**Таймеры общего назначения (TIM2 – TIM5).**

Эта секция применима ко всему семейству STM32F20x и STM32F21x, если не оговорено специально.

**Введение в таймеры TIM2-TIM5.**

Таймеры общего назначения состоят 16-битного или 32-битного автоперезагружаемого счетчика, управляемого программируемым предделителем.

Они могут быть использованы для различных целей, включая измерение длины импульса входных сигналов (захват входа) или генерации выходных сигналов (выход сравнения и ШИМ).

Длина импульса и период выходного сигнала могут быть модулированы в значения от нескольких микросекунд до нескольких миллисекунд, используя предделитель таймера и предделители контроллера тактовой частоты.

Таймеры полностью независимы, и не разделяют каких-либо ресурсов. Они могут быть синхронизированы вместе, как описано в секции 14.3.15.

**Главные особенности TIM2-TIM5.**

Особенности таймеров общего назначения TIMx включают:

* 16-битный (TIM3 и TIM4) или 32-битный (TIM2 и TIM5) суммирующий, вычитающий, реверсивный автоперезагружаемый счетчик.
* 16-битный программируемый предделитель, используемый для деления (также «на лету») тактовой частоты счетчика на любой коэффициент между 1 и 65536.
* До 4ех независимых каналов для:
  + Захвата входа.
  + Выхода сравнения.
  + генерации ШИМ (фронтовой и выровненный по центру режимы).
  + Режим вывода одиночного импульса.
* Схема синхронизации для управления таймером внешними сигналами и взаимодействия нескольких таймеров.
* Генерация прерывания/DMA по следующим событиям:
  + Обновление: счетчик переполнен/опустошен, инициализация счетчика (программно или внутренним/внешним триггером).
  + Событие триггера (старт счетчика, стоп, инициализация или счет по внутреннему/внешнему триггеру).
  + Захват входа.
  + Выход сравнения.
* Поддержка инкрементного кодера и схемы датчика Холла для позиционирования целей.
* Триггерный ввод для внешней частоты или cycle-by-cycle current management.

**Функциональное описание.**

**Основная единица времени.**

Главный блок программируемого таймера – это 16-битный/32-битный счетчик и связанный с ним автоперезагружаемый регистр. Счетчик может считать вверх, вниз, или в обоих направлениях. Тактовая частота счетчика может быть поделена на значение предделителя.

Счетчик, автоперезагружаемый регистр и регистр предделителя могут быть записаны или прочитаны программой. Это справедливо, даже когда счетчик считает.

Основная временная единица включает:

* Регистр счетчика (TIMx\_CNT)
* Регистр предделителя (TIMx\_PSC)
* Автоперезагружаемый регистр (TIMx\_ARR)

Регистр автоперезагрузки предварительно загружен. Запись в или чтение из автоперезагружаемого регистра осуществляет доступ к регистру предварительной загрузки. Содержимое регистра предварительной загрузки передается в теневой регистр безвозвратно (раз и на всегда) или на каждом событии обновления (UEV), в зависимости от бита разрешения автоперезагрузки предварительным значением (ARPE) в регистре TIMx\_CR1. Событие обновления отсылается, когда счетчик достигает переполнения (или опустошения когда – когда вычитает) и если UDIS бит равен 0 в регистре TIMx\_CR1. Это также может быть сгенерировано программно. Генерация события обновления описана в деталях для каждой конфигурации.

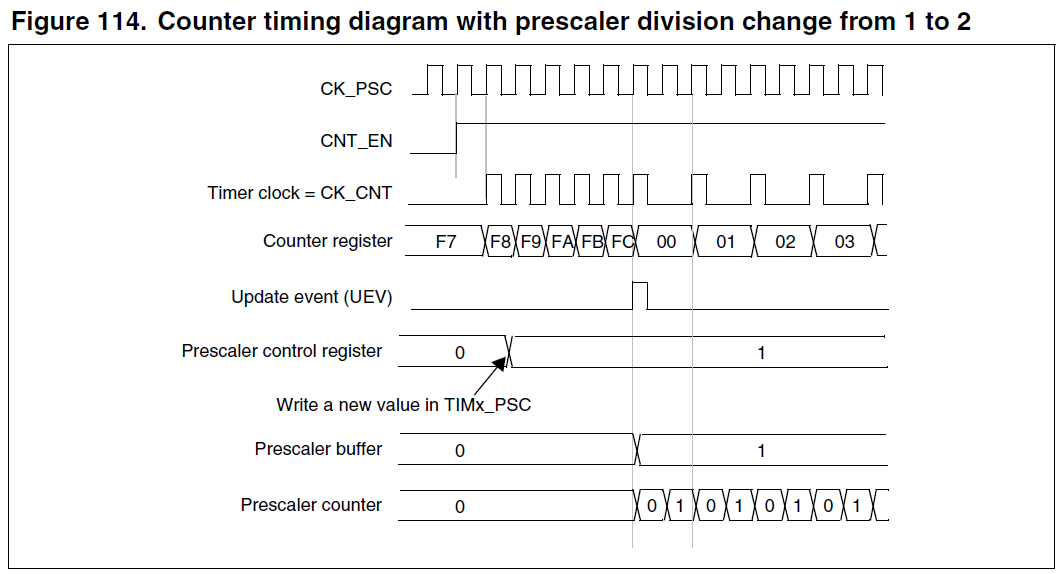
Счетчик тактируется выходом предделителя CK\_CNT, который активен, только когда бит разрешения счетчика (CEN) в регистре TIMx\_CR1 установлен.

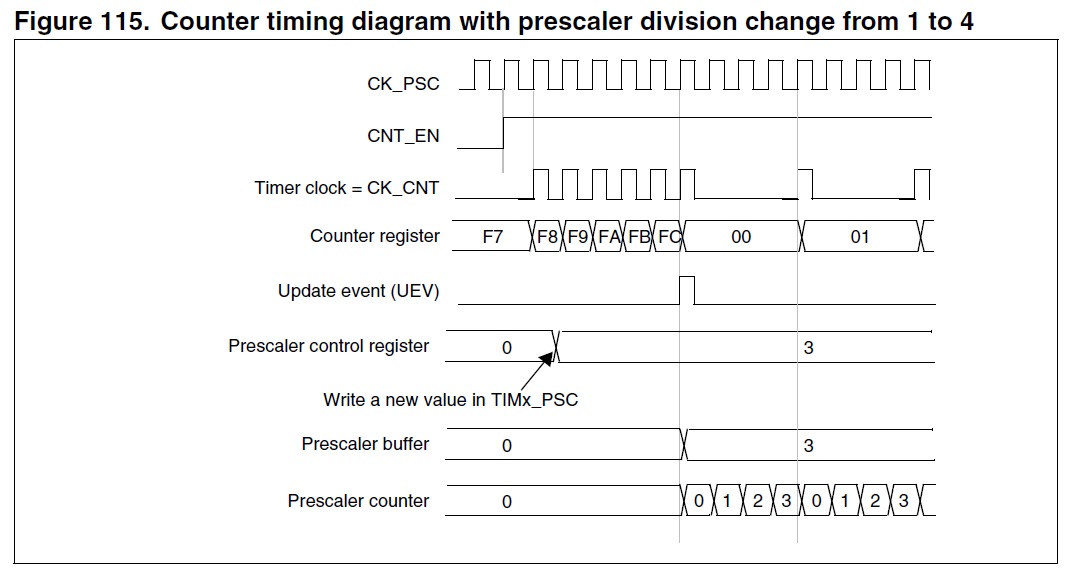
Заметьте, что реальный сигнал разрешения счета CNT\_EN устанавливается спустя один тактовый цикл после установки CEN.

**Описание предделителя.**

Предделитель может делить частоту тактирования счетчика на любой коэффициент между 1 и 65536. Предделитель основывается на 16-битном счетчике, управляемым через 16-битный/32-битный регистр (TIMx\_PSC регистр) (ну то есть он (счетчик делителя) там до скольких то досчитает, и выдаст тактовый импульс. Потом опять сосчитает сколько-то и снова выдаст тактовый импульс. А число счета задается в 16 битном регистре – примечание автора). Предделитель может перестраиваться на лету, поскольку этот управляющий регистр буферизирован. Новое значение предделителя будет взято (из этого самого буфера видимо – примечение автора) на следующем событии обновления.

Рисунок 114 и 115 дает некоторые примеры поведения счетчика, когда предделитель меняется на лету.





**Режимы счета.**

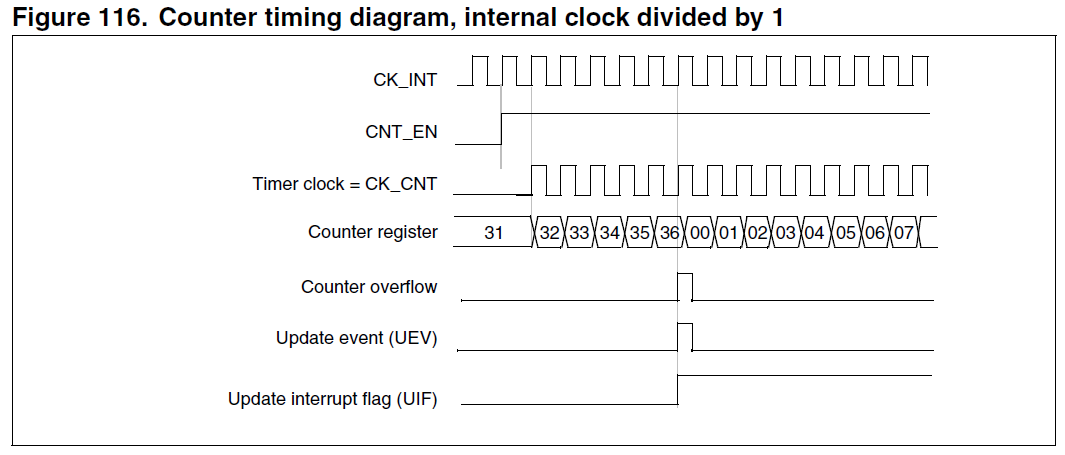
**Суммирующий режим.**

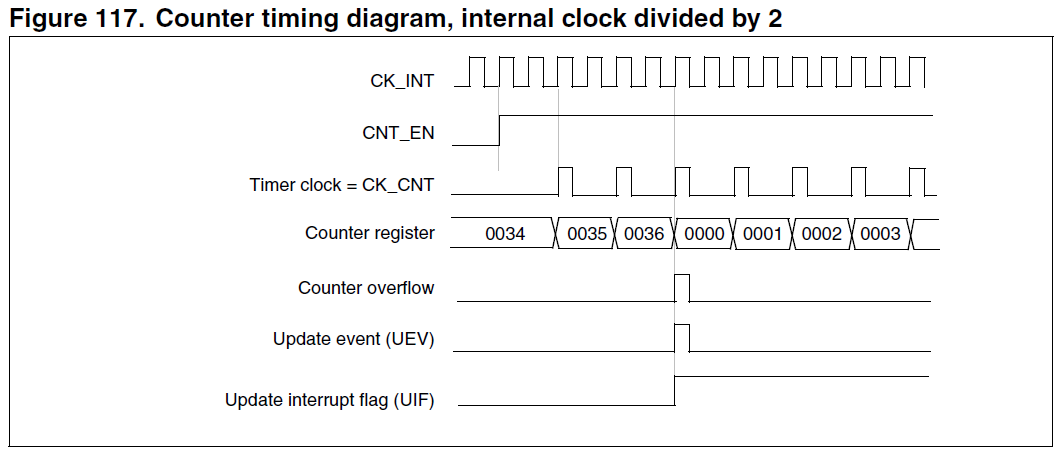
В суммирующем режиме счетчик считает от 0 до значения автоперезагрузки (содержимого регистра TIMx\_ARR), затем начинает заново от 0 и генерирует событие переполнения счетчика. Событие обновления может быть сгенерировано на каждом переполнении счетчика или установкой бита UG в регистре TIMx\_EGR (программно или используя контроллер режима ведомого). Событие UEV может быть запрещено программно, установкой бита UDIS в регистре TIMx\_CR1. Это необходимо для избежания обновления теневых регистров во время записи новых значений в регистры предварительной загрузки. Теперь события обновления не будут происходить до тех пор, пока бит UDIS не будет очищен. Однако, счетчик начинает заново с нуля, также как и счетчик предделителя (но скорость деления не изменяется). В дополнение, если бит URS (выбор запроса обновления) в регистре TIMx\_CR1 установлен, установка бита UG генерирует событие обновления UEV, но без установки UIF флага (таким образом не происходит запроса прерывания или DMA). Это необходимо для избежания прерываний как при обновлении, так и при захвате, когда очищается счетчик на событии захвата.

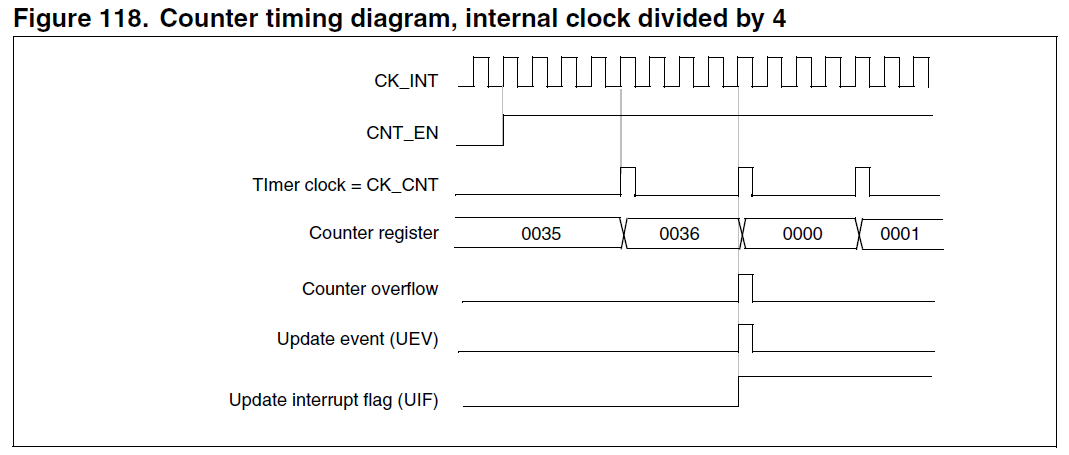
Когда происходит событие обновления, все регистры обновляются и флаг обновления (UIF бит в регистре TIMx\_SR) устанавливается (в зависимости от URS бита):

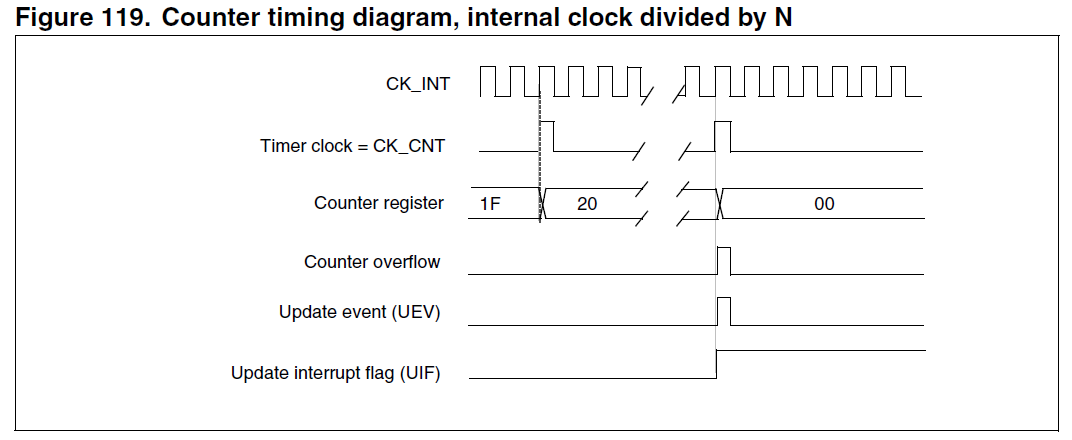
* Буфер предделителя перезагружается значением из регистра управления предделителем (содержимым TIMx\_PSC регистра).
* Теневой регистр автоперезагрузки обновляется значением регистра предзагрузки (TIMx\_ARR).

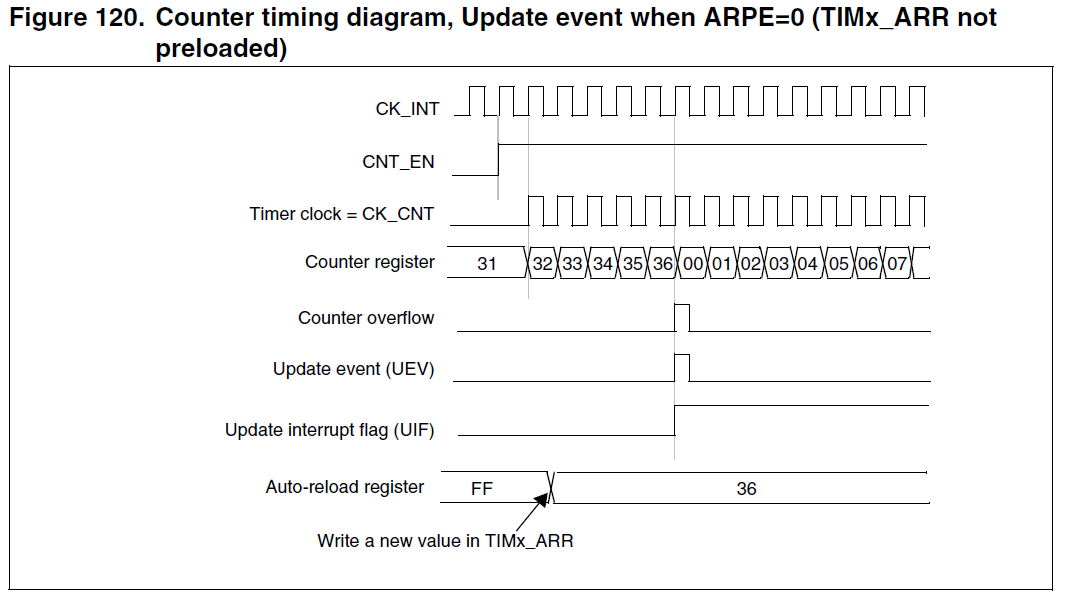
Следующие рисунки показывают несколько примеров счетчиков для различных тактовых частот, когда TIM\_ARR=0x36.

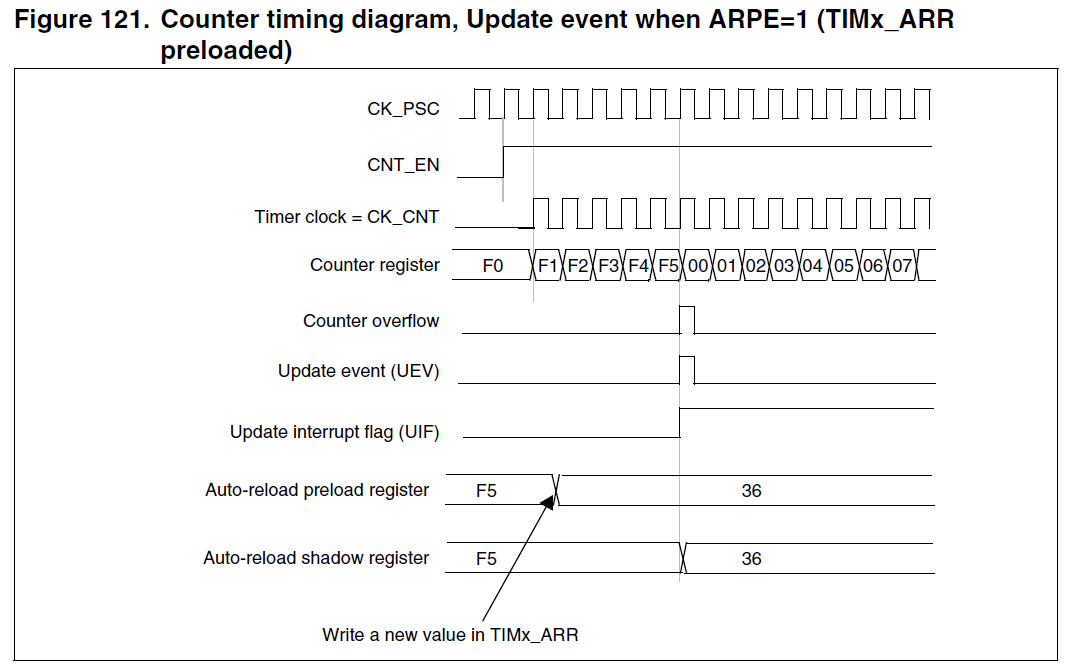












Примечание автора: теневой регистр видимо нужен для ускоренной возможности подгрузки значения порога срабатывания таймера. Раз записали в TIMx\_ARR, а дальше постоянно молотит из теневых регистров.

**Вычитающий режим.**

Потом напишу.

**Выровненный по центру режим (реверсивный).**

Тоже потом напишу.

**Выбор тактовой частоты.**

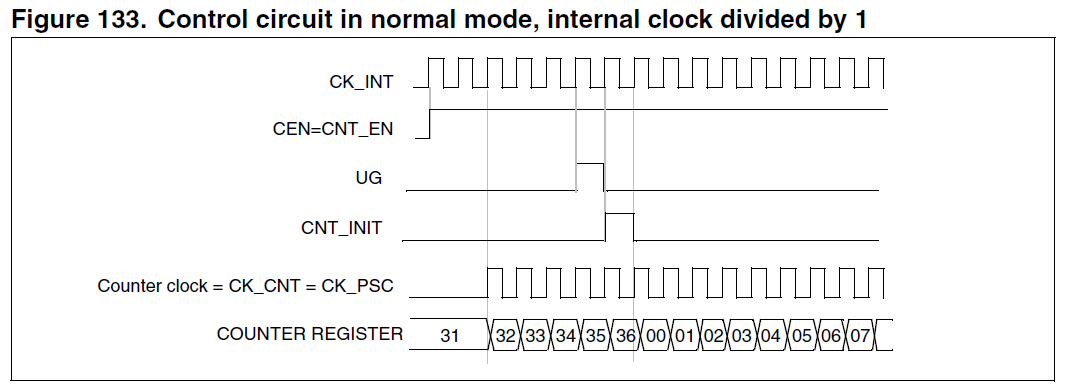
Частота тактирования счетчика может быть обеспечена следующими источниками:

* Внутренняя частота (CK\_INT).
* Внешняя частота, режим 1: внешний входной вывод (TIx).
* Внешняя частота, режим 2: внешний триггер-вход (ETR), доступный только в TIM2, TIM3 и TIM4.
* Внутренние триггер-входы (ITRx): используя один таймер как предделитель для другого таймера, для примера, вы можете настроить Timer1 для работы в качестве предделителя для Timer2.

**Внутренний тактовый источник (CK\_INT).**

Если контроллер режима ведомого отключен (SMS=000 в регистре TIMx\_SMCR), то биты CEN, DIR и UG являются актуальными управляющими битами и могут быть изменены только программой (за исключением бита UG, который остается очищенным автоматически). Вскоре после того, как бит CEN установлен в 1, предделитель тактируется внутренней частотой CK\_INT.

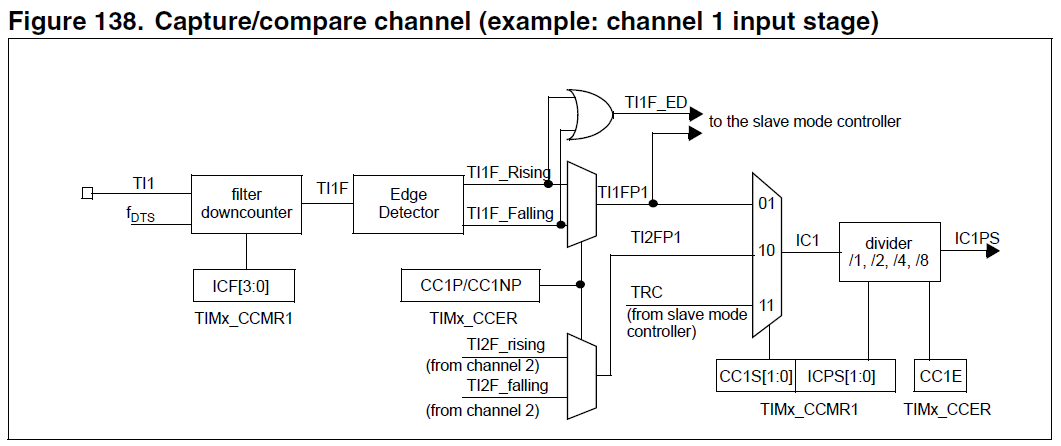
Рисунок 133 показывает поведение схемы управления и суммирующего счетчика в нормальном режиме, без предделителя.



**Каналы захвата/сравнения.**

Каждый канал захвата/сравнения выстроен вокруг регистра захвата/сравнения (включая теневой регистр), входной части для захвата (с цифровым фильтром, мультиплексированием и предделителем) и выходной частью (с компаратором и управлением выхода). Следующий рисунок дает представление об одном канале захвата сравнения.

Входная часть захватывает соответствующий вход TIx для генерации отфильтрованного сигнала TIxF. Затем, детектор фронта с выбранной полярностью генерирует сигнал (TIxFPx), который может попасть на предделитель, прежде чем он попадет в регистр захвата (ICxPS).



[дальше писать пока не буду, это мне пока не нужно]

**Режим выхода сравнения.**

Эта функция используется для управления выходным сигналом, или индикации пробега определенного периода времени. Когда находится совпадение между регистром захвата/сравнения, функция выхода сравнения:

* Задает на соответствующем выходном выводе программируемое значение, определенное режимом выхода сравнения (битами OCxM в регистре TIMx\_CCMRx) и выходную полярность (бит CCxP в регистре TIMx\_CCER). Выходной вывод может оставить свой уровень (без изменения, OCXM=000), установиться в активный уровень (OCxM=001), установиться в неактивный уровень (OCxM=010) или может переключиться (OCxM=011) при совпадении).
* Устанавливает флаг в регистре статуса прерывания (бит CCxIF в регистре TIMx\_SR).
* Генерирует прерывание, если соответствующая маска прерывания установлена (CCXIE бит в регистре TIMx\_DIER).
* Отсылает DMA запрос, если соответствующий бит разрешения установлен (бит CCxDE в регистре TIMx\_DIER, CCDS бит в TIMx\_CR2 регистре, для выбора запроса DMA).

Регистры TIMx\_CRRx могут быть запрограммированы с или без регистров предзагрузки, используя бит OCxPE в регистре TIMx\_CCMRx. Если этот бит сброшен, регистр предзагрузки в TIMx\_CCR1 отключен. TIMx\_CCR1 может быть записан в любое время, новое значение сразу принимается в расчет. Если этот бит установлен, регистр предзагрузки в TIMx\_CCR1 активен. Операции чтения/записи выполняются через него. Значение регистра предзагрузки загружается в активный регистр каждый раз, когда происходит событие обновления.

В режиме выхода сравнения, событие обновления UEV не производит никаких действий на ocxref и OCx выход. Дискретность по времени – один счет счетчика. Режим выхода сравнения может также быть использован для вывода одиночного испульса (в режиме один-импульс).

**Процедура.**

1. Выберете тактовый источник счетчика (внутренний, внешний, предделитель).
2. Запишите желаемые данные в регистры TIMx\_ARR и TIMx\_CCRx.
3. Установите CCxIE и/или CCxDE биты, если необходимо сгенерировать прерывание или запрос DMA.
4. Выберете режим выхода. Например, вы должны записать OCxM=11, OCxPE=0, CCxP=0 и CCxE=1 для переключения OCx выходного вывода, когда CNT совпадет с CCRx, CCRx предзагрузка не используется, OCx разрешен и активный уровень – высокий.
5. Разрешить счетчик установкой CEN бита в регистре TIMx\_CR1.

Регистр TIMx\_CCRx может быть обновлен в любое время программой для управления выходным сигналом (формой волны), при условии, что регистр предзагрузки не включен (OCxPE=0, иначе TIMx\_CCRx теневой регистр обновится только на следующем событии обновления UEV). На рисунке 142 приведен пример.

