**25.1 Введение**

**25.1.1 Назначение периферии**

Интерфейс общего назначения объединяет четыре модуля ввода/вывода общего назначения (GPIO). Каждый GPIO модуль предоставляет 32 выделенных контакта общего назначения с возможностями ввода и вывода; таким образом, интерфейс общего назначения поддерживает до 128 (4×32) контактов. Эти контакты могут быть настроены для следующих приложений:

- Ввод данных (захват)/вывод (управление)

- Клавиатурный интерфейс с антидребезгом

- Генерация прерываний в активном режиме при обнаружении внешних событий. Обнаруженные события обрабатывается двумя параллельными независимыми субмодулями генерации прерываний для поддержки бипроцессора операции.

- Формирование запроса на пробуждение в режиме ожидания при обнаружении внешних событий.

**25.1.2 Функции GPIO**

Каждый модуль GPIO состоит из 32 одинаковых каналов. Каждый канал можно настроить для использования в следующих приложениях:

- Ввод/вывод данных

- Клавиатурный интерфейс с ячейкой, предотвращающей дребезг

- Генерация синхронного прерывания (в активном режиме) при обнаружении внешних событий (сигнал переход(ы) и/или уровень(и) сигнала))

- Генерация запроса на пробуждение (в режиме ожидания) при обнаружении перехода (ов) сигнала.

Глобальные особенности интерфейса GPIO:

- Синхронные запросы на прерывание от каждого канала обрабатываются двумя идентичными генераторами прерываний. Подмодули для независимого использования подсистемой ARM

- Запросы на пробуждение от входных каналов объединяются, чтобы выдать системе один сигнал пробуждения.

- Доступ к общим регистрам можно получить по протоколу «Set & Clear».

**25.1.3 Неподдерживаемые функции GPIO**

Функция пробуждения модулей GPIO поддерживается только на GPIO0.

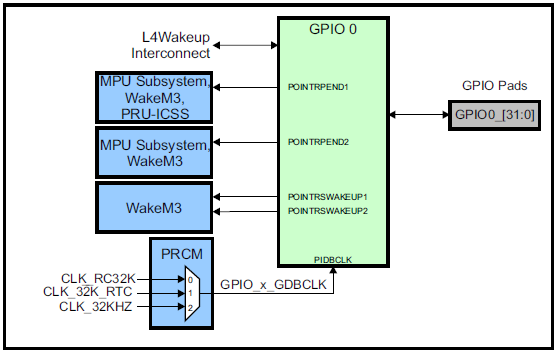
**25.2 Интеграция**

Устройство создает четыре модуля GPIO\_V2. Каждый модуль GPIO обеспечивает поддержку 32 выделенных пина с возможностью конфигурации ввода и вывода. Входные сигналы могут использоваться для генерации прерываний и сигнала пробуждения. Для двухпроцессорной работы доступны две линии прерывания. Пины могут использоваться в качестве контроллера клавиатуры.

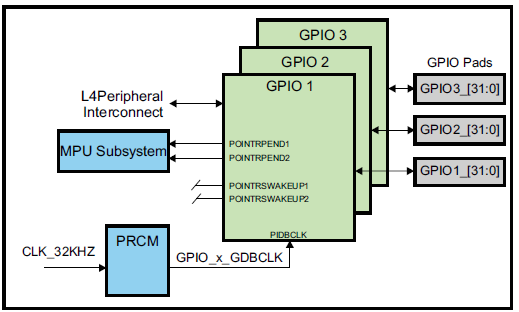
С четырьмя модулями GPIO устройство поддерживает максимум 128 контактов GPIO. (точное количество зависит от конфигурации устройства и мультиплексирования контактов.) GPIO0 находится в домене пробуждения (Wakeup domain) и может использоваться для пробуждения устройства через внешние источники. GPIO[1:3] расположены в периферийном домене.

Рисунок 25-1 и Рисунок 25-2 показывают интеграцию GPIO.

**Рисунок 25-1. Интеграция модуля GPIO0**



**Рисунок 25-2. Интеграция модуля GPIO[1–3]**



**25.2.1 Атрибуты подключения GPIO**

Общие атрибуты подключения для модулей GPIO в устройстве показаны в Таблице 25-1 и Таблице 25-2.

**Таблица 25-1. Атрибуты подключения GPIO0**

|  |  |
| --- | --- |
| Attributes | Type |
| Power Domain | Wakeup Domain |
| Clock Domain | PD\_WKUP\_L4\_WKUP\_GCLK (OCP)  GPIO\_0\_GDBCLK (Debounce) |
| Reset Signals | WKUP\_DOM\_RST\_N |
| Idle/Wakeup Signals | Smart Idle / Slave Wakeup |
| Interrupt Requests | Два прерывания:  INTRPEND1 (GPIOINT0A) to MPU subsystem, PRU-ICSS  (POINTRPEND1), and WakeM3  INTRPEND2 (GPIOINT0B) to MPU subsystem and WakeM3 |
| DMA Requests | Запросы на прерывание перенаправляются как запросы DMA: 1 DMA request (GPIOEVT0) |
| Physical Address | L4 Wakeup slave port |

**Таблица 25-2. Атрибуты подключения GPIO[1:3]**

|  |  |
| --- | --- |
| Attributes | Type |
| Power Domain | Peripheral Domain |
| Clock Domain | PD\_PER\_L4LS\_GCLK (OCP)  GPIO\_1\_GDBCLK (GPIO1 Debounce)  GPIO\_2\_GDBCLK (GPIO2 Debounce)  GPIO\_3\_GDBCLK (GPIO3 Debounce) |
| Reset Signals | PER\_DOM\_RST\_N |
| Idle/Wakeup Signals | Smart Idle |
| Interrupt Requests | Два прерывания:  INTRPEND1 (GPIOINTxA) to MPU subsystem  INTRPEND2 (GPIOINTxB) to MPU subsystem |
| DMA Requests | Запросы на прерывание перенаправляются как запросы DMA:1 DMA  request only for GPIO1 (GPIOEVT1) and GPIO2 (GPIOEVT2) |
| Physical Address | L4 Peripheral slave port |

**25.2.2 Тактирование GPIO и управление сбросом**

Для модулей GPIO требуется два тактовых генератора: тактовый сигнал устранения дребезга используется для ячеек устранения дребезга. Интерфейсное тактирование, обеспечиваемое периферийной шиной (интерфейс L4), также являются функциональными тактовыми импульсами и используются во всем модуле GPIO (кроме подмодуля подавления дребезга). Он синхронизирует интерфейс OCP и внутреннюю логику. Тактирование GPIO0 для антидребезга выбирается из одного из трех источников с помощью регистра **CLKSEL\_GPIO0\_DBCLK** в **PRCM**:

- Встроенный генератор частоты 32,768 кГц (CLK\_RC32K)

- PER PLL генерирует тактовую частоту 32,768 кГц (CLK\_32KHZ).

- Внешний генератор/такты 32,768 кГц (CLK\_32K\_RTC)

**Таблица 25-3. Тактовые сигналы GPIO**

|  |  |  |  |
| --- | --- | --- | --- |
| Clock Signal | Max Freq | Reference / Source | Comments |
| GPIO0 | | | |
| Functional / Interface clock | 100 MHz | CORE\_CLKOUTM4 / 2 | pd\_wkup\_l4\_wkup\_gclk  From PRCM |
| Debounce Functional clock | 32.768 KHz | CLK\_RC32K  CLK\_32KHZ  (PER\_CLKOUTM2 / 5859.375)  CLK\_32K\_RTC | pd\_wkup\_gpio0\_gdbclk  From PRCM |
| GPIO[1:3] | | | |
| Functional / Interface clock | 100 MHz | CORE\_CLKOUTM4 / 2 | pd\_per\_l4ls\_gclk  From PRCM |
| Debounce Functional clock  (GPIO1) | 32.768 KHz | CLK\_32KHZ  (PER\_CLKOUTM2 / 5859.375) | pd\_per\_gpio\_1\_gdbclk  From PRCM |
| Debounce Functional clock  (GPIO2) | 32.768 KHz | CLK\_32KHZ  (PER\_CLKOUTM2 / 5859.375) | pd\_per\_gpio\_2\_gdbclk  From PRCM |
| Debounce Functional clock  (GPIO3) | 32.768 KHz | CLK\_32KHZ  (PER\_CLKOUTM2 / 5859.375) | pd\_per\_gpio\_3\_gdbclk  From PRCM |

**25.2.3 Список выводов GPIO**

Каждый модуль GPIO включает 32 интерфейса ввода/вывода. Эти сигналы обозначаются, как показано в таблице 25-4.Обратите внимание, что для этого устройства большинство этих сигналов будут мультиплексированы с функциональными сигналами от других интерфейсов.

**Таблица 25-4. Список выводов GPIO**

|  |  |  |
| --- | --- | --- |
| Pin | Type | Description |
| GPIO0\_[31:0] | I/O | General Purpose Input-Output pins |
| GPIO1\_[31:0] |
| GPIO2\_[31:0] |
| GPIO3\_[31:0] |
| GPIO4\_[31:0] |
| GPIO5\_[31:0] |

**25.3 Функциональное описание**

В этом разделе обсуждаются рабочие детали и основные функции периферийного устройства GPIO.

**25.3.1 Режимы работы**

Для модуля определены четыре режима работы:

- Активный режим(**Active mode**): модуль работает синхронно с интерфейсным тактированием интерфейса, может генерироваться прерывание в соответствии с конфигурацией и внешними сигналами.

- Режим ожидания(**Idle mode**): модуль находится в состоянии ожидания, интерфейсное тактирование может быть остановлено, прерывания невозможны,сигнал пробуждения может быть сгенерирован в соответствии с конфигурацией и внешними сигналами. Проверьте функциональную спецификацию верхнего уровня чипа на предмет доступности тактирования устранения дребезга в режиме ожидания. Если такты подавления дребезга активны, ячейка устранения дребезга может использоваться для выборки и фильтрации входа генерации события пробуждения. В противном случае (тактирование антидребезга неактивно) ячейка устранения дребезга не может использоваться, так как она стробирует все входные сигналы.

- Неактивный режим(**Inactive mode**): модуль неактивен, тактирование интерфейса может быть остановлено, прерывание не может быть сгенерировано, а функция пробуждения запрещена.

- Отключенный режим(**Disabled mode**): модуль не используется, внутренние тактовые пути закрыты, запрос на прерывание или пробуждения не может быть сформирован.

Режимы **Idle** и **Inactive** настраиваются внутри модуля и активируются по запросу хоста через сигналы боковой полосы системного интерфейса. Режим **Disabled** устанавливается программно, через специальный бит конфигурации. Он безоговорочно блокирует внутренние пути синхронизации, которые не используются интерфейсом системы. Все регистры модуля являются 8-, 16- или 32-битными, доступными через OCP-совместимый интерфейс (little endian). В активном(**active**) режиме обнаружение события (уровня или перехода) выполняется в GPIO модуле с помощью интерфейсного тактирования. Точность обнаружения определяется частотой этих тактовых импульсов и выбранной схемой внутреннего стробирования.

**25.3.2 Тактирование и стратегия сброса**

**25.3.2.1 Тактирование**

Модуль GPIO работает с использованием двух источников тактирования:

- Тактирование антидребезга используются для логики субмодуля устранения дребезга (без соответствующих регистров конфигурации). Этот модуль может сэмплировать входную линию и фильтровать входной уровень, используя запрограммированную задержку.

- Интерфейсное тактирование, обеспечиваемое периферийной шиной (системный интерфейс, совместимый с OCP). Это используется во всем модуле GPIO (за исключением логики подмодуля антидребезга). Оно тактирует OCP интерфейс и внутреннюю логику. Функции управления синхронизацией позволяют адаптировать энергопотребление модуля к активности.

**25.3.2.2 Определение тактирования, стробирования и активного фронта**

Интерфейсный тактовый сигнал, обеспечиваемый периферийной шиной (системный интерфейс, совместимый с OCP), используется через весь модуль GPIO. Определены два домена тактирования: интерфейс OCP и внутренняя логика. Каждый тактирующий домен может управляться независимо. Операции выборки для сбора данных и событий обнаружение существляется с использованием нарастающего фронта. Данные, загруженные в регистр вывода данных (GPIO\_DATAOUT), устанавливается на выходных контактах GPIO синхронно с нарастающим фронтом тактового сигнала интерфейса.

Доступны пять функций стробирования тактирования:

- Тактирование для логики системного интерфейса может быть стробировано, когда к модулю нет доступа, если AUTOIDLE бит конфигурации в регистре конфигурации системы (GPIO\_SYSCONFIG) установлен. В противном случае эта логика свободно работает на интерфйсном тактировании.

- Тактирование для логики выборки входных данных может быть стробировано, когда доступ к данным в регистре не осуществляется.

- Четыре группы тактирования используются для логики обнаружения синхронных событий. Каждый 8-й вход GPIO\_V2 контактов микросхемы будет иметь отдельный сигнал включения в зависимости от настройки регистра обнаружения фронта/уровня. Если группа не требует обнаружения, тогда соответствующее тактирование будет закрыто. Все каналы также закрываются по схеме «один из N». N может принимать значения 1, 2, 4 или 8. Интерфейсное тактирование включается для этой логики один цикл каждые N циклов. Когда N равно 1, стробирование отсутствует и эта логика свободно работает на интерфейсном тактировании. Когда N находится в диапазоне от 2 до 8, эта логика работает со скоростью равной частоте тактовой частоте интерфейса, деленной на N.

- В неактивном(**inactive**) режиме все внутренние каналы синхронизации закрыты.

- В режиме **Disabled** все внутренние пути синхронизации, не используемые для системного интерфейса, блокируются. Все GPIO регистры доступны синхронно с интерфейсным тактированием.

**25.3.2.3 Запрос и подтверждение спящего режима**

По запросу спящего режима, выданному хост-процессором, модуль GPIO переходит в режим ожидания в соответствии с полем IDLEMODE в регистре конфигурации системы (GPIO\_SYSCONFIG).

• **IDLEMODE** = 0 (режим **Force-Idle**): GPIO переходит в неактивный режим независимо от внутреннего состояние модуля и безоговорочно отправляется подтверждение бездействия. В режиме **Force-Idle** модуль находится в неактивный режиме и его функция пробуждения полностью запрещена.

• **IDLEMODE** = 1h (режим без простоя): GPIO не переходит в режим бездействия, и подтверждение бездействия никогда не отправляется.

• **IDLEMODE** = 2h (режим **Smart-Idle**) или **IDLEMODE** = 3h (режим **Smart**-**Idle**): модуль GPIO оценивает свою внутреннюю способность отключать тактирование интерфейса. Когда больше нет внутренней активности (регистр ввода данных завершен для захвата входных контактов GPIO, нет ожидающих прерываний, все биты состояния прерывания сброшены, и нет доступа для записи в регистр GPIO\_DEBOUNCINGTIME ожидания синхронизации), подтверждается бездействие, и GPIO переходит в режим бездействия, готовый для выдачи запроса на пробуждение, когда ожидаемый переход происходит на включенном входном контакте GPIO. Этот запрос на пробуждение эффективно отправляется только в том случае, если бит **ENAWAKEUP** в **GPIO**\_**SYSCONFIG** установлен в положение **Enable**. Когда система пробуждается, запрос на бездействие становится неактивным, сигнал запроса на подтверждение бездействия и запроса на пробуждение (если именно GPIO инициировал пробуждение системы) немедленно отменяется, а асинхронный запрос на пробуждение (если существует) отражается в регистрах состояния синхронного прерывания.

*ПРИМЕЧАНИЕ. Запрос режима ожидания и подтверждение ожидания являются сигналами боковой полосы системного интерфейса. Однажды GPIO подтверждает запрос спящего режима (подтверждение бездействия было отправлено), интерфейсное тактирование может быть остановлено в любое время. После запроса спящего режима, выданного хост-процессором, модуль GPIO переходит в режим ожидания только в том случае, если в регистрах GPIO\_IRQSTATUS\_RAW\_n нет активного бита.*

**25.3.2.4 Сброс**

Сигнал аппаратного сброса OCP имеет действие глобального сброса на GPIO. Все регистры конфигурации, все DFF, синхронизированные с интерфейсным тактированием или тактированием устранения дребезга, и все внутренние конечные автоматы сбрасываются, когда аппаратный сброс OCP активен (низкий уровень). Бит **RESETDONE** в регистре состояния системы

(**GPIO\_SYSSTATUS**) отслеживает состояние внутреннего сброса: он устанавливается, когда сброс завершен на OCP и доменах тактирования антидребезга. Программный сброс (бит SOFTRESET в регистре конфигурации системы) имеет тот же эффект, что и сигнал аппаратного сброса OCP, а бит RESETDONE в GPIO\_SYSSTATUS обновляется в том же состоянии.

**25.3.3 Функции прерывания и пробуждения**

**25.3.3.1 Функциональное описание**

Чтобы сгенерировать запрос на прерывание к хост-процессору при определенном событии (уровень или логический переход) происходит на контакте GPIO, регистры конфигурации GPIO должны быть запрограммированы следующим образом:

- Прерывание для канала GPIO должно быть включены в регистрах GPIO\_IRQSTATUS\_SET\_0 и/или GPIO\_IRQSTATUS\_SET\_1.

- Ожидаемое(ые) событие(я) на входе GPIO для запуска запроса на прерывание должно быть выбрано в регистрах GPIO\_LEVELDETECT0, GPIO\_LEVELDETECT1, GPIO\_RISINGDETECT и GPIO\_FALLINGDETECT.

Чтобы сгенерировать запрос на пробуждение к хост-процессору при определенном событии (логическом переходе) происходящем на контакте GPIO, регистры конфигурации GPIO должны быть запрограммированы следующим образом:

- Канал GPIO должен быть включен в регистре GPIO\_IRQWAKEN.

- Ожидаемые события на входе GPIO, вызывающие прерывание (или пробуждение), должны быть выбраны в регистрах GPIO\_RISINGDETECT и GPIO\_FALLINGDETECT. Запрос на пробуждение может генерироваться только при повышении и/или при падении переходов. Например, генерация прерывания по обоим фронтам на входе k настраивается установкой в ​​1 k-го бита в регистре GPIO\_RISINGDETECT и GPIO\_FALLINGDETECT вместе с разрешением прерывания для одного или обеих линии прерывания (GPIO\_IRQSTATUS\_SET\_n).

*ПРИМЕЧАНИЕ. Все источники прерываний (или пробуждения) (32 входных канала GPIO) сливаются вместе с двумя синхронным запроса на прерывание 1 и 2 и двумя асинхронными запросами на пробуждение.*

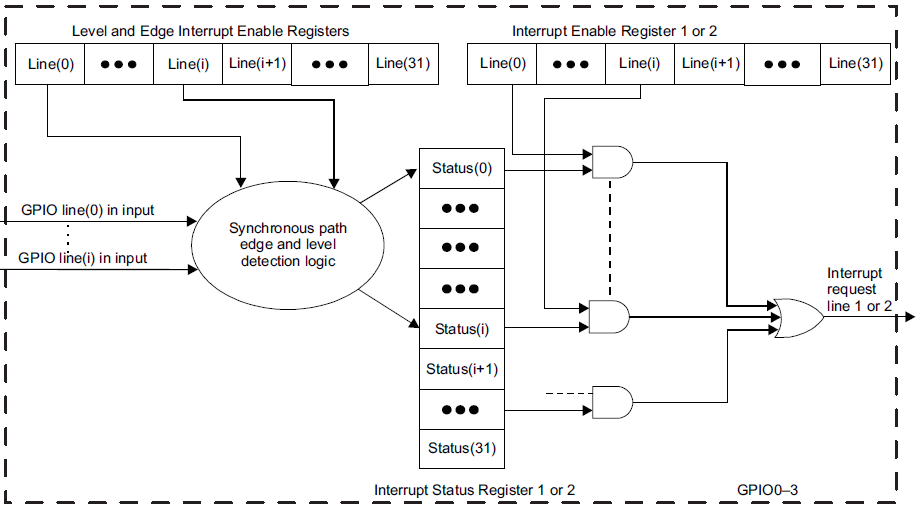
**25.3.3.2 Синхронный способ: генерация запроса прерывания**

В активном режиме после установки регистров конфигурации GPIO для активизации генерации прерываний синхронный тракт (рис. 25-3) выполняет выборку переходов и уровней на входном GPIO с внутренним тактовым сигналом стробируемого интерфейса. Когда событие соответствует запрограммированным настройкам, соответствующий бит в регистре GPIO\_IRQSTATUS\_RAW\_n установливается в 1, и в следующем такте интерфейса прерывание активируют линии 1 и/или 2 (в зависимости от GPIO\_IRQSTATUS\_SET\_n регистров).

Благодаря операции дискретизации минимальная длительность импульса на входном GPIO для запуска синхронного запроса на прерывание в два раза превышает тактовый период внутреннего интерфейса (тактовый сигнал внутреннего интерфейса) период равен N, умноженному на тактовый период интерфейса). Эта минимальная длительность импульса должна быть удовлетворена до и после любого обнаружения ожидаемого перехода уровня. Обнаружение уровня требует, чтобы выбранный уровень был стабильным для по меньшей мере, двойного превышения тактового периода внутреннего интерфейса для запуска синхронного прерывания.

Поскольку модуль является синхронным, задержка минимальна между ожидаемым возникновением события и активизацией линии (линий) прерывания. Это не должно превышать 3 внутренних тактовых цикла интерфейса + 2 тактовых цикла интерфейса, когда функция антидребезга не используется. Когда функция антидребезга активна, задержка зависит от значения регистра GPIO\_DEBOUNCINGTIME и должна быть меньше 3-х внутренних тактовых циклов стробируемого интерфейса + 2 интерфейсных тактовых цикла + GPIO\_DEBOUNCINGTIME тактовых циклов антидребезга + 3 тактовых цикла на антидребезга.

**Рис. 25-3. Создание запроса на прерывание**



**25.3.3.3 Асинхронный способ: создание запроса на пробуждение**

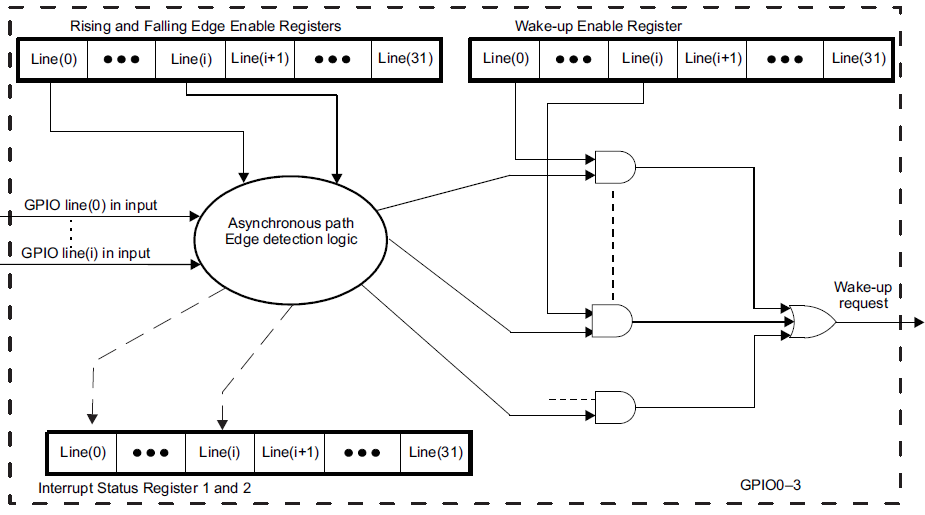
В режиме ожидания (тактовый сигнал интерфейса выключен, регистры конфигурации GPIO были ранее запрограммированы), асинхронный тракт (рис. 25-4) обнаруживает ожидаемый переход (и) на входном GPIO (в соответствии с программированием регистров) и активирует асинхронный запрос пробуждения, если пробуждение установлена регистре включения. Как видно , существует только одна внешняя линия пробуждения, так как все источники пробуждения слились вместе. Как только система входит в состояние пробуждения, интерфейсное тактирование запускается повторно и в соответствии с входным контактом GPIO, который инициировал запрос пробуждения, соответствующие биты в регистре **GPIO\_IRQSTATUS\_RAW\_n** синхронно устанавливаются в 1; на следующем внутреннем такте, линии 1 и/или 2 прерываний активны (активны на высоком уровне), когда соответствующие биты в **GPIO\_IRQSTATUS\_SET\_n** регистры установлены.

*ПРИМЕЧАНИЕ: Если антидребезг не включен, минимальная длительность входного импульса для запуска запроса на пробуждения отсутствует из-за отсутствия операции выборки.*

*Если используется антидребезг, минимальная длительность импульса устанавливается указанным временем антидребезга.*

*Бит ENAWAKEUP в регистре конфигурации системы (GPIO\_SYSCONFIG) включает или отключает функцию пробуждения GPIO глобально: если этот бит равен 0, GPIO\_IRQWAKEN не имеет эффекта.*

**Рис. 25-4. Создание запроса на пробуждение**



**25.3.3.4 Отпускание линии прерывания (или пробуждения)**

Когда хост-процессор получает запрос на прерывание, выданный модулем GPIO, он может считывать соответствующий регистр GPIO\_IRQSTATUS\_n, чтобы узнать, какой входной GPIO вызвал прерывание (или запрос на пробуждение). После обслуживания прерывания (или подтверждения запроса на пробуждение) процессор сбрасывает бит состояния и освобождает линию прерывания, записывая 1 соответствующий бит в регистр GPIO\_IRQSTATUS\_n. Если еще есть ожидающий запрос на прерывание для обслуживания (все биты в GPIO\_IRQSTATUS\_RAW\_n регистр не замаскированы PIO\_IRQSTATUS\_SET\_n, которые очищены путем установки GPIO\_IRQSTATUS\_CLR\_n) линия прерывания будет восстановлена.

**25.3.4 Базовая модель программирования интерфейса общего назначения**

**25.3.4.1 Энергосбережение за счет группирования обнаружения фронтов /уровней сигнала**

Каждый модуль GPIO реализует четыре стробированных тактовых сигнала, используемых логикой обнаружения границ/уровней для экономии энергии. Каждая группа из восьми входных контактов GPIO генерирует отдельный разрешающий сигнал в зависимости от границы/уровня установка регистра обнаружения (поскольку входной сигнал равен 32 битам, для каждого из них определены четыре группы из восьми входных сигналов модуля GPIO). Если группа не требует обнаружения границ/уровней, то соответствующий тактовый сигнал стробируется (отключается).

Группировка enable edge/level позволяет сэкономить потребляемую мощность модуля, как описано в разделе в следующем примере.

Если какой-либо из регистров:

- GPIO\_LEVELDETECT0

- GPIO\_LEVELDETECT1

- GPIO\_RISINGDETECT

- GPIO\_FALLINGDETECT

установлен в 0x0101\_0101, то все тактовые импульcы активны (энергопотребление высокое); если установлены в значение 0x0000\_00FF, то один тактовый сигнал активен.

*ПРИМЕЧАНИЕ: Когда тактирование включено путем записи в регистры GPIO\_LEVELDETECT0,GPIO\_LEVELDETECT1, GPIO\_RISINGDETECT и GPIO\_FALLINGDETECT, обнаружение начинается после 5 тактовых циклов. Этот период необходим для очистки пайплайна синзронизации обнаружения фронта/уровня.*

*Механизм не зависит от каждой группы синхросигналов. Если тактирование было запущено до выполнения новой настройка, рекомендуется: сначала установить необходимое новое обнаружение; во-вторых, отключить предыдущую настройку (при необходимости). Таким образом, соответствующий тактовый сигнал не стробируется, и обнаружение начинается немедленно.*

**25.3.4.2 Инструкции по установке и очистке**

Модуль GPIO реализует обновление регистров протоколов **set-and-clear** для вывода данных и разрешения прерывания и регистров включения пробуждения. Этот протокол является альтернативой атомарному тестированию и установочным операциям и состоит из операций записи по выделенным адресам (один адрес для установки бита [ов] и один адрес для сброса бита [ов]). Данные для записи равны 1 в битовых позициях для очистки (или для установки) и 0 в незатронутых битах.

Доступ к регистрам возможен двумя способами:

- Стандарт: Операции чтения и записи полного регистра по адресу первичного регистра

- Установить(**set**) и очистить(**clear**) (рекомендуется): Предусмотрены отдельные адреса для установки (и очистки) битов в регистрах. Запись 1 по этим адресам устанавливает (или очищает) соответствующий бит в эквивалентном регистре; запись 0 не имеет эффекта.

Поэтому для этих регистров определены три адреса для одного уникального физического регистра. Чтение этих адресов имеют тот же эффект и возвращает значение регистра.

**25.3.4.2.1 Инструкции по очистке**

**25.3.4.2.1.1 Очистка регистров включения прерываний (GPIO\_IRQSTATUS\_CLR\_0 и GPIO\_IRQSTATUS\_CLR\_1):**

- Операция записи в регистре clear interrupt enable1 (или enable2) очищает соответствующий бит в enable1 (или enable2) регистре прерывания, когда записанный бит равен 1; записанный бит в 0 не имеет эффекта.

- Считывание регистра clear interrupt enable1 (или enable2) возвращает значение параметра в регистре interrupt enable1 (или enable2).

**25.3.4.2.1.2 Очистка регистра включения пробуждения (GPIO\_CLEARWKUENA):**

- Операция записи в регистре разрешения сброса пробуждения очищает соответствующий бит в регистре включения пробуждения, когда записанный бит равен 1; бит, записанный в 0, не дает эффекта.

- Считывание открытого регистра разрешения пробуждения возвращает значение регистра разрешения пробуждения.

**25.3.4.2.1.3 Очистка регистр вывода данных (GPIO\_CLEARDATAOUT):**

- Операция записи в регистре вывода очищенных данных очищает соответствующий бит в регистре выводе данных, когда записанный бит равен 1; бит, записанный в 0, не дает эффекта.

- Считывание открытого выходного регистра данных возвращает значение выходного регистра данных.

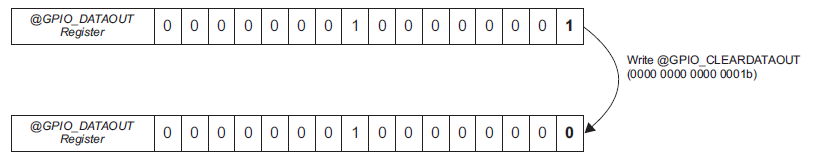
**25.3.4.2.1.4 Пример команды очистки**

Предположим, что регистр вывода данных (или один из регистров разрешения прерывания/пробуждения) содержит двоичное значение 0b0000\_0001\_0000\_0001 и требуется очистка бита 0.

С помощью функции команды **clear** запишите 0b0000\_0000\_0000\_0001 по адресу регистра вывода данных **clear** (или по адресу регистра разрешения сброса прерывания/пробуждения). После этой операции записи считывание выходного регистра данных (или регистра разрешения прерывания/пробуждения) возвращает 0b0000\_0001\_0000\_0000; бит 0 очищен.

*ПРИМЕЧАНИЕ: Несмотря на то, что регистры интерфейса общего назначения имеют ширину 32 бита, в этом примере представлены 16 значащих.*

**Рисунок 25-5. Пример регистра записи @ GPIO\_CLEARDATAOUT**



**25.3.4.2.2 Установка битов**

**25.3.4.2.2.1 Установка регистров включения прерываний (GPIO\_IRQSTATUS\_SET\_0 и GPIO\_IRQSTATUS\_SET\_1):**

- Операция записи в регистре **interrupt** **enable1** (или **enable2**) устанавливает соответствующий бит в регистре **interrupt enable1** (или **enable2**), когда записанный бит равен 1; запись бита в 0 не имеет эффекта.

- Считывание регистра set interrupt enable1 (или enable2) возвращает значение регистра interrupt enable1 (или enable2).

**25.3.4.2.2.2 Реестр включения пробуждения (GPIO\_SETWKUENA):**

- Операция записи в регистре разрешения пробуждения устанавливает соответствующий бит в регистре разрешения пробуждения, когда записанный бит равен 1; запись бита в 0 не имеет эффекта.

- Считывание установленного регистра разрешения пробуждения возвращает значение регистра разрешения пробуждения.

**25.3.4.2.2.3 Установка регистра вывода данных (GPIO\_SETDATAOUT):**

- Операция записи в заданном выходном регистре данных устанавливает соответствующий бит в выходном регистре данных когда записанный бит равен 1; запись бита в 0 не имеет эффекта.

- Считывание заданного выходного регистра данных возвращает значение выходного регистра данных.

**25.3.4.2.2.4 Пример инструкции по установке**

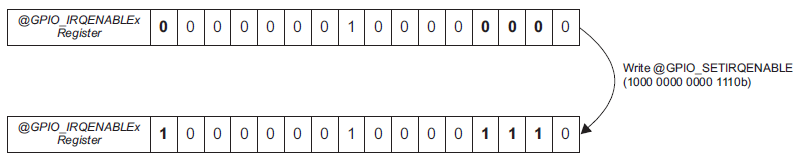
Предположим, что регистр interrupt enable1 (или enable2) (или регистр вывода данных) содержит двоичное значение, 0b0000\_0001\_0000\_0000, и необходимо установить биты 15, 3, 2 и 1.

С помощью функции команды set запишите 1000 0000 0000 1110h по адресу активизации регистра interrupt enable1(или enable2) (или по адресу установленного выходного регистра данных). После этой операции записи выполняется считывание регистра с interrupt enable1 (или enable1) (или регистра вывода данных) возвращает 0b1000\_0001\_0000\_1110;

биты 15, 3, 2 и 1 установлены.

ПРИМЕЧАНИЕ: Несмотря на то, что регистры интерфейса общего назначения имеют ширину 32 бита, только 16 незначительныхв этом примере представлены биты.

**Рисунок 25-6. Пример регистра записи @ GPIO\_SETIRQENABLEx**



**25.3.4.3 Ввод (захват )/вывод (привод) данных**

Регистр разрешения выхода (GPIO\_OE) управляет возможностью вывода/ввода для каждого контакта. При сбросе все контакты, связанные с GPIO и сконфигурированные как ввод или вывод, деактивированы. Этот регистр не используется внутри модуля; его единственной функцией является перенос конфигурации пинов.

При конфигурировании в качестве выходного сигнала (требуемый бит сбрасывается в GPIO\_OE), значение соответствующего бита в регистре GPIO\_DATAOUT включается на соответствующий контакт GPIO. Данные записываются в регистр выходных данных синхронно с тактовым сигналом интерфейса. Доступ к этому регистру можно получить с помощью операций чтения/записи или с помощью функции обновления альтернативного набора и очищенного протокольного регистра. Эта функция позволяет установить или очистить конкретные биты этого регистра с одним доступом на запись к установленному регистру вывода данных (GPIO\_SETDATAOUT) или в адрес регистра вывода чистых данных (GPIO\_CLEARDATAOUT). Если приложение использует контакт в качестве выхода и не хочет, чтобы происходило прерывание/пробуждение от этого контакта, приложение должно правильно сконфигурировать регистры активизации пробуждения и активизации прерывания.

При конфигурировании в качестве входа (требуемый бит установлен в 1 GPIO\_OE), состояние входа может быть считано из соответствующего бита в регистре GPIO\_DATAIN. Входные данные отбираются синхронно с интерфейсным тактированием и затем регистрируется в регистре ввода данных синхронно с интерфейсным тактированием. Когда уровни контактов GPIO изменяются, они фиксируются в этом регистре после двух тактовых циклов интерфейса (необходимые циклы для синхронизации и записи данных). Если приложение использует контакт в качестве входа, приложение должно правильно сконфигурировать активизацию пробуждения и регистры активизации прерывания для прерывания и пробуждения по мере необходимости.

**25.3.4.4 Время антидребезга**

Для активизации функции антидребезга для контакта регистры конфигурации GPIO должны быть запрограммированы следующим образом:

- Контакт GPIO должен быть сконфигурирован как вход в регистре разрешения выхода (запишите 1 в соответствующий бит регистра GPIO\_OE).

- Время антидребезга должно быть установлено в соответствующем регистре (GPIO\_DEBOUNCINGTIME).

Регистр GPIO\_DEBOUNCINGTIME используется для установки времени антидребезга для всех входных линий в модуля GPIO. Значение является глобальным для всех портов одного модуля GPIO, поэтому возможно установить до шести различных значнеий антидребезга. Ячейка отладки работает с тактовым сигналом отладки (32 кГц). Этот регистр представляет количество используемых тактовых циклов (один цикл составляет 31 микросекунду).

Следующая формула описывает необходимое время стабильности ввода, которое должно быть распространено на антидребезг выходных данных:

*Debouncing time = (DEBOUNCETIME + 1) × 31 μs*

Где значение поля DEBOUNCETIME в регистре GPIO\_DEBOUNCINGTIME находится в диапазоне от 0 до 255.

- Функция антидребезга должна быть включена в регистре включения отладки (запишите 1 в соответствующий бит DEBOUNCEENABLE в регистре GPIO\_DEBOUNCENABLE).

**25.3.4.5 GPIO как интерфейс клавиатуры**

Интерфейс общего назначения может использоваться в качестве интерфейса клавиатуры (рис. 25-7). Вы можете определить каналы на основе матрицы клавиатуры = \* c). На рис. 25-7 показана строка каналов, сконфигурированный как входы с включенной функцией антидребезга входных данных. Строка каналов подтянута к высокому уровню внешним подтягом. Колонка каналов конфигурируются как выходы и управляют низким уровнем.

При нажатии клавиши матрицы клавиатуры соответствующие строки строк и столбцов замыкаются и низкий уровень приводится в действие на соответствующем канале строки. При этом генерируется прерывание на основе соответствующего конфигурации (см. раздел 25.3.3).

При получении прерывания клавиатуры процессор может отключить прерывание клавиатуры и отсканировать каналы столбцов для ключевых координат.

- Последовательность сканирования имеет столько состояний, сколько каналов столбцов: Для каждого шага в последовательности, процессор приводит в движение один канал столбца на низком уровне, а другие - на высоком.

- Процессор считывает значения каналов строк и таким образом определяет, какие клавиши в столбце нажимают.

В конце последовательности сканирования процессор устанавливает, какие клавиши нажимаются. Интерфейс клавиатуры затем может быть переконфигурирован в состоянии ожидания прерывания.

**Рисунок 25-7. Интерфейс общего назначения, используемый в качестве интерфейса клавиатуры**

