

Таймеры-счётчики

Таймеры и процессоры событий

Большинство задач управления, которые реализуются с помощью МК, требуют исполнения их **в реальном времени**.

Под реальным временем понимается **способность системы получить информацию о состоянии управляемого объекта, выполнить необходимые расчетные процедуры (реализовать алгоритм управления) и выдать управляющие воздействия в течение интервала времени, достаточного для желаемого изменения состояния объекта.**

Возлагать функции формирования сигналов управления в реальном масштабе времени **только на центральный процессор неэффективно**, так как это занимает ресурсы, необходимые для расчетных процедур.

Поэтому в большинстве современных МК используется **аппаратная поддержка работы в реальном времени** с использованием **таймера** (таймеров).

Схемотехника МК обычно предусматривает возможность **использования таймера в режиме счетчика внешних событий**, поэтому его часто называют таймером/счетчиком.

Модули **таймеров/счётчиков** служат для приема информации о времени наступления тех или иных событий (от внутренних генераторов или от внешних датчиков событий), и также для формирования управляющих воздействий во времени.

Модуль таймера/счётчика 8-разрядного МК представляет собой **8-и или 16-и разрядный счетчик со схемой управления.**

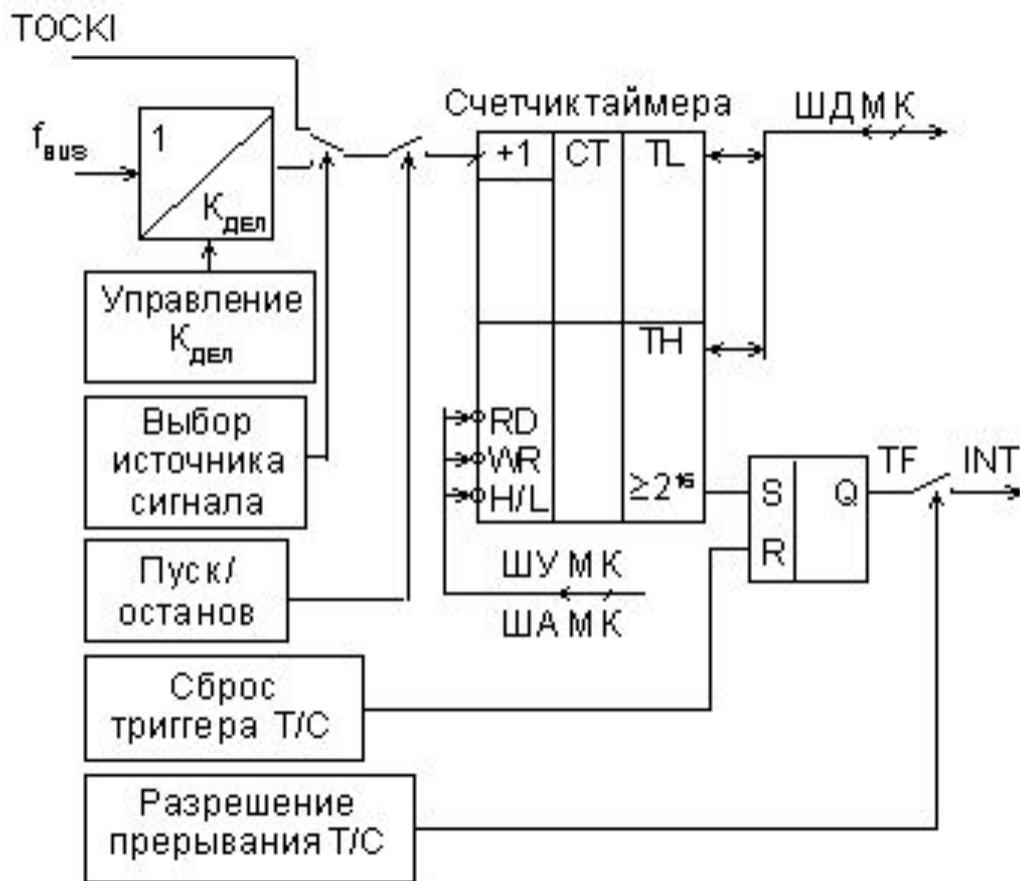


Рисунок 1 - Структура модуля таймера/счетчика.

В памяти МК 16-разрядный счетчик отображается двумя регистрами:

TH — старший байт счетчика,
TL — младший байт.

Регистры доступны для чтения и для записи.

Направление счета — только прямое, то есть содержимое счетчика **инкрементируется**.

Два источника входных сигналов:

- импульсная последовательность с выхода управляемого делителя частоты f_{BUS} (режим таймера);**
- сигналы внешних событий, поступающие на вход $T_{ОСКИ}$ контроллера (режим счётчика).**

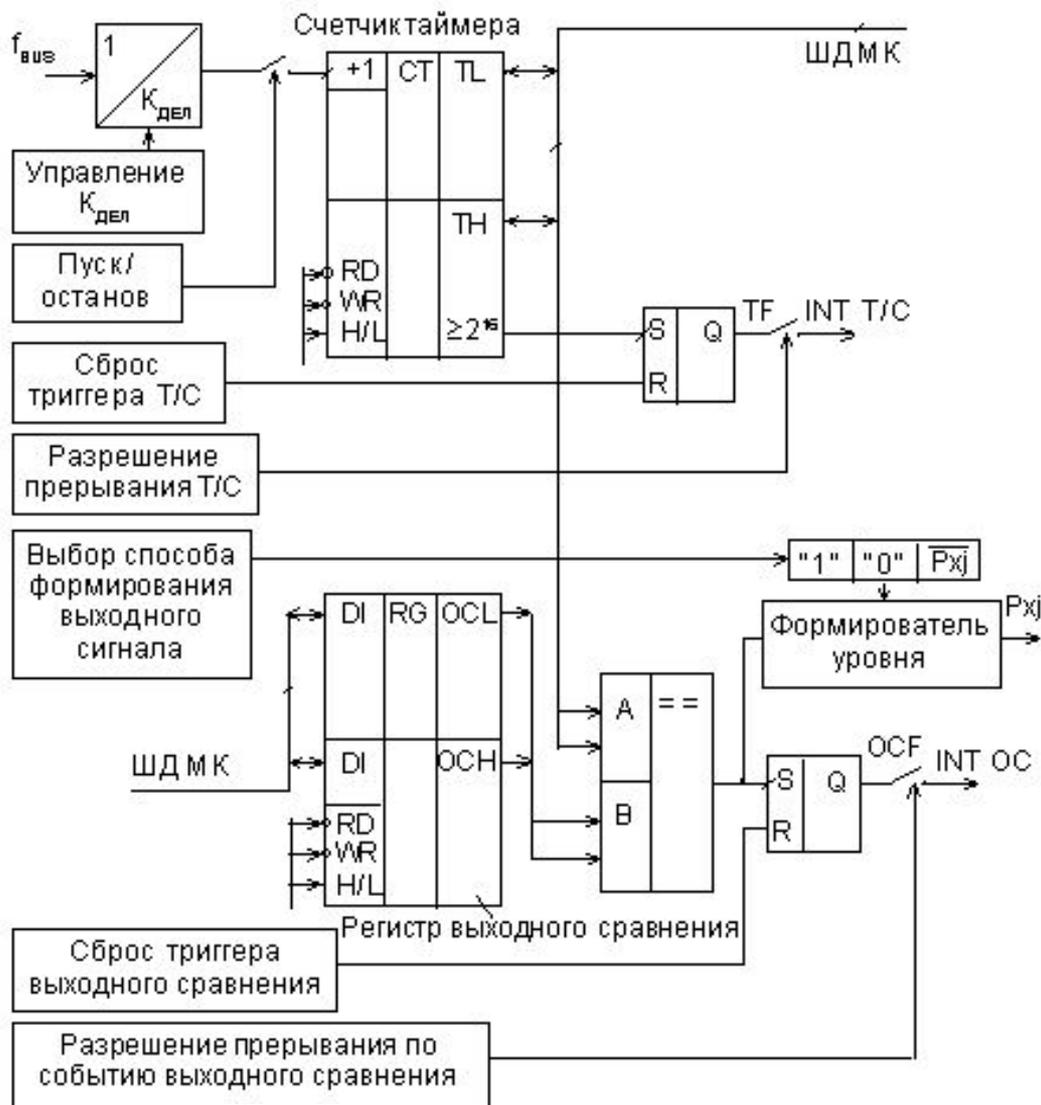


Рисунок 2 - Структурная схема канала выходного сравнения таймера

Цифровой компаратор

непрерывно сравнивает текущий код счетчика таймера с кодом, который записан в 16-разрядном регистре выходного сравнения. В момент равенства кодов на одном из выходов МК (P_{xj}) устанавливается заданный уровень логического сигнала. Предусмотрено три типа изменения сигнала на выходе P_{xj} в момент события выходного сравнения:

- установка **высокого** логического уровня;
- установка **низкого** логического уровня;
- **инвертирование** сигнала на выходе.

Микроконтроллеры семейства Atmel AVR Mega в зависимости от модели имеют в своем составе от двух до шести таймеров/счетчиков общего назначения. У ATmega128 – 4 таймера/счётчика.

Во всех моделях микроконтроллеров семейства AVR присутствуют как минимум два таймера/счетчика — T0 и T1.

Таймер/счетчик T0 – 8 разрядный. Имеет минимальный набор функций, зависящий от модели микроконтроллера.

Базовые функции: – **отсчёт и измерение временных интервалов**
– **счетчик внешних событий.**

Дополнительные функции:

- **генерация сигналов с широтно-импульсной модуляцией (ШИМ)** фиксированной разрядности (один или два канала),
- **работа в асинхронном режиме** в качестве **часов реального времени** (в моделях ATmega64x/128x).

Таймер/счетчик T1 – 16 разрядный. Может использоваться для **отсчета временных интервалов** и как **счетчик внешних событий**.

Кроме того, он может осуществлять запоминание своего состояния по **внешнему сигналу (захват)**. Как и таймер/счетчик T0, он может работать в качестве **2- или 3-канального ШИМ**, но уже **переменной разрядности**. Количество каналов ШИМ зависит от модели.

Остальные таймеры/счётчики: T2 идентичен T0; T3 и т.д. в той или иной мере повторяют функции T1.

В составе всех микроконтроллеров семейства имеется также **сторожевой таймер**, являющийся неизменным атрибутом всех современных микроконтроллеров.

Сторожевой таймер позволяет избежать случайного закливания программы, возникающего по тем или иным причинам.

Назначение выводов таймеров/счетчиков

Каждый таймер/счетчик использует один или несколько выводов микроконтроллера — линии портов ввода/вывода общего назначения, а функции, реализуемые этими выводами при работе совместно с таймерами/счетчиками, являются их **альтернативными функциями**.

Важно! При использовании альтернативных функций линий портов ввода/вывода **необходимо**, как правило, **самостоятельно сконфигурировать** эти **выводы в соответствии с их функциональным назначением** (для ввода или вывода).

Прерывания от таймеров/счетчиков

Для разрешения/запрещения прерываний от таймеров/счетчиков используется от одного до двух регистров ввода/вывода (**TIMSK** и ExpandedTIMSK – **ETIMSK**). Точно так же дело обстоит и с регистрами, содержащими флаги прерываний. Названия и адреса всех этих регистров

ATmega64x/128x	T0, T1, T2	TIMSK	\$37 (\$57)	TIFR	\$36 (\$56)
	T1, T3	ETIMSK	(\$7D)	ETIFR	(\$7C)

Другие – новые модели содержат большее число таких регистров (3...6), что равно числу счетчиков в конкретной модели.

Для разрешения какого-либо прерывания от таймера/счетчика необходимо установить в 1 соответствующий бит регистра **TIMSK/ETIMSK** и, разумеется, **флаг I** регистра **SREG** (разрешение обслуживания прерываний).

Форматы регистров **TIMSK** и **ETIMSK**, используемые для разрешения/запрещения прерываний от таймеров/счетчиков.

7	6	5	4	3	2	1	0
OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
R/W	R/W	R/W	R/W	P/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

ATmega8535x
ATmega16x/32x
ATmega64x/128x

Формат регистра TIMSK

7	6	5	4	3	2	1	0
—	—	TICIE3	OCIE3A	OCIE3B	TOIE3	OCIE3C	OCIE1C
R	R	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

ATmega64x
ATmega128x

Формат регистра ETIMSK

Биты регистров TIMSK, ETIMSK и TIMSK0...TIMSK5

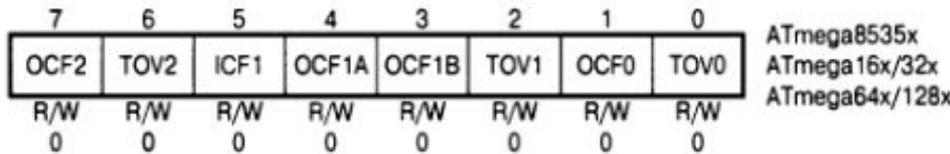
Название бита	Описание
TOIE n	Флаг разрешения прерывания по переполнению таймера/счетчика T_n ($n = 0...5$)
OCIE n	Флаг разрешения прерывания по событию «Совпадение» таймера/счетчика T_n ($n = 0, 2$)
OCIE nA	Флаг разрешения прерывания по событию «Совпадение А» таймера/счетчика T_n ($n = 0...5$)
OCIE nB	Флаг разрешения прерывания по событию «Совпадение В» таймера/счетчика T_n ($n = 0...5$)
OCIE nC	Флаг разрешения прерывания по событию «Совпадение С» таймера/счетчика T_n ($n = 1, 3...5$)
TICIE n	Флаг разрешения прерывания по событию «Захват» таймера/счетчика T_n ($n = 1, 3$)
ICIE n	Флаг разрешения прерывания по событию «Захват» таймера/счетчика T_n ($n = 1, 3...5$)

При наступлении какого-либо события соответствующий флаг регистра **TIFR/ETIFR** устанавливается в 1.

При запуске подпрограммы обработки прерывания он аппаратно сбрасывается в 0. Любой флаг может быть также сброшен программно, записью в него лог. 1.

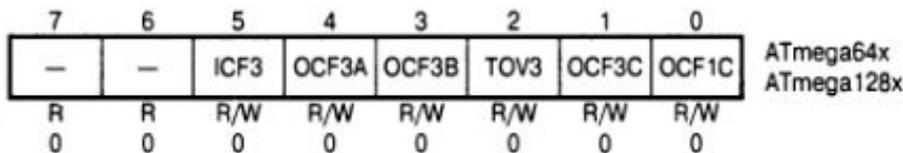
Биты регистра TIFR

Название бита	Описание
TOV n	Флаг прерывания по переполнению таймера /счетчика T n ($n = 0...5$)
OCF n	Флаг прерывания по событию «Совпадение» таймера/счетчика T n ($n = 0, 2$)
OCF nA	Флаг прерывания по событию «Совпадение А» таймера/счетчика T n ($n = 0...5$)
OCF nB	Флаг прерывания по событию «Совпадение В» таймера/счетчика T n ($n = 0...5$)
OCF nC	Флаг прерывания по событию «Совпадение С» таймера/счетчика T n ($n = 1, 3...5$)
ICF n	Флаг прерывания по событию «Захват» таймера/счетчика T n ($n = 1, 3...5$)



Формат регистра TIFR

Чтение (R)/Запись (W)
Начальное значение



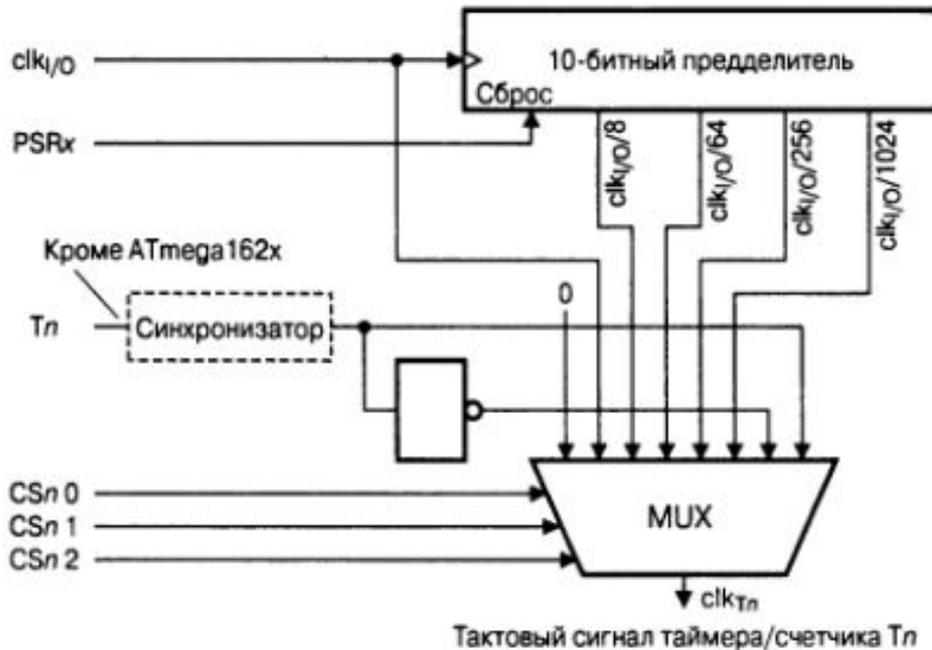
Формат регистра ETIFR

Чтение (R)/Запись (W)
Начальное значение

Предделители таймеров/счетчиков

Блоки предделителей предназначены для формирования тактовых сигналов таймеров/счетчиков clk_{T0} , clk_{T1} , clk_{T2} , clk_{T3} .

Упрощенная структурная схема блока предделителя для таймеров/счетчиков, не имеющих асинхронного режима работы



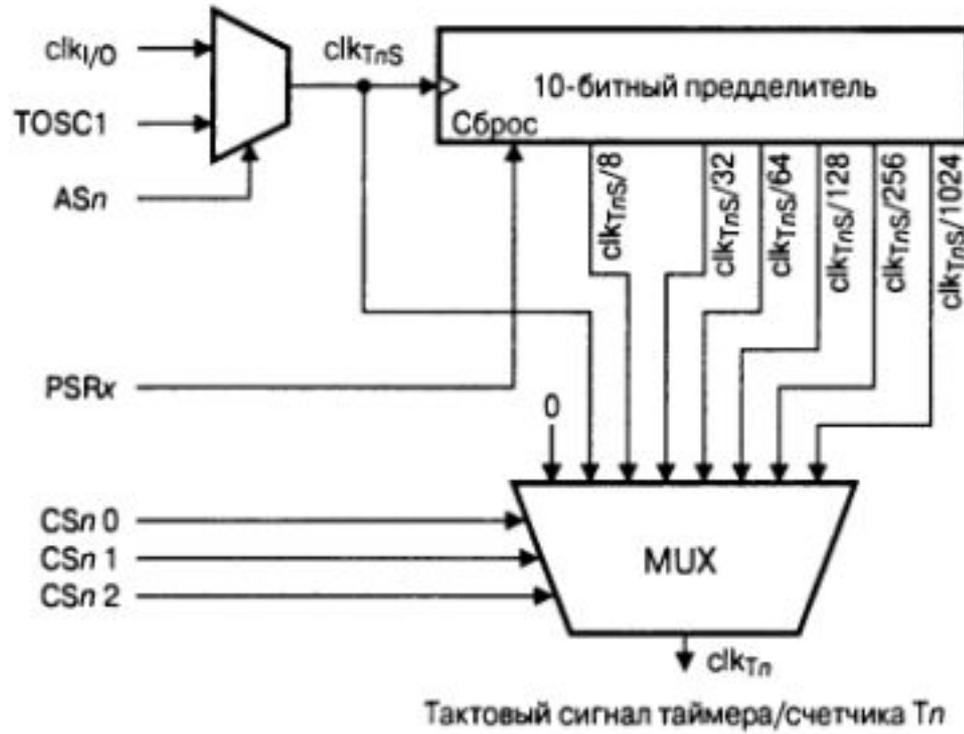
Тактовый сигнал таймера/счетчика T_n

- $n = 0, 1, 3$ — для ATmega162x ($PSRx = PSR310$),
- $1, 2, 3$ — для ATmega64x и ATmega128x ($PSRx = PSR321$),
- $0, 1$ — для ATmega8515x/8535x ATmega8x/16x/32x
и ATmega165x/325x/3250x/645x/6450x ($PSRx = PSR10$),
- $0, 1$ — для ATmega48x/88x/168x и ATmega164x/324x/644x ($PSRx = PSRSYNC$),
- $0, 1, 3, 4, 5$ — для ATmega640x/1280x/1281x/2560x/2561x ($PSRx = PSRSYNC$)

В состав каждого блока входят:
а) 10-битный предделитель,
б) выходной мультиплексор (селектор тактового сигнала),
в) входной мультиплексор исходного тактового сигнала (для таймеров, имеющих возможность работы в асинхронном режиме)

Так, например, выполнен предделитель таймера/счетчика T_0 моделей ATmega64x/128x и таймера/счетчика T_2 остальных моделей.

Структурная схема блока предделителя для таймеров/счетчиков, имеющих возможность работы в асинхронном режиме.

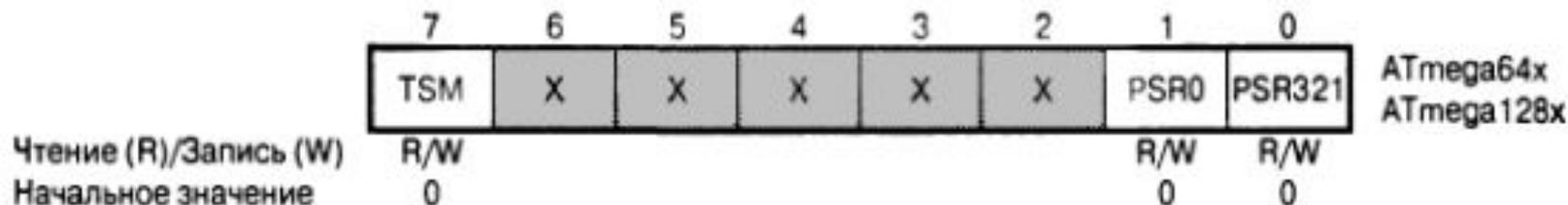


- $n = 1$ — для ATmega64x и ATmega128x ($PSR_x = PSR0$);
- 2 — для ATmega8535x, ATmega8x/16x/32x, ATmega162x и ATmega165x/325x/3250x/645x/6450x ($PSR_x = PSR2$);
- 2 — для ATmega48x/88x/168x, ATmega164x/324x/644x и ATmega640x/1280x/1281x/2560x/2561x ($PSR_x = PSRASY$).

Все таймеры/счетчики каждой модели семейства микроконтроллеров, не имеющие асинхронного режима работы, используют один и тот же 10-битный предделитель. При этом управление тактовым сигналом каждого таймера/счетчика осуществляется индивидуально.

Управление предделителями

Помимо управления тактовым сигналом таймера/счетчика, все микроконтроллеры семейства позволяют осуществлять **сброс предделителей**, а отдельные модели позволяют также осуществлять их **остановку**. Для этого используется регистр специальных функций **SFIOR**, расположенный по адресу \$23.



Формат регистра управления предделителями таймеров/счетчиков **SFIOR**. (биты, не используемые для управления предделителями таймеров/счетчиков, указаны на рисунке как X).

Для сброса предделителей таймеров/счетчиков используются биты **PSRx** регистра. При записи в эти биты лог. 1 предделители соответствующих таймеров/счетчиков переводятся в исходное состояние.

Биты сбрасываются в 0 аппаратно после выполнения операции сброса.

Важно! Один предделитель, как правило, используется несколькими таймерами/счетчиками, и соответственно **сброс предделителя повлияет на все таймеры/счетчики, которые его используют.**

Остановка всех предделителей микроконтроллера осуществляется **записью лог 1** в бит TSM регистра SFIOR.

Запуск предделителей осуществляется **записью лог. 0** в бит TSM.

Указанная функция может использоваться, в частности, **для синхронизации таймеров/счетчиков:**

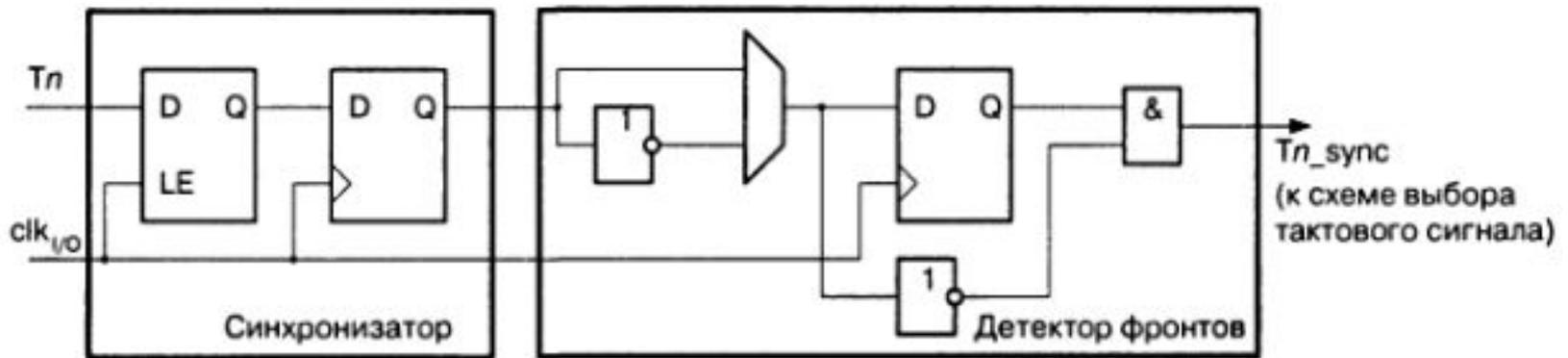
1. После установки бита TSM и битов PSRx соответствующие **таймеры/счетчики останавливаются** и могут быть проинициализированы требуемыми значениями.
2. После сброса бита TSM биты PSRx аппаратно сбрасываются и все **таймеры/счетчики начинают работать одновременно.**

Использование внешнего тактового сигнала

Практически все таймеры/счетчики, не имеющие асинхронного режима работы, могут тактироваться от внешнего сигнала.

Внешний сигнал, поступает на вход T_n ($n = 0...5$) микроконтроллера.

Прежде чем поступить на вход селектора тактового сигнала внешний сигнал, проходит через специальный узел, включающий схему синхронизации и детектор фронтов. В общем виде схема этого узла приведена на рисунке.



Входной каскад выводов T_n

Синхронизация внешнего сигнала осуществляется с частотой тактового сигнала микроконтроллера (состояние вывода T1 считывается по нарастающему фронту тактового сигнала clk_{T0}).

Поэтому **частота внешнего сигнала должна быть в 2 раза ниже частоты тактового сигнала микроконтроллера** ($f_{EXT} < f_{clk_{I/O}}/2$). Однако, чтобы гарантировать обнаружение фронтов внешнего сигнала во всем диапазоне возможных изменений частоты и скважности тактового сигнала микроконтроллера (из-за разброса параметров элементов тактового генератора), рекомендуется, чтобы частота внешнего сигнала была меньше, чем $f_{clk_{I/O}/2,5}$.

Следует помнить, что из-за влияния входного каскада происходит временная задержка между изменением состояния вывода и обновлением счетного регистра таймера/счетчика. **Величина задержки составляет от 2.5 до 3.5 тактов.**

Управление тактовым сигналом 8 разрядных Т/С

Формирование тактового сигнала таймера/счетчика $clk_{T0(T2)}$ осуществляется блоком предделителя.

В качестве тактового сигнала $clk_{T0(T2)}$ таймеров/счетчиков, не имеющих асинхронного режима, может использоваться:

- системный тактовый сигнал ($clk_{T0(T2)} = clk_{I/O}$).
- масштабированный системный тактовый сигнал ($clk_{T0(T2)} = clk_{I/O/n}$);
- внешний сигнал, поступающий на вход **Tn** микроконтроллера ($clk_{T0(T2)} = clk_{EXT}$)

Тактовый сигнал таймеров/счетчиков с асинхронным режимом может формироваться:

- либо из системного тактового сигнала $clk_{I/O}$ ($clk_{T0(T2)} = clk_{I/O/n}$),
- либо – в асинхронном режиме – из сигнала от дополнительного кварцевого резонатора ($clk_{T0(T2)} = clk_{TOSC1/n}$)

Переключение между синхронным и асинхронным режимами работы осуществляется с помощью бита **AS0 (AS2)** регистра **ASSR**.

Выбор источника тактового сигнала, а также запуск и остановка таймеров/счетчиков осуществляются с помощью битов **CS02...CS00** для **T0** (**CS22...CS20** для **T2**) регистров управления таймером **TCCRn (TCCRnA/TCCRnB)**.

7	6	5	4	3	2	1	0	ATmega8515x/8335x
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	ATmega16x/32x
Чтение (R)/Запись (W)	W	R/W	R/W	R/W	R/W	R/W	R/W	ATmega162x
Начальное значение	0	0	0	0	0	0	0	ATmega64x/128x

Формат
регистров
TCCR0
TCCR2

Выбор источника тактового сигнала таймеров/счетчиков T0
и T2

CSn2	CSn1	CSn0	Источник тактового сигнала		
			Обычный таймер/счетчик	Асинхронный таймер/счетчик	
				ASn = 0	ASn = 1
0	0	0	Таймер/счетчик остановлен	Таймер/счетчик остановлен	
0	0	1	clk _{I/O}	clk _{I/O}	clk _{TOSC1}
0	1	0	clk _{I/O} /8	clk _{I/O} /8	clk _{TOSC1} /8
0	1	1	clk _{I/O} /64	clk _{I/O} /32	clk _{TOSC1} /32
1	0	0	clk _{I/O} /256	clk _{I/O} /64	clk _{TOSC1} /64
1	0	1	clk _{I/O} /1024	clk _{I/O} /128	clk _{TOSC1} /128
1	1	0	Вывод Tn, счет осуществляется по спадающему фронту импульсов	clk _{I/O} /256	clk _{TOSC1} /256
1	1	1	Вывод Tn, счет осуществляется по нарастающему фронту импульсов	clk _{I/O} /1024	clk _{TOSC1} /1024

Например для работы в синхронном режиме запись TCCR0 = 0b00000101 установит работу таймера/счётчика 0 в режим предделения тактовой частоты на 1024, т.е. на одно изменение состояния внутреннего счетчика будет приходиться 1024 изменений тактового сигнала.

Режимы работы

Режим работы таймера/счетчика T0 (T2) определяется состоянием битов **WGMn2:WGMn0** регистра **TCCR0** (TCCRnA/TCCRnB).

Зависимость режима работы таймеров/счетчиков от состояния этих битов.

Режимы работы таймеров/счетчиков T0 и T2

Номер режима	WGMn2 ¹⁾	WGMn1	WGMn0	Режим работы таймера/счетчика T _n	Модуль счета (TOP)	Обновление регистров OCR _{nх}	Момент установки флага TOV _n
0	0	0	0	Normal	\$FF	Немедленно	\$FF
1	0	0	1	Phase correct PWM	\$FF	При TOP	\$00
2	0	1	0	CTC (сброс при совпадении)	OCR _n (OCRnA)	Немедленно	\$FF
3	0	1	1	Fast PWM	\$FF	При TOP	\$FF
4 ¹⁾	1	0	0	Зарезервировано	—	—	—
5 ¹⁾	1	0	1	Phase correct PWM	OCR _n (OCRnA)	При TOP	\$00
6 ¹⁾	1	1	0	Зарезервировано	—	—	—
7 ¹⁾	1	1	1	Fast PWM	OCR _n (OCRnA)	При TOP	TOP

¹⁾ В моделях ATmega48х/88х/168х, ATmega164х/324х/644х и ATmega640х/1280х/1281х/2560х/2561х.

Примечание. n = 0 или 2.

Режим Normal

Режим Normal - наиболее простой режим работы таймеров/счетчиков.

Для некоторых таймеров/счетчиков, например, T0 модели ATmega8x это вообще единственный режим.

В режиме Normal **счетный регистр функционирует как обычный суммирующий счетчик.**

По каждому импульсу тактового сигнала clk_{Tn} осуществляется инкрементирование счетного регистра TCNTn.

При переходе содержимого TCNTn через значение \$FF возникает переполнение, и счет продолжается со значения \$00. В том же такте сигнала clk_{Tn} , в котором обнуляется регистр TCNTn, флаг прерывания по переполнению TOVn устанавливается в 1.

При равенстве счетного регистра TCNTn и регистра сравнения OCRn устанавливается соответствующий флаг прерывания OCFn (OCFnA/OCFnB) и, если бит OCIEн (OCIEнA/OCIEнB) регистра маски TIMSKn установлен в 1, генерируется прерывание.

Наряду с установкой флага при равенстве счетного регистра и регистра сравнения **может изменяться состояние вывода OC_n (OC_nA/OC_nB)** микроконтроллера.

Способ (вид) изменения, определяется битами **$COM_{n1}:COM_{n0}$** регистра управления **$TCCR_n/TCCR_{nA}$**

Управление выводами OC_n

COM_{n1} (COM_{nx1})	COM_{n0} (COM_{nx0})	Описание
0	0	Таймер/счетчик T_n отключен от вывода OC_n (OC_{nx})
0	1	Состояние вывода меняется на противоположное
1	0	Вывод сбрасывается в 0
1	1	Вывод устанавливается в 1

Примечание. $n = 0$ или 2

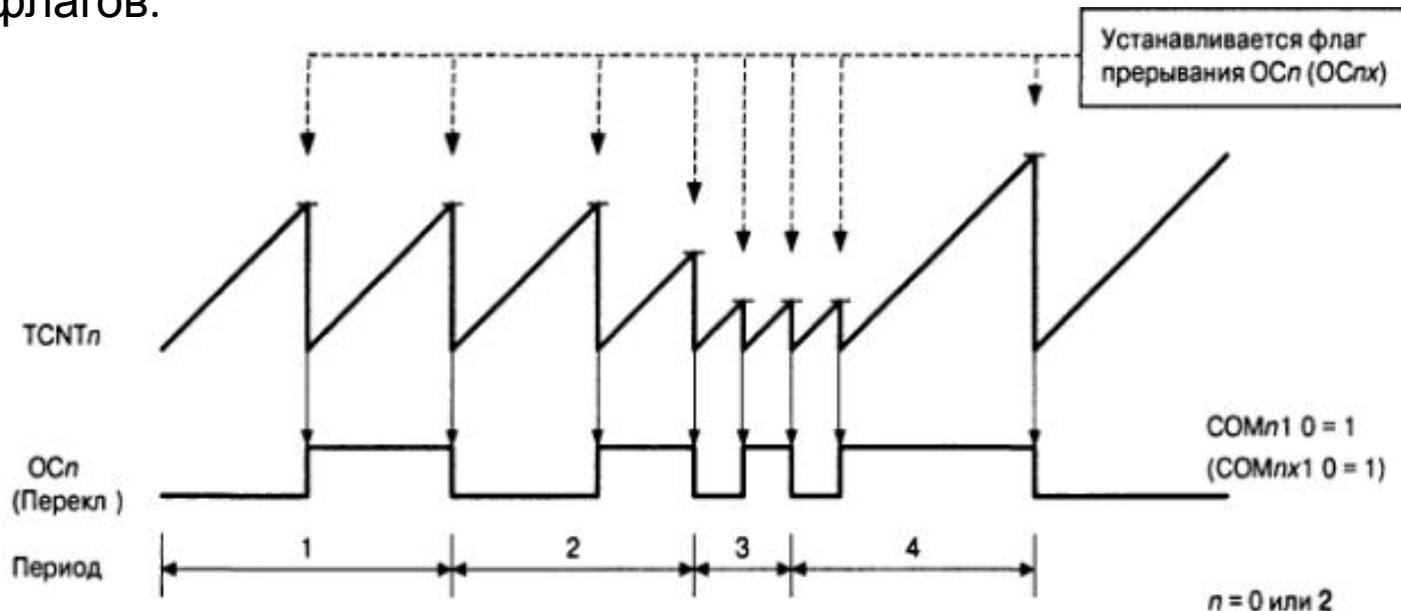
При необходимости состояние вывода **OC_n (OC_nA/OC_nB)** может быть изменено принудительно записью лог. 1 в бит **FOC_n (FOC_nA/FOC_nB)** соответствующего регистра управления. Прерывание при этом не генерируется.

Режим СТС (сброс при совпадении)

В режиме СТС счетный регистр TCNTn тоже функционирует как обычный суммирующий счетчик, инкрементирование которого осуществляется по каждому импульсу тактового сигнала clk_{Tn} .

Однако максимально возможное значение счетного регистра и, следовательно, разрешающая способность счетчика определяются регистром сравнения OCRn (OCRnA).

После достижения значения, записанного в регистре сравнения, счет продолжается со значения \$00. Если в регистре сравнения записано \$FF (OCR0 = 0xFF), то в том же такте сигнала clk_{Tn} , в котором обнуляется счетный регистр, устанавливается флаг прерывания по переполнению TOVn в соответствующем регистре флагов.



Временная диаграмма работы таймера/счетчика в режиме СТС

При достижении счетчиком максимального значения устанавливается флаг OCF_n (OCF_{nA}), и, если бит OClEn (OClEnA) соответствующего регистра маски **установлен в 1, генерируется прерывание**. Одновременно с установкой флага может изменяться состояние выводов OC_n (OC_{nA}/OC_{nB}) микроконтроллера. Состояние выводов определяется битами **COM_{n1}:COM_{n0}** (COM_{nx1}:COM_{nx0}) регистра управления TCCR_n/TCCR_{nA}

COM _{n1} (COM _{nx1})	COM _{n0} (COM _{nx0})	Описание
0	0	Таймер/счетчик T _n отключен от вывода OC _n (OC _{nx})
0	1	Состояние вывода меняется на противоположное
1	0	Вывод сбрасывается в 0
1	1	Вывод устанавливается в 1

Примечание. *n* = 0 или 2.

Для генерации сигнала заданной частоты необходимо записать в биты **COM_{n1}:COM_{n0}** (COM_{nx1}:COM_{nx0}) значение **01** (переключение вывода). Частота генерируемого сигнала будет определяться выражением

$$f_{OCn} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCR_{nx})},$$

где *N* — коэффициент деления предделителя. При необходимости состояние вывода OC_n (OC_{nA}/OC_{nB}) можно изменить принудительно записью лог. 1 в бит FOC_n (FOC_{nA}/FOC_{nB}) соответствующего регистра управления TCCR_n. Прерывание при этом не генерируется и сброса счетного регистра не производится.