**20.2 DMTimer 1ms**

**20.2.1 Введение**

Эта периферия представляет собой 32-битный таймер, предлагающий:

• Таймер счетчика с режимами сравнения и захвата

• Режим автоматической перезагрузки

• Режим старт-стоп

• Генерировать 1 мс тик с частотой 32768-Hz фукционального тактирования.

• Программируемый источник тактового сигнала делителя

• 16-32-битная адресация

• Регистры чтения/записи на лету

• Прерывания, генерируемые при переполнении, сравнении и захвате

• Включение прерывания

• Включение пробуждения

• Запись в режиме публикации

• Выделенный входной триггер для режима захвата и выделенный выходной триггер/сигнал ШИМ

• Выделенный выходной сигнал общего назначения PORGPOCFG

• Интерфейс OCP совместим

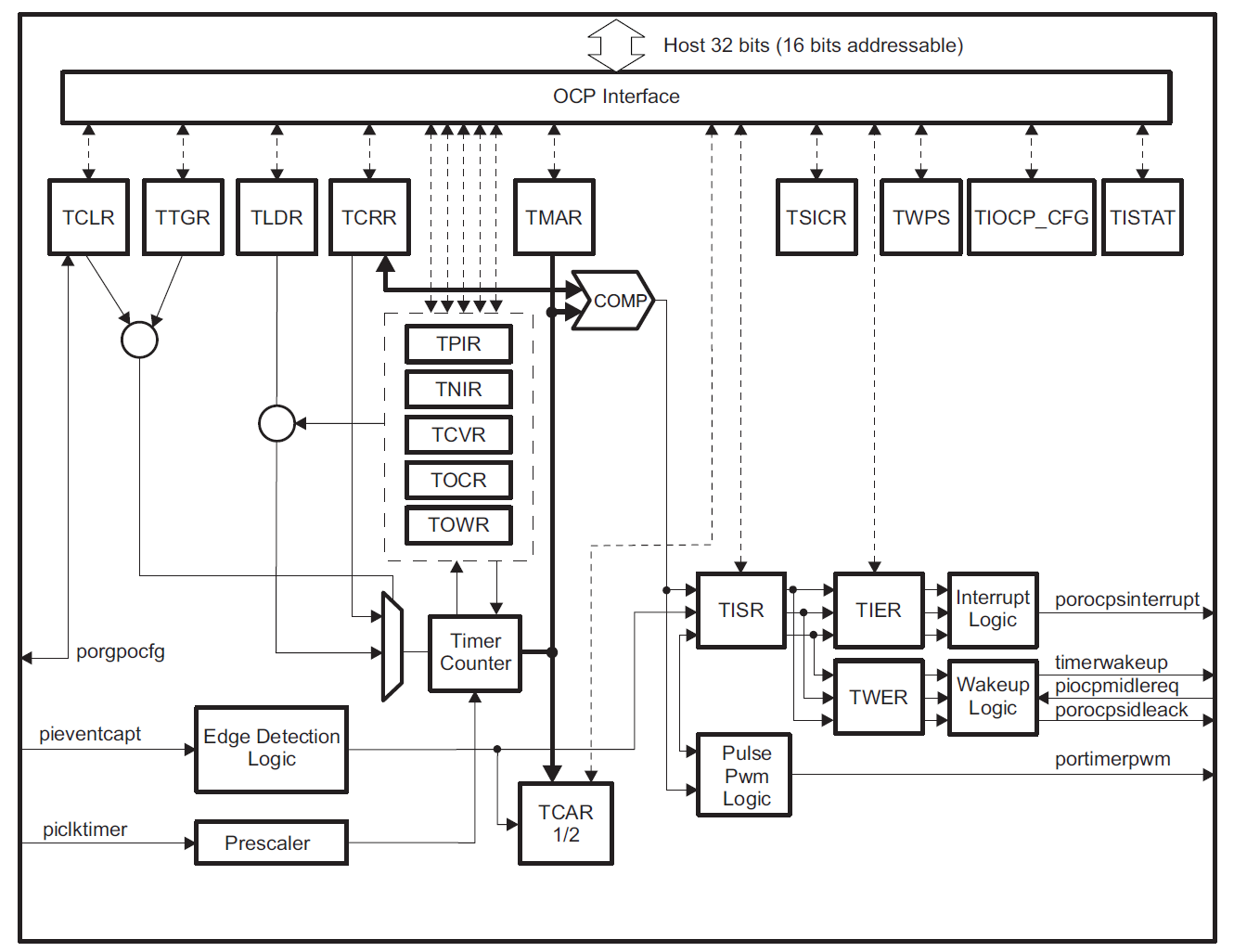
Модуль таймера содержит свободно работающий восходящий счетчик с возможностью автоматической перезагрузки при переполнении. Счетчик таймера может считываться и записываться «на лету» (при подсчете). Модуль таймера включает логику сравнения чтобы разрешить событие прерывания для значения, соответствующего программируемому счетчику.

Выделенный выходной сигнал может быть пульсирован или переключен на событие переполнения и совпадения. Этот вывод предлагает время сигнал триггера метки или источники сигнала ШИМ (широтно-импульсной модуляции). Выделенный выходной сигнал может быть использован для общего назначения PORGPOCFG (непосредственно управляется битом 14 регистра TCLR). Специальный входной сигнал используется для запуска автоматического захвата счетчика таймера и события прерывания на программируемом входе переходного сигнала. Программируемый делитель тактового сигнала (прескалер) позволяет уменьшить входной тактовый сигнал таймера. Все внутренние источники прерываний таймера объединены в одну линию прерываний модуля и одну линию пробуждения . Каждый внутренний источник прерывания может быть независимо включен/отключен с выделенным битом TIER в регистре функции прерывания и выделенным битом TWER для пробуждения.

Управление этим модулем осуществляется через периферийную шину OCP.

Поскольку два тактовых домена управляются внутри этого модуля, ресинхронизация осуществляется специальной логикой между доменом синхросигналов OCP и доменом синхросигналов таймера. При сбросе логика синхронизации позволяет использовать все отношения между тактированием OCP и таймером. Недостатком этого режима является то, что путь полной ресинхронизации используется с влиянием на задержку доступа с точки зрения тактовых циклов OCP. В целях улучшения модуля задержки доступа и в ограниченных условиях по соотношению тактового сигнала (ср. 7.1 Запись размещена), режим размещения записи может использоваться путем установки бита POSTED регистра системного управления (TSICR). В этом режиме запишите posted для включения режима, что означает, что команда записи OCP предоставляется до завершения процесса записи в домен таймера. Этот режим позволяет программному обеспечению (SW) выполнять параллельную запись в регистры таймера двойного режима и наблюдать за завершением (синхронизацией) процесса записи на уровне ПО, читая независимую запись опубликованныч битов состояния в регистре Write Posted Status Register (TWPS).

**Рис. 20-26. Блок-схема**



**20.2.2 Интеграция**

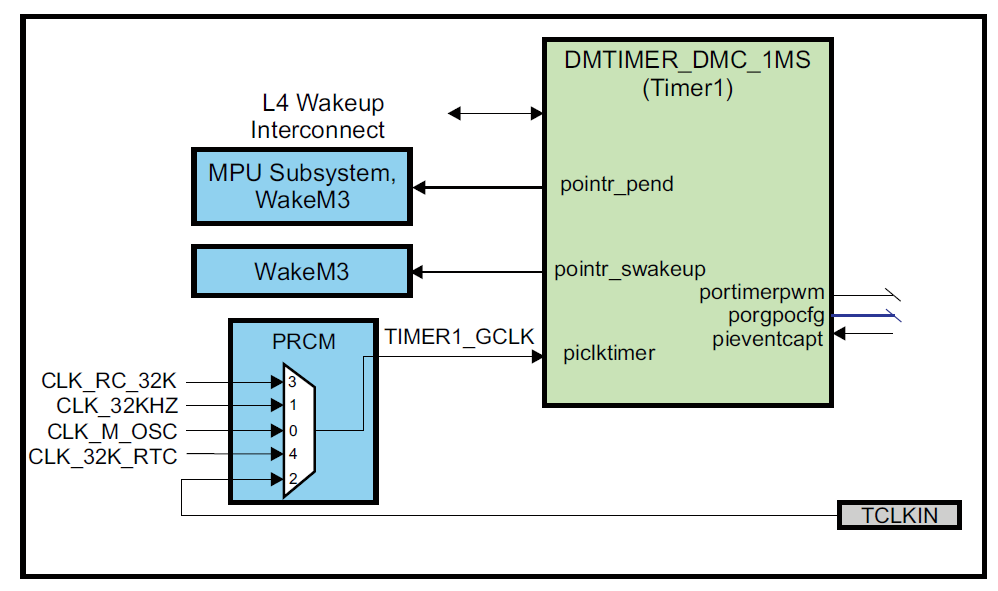
****

Рис. 20-27. Интеграция DMTimer 1 мс

**20.2.2.1 Атрибуты подключения таймера**

Таблица 20-28. Timer1 атрибуты связности

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| Power Domain | Wakeup Domain |
| Clock Domain | PD\_WKUP\_L4\_WKUP\_GCLK (OCP)  PD\_WKUP\_TIMER1\_GCLK (Func) |
| Reset Signals | WKUP\_DOM\_RST\_N |
| Idle/Wakeup Signals | Smart Idle / Slave Wakeup |
| Interrupt Requests | 1 to MPU Subsystem (TINT1\_1MS) and WakeM3 |
| DMA Requests | None |
| Physical Address | L4 Wakeup slave port |

**20.2.2.2 Управление таймером и сбросом**

Функциональное тактирование таймера DMTimer1 1 мс может быть выбрано из одного из пяти источников с помощью СLKSEL\_TIMER1MS\_CLK зарегистрированных в PRCM:

• Системный синхросигнал (CLK\_M\_OSC)

• Плата PER PLL генерирует тактовый сигнал 32,768 кГц (CLK\_32KHZ)

• Входной синхросигнал внешнего таймера TCLKIN

• Встроенный генератор ~ 32,768 кГц (CLK\_RC32K)

• Внешний генератор/тактовый генератор 32,768 кГц (CLK\_32K\_RTC)

**20.2.2.3 Синхросигналы таймера**

Таблица 20-29. Синхросигналы таймера

|  |  |  |  |
| --- | --- | --- | --- |
| **Clock Signal** | **Max Freq** | **Reference / Source** | **Comments** |
| **Timer1 (1ms) Clock Signals** |  |  |  |
| PICLKOCP  Interface clock | 100 MHz | CORE\_CLKOUTM4 / 2 | pd\_wkup\_l4\_wkup\_gclk  from PRCM |
| PICLKTIMER  Functional clock | 26 MHz(1) | CLK\_M\_OSC  CLK\_32KHZ  (PER\_CLKOUTM2 / 5859.375)  TCLKIN  CLK\_RC32K  CLK\_32K\_RTC | pd\_wkup\_timer1\_gclk  from PRCM |

*(1) PICLKTIMER должен быть меньше или равен PICLKOCP/4.*

**20.2.3 Функциональное описание**

Таймер общего назначения представляет собой восходящий счетчик. Он поддерживает три функциональных режима:

• Режим таймера

• Режим захвата

• Режим сравнения

По умолчанию после сброса ядра режимы захвата и сравнения отключены.

**20.2.3.1 Функции режима таймера**

Таймер представляет собой восходящий счетчик, который может быть запущен и остановлен в любое время через управление таймером в регистре (бит TCLR ST). Регистр счетчика таймера (TCRR) может быть загружен при остановке или на лету (при подсчете). TCRR может быть загружен непосредственно с помощью доступа записи TCRR с новым значением таймера. TCRR также может быть загружено значением, сохраненным в регистре загрузки таймера (TLDR) триггерным регистром (TTGR) доступа на запись. Загрузка TCRR выполняется независимо от значения, записанного в регистр TTGR. Таймер значение регистра счетчика TCRR может считываться, когда оно остановлено или захвачено «на лету» доступом чтения TCRR.

Таймер останавливается, и значение счетчика устанавливается в «0» при подтверждении сброса модуля. Таймер удерживается в состоянии остановки после сброса. Когда таймер остановлен, TCRR замораживается. Таймер может быть перезапущен из замороженного значения, если TCRR не был перезагружен с новым значением.

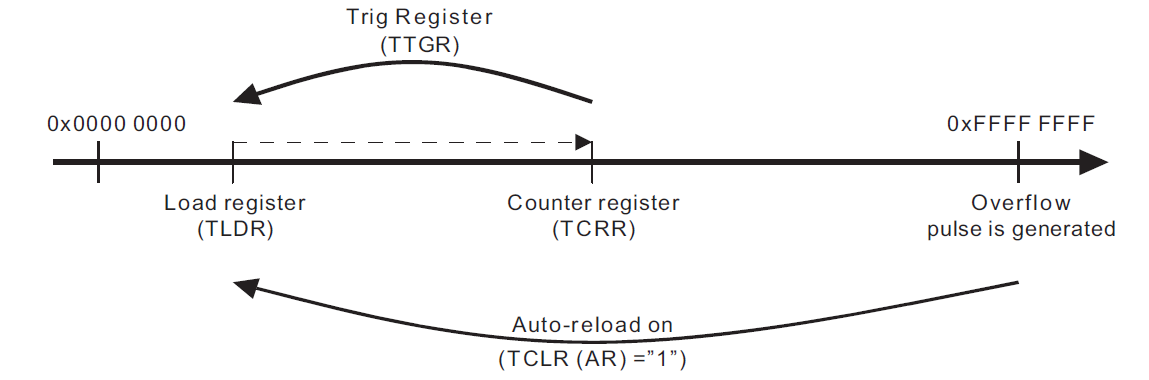
В режиме one shot (TCLR AR bit = «0») счетчик останавливается после подсчета переполнения (значение счетчика остается на нуле).

Если включен режим автоматической перезагрузки (бит TCLR AR = «1»), TCRR перезагружается со значением TLDR после переполнения счетчика.

Не рекомендуется помещать значение переполнения (0xFFFFFFFF) в TLDR, поскольку это может привести к нежелательному результату.

Прерывание может выдаваться при переполнении, если бит разрешения прерывания переполнения установлен в регистре таймера разрешения прерывания (TIER OVF\_IT\_ENA bit = «1»). Выход вывода (PORTIMERPWM) программируется через TCLR (биты TRG и PT) для генерации одного положительного импульса (длительность предварительного масштабирования) или для инвертирования текущего значения (режим переключения) при переполнении.

**Рис. 20-28. Значение синхронизации TCRR**



**20.2.3.1.1 Генерация тиков 1 мс**

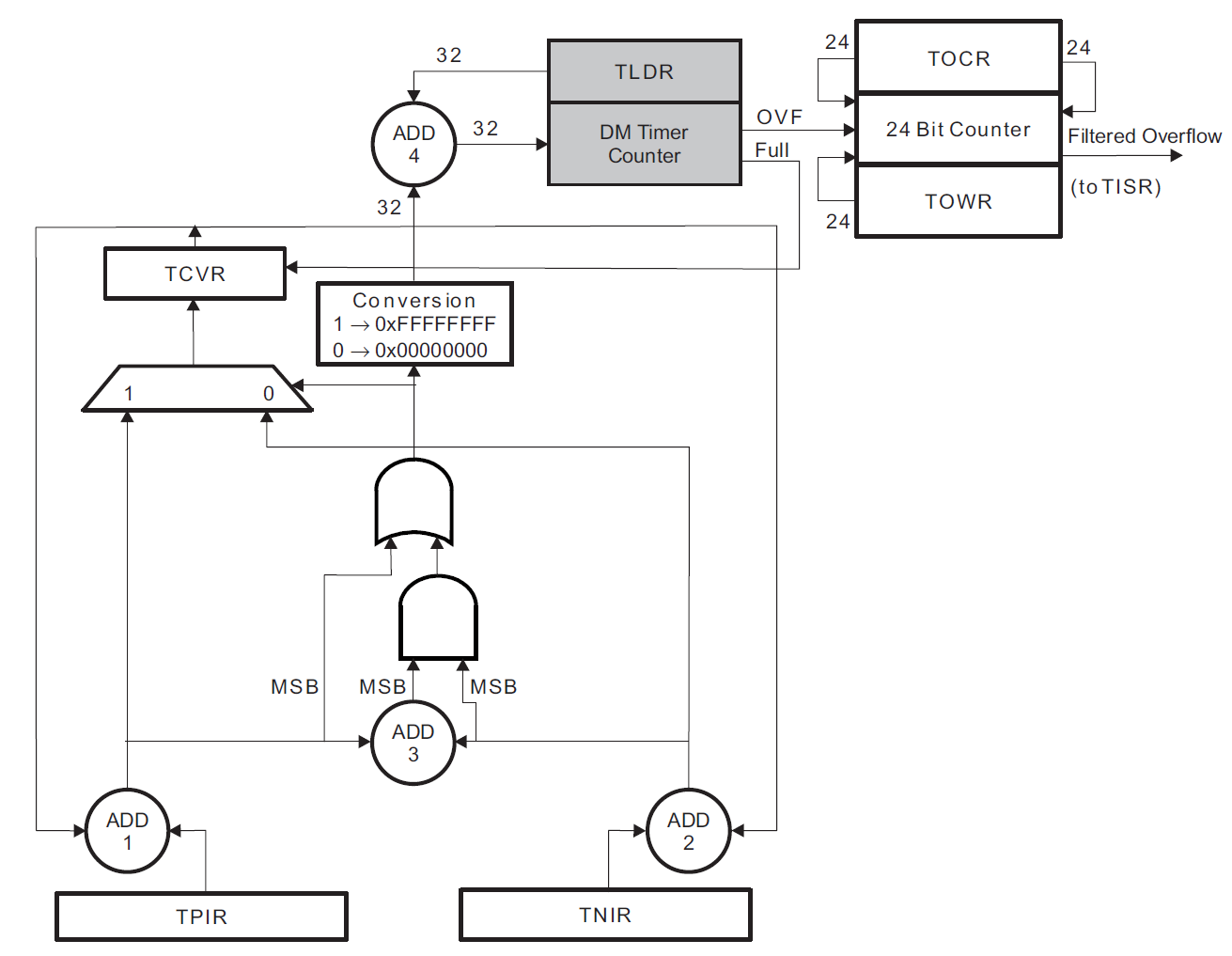
Чтобы минимизировать ошибку между истинным тиком 1 мс и тиком, сгенерированным таймером 32768 Гц, необходимо переключать последовательность периодов длительностью менее 1 мс и периодов длительностью свыше 1 мс.

Для исправления этой ошибки используется дополнительный блок (блок 1 мс).

В этой реализации инкрементное упорядочение автоматически управляется таймером для минимизации ошибкт. Значение регистра положительного приращения таймера (TPIR) и регистра отрицательного приращения таймера

(TNIR) должно быть определен только пользователем. Механизм автоматической адаптации используется для упрощения модели программирования.

**Рис. 20-29. Блок-схема модуля 1 мс**



TPIR, TNIR, TCVR и сумматоры Add1-3, используются для определения того, будет ли следующее значение загруженное в TCRR значением TLDR (значение подпериода) или значением TLDR - 1 (значение над периодом).

В следующей таблице приведено значение, загруженное в TCRR по знаку результата Add1, Add2 и Add3. MSB = «0» означает положительное значение, MSB = «1» означает отрицательное значение.

**Таблица 20-30. Значение, загруженное в TCRR для генерации отметки 1 мс**

|  |  |  |  |
| --- | --- | --- | --- |
| **Add1 MSB** | **Add2 MSB** | **Add3 MSB** | **TCRR** |
| 0 | 0 | 0 | TLDR |
| 0 | 0 | 1 | TLDR |
| 0 | 1 | 0 | TLDR |
|  | 1 | 1 | TLDR - 1 |
| 1 | 0 | 0 | N.A. |
| 1 | 0 | 1 | N.A. |
| 1 | 1 | 0 | TLDR - 1 |
| 1 | 1 | 1 | TLDR - 1 |

Значения регистров TPIR и TNIR вычисляются по формуле:

Положительное значение приращения = ((INTEGER [Fclk \* Ttick] + 1) \* 1e6) - (Fclk \* Ttick \* 1e6)

Отрицательное значение приращения = (INTEGER [Fclk \* Ttick] \* 1e6) - (Fclk \* Ttick \* 1e6)

где:

Fclk - тактовая частота (кГц)

Tтик - период тика (мс)

Регистр счетчика переполнения таймера (TOCR) и регистр обертывания переполнения таймера (TOWR) используется для фильтрации прерываний. Когда таймер переполняется, он увеличивает 24-битный регистр TOCR. Когда 24-битовое значение регистра TOCR совпадает со значением в 24-битовом регистре TOWR, и устанавливается переполнение таймера, TOCR сбрасывается и генерируется прерывание в TISR.

С блоком преобразования в состоянии сброса (регистр положительного приращения, регистр отрицательного приращения и Регистр значений счетчика обнулен), модель программирования и поведение DMtimer\_dmc1ms остаются неизменными.

Длч 1 мс с частотой тактирования 32768-Hz :

TPIR = 232000

TNIR = -768000

TLDR = 0xFFFFFFE0

*ПРИМЕЧАНИЕ: Любое значение периода тика может быть сгенерировано с соответствующими значениями регистровTPIR, TNIR и TLDR.*

*По умолчанию регистры TPIR, TNIR, TCVR, TOCR, TOWR и соответствующая логика находятся в режиме сброса (все 0s) и не имеют никаких действий на модели программирования DMtimer\_dmc1ms.*

**20.2.3.2 Функциональные возможности режима захвата**

Значение таймера в TCRR может быть зафиксировано и сохранено в TCAR1 или TCAR2 функции выбранного режима в TCLR через поле CAPT\_MODE при обнаружении перехода на входном контакте модуля (PIEVENTCAPT). Схема обнаружения краев контролирует переходы на входном контакте (PIEVENTCAPT).

Восходящий переход, нисходящий переход или оба могут быть выбраны в TCLR (бит TCM) для запуска захвата счетчика таймер. Модуль устанавливает TISR (TCAR\_IT\_FLAG бит) при обнаружении активного перехода и в то же время значение счетчика TCRR сохраняется в одном из регистров захвата таймера TCAR1 или TCAR2 следующим образом:

• Если CAPT\_MODE поле TCLR равно «0», то при первом включенном событии захвата значение счетчика в регистре TCAR1 сохраняется , и все последующие события игнорируются (нет обновления на TCAR1 и нет срабатывание прерывания) до тех пор, пока логика обнаружения не будет сброшена или регистр состояния прерывания не будет сброшен в TCAR записав в нее «1».

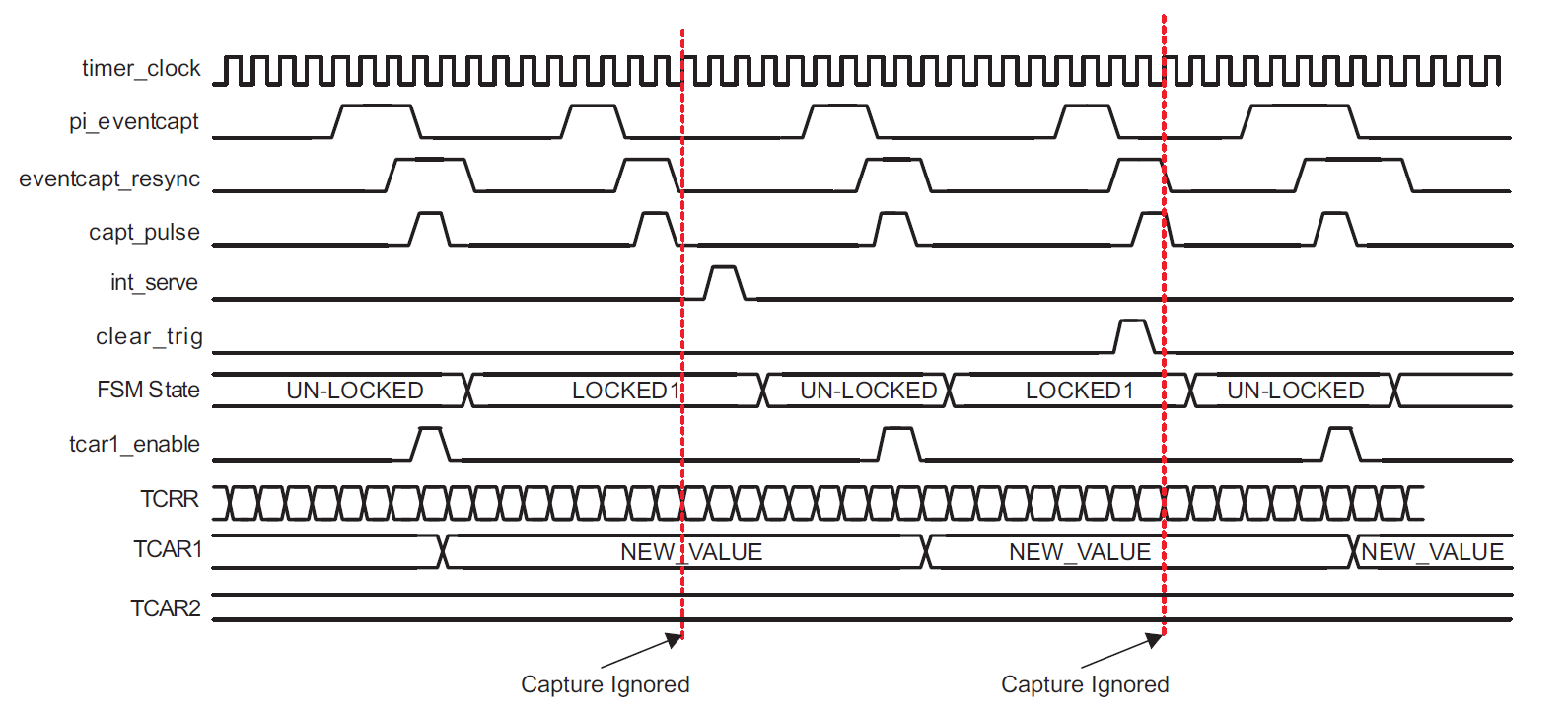
• Если CAPT\_MODE поле TCLR равно «1», то при первом включенном зафиксированном событии значение счетчика сохраняется в TCAR1 регистре, и при втором включенном событии захвата значение регистра счетчика сохраняются в TCAR2 регистре. Если прерывание захвата включено, прерывание будет установлено на втором захвату событий. Все остальные события игнорируются (без обновления при TCAR1/2 и без запуска прерывания) до тех пор, пока логика обнаружения сбрасывается или регистр состояния прерывания очищается в позиции TCAR, записывая «1» в него. Этот механизм полезен для расчета периода синхросигнала, если этот синхросигнал подключен к входному контакту PIEVENTCAPT.

Логика обнаружения границ сбрасывается (включается новый захват) при обслуживании активного прерывания захвата. TCAR\_IT\_FLAG бит TISR (ранее «1») очищается путем записи в него «1» или когда обнаружение края

биты режима TCLR (бит TCM), передаваемые из обнаружения режима отсутствия захвата в любые другие режимы. Функциональный тактовый синхросигнал таймера (вход в прескалер) используется для выборки входного контакта (PIEVENTCAPT). Отрицательные или положительные входные импульсы могут детектироваться, когда время импульса превышает функциональный тактовый период. Прерывание может выдаваться при обнаружении перехода, если бит разрешения прерывания захвата установлен в поле Timer Interrupt Enable регистра TIER (TCAR\_IT\_ENA бит).

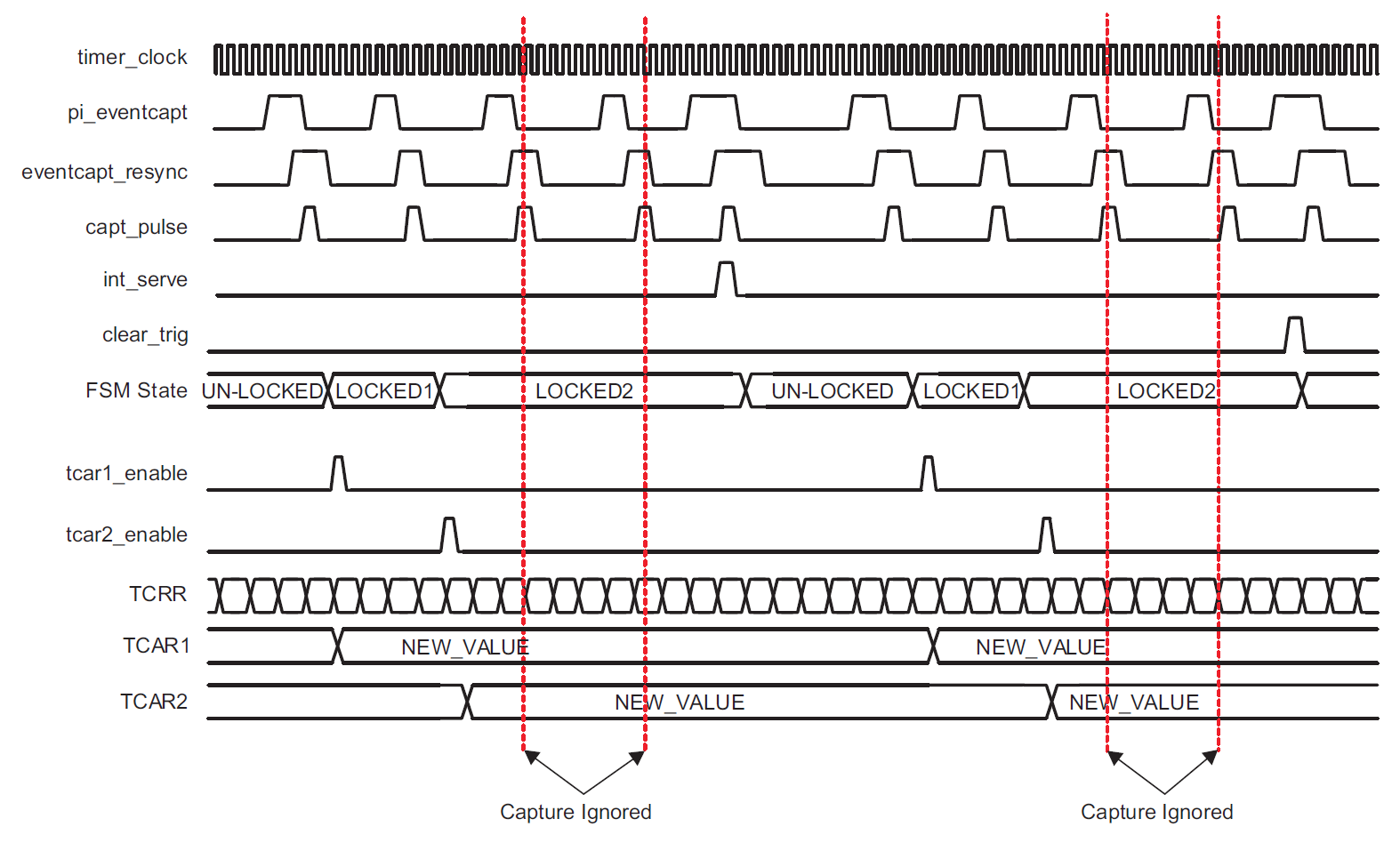
См. следующие примеры:

В следующей волне значение TCM равно «01», а CAPT\_MODE равно «0» - только передний фронт PIEVENTCAPT вызовет захват в TCAR, и только TCAR1 будет обновляться.

**Рис. 20-30. Пример волны захвата для CAPT\_MODE 0** 

В следующем примере значение TCM равно «01», а CAPT\_MODE - «1» - только передний фронт PIEVENTCAPT запустит захват в TCAR1 при первом включенном событии, и TCAR2 будет обновляться на второе включенное событие.

**Рис. 20-31. Пример волны захвата для CAPT\_MODE 1**



**20.2.3.3 Функциональные возможности режима сравнения**

Если для параметра Compare Enable TCLR (бит CE) установлено значение «1», значение таймера (TCRR) постоянно сравнивается с значением, хранящимся в регистре соответствия таймера (TMAR). Значение TMAR может быть загружено в любое время (отсчет таймера или стоп. Когда значения TCRR и TMAR совпадают, прерывание может быть выдано, если выставлен бит в TIER (MAT\_IT\_ENA бит). Правильная реализация заключается в записи значения сравнения в регистр TMAR перед установкой TCLR (Бит CE), чтобы избежать нежелательных прерываний из-за эффекта соответствия значений сброса.

Выделенный выходной контакт (PORTIMERPWM) может быть запрограммирован через TCLR (биты TRG и PT) на генерацию одного положительного импульса (длительность таймера) или инвертировать текущее значение (режим переключения), когда происходит переполнение или совпадение.

**20.2.3.4 Функциональность прескалера**

Для деления входной тактовой частоты счетчика таймера можно использовать счетчик предварительного масштабирования. Прескалер активизируется при установке бита 5 TCLR (PRE). Значение коэффициента деления 2n (PTV) может быть настроено в TCLR регистре.

Счетчик прескалера сбрасывается, когда счетчик таймера останавливается или перезагружается на лету.

Таблица 20-31. Сравнение значений предзагрузчика/таймера с контекстами

|  |  |  |
| --- | --- | --- |
| **Contexts** | **Prescaler Counter** | **Prescaler Counter** |
| Overflow (when Auto-reload on) | reset | TLDR |
| TCRR Write | reset | TCRR |
| TTGR Write | reset | TLDR |
| Stop | reset | Frozen |

20.2.3.5 Широтно-импульсная модуляция

Таймер может быть сконфигурирован для обеспечения программируемой широтно-импульсной модуляции (PORTIMERPWM) выход. Выходной контакт PORTIMERPWM может быть настроен на включение указанного события. TCLR (TRG бит) определяет, на каком значении регистра переключается вывод PORTIMERPWM. Переполнение или совпадение могут быть использованы для переключения вывода PORTIMERPWM при возникновении условия сравнения.

В случае переполнения и режима совпадения событие совпадения будет игнорироваться с момента настройки режима до возникновения первого события переполнения TCLR (бит SCPWM) может быть запрограммирован на установку или сброс выходного сигнала PORTIMERPWM, в то время как счетчик остановлен или триггер выключен. Это позволяет фиксировать детерминированное состояние выходного вывода когда модуляция остановлена. Модуляция синхронно прекращается, когда бит TRG очищается и происходит переполнение.

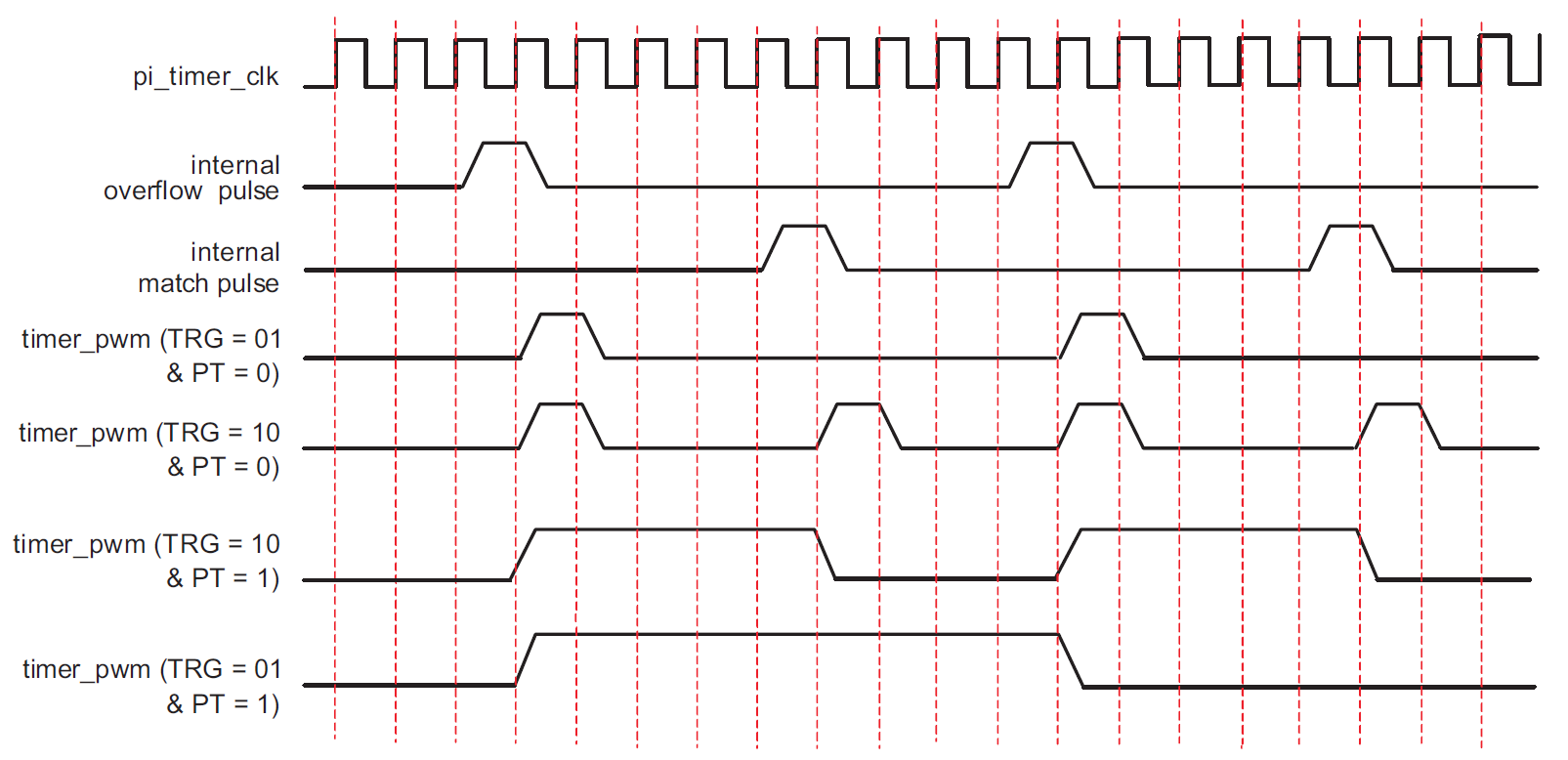
На следующей временной диаграмме каждый раз устанавливается внутренний импульс переполнения (0xFFFF FFFFF - TLDR + 1) достигается значение, и внутренний согласующий импульс устанавливается, когда счетчик достигает значения регистра TMAR.

В соответствии с программным значением TCLR (биты TRG и PT) таймер выдает импульс или PWM на выходе контакт (PORTIMERPWM).

Регистры TLDR и TMAR должны сохранять значения, меньшие, чем значение переполнения (0xFFFFFFFF) при не менее 2 единиц. В случае, если события триггера PWM одновременно переполнены и совпадают, разница между значениями, хранящиеся в регистре TMAR, и значение в TLDR должно быть не менее 2 единиц. При использовании события match должен быть установлен режим сравнения TCLR (CE).

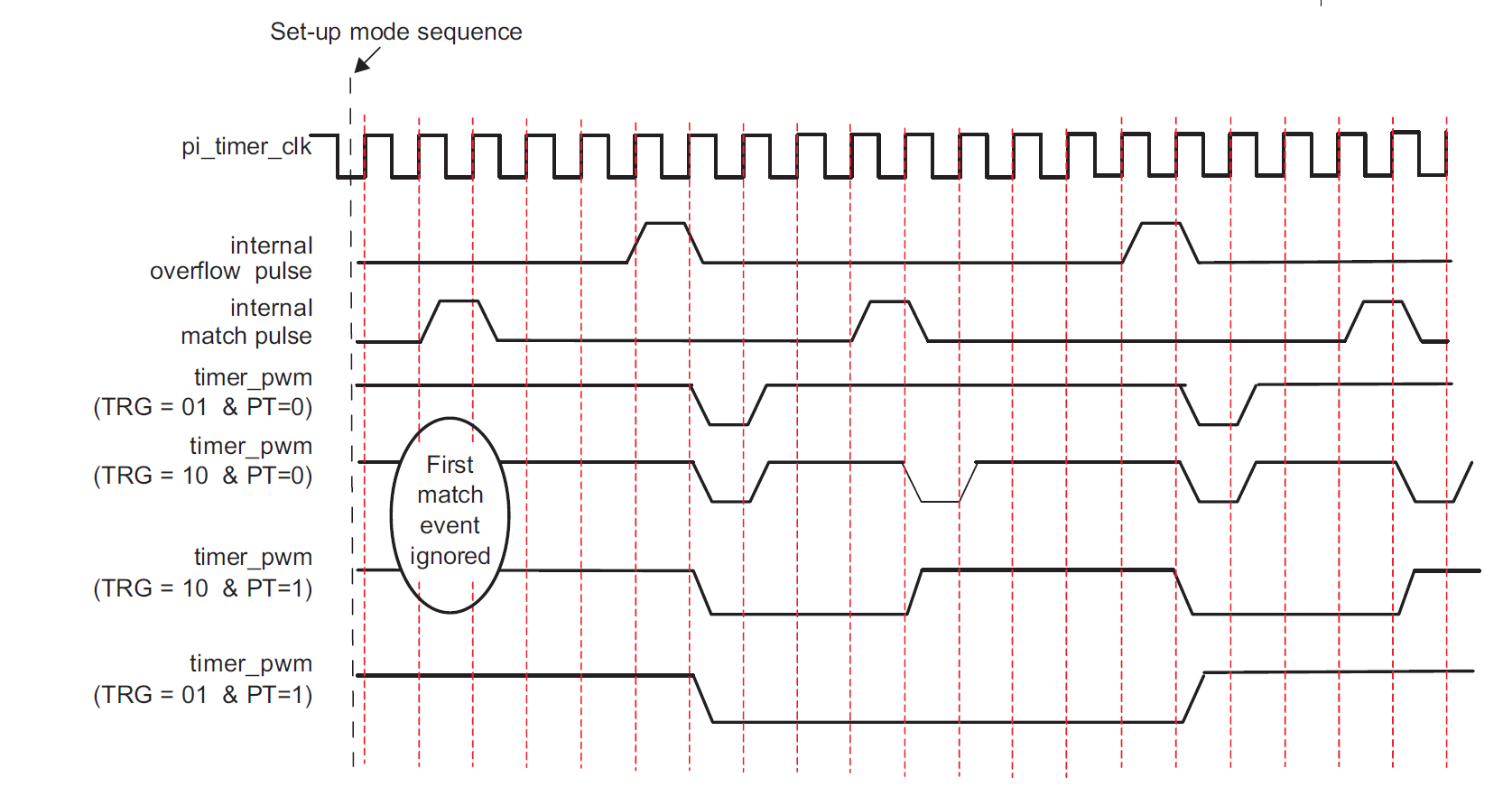
На следующей волне TCLR (бит SCPWM) устанавливается в «0».

**Рис. 20-32. Временная диаграмма широтно-импульсной модуляции, бит SCPWM = 0**



На следующей волне TCLR (бит SCPWM) устанавливается в «1».

**Рис. 20-33. Временная диаграмма широтно-импульсной модуляции, бит SCPWM = 1**

****

**20.2.3.6 Управление прерываниями таймера**

Таймер может выдать прерывание переполнения, прерывание совпадения таймера и прерывание захвата таймера. Каждый внутренний источник прерываний может быть независимо включен/отключен в регистре включения прерываний (TIER).

После выдачи события прерывания соответствующий бит состояния прерывания устанавливается в поле Timer Status.

Регистр (TISR). Событие отложенного прерывания сбрасывается, когда бит состояния установки перезаписывается значением «1».

Чтение регистра состояния прерывания и запись значения обратно обеспечивает быстрое подтверждение процесса прерывания.

**20.2.3.7 Запрос и подтверждение в спящем режиме**

По запросу спящего режима, выданному главным процессором (активный сигнал PIOCPMIDLEREQ запроса простоя), модуль таймера перейдет в спящий режим в соответствии с полем IdleMode системы регистра конфигурации (см. TIOCP\_CFG).

Если в поле IdleMode установлен режим No-Idle, таймер не переходит в режим Sleep и подтверждения сигнала Idle (POROCPSIDLEACK) никогда не устанавливается.

Если в поле IdleMode установлен режим Force-Idle, таймер переходит в спящий режим независимо от внутреннего

состояние модуля и безоговорочно устанавливается сигнал подтверждения простоя (POROCPSIDLEACK).

Если в поле IdleMode установлен режим Smart-Idle, модуль таймера оценивает свои внутренние возможности для выключения интерфейсного/функционального тактирования. В зависимости от поля ClockActivity, установка модуля таймера оценивает внутреннюю активность и устанавливает сигнал подтверждения простоя (POROCPSIDLEACK), вводя в спящий режим, готовый выдать запрос на пробуждение.

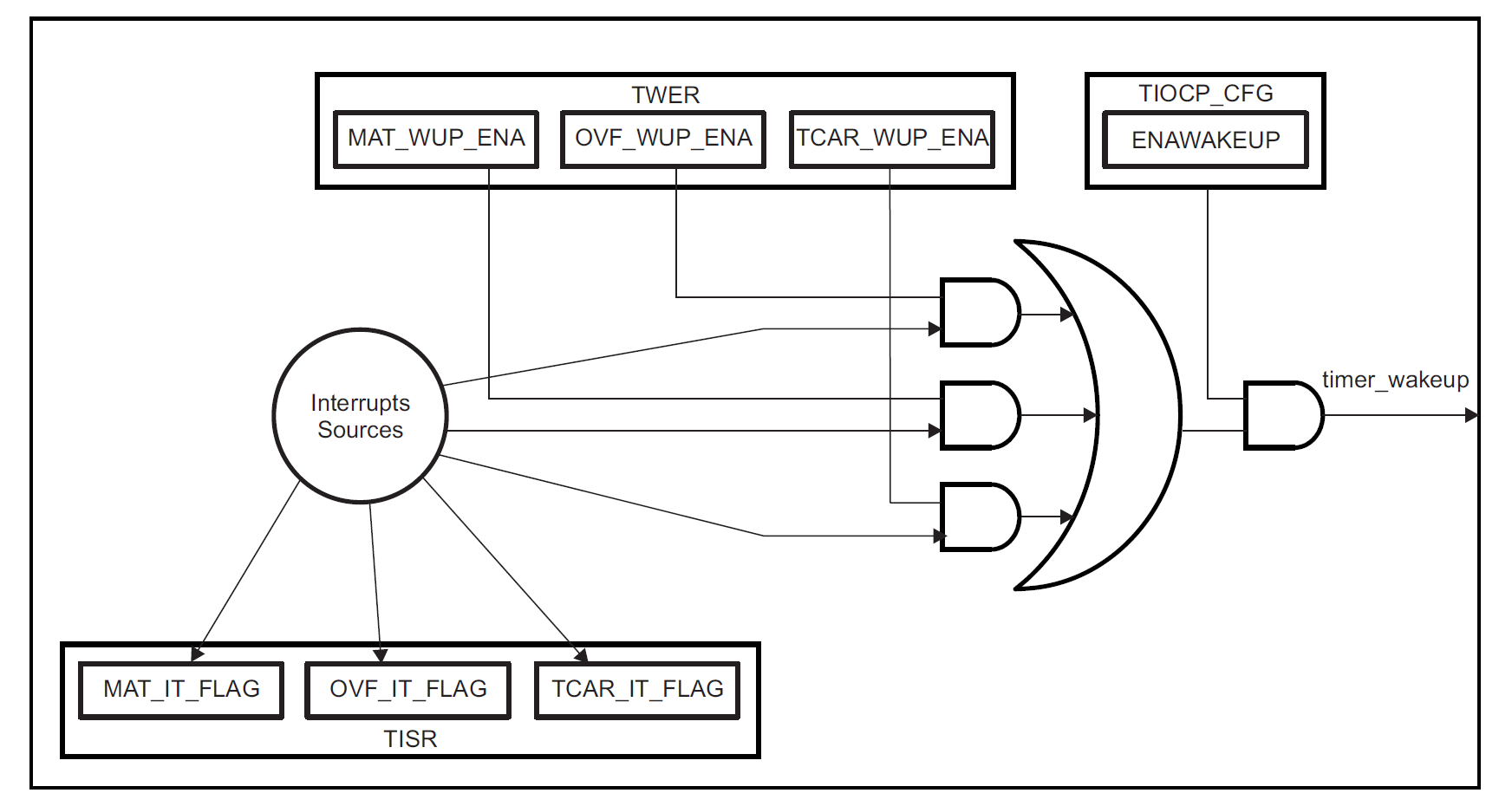
В следующей таблице описано поведение Smart Idle в соответствии с настройкой активности тактов:

**Таблица 20-32. SmartIdle - Настройка поля активности тактирования**

|  |  |  |  |
| --- | --- | --- | --- |
| **Clock Activity** | **Functional Clock** | **OCP Clock** | **Module Behavior** |
| 11 | ON | ON | Сигнал подтверждения холостого хода  утверждается, когда нет  ожидающих действии в домене OCP тактирования, без  оценка ожидающих  действий в домене функционального тактированя. (Модуль войдет в спящем режиме и, если событие отложенного прерывания завершено во время режима холостого хода  сигнал пробуждения будет  утверждается). |
| 10 | ON | OFF |
| 01 | OFF | ON | Сигнал подтверждения холостого хода  утверждается, когда нет  ожидающего действия в доменах функционального и OCP-синхросигнала (Улучшенная задержка в утверждение подтверждения Idle).  Возможность пробуждения  модуля отключена. |
| 00 | OFF | OFF |

Этот запрос на пробуждение эффективно отправляется только в том случае, если поле ENAWAKEUP TIOCP\_CFG включает возможность пробуждения таймера. При пробуждении системы сигнал Idle Request становится неактивным, и сигнал запроса на пробуждение также отменяется.

**Рис. 20-34. Создание запроса пробуждения**



**20.2.3.7.1 Освобождение линии пробуждения**

Когда хост-процессор получает запрос пробуждения, выданный периферийным устройством таймера, тактовый сигнал интерфейса повторно активизируется: хост-процессор деактивизирует PIOCPMIDLEREQ, таймер деактивизирует сигнал POROCPSIDLEACK, а затем хост может прочитать соответствующий бит в TISR, чтобы выяснить, какой источник прерывания инициировал запрос пробуждения. После подтверждения запроса пробуждения процессор сбрасывает бит состояния и освобождает линию прерывания, записывая «1» в соответствующий бит регистра TISR

**20.2.3.8 Частота подсчета таймера**

Счетчик dmtimer состоит из ступени прескалера и счетчика таймера.

Тактовым коэффициентом прескалера можно управлять, обратившись к полю определения коэффициента управляющего регистра

(PTV и PRE TCLR).

Скорость таймера определяется:

• Значением полей прескалера (PRE и PTV регистра TCLR)

• Значением, загруженном в регистр загрузки таймера (TLDR).

**Таблица 20-33. Значение тактовых коэффициентов прескалера**

|  |  |  |
| --- | --- | --- |
| **PRE** | **PTV** | **Divisor (PS)** |
| 0 | X | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 4 |
| 1 | 2 | 8 |
| 1 | 3 | 16 |
| 1 | 4 | 32 |
| 1 | 5 | 64 |
| 1 | 6 | 128 |
| 1 | 7 | 256 |

Уравнение скорости таймера выглядит следующим образом:

(0xFFFF FFFF - TLDR + 1) x период таймера x делитель тактового сигнала (PS)

С таймером Clock period = 1/timer Clock frequency и PS = 2 (PTV + 1).

В качестве примера, если мы рассмотрим вход таймера 32 кГц, с полем PRE равным «0», выход таймера период:

**Таблица 20-34. Значение и соответствующий период прерывания**

|  |  |
| --- | --- |
| **TLDR** | **Interrupt Period** |
| 0x0000 0000 | 37 h |
| 0xFFFF 0000 | 2 s |
| 0xFFFF FFF0 | 500 us |
| 0xFFFF FFFE | 62.5 us |

**20.2.3.9 Поведение таймера во время эмуляции**

Чтобы настроить таймер на остановку во время событий приостановки эмуляции (например, точек останова отладчика), настройте таймер в подсистеме отладки:

1. Установить TIOCP\_CFG.EMUFREE=0. Это позволит Suspend\_Control сигнал от подсистемы отладки (Глава 27), чтобы остановить и запустить таймер. Обратите внимание, что если EMUFREE = 1, Suspend\_Control сигнал

игнорируется, и таймер работает независимо от любого события debug suspend. Бит EMUFREE дает локальное управление с точки зрения модуля для получения сигнала приостановки, поступающего из подсистемы отладки.

2. Установите соответствующий регистр xxx\_Suspend\_Control = 0x9, как описано в разделе 27.1.1.1, *Поддержка приостановки отладки для периферийных устройств.* Выберите регистр, соответствующий периферийному устройству, которое вы хотите приостановить во время события приостановки.

Мо