

9 ТИПОВЫЕ СТРУКТУРНО-АРХИТЕКТУРНЫЕ РЕШЕНИЯ И ОСНОВЫ ПРИМЕНЕНИЯ ТАЙМЕРОВ ОБЩЕГО НАЗНАЧЕНИЯ

9.1 Общие вопросы.....	2
9.2 Принцип формирования задержек между двумя событиями посредством таймеров МК	3
9.3 Формирование сигналов с программно-задаваемыми частотно-временными параметрами посредством таймеров МК	5
9.3.1. Основные классы сигналов, формируемых посредством таймеров МК. Общий принцип формирования.....	5
9.3.2. Общий принцип ШИМ и основы ее применения. ШИМ-ЦАП. Калибровка источников ШИМ-сигналов	10
9.3.3. Основные режимы и принципы реализации программно-управляемой ШИМ на базе таймеров МК	19
9.3.4. Формирование ЧИМ-сигналов посредством таймеров МК	32
9.3.5. Формирование ФИМ-сигналов посредством таймеров МК.....	38
9.3.6. Реализация режима одновибратора посредством таймеров МК .	39
9.4 Преобразование частотно-временных параметров сигналов в цифровой код посредством таймеров МК.....	41
9.4.1. Общие вопросы преобразования частотно-временных параметров сигналов в цифровой код	41
9.4.2. Преобразование частоты сигналов в цифровой код посредством таймеров МК.....	46
9.4.3. Преобразование интервалов времени в цифровой код посредством таймеров МК.....	56
9.5 Типовые структурно-архитектурные решения таймеров МК общего назначения	62
9.5.1. Базовые структурно-архитектурные решения таймеров МК семейства <i>AVR</i>	63
9.5.2. Примеры программирования таймеров МК семейства <i>AVR</i>	78
9.5.3. Базовые структурно-архитектурные решения таймеров МК семейства <i>ARM Cortex-Mx</i>	96
9.5.4. Примеры программирования таймеров МК семейства <i>ARM Cortex-Mx</i>	126
9.6 Выводы по разделу 9.....	155

9.1 Общие вопросы

Как указано во введении, основной областью применения МК общего назначения являются средства контроля и управления техническими объектами. При их реализации часто возникает необходимость решения следующих задач:

- формирования программно задаваемых задержек (интервалов времени) между какими-либо событиями;
- генерации сигналов управления ИУ с программно задаваемыми частотно-временными параметрами (периодом или / и длительностью) данных сигналов.

Несколько реже, чем вышеперечисленные, встречаются также задачи преобразования в цифровой код частоты, длительности или периода выходных сигналов датчиков или каких-либо других сигналов, отражающих состояние объекта контроля и управления.

Все перечисленные группы задач могут быть объединены под общим названием **частотно-временных функций**. Для их реализации в составе практически всех моделей всех семейств современных МК общего назначения имеются специальные функциональные блоки, называемые **таймерами** (от английского *time* – время). Они разделяются на две группы:

- таймеры общего назначения;
- специализированные таймеры, основными разновидностями которых являются сторожевые таймеры, системные таймеры и счетчики реального времени.

Настоящий раздел посвящен структурно-архитектурным решениям таймеров общего назначения (в дальнейшем именуемых просто **таймерами**). Типовая структура и архитектура сторожевых таймеров рассмотрены в подразделе 5.4, системного таймера – в подпункте 7.3.2.5, счетчика реального времени – в пункте 3.3.3.

Нетрудно увидеть, что реализация частотно-временных функций первых двух из перечисленных групп фактически сводится к подсчету импульсов с известной частотой до заданного значения, по достижении которого происходит определенное событие, например, генерация запроса на прерывание или переключение некоторого вывода ПВВ МК. Преобразование частоты сигнала в

цифровой код реализуется подсчетом числа периодов данного сигнала за фиксированный интервал времени (формируемый, в свою очередь, способом подсчета импульсов с известной частотой, см. выше). Преобразование в код длительности интервала времени (периода или длительности сигнала) осуществляется подсчетом импульсов с фиксированной частотой за временной интервал, длительность которого подлежит преобразованию. Таким образом, центральным функциональным узлом таймера является двоичный счетчик; поэтому в технической документации ряда семейств МК используется термин «таймер / счетчик» (*Timer / Counter*), см., например, [8]. Как правило:

- счетчик таймера может работать в режиме как прямого, так и обратного счета (подробнее – см. подраздел 9.2);
- возможна предварительная загрузка в счетчик некоторого числа под управлением ПО;
- возможен программный выбор источника тактирования счетчика; в его качестве может служить как ГТИ МК, так и некоторый внешний по отношению к МК источник;
- возможен выбор частоты счетных импульсов посредством программно-управляемого предделителя (*Prescaler*) частоты выходного сигнала выбранного источника тактирования.

Рассмотрим общие принципы реализации вышеперечисленных частотно-временных функций посредством таймеров общего назначения.

9.2 Принцип формирования задержек между двумя событиями посредством таймеров МК

Ниже представлен обобщенный алгоритм формирования, посредством таймера МК, программно-задаваемой задержки между двумя событиями. Предполагается, что счетчик таймера изначально остановлен.

1. Выбрать источник тактирования счетчика и установить коэффициент деления предделителя частоты выходного сигнала источника, таким образом, чтобы обеспечивалось условие $2^{N_c} \times K_d / f_{CLK} > \Delta t$, где N_c – разрядность счетчика таймера, K_d – коэффициент деления предделителя, f_{CLK} – частота выходного

сигнала источника тактирования, Δt - требуемая длительность интервала задержки

2. По 1-му из событий (например, по включении 1-го реле) произвести загрузку в счетчик числа, равного:

$$N = N_{TOP} - \Delta t \times f_{CNT}; \quad (9.1)$$

если предполагается работа счетчика в режиме прямого счета (*Up-counting*), т. е. инкремента содержимого с поступлением каждого тактового импульса, или:

$$N = \Delta t \times f_{CNT}; \quad (9.2)$$

если счетчик предполагается использовать в режиме обратного счета (*Down-counting*), т. е. декремента содержимого с каждым очередным тактовым импульсом). В выражениях (9.1) и (9.2): N_{TOP} – верхний предел счета; f_{CNT} – частота сигнала тактирования счетчика (т. е. счетных импульсов).

3. После загрузки числа N в счетчик произвести его запуск (например, в таймерах МК семейства *ARM Cortex-Mx* – установкой в единицу бита разрешения счета в регистре управления таймером).

4. Переход счетчика от верхнего предела счета (N_{TOP}) к нижнему при его работе в режиме суммирования (переполнение, *Overflow*), или переход от нижнего предела (как правило, нулевого) к верхнему при работе в режиме вычитания (потеря значимости, *Underflow*) является признаком окончания формируемого интервала времени. По переполнении или потере значимости:

- устанавливается бит признака соответствующего события в регистре статуса таймера;

- вырабатывается запрос на прерывание по соответствующему событию (если данное прерывание разрешено).

При этом, например, в таймерах МК семейства *ARM Cortex-Mx* как при переполнении, так и при потере значимости в регистре статуса устанавливается один и тот же бит признака (*Update interrupt flag, UIF*), и вырабатывается запрос на прерывание с одним и тем же вектором.

5. По прерыванию (при его разрешении) или по выявленной путем программного опроса установке в единичное состояние

признака переполнения / потери значимости инициировать второе событие (например, включение 2-го реле), задержка которого относительно 1-го события должна быть равна Δt . После этого счетчик, в принципе, может быть остановлен.

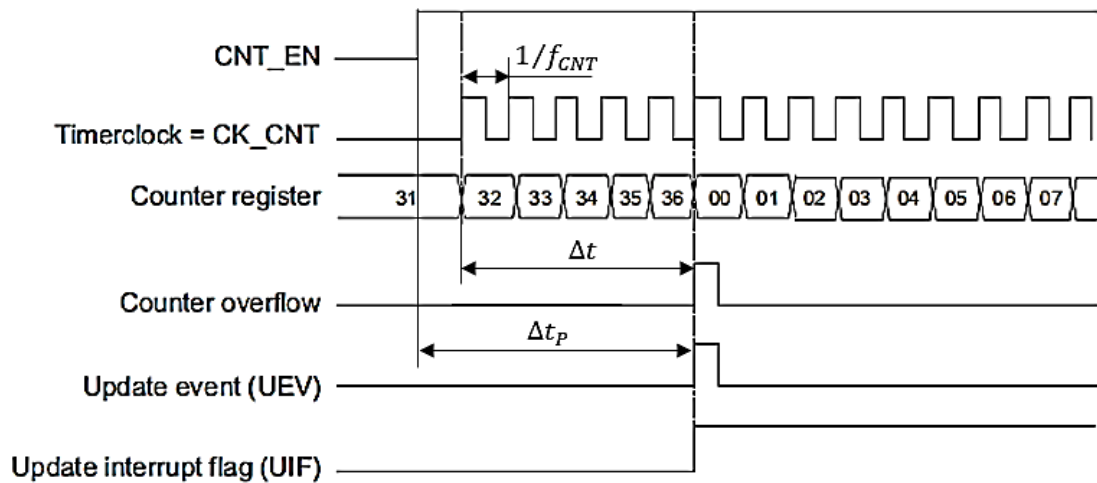
Временные диаграммы, поясняющие процедуру формирования задержки посредством таймера МК (на примере МК модельного ряда *STM32F030xx*) представлены на рис. 9.1. В обоих приведенных на рис. 9.1 примерах $N_{TOP} = 36$, $\Delta t = 5/f_{CNT}$. Предполагается также, что время реакции ЦП на переполнение / потерю значимости пренебрежимо мало по сравнению с периодом сигнала тактирования счетчика, равным $1/f_{CNT}$.

Примечание. По рис. 9.1 нетрудно заметить, что при формировании задержки по вышеприведенному алгоритму ее реальное значение, в общем случае, отличается от требуемого в большую сторону. Максимальное отклонение при этом равно $1/f_{CNT}$. В большинстве задач управления ИУ оно не существенно, т. к. $\Delta t \gg 1/f_{CNT}$. Однако, например, при формировании интервала счета для преобразования частоты в код (см. пункт 9.4.2) данное отклонение может быть критично. В подобных случаях может быть применен способ формирования интервала времени, описанный в подпункте 9.4.2.6, отличающийся более высокой точностью.

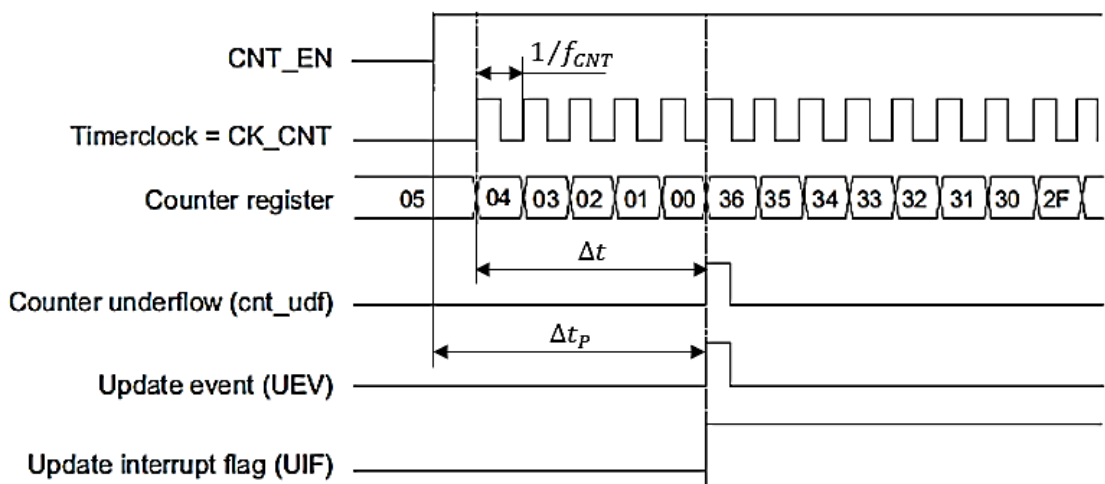
9.3 Формирование сигналов с программно-задаваемыми частотно-временными параметрами посредством таймеров МК

9.3.1. Основные классы сигналов, формируемых посредством таймеров МК. Общий принцип формирования

Архитектура таймеров практически всех семейств современных МК классов «*mainstream*» и «*high performance*» предоставляет возможность формирования, по крайней мере, 2-х классов сигналов с программно-задаваемыми частотно-временными параметрами:



a)



б)

CNT_EN – бит разрешения счета

$Timerclock = CK_CNT$ – сигнал тактирования счетчика

$Counter\ register$ – содержимое счетчика

$Counter\ overflow$ – переполнение счетчика

$Counter\ underflow$ – потеря значимости счетчиком

$Update\ event\ (UEV)$ – событие «Обновление таймера»

$Update\ interrupt\ flag$ – признак запроса прерывания по событию UEV

f_{CNT} – частота сигнала тактирования счетчика

Δt и Δt_P – требуемая и реальная длительность интервала задержки

Рис. 9.1. Временные диаграммы, поясняющие процедуру формирования задержки таймером МК модельного ряда *STM32F030xx* [15] в режимах прямого (а) и обратного счета (б) (см. также пояснения в тексте)

- широтно-модулированных импульсных сигналов (**ШИМ-сигналов**, *Pulse-Width Modulated signals, PWM-signals*),

характеризуемых фиксированной, однократно задаваемой при конфигурировании таймера частотой, и переменной длительностью, программно-управляемой в процессе работы МК;

- импульсных сигналов, **модулированных по частоте (ЧИМ-сигналов)**, характеризующихся фиксированной скважностью (отношением периода импульса к его длительности) и переменной частотой, программно-управляемой в процессе работы МК.

Архитектура таймеров ряда семейств МК (в частности, *ARM Cortex-Mx*) позволяет также:

- формировать импульсные сигналы, **модулированные по фазе (ФИМ-сигналы)**, с фиксированной частотой и скважностью и программно-управляемой начальной фазой;

- реализовывать режим **одновибратора (*One-pulse mode, OPM*)** [13 – 15], т. е. генерировать, по некоторому старт-событию, одиночный импульс с программно-управляемыми длительностью и задержкой относительно момента наступления старт-события.

Общий принцип формирования всех вышеперечисленных классов сигналов поясняет рис. 9.2. При их генерации счетчик таймера работает непрерывно, в одном из следующих режимов (в зависимости от класса и разновидности формируемого сигнала, см. пункты 9.3.3 – 9.3.5):

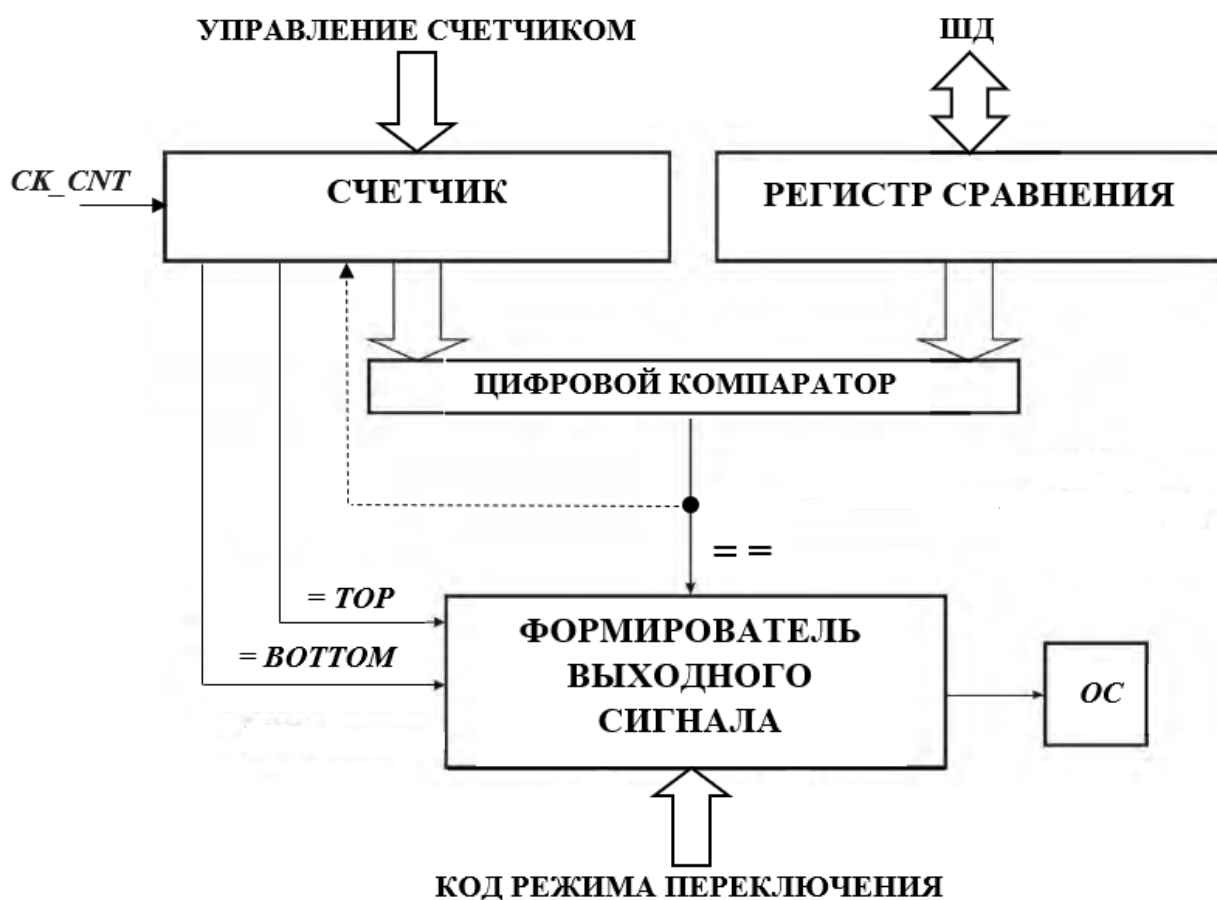
- прямого счета, при котором его содержимое увеличивается на единицу при поступлении каждого синхроимпульса; по достижении верхнего предела счета (*TOP*) происходит сброс содержимого счетчика до значения, равного нижнему пределу счета (*BOTTOM*), и возобновление счета;

- обратного счета, при котором его содержимое уменьшается на единицу при поступлении каждого синхроимпульса; по достижении нижнего предела счета в счетчик загружается число, равное верхнему пределу, и происходит возобновление счета;

- реверсивного счета, с рабочим циклом, состоящим из 2-х интервалов; в 1-ом из них осуществляется прямой счет, от нижнего до верхнего предела, во 2-м – обратный счет, от верхнего до нижнего предела, и т. д.

При этом:

- в таймерах большинства семейств МК нижний предел счета фиксирован и равен нулю;



CK_CNT – сигнал тактирования счетчика

ШД – шина данных

ОС – выход формируемого сигнала

Рис. 9.2. Обобщенная структурная схема блока формирования сигналов с программно-управляемыми частотно-временными параметрами
(см. пояснения в тексте)

- способы задания верхнего предела счета отличаются у различных семейств МК; в таймерах МК семейства *AVR*, в зависимости от класса формируемого сигнала и режима его генерации, верхний предел может быть или фиксированным, или равным содержимому регистра сравнения [6, 8] (см. пунктирную линию на рис. 9.2, а также рис. 9.17); в таймерах семейства *ARM Cortex-Mx* верхний предел счета задается программно-доступным регистром *ARR* [13 – 15] (см. рис. 9.19);

- режим обратного счета не поддерживается таймерами ряда семейств МК (в частности, семейства *AVR*).

Обязательными функциональными узлами таймера МК, реализующего формирование сигналов с программно-задаваемыми частотно-временными параметрами, являются программно-доступный **регистр сравнения** и **цифровой компаратор** (см. рис. 9.2). В регистр сравнения записывается число, задающее какой-либо из параметров генерируемого сигнала (например, длительность импульса ШИМ-сигнала или частоту ЧИМ-сигнала). В процессе работы счетчика его содержимое непрерывно сравнивается с содержимым регистра сравнения посредством цифрового компаратора. При достижении их равенства на выходе компаратора устанавливается активный уровень, по которому происходит переключение формируемого сигнала из некоторого состояния в некоторое другое состояние. Направление переключения зависит от класса формируемого сигнала и режима генерации. Например, при работе в режиме генерации не инвертированного асимметричного ШИМ-сигнала (см. рис. 9.7) по достижении равенства содержимых счетчика и регистра сравнения происходит переключение формируемого сигнала из единичного состояния в нулевое, а при генерации инвертированного асимметричного ШИМ-сигнала – из нулевого в единичное.

Возврат формируемого сигнала в состояние, из которого он переключился по срабатыванию компаратора, происходит, в зависимости от режима генерации сигнала, по достижении счетчиком нижнего или верхнего предела счета или по следующему срабатыванию цифрового компаратора (см. пункты 9.3.3 – 9.3.5). Например, при формировании не инвертированного асимметричного ШИМ-сигнала его переключение из нулевого состояния в единичное происходит по достижении счетчиком нижнего предела счета (см. рис. 9.7).

Переключение генерируемого сигнала в соответствующих направлениях осуществляется его формирователем, который управляется кодом, задающим режим переключения и располагаемым в одном из регистров таймера, а также выходным сигналом компаратора и сигналами « = *TOP*» и « = *BOTTOM*»,

устанавливаемыми в активное состояние по достижении счетчиком соответственно верхнего и нижнего пределов счета.

Отметим также, что при **жестких требованиях** к точности частотно-временных параметров формируемого сигнала, в качестве источника сигнала тактирования счетчика необходимо выбирать генератор с ПЭР (см. пункт 4.2.3).

Перейдем далее к рассмотрению принципов формирования и применения основных классов сигналов, генерируемых посредством таймеров МК.

9.3.2. Общий принцип ШИМ и основы ее применения. ШИМ-ЦАП. Калибровка источников ШИМ-сигналов

9.3.2.1. Широтно-импульсная модуляция (ШИМ) несколько более широко используется на практике, чем частотно-импульсная. Основная область ее применения – формирование аналоговых сигналов управления ИУ с программно-задаваемым средним значением (фактически – цифро-аналоговое преобразование управляющих сигналов ИУ).

Как указано ранее, принцип ШИМ состоит в формировании импульсного сигнала с фиксированным периодом и с длительностью, управляемой некоторой модулирующей величиной; применительно к ШИМ-сигналам таймеров МК – кодом, задающим их среднее значение.

Формируемые таймерами МК ШИМ-сигналы представляют собой прямоугольные (двухуровневые) импульсы напряжения с программно-управляемой длительностью. На рис. 9.3 приведены примеры временных диаграмм таких напряжений, при различных отношениях длительности к периоду.

Нетрудно увидеть, что среднее по времени значение прямоугольного ШИМ-напряжения описывается следующим выражением:

$$\overline{U_{PWM}} = \frac{(U_H - U_L)\tau}{T} + U_L; \quad (9.3)$$

где U_L и U_H – соответственно «низкий» и «высокий» уровни ШИМ-напряжения; τ и T – соответственно длительность и период

импульсов. Таким образом, среднее значение ШИМ-напряжения линейно зависит от длительности импульса, с коэффициентом, равным $(U_H - U_L)/T$. При фиксированных значениях U_L , U_H и T (что имеет место на практике) среднее значение ШИМ-напряжения однозначно задается длительностью импульса. Можно также сказать, что при фиксированных U_L и U_H значение $\overline{U_{PWM}}$ однозначно определяется отношением τ/T , называемым коэффициентом заполнения (*Duty cycle*), и обратно пропорциональным скважности ШИМ-сигнала.

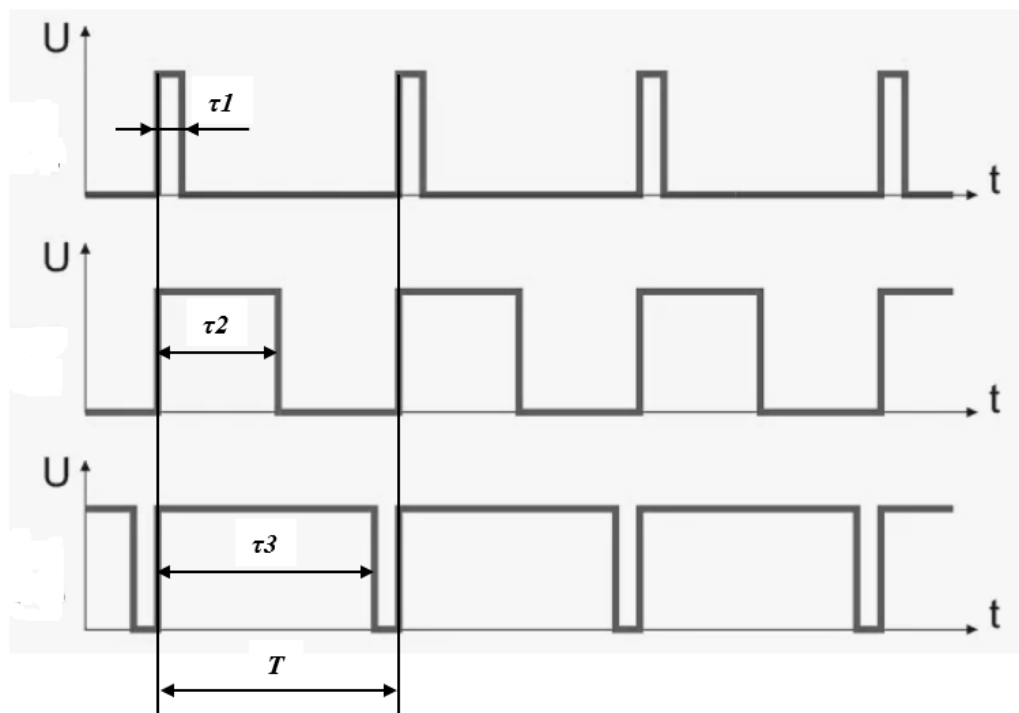


Рис. 9.3. Примеры временных диаграмм двухуровневых ШИМ-напряжений ($\tau_1 / T = 0,1$; $\tau_2 / T = 0,5$; $\tau_3 / T = 0,9$)

Вышесказанное справедливо и для случая, когда двухуровневый ШИМ-сигнал является током, а не напряжением (например, током в цепи, коммутируемой силовым переключателем, управляемым ШИМ-сигналом МК). При этом:

$$\overline{I_{PWM}} = \frac{(I_H - I_L)\tau}{T} + I_L. \quad (9.4)$$

9.3.2.2. Для выделения среднего значения (т. е., фактически, постоянной составляющей) ШИМ-сигнала он, при его подаче на ИУ, должен быть подвергнут фильтрации посредством ФНЧ. Как

правило, для фильтрации ШИМ-сигналов на практике применяются ФНЧ **1-го порядка**. В ряде случаев функции ФНЧ выполняет непосредственно ИУ (см. рис. 9.5 и пояснения к нему).

Базовым условием выбора параметров ФНЧ является обеспечение размаха остаточной переменной составляющей (пульсации) выходного напряжения ФНЧ, не превышающего максимального значения, допустимого в соответствии с требованиями к конкретному устройству. Как правило, задается максимально допустимое **относительное** значение пульсации, т. е. отношение ее размаха к постоянной составляющей ШИМ-сигнала. Аналитические выражения для выбора параметров ФНЧ достаточно сложны, поэтому ограничимся представленными в табл. 9.1 экспериментально полученными относительными значениями пульсации, обеспечиваемыми ФНЧ 1-го порядка при различных типовых значениях частоты среза и коэффициента заполнения ШИМ-сигнала.

Таблица 9.1

Зависимости относительных значений пульсаций на выходе ФНЧ 1-го порядка от частоты среза ФНЧ при фильтрации ШИМ-сигналов с различными коэффициентами заполнения

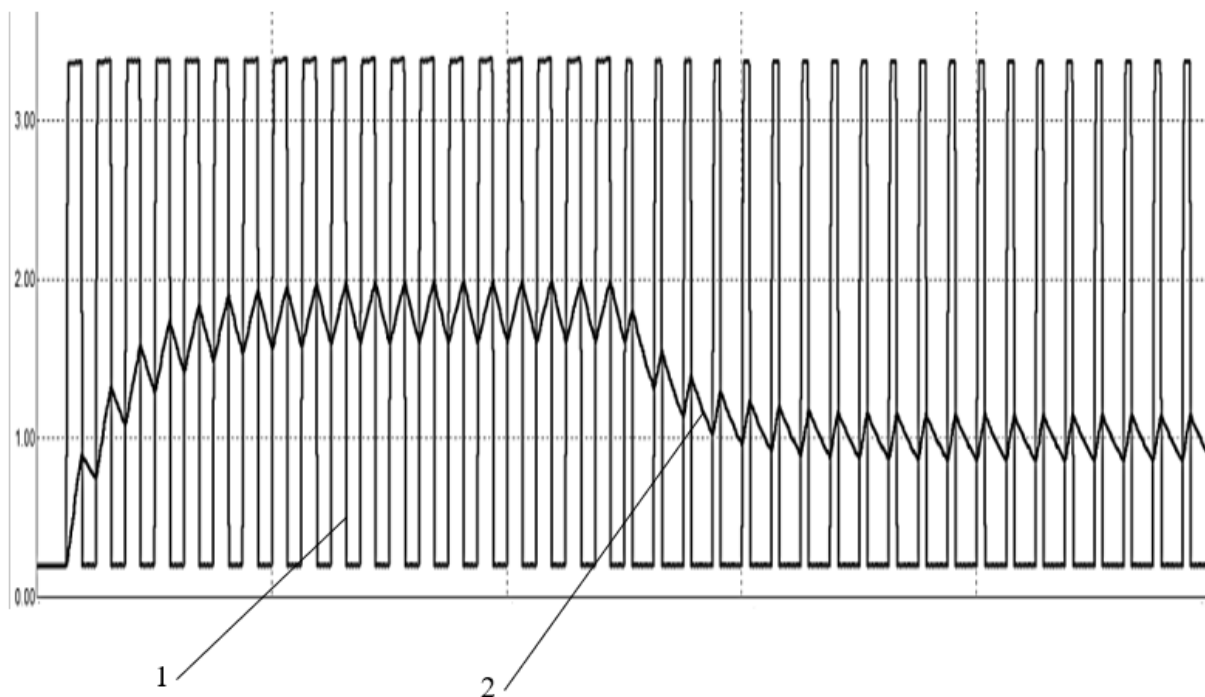
Коэффициент заполнения ШИМ-сигнала	Отношение размаха пульсации выходного сигнала ФНЧ к его постоянной составляющей при частоте среза ФНЧ, равной		
	0,0159 / T	0,0053 / T	0,0016 / T
2,5 %	0,096	0,033	0,010
5 %	0,094	0,031	0,009
10 %	0,087	0,029	0,009
25 %	0,073	0,024	0,007
50 %	0,049	0,016	0,005
75 %	0,024	0,008	0,003
90 %	0,009	0,003	0,001

Значения пульсации при частотах среза, **не указанных** в табл. 9.1, можно оценить на ее основе, исходя из того, что для каждого конкретного значения коэффициента заполнения относительное значение пульсации прямо пропорциональна частоте среза, при условии, что она много меньше значения $1 / T$. Например, при коэффициенте заполнения, равном 25 %, ФНЧ 1-го порядка с частотой среза, равной $0,01 / T$, обеспечит, в соответствии с табл. 9.1,

относительное значение пульсации, примерно равное $0,073 \times 0,01 / 0,0159$ или $0,024 \times 0,01 / 0,0053$, т. е. 0,045. В свою очередь, относительные значения пульсаций при не указанных в табл. 9.1 коэффициентах заполнения могут быть оценены методом интерполяции.

Примечание. При скачкообразном изменении постоянной составляющей фильтруемого сигнала, время установления выходного сигнала ФНЧ 1-го порядка до уровней 0,95; 0,99 и 0,999 от стационарного (установившегося) значения равно примерно $0,48/f_c$; $0,79/f_c$ и $1,11/f_c$ соответственно [20, 22].

На рис. 9.4 представлен пример временных диаграмм ШИМ-сигнала до и после фильтрации при скачкообразном изменении коэффициента заполнения с 50-ти до 25-ти процентов; в данном примере был задействован ФНЧ 1-го порядка с частотой среза, равной $0,08 / T$.



1, 2 – ШИМ-сигнал до и после фильтрации соответственно

Рис. 9.4. Пример временных диаграмм ШИМ-сигнала до и после фильтрации при скачкообразном изменении коэффициента заполнения (см. пояснения в тексте)

Максимально допустимый период (и, соответственно, минимально допустимая частота) ШИМ-сигнала при управлении ИУ

определяются сочетанием максимально допустимого уровня пульсаций ШИМ сигнала после фильтрации и максимально допустимого времени установления ФНЧ (которое должно быть существенно меньше минимальной длительности интервала времени, в течение которого параметры ШИМ-сигнала постоянны).

9.3.2.3. Минимально допустимый период (и, соответственно, максимально допустимая частота) ШИМ-сигнала при управлении ИУ, работающим в ключевом режиме, должны удовлетворять условию $DC_{min}T, (1 - DC_{max})T \gg t_{вкл/выкл}$ иу, где DC_{min} и DC_{max} – соответственно минимально и максимально возможное значение коэффициента заполнения, $t_{вкл/выкл}$ иу – длительность переходных процессов при включении / выключении ИУ.

9.3.2.4. Следует отметить, что ШИМ-управление не предъявляет жестких требований к точности частоты сигнала тактирования счетчика, поскольку при таком управлении важно отношение длительности импульса к его периоду, а не их абсолютные значения. В свою очередь, значения и длительности, и периода определяются частотой одних и тех же счетных импульсов (см. пункт 9.3.3), поэтому отклонения их частоты от номинальной (в пределах до нескольких процентов) не влияют на процесс ШИМ-управления.

9.3.2.5. На рис. 9.5 представлен один из примеров использования ШИМ – простейшая схема ШИМ-управления скоростью вращения маломощного (порядка единиц Ватт) коллекторного электродвигателя постоянного тока. Его скорость вращения, в первом приближении, может считаться прямо пропорциональной постоянной составляющей напряжения на его якоре, который подключен между выходом источника питания и коллектором транзистора $VT1$. Последний выполняет функцию ключа, осуществляющего включение / выключение напряжения на якоре электродвигателя. Транзистор управляется ШИМ-сигналом, подаваемым с порта МК. При этом напряжение на якоре также представляет собой ШИМ-сигнал, «высокий» и «низкий» уровни которого равны (с точностью до нескольких десятков милливольт) напряжению питания и нулю соответственно.

Функцию инерционного элемента (ФНЧ), выделяющего постоянную составляющую напряжения на якоре, выполняет сам

электродвигатель. Диод $VD1$ служит для защиты транзистора от выбросов напряжения, возникающих при его выключении из-за индуктивного характера цепи его нагрузки (см. подпункт 6.4.3.3). Конденсатор $C1$ предназначен для подавления импульсных помех, создаваемых электродвигателем.

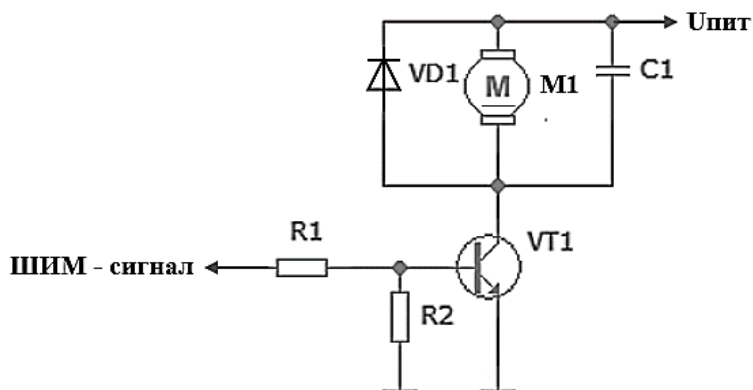


Рис. 9.5. Простейший вариант схемы ШИМ-управления скоростью вращения маломощного коллекторного электродвигателя постоянного тока (см. пояснения в тексте)

Разнообразные примеры применения МК для ШИМ-управления различными классами электродвигателей приведены в общедоступных Интернет-ресурсах. В частности, представляют интерес микроконтроллерные системы ШИМ-управления таким перспективным классом двигателей, как бесколлекторные электродвигатели постоянного тока (см., например, [49]).

9.3.2.5. Отметим, что ШИМ в сочетании с НЧ-фильтрацией может также применяться для формирования сигналов некоторой заданной формы (гармонических, треугольных, пилообразных и т. п.), с частотой, много меньшей частоты импульсов ШИМ-сигнала, т. е. $1 / T$. При этом НЧ-фильтр ШИМ-сигнала выполняет функцию сглаживания, а не выделения постоянной составляющей. Его частота среза должна быть много ниже, чем $1 / T$, но много выше частоты формируемого сигнала. Идеализированная временная диаграмма, поясняющая принцип ШИМ-формирования сигнала заданной формы (на примере косинусоиды со смещенным средним уровнем) представлена на рис. 9.6. На практике период импульсов ШИМ-сигнала должен быть минимум в несколько десятков раз меньше, чем период формируемого сигнала.

Более подробно вопросы ШИМ-формирования сигналов с заданными характеристиками и параметрами рассмотрены в **Примере 2** подпункта **9.5.2.2**.

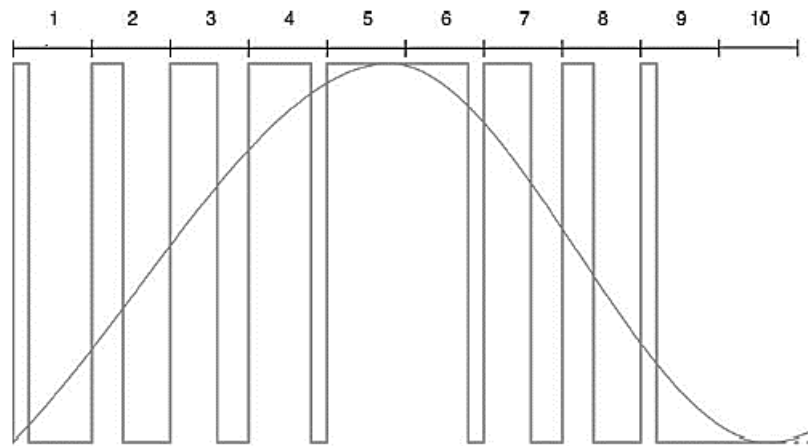


Рис. 9.6. Идеализированная временная диаграмма, поясняющая принцип формирования сигнала заданной формы способом ШИМ [9]

9.3.2.6. В большинстве типовых практических приложений ШИМ (в частности, в системах автоматического ШИМ-регулирования напряжения или тока в нагрузке) погрешности средних значений ШИМ-сигналов не критичны. Однако, например, при формировании способом ШИМ программно-задаваемых опорных напряжений или токов, а также при реализации ШИМ-ЦАП требования к погрешностям $\overline{U_{PWM}}$ или $\overline{I_{PWM}}$ могут быть достаточно серьезными (порядка десятых или сотых %). В таких случаях, как правило, необходима **калибровка** источника ШИМ-сигнала, т. е. коррекция его погрешностей до приемлемого уровня, определяемого конкретной решаемой задачей.

Как нетрудно заметить из выражений (9.3) и (9.4), основными источниками погрешности среднего значения ШИМ-напряжения или ШИМ-тока являются:

- погрешность значения U_L или, соответственно, I_L ;
- погрешность разности $U_H - U_L$ или, соответственно, $I_H - I_L$;
- погрешность отношения τ/T .

При условиях, что:

- таймер, управляющий ШИМ, тактируется выходным сигналом генератора на основе ПЭР (см. пункт 4.2.3);

- длительность переходных процессов при переключении ШИМ-сигнала на несколько порядков меньше его периода (что, как правило, имеет место в приложениях ШИМ, предъявляющих повышенные требования к точности значений $\overline{U_{PWM}}$ или $\overline{I_{PWM}}$); погрешность отношения τ/T может считаться пренебрежимо малой, и основными источниками погрешностей $\overline{U_{PWM}}$ или $\overline{I_{PWM}}$ являются первые два из перечисленных. Поэтому сосредоточимся на способах и алгоритмах минимизации погрешностей, обусловленных этими источниками.

Наиболее **очевидным** способом решения данной задачи является применение прецизионных источников напряжений U_H и U_L или, соответственно, токов I_H и I_L . Схемотехнические решения, реализующие этот подход, в принципе, известны [23, 38]. Они состоят в том, что ШИМ-напряжение или, соответственно, ШИМ-ток формируются поочередной подачей на выход его источника напряжений U_H и U_L или токов I_H и I_L , вырабатываемых прецизионными источниками постоянных напряжений или токов. При этом управление коммутацией напряжений / токов осуществляется ШИМ-сигналом, генерируемым таймером МК. Такой подход требует серьезных затрат электронных компонентов, внешних по отношению к МК, и поэтому он рационален для реализации только высокопрецизионных источников ШИМ-напряжения или тока.

Если в состав МК входит АЦП, возможен также **другой** подход, применимый при формировании **ШИМ-напряжений** непосредственно таймером, и состоящий в определении **реальных** значений U_H и U_L (т. е. выходных напряжений единицы и нуля вывода ПВВ, выполняющего функцию выхода таймера) на этапе калибровки, по результатам АЦ-преобразования $\overline{U_{PWM}}$ при фиксированных отношениях τ/T . В процессе работы, отношения τ/T , необходимые для формирования ШИМ-напряжений с требуемым $\overline{U_{PWM}}$, рассчитываются на основе полученных при калибровке реальных значений U_H и U_L . Данный подход характеризуется минимальными затратами внешних по отношению

к МК компонентов. Однако, при его использовании невозможно достижение остаточных погрешностей формирования ШИМ-напряжения, меньших погрешности АЦП.

Обобщенный алгоритм калибровки следующий.

1. Производится калибровка АЦП (см. пункт 10.3.9, а также подпункты 10.3.11.22 и 10.3.11.23).

2. Выполняется АЦ-преобразование $\overline{U_{PWM}}$ при 2-х значениях отношения τ/T (τ_1/T и τ_2/T), соответствующих начальной и конечной точкам рабочего диапазона $\overline{U_{PWM}}$ (например, при τ/T , равном 0,1 и 0,9). Для повышения точности калибровки, желательно выполнить несколько циклов АЦ-преобразования при каждом из значений τ/T , с усреднением полученных результатов преобразования.

3. В результате выполнения пункта 2 будет получена система из 2-х уравнений с 2-мя неизвестными, в качестве которых выступают значения $U_H - U_L$ и U_L (см. также выражение (9.3)):

$$\begin{cases} \overline{U_{PWM1}} = \frac{(U_H - U_L)\tau_1}{T} + U_L; \\ \overline{U_{PWM2}} = \frac{(U_H - U_L)\tau_2}{T} + U_L. \end{cases}$$

Решая данную систему, получаем:

$$U_H - U_L = \frac{(\overline{U_{PWM2}} - \overline{U_{PWM1}})}{\tau_2 - \tau_1} T;$$

$$U_L = \overline{U_{PWM2}} - \frac{(U_H - U_L)\tau_2}{T} = \overline{U_{PWM1}} - \frac{(U_H - U_L)\tau_1}{T}.$$

Для повышения точности определения U_L его желательно вычислять как среднеарифметическое 2-х данных значений, по следующему выражению:

$$U_L = \frac{1}{2} \left(\left(\overline{U_{PWM2}} - \frac{(U_H - U_L)\tau_2}{T} \right) + \left(\overline{U_{PWM1}} - \frac{(U_H - U_L)\tau_1}{T} \right) \right).$$

Калибровку следует проводить, как минимум, при каждом включении питания МК, а также после каждого сброса МК и после его выхода из энергосберегающего режима.

Полученные в результате калибровки реальные значения U_L и $U_H - U_L$ используются в процессе работы ШИМ-источника напряжения или ШИМ-ЦАП для вычисления отношения τ/T , необходимого для формирования ШИМ-напряжений с требуемым $\overline{U_{PWM}}$. Расчет производится по выражению:

$$\frac{\tau}{T} = \frac{\overline{U_{PWM}} - U_L}{U_H - U_L}.$$

9.3.3. Основные режимы и принципы реализации программно-управляемой ШИМ на базе таймеров МК

9.3.3.1. Известны два основных режима генерации ШИМ-сигналов посредством таймеров МК [6, 9]:

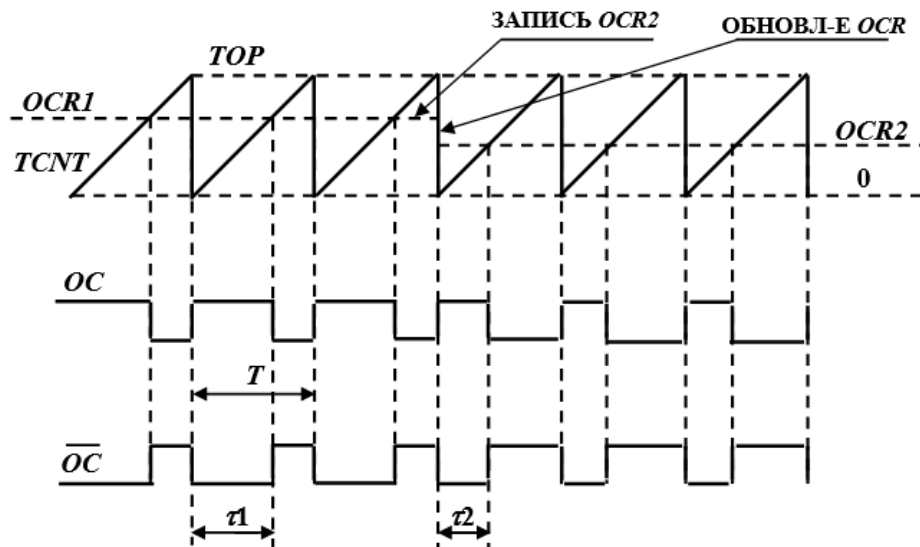
- асимметричная ШИМ, в документации на МК семейства *AVR* называемая «Быстрая ШИМ» (*Fast PWM*), а на МК семейства *ARM Cortex-Mx* – «ШИМ с выравниванием по фронту» (*Edge-aligned PWM*);

- центрированная ШИМ (*Center-aligned PWM*), в документации на МК семейства *AVR* – «ШИМ, корректная по фазе» (*Phase Correct PWM*).

9.3.3.2. Принцип реализации **асимметричной ШИМ** (на примере таймера МК семейства *AVR*) поясняют временные диаграммы, представленные на рис. 9.7 [6, 8].

При реализации асимметричной ШИМ счетчик таймера работает в режиме или прямого, или обратного счета (в МК семейства *AVR* – только прямого [6], см. рис. 9.7). Если счетчик работает в режиме прямого счета, ШИМ-сигнал устанавливается в единичное состояние по сбросу содержимого счетчика до значения, равного нижнему пределу счета (в таймерах семейств *AVR* и *ARM Cortex-Mx* – до нуля). Сброс ШИМ-сигнала в нулевое состояние при этом осуществляется по достижении равенства содержимых счетчика и регистра сравнения. Переключения инвертированного

ШИМ-сигнала происходят в направлениях, противоположных вышеуказанным (см. рис. 9.7). При асимметричной ШИМ с работой счетчика в режиме обратного счета (что реализуемо, например, в МК семейства *ARM Cortex-Mx*) ШИМ-сигнал устанавливается в единичное состояние по равенству содержимых счетчика и регистра сравнения, и сбрасывается в ноль при достижении нулевого значения содержимым счетчика.



$TCNT$ – содержимое счетчика (реально изменяется ступенчато, для упрощения рисунка показано изменяющимся непрерывно)

$OCR1, OCR2$ – содержимое регистра сравнения (OCR) на соответствующих интервалах времени

$OC, \overline{OC}$ – соответственно не инвертированный и инвертированный ШИМ-сигнал

T – период генерируемых импульсов

τ_1, τ_2 – длительность генерируемых импульсов на соответствующих интервалах времени

Рис. 9.7. Принцип реализации асимметричной ШИМ посредством таймера МК (на примере МК семейства *AVR*)

Регистры сравнения таймеров МК семейств *AVR* и *ARM Cortex-Mx* (как и большинства других семейств МК) **буферизированы**. В процессе ШИМ ПО может осуществлять запись в них новых значений в любые произвольные моменты времени. Однако при

выполнении команды записи регистра сравнения загружаемое в него число реально записывается в буферный регистр, а его загрузка в собственно регистр сравнения происходит в фиксированные моменты времени, в которые состояние генерируемого сигнала не зависит от содержимого данного регистра. Например, в режиме асимметричной ШИМ загрузка (обновление) регистра сравнения происходит в моменты перезапуска счетчика, как в МК семейства *AVR* (см. рис. 9.7), так и семейства *ARM Cortex-Mx*. Обновление регистра сравнения только в определенные, «не критичные» моменты времени устраняет генерацию импульсов с «не предсказуемой» длительностью в процессе программно-управляемой ШИМ. Следует отметить, что в таймерах семейства *ARM Cortex-Mx* может быть программно разрешена запись регистра сравнения без буферизации (см. подпункт 9.5.3.10) что, однако не рекомендуется [9].

На рис. 9.8 представлен фрагмент детализированной временной диаграммы формирования сигнала при асимметричной ШИМ и значениях *TOP* и *OCR*, равных 8-и и 4-м соответственно.

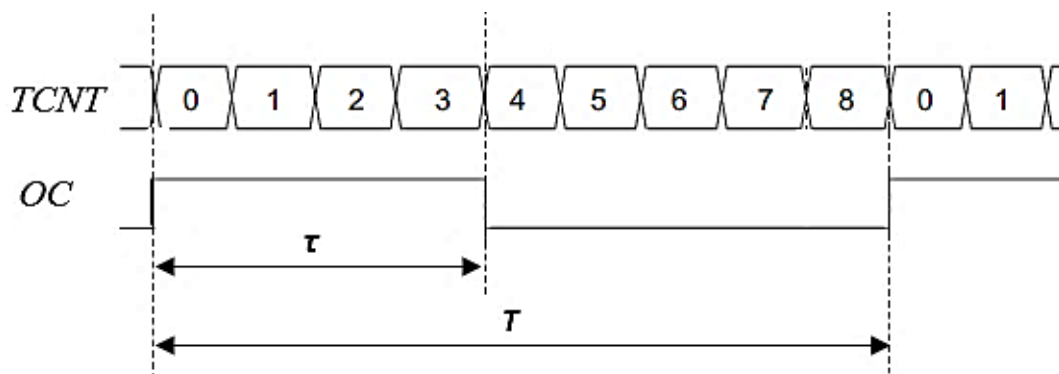


Рис. 9.8. Фрагмент детализированной временной диаграммы формирования сигнала при асимметричной ШИМ ($TOP = 8$, $OCR = 4$)

Из рис. 9.8 нетрудно увидеть, что в данном режиме ШИМ период генерируемых импульсов и их длительность описываются следующими выражениями:

$$T = \frac{TOP + 1}{f_{CNT}}; \quad (9.5)$$

$$\tau = \frac{OCR}{f_{CNT}}; \quad (9.6)$$

где f_{CNT} – частота сигнала тактирования счетчика (см. рис. 9.2).

Основным **достоинством** асимметричной ШИМ по сравнению с ШИМ с центрированием является в 2 раза **большая**, при прочих равных условиях, частота ШИМ-сигнала (откуда одно из ее названий – «**быстрая**» ШИМ), а, следовательно, возможность использования ФНЧ с 2 раза большей частотой среза и в 2 раза меньшим временем установления при тех же требованиях к остаточным пульсациям (см. подпункт 9.3.2.2). Основным **недостатком** асимметричной ШИМ является скачкообразное изменение начальной фазы ШИМ-сигнала при каждом изменении его длительности, что поясняет рис. 9.9. На нем продублированы временные диаграммы формирования ШИМ-сигнала (не инвертированного), приведенные на рис. 9.7, с указанием интервалов времени между серединами генерируемых импульсов.

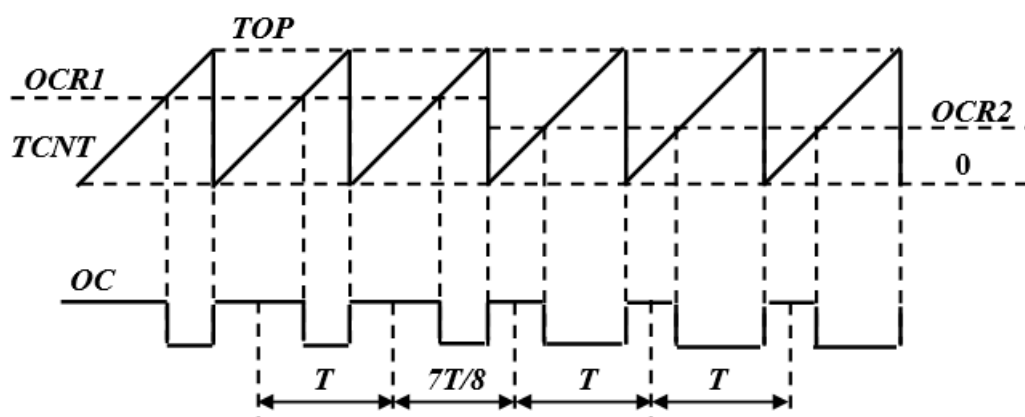


Рис. 9.9. Пример изменения начальной фазы сигнала при асимметричной ШИМ (см. пояснения в тексте)

Из рис.9.9 видно, что в стационарном режиме данные интервалы, естественно, равны периоду ШИМ-сигнала (T). Однако, при изменении (в процессе ШИМ) длительности импульсов с некоторого значения τ_1 на некоторое другое значение τ_2 , интервал времени между серединами последнего импульса с длительностью τ_1 и первого импульса с длительностью τ_2 будет отличаться от T на значение, в общем случае, равное $(\tau_2 - \tau_1)/2$; в приведенном на рис. 9.9 примере – на $T/8$. Таким образом, при каждом изменении

длительности импульсов в процессе асимметричной ШИМ происходит скачкообразное изменение начальной фазы ШИМ-сигнала на значение, равное $\pi \times (\tau_2 - \tau_1)/T$, что недопустимо, например, при решении ряда задач управления электроприводами [8].

От вышеописанного недостатка свободна центрированная ШИМ (ШИМ, **корректная по фазе**), см. подпункт 9.3.3.3.

9.3.3.3. Принцип реализации центрированной ШИМ (на примере таймера МК семейства *AVR*) поясняют временные диаграммы, представленные на рис. 9.10 [6, 8].

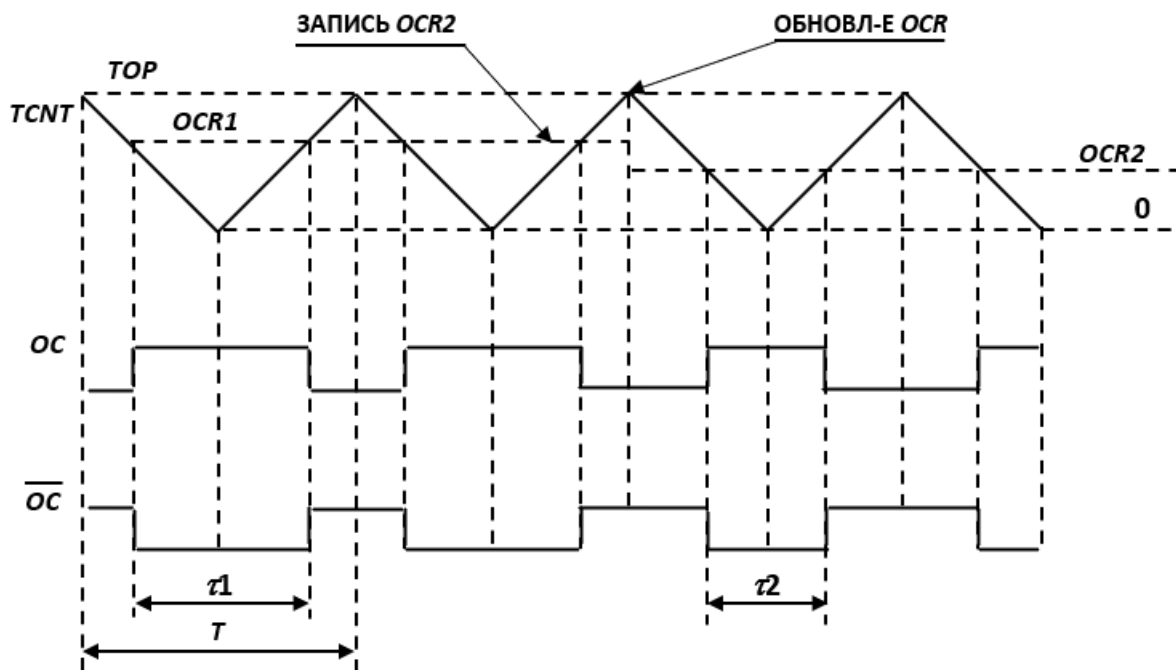


Рис. 9.10. Принцип реализации центрированной ШИМ посредством таймера МК (на примере МК семейства *AVR*) (условные обозначения соответствуют приведенным на рис. 9.7)

При центрированной ШИМ счетчик таймера работает в режиме реверсивного счета. Его рабочий цикл, как указано в пункте 9.3.1, при этом состоит из 2-х интервалов: прямого счета от нижнего до верхнего предела, и обратного счета от верхнего до нижнего предела (равного нулю у МК семейств *AVR* и *ARM Cortex-Mx*). Переключения ШИМ-сигнала из нуля в единицу происходят по достижении равенства содержимых счетчика и регистра сравнения на интервале обратного счета; переключение из единицы в ноль – по

аналогичному событию на интервале прямого счета (см. рис. 9.10). Инвертированный ШИМ-сигнал, естественно, переключается в противоположных направлениях.

Как и в режиме асимметричной ШИМ, программно-управляемая запись регистра сравнения может осуществляться в любой момент времени. Реальное обновление его содержимого, благодаря буферизации, происходит по достижении счетчиком верхнего предела счета (см. рис. 9.10), т. е. в моменты времени, в которые формирование импульса не происходит. Это предотвращает генерацию импульсов с «не предсказуемой» длительностью (см. также подпункт 9.3.3.2). Напомним, что в таймерах МК семейства *ARM Cortex-Mx*, в принципе, может быть программно разрешена запись регистров сравнения без буферизации (см. подпункт 9.5.3.10), что, однако, не рекомендуется [9].

Из рис. 9.10 нетрудно заметить, что середины (центры) импульсов ШИМ-сигнала при центрированной ШИМ всегда совпадают во времени с моментами достижения счетчиком нижнего предела счета, откуда происходит название данного режима ШИМ.

На рис. 9.11 представлен фрагмент детализированной временной диаграммы формирования сигнала центрированной ШИМ и значениях *TOP* и *OCR*, равных 4-м и 1-му соответственно.

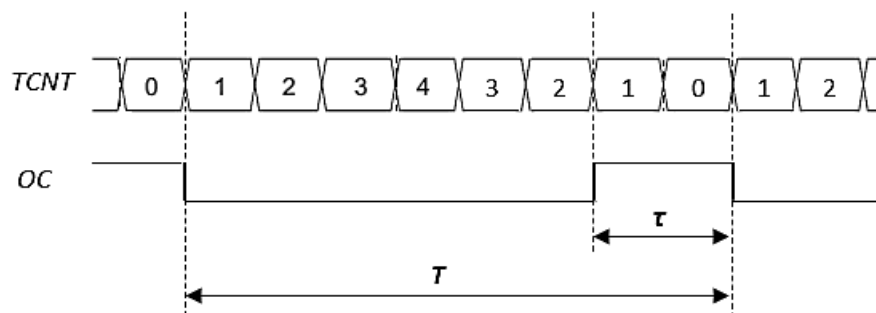


Рис. 9.11. Фрагмент детализированной временной диаграммы формирования сигнала центрированной ШИМ ($TOP = 4$, $OCR = 1$)

Из рис. 9.11 следует, что в данном режиме ШИМ период генерируемых импульсов и их длительность вычисляются по следующим выражениям:

$$T = \frac{2 \times TOP}{f_{CNT}}; \quad (9.7)$$

$$\tau = \frac{2 \times OCR}{f_{CNT}}. \quad (9.8)$$

Основным **достоинством** центрированной ШИМ по сравнению с асимметричной является отсутствие изменений начальной фазы ШИМ-сигнала при модуляции длительности импульсов (откуда другое название центрированной ШИМ – «ШИМ, корректная по фазе»). Неизменность начальной фазы обусловлена «привязкой» во времени середины импульса к моменту достижения счетчиком нижнего предела счета, а обновления регистра OCR – к моменту достижения верхнего предела, благодаря чему интервал времени между серединами импульсов не изменяется в процессе модуляции и остается равным периоду ШИМ-сигнала. Следовательно, остается неизменной и его начальная фаза. Данное свойство центрированной ШИМ поясняет рис. 9.12. Благодаря ему она предпочтительнее асимметричной при решении ряда задач управления электроприводами [8].

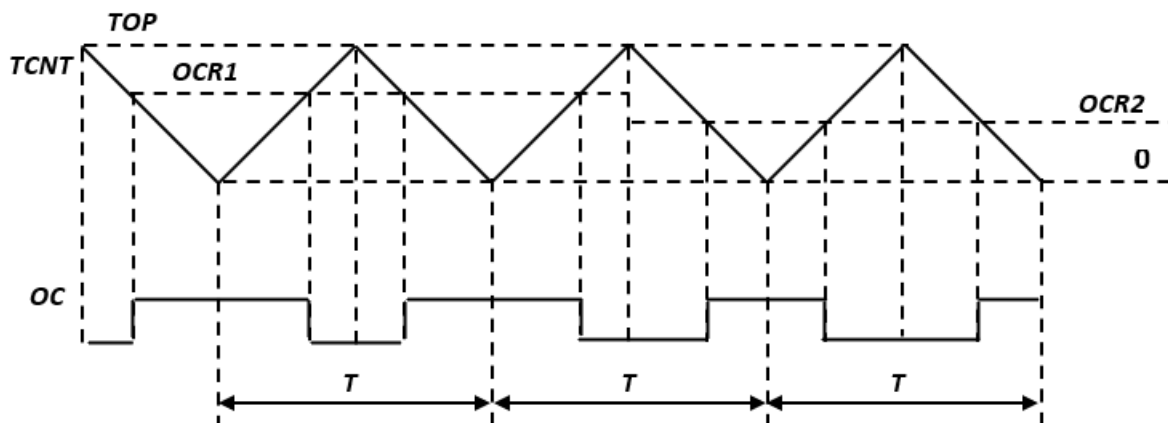


Рис. 9.12. Пояснение эффекта отсутствия изменения начальной фазы сигнала при центрированной ШИМ

Основным **недостатком** центрированной ШИМ по сравнению с асимметричной является в 2 раза меньшая (при прочих равных условиях) частота ШИМ-сигнала и, следовательно, необходимость применения ФНЧ с 2 раза меньшей частотой среза и в 2 раза большим временем установления при тех же требованиях к остаточным пульсациям (см. подпункт 9.3.2.2).

9.3.3.4. Практически все ранее изложенное в подпунктах 9.3.3.2 и 9.3.3.3 в равной мере относится и к реализации ШИМ посредством таймеров МК семейства *ARM Cortex-Mx*, с учетом того, что в таймерах данного семейства верхний предел счета равен содержимому регистра *ARR (Auto Reload Register)*, а в качестве регистра сравнения служит *CCR (Capture / Compare Register)* [13 – 15]. Также см. приведенные в подпункте 9.5.3.10 некоторые особенности реализации ШИМ, корректной по фазе, в МК семейства *ARM Cortex-Mx*.

9.3.3.5. Необходимо также остановиться на таком вопросе, как формирование двух **противофазных** ШИМ-сигналов, с задержкой между переходом одного из них в пассивное состояние и другого – в активное (см. рис. 9.13). В технической литературе данная задержка часто называется «мертвым временем» (*dead time*).

Типовым примером применения противофазных ШИМ-сигналов с задержкой между переключениями являются сигналы управления импульсными преобразователями напряжения, например, полумостовыми преобразователями. Функциональная схема простейшего преобразователя данного типа приведена на рис. 9.14 [50]. Противофазные ШИМ-сигналы с «мертвым временем» между переключениями используются для управления силовыми ключами на транзисторах *VT1* и *VT2*. В отсутствие задержки между переключениями управляющих сигналов, при каждом из переключений некоторое время один из ключей будет уже замкнут, а другой – еще не разомкнут. Это приведет к протеканию через них сквозного тока (на рис. 9.14 обозначен пунктирной линией), который неизбежно выведет из строя оба транзистора.

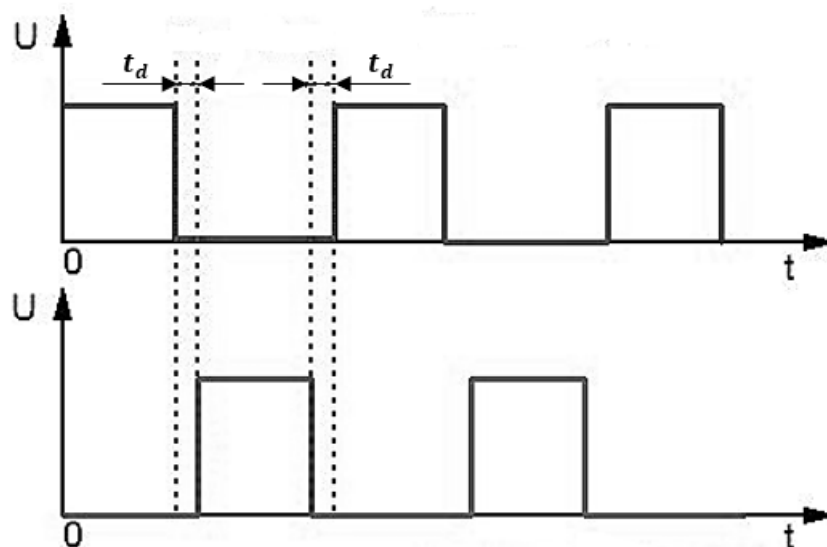
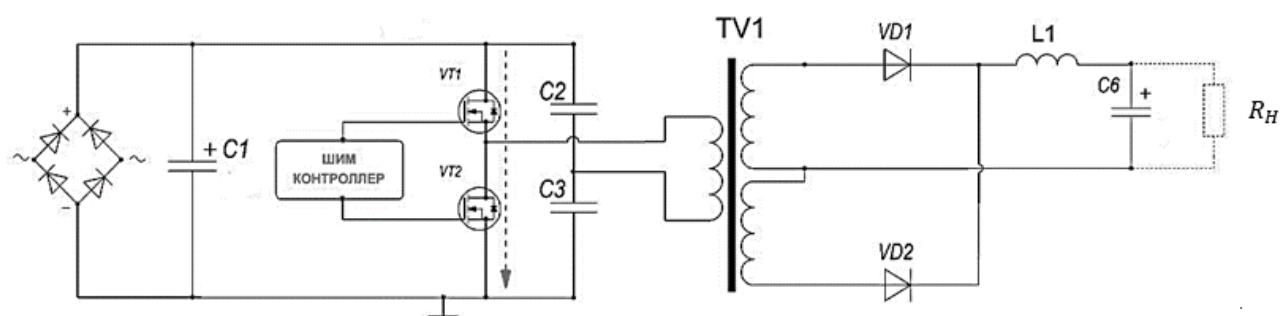


Рис. 9.13. Пример временных диаграмм противофазных импульсов с задержкой (t_d) между переключениями



R_H – сопротивление нагрузки

Рис. 9.14. Функциональная схема простейшего импульсного полумостового преобразователя [50]

В общем, предпочтительно применение **специализированных** ИС ШИМ-контроллеров, обеспечивающих формирование противофазных сигналов с задержкой между переключениями. Однако, на практике может возникнуть и необходимость формирования таких сигналов с помощью таймеров МК общего назначения. Возможность их генерации предоставляется архитектурой таймеров практически всех современных семейств МК классов «*mainstream*» и «*high performance*».

9.3.3.6. В составе МК семейства *ARM Cortex-Mx* имеется, как минимум, один таймер с «продвинутым» управлением (*Advanced-control timer*) [13 – 15], поддерживающий функцию «*dead-time*

insertion». Временные диаграммы, поясняющие данную функцию, приведены на рис. 9.15.

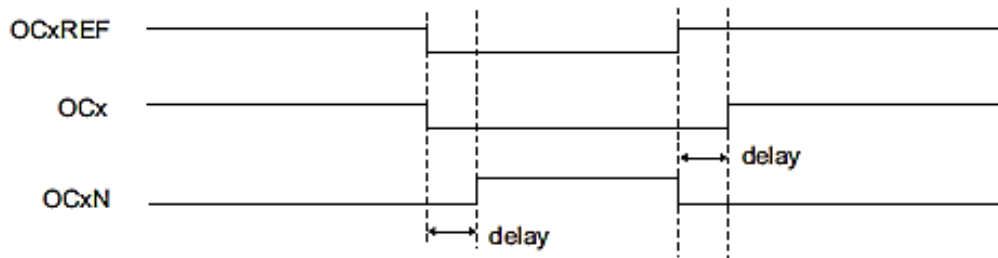


Рис. 9.15. Временные диаграммы, поясняющие функцию «*dead-time insertion*» таймеров МК семейства *ARM Cortex-Mx* (см. пояснения в тексте)

На рис. 9.15:

- *OCxREF* – внутренний импульс, вырабатываемый таймером по каналу *x*, в соответствии с выбранным режимом работы и содержимыми регистров *ARR* и *CCR* (см. подпункт 9.3.3.4);
- *OCx* и *OCxN* – сигналы на прямом и инверсном внешних выводах канала *x* таймера.

При этом переключение сигнала *OCx* в нулевое состояние осуществляется по спаду импульса *OCxREF*, а переход сигнала *OCxN* в единичное состояние – с задержкой (*delay*) относительно спада *OCxREF*. В свою очередь, переключение сигнала *OCxN* в нулевое состояние происходит по фронту импульса *OCxREF*, а переход сигнала *OCx* в единичное состояние – с задержкой *delay* относительно данного фронта. В результате на прямом и инверсном выходах таймера формируются противофазные импульсы с «мертвым временем» между переключениями (см. рис. 9.13).

Выбор режима «*dead-time insertion*», а также конкретного значения задержки осуществляются программно (см. подпункт 9.5.3.23).

Важно отметить следующее. При необходимости равенства длительностей импульсов *OCx* и *OCxN* (например, если они служат для управления ключами преобразователя, схема которого приведена на рис. 9.14), скважность импульсов *OCxREF* должна быть постоянна и равна 2-м. Это может быть реализовано:

- в режиме асимметричной ШИМ – при $CCR_x = (TOP + 1)/2$ (см. выражения (9.5) и (9.6) и подпункт 9.3.3.4);

- в режиме центрированной ШИМ – при $CCR_x = TOP/2$ (см. выражения (9.7) и (9.8) и подпункт 9.3.3.4);

- при работе формирователя выходного сигнала (см. рис. 9.2) в режиме *Toggle*, т. е. изменения состояния сигнала $OCxREF$ на противоположное при каждом совпадении содержимых счетчика и регистра сравнения (см. также пункт 9.3.4.3).

При равной 2-м скважности импульсов $OCxREF$ их длительность равна половине их периода, а длительность импульсов OCx и $OCxN$, как нетрудно увидеть из рис. 9.15, определяется выражением:

$$\tau = (T/2) - t_{delay}. \quad (9.9)$$

Таким образом, в процессе ШИМ длительность импульсов (и, соответственно, их скважность) задается одним параметром – t_{delay} , что удобно при реализации ШИМ. С другой стороны, основным **недостатком** данного способа ШИМ являются изменения начальной фазы сигнала при изменении его длительности в процессе как асимметричной, так и центрированной ШИМ.

9.3.3.7. Если таймером напрямую не поддерживается функция «*dead-time insertion*», что характерно для всех таймеров МК семейства *AVR* [6, 7] и таймеров категории *General-purpose timers* МК семейства *ARM Cortex-Mx* [13 – 15], формирование противофазных ШИМ-сигналов с задержкой между переключениями может быть реализовано в режиме центрированной ШИМ, с использованием 2-х каналов таймера. Принцип формирования (на примере таймера МК семейства *ARM Cortex-Mx*) поясняет рис. 9.16.

В представленном на рис. 9.16 примере задействованы два канала таймера – k -й и m -й. Первый из них сконфигурирован на генерацию инвертированного ШИМ-сигнала: переключение в единичное состояние происходит по совпадению содержимых счетчика и регистра сравнения на интервале прямого счета, переключение в нулевое состояние – по совпадению на участке обратного счета. Длительность импульсов, генерируемых на выходе k -го канала, при этом равна:

$$\tau_k = \frac{2 \times (TOP - CCRk)}{f_{CNT}}. \quad (9.10)$$

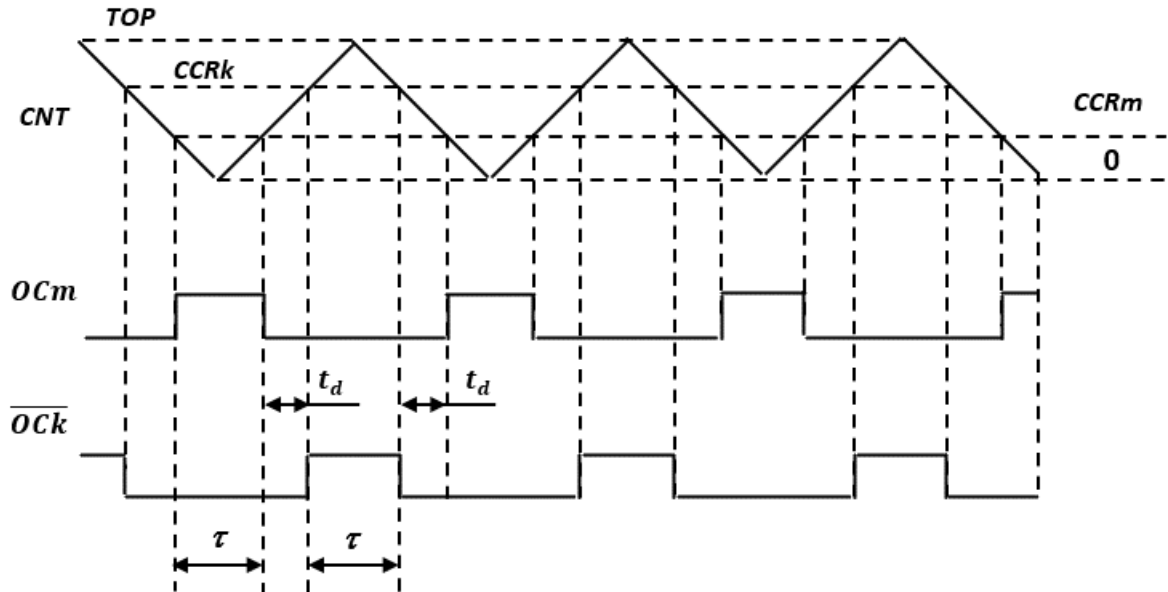


Рис. 9.16. Принцип формирования противофазных ШИМ-сигналов с задержкой между переключениями на таймере, не поддерживающем функцию «*dead-time insertion*» (см. пояснения в тексте)

В свою очередь, m -й канал сконфигурирован на генерацию не инвертированного ШИМ-сигнала: переключения в единичное и в нулевое состояния происходят на интервалах обратного и прямого счета соответственно. На выходе m -го канала генерируются импульсы с длительностью, равной:

$$\tau_m = \frac{2 \times CCRm}{f_{CNT}}. \quad (9.11)$$

Если необходимо равенство длительностей импульсов, генерируемых на выходах k -го и m -го каналов, некоторому одному и тому же значению τ (см. рис. 9.16), содержимое регистров $CCRk$ и $CCRm$ должно быть равно:

$$CCRk = TOP - \frac{\tau \times f_{CNT}}{2}; \quad (9.12)$$

$$CCRm = \frac{\tau \times f_{CNT}}{2}. \quad (9.13)$$

Задержка между переключениями по каналам при этом равна:

$$t_d = \frac{CCRk - CCRm}{f_{CNT}}. \quad (9.14)$$

Все вышесказанное в равной степени относится и к МК семейства AVR; необходимо только учесть, что их регистры сравнения носят имя OCR [6, 8].

Формирование противофазных ШИМ-сигналов вышеописанным способом может быть реализовано на таймерах практически всех семейств МК классов «*mainstream*» и «*high performance*». **Недостатком** данного способа по сравнению с описанным в подпункте 9.3.3.6 является необходимость программного изменения 2-х параметров (содержимого регистров CCRk и CCRm) в процессе ШИМ, для удовлетворения условий (9.12) и (9.13). С другой стороны, основным **достоинством** данного способа являются обеспечение ШИМ, корректной по фазе (см. рис. 9.12)

9.3.3.8. Следует отметить, что при реализации ШИМ способами, описанными в подпунктах 9.3.3.6 и 9.3.3.7, в ПО управления ШИМ должны быть **предусмотрены** меры против снижения «мертвого времени» ниже допустимого значения; естественно, особенно опасно его уменьшение до нуля. В частности:

- по включении и по сбросу МК, до активизации выходов таймера, должны быть установлены параметры его конфигурации, обеспечивающие гарантированно безопасное значение «мертвого времени» (см. рис. 9.15 и выражения (9.14));

- в процессе ШИМ необходимо устранить вероятность выхода параметров конфигурации таймера за пределы, вне которых значение «мертвого времени» ниже минимально допустимого.

9.3.3.9. Как правило, ШИМ-сигналы, формируемые таймерами МК, подаются на ИУ через **драйверы** нагрузок (см. пункт 6.4.3).

9.3.4. Формирование ЧИМ-сигналов посредством таймеров МК

9.3.4.1. Архитектура таймеров большинства современных семейств МК классов «*mainstream*» и «*high performance*» позволяет, кроме широтно-импульсной, реализовывать также частотно-импульсную модуляцию (ЧИМ), т. е. генерировать импульсы с фиксированной скважностью (как правило, равной 2-м) и с программно-управляемой частотой. На практике ЧИМ применяется несколько реже, чем ШИМ. Типовыми примерами использования ЧИМ в электронных средствах на базе МК являются:

- генерация двухчастотных или многочастотных сигналов звукового оповещения в системах охранной и противопожарной сигнализации (опыт показывает, что двухчастотный / многочастотный сигнал слышен на в разы большем расстоянии, чем одночастотный);

- генерация сигналов с программно задаваемой частотой в цифровых средствах связи (в первую очередь – беспроводной) на базе МК.

Формирование ЧИМ-сигнала, как и ШИМ-сигнала, осуществляется блоком, обобщенная структурная схема которого приведена на рис. 9.2. Наиболее распространенный способ генерации ЧИМ-сигнала посредством таймера МК состоит в следующем:

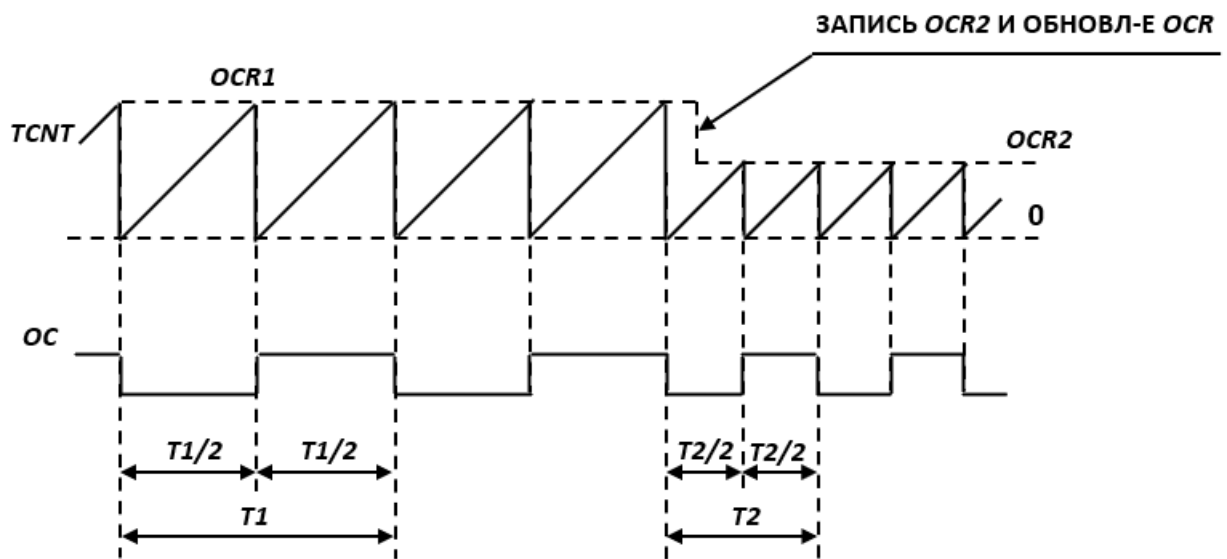
- частота сигнала задается или верхним пределом счета (*TOP*), или содержимым регистра сравнения;

- формирователь выходного сигнала работает в режиме *Toggle* (переключение состояния сигнала на противоположное при достижении равенства содержимых счетчика и регистра сравнения). Таким образом, формируется импульсный сигнал с программно-управляемой частотой и не зависящей от нее, фиксированной скважностью, равной 2-м.

Для краткости, в дальнейшем будем называть данный способ ***Toggle-ЧИМ***. Рассмотрим его подробнее.

9.3.4.2. На рис. 9.17 представлены временные диаграммы, поясняющие способ *Toggle-ЧИМ* на примере его реализации

таймером МК семейства *AVR*. В приведенном на рис. 9.17 примере используется режим работы таймера *CTC* (*Clear Timer on Compare Match*, «Сброс таймера по сравнению») [6, 8], формирователь выходного сигнала работает в режиме *Toggle*. Счетчик таймера при этом осуществляет прямой счет от нуля до верхнего предела, равного содержимому регистра, сравнения, по достижении которого происходит сброс счетчика, и возобновление счета с нуля. Переключение выходного сигнала происходит при сбросе счетчика.



$T1, T2$ – период генерируемого сигнала на соответствующих интервалах времени

Остальные условные обозначения соответствуют приведенным на рис. 9.7

Рис. 9.17. Пример реализации *Toggle*-ЧИМ посредством таймера МК семейства *AVR* в режиме *CTC*

На рис. 9.18 представлен фрагмент детализированной временной диаграммы формирования сигнала способом *Toggle*-ЧИМ посредством таймера МК семейства *AVR* в режиме *CTC*.

Из рис. 9.18 следует, что зависимость частоты ЧИМ-сигнала от содержимого регистра *OCR* подчиняется выражению:

$$f_{PFM} = \frac{f_{CNT}}{2 \times (OCR + 1)}. \quad (9.15)$$

Скважность формируемых импульсов не зависит от частоты и равна 2-м.

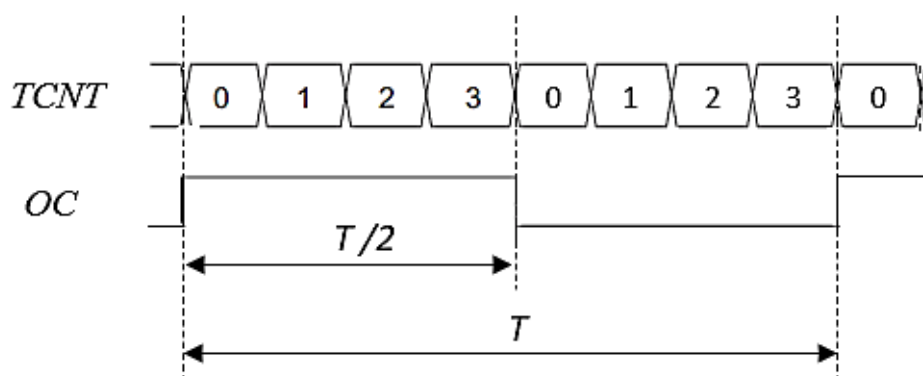


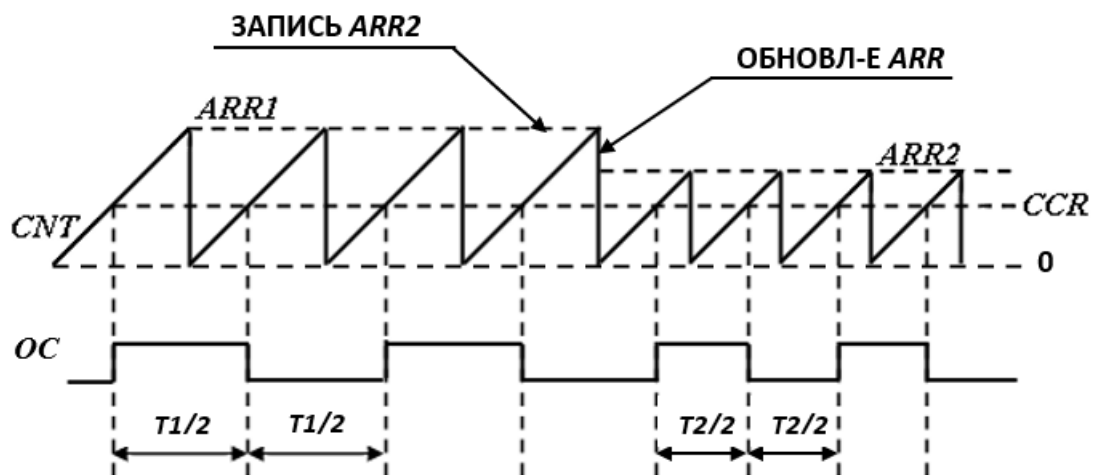
Рис. 9.18. Фрагмент детализированной временной диаграммы формирования ЧИМ-сигнала посредством таймера МК семейства *AVR* в режиме *CTC* ($OCR = 3$)

Необходимо заметить, что в режиме *CTC* регистр *OCR* МК семейства *AVR* не буферизирован, что не критично при формировании ЧИМ-сигнала.

Архитектура 16-битовых таймеров МК семейства *AVR* позволяет также формировать сигналы, фактически модулированные по частоте, в тех вариантах ШИМ, в которых верхний предел счета равен содержимому регистра *OCR* (при условии, что формирователь выходного сигнала запрограммирован на режим переключения *Toggle*) [6, 8]. Частота выходного сигнала при этом также описывается выражением (9.9); регистр *OCR* буферизирован, и его обновление происходит при перезапуске счетчика или по достижении верхнего предела счета (см. рис. 9.7 и 9.10).

9.3.4.3. Реализация способа *Toggle*-ЧИМ посредством таймеров МК семейства *ARM Cortex-Mx* характеризуется следующей особенностью: верхний предел счета и моменты переключения выходного сигнала (фактически – его частота и начальная фаза) задаются 2-мя различными регистрами. Верхний предел счета равен содержимому регистра *ARR* (*Auto Reload Register*), а переключение выходного сигнала осуществляется по достижении равенства содержимых счетчика и регистра *CCR* (*Capture / Compare Register*). Оба данных регистра программно доступны и буферизированы. Буферизация разрешается / запрещается программно; рекомендуется ее разрешить.

На рис. 9.19 представлен пример временных диаграмм формирования сигнала способом *Toggle*-ЧИМ посредством таймера МК семейства *ARM Cortex-Mx*. В данном примере счетчик таймера работает в режиме прямого счета, от нуля до верхнего предела, равного содержимому регистра *ARR*. После достижения верхнего предела счетчик сбрасывается в нулевое состояние, и счет возобновляется. Переключения формируемого сигнала, как указано выше, осуществляются при каждом совпадении содержимого счетчика и регистра *CCR*. Программно-управляемая запись регистра *ARR* может осуществляться в произвольные моменты времени, но при разрешенной буферизации реальное обновление его содержимого происходит при перезапуске счетчика (см. рис. 9.19).



CNT – содержимое счетчика (реально изменяется ступенчато, для упрощения рисунка показано изменяющимся непрерывно)

ARR1, *ARR2* – содержимое регистра *ARR* на соответствующих интервалах времени

CCR – содержимое регистра захвата / сравнения

T1, *T2* – период генерируемых импульсов на соответствующих интервалах времени

Рис. 9.19. Пример формирования ЧИМ-сигнала посредством таймера МК семейства *ARM Cortex-Mx* в режиме *Toggle*

На рис. 9.20 представлен фрагмент детализированной временной диаграммы сигнала, формируемого способом *Toggle*-ЧИМ посредством таймера МК семейства *ARM Cortex-Mx* (счет – прямой, *ARR* = 3, *CCR* = 1). Из рис. 9.20 нетрудно заметить, что частота сигнала при этом описывается выражением:

$$f_{PFM} = \frac{f_{CNT}}{2 \times (ARR + 1)}; \quad (9.16)$$

скважность импульсов не зависит от частоты и равна 2-м. То же справедливо и при работе счетчика в режиме обратного счета.

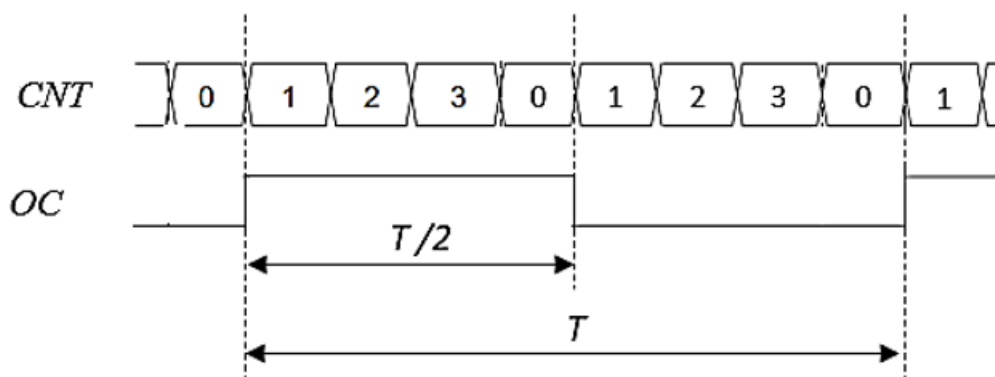


Рис. 9.20. Фрагмент детализированной временной диаграммы формирования ЧИМ-сигнала посредством таймера МК семейства *ARM Cortex-Mx* в режиме *Toggle* (счет – прямой, $ARR = 3$, $CCR = 1$)

В принципе, возможна и реализация способа *Toggle*-ЧИМ посредством таймера МК семейства *ARM Cortex-Mx* при работе его счетчика в реверсивном режиме (прямого счета от нуля до верхнего предела с последующим обратным счетом от верхнего предела до нуля). Однако, при этом, для обеспечения постоянной скважности импульсов, в процессе ЧИМ необходимо изменять содержимое и *ARR*, и *CCR*.

Благодаря тому, что частота и начальная фаза сигнала, формируемого таймером МК семейства *ARM Cortex-Mx* в режиме *Toggle*, задаются независимо одна от другой, различными регистрами (*ARR* и *CCR*), посредством таймеров данного семейства может быть реализована и модуляция импульсных сигналов по **фазе** (см. пункт 9.3.5).

9.3.4.4. Важным достоинством *Toggle*-ЧИМ является то, что в процессе ЧИМ достаточно изменять содержимое только одного регистра (*OCR* в таймерах МК семейства *AVR*, регистра *ARR* – в таймерах МК семейства *ARM Cortex-Mx*).

9.3.4.5. В принципе, кроме *Toggle*-ЧИМ, возможны и другие способы формирования ЧИМ-сигнала. В частности, оно может быть

реализовано при работе таймера в режиме ШИМ. ЧИМ реализуется посредством программного управления периодом сигнала, а постоянная скважность генерируемых импульсов обеспечивается за счет поддержки фиксированного (как правило, равного 2-м) отношения между периодом и длительностью генерируемых импульсов. Таким образом, в процессе модуляции изменяются и период, и длительность импульсов.

Вышеописанный способ поясняет рис. 9.21, на котором представлены временные диаграммы формирования ЧИМ-сигнала в режиме асимметричной ШИМ, на примере таймера МК семейства *ARM Cortex-Mx*. Частота сигнала при этом определяется по выражению:

$$f_{PFM} = \frac{f_{CNT}}{ARR + 1}. \quad (9.17)$$

Заметим, что она в 2 раза выше частоты сигнала, формируемого способом *Toggle*-ЧИМ, при том же значении *ARR*.

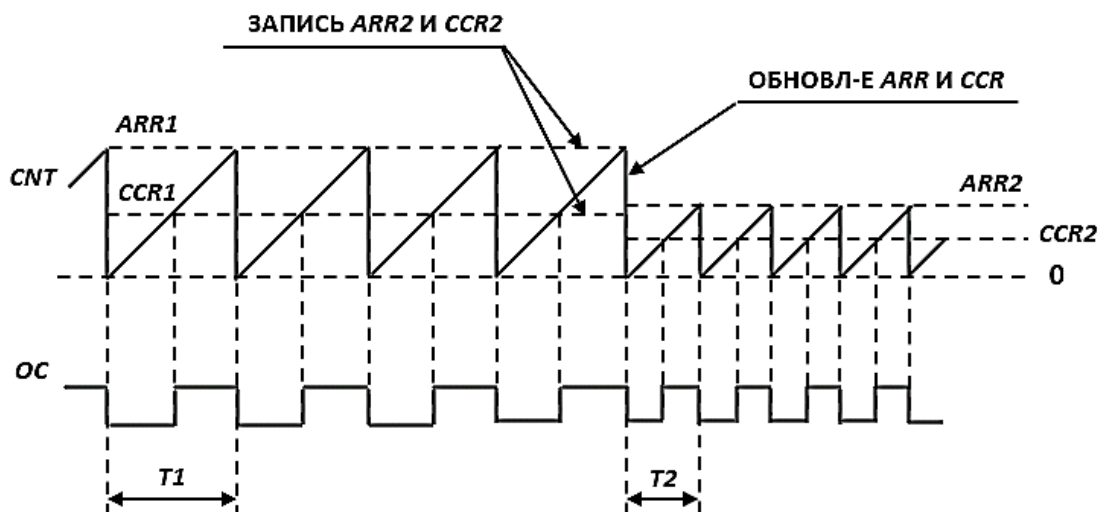


Рис. 9.21. Пример формирования ЧИМ-сигнала в режиме ШИМ посредством таймера МК семейства *ARM Cortex-Mx*

Для обеспечения скважности импульсов, равной 2-м, в процессе ЧИМ содержимое регистра *CCR* должно постоянно поддерживаться в 2 раза меньшим, чем *ARR + 1*. Таким образом, при каждом изменении содержимого регистра *ARR* необходимо пересчитывать и изменять также и содержимое *CCR*, что является существенным недостатком данного способа ЧИМ. Поэтому, несмотря на то что в

ряде конкретных случаев он может обеспечить формирование ЧИМ-сигнала на более высокой частоте, чем *Toggle*-ЧИМ, данный способ менее распространен на практике.

9.3.5. Формирование ФИМ-сигналов посредством таймеров МК

Как указано ранее, ФИМ-сигналы характеризуются фиксированной частотой и скважностью, и программно-управляемой начальной фазой. ФИМ, в частности, применяется в цифровых средствах связи на базе МК.

ФИМ может быть реализована, например, таймерами МК семейства *ARM Cortex-Mx* при работе выхода в режиме *Toggle*, за счет независимого задания частоты и начальной фазы сигнала в данном режиме (см. подпункт 9.3.4.3). Принцип реализации ФИМ в данном режиме поясняют временные диаграммы, представленные на рис. 9.22.

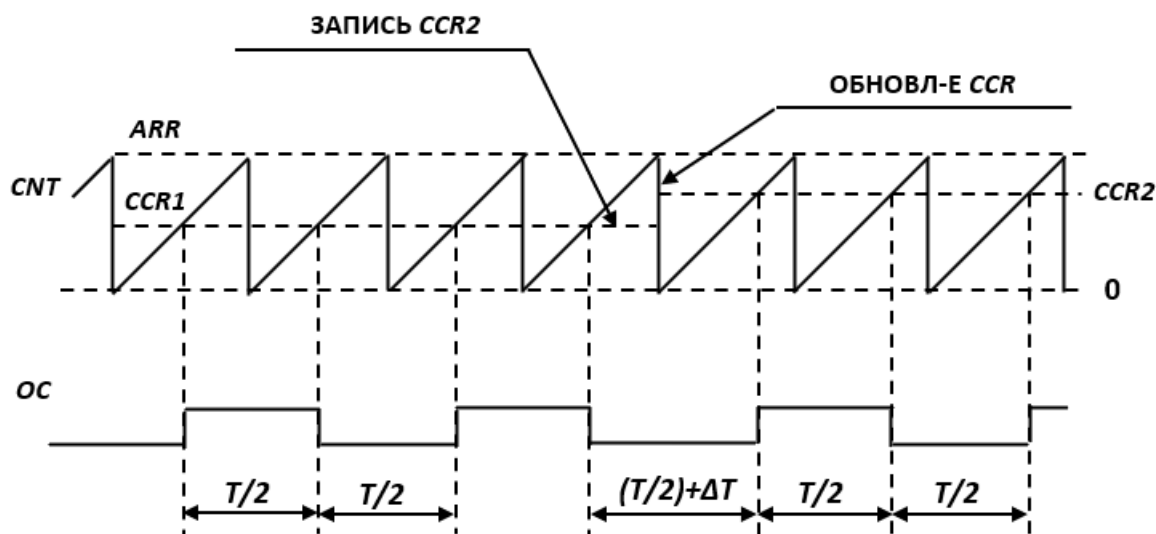


Рис. 9.22. Пример формирования ФИМ-сигнала посредством таймера МК семейства *ARM Cortex-Mx* в режиме *Toggle*

В приведенном на рис. 9.22 примере счетчик таймера работает в режиме прямого счета, от нуля до верхнего предела, равного содержимому регистра *ARR*, после достижения которого происходит сброс и перезапуск счетчика. Содержимое *ARR* не изменяется в процессе модуляции; частота генерируемого сигнала при этом

постоянна и определяется выражением (9.16). Фазовая модуляция осуществляется изменениями содержимого регистра CCR , каждое из которых вызывает фазовый сдвиг ФИМ-сигнала, равный (см. рис. 9.22):

$$\Delta\varphi = \frac{2\pi \times \Delta T}{T} = \frac{2\pi}{T} \times \frac{\Delta CCR}{f_{CNT}}; \quad (9.18)$$

где ΔCCR – разность между текущим и предыдущим значениями содержимого регистра CCR , в приведенном на рис. 9.22 примере равная $CCR2 - CCR1$.

9.3.6. Реализация режима одновибратора посредством таймеров МК

Как указано в пункте 9.3.1, архитектура таймеров ряда семейств МК (в частности, *ARM Cortex-Mx*) позволяет реализовывать режим одновибратора, т. е. генерировать, по некоторому старт-событию, одиночный импульс с программно-управляемыми длительностью и задержкой относительно момента наступления старт-события. Рассмотрим особенности реализации данного режима, на примере таймеров МК модельного ряда *STM32F030xx* [15].

Работу таймера МК данного модельного ряда (и семейства *ARM Cortex-Mx* в целом) в режиме одновибратора поясняет рис. 9.23. На нем в качестве типового примера приведена временная диаграмма формирования одиночных импульсов 1-м каналом 1-го таймера. Его счетчик в данном примере работает в режиме одностороннего прямого счета. Запуск счетчика осуществляется сигналом, обозначенным на рис. 9.23 как *TRG_SIG*, переход которого из пассивного в активное состояние служит старт-событием (см. выше). В качестве данного сигнала, в общем случае, может служить [15]:

- сигнал, генерируемый некоторым другим таймером, входящим в состав МК, по определенному событию (перезапуску, совпадению содержимых счетчика и регистра сравнения и т. д.);
- внешний (по отношению к МК) сигнал синхронизации;
- сигнал на одном из входов захвата таймера (см. подпункт 9.4.2.5).

В принципе, стартом-событием может служить программное разрешение работы таймера (см. Пример 3 в подпункте 9.5.4.3).

Для генерации одиночного импульса с активным единичным и пассивным нулевым уровнем выбран режим *PWM2* переключения выхода 1-го канала таймера, при котором (см. пункт 14.4.7 в [15]):

- переключение из нулевого состояния в единичное происходит по достижении равенства содержимых счетчика и регистра сравнения таймера;
- переключение из единичного состояния в нулевое осуществляется по сбросу счетчика в нулевое состояние.

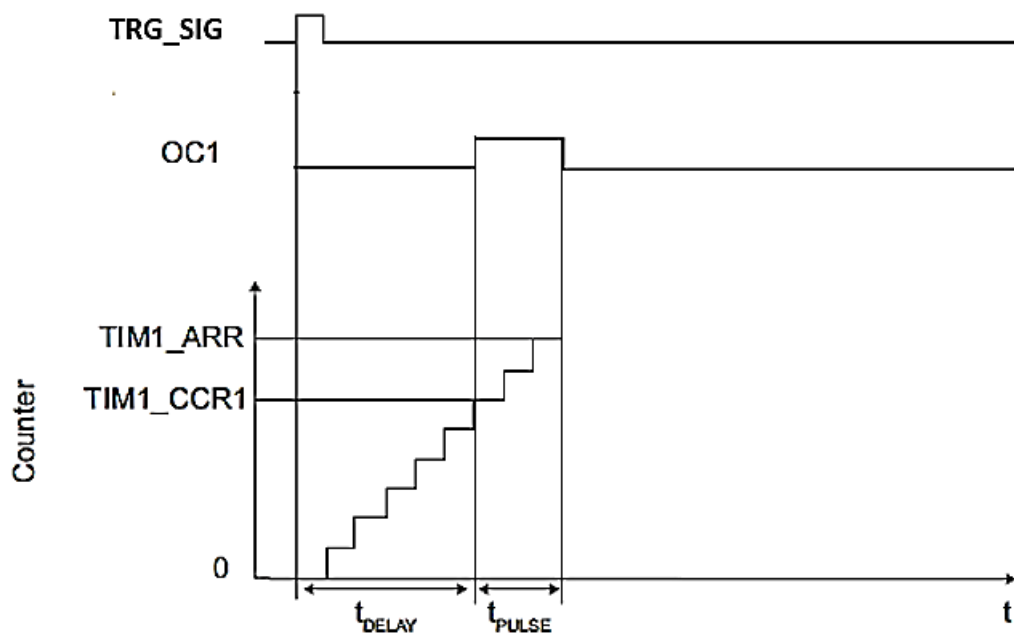


Рис. 9.23. Пример временных диаграмм работы таймера МК модельного ряда *STM32F030xx* [15] в режиме одновибратора (см. пояснения в тексте)

Выбор режима работы счетчика, сигнала запуска и режима переключения выхода осуществляются программно, посредством регистров управления таймером (см. пункт 14.3.10 в [15]).

Если счетчик таймера сконфигурирован на работу в режиме одностороннего прямого счета, по поступлении сигнала запуска на выходе *n*-го канала таймера (в рассматриваемом примере – 1-го) генерируется импульс с длительностью:

$$t_{PULSE} = \frac{ARR + 1 - CCRn}{f_{CNT}}; \quad (9.19)$$

с задержкой относительно старт-события, равной:

$$t_{DELAY} = \frac{CCRn}{f_{CNT}}. \quad (9.20)$$

Поскольку регистры *ARR* и *CCR* программно-доступны, значения t_{PULSE} и t_{DELAY} являются программно-задаваемыми и управляемыми. В примере, приведенном на рис. 9.23, $ARR = 8$, $CCRn = CCR1 = 6$, $t_{PULSE} = 3/f_{CNT}$, $t_{DELAY} = 6/f_{CNT}$.

Для работы в режиме одновибратора **должен быть** установлен в единицу бит ***OPM*** в 1-м регистре управления таймером. При единичном состоянии данного бита, по наступлении события «Обновление содержимого счетчика» (в рассматриваемом примере – по его сбросу в ноль) происходит его остановка, т. е. автоматический сброс бита *Counter enable (CEN)* 1-го регистра управления таймера. Для возможности генерации импульса по следующему сигналу запуска данный бит должен быть программно установлен в единицу.

Более подробно вопросы реализации режима одновибратора изложены в технической документации на соответствующие подсемейства / модельные ряды МК (см., например, [13 – 15]). Пример программного модуля с использованием режима одновибратора приведен в подпункте 9.5.4.3.

9.4 Преобразование частотно-временных параметров сигналов в цифровой код посредством таймеров МК

9.4.1. Общие вопросы преобразования частотно-временных параметров сигналов в цифровой код

Основными частотно-временными параметрами сигналов, задачи преобразования которых в код встречаются в типовых областях применения МК, являются частота (период) импульсных или гармонических сигналов, а также длительность интервала времени между 2-мя событиями [38]. Достаточно широко распространены датчики физических величин с **частотным**

выходом, принцип работы которых основан на преобразовании контролируемой величины в частоту выходного сигнала датчика. К ним относятся, например, большинство датчиков угловой и линейной скорости, ряд типов датчиков приближения, температуры, влажности и т. д. [51]. В свою очередь, на преобразовании контролируемой величины в **интервал времени** основан, в частности, принцип работы ультразвуковых датчиков приближения, уровня и расхода [51]. Также в ряде систем мониторинга и управления объектом на базе МК необходим контроль длительностей интервалов времени между некоторыми событиями (например, между моментами поступления сигналов запроса от объекта).

Естественно, при использовании датчиков с частотным или с временным выходом в системе на базе МК или при контроле интервалов времени между событиями при ее работе, необходимо преобразование частоты (периода) выходных сигналов датчиков или длительностей интервалов времени в цифровой код, с целью обработки и анализа посредством ПО МК.

Наиболее распространенным способом преобразования частотно-временных параметров сигналов в цифровой код является **электронно-счетный способ** [38].

Преобразование **частоты** сигнала в цифровой код данным способом реализуется подсчетом числа периодов сигнала, «укладывающихся» в интервал времени с известной длительностью (который назовем **опорным интервалом**). Результат подсчета при этом равен:

$$N_X = (\Delta t_R / T_X) \pm 1 = f_X \Delta t_R \pm 1; \quad (9.21)$$

где T_X и f_X – соответственно период и частота сигнала; Δt_R – длительность опорного интервала времени. Таким образом, результат счета прямо пропорционален преобразуемой частоте с абсолютной погрешностью, равной ± 1 . Ее причинами, в общем случае, являются:

- не кратность значения Δt_R периоду сигнала, частота которого подлежит преобразованию;
- не синхронизация начала и окончания опорного интервала времени (т. е. интервала счета) с начальной фазой сигнала, частота

которого подлежит преобразованию; из-за не синхронизации один его период может быть пропущен или в результат подсчета может быть включен один «лишний» период.

Значение преобразуемой частоты вычисляется как отношение $N_X/\Delta t_R$. При этом ее реальное значение находится в пределах:

$$f_X = (N_X/\Delta t_R) \pm (1/\Delta t_R); \quad (9.22)$$

Таким образом, относительная погрешность преобразования равна $\pm 1/(f_X \Delta t_R)$. Корректным выбором длительности опорного интервала времени Δt_R она может быть уменьшена до значений, приемлемых для большинства практических случаев.

В свою очередь, преобразование в код **длительности интервала времени** (в т. ч. периода сигнала) осуществляется подсчетом импульсов с известной (опорной) частотой, «укладывающихся» в соответствующий интервал. Результат подсчета при этом равен:

$$N_X = \Delta t_X f_R \pm 1; \quad (9.23)$$

где Δt_X – подлежащая преобразованию длительность интервала времени; f_R – значение опорной частоты. При этом результат счета прямо пропорционален значению Δt_X с абсолютной погрешностью ± 1 . Данная погрешность, как и при преобразовании частоты в код, возникает из-за не кратности значения Δt_X периоду счетных импульсов и не синхронизации начала и окончания интервала счета с их начальной фазой.

Длительность интервала Δt_X вычисляется как отношение N_X/f_R . Ее реальное значение при этом находится в пределах:

$$\Delta t_X = (N_X/f_R) \pm (1/f_R); \quad (9.24)$$

Относительная погрешность преобразования, таким образом, равна $\pm 1/(f_R \Delta t_X)$. Она может быть уменьшена до значений, приемлемых для большинства практических случаев, корректным выбором опорной частоты f_R .

Следует заметить, что преобразование частоты в код может быть заменено преобразованием периода с последующим программным пересчетом его в частоту, если необходимая погрешность

преобразования не может быть обеспечена при допустимых значениях Δt_R . Например, пусть необходимо реализовать преобразование в код частоты некоторого сигнала в диапазоне от 10 до 100 Гц с относительной погрешностью, не превышающей $\pm 0,001$ ($\pm 0,1\%$). Нетрудно увидеть, что для обеспечения этого условия потребуется интервал счета с длительностью Δt_R не менее $1/(10 \text{ Гц} \times 0,001)$, т. е. 100 с, что может быть не приемлемо во многих практических случаях. В то же время указанная погрешность обеспечивается при преобразовании в код периода данного сигнала и частоте счетных импульсов (опорной частоте), равной $1/(10 \text{ мс} \times 0,001)$, т. е. 100 кГц, что вполне реализуемо. Перерасчет периода в частоту, при необходимости, может быть выполнен программно.

Важно отметить, что общая абсолютная погрешность преобразования частоты или длительности интервала времени в код равна ± 1 только при **условии**, что по сравнению с ней погрешность значения Δt_R или, соответственно, f_R пренебрежимо мала. Данное условие выполнимо в абсолютном большинстве практических случаев, и его соблюдение является **необходимым**.

Следует также отметить, что в прецизионных средствах измерения частоты и интервалов времени используются дополнительные способы снижения погрешности (см., например, [52]), рассмотрение которых выходит за рамки настоящего пособия.

«Классическая» схема преобразователя частоты и длительности интервала времени в код электронно-счетным способом представлена на рис. 9.24. Принцип преобразования состоит в подсчете числа импульсов частотой f , «укладывающихся» в пределы «временного окна», задаваемого импульсом разрешения счета с длительностью Δt . При преобразовании частоты в код $f = f_X$, а $\Delta t = \Delta t_R$; при преобразовании длительности интервала времени – $f = f_R$, а $\Delta t = \Delta t_X$ (см. выражения (9.21) и (9.23)).

Естественно, для реализации преобразования по вышеприведенной схеме:

- сигнал, частоту или период которого необходимо определить, должен быть преобразован в последовательность прямоугольных импульсов с той же частотой (периодом), совместимых с

используемой элементной базой по напряжениям «низкого» и «высокого» уровней;

- при преобразовании в код длительности интервала времени между 2-мя событиями или длительности импульсного сигнала должен быть сформирован импульс разрешения счета с той же длительностью (см. рис. 9.24), также совместимый по напряжениям «низкого» и «высокого» уровней с применяемой элементной базой.

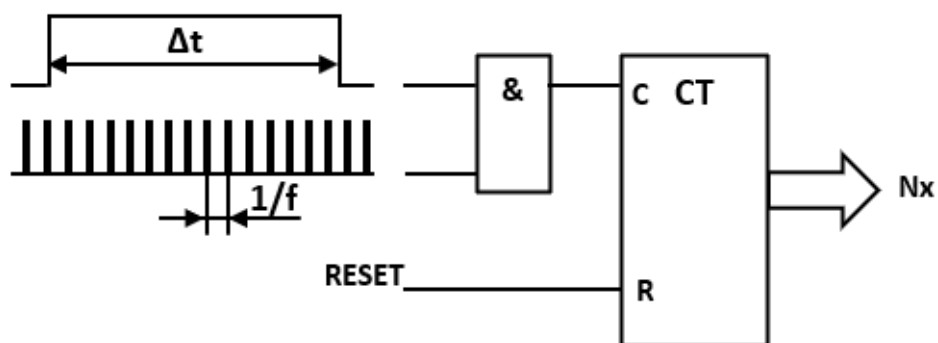


Рис. 9.24. «Классическая» схема преобразователя частоты и длительности интервала времени в код электронно-счетным способом (см. пояснения в тексте)

Формирование импульсов, в общем случае, реализуется посредством аналоговых компараторов [20, 22], подробнее – см. пункты 9.4.2 и 9.4.3. Необходимость в них отпадает, если сигнал, частотно-временные параметры которого необходимо преобразовать в код, является цифровым и совместимым по уровням напряжения с электронными компонентами преобразователя.

Из вышесказанного нетрудно увидеть, что в настоящее время преобразования частотно-временных параметров сигналов в код электронно-счетным способом рационально реализовывать на основе таймеров МК. Данный подход позволяет минимизировать аппаратные затраты, габариты и энергопотребление системы контроля и управления объектом, в процессе работы которой требуется выполнять указанные преобразования. В самом деле:

- в составе каждого из таймеров, естественно, имеется счетчик;
- интервал счета при преобразовании частоты в код также возможно формировать посредством таймера (см. подразделы 9.2 и 9.3);

- при преобразовании в код длительности интервала времени (см. выражение (9.23)) в качестве источника счетных импульсов с опорной частотой может быть использован высокочастотный или низкочастотный ГТИ МК на базе внешнего ПЭР (см. пункт 4.2.3), при необходимости – с делением частоты ГТИ до требуемого значения f_R ;

- для формирования импульсов (импульсных последовательностей), совместимых с МК по уровням напряжения, в структуре ряда семейств (подсемейств) МК имеются аналоговые компараторы.

Рассмотрим подробнее вопросы преобразования в код частотно-временных параметров сигналов посредством таймеров МК.

9.4.2. Преобразование частоты сигналов в цифровой код посредством таймеров МК

9.4.2.1. Как указано в пункте 9.4.1, преобразование частоты в цифровой код электронно-счетным способом сводится к подсчету числа периодов сигнала с неизвестной частотой, «укладывающихся» в опорный интервал времени, длительность которого известна (см. выражение (9.21)). Для решения данной задачи посредством таймера МК требуется реализовать:

- преобразование сигнала, частоту которого необходимо определить, в последовательность прямоугольных импульсов с той же частотой, совместимых с портами МК по напряжениям «низкого» и «высокого» уровней;

- формирование опорного интервала времени, в течение которого осуществляется счет;

- подсчет числа импульсов с частотой f_X , «укладывающихся» в опорный интервал времени.

9.4.2.2. Первая из перечисленных задач решается с помощью аналогового компаратора [20, 22]. Обобщенная структурная схема формирователя последовательности импульсов с частотой f_X при преобразовании частоты в код приведена на рис. 9.25.

Основой данной схемы служит аналоговый компаратор (АК). Фактически, он представляет собой дифференциальный усилитель с

весьма высоким коэффициентом усиления (от нескольких десятков тысяч и выше), выходной каскад которого работает в ключевом режиме, с 2-мя устойчивыми состояниями выходного напряжения – «отрицательного» и «положительного» насыщения. У промышленно выпускаемых АК (в том числе встроенных в МК) данные уровни равны напряжениям логического нуля и логической единицы цифровых ИС.

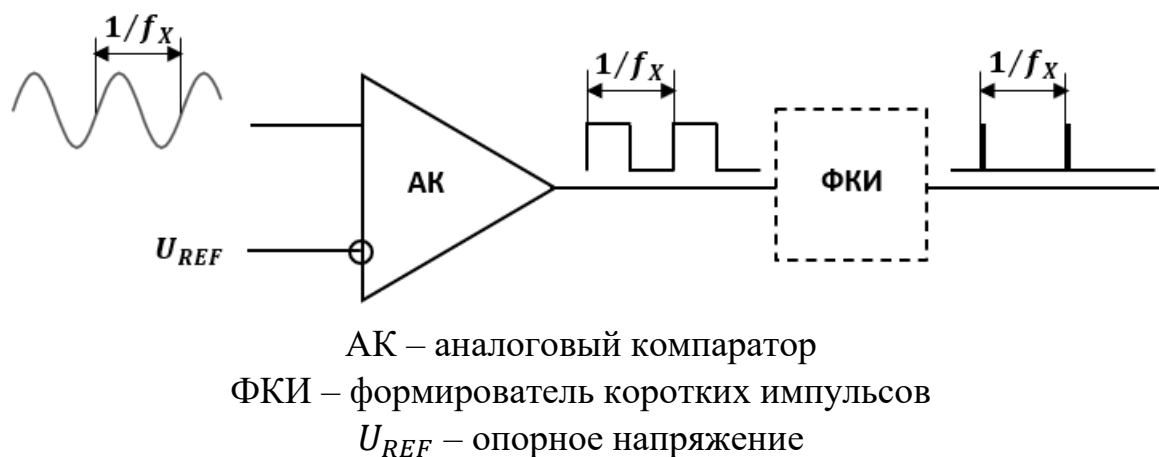


Рис. 9.25. Обобщенная структурная схема формирователя последовательности импульсов с частотой f_X при преобразовании частоты в код (см. пояснения в тексте)

Благодаря высокому коэффициенту усиления дифференциального сигнала, если напряжение на не инвертирующем входе АК на несколько сотен микровольт и более превышает напряжение на инвертирующем, выход АК переходит в состояние «положительного насыщения», т. е. логической единицы. Если же напряжение на инвертирующем входе на несколько сотен микровольт и более превышает напряжение на не инвертирующем, выход АК переключается в состояние «отрицательного насыщения», т. е. логического нуля. Таким образом, если на не инвертирующий вход АК подать сигнал, частота которого подлежит преобразованию в код, а на инвертирующий – опорное напряжение, равное среднему значению этого сигнала, выходной сигнал АК будет представлять собой последовательность прямоугольных импульсов с частотой f_X и с «низким» уровнем, равным напряжению логического нуля, а «высоким» - логической единицы (см. рис. 9.25).

Если преобразователь частоты в код реализуется по «классической» схеме, приведенной на рис. 9.24, желательно преобразование последовательности выходных импульсов АК в последовательность импульсов с той же частотой, но с высокой (порядка нескольких десятков) скважностью (так называемых «коротких» импульсов). Это позволяет минимизировать вероятность возникновения составляющей погрешности, вызванной не синхронизацией начала и окончания интервала счета с начальной фазой выходного сигнала АК. Эффект минимизации иллюстрирует рис. 9.26. На его 1-й и 2-й диаграммах представлены примеры выходного сигнала АК и ФКИ (см. рис. 9.25), на 3-й – импульса разрешения счета, на 4-й и 5-й – выходного сигнала логического элемента 2И (см. рис. 9.24) при использовании выходных импульсов соответственно АК и ФКИ в качестве счетных. В данном примере длительность Δt_R импульса разрешения счета кратна периоду сигнала и равна $10/f_X$, т. е. составляющая погрешности, вызываемая не кратностью значений Δt_R и $1/f_X$, теоретически должна отсутствовать. Тем не менее, если в качестве счетных выступают выходные импульсы АК, результат их подсчета за интервал времени Δt_R будет равен 11-ти, за счет паразитного счетного импульса (см. 4-ю диаграмму на рис. 9.26).

Если же в качестве счетных служат выходные импульсы ФКИ, в приведенном на рис. 9.22 примере результат счета будет корректен, и равен 10-ти (см. 5-ю диаграмму). В общем случае, в приведенной на рис. 9.24 схеме вероятность появления паразитного счетного импульса равна вероятности совпадения во времени счетного импульса с каким-либо из фронтов импульса разрешения счета; она **тем меньше, чем выше скважность** счетных импульсов.

ФКИ может быть реализован по одной из известных схем, приведенных, например, в [22].

Однако, при преобразовании частоты в код, например, способом, описанным в подпункте 9.4.2.6, скважность счетных импульсов не критична (см. данный подпункт). При этом ФКИ может отсутствовать, поэтому на рис. 9.21 он обозначен пунктиром.

В качестве АК, по возможности, рационально использовать встроенный АК (см. раздел 10).

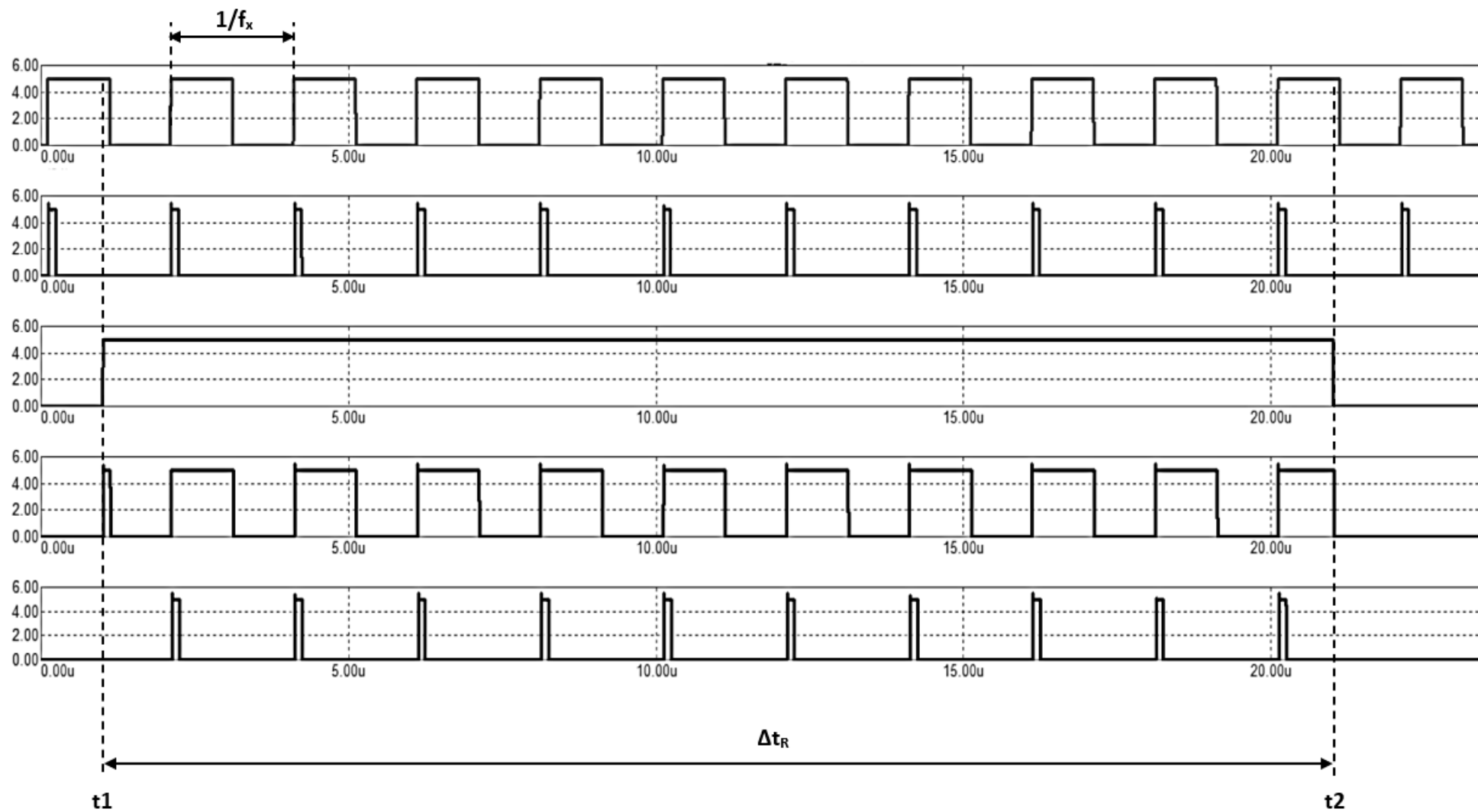


Рис. 9.26. Влияние скважности счетных импульсов на погрешность преобразования частоты в код (см. пояснения в тексте)

Если сигнал, частота которого преобразуется в код, является последовательностью прямоугольных импульсов с «низким» и «высоким» уровнями, равными напряжениям логических нуля и единицы, применение АК не требуется (но при использовании схемы, приведенной на рис. 9.24, может потребоваться ФКИ).

9.4.2.3. Преобразование частоты в код посредством таймеров МК, в принципе, может быть реализовано несколькими способами.

Наиболее очевидной является реализация по способу, описываемому схемой, приведенной на рис. 9.24. При этом для подсчета импульсов с частотой f_X следует использовать счетчик одного из таймеров МК, а формирование сигнала разрешения счета – осуществлять некоторым другим таймером (см. подраздел 9.3). Однако, такой способ потребует использования внешних по отношению к МК функциональных узлов (как минимум – логического элемента 2И, а также ФКИ, поскольку при преобразовании данным способом критична скважность счетных импульсов, см. подпункт 9.4.2.2), что противоречит общей тенденции к минимизации числа компонентов, внешних по отношению к МК. Поэтому такой способ преобразования частоты в код рационально применять только, если по каким-либо причинам нет возможности использования таймеров, реализующих функцию «захвата» (см. подпункт 9.4.2.6).

9.4.2.4. В принципе, возможна реализация преобразования частоты в код с формированием интервала счета по алгоритму, описанному в подразделе 9.2. При этом необходимы два таймера: первый – для подсчета импульсов с частотой f_X , второй – для задания интервала счета, причем в качестве источника его счетных импульсов должен быть выбран какой-либо из ГТИ на базе ПЭР (см. пункт 4.2.3).

Общий алгоритм преобразования следующий.

1. Изначально счетчики обоих таймеров должны быть остановлены.
2. Сконфигурировать первый таймер на тактирование от внешнего по отношению к МК источника, в качестве которого служит формирователь импульсов с частотой f_X (см. рис. 9.24).

3. Сконфигурировать второй таймер в соответствии с пунктом 1 алгоритма, приведенного в подразделе 9.2; при этом: $\Delta t = \Delta t_R$; в качестве источника тактирования счетчика 2-го таймера необходимо выбрать ГТИ на базе ПЭР (см. пункт 4.2.3).

4. Загрузить в счетчик второго таймера начальное значение в соответствии с выражениями (9.1) или (9.2).

5. Запустить счетчики обоих таймеров (см. пункт 3 представленного в разделе 9.2 алгоритма).

6. По признаку окончания интервала времени, формируемого вторым таймером (см. пункт 4 алгоритма, приведенного в разделе 9.2) остановить счетчики обоих таймеров.

7. Считать результат преобразования из счетчика первого таймера.

Преобразование по данному алгоритму характеризуется дополнительной (по сравнению со способами, описанными в подпунктах 9.4.2.3 и 9.4.2.6) составляющей погрешности из-за не синхронизации начала и окончания интервала счета с начальной фазой счетных импульсов. Данная составляющая возникает из-за задержек, вносимых ПО МК при запуске и при остановке счета. Поэтому описанный алгоритм может применяться только в отсутствие жестких требований к погрешности преобразования.

9.4.2.5. Рациональным сочетанием точности и аппаратных затрат характеризуется способ преобразования частоты в код, использующий функцию таймера, называемую «захват» (*capture*) [8, 9]. Она реализуется таймерами большинства семейств МК классов «*mainstream*» и «*high performance*» и состоит в следующем.

Функция захвата (в семействе *AVR* называемая **захватом по входу**, *Input Capture* [6, 8]) состоит в следующем. В момент наступления некоторого, программно назначаемого события, например, перепада сигнала на определенном выводе порта МК из нулевого состояния в единичное, содержимое счетчика автоматически, на аппаратном уровне, «захватывается» специальным регистром таймера (т. е. загружается в него). При этом в регистре статуса таймера устанавливается в активное состояние признак захвата, а также формируется запрос на прерывание по захвату (если оно разрешено).

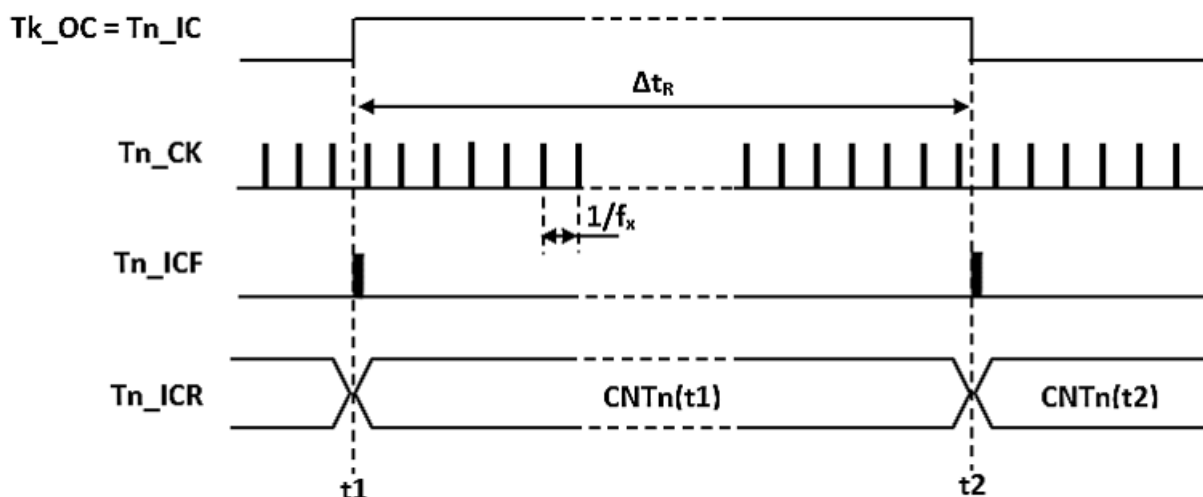
В таймерах МК семейства *AVR* для этой цели выделен регистр, называемый *Input Capture Register (ICR)* [6, 8] (как правило, один на таймер). В МК семейства *ARM Cortex-Mx* функция захвата и сравнения совмещены в одном регистре, называемом *Capture / Compare Register (CCR)* [13 – 15] (см. рис. 9.19 и пояснения к нему). Как правило, в таймерах данного семейства МК имеется несколько каналов захвата / сравнения, каждый из которых снабжен собственным *CCR*. Настройка *CCR* на выполнение функции захвата или сравнения осуществляется программно, посредством регистров конфигурации таймера.

Вообще говоря, типовая область применения функции захвата – преобразование в код интервала времени между 2-мя событиями (например, между фронтом и спадом импульса) по следующему обобщенному алгоритму [38]:

- счетчик таймера работает в режиме непрерывного прямого счета; его тактирование осуществляется импульсами с известной (опорной) частотой;
- в момент наступления каждого из событий содержимое счетчика автоматически загружается в регистр захвата и считывается из него центральным процессором (например, в подпрограмме обслуживания прерывания по захвату);
- длительность интервала времени между событиями определяется как число периодов счетных импульсов, «уложившихся» в данный интервал, т. е. как произведение их периода на разность содержимых счетчика в момент наступления каждого из событий.

Однако, аналогичным образом может быть реализовано и преобразование частоты в код, при условии, что длительность интервала времени между событиями, по которым происходит захват, является фиксированной и известной, а счетчик тактируется импульсами с частотой, подлежащей преобразованию (см. подпункт 9.4.2.6).

9.4.2.6. На рис. 9.27 представлены временные диаграммы, поясняющие способ преобразования частоты в код посредством таймеров МК с использованием функции захвата.



Tk_OC – выход генерируемого сигнала (см. рис. 9.2) k -го таймера
 Tn_IC – вход захвата n -го таймера
 Tn_CK – тактовый сигнал счетчика n -го таймера
 Tn_ICF – признак события «Захват содержимого счетчика n -го таймера»
 Tn_ICR – содержимое регистра захвата n -го таймера
 $CNTn(t1)$ и $CNTn(t2)$ – содержимое счетчика n -го таймера в моменты времени $t1$ и $t2$ соответственно

Рис. 9.27. Временные диаграммы, поясняющие способ преобразования частоты в код посредством таймеров МК с использованием функции захвата (см. пояснения в тексте)

Реализация данного способа, как и описанных в подпунктах 9.4.2.3 и 9.4.2.4, требует использования 2-х таймеров. Один из них, обозначенный на рис. 9.27 как некоторый k -й таймер МК, выполняет функцию формирования импульса длительностью Δt_R на выходе OC (см. рис. 9.2). Генерация данного импульса может осуществляться в режимах ШИМ или одновибратора (см. пункты 9.3.3 и 9.3.6).

Выходной сигнал k -го таймера поступает на вход захвата другого таймера, обозначенного на рис. 9.27 как n -й. Под входом захвата при этом подразумевается вывод порта МК, перепад сигнала на котором служит событием, вызывающим загрузку содержимого счетчика n -го таймера в его регистр захвата.

Счетчик n -го таймера работает в режиме непрерывного прямого счета, и тактируется импульсами с частотой f_x , подлежащей преобразованию в код. Загрузка его содержимого в регистр захвата n -го таймера происходит в моменты $t1$ и $t2$, по фронту и по спаду

импульса длительностью Δt_R на входе захвата. При этом (см. рис. 9.27):

- если $CNT(t2) > CNT(t1)$:

$$CNT(t2) - CNT(t1) = f_X \Delta t_R \pm 1; \quad (9.25)$$

- если $CNT(t2) \leq CNT(t1)$:

$$2^{N_c} - [CNT(t1) - CNT(t2)] = f_X \Delta t_R \pm 1; \quad (9.26)$$

где N_c – разрядность счетчика.

Значение преобразуемой частоты рассчитывается по выражениям:

$$f_X = \begin{cases} \frac{CNT(t2) - CNT(t1)}{\Delta t_R} & \text{при } CNT(t2) > CNT(t1); \\ \frac{2^{N_c} - [CNT(t1) - CNT(t2)]}{\Delta t_R} & \text{при } CNT(t2) \leq CNT(t1). \end{cases} \quad (9.27)$$

Относительная погрешность преобразования равна $\pm 1/(f_X \Delta t_R)$.

Обобщенный алгоритм преобразования частоты в код по данному способу следующий (см. рис. 9.27).

1. Сконфигурировать n -й таймер на режим непрерывного прямого счета. Выбрать источник импульсов с частотой f_X в качестве источника тактирования n -го таймера. Разрешить функцию захвата и назначить перепад из нуля в единицу на входе IC в качестве события, инициирующего захват. По возможности – разрешить прерывания по захвату.

2. Запустить счетчик n -го таймера.

3. Сконфигурировать k -й таймер на генерацию импульса (импульсов) с длительностью Δt_R , обеспечивающей требуемую погрешность преобразования (см. пояснение к выражению (9.27), а также выражения (9.6), (9.8) и (9.19)). В качестве источника тактирования счетчика k -го таймера необходимо выбрать генератор на основе ПЭР (см. пункт 4.2.3).

4. Установить счетчик k -го таймера в нулевое состояние, после чего запустить его.

5. По прерыванию, вызванному захватом содержимого счетчика n -го таймера или (если прерывание не разрешено) по установке в активное состояние признака захвата, ICF (см. рис. 9.27) – сбросить данный признак и считать содержимое регистра захвата, равное содержимому счетчика в момент $t1$ поступления фронта импульса, задающего интервал счета. В МК семейства AVR признак ICF будет сброшен автоматически, если обслуживание события захвата происходит по прерыванию.

6. Переназначить событие, инициирующее захват, выбрав в его качестве перепад из единицы в ноль на входе IC .

7. По вызванному очередным захватом прерыванию или по очередной установке в активное состояние признака захвата (см. рис. 9.27, а также пункт 5) – сбросить признак и считать содержимое регистра захвата, равное при этом содержимому счетчика в момент $t2$ поступления спада импульса, задающего интервал счета.

8. Рассчитать значение f_x по выражению (9.27).

Основными преимуществами данного способа преобразования частоты в код (по сравнению со способами, описанными в подпунктах 9.4.2.3 и 9.4.2.4) являются:

- высокая точность формирования интервала счета;
- минимальная вероятность возникновения погрешности от не синхронизации начала и окончания интервала счета с начальной фазой счетных импульсов; из рис. 9.26 и 9.27 нетрудно увидеть, что данная погрешность возникнет только при совпадении во времени активного перепада счетного импульса (т. е. перепада, которым тактируется счетчик) с фронтом или спадом импульса, задающего интервал счета);
- независимость вероятности возникновения погрешности не синхронизации от скважности счетных импульсов, а также от того, какой из их перепадов является активным (т. е. тактируется ли счетчик их фронтом или спадом).

В справедливости последнего утверждения можно убедиться, например, по рис. 9.26. Из него нетрудно увидеть, что, если преобразование реализуется по вышеописанному способу, т. е.:

- счетные импульсы поступают на счетчик таймера постоянно, независимо от состояния сигнала, приведенного на 3-й диаграмме;

- «захват» содержимого счетчика происходит в моменты времени t_2 и t_1 ;

то разность содержимых счетчика, «захватываемых» в моменты времени t_2 и t_1 , будет равна 10-ти, независимо ни от того, тактируется ли он импульсами, приведенными на 1-й или на 2-й диаграмме, ни от того, тактируется ли он их фронтом или спадом.

Таким образом, описанный в данном подпункте способ наиболее предпочтителен для преобразования частоты в код с помощью таймеров МК.

9.4.3. Преобразование интервалов времени в цифровой код посредством таймеров МК

9.4.3.1. Наиболее распространенным на практике способом преобразования интервалов времени между 2-мя событиями (в т. ч. периода и длительности сигналов) в цифровой код с помощью таймеров МК является способ, использующий функцию захвата [38]. Обобщенный алгоритм преобразования интервала времени в код данным способом приведен в подпункте 9.4.2.5. Временные диаграммы поясняющие преобразование в код длительности и периода сигнала данным способом, приведены на рис. 9.28а и 9.28б соответственно. Условные обозначения на них аналогичны таковым на рис. 9.27.

Для преобразования длительности или периода сигнала требуется один таймер. Его счетчик работает в режиме непрерывного прямого счета и тактируется импульсами опорной частоты (f_R). Если сигнал, длительность или период которого преобразуются в код, является прямоугольным и совместимым по уровням нуля и единицы с портом МК – он подается непосредственно на вход захвата таймера. Если же сигнал не соответствует данным требованиям, он подключается ко входу захвата через компаратор (см. рис. 9.25 и пояснения к нему). ФКИ при этом, естественно, не требуется.

«Захват» содержимого счетчика происходит:

- при преобразовании в код длительности импульса – по его фронту и спаду;

- при преобразовании периода сигнала – по 2-м соседним во времени фронтам (как на рис. 9.28б) или спадам импульсного сигнала с тем же периодом.

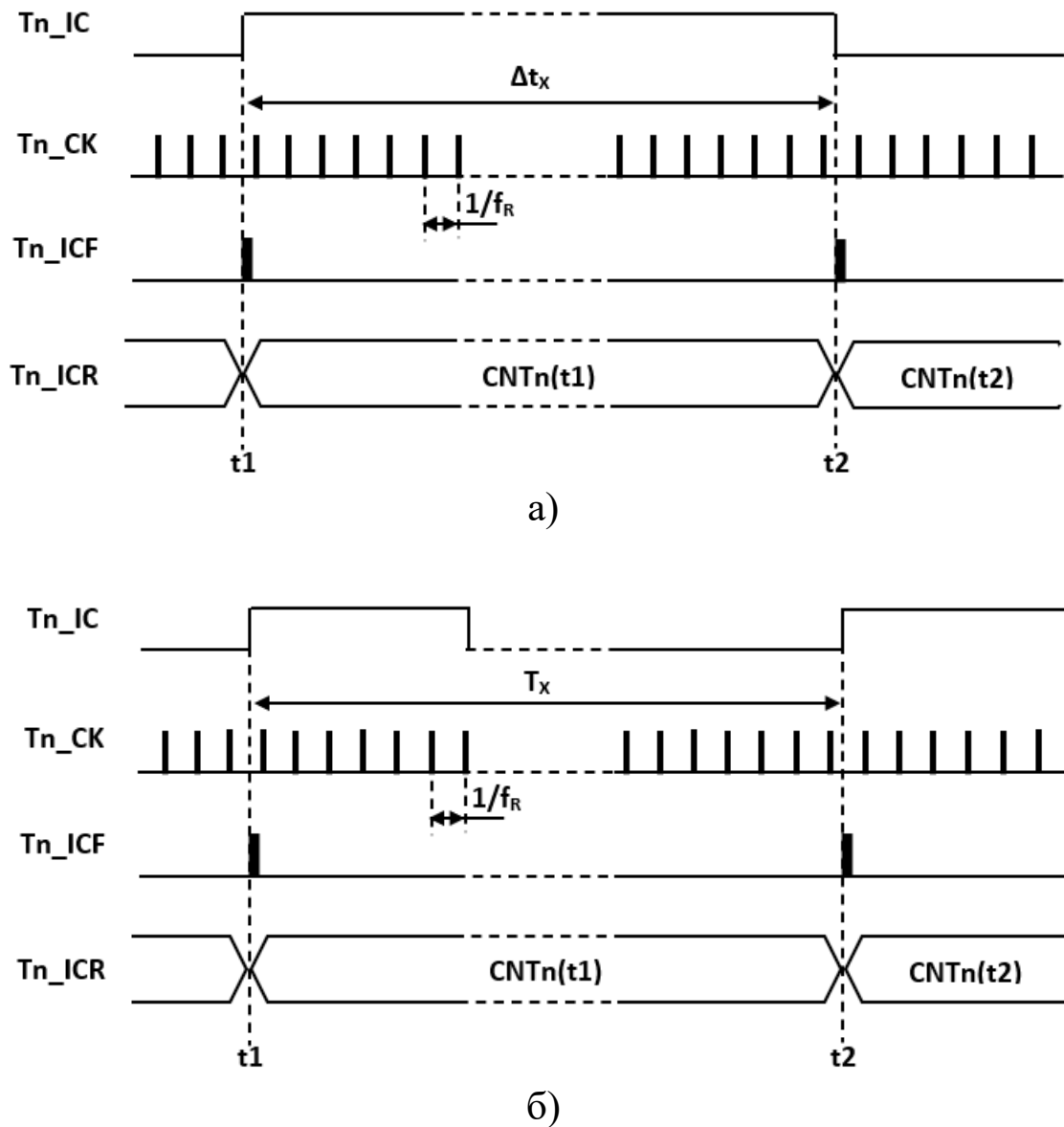


Рис. 9.28. Временные диаграммы, поясняющие преобразование длительности импульса (а) и периода сигнала (б) в код посредством таймеров МК с использованием функции захвата (см. пояснения в тексте)

При преобразовании в код длительности импульса (см. рис. 9.28а):

- если $CNT(t2) > CNT(t1)$:

$$CNT(t2) - CNT(t1) = f_R \Delta t_x \pm 1; \quad (9.28)$$

- если $CNT(t2) \leq CNT(t1)$:

$$2^{Nc} - [CNT(t1) - CNT(t2)] = f_R \Delta t_X \pm 1; \quad (9.29)$$

где Nc – разрядность счетчика.

Длительность импульса рассчитывается по выражениям:

$$\Delta t_X = \begin{cases} \frac{CNT(t2) - CNT(t1)}{f_R} & \text{при } CNT(t2) > CNT(t1); \\ \frac{2^{Nc} - [CNT(t1) - CNT(t2)]}{f_R} & \text{при } CNT(t2) \leq CNT(t1). \end{cases} \quad (9.30)$$

При преобразовании в код периода сигнала, в выражениях (9.28) – (9.30) Δt_X заменяется на T_X (см. рис. 9.28б).

Относительная погрешность преобразования равна $\pm 1/(f_R \Delta t_X)$ или, соответственно, $\pm 1/(f_R T_X)$.

Обобщенный алгоритм преобразования длительности или периода сигнала в код по данному способу следующий (см. рис. 9.28).

1. Сконфигурировать таймер на режим непрерывного прямого счета. В качестве источника тактирования его счетчика выбрать генератор на основе ПЭР (см. пункт 4.2.3). Выбрать частоту счетных импульсов (f_R), обеспечивающую требуемую погрешность преобразования (см. выше).

2. Разрешить функцию захвата и назначить в качестве события, инициирующего захват:

- при преобразовании в код интервала времени между фронтом и спадом импульса (как на рис. 9.28а) – фронт сигнала на входе IC ;

- при преобразовании интервала между спадом и фронтом импульса – спад сигнала на входе IC ;

- при преобразовании в код периода сигнала – или фронт (как на рис. 9.28б), или спад на входе IC .

3. По возможности – разрешить прерывания по захвату. Если преобразуется в код длительность одиночных или редко повторяющихся импульсов – разрешить прерывания по захвату крайне желательно.

4. Запустить счетчик таймера.

5. По прерыванию, вызванному захватом содержимого счетчика таймера или (если прерывание не разрешено) по установке в активное состояние признака захвата, ICF (см. рис. 9.28) – сбросить данный признак и считать содержимое регистра захвата, равное содержимому счетчика в момент $t1$. В МК семейства AVR признак будет сброшен автоматически, если обслуживание события захвата происходит по прерыванию.

6. При преобразовании в код длительности сигнала – переназначить событие, инициирующее захват, на противоположное ему (в примере, приведенном на рис. 9.28а, назначить спад сигнала на входе IC в качестве события захвата). При преобразовании периода сигнала переназначение не требуется.

7. По вызванному очередным захватом прерыванию или по очередной установке в активное состояние признака захвата (см. рис. 9.28, а также пункт 5) – сбросить признак и считать содержимое регистра захвата, равное при этом содержимому счетчика в момент $t2$.

8. Рассчитать значение Δt_x или, соответственно, T_x по выражению (9.30).

Следует заметить, что вышеприведенный алгоритм реализуем только при длительности или периоде сигнала, превышающих затраты времени на выполнение пунктов 5 – 7.

См. также **вариант** алгоритма преобразования Δt_x в код, описанный в подпункте 9.5.4.4.

9.4.3.2. В принципе, посредством таймера МК возможно преобразование длительности или периода сигнала в код на основе «классической» схемы, приведенной на рис. 9.24. При этом подсчет импульсов осуществляется счетчиком таймера, а в качестве источника импульсов опорной частоты служит один из ГТИ МК, выходной сигнал которого выводится на определенный пин одного из портов МК, одной из альтернативных функций которого является «*Clock Out*». Однако, такой способ мало распространен на практике, и более-менее рационален для применения только при отсутствии возможности использования таймеров, реализующих функцию захвата.

9.4.3.3. Отдельного рассмотрения требует случай преобразования в код интервала времени между 2-мя событиями, признаками которых служат два физически раздельных сигнала (например, импульсы «Старт» и «Стоп»). В принципе, из них может быть сформирован один импульс длительностью Δt_x , например, с помощью *RS*-триггера, устанавливаемого старт-импульсом и сбрасываемого импульсом «Стоп». Длительность сформированного импульса, в свою очередь, преобразуется в код способом, описанным в подпункте 9.4.3.1. Однако, такой подход не рационален, т. к. противоречит общей тенденции к минимизации числа компонентов, внешних по отношению к МК. Его использование допустимо только при невозможности использования способа, описанного в подпункте 9.4.3.4.

9.4.3.4. Более предпочтительны способы преобразования в код интервала времени между событиями, не требующие дополнительных внешних компонентов. Наиболее рациональным из них представляется способ, основанный на применении таймеров с многоканальным блоком захвата. К ним относятся, например, таймеры МК семейства *ARM Cortex-Mx*.

Временные диаграммы, поясняющие данный способ, приведены на рис. 9.29. Для его реализации требуется наличие в составе таймера, по крайней мере, 2-х независимых каналов захвата, каждый из которых снабжен собственным регистром захвата (у таймеров МК семейства *ARM Cortex-Mx* таких каналов, как правило, четыре [13 – 15]). На вход захвата одного из них (в приведенном на рис. 9.29 примере - k -го) должен поступать старт-импульс, другого (на рис. 9.29 – m -го) – импульс «Стоп». Каналы должны быть сконфигурированы на захват по соответствующим импульсам. Важно, чтобы по обоим каналам захват производился по одному и тому же перепаду импульса (или по фронту, или по спаду). Предпочтительно, чтобы по обоим каналам были разрешены прерывания по событию «Захват».

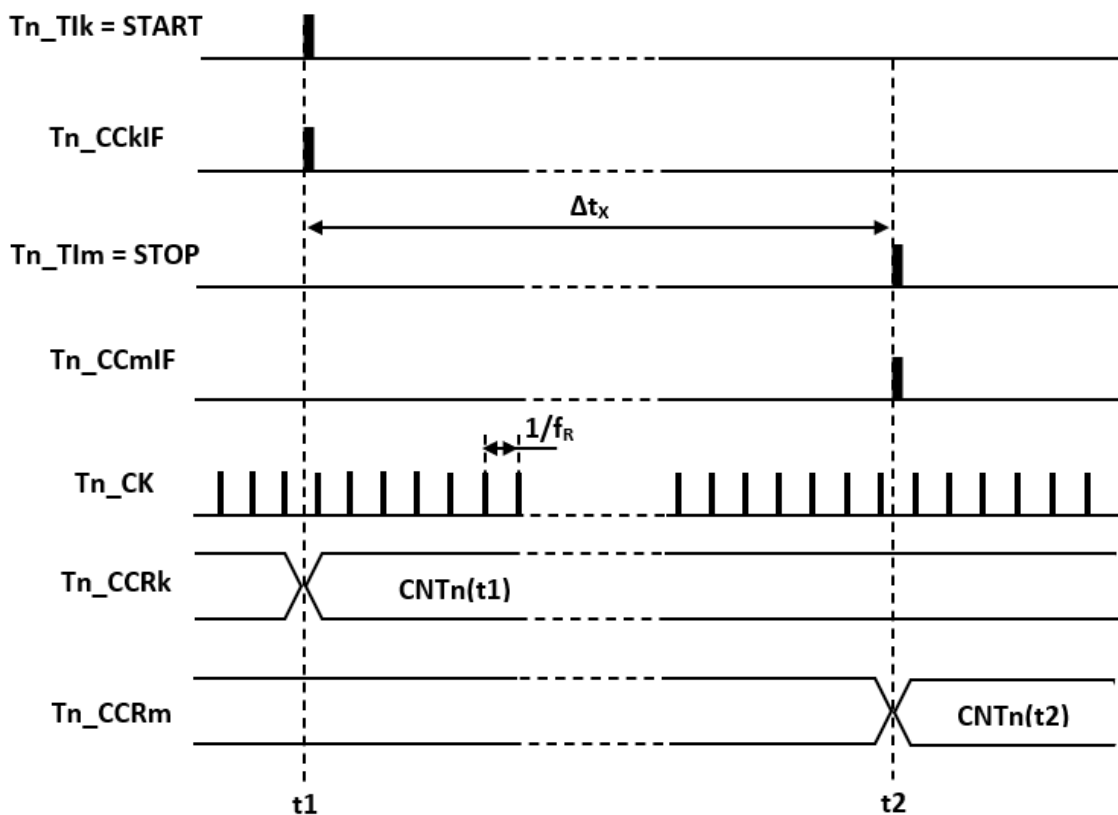
Обобщенный алгоритм преобразования при этом следующий.

1. Сконфигурировать каналы таймера в соответствии с приведенными выше требованиями.

2. Выбрать частоту счетных импульсов (f_R), обеспечивающую требуемую погрешность преобразования (см. пояснения к выражению (9.30)).

3. Запустить счетчик таймера.

4. По прерыванию, вызванному захватом содержимого счетчика по k -му каналу или (если прерывание не разрешено) по установке в активное состояние признака захвата по данному каналу, Tn_CCkIF (см. рис. 9.29) – сбросить признак Tn_CCkIF и считать содержимое регистра захвата k -го канала (Tn_CCRk), равное содержимому счетчика в момент $t1$.



Tn_Tlk и Tn_Tlm – сигналы на k -м и m -м входах захвата n -го таймера
 Tn_CCkIF и Tn_CCmIF – признаки события захвата по k -му и m -му входам
 n -го таймера

Tn_CK – сигнал тактирования n -го таймера

Tn_CCRk и Tn_CCRm – содержимое k -го и m -го регистров захвата / сравнения
 n -го таймера

Рис. 9.29. Временные диаграммы, поясняющие преобразование интервала времени между событиями в код посредством таймеров МК с многоканальным блоком захвата (на примере МК семейства *ARM Cortex-Mx*)

5. По прерыванию, вызванному захватом содержимого счетчика по m -му каналу или (если прерывание не разрешено) по установке в активное состояние признака захвата по данному каналу, Tn_CCmIF (см. рис. 9.29, а также пункт 4) – сбросить признак Tn_CCmIF и считать содержимое регистра захвата m -го канала (Tn_CCRm), равное содержимому счетчика в момент t_2 .

6. Рассчитать значение Δt_x по выражению (9.30).

Естественно, реализация вышеприведенного алгоритма возможна только при длительности преобразуемого интервала времени между событиями, превышающей затраты времени на выполнение пунктов 4 и 5.

9.4.3.5. Описанный в подпункте 9.4.3.4 способ, вообще говоря, не реализуем на таймере с только одним каналом захвата, что характерно, например, для таймеров МК семейства *AVR* [6, 8]. Однако, если в структуре МК имеется, как минимум, два таймера с одним каналом захвата в каждом (например, таким является МК 1887BE7T / *ATmega128* [8]), данный способ может быть реализован. При этом каналы захвата каждого из таймеров выполняет те же функции, что и каналы захвата таймера МК семейства *ARM Cortex-Mx* (см. подпункт 9.4.3.4). Алгоритм преобразования аналогичен приведенному в предыдущем подпункте.

9.4.3.6. На практике встречаются задачи преобразования в код относительно коротких интервалов времени, при котором не может быть обеспечена приемлемая погрешность преобразования даже при максимально возможной частоте f_R (см. выражение (9.24)). Однако, если существует возможность многократных преобразований с усреднением их результатов (например, в системах контроля динамических параметров ИС), погрешность, ввиду ее случайного характера, может быть уменьшена в $\sqrt{n}$ раз, где n – число усредняемых результатов преобразований [53]. См. Пример 4 в подпункте 9.5.4.4.

9.5 Типовые структурно-архитектурные решения таймеров МК общего назначения

Типовые структурно-архитектурные решения таймеров, как и других подсистем / функциональных блоков МК (см. разделы 3 – 8)

будем рассматривать на примерах МК семейств *AVR* и *ARM Cortex-Mx*. В целом, структура и архитектура таймеров обоих данных семейств МК характеризуется значительным сходством и, в свою очередь, является типовой для всех семейств МК классов «*mainstream*» и «*high performance*».

9.5.1. Базовые структурно-архитектурные решения таймеров МК семейства *AVR*

9.5.1.1. В состав большинства моделей МК подсемейства *Atmega* входят три основных разновидности таймеров общего назначения [6, 8]:

- 8-битовые таймеры с функцией ШИМ;
- 8-битовые таймеры с функцией ШИМ и с возможностью тактирования от специально выделяемого ГТИ, в документации на МК семейства *AVR* называемого асинхронным (т. е. не синхронизированным с основным ГТИ МК);
- 16-битовые таймеры с функциями ШИМ и захвата.

Типовые структурные схемы перечисленных разновидностей таймеров (на примере МК 1887BE7T / *Atmega128* [8]) приведены на рис. 9.30 – 9.32.

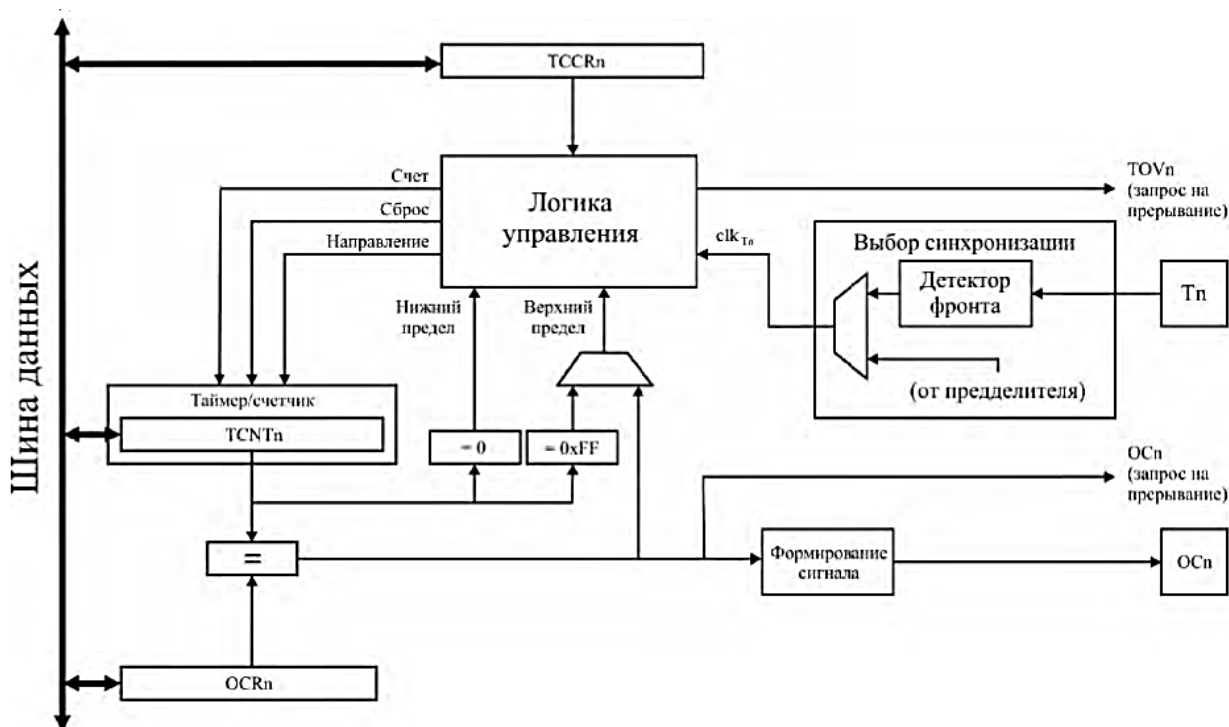


Рис. 9.30. Структурная схема 8-битового таймера с функцией ШИМ МК 1887BE7T / ATmega128 [8] (пояснения – в тексте)

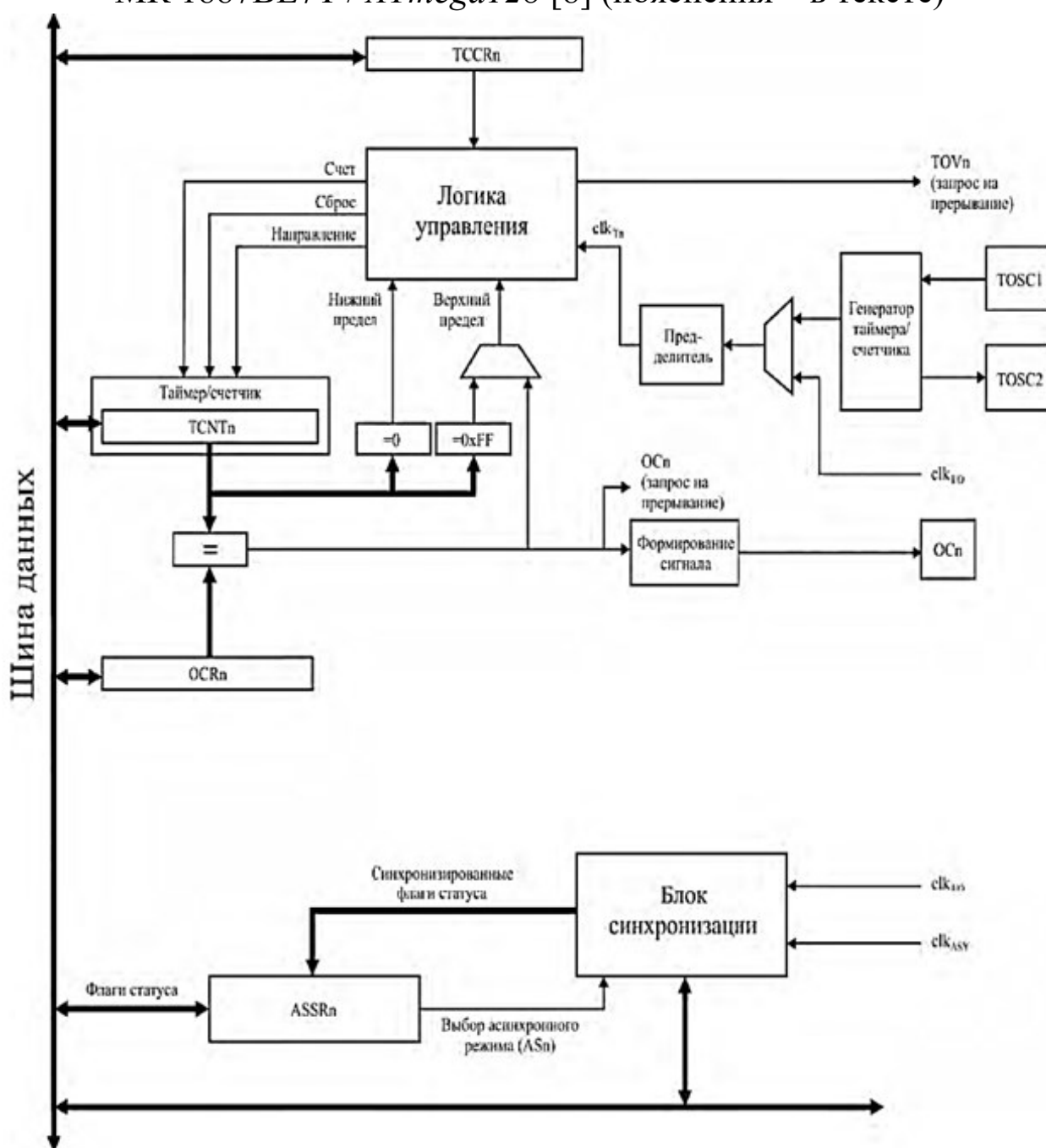


Рис. 9.31. Структурная схема 8-битового таймера МК 1887BE7T / ATmega128 с функцией ШИМ и с возможностью тактирования от асинхронного ГТИ [8] (пояснения – в тексте)

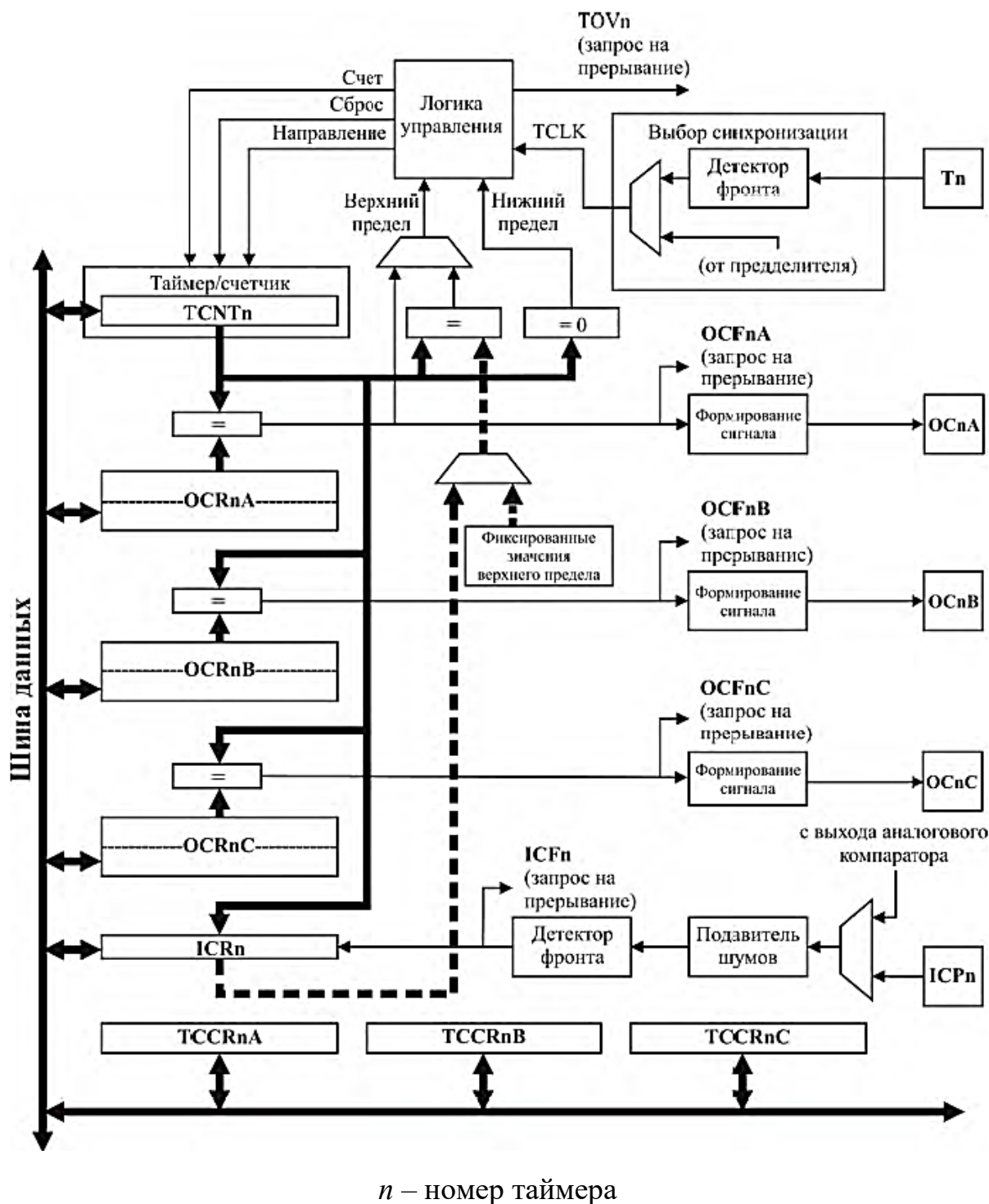


Рис. 9.32. Структурная схема 16-битового таймера МК 1887BE7T / *ATmega128* с функциями ШИМ и захвата [8]
(пояснения – в тексте)

В структуре МК подсемейства *ATtiny*, как правило, присутствуют один – два 8-битовых таймера, реализованных по структурной схеме, аналогичной приведенной на рис. 9.30 [7].

9.5.1.2. Основой каждого из таймеров служит двоичный счетчик (*TCNT*), соответственно 8- или 16-битовый. В схемах, приведенных на рис. 9.30 и 9.32, тактирование счетчика может осуществляться или от предделителя (*Prescaler*) частоты внутреннего синхросигнала МК, или от внешнего (по отношению к МК) источника тактовых импульсов, подаваемых на вход *Tn*. В схеме, представленной на рис. 9.31, счетчик тактируется от предделителя, входным сигналом которого может служить или сигнал собственного (асинхронного) ГТИ таймера, или синхросигнал ПУ МК, *clk_{I/O}* (см. рис. 4.14), частота которого совпадает с тактовой частотой МК. При этом асинхронный ГТИ строится на основе внешнего ПЭР, подключаемого к выводам *TOSC1* и *TOSC2*. Как правило, тактирование от асинхронного ГТИ используется при необходимости формирования интервалов времени или импульсов с относительно большими длительностью / периодом, которые не могут быть обеспечены при тактировании сигналом *clk_{I/O}*. Типовая резонансная частота ПЭР асинхронного ГТИ – 32768 Гц; функционально он эквивалентен генератору LSE подсистемы синхронизации МК семейства *ARM Cortex-Mx* (см. рис. 4.1).

Выбор источника тактирования счетчика и коэффициента деления предделителя осуществляется программно, битовым полем *CS* (*Clock Select*) регистров управления (*TCCR*), в таймере с возможностью тактирования от асинхронного ГТИ – также битом *AS* (*Asynchronous*) регистра *ASSR*. Остановка счетчика производится обнулением поля *CS*. Подробнее – см. рис. 9.34, 9.35 и табл. 9.2.

Счетчики таймеров программно-доступны и для записи, и для чтения, что позволяет реализовывать как описанный в подразделе 9.2 алгоритм формирования задержек между событиями, так и представленные в пунктах 9.4.2 и 9.4.3 способы преобразования в код частоты и длительности интервала времени.

Направление счета (прямой или обратный) задается логикой управления, в зависимости от режима работы. В отличие от таймеров МК семейства *ARM Cortex-Mx*, счетчики таймеров семейства *AVR* работают в направлении обратного счета (поочередно с прямым) только в режиме центрированной ШИМ (см. рис. 9.10); во всех других режимах возможен только прямой счет.

9.5.1.3. Все таймеры МК семейства *AVR* позволяют формировать ШИМ- и ЧИМ-сигналы (см. пункты 9.3.3 и 9.3.4): 8-битовые – по одному каналу, 16-битовые – как правило, по трем каналам (см. рис. 9.30 – 9.32). Формирование осуществляется в соответствии с представленной на рис. 9.2 обобщенной структурной схемой. Каждый из каналов снабжен собственным регистром сравнения (*Output Compare Register, OCR*), программно-доступным как для записи, так и для чтения. Данные регистры буферизированы (см. рис. 9.7 и 9.10, а также поясняющий их текст).

9.5.1.4. 16-битовые таймеры МК семейства *AVR* также позволяют реализовывать функцию захвата (см. подпункт 9.4.2.5). Канал захвата – один на таймер. Событием захвата может служить перепад сигнала на входе *ICP (Input Capture)* таймера или на выходе встроенного аналогового компаратора МК (см. рис. 9.32). Выбор события захвата, а также направления перепада сигнала, инициирующего захват (фронт или спад) осуществляется программно (см., в частности, пункты 1 и 6 алгоритма преобразования частоты в код, приведенного в подпункте 9.4.2.6).

9.5.1.5. Основными режимами работы таймеров МК семейства *AVR* являются следующие [6, 8]:

- нормальный (*Normal*), в котором счетчик постоянно работает в режиме прямого счета, со сбросом в ноль и перезапуском по достижении верхнего предела счета, который равен $2^8 - 1$ (255) у 8-битовых таймеров и $2^{16} - 1$ (65535) – у 16-битовых;

- *Clear Timer on Compare Match, CTC* – сброс счетчика таймера по совпадении с содержимым регистра сравнения (в 16-битовых таймерах – регистра сравнения *A* или регистра захвата); данный режим используется, в основном, при реализации ЧИМ (см. подпункт 9.3.4.2);

- асимметричная, «быстрая» ШИМ (*Fast PWM*), см. подпункт 9.3.3.2;

- центрированная, «корректная по фазе» ШИМ (*Phase Correct PWM*), см. подпункт 9.3.3.3.

В 8-битовых таймерах возможны только 4 режима работы – *Normal*; *CTC* с *OCR* в качестве регистра сравнения; *Fast PWM* и

Phase Correct PWM с верхним пределом счета (*TOP*, см. рис. 9.7 и 9.10), равным $0xFF$, т. е. 255.

В 16-битовых таймерах возможна реализация [6, 8]:

- 2-х вариантов режима *CTC*, использующих в качестве регистра сравнения соответственно *OCRnA* или регистр захвата;

- нескольких вариантов *Fast PWM* и *Phase Correct PWM*, различающихся между собой значением верхнего предела счета; в частности, возможны *Fast PWM* и *Phase Correct PWM* с *TOP*, равным $0xFF$ (255), $0x1FF$ (511), $0x3FF$ (1023), а также содержимому *OCRnA* или регистра захвата.

Выбор режима работы таймера осуществляется программно, битовым полем *WGM* (*Waveform Generation Mode*) регистров управления (*TCCR*); см. рис. 9.34, 9.35 и табл. 9.3.

9.5.1.6. Режим переключения формирователя выходного сигнала также задается программно, битовым полем *COM* (*Compare Match Output Mode*) регистров управления; при наличии нескольких каналов формирования выходных сигналов (см. рис. 9.32) – индивидуальным битовым полем *COM* каждого из каналов. На практике обычно применяются следующие режимы переключения:

- при формировании ЧИМ-сигнала – режим «*Toggle*», т. е. переключения состояния выхода на противоположное в момент совпадения содержимых счетчика и регистра сравнения (см. рис. 9.17 и поясняющий его текст);

- при формировании не инвертированного сигнала «быстрой» ШИМ (см. рис. 9.7) – переключение выхода в единичное состояние по сбросу счетчика, в нулевое – по совпадении содержимых счетчика и регистра сравнения;

- при формировании инвертированного сигнала «быстрой» ШИМ – переключение выхода в нулевое состояние по сбросу счетчика, в единичное – по совпадении содержимых счетчика и регистра сравнения (см. рис. 9.7);

- при формировании не инвертированного сигнала центрированной ШИМ (см. рис. 9.10) – переключение из нуля в единицу по совпадении содержимых счетчика и регистра сравнения на интервале обратного счета; переключение из единицы в ноль – по аналогичному событию на интервале прямого счета;

- при формировании инвертированного сигнала центрированной ШИМ (см. рис. 9.10) – переключение из нуля в единицу по совпадении содержимых счетчика и регистра сравнения на интервале прямого счета; переключение из единицы в ноль – по аналогичному событию на интервале обратного счета.

Подробнее – см. рис. 9.34, 9.35 и табл. 9.4.

9.5.1.7. Таймеры МК семейства *AVR* могут служить источниками запросов на прерывания (при их разрешении) по следующим событиям (см. рис. 9.30 – 9.32):

- переполнение счетчика (*TOV*), т. е. по его сбросу после достижения им верхнего предела счета (см. пример на рис. 9.33);
 - совпадение содержимых счетчика и регистра сравнения (*OC*);
- в таймерах с несколькими каналами формирования сигналов запросы на прерывания формируются независимо по каждому из каналов (см. рис. 9.32);
- захват содержимого счетчика регистром *ICR* (*IC*).



Рис. 9.33. Формирование запроса на прерывание по переполнении счетчика таймера МК 1887BE7T / *Atmega128* при прямом счете [8] (коэффициент деления предделителя равен 8-и)

Разрешение / запрет прерываний по вышеперечисленным событиям осуществляется соответствующими битами регистра маскирования прерываний таймера (*Timer / Counter Interrupt Mask Register, TIMSK*). События, являющиеся потенциальными источниками прерываний, фиксируются в регистре признаков («флагов») прерываний соответствующего таймера (*Timer / Counter Interrupt Flag Register, TIFR*). По состоянию его битов может быть

выполнена проверка наличия / отсутствия соответствующих событий, если прерывания по ним запрещены по каким-либо причинам. Подробнее – см. подпункт 9.5.1.11.

9.5.1.8. Регистровые модели таймеров МК семейства *AVR* включают в себя (см. рис. 9.30 – 9.32):

- 8- или 16-битовый счетчик (*TCNT*);
- регистр (регистры) сравнения (*OCR*);
- регистр захвата, *ICR* (в 16-битовых таймерах МК подсемейства *ATmega*, а также в 8-битовых таймерах ряда моделей подсемейства *ATtiny* [7]);
- регистр (регистры) управления таймером (*TCCR*);
- регистры маскирования прерываний (*TIMSK*) и «флагов» прерываний (*TIFR*); в ряде моделей МК регистры *TIMSK* и *TIFR* являются общими для всех таймеров; для каждого из них назначены отдельные биты в данных регистрах (см. рис. 9.37).

9.5.1.9. На рис. 9.34 в качестве примера представлен формат регистра управления 8-битового таймера с функцией ШИМ (2-го таймера) МК 1887BE7T / *ATmega128* [8], структурная схема которого приведена на рис. 9.30.

Бит	7	6	5	4	3	2	1	0	
	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	TCCR2
Доступ	3	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	
Начальное значение	0	0	0	0	0	0	0	0	

Рис. 9.34. Формат регистра управления 2-го таймера (*TCCR2*) МК 1887BE7T / *ATmega128* [8] (пояснения приведены в тексте)

Принцип обозначения битов данного регистра (как и битов всех регистров каждого из таймеров) следующий: буквенная аббревиатура указывает на назначение бита или битового поля, 1-я цифра – номер таймера (в рассматриваемом примере – 2-й), 2-я (при наличии) – номер бита в соответствующем битовом поле. **На рис. 9.34 и далее:** Ч/З – бит доступен и для чтения, и для записи, Ч – только для чтения, З – только для записи.

Биты с 0-го по 2-й (*CS, Clock Select*) регистра управления 2-м таймером задают источник и частоту тактирования, в соответствии с табл. 9.2.

Таблица 9.2

Источник и частота тактирования счетчика 2-го таймера МК 1887BE7T / ATmega128 в зависимости от битового поля CS регистра управления [8]

CS22	CS21	CS20	Источник и частота тактирования
0	0	0	Счетчик остановлен
0	0	1	$clk_{I/O}$
0	1	0	$clk_{I/O}/8$
0	1	1	$clk_{I/O}/64$
1	0	0	$clk_{I/O}/256$
1	0	1	$clk_{I/O}/1024$
1	1	0	Спад импульса на внешнем входе T_n (см. рис. 9.30)
1	1	1	Фронт импульса на внешнем входе T_n (см. рис. 9.30)

Биты 3-й и 6-й (*WGM, Waveform Generation Mode*) регистра управления определяют режим работы таймера, в соответствии с табл. 9.3.

Таблица 9.3

Режим работы 2-го таймера МК 1887BE7T / ATmega128 в зависимости от битового поля WGM регистра управления [8]

WGM21	WGM20	Режим работы
0	0	Нормальный
0	1	Центрированная ШИМ (<i>Phase Correct PWM</i>)
1	0	<i>CTC</i>
1	1	Асимметричная ШИМ (<i>Fast PWM</i>)

Биты 4-й и 5-й (*COM, Compare Match Output Mode*) регистра управления задают режим переключения формирователя выходного сигнала, в соответствии с табл. 9.4.

7-й бит (*FOC, Force Output Compare*), доступный только для записи, служит для принудительного переключения выхода в состояние, в которое он перешел бы по совпадении содержимых счетчика и регистра сравнения, в соответствии с состоянием битового поля *COM* (см. табл. 9.4). Переключение производится

записью единицы в бит *FOC*. Данная функция может быть использована, например, для принудительного отключения или включения некоторого ИУ в аварийной ситуации. Запись единицы в бит *FOC* не вызывает эффекта в режимах ШИМ.

Таблица 9.4

Режим переключения формирователя выходного сигнала 2-го таймера МК 1887BE7T / ATmega128 в зависимости от битового поля COM регистра управления [8]

<i>COM21</i>	<i>COM20</i>	Режим работы (см. также подпункт 9.5.1.6)
0	0	Выход формирователя отключен от вывода ПВВ
0	1	В режимах <i>Normal</i> и <i>CTC – Toggle</i> В режимах асимметричной и центрированной ШИМ – не используется (зарезервировано)
1	0	В режимах <i>Normal</i> и <i>CTC</i> – сброс выхода в ноль по совпадении содержимых счетчика и регистра сравнения* В режимах асимметричной и центрированной ШИМ – формирование не инвертированного ШИМ-сигнала
1	1	В режимах <i>Normal</i> и <i>CTC</i> – установка выхода в единицу по совпадении содержимых счетчика и регистра сравнения* В режимах асимметричной и центрированной ШИМ – формирование инвертированного ШИМ-сигнала
*Может использоваться при формировании интервалов времени между включениями / выключениями ИУ.		

В состав 16-битовых таймеров МК семейства *AVR*, как правило, входит несколько регистров управления (см. рис. 9.32). На рис. 9.35 в качестве типового примера приведены форматы регистров управления 1-го таймера МК 1887BE7T / *ATmega128* [8]. Наличие трех регистров управления вместо одного обусловлено, в первую очередь:

- наличием трех каналов формирования сигналов (*A*, *B* и *C*, см. рис. 9.32), для каждого из которых выделено 2-битовое поле управления формирователем сигнала (*COM1A*, *COM1B* и *COM1C* соответственно); при этом режимы переключения по каналам в зависимости от состояния данных битовых полей также определяются табл. 9.4;

- наличием не 4-х, а 16-ти режимов работы таймера (см. подпункт 9.5.1.5), для задания которых требуется 4-битовое поле *WGM*, 2 младших бита которого расположены в регистре *TCCR1A*, два старших – в регистре *TCCR1B*; таблица соответствия режимов работы таймера состоянию данных битов приведена в [8];

- необходимостью введения битов принудительного переключения по всем 3-м выходам (см. рис. 9.35в).

Бит	7	6	5	4	3	2	1	0
	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10
Доступ	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З
Начальное значение	0	0	0	0	0	0	0	0

а)

Бит	7	6	5	4	3	2	1	0
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
Доступ	Ч/З	Ч/З	Ч	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З
Начальное значение	0	0	0	0	0	0	0	0

б)

Бит	7	6	5	4	3	2	1	0
	FOC1A	FOC1B	FOC1C	-	-	-	-	-
Доступ	З	З	З	Ч	Ч	Ч	Ч	Ч
Начальное значение	0	0	0	0	0	0	0	0

в)

Рис. 9.35. Форматы регистров управления *TCCR1A* (а), *TCCR1B* (б) и *TCCR1C* (в) 1-го таймера МК 1887BE7T / *ATmega128* [8]
(пояснения приведены в тексте)

Также, поскольку данный таймер поддерживает функцию захвата, в регистр *TCCR1B* введены:

- бит *ICES1*, определяющий, по какому из перепадов (фронту или спаду) сигнала на входе *ICP1* или на выходе компаратора (см. рис. 9.32) происходит захват;

- бит *ICNC1*, управляющий включением / отключением подавителя шумов (см. рис. 9.32).

Выбор сигнала, инициирующего захват, производится битом *AC1C* в регистре управления компаратором: при единичном

состоянии данного бита захват инициируется выходным сигналом компаратора, при нулевом – сигналом на входе *ICP1* [8].

Битовое поле *CS1* выполняет те же функции, что и аналогичное битовое поле в регистре управления 2-м таймером (см. рис. 9.34). Его содержимое задается также в соответствии с табл. 9.2.

9.5.1.10. В таймерах с возможностью тактирования от асинхронного ГТИ имеется дополнительный регистр управления и статуса асинхронного режима – *ASSR* (*Asynchronous Status Register*), см. рис. 9.31. Формат регистра *ASSR* 0-го таймера МК 1887BE7T / *ATmega128* [8] приведен на рис. 9.36.

Бит	7	6	5	4	3	2	1	0	
	-	-	-	-	AS0	TCN0UB	OCR0UB	TCR0UB	ASSR
Доступ	ч	ч	ч	ч	ч/з	ч	ч	ч	
Начальное значение	0	0	0	0	0	0	0	0	

Рис. 9.36. Формат регистра управления и статуса асинхронного режима 0-го таймера МК 1887BE7T / *ATmega128* [8]
(пояснения приведены в тексте)

Бит *AS0* задает источник тактирования таймера; при его нулевом состоянии на вход предделителя (см. рис. 9.31) поступает сигнал *clk_{I/O}*, а при нулевом – с выхода генератора 0-го таймера, не синхронизированного с основным ГТИ МК. Биты *TCN0UB* (*TCNT0 Update Busy*), *OCR0UB* (*OCR0 Update Busy*) и *TCR0UB* (*TCCR0 Update Busy*) доступны только для чтения. Они используются при тактировании таймера от ГТИ, не синхронизированного с основным. Нулевое состояние данных битов оповещает ЦП о доступности соответствующих регистров для обновления их содержимого.

9.5.1.11. Для большинства моделей МК семейства *AVR* характерно наличие одного регистра маскирования / разрешения прерываний и одного регистра «флагов» прерываний на все таймеры. Примеры форматов этих регистров приведены на рис. 9.37. На нем:

- *OCIE_n* (*n* = 0, 2) – бит разрешения прерываний по совпадении содержимых счетчика *n*-го таймера и регистра *OCR_n*;

- $OCFn$ ($n = 0, 2$) – бит признака совпадения содержимых счетчика n -го таймера и регистра $OCRn$;
- $OCIE1A$, $OCIE1B$ – биты разрешения прерываний по совпадении содержимых счетчика 1-го таймера и регистра $OCR1A$ или $OCIE1B$ (см. рис. 9.32);
- $OCF1A$, $OCF1B$ – биты признаков совпадения содержимых счетчика 1-го таймера и регистра $OCR1A$ или $OCIE1B$;
- $TOIE_n$ и TOV_n – бит разрешения прерываний по переполнении счетчика n -го таймера и бит признака переполнения соответственно;
- $TICIE_n$ и ICF_n – бит разрешения прерывания по захвату и бит признака захвата соответственно.

Примечание. Биты $OCIE1C$ и $OCF1C$ располагаются в регистрах $ETIMSK$ и $ETIFR$ соответственно [8].

Бит	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Доступ	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	
Начальное значение	0	0	0	0	0	0	0	0	

Бит	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Доступ	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	
Начальное значение	0	0	0	0	0	0	0	0	

Рис. 9.37. Форматы регистров маскирования и «флагов» прерываний таймеров МК 1887BE7T / *ATmega128* [8]
(пояснения приведены в тексте)

Разрешение прерывания какому-либо событию производится записью единицы в соответствующий бит регистра $TIMSK$ ($ETIMSK$). По наступлении события бит его признака устанавливается в единичное состояние. При переходе к подпрограмме обслуживания прерывания по соответствующему событию бит его признака сбрасывается автоматически. Если прерывание запрещено, сброс осуществляется записью единицы в данный бит. Напомним, что в МК семейства *AVR* каждому событию,

являющемуся потенциальным источником прерываний, назначается отдельная позиция в таблице векторов прерываний (см. пункт 7.2.1).

9.5.1.12. Необходимо также остановиться на вопросах записи и чтения 16-битовых регистров, входящих в состав таймеров МК семейства *AVR*.

МК данного семейства являются 8-битовыми, такова же разрядность и их шины данных. Для возможности чтения / записи 16-битовых регистров через 8-битовую шину, используется прием, поясняемый рис. 9.38 (на примере счетчика 16-битового таймера).

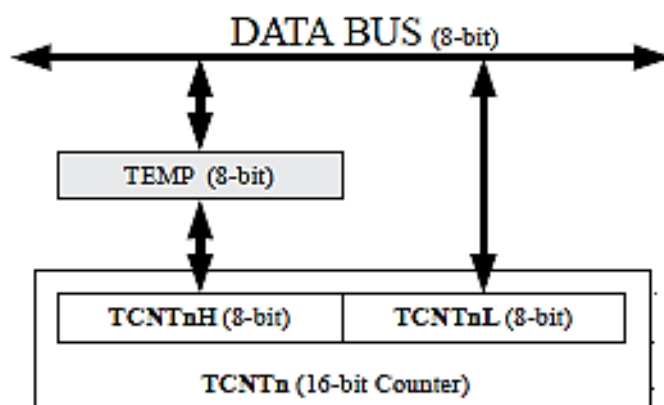


Рис. 9.38. Доступ к 16-битовым регистрам таймеров МК семейства *AVR* (на примере счетчика таймера). Пояснения приведены в тексте

16-битовые регистры разбиты на два 8-битовых, с собственными именами и адресами, программно-доступных независимо друг от друга. Например, в регистровых моделях 16-битовых таймеров их счетчики представлены 2-мя регистрами: *TCNTnH* и *TCNTnL* (соответственно старший и младший байт счетчика). Для обеспечения одновременного изменения обоих байтов при записи, а также считывания содержимого обоих байтов по состоянию на один и тот же момент времени, старший байт снабжен регистром временного хранения (*TEMP*). При этом:

- данные, записываемые в старший байт, реально загружаются в регистр *TEMP*, а при записи младшего байта, одновременно с ней, содержимое регистра *TEMP* загружается в старший байт;

- при считывании младшего байта одновременно происходит загрузка старшего в регистр *TEMP*; по команде чтения старшего байта реально считывается содержимое регистра *TEMP*.

Таким образом:

- для одновременного изменения содержимого и старшего, и младшего байта 16-битового регистра при записи, необходимо вначале выполнить команду записи старшего байта, затем – младшего;

- для считывания содержимого обоих байтов по состоянию на один и тот же момент времени, необходимо вначале выполнить команду чтения младшего байта, затем – старшего.

Отметим, что вышеописанный порядок команд при записи и чтения 16-битовых регистров должен соблюдаться в программах на языке Ассемблера МК семейства *AVR*. При разработке программ на языке *C* в *IDE* ПО МК данного семейства допускается загрузка и чтение 16-битовых регистров одной командой, как 16-битовых переменных. Однако, при компиляции *C*-программы данная команда транслируется в последовательность ассемблерных команд, реализующую запись / чтение регистра в соответствии с вышеописанным порядком. Например, следующая команда загрузки 16-битового счетчика 1-го таймера МК *ATmega128 / 1887BE7T*:

TCNT1 = 0xFEFC;

в *IDE Atmel Studio 7* транслируется в следующую последовательность ассемблерных команд (см. табл. 2.7):

```
//Загрузка в РОН R18 числа, подлежащего записи в младший байт счетчика  
LDI R18,0xFC  
// Загрузка в РОН R19 числа, подлежащего записи в старший байт счетчика  
LDI R19,0xFE  
//Вывод содержимого РОН R19 в старший байт счетчика  
OUT 0x2D,R19  
//Вывод содержимого РОН R18 в младший байт счетчика  
OUT 0x2C,R18
```

В свою очередь, команда присваивания некоторой переменной *x* значения, равного содержимому счетчика (16-битового) 1-го таймера МК *ATmega128 / 1887BE7T*:

x = TCNT1;

транслируется в следующую последовательность ассемблерных команд:

```
//Ввод младшего байта счетчика в РОН R18  
IN R18,0x2C  
//Ввод старшего байта счетчика в РОН R19  
IN R19,0x2D
```

При этом, например, *IDE Atmel Studio 7* допускает и обращение к отдельным байтам 16-битовых регистров:

```
TCNT1H = 0xFE;  
TCNT1L = 0xFC;
```

Приемы, аналогичные вышеописанным, используются и в других семействах МК при чтении и записи регистров, разрядность которых превышает разрядность шины данных МК.

9.5.2. Примеры программирования таймеров МК семейства AVR

В данном пункте приведены примеры программных модулей реализации наиболее распространенных типовых функций таймеров МК семейства AVR (см. пункт 9.1):

- формирования программно задаваемой задержки между двумя событиями;
- цифро-аналогового преобразования способом ШИМ;

9.5.2.1. Пример 1. Пусть необходимо, после включения некоторого реле, через $1\text{ с} \pm 50\text{ мс}$ включить другое реле. Модель МК – 1887BE7T. Включение 1-го реле производится логической единицей на 0-м выводе ПВВ А, 2-го – на 1-м выводе того же ПВВ. Тактирование МК производится ГТИ частотой 8 МГц с внешним ПЭР. Применение дополнительного низкочастотного ПЭР (см. рис. 9.31) или внешнего ГТИ таймеров условиями проектирования не предусмотрено.

Формирование интервала времени между включениями 1-го и 2-го реле будем реализовывать по алгоритму, представленному в подразделе 9.2.

1) В первую очередь, необходимо определить, посредством какого таймера, 8- или 16-битового, может быть сформирован интервал времени с длительностью 1 с при вышеприведенных условиях.

В рассматриваемом примере источником тактирования таймеров является ГТИ МК. Частота счетных импульсов определяется коэффициентом деления предделителя (*Prescaler*), см. рис. 9.30 – 9.32, который может принимать значения, приведенные в табл. 9.2 (у 2-го таймера, схема которого приведена на рис. 9.31 – также 32 и 128). Таким образом, в данном примере минимально возможная частота счетных импульсов составит 8 МГц / 1024, т. е. 7,8125 кГц, а их максимально возможный период – 128 мкс. Максимальная длительность интервала времени, который при этом может быть сформирован 8-битовым таймером, равна $128 \text{ мкс} \times 2^8$, т. е. 32,768 мс, а 16-битовым – $128 \text{ мкс} \times 2^{16}$, т. е. 8388,608 мс (примерно 8,389 с). Следовательно, для решения поставленной задачи необходим 16-битовый таймер. В состав МК 1887BE7T входит два таймера с такой разрядностью – 1-й и 3-й. Для определенности выберем 1-й таймер.

2) Далее следует выбрать конкретное значение коэффициента деления предделителя. С одной стороны, он должен удовлетворять условию: $2^{16} \times K_d / 8 \text{ МГц} > 1 \text{ с}$ (см. п. 1 алгоритма, приведенного в подразделе 9.2). С другой стороны, для обеспечения требуемой погрешности формирования интервала времени ($\pm 50 \text{ мс}$) необходимо удовлетворение условия: $K_d / 8 \text{ МГц} \leq \pm 50 \text{ мс}$ (см. Примечание к алгоритму, представленному в подразделе 9.2).

Первое из условий может быть удовлетворено при коэффициентах деления предделителя, равных 256-ти и 1024-м, второе – при обоих данных коэффициентах. Выберем коэффициент деления, равный 256-ти.

3) Расчет числа N , которое необходимо загрузить в счетчик таймера по включении 1-го реле, должен производиться по выражению (9.1), т. к. при реализации приведенного в подразделе 9.2 алгоритма счетчики таймеров МК семейства AVR могут работать только в режиме прямого счета. В рассматриваемом примере $N_{TOP} =$

$2^{16} - 1 = 65535$, $\Delta t = 1$ с, $f_{CNT} = 8 \text{ МГц}/256 = 31,25 \text{ кГц}$. В соответствии с выражением (9.1) получаем, что $N = 34285$ ($0x85ED$).

4) Далее представлены два варианта программного фрагмента, реализующего формирование интервала времени в соответствии с поставленной задачей, вышеприведенными расчетами и алгоритмом, представленным в подразделе 9.2. Первый из вариантов использует завершение формируемого интервала по прерыванию от таймера, второй – по результатам программного опроса признака переполнения счетчика (см. пункты 4 и 5 алгоритма, представленного в подразделе 9.2). *IDE – Atmel Studio 7*. Уровень программирования – регистровый, представление операндов – числовое (см. подпункт 2.6.1.14 и Приложение В).

Вариант 1 (с использованием прерывания от таймера)

```
//Подключение библиотеки моделей ПУ МК семейства AVR
#include <avr/io.h>
/*
Подключение библиотеки стандартных функций обработки прерываний МК
семейства AVR
*/
#include <avr/interrupt.h>
//Объявление переменных, используемых в программе
.
.
.
////////////////////////////////////
//Код основного блока программы
int main(void)
{
/*
Конфигурирование функциональных блоков МК, кроме 1-го таймера и ПВВ А
*/
.
.
.
/*
Конфигурирование 0-го и 1-го выводов ПВВ А на работу в режиме цифрового
двухтактного выхода, выполняемое записью единицы в 0-й и 1-й разряды
регистра DDRA (см. табл. 6.1)
*/
DDRA = 0x03;
```



```
/*
```

Обнуление бита признака переполнения (*TOV1*) в регистре *TIFR* (см. рис. 9.37) записью **единицы** в данный бит (см. подпункт 7.2.4.5).

Примечание. Вообще говоря, сброс должен происходить автоматически, при переходе к подпрограмме обслуживания прерывания по переполнению счетчика 1-го таймера. Однако, как показывает опыт, в «клонах» МК семейства *AVR* ряда производителей сброс признака события необходимо производить в подпрограмме обслуживания прерывания по данному событию. Необходимость программного сброса признака может быть выяснена только в процессе отладки ПО на конкретном образце МК

```
*/
```

```
    TIFR = TIFR|0x04;
```

```
/*
```

Остановка счетчика 1-го таймера, выполняемая обнулением битового поля *CS* регистра *TCCR1B* (см. рис. 9.35)

```
*/
```

```
    TCCR1B = 0b00000000;
```

```
//Включение второго реле записью единицы в 1-й разряд ПВВ А
```

```
    PORTA = PORTA|0x02;
```

```
}
```

Вариант 2 (с использованием программного опроса признака переполнения счетчика).

Комментариями снабжены только команды, отсутствующие в вышеприведенном 1-м варианте программного фрагмента, реализующего ту же функцию

```
#include <avr/io.h>
```

```
#include <avr/interrupt.h>
```

```
//Объявление переменных, используемых в программе
```

```
•  
•  
•
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
int main(void)
```

```
{
```

```
    DDRA=0x03;
```

```
    TCNT1 = 0x85ED;
```

```
    PORTA=0x01;
```

```
    TCCR1B = TCCR1B|0b00000100;
```

```
//Ожидание установки в единицу бита TOV1 в регистре TIFR (см. рис. 9.37)
```

```
    while((TIFR&0x04)==0);
```

```
//Остановка счетчика 1-го таймера
```

```

    TCCR1B = 0b00000000;
/*
Обнуление бита признака переполнения (TOV1) в регистре TIFR (см. рис.
9.37)
*/
    TIFR = TIFR|0x04;
//Включение второго реле записью единицы в 1-й разряд ПБВ А
    PORTA = PORTA|0x02;
//Продолжение основной программы
.
.
.
}

```

9.5.2.2. Пример 2. Формирование гармонического сигнала посредством ШИМ-ЦАП.

Пусть необходимо способом прямого цифрового синтеза сформировать гармонический (синусоидальный) сигнал с частотой 1 кГц, смещенный на величину амплитуды относительно нулевого уровня. Амплитуда сигнала должна быть максимально возможной при номинальном напряжении питания МК. Эквивалентная разрядность ЦАП – не менее 6-и бит. Минимально допустимое отношение периода формируемого сигнала к периоду ШИМ-импульсов – 100. Модель МК – 1887BE7T (*ATmega128*). МК тактируется ГТИ частотой 8 МГц с внешним ПЭР. Выбор таймера и канала формирования сигнала должны производиться в процессе разработки ПО. Калибровка ЦАП не требуется.

На данном примере также будут рассмотрены общие вопросы формирования сигналов с заданной формой и параметрами способом ШИМ.

1) Начинаем с выбора режима ШИМ, от которого зависит эквивалентная разрядность ШИМ-ЦАП в битах (N_{EPWM}), которая, в свою очередь, равна $\lfloor \log_2(N_{PW}) \rfloor$, где $\lfloor . \rfloor$ – оператор округления до ближайшего меньшего целого, N_{PW} – максимальное число значений длительности импульса ШИМ-сигнала при заданном значении его периода (T). Нетрудно увидеть, что $N_{PW} = T/\tau_{min}$, где τ_{min} – минимальная длительность импульса при заданном T . Из выражений (9.6) и (9.8) следует, что при асимметричной ШИМ τ_{min} равна $1/f_{CNT}$, а при центрированной –

$2/f_{CNT}$. Следовательно, эквивалентная разрядность ШИМ-ЦАП при заданном периоде импульсов описывается следующими выражениями:

- в режиме асимметричной ШИМ:

$$N_{E\,PWM} = \lfloor \log_2(f_{CNT}T) \rfloor \text{ бит}; \quad (9.31)$$

- в режиме центрированной ШИМ:

$$N_{E\,PWM} = \lfloor \log_2(f_{CNT}T) \rfloor - 1 \text{ бит}; \quad (9.32)$$

Данные выражения верны при условии, что во всем диапазоне изменения выходного сигнала ШИМ-ЦАП отношение размаха пульсации сглаженного ШИМ-сигнала к его максимальному значению пренебрежимо мало по сравнению с $1/N_{PW}$, т. е. приведенная к конечной точке шкалы ЦАП погрешность от неполного подавления пульсации пренебрежимо мала по сравнению с приведенной погрешностью квантования. Подробнее – см. пункт 8).

Таким образом, асимметричная ШИМ обеспечивает на 1 бит большую разрядность ЦАП, чем центрированная, при тех же значениях f_{CNT} и T . С другой стороны, для ШИМ-ЦАП не существенно основное преимущество центрированной ШИМ – отсутствие «скачков» начальной фазы импульсов при изменении их длительности. Поэтому для ШИМ-формирования сигналов рационально выбрать режим асимметричной («быстрой») ШИМ.

2) Определяем максимально допустимое значение периода ШИМ-импульсов. По вышеприведенным условиям, он не должен превышать 0,01 периода формируемого сигнала, равного $1/1 \text{ кГц} = 1 \text{ мс}$. Следовательно, максимально допустимое значение периода ШИМ-импульсов равно 10 мкс.

3) Далее необходимо определить его минимально возможное значение (как нетрудно заметить из табл. 9.1, чем он меньше, тем меньше относительная пульсация сглаженного ШИМ-сигнала). Минимальное значение периода ШИМ-сигнала определяется требуемой эквивалентной разрядностью ШИМ-ЦАП, которая в выбранном асимметричном режиме ШИМ равна $\lfloor \log_2(f_{CNT}T) \rfloor$ (см. подпункт 9.3.2.5). В рассматриваемом случае разрядность должна

быть равна 6-и битам, частота счетных импульсов, очевидно, не может превышать частоту тактирования МК (8 МГц). Следовательно, минимально возможное значение T равно 8 мкс. **Важно** отметить, что период формируемого сигнала должен содержать **целое** число периодов ШИМ-импульсов. В данном примере это условие соблюдается ($1 \text{ мс} / 8 \text{ мкс} = 125$). При его несоблюдении период T **должен** быть скорректирован до ближайшего **большого** значения, удовлетворяющего данному условию.

4) Период ШИМ-импульсов в режиме асимметричной ШИМ определяется выражением (9.5). В соответствии с ним, при $f_{CNT} = 8 \text{ МГц}$ и $T = 8 \text{ мкс}$ значение верхнего предела счета (TOP) должно быть равно **63-м**.

5) Выбираем таймер МК и режим его работы. Верхний предел счета 8-битовых таймеров МК 1887BE7T в режиме асимметричной («быстрой») ШИМ является фиксированным и равным 255-ти ($0xFF$) [8]. В 16-битовых таймерах возможны варианты «быстрой» ШИМ с верхним пределом счета, равным содержимому регистра захвата или одного из регистров сравнения [8]. Таким образом, в рассматриваемом случае формирование импульсов «быстрой» ШИМ с периодом 8 мкс возможно только при использовании одного из 16-битовых таймеров МК 1887BE7T. В данном примере выбран 1-й таймер; режим его работы – 15-й («быстрая» ШИМ с верхним пределом счета, равным содержимому регистра сравнения $OCR1A$). Выдача формируемого сигнала осуществляется на выход канала B 1-го таймера, $OC1B$ (см. рис. 9.32). Значения отсчетов формируемого сигнала прямо пропорциональны длительности ШИМ-импульсов, которая, в свою очередь, описывается выражением (9.6), в котором, в данном конкретном случае, в качестве OCR служит содержимое регистра $OCR1B$.

6) Далее необходимо вывести выражение для расчета загружаемых в регистр сравнения чисел, как функции от отсчетов формируемого сигнала. Ограничимся режимом асимметричной ШИМ, предпочтительным для применения в ШИМ-ЦАП (см. пункт 1)).

Длительность ШИМ-импульса теоретически может изменяться в пределах от 0 до T . Однако, на практике следует избегать крайних значений данного диапазона (т. е. коэффициентов заполнения ШИМ-сигнала, равных 0% и 100%) [8]. Рационально ограничиться пределами изменения от τ_{min} до $T - \tau_{min}$. В режиме асимметричной ШИМ им соответствуют значения OCR , равные 1 и TOP соответственно.

В большинстве практических случаев (в том числе в рассматриваемом примере) зависимость длительностей ШИМ-импульсов и, соответственно, содержимого OCR (см. выражение (9.6)) от значений отсчетов формируемого сигнала должна быть линейной, т. е.:

$$OCR[iT] = a \times y[iT] + b; \quad (9.33)$$

где $OCR[iT]$ и $y[iT]$ – соответственно содержимое OCR и значение отсчета формируемого сигнала в момент времени iT ($i = 0, 1, 2, \dots$); a и b – коэффициенты, определяемые из следующей системы уравнений:

$$\begin{cases} a \times y_{max} + b = TOP; \\ a \times y_{min} + b = 1; \end{cases} \quad (9.34)$$

где y_{min} и y_{max} – минимальное и максимальное значение формируемого сигнала, при которых содержимое OCR должно быть равно 1 и TOP соответственно (см. предыдущий абзац).

Решая систему уравнений (9.33) относительно коэффициентов a и b , получаем, что в режиме асимметричной ШИМ:

$$a = \frac{TOP - 1}{y_{max} - y_{min}}; \quad (9.35)$$

$$b = TOP - \frac{TOP - 1}{y_{max} - y_{min}} y_{max}; \quad (9.36)$$

На основе выражений (9.33), (9.35) и (9.36), с учетом того, что содержимое OCR может быть только целым положительным числом, после соответствующих преобразований получаем следующее выражение для его расчета:

$$OCR[iT] = round\left\{\frac{TOP - 1}{y_{max} - y_{min}}y[iT] + \frac{y_{max} - y_{min}TOP}{y_{max} - y_{min}}\right\}; \quad (9.37)$$

где $round\{.\}$ – оператор округления до ближайшего целого.

В рассматриваемом примере $y_{min} = -1$, $y_{max} = +1$, $TOP = 63$. Следовательно, в соответствии с выражением (9.37):

$$OCR[iT] = round\{31 \times \sin[iT] + 32\}. \quad (9.38)$$

7) Ниже представлен программный модуль, реализующий ШИМ-формирование синусоидального сигнала с параметрами и характеристиками, соответствующими результатам выполнения предыдущих пунктов. *IDE – Atmel Studio 7*. Уровень программирования – регистровый, представление операндов – числовое (см. подпункт 2.6.1.14 и Приложение В).

Примечание. В приведенном программном модуле расчет значений формируемого сигнала (синусоиды) осуществляется с использованием стандартной библиотеки математических функций языка C.

```
//Подключение библиотеки моделей ПУ МК семейства AVR
#include <avr/io.h>
/*
Подключение библиотеки стандартных функций обработки прерываний МК
семейства AVR
*/
#include <avr/interrupt.h>
//Подключение библиотеки математических функций
#include <math.h>
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Объявление переменных, используемых в программе
/*
Объявление массива 8-битовых целых чисел без знака, загружаемых в регистр
сравнения в процессе формирования синусоиды
*/
unsigned char sn[150];

/*
Объявление аргумента (t) и значения (y) функции синуса как переменных с
плавающей точкой
*/
```

```
float t, y;
```

```
/*
```

Объявление используемых в программе индексных переменных как целочисленных

```
*/
```

```
int i, j;
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//Код основного блока программы
```

```
int main(void)
```

```
{
```

```
/*
```

Вычисление отсчетов синусоиды частотой 1 кГц с периодом дискретизации, равным выбранному периоду ШИМ-сигнала (8 мкс), и расчет чисел, загружаемых в регистр сравнения, по выражению (9.38). Округление осуществляется за счет преобразования типов переменных *float* → *unsigned char*. Слагаемое 0.5 в выражении для расчета *sn[i]* обеспечивает математически корректное округление до **ближайшего целого**; в отсутствие данного слагаемого *sn[i]* было бы равно целой части значения $31*y+32$.

```
*/
```

```
    t = 0.000000;
```

```
    for(i=0;i<125;i++)
```

```
    {
```

```
        y = sin(2*3.14159*1000*t);
```

```
        sn[i]=31*y+32+0.5;
```

```
        t=t+0.000008;
```

```
    }
```

```
/*
```

Конфигурирование 6-го вывода ПВВ *B* как цифрового двухтактного выхода (см. табл. 6.1). В рассматриваемом примере данный вывод используется по альтернативной функции, в качестве выхода канала *B* 1-го таймера (*OC1B*), т. е. выхода формируемого сигнала.

Настройка 6-го вывода ПВВ *B* на выполнение функции *OC1B* обеспечивается конфигурированием канала *B* 1-го таймера (см. далее), однако настройку данного вывода на работу в режиме выхода необходимо выполнить отдельно [8].

```
*/
```

```
    DDRB = DDRB|0b01000000;
```

```
/*
```

Конфигурирование 1-го таймера, форматы регистров управления которого приведены на рис. 9.35:

- выбор 15-го режима работы («быстрый» ШИМ с $TOP = OCR1A$) записью кода 1111 в битовое поле *WGM* (0-й и 1-й биты регистра *TCCR1A*, 3-й и 4-й биты регистра *TCCR1B*);

- выбор режима формирования не инвертированного ШИМ-сигнала по каналу *B*, записью кода 10 в битовое поле *COM1B* (4-й и 5-й биты регистра *TCCR1A*, см. также табл. 9.4 и пояснения к рис. 9.35);
- запись верхнего предела счета (*TOP*) в регистр *OCR1A*;
- обнуление старшего байта регистра *OCR1B*, задающего в данном примере длительность ШИМ-импульса; ввиду того, что она задается с 6-битовой точностью, старший байт регистра *OCR1B* в процессе ШИМ не задействован;
- запись числа *sn[0]* в младший байт регистра *OCR1B* (реально – в его буфер, см. комментарии к подпрограмме обслуживания прерываний по переполнении счетчика 1-го таймера);
- разрешение прерываний по переполнении счетчика 1-го таймера (установка в единицу 2-го бита регистра *TIMSK*, см. рис. 9.37).

$$TCCR1A = 0b00100011;$$
$$OCR1A=63;$$
$$OCR1BL=sn[0];$$

/*

*/

$$TCCR1B = TCCR1B|0x01;$$
$$i=0;$$

Глобальное разрешение прерываний установкой в единицу бита I в регистре статуса МК (см. рис. 2.24)

```
sei();
```

Собственно ШИМ-формирование синусоидального сигнала в «бесконечном» цикле. Загрузка в регистр *OCR1B* числа, задающего длительность очередного импульса, производится по прерыванию, запрос на которое генерируется по переполнении счетчика таймера (см. рис. 9.33)

while (1);

////////////////////////////////////

Подпрограмма обслуживания прерывания по переполнении счетчика 1-го таймера

SIGNAL (TIMER1_OVF_vect)

{
/*

Сброс признака переполнения счетчика 1-го таймера записью единицы во 2-й бит регистра *TIFR* (см. рис. 9.37). См. также комментарий к аналогичной команде в Варианте 1 программного фрагмента, приведенного в подпункте 9.5.2.1

*/

TIFR = TIFR|0x04;

/*

Запись в младший байт регистра *OCR1B* (фактически – в его буфер) числа *sn[i]*, задающего длительность импульса, который будет сформирован после **следующего** переполнения счетчика. По **текущем** переполнении в *OCR1BL* из буфера автоматически загрузится число, записанное в него подпрограммой обслуживания прерывания при **предыдущем** переполнении счетчика. См. рис. 9.39, а также рис. 9.7 и пояснения к нему.

*/

OCR1BL=sn[i];

//Инкремент счетчика импульсов

i=i+1;

/*

Если сформированы все 125 ШИМ-импульсов очередного периода генерируемой синусоиды (с 0-го по 124-й) – обнуление счетчика импульсов и переход к формированию следующего периода синусоиды

*/

if(i>124) i=0;

}



Рис. 9.39. Временные диаграммы перезагрузки регистров сравнения таймеров МК семейства AVR в режиме ШИМ (*n* – номер счетчика; *x* – номер канала; пунктирными линиями обозначено положение во времени активных перепадов счетных импульсов) [8]

8) В заключение необходимо отметить, что ШИМ-сигнал, сформированный на выходе *OC1B*, естественно, требует

сглаживания посредством ФНЧ. Остановимся вкратце на выборе его параметров.

Базовыми **требованиями** к сглаживающему ФНЧ ШИМ-ЦАП являются следующие:

- во всем диапазоне изменения выходного сигнала ШИМ-ЦАП **приведенная** к конечной точке шкалы ЦАП **погрешность** от неполного подавления пульсаций должна быть пренебрежимо мала по сравнению с приведенной погрешностью квантования, в рассматриваемом примере равной $1/2^6 \approx 0,0156$ (см. пояснения к выражениям (9.31) и (9.32));

- с другой стороны, **неравномерность АЧХ** ФНЧ, нормированной относительно нулевой частоты (т. е. отношение $|H(f)|/|H(0)|$) в диапазоне частот формируемого «полезного» сигнала (в рассматриваемом случае - синусоиды), очевидно, также должна быть пренебрежимо мала по сравнению с приведенной погрешностью квантования.

В зависимости от конкретной решаемой задачи, первостепенное значение может иметь **погрешность** от неполного подавления пульсаций или во **временной**, или в **частотной** области, т. е.:

- или вызванное неполным подавлением пульсаций максимальное отклонение выходного напряжения ФНЧ от «идеальной» временной диаграммы сигнала, подлежащего формированию ШИМ-ЦАП;

- или суммарный уровень паразитных спектральных составляющих выходного сигнала ФНЧ, обусловленных наличием в нем остаточных пульсаций;

(см. также подпункты 10.4.5.3 и 10.4.5.4).

Погрешности во **временной** области фактически представляют собой значения остаточных пульсаций на выходе ФНЧ. Зависимости относительных значений пульсаций на выходе ФНЧ 1-го порядка от его частоты среза и от коэффициента заполнения ШИМ-сигналов приведены в табл. 9.1. Они являются, по существу, относительными (но не приведенными) погрешностями от неполного подавления пульсации ШИМ-сигнала при различных коэффициентах заполнения. В рассматриваемом случае относительная погрешность равна $\Delta/\overline{U_{PWM}}$, а приведенная – $\Delta/\overline{U_{PWM\ max}}$, где Δ - размах

пульсации, $\overline{U_{PWM}}$ и $\overline{U_{PWM\ max}}$ – соответственно среднее значение ШИМ-напряжения при заданном и при максимальном (т. е. равном 100%) коэффициентах заполнения (см. выражение (9.3)). Таким образом, приведенная погрешность от неполного подавления пульсации равна произведению относительной погрешности на отношение $\overline{U_{PWM}}/\overline{U_{PWM\ max}}$, равное, в свою очередь, коэффициенту заполнения ШИМ-сигнала.

В табл. 9.5 приведены значения приведенной погрешности от неполного подавления пульсаций ШИМ-сигнала (т. е. приведенные значения остаточных пульсаций), рассчитанные на основе табл. 9.1 при 1-м порядке ФНЧ.

Таблица 9.5

Значения приведенной погрешности от неполного подавления пульсаций ШИМ-сигнала, рассчитанные на основе табл. 9.1

Коэффициент заполнения ШИМ-сигнала	Приведенная погрешность от неполного подавления пульсаций ШИМ-сигнала при частоте среза ФНЧ, равной		
	0,0159 / T	0,0053 / T	0,0016 / T
2,5 %	0,00192	0,00066	0,0002
5 %	0,0047	0,00155	0,00045
10 %	0,0087	0,0029	0,0009
25 %	0,01825	0,006	0,00175
50 %	0,0245	0,008	0,0025
75 %	0,018	0,006	0,00225
90 %	0,0081	0,0027	0,0009

Из табл. 9.5 можно сделать следующий основной вывод: приведенная погрешность от неполного подавления пульсаций максимальна при коэффициенте заполнения, равном 50%. Поэтому, если приоритетными являются погрешности формирования сигнала во **временной** области, основными критериями выбора сглаживающего ФНЧ ШИМ-ЦАП должны служить:

- обеспечение приведенного значения остаточных пульсаций ШИМ-сигнала, пренебрежимо малого по сравнению с приведенной погрешностью квантования, при коэффициенте заполнения ШИМ-сигнала, равном 50%;

- неравномерность нормированной АЧХ ФНЧ в диапазоне частот формируемого «полезного» сигнала, пренебрежимо малая по сравнению с приведенной погрешностью квантования.

В **частотной** области неполное подавление пульсаций ШИМ-сигнала вызывает появление паразитных спектральных составляющих в выходном напряжении ШИМ-ЦАП (т. е. сглаживающего ФНЧ). Погрешность ШИМ-ЦАП в **частотной** области, обусловленная этим фактором, в общем случае, характеризуется отношением суммарного среднеквадратического значения (СКЗ) паразитных спектральных составляющих выходного сигнала ФНЧ к суммарному СКЗ «полезных» составляющих. Как правило, данная погрешность нормируется при формировании **гармонических** сигналов. В этом частном случае она представляет собой **суммарный коэффициент гармонических составляющих** (*Total Harmonic Distortion, THD*) выходного напряжения ФНЧ, определяемый по следующему выражению (см. также подпункт 10.4.5.15):

$$THD = \left(\sqrt{\sum_{h=2}^{H_{max}} U_{RMS\ h}^2} \right) / U_{RMS\ 1};$$

где $U_{RMS\ h}$ – СКЗ напряжения паразитной гармонической составляющей (гармоники) с номером h , т. е. с частотой, в h раз большей частоты основной, «полезной» гармонической составляющей, называемой 1-ой гармоникой; $U_{RMS\ 1}$ – СКЗ напряжения 1-ой гармоники; $h = 2, 3, 4, \dots, H_{max}$; H_{max} – номер наиболее высокочастотной из паразитных гармоник, подлежащих учету при определении THD (для ШИМ-ЦАП – как правило, гармоники с частотой, равной $3 / T$).

Таким образом, если приоритетными являются погрешности ШИМ-формирования сигнала в **частотной** области, то, в наиболее распространенном на практике частном случае, при формировании гармонических сигналов, основными критериями выбора сглаживающего ФНЧ ШИМ-ЦАП должны служить:

- обеспечение суммарного коэффициента гармонических составляющих (THD), пренебрежимо малого по сравнению с приведенной погрешностью квантования;

- неравномерность нормированной АЧХ ФНЧ на частоте формируемого «полезного» сигнала, пренебрежимо малая по сравнению с приведенной погрешностью квантования.

В соответствии с [53], пренебрежимо малой могут считаться погрешность / THD / неравномерность АЧХ, в 3 раза меньшие приведенной погрешности квантования, т. е. меньшие $1/(3 \times 2^N)$, где N – разрядность ШИМ-ЦАП.

В качестве сглаживающего ФНЧ ШИМ-ЦАП, в общем случае, предпочтительно применять ФНЧ Баттерворта [20, 23], характеризуемый максимальной равномерностью АЧХ в полосе пропускания и приемлемой для большинства практических случаев скоростью спада АЧХ в полосе заграждения. При этом, ввиду необходимости минимизации числа внешних по отношению к МК компонентов, рационально использование ФНЧ с возможно меньшим порядком (не выше 2-го).

В табл. 9.6 представлены:

- выраженные в периоде ШИМ-сигнала максимальные частоты среза ($f_{c\ max}$) ФНЧ Баттерворта 1-го и 2-го порядков, при которых обеспечиваются приведенные значения остаточных пульсаций сглаженного ШИМ-сигнала, пренебрежимо малые по сравнению с приведенной погрешностью квантования, при коэффициенте заполнения ШИМ-сигнала, равном 50%, и разрядности ШИМ-ЦАП, равной 6-и, 8-и и 10-ти битам;

- выраженные в частоте среза ФНЧ максимальные частоты спектральных составляющих формируемого сигнала ($f_{s\ max}$), при которых обеспечивается неравномерность АЧХ ФНЧ Баттерворта 1-го и 2-го порядка, пренебрежимо малая по сравнению с погрешностью квантования 6-и, 8-и и 10-битового ШИМ-ЦАП;

- значения THD формируемого ШИМ-ЦАП гармонического сигнала, обеспечиваемые при частоте среза ФНЧ и частоте сигнала, равных $f_{c\ max}$ и $f_{s\ max}$ соответственно.

Таблица 9.6

*Исходные данные для выбора сглаживающего ФНЧ
ШИМ-ЦАП*

Разрядность ШИМ-ЦАП	Приведенная погрешность квантования	Порядок ФНЧ	$f_{C\max}^{1)}$	$f_{S\max}^{1)}$	THD при $f_S = f_{S\max}$ и $f_C = f_{C\max}$
6 бит	0,015625 (1,5625%)	1	$0,0032 / T$	$0,100f_C$	0,27%
		2	$0,0636 / T$	$0,315f_C$	0,31%
8 бит	0,00391 (0,391%)	1	$0,00086 / T$	$0,050f_C$	0,07%
		2	$0,0358 / T$	$0,225f_C$	0,10%
10 бит	0,000977 (0,0977%)	1	$0,00022 / T$	$0,025f_C$	0,02%
		2	$0,0215 / T$	$0,155f_C$	0,05%
			$0,0016 / T^{2)}$		0,03%

¹⁾ См. пояснения данных обозначений в тексте на предыдущей странице.

²⁾ Данная частота среза является максимальной, при которой значение THD пренебрежимо мало по сравнению с приведенной погрешностью квантования 10-битового ЦАП. Пренебрежимо малые по сравнению с ней приведенные значения остаточных пульсаций выходного напряжения ФНЧ обеспечиваются при частоте среза, равной $0,0215 / T$.

Примечание. Как можно увидеть из табл. 9.6, при частоте среза ФНЧ, обеспечивающей приведенные значения остаточных пульсаций сглаженного ШИМ-сигнала, пренебрежимо малые по сравнению с приведенной погрешностью квантования, значения THD также пренебрежимо малы по сравнению с ней. Исключением является только вариант, при котором требуемая разрядность ШИМ-ЦАП равна 10-ти битам, а в качестве сглаживающего применяется ФНЧ Баттерворта 2-го порядка. В данном случае значение THD , соответствующее 10-битовой разрядности ШИМ-ЦАП, обеспечивается при частоте среза ФНЧ, меньшей, чем та, при которой возможно обеспечение приведенного значения остаточных пульсаций, пренебрежимо малого по сравнению с погрешностью квантования 10-битового ЦАП (см. табл. 9.6).

Типовые схемы реализации ФНЧ (в т. ч. ФНЧ Баттерворта) и руководства по их расчету приведены, например, в [20] и в [23].

В рассматриваемом примере разрядность ШИМ-ЦАП равна 6-и битам, период T ШИМ-сигнала – 8 мкс, формируемый сигнал – гармонический, с частотой 1 кГц. Как следует из табл. 9.6, при данном сочетании периода ШИМ и частоты формируемого сигнала, соответствующие 6-битовой разрядности требования к приведенному значению остаточных пульсаций, коэффициенту THD и неравномерности АЧХ ФНЧ на частоте формируемого сигнала удовлетворяются при использовании ФНЧ Баттерворта **2-го порядка** с частотой среза, равной $0,0636 / 8$ мкс, т. е. 7950 Гц. При этом неравномерность АЧХ ФНЧ в полосе пропускания, соответствующая 6-битовой разрядности ЦАП, обеспечивается до частоты, равной 2,5 кГц, т. е. существенно превышающей частоту формируемого сигнала (1 кГц). Заметим, что при применении ФНЧ **1-го порядка** его частота среза должна быть равна не более $0,0032 / 8$ мкс, т. е. 400 Гц, однако при такой частоте среза нормированная АЧХ ФНЧ на частоте 1 кГц будет равна 0,371, что существенно ниже минимально допустимого значения, соответствующего 6-битовой разрядности ЦАП (см. табл. 9.6).

9.5.3. Базовые структурно-архитектурные решения таймеров МК семейства *ARM Cortex-Mx*

9.5.3.1. В состав большинства моделей МК семейства *ARM Cortex-Mx* входят таймеры общего назначения следующих категорий [9, 13 – 15]:

- «базовые» таймеры (*Basic timers*);
- собственно таймеры общего назначения (*General-purpose timers*);
- таймеры с «продвинутым» управлением (*Advanced-control timers*), которые корректнее назвать таймерами с расширенными функциональными возможностями

Рассмотрим типовые структурно-архитектурные решения таймеров первых трех перечисленных категорий (на примере таймеров МК модельного ряда *STM32F030xx* [15]). Типовая структура и архитектура счетчика реального времени, а также основы его применения рассмотрены в пункте 3.3.3.

9.5.3.2. В целом, общие структурные и архитектурные решения таймеров МК семейства *ARM Cortex-Mx* аналогичны таковым таймеров МК семейства *AVR* (и всех современных семейств МК классов «*mainstream*» и «*high performance*»). Основными особенностями архитектуры таймеров МК семейства *ARM Cortex-Mx*, характерными для всех их категорий (кроме счетчика реального времени) являются [9, 13 – 15]:

- разрядность счетчиков большинства таймеров равна 16-ти битам; исключение составляют некоторые из таймеров общего назначения ряда подсемейств (например, 2-й и 5-й таймеры МК STM32F405xx/07xx подсемейства *ARM Cortex-M4* [14]), разрядность счетчиков которых равна 32-м битам;

- верхний предел счета задается программно-доступным 16-битовым буферизированным регистром автоматической перезагрузки (*Auto-reload register, ARR*) с возможностью программного разрешения / запрета буферизации (см. подпункт 9.3.4.3);

- коэффициенты деления предделителей (*Prescaler*) частоты тактирования счетчиков всех таймеров могут задаваться ПО в диапазоне от 1 до 2^{16} с шагом 1;

- в отличие от таймеров МК семейства *AVR*, таймеры МК семейства *ARM Cortex-Mx* могут, по определенным событиям (например, по переполнению счетчика и т. п.), вырабатывать не только запросы на прерывание, но и запросы на ПДП; данная возможность полезна, например, при необходимости синхронизации циклов обмена в режиме ПДП с определенными моментами времени (при этом собственно таймер может и не быть абонентом процесса обмена данными, см., например, подпункт 9.5.3.16);

- большинство таймеров МК семейства *ARM Cortex-Mx* могут служить источниками сигналов тактирования других таймеров и / или каких-либо функциональных блоков МК (АЦП, ЦАП и т. п.); сигналы тактирования вырабатываются по некоторому, выбираемому программно событию (переполнению счетчика таймера и т. п.).

Источником тактирования счетчика таймера может служить:

- синхросигнал домена, к которому относится таймер (см. подпункт 4.4.2.7);
 - синхросигнал, вырабатываемый некоторым другим таймером по определенному событию (см. выше);
 - выходной сигнал ГТИ, внешнего по отношению к МК;
- при этом выбор источника тактирования осуществляется программно.

Таймеры категории «*Basic*», как правило, могут тактироваться только синхросигналом их домена. То же характерно и для некоторых из таймеров категории «*General-purpose*».

Basic timers предназначены, в основном, для формирования задержек между событиями (см. подраздел 9.2), и не содержат узлов и блоков, необходимых для формирования ШИМ- и ЧИМ-сигналов, а также для преобразования в код частоты и интервалов времени. Таймеры категорий «*General-purpose*» и «*Advanced-control*» ориентированы на реализацию всех частотно-временных функций, перечисленных в подразделе 9.1.

9.5.3.3. Типовой пример структурной схемы таймера категории «*Basic*» приведен на рис. 9.40 [15].

Основой таймера служит 16-битовый **суммирующий** счетчик (*COUNTER*), программно-доступный как для записи, так и для чтения. Тактирование счетчика осуществляется выходным сигналом предделителя (*Prescaler*), на вход которого поступает синхросигнал домена *APB1*. Частота счетных импульсов, $f_{СК_CNT}$, описывается выражением:

$$f_{СК_CNT} = f_{СК_PSC} / (PSC + 1); \quad (9.39)$$

где *PSC* – содержимое программно-доступного 16-битового регистра, задающего коэффициент деления предделителя. *PSC* может принимать значения от 0 до $2^{16} - 1$; следовательно, коэффициент деления может изменяться в пределах от 1 до 2^{16} с шагом 1.

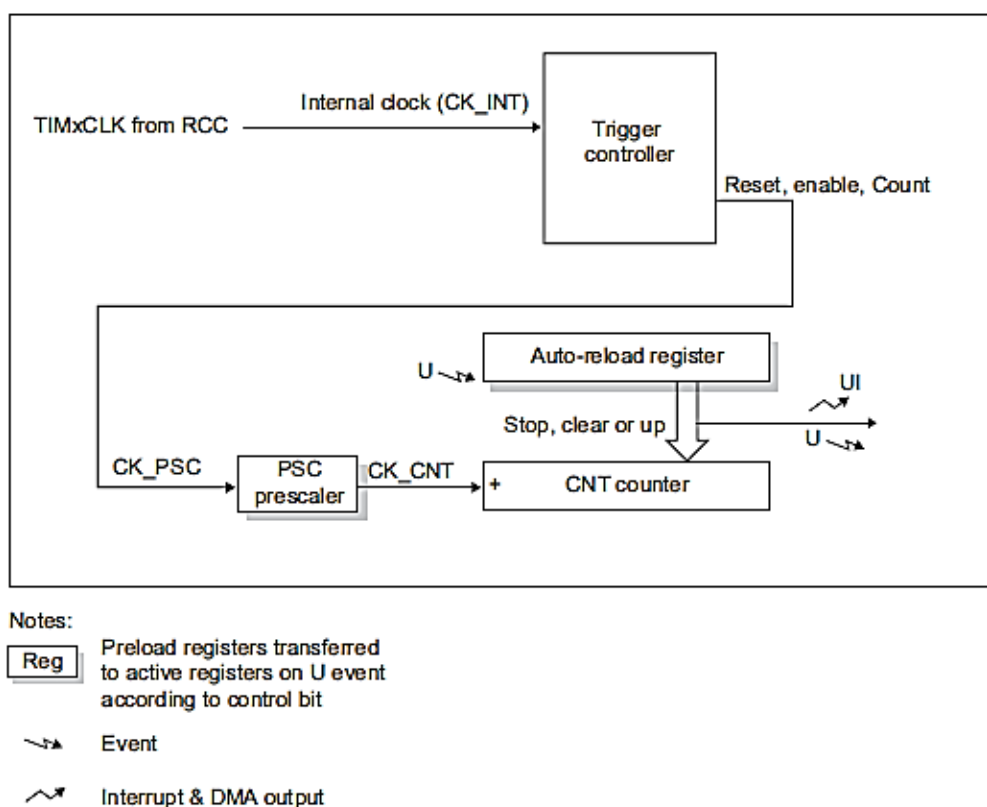


Рис. 9.40. Типовая структурная схема таймера категории *Basic timer* МК семейства *ARM Cortex-Mx* (на примере таймеров *TIM6* и *TIM7* МК модельного ряда *STM32F030xx* [15]).
Пояснения приведены в тексте

Запуск / остановка счетчика осуществляются соответственно установкой / сбросом бита *CEN*) в 1-м регистре управления таймером (*CRI*). Заметим, что конфигурирование и работа таймера возможны только, если установлен в единицу бит разрешения его тактирования (*TIMnEN*, где *n* – номер таймера) в регистре *RCC_APB1ENR* подсистемы синхронизации и сброса (см. подпункт 4.4.2.7).

Верхний предел счета задается регистром автоматической перезагрузки (*Auto-reload register, ARR*). Временные диаграммы работы счетчика при *ARR = 36* полностью аналогичны приведенным на рис. 9.1а.

Заметим, что регистры *ARR* и *PSC* снабжены буферными регистрами (*Preload registers*), обозначенными на рис. 9.40 как «тень» регистров *ARR* и *PSC*. Команда записи *ARR* или *PSC* реально загружает соответствующий буферный регистр, содержимое которого передается в «*active register*», т. е. собственно в *ARR* или

PSC, по событию «Обновление таймера» (*Update event*), обозначенному на рис. 9.40 как *U*. Буферизация регистра *ARR* может быть запрещена записью нуля в бит *ARPE* 1-го регистра управления таймером.

Источники возникновения события «Обновление таймера» задаются программно, состоянием бита *URS* (*Update Request Source*) 1-го регистра управления таймером. Ими могут быть:

- при нулевом значении данного бита – переполнение счетчика (см. рис. 9.1), обновление сигналом от ведущего устройства или программное обновление таймера, записью единицы в бит *UG* (*Update Generation*) регистра *EGR* (*Event Generation Register*) таймера;

- при единичном состоянии – только переполнение счетчика.

По событию *Update event* может быть сгенерирован запрос на прерывание или на ПДП (*UI* на рис. 9.40), если они разрешены. Их разрешение / запрет осуществляются установкой / сбросом соответственно битов *UIE* и *UDE* регистра *DIER* (*DMA/Interrupt Enable Register*) таймера.

Более подробное описание структуры и архитектуры таймеров категории «*Basic*» различных подсемейств / модельных рядов МК семейства *ARM Cortex-Mx* приводится в руководствах по применению соответствующих модельных рядов МК (см, например, [13 – 15]).

9.5.3.4. Типовой пример структурной схемы таймера категории «*General-purpose*» МК семейства *ARM Cortex-Mx* приведен на рис. 9.41 [15]. Рассмотрим на его основе базовые особенности структуры и архитектуры таймеров данной категории.

9.5.3.5. Счетчик таймера категории «*General-purpose*» может работать в режимах прямого (*up-counting*), обратного (*down-counting*) и реверсивного (*center-aligned*) счета (см. описания данных режимов в пункте 9.3.1). В отличие от таймеров МК семейства *AVR*, работа в каждом из этих режимов возможна независимо от того, используется ли таймер для формирования интервалов времени (см. рис. 9.1а и 9.1б), генерации сигналов с программно-управляемыми частотно-временными параметрами или для реализации функций захвата. Режим работы счетчика задается битом *DIR* (*Direction*) и

битовым полем *CMS* (*Center-aligned Mode Selection*) 1-го регистра управления таймером (*CR1*).

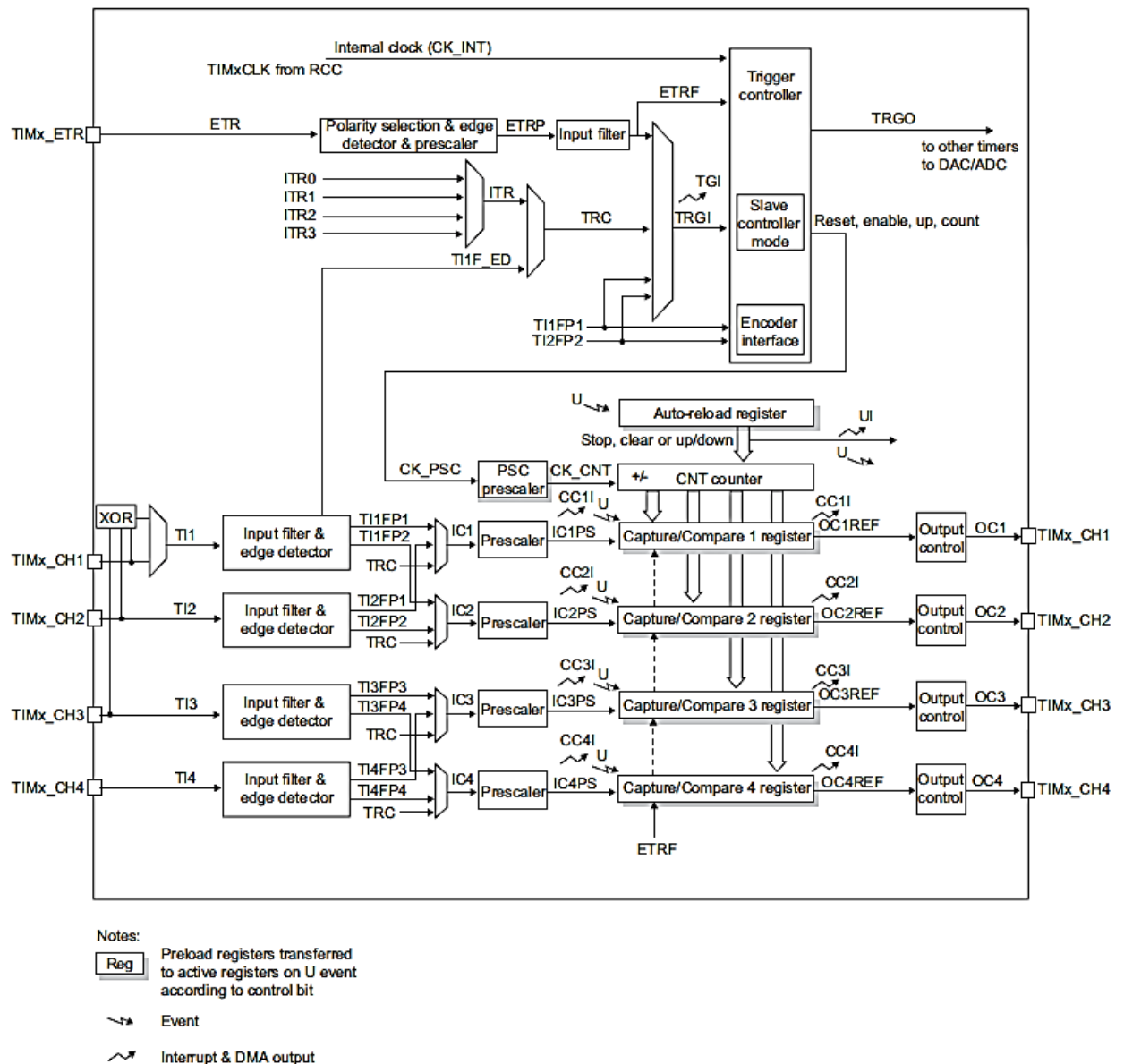


Рис. 9.41. Типовая структурная схема таймера категории «*General-purpose*» МК семейства *ARM Cortex-Mx* (на примере таймера *TIM3* МК модельного ряда *STM32F030xx* [15]).

Пояснения приведены в тексте

При *CMS*, равном 00, счетчик работает в режиме одностороннего счета (прямого, если *DIR* = 0, и обратного – при *DIR* = 1). При значении *CMS*, не равном 00, режим счета – реверсивный, причем, если *CMS* = 01, запрос на прерывание по совпадению содержимых счетчика и регистра сравнения генерируется только на интервале обратного счета, при *CMS* = 10 – прямого, а при *CMS* = 11 – как прямого, так и обратного. Наличие данных вариантов работы

счетчика, в частности, может быть использовано при реализации ШИМ, корректной по фазе (см. подпункт 9.5.3.10).

Верхний предел счета, как и во всех таймерах МК семейства *ARM Cortex-Mx*, задается регистром *ARR*, с возможностью его буферизации. Разрешение / запрет буферизации осуществляются установкой / сбросом бита *ARPE* 1-го регистра управления таймером. Если регистр *ARR* буферизирован (что рекомендуется), его загрузка, как и в таймерах категории «*Basic*», происходит по событию «Обновление таймера». При этом источниками возникновения события «Обновление таймера», в зависимости от состояния бита *URS* (*Update Request Source*) регистра *CR1*, могут быть:

- при $URS = 0$ – переполнение счетчика (при прямом счете) или потеря значимости (при обратном счете) (см. рис. 9.1), а также обновление сигналом от ведущего устройства (см. подпункт 9.5.3.19) или программное обновление таймера, записью единицы в бит *UG* (*Update Generation*) регистра *EGR* (*Event Generation Register*) таймера;

- при $URS = 1$ – только переполнение / потеря значимости.

9.5.3.6. Источником тактирования таймера категории «*General-purpose*», в общем случае, может служить (см. рис. 9.41):

- синхросигнал домена, к которому относится таймер (обозначен на рис. 9.41 как *Internal clock*);

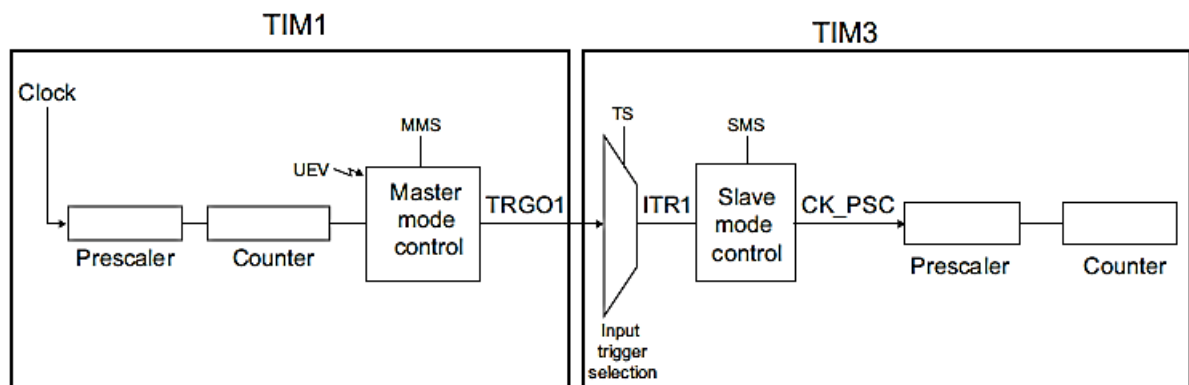
- выходной сигнал ГТИ, внешнего по отношению к МК (*ETR* на рис. 9.41);

- входной сигнал одного из каналов захвата (см. также подпункт 9.5.3.8), *TI1* или *TI2* на рис. 9.41;

- сигнал синхронизации, вырабатываемый некоторым другим таймером категории «*General-purpose*» или «*Advanced-control*» на его выходе *TRGO*; данные сигналы обозначены на рис. 9.41 как *ITR0* – *ITR3*.

В частности, последний из перечисленных вариантов тактирования позволяет использовать один таймер в качестве предделителя другого таймера, т. е. включение таймеров по схеме «Ведущий / Ведомый» (*Master / Slave*), пример которой приведен на рис. 9.42. Данная схема включения позволяет существенно

расширить диапазон формируемых интервалов времени и частотный диапазон генерируемых сигналов. Например, если 1-й таймер тактируется синхроимпульсами частотой 8 МГц, а коэффициенты деления предделителей обоих таймеров равны $2^{16} = 65536$, то, с учетом того, что верхний предел счета 16-битовых таймеров (N_{TOP}) равен $2^{16} - 1$, посредством 2-го таймера могут быть сформированы интервалы времени (см. подраздел 9.2) длительностью до $(1/8 \text{ МГц}) \times 2^{16} \times 2^{16} \times (N_{TOP} + 1) \approx 35184372 \text{ с}$, т. е. примерно до 407-и суток.



MMS – Master Mode Selection

SMS – Slave Mode Selection

TS – Trigger Selection

UEV – Update Event

Рис. 9.42. Пример включения таймеров МК семейства *ARM Cortex-Mx* по схеме «Ведущий / Ведомый» [15]

Выбор источника импульсов тактирования счетчика и управления им осуществляется 3-битовым полем *TS* (*Trigger Selection*), а режима тактирования и управления счетчиком – 3-битовым полем *SMS* (*Slave Mode Selection*) регистра *SMCR* (*Slave Mode Control Register*) таймера. В частности, при *SMS* = 000 предделитель счетчика тактируется фронтом (перепадом из 0 в 1) сигнала *Internal clock*; при *SMS* = 111 – фронтом сигнала *TRGI* (см. рис. 9.41), выбираемого, в свою очередь, битовым полем *TS* (например, при *TS* = 000 в качестве *TRGI* служит сигнал *ITR0*, при *TS* = 001 – *ITR1*, при *TS* = 111 – *ETRF* и т. п.). При *SMS* = 100 по фронту сигнала *TRGI* происходит обновление таймера, а предделитель его счетчика тактируется фронтом сигнала *Internal clock*.

Источники сигналов *ITR0* – *ITR3* таймера *TIM3* МК модельного ряда *STM32F030xx* приведены в табл. 9.7.

Таблица 9.7

Источники сигналов ITR0 – ITR3 таймера TIM3 МК модельного ряда STM32F030xx [15]

Slave TIM	ITR0 (TS = 000)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM3	TIM1	TIM15	TIM14

Выбор коэффициента деления предделителя частоты сигнала внешнего ГТИ производится битовым полем *ETPS* (*External Trigger Prescaler*) регистра *SMCR*, а коэффициента деления предделителя счетчика таймера – 16-битовым регистром *PSC*, как и в таймерах категории «*Basic*». Частота счетных импульсов рассчитывается по выражению (9.39).

Подробная информация о выборе источников и режимов тактирования таймеров категории «*General-purpose*» приводится в руководствах по применению соответствующих модельных рядов МК (см., например, пункт 14.4.3 руководства [15]).

Специализированный вариант тактирования счетчика, используемый в режиме интерфейса с **поворотными инкрементными энкодерами** (*Encoder interface* на рис. 9.41), рассмотрен в подпункте 9.5.3.18.

9.5.3.7. В свою очередь, таймеры категории «*General-purpose*» могут служить **источниками тактирования** других таймеров той же категории (см. рис. 9.42 и табл. 9.7), а также **запуска АЦП** и **стробирования ЦАП**, выполняя по отношению к ним функцию ведущего (*Master*). Сигнал тактирования / запуска / стробирования ведомых вырабатывается на выходе *TRGO* (*Trigger Output*), см. рис. 9.41. Событие, по которому осуществляется генерация сигнала *TRGO*, задается 3-битовым полем *MMS* (*Master Mode Selection*) 2-го регистра управления таймером (*CR2*). Например, при *MMS* = 010 сигнал *TRGO* вырабатывается по событию «Обновление таймера»; при *MMS* = 100 – сигнал *OC1REF*, генерируемый по совпадении содержимых счетчика и регистра *CCR1* (см. рис. 9.41) и т. п.; подробнее – см. пункт 14.4.2 руководства [15].

9.5.3.8. Поскольку *General-purpose timers* предназначены для реализации всех частотно-временных функций, перечисленных в подразделе 9.1, их структура (см. рис. 9.41) **дополнена** функциональными узлами, позволяющими осуществлять:

- генерация импульсных сигналов с программно-задаваемыми частотно-временными параметрами, в том числе ШИМ, ЧИМ и ФИМ (см. пункты 9.3.3 – 9.3.5), а также одиночных импульсов с программно-управляемой длительностью (см. пункт 9.3.6);

- «захват» содержимого счетчика по перепаду сигнала на некотором входе, при программно-задаваемых номере входа и направлении перепада, что, в свою очередь, позволяет реализовывать преобразование в код частоты, периода и длительности сигналов (см. подраздел 9.4).

В отличие от таймеров МК семейства *AVR*, для таймеров МК семейства *ARM Cortex-Mx* характерно совмещение функций регистров сравнения и захвата одним и тем же регистром, называемым *Capture / Compare Register* (регистр захвата / сравнения), *CCR*. В структуру таймера *TIM3* МК модельного ряда *STM32F030xx* (см. рис. 9.41) входит 4 канала захвата / сравнения и, соответственно, 4 независимых, программно доступных регистра *CCR* (*CCR1 – CCR4*, см. рис. 9.41). В состав МК данного модельного ряда также входят таймеры с 2-мя подобными каналами.

В режиме генерации импульсных сигналов с программно-управляемыми частотно-временными параметрами эти параметры задаются содержимым *CCR* (см., например, рис. 9.21 и 9.22). Функции формирователей выходного сигнала (см. рис. 9.2) выполняют программно-управляемые блоки, обозначенные на рис. 9.41 как *Output control*.

В режиме захвата содержимое счетчика загружается также в регистр *CCR*. В состав каждого из каналов захвата также входят (см. рис. 9.41):

- фильтр входного сигнала и детектор его активного перепада (*Input filter & edge detector*);

- программно-управляемый мультиплексор, посредством которого выбирается сигнал, инициирующий захват;

- предделитель (*Prescaler*) канала захвата, в зависимости от программно-задаваемого коэффициента деления которого захват может производиться по каждому, каждому 2-му, каждому 4-му или каждому 8-му активному перепаду сигнала, инициирующего захват.

В качестве как выходов каналов (в режиме генерации сигналов), так и их входов (в режиме захвата) служат одни и те же выводы ПВВ МК, которые при этом должны быть настроены на выполнение альтернативной функции *TIMx_CHn* (где *x* – номер таймера, *n* – номер канала). Настройка канала на конкретный режим работы (генерация сигнала или захват) осуществляется программно (см. подпункт 9.5.3.9).

9.5.3.9. Конфигурирование каждого из каналов осуществляется независимо от других, посредством 3-х 16-битовых регистров: *CCER* (*Capture / Compare Enable Register*), *CCMR1* и *CCMR2* (*Capture / Compare Mode Register 1 & 2*), форматы которых представлены на рис. 9.43.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	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

а)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	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

б)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

в)

Рис. 9.43. Форматы регистров *CCMR1* (а), *CCMR2* (б) и *CCER* таймера *TIM3* МК модельного ряда *STM32F030xx* [15]
(пояснения приведены в тексте)

В регистре *CCMR1* выделено по одному байту под конфигурирование 1-го и 2-го каналов, в регистре *CCMR2* – по байту под конфигурирование 3-го и 4-го каналов. Одни и те же биты выполняют различные функции, в зависимости от режима работы канала – генерация сигнала или захват по входу (см. рис. 9.43).

Выбор режима работы канала осуществляется 2-битовым полем $CCnS$, где n – номер канала (см. рис. 9.43).

9.5.3.10. При $CCnS$, равном 00, канал работает в режиме генерации сигнала с программно-задаваемыми типом (ШИМ-, ЧИМ и т. п.) и параметрами (см. далее). Выдача сигнала производится на вывод $TIMx_CHn$.

Важно отметить, что соответствующий вывод ПВВ должен быть не только настроен на выполнение альтернативной функции $TIMxCHn$, но и программно сконфигурирован как выход. Режим переключения выхода OCn (см. рис. 9.41) задается битовым полем $OCnM$ (см. рис. 9.43). Наиболее часто используемыми на практике являются следующие режимы переключения:

- *Toggle* ($OCnM = 011$), при котором состояние выхода изменяется на противоположное при каждом совпадении содержимых счетчика и регистра $CCRn$ (данный режим удобен для реализации ЧИМ, см. пункт 9.3.4);

- принудительная установка выхода в пассивное ($OCnM = 100$) или в активное ($OCnM = 101$) состояние; используется для форсированного перевода выхода канала в «безопасное» состояние в нештатных ситуациях, аналогично установке в активное состояние битов FOC в регистрах управления таймерами МК семейства AVR (см. рис. 9.34 и 9.35в и пояснения к ним);

- ШИМ, вариант 1 ($OCnM = 110$) – уровень выходного сигнала активен при содержимом счетчика, меньшем $CCRn$, пассивен в противном случае;

- ШИМ, вариант 2 ($OCnM = 111$) – уровень выходного сигнала активен при содержимом счетчика, большем $CCRn$, пассивен в противном случае;

при этом активный уровень сигнала задается битом $CCnP$ регистра $CCER$ (см. подпункт 9.5.3.12).

Единичное состояние бита $OCnCE$ разрешает принудительный сброс сигнала $OCnREF$ по «высокому» уровню сигнала $ETRF$ (см. рис. 9.41). Данная функция может быть использована для форсированного перевода выхода канала в пассивное состояние при аварийных ситуациях (например, по срабатывании подключенного ко входу $TIMx_ETR$ МК детектора перегрузки устройства,

управляемого ШИМ-сигналом с выхода n -го канала таймера). При этом, поскольку управление переключением происходит полностью на аппаратном уровне, задержка между моментом возникновения аварийной ситуации и моментом сброса сигнала $OCnREF$ будет значительно меньше, чем при программном сбросе записью кода 100 или 101 в битовое поле $OCnM$ (см. выше).

Битом **$OCnPE$** осуществляется разрешение (при $OCnPE = 1$) или запрет (при $OCnPE = 0$) **буферизации** регистра $CCRn$. Если буферизация разрешена, что рекомендуется (см. пункт 9.3.3), обновление содержимого $CCRn$, как и регистра ARR (см. подпункт 9.5.3.5), происходит по событию «Обновление таймера».

Заметим, что в режиме реверсивного счета данное событие происходит как по переполнению, так по обнулению счетчика. С другой стороны, для реализации ШИМ, корректной по фазе (см. подпункт 9.3.3.3) необходимо обеспечить обновление содержимого $CCRn$ **только** по переполнению. Один из вариантов решения данной задачи – производить запись нового значения в буфер регистра $CCRn$ только на интервале прямого счета, для чего:

- использовать режим реверсивного счета с генерацией запроса на прерывание по совпадению содержимых счетчика и $CCRn$ только на интервале прямого счета ($CMS = 10$, см. подпункт 9.5.3.5);
- записывать новое значение в буфер $CCRn$ по данному прерыванию.

Реальное обновление содержимого регистра $CCRn$ при этом произойдет по ближайшему во времени событию «Обновление таймера», т. е. по переполнению счетчика («на вершине», см. рис. 9.10, с учетом того, что регистр OCR функционально эквивалентен регистру CCR).

Необходимо также напомнить, что посредством таймеров категории «*General-purpose*» возможна генерация **одинокных импульсов** по некоторому событию, подробнее – см. пункт 9.3.6.

9.5.3.11. При $CCnS$, не равном 00, канал настраивается на режим захвата содержимого счетчика по перепаду сигнала на одном из входов, в зависимости от конкретного значения битового поля $CCnS$. В режиме захвата параметры цифрового фильтра входного сигнала (частота и длина выборки) задаются битовым полем $ICnF$, а

коэффициент деления предделителя канала захвата (см. рис. 9.41) – битовым полем *ICnPSC*. Подробнее – см. [15]. Специализированный вариант режима захвата, «**Исключающее ИЛИ по входам**», рассмотрен в подпункте 9.5.3.19.

9.5.3.12. В регистре *CCER* под конфигурирование каждого из каналов выделено 3-битовое поле, включающее в себя:

- бит *CCnE* (*Capture / Compare n Enable*, где *n* – номер канала), единичное состояние которого разрешает выдачу сигнала на выход (если канал сконфигурирован на генерацию сигнала); или захват содержимого счетчика по фронту или по спаду входного сигнала (в зависимости от состояния бита *CCnP*, см. далее), если канал сконфигурирован на ввод;

- биты *CCnP* (*Capture / Compare n Polarity*) и *CCnNP*; если канал сконфигурирован на генерацию сигнала, бит *CCnP* задает активный уровень выхода, бит *CCnNP* должен быть равен нулю; если канал сконфигурирован на функцию захвата, биты *CCnP* и *CCnNP* определяют, по какому из перепадов входного сигнала происходит захват (см. пункт 14.4.9 руководства [15]).

9.5.3.13. Более подробное описание форматов регистров *CCER*, *CCMR1* и *CCMR2*, в т. ч. назначения их битов / битовых полей представлено в пунктах 14.4.7 – 14.4.9 руководства [15].

9.5.3.14. Таймеры категории «*General-purpose*» могут служить источниками **запросов** как прерываний, так и прямого доступа к памяти. В таймере *TIM3* МК модельного ряда *STM32F030xx* запросы могут генерироваться по следующим событиям [15]:

- «Обновление таймера» (запрос на прерывание / ПДП по данному событию обозначен на рис. 9.41 как *UI*);

- «Сравнение / захват по *n*-му каналу» (на рис. 9.41 запросы по этим событию обозначены как *CCnI*, *n* = 1...4);

- поступление импульса запуска от внешнего по отношению к МК источника или от ведущего таймера (запрос по данному событию обозначен на рис. 9.41 как *TGI*).

Разрешение / запрет запросов на прерывания и на ПДП по каждому из перечисленных событий осуществляются посредством регистра *DIER* (*DMA / Interrupt Enable Register*) таймера. В нем выделено по одному биту для разрешения (единицей) или запрета

(нулем) запроса на прерывание по каждому из данных событий и по одному – для разрешения / запрета запроса на ПДП по каждому из них. Факты возникновения каждого из событий отображаются в регистре статуса таймера.

9.5.3.15. Для таймеров категории «*General-purpose*», как и для большинства функциональных блоков МК семейства *ARM Cortex-Mx*, выделяется только **одна позиция** в таблице векторов прерываний по всем событиям, которые могут инициировать запрос на прерывание от соответствующего блока, в данном случае – таймера (см., например, позицию *TIM5 global interrupt* на рис. 7.8). Таким образом, запросы от некоторого таймера по каждому из событий, прерывания по которым разрешены, вызовут переход к одной и той же подпрограмме обслуживания. В нее при этом должен быть включен опрос признаков данных событий, с целью корректного ее выполнения. После опроса признаки должны быть сброшены, для предотвращения автоматического возобновления выполнения подпрограммы обслуживания прерывания после выхода из нее. См. подпункты 7.3.2.15 и 7.3.2.16.

9.5.3.16. Простой пример **использования запросов ПДП** от таймера приведен в пункте 11.2.3 источника [9]: включение / выключение светодиодов по переполнении счетчика таймера. В данном примере (см. также подпункт 8.3.1.11):

- событием, инициирующим запросы на ПДП, является обновление таймера;
- в качестве адреса ПУ выступает адрес ПВВ, к которому подключены светодиоды;
- базовым адресом ПД служит начальный адрес массива кодов, управляющих светодиодами;
- направление передачи – ПД → ПУ;
- режим работы – циклический.

Более сложный пример использования запросов ПДП от таймера (загрузка из памяти в режиме ПДП содержимого регистров *CCRn* по событию «Обновление таймера») приведен в пункте 14.4.18 руководства [15].

9.5.3.17. Генерация запросов на прерывания и на ПДП по событиям, перечисленным в подпункте 9.5.3.14, может также

инициироваться **программно**, установкой в единичное состояние соответственно битов *UG*, *CC1G*, *CC2G*, *CC3G*, *CC4G* и *TG* в регистре *EGR* (*Event Generation Register*) таймера. При записи единицы в какой-либо из этих битов вырабатывается запрос на прерывание / ПДП по соответствующему событию (если он разрешен), а также устанавливается в единицу его признак в регистре статуса таймера. Например, при записи единицы в бит *UG* регистра *EGR* генерируется (если он разрешен) запрос на прерывание или на ПДП по событию «Обновление таймера» и устанавливается в единичное состояние признак данного события (*UIF*) в регистре статуса таймера. Следует отметить, что:

- при установке в единицу бита *UG* регистра *EGR* и нулевом состоянии бита *URS* в 1-м регистре управления таймером реально произойдет обновление таймера;

- если *n*-й канал таймера сконфигурирован на режим захвата, то при записи единицы в бит *CCnG* регистра *EGR* реально происходит загрузка содержимого счетчика в регистр *CCRn*.

Программная инициализация запросов на прерывание / на ПДП по соответствующим событиям, в частности, может быть использована в нештатных ситуациях, для форсированного выполнения процедур, подлежащих реализации при возникновении данных событий.

9.5.3.18. Необходимо также остановиться на 2-х **специализированных** режимах работы таймеров общего назначения.

Первый из них называется режимом интерфейса с **поворотными инкрементными энкодерами** (*Encoder interface* на рис. 9.41) [9, 13]. Они представляют собой широко применяемые в системах контроля и управления электромеханические, оптические или электромагнитные датчики, преобразующие угловую или линейную скорость объекта в прямо пропорциональную ей частоту импульсов, на основе которой, в свою очередь, может быть получена информация о положении, угловом или линейном перемещении объекта [51]. Как правило, поворотные инкрементные энкодеры снабжаются 2-мя выходами (*A* и *B*), на которых формируются две последовательности импульсов с одинаковой частотой и с взаимным фазовым сдвигом, модуль которого равен 90°, а знак указывает

- битовыми полями *IC1F* и *IC2F* регистра *CCMRI* должны быть установлены необходимые параметры входных фильтров 1-го и 2-го каналов (см. рис. 9.41 и 9.43);

- посредством битового поля *SMS* регистра *SMCR* (см. подпункт 9.5.3.6) должен быть выбран один из режимов тактирования *Encoder mode* 1, 2 или 3, которым соответствуют значения *SMS*, равные 001, 010 и 011;

- один из каналов какого-либо другого таймера должен быть настроен на режим захвата; на его вход захвата необходимо подать сигнал с выхода канала *A* или *B* энкодера.

При выполнении всех вышеперечисленных условий:

- в режиме *Encoder mode* 1 счетчик таймера срабатывает по **каждому** перепаду (и по фронту, и по спаду) сигнала на входе 1-го канала таймера (см. рис. 9.45а), в режиме *Encoder mode* 2 – на входе 2-го канала, в режиме *Encoder mode* 3 – на входах обоих каналов (см. рис. 9.45б);

- если направление вращения / перемещения объекта соответствует временным диаграммам, приведенным на рис. 9.44а (выходной сигнал канала *B* опережает по фазе выходной сигнал канала *A*), в любом из вариантов режима *Encoder mode* счетчик таймера производит обратный счет, а бит *DIR* 1-го регистра **автоматически** устанавливается в единичное состояние, соответствующее работе счетчика на вычитание (см. рис. 9.45);

- если направление вращения / перемещения объекта соответствует временным диаграммам, приведенным на рис. 9.44б (выходной сигнал канала *A* опережает по фазе выходной сигнал канала *B*), в любом из вариантов режима *Encoder mode* счетчик таймера производит прямой счет, а бит *DIR* 1-го регистра **автоматически** устанавливается в нулевое состояние, соответствующее работе счетчика на суммирование (см. рис. 9.45);

- направление вращения / перемещения объекта определяется программным опросом бита *DIR*;

- скорость вращения / перемещения определяется посредством преобразования в код периода сигнала на выходе канала *A* или *B* энкодера; преобразование осуществляется с помощью другого таймера, на вход захвата которого подан соответствующий сигнал; способ преобразования описан в пункте 9.4.3.

Более подробно вопросы практической реализации интерфейса МК с поворотными инкрементными энкодерами рассмотрены, например, в пункте 14.3.12 руководства [15], а также в пункте 11.3.9 источника [9].

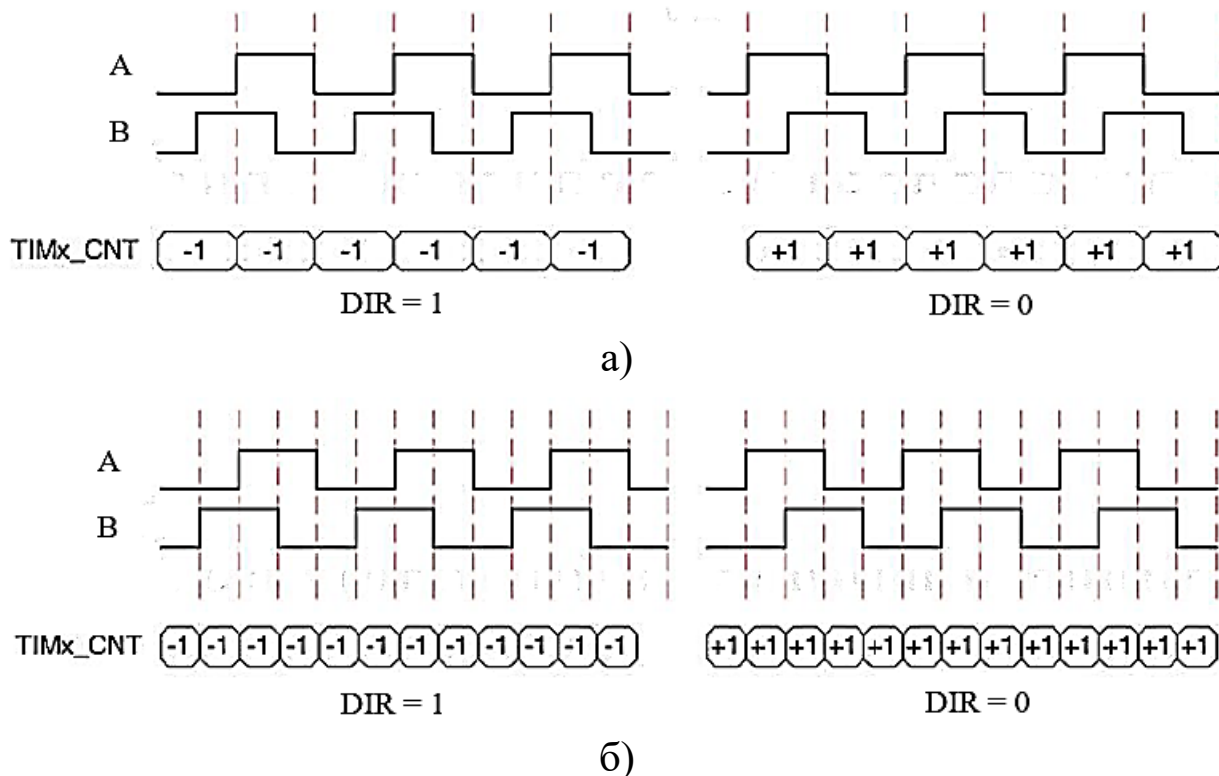


Рис. 9.45. Временные диаграммы работы счетчика таймера в режимах *Encoder mode 1* (а) и *Encoder mode 3* (б) (см. пояснения в тексте)

9.5.3.19. Второй из подлежащих рассмотрению специализированных режимов работы таймеров категории «*General-purpose*» - режим «Исключающее ИЛИ (*XOR*) по входам» [9, 13].

Данный режим применяется, в первую очередь, для контроля и управления бесщеточными трехфазными электродвигателями постоянного тока (*Brush-Less DC motors, BLDC*), в качестве датчиков положения ротора которых используются датчики Холла [51], по одному на каждую фазу (откуда другое название данного режима – режим интерфейса с датчиками Холла). Выходные сигналы датчиков подключаются ко входам 1-го, 2-го и 3-го каналов таймера. Сигнал, полученный в результате их объединения посредством элемента «Исключающее ИЛИ» (*XOR*) (см. рис. 9.41), служит входным сигналом 1-го канала, который при этом конфигурируется

на режим захвата по каждому из перепадов на выходе элемента *XOR*.
Для этого:

- бит *TIIS* (*TI Selection*) 2-го регистра управления таймером устанавливается в единичное состояние, при котором в качестве входного сигнала 1-го канала таймера, *TI1* (см. рис. 9.41) выступает выходной сигнал элемента *XOR*;

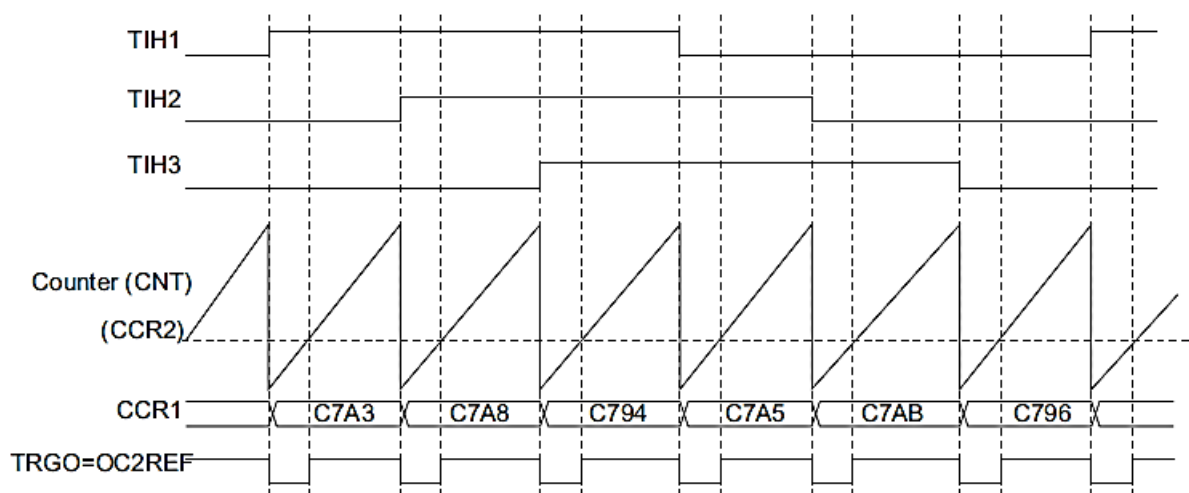
- в битовое поле *CCIS* регистра *CCMR1* (см. рис. 9.43 и пояснения к нему) записывается кодовая комбинация 11, которая соответствует режиму захвата по перепадам сигнала *TRC* (см. рис. 9.41);

- в качестве источника сигнала *TRC* выбирается (записью кода 101 в битовое поле *TS* регистра *SMCR*, см. подпункт 9.5.3.6) блок фильтра и детектора перепадов 1-го канала (*Input filter & edge detector* на рис. 9.41), вход которого при *TIIS* = 1 (см. выше) подключен к выходу элемента *XOR*;

- в битовое поле *SMS* регистра *SMCR* (см. подпункт 9.5.3.6) записывается код 100, что соответствует обновлению таймера по сигналу *TRGI*, при вышеуказанных содержимых битовых полей *TIIS*, *CCIS* и *TS* – по перепадам выходного сигнала элемента *XOR*; тактирование счетчика таймера при этом осуществляется синхросигналом домена, к которому он принадлежит (*APB1*).

Пример временных диаграмм работы таймера в режиме «Исключающее ИЛИ по входам» при вышеописанном профиле конфигурации приведены на рис. 9.46 [15]. По каждому изменению состояния сигнала на выходе каждого из датчиков Холла происходит сброс счетчика (при нулевом состоянии бита *URS*, см. подпункт 9.5.3.5 – также обновление таймера), а в регистре *CCR1* фиксируется содержимое счетчика, достигнутое за интервал времени от предыдущего перепада сигнала на выходе какого-либо из датчиков. Оно несет информацию о скорости вращения двигателя и о положении ротора. В данном примере также, по 2-му каналу таймера, осуществляется генерация импульса, инициирующего изменение параметров конфигурации таймера категории «*Advanced-control*», управляющего двигателем (подробнее – см. подпункт 9.5.3.23).

Более подробно вопросы практической реализации режима «Исключающее ИЛИ по входам» рассмотрены, например, в пункте 13.3.17 руководства [15], а также в подпункте 11.3.10.1 источника [9].



$TIH1 \dots TIH3$ – выходные сигналы датчиков Холла

Рис. 9.46. Пример временных диаграмм работы таймера категории «*General-purpose*» МК модельного ряда *STM32F030xx* в режиме «Исключающее ИЛИ по входам» [15]

9.5.3.20. Более подробное описание структуры и архитектуры таймеров категории «*General-purpose*» различных подсемейств / модельных рядов МК семейства *ARM Cortex-Mx* приводится в руководствах по применению соответствующих модельных рядов МК (см, например, [13 – 15]).

9.5.3.21. Типовой пример структурной схемы таймера категории «*Advanced-control*» МК семейства *ARM Cortex-Mx* приведен на рис. 9.47 [15].

Примечание. Регистры *ARR*, *CCR*, *PSC* и *REP* буферизированы (что не отражено на рис. 9.47 в явном виде). Обновление их содержимого осуществляется по событию «Обновление таймера», аналогично буферизированным регистрам таймеров категорий «*Basic*» и «*General-purpose*». Буферизация регистров *ARR* и *CCRn* может быть разрешена или запрещена (что не рекомендуется) программно: буферизация регистра *ARR* – установкой / сбросом бита

ARPE в 1-м регистре управления; буферизация регистров *CCRn* – установкой / сбросом битов *OCnPE* в регистрах *CCMR* (см. рис. 9.43 и пояснения к нему).

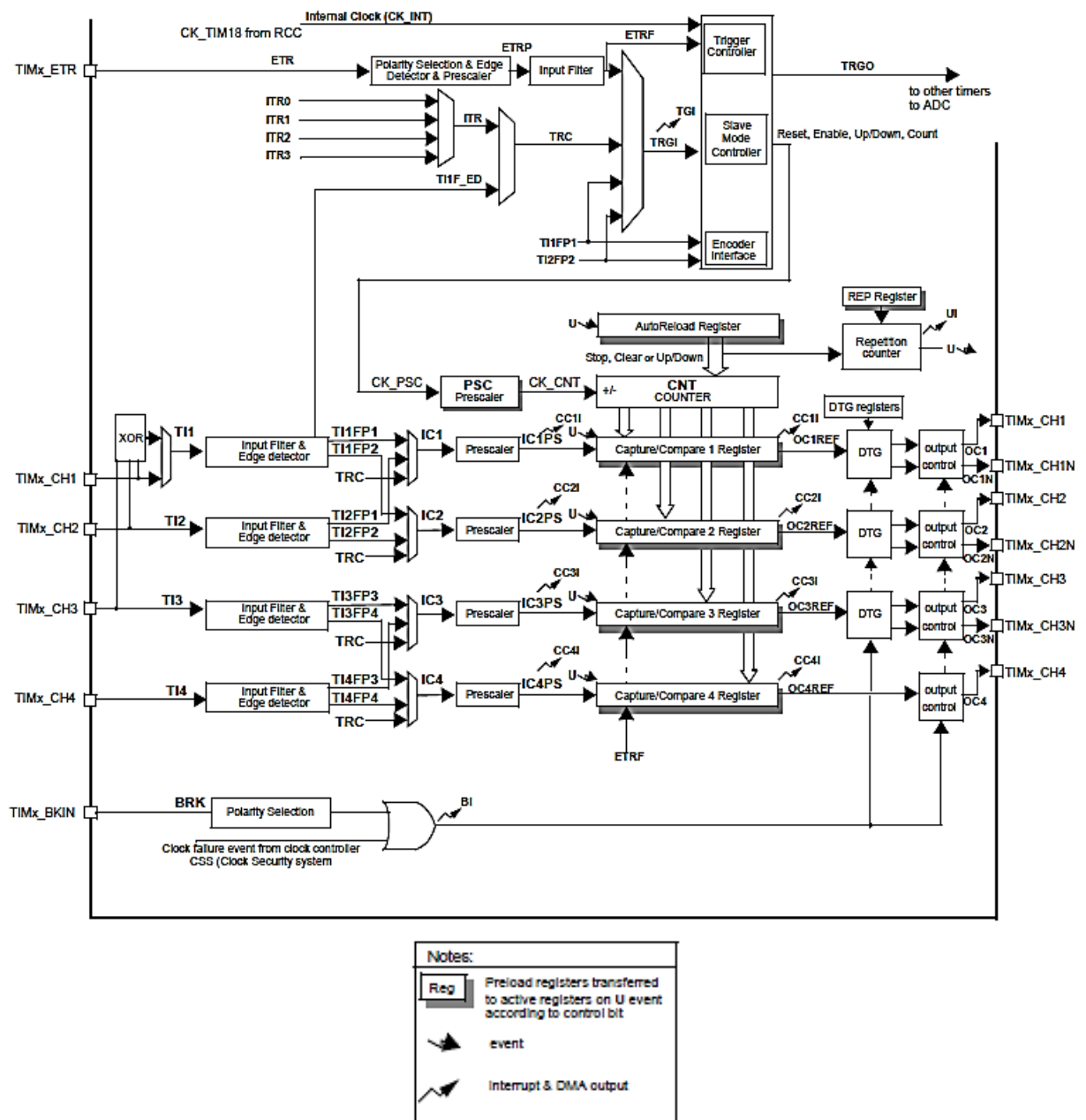


Рис. 9.47. Типовая структурная схема таймера категории «*General-purpose*» МК семейства *ARM Cortex-Mx* (на примере таймера *TIM1* МК модельного ряда *STM32F030xx* [15]).

Пояснения приведены в тексте

Базовые структурно-архитектурные решения таймеров категорий «*Advanced-control*» и «*General-purpose*» (см. подпункты 9.5.3.4 – 9.5.3.19), в целом, аналогичны. Необходимо остановиться

только на основных отличительных особенностях структуры и архитектуры таймеров категории «*Advanced-control*». К ним относятся (см. рис. 9.47):

- наличие **счетчика повторов** (*Repetition counter*) (см. подпункт 9.5.3.22);

- возможность формирования (по 3-м из 4-х каналов) **противофазных сигналов** с программно-задаваемым «**мертвым временем**» между переключениями (см. подпункт 9.5.3.23);

- возможность **аварийной остановки** таймера на аппаратном уровне (без участия ПО), с принудительным переводом выходов каналов таймера в пассивное (безопасное) состояние (см. подпункт 9.5.3.24).

9.5.3.22. Таймеры категории «*Advanced-control*» снабжены 8-битовым **счетчиком повторов** (*Repetition counter* на рис. 9.47), работающим в режиме обратного счета и позволяющим расширить (в сторону увеличения) диапазон интервалов времени, формируемых таймером (см. подраздел 9.2). Обновление таймеров категории «*Advanced-control*», в отличие от таймеров категорий «*Basic*» и «*General-purpose*», происходит не по переполнению / потере значимости счетчика таймера, а по потере значимости (*Underflow*) счетчиком повторов, содержимое которого уменьшается на единицу (декрементируется):

- по каждому переполнению счетчика таймера, если он работает в режиме прямого счета;

- по каждой потере значимости счетчиком таймера, если он работает в режиме обратного счета;

- по каждой потере значимости и по каждому переполнению счетчика таймера, если он работает в реверсивном режиме (*Center-aligned*).

Таким образом, если счетчик таймера работает в режиме непрерывного одностороннего счета, обновление таймера происходит с периодом, равным:

$$T_U = \frac{(ARR + 1)(RCR + 1)}{f_{CK_CNT}}; \quad (9.40)$$

т. е. только по каждому $(RCR + 1)$ -му переполнению или, соответственно, потере значимости счетчика таймера, где RCR – содержимое 8-битового программно-доступного буферизированного регистра *REP Register*, задающего верхний предел счета счетчика повторов (см. рис. 9.47). В реверсивном режиме работы счетчика таймера обновление таймера происходит с периодом:

$$T_U = \frac{ARR(RCR + 1)}{f_{CK_CNT}}. \quad (9.41)$$

Таким образом, при прочих равных условиях, посредством таймера категории «*Advanced-control*» может быть сформирован интервал времени с длительностью, в $RCR + 1$ раз большей, чем посредством таймера категории «*General-purpose*».

Нетрудно увидеть, что при содержимом RCR , равном нулю, период обновления таймера категории «*Advanced-control*» равен периоду обновления таймера категории «*General-purpose*» при тех же значениях f_{CK_CNT} и ARR .

9.5.3.23. Следующим важным элементом архитектуры таймеров категории «*Advanced-control*» является возможность формирования (по 3-м из 4-х каналов) **противофазных сигналов** с программно-задаваемым «мертвым временем» между переключениями. Данная возможность особенно востребована в задачах управления электроприводом (см. подпункты 9.3.3.5 и 9.3.3.6). Общие вопросы реализации данной функции в таймерах категории «*Advanced-control*» изложены в подпункте 9.3.3.6. Необходимо только дополнить их некоторыми подробностями.

1. Значение t_{delay} (см. рис. 9.15) задается битовым полем DTG регистра *BDTR* (*Break & Dead Time Register*). Оно может находиться в пределах от 0 до $1008T_{DTS}$, где T_{DTS} (*Dead – Time & Sampling clock*) – дискретность задания «мертвого» времени и периода выборки цифровых фильтров таймера (см. рис. 9.47). Значение T_{DTS} может быть равно одному, 2-м или 4-м периодам синхросигнала домена МК, к которому относится таймер (сигнала CK_INT на рис. 9.47). Подробности – см. пункты 13.4.1 и 13.4.18 руководства [15].

2. Режимы работы прямого (OCx) и инверсного ($OCxN$) выходов канала (см. рис. 9.15 и 9.47) задаются независимо один от другого, состояниями битов $CCxE$, $CCxP$, $CCxNE$ и $CCxNP$ регистра *Capture / Compare Enable Register*, $CCER$, а также битов MOE , $OSSR$ и $OSSI$ регистра $BDTR$ (см. пункты 13.4.9 и 13.4.18 руководства [15]). В отличие от аналогичных по назначению битов регистра $CCER$ МК категории «*General-purpose*», биты $CCxE$ и $CCxNE$ буферизированы. Также буферизированы битовые поля $OCxM$ регистров $CCMR1$ и $CCMR2$, формат и назначение битов которых аналогичен представленному на рис. 9.43 формату одноименных регистров таймеров категории «*General-purpose*». Буферизация может быть запрещена обнулением бита $CCPC$ 2-го регистра управления таймером ($CR2$). Если она разрешена, обновление битов $CCxE$, $CCxNE$ и битового поля $OCxM$ происходит:

- если бит $CCUS$ регистра $CR2$ установлен – по событию «*Commutation event*» (COM) – фронту (перепаду из 0 в 1) сигнала $TRGI$ (см. рис. 9.47), или при программной установке в единицу бита $COMG$ регистра EGR (*Event Generation Register*);
- если бит $CCUS$ сброшен – только при программной установке в единицу бита $COMG$.

В зависимости от состояния битов $CCxE$, $CCxP$, $CCxNE$, $CCxNP$, AOE , $OSSR$ и $OSSI$ режимы работы выходов OCx и $OCxN$ могут быть следующими:

- выход отключен от выхода таймера и находится в нулевом состоянии;
- выход подключен, но находится в неактивном состоянии (нулевом или единичном, в зависимости от состояния бита $CCxP$ или, соответственно, $CCxNP$ регистра *Capture / Compare Enable Register*, $CCER$);
- выход активен и управляется сигналом $OCxREF$ таймера (см. рис. 9.47), в соответствии с режимом генерации сигнала, задаваемым битовым полем $OCxM$ регистра $CCMR1$ или $CCMR2$.

Зависимости режимов работы выходов OCx и $OCxN$ от состояний битов $CCxE$, $CCxP$, $CCxNE$, $CCxNP$, AOE , $OSSR$ и $OSSI$ представлены в табл. 46 руководства [15].

В частности, временные диаграммы, приведенные на рис. 9.15, соответствуют активному режиму работы как выхода OCx , так и $OCxN$, причем активное состояние каждого из них – единичное ($CCxP = CCxNP = 0$).

Показательным примером применения различных сочетаний режимов работы выходов OCx и $OCxN$ является управление $BLDC$ -двигателем по результатам контроля положения ротора и скорости его вращения посредством датчиков Холла (см. подпункт 9.5.3.19). Пример временных диаграмм контроля и управления приведен на рис. 9.48 [15].

Контроль осуществляется «интерфейсным» таймером категории «*General-purpose*» в режиме «Исключающее ИЛИ по входам» (см. рис. 9.46), а управление двигателем – посредством таймера категории «*Advanced-control*», являющегося ведомым по отношению к интерфейсному таймеру. Последний, по каждому перепаду выходного сигнала каждого из датчиков Холла, фиксирует в регистре $CCR1$ содержимое счетчика, несущее информацию о положении и скорости вращения ротора, а также генерирует импульс $TRGO$ синхронизации ведомого, поступающий на один из входов ITR (см. рис. 9.47). Перепад из 0 в 1 данного импульса служит источником события «*Commutation event*» (COM) для ведомого таймера (см. рис. 9.48), по которому обновляются биты, задающие режимы работы выходов OCx и $OCxN$, управляющих двигателем. Управление осуществляется ШИМ-сигналами с постоянной составляющей (т. е. с коэффициентом заполнения), являющейся функцией от текущей скорости вращения и положения ротора, определяемых по содержимому регистра $CCR1$ (см. рис. 9.48).

При этом (см. табл. 46 руководства [15]):

- биты MOE , $OSSR$ и $OSSI$ регистра $BDTR$ ведомого таймера установлены в единицу;

- $CC1NE = CC2NE = CC3NE = 0$ (инверсные выходы каналов находятся в неактивном состоянии);

- на интервале времени от t_1 до t_2 $CC1E = 1$ (прямой выход 1-го канала активен, и работает в режиме ШИМ); $CC1NP = 0$ (неактивный инверсный выход 1-го канала находится в нулевом состоянии); $CC2E = CC2P = 0$ (прямой выход 2-го канала не активен и находится

в нулевом состоянии); $CC2NP = 1$ (инверсный выход 2-го канала находится в единичном состоянии); $CC3E = CC3P = 0$ (прямой выход 3-го канала не активен и находится в нулевом состоянии); $CC3NP = 0$ (инверсный выход 3-го канала находится в нулевом состоянии);

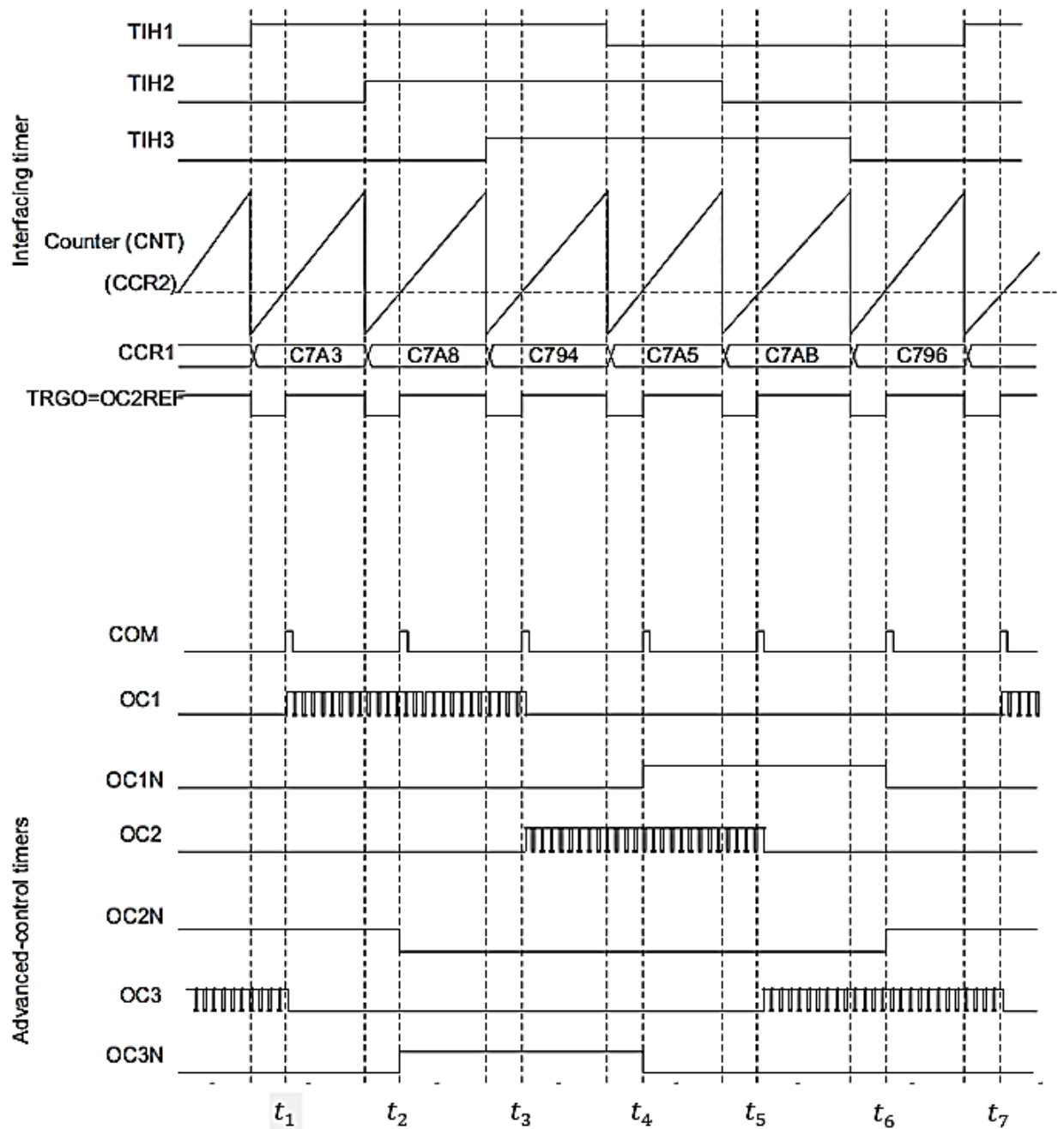


Рис. 9.48. Пример временных диаграмм контроля и управления *BLDC*-двигателем [15] (см. пояснения в тексте)

- на интервале времени от t_2 до t_3 изменяются, по сравнению с интервалом от t_1 до t_2 , только состояния инверсных выходов 2-го и 3-го канала; инверсный выход 2-го канала переключается в ноль

(обнулением бита *CC2NP*), 3-го канала – в единицу (установкой в единицу бита *CC3NP*);
и так далее.

Более подробное изложение вопросов формирования противофазных сигналов таймерами категории «*Advanced-control*» приведено в руководствах по применению МК соответствующих модельных рядов (см., например, [13 – 15]).

9.5.3.24. Еще одним элементом архитектуры таймеров категории «*Advanced-control*», отсутствующим у таймеров категорий «*Basic*» и «*General-purpose*», является возможность **аварийной остановки** таймера (*Break function*) на аппаратном уровне, без участия ПО, по некоторому внешнему по отношению к МК событию (например, по срабатыванию детектора перегрузки), или по сбою системы синхронизации (*Clock failure*, см. подпункт 4.4.2.6). При аварийной остановке выходы каналов таймера автоматически переводятся в пассивное (безопасное) состояние, задаваемое программно при конфигурировании таймера. Функция аварийной остановки, как и генерация противофазных управляющих сигналов (см. подпункт 9.5.3.23), ориентирована, в первую очередь, на применение в системах управления электроприводом.

Разрешение / запрет аварийной остановки осуществляется записью единицы или, соответственно, нуля в бит *BKE* (*Break Enable*) регистра *BDTR*. Если данный бит находится в единичном состоянии, то при активном уровне сигнала на входе *BKIN* таймера (задаваемом значением бита *BKP* регистра *BDTR*) или при генерации события «*Clock failure*» блоком *CSS* (см. подпункт 4.4.2.6) происходит следующее [15]:

- если задействованы только прямые выходы (*OCx*), они автоматически устанавливаются в состояния, указанные в битах *OISx* (*Output x Idle State*) 2-го регистра управления таймером, независимо от того, работает ли ГТИ МК или нет;

- если задействованы и прямые (*OCx*), и инверсные (*OCxN*) выходы, они устанавливаются в неактивное состояние, которое равно значениям битов *CCxP* и *CCxNP* регистра *CCER*; затем, если работа подсистемы синхронизации МК не нарушена, то, по истечении интервала времени t_{delay} после момента аварийной

остановки, выходы OCx и $OCxN$ устанавливаются в состояния, указанные в битах $OISx$ и $OISxN$ 2-го регистра управления таймером.

При этом значения $CCxP$, $CCxNP$, $OISx$ и $OISxN$ должны быть выбраны в зависимости от конкретной схемы управляемого устройства, таким образом, чтобы при установлении выходов OCx и $OCxN$ в состояния $CCxP$ и $CCxNP$ или, соответственно, $OISx$ и $OISxN$ был обеспечен максимально безопасный режим его работы.

Также по наступлении события, вызывающего аварийную остановку:

- в регистре состояния таймера устанавливается в единицу признак аварийного останова (BIF);

- если разрешены прерывания по аварийной остановке, т. е. при единичном состоянии бита BIE в регистре $DIER$ ($DMA / Interrupt Enable Register$), то генерируется запрос на прерывание.

Подробнее о функции аварийной остановки в таймерах категории «*Advanced-control*» - см., например, пункт 13.3.12 руководства [15].

9.5.3.25. Источниками прерываний от таймеров категории «*Advanced-control*» МК модельного ряда $STM32F030xx$ могут служить следующие события (см. рис. 9.47):

- «Обновление таймера» (UI);
- «Сравнение / захват по n -му каналу» ($CCnI$, $n = 1 \dots 4$);
- поступление импульса запуска от внешнего по отношению к МК источника или от ведущего таймера (TGI);
- событие «*Commutation event*» ($COMI$), см. подпункт 9.5.3.23;
- аварийная остановка таймера (BI).

Те же события, кроме аварийной остановки, могут служить источниками запросов на ПДП.

Разрешение / запрет прерываний и запросов на ПДП осуществляется программной установкой / сбросом соответствующих битов разрешения в регистре $DIER$ ($DMA / Interrupt Enable Register$); факты возникновения каждого из событий отображаются в регистре статуса таймера (см. пункты 13.4.4 и 13.4.5 руководства [15]).

Как и в таймерах категории «*General-purpose*» (см. подпункт 9.5.3.17), генерация запросов на прерывания и на ПДП по

вышеперечисленным событиям может также инициироваться **программно**, установкой в единичное состояние соответствующих битов в регистре *EGR (Event Generation Register)* таймера (см. пункт 13.4.6 руководства [15]). При записи единицы в какой-либо из этих битов вырабатывается запрос на прерывание / ПДП по соответствующему событию (если он разрешен), а также устанавливается в единицу его признак в регистре статуса таймера.

Необходимо отметить, что в таблице векторов прерываний МК семейства *ARM Cortex-Mx* прерываниям от таймеров категории «*Advanced-control*», в отличие от категории «*General-purpose*», выделяется несколько позиций, отдельно для прерываний по каждому из событий / группе событий. См. приведенные на рис. 9.49 фрагменты таблиц векторов прерываний МК модельных рядов *STM32F030xx* [15] и *STM32F10xx* [13].

Position	Priority	Type of priority	Acronym	Description	Address
13	20	settable	TIM1_BRK_UP_TRG_COM	TIM1 break, update, trigger and commutation interrupt	0x0000 0074
14	21	settable	TIM1_CC	TIM1 capture compare interrupt	0x0000 0078

а)

Position	Priority	Type of priority	Acronym	Description	Address
43	50	settable	TIM8_BRK	TIM8 Break interrupt	0x0000_00EC
44	51	settable	TIM8_UP	TIM8 Update interrupt	0x0000_00F0
45	52	settable	TIM8_TRG_COM	TIM8 Trigger and Commutation interrupts	0x0000_00F4
46	53	settable	TIM8_CC	TIM8 Capture Compare interrupt	0x0000_00F8

б)

Рис. 9.49. Фрагменты таблиц векторов прерываний МК модельных рядов *STM32F030xx* [15] (а) и *STM32F10xx* [13] (б)
(*TIM1*, *TIM8* – таймеры категории «*Advanced-control*»)

9.5.3.26. Следует также отметить, что ряд элементов архитектуры таймеров категории «*Advanced-control*» присутствует и в некоторых таймерах категории «*General-purpose*» (см., например, раздел 17 руководства [15]).

9.5.3.27. Более подробное описание структуры и архитектуры таймеров категории «*Advanced-control*» различных подсемейств / модельных рядов МК семейства *ARM Cortex-Mx* приводится в руководствах по применению соответствующих модельных рядов МК (см, например, [13 – 15]).

9.5.4. Примеры программирования таймеров МК семейства *ARM Cortex-Mx*

В данном пункте приведены примеры программных модулей реализации наиболее распространенных типовых функций таймеров МК семейства *ARM Cortex-Mx* (см. пункт 9.1):

- формирования программно задаваемой задержки между двумя событиями;
- формирования сигналов с программно задаваемыми частотно-временными параметрами, в том числе ШИМ-сигналов и одиночных импульсов;
- преобразования в код длительности импульсов.

Уровень программирования – *CMSIS*, представление операндов – числовое (см. подпункт 2.6.1.14 и Приложение В).

9.5.4.1. Пример 1. Формирование программно задаваемой задержки между двумя событиями.

В качестве простого примера формирования программно задаваемой задержки между двумя событиями рассмотрим программный модуль включения / выключения светодиода с интервалами времени длительностью 1 с. Допустимая погрешность формирования задержки между включением / выключением - ± 50 мс. Модель МК - *STM32F030F4* модельного ряда *STM32F030xx* (см. пункт 9.5.3). Для формирования задержки выделен таймер *TIM3* категории «*General-purpose*». МК в целом, в том числе *TIM3* тактируется ГТИ с внешним ПЭР частотой 8 МГц. Светодиод подключен к 4-му выводу ПВВ *A*; включение светодиода

осуществляется подачей единицы на данный вывод, выключение – подачей нуля.

Для формирования задержки будем использовать алгоритм, описанный в подразделе 9.2, несколько модифицированный с учетом особенностей архитектуры таймеров МК семейства *ARM Cortex-Mx*. Как указано ранее (см. подпункт 9.5.3.2), верхний предел счета их счетчиков (N_{TOP}) равен содержимому регистра ARR , и может задаваться программно в широких пределах, от 1 до $2^{16} - 1$. Поэтому в ряде практических случаев (в т. ч. в рассматриваемом) длительность задержки рационально задавать через содержимое ARR . При этом счетчик работает в режиме прямого или обратного счета, в пределах от 0 до ARR или, соответственно, от ARR до 0, т. е. $N = 0$, $N_{TOP} = ARR$ (см. выражения (9.1) и (9.2)). Признаком окончания формируемого интервала времени служит переполнение счетчика или, соответственно, потеря значимости, а его длительность равна $(ARR + 1)/f_{CNT}$, где f_{CNT} – частота счетных импульсов.

Необходимо выбрать коэффициент деления предделителя счетчика, PSC , и значение ARR . С точки зрения энергопотребления, желательно выбрать частоту счетных импульсов, минимально необходимую для обеспечения заданной погрешности формирования задержки, т. е. для обеспечения условия $1/f_{CNT} \leq 50$ мс (см. подраздел 9.2). С другой стороны, желательно, чтобы для формирования импульсов с частотой f_{CNT} не потребовался дополнительный предделитель, кроме входящего в состав задействованного таймера. Также, поскольку в состав задействованного таймера не входит счетчик повторов, желательно, чтобы для формирования интервалов времени с заданной длительностью не потребовался программный подсчет повторов, т. е. чтобы она была меньше значения $(ARR_{max} + 1)/f_{CNT}$.

Разрядность предделителя таймера $TIM3$, а также его регистра ARR равна 16-ти битам, как и у большинства таймеров МК семейства *ARM Cortex-Mx*. Поэтому, при частоте сигнала CK_PSC (см. рис. 9.41), равной 8 МГц, минимально возможная частота счетных импульсов таймера равна $8 \text{ МГц}/2^{16} \approx 122 \text{ Гц}$. Значение $1/122 \text{ Гц} \ll 50 \text{ мс}$, поэтому при частоте счетных импульсов, равной

122 Гц, требуемая погрешность формирования задержек обеспечивается с запасом. С другой стороны, $ARR_{max} = 2^{16} - 1$, поэтому $(ARR_{max} + 1)/122 \text{ Гц} \gg 1 \text{ с}$, поэтому программный подсчет повторов при f_{CNT} , равной 122 Гц, также не требуется. Однако, длительность формируемого интервала времени не кратна значению $1/122 \text{ Гц}$. Поэтому удобнее выбрать ближайшую к 122 Гц частоту счетных импульсов, при которой данная длительность кратна их периоду. Такой является частота 125 Гц. При этом коэффициент деления предделителя счетчика должен быть равен 64000, а содержимое регистра $PSC - 63999$ (см. выражение (9.39)). Для обеспечения равенства $(ARR + 1)/125 \text{ Гц} = 1 \text{ с}$ значение ARR должно быть равно 124-м.

Для определенности, выберем режим прямого счета. Фиксацию моментов переполнения счетчика будем осуществлять по прерываниям, генерируемым по событию «Обновление таймера» (см. рис. 9.1). По каждому из прерываний состояние светодиода изменяется на противоположное.

Код программного модуля с необходимыми комментариями представлен ниже.

```
//Подключение файла описания МК модельного ряда stm32f0xx
#include "stm32f0xx.h"
/*
Объявление целочисленной переменной i, управляющей состоянием
светодиода
*/
int i;
/////////////////////////////////////////////////////////////////
//Основной блок программы
void main(void)
{
/*
Конфигурирование источника синхронизации МК. Код полностью аналогичен
приведенному в подпункте 4.4.2.4
*/
RCC->CR |= 0x00010000;
while((RCC->CR)&(0x00020000)==0);
RCC->CFGR &= 0xFFFFF0FC;
RCC->CFGR |= 0x00000001;
```

```
while(((RCC->CFGR)&(0x0000000C))!=0x00000004);
```

```
RCC->CR &= 0xFFFFFFF0;
```

```
//Конфигурирование ПБВ А, к которому подключен светодиод
```

```
/*
```

Разрешение тактирования ПБВ А установкой в единицу 17-го бита (*IOPAEN*) регистра *AHBENR* подсистемы управления синхронизацией и сбросом (см. пункт 7.4.6 руководства [15])

```
*/
```

```
RCC->AHBENR |= 0x00020000;
```

```
/*
```

Конфигурирование 4-го вывода ПБВ А, управляющего светодиодом, на работу в режиме цифрового выхода общего назначения, записью кода 01 в битовое поле *MODER4* регистра *GPIOA_MODER* (см. пункт 8.4.1 руководства [15])

```
*/
```

```
GPIOA->MODER |= 0x00000100;
```

```
//Задание начального значения переменной i
```

```
i=0;
```

```
/*
```

Разрешение прерываний от 3-го таймера (*TIM3 global interrupt*, см. позицию 16 в таблице векторов прерываний, пункт 11.1.3 руководства [15]) установкой в 16-го бита в регистре *ISER[0]* (*Interrupt Set-Enable Register 0*) контроллера прерываний (*NVIC*), см. пункт 4.2.2 руководства [19]

```
*/
```

```
NVIC->ISER[0] |= 0x00010000;
```

```
//Конфигурирование 3-го таймера
```

```
/*
```

Разрешение тактирования 3-го таймера установкой в единицу 1-го бита (*TIM3EN*) регистра *APB1ENR* подсистемы управления синхронизацией и сбросом (см. пункт 7.4.8 руководства [15]).

```
*/
```

```
RCC->APB1ENR |= 0x00000002;
```

```
/*
```

Запись в регистры *PSC* и *ARR* 3-го таймера ранее вычисленных значений коэффициента деления предделителя (уменьшенного на 1) и верхнего предела счета

```
*/
```

```
TIM3->PSC = 63999;
```

```
TIM3->ARR = 124;
```

```
/*
```

Разрешение буферизации регистра *ARR* установкой в единицу 7-го бита (*ARPE*) регистра *CR1* 3-го таймера (см. пункт 14.4.1 руководства [15]).

Примечание 1. Состояния 4-го бита (*DIR*) и битов 5-го и 6-го (битовое поле *CMS*) оставлены нулевыми («по умолчанию»), что соответствует режиму одностороннего прямого счета.

Примечание 2. В качестве входного сигнала предделителя счетчика выбран синхросигнал домена *APB1*, к которому относится *TIM3* (содержимое битового поля *SMS* регистра *TIM3_SMCR* оставлено нулевым, «по умолчанию»). В данном примере частота этого синхросигнала равна 8 МГц.

*/

```
TIM3->CR1 |= 0x00000080;
```

/*

Разрешение прерываний по обновлению 3-го таймера установкой в единицу 0-го бита (*UIE*, *Update Interrupt Enable*) регистра *TIM3_DIER* (см. пункт 14.4.4 руководства [15])

*/

```
TIM3->DIER |= 0x00000001;
```

/*

Запуск 3-го таймера установкой в единицу 0-го бита (*CEN*, *Counter Enable*) регистра *TIM3_CR1*

*/

```
TIM3->CR1 |= 0x00000001;
```

```
//«Бесконечный» цикл
```

```
    while(1);
```

```
}
```

```
////////////////////////////////////
```

/*

Подпрограмма обработки прерывания от 3-го таймера (при этом разрешена генерация запросов на прерывание только по событию «Обновление таймера», см. выше)

*/

```
void TIM3_IRQHandler(void)
```

```
{
```

/*

Сброс признака генерации прерывания по событию «Обновление таймера» обнулением 0-го бита (*UIF*) в регистре статуса таймера (см. пункт 14.4.5 руководства [15]). Необходимость сброса пояснена в подпункте 7.3.2.16.

Примечание. Под все прерывания от 3-го таймера выделена только одна позиция в таблице векторов прерываний (см. пункт 11.1.3 руководства [15]). Поэтому, в общем случае, в подпрограмму обслуживания прерываний от 3-го таймера должен быть включен опрос признаков всех событий, которые могли бы быть источниками прерывания (см. подпункт 7.3.2.15). Однако, в данном конкретном случае разрешены прерывания только по событию «Обновление таймера», поэтому необходимости в опросе признаков нет.

*/

TIM3->SR &= 0xFFFFFEE;

/*

Код управления светодиодом:

- если в предыдущем интервале времени длительностью 1 с он был выключен ($i = 0$) – включение светодиода установкой в единицу 4-го вывода ПБВ *A* и присвоение переменной *i* значения 1;
- в противном случае ($i = 1$) – выключение светодиода сбросом в единицу 4-го вывода ПБВ *A* и присвоение переменной *i* значения 0.

*/

switch (i)

{

case 0:

GPIOA->ODR |= 0x00000010;

i=1;

break;

case 1:

GPIOA->ODR &= 0xFFFFFEEF;

i=0;

break;

}

}

9.5.4.2. Пример 2. В данном примере производится ШИМ-управление яркостью свечения светодиода. С интервалами в 1 с среднее значение тока через светодиод увеличивается на $0,2I_{max}$, в пределах от $0,1I_{max}$ до $0,9I_{max}$. По окончании интервала времени длительностью 1 с, в течение которого среднее значение тока равно $0,9I_{max}$, оно уменьшается до минимального ($0,1I_{max}$), и описанный процесс повторяется. Модель МК - *STM32F030F4* модельного ряда *STM32F030xx* (см. пункт 9.5.3). Для формирования задержки выделен таймер *TIM3* категории «*General-purpose*», для формирования ШИМ-сигнала - таймер *TIM14* той же категории. Как и в предыдущем примере, МК в целом, в том числе таймеры *TIM3* и *TIM14* тактируются ГТИ с внешним ПЭР частотой 8 МГц; светодиод подключен к 4-му выводу ПБВ *A* (одной из альтернативных функций которого является выход 1-го канала таймера *TIM14* [26]). Включение светодиода осуществляется подачей единицы на данный вывод, выключение – подачей нуля.

Параметры конфигурации 3-го таймера, посредством которого формируются интервалы времени длительностью 1 с, очевидно, должны быть такими же, как и в предыдущем примере (см. подпункт 9.5.4.1). На выборе параметров конфигурации таймера *TIM14*, формирующего ШИМ-сигнал управления светодиодом, следует остановиться подробнее.

Таймером *TIM14* поддерживается только режим асимметричной ШИМ с работой счетчика в режиме суммирования (прямого счета) (см. раздел 16.2 руководства [15]). Основной недостаток асимметричной ШИМ – скачкообразное изменение начальной фазы ШИМ-сигнала при изменениях его длительности (см. рис. 9.9), не критичен при управлении яркостью светодиодов.

В первую очередь, необходимо выбрать **период** ШИМ-сигнала. Как указано в подпункте 9.3.2.2, его максимально возможное значение определяется допустимым уровнем пульсаций выходного сигнала ИУ после сглаживания, в рассматриваемом случае – воспринимаемых пользователем пульсаций светового потока, излучаемого светодиодом. При этом функцию сглаживающего ФНЧ выполняет зрительная система пользователя. Общепринято, что период ШИМ-сигнала, управляющего яркостью светодиода, должен быть менее 20 мс; в противном случае излучение светодиода не будет восприниматься как непрерывное.

Минимально допустимый период определяется условием $DC_{min}T, (1 - DC_{max})T \gg t_{\text{вкл/выкл иу}}$ (см. подпункт 9.3.2.3). В данном конкретном случае $t_{\text{вкл/выкл иу}}$ – время включения / выключения светодиода. На практике, с учетом задержек, вносимых токоограничивающей цепью светодиода и паразитными емкостями монтажа, оно составляет от десятков нс (для маломощных светодиодов) до единиц мкс (для мощных, осветительных) [39].

В рассматриваемом примере ток через светодиод изменяется в пределах в пределах от $0,1I_{max}$ до $0,9I_{max}$; следовательно, $DC_{min} = 1 - DC_{max} = 0,1$, и выбор периода ШИМ-сигнала сводится к удовлетворению условия $0,1T \gg t_{\text{вкл/выкл сд}}$. Для этого значение $0,1T$ должно находиться в пределах от сотен нс (для маломощных светодиодов) до десятков мкс (для мощных), а период ШИМ-сигнала – соответственно от единиц до сотен мкс. С точки зрения

возможности управления светодиодами различной мощности, а также минимизации энергопотребления, желательно выбрать значение периода, близкое к верхней границе данного диапазона, т. е. порядка нескольких сотен мкс. Выбираем его равным 500 мкс; при этом частота ШИМ-сигнала равна 2 кГц, что с запасом обеспечивает восприятие излучения светодиода как непрерывного светового потока.

Дискретность задания коэффициента заполнения ШИМ-сигнала равна, согласно исходным требованиям, 0,1. Следовательно, в режиме асимметричной ШИМ, период тактовых импульсов счетчика таймера достаточно выбрать в 10 раз меньшим T , а частоту (f_{CNT}) – в 10 раз большей частоты ШИМ-сигнала, т. е. равной 20 кГц. Таким образом, коэффициент деления предделителя счетчика должен быть равным 8 МГц / 20 кГц, т. е. 400, а содержимое регистра *PSC* таймера *TIM14* – 399 (см. выражение (9.39)). Верхний предел счета, и, соответственно, содержимое регистра *ARR* таймера *TIM14* при этом, согласно выражению (9.5), должны быть равны 9-и. В соответствии с выражением (9.6) для задания тока через светодиод в пределах от $0,1I_{max}$ до $0,9I_{max}$ с шагом $0,2I_{max}$, содержимое регистра *CCR1* таймера должно изменяться в пределах от 1 до 9 с шагом 2. Во избежание формирования импульсов с некорректной длительностью при изменении содержимого регистра *CCR1*, должна быть разрешена его буферизация, с перезагрузкой по обновлению таймера (по аналогии с рис. 9.7).

Код программного модуля с необходимыми комментариями представлен ниже.

```
//Подключение файла описания МК модельного ряда stm32f0xx
#include "stm32f0xx.h"
/*
Объявление целочисленной переменной i, управляющей состоянием
светодиода
*/
int i;
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Основной блок программы
void main(void)
{
```

/*

Конфигурирование источника синхронизации МК. Код полностью аналогичен приведенному в подпункте 4.4.2.4

*/

```
RCC->CR |= 0x00010000;
```

```
while((RCC->CR)&(0x00020000)==0);
```

```
RCC->CFGR &= 0xFFFFFFFFFC;
```

```
RCC->CFGR |= 0x00000001;
```

```
while(((RCC->CFGR)&(0x0000000C))!=0x00000004);
```

```
RCC->CR &= 0xFFFFFFFFE;
```

```
//Конфигурирование ПБВ А, к которому подключен светодиод
```

```
//См. комментарии к аналогичным строкам кода в подпункте 9.5.4.1
```

```
RCC->AHBENR |= 0x00020000;
```

```
GPIOA->MODER |= 0x00000100;
```

/*

Настройка 4-го вывода ПБВ А на выполнение 4-й альтернативной функции («1-й канал таймера *TIM14*») записью кода 0100 в битовое поле *AFSEL4* регистра *GPIOA_AFRL* ПБВ А (см. пункт 8.4.9 руководства [15] и табл. 12 *datasheet* [26])

*/

```
GPIOA->AFR[0] |= 0x00040000;
```

```
//Задание начального значения переменной i
```

```
i=0;
```

```
//См. комментарий к аналогичной строке кода в подпункте 9.5.4.1
```

```
NVIC->ISER[0] |= 0x00010000;
```

/*

Разрешение тактирования таймеров *TIM3* и *TIM14* установкой в единицу 1-го (*TIM3EN*) и 8-го (*TIM14EN*) регистра *APB1ENR* подсистемы управления синхронизацией и сбросом (см. пункт 7.4.8 руководства [15])

*/

```
RCC->APB1ENR |= 0x00000102;
```

/*

Конфигурирование таймера *TIM3* (см. комментарии к аналогичному фрагменту кода в подпункте 9.5.4.1)

*/

```
TIM3->PSC = 63999;
```

```
TIM3->ARR = 124;
```

```
TIM3->CR1 |= 0x00000080;
```

```
TIM3->DIER |= 0x00000001;
```

```
TIM3->CR1 |= 0x00000001;
```

/*

Конфигурирование 1-го канала таймера *TIM14* (см. пункты 16.4.5 и 16.4.6 руководства [15]).

Задание режима работы 1-го канала (ШИМ1, не инвертированный ШИМ, см. рис. 9.7) записью кода 110 в биты с 4-го по 6-й (битовое поле OC1M) регистра *TIM14_CCMR1*.

Разрешение буферизации регистра *CCR1* установкой в единицу 3-го бита (*OC1PE*) регистра *TIM14_CCMR1*.

Состояния 0-го и 1-го битов (*CCIS*) регистра *TIM14_CCMR1* оставлены нулевыми («по умолчанию»), что соответствует работе 1-го канала таймера *TIM14* в режиме генерации сигнала, с его выдачей на вывод *TIM14_CH1*, т. е. на 4-й вывод ПБВ А.

Разрешение работы канала *TIM14_CH1* записью единицы в 0-й бит (*CC1E*) регистра *TIM14_CCER*.

*/

```
TIM14->CCMR1 = 0x0068;
```

```
TIM14->CCER |= 0x0001;
```

/*

Запись в регистры *PSC* и *ARR* таймера *TIM14* ранее вычисленных значений коэффициента деления предделителя (уменьшенного на 1) и верхнего предела счета.

Примечание. В качестве входного сигнала предделителя счетчика может служить только синхросигнал домена *APB1*, к которому относится *TIM14* (см. пункт 16.2 руководства [15]). В данном примере частота этого синхросигнала равна 8 МГц.

*/

```
TIM14->PSC = 399;
```

```
TIM14->ARR = 9;
```

/*

Запись в регистр *TIM14_CCR1* начального значения (соответствующего току через светодиод, равному $0,1I_{max}$)

*/

```
TIM14->CCR1 = 1;
```

/*

Разрешение буферизации регистра *ARR* установкой в единицу 7-го бита (*ARPE*) регистра *CR1* таймера *TIM14* (см. пункт 16.4.1 руководства [15])

*/

```
TIM14->CR1 |= 0x0080;
```

/*

Запуск таймера *TIM14* установкой в единицу 0-го бита (*CEN*, *Counter Enable*) регистра *TIM14_CR1* (см. пункт 16.4.1 руководства [15])

*/

```
TIM14->CR1 |= 0x0001;
```

```
//«Бесконечный» цикл
```

```
while(1);
```

```
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
/*
```

Подпрограмма обработки прерывания от 3-го таймера (разрешена генерация запросов на прерывание только по событию «Обновление таймера», см. выше)

```
*/
```

```
void TIM3_IRQHandler(void)
```

```
{
```

```
//См. комментарий к аналогичной строке кода в подпункте 9.5.4.1
```

```
TIM3->SR &= 0xFFFFF0FE;
```

```
/*
```

Код управления током через светодиод. По истечении каждого из интервалов длительностью 1 с в буфер регистра *TIM14_CCR1* загружается число, задающее коэффициент заполнения ШИМ-сигнала в течение очередного секундного интервала. Указателем данного числа, в свою очередь, служит переменная *i*. После загрузки буфера регистра *TIM14_CCR1* значение указателя устанавливается соответствующим коэффициенту заполнения, который должен быть в следующем (после очередного) интервале длительностью 1 с.

```
*/
```

```
switch (i)
```

```
{
```

```
case 0:
```

```
TIM14->CCR1 = 1;
```

```
i=1;
```

```
break;
```

```
case 1:
```

```
TIM14->CCR1 = 3;
```

```
i=2;
```

```
break;
```

```
case 2:
```

```
TIM14->CCR1 = 5;
```

```
i=3;
```

```
break;
```

```
case 3:
```

```
TIM14->CCR1 = 7;
```

```
i=4;
```

```
break;
```

```

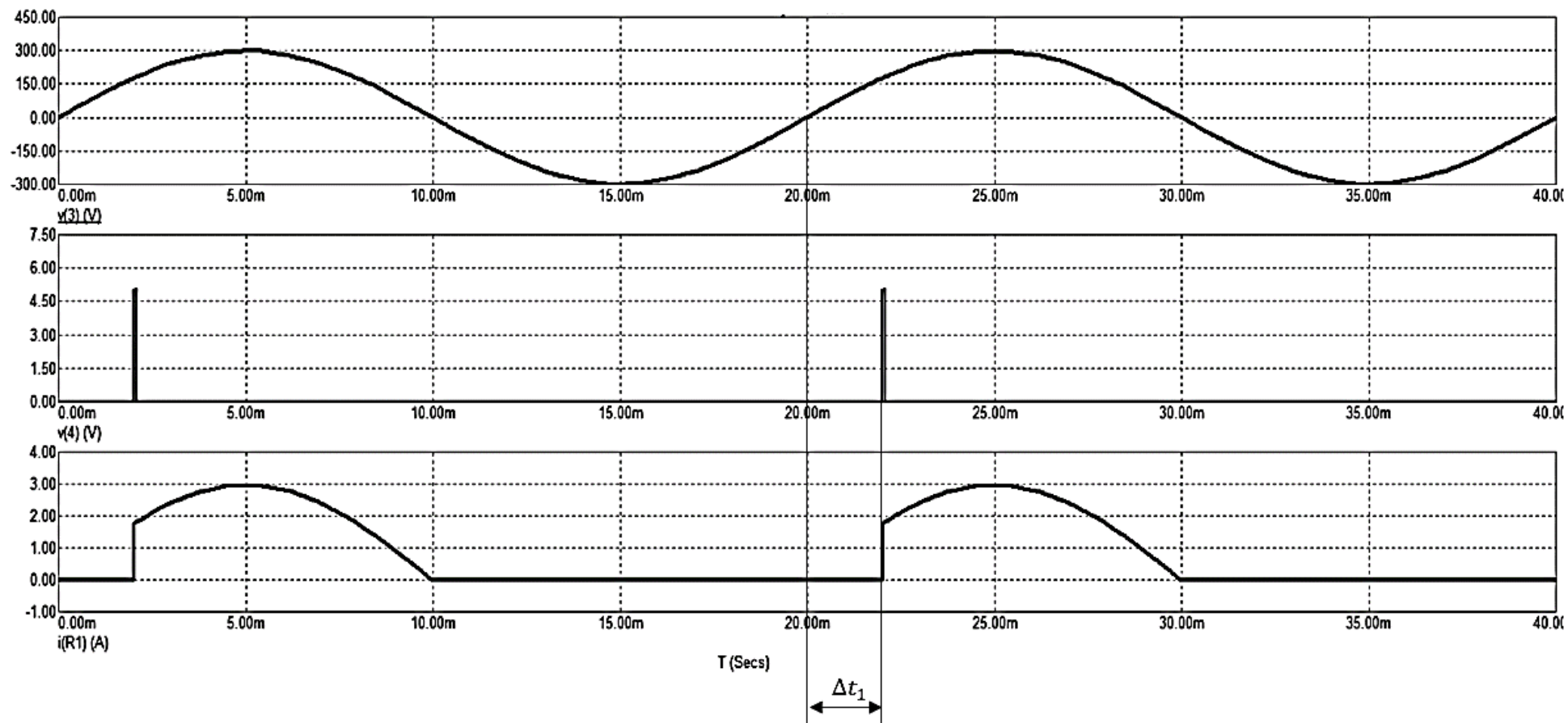
case 4:
TIM14->CCR1 = 9;
i=0;
break;
}
}

```

9.5.4.3. Пример 3. Работа таймера в режиме одновибратора (*One-pulse mode*, см. пункт 9.3.6).

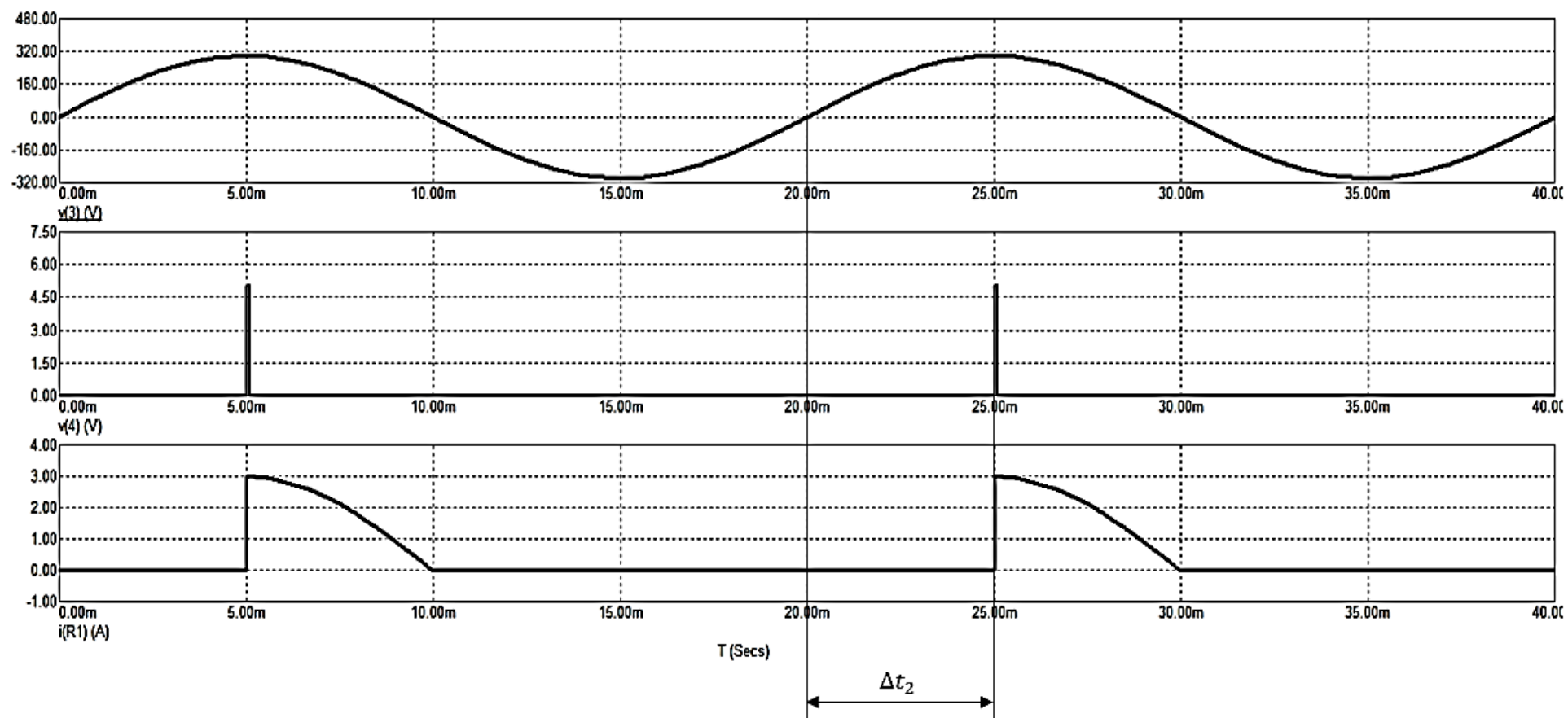
Пусть необходимо реализовать управление простейшим тиристорным регулятором мощности в некоторой нагрузке (импеданс которой может считаться активным). Регулирование осуществляется путем изменения задержки момента включения тиристора относительно момента пересечения нулевого уровня напряжением питания силовой цепи. Принцип регулирования поясняет рис. 9.50. На нем: $v(3)$ – напряжение питания силовой цепи; $v(4)$ – сигнал управления тиристором; $i(R1)$ – ток в нагрузке; Δt_1 и Δt_2 – задержка включения тиристора при различной мощности в нагрузке (рисунок 9.50а соответствует большей мощности, 9.50б – меньшей).

Модель МК - *STM32F030F4* модельного ряда *STM32F030xx* (см. пункт 9.5.3). Для формирования импульсов управления тиристором выделен 4-й канал таймера *TIM3* категории «*General-purpose*», импульсы поступают на управляющий электрод через цепь гальванической развязки. Длительность импульса включения тиристора должна быть равна $50 \text{ мкс} \pm 10\%$. Моменты пересечения нулевого уровня напряжением питания силовой цепи фиксируются внешним (по отношению к МК) аналоговым компаратором; данное напряжение подается на него через аттенюатор с гальваническим разделением. Выходной сигнал компаратора поступает на 0-й вход ПВВ А МК и служит источником запроса на прерывание по событию *EXTI0* (см. подпункт 7.3.2.17). Уровень логической единицы на выходе компаратора соответствует положительной полувольте напряжения питания силовой цепи, уровень логического нуля – отрицательной.



a)

Рис. 9.50, лист 1. Временные диаграммы, поясняющие принцип работы простейшего тиристорного регулятора мощности (см. текст)



б)

Рис. 9.50, лист 2

Период напряжения питания силовой цепи – 20 мс; допустимые пределы изменения интервала Δt (см. рис. 9.50) – от 1 до 9 мс с дискретностью 50 мкс.

МК в целом, в том числе таймер *TIM3* тактируются ГТИ с внешним ПЭР частотой 8 МГц.

Для решения поставленной задачи удобно использовать режим одновибратора (см. пункт 9.3.6, в т. ч. рис. 9.23). В качестве интервала времени Δt (см. рис. 9.50) при этом служит интервал t_{DELAY} (см. рис. 9.23), а длительность генерируемого импульса равна t_{PULSE} .

В данном примере в качестве старт-события при генерации импульса было выбрано программное разрешение работы счетчика (установка в активное состояние бита *CEN* в 1-м регистре управления таймером), выполняемое в подпрограмме обслуживания запроса на прерывание, генерируемого компаратором. Это позволяет реализовать полностью программно-контролируемое оперативное управление значениями t_{DELAY} и t_{PULSE} , задавая их перед формированием каждого импульса, до активизации счетчика, что исключает генерацию импульсов с «не предсказуемыми» t_{DELAY} и t_{PULSE} . Недостатком данного способа запуска генерации является задержка начала формирования интервала t_{DELAY} относительно момента срабатывания компаратора, вносимая процедурой вызова подпрограммы (15 циклов ЦП [9]), командами записи *CCR* и *ARR* и программного запуска счетчика, а также запаздыванием начала работы счетчика относительно момента разрешения счета (см. рис. 9.1). Однако, при тактовой частоте ЦП, равной 8 МГц, и частоте тактирования счетчика таймера не менее 1 МГц, данная задержка составляет порядка нескольких микросекунд, и пренебрежимо мала по сравнению с t_{DELAY} (от 1 до 9 мс) и дискретностью его задания (50 мкс).

Альтернативным вариантом запуска генерации импульса могло бы служить подключение выхода компаратора ко входу 1-го или 2-го канала таймера, с использованием фронта сигнала *TRGI* (см. рис. 9.41) в качестве старт-события; подробнее – см. примеры в пункте 14.3.10 и Приложении (*Appendix*) *A.8.16* руководства [15]. При таком подходе отсутствует задержка начала формирования

интервала t_{DELAY} относительно момента срабатывания компаратора. Однако, при этом существенно сложнее реализовать оперативное и корректное управление значениями t_{DELAY} и t_{PULSE} ввиду того, что старт-событие «не подконтрольно» программному обеспечению МК. Поэтому при длительностях интервала t_{DELAY} от нескольких сотен мкс и более данный вариант запуска менее предпочтителен.

Необходимо остановиться на выборе частоты тактирования счетчика таймера (f_{CNT}). Поскольку формирование интервала t_{DELAY} запускается установкой в единицу бита разрешения счета (CEN), максимальное отклонение реального значения t_{DELAY} от задаваемого выражением (9.20) равно сумме $1/f_{CNT}$ (см. рис. 9.1) и равной порядка нескольких микросекунд задержки, вносимой процедурой вызова подпрограммы, командами записи регистров CCR и ARR и программного запуска счетчика. Очевидно, максимальное отклонение должно быть пренебрежимо мало по сравнению с дискретностью задания t_{DELAY} (50 мкс). Данное условие гарантированно выполняется при $1/f_{CNT} = 1$ мкс, т. е. при $f_{CNT} = 1$ МГц. Коэффициент деления предделителя счетчика при этом должен быть равен 8-и, а содержимое регистра PSC – 7-и, в соответствии с выражением (9.39). Согласно выражениям (9.19) и (9.20):

- для изменения t_{DELAY} в пределах от 1 до 9 мс с дискретностью 50 мкс содержимое $CCR4$ должно изменяться в пределах от 1000 до 9000 с шагом 50;

- для формирования импульса включения тиристора длительностью 50 мкс содержимое регистра ARR должно быть равным $CCR4 + 49$.

Ниже представлены коды блоков программного модуля управления тиристором, относящихся к тематике настоящего раздела, с необходимыми комментариями.

```
/*
```

```
Включение библиотеки модулей ядра и памяти МК модельного ряда  
STM32F0xx
```

```
*/
```

```
#include "stm32f0xx.h"
```

```

/*
Объявление переменных, задающих значения  $t_{DELAY}$  и  $t_{PULSE}$ , как
беззнаковых целых
*/
unsigned int delay, pulse;
//Объявление других переменных
. . .
//Объявление подпрограммы конфигурирования 3-го таймера
void timer3_conf(void);

/*
Объявление подпрограммы конфигурирования контроллера внешних
прерываний
*/
void exti_conf(void);

/*
Объявление подпрограммы конфигурирования 0-го вывода порта A, на
который поступает выходной сигнал компаратора, и 1-го вывода порта B, на
который поступает выходной сигнал 4-го канала 3-го таймера
*/
void gpio_conf(void);

//Объявление других подпрограмм
. . .

/////////////////////////////////////////////////////////////////
// Подпрограмма конфигурирования 3-го таймера
void timer3_conf(void)
{
/*
Разрешение тактирования 3-го таймера установкой в единицу 1-го бита
(TIM3EN) в регистре RCC_APB1ENR (см. пункт 7.4.8 руководства [15]).
*/
RCC->APB1ENR |= 0x00000002;
/*
Задание режима работы счетчика 3-го таймера посредством регистра
TIM3_CR1 (см. пункт 14.4.1 руководства [15]).
Установка режима одновибратора (One Pulse Mode) записью единицы в 3-й
бит (OPM) регистра TIM3_CR1.
Буферизация регистра ARR не разрешена (бит ARPE регистра TIM3_CR1
оставлен в нулевом состоянии, «по умолчанию»), что позволяет программно
загружать регистр ARR (и, соответственно, задавать значения  $t_{DELAY}$  и  $t_{PULSE}$ ,

```

см. выражения (9.19) и (9.20)) перед формированием каждого импульса включения тиристора.

Состояния бита *DIR* и битового поля *CMS* регистра *TIM3_CR1* оставлены нулевыми («по умолчанию»), что соответствует режиму одностороннего прямого счета.

Разрешение работы счетчика (запись единицы в бит *CEN* регистра *TIM3_CR1*) производится в подпрограмме обслуживания прерывания по срабатыванию компаратора.

Примечание. В качестве входного сигнала предделителя счетчика выбран синхросигнал домена *APB1*, к которому относится *TIM3* (содержимое битового поля *SMS* регистра *TIM3_SMCR* оставлено нулевым, «по умолчанию»). В данном примере частота этого синхросигнала равна 8 МГц.

*/

```
TIM3->CR1 |= 0x00000008;
```

/*

Установка коэффициента деления предделителя равным 7, при этом частота счетных импульсов таймера равна $8 \text{ МГц} / (7 + 1) = 1 \text{ МГц}$

*/

```
TIM3->PSC = 7;
```

/*

Конфигурирование 4-го канала таймера *TIM3* посредством регистров *TIM3_CCMR2* и *TIM3_CCMR* (см. пункты 14.4.7 – 14.4.9 руководства [15] и рис. 9.43).

Состояния 8-го и 9-го битов (*CC4S*) регистра *TIM14_CCMR1* оставлены нулевыми («по умолчанию»), что соответствует работе 4-го канала таймера *TIM3* в режиме генерации сигнала, с его выдачей на вывод *TIM3_CH4*, т. е. на 1-й вывод ПВВ В.

Задание режима переключения *PWM2* по 4-му выходу записью кода 111 в биты 12 - 14 регистра *TIM3_CCMR2*: выход находится в пассивном состоянии, если содержимое счетчика меньше содержимого регистра *CCR4*, в противном случае - в активном состоянии (см. рис. 9.23).

13-й бит (*CC4P*) регистра *TIM3_CCER*, задающий полярность активного уровня на выходе, оставлен в нулевом состоянии («по умолчанию»), что соответствует единичному активному состоянию выхода.

Буферизация регистра *CCR4* **не разрешена** (бит *OC4PE* регистра *TIM3_CCMR2* оставлен в нулевом состоянии, «по умолчанию»), что позволяет программно загружать регистр *CCR4* (и, соответственно, задавать значение t_{DELAY} , см. выражение (9.20)) перед формированием каждого импульса включения тиристора.

Разрешение работы канала *TIM3_CH4* производится в подпрограмме обслуживания прерывания по срабатыванию компаратора.

*/

```
TIM3->CCMR2 |= 0x7000;
```

```

TIM3->CCER = 0x0000;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Подпрограмма конфигурирования контроллера внешних прерываний
void exti_conf(void)
{
/*
Разрешение тактирования контроллера системной конфигурации (SYSCFG)
установкой в единицу 0-го бита регистра RCC_APB2ENR (см. пункт 7.4.7
руководства [15]).
*/
RCC->APB2ENR |= 0x00000001;
/*
Глобальное разрешение прерываний с номером вектора 5 (т. е. прерываний по
событиям EXTI0 и EXTI1) установкой в единицу 5-го бита регистра ISER[0]
контроллера прерываний. Реально в данном примере задействованы только
запросы по событию EXTI0.
Примечание 1. Чтобы МК обслуживал запросы от этих источников,
необходимо также дополнительно установить биты разрешения обслуживания
данных запросов в регистре EXTI_IMR контроллера внешних прерываний (см.
далее)
*/
NVIC->ISER[0] |= 0x00000020;
/*
Выбор перепада из 0 в 1 (Rising Edge) в качестве источника запроса на
прерывание по событию EXTI0. Выбор производится установкой в единицу 0-
го бита регистра EXTI_RTSR (см. пункт 11.3.3 руководства [15], а также
подпункт 7.3.2.17).
*/
EXTI->RTSR |= 0x00000001;
/*
Разрешение обслуживания запросов на прерывания по событию EXTI0
установкой в единицу 0-го бита регистра EXTI_IMR (см. пункт 11.3.1
руководства [15]).
*/
EXTI->IMR |= 0x00000001;
/*

```

Примечание 2. При конфигурировании контроллера внешних прерываний, вообще говоря, должно быть указано, 0-й вывод **какого** из ПВВ служит источником события *EXTI0* (или, в общем случае, *i*-й вывод **какого** из ПВВ является источником события *EXTIi*), см. пункт 11.2.5 руководства [15], а также рис. 7.11. Указание ПВВ – источника соответствующего события производится посредством регистров *SYSCFG_EXTICR* (см. пункты 9.1.2 – 9.1.5 руководства [15]). По умолчанию, источниками событий *EXTIi* являются *i*-е выводы ПВВ *A*. В рассматриваемом примере источником события *EXTI0*

является 0-й вывод именно ПБВ *A*, поэтому программирование регистров *SYSCFG_EXTICR* не требуется.

*/

}

//

/*

Подпрограмма конфигурирования порта *A*, на 0-й вывод которого поступает выходной сигнал компаратора, и порта *B*, на 1-й вывод которого поступает выходной сигнал 4-го канала 3-го таймера

*/

void gpio_conf(void)

{

/*

Разрешение тактирования портов *A* и *B* установкой в единицу битов 17-го (*IOPAEN*) и 18-го (*IOPBEN*) регистра *RCC_AHBENR* (см. пункт 7.4.6 руководства [15]).

*/

RCC->AHBENR |= 0x00060000;

/*

Задание режима работы задействованного (1-го) вывода порта *B*: предписано выполнение альтернативной функции записью кода 10 в биты 2-й и 3-й регистра *GPIOB_MODER* (см. пункт 8.4.1 руководства [15]).

*/

GPIOB->MODER |= 0x00000008;

/*

Настройка 1-го вывода ПБВ *B* на выполнение 1-й альтернативной функции («4-й канал таймера *TIM3*») записью кода 0001 в битовое поле *AFSEL1* регистра *GPIOB_AFR1* ПБВ *B* (см. пункт 8.4.9 руководства [15] и табл. 13 *datasheet* [26]).

*/

GPIOB->AFR[0] |= 0x00000010;

/*

Примечание. Задействованный (0-й) вывод порта *A* работает в режиме «по умолчанию» (цифрового «плавающего» входа), поэтому специальные команды настройки его режима не требуются.

*/

}

//

//Основной блок программы

void main(void)

{

```
/*
```

Конфигурирование источника синхронизации МК. Код полностью аналогичен приведенному в подпункте 4.4.2.4

```
*/
```

```
RCC->CR |= 0x00010000;
while((RCC->CR)&(0x00020000)==0);
RCC->CFGR &= 0xFFFFFFFF;
RCC->CFGR |= 0x00000001;
while(((RCC->CFGR)&(0x0000000C))!=0x00000004);
RCC->CR &= 0xFFFFFFFF;
```

```
/*
```

Задание (в периодах счетных импульсов 3-го таймера, т. е. в микросекундах) значения t_{PULSE} и начального значения t_{DELAY} , соответствующего минимальной мощности в нагрузке

```
*/
```

```
pulse = 50;
delay = 9000;
```

```
/*
```

Задание начальных значений других переменных

```
*/
```

```
. . .
```

```
/*
```

Выполнение подпрограмм конфигурирования 3-го таймера, ПВВ и контроллера внешних прерываний, который конфигурируется в **последнюю** очередь, т. к. после его активизации начинается собственно процесс управления тиристором, поэтому все задействованные в нем блоки МК должны быть сконфигурированы до активизации контроллера внешних прерываний.

```
*/
```

```
timer3_conf();
gpio_conf();
exti_conf();
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
/*
```

«Бесконечный» цикл, в котором осуществляется мониторинг мощности в нагрузке и модифицирование переменной *delay* по его результатам, с целью поддержания мощности на заданном уровне.

```
*/
```

```
while (1)
```

```
{
```

```
. . .
```

```
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
}
```


По истечении интервала времени $t_{DELAY} + t_{PULSE}$ бит разрешения работы счетчика (*CEN*) автоматически сбрасывается, счетчик останавливается, генерируется событие «Обновление таймера», и он переходит в режим ожидания очередного запуска, который произойдет по очередному срабатыванию компаратора. Таким образом, производится формирование импульсов управления тиристором в соответствии с приведенными на рис. 9.50 временными диаграммами.

*/

```
TIM3->CR1 |= 0x00000001;
```

/*

Ожидание установки в единицу 0-го бита (*Update interrupt flag*) регистра статуса 3-го таймера (см. пункт 14.4.5 руководства [15]), что является признаком завершения генерации импульса.

*/

```
while(((TIM3->SR)&0x0001)==0);
```

//Сброс 0-го бита регистра статуса.

```
TIM3->SR &= 0xFFFE;
```

/*

Запрет выдачи сигналов на вывод *TIM3_CH4* записью нуля в 12-й бит (*CC4E*) регистра *TIM3_CCER* (см. пункт 14.4.9 руководства [15]).

*/

```
TIM3->CCER &= 0xEFFF;
```

/*

Выход из подпрограммы. Ожидание очередного запроса на прерывание по событию *EXTI0* (т. е. по срабатыванию компаратора).

*/

}

9.5.4.4. Пример 4. Преобразование в код длительности импульса.

В процессе контроля частотно-временных параметров некоторой интегральной микросхемы (ИМС) необходимо осуществлять преобразование в цифровой код длительности импульсов, генерируемых на одном из выходов ИМС. На нем формируется последовательность прямоугольных импульсов с «высоким» и «низким» уровнями, совместимыми с уровнями 0 и 1 ПВВ МК, и с номинальными значениями периода и длительности, равными $5 \text{ мкс} \pm 10\%$ и $1 \text{ мкс} \pm 10\%$ соответственно. Длительности фронта и спада импульсов не превышают 50 нс.

Относительная погрешность преобразования длительности импульса в код должна находиться в пределах $\pm 1\%$.

Установка контроля построена на МК модели *STM32F407VG* подсемейства *ARM Cortex-M4* [14, 29]. Тактирование МК в целом и его таймеров осуществляется блоком ФАПЧ, тактовая частота – 40

МГц. В качестве первичного ГТИ блока ФАПЧ служит генератор с внешним ПЭР частотой 8 МГц.

Генерируемая ИМС последовательность импульсов поступает на вход 2-го канала 3-го таймера МК.

Результат преобразования длительности импульса в код передается на хост-ПК через *USART* (см. раздел 11) для последующей обработки и протоколирования.

Преобразование длительности импульса в код осуществляется способом, иллюстрируемым рис. 9.51, и представляющим собой модификацию способа, описываемого приведенными на рис. 9.29 временными диаграммами, учитывающую особенности поставленной задачи.

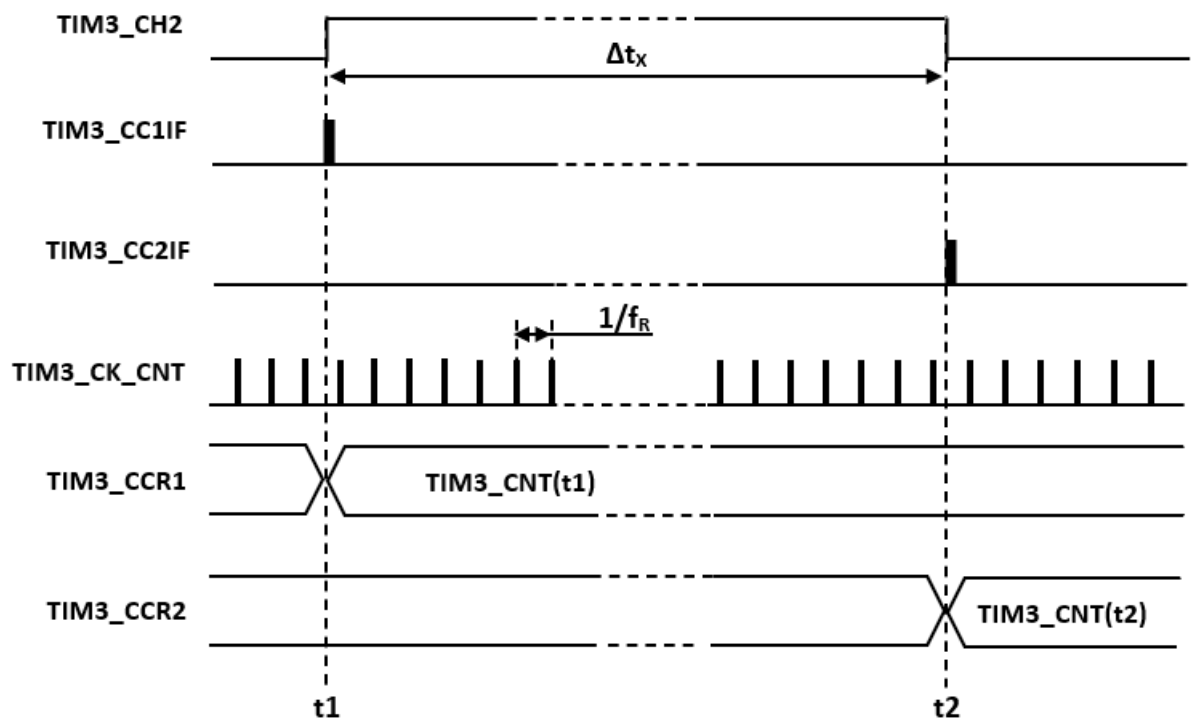


Рис. 9.51. Временные диаграммы, поясняющие используемый в данном примере способ преобразования длительности импульса в код (см. пояснения в тексте)

В реализуемом способе преобразования (см. рис. 9.51):

- используются два канала 3-го таймера (1-й и 2-й), сконфигурированные на функцию захвата содержимого счетчика соответственно по фронту и по спаду сигнала на входе одного и того же, 2-го канала (что допустимо, см. пункт 18.4.7 руководства [14]); благодаря этому нет необходимости в переназначении события,

инициирующего захват, как в п. 6 алгоритма, приведенного в подпункте 9.4.3.1;

- в качестве входного сигнала 2-го канала таймера служат непосредственно импульсы, длительность которых преобразуется в код; при этом отсутствует необходимость формирования старт- и стоп-импульса (см. рис. 9.29);

- признаками захвата по каналам служит установка в единичное состояние соответственно битов *CC1IF* и *CC2IF* в регистре статуса 3-го таймера;

- захват выявляется программным опросом его признаков; прерывания не используются из-за относительно небольшого значения Δt_x и, как следствие, повышенных требований к времени реакции на захват, которые сложнее удовлетворить при его обслуживании по прерываниям (в МК подсемейства *ARM Cortex-M4* время перехода к подпрограмме обслуживания прерывания составляет 12 тактов ЦП [9], в рассматриваемом примере оно было бы равно 300 нс по каждому из захватов);

- в качестве f_R служит тактовая частота МК, равная 40 МГц;

- значение Δt_x вычисляется по выражениям (9.30).

При указанных ранее значениях Δt_x ($1 \text{ мкс} \pm 10\%$) и частоты f_R (40 МГц) относительная погрешность преобразования Δt_x в код равна $\pm 2,5\%$, что не соответствует требуемому значению ($\pm 1\%$). Для снижения погрешности до $\pm 1\%$ используется усреднение результатов нескольких преобразований. Как указано в подпункте 9.4.3.6, поскольку погрешность преобразования носит случайный характер, при усреднении n результатов преобразований она уменьшается в $\sqrt{n}$ раз [53]. В рассматриваемом примере необходимо снижение погрешности в 2,5 раза; следовательно, необходимо усреднение минимум 7-и результатов преобразования. Реально производится усреднение 8-и результатов (с запасом).

Ниже представлены коды относящихся к тематике настоящего раздела блоков программного модуля преобразования длительности импульсов в код, с необходимыми комментариями.

/*

Включение библиотеки модулей ядра и памяти МК модельного ряда *STM32F4xx*

*/

```
#include "stm32f4xx.h"
```

```
/*
```

Объявление, как целых, переменных:

- *i*, служащей счетчиком цикла;
- *ct1* и *ct2*, равные содержимому счетчика, захватываемому по фронту и по спаду импульса (см. рис. 9.51);
- *n*, равной числу усредняемых результатов преобразования;
- *dtx*, равной текущему результату преобразования длительности импульса в код;
- *summ*, равной сумме результатов однократных преобразований;
- *avrg*, равной среднему значению *n* результатов преобразования.

```
*/
```

```
int i, ct1, ct2, n, dtx, summ, avrg;
```

```
/*
```

Объявление результата преобразования, передаваемого на хост-ПК, как беззнаковой 8-битовой переменной *dath*

```
*/
```

```
unsigned char dath;
```

```
//Объявление других переменных
```

```
• • •
```

```
//Объявление подпрограмм, используемых в программном модуле
```

```
• • •
```

```
////////////////////////////////////  
////////////////////////////////////
```

```
//Основной блок программы
```

```
int main(void)
```

```
{
```

```
//Задание числа усредняемых результатов преобразования.
```

```
n = 8;
```

```
////////////////////////////////////  
/*
```

Конфигурирование источника синхронизации МК. Код полностью аналогичен приведенному в подпункте 4.4.2.5.

```
*/
```

```
RCC->CR |=0x00010000;
```

```
while(((RCC->CR)&(0x00020000))==0x00000000);
```

```
RCC->PLLCFGR = 0x22411404;
```

```
RCC->CR |=0x01000000;
```

```
while(((RCC->CR)&(0x02000000))==0x00000000);
```

```
RCC->CFGR = 0x00000002;
```

```
while(((RCC-> CFGR)&(0x0000000C))!=0x00000008);
```

```
RCC->CR = (RCC->CR)&(0xFFFFF0FE);
```

```

//////////////////////////////////////////////////
/*
Настройка 7-го вывода ПБВ А на выполнение альтернативной функции входа
2-го канала 3-го таймера
*/
/*
Разрешение тактирования ПБВ А установкой в единицу 0-го бита (GPIOAEN)
регистра RCC_AHB1ENR (см. пункт 7.3.10 руководства [14]).
*/
RCC->AHB1ENR |= 0x00000001;
/*
Настройка 7-го вывода ПБВ А на выполнение альтернативной функции
записью двоичного кода 10 в битовое поле MODER7 (14-й и 15-й биты)
регистра GPIOA_MODER (см. пункт 8.4.1 руководства [14]).
*/
GPIOA->MODER |= 0x00008000;
/*
Указание номера альтернативной функции, выполняемой 7-м выводом ПБВ А.
В данном примере ее номер – 2 (TIM3_CH2, см. табл. 9 datasheet [29]). Номер
альтернативной функции указывается записью двоичного кода 0010 в битовое
поле AFRL7 (биты с 28-го по 31-й) регистра GPIOA_AFRL (см. пункт 8.4.9
руководства [14]).
*/
GPIOA->AFR[0] = 0x20000000;
//////////////////////////////////////////////////
/*
Конфигурирование 3-го таймера, посредством которого выполняется
преобразование длительности импульса в код.
*/
/*
Разрешение тактирования 3-го таймера установкой в единицу 1-го бита
(TIM3EN) регистра RCC_APB1ENR (см. пункт 7.3.13 руководства [14]).
*/
RCC->APB1ENR |= 0x00000002;
/*
Конфигурирование 1-го и 2-го каналов 3-го на функцию захвата,
инициируемого сигналом, поступающим на вывод TIM3_CH2.
Конфигурирование производится записью двоичных кодов 10 и 01 в битовые
поля соответственно CC1S и CC2S регистра TIM3_CCMR1 (см. пункт 18.4.7
руководства [14]).
*/
TIM3->CCMR1=0x0102;

```

/*

Разрешение работы 1-го и 2-го каналов 3-го таймера установкой в единицу битов 0-го (CC1E) и 4-го (CC2E) регистра *TIM3_CCER*.

Выбор фронта импульса и его спада в качестве событий, инициирующего захват соответственно по 1-му и 2-му каналам. Выбор производится записью нуля в 1-й бит (CC1P) и в 5-й бит (CC2P) регистра *TIM3_CCER*.

См. пункт 18.4.9 руководства [14].

*/

TIM3->CCER=0x0031;

/*

Примечание. Все остальные параметры конфигурации 3-го таймера оставлены «по умолчанию».

*/

/*

Запуск счетчика 3-го таймера записью единицы в 0-й бит регистра *TIM3_CR1* (см. пункт 18.4.1 руководства [14]).

*/

TIM3->CR1=0x0001;

//

/*

Конфигурирование других функциональных блоков МК, используемых при решении данной задачи, в т. ч. USART

*/

• • •

//

//«Бесконечный» цикл

while(1)

{

//

//Преобразование длительности импульса в код

/*

Присваивание начального (нулевого) значения сумме результатов однократных преобразований

*/

summ=0;

/*

Сброс 1-го (*CC1IF*) и 2-го (*CC2IF*) битов регистра статуса 3-го таймера (см. пункт 18.4.1 руководства [14]), являющихся признаками захвата / сравнения по 1-му и 2-му каналу.

*/

TIM3->SR &= 0xFFF9;

```

/*
n-кратное (8-кратное) преобразование длительности импульса в код
*/
for (i=1;i<=n;i++)
{
/*
Ожидание установки в единицу 1-го бита регистра статуса, служащего
признаком захвата по 1-му каналу (в данном примере инициируемого фронтом
импульса)
*/
while(((TIM3->SR)&0x0002)==0);
/*
Ожидание установки в единицу 2-го бита регистра статуса, служащего
признаком захвата по 2-му каналу (в данном примере инициируемого спадом
импульса)
*/
while(((TIM3->SR)&0x0004)==0);
//Сброс признаков захвата
TIM3->SR &= 0xFFFF;
/*
Считывание значений TIM3_CNT(t1) и TIM3_CNT(t2) из регистров CCR1 и
CCR2 соответственно.
*/
ct1 = TIM3->CCR1;
ct2 = TIM3->CCR2;
/*
Расчет длительности импульса (в периодах тактовой частоты счетчика) по
выражениям (9.30).
*/
if(ct2>ct1) dtx = ct2 - ct1;
else dtx = 65536 - ct1 + ct2;
/*
Прибавление очередного результата преобразования к текущему значению
суммы результатов однократных преобразований
*/
summ = summ + dtx;
}
/*
Вычисление среднего значения результатов n преобразований и его
присвоение результату преобразования, передаваемому на хост-ПК.
*/
avrg=summ/n;

```

```

dath=avrg;
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Передача результата преобразования на хост-ПК
.      .      .
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
}

```

9.6 Выводы по разделу 9

9.6.1. Таймеры общего назначения входят в состав практически всех моделей МК всех классов. Их назначение – реализация **частотно-временных функций**, в том числе:

- формирование программно задаваемых задержек (интервалов времени) между какими-либо событиями, например, между включениями / выключениями ИУ;
- генерация сигналов управления ИУ с программно задаваемыми частотно-временными параметрами (периодом или / и длительностью) данных сигналов;
- преобразование в цифровой код частоты, длительности или периода выходных сигналов датчиков или каких-либо других сигналов, отражающих состояние объекта контроля и управления.

В составе ряда моделей простейших МК класса «*cost-sensitive*» имеется только один таймер общего назначения (см., например, рис. 1.1 и 1.2), как правило, позволяющий реализовывать только первую из перечисленных частотно-временных функций. В структуре МК класса «*mainstream*», как правило, имеется 3 – 4 таймера общего назначения (см. рис. 1.3), а в составе МК класса «*high performance*» - от 3-х – 4-х (см. рис. 1.4 и 1.5) до 10-и – 15-и [9, 14]. Таймеры общего назначения практически всех моделей МК классов «*mainstream*» и «*high performance*», в совокупности, позволяют реализовывать все три перечисленные группы частотно-временных функций.

9.6.2. Центральным функциональным узлом всех таймеров является двоичный счетчик; поэтому в технической документации ряда семейств МК используется термин «таймер / счетчик» (*Timer / Counter*), см., например, [8]. Кроме него, в структуру таймеров

общего назначения МК классов «*mainstream*» и «*high performance*», как правило, входят (см. пункты 9.5.1 и 9.5.3):

- программно-управляемый предварительный делитель частоты источника тактирования счетчика;
- у ряда семейств МК (например, *ARM Cortex-Mx*) – программно-доступный регистр, задающий верхний предел счета
- программно-доступные регистры сравнения, используемые при формировании сигналов с программно-задаваемыми частотно-временными параметрами;
- программно-доступные регистр / регистры захвата (в ряде семейств, например, *ARM Cortex-Mx* – схемотехнически совмещенные с регистрами сравнения), используемые при преобразовании в код частотно-временных параметров сигналов;
- программно конфигурируемые блоки предварительной обработки (в т. ч. фильтрации) входных сигналов таймера и блоки формирования выходных сигналов;
- программно-доступные регистры конфигурирования таймера и контроля его состояния.

9.6.3. Типовая структура и архитектура таймеров общего назначения современных МК классов «*mainstream*» и «*high performance*» (см. пункты 9.5.1 и 9.5.3) обеспечивают следующие функциональные возможности:

- работы счетчика таймера в режиме как прямого, так и обратного счета (у ряда семейств, в частности, *ARM Cortex-Mx* – с возможностью программного выбора направления счета, в то время как, например, в таймерах МК семейства *AVR* режим обратного счета используется только при реализации центрированной ШИМ);
- предварительной загрузки в счетчик некоторого числа под управлением ПО;
- программного выбора источника тактирования счетчика; в его качестве может служить как ГТИ МК, так и некоторый внешний по отношению к МК источник;
- выбора частоты счетных импульсов посредством программно-управляемого предделителя (*Prescaler*) частоты выходного сигнала выбранного источника тактирования;

- у ряда семейств МК, например, *ARM Cortex-Mx* – программного выбора верхнего предела счета;
- формирования импульсных сигналов, с программно-задаваемыми, посредством регистров сравнения, частотой, длительностью, активным и пассивным уровнями, а также (например, у таймеров категории «*Advanced-control*» МК семейства *ARM Cortex-Mx*) формирования противофазных импульсных сигналов с «мертвым временем» между переключениями;
- фиксации, по определенным, программно задаваемым событиям, содержимого счетчика в регистре / регистрах захвата; данная функциональная возможность используется при преобразовании частоты и длительности интервалов времени в код;
- формирования запросов на прерывания (у ряда семейств – также на ПДП) по характерным событиям – переполнению счетчика, совпадению его содержимого с содержимым регистра сравнения, фиксации его содержимого в регистре захвата.

Вышеперечисленные возможности, в целом, позволяют реализовывать все функции, перечисленные в пункте 9.6.1.

9.6.4. Формирование, посредством таймеров, программно задаваемых интервалов времени между событиями сводится к подсчету импульсов с известной частотой до заданного значения, по достижении которого выполняются определенные действия, например, включение или выключение некоторого ИУ. Обобщенный алгоритм формирования программно задаваемой задержки описан в подразделе 9.2. См. также примеры, приведенные в подпунктах 9.5.2.1 и 9.5.4.1.

9.6.5. Кроме формирования интервалов времени, одной из базовых областей применения таймеров общего назначения является генерация сигналов с программно-задаваемыми частотно-временными параметрами. Архитектура таймеров общего назначения практически всех семейств современных МК классов «*mainstream*» и «*high performance*» предоставляет возможность формирования, по крайней мере, 2-х классов сигналов с программно-задаваемыми частотно-временными параметрами:

- **ШИМ-сигналов**, с фиксированной, однократно задаваемой при конфигурировании таймера частотой, и переменной длительностью, программно-управляемой в процессе работы МК;

- **ЧИМ-сигналов**, с фиксированной скважностью и переменной частотой, программно-управляемой в процессе работы МК.

Архитектура таймеров ряда семейств МК (в частности, *ARM Cortex-Mx*) позволяет также:

- формировать **ФИМ-сигналы**, с фиксированной частотой и скважностью и программно-управляемой начальной фазой;

- реализовывать режим **одновибратора**, т. е. генерировать, по некоторому старт-событию, одиночный импульс с программно-управляемыми длительностью и задержкой относительно момента наступления старт-события.

9.6.6. Общие принципы формирования сигналов с программно-задаваемыми частотно-временными параметрами изложены в пункте 9.3.1. Обязательными функциональными узлами таймера МК, реализующего формирование данных сигналов, являются программно-доступный регистр сравнения и цифровой компаратор (см. рис. 9.2). В регистр сравнения записывается число, задающее какой-либо из параметров генерируемого сигнала (например, длительность импульса ШИМ-сигнала или частоту ЧИМ-сигнала). В процессе работы счетчика его содержимое непрерывно сравнивается с содержимым регистра сравнения посредством цифрового компаратора. При достижении их равенства происходит переключение формируемого сигнала из некоторого состояния в некоторое другое состояние. Направление переключения зависит от класса формируемого сигнала и режима генерации. Возврат формируемого сигнала в состояние, из которого он переключился по срабатывании компаратора, происходит, в зависимости от режима генерации сигнала, по достижении счетчиком нижнего или верхнего предела счета или по следующему срабатыванию цифрового компаратора (см. пункты 9.3.3 – 9.3.5).

9.6.7. Из перечисленных в пункте 9.6.4 классов сигналов, формируемых таймерами МК, наиболее широкое применение на практике находят ШИМ-сигналы. Основными областями их использования являются:

- ШИМ-регулирование мощности ИУ под управлением ПО МК (см. пункт 9.3.2, а также подпункт 9.5.4.2);
- цифро-аналоговое преобразование низкочастотных сигналов (см. рис. 9.6 и подпункт 9.5.2.2), в т. ч. формирование программно-управляемых постоянных напряжений и токов.

Основы реализации ШИМ посредством таймеров МК изложены в пункте 9.3.3, ряд вопросов ее практического применения (в т. ч. выбора параметров сглаживающего ФНЧ и калибровки источников ШИМ-сигналов) – в пункте 9.3.2 и в подпункте 9.5.2.2. Пример программной реализации ШИМ-ЦАП представлен в подпункте 9.5.2.2, ШИМ-регулирования тока в нагрузке – в подпункте 9.5.4.2.

9.6.8. Основы реализации ЧИМ и ФИМ на базе таймеров МК описаны в пунктах 9.3.4 и 9.3.5 соответственно; работы таймера в режиме одновибратора – в пункте 9.3.6. Пример программной реализации режима одновибратора при тиристорном управлении током в нагрузке представлен в подпункте 9.5.4.3.

9.6.9. Преобразование частоты сигнала в цифровой код реализуется подсчетом числа периодов данного сигнала за фиксированный интервал времени (формируемый, в свою очередь, способом подсчета импульсов с известной частотой). Рациональным сочетанием точности и аппаратурных затрат характеризуется описанный в подпункте 9.4.2.6 способ преобразования частоты в код, использующий функцию таймера, называемую «захват». Данный способ использует два таймера, один из которых формирует импульс с длительностью интервала счета (например, способом ШИМ), а счетчик второго осуществляет непрерывный счет импульсов с подлежащей преобразованию частотой. Его содержимое фиксируется в регистре захвата таймера в моменты фронта и спада импульса, задающего интервал счета. Значение частоты вычисляется по выражениям (9.27).

9.6.10. Преобразование в код длительности интервала времени (периода или длительности сигнала) осуществляется подсчетом импульсов с фиксированной частотой за временной интервал, длительность которого подлежит преобразованию. Реализация данного преобразования посредством таймеров МК также базируется на функции захвата (см. пункт 9.4.3). Счетчик таймера

при этом осуществляет непрерывный счет импульсов с опорной частотой, его содержимое фиксируется в регистре захвата по началу и по окончании интервала времени, длительность которого подлежит преобразованию. Ее значение рассчитывается по выражениям (9.30).

Пример программной реализации преобразования длительности импульсов в код приведен в подпункте 9.5.4.4.

9.6.11. В целом, рассмотренные в настоящем разделе структурно-архитектурные решения таймеров общего назначения являются типовыми для всех семейств МК классов «*mainstream*» и «*high performance*».