.

## Модель ЛРДТ по 5-проводной схеме

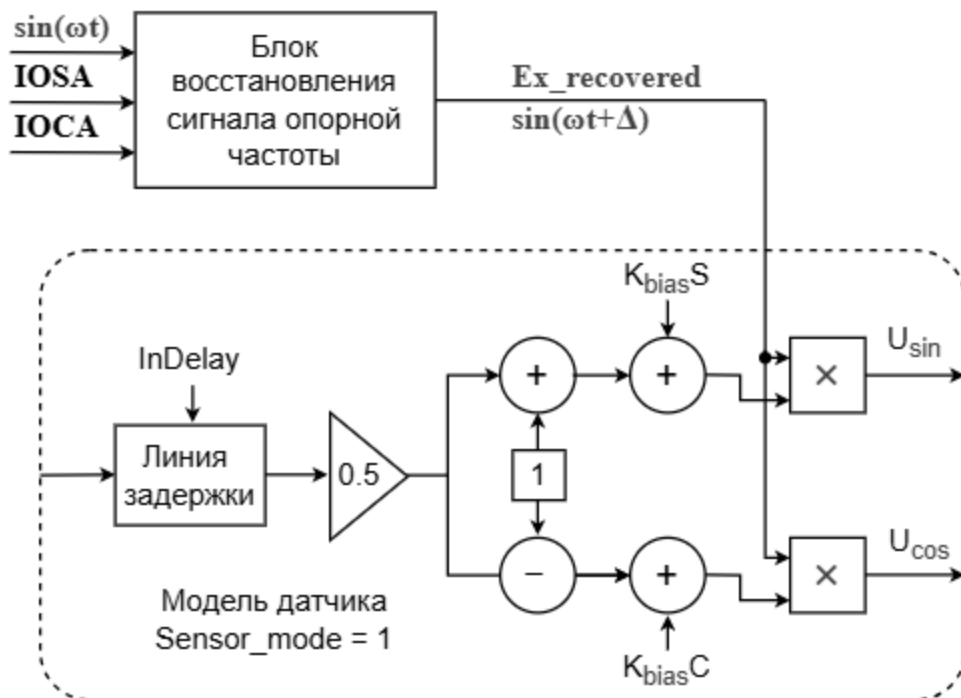


Схема модели ЛРДТ с двумя независимыми выходами (Sensor\_mode==1)

$$U_S = Ex\_recovered \cdot \left( 1 + 0,5\phi + KbiasS \cdot \frac{0,25}{32768} \right), \quad (11)$$

$$U_C = Ex\_recovered \cdot \left( 1 - 0,5\phi + KbiasC \cdot \frac{0,25}{32768} \right), \quad (12)$$

где:

- **KbiasS, KbiasC** – коэффициенты смещения нуля по каналам S и C соответственно (знаковое число в дополнительном коде);
  - $\phi$  – координата в диапазоне от  $-1$  до  $+1$ , вычисленная на предыдущем шаге, задержанная на **InDelay** тактов  $f_{clk}$ .
  - Умножение координаты  $\phi$  на 0.5 растягивает выход контура на рабочий диапазон датчика, у которого наведенная ЭДС выходных независимых обмоток не меняет своей полярности.
- 

Последнее обновление **22 мая 2026 г.** от **islepchenkov**

# Полоса пропускания контура

## Полоса пропускания преобразователя

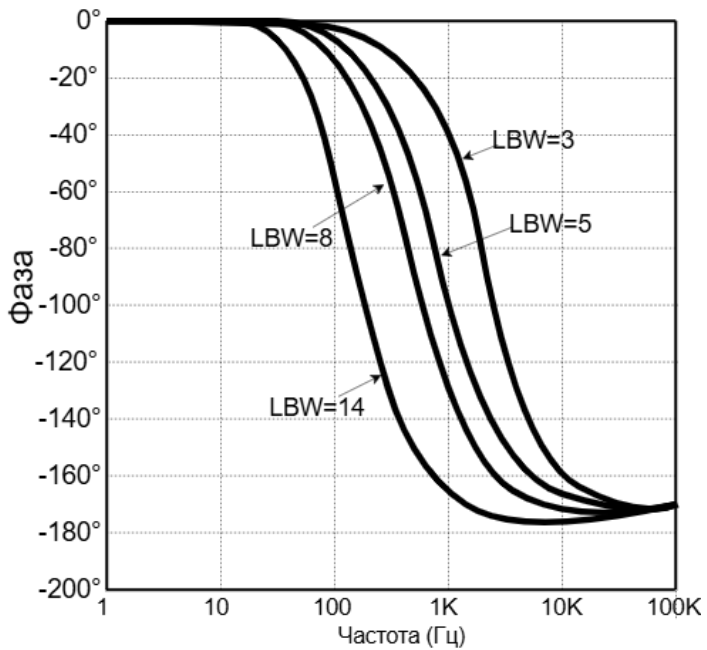
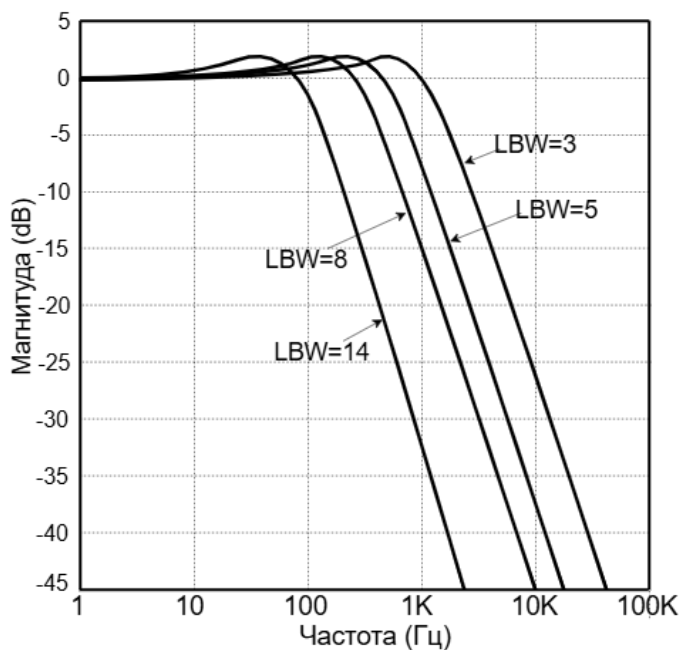
Полоса пропускания контура настраивается пользователем путем установки значения в битах **LBW** регистров **C1KonturStngs/C2KonturStngs**.

Таблица 9 – Полоса пропускания преобразователя

| <b>LBW</b> | <b>Полоса пропускания, Гц</b> | <b>Разрядность, бит (в лабораторных условиях, fex = 10 кГц)</b> | <b>Длительность переходного процесса, мс</b> |
|------------|-------------------------------|---|--|
| 3          | 5617                          | 10  | 0.8  |
| 4          | 3984                          | 12  | 1  |
| 5          | 2840                          | 12  | 1.5  |
| 6          | 2015                          | 12  | 2  |
| 7          | 1428                          | 12  | 3  |
| 8          | 1011                          | 13  | 4  |
| 9          | 716                           | 13  | 6  |
| 10         | 507                           | 14  | 8  |
| 11         | 358                           | 14  | 12   |
| 12         | 253                           | 14  | 16   |

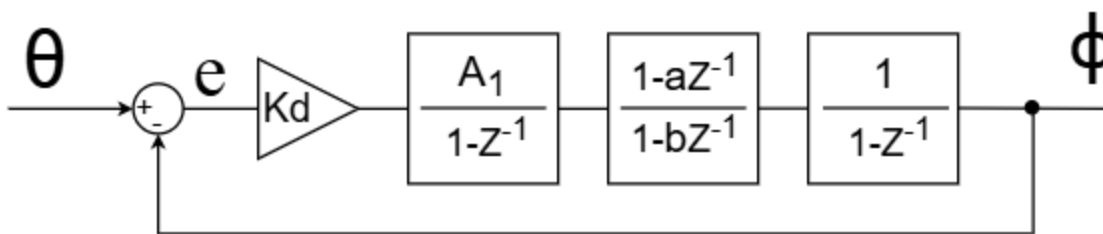
| <b>LBW</b> | <b>Полоса пропускания, Гц</b> | <b>Разрядность, бит (в лабораторных условиях, fex = 10 кГц)</b> | <b>Длительность переходного процесса, мс</b> |
|------------|-------------------------------|---|--|
| 13         | 179                           | 14  | 24   |
| 14         | 127                           | 15  | 32   |
| 15         | 89                            | 15  | 48   |
| 16         | 63                            | 16  | 64   |
| 17         | 45                            | 16+   | 96   |
| 18         | 32                            | 16+   | 128  |
| 19         | 22.4                          | 16+   | 192  |
| 20         | 15.9                          | 16+   | 256  |

Выбор значения **LBW** меняет коэффициенты петлевого фильтра таким образом, чтобы обеспечить переходной процесс, близкий по форме к критическому, при номинальных сигналах на входах микросхемы. При выборе полосы пропускания контура **LBW** необходимо найти компромисс между требуемой разрядностью и динамикой преобразования. Чем уже полоса пропускания контура, тем большая ошибка возникает при ускорении вала, и тем длительнее переходной процесс первоначального нахождения угла. В то же время, чем шире полоса пропускания контура, тем меньшую разрядность можно получить на выходе преобразователя. Значение полосы пропускания контура, записываемое в регистр **LBW**, выбирается из таблицы 9 или настраивается экспериментально, в зависимости от используемой схемы подключения преобразователя, необходимой динамики преобразования и разрядности. Если частота цикла преобразования  $f_{clk}$  отличается от номинальной (1024 кГц), полоса пропускания контура меняется пропорционально изменению частоты.



АЧХ и ФЧХ контура для разных LBW

Форма переходного процесса также остается постоянной, изменяется только время переходного процесса. Вид переходного процесса в контуре при номинальной амплитуде входного сигнала близок к критическому.



Контур 1

Передаточная функция открытого контура (с незамкнутой обратной связью) выражается следующей функцией

$$W(z) = \frac{A_1 \cdot K_d}{(1 - z^{-1})^2} \cdot \frac{(1 - a \cdot z^{-1})}{(1 - b \cdot z^{-1})} = \frac{A_1 \cdot \left[ \frac{\pi}{2^{8+LBW}} \right]}{(1 - z^{-1})^2} \cdot \frac{1 - \left[ \frac{2^{6+0.5LBW}}{2^{6+0.5LBW}} \right] \cdot z^{-1}}{1 - \left[ \frac{2^{6+0.5LBW}}{2^{6+0.5LBW}} \right] \cdot z^{-1}}, \quad (13)$$

где LBW - значение, записанное в регистр LBW[4:0].

$A_1$  принимает следующие значения:

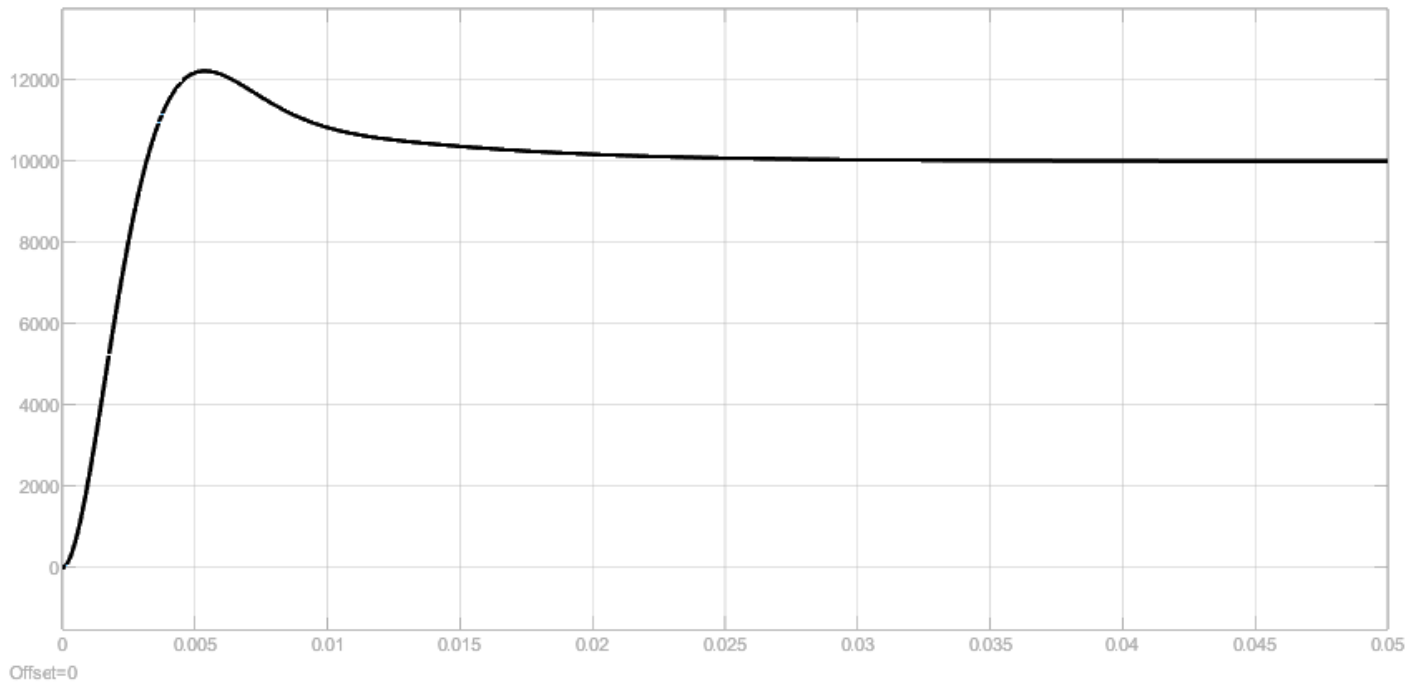
- $A_1 = 1$  при  $Sensor\_mode = 0$ ;
- $A_1 = 6 (\sim 4 \cdot \sqrt{2})$  при  $Sensor\_mode == 1$ ;

- $A_1 = 3 (\sim 2 \cdot \sqrt{2})$  при  $Sensor\_mode == 2$ .

Если установлен бит  $DC\_carrier$ , то:

- $A_1 = 0,75 (\sim \frac{1}{\sqrt{2}})$  при  $Sensor\_mode == 0$ ;
- $A_1 = 4$  при  $Sensor\_mode == 1$ ;
- $A_1 = 2$  при  $Sensor\_mode == 2$ .

Формула (13) верна для максимальной амплитуды сигнала на входах преобразователя  $\pm 0,8$  (в микросхеме  $\pm 0,8$  соответствует сигналу на входе микросхемы  $2,0$  В [пик-пик] и значениям в регистрах юстировки по умолчанию). Масштабирование сигналов осуществляется в блоке масштабирования и преобразования координат.



Переходной процесс в замкнутом контуре (размерность времени для  $LBW = 14$ )

Преобразуя  $K_d$  из формулы (13), можно рассчитать максимальную ошибку при ускорении:

$$K_a = \frac{\text{Ускорение} \left( \frac{\text{рад}}{c^2} \right)}{\text{Ошибка (рад)}} = \frac{A_1 \cdot K_d \cdot (f_{\text{clk}})^2}{10} = \frac{A_1 \cdot 2 \cdot \pi \cdot (f_{\text{clk}})^2}{10 \cdot 2^{8+LBW}} \left( \frac{1}{c^2} \right). \quad (14)$$

Полоса преобразователя выбирается пользователем в зависимости от соотношения сигнал/шум на входе микросхемы, максимальной допустимой ошибки при ускорении и времени переходных процессов. Например, при повышенном шуме и помехах внешней схемы возможно получение необходимой разрядности путем настройки меньшей полосы

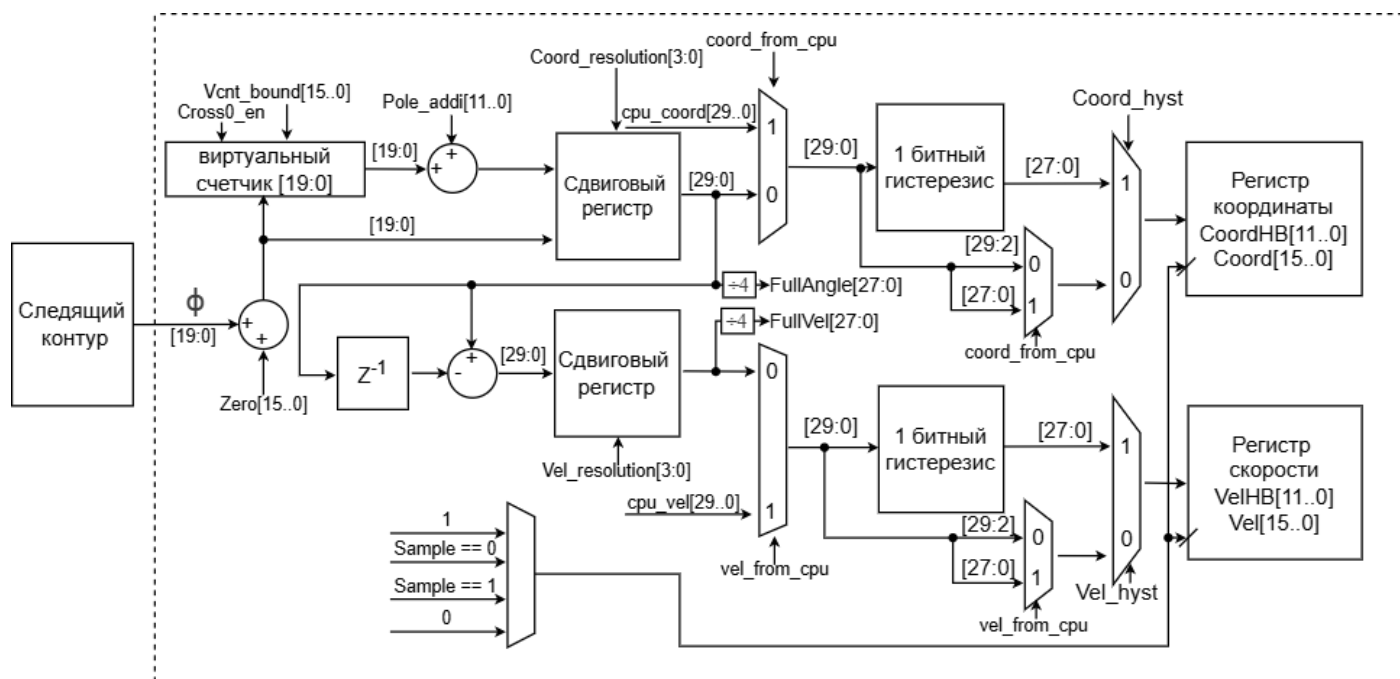
пропускания контура. При этом время переходных процессов и ошибка при ускорении увеличиваются.

*Последнее обновление **22 мая 2026 г.** от **islepchenkov***

# Блок обработки результатов

Блок позволяет представить результат преобразования в формате удобном для пользователя.

При выборе **Sensor\_mode=0** результат преобразования беззнаковый и соответствует углу от 0 до 2π. При выборе **Sensor\_mode != 0** результат преобразования знаковый и соответствует координате от -1 до +1.



Структура блока обработки результата

Координата, вычисляемая в следящем контуре, всегда имеет разрядность 20 бит (младшие 2 бита требуются для гистерезиса). Координата может быть скорректирована на постоянную величину, задаваемую в регистре **Zero**. Коррекция производится путем знакового сложения 20-битной координаты следящего контура с значением, заданным в регистре **Zero**, помноженной на 16.

В некоторых применениях СКВТ требуется отслеживание переходов через нуль выходного угла (многополюсные СКВТ, недостаточный темп опроса датчика при высокой частоте вращения, малые ресурсы для вычислений и т.д.), для чего в преобразователь

введен 20 битный виртуальный счетчик переходов через нуль выходной координаты следящего контура. Счетчик в темпе работы преобразователя добавляет единицу если значение двух старших разрядов [19:18] координаты изменилось из 3 в 0, и наоборот счетчик декрементируется если значение старших разрядов изменилось из 0 в 3. Для включения счетчика требуется установить бит `En_cross0` регистра `KonturStngs` в единицу (при `En_cross0` значение счетчика равно нулю при `Sensor_mode==0` или разumnoжено знаком старшего [19] разряда выходной координаты следящего контура при `Sensor_mode != 0`). Значение 20 битного виртуального счетчика переполняется как двоичное число (`C1Vcnt_bound==65535`), однако для датчиков с количеством полюсов некратным степени двойки возможно введение другого порога с помощью записи в регистр `C1Vcnt_bound/C2Vcnt_bound` значения, которое умножается на 16 и добавляется 15. В этом случае координата может быть приведена к двоичной простым умножением на коэффициент.

20 разрядов координаты следящего контура дополняются слева 20 разрядами виртуального счетчика и получившееся значение подается на регистр сдвига "вправо". Величина сдвига задается битами `Coord_resolution` в регистре `ResCntrl` (по умолчанию равно 2, что дает 16 битный код угла следящего контура в регистрах `C1Coord/C2Coord`). 28 битов [29:2] с регистра сдвига могут быть считаны по SPI по адресам `C1CoordHB`, причем старшая часть (`C1CoordHB`) всегда синхронизирована с младшей на уровне интерфейса (значения тактируемого регистра с нечетным адресом фиксируется регистра с адресом на единицу меньше (четного)).

При необходимости может быть включен гистерезис координаты (с помощью установки бита `Coord_hist`), который для своей работы использует младшие два бита с выхода регистра сдвига. При включенной схеме гистерезиса границы переключения значения выхода координаты сдвигаются на  $\pm 0,75$  от предыдущего значения. Если отклонение нового значения координаты составило менее  $\pm 0,75$  от значения, вычисленного на предыдущем шаге, регистр `Coord` не меняется. Таким образом, при неподвижном датчике, и, если позволяет соотношение сигнал шум на входе преобразователя, значение в регистре `Coord` будет постоянным. Даже, если координата датчика находится на границе между соседними значениями, гистограмма кодов будет содержать только одно значение. При использовании постобработки измеренной координаты (например, фильтрации) следует отключить схему гистерезиса, т.к. она является нелинейным элементом и уменьшает точность преобразования. При отключенной схеме гистерезиса, если координата датчика находится посередине между двумя значениями, гистограмма кодов будет содержать примерно одинаковое количество этих значений.

Скорость вычисляется в следящем контуре как приращение координаты на каждом такте преобразования  $f_{clk}$ .

В зависимости от настройки битов `Sample_src` (регистр `Mode_config`) возможно запоминание вычисленных преобразователем значений координаты, скорости.

**Таблица 10 – Настройка режима запоминания**

| Значение битов<br><code>Sample_src[1:0]</code> | Режим запоминания значений в регистрах <code>Coord</code> , <code>Vel</code> ,<br><code>Stat</code>                             |
|--|---|
| 00   | Сохраняются предыдущие значения   |
| 01   | Вход <code>Sample == 1</code> – защелкиваются новые значения<br>Вход <code>Sample == 0</code> – сохраняются предыдущие значения |
| 10   | Вход <code>Sample == 1</code> – сохраняются предыдущие значения<br>Вход <code>Sample == 0</code> – защелкиваются новые значения |
| 11 (по умолчанию)                              | Защелкивание в регистры происходит во время транзакции<br>чтения по интерфейсу SPI  |

Биты `Sample_src[1:0]` и вход `Sample` не оказывают влияния на блок эмуляции квадратурного энкодера, работающий на тактовой частоте микросхемы.

Последнее обновление **22 мая 2026 г.** от *islepchenkov*

# Эмуляция квадратурного энкодера

## Эмуляция квадратурного энкодера

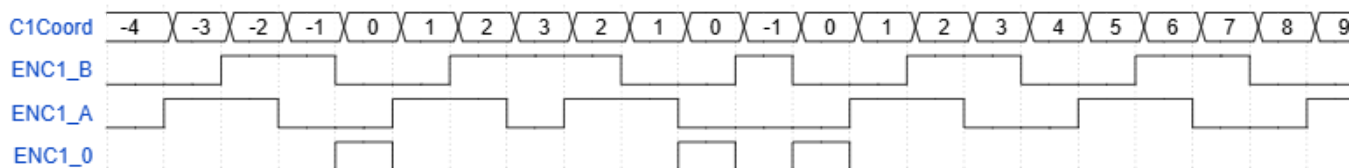
Микросхема содержит выходы эмуляции квадратурного энкодера для замены датчиков типа энкодер на СКВТ, а также для подключения микросхемы к устройствам, принимающим на вход сигналы типа энкодер.

Эмуляция квадратурного энкодера использует те же разряды координаты, что попадают в регистр C1Coord/C2Coord .

Младшие N разрядов координаты и скорости могут быть замаскированы (занулены) с помощью настройки N = `Vel_resolution[3..0]` регистров `C1ResCntnl/C2ResCntnl` (предполагается при использовании режима энкодера значение скорости рассчитанное контуром не будет использоваться).

Сигнал `ENC_0` устанавливается в состояние логической единицы, когда вычисленная координата равна 0.

После сброса микросхемы `ENC_A`, `ENC_B` устанавливаются в состояние логического нуля, `ENC_0` устанавливается в состояние логической единицы.



Временная диаграмма сигналов эмуляции квадратурного энкодера

Максимальная скорость переключения сигналов `ENC_A`, `ENC_B`, `ENC_0` задается битами `Enc_presc[2:0]` в регистрах `C1ResCntnl/C2ResCntnl` путём деления тактовой частоты генератора микросхемы `FINT` (см. таблицу ниже). `FINT` определяется настройкой PLL.

**Таблица 11 – Временные характеристики сигналов эмуляции квадратурного энкодера**

| <code>Enc_presc[2:0]</code> | Частота переключения сигналов <code>ENC_A</code> , <code>ENC_B</code> , <code>ENC_0</code> , <code>DIR</code> |
|-----------------------------|---|
| <code>000</code>            | <code>FINT/2</code>   |
| <code>001</code>            | <code>FINT/3</code>   |
| <code>010</code>            | <code>FINT/4</code>   |
| <code>011</code>            | <code>FINT/5</code>   |
| <code>100</code>            | <code>FINT/6</code>   |
| <code>101</code>            | <code>FINT/16</code>  |
| <code>110</code>            | <code>FINT/32</code>  |
| <code>111</code>            | <code>FINT/64</code>  |

Последнее обновление **17 июн. 2026 г.** от **Ilya Slerchenkov**

# Блок определения ошибок подключения

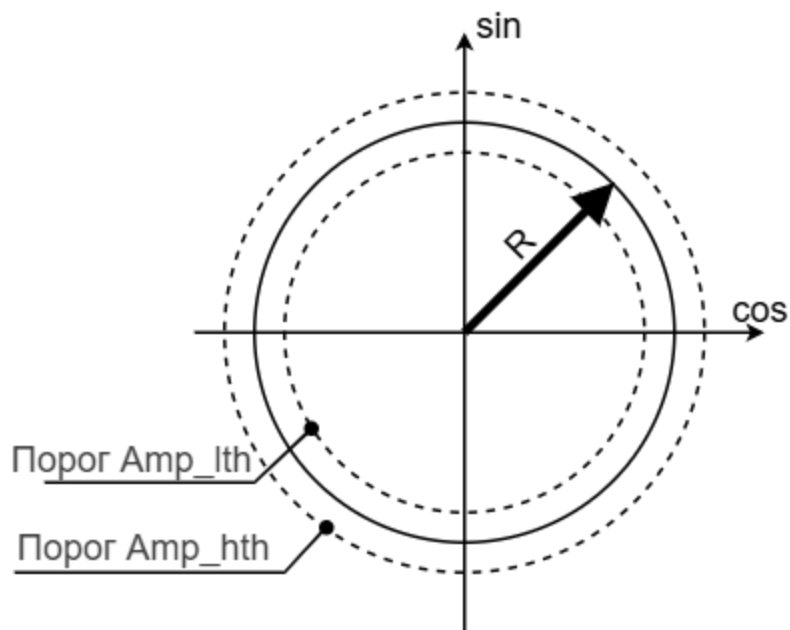
Микросхема производит определение корректности подключения датчика и детектирование ошибок преобразования. Причина появления ошибки записывается в регистры **C1Stat/C2Stat**. При наличии ошибки в регистрах **C1Stat/C2Stat** значения скорости и координаты могут принимать некорректные значения.

- Микросхема проверяет корректность сигналов на входе микросхемы. При превышении входных сигналов допустимой амплитуды выставляются флаги **CLIP\_SIN** и **CLIP\_COS**.
- При наличии в сигналах на входе микросхемы большой постоянной составляющей выставляется флаг **ADC\_OVF**.
- При задании слишком больших коэффициентов **C1KampS**, **C1KampC** или **C2KampS**, **C2KampC** в блоке масштабирования и преобразования координат и вызванном этим переполнении внутри микросхемы выставляется флаг **CORR\_OVF**.
- При большом сдвиге фазы между сигналами на входах микросхемы и сигналом **Ex\_ref** выставляется флаг **EX\_PH\_OUTRANGE**.

Микросхема производит определение корректной амплитуды сигнала на входе следящего контура путем сравнения амплитуды входного сигнала, представленного в полярных координатах с порогами. В режиме **Sensor\_mode==0** амплитуда сигнала **R**, используемого для сравнения с порогами, вычисляется по формуле

$$R = \sqrt{\text{Sin}1^2 + \text{Cos}1^2}, \quad (15)$$

где **Sin1** и **Cos1** – сигналы на входе следящего контура (амплитудой от -1 до +1).

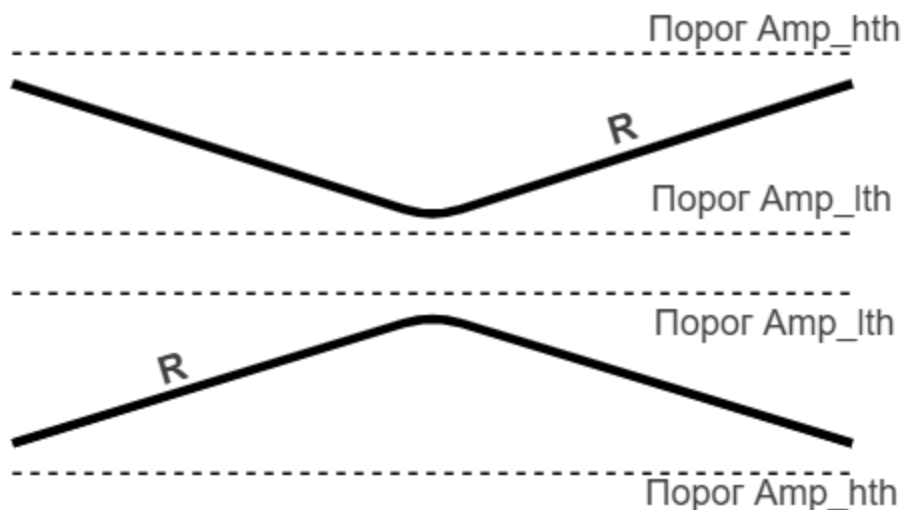


Определение амплитуды и наличия сигналов на входе преобразователя в режиме  $Sensor\_Mode=0$

В режимах  $Sensor\_mode==1$  и  $Sensor\_mode==2$  амплитуда сигнала  $R$ , используемого для сравнения с порогами, вычисляется по формуле

$$R = |\sin 1 + \cos 1|, \quad (16)$$

где  $\sin 1$  и  $\cos 1$  – сигналы на входе следящего контура (амплитудой от -1 до +1).



Определение амплитуды и наличия сигналов на входе преобразователя в режимах  $Sensor\_Mode=1, Sensor\_Mode=2$

При  $R > Amp\_hth$  выставляется флаг  $UIN\_HIGH = 1$ . При  $R < Amp\_lth$  выставляется флаг  $UIN\_LOW = 1$ . Данные флаги могут использоваться для определения пропадания

сигналов на входе микросхемы, присутствия сигналов неправильной формы или амплитуды.

Флаг **NLock** выставляется в состояние логической единицы при сравнении разницы между сигналами преобразователя и сигналами внутренней модели с порогом, заданным в регистре **Lock\_tn**= 1. Флаг может использоваться для определения большого отклонения вычисленной координаты от координаты датчика (например, из-за ускорения вала).

Биты в регистре **Mask** маскируют соответствующие биты в регистре **Stat**. При записи логического нуля в один из битов регистра **Mask** вычисление соответствующего ему бита регистра **Stat** блокируется и принимается равным логическому нулю. При равенстве всех битов регистра **Stat** нулю выставляются сигналы **Ready1** (преобразователь 1) и **Ready2** (преобразователь 2), означающие отсутствие ошибок преобразователя.

*Последнее обновление 22 мая 2026 г. от islepchenkov*

# Интерфейс SPI

## Описание структуры SPI

Взаимодействие управляющего микроконтроллера с микросхемой 5400TP065A-022 осуществляется через последовательный SPI интерфейс с помощью четырех сигналов: **SCLK**, **SSTR**, **SDO**, **SDI**.

- **SSTR** — сигнал разрешения работы контроллера. Активный уровень "0". Переход из 1 в 0 является событием начала транзакции (или цепочки транзакций). При **SSTR** = 1 производится сброс счетчиков синхроимпульсов **SCLK**.
- **SCLK** — сигнал синхронизации. Микросхема защелкивает входные данные (вход **SDI**) по переднему фронту **SCLK** и выставляет выходные данные **SDO** по заднему фронту **SCLK**. Ведущее устройство должно принимать данные с входа **SDO** по переднему фронту и передавать **SDI** по заднему фронту **SCLK**.
- **SDI** — порт для приема и выдачи данных. Выдача данных на порт производится только при однократном переводе контроллера SPI в режиме полудуплексного чтения.
- **SDO** — порт выдачи данных, при неактивном уровне **SSTR**, **SDO** всегда переводится в состояние высокого импеданса ("Z")

Контроллер SPI работает только по синхроимпульсам **SCLK** и не зависит от наличия системной частоты FINT. Взаимодействие с контроллером осуществляется через последовательности кадров. Каждый кадр состоит из **16 полных синхроимпульсов SCLK** (имеющих и передний, и задний фронт).

- Контроллер имеет два 4-битных счетчика:
  - Счетчик синхроимпульсов запроса (инкрементируется передними фронтами **SCLK**)
  - Счетчик синхроимпульсов ответа (инкрементируется задними фронтами **SCLK**)
- При **SSTR** = 1 счетчики сброшены.
- При наступлении **17-го синхроимпульса** счетчики переполняются, что сигнализирует о начале нового кадра (позволяет использовать DMA).

Существуют три вида кадра

- прием на **SDI** командного слова и одновременная выдача на **SDO** данного по предыдущему адресу
- прием на **SDI** данного для записи (на выводе **SDO** выдается существующее данное по адресу записи)
- выдачей данных на **SDI** в полудуплексном режиме (вывод **SDO** дублирует выдачу SDI)

Формат командного слова: **M<sub>2</sub>,M<sub>1</sub>,M<sub>0</sub>, A<sub>10</sub>,A<sub>9</sub>,A<sub>8</sub>, A<sub>7</sub>, A<sub>6</sub>, A<sub>5</sub>, A<sub>4</sub>, A<sub>3</sub>, A<sub>2</sub>, A<sub>1</sub>, A<sub>0</sub>, 0, P**

- **M<sub>2</sub>, M<sub>1</sub>, M<sub>0</sub>** — код операции (см. таблицу ниже)
- **A<sub>10</sub>...A<sub>0</sub>** — адрес
- **P** — бит четности

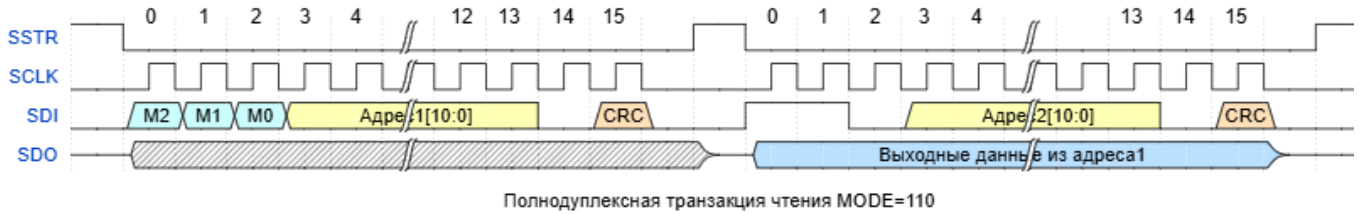
## Коды операций

| Код операции<br>(M <sub>2</sub> ,M <sub>1</sub> ,M <sub>0</sub> ) | Описание   |
|---|--|
| 100   | Запись по адресу значением из следующего кадра                                 |
| 110   | Вывод значения ячейки по адресу в следующем кадре (полнодуплексная транзакция) |
| 001   | Вывод значения ячейки по адресу в следующем кадре (полудуплексная транзакция)  |
| 010   | Команда блокировки значения в выходном регистре                                |
| 101   | Команда разблокировки записи выходного регистра                                |

Контроллер SPI может находиться в одном из 3 состояний:

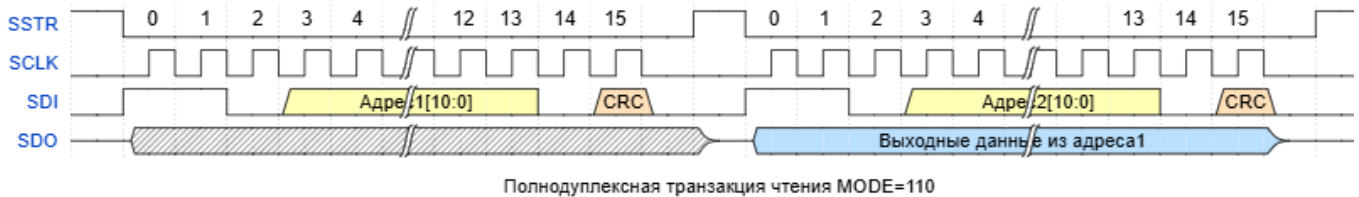
1. Прием командного слова
2. Прием слова данных (если предыдущее командное слово интерпретировалось как запись)
3. Выдача слова данных в полудуплексном режиме

Коды операции "110", "010", "101" оставляют контроллер SPI в 1 состоянии, следующий кадр может содержать командное слово, а на выводе SDO будет передаваться значение адреса зафиксированного в предыдущем кадре.

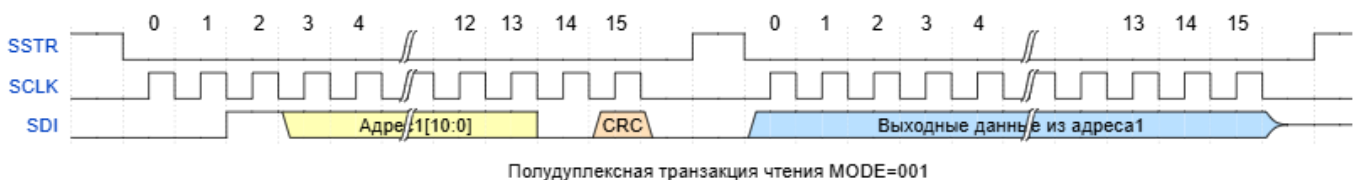


### Полнодуплексная транзакция чтения режим 110

Переход в 2,3 состояние из состояния 1 возможен при корректно принятым командным словом с кодом операции "100" или "001". При нахождении контроллера в 2 и 3 состоянии данные на выводе SDI не интерпретируются как командное слово, по окончании кадра контроллер переходит в состояние 1.



### Транзакция записи — режим 100

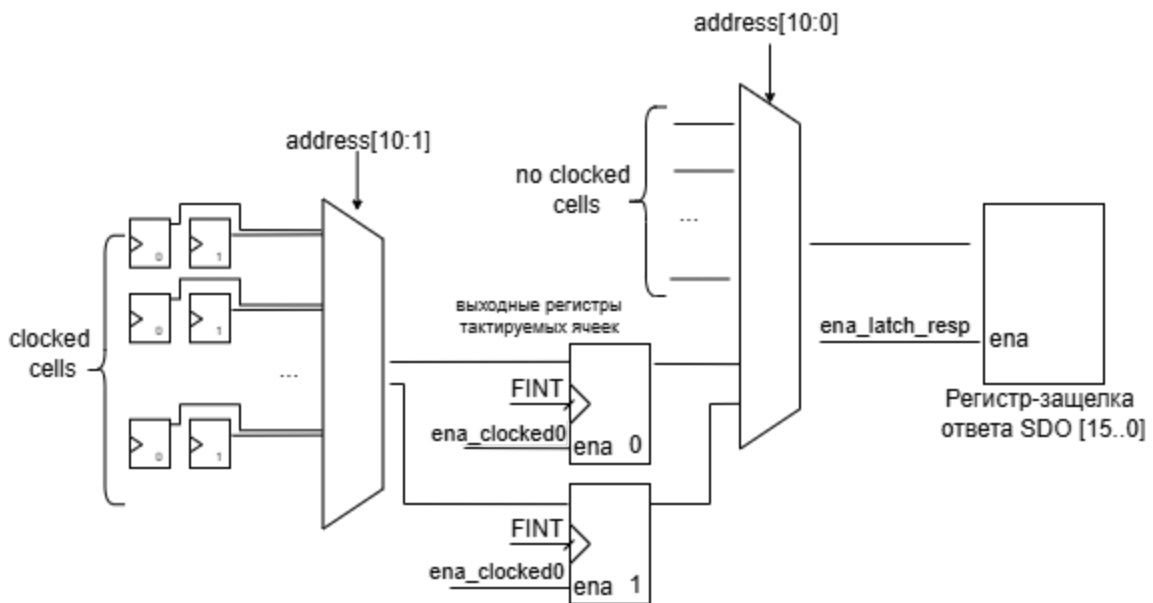


### Полудуплексная транзакция чтения режим 001

Доступ к ячейкам памяти — параллельный: один мультиплексор подключает нужную ячейку к контроллеру SPI по заданному адресу.

Фиксация адреса, мультиплексирование по адресу регистра для чтения производится при приеме 14-го переднего фронта командного слова, не дожидаясь завершения окончания транзакции и приема бита четности. Однако контроллер перейдет в другое состояние только если поступило 16 передних фронтов SCLK, 14-й бит регистра запроса равен 0 и корректен бит четности (или отключен контроль).

Для тактируемых FINT ячеек выход мультиплексора подключен к промежуточному регистру. Разрешение копирования данных на него снимается "ena\_clocked" при заднем фронте **SSTR** или 16-м заднем фронте **SCLK** (при непрерывной циклической передаче). Сигнал "ena\_clocked" (см. рис. ниже) пересинхронизируется двумя триггерами на FINT, что требует паузы до первого фронта **SCLK** длиной в 3 периода FINT (параметр данной задержки конфигурируется в большинстве архитектур микроконтроллеров). Нечетные адреса всех тактируемых ячеек защелкиваются в промежуточный регистр только при чтении четного адреса на единицу меньше. При повторном чтении нечетного адреса значение в промежуточном регистре не обновится. Для чтения актуального значения нечетного адреса возможно при последовательной передаче транзакций команд чтения сначала четного адреса, затем нечетного адреса на единицу больше.

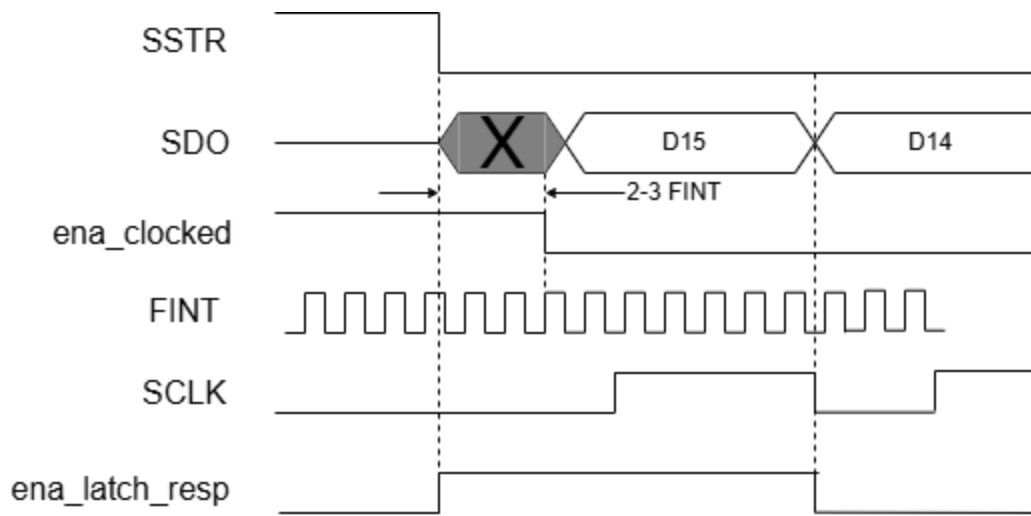


Формат командного слова SPI



Чтение тактируемой ячейки с нечетным адресом

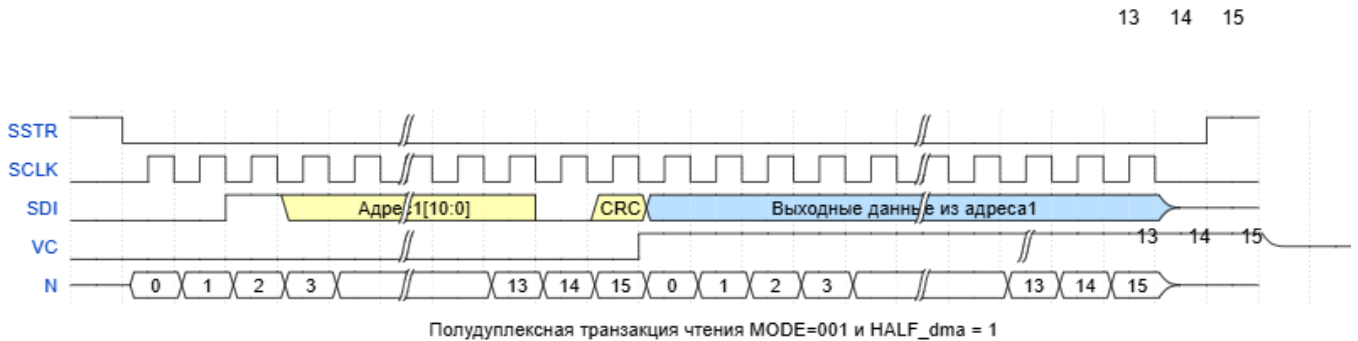
Чётный/нечётный адрес SPI



Временная диаграмма

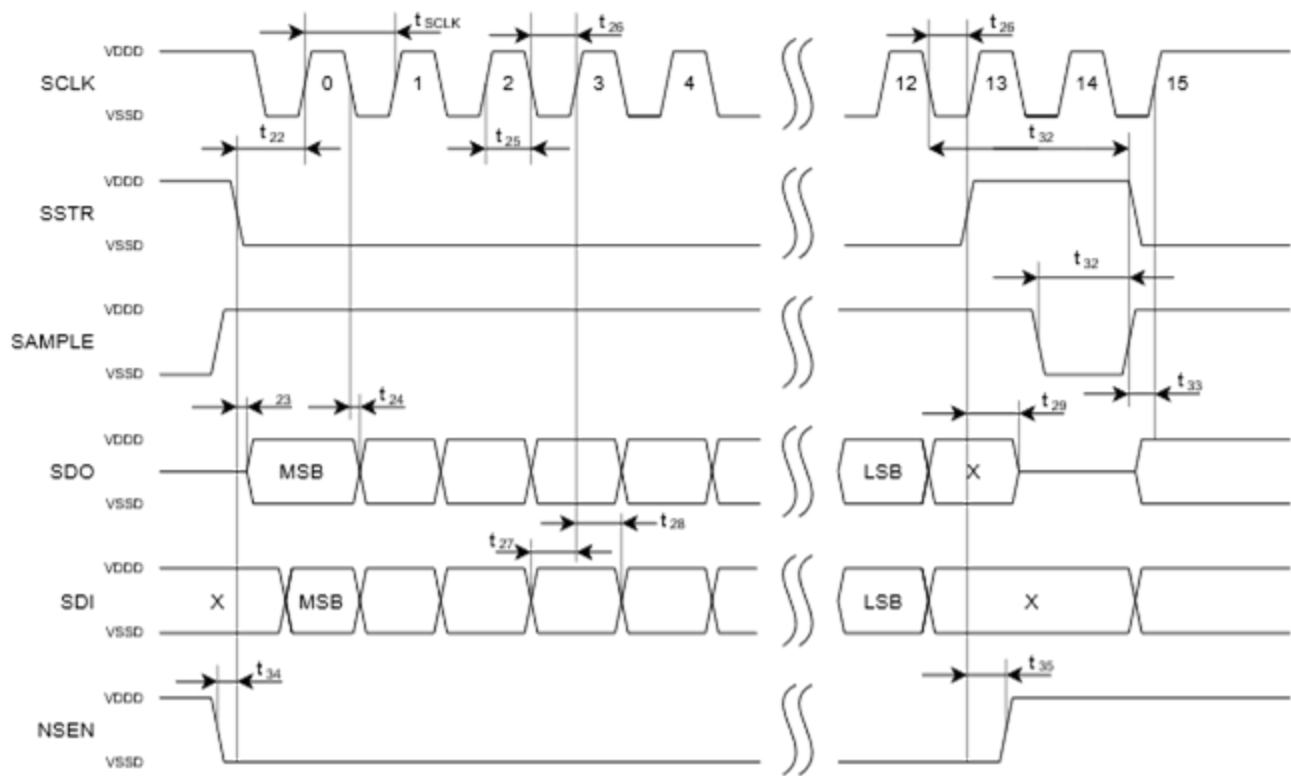
Выходной регистр-защелка не может поменять свое значение начиная с 1 по 15 заднего фронта **SCLK**.

Контроллер SPI может быть переведен в режим полудуплексной передачи без перевода **SSTR** в 1 (циклический режим). В этом случае требуется установить бит **HALF\_dma** = 1, что даст указание контроллеру переводить вывод **SDI** и приемо-передатчик сигналом **VC** в активное состояние выдачи на спаде 16 **SCLK** текущего кадра приема командного слова.



Полудуплексная транзакция чтения режим 001 и HALF\_dma=1

Вывод **nSEN** маскирует сигнал **SSTR**. При **nSEN** = 1 интерфейс SPI отключается (используется при подключении нескольких микросхем). Если обмен только с одной микросхемой, **nSEN** должен быть подтянут к 0.



Временная диаграмма

## Временные параметры последовательного интерфейса

| Параметр, единица измерения   | не менее | типовое | не более |
|---|----------|---------|----------|
| Период <b>SCLK</b> ( $t_{SCLK}$ ), нс   | 50       |         |          |
| Задержка от заднего фронта <b>SSTR</b> до переднего фронта <b>SCLK</b> ( $t_{22}$ ), нс | 6,0      |         |          |
| Задержка от заднего фронта <b>SSTR</b> до занятия линии <b>SDO</b> ( $t_{23}$ ), нс     |          |         | 39       |
| Задержка от заднего фронта <b>SCLK</b> до переключения <b>SDO</b> ( $t_{24}$ ), нс      |          |         | 41       |
| Длительность лог. «1» на <b>SCLK</b> ( $t_{25}$ ), нс                                   | 16       |         |          |

| Параметр, единица измерения  | не менее | типовое | не более |
|--|----------|---------|----------|
| Длительность лог. «0» на SCLK (t_26), нс   | 16       |         |          |
| Задержка от выставления SDI до переднего фронта SCLK (t_27), нс                            | 6,0      |         |          |
| Задержка от переднего фронта SCLK до изменения SDI (t_28), нс                              | 6,0      |         |          |
| Задержка от заднего фронта SSTR до освобождения SDO (t_29), нс                             |          |         | 29       |
| Задержка от заднего фронта SCLK до переднего фронта SSTR (t_30), нс                        | 6,0      |         |          |
| Задержка от заднего фронта SCLK при SSTR=0 до переднего фронта Sample (t_31), нс           | 6,0      |         |          |
| Длительность лог. «0» на Sample (t_32), нс   | 50       |         |          |
| Задержка от переднего фронта Sample до первого переднего фронта SCLK при SSTR=0 (t_33), нс | 4,0      |         |          |
| Задержка от заднего фронта nSEN до заднего фронта SSTR (t_34), нс                          | 0        |         |          |
| Задержка от переднего фронта SSTR до переднего фронта nSEN (t_35), нс                      | 0        |         |          |

## Дополнительные режимы SPI

Целевое применение 5400TP065A-022 - индуктивные датчики угла и положения в аппаратуре специального назначения. Для снижения уровня помех и искажений рекомендуется размещение 5400TP065A-022 в непосредственной близости от источника ЭДС. Для обеспечения цифрового обмена рекомендуется использование

дифференциальных пар - передача сигналов по току (RS-485, M-LVDS, LVDS и др.). Для повышения надежности передачи данных и упрощения использования множества преобразователей в микросхеме 5400TP065A-022 реализовано несколько дополнительных функций.

### Фиксация значения выходной ячейки для множественного чтения.

При передаче кода операции "010" сигнал разрешения с регистра (четный и нечетный адрес) тактируемых ячеек маскируется, значение его не обновляется. Это дает возможность прочитать значения несколько раз и сравнить целостность передачи данных. Разблокировка производится кодом операции "101".

### Чтение ранее переданной транзакции

Чтение адреса 73 `SPI_req` возвращает ранее переданный в контроллер SPI кадр. Это позволяет проверить соответствие принятого значения с запрошенным адресом.



Чтение предыдущей транзакции по адресу 73

### Блокировка записи слов

При записи в регистр `WR_Lock` значения отличного от нуля, запись в другие регистры блокируется (кроме `BUS_addr`), только чтение.

### Программный адрес

При наличии нескольких микросхем подключенных к одной общей шине для обеспечения индивидуального доступа к каждой микросхеме в их регистры `IC_addr` могут быть предварительно записаны различные номера в диапазоне [1..255]. Чтобы команда чтения или записи сработала для конкретной микросхемы мастер устройство должно предварительно записать в регистр `BUS_addr` значение номера необходимой микросхемы, и затем следующие необходимые транзакции записи и чтения. Запись в регистр `BUS_addr` не блокируется значениями регистров `WR_Lock` и `IC_addr`.

Контроллер SPI конкретной микросхемы реагирует на команды только в следующих случаях

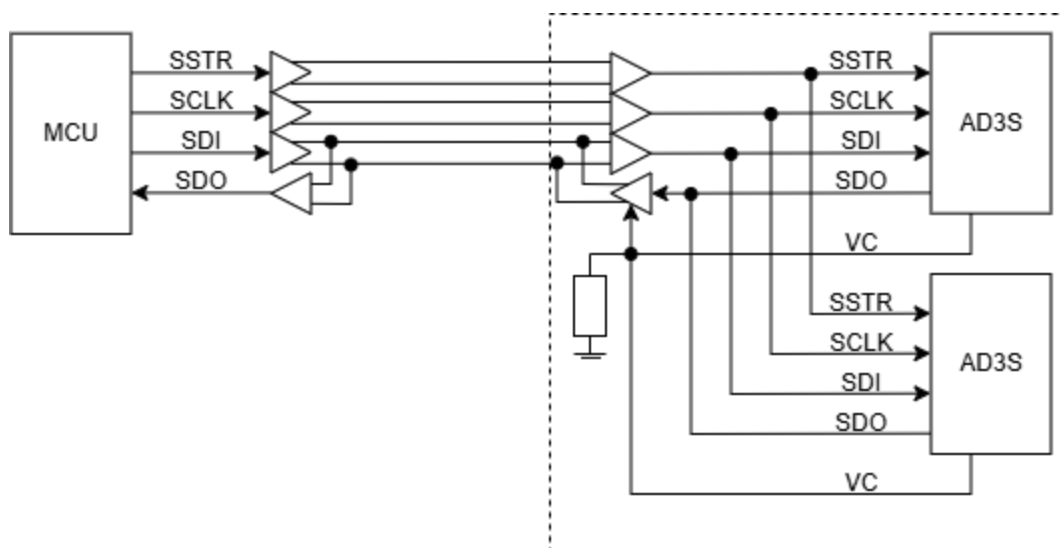
- значение IC\_addr равно 0
- IC\_addr != 0, BUS\_addr == IC\_addr
- IC\_addr != 0, BUS\_addr == 0, флаг BUS0\_mode == 1

В остальных случаях контроллер не реагирует (за исключением перезаписи BUS\_addr).

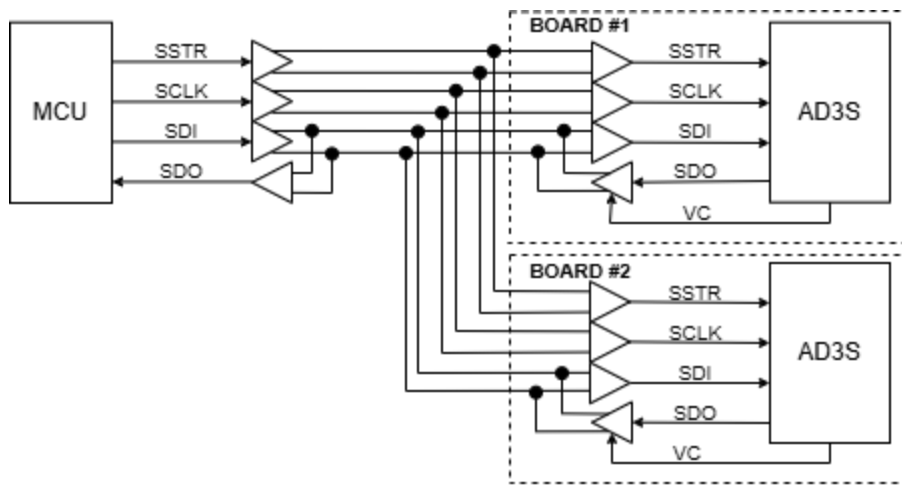
Подразумевается, что конкретное значение IC\_addr записывается в ПЗУ микросхемы до установки в общий интерфейс аппаратуры.

### Управление внешним приемо-передатчиком с помощью вывода VC

Установка бита VC\_mode в 1 регистра AFE\_config переводит вывод VC в режим управления внешним приемо-передатчиком. Бит DE\_half определяет когда вывод VC переходит в активное состояние: при DE\_half = 1 - только для ответа при полудуплексных запросах, при DE\_half = 0 при всех запросах когда микросхема должна отвечать (когда SSTR = 0, nSEN = 0 и есть соответствие адреса микросхемы и шины адреса). Бит DE\_inv определяет какой уровень VC является активным для передатчика: при DE\_inv = 0 активный 1, при DE\_inv = 1 активный уровень 0. Установка бита DE\_turnZ в 1 переводит вывод VC в состояние высокого импеданса ("Z") между передачами, а сигнал VC должен быть доопределен подтяжкой резистора (требуется при подключении нескольких микросхем к одному передатчику). На рисунках ниже представлены некоторые варианты включения микросхемы.



Подключение двух преобразователей на одной плате с единым интерфейсом в полудуплексном режиме передачи



Подключение двух независимых преобразователей на один интерфейс в полудуплексном режиме передачи

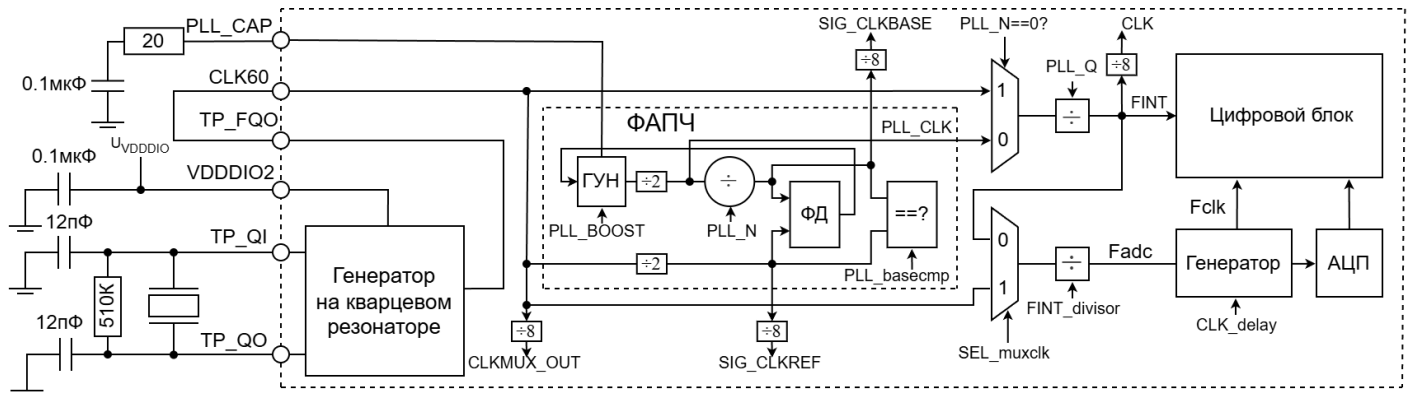
## Режим параллельной передачи результата

Режим включается при  $SPI\_ext\_en = 1$ . Выводы  $ENC1\_A$ ,  $ENC1\_B$ ,  $ENC1\_0$ ,  $ENC2\_A$ ,  $ENC2\_B$ ,  $ENC2\_0$  переключаются параллельно с  $SDO$  и передают данные из регистров  $C1Coord/C2Coord$ ,  $C1CoordHB/C2CoordHB$ ,  $C1Vel/C2Vel$ ,  $C1VelHB/C2VelHB$ .

## Назначение выводов в режиме параллельной передачи результата

| SDI | VC | ENC1_A  | ENC1_B  | ENC1_0    | SDO       | ENC2_A  | ENC2_B  | ENC2_0    |
|-----|----|---------|---------|-----------|-----------|---------|---------|-----------|
| 0   | 0  | C1Coord | C1Vel   | C1CoordHB | C1VelHB   | C2Coord | C2Vel   | C2CoordHB |
| 0   | 1  | C1Vel   | C1Coord | C1VelHB   | C1CoordHB | C2Vel   | C2Coord | C2VelHB   |
| 1   | 0  | C2Coord | C2Vel   | C2CoordHB | C2VelHB   | C1Coord | C1Vel   | C1CoordHB |
| 1   | 1  | C2Vel   | C2Coord | C2VelHB   | C2CoordHB | C1Vel   | C1Coord | C1VelHB   |

# Тактирование

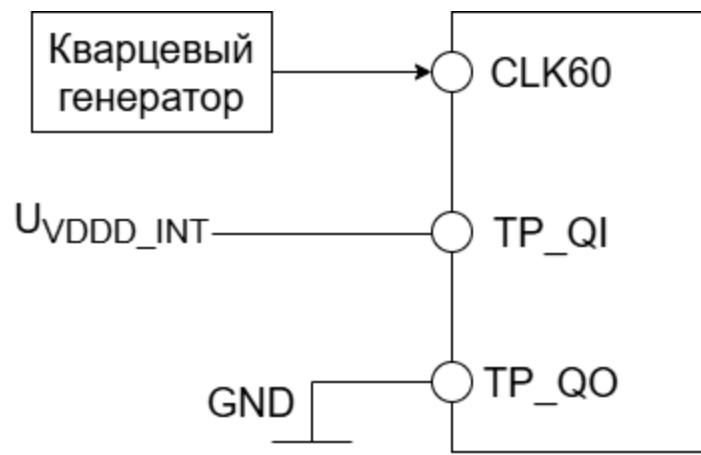


Организация тактирования

## Выбор источника тактирования

Источник тактирования определяется схематически одним из двух вариантов:

1. **С помощью встроенного генератора на внешнем кварцевом резонаторе**, подключенного к выводам **TP\_QI** (28) и **TP\_QO** (27). Выход генератора должен **TP\_FQO** (26) быть подключен на тактовый вход **CLK60** (39). Для работы генератора требуется подать на вывод **VDDIO2** питание, равное питанию **VDDIO**. Генератор может работать с кварцевыми резонаторами с номиналом по частоте от 2 МГц до 20 МГц. Следует учитывать, что выход на режим генератора составляет от нескольких миллисекунд.
2. **С помощью внешнего тактового генератора** с частотой в диапазоне 10 МГц...60 МГц, подключенного к выводу **CLK60** (39). Вывод **TP\_QI** (28) должен быть подключен к **VDDD\_INT** (25), вывод **TP\_QO** (27) должен быть подключен к GND.



## Настройка умножителя частоты в ФАПЧ

Умножитель частот может быть задействован, если входная частота тактирования не более 20 МГц. Для включения умножителя требуется задать `PLL_N` в диапазоне `3..74`. Если поле `PLL_N` равно нулю, то цифровой блок использует тактирование со входа микросхемы.

Частота сигнала `CLK_PLL` с выхода умножителя определяется произведением входной частоты на коэффициент (`PLL_N`) и не должна превышать 60 МГц.

Время установления частоты и величина пульсаций определяется номиналом петлевого фильтра, состоящего из внутреннего резистора 10 кОм и внешнего конденсатора, подключенного к выводу `PLL_CAP`. При увеличении емкости (относительно номинала 0,1 мкФ) уровень пульсаций уменьшается, однако время выхода на режим увеличивается. При уменьшении номинала время установления частоты уменьшается, однако уровень пульсаций увеличивается. Уровень пульсаций не должен превышать 5% от частоты.

Для ускорения зарядки петлевого фильтра (уменьшения времени переходного процесса) требуется задать бит `PLL_boost` в единицу. Момент отключения предзаряда петлевого фильтра определяется полем `PLL_basecmp` регистра `PLL_config`. При его задании включается счетчик импульсов опорной частоты `CLKREF` и частоты с выхода делителя `PLL_BASE`. Компаратор удерживает сигнал сброса до того момента пока количество импульсов с делителя ГУН не будет превышать количество использованных импульсов опорной частоты `CLK_REF` на  $2^{\text{PLL\_basecmp}}$  импульсов. Сигнал готовности компаратора выведен на бит SPI `CLKCOMP_rdy`.

Для анализа работы ФАПЧ в микросхеме предусмотрен отладочный вывод DB. При записи в поле `DB_mode` регистра `AFE_config` одного из значений "0", "1", "2", "3" на выводе DB будут выведены один из сигналов `CLKMUX_OUT`, `DB_SIG_PLLBASE`, `DB_SIG_CLKREF`, `CLK` поделенные на 8.

## Настройка частоты FINT

Для деления входной частоты требуется задать поле `PLL_Q` регистра `PLL_config` значением в диапазоне `1..15`. Если делить частоту не требуется, то задать `PLL_Q` нулем.

Для компенсации длительного выхода на рабочий режим источника тактирования (внешний тактовый генератор или внутренний генератор на кварцевом резонаторе) требуется ввести значение в диапазоне **1..3** поля **CLK\_delay** регистра **PLL\_config**. В этом случае тактируемая цифровая логика будет удерживаться в состоянии «сброс».

## Настройка частоты АЦП

С помощью бита **SEL\_muxclk** определяется источник тактирования АЦП. При использовании встроенного ФАПЧ для тактирования АЦП рекомендуется использовать сигнал напрямую с кварцевого генератора. Для работы АЦП требуется обеспечить тактирование с частотой  $F_{adc}$  не превышающей 20 МГц, для чего пользователь должен выбрать делитель частоты **FINT\_divisor**. Для одного преобразования АЦП требуется 16 тактов  $F_{adc}$ , а для работы следящего контура требуется не менее 32 тактов FINT. Если время цикла работы следящего контура превышает время одного преобразования АЦП, то пользователю необходимо замедлить АЦП добавлением пустых тактов  $F_{adc}$ . Их количество определяется битами **DELAY\_cycles**.

## Выключение тактирования

Для снижения энергопотребления, когда микросхема запитана, но не используется, имеется возможность выключить тактирование с цифрового блока с помощью бита **OSC\_DIS** или подачей активного уровня **1** на вывод **STNDBY**. Состояние включено/выключено тактирование может быть определено по биту **CLK\_rdy**.

*Последнее обновление 17 июн. 2026 г. от Ilya Slepchenkov*

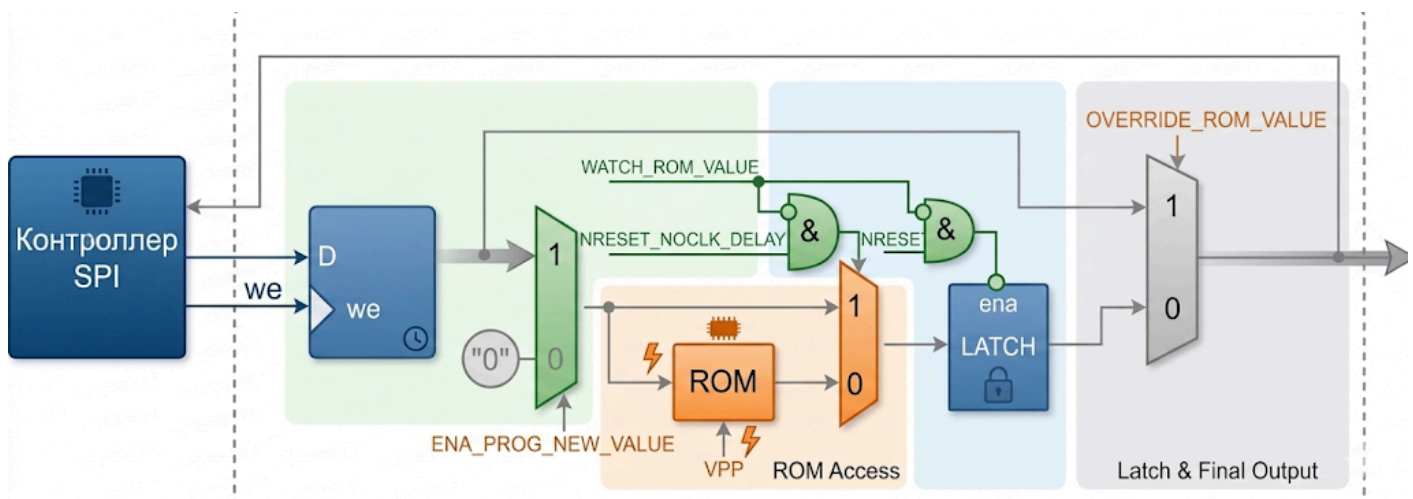
# Постоянное запоминающее устройство

## Структура

Микросхема содержит два сегмента ПЗУ для хранения пользовательских настроек, необходимых для автоматической инициализации регистров после подачи напряжения питания или сигнала сброса.

## ПЗУ с параллельным доступом

ПЗУ с параллельным доступом активируется непосредственно при включении питания и содержит 2 ячейки, которые неразрывно связаны с управляющими регистрами `PLL_config` и `INIT_conf`. На рисунке ниже представлена структура ячейки ПЗУ.



Структура ячейки ПЗУ с параллельным доступом

Доступ к настройкам `PLL_config` и `INIT_conf` осуществляется через управляющий регистр `UOTP_ctrl`, который содержит три бита: `WATCH_ROM_UVAL`, `PROG_NEW_UVAL` и `OVERRIDE_UVAL`. По умолчанию значения регистров `PLL_config` и `INIT_conf` равны нулю.

Для временного переопределения этих значений (до снятия питания) необходимо записать требуемые данные в соответствующие регистры (`PLL_config` и `INIT_conf`), а затем установить бит `OVERRIDE_UVAL` в логическую «1». При этом заданные значения

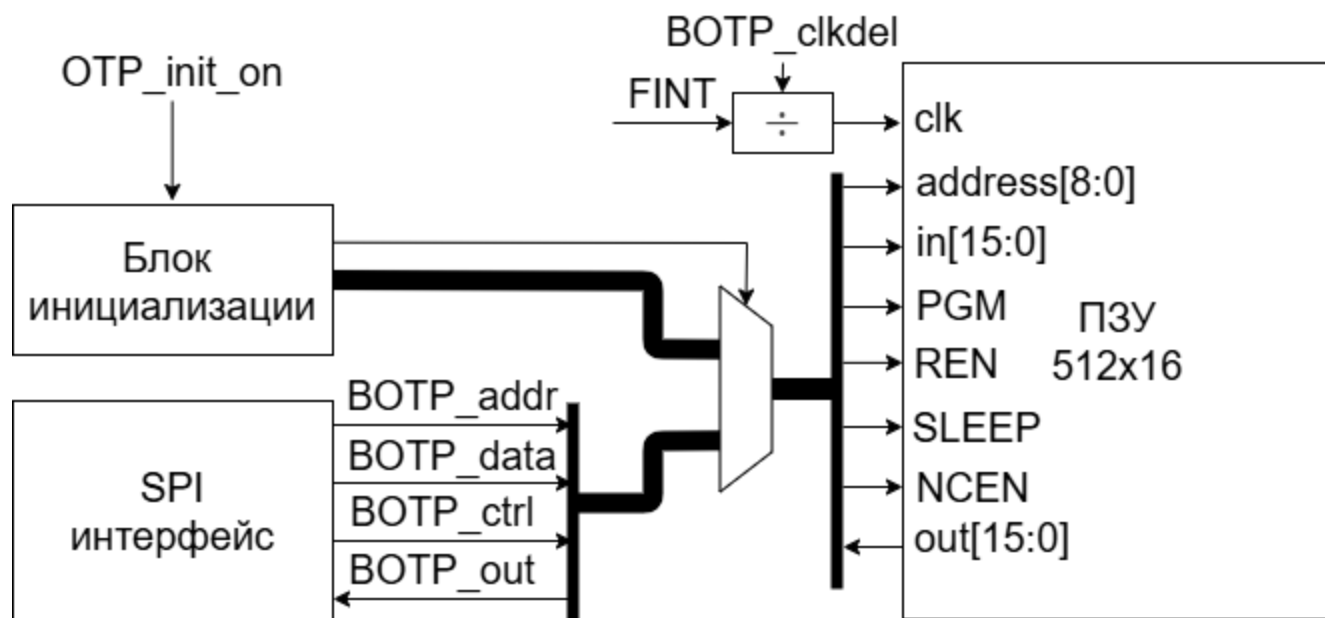
сохраняются даже при выполнении программного сброса с помощью бита `MANUAL_NRST_PLL` (путем записи нуля, а затем единицы).

Для программирования этих значений в энергонезависимую память ПЗУ необходимо выполнить следующую последовательность действий:

1. Записать в регистры `PLL_config` и `INIT_conf` нужные параметры.
2. Убедиться в корректности записи путем обратного чтения (при `OVERRIDE_UVAL = 1`).
3. Установить бит `PROG_NEW_UVAL` и подать напряжение программирования 9.8 В+0.3 В на вывод `VPP`.
4. По истечении заданного времени снять напряжение `VPP` и сбросить бит `PROG_NEW_UVAL`.
5. Установить бит `WATCH_ROM_UVAL` в логическую «1».
6. Считать значения из `PLL_config` и `INIT_conf`, чтобы подтвердить успешность записи.

Если считанные значения не совпадают с заданными, операцию программирования следует повторить.

## ПЗУ с последовательным доступом



ПЗУ с последовательным доступом

Второй сегмент ПЗУ предназначен для инициализации регистров конфигурации и ячеек памяти микровычислителей при включении питания. ПЗУ содержит 512 16-битных слов. Если выставлен бит `OTP_init_on`, после снятия сигнала сброс (при наличии тактирования),



| +       | 2 x 0        | 2 x 1    | 2 x 2        | 2 x 3     | 2 x 4       | 2 x 5   |
|---------|--------------|----------|--------------|-----------|-------------|---------|
| 2 x 484 | 232 cpu2     | C2KampS  | C2KampC      | C2KbiasS  | C2KbiasC    | C2fbias |
| 2 x 492 | C2EXInc      | C2Amp_th | C2InputStngs | C2Lock_th | C2Zero      | C2Mask  |
| 2 x 500 | C2Vcnt_bound | IC_addr  | ADC_config   | Mask_Stat | Flags_delay | MR_lock |
| 2 x 508 | Mode_config  | -        | -            | -         | -           | -       |

Запись и чтение ПЗУ производится с помощью управляющих регистров [BOTP\\_addr](#), [BOTP\\_data](#), [BOTP\\_out](#), [BOTP\\_ctrl](#).

Для корректной работы ПЗУ (записи и чтения, в т.ч. при инициализации после включения питания) требуется настройка тактирования. Исходя из значения частоты [FINT](#) требуется записать в регистр [BOTP\\_clkdel](#) такое значение для деления частоты [FINT](#) (в соответствии с таблицей из описания регистров), чтобы тактовая частота на блок ПЗУ не превышала 10 МГц (значение будет уточнено после проведения испытаний на плюсе и минусе).

Для записи данных в ячейку ПЗУ по указанному адресу требуется записать целевой адрес в регистр [BOTP\\_addr](#), а записываемое значение — в регистр [BOTP\\_data](#). Установить бит [NCEN](#) в лог. «0», бит [SLEEP](#) в лог. «0», бит [PGM](#) в лог. «1». Затем установить напряжение [VPP](#) с 4.0В на 9.5В на время 200 мс, переключить напряжение [VPP](#) на 4.0В, сбросить бит [PGM](#), установить бит [REN](#) и проверить записанное значение с помощью считывания ячейки [BOTP\\_data](#).

Внимание! Если Вы прожгли бит [INIT\\_on](#), при подаче питания микросхема не отвечает, то вероятной причиной является отсутствие надлежащей настройки делителя частоты. Для того чтобы микросхема ответила в этом состоянии нужно найти её [IC\\_addr](#), путем последовательной записи в регистр [BUS\\_addr](#) всех 255 возможных значений от 1 до 255. После нахождения адреса, нужно запрограммировать делитель [BOTP\\_clkdel](#) на значение не менее ## (определяется по результатам испытаний)

Внимание! Чтение ячеек последовательной ПЗУ после инициализации по событию подачи питания или сброса будет давать нулевые значения. Для того чтобы прочитать необходимо выставить бит OVERRIDE\_UVAL, снять бит INIT\_on, далее снять MANUAL\_NRST\_PLL, далее установить MANUAL\_NRST\_PLL, и тогда ячейки ПЗУ будут читаться.

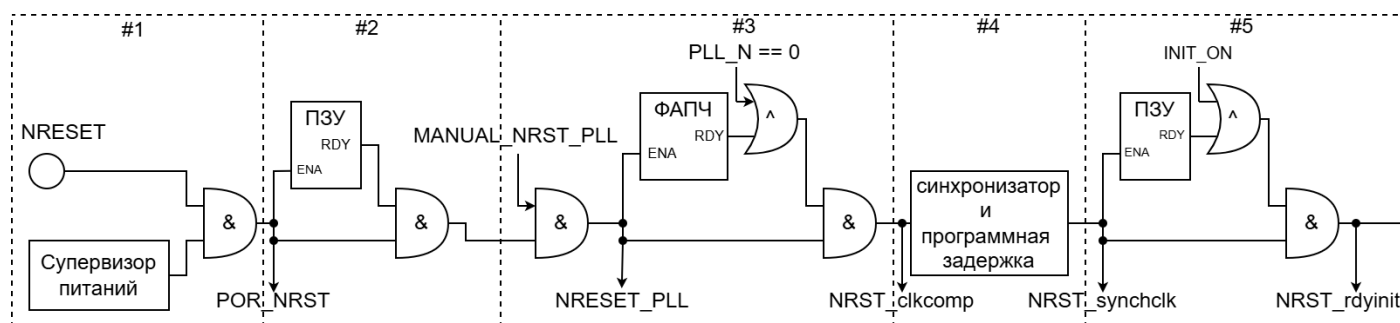
*Последнее обновление **17 июн. 2026 г.** от **Ilya Slepchenkov***

# Подсистема сброса

Система сброса решает следующие задачи:

- Блокирование работы микросхемы до установления требуемых уровней питания, блокирование работы цифрового блока до установления тактирования и инициализации начальными данными.
- Очистка памяти и установка всех регистров в начальное состояние по внешнему сигналу "сброс"

Сигнал "сброс" может быть подан сигналом низкого уровня на вывод RESET



Цепочка распространения сигнала сброс

## Таблица этапов распространения сигнала сброс

| # | Бит SPI  | Этап  | Описание  |
|---|----------|---|---|
| 1 | NRST_pll | Формирование                                | сигнал с вывода NRESET объединяется с сигналом готовности супервизора питания                       |
| 2 |          | Чтение настроек ПЗУ с параллельным доступом | Производится перезапись из ПЗУ с параллельным доступом настроек ИОН, ФАПЧ, и режимов инициализации. |

| # | Бит SPI       | Этап  | Описание  |
|---|---------------|---|---|
| 3 | NRST_clkcomp  | ФАПЧ  | Ожидание готовности аналоговой части ФАПЧ           |
| 4 | NRST_synchclk | Программная задержка                            | Синхронизация сигнала сброса и программная задержка |
| 5 | NRST_rdyinit  | Чтение настроек ПЗУ с последовательным доступом | Ожидание готовности ФАПЧ и программной задержки     |

Последнее обновление **22 мая 2026 г.** от *islepchenkov*

# Микровычислители

Микросхема содержит два независимых специализированных микровычислителя (CPU1 и CPU2), предназначенных для:

- коррекции статических нелинейностей датчика угла;
- согласования отсчётов в многополюсных датчиках;
- расчёта угловой скорости, ускорения и фильтрации;
- буферизации внутренних сигналов преобразователей.

Наличие двух CPU позволяет одному выполнять коррекцию нелинейностей, а другому — расчёт скорости и ускорения, что обеспечивает средний темп вычислений 50–150 мкс.

Каждый CPU имеет нестандартную разрядность 14 бит, сокращённый набор из 48 команд, 8 регистров общего назначения (R0–R5 по 14 бит, R6 и R7 по 28 бит) и 512 ячеек памяти программ и данных по 14 бит.

## Архитектура конвейера

Каждый CPU реализован на основе 4-ступенчатого конвейера:

| Ступень | Назначение  |
|---------|---|
| IF      | Выборка инструкции из памяти по адресу счётчика команд (PC)             |
| ID      | Декодирование инструкции, чтение регистров                              |
| EX      | Исполнение: АЛУ, CORDIC, умножитель, вычисление адреса, запись в память |
| MEM/WB  | Выбор результата, запись в регистр                                      |

Базовое время выполнения одной инструкции — 4 такта FINT. Результат умножения или операции CORDIC может быть готов позже, но не блокирует выполнение следующих

инструкций, если они не используют регистры R6 или R7.

Исключения:

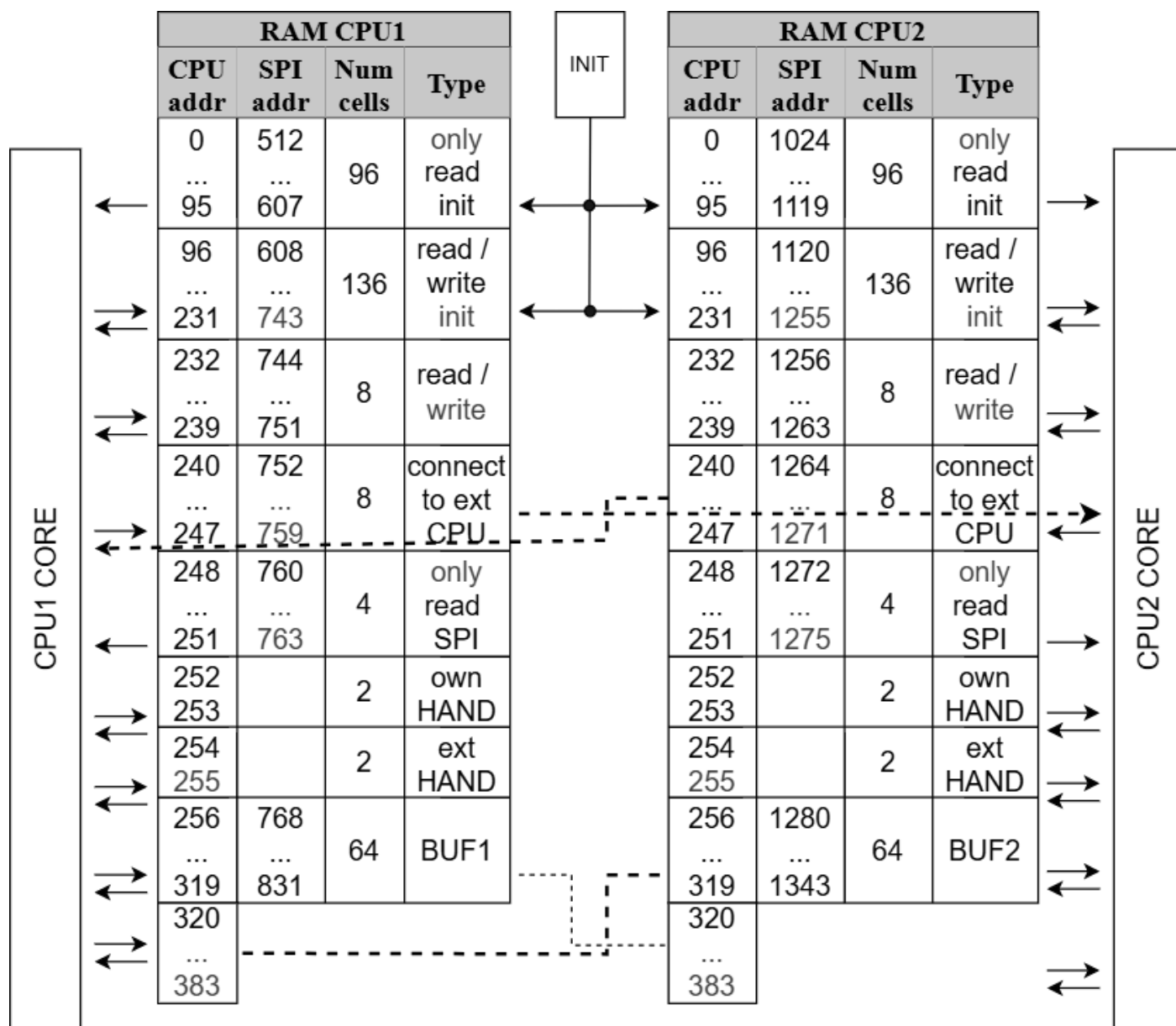
- **CORDIC** (SIN, COS, ATAN) — 16 итераций, ~17 тактов;
- **Умножение** (MULTI, MULTF и др.) — 5-10 тактов;
- **NOP N** — N+1 тактов (задержка);
- **WAIT\_\*** — блокировка до наступления события.

## Регистры общего назначения

| Регистр | Разрядность | Назначение  |
|---------|-------------|---|
| R0-R5   | 14 бит      | Регистры общего назначения. При чтении знакорасширяются до 28 бит   |
| R6      | 28 бит      | Результат умножения/сдвига. Всегда перезаписывается командами MULTI, MULTI_ACC, MULTF, MULTF_ACC, MULTCUBE_ACC, DEC2FLOAT |
| R7      | 28 бит      | Аккумулятор. Всегда перезаписывается командами ACC, ACC2, MULTI_ACC, MULTF_ACC, MULTCUBE_ACC                              |

Загрузка и сохранение R6 и R7 возможна только форматами по 28 бит (2 ячейки) или 56 бит (4 ячейки). Загрузить или сохранить только младшие 14 бит R6/R7 нельзя. Для операций с R6/R7 использовать команды LOAD32/STORE32, CLOAD32/CSTORE32.

## Структура памяти



Структура памяти CPU

Объём памяти каждого CPU — 512 ячеек по 14 бит. Карта адресов:

| Адреса  | Размер | Назначение                   | Доступ CPU            |
|---------|--------|------------------------------|-----------------------|
| 0-95    | 96     | Программный код (инструкции) | Только чтение         |
| 96-239  | 144    | Данные                       | Чтение/запись         |
| 240-247 | 8      | Обмен между CPU1 и CPU2      | Чтение/запись (кросс) |

| Адреса  | Размер | Назначение                                 | Доступ CPU    |
|---------|--------|--|---------------|
| 248–251 | 4      | Ячейки SPI                                 | Чтение/запись |
| 252–253 | 2      | HAND (данные преобразователя)              | Чтение/запись |
| 254–255 | 2      | HAND ext (данные внешнего преобразователя) | Чтение/запись |
| 256–383 | 128    | Буфер (BUF)                                | Чтение/запись |

При включённой функции инициализации (бит [INIT\\_on](#)) в первые 232 ячейки обоих CPU переписываются данные из ПЗУ с последовательным доступом.

## Ячейки обмена данными CPU

Ячейки с адресами 240–247 предназначены для передачи данных между CPU1 и CPU2. Шины чтения подключены кросс-образованием: чтение CPU1 по адресам 240–247 возвращает данные, записанные CPU2, и наоборот.

## Ячейки обмена с интерфейсом SPI

Ячейки памяти CPU доступны для чтения и записи через интерфейс SPI через регистры Dсру1LB/Dсру1HB и Dсру2LB/Dсру2HB. Адрес ячейки задаётся в составе транзакции SPI. При записи данные фиксируются в соответствующей ячейке памяти CPU. При чтении возвращается текущее значение ячейки.

## Буферы

В микросхему добавлены два дополнительных блока памяти BUF1 и BUF2 по 64 ячейки каждый (адреса 256–319). По умолчанию BUF1 расширяет память CPU1, а BUF2 — CPU2. При необходимости буфер может быть переключён на другой CPU с помощью битов [SHRD\\_RAM](#) и [SHRD\\_CPU2](#), что позволяет одному CPU работать с 128 дополнительными ячейками (адреса 256–383).

# Адресация памяти

Команды загрузки и сохранения используют 7-битное поле адреса. Фактический адрес зависит от типа команды и регистра:

| Команда          | Регистр | Ячейки | Диапазон адресов | Формула адреса              |
|------------------|---------|--------|------------------|-----------------------------|
| STORE/CSTORE     | R0–R5   | 1      | 128–255          | $\text{addr} + 128$         |
| LOAD/CLOAD       | R0–R5   | 1      | 128–255          | $\text{addr} + 128$         |
| STORE32/CSTORE32 | R0–R5   | 2      | 64–318 (чётные)  | $\text{addr} \times 2 + 64$ |
| LOAD32/CLOAD32   | R0–R5   | 2      | 64–318 (чётные)  | $\text{addr} \times 2 + 64$ |
| STORE/CSTORE     | R6, R7  | 4      | 64–318 (чётные)  | $\text{addr} \times 2 + 64$ |
| LOAD/CLOAD       | R6, R7  | 4      | 64–318 (чётные)  | $\text{addr} \times 2 + 64$ |
| STORE32/CSTORE32 | R6, R7  | 4      | 64–318 (чётные)  | $\text{addr} \times 2 + 64$ |
| LOAD32/CLOAD32   | R6, R7  | 4      | 64–318 (чётные)  | $\text{addr} \times 2 + 64$ |

Для 32-битных операций (STORE32, LOAD32) и операций с R6/R7 адрес всегда умножается на 2. Ячейка с чётным адресом помещается в младшую часть регистра, с нечётным (адрес +1) — в старшую. Для 4-ячейковых операций: `{data[addr+3], data[addr+2], data[addr+1], data[addr]}`.

## Форматы данных

### 28-битный формат с плавающей запятой

Формат является усечённым IEEE754 float32. Нет NaN, Inf, denormal.

#### Структура формата

| Поле | Биты   | Длина | Описание                         |
|------|--------|-------|----------------------------------|
| Sign | 27     | 1     | 0 = +, 1 = -                     |
| Exp  | 26..20 | 7     | Порядок, bias = 63               |
| Mant | 19..0  | 20    | Мантисса (IEEE754 fraction >> 3) |

### Кодирование (Float32 → Float28)

1.  $v = \text{single}(x)$  (IEEE754 float32)
2. Извлечь поля IEEE754:  $\text{Sign} = \text{bit}31$ ,  $\text{Exp}32 = \text{bits}30..23$ ,  $\text{Frac}32 = \text{bits}22..0$
3. Порядок: если  $\text{Exp}32 == 0$ , то  $\text{Exp}28 = 0$ , иначе  $\text{Exp}28 = \text{Exp}32 - 127$
4.  $\text{Exp} = \text{Exp}28 + 63$
5.  $\text{Mant} = \text{Frac}32 \gg 3$
6. Упаковка:  $\text{Float}28 = (\text{Sign} \ll 27) \mid (\text{Exp} \ll 20) \mid \text{Mant}$

### Float28 → Float32

1. Извлечь:  $\text{Sign} = \text{bit}27$ ,  $\text{Exp} = \text{bits}26..20$ ,  $\text{Mant} = \text{bits}19..0$
2. Восстановление:  $\text{Exp}32 = \text{Exp} - 63 + 127$ ,  $\text{Frac}32 = \text{Mant} \ll 3$
3.  $\text{Value} = \text{Sign} \times (1 + \text{Frac}32 / 2^{23}) \times 2^{(\text{Exp} - 63)}$

### Float28 → Integer

$\text{INTvalue} = \text{Sign} \times \text{Mant} \times 2^{(\text{Order} - 63)}$ , где  $\text{Sign} = +1$  если бит 27 = 0, иначе -1.

## Система команд

Все инструкции кодируются 14-битным словом. Биты [5:3] кодируют регистр-источник B (SEL\_B), биты [2:0] — регистр-приёмник A (SEL\_A). Регистры R0–R7 кодируются значениями 000–111.

## Арифметико-логические команды

| Команда | Код          | Аргументы | Описание  |
|---------|--------------|-----------|---|
| NOP     | 000000000000 | [N]       | Нет операции. При $N > 0$ — задержка $N+1$ тактов |
| CLR     | 000000000001 | A         | $A = 0$   |
| INC     | 000000000010 | A         | $A = A + 1$                                       |
| DEC     | 000000000011 | A         | $A = A - 1$                                       |
| ABS     | 000000001000 | A         | $A =  A $   |
| ACC     | 000000001111 | A         | $R7 = R7 + A$                                     |
| ACC2    | 00000001     | B, A      | $R7 = R7 + A + B$                                 |
| ADD     | 00000011     | B, A      | $A = A + B$                                       |
| SUB     | 00000100     | B, A      | $A = A - B$                                       |
| ORL     | 00000101     | B, A      | $A = A   B$                                       |
| ANL     | 00000110     | B, A      | $A = A \& B$                                      |
| MOVE    | 00000111     | B, A      | $A = B$   |
| CONST   | 1101         | val, A    | $A = \text{знакорасширенное val } (-64\dots+63)$  |

## Команды сдвига

| Команда | Код    | Аргументы | Описание   |
|---------|--------|-----------|--|
| LSL     | 000010 | count, A  | Логический сдвиг влево: $A = A \ll (\text{count} + 1)$ |

| Команда | Код    | Аргументы | Описание  |
|---------|--------|-----------|---|
| ASR     | 000011 | count, A  | Арифметический сдвиг вправо: $A = A \gg (\text{count} + 1)$ |

## Команды умножения и преобразования форматов

| Команда      | Код         | Аргументы | Описание   |
|--------------|-------------|-----------|--|
| MULTI        | 11110100    | B, A      | $R6 = A \times B$ (целое, A — 28 бит, B — 24 бит, R6 — 52 бит)                   |
| MULTI_ACC    | 11110101    | B, A      | $R6 = A \times B$ ; $R7 = R7 + R6$ (целое MAC)                                   |
| MULTF        | 11110110    | B, A      | $R6 = A \times B$ (A — целое 28 бит, B — float 28 бит, R6 — 56 бит)              |
| MULTF_ACC    | 11110111    | B, A      | $R6 = A \times B$ ; $R7 = R7 + R6$ (float MAC)                                   |
| MULTK24      | 11111011    | B, A      | $A = \text{RES}[51:22]$ , $\text{RES} = A \times B$ (целое, 24-битная константа) |
| MULTK14      | 11111100    | B, A      | $A = \text{RES}[40:13]$ , $\text{RES} = A \times B$ (целое, 14-битная константа) |
| MULTCUBE_ACC | 11111101    | B, A      | $R6 = (\text{ch1}+1)^{(\text{ch7}+1)} \times (B + A)$ ; $R7 = R7 + R6$           |
| DEC2FLOAT    | 11111110000 | A         | $R6 = \text{float}(A)$ — преобразование целого в формат float28                  |

## Команды работы с памятью

| Команда | Код  | Аргументы | Описание   |
|---------|------|-----------|--|
| STORE   | 0001 | addr, A   | Запись 14 бит: ram[addr] = A[13:0]                         |
| STORE32 | 0010 | addr, A   | Запись 28 бит: ram[addr] = A[13:0], ram[addr+1] = A[27:14] |
| LOAD    | 0011 | addr, A   | Чтение 14 бит: A = знакорасширение(ram[addr])              |
| LOAD32  | 1100 | addr, A   | Чтение 28 бит: A = {ram[addr+1], ram[addr]}                |

## Индексные команды работы с памятью

Индексные команды CLOAD/CSTORE используют дополнительное смещение rel (см. раздел [Канальные регистры](#)). Для работы этих команд необходимо включить режим командой LOADSTORE\_ON.

| Команда  | Код  | Аргументы | Описание   |
|----------|------|-----------|--|
| CSTORE   | 1001 | addr, A   | Запись 14 бит: ram[addr + rel] = A[13:0]                                 |
| CSTORE32 | 1010 | addr, A   | Запись 28 бит: ram[addr + rel] = A[13:0], ram[addr + rel + 1] = A[27:14] |
| CLOAD    | 1000 | addr, A   | Чтение 14 бит: A = знакорасширение(ram[addr + rel])                      |
| CLOAD32  | 1011 | addr, A   | Чтение 28 бит: A = {ram[addr + rel + 1], ram[addr + rel]}                |

## Команды условного перехода

Все переходы используют относительное смещение от текущего PC.

| Команда | Код         | Аргументы | Описание   |
|---------|-------------|-----------|--|
| EQUAL   | 010         | rel, B, A | PC = PC + rel если A == B (rel: -16...+15)                 |
| MORE    | 011         | rel, B, A | PC = PC + rel если A > B, знаковое (rel: -16...+15)        |
| DJNZ    | 11100       | rel, A    | A = A - 1; PC = PC + rel если A ≠ 0 (rel: -32...+31)       |
| CDJNZ   | 111010      | rel, CH#  | CH# = CH# - 1; PC = PC + rel если CH# ≠ 0 (rel: -16...+15) |
| EQUAL0  | 111011      | rel, A    | PC = PC + rel если A == 0 (rel: -16...+15)                 |
| JUMP    | 111100      | rel       | PC = PC + rel (rel: -128...+127)                           |
| RJUMP   | 11111111100 | A         | PC = A (абсолютный переход по регистру)                    |
| STOREPC | 11111111101 | A         | A = PC + 1 (сохранение адреса возврата)                    |

## CORDIC-команды

Аппаратный CORDIC-сопроцессор выполняет 16 итераций за ~17 тактов. SIN и COS вычисляются одновременно.

| Команда | Код         | Аргументы | Описание   |
|---------|-------------|-----------|--|
| SIN     | 00000000101 | A         | Вычисление $\sin(A)$ . Результат: A = знакорасширенный cordic_sin[15:0]. Аргумент — 18-битный угол (2 старших бита — квадрант) |
| COS     | 00000000110 | A         | Вычисление $\cos(A)$ . Результат: A = знакорасширенный cordic_cos[15:0]  |

| Команда | Код      | Аргументы | Описание  |
|---------|----------|-----------|---|
| ATAN    | 00000010 | B, A      | Вычисление atan2(A, B). Результат: A — 18-битный угол, B — аргумент cos, A — аргумент sin |

## Канальные регистры

Канальные регистры используются для индексной адресации памяти и организации циклов.

| Регистр | Разрядность | Команда загрузки | Источник данных |
|---------|-------------|------------------|-----------------|
| ch0     | 9 бит       | CFIX0            | A[8:0]          |
| ch1     | 8 бит       | CFIX1            | A[7:0]          |
| ch2     | 7 бит       | CFIX2            | A[6:0]          |
| ch3     | 6 бит       | CFIX3            | A[5:0]          |
| ch4     | 5 бит       | CFIX4            | A[4:0]          |
| ch7     | 2 бит       | CFIX7            | A[1:0]          |

Смещение rel для индексных команд вычисляется как взвешенная сумма:  $rel = ch0 + ch1 \times 2 + ch2 \times 4 + ch3 \times 8 + ch4 \times 16$

| Команда | Код         | Аргументы | Описание     |
|---------|-------------|-----------|--------------|
| CFIX0   | 11111110100 | A         | ch0 = A[8:0] |
| CFIX1   | 11111110101 | A         | ch1 = A[7:0] |
| CFIX2   | 11111110110 | A         | ch2 = A[6:0] |

| Команда    | Код         | Аргументы | Описание  |
|------------|-------------|-----------|---|
| CFIX3      | 11111110111 | A         | ch3 = A[5:0]  |
| CFIX4      | 11111111000 | A         | ch4 = A[4:0]  |
| CFIX7      | 11111111001 | A         | ch7 = A[1:0]  |
| CLR_CH     | 11111111010 | CH#       | CH# = 0. При CH# = 5 — очистка всех канальных регистров |
| CLOAD_CUBE | 11111111011 | A         | $A = (ch1+1)^{(ch7+1)}$ — полиномиальный базис          |

NUMCUBE: значение  $(ch1+1)^{(ch7+1)}$  определяет порядок полинома:

| ch7 | Значение      | Базис        |
|-----|---------------|--------------|
| 0   | ch1 + 1       | Линейный     |
| 1   | $(ch1 + 1)^2$ | Квадратичный |
| 2   | $(ch1 + 1)^3$ | Кубический   |
| 3   | 1             | Константа    |

## Команды управления HAND-интерфейсом

| Команда    | Код         | Аргументы | Описание                                      |
|------------|-------------|-----------|---|
| SET_A_HAND | 11111110001 | const     | Выбор номера пары ячеек преобразователя (0–7) |
| HFIX       | 11111110010 | A         | Установка номера HAND из регистра A[2:0]      |

## Команды управления и синхронизации

| Команда       | Код            | Описание  |
|---------------|----------------|---|
| SETB_STP      | 11111111110000 | Установка флага STP = 1                         |
| CLRB_STP      | 11111111110001 | Сброс флага STP = 0                             |
| SW_HTE1       | 11111111110010 | Переключение мультиплексора HAND1 на другой CPU |
| SW_HTE2       | 11111111110011 | Переключение мультиплексора HAND2 на другой CPU |
| WAIT_EXT_CPU  | 11111111111000 | Ожидание импульса от другого CPU                |
| PULSE_EXT_CPU | 11111111111001 | Генерация импульса для другого CPU              |
| WAIT_OWN_HAND | 11111111111010 | Ожидание данных от «родного» преобразователя    |
| WAIT_EXT_HAND | 11111111111011 | Ожидание данных от «внешнего» преобразователя   |
| LOADSTORE_ON  | 11111111111100 | Разрешение работы команд CLOAD/CSTORE           |
| LOADSTORE_OFF | 11111111111101 | Запрет работы команд CLOAD/CSTORE               |
| IDLE          | 11111111111111 | Остановка CPU (бесконечный NOP)                 |

## Взаимодействие с преобразователем

Подключение CPU1 и CPU2 к преобразователям (HAND1 и HAND2) осуществляется симметрично: для CPU1 «родным» (own) является HAND1, а внешним (ext) — HAND2. Для CPU2 — наоборот. Считывание данных производится с двойной шириной слова (28 бит): с own по адресам 252–253, с ext по адресам 254–255.

Выбор номера пары ячеек преобразователя выполняется командой SET\_A\_HAND.

| #ADDR | Содержимое   |
|-------|--|
| 0     | FullAngle[27:0]  |
| 1     | Wca[11:0], BScos[12], ex_shifted, Wsa[11:0], BSsin[12], ex_ref |
| 2     | VirtualS[12:0], ex_recovered, BSsin[12:0], ex_recovered_90dgr  |
| 3     | Amp_metric[11:0], Err_metric[15:0]                             |
| 4     | FullVel[27:0]  |
| 5     | PhiC[15:2], PhiS[15:2]   |
| 6     | VirtualC[12:0], ex_recovered, BScos[12:0], ex_recovered_90dgr  |
| 7     | Pole_addi, STAT  |

## Запись данных в преобразователь

Доступ к записи в каждый преобразователь есть только у одного CPU. Выбор CPU с доступом на запись определяется конфигурационным битом [HandToEXT](#). CPU может инвертировать действие флага HandToEXT с помощью команд SW\_HTE1 и SW\_HTE2 (цифра — номер преобразователя).

## Запись Coord/Vel

Результаты вычислений CPU могут быть записаны в блок гистерезиса скорости и координаты. Для этого установите флаг [Coord\\_from\\_cpu](#) или [Vel\\_from\\_cpu](#) и запишите данные, установив командой SET\_A\_HAND соответствующий адрес (0 для координаты, 4 для скорости).

## Запись Pole\_addi

Для записи добавочного значения полюса `Pole_addi` выставите адрес 3. Запись возможна, если источник определён как CPU с помощью бита `PoleAddi_src`.

## Синхронизация двух CPU

CPU1 и CPU2 могут обмениваться данными и синхронизировать работу:

- **Обмен данными** — через ячейки 240–247 (кросс-подключение шин чтения).
- **Программная синхронизация** — `WAIT_EXT_CPU` ожидает импульс от другого CPU, `PULSE_EXT_CPU` генерирует импульс.
- **Синхронизация с преобразователем** — `WAIT_OWN_HAND` ожидает обновление данных от «родного» преобразователя, `WAIT_EXT_HAND` — от «внешнего».
- **Флаг STP** — `SETB_STP/CLRB_STP` управляют однобитным выходом CPU, может использоваться для внешней сигнализации.

## Отладчик

Каждый CPU имеет встроенный отладчик, управляемый через SPI-регистры `DBG1_ctrl/DBG1_data` и `DBG2_ctrl/DBG2_data`.

### Управление выполнением

Регистр `DBG_ctrl`:

| Биты    | Назначение                            |
|---------|---------------------------------------|
| [12:11] | Команда: 0 = RUN, 1 = STOP, 2 = STEP  |
| [10]    | Триггер обработки команды (по фронту) |
| [9]     | Разрешение точки останова 1           |
| [8:0]   | Адрес точки останова 1                |

Регистр `DBG_data`:

| Биты  | Назначение                  |
|-------|-----------------------------|
| [9]   | Разрешение точки останова 2 |
| [8:0] | Адрес точки останова 2      |

## Чтение регистров

Чтение внутренних регистров CPU производится через SPI-команду READ\_CPU\_REGS. Поле sel\_reg (биты [13:10]) выбирает данные:

| sel_reg | Данные                                |
|---------|---------------------------------------|
| 0-5     | R0-R5 (14 бит)                        |
| 6-7     | R6/R7 младшие 14 бит                  |
| 8-9     | R6/R7 старшие 14 бит                  |
| 10      | Статус: DBG_STOP, STP, PC, инструкция |
| 11      | Каналы: a_hand, ch1, ch4, ch0         |
| 12      | Каналы: ch7, ch_rel, ch2, ch3         |

## Пример программы

### CPU1: Коррекция угла по 1-й гармонике

Программа читает угол из преобразователя, вычисляет  $\sin(\text{угол})$  через CORDIC, умножает на float-коэффициент из памяти и вычитает полученную коррекцию из исходного угла. Результат передаётся CPU2 через ячейки обмена.

```
SET_A_HAND 0
```

```
// Выбрать FullAngle
```

```

LOOP: WAIT_OWN_HAND // Ждать обновление данных
LOAD32 HAND1_L R0 // R0 = текущий угол (28 бит,
адреса 252-253)

CLR R7 // R7 = 0 (аккумулятор коррекции)
SIN R0 // CORDIC: R0 = sin(угол)
LOAD32 228 R1 // R1 = коэффициент (float28,
адреса 228-229)
MULTF_ACC R1 R0 // R6 = R0(целое) × R1(float); R7
+= R6

ASR 15 R7 // Масштабирование: R7 >>= 16
MOVE R7 R2 // R2 = коррекция

LOAD32 HAND1_L R0 // R0 = исходный угол (пересчитать)
SUB R2 R0 // R0 = угол - коррекция

STORE32 EXT_CPU_0 R0 // Исправленный угол → CPU2 (адреса
240-241)
PULSE_EXT_CPU // Сигнал готовности CPU2
JUMP LOOP // Следующий цикл

```

**Используемые переменные** (из файла `cpu1_ram.txt`):

| Переменная  | Адрес   | Описание                          |
|-------------|---------|-----------------------------------|
| HAND1_L     | 252     | Младшие 14 бит OWN HAND           |
| HAND1_H     | 253     | Старшие 14 бит OWN HAND           |
| EXT_CPU_0   | 240     | Младшие 14 бит обмена CPU1 → CPU2 |
| EXT_CPU_1   | 241     | Старшие 14 бит обмена CPU1 → CPU2 |
| (адрес 228) | 228–229 | Float28 коэффициент коррекции     |

**Пошаговое описание:**

1. `SET_A_HAND 0` — выбирает пару ячеек FullAngle для чтения из преобразователя.

2. `WAIT_OWN_HAND` — останавливает CPU до появления новых данных от «родного» преобразователя.
3. `LOAD32 HAND1_L R0` — загружает 28-битный угол из ячеек 252 (младшие) и 253 (старшие).
4. `CLR R7` — обнуляет аккумулятор.
5. `SIN R0` — CORDIC вычисляет  $\sin(R0)$ , результат записывается в R0. CORDIC одновременно вычисляет  $\cos$ , результат доступен для следующей команды `COS`.
6. `LOAD32 228 R1` — загружает 28-битный float-коэффициент из ПЗУ.
7. `MULTF_ACC R1 R0` —  $R6 = R0$  (целое  $\sin$ )  $\times$   $R1$  (float коэф.);  $R7 += R6$ .
8. `ASR 15 R7` — арифметический сдвиг R7 вправо на 16 бит (масштабирование накопленной суммы).
9. `MOVE R7 R2` —  $R2 =$  коррекция.
10. `LOAD32 HAND1_L R0` — перечитывает исходный угол (данные в HAND защёлкнуты).
11. `SUB R2 R0` —  $R0 =$  исходный угол  $-$  коррекция.
12. `STORE32 EXT_CPU_0 R0` — записывает 28-битный результат в ячейки 240–241 для CPU2.
13. `PULSE_EXT_CPU` — генерирует импульс, по которому CPU2 узнаёт о готовности данных.

## Сводная таблица команд

| Имя | Код операции | Аргументы | Описание                              |
|-----|--------------|-----------|---------------------------------------|
| NOP | 000000000000 | [count]   | Нет операции; задержка count+1 тактов |
| CLR | 000000000001 | A         | $A = 0$                               |
| INC | 000000000010 | A         | $A = A + 1$                           |
| DEC | 000000000011 | A         | $A = A - 1$                           |
| ABS | 000000000100 | A         | $A =  A $                             |

| Имя     | Код операции | Аргументы | Описание  |
|---------|--------------|-----------|---|
| SIN     | 00000000101  | A         | $A = \sin(A)$ , CORDIC 16 итераций                    |
| COS     | 00000000110  | A         | $A = \cos(A)$ , CORDIC 16 итераций                    |
| ACC     | 00000000111  | A         | $R7 = R7 + A$   |
| ACC2    | 00000001     | B, A      | $R7 = R7 + A + B$                                     |
| ATAN    | 00000010     | B, A      | $A = \text{atan2}(A, B)$                              |
| ADD     | 00000011     | B, A      | $A = A + B$   |
| SUB     | 00000100     | B, A      | $A = A - B$   |
| ORL     | 00000101     | B, A      | $A = A   B$   |
| ANL     | 00000110     | B, A      | $A = A \& B$  |
| MOVE    | 00000111     | B, A      | $A = B$   |
| LSL     | 000010       | count, A  | $A = A \ll (\text{count} + 1)$                        |
| ASR     | 000011       | count, A  | $A = A \gg (\text{count} + 1)$                        |
| STORE   | 0001         | addr, A   | $\text{ram}[\text{addr}] = A[13:0]$                   |
| STORE32 | 0010         | addr, A   | $\text{ram}[\text{addr}..\text{addr}+1] = A[27:0]$    |
| LOAD    | 0011         | addr, A   | $A =$<br>знакорасширение( $\text{ram}[\text{addr}]$ ) |
| EQUAL   | 010          | rel, B, A | $\text{PC} += \text{rel}$ если $A == B$               |
| MORE    | 011          | rel, B, A | $\text{PC} += \text{rel}$ если $A > B$                |

| Имя       | Код операции | Аргументы | Описание                             |
|-----------|--------------|-----------|--------------------------------------|
| CLOAD     | 1000         | addr, A   | A = знакорасширение(ram[addr + rel]) |
| CSTORE    | 1001         | addr, A   | ram[addr + rel] = A[13:0]            |
| CSTORE32  | 1010         | addr, A   | ram[addr+rel..addr+rel+1] = A[27:0]  |
| CLOAD32   | 1011         | addr, A   | A = {ram[addr+rel+1], ram[addr+rel]} |
| LOAD32    | 1100         | addr, A   | A = {ram[addr+1], ram[addr]}         |
| CONST     | 1101         | val, A    | A = знакорасширение(val), -64...+63  |
| DJNZ      | 11100        | rel, A    | A--; PC += rel если A ≠ 0            |
| CDJNZ     | 111010       | rel, CH#  | CH#--; PC += rel если CH# ≠ 0        |
| EQUAL0    | 111011       | rel, A    | PC += rel если A == 0                |
| JUMP      | 111100       | rel       | PC += rel                            |
| MULTI     | 11110100     | B, A      | R6 = A × B, целое                    |
| MULTI_ACC | 11110101     | B, A      | R6 = A × B; R7 += R6, целое          |
| MULTF     | 11110110     | B, A      | R6 = A × B, A — целое, B — float     |
| MULTF_ACC | 11110111     | B, A      | R6 = A × B; R7 += R6, float          |
| MULTK24   | 11111011     | B, A      | A = (A × B)[51:22]                   |

| Имя          | Код операции   | Аргументы | Описание   |
|--------------|----------------|-----------|--|
| MULTK14      | 11111100       | B, A      | $A = (A \times B)[40:13]$                          |
| MULTCUBE_ACC | 11111101       | B, A      | $R6 = (ch1+1)^{(ch7+1)} \times (B+A)$ ; $R7 += R6$ |
| DEC2FLOAT    | 11111110000    | A         | $R6 = \text{float}(A)$                             |
| SET_A_HAND   | 11111110001    | const     | Выбор адреса HAND (0-7)                            |
| HFIX         | 11111110010    | A         | Установка HAND из $A[2:0]$                         |
| CFIX0        | 11111110100    | A         | $ch0 = A[8:0]$                                     |
| CFIX1        | 11111110101    | A         | $ch1 = A[7:0]$                                     |
| CFIX2        | 11111110110    | A         | $ch2 = A[6:0]$                                     |
| CFIX3        | 11111110111    | A         | $ch3 = A[5:0]$                                     |
| CFIX4        | 11111111000    | A         | $ch4 = A[4:0]$                                     |
| CFIX7        | 11111111001    | A         | $ch7 = A[1:0]$                                     |
| CLR_CH       | 11111111010    | CH#       | $CH\# = 0$ ; при $CH\#=5$ очистка всех             |
| CLOAD_CUBE   | 11111111011    | A         | $A = (ch1+1)^{(ch7+1)}$                            |
| RJUMP        | 11111111100    | A         | $PC = A$   |
| STOREPC      | 11111111101    | A         | $A = PC + 1$                                       |
| SETB_STP     | 11111111110000 |           | $STP = 1$  |
| CLRB_STP     | 11111111110001 |           | $STP = 0$  |

| Имя           | Код операции   | Аргументы | Описание                 |
|---------------|----------------|-----------|--------------------------|
| SW_HTE1       | 11111111110010 |           | Переключение HAND1       |
| SW_HTE2       | 11111111110011 |           | Переключение HAND2       |
| WAIT_EXT_CPU  | 11111111111000 |           | Ожидание импульса от CPU |
| PULSE_EXT_CPU | 11111111111001 |           | Импульс для CPU          |
| WAIT_OWN_HAND | 11111111111010 |           | Ожидание own HAND        |
| WAIT_EXT_HAND | 11111111111011 |           | Ожидание ext HAND        |
| LOADSTORE_ON  | 11111111111100 |           | Включить CLOAD/CSTORE    |
| LOADSTORE_OFF | 11111111111101 |           | Выключить CLOAD/CSTORE   |
| IDLE          | 11111111111111 |           | Остановка CPU            |

Последнее обновление **17 юн. 2026 г.** от **Ilya Slepchenkov**

# Регистры конфигурации

## Таблица адресов регистров

| +            | 2 x 0    | 2 x 1        | 2 x 2     | 2 x 3    | 2 x 4   | 2 x 5        |
|--------------|----------|--------------|-----------|----------|---------|--------------|
| 2<br>x<br>0  | C1KampS  | C1KampC      | C1KbiasS  | C1KbiasC | C1fbias | C1ExPhShft   |
| 2<br>x<br>8  | C1Amp_th | C1InputStngs | C1Lock_th | C1Zero   | C1Mask  | C1KonturStng |
| 2<br>x<br>16 | C1Coord  | C1CoordHB    | C1AdcS    | C1AdcC   | C1OutS  | C1VirtualS   |
| 2<br>x<br>24 | C1Vel    | C1VelHB      | C1PhiS    | C1PhiC   | C1OutC  | C1VirtualC   |
| 2<br>x<br>32 | C2KampS  | C2KampC      | C2KbiasS  | C2KbiasC | C2fbias | C2ExPhShft   |
| 2<br>x<br>40 | C2Amp_th | C2InputStngs | C2Lock_th | C2Zero   | C2Mask  | C2KonturStng |
| 2<br>x<br>48 | C2Coord  | C2CoordHB    | C2AdcS    | C2AdcC   | C2OutS  | C2VirtualS   |

| +            | 2 x 0      | 2 x 1      | 2 x 2     | 2 x 3       | 2 x 4     | 2 x 5      |
|--------------|------------|------------|-----------|-------------|-----------|------------|
| 2<br>x<br>56 | C2Vel      | C2VelHB    | C2PhiS    | C2PhiC      | C2OutC    | C2VirtualC |
| 2<br>x<br>64 | IC_addr    | ADC_config | Mask_Stat | Flags_delay | WR_lock   | CMP_lth    |
| 2<br>x<br>72 | NOCLK_stat | SPI_req    | alive_cnt | Stat_main   | Dcpu1LB   | Dcpu1HB    |
| 2<br>x<br>80 | PLL_config | INIT_conf  | UOTP_ctrl | BUS_addr    | BOTP_addr | BOTP_data  |
| 2<br>x<br>88 | -          | -          | -         | -           | P1BG_ctrl | P1BG_data  |
| 2<br>x<br>96 | -          | -          | -         | -           | -         | -          |

## C1KampS/C2KampS

Адрес: 0/32

**Описание:** KampS [15:0] – коэффициент усиления по каналу АЦП *IOSA1* (преобразователь 1), *IOSA2* (преобразователь 2). Беззнаковое значение, всегда положительное. Амплитуда сигналов на этих входах микросхемы умножается на значение из данного регистра и делится на 1024. Значение по умолчанию соответствует амплитуде сигнала на входе равному входному диапазону АЦП (0 ÷ 2,5 В) для режима СКВТ. При изменении значений в

данном регистре необходимо следить за флагами переполнения, а также за срабатыванием компараторов порогов.

**Тип доступа:** Чтение/Запись (R/W)

---

## C1KampC/C2KampC

Адрес: 1/33

**Описание:** KampC [15:0] – коэффициент усиления по каналу АЦП [IOCA1](#) (преобразователь 1), [IOCA2](#) (преобразователь 2). Беззнаковое значение, всегда положительное. Амплитуда сигналов на этих входах микросхемы умножается на значение из данного регистра и делится на 1024. Значение по умолчанию соответствует амплитуде сигнала на входе равному входному диапазону АЦП ( $0 \div 2,5$  В) для режима СКВТ. При изменении значений в данном регистре необходимо следить за флагами переполнения, а также срабатыванием компараторов порогов. В режиме Sensor\_mode=01 максимальное значение 1024.

**Тип доступа:** Чтение/Запись (R/W)

---

## C1KbiasS/C2KbiasS

Адрес: 2/34

**Описание:** KbiasS [15:0] – смещение нуля по каналу АЦП [IOSA1](#) (преобразователь 1), [IOSA2](#) (преобразователь 2). Знаковое значение в дополнительном коде. Максимум +32767, минимум -32767.

**Тип доступа:** Чтение/Запись (R/W)

---

## C1KbiasC/C2KbiasC

Адрес: 3/35

**Описание:** KbiasC [15:0] – смещение нуля по каналу АЦП [IOCA1](#) (преобразователь 1), [IOCA2](#) (преобразователь 2). Знаковое значение в дополнительном коде. Максимум +32767, минимум -32767.

**Тип доступа:** Чтение/Запись (R/W)

---

## C1fbias/C2fbias

Адрес: 4/36

**Описание:** fbias[15:0] – коррекция неортогональности обмоток СКВТ. Смещение фазы обмотки sin. Знаковое значение в дополнительном коде.

**Тип доступа:** Чтение/Запись (R/W)

---

## C1ExPhShft/C2ExPhShft

Адрес: 5/37

**Описание:** ExPhShft[15:0] – задает сдвиг по фазе сигнала с EXO1 (преобразователь 1), EXO2 (преобразователь 2) на плате до входов IOSA1, IOCA1 (преобразователь 1), IOSA2, IOCA2 (преобразователь 2). Используется для определения квадранта положения СКВТ. Знаковое значение в дополнительном коде.

**Тип доступа:** Чтение/Запись (R/W)

---

## C1ExoStngs/C2ExoStngs

Адрес: 6/38

**Описание:** Настройки Amp\_code

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты   | Имя поля | Доступ | Сброс | Описание   |
|---|--------|----------|--------|-------|--|
| 1 | 11..10 | EXO_mode | R/W    | 0     | EXO_mode - режим формирования опоры возбуждения датчика 00 - |

| № | Биты | Имя поля | Доступ | Сброс | Описание   |
|---|------|----------|--------|-------|--|
|   |      |          |        |       | выключено. При EXO_inv=0 на EXO1(2) выводится 0 В, при EXO_inv=1 на EXO1(2) выводится VDDEXO1(2) 01 - вывод меандра 10 - вывод синусоидального напряжения с помощью ЦАП 11 - вывод постоянного значения C2EXInc/C2EXInc По умолчанию EXO_mode = 00 |
| 2 | 9    | PWR_X2   | R/W    | 0     | Добавление усилительного каскада при формировании меандра на EXO1(2) для увеличения выходного тока: 0 - без увеличения 1 - с увеличением По умолчанию 0.   |
| 3 | 8    | EXO_inv  | R/W    | 0     | Инверсия выхода EXO1(2) в состоянии выключено, или при выводе меандра: 0 - без инверсии 1 - с инверсией По умолчанию 0.  |
| 4 | 7..0 | Amp_code | R/W    | 0     | Коэффициент усиления выходного напряжения ЦАП  |

## C1EXInc/C2EXInc

Адрес: 7/39

**Описание:** EXInc[15:0] задает приращение фазы синусоидального сигнала на каждом такте ЦАП (Fclk). Значение рассчитывается по формуле (5). Значение по умолчанию соответствует частоте 12 кГц.

**Тип доступа:** Чтение/Запись (R/W)

# C1Amp\_th/C2Amp\_th

Адрес: 8/40

**Описание:** Порог компаратора для вычисления флагов UIN в регистре C1Stat (преобразователь 1), C2Stat (преобразователь 2).

**Тип доступа:** Чтение/Запись (R/W)

## Поля регистра

| № | Биты  | Имя поля       | Доступ | Сброс | Описание  |
|---|-------|----------------|--------|-------|---|
| 1 | 15..8 | <b>Amp_hth</b> | R/W    | 187   | Порог компаратора для вычисления флага UIN_HIGH в регистре C1Stat (преобразователь 1), C2Stat (преобразователь 2). Amp_hth безразмерная относительная величина. Рекомендуется подобрать значение этой величины такой, чтобы при нормальной работе преобразователя не возникало срабатывание компаратора. Текущее значение величины можно считать из регистров C1Amp_metric, C2Amp_metric. |
| 2 | 7..0  | <b>Amp_lth</b> | R/W    | 62    | Порог компаратора для вычисления флага UIN_LOW в регистре C1Stat (преобразователь 1), C2Stat (преобразователь 2). Amp_lth безразмерная относительная величина. Рекомендуется подобрать значение этой величины такой, чтобы при нормальной работе преобразователя не возникало срабатывание компаратора. Текущее значение величины можно считать из регистров C1Amp_metric, C2Amp_metric   |

# C1InputStngs/C2InputStngs

Адрес: 9/41

**Описание:** Настройки входных групп: источник опорного сигнала, инверсия, фильтрация, калибровка АЦП, компенсация смещения

**Тип доступа:** Чтение/Запись (R/W)

## Поля регистра

| № | Биты   | Имя поля              | Доступ | Сброс | Описание  |
|---|--------|-----------------------|--------|-------|---|
| 1 | 14..13 | <b>EXI_insentv</b>    | R/W    | 0     | Глубина фильтра дребезга переключения сигнала EXI. 0 – без фильтра 1 – 2 цикла 2 – 4 цикла 3 – 8 циклов   |
| 2 | 12     | <b>EXI_inv</b>        | R/W    | 0     | Инверсия входа <b>EXI1</b> : 0 – вход EXI поступает в схему без инверсии 1 – вход EXI инвертируется   |
| 3 | 11     | <b>Ex_recovery_en</b> | R/W    | 1     | Включение блока восстановления сигнала опорной частоты: 0 – блок восстановления сигнала опорной частоты отключен 1 – блок восстановления сигнала опорной частоты включен  |
| 4 | 10..9  | <b>Ex_source</b>      | R/W    | 0     | Биты выбора источника сигнала Ex_ref для блока восстановления сигнала опорной частоты. 3 – Ex_ref преобразователя 2 дублирует Ex_recovered от преобразователя 1 (невозможно установить для преобразователя 1); 2 – Ex_ref преобразователя 2 |

| № | Биты | Имя поля       | Доступ | Сброс | Описание   |
|---|------|----------------|--------|-------|--|
|   |      |                |        |       | дублирует Ex_ref от преобразователя 1 (невозможно установить для преобразователя 1); 1 – с входа EXI1 для преобразователь 1 и для преобразователя 2; Примечание – Подача с EXI2 опорного сигнала не предусмотрена. Если включен режим сельсин, то используется вход EXI1 для обоих преобразователей. Если включен один из сдвоенных режимов, используется EXI1. 0 – с генератора опорной частоты |
| 5 | 8    | <b>OPA_en</b>  | R/W    | 0     | Включение входных операционных усилителей преобразователя: 0 – входные усилители отключены. Сигналы на АЦП подаются со входов IOSA, IOCA 1 – входные усилители включены. Сигналы на АЦП подаются с усилителей  |
| 6 | 7    | <b>ADC_CAL</b> | R/W    | 0     | Калибровка АЦП: 0 – обычный режим работы 1 – режим калибровки смещения АЦП В режиме калибровки смещения ОУ и АЦП отключаются от выводов микросхемы, и на вход ОУ подается опорное напряжение 1,25 В. Коды, получаемые с АЦП, усредняются для вычисления смещения АЦП, пока бит установлен в 1. После установки   |

| № | Биты | Имя поля              | Доступ | Сброс | Описание   |
|---|------|-----------------------|--------|-------|--|
|   |      |                       |        |       | бита в 0, полученное значение смещения вычитается из кодов, получаемых с АЦП. Калибровка АЦП прямо влияет на погрешность микросхемы в режимах с немодулированным сигналом датчика. В режимах с модулированным сигналом датчика калибровка не обязательна. Чем больше время калибровки, тем более точная будет калибровка. Рекомендуемое время калибровки ~350 мс или больше. |
| 7 | 6    | <b>Autooff_adccal</b> | R/W    | 0     | Отключение режима калибровки АЦП после первого расчета корректирующих коэффициентов: 0 – режим калибровки не отключается 1 – автоотключение режима калибровки при DC_carrier = 1 после установки флага HW_NotRDY = 0   |
| 8 | 5    | <b>DC_carrier</b>     | R/W    | 0     | Входные сигналы без модуляции: 0 – входные сигналы модулированы 1 – входные сигналы без модуляции  |
| 9 | 4    | <b>DC_correction</b>  | R/W    | 1     | Включение компенсации среднего уровня сигнала для АЦП: 0 – компенсация отключена 1 – компенсация включена При подаче на вход немодулированных сигналов этот  |

| №  | Биты | Имя поля        | Доступ | Сброс | Описание   |
|----|------|-----------------|--------|-------|--|
|    |      |                 |        |       | бит должен быть установлен в состояние лог. «0»  |
| 10 | 3..0 | <b>DC_depth</b> | R/W    | 0     | Глубина коррекции смещения. Задается значение в диапазоне от 0 до 15. Расчет первых коэффициентов смещения производится после 4 циклов работы блока коррекции смещения Tdc_dep. Период одного цикла Tdc_dep определяется следующими зависимостями: (Tclk - период работы контура, для примера взята 1 мкс) 0 - 1 128Tclk ~ 128 мкс 1 - 2 256Tclk ~ 256 мкс 2 - 4 512*Tclk ~ 512 мкс ... Значение по умолчанию 0. |

## C1Lock\_th/C2Lock\_th

Адрес: 10/42

**Описание:** Порог компаратора для вычисления флага NLock в регистре C1Stat (преобразователь 1), C2Stat (преобразователь 2). Lock\_th безразмерная относительная величина. Рекомендуется подобрать значение этой величины такой, чтобы при нормальной работе преобразователя не возникало срабатывание компаратора.

**Тип доступа:** Чтение/Запись (R/W)

## C1Zero/C2Zero

Адрес: 11/43

**Описание:** Zero[15:0] – коррекция вычисленной координаты. Значение Zero прибавляется к вычисленной преобразователем координате. При коррекции угол представлен 16-битным значением, вне зависимости от настроек, заданных в C1ResCntrl или C2ResCntrl.

**Тип доступа:** Чтение/Запись (R/W)

## C1Mask/C2Mask

Адрес: 12/44

**Описание:** Регистр маски Mask Значение, записанное в регистр маски Mask, включает работу соответствующих бит регистра C1Stat.

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты   | Имя поля            | Доступ | Сброс | Описание  |
|---|--------|---------------------|--------|-------|---|
| 1 | 15     | <b>MSK_NLock</b>    | R/W    | 1     | Маска для бита NLock (контур в неустановившемся режиме): 0 – бит NLock устанавливается в состояние лог. «0» 1 – бит NLock отражает действительное состояние флага |
| 2 | 14..13 | <b>MSK_quadrant</b> | R/W    | 0     | Маска для бита quadrant (квадрант): 0 – биты quadrant[1:0] устанавливаются в состояние лог. «0» 1 – биты quadrant[1:0] отражают действительное состояние флагов   |

| № | Биты | Имя поля                 | Доступ | Сброс | Описание   |
|---|------|--------------------------|--------|-------|--|
| 3 | 12   | <b>MSK_Kontur_NotENA</b> | R/W    | 1     | Флаг запуска контура сигнализирует, что контур запустился в соответствии с настройками. 0 - бит Kontur_NotENA устанавливается в состояние лог. «0» 1 - бит Kontur_NotENA функционирует |
| 4 | 11   | <b>MSK_RCV_NotRDY</b>    | R/W    | 1     | Маска для бита RCV_NotRDY: 0 - бит RCV_NotRDY устанавливается в состояние лог. «0» 1 - бит RCV_NotRDY функционирует  |
| 5 | 10   | <b>MSK_MISS_EXREF</b>    | R/W    | 1     | Маска для бита MISS_EXREF: 0 - бит MISS_EXREF устанавливается в состояние лог. «0» 1 - бит MISS_EXREF функционирует  |
| 6 | 9    | <b>MSK_EX_PH_OUTR</b>    | R/W    | 1     | Маска для битов EX_PH_HIGH, MISS_EXREF: 0 – биты EX_PH_HIGH, MISS_EXREF устанавливаются в состояние лог. «0» 1 – биты EX_PH_HIGH, MISS_EXREF отражают действительное состояние флагов  |
| 7 | 7    | <b>MSK_C_LOOP_OVF</b>    | R/W    | 1     | Маска для бита C_LOOP_OVF (переполнение в следящем контуре): 0 – бит C_LOOP_OVF устанавливается в состояние  |

| №  | Биты | Имя поля            | Доступ | Сброс | Описание   |
|----|------|---------------------|--------|-------|--|
|    |      |                     |        |       | лог. «0» 1 – бит C_LOOP_OVF отражает действительное состояние флага  |
| 8  | 6    | <b>MSK_UIN_HIGH</b> | R/W    | 1     | Маска для бита UIN_HIGH (амплитуда сигналов слишком велика): 0 – бит UIN_HIGH устанавливается в состояние лог. «0» 1 – бит UIN_HIGH отражает действительное состояние флага  |
| 9  | 5    | <b>MSK_UIN_LOW</b>  | R/W    | 1     | Маска для бита UIN_LOW (амплитуда сигналов слишком мала): 0 – бит UIN_LOW устанавливается в состояние лог. «0» 1 – бит UIN_LOW отражает действительное состояние флага   |
| 10 | 4    | <b>MSK_CORR_OVF</b> | R/W    | 1     | Переполнение после коррекции амплитуды сигналов. Маска для бита CORR_OVF (переполнение после коррекции амплитуды сигналов). 0 – бит CORR_OVF устанавливается в состояние лог. «0» 1 – бит CORR_OVF отражает действительное состояние флага |
| 11 | 3    | <b>MSK_ADC_OVF</b>  | R/W    | 1     | Маска для бита ADC_OVF (переполнение из-за   |

| №  | Биты | Имя поля             | Доступ | Сброс | Описание  |
|----|------|----------------------|--------|-------|---|
|    |      |                      |        |       | большой постоянной составляющей сигналов на входе АЦП): 0 – бит ADC_OVF устанавливается в состояние лог. «0» 1 – бит ADC_OVF отражает действительное состояние флага      |
| 12 | 2    | <b>MSK_CLIP_COS</b>  | R/W    | 1     | Маска для бита CLIP_COS (переполнение АЦП по каналу cos): 0 – бит CLIP_COS устанавливается в состояние лог. «0» 1 – бит CLIP_COS отражает действительное состояние флага  |
| 13 | 1    | <b>MSK_CLIP_SIN</b>  | R/W    | 1     | Маска для бита CLIP_SIN (переполнение АЦП по каналу sin): 0 – бит CLIP_SIN устанавливается в состояние лог. «0» 1 – бит CLIP_SIN отражает действительное состояние флага  |
| 14 | 0    | <b>MSK_HW_NotRDY</b> | R/W    | 1     | Маска для бита HW_NotRDY (коррекция АЦП не произведена): 0 – бит HW_NotRDY устанавливается в состояние лог. «0» 1 – бит HW_NotRDY отражает действительное состояние флага |

Адрес: 13/45

**Описание:** Регистры режимов работы преобразователей

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты   | Имя поля            | Доступ | Сброс | Описание  |
|---|--------|---------------------|--------|-------|---|
| 1 | 15     | <b>Wait_1offset</b> | R/W    | 0     | Режим начала работы блока восстановления частоты при DC_carrier=0, или контура при DC_carrier=1: 0 – включать работу блоков до расчета первых коэффициентов 1 – удерживать блоки в сбросе пока не рассчитаются первые коэффициенты смещений |
| 2 | 14     | <b>Vel_hist</b>     | R/W    | 0     | Гистерезис значений координаты: 0 – гистерезис отключен 1 – гистерезис включен  |
| 3 | 13     | <b>Coord_hist</b>   | R/W    | 0     | Гистерезис значений координаты: 0 – гистерезис отключен 1 – гистерезис включен  |
| 4 | 12     | <b>HandToEXT</b>    | R/W    | 0     | Адресная шина мультиплексора подключается не к своему CPU: 0 - HAND подключается к OWN CPU 1 - HAND подключается к EXT CPU  |
| 5 | 11..10 | <b>Sensor_mode</b>  | R/W    | 0     | 00 – режим СКВТ или сельсин. В модели датчика используется тригонометрическая функция (Рисунок #).01 – режим ЛРДТ с подключением по 5-ти проводной  |

| № | Биты | Имя поля            | Доступ | Сброс | Описание  |
|---|------|---------------------|--------|-------|---|
|   |      |                     |        |       | схеме. В модели датчика используется линейная функция. Модель датчика согласно рисунку #. 10 – режим ЛРДТ с подключением по схеме с последовательным соединением обмоток. В модели датчика используется линейная функция. Модель датчика согласно рисунку #.  |
| 6 | 9    | <b>En_cross0</b>    | R/W    | 0     | Режим виртуального грубого отсчета: 0 - режим выключен 1 - режим включен. В зависимости от направления вращения датчика при переходе через "0" координаты, производится инкремент или декремент виртуального счетчика. Значение счетчика может быть перезаписано из CPU1/CPU2 для согласования отсчетов для датчиков с редукцией грубого и точного каналов. |
| 7 | 8    | <b>PoleAddi_src</b> | R/W    | 0     | Источник добавки к виртуальному счетчику полюсов: 0 - SPI 1 - CPU   |
| 8 | 7..5 | <b>InDelay</b>      | R/W    | 0     | InDelay [2:0] – компенсация задержки входного тракта микросхемы и фильтров на плате. Задается в тактах частоты Fclk. Беззнаковое положительное значение.  |
| 9 | 4..0 | <b>LBW</b>          | R/W    | 4     | Настройка полосы пропускания следящего контура ((см. таблицу 9))  |

# C1ResCntrl/C2ResCntrl

Адрес: 14/46

**Описание:** Регистр настройки выходной информации преобразователя

**Тип доступа:** Чтение/Запись (R/W)

## Поля регистра

| № | Биты   | Имя поля          | Доступ | Сброс | Описание  |
|---|--------|-------------------|--------|-------|---|
| 1 | 15     | <b>Enc_en</b>     | R/W    | 0     | Включение блока эмуляции квадратурного энкодера: 0 – блок эмуляции квадратурного энкодера отключен 1 – блок эмуляции квадратурного энкодера включен<br>Coord_resolution должен быть равным 2 (16 битный код)<br>Значение Vel_resolution равно количеству младших битов, которые будут замаскированы<br>Например, при Vel_resolution=8, энкодер будет работать 8 битным кодом при Vel_resolution=12, энкодер работает с 4 битным кодом |
| 2 | 14..12 | <b>Enc_presc</b>  | R/W    | 0     | Делитель частоты для сигналов энкодера: 000 – FINT/2 001 – FINT/3 010 – FINT/4 011 – FINT/5 100 – FINT/8 101 – FINT/16 110 – FINT/32 111 – FINT/64  |
| 3 | 11     | <b>SPI_ext_en</b> | R/W    | 0     | Разрешает режим прямой параллельной передачи  |

| № | Биты | Имя поля                | Доступ | Сброс | Описание  |
|---|------|-------------------------|--------|-------|---|
|   |      |                         |        |       | результата: 0 – режим отключен<br>1 – режим включен При ENC_en == 1 состояние данного бита не имеет значения  |
| 4 | 9    | <b>Vel_from_cpu</b>     | R/W    | 0     | Значения скорости режима прямой/параллельной передачи переопределить значениями из CPU (соответствующему данному каналу) 0 - использовать значения из контура 1 - переопределить скорость значениями из CPU     |
| 5 | 8..5 | <b>Vel_resolution</b>   | R/W    | 7     | Устанавливает разрешение в регистре C1Vel (преобразователь 1), C2Vel (преобразователь 2). Значение из таблицы 9   |
| 6 | 4    | <b>Coord_from_cpu</b>   | R/W    | 0     | Значения энкодера, режима прямой/параллельной передачи переопределить значениями из CPU (соответствующего данному каналу) 1 - переопределить координату значениями из CPU 0 - использовать значения из контура  |
| 7 | 3..0 | <b>Coord_resolution</b> | R/W    | 2     | Устанавливает разрешение в регистре C1Coord (преобразователь 1), C2Coord (преобразователь 2). Значение может быть вычислено по формуле: Количество бит = $18 \text{ shl } \text{Coord\_resolution}[3:0]$ или из |

| № | Биты | Имя поля | Доступ | Сброс | Описание  |
|---|------|----------|--------|-------|---|
|   |      |          |        |       | таблицы 9. При получении данных с выхода эмуляции квадратурного энкодера устанавливать эти биты в значение #. |

---

## C1Vcnt\_bound/C2Vcnt\_bound

Адрес: 15/47

**Описание:** Порог переполнения виртуального счетчика в старших разрядах [15:0]

**Тип доступа:** Чтение/Запись (R/W)

---

## C1Coord/C2Coord

Адрес: 16/48

**Описание:** Coord – координата, вычисленная в преобразователе. Разрядность зависит от настроек в регистрах C1ResCntrl и C2ResCntrl.

**Тип доступа:** Только чтение (RO)

---

## C1CoordHB/C2CoordHB

Адрес: 17/49

**Описание:** CoordHB[11..0] - старшие 12 разрядов координаты

**Тип доступа:** Только чтение (RO)

---

## C1AdcS/C2AdcS

Адрес: 18/50

**Описание:** Выход каналов АЦП и опорных сигналов

**Тип доступа:** Только чтение (RO)

### Поля регистра

| № | Биты  | Имя поля           | Доступ | Сброс | Описание   |
|---|-------|--------------------|--------|-------|--|
| 1 | 15    | <b>msb_arg_cos</b> | RO     | 0     | Старший разряд данных поступающих на контур канала COS |
| 2 | 14    | <b>Ex_shifted</b>  | RO     | 0     | Опорный сигнал EX_REF после сдвига на ExPhShift        |
| 3 | 13..2 | <b>Dadc_sin</b>    | RO     | 0     | Выход АЦП канал SIN                                    |
| 4 | 1     | <b>msb_arg_sin</b> | RO     | 0     | Старший разряд данных поступающих на контур канала SIN |
| 5 | 0     | <b>Ex_ref</b>      | RO     | 0     | Опорный сигнал EX_REF                                  |

## C1AdcC/C2AdcC

Адрес: 19/51

**Описание:** Выход каналов АЦП и опорных сигналов

**Тип доступа:** Только чтение (RO)

### Поля регистра

| № | Биты   | Имя поля        | Доступ | Сброс | Описание   |
|---|--------|-----------------|--------|-------|--|
| 1 | 15..12 | <b>crc4_adc</b> | RO     | 0     | Контрольная сумма по [AdcC[13:0], AdcS[15:0]], включается битом HAND_CRC4_en |
| 2 | 11..0  | <b>Dadc_cos</b> | RO     | 0     | Выход АЦП канал COS  |

## C1OutS/C2OutS

Адрес: 20/52

**Описание:** Код канала АЦП SIN после коррекции смещения и амплитуды. Является входным аргументом в контур.

**Тип доступа:** Только чтение (RO)

### Поля регистра

| № | Биты  | Имя поля               | Доступ | Сброс | Описание  |
|---|-------|------------------------|--------|-------|---|
| 1 | 15    | <b>VirtualSin0</b>     | RO     | 0     | Младший бит виртуального сигнала датчика канала синус |
| 2 | 14    | <b>Ex_recovered_vs</b> | RO     | 0     | Восстановленный сигнал опорной частоты                |
| 3 | 13..1 | <b>arg_sin_kontur1</b> | RO     | 0     | Данные 13-бит поступающие на контур канала SIN        |

| № | Биты | Имя поля             | Доступ | Сброс | Описание   |
|---|------|----------------------|--------|-------|--|
| 4 | 0    | Ex_recovered90dgr_vs | RO     | 0     | Восстановленный сигнал опорной частоты, сдвинутый на 90 градусов |

## C1VirtualS/C2VirtualS

Адрес: 21/53

**Описание:** Виртуальные значения Sin участвующие в свертке

**Тип доступа:** Только чтение (RO)

### Поля регистра

| № | Биты   | Имя поля       | Доступ | Сброс | Описание   |
|---|--------|----------------|--------|-------|--|
| 1 | 15..12 | crc4_vs        | RO     | 0     | Контрольная сумма по битам [VirtualS[13:0], OutS[15:0]], включается битом HAND_CRC4_en |
| 2 | 11..0  | VirtualSin12_1 | RO     | 0     | Старшие 12 бит виртуального сигнала датчика канала синус                               |

## C1Err\_metric/C2Err\_metric

Адрес: 22/54

**Описание:** Метрика ошибки угла на выходе микросхемы. По модулю этой метрики срабатывает компаратор флага NLock.

**Тип доступа:** Только чтение (RO)

## C1Amp\_metric/C2Amp\_metric

Адрес: 23/55

**Описание:** Метрика амплитуды сигнала на входе микросхемы. По этой метрике срабатывают компараторы флагов UIN\_High, UIN\_Low. Номинальное значение 400.

**Тип доступа:** Только чтение (RO)

---

## C1Vel/C2Vel

Адрес: 24/56

**Описание:** Vel – скорость, вычисленная в преобразователе. Разрядность зависит от настроек в регистрах C1ResCntrl и C2ResCntrl.

**Тип доступа:** Только чтение (RO)

---

## C1VelHB/C2VelHB

Адрес: 25/57

**Описание:** Старшие разряды скорости Vel

**Тип доступа:** Только чтение (RO)

---

## C1PhiS/C2PhiS

Адрес: 26/58

**Описание:** Выход контура с учетом модели датчика, коэффициентов InDelay, KbiasS, Fbias

**Тип доступа:** Только чтение (RO)

**Поля регистра**

| № | Биты   | Имя поля               | Доступ | Сброс | Описание  |
|---|--------|------------------------|--------|-------|---|
| 1 | 15..14 | <b>phimodel_cos1_0</b> | RO     | 0     | Младшие 2 бит виртуального сигнала датчика канала COS |
| 2 | 13..0  | <b>phimodel_sin</b>    | RO     | 0     | 14 бит виртуального сигнала датчика канала SIN        |

## C1PhiC/C2PhiC

Адрес: 27/59

**Описание:** Выход контура с учетом модели датчика, коэффициентов InDelay, KbiasC, Fbias

**Тип доступа:** Только чтение (RO)

### Поля регистра

| № | Биты   | Имя поля                | Доступ | Сброс | Описание   |
|---|--------|-------------------------|--------|-------|--|
| 1 | 15..12 | <b>crc4_ph</b>          | RO     | 0     | Контрольная сумма по битам [PhiC[13:0], PhiS[15:0]], включается битом HAND_CRC4_en |
| 2 | 11..0  | <b>phimodel_cos13_2</b> | RO     | 0     | Старшие 12 бит виртуального сигнала датчика канала COS                             |

## C1OutC/C2OutC

Адрес: 28/60

**Описание:** Код канала АЦП COS после коррекции смещения и амплитуды. Является входным аргументом в контур.

**Тип доступа:** Только чтение (RO)

### Поля регистра

| № | Биты  | Имя поля                    | Доступ | Сброс | Описание   |
|---|-------|-----------------------------|--------|-------|--|
| 1 | 15    | <b>VirtualCos0</b>          | RO     | 0     | Младший бит виртуального сигнала датчика канала COS              |
| 2 | 14    | <b>Ex_recovered_vc</b>      | RO     | 0     | Восстановленный сигнал опорной частоты                           |
| 3 | 13..1 | <b>arg_cos_kontur1</b>      | RO     | 0     | Данные 13-бит поступающие на контур канала COS                   |
| 4 | 0     | <b>Ex_recovered90dgr_vc</b> | RO     | 0     | Восстановленный сигнал опорной частоты, сдвинутый на 90 градусов |

## C1VirtualC/C2VirtualC

Адрес: 29/61

**Описание:** Виртуальные значения Cos для вычисления ошибки в контуре

**Тип доступа:** Только чтение (RO)

### Поля регистра

| № | Биты   | Имя поля       | Доступ | Сброс | Описание   |
|---|--------|----------------|--------|-------|--|
| 1 | 15..12 | <b>crc4_vc</b> | RO     | 0     | Контрольная сумма по битам [VirtualS[13:0], OutS[15:0]], включается битом HAND_CRC4_en |

| № | Биты  | Имя поля              | Доступ | Сброс | Описание   |
|---|-------|-----------------------|--------|-------|--|
| 2 | 11..0 | <b>VirtualCos12_1</b> | RO     | 0     | Старшие 12 бит виртуального сигнала датчика канала COS |

## C1Stat/C2Stat

Адрес: 30/62

**Описание:** Stat - регистр ошибок/состояния канала преобразователя.

**Тип доступа:** Только чтение (RO)

### Поля регистра

| № | Биты   | Имя поля        | Доступ | Сброс | Описание  |
|---|--------|-----------------|--------|-------|---|
| 1 | 15     | <b>NLock</b>    | RO     | 1     | Контур в неустановившемся режиме: 0 – ошибка в следящем контуре меньше чем значение в регистре C1Lock_th (преобразователь 1), C2Lock_th (преобразователь 2). 1 – ошибка в следящем контуре больше чем значение в регистре C1Lock_th (преобразователь 1), C2Lock_th (преобразователь 2). |
| 2 | 14..13 | <b>quadrant</b> | RO     | 0     | Квадрант. Старшие два бита результата (без учета сдвига в блоке обработки результатов). При записи в Coord_resolution 1 или 0 позволяет получить старший бит или биты   |

| № | Биты | Имя поля              | Доступ | Сброс | Описание   |
|---|------|-----------------------|--------|-------|--|
|   |      |                       |        |       | результата. Биты не влияют на состояние бит Ready  |
| 3 | 11   | <b>Kontur_NotENA</b>  | RO     | 0     | Флаг запуска контура сигнализирует, что контур запустился в соответствии с настройками: 0 - контур запущен 1 - контур в ожидании наступления заданных условий  |
| 4 | 10   | <b>RCV_NotRDY</b>     | RO     | 0     | Флаг готовности блока восстановления опорной частоты: пока RCV_NotRDY=1, контур удерживается в сбросе: 0 - опорная частота восстановлена 1 - опорная частота не восстановлена  |
| 5 | 9    | <b>MISS_EXREF</b>     | RO     | 0     | Отсутствие опорного сигнала: 0 - EX_REF переключается чаще 10 Гц 1 - за секунду изменений логического уровня EX_REF не обнаружено  |
| 6 | 8    | <b>EX_PH_OUTRANGE</b> | RO     | 0     | Большой сдвиг фазы: 0 – сдвиг фазы между опорным и восстановленным сигналом меньше $\pm 40^\circ$ 1 – сдвиг фазы между опорным и восстановленным сигналом больше $\pm 40^\circ$ , что может приводить к избыточному шуму на выходе микросхемы. |

| № | Биты | Имя поля          | Доступ | Сброс | Описание  |
|---|------|-------------------|--------|-------|---|
| 7 | 7    | <b>C_LOOP_OVF</b> | RO     | 0     | <p>Переполнение в следящем контуре: 0 – нет ошибки в следящем контуре 1 – ошибка в следящем контуре. Возможна некорректная работа микросхемы. В режимах Sensor_mode==0 установка данного бита может быть вариантом нормальной работы преобразователя, указывающим на то, что координата датчика достигла максимального/минимального значения.</p> |
| 8 | 6    | <b>UIN_HIGH</b>   | RO     | 0     | <p>Амплитуда сигналов слишком велика: 0 – амплитуда сигналов на входе следящего контура меньше заданного порога 1 – большая амплитуда сигналов на входе следящего контура. Определяется сравнением с порогом заданном в регистре Amp_hth (преобразователь 1), Amp_hth (преобразователь 2).</p>  |
| 9 | 5    | <b>UIN_LOW</b>    | RO     | 0     | <p>Амплитуда сигналов слишком мала: 0 – амплитуда сигналов на входе следящего контура больше заданного порога 1 – малая амплитуда сигналов на входе следящего контура. Определяется сравнением с порогом заданном в регистре</p>  |

| №  | Биты | Имя поля        | Доступ | Сброс | Описание  |
|----|------|-----------------|--------|-------|---|
|    |      |                 |        |       | C1Amp_lth (преобразователь 1),<br>C2Amp_lth (преобразователь 2).  |
| 10 | 4    | <b>CORR_OVF</b> | RO     | 0     | Переполнение после коррекции амплитуды сигналов: 0 – переполнение после коррекции усиления отсутствует 1 – слишком большой сигнал после коррекции усиления. Ошибка вызвана слишком большими коэффициентами C1KampS, C1KampC или C2KampS, C2KampC. Флаг сбрасывается если в течение 2,5 мс не было переполнений. |
| 11 | 3    | <b>ADC_OVF</b>  | RO     | 0     | Переполнение из-за большой постоянной составляющей сигналов на входе АЦП: 0 – переполнение из-за большой постоянной составляющей отсутствует 1 – постоянная составляющая сигнала(ов) вне диапазона. Флаг сбрасывается если в течение 18 с не было выхода за диапазон.   |
| 12 | 2    | <b>CLIP_COS</b> | RO     | 0     | Переполнение АЦП по каналу cos: 0 – амплитуда сигнала в канале cos не выходит за диапазон АЦП 1 – амплитуда сигнала в канале cos выходит за диапазон АЦП. Флаг сбрасывается если в течение  |

| №  | Биты | Имя поля         | Доступ | Сброс | Описание  |
|----|------|------------------|--------|-------|---|
|    |      |                  |        |       | 2,5 мс не было выхода за диапазон.  |
| 13 | 1    | <b>CLIP_SIN</b>  | RO     | 0     | Переполнение АЦП по каналу sin: 0 – амплитуда сигнала в канале sin не выходит за диапазон АЦП 1 – амплитуда сигнала в канале sin выходит за диапазон АЦП. Флаг сбрасывается если в течение 2,5 мс не было выхода за диапазон.   |
| 14 | 0    | <b>HW_NotRDY</b> | RO     | 0     | Коррекция АЦП не произведена: 0 – коррекция АЦП пройдена. Бит устанавливается 0 после первого расчета корректирующих коэффициентов АЦП (происходит за время ~18 с). При установке бита DC_carrier регистра InputStngs, этот бит устанавливается в 0. 1 – устанавливается после сброса |

## C1Pole\_addi/C2Pole\_addi

Адрес: 31/63

**Описание:** Корректировка номера полюса [11:0]. Значение добавляется к виртуальному счетчику.

**Тип доступа:** Чтение/Запись (R/W)

## IC\_addr

Адрес: 64

**Описание:** Текущий адрес запросов к устройству Для того чтобы микросхема принимала и выдавала значения, необходимо установить  $BUS\_addr = 0$  или  $BUS\_addr == IC\_addr$

**Тип доступа:** Чтение/Запись (R/W)

## ADC\_config

Адрес: 65

**Описание:** Настройки периода работы преобразователей и частоты тактирования АЦП. Период работы преобразователя вычисляется по следующей формуле:  $Tclk = 16 * Tclk\_adc + Delay\_cycles * Tfint$  где FINT - частота тактирования цифрового блока (внешняя или с PLL)  $Tclk\_adc = 1/Fclk\_adc$ ,  $Fadc = FINT/(FINT\_divisor+1)$

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты  | Имя поля            | Доступ | Сброс | Описание   |
|---|-------|---------------------|--------|-------|--|
| 1 | 12    | <b>SEL_muxclk</b>   | R/W    | 0     | Выбор источника тактирования блока формирования синхрочастоты для АЦП и синусоидального генератора: 0 - частота цифрового блока 1 - частота с мультиплексора MUX_CLK           |
| 2 | 11..8 | <b>FINT_divisor</b> | R/W    | 3     | FINT_divisor - коэффициент деления частоты для тактирования АЦП $Fclk\_adc = FINT/(FINT\_divisor+1) = 1/Tclk\_adc$ , где Fclk_adc не должен превышать 20 МГц. FINT_divisor - 4 |

| № | Биты | Имя поля            | Доступ | Сброс | Описание   |
|---|------|---------------------|--------|-------|--|
|   |      |                     |        |       | битное беззнаковое значение в диапазоне [0..15]  |
| 3 | 7..0 | <b>DELAY_cycles</b> | R/W    | 0     | DELAY_cycles - значение добавочного количества тактов Fclk для составления необходимого периода преобразования. Период преобразования микросхемы Tadc = 16 * Tclk_adc + Delay_cycles*Tclk_adc<br>DELAY_cycles - 8 битное беззнаковое значение в диапазоне [0..255] |

## Mask\_Stat

Адрес: 66

**Описание:** маски для регистров C1Stat и C2Stat

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты | Имя поля               | Доступ | Сброс | Описание   |
|---|------|------------------------|--------|-------|--|
| 1 | 4    | <b>MSK_CLK_not_RDY</b> | R/W    | 1     | Маска для бита CLK_not_RDY: 0 – бит CLK_not_RDY устанавливается в состояние лог. «0» 1 – бит CLK_not_RDY отражает действительное состояние флага |
| 2 | 3    | <b>MSK_SPI_err</b>     | R/W    | 1     | Маска для бита SPI_err (ошибка при передаче по SPI): 0 – бит SPI_err устанавливается в   |

| № | Биты | Имя поля             | Доступ | Сброс | Описание  |
|---|------|----------------------|--------|-------|---|
|   |      |                      |        |       | состояние лог. «0» 1 – бит SPI_err отражает действительное состояние флага  |
| 3 | 2    | <b>MSK_Not_equal</b> | R/W    | 1     | Маска для бита Not_Equal (результаты преобразований не совпадают): 0 – бит Not_Equal устанавливается в состояние лог. «0» 1 – бит Not_Equal отражает действительное состояние флага |
| 4 | 1    | <b>MSK_nReady2</b>   | R/W    | 1     | Маска для бита nReady2 (преобразователь 2 Готов): 0 – бит nReady2 устанавливается в состояние лог. «0» 1 – бит nReady2 отражает действительное состояние флага                      |
| 5 | 0    | <b>MSK_nReady1</b>   | R/W    | 1     | Маска для бита nReady1 (преобразователь 1 Готов): 0 – бит nReady1 устанавливается в состояние лог. «0» 1 – бит nReady1 отражает действительное состояние флага                      |

## Flags\_delay

Адрес: 67

**Описание:** Flags\_delay [15:0] выполняет следующие функции: задает время обновления регистров Amp\_metric и флагов UIN\_HIGH, UIN\_LOW (с ревизии 3); устанавливает время удержания флагов. Единица времени 4/Fclk. После пропадания ошибки время удержания флагов (3×65535 – 4×65535) мкс. Желательно устанавливать время удержания флагов больше периода сигнала возбуждения датчика, чтобы избежать постоянного сброса и обратной установки флагов. В то же время, установка слишком большого значения нежелательна, т.к. время обновления флагов увеличивается.

**Тип доступа:** Чтение/Запись (R/W)

---

## WR\_lock

Адрес: 68

**Описание:** Блокировка записи настроек в микросхему. Если WR\_lock == 0, запись разрешена. При WR\_lock != 0, команды записи не исполняются

**Тип доступа:** Чтение/Запись (R/W)

---

## CMP\_lth

Адрес: 69

**Описание:** Максимальное допустимое различие результатов преобразования каналов 1 и 2 для выставления флага Not\_Equal в регистре Stat\_main.

**Тип доступа:** Чтение/Запись (R/W)

---

## AFE\_config

Адрес: 70

**Описание:** Регистр настройки аналоговых блоков

**Тип доступа:** Чтение/Запись (R/W)

## Поля регистра

| № | Биты   | Имя поля       | Доступ | Сброс | Описание  |
|---|--------|----------------|--------|-------|---|
| 1 | 15     | <b>OSC_DIS</b> | R/W    | 0     | Отключение подачи тактового сигнала на микросхему: 0 – частота подается 1 – частота не подается<br>При установлении бита в состояние лог. «1» тактовая частота внутри микросхемы отсутствует, и работа микросхемы останавливается. При этом возможно осуществлять запись регистров по SPI   |
| 2 | 14..12 | <b>Mode</b>    | R/W    | 0     | Выбор режима работы преобразователя. 000 – каналы преобразователя работают независимо 010 – каналы преобразователя работают параллельно и подключены к входам <b>IOSA1, IOCA1, EXI1, EXO1</b> 011 – каналы преобразователя работают параллельно и подключены к входам <b>IOSA2, IOCA2, EXI2, EXO2</b> 100 – каналы преобразователя работают параллельно в режиме Сельсин и подключены к входам <b>IOSA1, IOCA1, IOSA2, EXI1, EXO1</b> |
| 3 | 11     | <b>VC_mode</b> | R/W    | 0     | Режим работы пина VC: 0 – VC функционирует согласно описанию в разделе # 1 – режим выдачи разрешения выдачи <b>SDO</b> на приемопередатчик  |
| 4 | 10     | <b>DE_inv</b>  | R/W    | 0     | Полярность разрешения выдачи <b>SDO</b> на приемопередатчик: 0 –  |

| № | Биты | Имя поля         | Доступ | Сброс | Описание  |
|---|------|------------------|--------|-------|---|
|   |      |                  |        |       | выдача разрешена при DE = "1", при DE="0" прием 1 – выдача разрешена при DE = "0", при DE="1" прием   |
| 5 | 9    | <b>DE_half</b>   | R/W    | 0     | Режим выдачи сигнала DE: 0 – для полнодуплексного режима, DE выдается при nSEN=0, <b>SSTR</b> =0 1 – для полудуплексного режима, DE=1 только при запросах в режиме 001  |
| 6 | 8    | <b>Bus0_mode</b> | R/W    | 0     | Отвечать ли при групповом адресе шины BUS_addr = 0 и наличии адреса (IC_addr !=0): 0 – не отвечать 1 – отвечать   |
| 7 | 7    | <b>HALF_dma</b>  | R/W    | 0     | Переводить приемопередатчик и вывод <b>SDI</b> на спаде 16 такта <b>SCLK</b> текущей транзакции. Этот режим необходим для DMA транзакций в полудуплексом режиме. (Трехпроводный режим) 0 – открывать транзакцию только при новой транзакции SPI 1 – переводить приемопередатчик на последней транзакции |
| 8 | 6    | <b>DE_turnZ</b>  | R/W    | 0     | Перевод в неактивное состояние DE вывод <b>VC</b> в состояние высокого импеданса (high Z): 0 – выводить жесткий 0 (или 1 при инверсном режиме) - повышенная производительность 1 – для выключения передатчика переводить <b>VC</b> в состояние высокого   |

| №  | Биты | Имя поля         | Доступ | Сброс | Описание  |
|----|------|------------------|--------|-------|---|
|    |      |                  |        |       | импеданса (необходима подтяжка резистором)  |
| 9  | 4    | <b>EXO_sync</b>  | R/W    | 0     | Синхронизация таймеров счетчиков в режиме Меандр: 0 - синхронизация выключена 1 - синхронизация включена<br>Синхронизация требуется для управления комплементарными транзисторами накачки возбуждения.  |
| 10 | 3    | <b>SHRD_RAM</b>  | R/W    | 0     | Включение режима передачи массива ячеек памяти данных по адресам [256:384] одного из микровычислителей для расширения памяти данных другого вычислителя: 0 – массивы ячеек памяти данных [256:384] подключены к своим микровычислителям и обращение к ним производится независимо 1 – массив ячеек памяти данных [256:384] одного из микровычислителей подключается к расширению памяти другого |
| 11 | 2    | <b>SHRD_CPU2</b> | R/W    | 0     | Выбор микровычислителя для применения режима расширения памяти данных на 128 ячеек. Выбор активизируется при SHRD_RAM = 1. 0 – расширение производится для CPU1 1 – расширение производится для CPU2  |

| №  | Биты | Имя поля           | Доступ | Сброс | Описание  |
|----|------|--------------------|--------|-------|---|
| 12 | 1    | <b>VREF_DAC_en</b> | R/W    | 0     | Включение опорного сигнала для генератора 2.5 В синусоидального возбуждения: 0 – опорный сигнал выключен 1 – опорный сигнал включен |
| 13 | 0    | <b>VREF_en</b>     | R/W    | 0     | Включение источника опорного напряжения 2.5 В: 0 – источник опорного напряжения выключен 1 – источник опорного напряжения включен   |

## Mode\_config

Адрес: 71

**Описание:** Регистр общей настройки микросхемы

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты | Имя поля          | Доступ | Сброс | Описание   |
|---|------|-------------------|--------|-------|--|
| 1 | 15   | <b>SPI_CRC_en</b> | R/W    | 1     | Включение бита четности CRC: 0 – бит четности при передаче по SPI игнорируется 1 – бит четности требуется для записи и чтения из SPI |
| 2 | 14   | <b>ADC_en</b>     | R/W    | 0     | Включение опроса АЦП: 0 - опрос АЦП выключен 1 - опрос АЦП работает  |

| № | Биты | Имя поля          | Доступ | Сброс | Описание   |
|---|------|-------------------|--------|-------|--|
| 3 | 13   | <b>CPU2_en</b>    | R/W    | 0     | Включение CPU2: 0 - CPU2 выключен, находится в состоянии сброса 1 - CPU2 включен   |
| 4 | 12   | <b>CPU1_en</b>    | R/W    | 0     | Включение CPU1: 0 - CPU1 выключен, находится в состоянии сброса 1 - CPU1 включен   |
| 5 | 11   | <b>CONV2_en</b>   | R/W    | 0     | Включение преобразователя 2: 0 – преобразователь 2 отключен и находится в состоянии сброса 1 – преобразователь 2 включен                       |
| 6 | 10   | <b>CONV1_en</b>   | R/W    | 0     | Включение преобразователя 1: 0 – преобразователь 1 отключен и находится в состоянии сброса 1 – преобразователь 1 включен                       |
| 7 | 9    | <b>EXO1_en</b>    | R/W    | 0     | Включение тактирования формирователя частоты на EXO1: 0 – формирователь EXO1 не тактируется 1 – формирователь EXO1 тактируется                 |
| 8 | 8    | <b>EXO2_en</b>    | R/W    | 0     | Включение тактирования формирователя частоты на EXO2: 0 – формирователь EXO2 не тактируется 1 – формирователь EXO2 тактируется                 |
| 9 | 7..6 | <b>Sample_src</b> | R/W    | 3     | Выбор строба для выборки данных в SPI. 11 – Выборка производится сигналами SPI в момент чтения регистров 10 – при входе Sample равном лог. «0» |

| №  | Биты | Имя поля            | Доступ | Сброс | Описание  |
|----|------|---------------------|--------|-------|---|
|    |      |                     |        |       | данные защелкиваются в регистры 01 – при входе Sample равном лог. «1» данные защелкиваются в регистры 00 – регистры координаты, скорости и состояния сохраняют свое предыдущее состояние  |
| 10 | 4..2 | <b>DB_mode</b>      | R/W    | 0     | Управление источником цифрового сигнала DB. 000 - выход DB подключен к CLKMUX_OUT 001 - выход DB подключен к DB_SIG_PLLBASE 010 - выход DB подключен к DB_SIG_CLKREF 011 - выход DB подключен к CLK 100 - выход DB подключен к STP CPU1 101 - выход DB подключен к STP CPU2 110 - выдача на DB лог. «0» 111 - выдача на DB лог. «1» |
| 11 | 1    | <b>CPU_CRC4_en</b>  | R/W    | 0     | Включение выдачи CRC4 при чтении ячеек CPU: 0 – Четные адреса: [d1[1:0], d0[13:0]], нечетные адреса [ 2'b00, d1[13:0] ] 1 – Четные адреса: [d1[1:0], d0[13:0]], нечетные адреса [ crc4[3:0], d1[13:2] ]   |
| 12 | 0    | <b>HAND_CRC4_en</b> | R/W    | 0     | Включение выдачи CRC4 при чтении выходных тактируемых ячеек HAND1 и HAND2: 0 – Четные адреса: d0[15:0], нечетные адреса [ 4'b00000, d1[13:0] ] 1 – Четные адреса: d0[15:0], нечетные адреса [ crc4[3:0], d1[13:0] ]   |

# NOCLK\_stat

Адрес: 72

**Описание:** Биты состояния микросхемы

**Тип доступа:** Только чтение (RO)

## Поля регистра

| №  | Биты | Имя поля             | Доступ | Сброс | Описание   |
|----|------|----------------------|--------|-------|--|
| 1  | 10   | <b>SAMPLE_pin</b>    | RO     | 1     | Логический уровень на входе Sample                         |
| 2  | 9    | <b>VC_pin</b>        | RO     | 1     | Логический уровень на входе VC                             |
| 3  | 8    | <b>CLKDLY_rdy</b>    | RO     | 1     | Clock delay ready  |
| 4  | 7    | <b>CLKCOMP_rdy</b>   | RO     | 1     | Clock comp ready   |
| 5  | 6    | <b>CLK_rdy</b>       | RO     | 1     | CLK ready  |
| 6  | 5    | <b>STNDBY</b>        | RO     | 0     | Pad <b>STNDBY</b>  |
| 7  | 4    | <b>POR_NRST</b>      | RO     | 1     | (POR & <b>NRESET</b> pin) after delay                      |
| 8  | 3    | <b>NRST_pll</b>      | RO     | 1     | Сигнал сброса логики уровня PLL                            |
| 9  | 2    | <b>NRST_clkcomp</b>  | RO     | 1     | Сигнал сброса логики уровня блока сравнения частот         |
| 10 | 1    | <b>NRST_synchclk</b> | RO     | 1     | Сигнал сброса логики уровня синхронизатора сброса          |
| 11 | 0    | <b>NRST_rdyinit</b>  | RO     | 1     | Сигнал сброса логики уровня готовности блока инициализации |

---

## SPI\_req

Адрес: 73

**Описание:** Предыдущая транзакция SPI

**Тип доступа:** Только чтение (RO)

---

## alive\_cnt

Адрес: 74

**Описание:** alive\_cnt [15:0] – счетчик считает во время работы микросхемы. Единица времени 32768/Fclk

**Тип доступа:** Только чтение (RO)

---

## Stat\_main

Адрес: 75

**Описание:** Регистр состояния микросхемы.

**Тип доступа:** Только чтение (RO)

### Поля регистра

| № | Биты | Имя поля           | Доступ | Сброс | Описание   |
|---|------|--------------------|--------|-------|--|
| 1 | 4    | <b>CLK_not_RDY</b> | RO     | 1     | Флаг готовности тактовой частоты: 0 – синхросигнал не подается на цифровой блок 1 – синхросигнал подается на цифровой блок (прошла задержка) |

| № | Биты | Имя поля         | Доступ | Сброс | Описание   |
|---|------|------------------|--------|-------|--|
| 2 | 3    | <b>SPI_err</b>   | RO     | 1     | Флаг ошибки при передаче по SPI: 0 – при приеме предыдущего кадра по интерфейсу SPI ошибок не было 1 – при приеме предыдущего кадра по интерфейсу SPI возникла ошибка. Необходимо выполнить сброс микросхемы. Флаг может быть сброшен записью лог. «1»   |
| 3 | 2    | <b>Not_equal</b> | RO     | 0     | Результаты преобразований не совпадают: 0 – результаты преобразований конвертеров 1 и 2 совпадают или различаются не больше, чем на величину, установленную в регистре CMP_lth 1 – результаты преобразований конвертеров 1 и 2 различаются больше, чем на величину, установленную в регистре CMP_lth |
| 4 | 1    | <b>nReady2</b>   | RO     | 1     | Преобразователь 2 готов: 0 – регистр C2Stat не содержит установленных в «1» битов 1 – регистр C2Stat содержит установленные в «1» биты   |
| 5 | 0    | <b>nReady1</b>   | RO     | 1     | Преобразователь 1 готов: 0 – регистр C1Stat не содержит установленных в «1» битов 1 – регистр C1Stat содержит установленные в «1» биты   |

## Дсру1LB

Адрес: 76

**Описание:** Выходная шина с регистров CPU1, младшее слово.

**Тип доступа:** Только чтение (RO)

---

## Дсру1НВ

Адрес: 77

**Описание:** Выходная шина с регистров CPU1, старшее слово.

**Тип доступа:** Только чтение (RO)

---

## Дсру2LB

Адрес: 78

**Описание:** Выходная шина с регистров CPU2, младшее слово.

**Тип доступа:** Только чтение (RO)

---

## Дсру2НВ

Адрес: 79

**Описание:** Выходная шина с регистров CPU2, старшее слово.

**Тип доступа:** Только чтение (RO)

---

## PLL\_config

Адрес: 80

**Описание:** Регистр настройки режимов тактирования микросхемы

**Тип доступа:** Чтение/Запись (R/W)

## Поля регистра

| № | Биты   | Имя поля           | Доступ | Сброс | Описание  |
|---|--------|--------------------|--------|-------|---|
| 1 | 15..14 | <b>PLL_basecmp</b> | R/W    | 0     | <p>Параметр критерия готовности умножителя частоты PLL. Если задано ненулевое значение, то производится подсчет импульсов выходной частоты и сравнение с опорной на 7 битном счетчике: если значения расходятся, то происходит удержание сигнала RESET 00 – признак не влияет на удержание сброса 01 – приведенная к базовой частота PLL отличается не более чем на [-1..1] импульсов 10 – приведенная к базовой частота PLL отличается не более чем на [-2..2] импульсов 11 – приведенная к базовой частота PLL отличается не более чем на [-4..4] импульсов</p> |
| 2 | 13..12 | <b>CLK_delay</b>   | R/W    | 0     | <p>Количество импульсов синхрочастоты, при которых удерживается сигнал RESET: 0 – 8192 импульса синхрочастоты 1 – 4096 импульса синхрочастоты 2 – 2048 импульса синхрочастоты 3 – сброс не удерживается</p>   |
| 3 | 11..8  | <b>PLL_Q</b>       | R/W    | 0     | <p>Коэффициент деления для получения внутренней тактовой частоты. <math>FINT = 0,5 \times fOSC \times PLL\_N / (PLL\_Q + 1)</math> PLL_Q = 0....15, fOSC – частота на входе <b>TECH2</b><br/>Примечание: значение</p>   |

| № | Биты | Имя поля  | Доступ | Сброс | Описание   |
|---|------|-----------|--------|-------|--|
|   |      |           |        |       | fOSC/(PLL_Q+1) должно быть в диапазоне от 2 до 16 МГц  |
| 4 | 7    | PLL_BOOST | R/W    | 1     | Выдача начального напряжения на PLL для ускорения достижения заданной частоты 0 – без начального напряжения 1 – вместе с начальным напряжением   |
| 5 | 6..0 | PLL_N     | R/W    | 0     | Коэффициент умножения для получения внутренней тактовой частоты. $FINT = 0,5 \times fOSC \times PLL\_N / (PLL\_Q + 1)$ PLL_N = 3 до 74, fOSC – частота на входе TECH2 Если PLL_N = 0, блок ФАПЧ отключается и переходит в низко потребляющий режим. Тактирование производится от внешнего тактового генератора |

## INIT\_conf

Адрес: 81

**Описание:** Регистр управления режимом начальной конфигурации микросхемы

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты | Имя поля | Доступ | Сброс | Описание  |
|---|------|----------|--------|-------|---|
| 1 | 5    | LVL_pin  | R/W    | 0     | Выключение проставок в цифровых портах: 0 - проставки включены 1 - проставки закорочены |

| № | Биты | Имя поля           | Доступ | Сброс | Описание   |
|---|------|--------------------|--------|-------|--|
| 2 | 4    | <b>OTP_init_on</b> | R/W    | 0     | Инициализация из ВОТР памяти: 0 - инициализация включена 1 - инициализация выключена   |
| 3 | 3..0 | <b>ВОТР_clkdel</b> | R/W    | 4     | Делитель частоты, для функционирования ВОТР $F_{clk\_rom} = F_{INT}/(3 + \text{ВОТР\_clkdel})$ , где $F_{clk\_rom}$ должен быть не более 10 МГц<br>ВОТР_clkdel - 4 битное беззнаковое значение [0..15] |

## UOTP\_ctrl

Адрес: 82

**Описание:** Регистр управления записи и чтения пользовательской памяти прямого доступа (UOTP), отвечает за регистры PLL\_CONFIG, INIT\_conf

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты | Имя поля               | Доступ | Сброс | Описание  |
|---|------|------------------------|--------|-------|---|
| 1 | 3    | <b>MANUAL_NRST_PLL</b> | R/W    | 1     | Переключить выход пользовательской ячейки SPI на реальное значение OTP памяти |
| 2 | 2    | <b>WATCH_ROM_UVAL</b>  | R/W    | 0     | Переключить выход пользовательской ячейки SPI на реальное значение OTP памяти |

| № | Биты | Имя поля             | Доступ | Сброс | Описание  |
|---|------|----------------------|--------|-------|---|
| 3 | 1    | <b>PROG_NEW_UVAL</b> | R/W    | 0     | Разрешить запись значения из ячейки SPI в память OTP                        |
| 4 | 0    | <b>OVERRIDE_UVAL</b> | R/W    | 0     | Переопределить текущее значение пользовательской настройки значением из SPI |

---

## BUS\_addr

Адрес: 83

**Описание:** Текущий адрес запросов к устройству Если задан IC\_addr, чтобы микросхема принимала и выдавала значения необходимо установить BUS\_addr = 0 или BUS\_addr == IC\_addr

**Тип доступа:** Чтение/Запись (R/W)

---

## BOTP\_addr

Адрес: 84

**Описание:** Регистр адреса блока OTP памяти 512x16 бит (BOTP)

**Тип доступа:** Чтение/Запись (R/W)

---

## BOTP\_data

Адрес: 85

**Описание:** Регистр данных блока OTP памяти 512x16 бит (BOTP)

**Тип доступа:** Чтение/Запись (R/W)

---

## BOTP\_ctrl

Адрес: 86

**Описание:** Регистр управления записи и чтения блока OTP памяти 512x16 бит (BOTP)

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты | Имя поля     | Доступ | Сброс | Описание  |
|---|------|--------------|--------|-------|---|
| 1 | 3    | <b>PGM</b>   | R/W    | 0     | PGM - бит разрешения записи: 0 - запись не производится 1 - запись по адресу, записанному в регистре <b>BOTP_addr</b> и данного, записанного в регистр <b>BOTP_data</b> |
| 2 | 2    | <b>REN</b>   | R/W    | 0     | REN - режим чтения: 0 - чтение не производится 1 - чтение по адресу, записанному в регистре <b>BOTP_addr</b>  |
| 3 | 1    | <b>NCEN</b>  | R/W    | 1     | NCEN - разрешение транзакции для <b>BOTP</b> : 0 - <b>BOTP</b> принимает команды, адреса и данные 1 - <b>BOTP</b> игнорирует  |
| 4 | 0    | <b>SLEEP</b> | R/W    | 1     | Сигнал перехода блока <b>BOTP</b> в режим пониженного энергопотребления: 0 - <b>BOTP</b> включен, готов к чтению/записи 1 - <b>BOTP</b> выключен, недоступен            |

## BOTP\_out

Адрес: 87

**Описание:** Выход блока OTP памяти 512x16 бит (BOTP)

**Тип доступа:** Только чтение (RO)

## P1BG\_ctrl/P2BG\_ctrl

Адрес: 92/94

**Описание:** Регистр управления отладкой CPU

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты   | Имя поля         | Доступ | Сброс | Описание  |
|---|--------|------------------|--------|-------|---|
| 1 | 12..11 | <b>command</b>   | R/W    | 0     | 00 - запустить CPU до точки останова, если она активна<br>01 - остановить CPU как есть<br>10 - выполнить текущую операцию |
| 2 | 10     | <b>tap_event</b> | R/W    | 0     | Переход из 0 в 1 активизирует одну из команд (00, 01, 10)   |
| 3 | 9      | <b>stop1_ena</b> | R/W    | 0     | Включение режима отслеживания точки останова 1  |
| 4 | 8..0   | <b>stop1_pc</b>  | R/W    | 0     | Адрес точки останова 1  |

## P1BG\_data/P2BG\_data

Адрес: 93/95

**Описание:** Данные управления отладкой CPU

**Тип доступа:** Чтение/Запись (R/W)

### Поля регистра

| № | Биты   | Имя поля         | Доступ | Сброс | Описание                                       |
|---|--------|------------------|--------|-------|--|
| 1 | 13..10 | <b>sel_reg</b>   | R/W    | 0     | Выбор регистра из CPU для чтения               |
| 2 | 9      | <b>stop2_ena</b> | R/W    | 0     | Включение режима отслеживания точки останова 2 |
| 3 | 8..0   | <b>stop2_pc</b>  | R/W    | 0     | Адрес точки останова 2                         |

---

Последнее обновление **17 июн. 2026 г.** от **Илья Slepchenkov**

# Методика подключения и настройки

Настройка должна производиться в порядке, заданном нумерацией разделов.

## 1. Проверка подключения

а. С помощью осциллографа убедитесь в наличии напряжения питания на выводах **VDDA**(63) = 5,0 В и **VDDD**(33) = 5,0 В, а также на выводах **VDDIO**(45) = 3,3 В, **VDDIO2**(25) = 3,3 В, **VPP**(35) = 3,3 В. При использовании генератора опорной частоты убедитесь в наличии напряжения питания на выводах **VDDEXO1**(5) = 5,0 В и/или **VDDEXO2**(4) = 5,0 В.

б. Убедитесь в наличии напряжения на выходах встроенных линейных регуляторов: **VDDD1P8**(36, 61)  $\approx$  1,8 В, **VDDD\_INT**(31)  $\approx$  4,0 В, **VDDA\_INT**(30)  $\approx$  4,0 В.

в. При тактировании с использованием кварцевого резонатора убедитесь в наличии тактовой частоты на выходе генератора **TP\_FQO**(26). Убедитесь в наличии тактового сигнала на технологическом выводе **DB**(41) — по умолчанию на нём присутствует сигнал **CLKMUX\_OUT**, поделённый на 8 (поле **DB\_mode** регистра **Mode\_config** = 000 после сброса). Если на выводе **DB** тактовый сигнал отсутствует — проверьте подключение служебных выводов **TECH1**(37) и **TECH2**(38).

г. Убедитесь, что служебные выводы подключены корректно: **TECH1**(37) подключён к GND, **TECH2**(38) подключён к **VDDD\_INT**.

## 2. Настройка тактовой частоты микросхемы

а. При тактировании от кварцевого резонатора (подключённого к **TP\_QI**(28), **TP\_QO**(27), выход **TP\_FQO**(26) подключён к **CLK60**(39)) — дополнительная настройка источника тактирования не требуется.

b. При тактировании от внешнего генератора (подключённого к CLK60(39)): вывод TP\_QI(28) должен быть подключён к VDDD\_INT, вывод TP\_QO(27) — к GND.

c. Запишите в регистр PLL\_config (адрес 80) рассчитанные значения PLL\_N[6:0], PLL\_Q[11:8]. Для ускорения выхода PLL на режим установите PLL\_BOOST[7] = 1. Для контроля готовности PLL установите PLL\_basecmp[15:14] ≠ 00. Если PLL не используется, запишите PLL\_N = 0 — тактирование будет производиться напрямую от внешнего генератора на CLK60.

d. Запишите в регистр ADC\_config (адрес 65) значения FINT\_divisor[11:8] и DELAY\_cycles[7:0]. Частота тактирования АЦП  $F_{clk\_adc} = F_{INT} / (F_{INT\_divisor} + 1)$  не должна превышать 20 МГц. Период работы преобразователя  $T_{clk} = 16 \times T_{clk\_adc} + DELAY\_cycles \times T_{fint}$ .

e. Убедитесь в готовности тактовой частоты: считайте регистр NOCLK\_stat (адрес 72) — биты CLK\_rdy[6] и CLKCOMP\_rdy[7] должны быть равны 1. Также убедитесь, что AFE\_config.OSC\_DIS[15] = 0.

#### **ⓘ ПРИМЕЧАНИЕ**

При некорректной настройке тактовой частоты может неправильно работать чтение регистров SPI. В этом случае произведите внешний сброс микросхемы (вывод NRESET), либо запишите корректные значения в регистры PLL\_config, AFE\_config (запись в эти регистры функционирует при отсутствующей или слишком большой тактовой частоте).

## **3. Настройка режима работы аналоговой части**

a. Для использования встроенного источника опорного напряжения АЦП запишите AFE\_config.VREF\_en (адрес 70, бит 0) = 1. При этом на выводе VREF2P5(11) появится напряжение 2,5 В. Если используется внешний источник опорного напряжения, подайте его на вывод VREF2P5(11) и установите VREF\_en = 0.

b. При использовании встроенного синусоидального генератора возбуждения запишите AFE\_config.VREF\_DAC\_en (бит 1) = 1. При этом опорное напряжение ЦАП будет

формироваться внутри микросхемы и подаваться на вывод REFDAC(64), который необходимо зашунтировать конденсатором 0,1 мкФ.

с. Запишите в поле **Mode** регистра **AFE\_config** (адрес 70, биты [14:12]) требуемый режим:

- 000 — каналы преобразователя работают независимо
- 010 — каналы работают параллельно, подключены к входам **IOSA1**, **IOCA1**, **EXI1**, **EXO1**
- 011 — каналы работают параллельно, подключены к входам **IOSA2**, **IOCA2**, **EXI2**, **EXO2**
- 100 — режим сельсин (оба канала параллельно с преобразованием 3 фазы в 2)

d. При необходимости использования встроенных операционных усилителей на входах сигналов с датчиков запишите **C1InputStngs.OPA\_en** (адрес 9, бит 8) = 1. При **OPA\_en** = 0 сигналы подаются на АЦП непосредственно через выводы **IOSA1**, **IOCA1**.

e. Для модулированных сигналов датчика (СКВТ, сельсин, ЛРДТ): **C1InputStngs.DC\_carrier** (бит 5) = 0, **DC\_correction** (бит 4) = 1.

f. Убедитесь, что **WR\_lock** (адрес 68) = 0. Если значение отлично от нуля, запишите 0 для разрешения записи в регистры.

## 4. Настройка сигнала возбуждения датчика (на примере канала 1)

a. При использовании внешнего сигнала возбуждения: **C1ExoStngs.EXO\_mode** (адрес 6, биты [11:10]) = 00 (генератор выключен), **C1InputStngs.Ex\_source** (адрес 9, биты [10:9]) = 1 (внешний сигнал с **EXI1**). На этом настройка возбуждения окончена, переходите к разделу 5.

b. При использовании встроенного генератора в режиме синусоидального сигнала: запишите **C1ExoStngs.EXO\_mode** (адрес 6, биты [11:10]) = 10, **Amp\_code** (биты [7:0]) — начальное значение амплитуды. Включите тактирование генератора: **Mode\_config.EXO1\_en** (адрес 71, бит 9) = 1.

с. Запишите в регистр **C1EXInc** (адрес 7) значение, соответствующее выбранной частоте возбуждения датчика (см. формулу (5) в разделе **Программируемый генератор**). Частота рассчитывается по формуле:  $f_{ex} = EXInc / 2^{20} \times f_{clk}$ .

d. Используя осциллограф, подключённый к выводу **EXO1(6)**, убедитесь в наличии и параметрах сигнала (амплитуда, частота, отсутствие нелинейных искажений). Амплитуда сигнала должна быть ниже напряжения питания **VDDEXO1** как минимум на 0,4 В. При необходимости скорректируйте **Amp\_code** или **C1EXInc**.

e. Подключите датчик. Осциллографом убедитесь в параметрах сигнала возбуждения на входе датчика (амплитуда, искажения) и сигналов вторичных обмоток, подключённых к микросхеме. Убедитесь в отсутствии насыщения сердечника датчика. При необходимости скорректируйте схему усилителя-фильтра на плате.

## 5. Настройка входных каскадов (на примере канала 1)

a. Вращая/двигая датчик, оцените параметры сигнала на входах АЦП **IOSA1(24)**, **IOCA1(21)**. Максимальная амплитуда сигналов должна быть  $2 \text{ В} \pm 10 \%$  (при **VREF2P5** = 2,5 В). Средний уровень (синфазное напряжение)  $1,25 \text{ В} \pm 10 \%$  (при **VREF2P5** = 2,5 В). Убедитесь в отсутствии значительных нелинейных искажений.

b. При подключении датчиков типа ЛРДТ с последовательно соединёнными обмотками (**Sensor\_mode** = 10) дополнительно убедитесь в совпадении фаз сигналов **IOSA1**, **IOCA1**. При необходимости скорректируйте сдвиг фаз с помощью RC-цепи в одном из каналов. Разность фаз между сигналами удобно измерять в режиме осциллографа XY — при одинаковых фазах овал сливается в линию.

c. При использовании внешнего источника напряжения возбуждения: подайте внешний сигнал на вход **EXI1(50)** через компаратор (см. [рекомендуемую схему](#)). Сигнал на входе **EXI1** — цифровой с уровнями 0 и 1.

Убедитесь в совпадении фаз сигналов на входе **EXI1** и входах **IOSA1**, **IOCA1**. Для этого установите СКВТ примерно на угол  $45^\circ$ , ЛРДТ — в крайнее положение. Осциллографом контролируйте соотношение фаз сигналов на входе **IOSA1** и **EXI1**. При необходимости сдвиньте фазу сигнала на входе **EXI1** с помощью RC-цепи.

- Если **C1InputStngs.Ex\_recovery\_en** = 0: необходимо **точное** совпадение фаз, т.к. сигнал со входа **EXI1** используется для демодуляции. Сдвиг фазы в датчике зависит от температуры. Несовпадение фаз ведёт к росту шума и значительному падению точности. Данный режим не рекомендуется.

- Если `Ex_recovery_en` = 1: необходимо **приблизительное** совпадение фаз (в пределах  $\pm 45^\circ$ ). Сигнал `EXI1` используется для определения квадранта, а демодуляция производится сигналом, выделенным из АЦП. Температурная зависимость отсутствует. Для датчиков типа ЛРДТ при `Ex_recovery_en` = 1 можно подтянуть вход `EXI1` к уровню логического нуля или единицы.

с. При использовании встроенного генератора возбуждения на выводе `EXI1` (50) появляется сигнал `Ex_ref`, который используется внутри микросхемы для демодуляции (при `Ex_recovery_en` = 0) или определения квадранта (при `Ex_recovery_en` = 1).

Необходимо добиться совпадения фаз сигналов на выходе `EXI1` и на входах `IOSA1`, `IOCA1`. Для этого установите СКВТ примерно на угол  $45^\circ$ , ЛРДТ — в крайнее положение. Осциллографом контролируйте соотношение фаз. С помощью регистра `C1ExPhShft` (адрес 5) сдвиньте фазу сигнала `Ex_ref` так, чтобы она совпала с фазами `IOSA1`, `IOCA1`.

- При `Ex_recovery_en` = 0: необходимо точное совпадение фаз. Температурная зависимость присутствует. Режим не рекомендуется.
- При `Ex_recovery_en` = 1: необходимо приблизительное совпадение фаз (в пределах  $\pm 45^\circ$ ). Температурная зависимость отсутствует. Настройку `C1ExPhShft` необходимо производить и для датчиков типа ЛРДТ.

### ВНИМАНИЕ

При неправильной настройке блока восстановления опорного сигнала преобразователь может давать ошибку  $180^\circ$ , при этом флаги ошибок будут сброшены.

## 5. Диагностика подключения (на примере канала 1)

а. Запишите `C1Mask` (адрес 12) = 0xFFFFE. Запишите тип датчика в поле `C1KonturStngs.Sensor_mode` (адрес 13, биты [11:10]). При подключении сельсина установите `AFE_config.Mode` = 100. Включите АЦП и преобразователь: `Mode_config.ADC_en` (адрес 71, бит 14) = 1, `CONV1_en` (бит 10) = 1.

- b. Запишите `C1KonturStngs.LBW` (биты [4:0]) = 14 (или по таблице полос пропускания). Запишите `C1ResCntrl.Vel_resolution` (биты [8:5]) = 7, `Coord_resolution` (биты [3:0]) = 2 (16-битная координата).
- c. Двигая датчик, считывайте регистр `C1Stat` (адрес 30). В первую очередь убедитесь, что флаги `CLIP_COS` = 0, `CLIP_SIN` = 0, `ADC_OVF` = 0 всегда.
- d. Флаги `UIN_LOW`, `UIN_HIGH`, `CORR_OVF` служат для диагностики сигналов после коррекции. При подключении СКВТ/сельсина и выполнении разделов 4 и 5 они должны быть равны 0. При изменении регистров `C1KampC`, `C1KampS`, `C1ExPhShft`, `C1KbiasC`, `C1KbiasS`, `C1fbias` убедитесь, что новые флаги ошибки не появились. Для датчиков типа ЛРДТ, чтобы снять ошибку `UIN_HIGH`, запишите в `C1Amp_th.Amp_lth` (адрес 8, биты [7:0]) новое значение (рекомендуемое 1000).
- e. Вращая/двигая датчик, считывайте значения координаты `C1Coord` (адрес 16) и скорости `C1Vel` (адрес 24).
- Если координата изменяется случайным образом по всему диапазону при неподвижном датчике: выключите преобразователь (`CONV1_en` = 0), затем снова включите (`CONV1_en` = 1). Если не помогло — повторите все пункты сначала.
  - Если угол отличается на 180° или ЛРДТ находится на упоре: проверьте сигнал `EXI1` — вероятно, он инверсный. Либо повторите настройку раздела 5.
  - В режиме сельсина при нелинейном изменении угла (например, поворот на 90° даёт изменение только на 60°): неправильно подключены обмотки сельсина.
  - Для сдвига значения координаты относительно получаемой с датчика запишите значение в регистр `C1Zero` (адрес 11).

## 6. Калибровка усиления (на примере канала 1)

a. Если для схемы входных каскадов использованы компоненты с разбросом значений больше 0,01 %, создаётся заметная дополнительная ошибка преобразования. В этом случае рекомендуется калибровка разницы усиления по каналам платы. Для режима ЛРДТ с последовательно соединёнными обмотками (`Sensor_mode` = 10) калибровка не имеет смысла, т.к. сигналы несимметричные по своей природе.

b. Запишите `C1InputStngs.Ex_recovery_en = 1`, `Ex_source = 1`. Запишите `C1KonturStngs.LBW = 16`. Запишите `C1ResCntrl.Coord_resolution = 2`.

c. С генератора сигналов подайте на входы платы в канал синуса и косинуса абсолютно одинаковые сигналы (синусоидальные, можно попробовать меандр). Частота должна быть равной частоте сигнала возбуждения датчика. Выход синхронизации генератора подайте на вход `EXI1`.

d. Для режима СКВТ (`Sensor_mode = 00`): считайте значение угла — должно получиться примерно  $45^\circ$  или  $-45^\circ$  (в зависимости от полярности на `EXI1`). С помощью регистров `C1KampS` (адрес 0), `C1KampC` (адрес 1) добейтесь получения кодов 8192 или 40960.

e. Для режима 5-ти проводной ЛРДТ (`Sensor_mode = 01`): считайте значение координаты — должно получиться примерно 0. С помощью регистров `C1KampS`, `C1KampC` добейтесь получения кода 0.

f. Верните изменённые регистры в исходное состояние.

## 7. Настройка флагов состояния (на примере канала 1)

a. После выбора параметров преобразователей настройте пороги флагов ошибок.

b. В регистре `C1Amp_th` (адрес 8) содержатся пороги `Amp_hth[15:8]` и `Amp_lth[7:0]` для флагов `UIN_HIGH` и `UIN_LOW`. Для режима СКВТ значения по умолчанию (`Amp_hth = 187`, `Amp_lth = 62`) подходят. Вместо расширения порогов предпочтительно скорректировать амплитуду сигнала на входе или в регистрах `C1KampS`, `C1KampC`.

c. В регистре `C1Lock_th` (адрес 10) содержится порог срабатывания для флага `NLock`. Существует 2 сценария использования:

- Диагностика: рекомендуется ставить `C1Lock_th` в большие значения.
- Переключение полосы пропускания при ускорении: подберите порог на срабатывание при минимальном ускорении, но при этом обеспечьте отсутствие срабатывания в положении покоя.

d. При включении преобразователя с малыми значениями `C1Lock_th` может наблюдаться повышенный шум координаты (модуляция частотой  $2 \times f_{ex}$ ) и флаг `NLock`.

Преобразователь убирает эти компоненты в процессе работы (до 30 с). Значение координаты и скорости при этом корректно. Возможные решения: а) поднять порог `C1Lock_th` (временно или постоянно); б) замаскировать бит `NLock` в `C1Mask`; в) установить `C1InputStngs.DC_correction = 0`.

е. Неиспользованные биты в регистре `C1Stat` должны быть замаскированы установкой 0 в соответствующий бит `C1Mask` (адрес 12).

ф. При получении всех флагов в регистре `C1Stat` равными 0: `Stat_main.nReady1` (адрес 75, бит 0) = 0.

г. Неиспользованные биты регистра `Stat_main` должны быть замаскированы в регистре `Mask_Stat` (адрес 66). При всех замаскированных флагах равными 0 — микросхема готова к работе.

h. Регистр `Flags_delay` (адрес 67) задаёт время обновления метрик и удержания флагов (единица времени  $4/F_{clk}$ ). Рекомендуется устанавливать значение больше периода сигнала возбуждения датчика.

## 8. Пример настройки для подключения СКВТ

Пример для кварцевого резонатора 10 МГц,  $F_{INT} = 60$  МГц, частота преобразования  $F_{clk} \approx 500$  кГц, частота возбуждения  $\approx 10$  кГц.

`PLL_config` (адрес 80) = `0x4086`

- `PLL_basecmp[15:14] = 01` — контроль готовности PLL
- `CLK_delay[13:12] = 00` — задержка 8192 такта
- `PLL_Q[11:8] = 0`
- `PLL_BOOST[7] = 1` — ускоренный запуск PLL
- `PLL_N[6:0] = 6`

`ADC_config` (адрес 65) = `0x0218`

- `FINT_divisor[11:8] = 2` →  $F_{clk\_adc} = 60/3 = 20$  МГц
- `DELAY_cycles[7:0] = 24` →  $T_{clk} = (16+24) \times 50$  нс = 2000 нс,  $f_{clk} = 500$  кГц

INIT\_conf (адрес 81) = 0x0003

- BOTP\_clkdel[3:0] = 3 → Fclk\_rom = 60/(3+3) = 10 МГц

AFE\_config (адрес 70) = 0x0003

- Mode[14:12] = 000 — независимые каналы
- VREF\_DAC\_en[1] = 1 — ИОН ЦАП включен
- VREF\_en[0] = 1 — ИОН АЦП включен

Mode\_config (адрес 71) = 0xC6C0

- SPI\_CRC\_en[15] = 1 — контроль чётности SPI
- ADC\_en[14] = 1 — АЦП включен
- CONV1\_en[10] = 1 — преобразователь 1 включен
- EXO1\_en[9] = 1 — генератор EXO1 тактируется
- Sample\_src[7:6] = 11 — выборка по SPI

Mask\_Stat (адрес 66) = 0x001D

- Маскируем nReady2 (преобразователь 2 не используется)

C1Mask (адрес 12) = 0xFFFFE

C1ExoStngs (адрес 6) = 0x0850

- EXO\_mode[11:10] = 10 — синусоидальный сигнал
- Amp\_code[7:0] = 80

C1EXInc (адрес 7) = 20972

- $f_{ex} \approx 20972 / 2^{20} \times 500000 \approx 10$  кГц

C1InputStngs (адрес 9) = 0x0810

- Ex\_recovery\_en[11] = 1 — восстановление опоры включено
- Ex\_source[10:9] = 00 — от внутреннего генератора
- DC\_correction[4] = 1 — коррекция синфазного напряжения

C1KonturStngs (адрес 13) = 0x000E

- Sensor\_mode[11:10] = 00 — СКВТ
- LBW[4:0] = 14 — полоса пропускания

C1ResCntrl (адрес 14) = 0x00E2

- Vel\_resolution[8:5] = 7 — 15-битная скорость
- Coord\_resolution[3:0] = 2 — 16-битная координата

C1ExPhShft (адрес 5) — подбирается согласно методике (раздел 5).

C1Lock\_th (адрес 10) = 1000 — используется для диагностики, порог повышен.

### Рекомендуемая последовательность записи регистров:

1. PLL\_config = 0x4086
2. INIT\_conf = 0x0003
3. Ожидание готовности тактовой частоты (NOCLK\_stat.CLK\_rdy = 1)
4. AFE\_config = 0x0003
5. ADC\_config = 0x0218
6. Mask\_Stat = 0x001D
7. C1ExoStngs = 0x0850
8. C1EXInc = 20972
9. C1InputStngs = 0x0810
10. C1KonturStngs = 0x000E
11. C1ResCntrl = 0x00E2
12. C1Lock\_th = 1000
13. C1Mask = 0xFFFFE
14. Mode\_config = 0xC6C0 — включение последним

## 9. Пример настройки для подключения сельсин

Отличия от примера для СКВТ (раздел 8):

**AFE\_config** (адрес 70) = 0x4003

- **Mode**[14:12] = 100 — режим сельсин

**Mode\_config** (адрес 71) = 0xCEC0

- **CONV2\_en**[11] = 1 — оба преобразователя включены

**C2InputStngs** (адрес 41) = 0x0810 — аналогично C1InputStngs

**C2KonturStngs** (адрес 45) = 0x000E — аналогично C1KonturStngs

**C2ResCntrl** (адрес 46) = 0x00E2 — аналогично C1ResCntrl

**C2Mask** (адрес 44) = 0xFFFFE

**Mask\_Stat** (адрес 66) = 0x001F

**C1EXInc** (адрес 7) — зависит от требуемой частоты возбуждения. Для сельсина с частотой возбуждения 50 Гц:  $EXInc = 50 \times 2^{20} / 1250000 \approx 42$ .

Подключение обмоток: **IOSA1** =  $\sin(\theta + 0^\circ)$ , **IOCA1** =  $\sin(\theta + 120^\circ)$ , **IOSA2** =  $\sin(\theta - 120^\circ)$ . Вывод **IOCA2** не используется. Допускается подключение только одного сельсина.

#### **PRIMECHANIYE**

Для сельсина с низкой частотой возбуждения (например, 50 Гц) необходимо установить большое значение **LBW** (узкая полоса пропускания), чтобы фильтр подавлял вторую гармонику до -100 дБ.

## 10. Пример настройки для подключения ЛРДТ

Отличия от примера для СКВТ (раздел 8):

**ЛРДТ с подключением по 5-ти проводной схеме:**

**C1KonturStngs** (адрес 13) = 0x040E

- `Sensor_mode[11:10] = 01` — ЛРДТ 5-проводной

**ЛРДТ с последовательным соединением обмоток:**

`C1KonturStngs` (адрес 13) = `0x080E`

- `Sensor_mode[11:10] = 10` — ЛРДТ с последовательным соединением

**Особенности настройки ЛРДТ:**

- Для снятия ошибки `UIN_HIGH` увеличьте `C1Amp_th.Amp_lth` (рекомендуемое значение 1000).
- Выходная координата — знаковое значение в диапазоне от  $-1$  до  $+1$  (в отличие от угла  $0...2\pi$  для СКВТ).
- Для ЛРДТ с последовательно соединёнными обмотками (`Sensor_mode = 10`) критично совпадение фаз сигналов `IOSA1` и `IOCA1` (см. раздел 5.b).

## 11. Блокировка записи и инициализация из ПЗУ

а. После завершения настройки всех регистров запишите ненулевое значение в регистр `WR_lock` (адрес 68). Это предотвратит случайное изменение конфигурации. При `WR_lock ≠ 0` команды записи не исполняются, за исключением регистра `BUS_addr` (адрес 83). Для разблокировки запишите 0.

б. Для серийного производства конфигурация может быть сохранена в блок OTP-памяти (ВОТР, 512×16 бит). При установке `INIT_conf.OTP_init_on` (бит 4) = 0 конфигурация из ВОТР загружается автоматически при каждом сбросе микросхемы. Подробнее см. раздел ПЗУ.

Последнее обновление **17 июн. 2026 г.** от *Ilya Slepchenkov*

# Описание ПО

Последнее обновление **17 июн. 2026 г.** от **Ilya Slerchenkov**