

## 12 Таймеры улучшенного управления(TIM1)

Этот раздел касается всего семейства STM32F100xx если не указано иное.

### 12.1 Введение в TIM1

Таймер улучшенного управления(TIM1) состоит из 16-битного автоперезагружаемого счетчика, управляемого программируемым предделителем.

Он может использоваться для различных целей, включая измерение ширины импульсов входных сигналов(захват по входу) или генерацию сигналов(выход по сравнению, ШИМ, комплементарный ШИМ с вставкой мертвого времени).

Длина импульсов и период сигналов может модулироваться от нескольких микросекунд до миллисекунд используя предделитель таймера и предделители контроллера RCC.

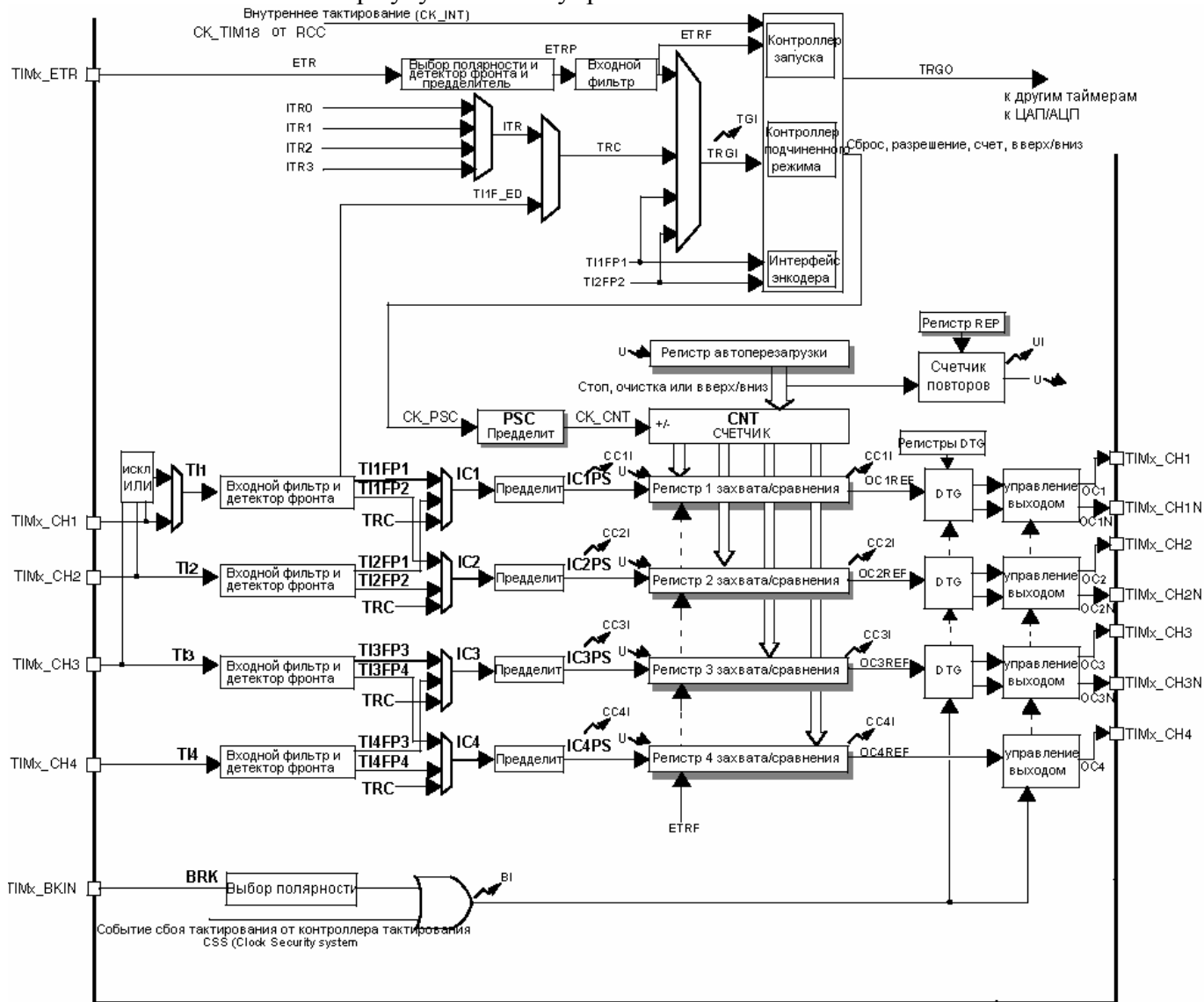
Таймеры улучшенного управления(TIM1) и общего назначения(TIMx) полностью независимы и не используют какие-либо ресурсы. Они способны синхронизироваться вместе как это описано в разделе 12.3.20.

### 12.2 Основные возможности TIM1

Возможности включают:

- 16-битный автоперезагружаемый счет вверх, вниз, вверх/вниз;
- 16-битный программируемый предделитель, позволяющий делить(также «на лету») тактовую частоту на любой коэффициент между 1 и 65535.
- До 4-х независимых каналов для:
  - Захвата входа
  - Выхода сравнения
  - Генерации ШИМ(режим выравнивания по центру и по фронту)
  - Режим вывода одиночного импульса.
- Комплементарные выходы с программируемым мертвым временем
- Схема синхронизации для управления таймером внешними сигналами и соединения нескольких таймеров вместе
- Счетчик повторений для обновления регистров таймера только после данного числа циклов счетчика
- Вход останова для перевода сигналов таймера в состояние сброса или известное состояние
- Генерация прерывания/DMA по следующим событиям:
  - Обновление: переполнение счета, инициализация счетчика(программно или внутр/внешн запуск)
  - Событие запуска(начало счета, остановка, инициализация или счет по внутр/внешн запуску)
  - Выход сравнения
  - Вход останова
- Поддерживает инкрементальный(квадратурный) энкодер и датчик Холла для целей позиционирования
- Входы запуска внешнего тактирования или управления цикл за циклом

Рис.40. Блок-схема таймера улучшенного управления



Примечание:

Reg Предзаруженные регистры передаются в активные рег-ры по событию U в соответствии с битом упр-я событие

↘ прерывание и выход DMA

## 12.3 Функциональное описание TIM1

### 12.3.1 Блок временной базы

Главный блок таймера – 16-битный счетчик с регистром автозагрузки. Счетчик может считать вверх, вниз и вверх-вниз. Тактовая частота счетчика может делиться предделителем.

Счетчик, регистр автозагрузки и регистр предделителя могут быть прочитаны и записаны программой. Это верно даже во время работы счетчика.

Регистр автоперезагрузки предварительно загружается. Запись или чтение в регистр автоперезагрузки осуществляет доступ в предварительно загруженный регистр. Содержимое предзагруженного регистра передается в теневой регистр постоянно или по каждому событию обновления(UEV), в зависимости от бита(ARPE) разрешения предварительной автоперезагрузки регистра TIMx\_CR1. Событие обновления посылается когда счетчик достигает переполнения(счет вверх или вниз) и если бит UDIS равен 0. Оно также может генерироваться программно.

Счетчик тактируется выходом предделителя CK\_INT, который разрешен только когда установлен бит разрешения счетчика(CEN) регистра TIMx\_CR1.

Прим: счетчик начинает счет через 1 тактовый цикл после установки бита CEN.

### Описание предделителя

Предделитель может делить тактовую частоту счетчика на любой коэффициент между 1 и 65536. Он базируется на 16-битном счетчике, управляемым через 16-битный регистр TIMx\_PSC. Он может изменяться налету т.к. он буферизован. Новое значение предделителя принимается по следующему событию обновления.

Рис. 42 и рис.43 показывают примеры поведения счетчика когда предделитель меняется налету.

Рис.41. Временная диаграмма счетчика когда предделитель меняется с 1 на 2

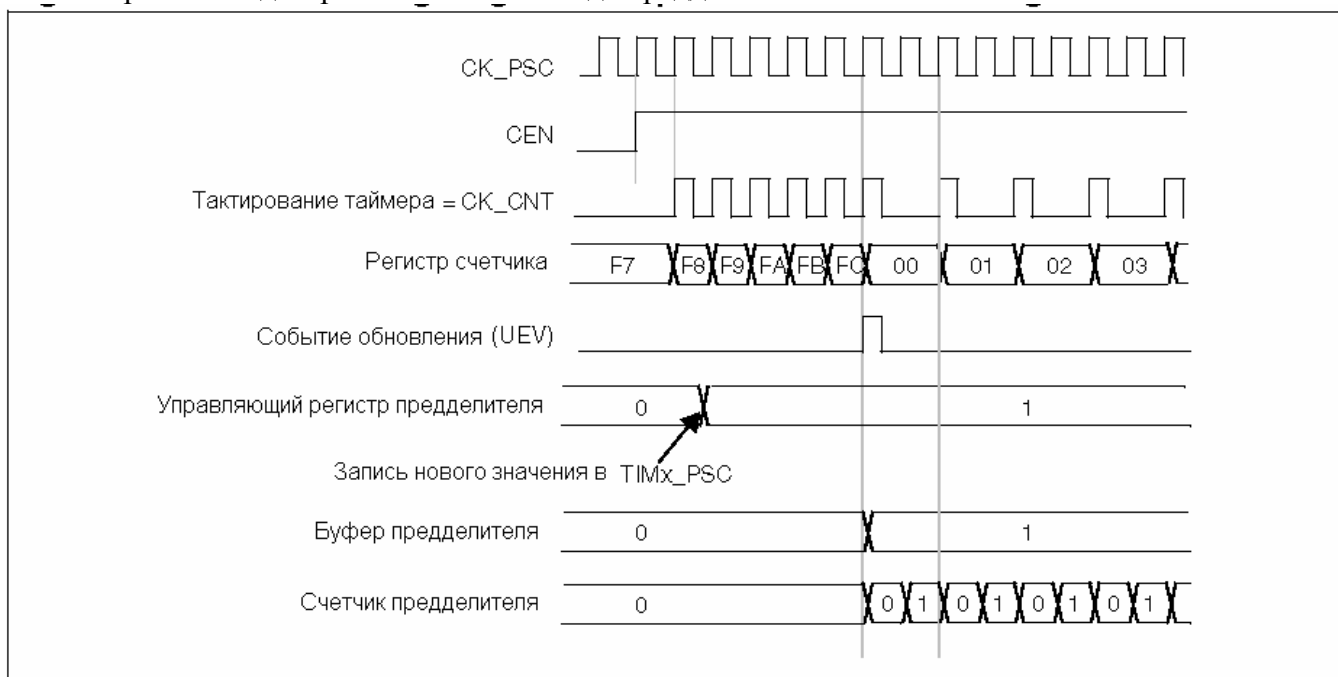
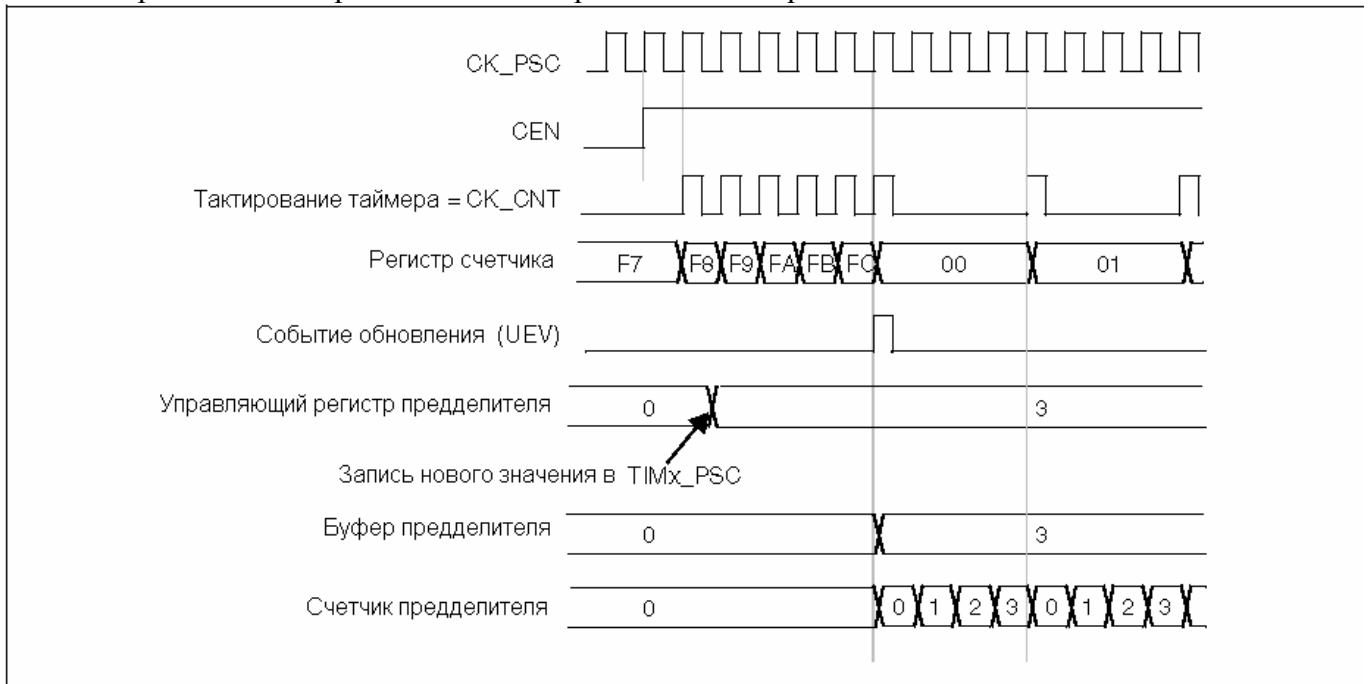


Рис. 42. Временная диаграмма счетчика при изменении предделителя с 1 на 4



### 12.3.2 Режимы счетчика

#### Режим счета вверх

Счетчик считает с 0 до значения автоперезагрузки (содержимое регистра TIMx\_ARR), затем перезапускается с 0 и генерирует событие переполнения счетчика. Если используется счетчик повторов, то событие обновления (UEV) генерируется после повторения количества счетов вверх, которое запрограммировано в регистре TIMx\_RCR. Иначе событие обновления генерируется по каждому переполнению счетчика.

Установка бита UG в регистре TIMx\_EGR (программно или используя контроллер подчиненного режима) также генерирует событие обновления.

Событие UEV может быть запрещено программно установкой бита UDIS в регистре TIMx\_CR1. Это позволяет избежать обновления теневого регистра при записи новых значений в предзагружаемые регистры. Т.о. события обновления не происходят до записи бита UDIS в 0. Однако, счетчик перезапускается с нуля, также как и счетчик предделителя (но значения предделителя не меняется). В дополнение, если установлен бит URS (выбор запроса обновления) регистра TIMx\_CR1, установка бита UG генерирует событие обновления UEV, но без установки флага UIF (прерывание или запрос DMA не посылаются). Это для избежания генерации обновления и захвата прерываний при очистке счетчика по событию захвата.

Когда происходит событие обновления, то все регистры обновляются и устанавливается флаг обновления (бит UIF регистра TIMx\_SR, в зависимости от бита URS):

- Счетчик повторений перезагружается содержимым регистра TIMx\_RCR
- Автоперезагружаемый теневой регистр обновляется предзагруженным значением (TIMx\_ARR)
- Буфер предделителя перезагружается предзагруженным значением (TIMx\_PSC)

Следующие рисунки показывают некоторые примеры поведения счетчика при различных тактовых частотах когда TIMx\_ARR=0x36

Рис. 43. Внутреннее тактирование делится на 1

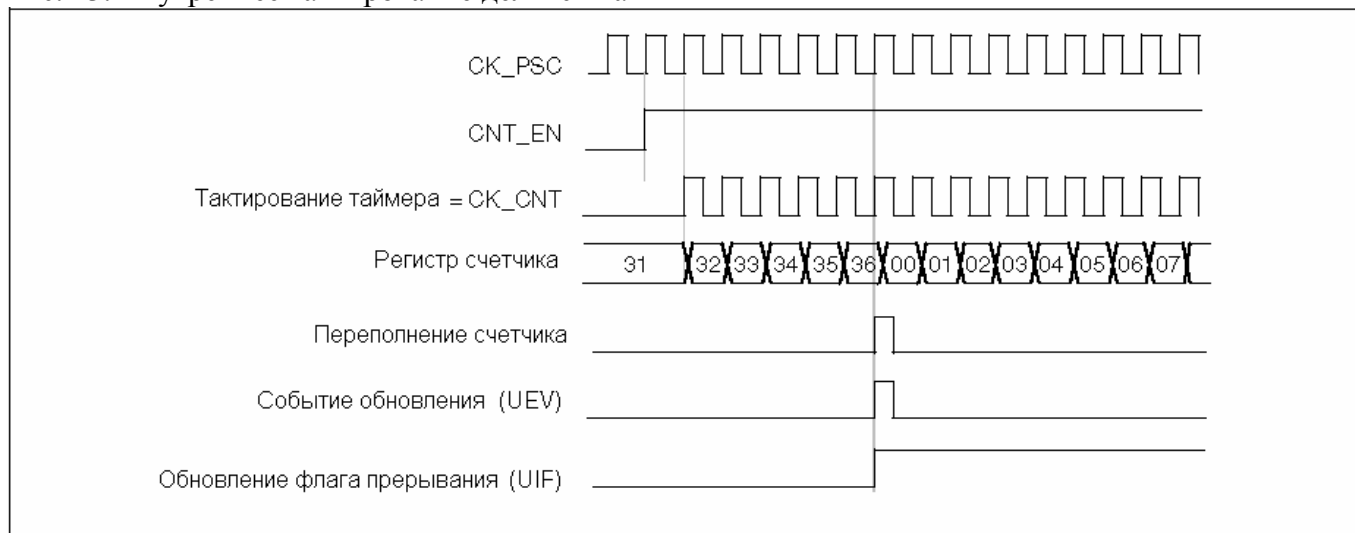


Рис.44. Внутреннее тактирование делится на 2

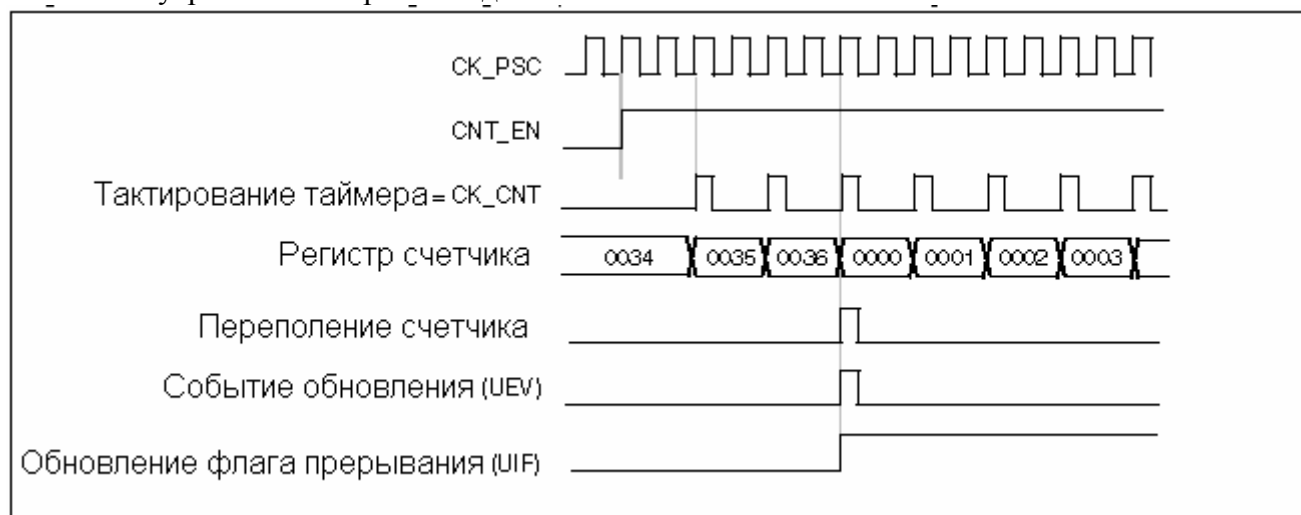


Рис.45. Внутреннее тактирование делится на 4

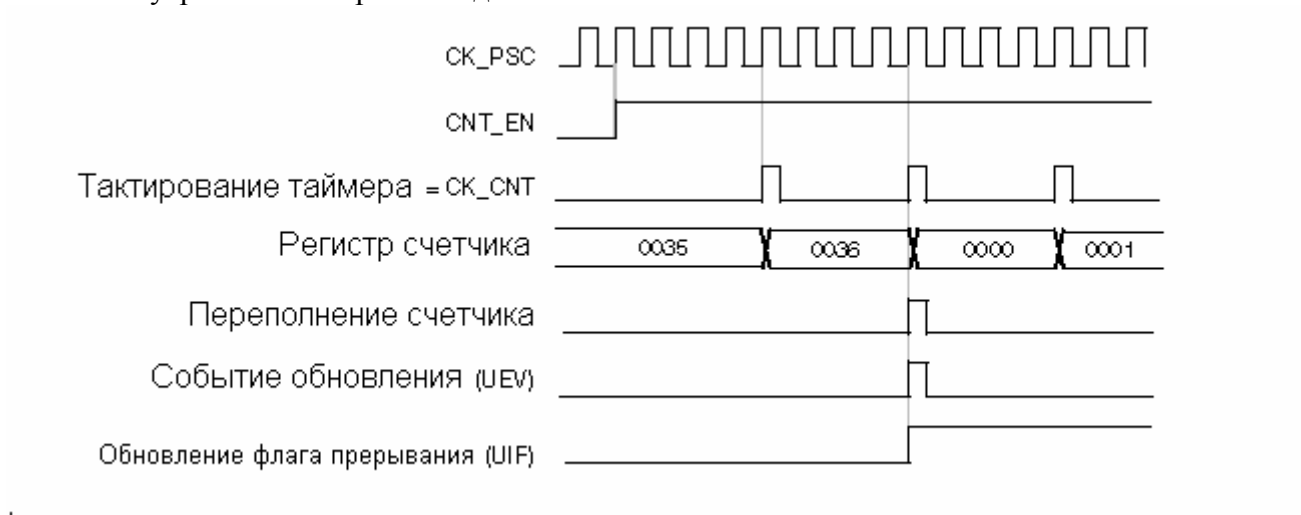


Рис. 46. Внутреннее тактирование делится на N

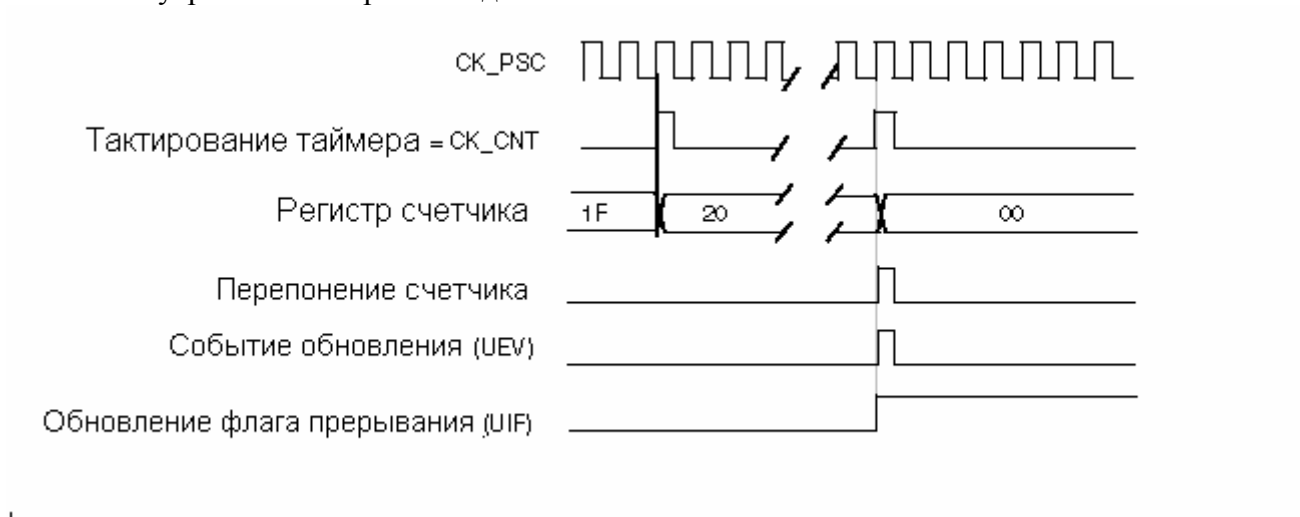


Рис. 47. Событие обновления когда ARPE=0(TIMx\_ARR не предзагружен)

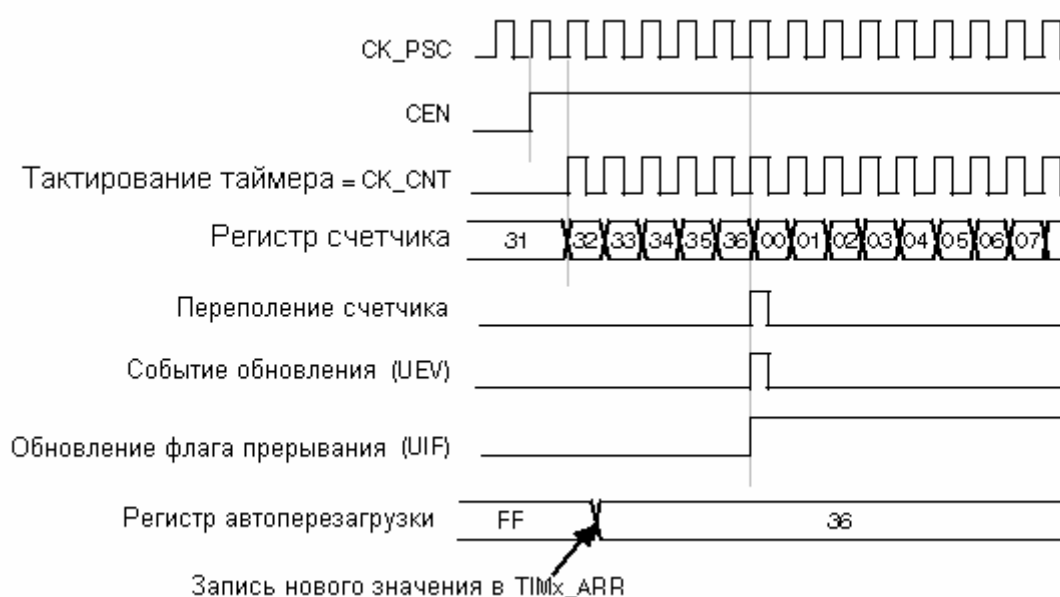
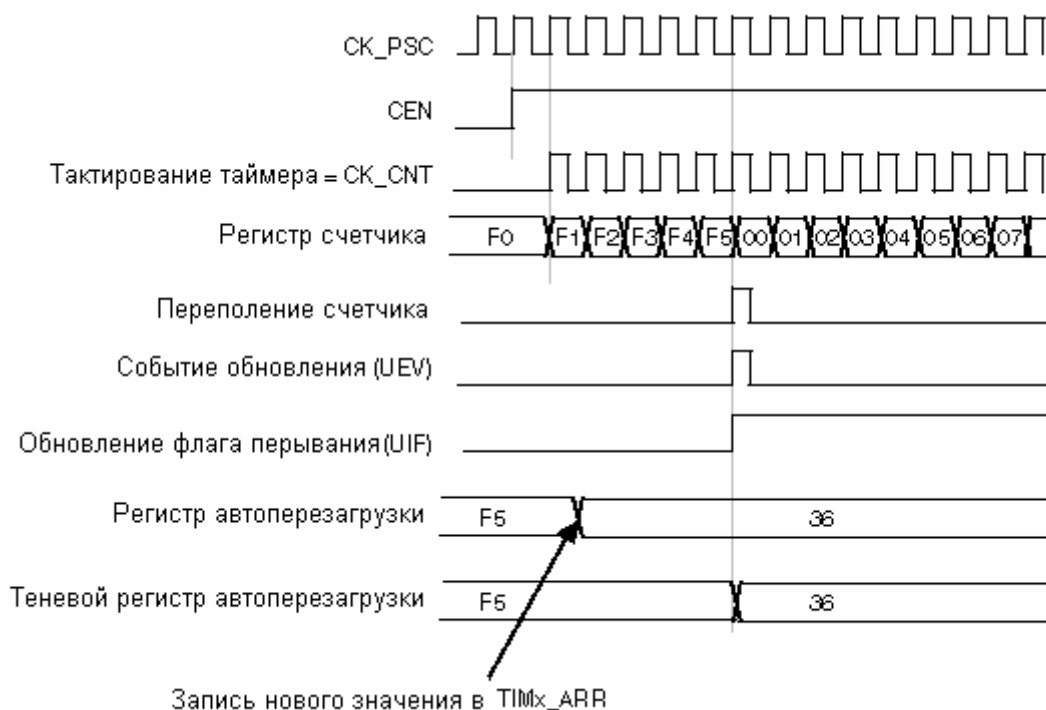


Рис. 48. Событие обновления когда ARPE=1(TIMx\_ARR предзагружен)



### Режим счета вниз

В режиме счета вниз счетчик считает со значения автозагрузки(содержимое регистра TIMx\_ARR) вниз до 0, затем перезапускается со значения автоперезагрузки и генерирует событие переполнения счетчика.

Если используется счетчик повторов, то событие обновления(UEV) генерируется после повторения определенного числа переполнений, запрограммированных в регистр счетчика повторений(TIMx\_RCR). Иначе событие обновления генерируется по каждому переполнению.

Установка бита UG в регистре TIMx\_EGR(программно или используя контроллер подчиненного режима) также генерирует событие обновления.

Событие обновления UEV может быть запрещено программой через установку бита UDIS в регистре TIMx\_CR1. Это позволяет избежать обновления теневых регистров при записи новых значений в регистры предварительной загрузки. События обновления не происходят пока бит UDIS не будет записан в 0. Однако счетчик перезапускается со значения автоперезагрузки несмотря на то, что счетчик предделителя перезапускается с 0(но коэффициент предделителя не меняется).

В дополнение, если установлен бит URS(выбор запроса обновления) регистра TIMx\_CR1, то установка бита UG генерирует событие обновления без установки флага UIF(поэтому прерывания и запросы DMA не посылаются). Это для избежания генерации обновления и захвата прерываний при очистке счетчика по событию захвата.

Когда происходит событие обновления, все регистры обновляются и устанавливается(зависит от бита URS) флаг обновления(UIF) :

- Счетчик повторов перезагружается содержимым регистра TIMx\_RCR
- Буфер предделителя перезагружается предварительно загруженным значением (содержимое TIMx\_PSC)
- Активный регистр автоперезагрузки обновляется предзагруженным значением(содержимое TIMx\_ARR). Прим: автоперезагрузка обновляется перед перезагрузкой счетчика, поэтому следующий период – ожидаемый.

Следующие рисунки показывают примеры поведения счетчика на различных частотах когда TIMx\_ARR=0x36.

Рис. 49. Внутреннее тактирование делится на 1

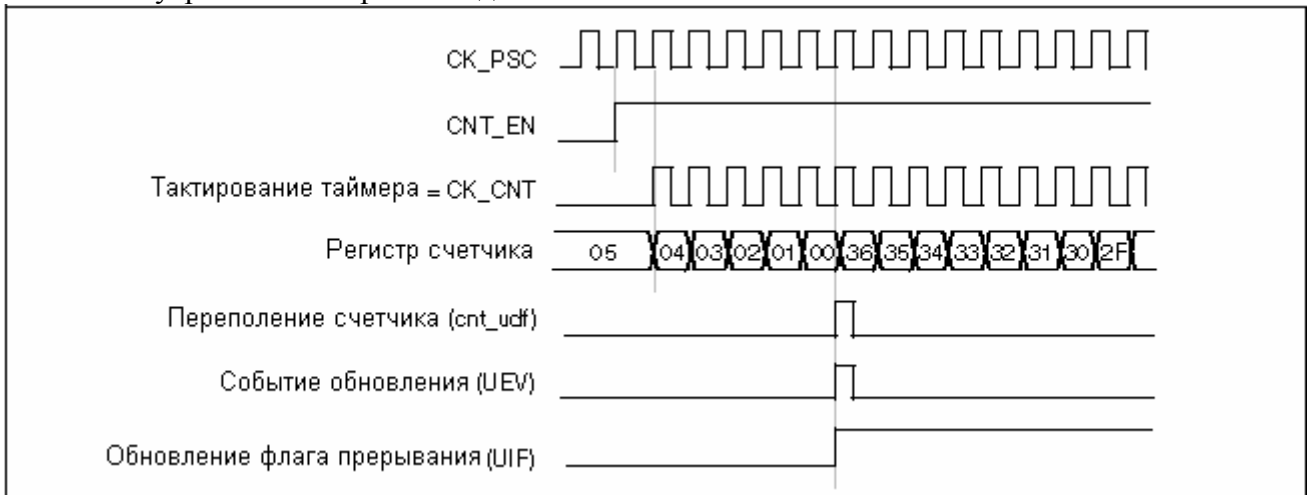


Рис.50. Деление на 2

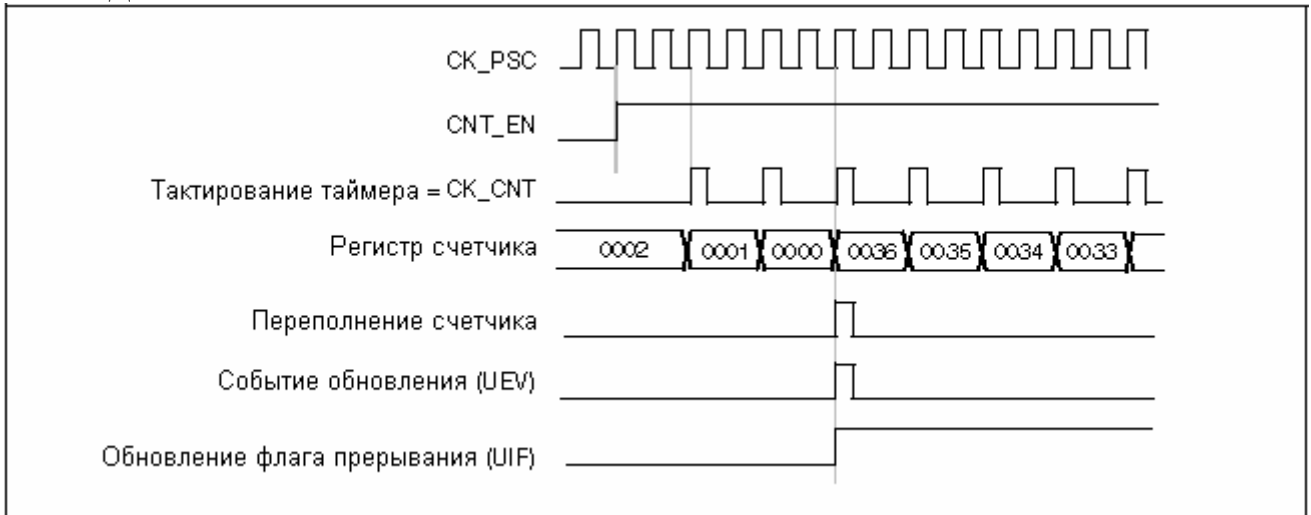


Рис.51. Деление на 4

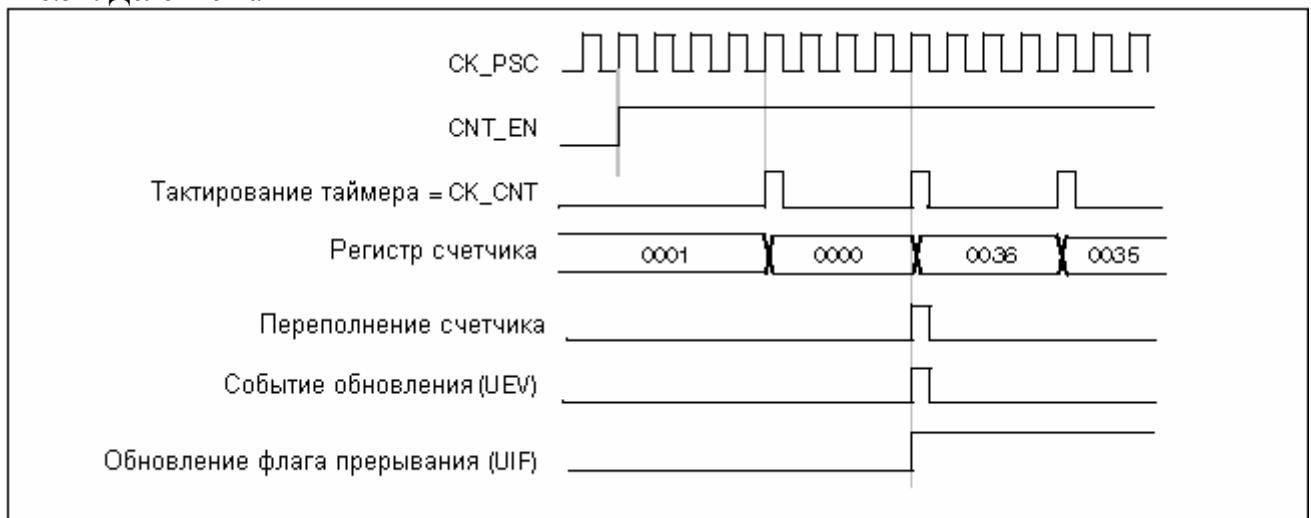


Рис. 52. Деление на N

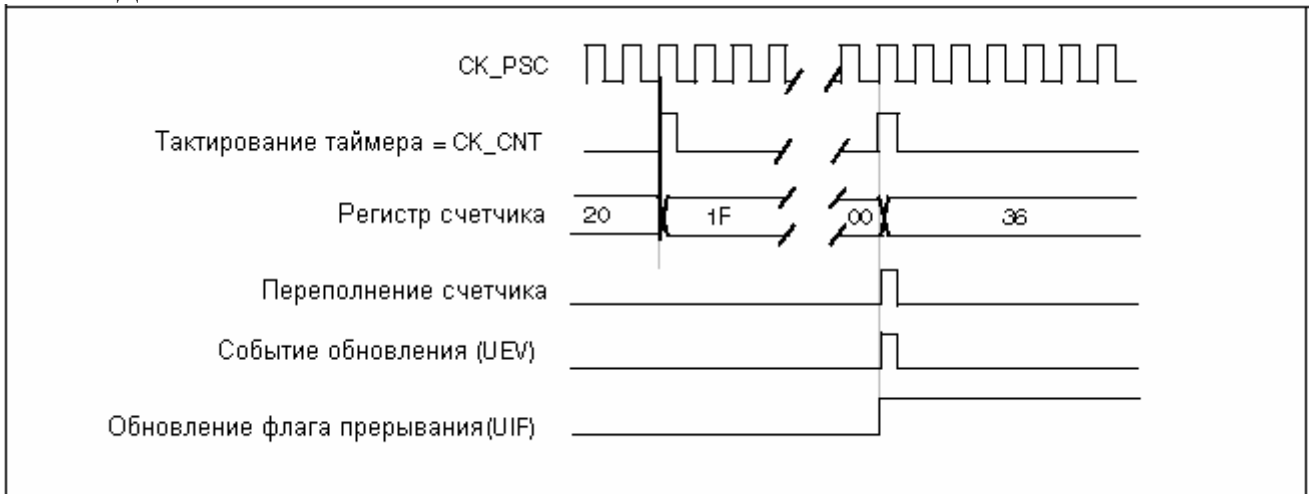
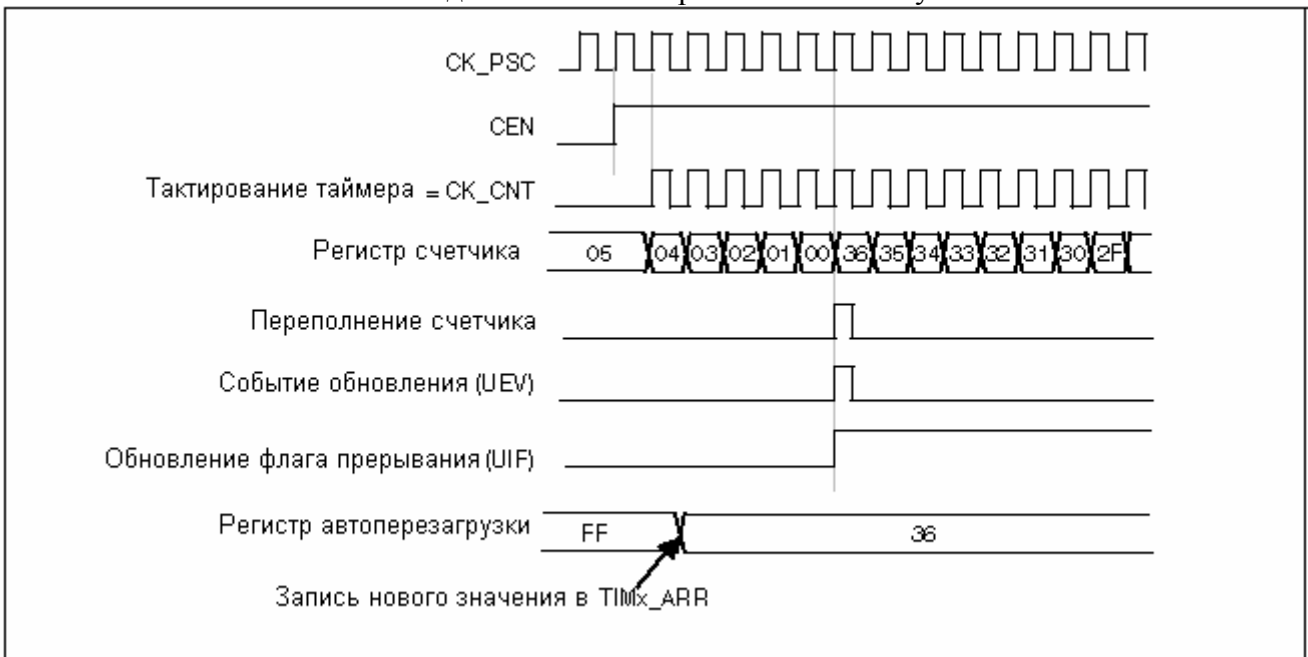




Рис.53. Событие обновления когда счетчик повторений не используется



### Режим выравнивания по центру(счет вверх-вниз)

Счетчик считает с 0 до значения автоперезагрузки (содержимое TIMx\_ARR)-1, генерирует событие переполнения, затем считает со значения автоперезагрузки до 1 и генерирует событие переполнения. Затем начинает счет с 0.

Режим выравнивания по центру активен, когда биты CMS в регистре TIMx\_CR1 не равны '00'. Флаг прерывания сравнения выхода каналов, сконфигурированных на выход, устанавливается когда: счетчик считает вниз(режим 1 выравнивания по центру, CMS='01'), счетчик считает вверх(режим 2, CMS='10'), счетчик считает вверх и вниз(режим 3, CMS='11').

В этом режиме бит направления DIR регистра TIMx\_CR1 не может быть записан. Он обновляется аппаратно и предоставляет текущее направление счетчика.

Событие обновления может генерироваться по каждому переполнению счетчика(вверх или вниз), или установкой бита UG(программно или контроллером подчиненного режима). В этом случае счетчик перезапускает счет с 0, также как и счетчик предделителя.

Событие обновления UEV может быть запрещено программно установкой бита UDIS. Это для избежания обновления теневых регистров при записи новых значений в регистры предзагрузки. Событие обновления не происходит, пока UDIS не будет записан в 0. Однако счетчик продолжает счет вверх-вниз базирясь по текущему значению автоперезагрузки.

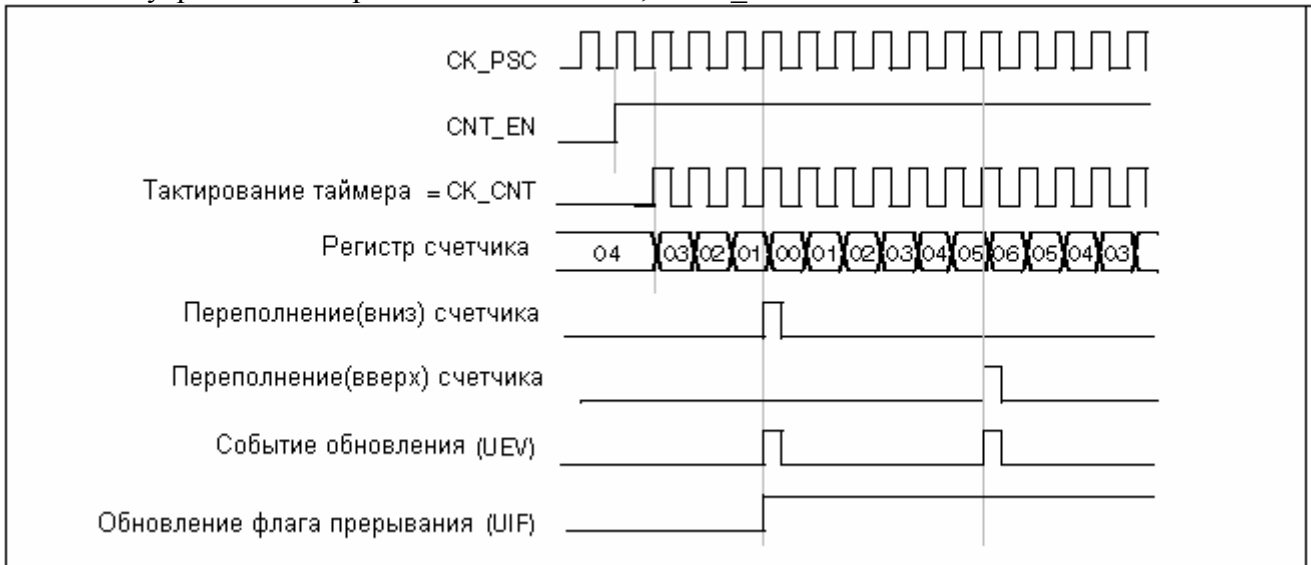
В дополнение, если установлен бит URS(выбор запроса обновления), то установка бита UG генерирует событие обновления UEV без установки флага UIF(Поэтому прерывание и запросы DMA не посылаются). Это для избежания генерации обновления и захвата прерываний при очистке счетчика по событию захвата.

Когда происходит событие обновления, то все регистры обновляются и устанавливается(в зависимости от бита URS) флаг обновления(UIF):

- Счетчик повторений перезагружается содержимым регистра TIMx\_RCR
- Буфер предделителя перезагружается значением предзагрузки(содержимое TIMx\_PSC)
- Активный регистр автоперезагрузки обновляется предзагруженным значением(содержимое TIMx\_ARR). Прим: если источником обновления является переполнение счетчика, то автоперезагрузка обновляется перед перезагрузкой счетчика. Поэтому следующий период является ожидаемым(счетчик загружается новым значением).

Следующие рисунки показывают поведение счетчика на различных тактовых частотах.

Рис.54. Внутреннее тактирование делится на 1, TIMx\_ARR=0x6.



1. Здесь используется режим 1 выравнивания по центру.

Рис. 55. Деление на 2

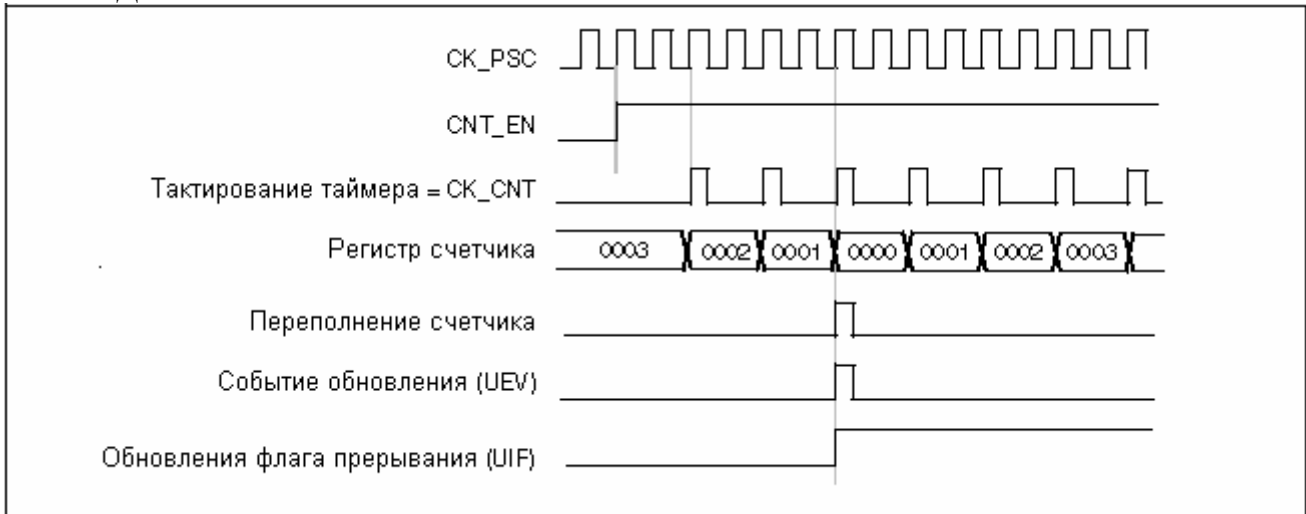
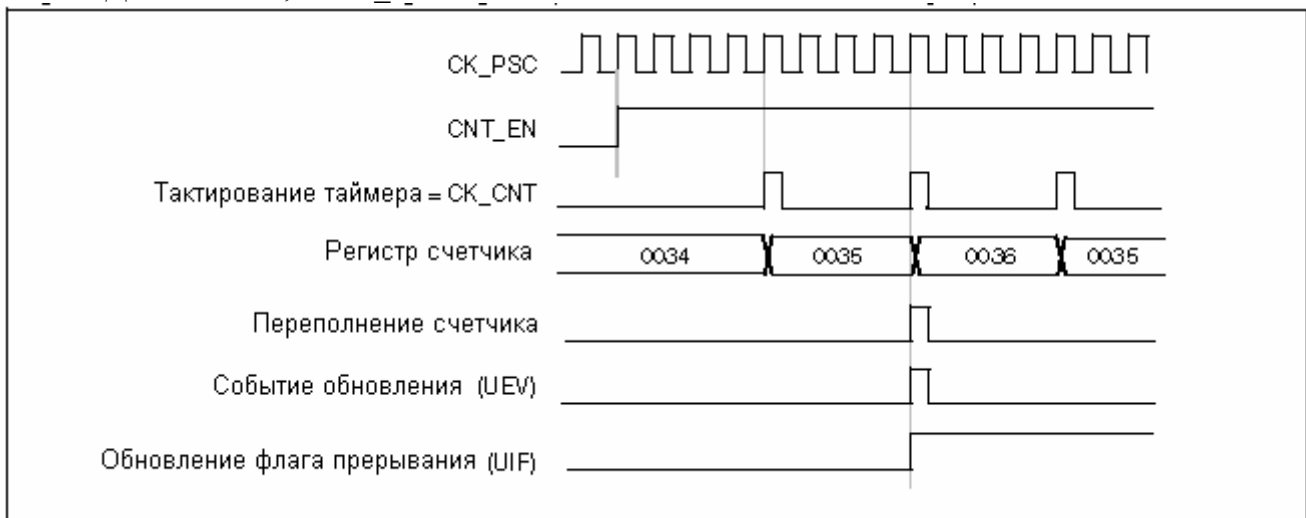


Рис.56. Деление на 4, TIMx\_ARR=0x36



1. Используются режимы 2 и 3 выравнивания по центру с UIF по переполнению.

Рис.57. Деление на N

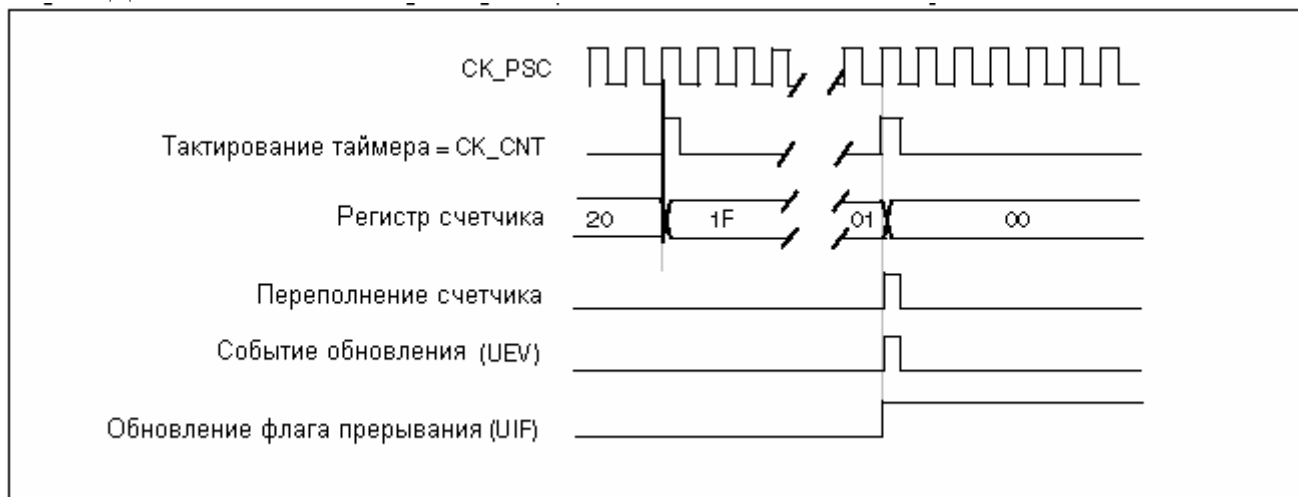


Рис.58. Событие обновления с ARPE=1(переполнение вниз)

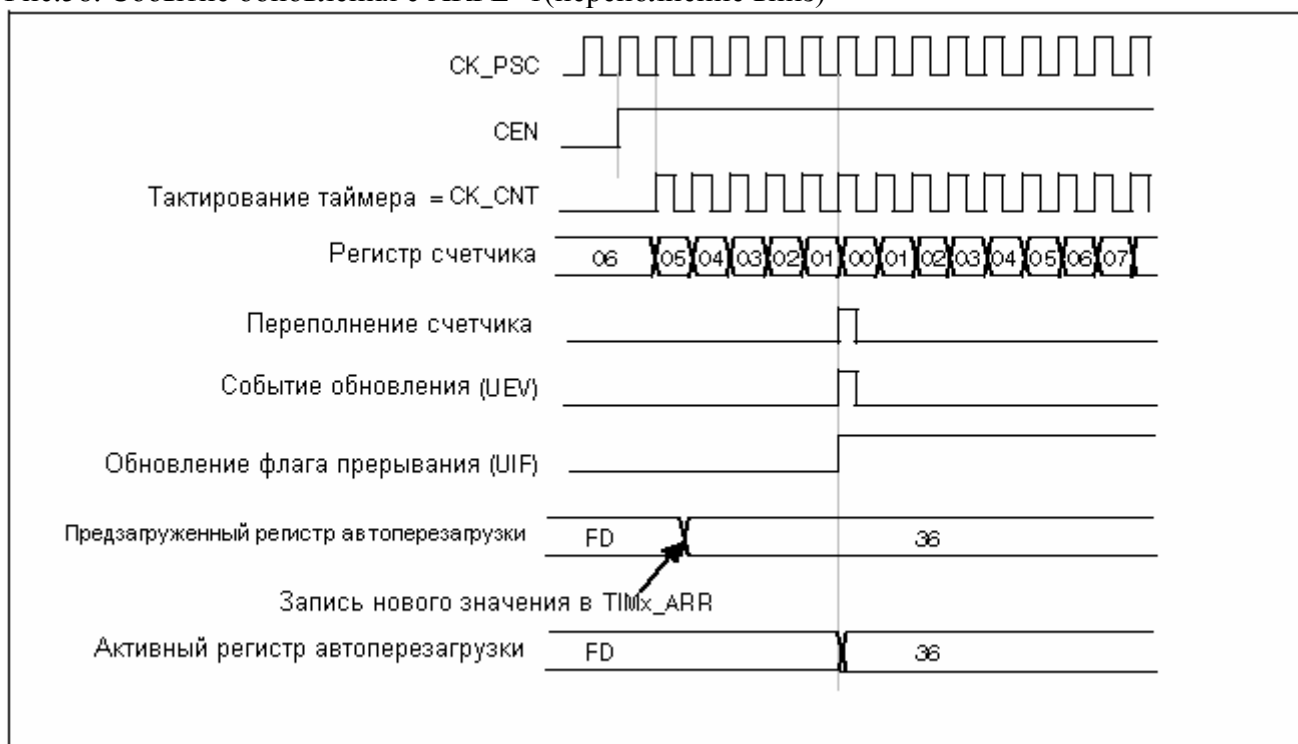
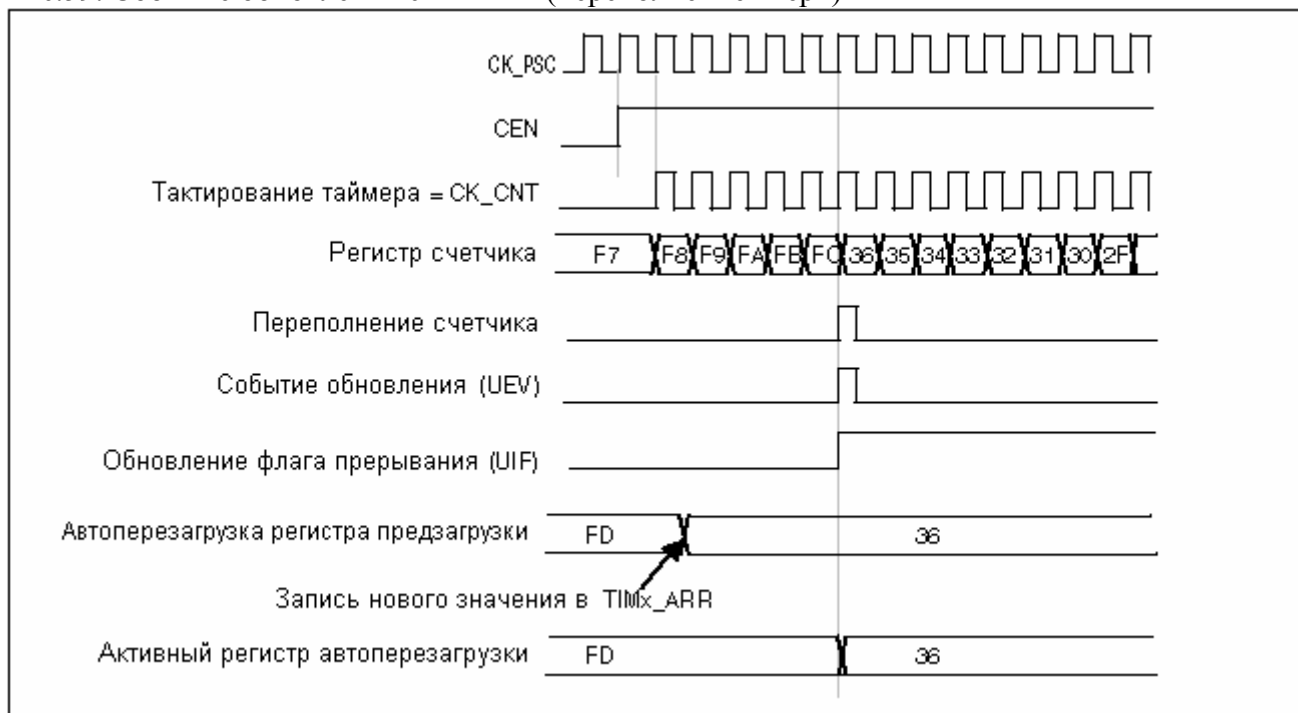


Рис.59. Событие обновления с ARPE=1(переполнение вверх)



### 12.3.3 Счетчик повторов

Раздел 12.3.1 описывает как генерируется событие обновления относительно переполнений счетчика. Оно в действительности генерируется когда счетчик повторов достигает нуля. Это может быть полезным при генерации сигналов ШИМ.

Это означает, что данные передаются из предзагруженных регистров в теньевые регистры (TIMx\_ARR, TIMx\_PSC, и TIMx\_CCR - в режиме сравнения) каждые N переполнений счетчика, где N – значение регистра счетчика повторов TIMx\_RCR.

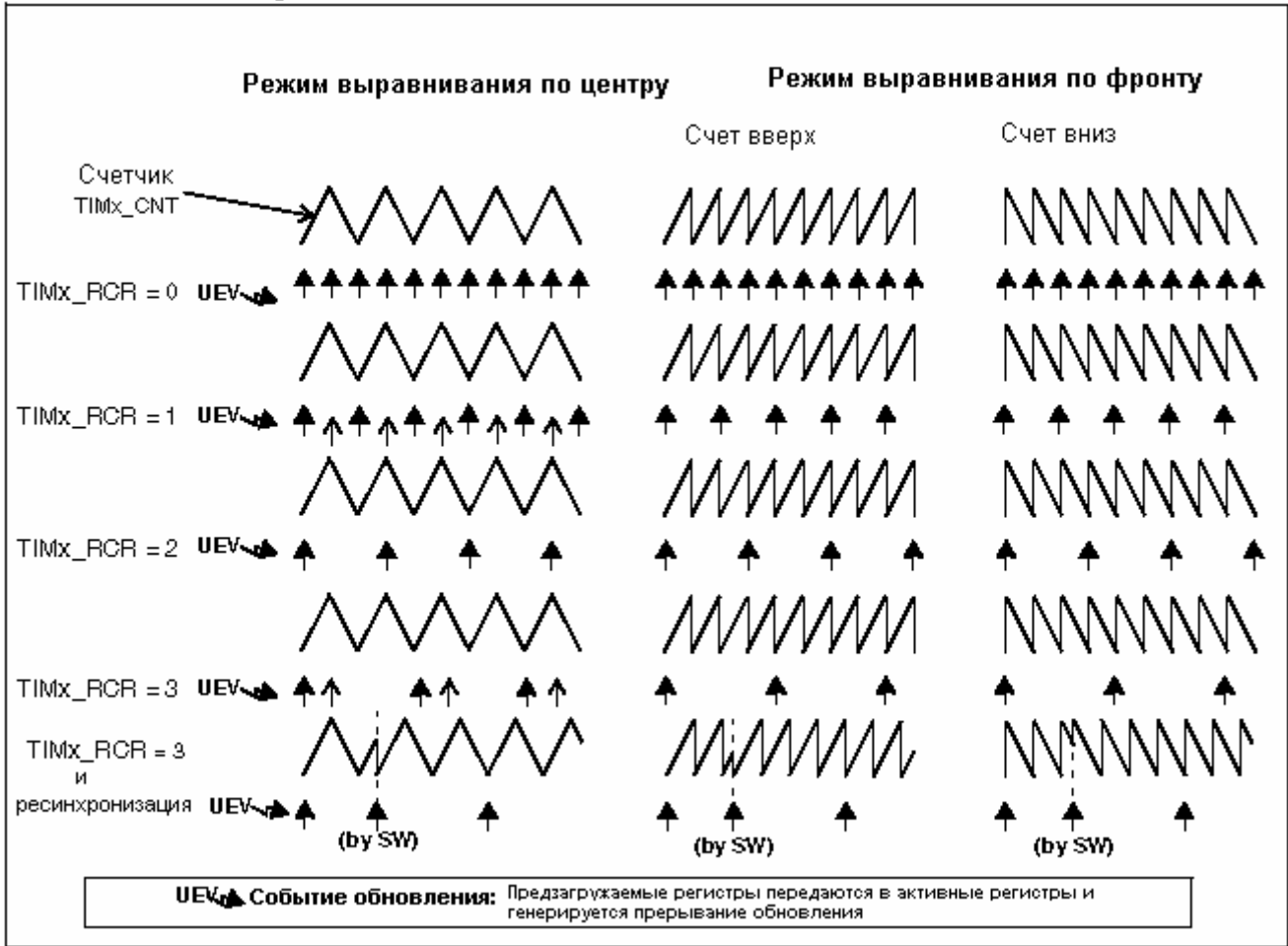
Счетчик повторов декрементируется:

- По каждому переполнению при счете вверх
- По каждому переполнению в режиме счета вниз
- По каждому переполнению в режиме выравнивания по центру.

Хотя максимальное число повторений ограничено до 128 циклов ШИМ, возможно обновлять рабочий цикл ШИМ дважды за период ШИМ. При обновлении регистров сравнения один раз за период ШИМ в режиме выравнивания по центру, максимальное разрешение  $2 \times T_{ck}$  из-за симметричности конфигурации.

Счетчик повторений является автоперезагружаемым. Скорость повторения определяется значением регистра TIMx\_RCR. Когда событие обновления генерируется программно (установкой бита UG) или аппаратно через контроллер подчиненного режима, тогда оно происходит немедленно, какое бы ни было значение счетчика повторов и содержимое регистра TIMx\_RCR.

Рис. 60. Примеры обновления в зависимости от режима и установок регистра TIMx\_RCR



### 12.3.4 Выбор тактирования

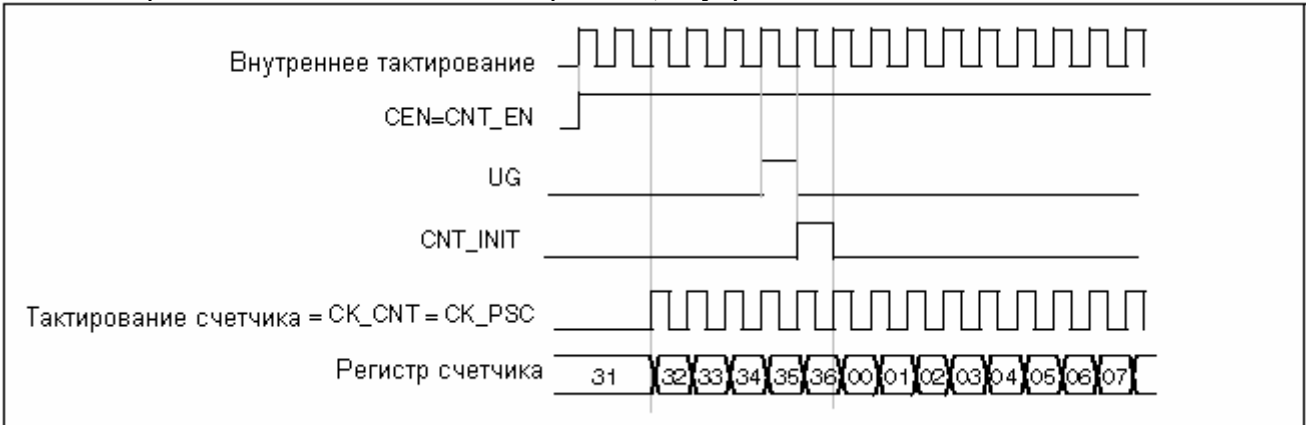
Тактирование счетчика может быть предоставлено следующими источниками:

- Внутреннее тактирование (CK\_INT)
- Режим 1 внешнего тактирования: внешний вход
- Режим 2 внешнего тактирования: вход внешнего запуска ETR
- Входы внутреннего запуска (ITRx): используя один таймер как предделитель, вы можете сконфигурировать таймер 1 как предделитель для таймера 2.

#### Источник внутреннего тактирования (CK\_INT)

Если запрещен контроллер подчиненного режима (SMS=000), тогда биты CEN, DIR и UG являются управляющими битами и могут изменяться только программно (за исключением UG, кот остается очищенным автоматически). Как только бит CEN записывается в 1, предделитель начинает тактироваться внутренним источником CK\_INT. Рис.61 показывает поведение управляющей схемы и счетчика вверх в обычном режиме без предделителя.

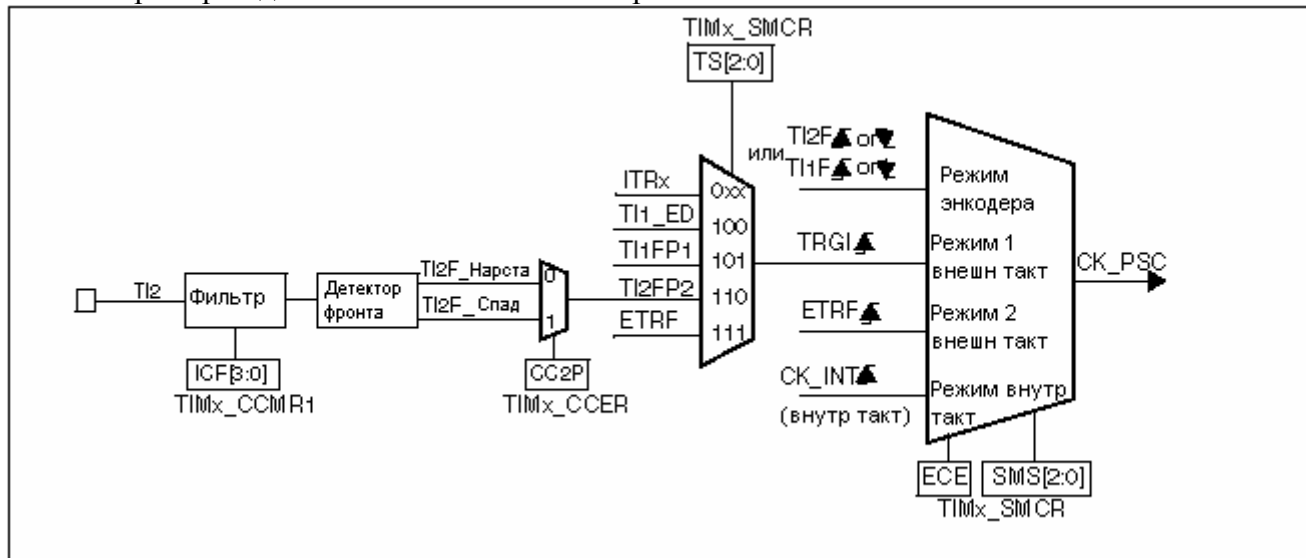
Рис.61. Управляющая схема в обычном режиме, внутр такт делится на 1



### Режим 1 внешнего источника тактирования

Этот режим выбирается когда SMS=111. Счетчик может считать по каждому нарастающему и спадающему фронту на выбранном входе.

Рис.62. Пример подключения внешнего тактирования TI2

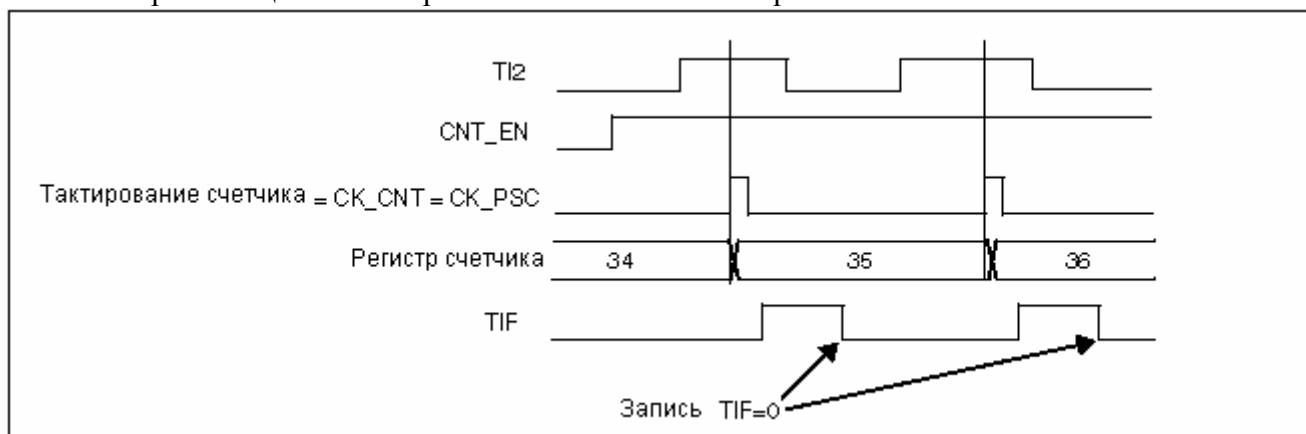


Например, чтобы сконфигурировать счет вверх в ответ на нарастающий фронт по входу TI2 используйте процедуру:

1. Сконфигурировать канал 2 на определение нарастающих фронтов на входе TI2 записью CC2S='01' в TIMx\_CCMR
2. Сконфигурировать длительность входного фильтра записью битов IC2F[3:0](если фильтр не нужен, то IC2F=0000)
3. Выбрать полярность нарастающего фронта записью CC2P=0 в TIMx\_CCER
4. Сконфигурировать таймер в режиме 1 внешнего тактирования записью SMS=111 в TIMx\_SMCR
5. Выбрать TI2 как источник входа запуска записью TS=110 в TIMx\_SMCR
6. Разрешить счетчик записью CEN=1 в TIMx\_CR1

Прим: для запуска прерыватель захвата не используется, поэтому не нужно его конфигурировать.

Рис.63. Управляющая схема в режиме 1 внешнего тактирования

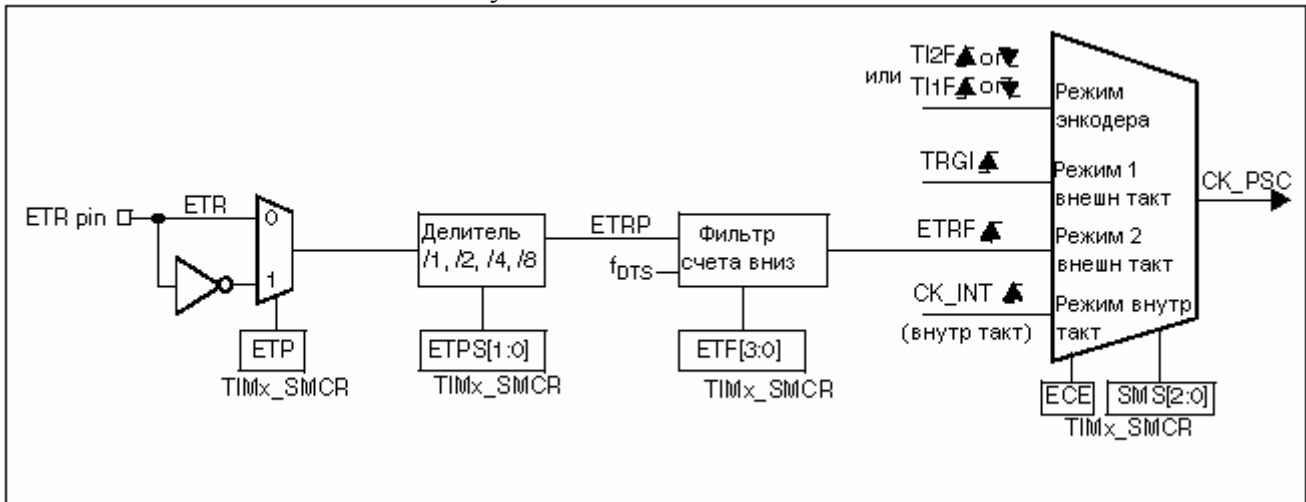


### Режим 2 внешнего источника тактирования

Режим выбирается записью ECE=1 в TIMx\_SMCR

Счетчик может считать по каждому нарастающему или спадающему фронту на входе внешнего запуска ETR.

Рис.64. Входной блок внешнего запуска

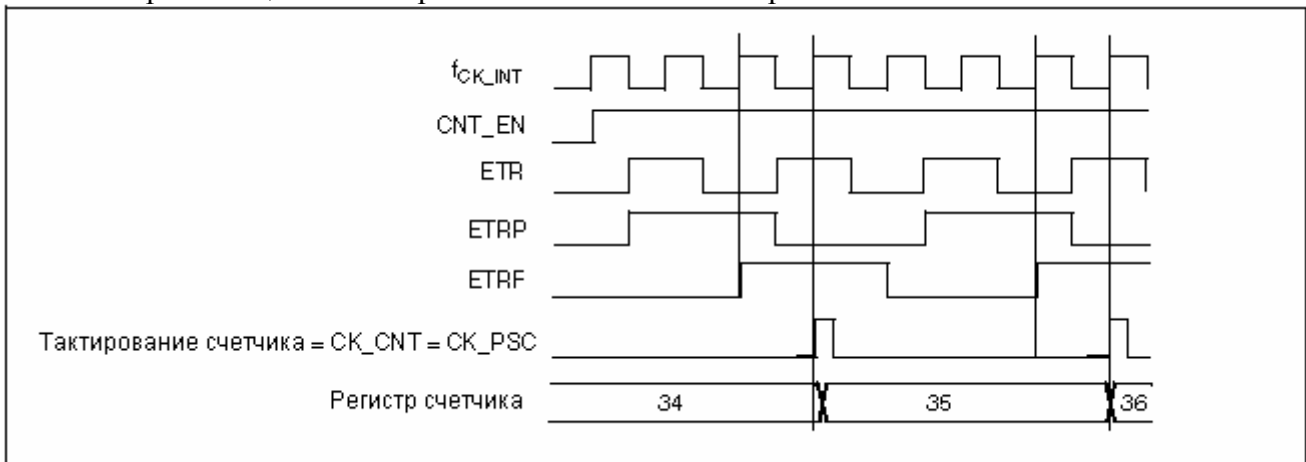


Например, для конфигурации счетчика вверх на счет по каждому 2-м нарастающим фронтам на ETR используйте процедуру:

1. Т.к. в данном примере фильтр не нужен, то записать ETF[3:0]=0000 в TIMx\_SMCR
2. Установить предделитель записью ETPS[1:0]=01 в TIMx\_SMCR
3. Выбрать определение нарастающего фронта на входе ETR записью ETP=0 в TIMx\_SMCR
4. Разрешить режим 2 внешнего тактирования записью ECE=1 в TIMx\_SMCR
5. Разрешить счет записью CEN=1 в TIMx\_CR1

Счетчик считает один раз за каждые 2 нарастающих фронта ETR. Задержка между нарастающим фронтом на ETR и действительным тактом счетчика вызвана схемой ресинхронизации сигнала ETRP.

Рис.65. Управляющая схема в режиме 2 внешнего тактирования



### 12.3.5 Каналы захвата/сравнения

Каждый канал захвата/сравнения состоит из регистра захвата/сравнения(включая теневого регистра), входного каскада для захвата(с цифровым фильтром, мультиплексором и предделителем) и выходного каскада(с компаратором и управлением выходом).

Рис. 66 – Рис.69 предоставляют обзор одного канала захвата/сравнения.

Входной каскад производит выборку соответствующего входа TIx чтобы сгенерировать отфильтрованный сигнал TIxF. Затем детектор фронта с выбираемой полярностью генерирует сигнал(TIxFPx) который может быть использован как вход запуска для контроллера подчиненного режима или команда захвата. Перед захватом регистра он предварительно делится(ICxPS).

Рис.66 Канал захвата/сравнения(в документе рисунок отсутствует)

Выходной каскад генерирует промежуточный сигнал, который затем используется как опорный:

OCxRef(активный высокий). Полярность действует в конце цепи.

Рис.67. Основная схема канала 1 захвата/сравнения

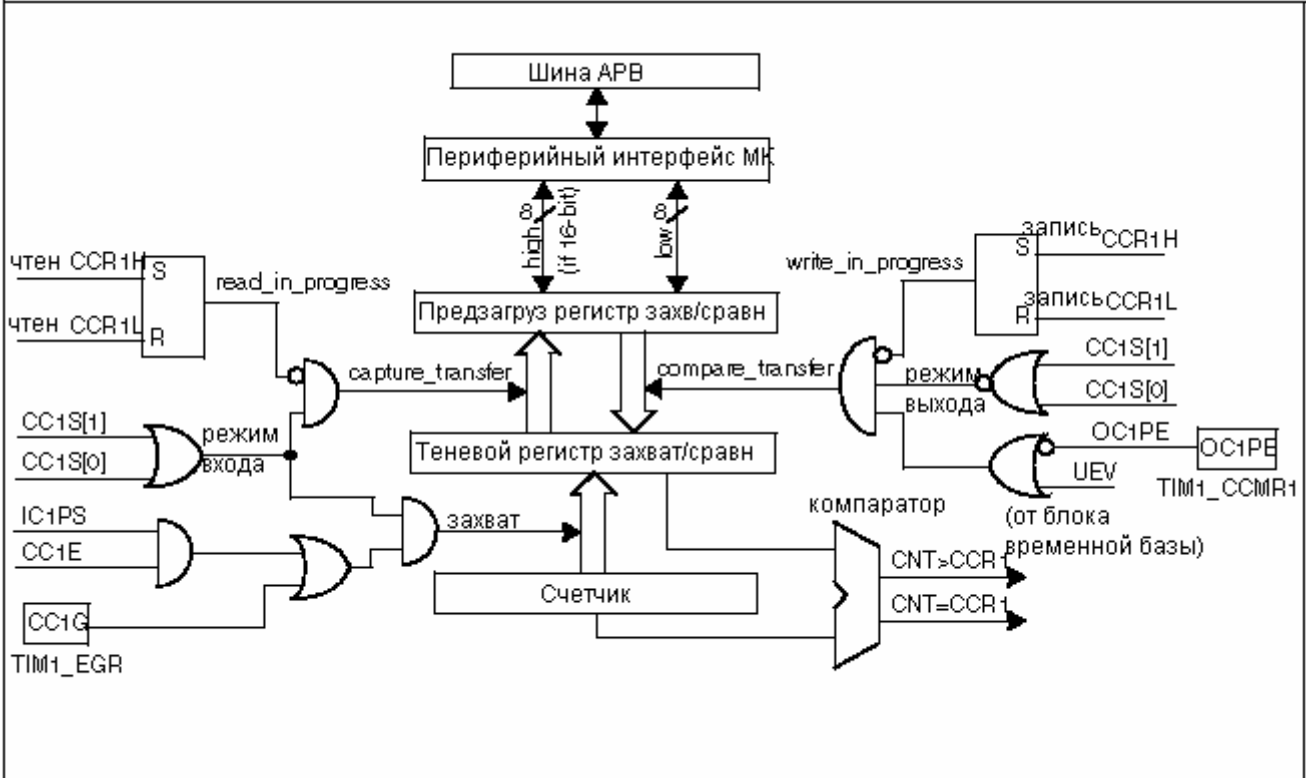


Рис.68. Выходной каскад канала захвата/сравнения(канал 1-3)

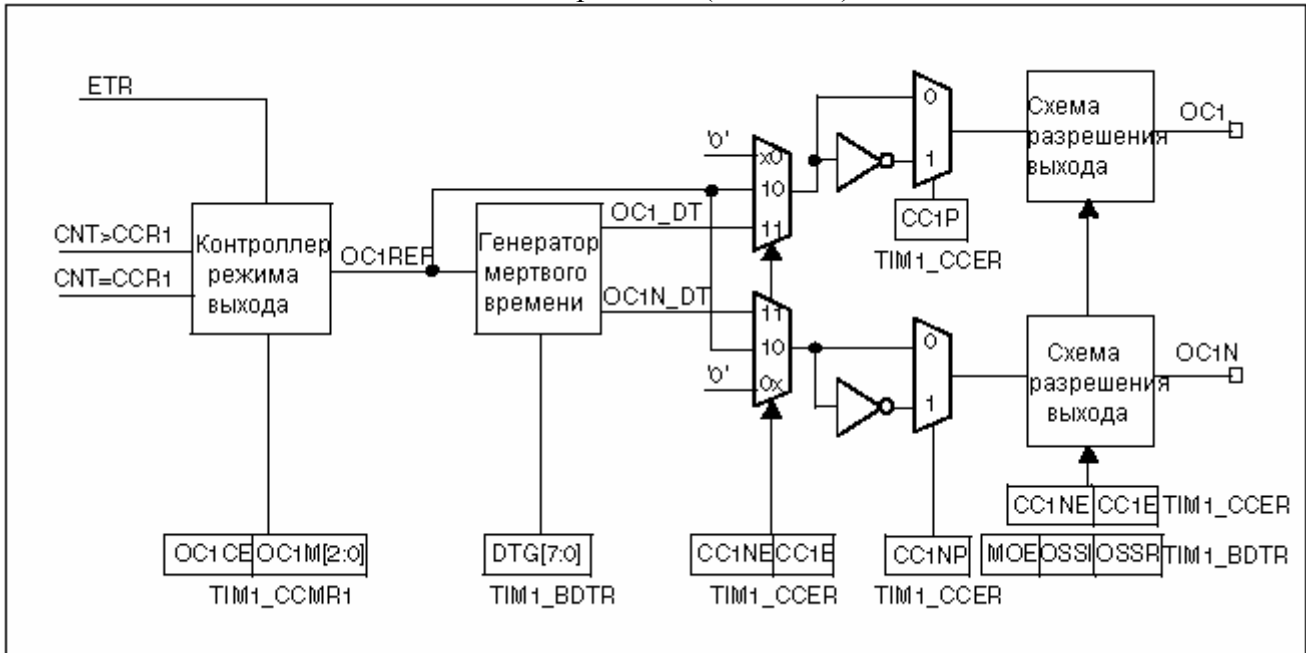
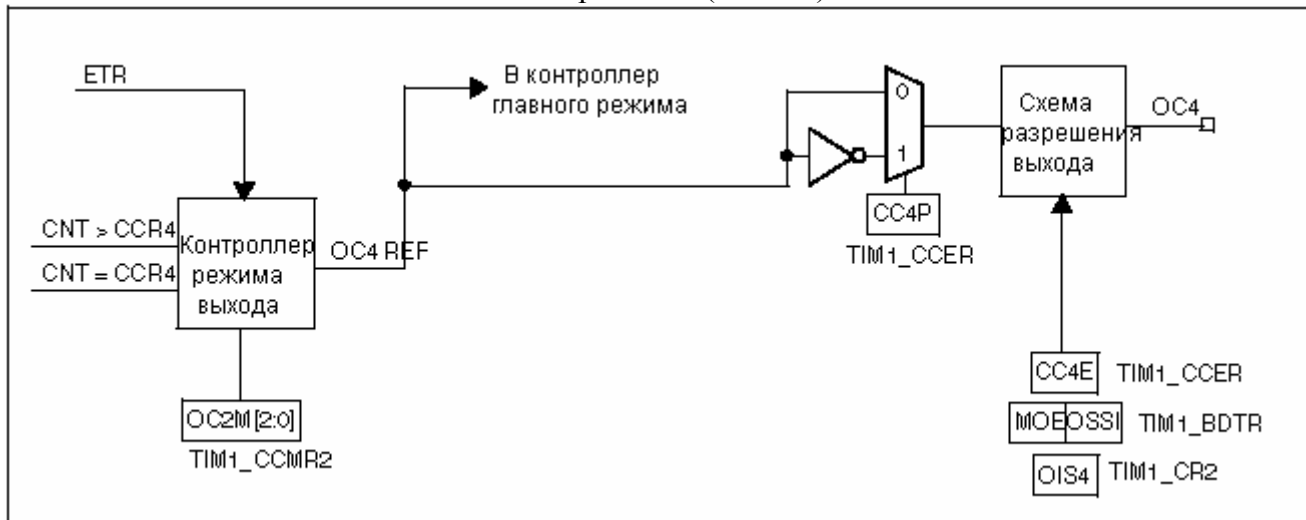




Рис.69. Выходной каскад канала захвата/сравнения(канал 4)



Блок захвата/сравнения состоит из одного регистра предварительной загрузки и одного теневого регистра. При записи и чтении обращение всегда идет к предварительно загруженному регистру. В режиме захвата захват производится в теновой регистр, который копируется в регистр предварительной загрузки.

В режиме сравнения регистр предзагрузки копируется в теновой регистр, который сравнивается со счетчиком.

### 12.3.6 Режим захвата входа

В режиме захвата входа регистры захвата/сравнения (TIMx\_CCRx) используются для зашелкивания значения счетчика после обнаружения перехода на соответствующем сигнале ICx. Когда происходит захват, то устанавливается соответствующий флаг CCxIF (регистр TIMx\_SR) и прерывание либо запрос DMA могут посылаются если они разрешены. Если захват происходит когда флаг CCxIF уже был высоким, тогда устанавливается флаг перезахвата CCxOF (TIMx\_SR). CCxIF может быть очищен программно, путем записи в него 0, или чтением захваченных данных из регистра TIMx\_CCRx. CCxOF очищается записью в него 0.

Следующий пример показывает как захватить значение счетчика в TIMx\_CCR1 когда нарастает ТП1:

- Выбрать активный вход: TIMx\_CCR1 должен быть связан с входом ТП1, поэтому запишите биты CC1S=01 в TIMx\_CCMR1. Как только CC1S становится отличным от нуля, канал конфигурируется на вход и регистр TIMx\_CCR1 становится только для чтения.
- Запрограммировать требуемую длительность входного фильтра относительно сигнала, подключаемого к таймеру (биты ICxP регистра TIMx\_CCRx). Представим, что при смене входного сигнала он нестабилен 5 тактовых циклов. Мы должны запрограммировать длительность фильтра больше этих 5 тактовых циклов. Мы можем принять переход на ТП1 когда было обнаружено 8 последовательных выборок с новым уровнем (выбранные на частоте fDTS). Тогда запишите биты IC1F=0011 в регистр TIMx\_CCMR1.
- Выбрать фронт активного перехода на ТП1 записью CC1P=0 в TIMx\_CCER (нарастающий фронт)
- Запрограммировать входной предделитель. В нашем случае мы хотим делать захват по каждому действительному переходу поэтому предделитель отключен (IC1PS=00 в TIMx\_CCMR1)
- Разрешить захват из счетчика в регистр захвата установкой бита CC1E в TIMx\_CCER
- Если нужно, то разрешить соответствующий запрос прерывания установкой бита CC1IE и/или запрос DMA установкой бита CC1DE в TIMx\_DIER.

Когда происходит захват входа:

- Регистр TIMx\_CCR1 по активному переходу получает значение счетчика
- Устанавливается флаг CC1IF(флаг прерывания). CC1OF также устанавливается если произошло по крайней мере два последовательных захвата когда флаг не был очищен.
- Генерируется прерывание в зависимости от бита CC1IE
- Генерируется запрос DMA в зависимости от бита CC1DE

Чтобы обработать перезахват рекомендуется читать данные перед флагом перезахвата. Это предотвратит пропуск перезахвата, который может произойти после чтения флага и до чтения данных.

Прим: Прерывание и/или запрос DMA от IC может генерироваться программно установкой соответствующего бита CCxG регистра TIMx\_EGR.

### 12.3.7 Режим входа ШИМ

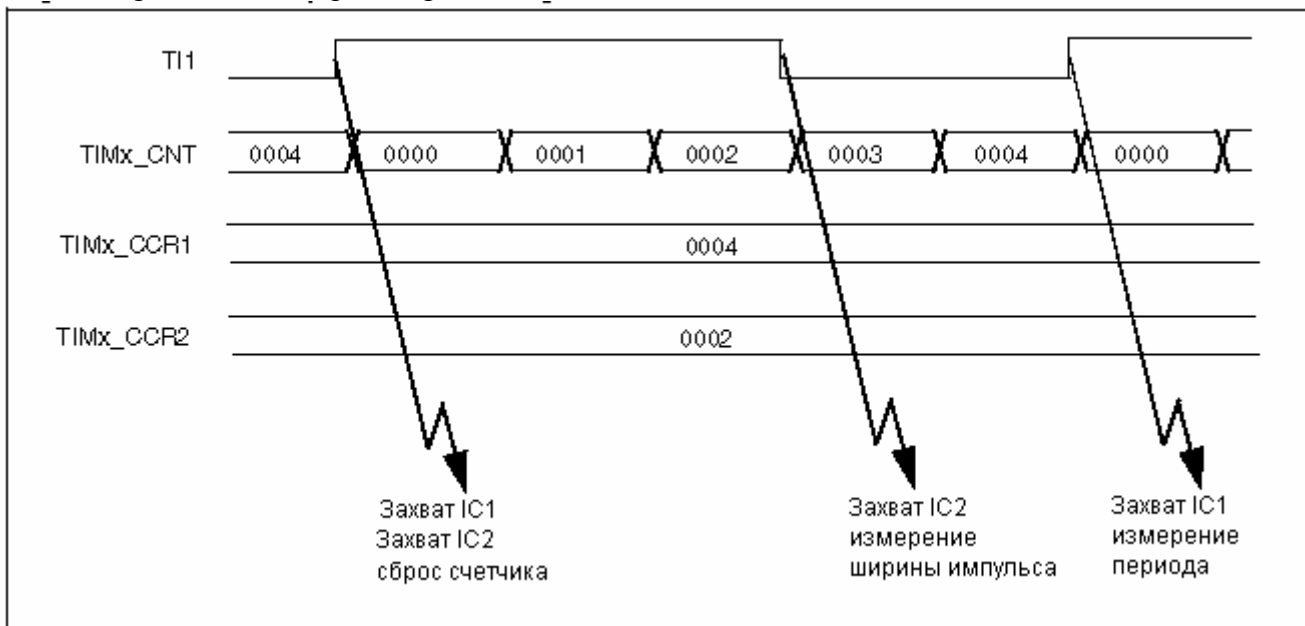
Это частный случай режима захвата входа. Процедура та же за исключением:

- Два сигнала ICx отображаются на один вход TIx
- Эти 2 сигнала ICx активны по фронтам с противоположной полярностью
- Один из двух сигналов TIxFP выбирается как вход запуска и контроллер подчиненного режима конфигурируется в режиме сброса.

Пример. Вы можете измерить период(TIMx\_CCR1) и рабочий цикл(TIMx\_CCR2) ШИМ, подключенного к TI1 используя следующую процедуру(зависит от частоты CK\_INT и значения делителя):

- Для TIMx\_CCR1 выбрать активный вход: записать CC1S=01 в TIMx\_CMR1(выбран TI1)
- Выбрать активную полярность для TI1FP1(используется для захвата в TIMx\_CCR1 и очистки счетчика): записать CC1P=0(активен по нарастающему фронту)
- Выбрать активный вход для TIMx\_CCR2: записать CC2S=10 в TIMx\_CCMR1(выбран TI1)
- Выбрать активную полярность для TI1FP2(используется для захвата в TIMx\_CCR2): записать CC2P=1(активен по спадающему фронту)
- Выбрать действующий вход запуска: записать TS=101 в TIMx\_SMCR(выбран TI1FP1)
- Сконфигурировать контроллер подчиненного режима: записать SMS=100 в TIMx\_SMCR
- Разрешить захваты: записать 1 в биты CC1E и CC2E

Рис. 70. Временная диаграмма режима входа ШИМ



### 12.3.8 Режим форсированного выхода

В режиме выхода (CCxS=00 в TIMx\_CCMRx) каждый сигнал выхода сравнения (OCxREF и OCx/OCxN) может быть напрямую программно переведен в активный или неактивный уровень вне зависимости от состояния сравнения между регистром сравнения выхода и счетчиком.

Чтобы форсировать выходной сигнал сравнения (OCxREF/OCx) в его активный уровень вам нужно записать OCxM=101 в TIMx\_CCMRx. Соответственно, OCxREF форсируется высоким (OCxREF всегда активный высокий), а OCx получает противоположное значение полярности бита CCxP.

Например: CCxP=0 (OCx активный высокий) => OCx форсируется высоким уровнем.

Сигнал OCxREF может форсироваться низким уровнем через запись 100 в биты OCxM регистра TIMx\_CCMRx.

В любом случае все еще производится сравнение между теневым регистром TIMx\_CCRx и счетчиком. И это позволяет устанавливать флаг. Соответственно могут посылаться прерывания и запросы DMA.

### 12.3.9 Режим сравнения выхода

Эта функция используется для управления выходным сигналом или индикации когда истек период времени. Когда обнаружено совпадение между регистром захвата/сравнения и счетчиком, то функция сравнения выхода:

- Переводит соответствующий выход в запрограммированное значение, определяемое режимом сравнения выхода (биты OCxM в регистре TIMx\_CCMRx) и выходной полярностью (CCxP в TIMx\_CCRx). Выход по совпадению может удерживать свой уровень (OCxM=000), устанавливаться активным (OCxM=001), устанавливаться неактивным (OCxM=010) или изменяться (OCxM=011)
- Устанавливает флаг в регистре статуса прерывания (бит OCxIF в TIMx\_SR)
- Генерирует прерывание если установлена соответствующая маска прерывания (бит CCxIE в TIMx\_DIER)
- Посылает запрос DMA если установлен соответствующий бит разрешения (бит CCxDE в TIMx\_DIER, бит CCDS в TIMx\_CR2 для выбора запроса DMA).

Регистры TIMx\_CCRx могут программироваться с- или без регистров предварительной загрузки путем использования бита OCxPE в TIMx\_CCMRx.

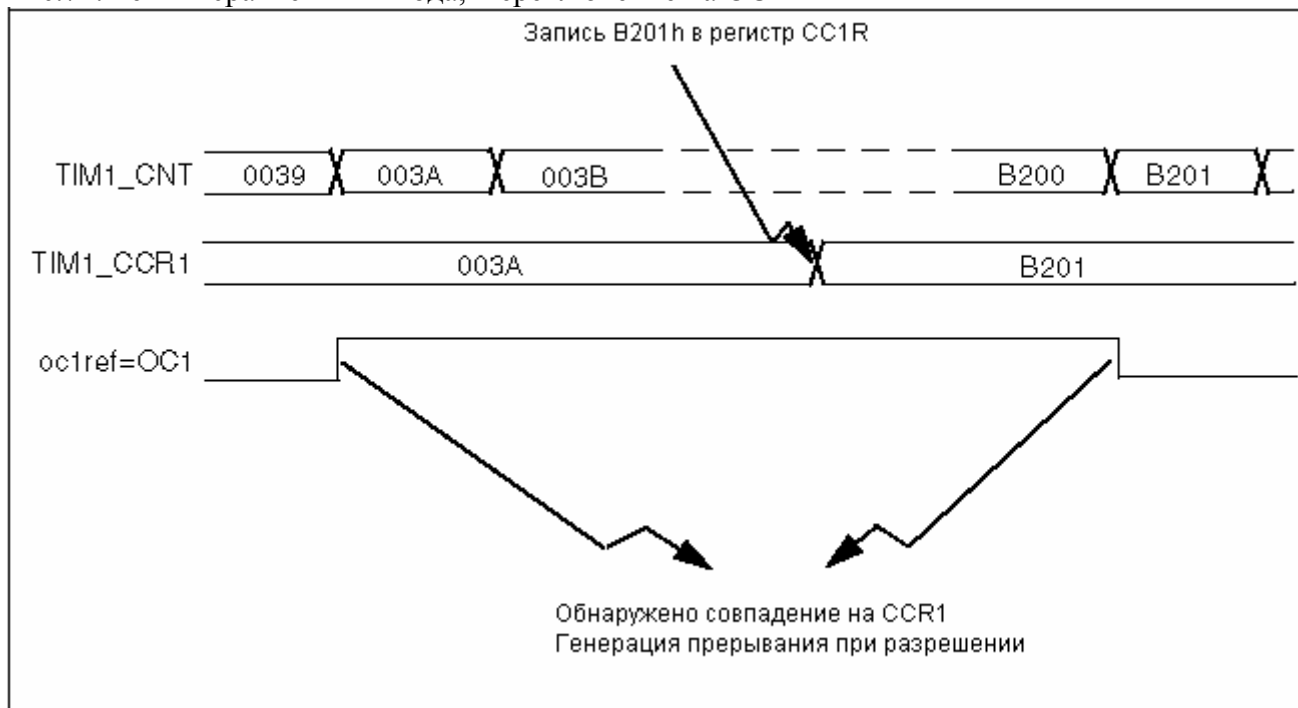
В режиме сравнения выхода событие обновления UEV не влияет на выходы OCxREF и OCx. Временное разрешение – 1 такт счетчика. Режим сравнения выхода может также использоваться для вывода одиночного импульса (режим одного импульса).

Процедура:

1. выбрать тактирование счетчика (внутр, внешн, предделитель)
2. записать желаемые данные в регистры TIMx\_ARR и TIMx\_CCR
3. установить бит CCxIE если нужна генерация запроса прерывания
4. выбрать режим выхода. Например:
  - записать OCxM=011 для смены выхода OCx когда CNT совпадает с CCRx
  - OCxPE=0 для запрещения регистра предзагрузки
  - CCxP=0 для выбора высокой активной полярности
  - CCxE=1 для разрешения выхода
5. Разрешить счетчик установкой бита CEN регистра TIMx\_CR1

Регистр TIMx\_CCRx может обновляться программно в любой момент для управления выходным сигналом при условии, что регистр предзагрузки не разрешен (OCxPE=0, иначе теневой регистр TIMx\_CCRx обновляется только по следующему событию обновления UEV). Пример на рисунке 71.

Рис.71. Режим сравнения выхода, переключение на OC1



### 12.3.10 Режим ШИМ

Режим Широтно-Импульсной модуляции позволяет генерировать сигнал, частота которого определяется значением регистра TIMx\_ARR, а рабочий цикл – значением регистра TIMx\_CCRx. ШИМ может быть выбран независимо на каждом канале (один ШИМ на один выход OCx) записью в биты OCxM регистра TIMx\_CCMRx значений '110' (режим 1 ШИМ) или '111' (режим 2 ШИМ). Вы должны разрешить соответствующую предварительную загрузку регистра установкой бита OCxPE в TIMx\_CCMRx и в конечном счете регистр предварительной автоперезагрузки (в режимах счета вверх или выравнивания по центру) установкой бита ARPE в регистре TIMx\_CR1.

Т.к. предварительно загруженные регистры передаются в теньевые регистры только когда происходит событие обновления, то перед запуском счетчика вы должны инициализировать все регистры установкой бита UG в TIMx\_EGR.

Полярность OCx задается программно через бит CCxP регистра TIMx\_CCER. Она может быть запрограммирована как активный высокий уровень или активный низкий. Выход OCx разрешается комбинацией битов CCxE, CCxNE, MOE, OSSI и OSSR (регистры TIMx\_CCER и TIMx\_BDTR).

В режиме ШИМ (1 или 2) TIMx\_CNT и TIMx\_CCRx всегда сравниваются на условие TIMx\_CCRx ≤ TIMx\_CNT или TIMx\_CNT ≤ TIMx\_CCRx (в зависимости от направления счетчика).

Таймер может генерировать ШИМ в режиме выравнивания по фронту или по центру в зависимости от битов CMS регистра TIMx\_CR1.

#### Режим ШИМ с выравниванием по фронту

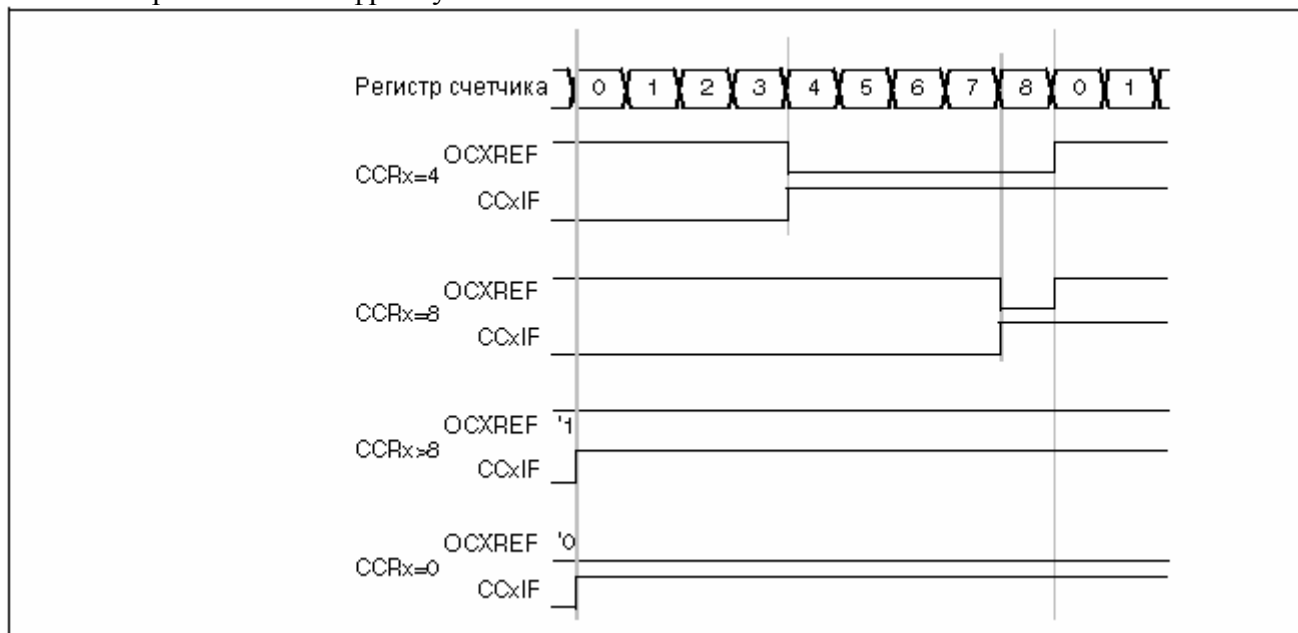
- Счет вверх

Счет вверх активен, когда бит DIR регистра TIMx\_CR1 низкий.

В следующем примере мы обратимся к режиму 1. Опорный сигнал ШИМ OCxREF высокий пока TIMx\_CNT < TIMx\_CCRx, иначе он становится низким. Если значение сравнения регистра TIMx\_CCRx больше значения автоперезагрузки (TIMx\_ARR), тогда OCxREF удерживается в 1. Если значение сравнения 0, тогда OCxREF удерживается на 0.

Рис.72 показывает примеры некоторых сигналов, выровненных по фронту ШИМ, когда TIMx\_ARR=8.

Рис.72. Выравненные по фронту сигналы ШИМ



- Счет вниз

Счет вниз активен когда бит DIR регистра высокий. В режиме 1 ШИМ опорный сигнал OCxREF низкий пока  $TIMx\_CNT > TIMx\_CCRx$ , иначе он становится высоким. Если сравниваемое значение в  $TIMx\_CCRx$  больше чем автоперезагружаемое значение  $TIMx\_ARR$ , тогда OCxREF удерживается в 1. 0% ШИМ невозможна в этом режиме.

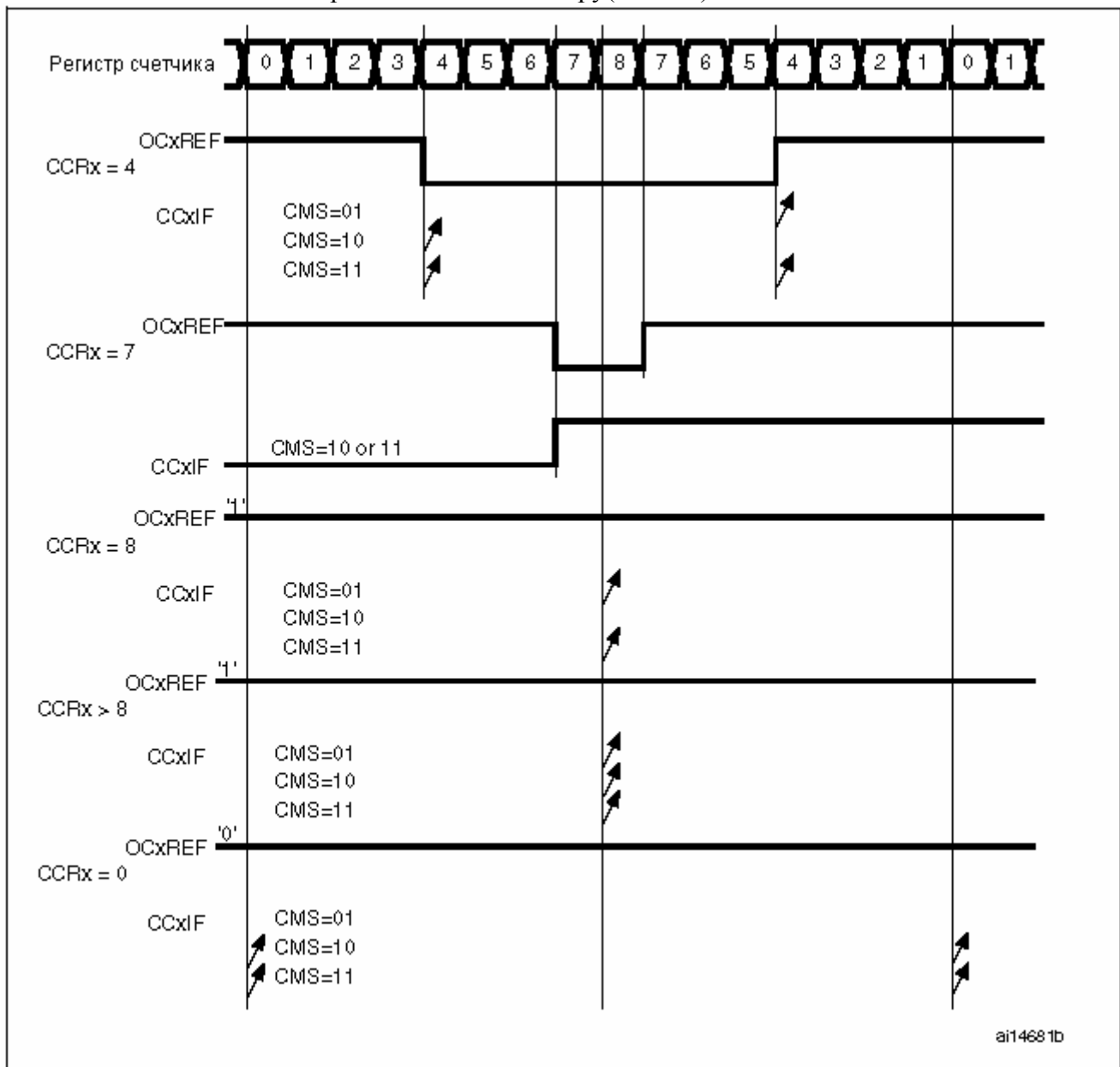
### Режим ШИМ с выравниванием по центру

Режим с выравниванием по центру активируется когда биты CMS в регистре  $TIMx\_CR1$  отличны от 00(все остальные конфигурации имеют одинаковое влияние на сигналы OCxREF/ OCx). Флаг сравнения устанавливается когда счетчик считает вверх, когда он считает вниз или когда считает вверх-вниз, в зависимости от конфигурации битов CMS. Бит направления (DIR) регистра  $TIMx\_CR1$  обновляется аппаратно и не должен изменяться программой.

Рис.73 показывает сигналы выравненной по центру ШИМ где:

- $TIMx\_ARR=8$
- Режим 1 ШИМ
- Флаг устанавливается когда счетчик считает вниз в соответствии с режимом 1 с выравниванием по центру, выбранным для CMS=01 в регистре  $TIMx\_CR1$ .

Рис.73. Сигналы ШИМ с выравниванием по центру(ARR=8)



### Подсказки по использованию режима с выравниванием по центру:

- При запуске в режиме выравнивания по центру используется текущая конфигурация вверх-вниз. Это значит, что счетчик считает вверх или вниз в зависимости от значения, записанного в бит DIR регистра TIMx\_CR1. Более того, биты DIR и CMS не должны меняться программой в одно время.
- Запись в счетчик, работающий в режиме выравнивания по центру, не рекомендуется т.к. это может привести к непредсказуемым результатам. В частности:
  - Направление не обновляется если вы записываете значение в счетчик, которое больше чем значение автоперезагрузки (TIMx\_CNT > TIMx\_ARR). Например, если счетчик считал вверх, то он продолжает счет вверх.
  - Направление обновляется если вы записываете 0 или записываете значение TIMx\_ARR в счетчик, но событие обновления UEV не генерируется.
- Самый безопасный путь использования режима выравнивания по центру – генерировать обновление программно (установкой бита UG регистра TIMx\_EGR) перед запуском счетчика и не записывать в счетчик пока он работает.

### 12.3.11 Комплементарные выходы и вставка мертвого времени

Таймер улучшенного управления (TIM1) может выводить два комплементарных сигнала и управлять моментами включения/отключения выходов. Это время известно как «мертвое время» и вам требуется подстраивать его под характеристики устройства, которое подключено к выходам (внутренние задержки преобразователей уровня, задержки переключения питания...).

Вы можете выбирать полярность выходов (главный выход OCx или комплементарный OCxN) независимо под каждый выход. Это делается записью в биты CCxP и CCxNP регистра TIMx\_CCER. Комплементарные сигналы OCx и OCxN активируются комбинацией нескольких управляющих битов: CCxE и CCxNE в регистре TIMx\_CCER и MOE, OISx, OISxN, OSSI и OSSR в TIMx\_BDTR и TIMx\_CR2. В частности, мертвое время активируется при переключении в состояние простоя (MOE переходит в 0).

Вставка мертвого времени разрешается установкой битов CCxE и CCxNE и бита MOE если присутствует схема аварийного останова. Для каждого канала имеется 10-битный генератор мертвого времени. Из опорного сигнала OCxREF он генерирует два выхода OCx и OCxN. Если OCx и OCxN активные высокие:

- Выходной сигнал OCx такой же как опорный, за исключением того, что нарастающий фронт задерживается относительно опорного нарастающего фронта
- Выходной сигнал OCxN противоположен опорному сигналу за исключением того, что нарастающий фронт задерживается относительно опорного спадающего фронта.

Если задержка больше ширины активного выхода (OCx или OCxN), тогда соответствующий импульс не генерируется. Следующие рисунки показывают соотношения между выходными сигналами генератора мертвого времени и опорным сигналом OCxREF. (в этих примерах мы полагаем что CCxP=0, CCxNP=0, MOE=1, CCxE=1 и CCxNE=1)

Рис. 74. Комплементарный выход с вставкой мертвого времени

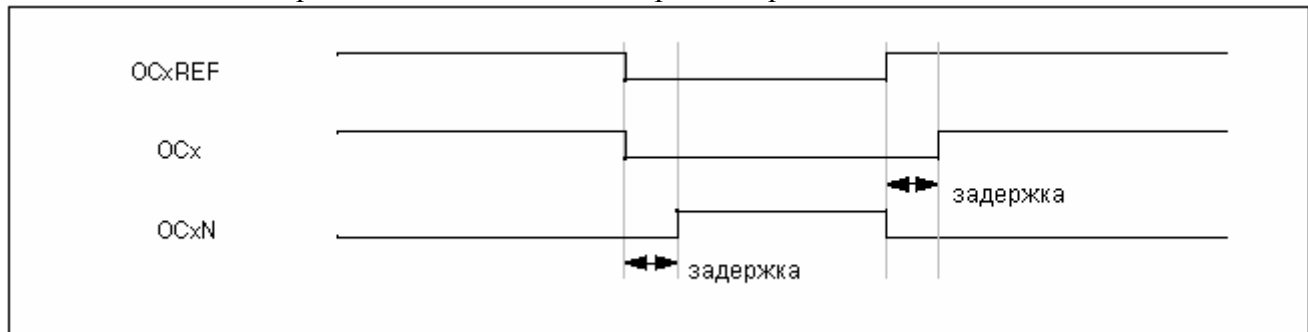


Рис. 75. Сигнал мертвого времени с задержкой больше чем отрицательный импульс

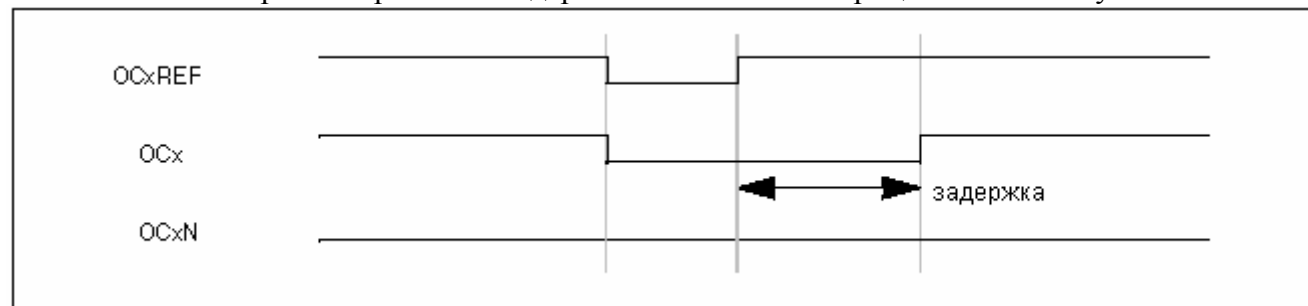
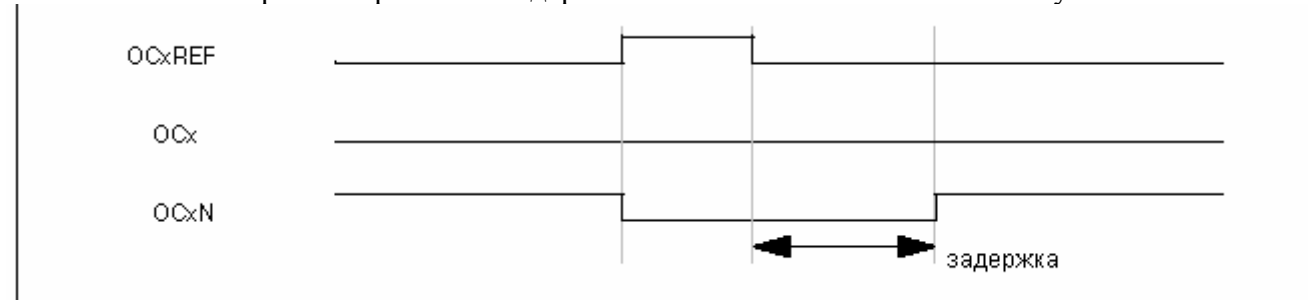


Рис. 76. Сигнал мертвого времени с задержкой больше положительного импульса



Задержка мертвого времени одинакова для каждого канала и программируется через биты DTG регистра TIMx\_BDTR.

### **Перенаправление OCxREF на OCx или OCxN**

В режиме выхода(форсированный, выход сравнения или ШИМ) OCxREF может быть перенаправлен на выход OCx или OCxN через конфигурирование битов CCxE и CCxNE регистра TIMx\_CCER. Это позволяет послать специфичные сигналы(как ШИМ или статический уровень) на один выход, когда комплементарный остается в своем неактивном уровне. Другие альтернативные возможности – получить два выхода с неактивным уровнем или два активных выхода и комплементарных с мертвым временем.

Прим: Когда разрешен только OCxN(CCxE=0, CCxNE=1) он некомплементарный и остается активным пока OCxREF высокий. Например, если CCxNP=0, тогда OCxN= OCxREF. С другой стороны, когда оба OCx и OCxN разрешены(CCxE= CCxNE=1) OCx остается активным когда OCxREF высокий, тогда как OCxN - комплементарный и остается активным когда OCxREF низкий.

### **12.3.12 Использование функции аварийного останова**

При использовании функции останова сигналы разрешения выхода и неактивные уровни модифицируются в соответствии с битами дополнительного управления(MOE, OSSI, OSSR регистра TIMx\_BDTR, OISx и OISxN регистра TIMx\_CR2). В любом случае в данный момент на выходах OCx и OCxN не может быть установлен активный уровень.

Источником останова может быть как вход останова, так и событие сбоя тактирования, генерируемое Системой Безопасности Тактирования(CSS).

По выходу из сброса схема останова отключена и бит MOE низкий. Вы можете разрешить функцию останова установкой бита VKE регистра TIMx\_BDTR. Полярность входа останова выбирается конфигурацией бита VKP в этом регистре. VKE и VKP могут модифицироваться в один момент. Когда записываются биты VKE и VKP, то имеет место задержка в 1 тактовый цикл APB перед тем как запись начинает действовать. Поэтому после операции записи нужно выждать 1 тактовый период APB чтобы корректно прочитать бит.

Т.к. спадающий фронт MOE может быть асинхронным, то была добавлена схема ресинхронизации между действительным сигналом(действующим на выводах) и битом синхронного управления(регистра TIMx\_BDTR). В результате имеется некоторая задержка между асинхронным и синхронным сигналами. В частности, если вы записываете MOE в 1 когда он был низким, тогда вы должны будете вставить задержку(пустую инструкцию) перед его корректным чтением. Это вызвано тем, что вы записываете асинхронный, а читаете синхронный сигнал.

Когда происходит останов(выбранный уровень на входе останова):

- Асинхронно очищается бит MOE, помещая выходы в неактивное состояние, состояние простоя или в состояние сброса(выбирается битом OSSI). Эта возможность функционирует даже тогда, когда генератор МК отключен.
- Каждый выходной сигнал переводится в уровень, запрограммированный битом OISx регистра TIMx\_CR2 пока MOE=0. Если OSSI=0 тогда таймер отпускает разрешение выхода, иначе разрешение выхода остается высоким.
- Когда используются комплементарные выходы:
  - Выходы сперва помещаются в состояние сброса/неактивное состояние(в зависимости от полярности). Это происходит асинхронно, поэтому функционирует даже если тактирование не подается на таймер.
  - Если тактирование таймера присутствует, тогда активизируется генератор мертвого времени чтобы после мертвого времени перевести выходы в уровни, запрограммированные битами OISx и OISxN. Даже в этом случае OCx и OCxN не могут вместе переводиться в их активный уровень. Прим: из-за ресинхронизации на MOE, длительность мертвого времени немного больше чем обычно(около 2 тактовых циклов ck\_tim).
  - Если OSSI=0, тогда таймер отпускает разрешение выходов, иначе разрешенные выходы остаются или становятся высокими как только один из битов CCxE или CCxNE будет высокий.



- Устанавливается статусный флаг останова(бит BIF регистра TIMx\_SR). Если установлен бит BIE регистра TIMx\_DIER, то может генерироваться прерывание. Если установлен бит BDE регистра TIMx\_DIER, то может быть послан запрос DMA.
- Если установлен бит AOE в TIMx\_BDTR, тогда бит MOE снова автоматически устанавливается по следующему событию обновления UEV. Это, к примеру, может использоваться чтобы произвести регулировку. В противном случае MOE остается низким пока вы снова не запишите туда 1. В этом случае он может использоваться для целей безопасности когда вы подключаете ко входу останова сигнал от мощных драйверов, датчика температуры или другие компоненты безопасности.

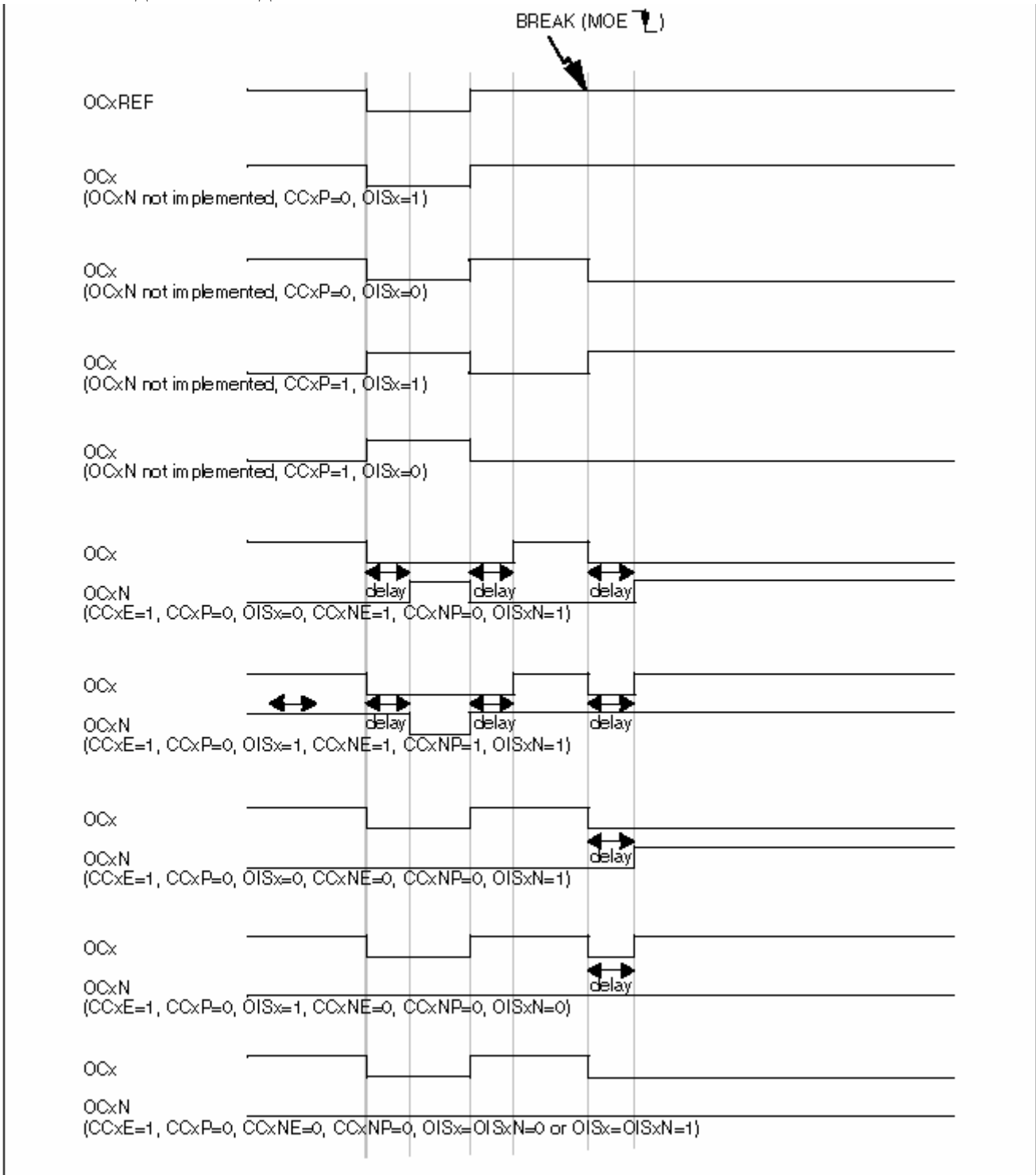
Прим: Входы останова действуют по уровню. Поэтому MOE не может быть установлен пока активен вход останова(ни программно, ни автоматически). В это время статусный флаг BIF не может быть очищен.

Останов может генерироваться входом BRK с программируемой полярностью и битом разрешения VKE регистра TIMx\_BDTR.

Дополнительно ко входу останова и управлением выходами в схему останова для безопасности приложения была добавлена защита записи. Она позволяет заморозить конфигурацию по нескольким параметрам(длительность мертвого времени, полярность OCx/OCxN, конфигурацию OCxM, разрешение и полярность останова). Вы можете выбрать из 3 уровней защиты через выбор битов LOCK регистра TIMx\_BDTR. Биты LOCK могут быть записаны только после сброса МК.

Рис.77 показывает пример поведения выходов в ответ на останов.

Рис.77. Поведение выхода в ответ на останов



### 12.3.13 Очистка сигнала OCxREF по внешнему событию

Сигнал OCxREF на данном канале может переводиться в низкое состояние путем подачи высокого уровня на вход ETRF(бит разрешения OCxCE соответствующего регистра TIMx\_CCMRx установлен в 1). Сигнал OCxREF остается низким пока не произойдет следующее событие обновления UEV. Данная функция может использоваться только в режимах сравнения выхода и ШИМ, и не работает в форсированном режиме.

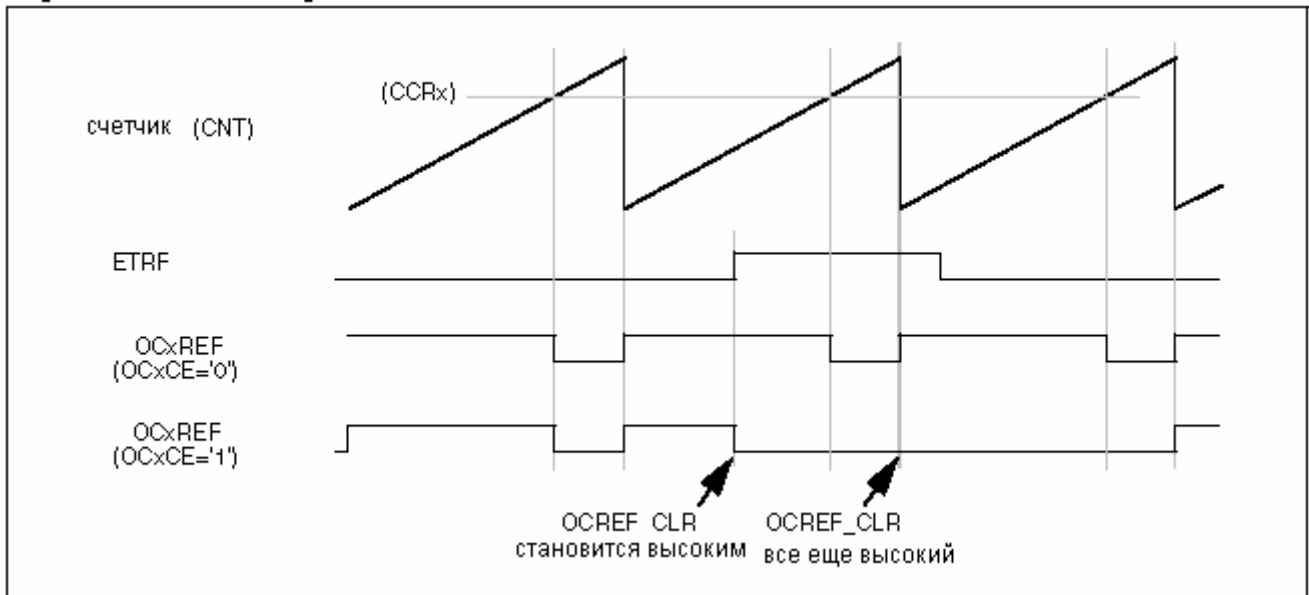
Например, сигнал OCxREF может быть подключен к выходу компаратора для работы с током. В этом случае ETR должен быть сконфигурирован следующим образом:

1. Пределитель внешнего запуска должен быть отключен: биты ETPS[1:0] регистра TIMx\_SMCR установлены в 00.
2. Режим 2 внешнего тактирования должен быть запрещен: бит ECE регистра TIMx\_SMCR установлен в 0.

3. Полярность внешнего запуска(ETP) и фильтр внешнего запуска(ETF) могут конфигурироваться под нужды пользователя.

Рис. 78 показывает поведение сигнала OCxREF когда вход ETRF становится высоким для обоих значений бита разрешения OCxCE. В данном примере таймер TIMx запрограммирован в режиме ШИМ.

Рис.78. Очистка TIMx\_OCxREF



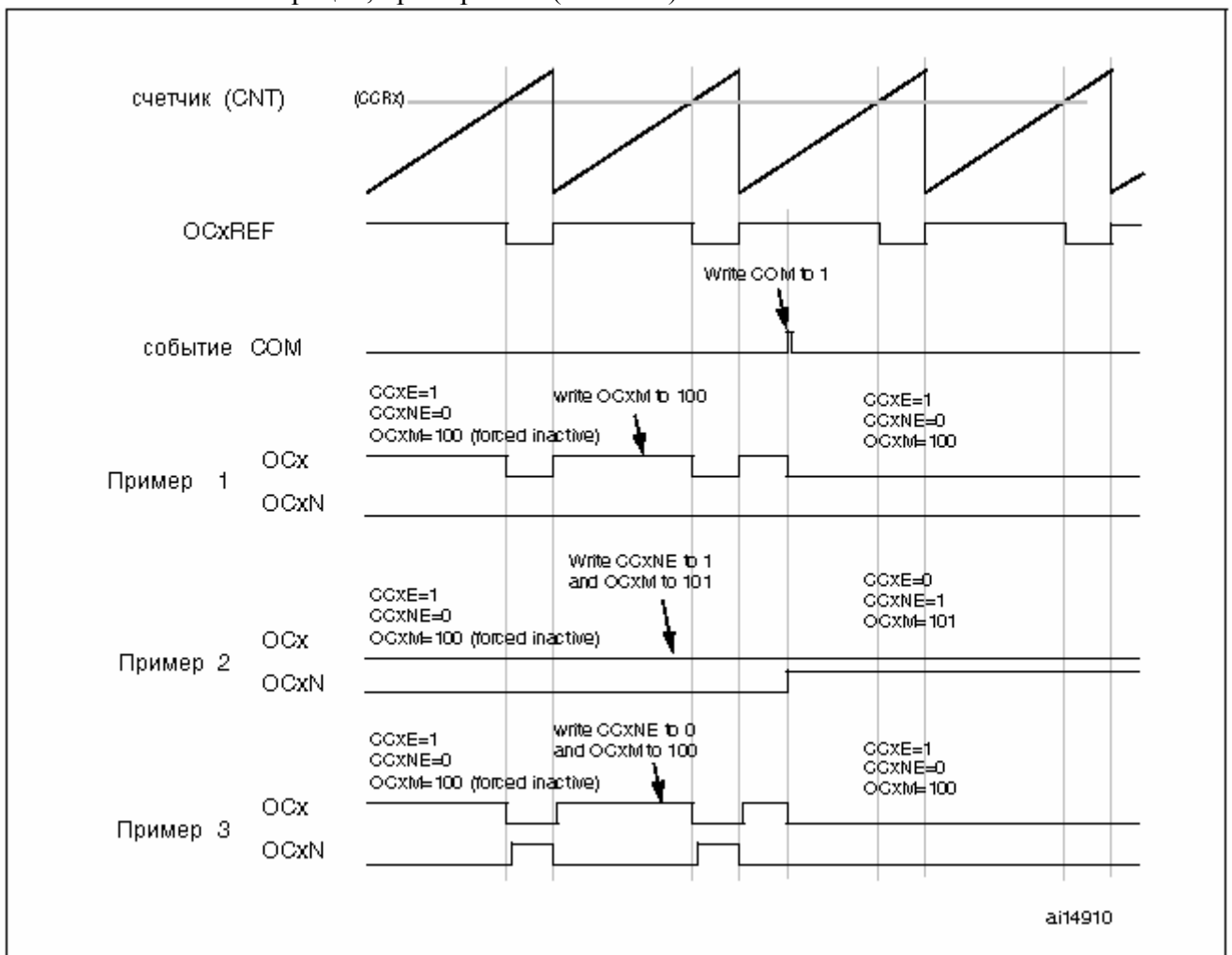
### 12.3.14 6-шаговая генерация ШИМ

Когда в канале используются комплементарные выходы, тогда биты предварительной загрузки доступны через биты OCxM, CCxE и CCxNE. Биты предварительной загрузки передаются в теньевые биты по коммутационному событию COM. Поэтому для следующего шага вы можете заранее запрограммировать конфигурацию и одновременно изменить конфигурацию всех каналов. COM может генерироваться программно через установку бита COM в регистре TIMx\_EGR либо аппаратно по нарастающему фронту TRGI.

Когда происходит событие COM, то устанавливается флаг(бит COMIF в TIMx\_SR), который может сгенерировать прерывание(если установлен бит COMIE в TIMx\_DIER) или запрос DMA(если установлен бит COMDE в TIMx\_DIER).

Рис.79 описывает поведение выходов OCx и OCxN в трех различных конфигурациях когда происходит событие COM

Рис.79. 6-шаговая генерация, пример COM(OSSR=1)



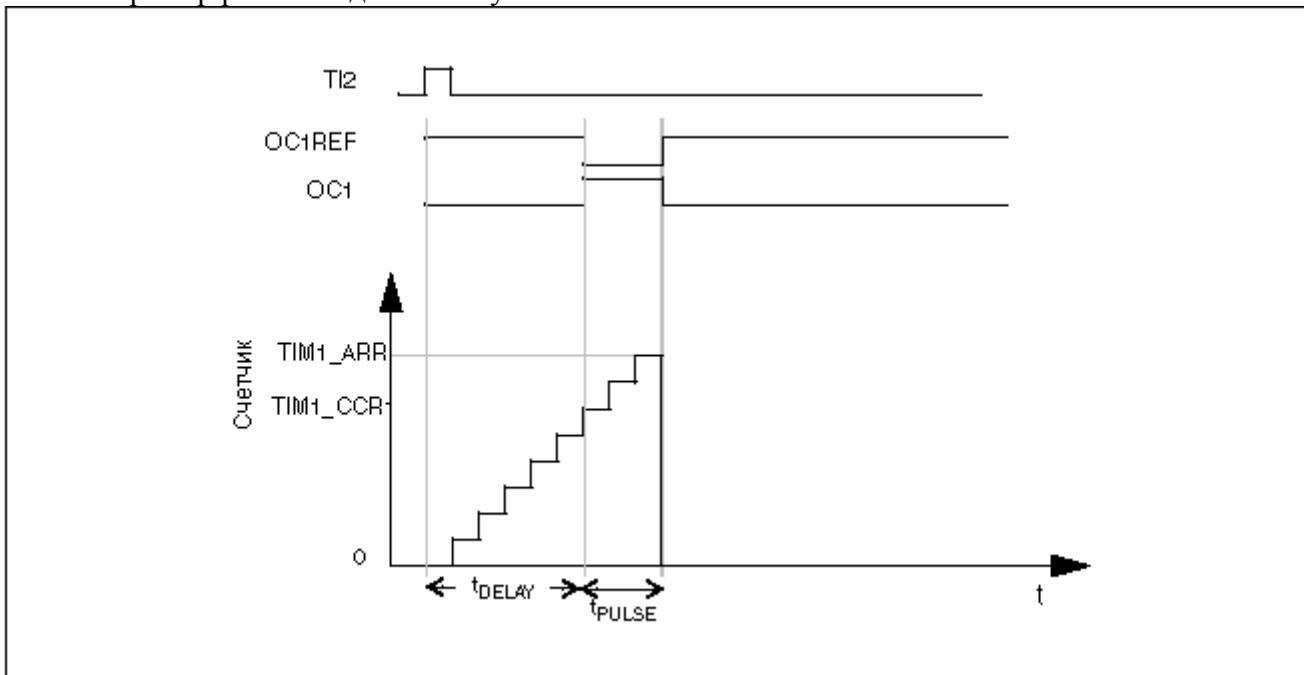
### 12.3.15 Режим одного импульса

Режим одного импульса(OPM) – это частный случай предыдущих режимов. Он позволяет счетчику запускаться в ответ на воздействие и генерировать импульс программируемой длины после программируемой задержки. Запуск счетчика может управляться через контроллер подчиненного режима. Генерация сигнала может происходить в режиме сравнения выхода или режиме ШИМ. Режим одного импульса выбирается через установку бита OPM регистра TIMx\_CR1. Это позволяет автоматически останавливать счетчик по следующему событию обновления UEV.

Импульс может генерироваться правильно только если значение сравнения отлично от начального значения счетчика. Перед запуском(когда таймер ожидает запуск) должна быть следующая конфигурация:

- При счете вверх:  $CNT < CCRx \leq ARR$  (в частности,  $0 < CCRx$ )
- При счете вниз:  $CNT > CCRx$

Рис.80. Пример режима одного импульса



Пример. Как только будет обнаружен положительный фронт на входе TI2, после задержки  $t_{DELAY}$ , генерируем положительный импульс на OC1 длиной  $t_{PULSE}$ .

Используем TI2FP2 как запуск1:

- Отобразить TI2FP2 на TI2 записью  $CC2S=01$  в регистр TIMx\_CCMR1
- TI2FP2 должен обнаруживать нарастающий фронт, запишите  $CC2P=0$  в TIMx\_CCER
- Сконфигурировать TI2FP2 как запуск для контроллера подчиненного режима (TRGI) записью  $TS=110$  в TIMx\_SMCR
- TI2FP2 используется для запуска счетчика, запишем в  $SMS=110$  в TIMx\_SMCR (режим запуска)

Сигнал OPM определяется через запись в регистры сравнения (принимая во внимание тактовую частоту и делитель счетчика).

- $t_{DELAY}$  определяется значением регистра TIMx\_CCR1
- $t_{PULSE}$  определяется разностью между значением автоперезагрузки и значением сравнения (TIMx\_ARR - TIMx\_CCR1)
- Предположим, что вы хотите создать сигнал с переходом 0 в 1 когда происходит совпадение при сравнении и переходом 1 в 0, когда счетчик достигает значения автоперезагрузки. Разрешите режим 2 ШИМ записью  $OC1M=111$  в TIMx\_CCMR1. Вы дополнительно можете разрешить регистры предварительной загрузки записью  $OC1PE=1$  в TIMx\_CCMR1 и  $ARPE=1$  в TIMx\_CR1. В данном случае вам нужно записать значение сравнения в регистр TIMx\_CCR1, значение автоперезагрузки в TIMx\_ARR, сгенерировать обновление установкой бита UG и подождать события внешнего запуска на TI2. В этом примере CCP записан в 0.

В нашем примере биты DIR и SMS регистра TIMx\_CR1 должны быть низкими.

Т.к. нужен 1 импульс (одиночный режим), то запишите 1 в бит OPM регистра TIMx\_CR1 чтобы остановить счетчик по следующему событию обновления (когда счетчик переходит назад, со значения автоперезагрузки до 0). Когда бит OPM регистра TIMx\_CR1 установлен в 0, тогда выбирается повторяющийся режим.

#### Частный случай: быстрое разрешение OCx:

В режиме одного импульса при определении фронта на входе TI1 устанавливается бит CEN, который разрешает счетчик. Затем при сравнении значения счетчика и значения сравнения меняется выход. Но на эти операции потребуется несколько тактовых циклов что ограничивает минимальную задержку  $t_{DELAY}$ , которую возможно получить. Если вы хотите получить выходной сигнал с минимальной задержкой, то можно установить бит OCxFE регистра

TIMx\_CCMRx. OCxREF(и OCx) форсируются в ответ на воздействие без результата сравнения. Его новый уровень такой же, как если бы произошло совпадение при сравнении. OCxFE действует только тогда, когда канал сконфигурирован в режиме ШИМ1 или ШИМ2.

### 12.3.16 Режим интерфейса энкодера

Чтобы выбрать режим интерфейса энкодера запишите SMS=001 в регистр TIMx\_SMCR если счетчик считает только по фронтам TI2, SMS=010 если считает только по фронтам TI1 и SMS=011 если считает по фронтам обоих входов TI1 и TI2. Выберите полярность TI1 и TI2 через биты CC1P и CC2P в TIMx\_CCER. Когда потребуется, вы также можете запрограммировать входной фильтр. Два входа TI1 и TI2 используются для связи с инкрементальным энкодером. Счетчик тактируется по каждому действующему переходу на TI1FP1 и TI2FP2 (TI1 и TI2 после входного фильтра и выбора полярности, TI1FP1= TI1 если не фильтруется и не инвертируется, TI2FP2= TI2 если не фильтруется и не инвертируется) полагая что он разрешен(бит SEN в TIMx\_CR1 записан в 1). Оценивается последовательность переходов на двух входах и генерируются счетные импульсы с сигналом направления. В зависимости от последовательности, счетчик считает вверх или вниз, соответственно аппаратно модифицируется бит DIR регистра TIMx\_CR1. Бит DIR рассчитывается по каждому переходу любого входа(TI1 или TI2), вне зависимости от того, считает ли счетчик только по TI1, только по TI2 или по обоим TI1 и TI2.

Режим интерфейса энкодера действует просто как внешнее тактирование с выбором направления. Это означает, что счетчик продолжительно считает между 0 и значением регистра автоперезагрузки TIMx\_ARR(от 0 до ARR или от ARR до 0 в зависимости от направления). Поэтому перед запуском вы должны сконфигурировать TIMx\_ARR. Захват, сравнение, предделитель, счетчик повторений, выход запуска продолжают работать как обычно. Режим 2 внешнего тактирования несовместим и не должен выбираться.

В этом режиме счетчик модифицируется следуя за скоростью, направлением и содержимым инкрементального энкодера. Следовательно, всегда воспроизводит позицию энкодера. Направление счета соответствует направлению вращения подключенного датчика. Таблица обобщает возможные комбинации, предполагая что TI1 и TI2 не переключаются в один и тот же момент.

Таблица 66. Направление счета в зависимости от сигналов энкодера

| Активный фронт     | Уровень на противоположном сигнале(TI1FP1 для TI2, TI2FP2 для TI1) | Сигнал TI1FP1 |           | Сигнал TI2FP2 |           |
|--------------------|--|---------------|-----------|---------------|-----------|
|                    |  | Нарастающий   | Спадающий | Нарастающий   | Спадающий |
| Счет только по TI1 | Высокий  | Вниз          | Вверх     | Нет счета     | Нет счета |
|                    | Низкий   | Вверх         | Вниз      | Нет счета     | Нет счета |
| Счет только по TI2 | Высокий  | Нет счета     | Нет счета | Вверх         | Вниз      |
|                    | Низкий   | Нет счета     | Нет счета | Вниз          | Вверх     |
| Счет по TI1 и TI2  | Высокий  | Вниз          | Вверх     | Вверх         | Вниз      |
|                    | Низкий   | Вверх         | Вниз      | Вниз          | Вверх     |

Внешний инкрементальный энкодер может подключаться напрямую к МК без внешней согласующей логики. Однако обычно используются компараторы для согласования дифференциальных выходов энкодера с цифровыми сигналами. Третий выход энкодера, показывающий нулевую механическую позицию, может быть подключен ко входу внешнего прерывания для сброса счетчика.

Рис.81 показывает пример работы счетчика. Он показывает управление направлением и генерацию счетного сигнала. Он также показывает как компенсируется дрожание входа(джиттер) когда выбираются оба фронта. Это может происходить если датчик позиционируется вблизи к одной из переключающих точек. Для данного примера мы предполагаем следующую конфигурацию:

- CC1S=01(регистр TIMx\_CCMR1, TI1FP1 отображается на TI1)
- CC2S=01(регистр TIMx\_CCMR2, TI1FP2 отображается на TI2)

- CC1P=0(регистр TIMx\_CCER, TI1FP1 не инвертирован, TI1FP1= TI1)
- CC2P=0(регистр TIMx\_CCER, TI1FP2 не инвертирован, TI1FP2= TI2)
- SMS=011(регистр TIMx\_SMCR, оба входа активны по нарастающим и спадающим фронтам)
- CEN=1(регистр TIMx\_CR1, счетчик разрешен)

Рис.81. Пример работы счетчика в режиме интерфейса энкодера

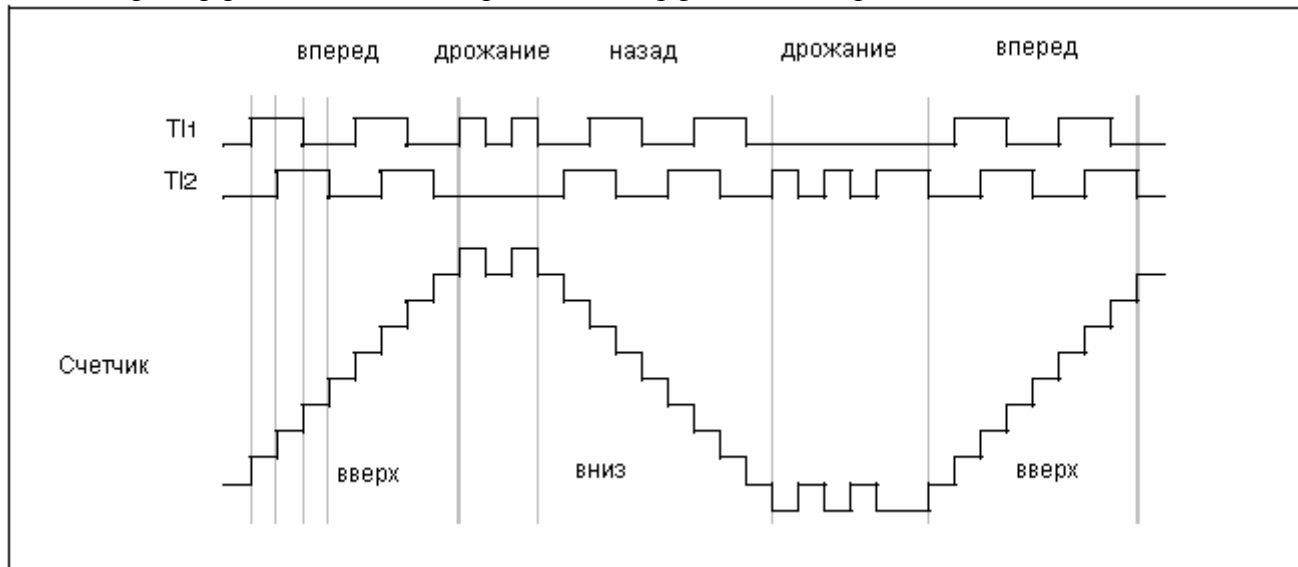
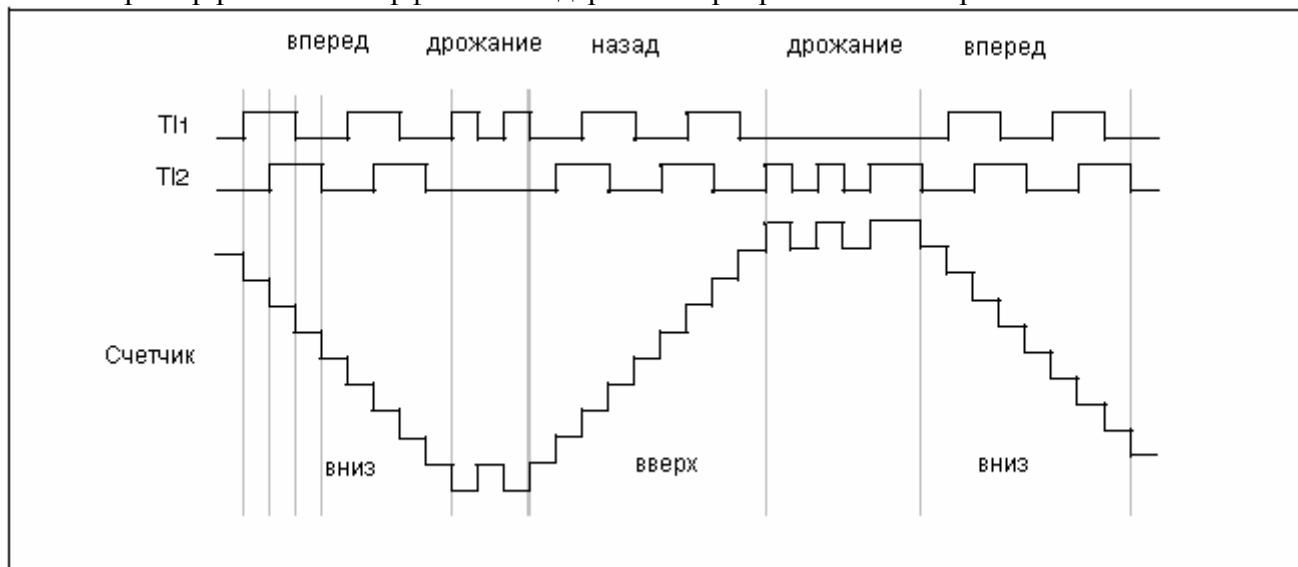


Рис.82 показывает пример поведения счетчика когда инвертирована полярность TI1FP1(такая же конфигурация как и выше за исключением того, что CC1P=1).

Рис.82.Пример режима интерфейса энкодера с инвертированной полярностью TI1FP1



При конфигурации в режиме интерфейса энкодера таймер предоставляет информацию о текущей позиции датчика. Вы можете получить динамическую информацию(скорость, ускорение, замедление) измеряя период между двумя событиями энкодера с использованием второго таймера в режиме захвата. Для данной цели может быть использован выход энкодера, показывающий механический ноль. В зависимости от времени между двумя событиями счетчик может быть прочитан через определенные моменты времени. Вы можете сделать это зашелкивая значение счетчика в третий регистр захвата входа если он доступен(тогда сигнал захвата должен быть периодичным и может генерироваться другим таймером). Также возможно читать данное значение через запрос DMA, генерируемый часами реального времени.

### 12.3.7 Функция Исключающего ИЛИ входа таймера

Бит TI1S регистра TIMx\_CR2 позволяет подключить вход фильтра канала 1 к выходу вентиля исключающего ИЛИ с объединением трех входов TIMx\_CN1, TIMx\_CN2 и TIMx\_CN3.

Выход исключающего ИЛИ может использоваться со всеми входными функциями таймера, такими как запуск или захват входа.

### 12.3.18 Сопряжение с датчиками Холла

Это делается с использованием таймеров улучшенного управления (TIM1) для генерации ШИМ сигналов управления мотором и другого таймера TIMx (TIM2, TIM3, TIM4 или TIM5), который рассматривается как «связующий таймер». «Связующий таймер» захватывает 3 входа (CC1, CC2, CC3), подключенные через исключающее ИЛИ к входному каналу TI1 (выбирается установкой бита TI1S регистра TIMx\_CR2).

Контроллер подчиненного режима конфигурируется в режиме сброса: TI1F\_ED – подчиненный вход. Поэтому каждый раз, когда изменяется один из 3 входов, счетчик перезапускает счет с 0. Это создает временную базу, запускаемую любым изменением на входах Холла.

В «связующем таймере» канал 1 захвата/сравнения конфигурируется в режиме захвата, сигнал захвата – TRC. Захваченное значение, соответствующее пройденному времени между 2 изменениями на входах, предоставляет информацию о скорости мотора.

«Связующий таймер» может использоваться в режиме выхода для генерации импульса, который изменяет конфигурацию каналов таймера улучшенного управления (TIM1, запуском события COM). Таймер TIM1 используется для генерации сигналов ШИМ управления мотором. Чтобы сделать это, канал связующего таймера должен быть запрограммирован так, чтобы генерировался положительный импульс после запрограммированной задержки (в режиме сравнения выхода или ШИМ). Этот импульс посылается таймеру улучшенного управления (TIM1) через выход TRGO.

Пример: вы хотите изменять конфигурацию ШИМ таймера улучшенного управления TIM1 после запрограммированной задержки каждый раз, когда происходит событие на входах Холла, подключенных к одному из таймеров TIMx.

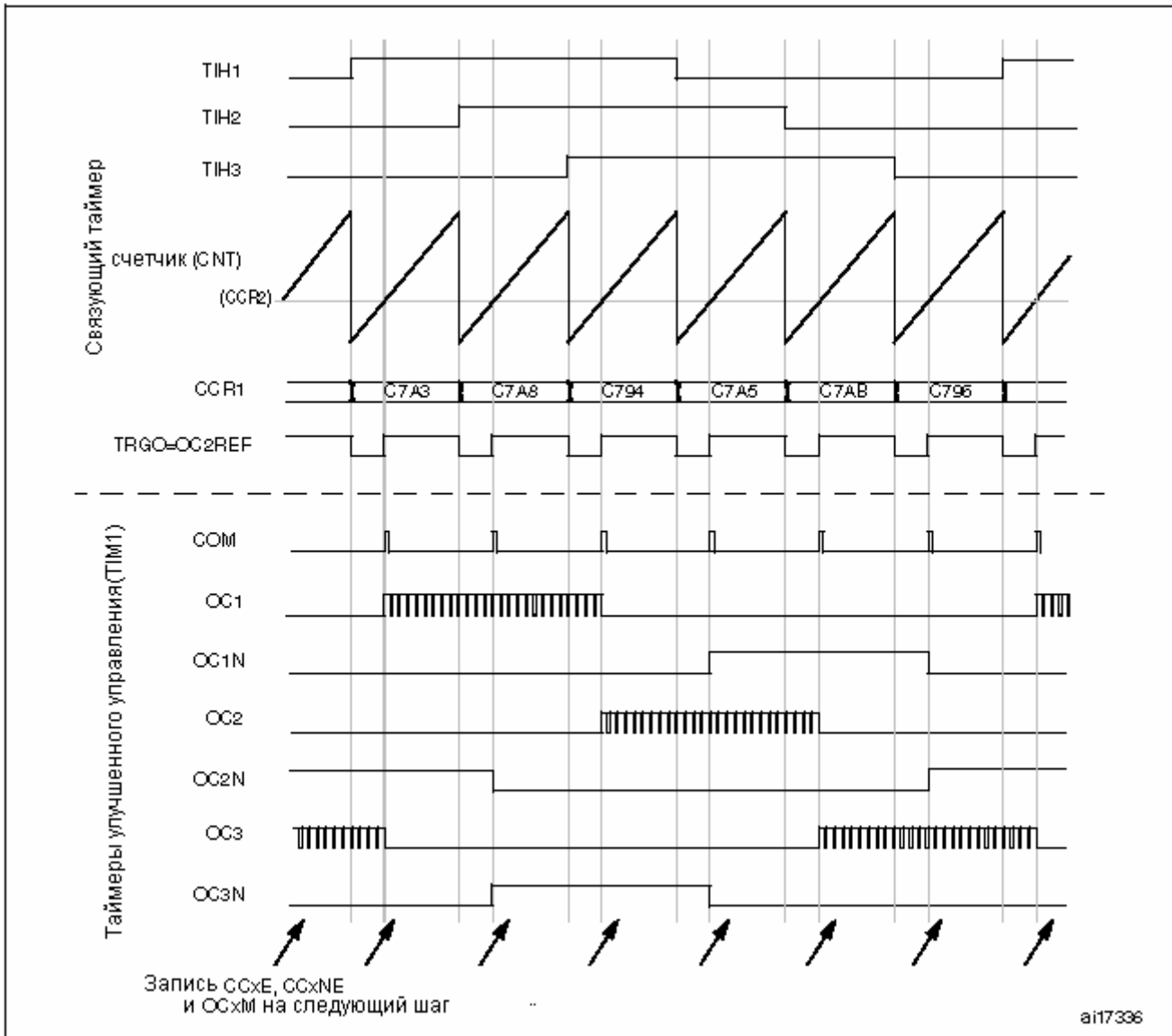
- Сконфигурировать 3 входа таймера через ИЛИ на входной канал TI1 через запись бита TI1S=1 регистра TIMx\_CR2
- Запрограммировать временную базу: записать в TIMx\_ARR максимальное значение (счетчик должен очищаться изменением TI1). Установите предделитель чтобы получить максимальный счетный период больше, чем время между двумя изменениями от датчика
- Запрограммировать канал 1 в режиме захвата (выбран TRC): записать биты CC1S=01 в регистре TIMx\_CCMR1. Если требуется, то вы можете запрограммировать цифровой фильтр
- Запрограммировать канал 2 в режиме ШИМ 2 с желаемой задержкой: записать биты OC2M=111 и биты CC2S=00 в регистре TIMx\_CCMR1
- Выбрать OC2REF как выход запуска на TRGO: записать биты MMS=101 в регистре TIMx\_CR2

У таймера улучшенного управления TIM1 правый вход ITR должен быть выбран как вход запуска, таймер программируется на генерацию сигналов ШИМ, управляющие сигналы захвата/сравнения предварительно загружаются (CCPC=1 в регистре TIMx\_CR2) и событие COM управляется входом запуска (CCUS=1 в регистре TIMx\_CR2). Управляющие биты ШИМ (CCxE, OCxM) записываются в следующем шаге после события COM (это можно сделать в подпрограмме обработки прерывания, генерируемым нарастающим фронтом OC2REF).

Рис.83 описывает данный пример.



Рис.83. Пример связи с датчиком Холла



### 12.3.19 TIMx и синхронизация внешнего запуска

Таймер TIMx может быть синхронизирован по внешнему запуску в нескольких режимах: режим сброса, режим стробирования и режим запуска.

#### Подчиненный режим: Режим сброса

Счетчик с предделителем может реинициализироваться в ответ на событие по входу запуска. Более того, генерируется событие обновления если бит URS в регистре TIMx\_CR1 низкий. Затем обновляются все регистры предварительной загрузки (TIMx\_ARR, TIMx\_CCRx).

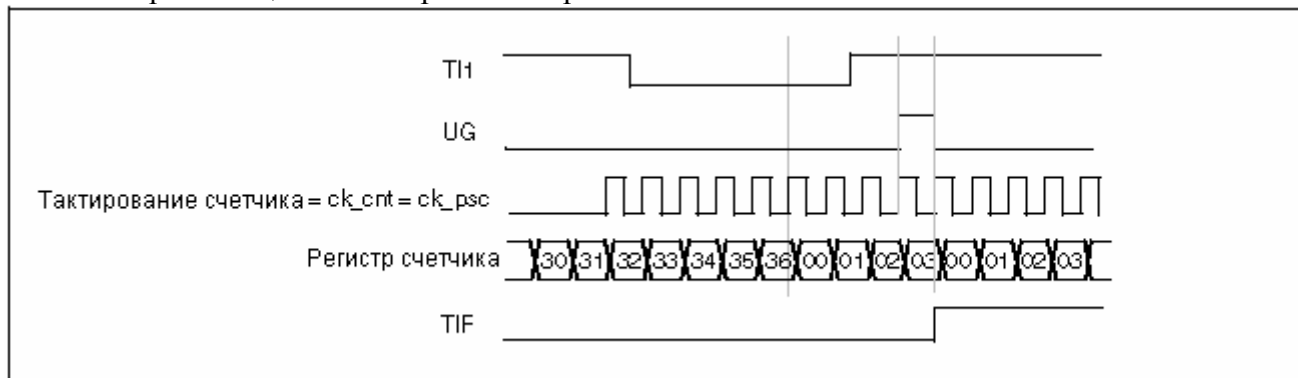
В следующем примере счет вверх очищается в ответ на нарастающий фронт на входе TI1:

- Сконфигурировать канал 1 на определение нарастающего фронта на TI1. Сконфигурировать длительность входного фильтра (в данном примере фильтр не нужен, поэтому IC1F=0000). Предделитель захвата для запуска не используется, поэтому его конфигурировать не нужно. Битами CC1S=01 регистра TIMx\_CCMR1 выбрать только источник захвата входа. Записать CC1P=0 в регистр TIMx\_CCER для подтверждения полярности (и определения только нарастающих фронтов).
- Сконфигурировать таймер в режиме сброса записью SMS=100 в регистр TIMx\_SMCR. Выбрать TI1 как входной источник записью TS=101 в TIMx\_SMCR.
- Запустить счетчик записью CEN=1 в регистр TIMx\_CR1.

Счетчик начинает считать от внутреннего тактирования и ведет себя как обычно до появления нарастающего фронта TI1. Когда нарастает TI1, тогда счетчик очищается и перезапускается с 0.

Тем временем устанавливается флаг запуска(бит TIF регистра TIMx\_SR) и, при разрешении, посылается прерывание или запрос DMA(зависит от битов TIE и TDE регистра TIMx\_DIER).  
 Следующий рисунок показывает поведение, когда регистр автоперезагрузки TIMx\_ARR=0x36. Задержка между нарастающим фронтом на TI1 и действительным сбросом счетчика вызвана схемой ресинхронизации на входе TI1.

Рис.84. Управляющая схема в режиме сброса



### Подчиненный режим: Режим стробирования

Счетчик может разрешаться в зависимости от уровня на выбранном входе.

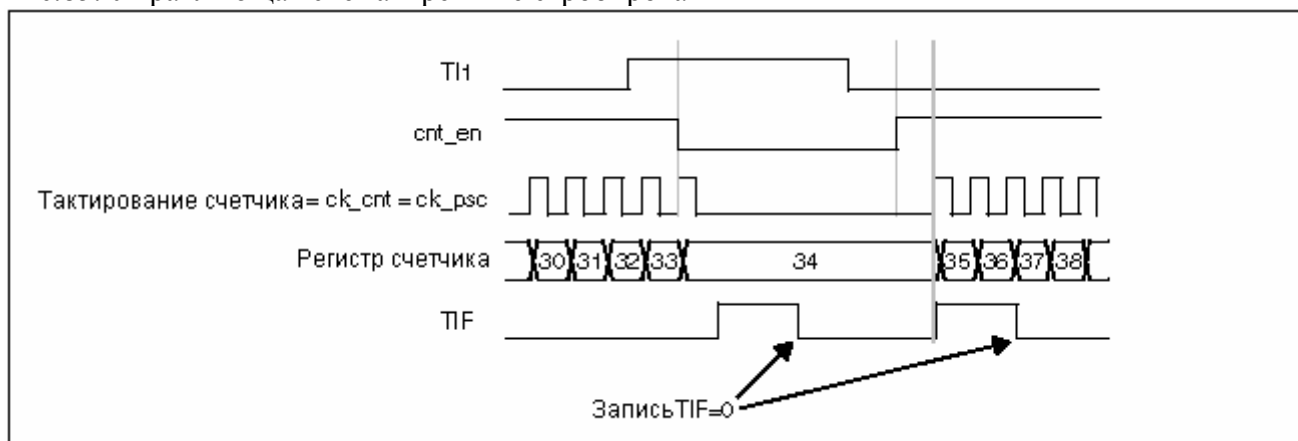
В следующем примере счетчик считает вверх только когда вход TI1 низкий:

- Сконфигурировать канал 1 на определение низких уровней на TI1. Сконфигурировать длительность входного фильтра(в этом примере мы не используем фильтр, поэтому IC1F=0000). Предделитель захвата для запуска не используется, поэтому вам не нужно его конфигурировать. Биты CC1S выбирают только источник входа захвата, CC1S=01 в регистре TIMx\_CCMR1. Для подтверждения полярности(и определения только низких уровней) запишите в регистр TIMx\_CCER бит CC1P=1.
- Сконфигурировать таймер в режиме стробирования записью SMS=101 в регистр TIMx\_SMCR. Выбрать TI1 как входной источник записью TS=101 в регистр TIMx\_SMCR.
- Разрешить счетчик записью CEN=1 в TIMx\_CR1(в режиме стробирования счетчик не запускается если CEN=0, какой бы ни был входной уровень запуска).

Счетчик начинает считать от внутреннего тактирования пока TI1 низкий и останавливается когда TI1 становится высоким. Когда счетчик запускается или останавливается, то выставляется флаг TIF в регистре TIMx\_SR.

Задержка между нарастающим фронтом на TI1 и действительной остановкой счетчика вызвана схемой ресинхронизации на входе TI1.

Рис.85. Управляющая схема в режиме стробирования



### Подчиненный режим: Режим запуска

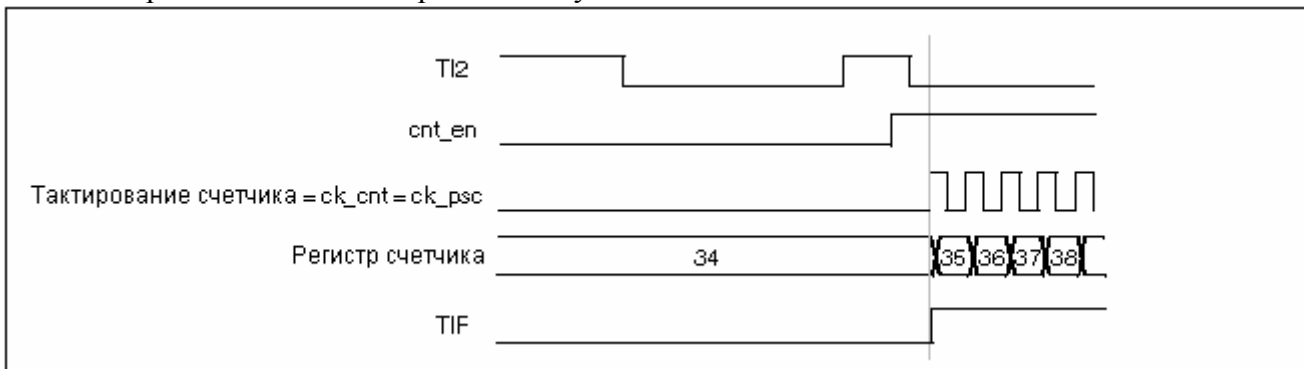
Счетчик может запускаться в ответ на событие на выбранном входе.

В следующем примере начинается счет вверх в ответ на нарастающий фронт на входе TI2:

- Сконфигурировать канал 2 на определение нарастающих фронтов на TI2. Сконфигурировать длительность входного фильтра(в этом примере фильтр не нужен, поэтому IC2F=0000). Предделитель захвата для запуска не используется, поэтому не нужно его конфигурировать. Биты CC2S=01 в TIMx\_CCMR1 сконфигурированы только на выбор источника захвата входа. Записать CC2P=1 в TIMx\_CCER чтобы выставить полярность (и определение только низкого уровня).
- Сконфигурировать таймер в режиме запуска через запись SMS=110 в TIMx\_SMCR. Выбрать TI2 как входной источник записью TS=110 в TIMx\_SMCR.

Когда появляется нарастающий фронт на TI2, то счетчик начинает счет и выставляет флаг TIF. Задержка между нарастающим фронтом на TI2 и действительным запуском счетчика вызвана схемой ресинхронизации на входе TI2.

Рис.86. Управляющая схема в режиме запуска



### Подчиненный режим: Режим 2 внешнего тактирования + Режим запуска

Режим 2 внешнего тактирования может использоваться в дополнение к другому подчиненному режиму(за исключением режима 1 внешнего тактирования и режима энкодера). В этом случае сигнал ETR используется как вход внешнего тактирования, а другой вход может выбираться как вход запуска(в режиме сброса, режиме стробирования или режиме запуска). Не рекомендуется выбирать ETR как TRGI через биты TS регистра TIMx\_SMCR.

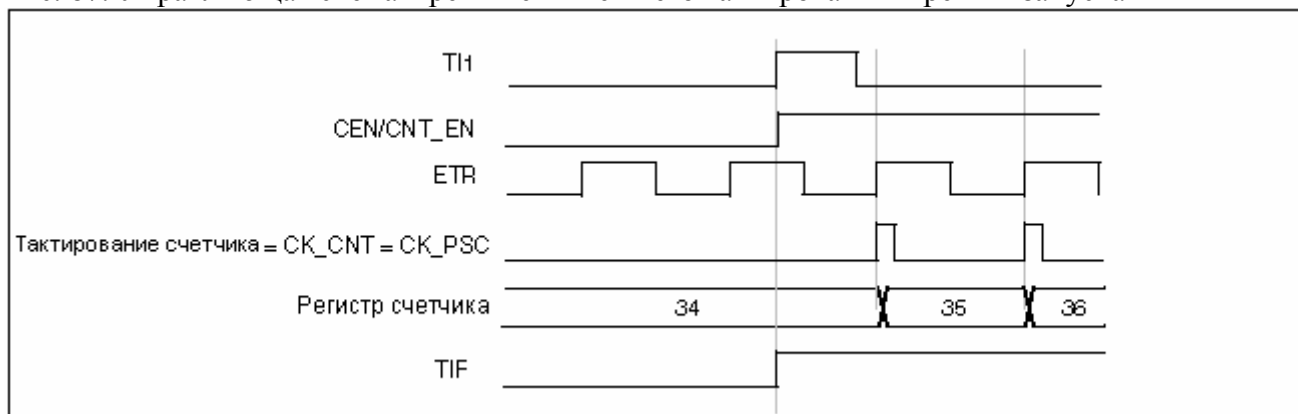
В следующем примере счет вверх инкрементируется по каждому нарастающему фронту сигнала ETR как только происходит нарастающий фронт TI1:

1. Сконфигурировать схему входа внешнего запуска программированием регистра TIMx\_SMCR:
  - ETF=0000: без фильтра
  - ETPS=00: предделитель запрещен
  - ETP=0: определение нарастающих фронтов на ETR и ECE=1 для разрешения режима 2 внешнего тактирования.
2. Сконфигурировать канал 1 для определения нарастающих фронтов на TI1:
  - IC1F=0000: без фильтра
  - Предделитель захвата для запуска не используется и не конфигурируется
  - CC1S=01 в TIMx\_CCMR1 для выбора только источника захвата входа
  - CC1P=0 в TIMx\_CCER для выбора полярности(и обнаружения только нарастающего фронта)
3. Сконфигурировать таймер в режиме запуска записью SMS=110 в регистр TIMx\_SMCR. Выбрать TI1 как источник входа записью TS=101 в регистре TIMx\_SMCR.

Нарастающий фронт на TI1 разрешает счетчик и устанавливает флаг TIF. Затем счетчик считает по нарастающим фронтам ETR.

Задержка между нарастающим фронтом сигнала ETR и действительным сбросом счетчика вызвана схемой ресинхронизации на входе ETRP.

Рис. 87. Управляющая схема в режиме 2 внешнего тактирования + режим запуска



### 12.3.20 Синхронизация таймера

Таймеры TIM связываются вместе для синхронизации или объединения в цепочки.

### 12.3.21 Режим отладки

Когда микроконтроллер входит в режим отладки (остановлено ядро Cortex-M3) счетчик таймера продолжает функционировать или останавливается в зависимости от конфигурационного бита DBG\_TIMx\_STOP в модуле DBG.

## 12.4 Регистры TIM1

### 12.4.1 Управляющий регистр 1 TIM1 (TIMx\_CR1)

Смещение: 0x00

Значение сброса: 0x0000

|          |    |    |    |    |    |          |    |      |          |    |     |     |     |      |     |
|----------|----|----|----|----|----|----------|----|------|----------|----|-----|-----|-----|------|-----|
| 15       | 14 | 13 | 12 | 11 | 10 | 9        | 8  | 7    | 6        | 5  | 4   | 3   | 2   | 1    | 0   |
| Reserved |    |    |    |    |    | CKD[1:0] |    | ARPE | CMS[1:0] |    | DIR | OPM | URS | UDIS | CEN |
|          |    |    |    |    |    | rw       | rw | rw   | rw       | rw | rw  | rw  | rw  | rw   | rw  |

| Бит   | Описание   |
|-------|--|
| 15:10 | Резерв, всегда читается как 0  |
| 9:8   | <b>CKD[1:0]</b> : деление тактирования<br>Это битовое поле показывает соотношение между тактированием таймера (CK_INT), мертвым временем и частотой выборок (tDTS), используемое генераторами мертвого времени и цифровыми фильтрами (ETR, TIX),<br>00: tDTS=tCK_INT<br>01: tDTS=2*tCK_INT<br>10: tDTS=4*tCK_INT<br>11: Резерв, не программируйте это значение   |
| 7     | <b>ARPE</b> : разрешение предварительной загрузки для автоперезагрузки<br>0: регистр TIMx_ARR не буферизован<br>1: регистр TIMx_ARR буферизован  |
| 6:5   | <b>CMS[1:0]</b> : выбор режима выравнивания по центру<br>00: Режим выравнивания по фронту. Счетчик считает вверх или вниз в зависимости от бита направления (DIR).<br>01: Режим 1 выравнивания по центру. Счетчик считает вверх и дополнительно вниз. Флаги прерывания сравнения выхода от каналов, сконфигурированных на выход (CCxS=00 в TIMx_CCMRx), устанавливаются только тогда, когда счетчик считает вниз<br>10: Режим 2 выравнивания по центру. Счетчик считает вверх и дополнительно вниз. Флаги прерывания сравнения выхода от каналов, сконфигурированных на выход (CCxS=00 в TIMx_CCMRx), устанавливаются только тогда, когда счетчик считает вверх. |

|   |  |
|---|--|
|   | <p>11: Режим 3 выравнивания по центру. Счетчик считает вверх и дополнительно вниз. Флаги прерывания сравнения выхода от каналов, сконфигурированных на выход (CCxS=00 в TIMx_CCMRx) устанавливаются когда счетчик считает вверх и вниз. Прим: Когда счетчик разрешен(CEN=1), то не допускается переключаться из режима выравнивания по фронту в режим выравнивания центру.</p>   |
| 4 | <p><b>DIR:</b> Направление<br/> 0: счетчик считает вверх<br/> 1: счетчик считает вниз<br/> Прим: Этот бит только для чтения когда счетчик сконфигурирован в режиме выравнивания по центру или режиме энкодера.</p>   |
| 3 | <p><b>OPM:</b> Режим одного импульса<br/> 0: счетчик не останавливается по событию обновления<br/> 1: счетчик останавливает счет по следующему событию обновления(очистка бита CEN)</p>  |
| 2 | <p><b>URS:</b> Источник запроса обновления<br/> Этот бит устанавливается и очищается программно для выбора источников события UEV.<br/> 0: Любое из следующих событий генерирует обновление прерывания или запрос DMA при разрешении. Этими событиями могут быть:<br/> - переполнение счетчика(вверх/вниз)<br/> - установка бита UG<br/> - генерация обновления через контроллер подчиненного режима<br/> 1: Только переполнение счетчика(вверх-вниз) генерирует прерывание обновления или запрос DMA при разрешении.</p>  |
| 1 | <p><b>UDIS:</b> Запрещение обновления<br/> Этот бит устанавливается и очищается программно для разрешения/запрещения генерации события UEV.<br/> 0: UEV разрешен. Событие обновления(UEV) генерируется одним из следующих событий:<br/> - переполнение счетчика(вверх/вниз)<br/> - установка бита UG<br/> - генерация обновления через контроллер подчиненного режима<br/> Затем буферизованные регистры загружаются своими предварительно загруженными значениями.<br/> 1: UEV запрещен. Событие обновления не генерируется, теньевые регистры удерживают свои значения(ARR, PSC, CCRx). Однако счетчик и предделитель реинициализируются если устанавливается бит UG или если получен аппаратный сброс от контроллера подчиненного режима.</p> |
| 0 | <p><b>CEN:</b> Разрешение счетчика<br/> 0: счетчик запрещен<br/> 1: счетчик разрешен<br/> Прим: Режимы внешнего тактирования, стробирования и энкодера могут работать только если до этого был программно установлен бит CEN. Однако режим запуска может автоматически аппаратно устанавливать бит CEN.</p>  |

## 12.4.2 Управляющий регистр 2 TIM1(TIMx\_CR2)

Смещение адреса: 0x04

Значение сброса: 0x0000

|      |      |       |      |       |      |       |      |      |          |    |    |      |      |      |      |   |
|------|------|-------|------|-------|------|-------|------|------|----------|----|----|------|------|------|------|---|
|      | 15   | 14    | 13   | 12    | 11   | 10    | 9    | 8    | 7        | 6  | 5  | 4    | 3    | 2    | 1    | 0 |
| Res. | OIS4 | OIS3N | OIS3 | OIS2N | OIS2 | OIS1N | OIS1 | TI1S | MMS[2:0] |    |    | CCDS | CCUS | Res. | CCPC |   |
|      | rw   | rw    | rw   | rw    | rw   | rw    | rw   | rw   | rw       | rw | rw | rw   | rw   |      | rw   |   |

| Бит | Описание   |
|-----|--|
| 15  | Резерв, всегда читается как 0  |
| 14  | <b>OIS4</b> : Состояние бездействия выхода 4(выход OC4)  |
| 13  | <b>OIS3N</b> : Состояние бездействия выхода 3(выход OC3N)  |
| 12  | <b>OIS3</b> : Состояние бездействия выхода 3(выход OC3)  |
| 11  | <b>OIS2N</b> : Состояние бездействия выхода 2(выход OC2N)  |
| 10  | <b>OIS2</b> : Состояние бездействия выхода 2(выход OC2)  |
| 9   | <b>OIS1N</b> : Состояние бездействия выхода 1(выход OC1N)<br>0: когда MOE=0 после мертвого времени OC1N=0<br>1: когда MOE=0 после мертвого времени OC1N=1<br>Прим: этот бит не может быть модифицирован когда запрограммирован уровень LOCK 1, 2 или 3.  |
| 8   | <b>OIS1</b> : Состояние бездействия выхода 1(выход OC1)<br>0: когда MOE=0, OC1=0(после мертвого времени если используется OC1N)<br>1: когда MOE=0, OC1=1(после мертвого времени если используется OC1N)<br>Прим: этот бит не может быть модифицирован когда запрограммирован уровень LOCK 1, 2 или 3.  |
| 7   | <b>TI1S</b> : Выбор TI1<br>0: вывод TIMx_CH1 подключен ко входу TI1<br>1: выходы TIMx_CH1, CH2 и CH3 подключены ко входу TI1(объединение через исключающее ИЛИ)  |
| 6:4 | <b>MMS</b> : Выбор режима главного<br>Эти биты в главном режиме позволяют выбирать посылаемую информацию подчиненным таймерам для синхронизации(TRGO).<br>000: <b>Сброс</b> – бит UG из регистра TIMx_EGR используется как выход запуска(TRGO). Если сброс генерируется входом запуска(контроллер подчиненного режима сконфигурирован в режиме сброса), тогда сигнал на TRGO задерживается по сравнению с действительным сбросом.<br>001: <b>Разрешение</b> – сигнал разрешения счетчика используется как выход запуска(TRGO). Он полезен для запуска нескольких таймеров в одно время или для управления окном, в котором разрешен подчиненный таймер. Сигнал разрешения счетчика генерируется через логическое ИЛИ между управляющим битом SEN и входом запуска при конфигурации в режиме стробирования. На TRGO есть задержка, когда сигнал разрешения счетчика управляется входом запуска за исключением выбора режима главного/подчиненного.<br>010: <b>Обновление</b> – событие обновления выбирается как выход запуска(TRGO). Например главный таймер может использоваться как предделитель подчиненного таймера.<br>011: <b>Импульс сравнения</b> – выход запуска посылает положительный импульс когда устанавливается флаг CC1IF(даже если он уже высокий) как только происходит совпадение при сравнении или захват.<br>100: <b>Сравнение</b> – сигнал OC1REF используется как выход запуска(TRGO)<br>101: <b>Сравнение</b> – сигнал OC2REF используется как выход запуска(TRGO)<br>110: <b>Сравнение</b> – сигнал OC3REF используется как выход запуска(TRGO)<br>111: <b>Сравнение</b> – сигнал OC4REF используется как выход запуска(TRGO) |
| 3   | <b>CCDS</b> : выбор DMA захвата/сравнения<br>0: посылается запрос DMA CCx когда происходит событие CCx<br>1: посылается запрос DMA CCx когда происходит событие обновления   |

|   |   |
|---|---|
| 2 | <p><b>CCUS:</b> Выбор управления обновления захвата/сравнения</p> <p>0: Когда биты управления захвата/сравнения предварительно загружены(CCPC=1), они обновляются только установкой бита COMG</p> <p>1: Когда биты управления захвата/сравнения предварительно загружены(CCPC=1), они обновляются установкой бита COMG либо по нарастающему фронту на TRGI</p> <p>Прим: этот бит действует только на каналах с комплементарным выходом.</p> |
| 1 | Резерв, всегда читается как 0   |
| 0 | <p><b>CCPC:</b> Предварительно загруженное управление захвата/сравнения</p> <p>0: биты CCxE, CCxNE и OCxM предварительно не загружаются</p> <p>1: биты CCxE, CCxNE и OCxM предварительно загружаются после записи</p> <p>Прим: этот бит работает только на каналах с комплементарным выходом.</p>   |

### 12.4.3 Регистр управления подчиненным режимом TIM1 (TIMx\_SMCR)

Смещение адреса: 0x08

Значение сброса: 0x0000

|     |     |           |    |          |    |    |    |     |         |    |    |    |      |          |    |    |
|-----|-----|-----------|----|----------|----|----|----|-----|---------|----|----|----|------|----------|----|----|
| 15  | 14  | 13        | 12 | 11       | 10 | 9  | 8  | 7   | 6       | 5  | 4  | 3  | 2    | 1        | 0  |    |
| ETP | ECE | ETPS[1:0] |    | ETF[3:0] |    |    |    | MSM | TS[2:0] |    |    |    | Res. | SMS[2:0] |    |    |
| rw  | rw  | rw        | rw | rw       | rw | rw | rw | rw  | rw      | rw | rw | rw | Res. | rw       | rw | rw |

| Бит   | Описание   |
|-------|--|
| 15    | <p><b>ETP:</b> Полярность внешнего запуска</p> <p>Этот бит выбирает, используется ли ETP или /ETP для операций запуска</p> <p>0: ETP неинвертирован, активный высокий уровень или нарастающий фронт</p> <p>1: ETP инвертирован, активный низкий уровень или спадающий фронт</p>  |
| 14    | <p><b>ECE:</b> Разрешение внешнего тактирования</p> <p>Этот бит разрешает режим 2 внешнего тактирования.</p> <p>0: Режим 2 внешнего тактирования запрещен</p> <p>1: Режим 2 внешнего тактирования разрешен. Счетчик тактируется любым активным фронтом сигнала ETRF.</p> <p>Прим: 1. Установка бита ECE имеет такой же эффект, как выбор режима 1 внешнего тактирования с TRGI подключенным к ETRF(SMS=111 и TS=111)</p> <p>2. Возможно одновременно использовать режим 2 внешнего тактирования со следующими подчиненными режимами: режим сброса, режим стробирования и режим запуска. Однако в этом случае TRGI не должен подключаться к ETRF(биты TS не должны быть 111).</p> <p>3. Если в одно время разрешается режим 1 внешнего тактирования и режим 2 внешнего тактирования, то внешний вход тактирования – ETRF.</p> |
| 13:12 | <p><b>ETPS[1:0] :</b> Предделитель внешнего запуска</p> <p>Частота сигнала внешнего запуска ETRP должна составлять по крайней мере 1/4 от частоты TIMxCLK. Предделитель может быть разрешен для уменьшения частоты ETRP. Он полезен когда вход имеет слишком быстрое тактирование.</p> <p>00: предделитель отключен</p> <p>01: частота ETRP делится на 2</p> <p>10: частота ETRP делится на 4</p> <p>11: частота ETRP делится на 8</p>   |
| 11:8  | <p><b>ETF[3:0] :</b> Фильтр внешнего запуска</p> <p>Это битовое поле определяет частоту, используемую для выборки сигнала ETRP и длину цифрового фильтра, применяемого к ETRP. Цифровой фильтр состоит из счетчика событий для которого требуется N событий, чтобы подтвердить передачу на выход.</p> <p>0000: Без фильтра, выборка делается на fDTS</p> <p>0001: fSAMPLING=fCK_INT, N=2</p> <p>0010: fSAMPLING=fCK_INT, N=4</p> <p>0011: fSAMPLING=fCK_INT, N=8</p> <p>0100: fSAMPLING=fDTS/2, N=6</p>  |

|     |   |
|-----|---|
|     | <p>0101: fSAMPLING=fDTS/2, N=8<br/> 0110: fSAMPLING=fDTS/4, N=6<br/> 0111: fSAMPLING=fDTS/4, N=8<br/> 1000: fSAMPLING=fDTS/8, N=6<br/> 1001: fSAMPLING=fDTS/8, N=8<br/> 1010: fSAMPLING=fDTS/16, N=5<br/> 1011: fSAMPLING=fDTS/16, N=6<br/> 1100: fSAMPLING=fDTS/16, N=8<br/> 1101: fSAMPLING=fDTS/32, N=5<br/> 1110: fSAMPLING=fDTS/32, N=6<br/> 1111: fSAMPLING=fDTS/32, N=8</p>  |
| 7   | <p><b>MSM</b>: Режим главного/подчиненного<br/> 0: Бездействует<br/> 1: Действие события по входу запуска(TRGI) задерживается чтобы совершить синхронизацию между текущим таймером и его подчиненными(через TRGO). Это полезно если мы захотим синхронизировать несколько таймеров по одному внешнему событию.</p>  |
| 6:4 | <p><b>TS[2:0]</b> : Выбор запуска<br/> Это битовое поле выбирает вход запуска, используемый для синхронизации счетчика.<br/> 000: Внутренний запуск 0 (ITR0)<br/> 001: Внутренний запуск 1 (ITR1)<br/> 010: Внутренний запуск 2 (ITR2)<br/> 011: Внутренний запуск 3 (ITR3)<br/> 100: Детектор фронта TI1(TI1F_ED)<br/> 101: Отфильтрованный вход 1 таймера(TI1FP1)<br/> 110: Отфильтрованный вход 2 таймера(TI2FP2)<br/> 111: Вход внешнего запуска(ETRF)<br/> Прим: Эти биты должны изменяться только когда не используются(когда SMS=000) чтобы избежать неверного определения фронтов в момент перехода.</p>  |
| 3   | <p>Резерв, всегда читается как 0</p>  |
| 2:0 | <p><b>SMS</b> : Выбор режима подчиненного<br/> Когда выбираются внешние сигналы, то активный фронт сигнала запуска(TRGI) связывается с выбранной полярностью внешнего входа.<br/> 000: Подчиненный режим запрещен – если CEN=1, тогда предделитель тактируется напрямую от внутреннего тактирования.<br/> 001: Режим 1 энкодера – счетчик считает вверх/вниз по фронту TI2FP2 в зависимости от уровня TI1FP1.<br/> 010: Режим 2 энкодера – счетчик считает вверх/вниз по фронту TI1FP1 в зависимости от уровня TI2FP2.<br/> 011: Режим 3 энкодера – счетчик считает вверх/вниз по обоим фронтам TI1FP1 и TI2FP2 в зависимости от уровня другого входа.<br/> 100: Режим сброса – нарастающий фронт выбранного входа запуска(TRGI) реинициализирует счетчик и генерирует обновление регистров.<br/> 101: Режим стробирования – тактирование счетчика разрешено когда высокий вход запуска(TRGI). Счетчик останавливается(но не сбрасывается) как только вход запуска становится низким. Управляется запуск и остановка счетчика.<br/> 110: Режим запуска – счетчик запускается по нарастающему фронту запуска TRGI(но не сбрасывается). Управляется только запуск счетчика.<br/> 111: Режим 1 внешнего тактирования – нарастающие фронты выбранного входа запуска(TRGI) тактируют счетчик.<br/> Прим: Режим стробирования не должен выбираться если TI1F_ED выбирается как вход запуска(TS=100). Действительно, TI1F_ED выводит 1 импульс на каждый переход TI1F, несмотря на то, что режим стробирования контролирует уровень сигнала запуска.</p> |



|                  |                   |              |              |              |
|------------------|-------------------|--------------|--------------|--------------|
| Подчиненный TIM1 | ITR0(TS=000)      | ITR1(TS=001) | ITR2(TS=010) | ITR3(TS=011) |
| TIM1             | TIM5 или TIM15(1) | TIM2         | TIM3         | TIM4         |

1. TIM5 только в устройствах высокой плотности. Выбор TIM5 или TIM15 зависит от бита MISC\_REMAP регистра AFIO\_MAPR2.

#### 12.4.4 Регистр разрешения прерываний/DMA TIM1 (TIMx\_DIER)

Смещение адреса: 0x0C

Значение сброса: 0x0000

|      |     |       |       |       |       |       |     |     |     |       |       |       |       |       |     |    |
|------|-----|-------|-------|-------|-------|-------|-----|-----|-----|-------|-------|-------|-------|-------|-----|----|
|      | 15  | 14    | 13    | 12    | 11    | 10    | 9   | 8   | 7   | 6     | 5     | 4     | 3     | 2     | 1   | 0  |
| Res. | TDE | COMDE | CC4DE | CC3DE | CC2DE | CC1DE | UDE | BIE | TIE | COMIE | CC4IE | CC3IE | CC2IE | CC1IE | UIE |    |
|      | rw  | rw    | rw    | rw    | rw    | rw    | rw  | rw  | rw  | rw    | rw    | rw    | rw    | rw    | rw  | rw |

| Бит | Описание   |
|-----|--|
| 15  | Резерв, всегда читается как 0  |
| 14  | <b>TDE</b> : Разрешение запроса запуска DMA<br>0: Запрос запуска DMA запрещен<br>1: Запрос запуска DMA разрешен                |
| 13  | <b>COMDE</b> : Разрешение запроса COM DMA<br>0: Запрос COM DMA запрещен<br>1: Запрос COM DMA разрешен                          |
| 12  | <b>CC4DE</b> : Разрешение запроса DMA захвата/сравнения 4<br>0: Запрос DMA CC4DE запрещен<br>1: Запрос DMA CC4DE разрешен      |
| 11  | <b>CC3DE</b> : Разрешение запроса DMA захвата/сравнения 3<br>0: Запрос DMA CC3DE запрещен<br>1: Запрос DMA CC3DE разрешен      |
| 10  | <b>CC2DE</b> : Разрешение запроса DMA захвата/сравнения 2<br>0: Запрос DMA CC2DE запрещен<br>1: Запрос DMA CC2DE разрешен      |
| 9   | <b>CC1DE</b> : Разрешение запроса DMA захвата/сравнения 1<br>0: Запрос DMA CC1DE запрещен<br>1: Запрос DMA CC1DE разрешен      |
| 8   | <b>UDE</b> : Разрешение запроса обновления DMA<br>0: Запрос обновления DMA запрещен<br>1: Запрос обновления DMA разрешен       |
| 7   | <b>BIE</b> : Разрешение прерывания аварийного останова<br>0: Прерывание останова запрещено<br>1: Прерывание останова разрешено |
| 6   | <b>TIE</b> : Разрешение прерывания запуска<br>0: Прерывание запуска запрещено<br>1: Прерывание запуска разрешено               |
| 5   | <b>COMIE</b> : Разрешение прерывания COM<br>0: Прерывание COM запрещено<br>1: Прерывание COM разрешено                         |
| 4   | <b>CC4IE</b> : Разрешение прерывания захвата/сравнения 4<br>0: Прерывание CC4 запрещено<br>1: Прерывание CC4 разрешено         |
| 3   | <b>CC3IE</b> : Разрешение прерывания захвата/сравнения 3<br>0: Прерывание CC3 запрещено<br>1: Прерывание CC3 разрешено         |
| 2   | <b>CC2IE</b> : Разрешение прерывания захвата/сравнения 2<br>0: Прерывание CC2 запрещено<br>1: Прерывание CC2 разрешено         |

|   |   |
|---|---|
| 1 | <b>CC1IE</b> : Разрешение прерывания захвата/сравнения 1<br>0: Прерывание CC1 запрещено<br>1: Прерывание CC1 разрешено    |
| 0 | <b>UIE</b> : Разрешение прерывания обновления<br>0: Прерывание обновления запрещено<br>1: Прерывание обновления разрешено |

#### 12.4.5 Регистр статуса TIM1 (TIMx\_SR)

Смещение адреса: 0x10

Значение сброса: 0x0000

|          |    |       |       |       |       |      |       |       |       |       |       |       |       |       |       |
|----------|----|-------|-------|-------|-------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 15       | 14 | 13    | 12    | 11    | 10    | 9    | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| Reserved |    | CC4OF | CC3OF | CC2OF | CC1OF | Res. | BIF   | TIF   | COMIF | CC4IF | CC3IF | CC2IF | CC1IF | UIF   |       |
|          |    | rc_w0 | rc_w0 | rc_w0 | rc_w0 | Res. | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 |

| Бит   | Описание  |
|-------|---|
| 15:13 | Резерв, всегда читается как 0   |
| 12    | <b>C4OF</b> : Флаг перезахвата захвата/сравнения 4  |
| 11    | <b>C3OF</b> : Флаг перезахвата захвата/сравнения 3  |
| 10    | <b>C2OF</b> : Флаг перезахвата захвата/сравнения 2  |
| 9     | <b>C1OF</b> : Флаг перезахвата захвата/сравнения 1<br>Этот флаг устанавливается аппаратно только когда соответствующий канал сконфигурирован в режиме захвата входа. Он очищается программно через запись в него '0'.<br>0: Перезахват не обнаружен<br>1: Значение счетчика было захвачено в регистр TIMx_CCR1 когда уже был установлен флаг CC1F.  |
| 8     | Резерв, всегда читается как 0   |
| 7     | <b>BIF</b> : Флаг прерывания аварийного останова<br>Этот флаг устанавливается аппаратно как только вход останова становится активным. Он очищается программно если вход останова неактивен.<br>0: Событие останова не произошло<br>1: Активный уровень был обнаружен на входе останова  |
| 6     | <b>TIF</b> : Флаг прерывания запуска<br>Этот флаг устанавливается аппаратно по событию запуска(обнаружен активный фронт на входе TRGI когда разрешен контроллер подчиненного режима во всех режимах кроме режима стробирования). Он очищается программно.<br>0: Событие запуска не произошло<br>1: Запрос прерывания запуска  |
| 5     | <b>COMIF</b> : Флаг прерывания COM<br>Этот флаг устанавливается аппаратно по событию COM(когда обновляются управляющие биты захвата/сравнения – CCxE, CCxNE, OCxM).<br>0: Событие COM не произошло<br>1: Запрос прерывания COM  |
| 4     | <b>CC4IF</b> : Флаг прерывания захвата/сравнения 4  |
| 3     | <b>CC3IF</b> : Флаг прерывания захвата/сравнения 3  |
| 2     | <b>CC2IF</b> : Флаг прерывания захвата/сравнения 2  |
| 1     | <b>CC1IF</b> : Флаг прерывания захвата/сравнения 1<br><b>Если канал CC1 сконфигурирован как выход:</b><br>Этот флаг устанавливается аппаратно когда значение счетчика совпадает со значением сравнение за некоторым исключением в режиме выравнивания по центру. Он очищается программно.<br>0: нет совпадения<br>1: Содержимое счетчика TIMx_CNT совпало с содержимым регистра TIMx_CCR1. Когда содержимое TIMx_CCR1 больше содержимого TIMx_ARR, бит CC1IF становится |

|   |   |
|---|---|
|   | <p>высоким при переполнении счетчика(в режиме счета вверх, вниз и вверх/вниз).<br/> <b>Если канал СС1 сконфигурирован на вход:</b><br/> Этот бит устанавливается аппаратно при захвате. Он очищается программно или чтением регистра TIMx_CCR1.<br/> 0: Захват входа не произошел<br/> 1: Значение счетчика было захвачено в регистр TIMx_CCR1(Обнаружен фронт на IC1, который совпадает с выбранной полярностью).</p>  |
| 0 | <p><b>UIF:</b> Флаг прерывания обновления.<br/> Этот бит устанавливается аппаратно по событию обновления. Он очищается программно.<br/> 0: Обновление не произошло<br/> 1: Установка запроса прерывания обновления. Этот бит устанавливается аппаратно когда обновляются регистры:<br/> - По переполнению относительно счетчика повторов(обновление если счетчик повторов = 0) и если UDIS=0 в TIMx_CR1.<br/> - Когда CNT реинициализируется программно используя бит UG регистра TIMx_EGR если URS=0 и UDIS=0 в TIMx_CR1.<br/> - Когда CNT реинициализируется событием запуска если URS=0 и UDIS=0 в TIMx_CR1.</p> |

#### 12.4.6 Регистр генерации событий TIM1 (TIMx\_EGR)

Смещение адреса: 0x14

Значение сброса: 0x0000

|          |    |    |    |    |    |   |   |    |    |      |      |      |      |      |    |
|----------|----|----|----|----|----|---|---|----|----|------|------|------|------|------|----|
| 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7  | 6  | 5    | 4    | 3    | 2    | 1    | 0  |
| Reserved |    |    |    |    |    |   |   | BG | TG | COMG | CC4G | CC3G | CC2G | CC1G | UG |
|          |    |    |    |    |    |   |   | w  | w  | w    | w    | w    | w    | w    | w  |

| Бит  | Описание   |
|------|--|
| 15:8 | Резерв, всегда читается как 0  |
| 7    | <p><b>BG:</b> Генерация аварийного останова<br/> Этот бит устанавливается программно, чтобы сгенерировать событие, он автоматически очищается аппаратно.<br/> 0: Не воздействует<br/> 1: Генерируется событие останова. Бит МОЕ очищается и устанавливается флаг BIF.<br/> При разрешении может происходить связанное прерывание или передача DMA.</p> |
| 6    | <p><b>TG:</b> Генерация запуска<br/> Этот бит устанавливается программно чтобы сгенерировать событие, он автоматически очищается аппаратно.<br/> 0: Не действует<br/> 1: Устанавливается флаг TIF в регистре TIMx_SR. Связанное прерывание или передача DMA могут происходить при разрешении.</p>  |
| 5    | <p><b>COMG:</b> Генерация обновления управления захватом/сравнением<br/> Этот бит устанавливается программно, автоматически очищается аппаратно.<br/> 0: Не действует<br/> 1: Когда установлен бит CCPSC, он позволяет обновить биты CCxE, CCxNE и OCxM.<br/> Прим: Этот бит действует только на каналах с комплементарным выходом.</p>                |
| 4    | <b>CC4G:</b> Генерация захвата/сравнения 4   |
| 3    | <b>CC3G:</b> Генерация захвата/сравнения 3   |
| 2    | <b>CC2G:</b> Генерация захвата/сравнения 2   |
| 1    | <p><b>CC1G:</b> Генерация захвата/сравнения 1<br/> Этот бит устанавливается программно чтобы сгенерировать событие. Он автоматически очищается аппаратно.<br/> 0: Не действует<br/> 1: Генерируется событие захват/сравнение канала 1</p>  |

|   |   |
|---|---|
|   | <p><b>Если канал CC1 сконфигурирован на выход:</b><br/>Устанавливается флаг CC1IF, при разрешении посылается соответствующее прерывание или запрос DMA.</p> <p><b>Если канал CC1 сконфигурирован на вход:</b><br/>Текущее значение счетчика захватывается в регистр TIMx_CCR1. Устанавливается флаг CC1IF, при разрешении посылается соответствующее прерывание или запрос DMA. Флаг CC1OF устанавливается если флаг CC1IF уже был высоким.</p>                             |
| 0 | <p><b>UG:</b> Генерация обновления<br/>Этот бит может устанавливается программно, он автоматически очищается аппаратно.<br/>0: Не действует<br/>1: Реинициализирует счетчик и генерирует обновление регистров. Прим: счетчик предделителя также очищается(значение предделителя не затрагивается). Счетчик очищается если выбран режим выравнивания по центру или если DIR=0(счет вверх), иначе он принимает значение автоперезагрузки(TIMx_ARR) если DIR=1(счет вниз).</p> |

#### 12.4.7 Регистр 1 режима захвата/сравнения TIM1 (TIMx\_CCMR1)

Смещение адреса: 0x18

Значение сброса: 0x0000

Каналы могут использоваться на вход(режим захвата) или на выход(режим сравнения). Направление канала определяется конфигурацией соответствующих битов CCxS. Все остальные биты имеют различные функции в режимах входа и выхода. Для данного бита OCxx описывает его функцию когда канал сконфигурирован на выход, ICxx описывает его функцию когда канал сконфигурирован на вход. Поэтому вы должны быть осторожны т.к. один и тот же бит может иметь различное значение.

|           |           |    |    |             |           |           |    |           |           |    |    |             |           |           |    |
|-----------|-----------|----|----|-------------|-----------|-----------|----|-----------|-----------|----|----|-------------|-----------|-----------|----|
| 15        | 14        | 13 | 12 | 11          | 10        | 9         | 8  | 7         | 6         | 5  | 4  | 3           | 2         | 1         | 0  |
| OC2<br>CE | OC2M[2:0] |    |    | OC2<br>PE   | OC2<br>FE | CC2S[1:0] |    | OC1<br>CE | OC1M[2:0] |    |    | OC1<br>PE   | OC1<br>FE | CC1S[1:0] |    |
| IC2F[3:0] |           |    |    | IC2PSC[1:0] |           |           |    | IC1F[3:0] |           |    |    | IC1PSC[1:0] |           |           |    |
| rw        | rw        | rw | rw | rw          | rw        | rw        | rw | rw        | rw        | rw | rw | rw          | rw        | rw        | rw |

#### В режиме сравнения выхода:

| Бит   | Описание  |
|-------|---|
| 15    | <b>OC2CE:</b> Разрешение очистки сравнения выхода 2   |
| 14:12 | <b>OC2M[2:0]:</b> Режим сравнения выхода 2  |
| 11    | <b>OC2PE:</b> Разрешение предварительной загрузки сравнения выхода 2  |
| 10    | <b>OC2FE:</b> Быстрое разрешение сравнения выхода 2   |
| 9:8   | <p><b>CC2S[1:0]:</b> Выбор захвата/сравнения 2<br/>Это битовое поле определяет направление канала(вход/выход) также как и используемый вход.<br/>00: Канал CC2 сконфигурирован как выход<br/>01: Канал CC2 сконфигурирован как вход, IC2 отображается на TI2<br/>10: Канал CC2 сконфигурирован как выход, IC2 отображается на TI1<br/>11: Канал CC2 сконфигурирован как выход, IC2 отображается на TRC. Этот режим работает только если выбран внутренний запуск через бит TS(регистр TIMx_SMCR).<br/>Прим: Биты CC2S только для записи когда канал отключен(CC2E=0 в TIMx_CCR1).</p> |
| 7     | <p><b>OC1CE:</b> Разрешение очистки сравнения выхода 1<br/>0: Вход ETRF не влияет на OC1Ref<br/>1: OC1Ref очищается как только обнаружен высокий уровень на входе ETRF.</p>   |
| 6:4   | <p><b>OC1M:</b> Режим сравнения выхода 1<br/>Эти биты определяют поведение опорного выходного сигнала OC1REF из которого получаются OC1 и OC1N. OCREF активный высокий тогда как активный уровень OC1 и</p>   |

|     |   |
|-----|---|
|     | <p>OC1N зависит от битов CC1P и CC1NP.</p> <p>000: Заморожен – сравнение между выходным регистром TIMx_CCR1 и счетчиком TIMx_CNT не влияет на выходы(этот режим используется для генерации временной базы).</p> <p>001: По совпадению устанавливает канал 1 в активный уровень. Сигнал OC1REF форсируется высоким когда счетчик TIMx_CNT совпадает с регистром захвата/сравнения 1(TIMx_CCR1).</p> <p>010: По совпадению устанавливает канал 1 в неактивный уровень.</p> <p>011: Изменяет - OC1REF изменяется когда TIMx_CNT= TIMx_CCR1.</p> <p>100: Форсирует неактивный уровень - OC1REF форсируется низким.</p> <p>101: Форсирует активный уровень - OC1REF форсируется высоким.</p> <p>110: Режим ШИМ 1 – При счете вверх канал 1 активен пока TIMx_CNT&lt; TIMx_CCR1 иначе неактивен. При счете вниз канал 1 активен(OC1REF=0) пока TIMx_CNT&gt; TIMx_CCR1, иначе неактивен.</p> <p>111: Режим ШИМ 2 – При счете вверх канал 1 неактивен пока TIMx_CNT&lt; TIMx_CCR1 иначе активен. При счете вниз канал 1 активен пока TIMx_CNT&gt; TIMx_CCR1, иначе неактивен.</p> <p>Прим:</p> <ol style="list-style-type: none"> <li>1. Эти биты не могут модифицироваться при программировании LOCK-уровня 3 и CC1S=00(канал сконфигурирован на выход).</li> <li>2. В режиме ШИМ 1 и 2, уровень OCREF изменяется только когда изменяется результат сравнения или когда режим сравнения выхода переключается из «замороженного» режима в режим «ШИМ».</li> </ol> |
| 3   | <p><b>OC1PE:</b> Разрешение предварительной загрузки сравнения выхода 1</p> <p>0: Предзагрузочный регистр для TIMx_CCR1 запрещен. TIMx_CCR1 может быть записан в любой момент, новое значение принимается немедленно.</p> <p>1: Предзагрузочный регистр для TIMx_CCR1 разрешен. Операции доступа чтения/записи обращаются к предзагрузочному регистру. Предзагруженное значение TIMx_CCR1 загружается в активный регистр по каждому событию обновления.</p> <p>Прим:</p> <ol style="list-style-type: none"> <li>1. Эти биты не могут модифицироваться при программировании LOCK-уровня 3 и CC1S=00(канал сконфигурирован на выход).</li> <li>2. Режим ШИМ может использоваться без подтверждения предзагрузочного регистра только в режиме одного импульса(установлен бит OPM в регистре TIMx_CR1). В противном случае поведение не гарантируется.</li> </ol>   |
| 2   | <p><b>OC1FE:</b> Быстрое разрешение сравнения выхода 1</p> <p>Этот бит используется для ускорения воздействия от события по входу запуска на выход CC.</p> <p>0: CC1 ведет себя обычно в зависимости от значений счетчика и CCR1, даже когда включен запуск. Минимальная задержка от появления фронта на входе запуска и до активации выхода CC1 – 5 тактовых циклов.</p> <p>1: Активный фронт на входе запуска действует на выход CC1 также как совпадение сравнения. Затем ОС устанавливает уровень сравнения независимо от результата сравнения. Задержка от выборки входа запуска и активации выхода CC1 уменьшается до 3 тактовых циклов. OC1FE действует только когда канал сконфигурирован в режиме ШИМ 1 или ШИМ 2.</p>   |
| 1:0 | <p><b>CC1S:</b> Выбор захвата/сравнения 1</p> <p>Это битовое поле определяет направление канала(вход/выход) и используемый вход.</p> <p>00: Канал CC1 сконфигурирован на выход</p> <p>01: Канал CC1 сконфигурирован на вход, IC1 отображается на TI1</p> <p>10: Канал CC1 сконфигурирован на вход, IC1 отображается на TI2</p> <p>11: Канал CC1 сконфигурирован на вход, IC1 отображается на TRC. Этот режим работает только когда выбирается вход внутреннего запуска через бит TS (CC1E=0 в TIMx_CCER)</p>  |

**В режиме захвата входа:**

| Бит   | Описание   |
|-------|--|
| 15:12 | <b>IC2F:</b> Фильтр входа захвата 2  |
| 11:10 | <b>IC2PSC[1:0]</b> : Предделитель входа захвата 2  |
| 9:8   | <b>CC2S:</b> Выбор захвата/сравнения 2<br>Это битовое поле определяет направление канала(вход/выход) также как и используемый вход.<br>00: Канал CC2 сконфигурирован как выход<br>01: Канал CC2 сконфигурирован как вход, IC2 отображается на TI2<br>10: Канал CC2 сконфигурирован как выход, IC2 отображается на TI1<br>11: Канал CC2 сконфигурирован как выход, IC2 отображается на TRC. Этот режим работает только если выбран внутренний запуск через бит TS(регистр TIMx_SMCR).<br>Прим: Биты CC2S только для записи когда канал отключен(CC2E=0 в TIMx_CCER).  |
| 7:4   | <b>IC1F[3:0]:</b> Фильтр входа захвата 1<br>Это битовое поле определяет частоту, используемую для выборки входа TI1 и длину цифрового фильтра, применяемого к TI1. Цифровой фильтр состоит из счетчика событий, которому требуется N событий чтобы передать переход на выход:<br>0000: Без фильтра, выборка делается на fDTS<br>0001: fSAMPLING=fCK_INT, N=2<br>0010: fSAMPLING=fCK_INT, N=4<br>0011: fSAMPLING=fCK_INT, N=8<br>0100: fSAMPLING=fDTS/2, N=6<br>0101: fSAMPLING=fDTS/2, N=8<br>0110: fSAMPLING=fDTS/4, N=6<br>0111: fSAMPLING=fDTS/4, N=8<br>1000: fSAMPLING=fDTS/8, N=6<br>1001: fSAMPLING=fDTS/8, N=8<br>1010: fSAMPLING=fDTS/16, N=5<br>1011: fSAMPLING=fDTS/16, N=6<br>1100: fSAMPLING=fDTS/16, N=8<br>1101: fSAMPLING=fDTS/32, N=5<br>1110: fSAMPLING=fDTS/32, N=6<br>1111: fSAMPLING=fDTS/32, N=8 |
| 3:2   | <b>IC1PSC:</b> Предделитель входа захвата 1<br>Это битовое поле определяет значение предделителя, действующего на входе CC1(IC1). Предделитель сбрасывается когда CC1E=0(регистр TIMx_CCER).<br>00: без предделителя, захват производится каждый раз, когда обнаруживается фронт на входе захвата<br>01: захват делается один раз за каждые 2 события<br>10: захват делается один раз за каждые 4 события<br>11: захват делается один раз за каждые 8 событий  |
| 1:0   | <b>CC1S:</b> Выбор захвата/сравнения 1<br>Это битовое поле определяет направление канала(вход/выход), также как и используемый вход.<br>00: Канал CC1 сконфигурирован как выход<br>01: Канал CC1 сконфигурирован как вход, IC1 отображается на TI1<br>10: Канал CC1 сконфигурирован как выход, IC1 отображается на TI2<br>11: Канал CC1 сконфигурирован как выход, IC1 отображается на TRC. Этот режим работает только если выбран внутренний запуск через бит TS(регистр TIMx_SMCR).<br>Прим: Биты CC1S только для записи когда канал отключен(CC1E=0 в TIMx_CCER).   |

### 12.4.8 TIM1 Регистр 2 режима захвата/сравнения (TIMx\_CCMR2)

Смещение адреса: 0x1C

Значение сброса: 0x0000

См. описание регистра CCMR1.

|           |           |    |    |             |           |           |    |           |           |    |    |             |           |           |    |
|-----------|-----------|----|----|-------------|-----------|-----------|----|-----------|-----------|----|----|-------------|-----------|-----------|----|
| 15        | 14        | 13 | 12 | 11          | 10        | 9         | 8  | 7         | 6         | 5  | 4  | 3           | 2         | 1         | 0  |
| OC4<br>CE | OC4M[2:0] |    |    | OC4<br>PE   | OC4<br>FE | CC4S[1:0] |    | OC3<br>CE | OC3M[2:0] |    |    | OC3<br>PE   | OC3<br>FE | CC3S[1:0] |    |
| IC4F[3:0] |           |    |    | IC4PSC[1:0] |           |           |    | IC3F[3:0] |           |    |    | IC3PSC[1:0] |           |           |    |
| rw        | rw        | rw | rw | rw          | rw        | rw        | rw | rw        | rw        | rw | rw | rw          | rw        | rw        | rw |

### 12.4.9 TIM1 Регистр разрешения захвата/сравнения (TIMx\_CCER)

Смещение адреса: 0x20

Значение сброса: 0x0000

|          |    |      |      |       |       |      |      |       |       |      |      |       |       |      |      |
|----------|----|------|------|-------|-------|------|------|-------|-------|------|------|-------|-------|------|------|
| 15       | 14 | 13   | 12   | 11    | 10    | 9    | 8    | 7     | 6     | 5    | 4    | 3     | 2     | 1    | 0    |
| Reserved |    | CC4P | CC4E | CC3NP | CC3NE | CC3P | CC3E | CC2NP | CC2NE | CC2P | CC2E | CC1NP | CC1NE | CC1P | CC1E |
|          |    | rw   | rw   | rw    | rw    | rw   | rw   | rw    | rw    | rw   | rw   | rw    | rw    | rw   | rw   |

| Бит   | Описание  |
|-------|---|
| 15:14 | Резерв, всегда читается как 0   |
| 13    | <b>CC4P</b> : Полярность выхода захвата/сравнения 4   |
| 12    | <b>CC4E</b> : Разрешение выхода захвата/сравнения 4   |
| 11    | <b>CC3NP</b> : Полярность комплементарного выхода захвата/сравнения 3   |
| 10    | <b>CC3NE</b> : Разрешение комплементарного выхода захвата/сравнения 3   |
| 9     | <b>CC3P</b> : Полярность выхода захвата/сравнения 3   |
| 8     | <b>CC3E</b> : Разрешение выхода захвата/сравнения 3   |
| 7     | <b>CC2NP</b> : Полярность комплементарного выхода захвата/сравнения 2   |
| 6     | <b>CC2NE</b> : Разрешение комплементарного выхода захвата/сравнения 2   |
| 5     | <b>CC2P</b> : Полярность выхода захвата/сравнения 2   |
| 4     | <b>CC2E</b> : Разрешение выхода захвата/сравнения 2   |
| 3     | <b>CC2NP</b> : Полярность комплементарного выхода захвата/сравнения 2<br>0: OC1N активный высокий<br>1: OC1N активный низкий<br>Прим: Этот бит не записывается когда был запрограммирован уровень LOCK-2 или 3 (TIMx_BDTR) и CC1S=00(канал сконфигурирован на выход).   |
| 2     | <b>CC1NE</b> : Разрешение комплементарного выхода захвата/сравнения<br>0: Отключен - OC1N не активный. Уровень OC1N является функцией битов MOE, OSSI, OSSR, OIS1, OIS1N и CC1E.<br>1: Включен – сигнал OC1N выводится на соответствующий выход в зависимости от битов MOE, OSSI, OSSR, OIS1, OIS1N и CC1E.   |
| 1     | <b>CC1P</b> : Полярность выхода захвата/сравнения 1<br><b>Канал CC1 сконфигурирован на выход:</b><br>0: OC1 активный высокий<br>1: OC1 активный низкий<br><b>Канал CC1 сконфигурирован на вход:</b><br>Этот бит выбирает IC1 для операций захвата или запуска.<br>0: неинвертирован: захват происходит по нарастающему фронту IC1. При использовании как внешний запуск, IC1 не инвертируется.<br>1: инвертирован: захват происходит по спадающему фронту IC1. При использовании для внешнего запуска IC1 инвертируется.<br>Прим: Этот бит не записывается когда был запрограммирован уровень LOCK-2 или 3 (TIMx_BDTR). |
| 0     | <b>CC1E</b> : Разрешение выхода захвата/сравнения 1<br><b>Канал CC1 сконфигурирован на выход:</b>   |

|  |
|--|
| <p>0: Отключен – OC1 не активен. Уровень OC1 является функцией битов MOE, OSSI, OSSR, OIS1, OIS1N и CC1NE.</p> <p>1: Включен – сигнал OC1 выводится на соответствующий вывод в зависимости от битов MOE, OSSI, OSSR, OIS1, OIS1N и CC1NE.</p> <p><b>Канал CC1 сконфигурирован на вход:</b></p> <p>Этот бит определяет, делается ли захват значения счетчика в регистр захвата/сравнения(TIMx_CCR1).</p> <p>0: Захват запрещен</p> <p>1: Захват разрешен.</p> |
|--|

Таблица 68. Биты управления каналами с комплементарными выходами OCx и OCxN и возможностью останова.

| Управляющие биты |          |          |          |           | Состояние выходов(1)   |   |
|------------------|----------|----------|----------|-----------|--|---|
| Бит MOE          | Бит OSSI | Бит OSSR | Бит CCxE | Бит CCxNE | Состояние выхода OCx   | Состояние выхода OCxN   |
| 1                | X        | 0        | 0        | 0         | Выход запрещен(не управляется таймером)<br>OCx=0, OCx_EN=0                               | Выход запрещен(не управляется таймером)<br>OCx=0, OCx_EN=0                                  |
|                  |          | 0        | 0        | 1         | Выход запрещен(не управляется таймером)<br>OCx=0, OCx_EN=0                               | OCxREF + Полярность<br>OCxN=OCxREF<br>хор CCxNP, OCxN_EN=1                                  |
|                  |          | 0        | 1        | 0         | OCxREF + Полярность<br>OCx=OCxREF хор CCxP,<br>OCx_EN=1                                  | Выход запрещен(не управляется таймером)<br>OCx=0, OCx_EN=0                                  |
|                  |          | 0        | 1        | 1         | OCREF + Полярность +<br>мертвое время<br>OCx_EN=1  | Комплементарный к OCREF<br>(не OCREF) + Полярность +<br>мертвое время<br>OCxN_EN=1          |
|                  |          | 1        | 0        | 0         | Выход запрещен(не управляется таймером)<br>OCx=CCxP, OCx_EN=0                            | Выход запрещен(не управляется таймером)<br>OCxN=CCxNP, OCxN_EN=0                            |
|                  |          | 1        | 0        | 1         | Отключенное состояние<br>(выход разрешен,<br>неактивное состояние)<br>OCx=CCxP, OCx_EN=1 | OCxREF + Полярность<br>OCxN=OCxREF хор CCxNP,<br>OCxN_EN=1                                  |
|                  |          | 1        | 1        | 0         | OCxREF + Полярность<br>OCx=OCxREF хор CCxP,<br>OCx_EN=1                                  | Отключенное состояние<br>(выход разрешен, неактивное<br>состояние)<br>OCxN=CCxNP, OCxN_EN=1 |
|                  |          | 1        | 1        | 1         | OCREF + Полярность +<br>мертвое время<br>OCx_EN=1  | Комплементарный к OCREF<br>(не OCREF) + Полярность +<br>Мертвое время<br>OCxN_EN=1          |

1. Когда не используются оба выхода канала(CCxE=CCxNE=0), тогда должны быть очищены биты OISx, OISxN, CCxP и CCxNP.

Прим: Состояние внешних выводов I/O, подключенных к комплементарным каналам OCx и OCxN, зависит от состояния канала OCx и OCxN, и регистров GPIO.



### 12.4.10 Счетчик TIM1 (TIMx\_CNT)

Смещение адреса: 0x24

Значение сброса: 0x0000

|           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 15        | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| CNT[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| rw        | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Бит 15:0      **CNT[15:0]** : Значение счетчика

### 12.4.11 Предделитель TIM1 (TIMx\_PSC)

Смещение адреса: 0x28

Значение сброса: 0x0000

|           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 15        | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| PSC[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| rw        | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Бит 15:0      **PSC[15:0]** : Значение предделителя  
 Тактовая частота счетчика(CK\_INT) равна  $f_{CK\_PSC} / (PSC[15:0] + 1)$ .

PSC содержит значение, загружаемое в активный регистр предделителя по каждому событию обновления(включая случай когда счетчик очищается через бит UG в регистре TIMx\_EGR или контроллер запуска, сконфигурированный в «режиме сброса»).

### 12.4.12 Регистр автоперезагрузки TIM1 (TIMx\_ARR)

Смещение адреса: 0x2C

Значение сброса: 0x0000

|           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 15        | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| ARR[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| rw        | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Бит 15:0      **ARR[15:0]** : Значение ARR загружается в действующий регистр автоперезагрузки. Счетчик блокируется когда содержит нулевое значение автоперезагрузки.

### 12.4.13 Регистр счетчика повторов TIM1 (TIMx\_RCR)

Смещение адреса: 0x30

Значение сброса: 0x0000

|          |    |    |    |    |    |   |   |          |    |    |    |    |    |    |    |
|----------|----|----|----|----|----|---|---|----------|----|----|----|----|----|----|----|
| 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| Reserved |    |    |    |    |    |   |   | REP[7:0] |    |    |    |    |    |    |    |
|          |    |    |    |    |    |   |   | rw       | rw | rw | rw | rw | rw | rw | rw |

| Бит  | Описание   |
|------|--|
| 15:8 | Резерв, всегда читается как 0  |
| 7:0  | <p><b>REP[7:0]</b> : Значение счетчика повторов</p> <p>Эти биты позволяют пользователю установить скорость обновления регистров сравнения(т.е. периодичность пересылок из предзагрузочных в активные регистры) когда разрешены предзагрузочные регистры, также как и скорость генерации прерываний обновления если это прерывание разрешено.</p> <p>Каждый раз, когда счетчик REP_CNT считая вниз, достигает нуля, генерируется событие обновления и он перезапускает счет со значения REP. Т.к. REP_CNT перезагружается значением REP только при событии обновления повтора U_RC, то любая запись не принимается до момента следующего события повтора.</p> <p>Это значит, что в режиме ШИМ(REP+1) соответствует:</p> <ul style="list-style-type: none"> <li>- количеству периодов ШИМ в режиме выравнивания по фронту</li> <li>- количеству полупериодов ШИМ в режиме выравнивания по центру.</li> </ul> |



## 12.4.18 Регистр аварийного останова и мертвого времени TIM1 (TIMx\_BDTR)

Смещение адреса: 0x44

Значение сброса: 0x0000

|     |     |     |     |      |      |           |    |          |    |    |    |    |    |    |    |
|-----|-----|-----|-----|------|------|-----------|----|----------|----|----|----|----|----|----|----|
| 15  | 14  | 13  | 12  | 11   | 10   | 9         | 8  | 7        | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| MOE | AOE | BKP | BKE | OSSR | OSSI | LOCK[1:0] |    | DTG[7:0] |    |    |    |    |    |    |    |
| rw  | rw  | rw  | rw  | rw   | rw   | rw        | rw | rw       | rw | rw | rw | rw | rw | rw | rw |

Прим: Т.к. биты AOE, BKP, BKE, OSSI, OSSR и DTG[7:0] могут быть заблокированы от записи в зависимости от конфигурации LOCK, то может потребоваться сконфигурировать их всех во время первой записи в регистр TIMx\_BDTR.

| Бит | Описание   |
|-----|--|
| 15  | <p><b>MOE</b> : Главное разрешение выхода</p> <p>Этот бит очищается асинхронно аппаратно как только вход останова становится активным. Он устанавливается программно или автоматически в зависимости от бита AOE. Он действует на каналы, сконфигурированные на выход.</p> <p>0: выходы OC и OCN запрещены или форсированы в состоянии ожидания.</p> <p>1: выходы OC и OCN разрешены если установлены их соответствующие биты разрешения (CCxE и CCxNE в регистре TIMx_CCER).</p>  |
| 14  | <p><b>AOE</b> : Автоматическое разрешение выхода</p> <p>0: MOE может устанавливаться только программно</p> <p>1: MOE может устанавливаться программно или автоматически по следующему событию обновления(если вход останова не активен).</p> <p>Прим: Этот бит не может модифицироваться если запрограммирован уровень 1 LOCK.</p>   |
| 13  | <p><b>BKP</b> : Полярность аварийного останова.</p> <p>0: Вход останова BRK активный низкий</p> <p>1: Вход останова BRK активный высокий.</p> <p>Прим: Этот бит не может модифицироваться если запрограммирован уровень 1 LOCK.</p> <p>Прим: Любая операция записи в этот бит перед принятием требует задержку в 1 тактовый цикл APB</p>   |
| 12  | <p><b>BKE</b> : Разрешение аварийного останова</p> <p>0: входы останова(BRK и событие сбоя тактирования CCS) запрещены</p> <p>1: входы останова(BRK и событие сбоя тактирования CCS) разрешены</p> <p>Прим: Этот бит не может модифицироваться если запрограммирован уровень 1 LOCK.</p> <p>Прим: Любая операция записи в этот бит перед принятием требует задержку в 1 тактовый цикл APB</p>  |
| 11  | <p><b>OSSR</b> : Выбор отключенного состояния в рабочем режиме</p> <p>Этот бит используется когда MOE=1 на каналах, имеющих комплементарный выход которые сконфигурированы как выходы. OSSR не действует если в таймере не задействованы комплементарные выходы.</p> <p>0: Когда не активный, выходы OC/OCN запрещены(сигнал разрешения выхода OC/OCN=0).</p> <p>1: Когда не активный, выходы OC/OCN разрешаются с их неактивным уровнем как только CCxE=1 или CCxNE=1. Тогда сигнал разрешения выхода OC/OCN=1.</p> <p>Прим: Этот бит не может модифицироваться если запрограммирован уровень 2 LOCK.</p> |
| 10  | <p><b>OSSI</b> : Выбор отключенного состояния для режима ожидания</p> <p>Этот бит используется когда MOE=0 на каналах, сконфигурированных как выходы.</p> <p>0: Когда не активный, выходы OC/OCN запрещены(сигнал разрешения выхода OC/OCN=0)</p> <p>1: Когда не активный, выходы OC/OCN сперва переводятся в их уровень ожидания когда CCxE=1 или CCxNE=1. Сигнал разрешения выхода OC/OCN=1.</p> <p>Прим: Этот бит не может модифицироваться если запрограммирован уровень 2 LOCK.</p>   |
| 9:8 | <p><b>LOCK[1:0]</b> : Конфигурация блокировки.</p>   |

|     |  |
|-----|--|
|     | <p>Эти биты предлагают защиту от записи в случае программных ошибок.</p> <p>00: блокировка отключена – нет защиты от записи битов.</p> <p>01: LOCK уровень 1 = биты DTG в TIMx_BDTR, биты OISx и OISxN в регистре TIMx_CR2 и биты BKE/BKP/AOE в регистре TIMx_BDTR более не могут записываться.</p> <p>10: LOCK уровень 2 = LOCK уровень 1 + биты полярности CC (биты CCxP/CCxNP в регистре TIMx_CCER, также как и относящийся канал, сконфигурированный на выход через биты CCxS) также как биты OSSR и OSSI более не могут записываться.</p> <p>11: LOCK уровень 3 = LOCK уровень 2 + управляющие биты CC(биты OCxM и OCxPE в регистре TIMx_CCMRx, также как и относящийся канал, сконфигурированный на выход через биты CCxS) больше не могут записываться.</p> <p>Прим: Биты LOCK могут записываться только после сброса. После записи регистра TIMx_BDTR их содержимое замораживается до следующего сброса.</p> |
| 7:0 | <p><b>DTG[7:0]</b> : Установка генератора мертвого времени</p> <p>Это битовое поле определяет длительность мертвого времени, вставляемого между комплементарными выходами. DT соответствует этой длительности.</p> <p>DTG[7:5]=0xx =&gt; DT=DTG[7:0]x tdtg с tdtg=tDTS.</p> <p>DTG[7:5]=10x =&gt; DT=(64+DTG[5:0])xtdtg с Tdtg=2xtDTS.</p> <p>DTG[7:5]=110 =&gt; DT=(32+DTG[4:0])xtdtg с Tdtg=8xtDTS.</p> <p>DTG[7:5]=111 =&gt; DT=(32+DTG[4:0])xtdtg с Tdtg=16xtDTS.</p> <p>Пример. Если TDTS=125ns (8MHz), то возможные значения мертвого времени:</p> <p>0 до 15875 ns, шаг 125 ns,</p> <p>16 us до 31750 ns, шаг 250 ns,</p> <p>32 us до 63us , шаг 1 us,</p> <p>64 us до 126 us , шаг 2 us</p> <p>Прим: Это битовое поле не может быть модифицировано когда запрограммирован уровень LOCK 1, 2 или 3.</p>   |

#### 12.4.19 Регистр управления DMA TIM1 (TIMx\_DCR)

Смещение адреса: 0x48

Значение сброса: 0x0000

|          |    |    |    |    |          |    |    |    |    |          |   |   |   |   |    |    |    |    |    |
|----------|----|----|----|----|----------|----|----|----|----|----------|---|---|---|---|----|----|----|----|----|
| 15       | 14 | 13 | 12 | 11 | 10       | 9  | 8  | 7  | 6  | 5        | 4 | 3 | 2 | 1 | 0  |    |    |    |    |
| Reserved |    |    |    |    | DBL[4:0] |    |    |    |    | Reserved |   |   |   |   |    |    |    |    |    |
|          |    |    |    |    | rw       | rw | rw | rw | rw |          |   |   |   |   | rw | rw | rw | rw | rw |

| Бит   | Описание   |
|-------|--|
| 15:13 | Резерв, всегда читается как 0  |
| 12:8  | <p><b>DBL[4:0]</b> : Длина вспышки DMA</p> <p>Этот 5-битовый вектор определяет длину пересылок DMA(таймер распознает пересылку вспышки когда производится доступ чтения или записи по адресу TIMx_DMAR) т.е. количество пересылок. Пересылками могут быть полуслова или байты(см пример ниже).</p> <p>00000: 1 пересылка</p> <p>00001: 2 пересылки</p> <p>00010: 3 пересылки</p> <p>...</p> <p>10001: 18 пересылок</p> |
| 7:5   | Резерв, всегда читается как 0  |
| 4:0   | <p><b>DBA[4:0]</b> : Базовый адрес DMA</p> <p>Этот 5-битный вектор определяет базовый адрес для пересылок DMA(когда производится доступ чтения/записи через адрес TIMx_DMAR). DBA определяется как смещение начиная с адреса регистра TIMx_CR1.</p> <p>Пример:</p> <p>00000: TIMx_CR1,</p> <p>00001: TIMx_CR2,</p>   |

00010: TIMx\_SMCR,

...

Пример: Рассмотрим следующую пересылку: DBL=7 пересылок и DBA=TIMx\_CR1. В этом случае пересылка делается в/из 7 регистров, начиная с адреса TIMx\_CR1.

#### 12.4.20 Адрес полной пересылки DMA TIM1(TIMx\_DMAR)

Смещение адреса: 0x4C

Значение сброса: 0x0000

|           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 15        | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| DMA[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| rw        | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Бит 15:0 **DMA[15:0]** : Регистр DMA для доступа вспышки

Доступ чтения или записи к регистру DMAR производит доступ к регистру, расположенному по адресу (TIMx\_CR1 address) + (DBA + DMA index) x 4 .

Где адрес TIMx\_CR1 является адресом управляющего регистра 1, DBA – это базовый адрес DMA, сконфигурированный в регистре TIMx\_DCR, индекс DMA автоматически управляется пересылкой DMA и находится в пределах от 0 до DBL(DBL конфигурируется в TIMx\_DCR).

#### Пример использования возможности вспышки DMA

В данном примере используется возможность вспышки DMA таймера для обновления содержимого регистров CCRx (x = 2, 3, 4) с полусловной передачей в регистры CCRx. Это делается в следующих шагах:

1. Сконфигурировать соответствующий канал DMA:

- периферийным адресом канала DMA является адрес регистра DMAR
- адресом памяти канала DMA является адрес буфера в ОЗУ, содержащий данные, пересылаемые через DMA в регистры CCRx.
- количество пересылаемых данных = 3(см. примечание ниже)
- циклический режим запрещен

2. Сконфигурировать регистр DCR через конфигурацию битовых полей DBA и DBL: DBL = 3 пересылки, DBA = 0xE.

3. Разрешить DMA-запрос обновления TIMx(установить бит UDE в регистре DIER).

4. Разрешить TIMx.

5. Разрешить канал DMA.

Прим: Данный пример для случая, когда каждый регистр CCRx обновляется один раз. Если каждый регистр CCRx обновляется к примеру дважды, то количество данных на пересылку должно быть 6. Например, буфер в ОЗУ содержит data1, data2, data3, data4, data5 и data6. Данные пересылаются в регистры CCRx следующим образом: по первому запросу обновления DMA data1 передается в CCR2, data2 передается в CCR3, data3 передается в CCR4 и по второму запросу обновления DMA, data4 передается в CCR2, data5 передается в CCR3 и data6 передается в CCR4.

#### 12.4.21 Карта регистров TIM1

Регистры TIM1 отображаются как 16-битные адресуемые регистры.



| Offset | Register    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17         | 16  | 15  | 14  | 13       | 12   | 11         | 10      | 9        | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |
|--------|-------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|-----|-----|-----|----------|------|------------|---------|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x34   | TIMx_CCR1   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | CCR1[15:0] |     |     |     |          |      |            |         |          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|        | Reset value |          |    |    |    |    |    |    |    |    |    |    |    |    |    | 0          | 0   | 0   | 0   | 0        | 0    | 0          | 0       | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x38   | TIMx_CCR2   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | CCR2[15:0] |     |     |     |          |      |            |         |          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|        | Reset value |          |    |    |    |    |    |    |    |    |    |    |    |    |    | 0          | 0   | 0   | 0   | 0        | 0    | 0          | 0       | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x3C   | TIMx_CCR3   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | CCR3[15:0] |     |     |     |          |      |            |         |          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|        | Reset value |          |    |    |    |    |    |    |    |    |    |    |    |    |    | 0          | 0   | 0   | 0   | 0        | 0    | 0          | 0       | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x40   | TIMx_CCR4   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | CCR4[15:0] |     |     |     |          |      |            |         |          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|        | Reset value |          |    |    |    |    |    |    |    |    |    |    |    |    |    | 0          | 0   | 0   | 0   | 0        | 0    | 0          | 0       | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x44   | TIMx_BDTR   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | MOE        | AOE | BKP | BKE | OSSR     | OSSI | LOCK [1:0] | DT[7:0] |          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|        | Reset value |          |    |    |    |    |    |    |    |    |    |    |    |    |    | 0          | 0   | 0   | 0   | 0        | 0    | 0          | 0       | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x48   | TIMx_DCR    | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | DBL[4:0]   |     |     |     | Reserved |      |            |         | DBA[4:0] |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|        | Reset value |          |    |    |    |    |    |    |    |    |    |    |    |    |    | 0          | 0   | 0   | 0   | 0        | 0    | 0          | 0       | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x4C   | TIMx_DMAR   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | DMAB[15:0] |     |     |     |          |      |            |         |          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|        | Reset value |          |    |    |    |    |    |    |    |    |    |    |    |    |    | 0          | 0   | 0   | 0   | 0        | 0    | 0          | 0       | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Перевод основывается на тексте 3-й и 4-й версий руководства RM0041.

Если у вас что-то упорно не получается, попробуйте заглянуть в оригинал. Вполне возможно, что перевод где-то содержит ошибки.

Автор перевода: UK8AMK