

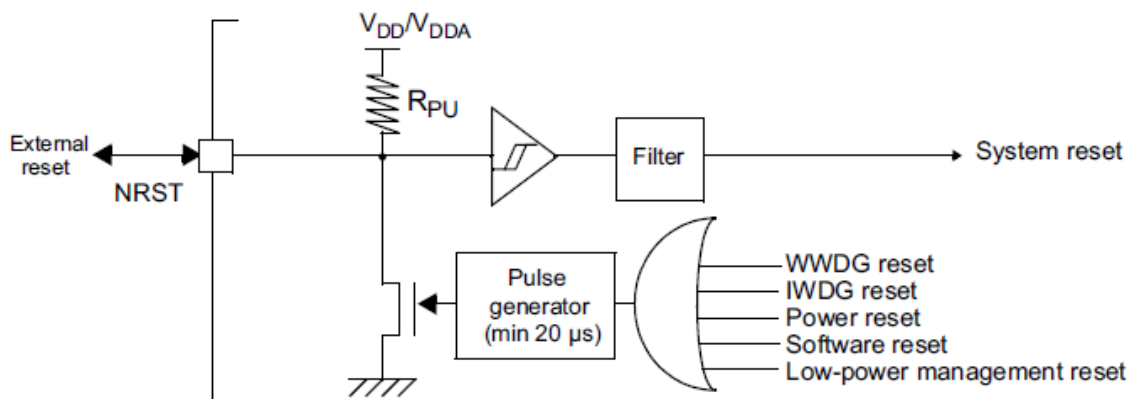
5 ТИПОВЫЕ СТРУКТУРНО-АРХИТЕКТУРНЫЕ РЕШЕНИЯ ПОДСИСТЕМЫ СБРОСА МК ОБЩЕГО НАЗНАЧЕНИЯ

5.1 Общие вопросы.....	2
5.2 Сброс МК по включении питания и при сбоях по питанию	6
5.3 Внешний сброс МК	8
5.4 Сброс от сторожевых таймеров.....	11
5.4.1. Назначение и принципы применения сторожевых таймеров...	11
5.4.2. Общие принципы работы сторожевых таймеров	12
5.4.3. Сторожевой таймер МК семейства <i>AVR</i>	13
5.4.4. Сторожевые таймеры МК семейства <i>ARM Cortex-Mx</i>	18
5.5 Сброс МК, инициируемый ПО	26
5.6 Сброс МК по инициализации входа в энергосберегающие режимы и по выходу из них	26
5.7 Сброс МК через отладочный интерфейс	26
5.8 Формирование внутреннего сигнала сброса МК.....	27
5.9 Индивидуальный сброс функциональных блоков МК.....	28
5.10 Выводы по разделу 5.....	29

5.1 Общие вопросы

Подсистема сброса, наряду с подсистемами питания и синхронизации и с портами ввода / вывода, входит в состав функциональных блоков / подсистем МК, минимально необходимых для его практического применения. Ее основное назначение – установка всех энергозависимых ЗУ МК (в 1-ю очередь – РОН и РСФ) в некоторые определенные и предсказуемые состояния при включении питания, при штатной перезагрузке МК или при возникновении нештатных ситуаций, например, сбоях по питанию, невыполнении условий выхода из цикла и т. п. (см. табл. 1.1). Состояния, в которые устанавливаются регистры при сбросе, называются *Reset value*, *Initial value* или *Default value*, и всегда приводятся в технической документации на МК.

Структурно-архитектурные решения подсистем сброса большинства семейств МК, в целом, аналогичны. Рассмотрим их на 2-х типовых примерах: подсистемы сброса МК семейства *ARM Cortex-Mx* и МК подсемейства *ATmega*, упрощенные структурные схемы которых представлены на рис. 5.1 и 5.2 соответственно.



R_{PU} – подтягивающий резистор

IWDG, *WWDG* – соответственно независимый и оконный сторожевые таймеры

Рис. 5.1. Упрощенная структурная схема подсистемы сброса МК подсемейства *ARM Cortex-M3* [13] (пояснения см. в тексте)

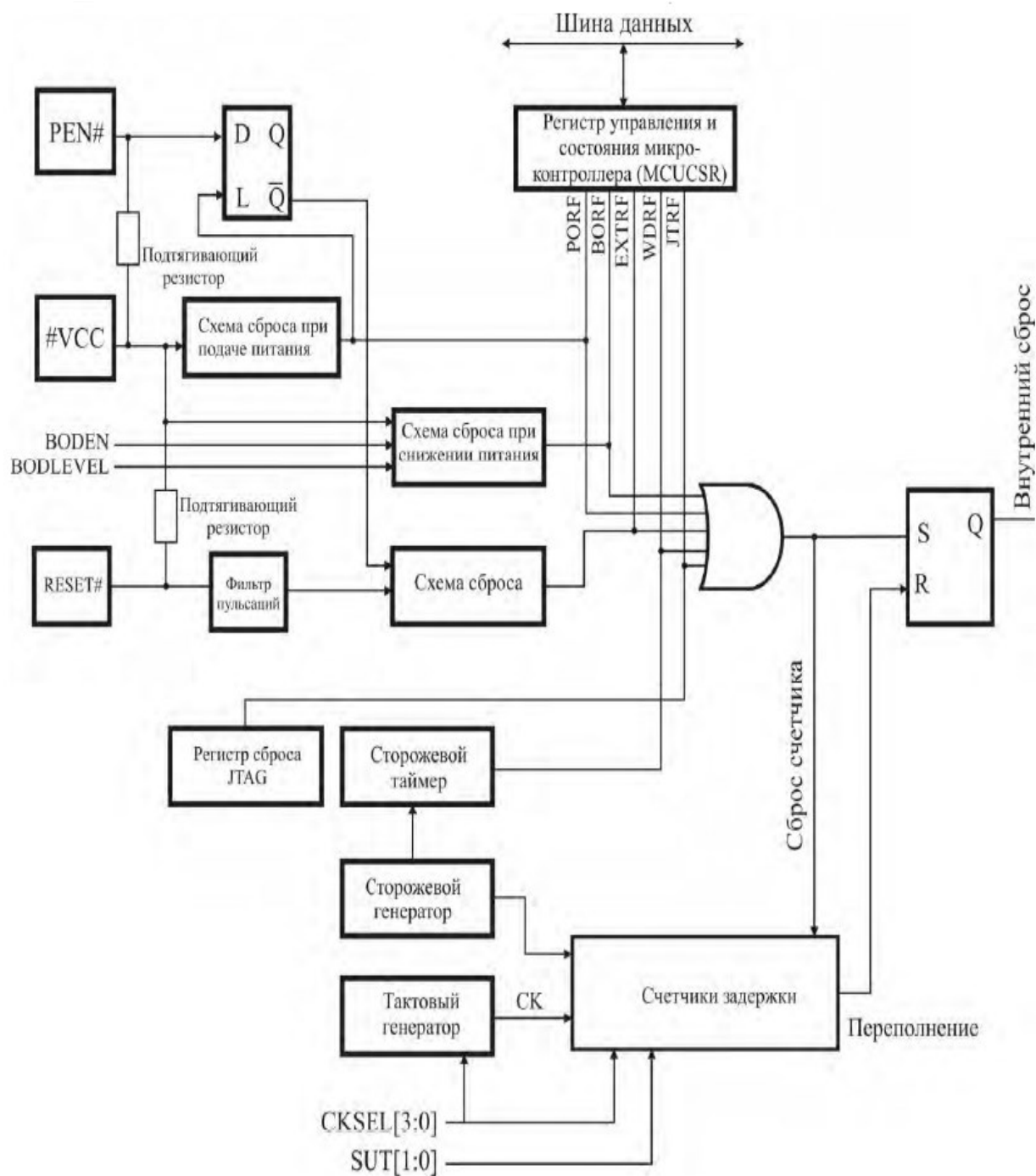


Рис. 5.2. Упрощенная структурная схема подсистемы сброса МК подсемейства *ATmega* [8] (пояснения см. в тексте)

Сопоставляя и обобщая структурные схемы, представленные на рис. 5.1 и 5.2, нетрудно увидеть, что типовая подсистема сброса МК включает в себя следующие основные функциональные узлы:

- источники сигналов сброса;
- схему объединения сигналов сброса по логическому ИЛИ;
- блок формирования внутреннего (системного) сигнала сброса.

В абсолютном большинстве семейств МК всех классов **основными событиями**, вызывающими сброс, являются:

- включение питания;
- активный уровень входа внешнего сброса (*NRST* на рис. 5.1, *RESET#* на рис. 5.2), как правило, формируемый механическим переключателем;
- окончание интервала времени ожидания, формируемого сторожевым таймером (см. подраздел 5.4).

Соответственно, источниками сигнала сброса при этом являются схемы его формирования по соответствующим событиям (включении питания, активному уровню входа внешнего сброса, окончании интервала ожидания). На рис. 5.2 данные схемы представлены в виде отдельных блоков. На рис. 5.1 1-я и 3-я из вышеперечисленных схем не показаны; их выходные сигналы обозначены как *Power reset* и *IWDG* соответственно. Схема формирования сигнала сброса по входу *NRST* (его активное состояние – нулевое) включает в себя подтягивающий резистор R_{PU} , триггер Шмидта (пороговый элемент с гистерезисом [22]) и фильтр.

Кроме перечисленных выше событий, вызывающих сброс, в большинстве семейств МК классов «*mainstream*» и «*high performance*» существует ряд **дополнительных** источников сброса. В частности, в МК подсемейства *ATmega* также реализуется (см. рис. 5.2):

- сброс по уменьшению напряжения питания ниже допустимого уровня (*Brown Out Reset, BOR*);
- сброс через отладочный интерфейс *JTAG*.

В свою очередь, в МК семейства *ARM Cortex-Mx* (рис. 5.1):

- сигнал *Power reset* формируется также при выходе из энергосберегающего режима «Ожидание» (*Standby*) (см. табл. 3.2);
- в ряде подсемейств, в частности *ARM Cortex-M4*, реализуется функция *BOR* (см. выше); при данном событии также вырабатывается сигнал *Power reset*;
- может быть реализован сброс, инициируемый ПО, при котором вырабатывается сигнал *Software reset*;
- программно может быть разрешен автоматический сброс по выполнении МК последовательности операций по переходу в ЭСР *Standby* и *Stop* (см. табл. 3.2); при этом вырабатывается сигнал *Low-power management reset* и, **вместо** перехода в ЭСР, происходит сброс МК;

- в состав большинства моделей МК семейства *ARM Cortex-Mx* кроме основного сторожевого таймера, называемого независимым (*Independent watchdog, IWDG*) входит так называемый оконный сторожевой таймер (*Window watchdog, WWDG*), вырабатывающий сигнал сброса при времени ожидания некоторого события, как большем некоторого максимально допустимого, так и меньшем заданного минимального (см. пункт 5.4.4).

В целом, перечисленные события, вызывающие сброс МК, как основные, так и дополнительные, составляют практически всю совокупность типовых источников сброса, которые могут встретиться в МК различных классов и семейств. Особенности формирования сигналов сброса от данных источников описаны далее, в подразделах 5.2 – 5.7.

Сигналы сброса от источников, имеющихся в составе конкретной модели МК, объединяются по логическому ИЛИ (см. рис. 5.1 и 5.2) и поступают на формирователь внутреннего импульса сброса МК. Основное назначение данного формирователя – генерация импульса сброса с длительностью, достаточной для установления напряжения питания МК, а также для входа ГТИ, служащего источником синхронизации МК, в рабочий режим (см. пункт 4.2.6, а также рис. 3.7).

С точки зрения архитектуры МК процедура сброса является **прерыванием** (см. раздел 7). Прерывание по сбросу не может быть запрещено (замаскировано). В ряде моделей МК класса «cost-sensitive», в частности, во многих моделях *PIC*-МК младшего подсемейства, сброс является единственным источником прерывания.

При каждом сбросе МК в некотором РСФ **фиксируется** информация о факте сброса и об его источнике. В МК *ATmega128* (1887BE7, как и в ряде других моделей подсемейства *ATmega* [6] данным РСФ является *MCUCSR* (*Micro Controller Unit Control and Status Register*) (см. рис. 5.2). Бит *PORF* устанавливается в единичное состояние, если имел место сброс по включении питания, *EXTRF* – внешний сброс, *BORF* – сброс из-за сбоя по питанию, *WDRF* – сброс от сторожевого таймера, *JTRF* – сброс через отладочный интерфейс *JTAG*. Аналогичным образом реализуется фиксация событий сброса и его источников в МК семейства *ARM Cortex-Mx*. В большинстве моделей данного семейства для этой цели используется регистр управления и статуса подсистемы сброса и синхронизации

(*RCC_CSR*). В нем также выделены специальные биты (флаги), которые устанавливаются в единичное состояние при сбросе от определенного источника (например, бит *PORRSTF* – при сбросе по включении питания, *PINRSTF* – при сбросе от внешнего источника, *IWDGRSTF* – при сбросе от независимого сторожевого таймера и т. п.).

Биты – индикаторы сброса и его источников доступны для чтения и для обнуления прикладным ПО. При этом запись нуля в данные биты регистра *MCUCSR* МК подсемейства *ATmega* осуществляется отдельно для каждого бита; в МК семейства *ARM Cortex-Mx* осуществляется общий сброс этих битов записью единицы в бит *RMVF* (*Remove reset flag*) регистра *RCC_CSR*.

Фиксация событий сброса и его источников позволяет прикладному ПО, в частности, определять причины нештатных ситуаций и выдавать сообщения о них (см. пример в подразделе 5.4).

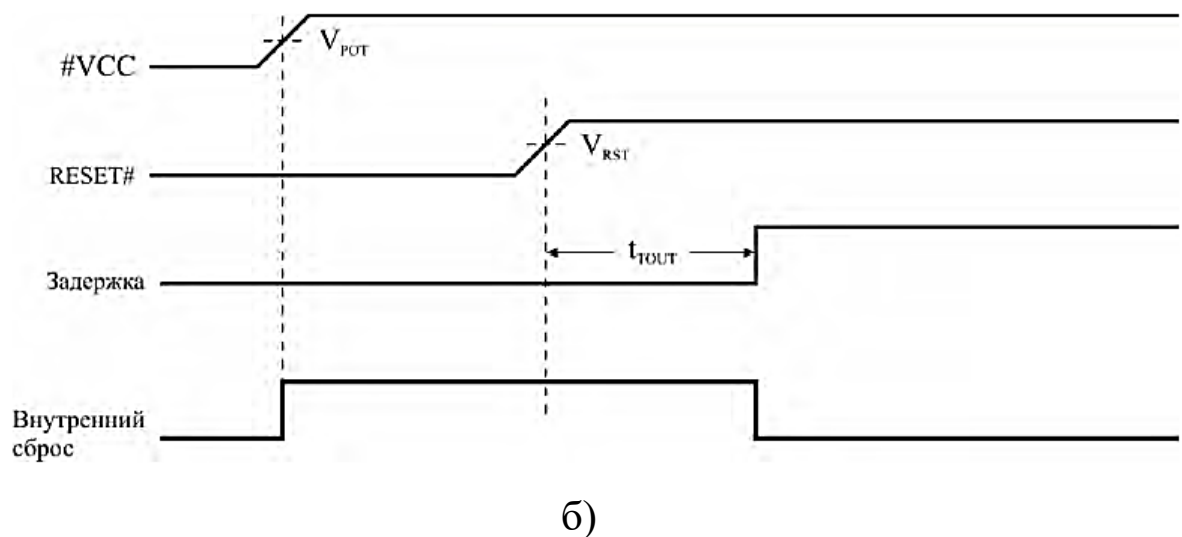
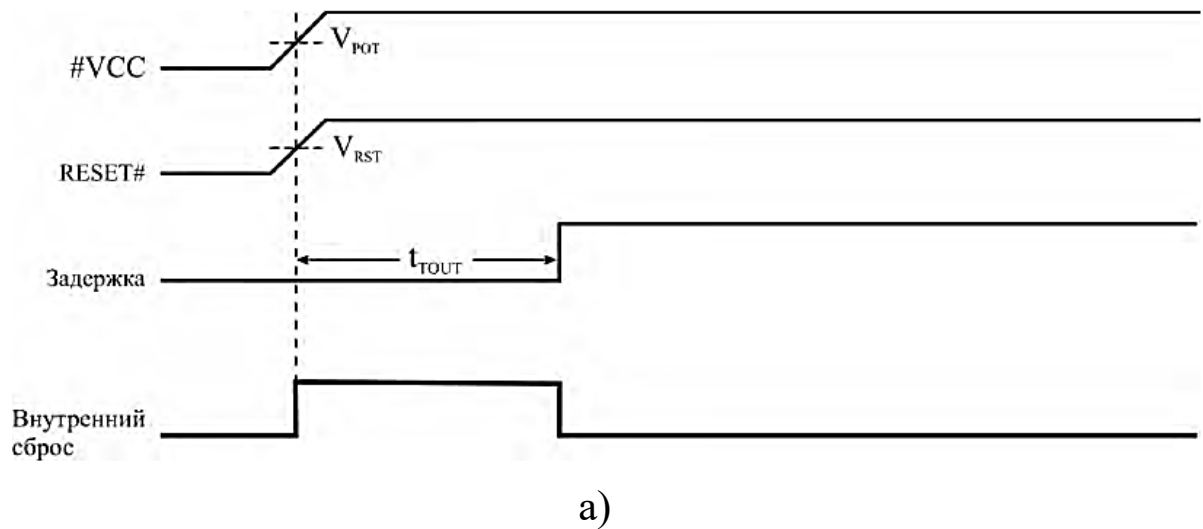
Рассмотрим принципы реализации базовых функциональных узлов подсистем сброса МК, а также особенности их структуры и архитектуры, которые необходимо учитывать при проектировании устройств на МК. В качестве типовых примеров, как и ранее, будут использоваться подсистемы сброса МК семейства *ARM Cortex-Mx* и подсемейства *ATmega* (см. рис. 5.1 и 5.2 соответственно).

5.2 Сброс МК по включении питания и при сбоях по питанию

Вопросы реализации сброса МК по включении питания и при сбоях по питанию частично рассмотрены в пункте 3.3.1.

Внутренняя цепь автоматического формирования сигнала сброса по включении питания имеется в составе абсолютного большинства МК всех классов и семейств. Может отсутствовать в ряде моделей МК класса «*cost-sensitive*»; в таких случаях она должна быть реализована на внешних элементах, например, по типовой схеме, представленной на рис. 3.6. Описание принципа ее работы приведено в сопутствующем тексте пункта 3.3.1. В целом, таков же и принцип работы внутренних схем автоматизированного сброса МК по включении питания.

Временные диаграммы формирования сигнала сброса МК подсемейства *ATmega* по включении питания приведены на рис. 5.3 [6, 8].



V_{POT} – пороговое напряжение сброса при подаче питания

V_{RST} – пороговое напряжение по входу $RESET\#$

t_{TOUT} – минимально необходимая длительность внутреннего импульса сброса МК (см. пункт 4.4.1)

Рис. 5.3. Временные диаграммы формирования сигнала сброса МК подсемейства *ATmega* [6, 8] при не задействованном входе $RESET\#$ (а) и при управлении входом $RESET\#$ от внешнего источника (б) (см. также пояснения в тексте)

Как видно из рис. 5.3, сигнал внутреннего сброса МК устанавливается в активное (единичное) состояние по достижении напряжением питания порогового уровня. Отсчет интервала времени t_{TOUT} начинается от:

- момента установки в единицу сигнала внутреннего сброса, если вход *RESET#* не задействован, т. е. подключен через подтягивающий резистор к напряжению питания (см. рис. 5.2);

- момента достижения порогового уровня напряжением на входе *RESET#*, если данный вход управляется сигналом от внешнего источника (активное состояние входа *RESET#* - нулевое).

Вопросы формирования интервала времени t_{TOUT} освещены в подразделе 5.8.

Временные диаграммы формирования сигнала сброса МК семейства *ARM Cortex-Mx* по включении питания, а также их описание приведены в пункте 3.3.1 (см. рис. 3.8 и сопроводительный текст к нему).

Вопросы реализации функции сброса МК при сбоях по питанию достаточно подробно освещены в пункте 3.3.1, и возвращаться к ним в данном подразделе нет необходимости.

5.3 Внешний сброс МК

Внешним называют сброс МК по активному уровню сигнала на некотором выводе БИС МК, специально выделенном или сконфигурированном для данной цели. Например, в МК 1887BE7T под вход внешнего сброса *RESET#* (см. рис. 5.2) выделен вывод БИС МК, предназначенный для использования только по данному назначению. Под вход *NRST* (см. рис. 5.1) большинства моделей МК семейства *ARM Cortex-Mx* также специально выделен некоторый вывод БИС МК. В тоже время в ряде моделей *PIC*-МК [4, 17] под вход внешнего сброса конфигурируется определенный вывод одного из портов.

Как правило, источником внешнего сброса служит механический переключатель, обычно – кнопочный, без фиксации. Типовая схема его подключения ко входу внешнего сброса МК представлена на рис. 5.4 [6, 10]. Предполагается, что активное состояние данного входа – нулевое (что характерно для большинства семейств МК). Резистор *R1* является подтягивающим, и обеспечивает надежный уровень логической единицы при разомкнутом контакте. Рекомендуемое значение его сопротивления для МК подсемейства *ATmega* – 10 кОм, для семейства *ARM Cortex-Mx* – 100 кОм (в схемах подключения ряда моделей может отсутствовать, его функцию выполняет внутренний подтягивающий

резистор, следует руководствоваться *datasheet*). Конденсатор *C1* служит для обеспечения ширины импульса сброса, не меньшей, чем минимально допустимая, а также для фильтрации «дребезга» механического контакта и защиты от ложных сбросов. Типовое значение его емкости – 0,1 мкФ.

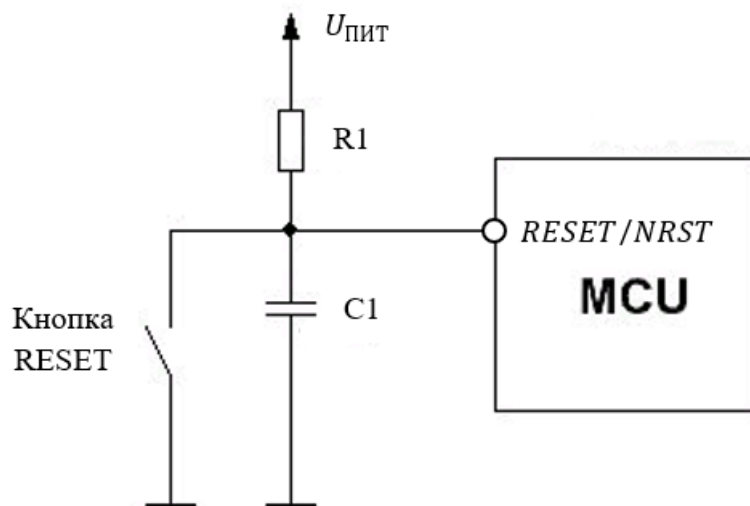


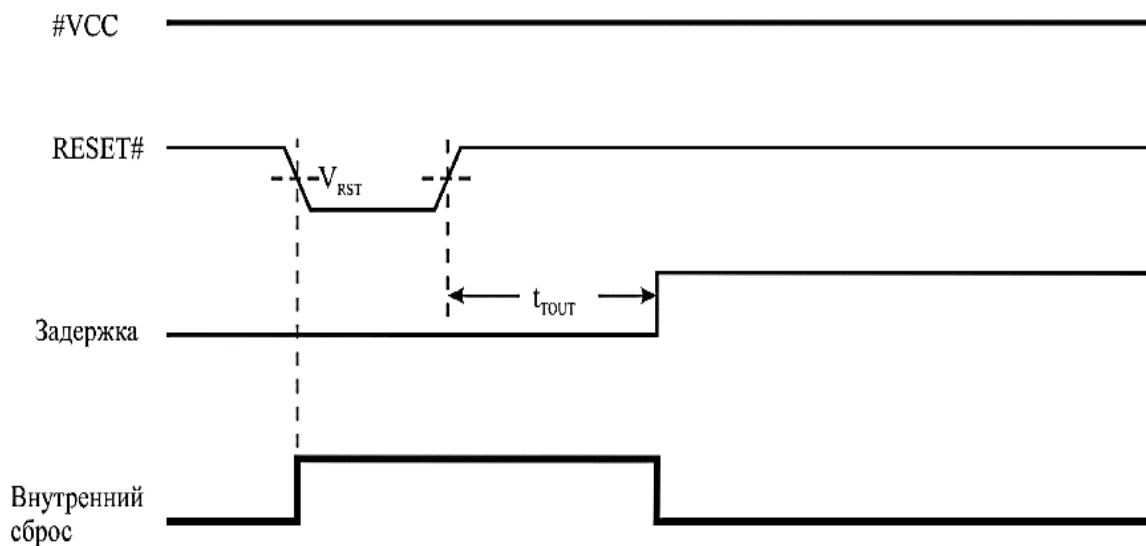
Рис. 5.4. Типовая схема подключения кнопочного переключателя ко входу внешнего сброса МК

В принципе, возможен внешний сброс не только механическим переключателем, но и от некоторого электронного источника сигнала сброса.

Важно отметить, что длительность интервала удержания входа внешнего сброса МК в активном состоянии должна быть не меньше указываемой в технической документации на соответствующую модель МК.

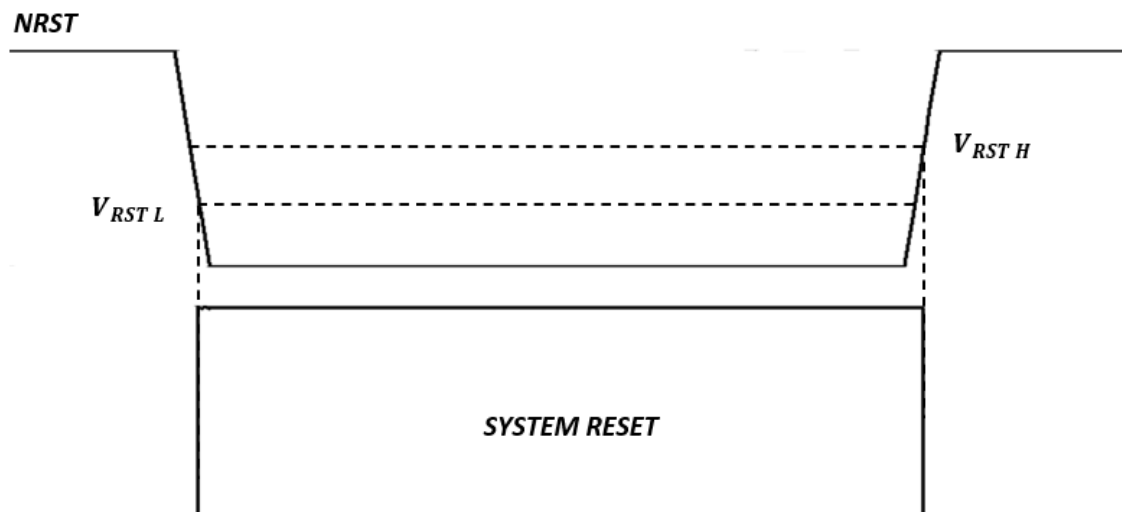
Следует также отметить, что в ряде подсемейств / моделей МК функция внешнего сброса может быть запрещена. Например, в ряде моделей младшего подсемейства *PIC*-МК [4, 17] разрешение / запрет данной функции осуществляются битом *MCLR* слова конфигурации МК, доступного только в режиме программирования энергонезависимой памяти. При активном (единичном) состоянии данного бита выделенный под вход сброса вывод одного из портов МК может использоваться для подачи сигнала сброса от внешнего источника. При нулевом состоянии бита *MCLR* соответствующий вывод порта является входом общего назначения, а внутренняя шина сигнала внешнего сброса МК автоматически переводится в пассивное состояние.

Временные диаграммы формирования внутреннего сигнала сброса МК подсемейства *ATmega* по активизации входа внешнего сброса приведены на рис. 5.5. Аналогичные диаграммы для МК семейства *ARM Cortex-Mx* представлены на рис. 5.6.



V_{RST} – пороговое напряжение по входу внешнего сброса

Рис. 5.5. Временные диаграммы формирования внутреннего сигнала сброса МК подсемейства *ATmega* по активизации входа внешнего сброса [8]



$V_{RST L}$ и $V_{RST H}$ – соответственно нижнее и верхнее пороговое напряжение по входу внешнего сброса

Рис. 5.6. Временные диаграммы формирования внутреннего сигнала сброса МК семейства *ARM Cortex-Mx* по активизации входа внешнего сброса [13 – 15]

5.4 Сброс от сторожевых таймеров

5.4.1. Назначение и принципы применения сторожевых таймеров

Сторожевым таймером (*Watchdog Timer* или просто *Watchdog*) МК называют функциональный блок, основное назначение которого – предотвращение «зависаний» МК, вызванных сбоями в работе его аппаратных средств или / и ПО. Они могут привести к серьезным последствиям, особенно если МК работает в составе автоматизированной системы контроля и управления техническим объектом, функционирующей без участия человека-оператора.

По крайней мере, один сторожевой таймер входит в состав абсолютного большинства моделей современных МК общего назначения всех классов (см. рис. 1.1 – 1.5). В структуре МК ряда семейств, например, *ARM Cortex-Mx*, кроме основного сторожевого таймера (*IWDG*, см. рис. 5.1), входит сторожевой таймер, называемый оконным (*WWDG*, также см. рис. 5.1).

Основная функция «типового» (не оконного) сторожевого таймера – сброс МК, если длительность выполнения некоторого этапа прикладной программы МК (в первую очередь – цикла) превысит максимально допустимое значение. Оконный таймер осуществляет сброс МК, если указанная длительность как меньше нормального значения, так и больше максимально допустимой.

Как правило, прикладная программа работы МК представляет собой «бесконечный» цикл: ее запуск происходит по включении питания, а останов – по его выключению. Поэтому наиболее «радикальным» алгоритмом применения сторожевого таймера является следующий.

1. По включении питания – конфигурирование сторожевого таймера на отсчет интервала времени с длительностью, равной максимально допустимому времени выполнения общего цикла прикладной программы МК.

2. Запуск прикладной программы с одновременным запуском отсчета времени сторожевым таймером.

3. Если очередной шаг общего цикла программы МК завершился до окончания интервала времени, отсчет которого производится сторожевым таймером – производится его сброс (см. команды *CLRWDT* и *WDR* в табл. 2.6 и 2.7 соответственно) или перезагрузка,

в зависимости от архитектуры семейства, к которому принадлежит МК. Затем осуществляется переход к следующему шагу общего цикла, и снова запускается отсчет времени сторожевым таймером.

4. Если интервал времени, отсчитываемый сторожевым таймером, истек, а его обнуление или перезагрузка не произведены – произойдет сброс МК сторожевым таймером. В подпрограмму обслуживания прерывания по сбросу при этом должен быть включен опрос битов – указателей его источника (см., например, регистр *MCUCSR* на рис. 5.2). Если в результате опроса выявлено, что источником сброса был сторожевой таймер – осуществляется переход к обработке соответствующей нештатной ситуации (например, к анализу значений переменных, состояний регистров и т. п., которые потенциально могли вызвать «зависание», с их корректировкой).

Примечание. В ряде конкретных случаев признаком нештатной ситуации может быть не только время выполнения общего цикла, большее максимально допустимого, но и меньшее некоторого минимально возможного. В таких случаях рационально применение оконного сторожевого таймера, который осуществит сброс МК при:

- попытке сброса / перезагрузки таймера, если он еще не отсчитал минимально допустимое время выполнения общего программного цикла МК;

- если истекло максимально допустимое время выполнения цикла, а сброс / перезагрузка таймера не произведены.

Описанный выше алгоритм обеспечивает сброс МК сторожевым таймером при «зависаниях» по практически любой из возможных причин. В ряде случаев может потребоваться контроль сторожевым таймером времени выполнения только некоторого частного цикла прикладной программы (например, установления связи с другим МК со сбросом МК при потере связи). Алгоритм применения сторожевого таймера при этом аналогичен вышеописанному: запуск таймера при входе в цикл; его программные сброс или перезагрузка при нормальном выходе из цикла.

5.4.2. Общие принципы работы сторожевых таймеров

Основным функциональным узлом сторожевого таймера является счетчик, суммирующий или вычитающий (в зависимости от архитектуры конкретного семейства / подсемейства МК). В

частности, сторожевой таймер МК семейства *AVR* реализован на основе суммирующего счетчика [6], а большинства моделей МК семейства *ARM Cortex-Mx* – вычитающего счетчика с предустановкой, т. е. с возможностью предварительной записи в него некоторого числа перед запуском [9].

Тактирование счетчика сторожевого таймера осуществляется ГТИ, не останавливаемым ни в одном из энергосберегающих режимов: например, в МК семейства *AVR* – специально выделенным ГТИ (см. рис. 5.2), в большинстве моделей МК семейства *ARM Cortex-Mx* – генератором *LSI* (см. рис. 4.1). Тактовые импульсы поступают на счетчик сторожевого таймера через предделитель с программно-управляемым коэффициентом деления.

«Типовой» (не оконный) сторожевой таймер на основе суммирующего счетчика вырабатывает сигнал сброса при его переполнении, а на основе вычитающего счетчика – после обнуления его содержимого. Оконный таймер, как правило, строится на основе вычитающего счетчика, и вырабатывает сигнал сброса:

- при попытке его программной переустановки, если содержимое счетчика еще не достигло верхнего порогового значения (т. е. если он еще не отсчитал минимально допустимый интервал времени);

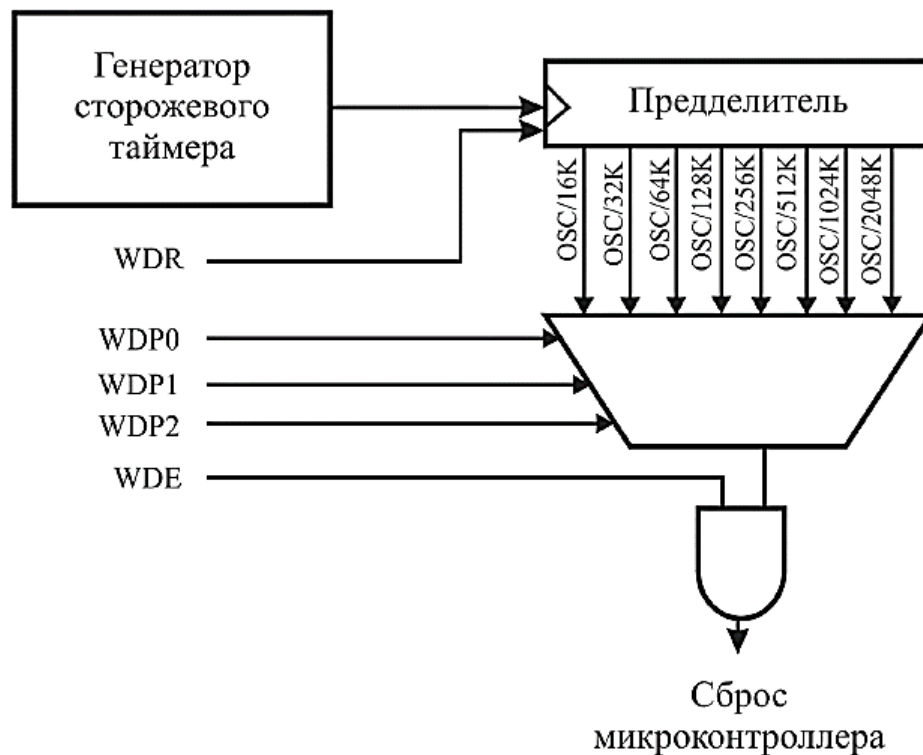
- по достижении содержимым счетчика нижнего порогового значения (т. е. если он отсчитал максимально допустимый интервал времени);

(см. также рис. 5.13).

В качестве типовых примеров рассмотрим структуру и архитектуру сторожевых таймеров МК семейства *AVR* и семейства *ARM Cortex-Mx*.

5.4.3. Сторожевой таймер МК семейства *AVR*

В структуру МК данного семейства входит один сторожевой таймер, структура и архитектура которого достаточно схожи практически у всех моделей МК данного семейства. Будем рассматривать их на примере сторожевого таймера МК *ATmega128* (1887BE7T), упрощенная структурная схема которого представлена на рис. 5.7.



OSC – частота генератора сторожевого таймера
 $16K = 16 \times 1024$, $32K = 32 \times 1024$ и т. д. – коэффициенты деления частоты
 WDR – сигнал сброса таймера
 $WDP0...WDP2$ и WDE – соответственно биты выбора коэффициента деления предделителя и бит разрешения работы сторожевого таймера в регистре управления им

Рис. 5.7. Упрощенная структурная схема сторожевого таймера МК 1887BE7T [8]

Сторожевой таймер включает в себя:

- ГТИ;
- предделитель, одновременно служащий и счетчиком таймера;
- программно-управляемый мультиплексор, посредством которого осуществляется выбор выходного сигнала предделителя, поступающего на логику формирования сигнала сброса;
- формирователь сигнала сброса.

Тактирование сторожевого таймера осуществляется специально выделенным ГТИ, работа которого не прекращается ни в одном из энергосберегающих режимов. Его частота, обозначенная на рис. 5.7 как OSC , зависит от конкретной модели МК. Например, у МК 1887BE7T ее номинальное значение равно 500 кГц при напряжении питания, равном 5 В. После запуска счетчика- предделителя на

каждом из его выходов через интервал времени, равный K_{di}/OSC (где K_{di} – коэффициент деления частоты по i -му выходу, см. рис. 5.7), сформируется импульс переполнения. С выбранного мультиплексором выхода импульс переполнения поступает на один из входов логического элемента 2И, на другой вход которого подается сигнал разрешения работы сторожевого таймера. Если этот сигнал находится в активном (единичном) состоянии, и до формирования импульса переполнения по выбранному мультиплексором выходу таймер не был сброшен, данный импульс запустит сброс МК.

Нетрудно вычислить, что при номинальном значении частоты ГТИ сторожевого таймера, равном 500 кГц, длительность интервала времени от момента запуска таймера до формирования импульса переполнения на младшем выходе предделителя равна примерно 32,8 мс, а на старшем – примерно 4,19 с. Реальные типовые длительности данного интервала несколько меньше: по младшему выходу – порядка 28 мс, по старшему – порядка 3,6 с [8], ввиду того что типовая частота ГТИ сторожевого таймера несколько выше номинальной (что характерно для многих моделей МК семейства AVR). Типовые длительности интервала времени от запуска сторожевого таймера до формирования сигнала переполнения по каждому из выходов его предделителя всегда приводятся в технической документации на конкретную модель МК.

Для сброса МК при «зависании» выбирается выход предделителя, значение K_{di}/OSC по которому является ближайшим большим к максимально допустимому времени выполнения цикла, предохраняемого сторожевым таймером от «зависания». Выбор осуществляется битами $WDP0...WDP2$ регистра управления сторожевым таймером (см. рис. 5.7): битовой комбинации 000 соответствует младший выход предделителя, 111 – старший.

Временные диаграммы формирования внутреннего сигнала сброса МК семейства AVR при переполнении сторожевого таймера представлены на рис. 5.8.

Конфигурирование сторожевого таймера МК семейства AVR осуществляется посредством регистра управления им, которому в большинстве моделей семейства присвоено имя $WDTCR$ (*Watchdog Timer Control Register*), в ряде моделей – $WDTCR$ (*Watchdog Timer Control and Status Register*). На рис. 5.9 в качестве типового примера

представлен формат регистра управления сторожевым таймером МК 1887BE7T [8].

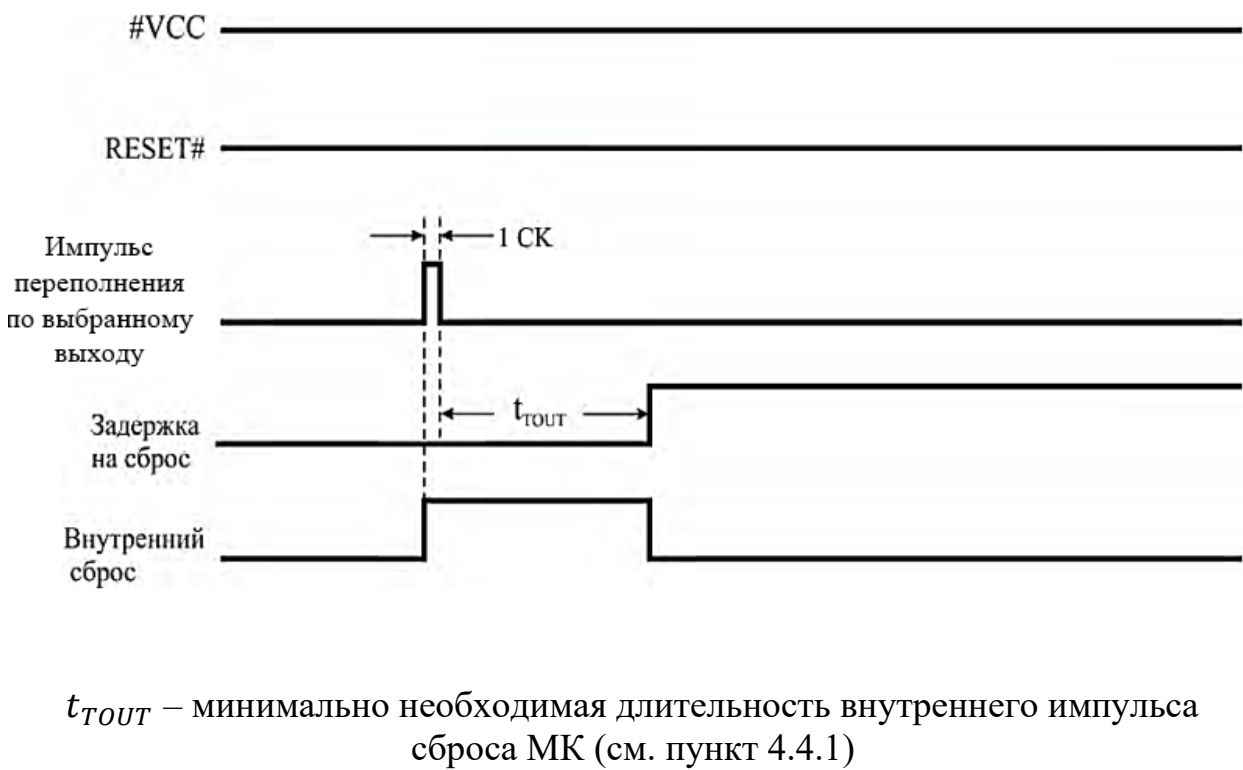


Рис. 5.8. Временные диаграммы формирования внутреннего сигнала сброса МК семейства AVR при переполнении сторожевого таймера [8]

Бит	7	6	5	4	3	2	1	0	
	-	-	-	WDCE	WDE	WDP2	WDP1	WDP0	WDTCR
Чтение/запись	Ч	Ч	Ч	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	
Начальное значение	0	0	0	0	0	0	0	0	

WDCE – бит разрешения конфигурирования сторожевого таймера
Назначение битов с 3-го по 0-й – см. рис. 5.7

Рис. 5.9. Формат регистра управления сторожевым таймером МК ATmega128 (1887BE7T) [8]

Ввиду того, что сторожевой таймер является функциональным блоком системного назначения, имеются некоторые особенности процедур его конфигурирования, запуска и остановки, позволяющие

избегать сбоев на системном уровне при выполнении данных процедур.

После установки бита *WDCE* в единицу он автоматически сбрасывается по истечении 4-х тактов ЦП. Поэтому запись битов регистра управления сторожевым таймером должна производиться в течение 4-х тактов после записи единицы в бит *WDCE* (имеются исключения, см. далее).

Существуют три уровня безопасности при работе со сторожевым таймером (0-й, 1-й и 2-й). Уровень безопасности задается состояниями 1-го (*M103C*) и 0-го (*WDTON*) битов расширенного конфигурационного байта МК, доступного для записи только в режиме программирования энергонезависимой памяти. 0-му уровню безопасности соответствуют значения *WDTON* = 1, *M103C* = 0 (это их состояния по умолчанию); 1-му - *WDTON* = 1, *M103C* = 1; 2-му - *WDTON* = 0, независимо от состояния бита *M103C*.

На **0-м уровне** сторожевой таймер, по умолчанию, отключен. Его включение производится записью единицы в бит *WDE*, выбор выхода предделителя – записью соответствующего кода в биты *WDP*. Установка бита *WDCE* при этом не требуется. Остановка сторожевого таймера должна производиться следующей последовательностью операций.

1. Записать логическую единицу в *WDCE* и *WDE* одной инструкцией. Логическая единица должна быть записана в бит *WDE*, даже если до выполнения данной операции в нем уже была записана логическая единица.

2. В течение следующих четырех тактов записать логический ноль в *WDE*, что приводит к отключению сторожевого таймера.

На **1-м уровне** таймер, как и на 0-м, отключен по умолчанию. Его включение также производится записью единицы в бит *WDE*, без установки бита *WDCE*. Остановка таймера, а также выбор выхода предделителя должны выполняться следующим образом.

1. Записать логическую единицу в биты *WDCE* и *WDE* одной инструкцией. Логическая единица должна быть записана в *WDE* независимо от предыдущего значения бита *WDE*.

2. В течение следующих четырех тактов одной инструкцией записать желаемое значение в биты *WDE* и *WDP*, одновременно сбрасывая бит *WDCE*.

Наконец, на **2-м уровне** таймер включен постоянно, его остановка невозможна. Выбор выхода предделителя должен производиться следующей последовательностью операций.

1. Записать логическую единицу в *WDCE* и *WDE* одной инструкцией. Несмотря на то, что бит *WDE* всегда установлен, для запуска временной последовательности в него все равно необходимо записывать логическую единицу.

2. В течение следующих четырех тактов одной инструкцией записать желаемое значение в биты *WDP*, одновременно сбрасывая бит *WDCE*. Значение, записываемое в бит *WDE*, не оказывает никакого влияния.

Описанные выше процедуры работы со сторожевым таймером, на первый взгляд, могут показаться излишне сложными. Однако, они обеспечивают его защиту от случайных отключений или изменений интервала времени от запуска до переполнения, которые могут вызвать серьезные сбои в работе МК. Следует отметить, что с той же целью преднамеренные «сложности» введены и в процедуры конфигурирования и запуска сторожевых таймеров семейства *ARM Cortex-Mx* (см. пункт 5.4.4).

В заключение необходимо отметить, что в ряде моделей МК семейства *AVR* (в частности, подсемейства *ATtiny* [7]) возможен вариант конфигурирования сторожевого таймера, при котором он по переполнении не производит сброс МК, а только генерирует запрос на прерывание.

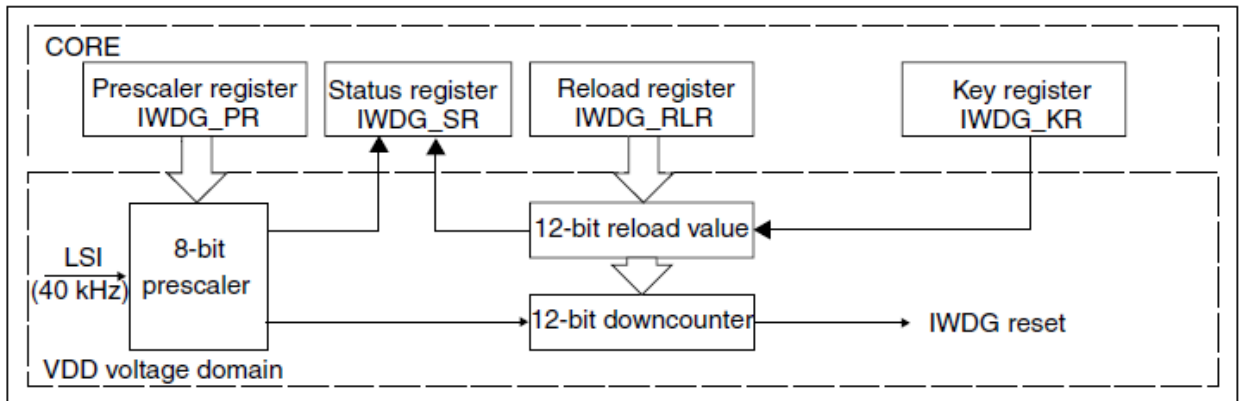
5.4.4. Сторожевые таймеры МК семейства *ARM Cortex-Mx*

В состав большинства моделей МК семейства *ARM Cortex-Mx* входят два сторожевых таймера: независимый (*IWDG*) и оконный (*WWDG*).

5.4.4.1. Независимый сторожевой таймер реализуется по обобщенной структурной схеме, приведенной на рис. 5.10. Слово «независимый» в его названии объясняется тем, что он тактируется от генератора *LSI*, который всегда работает, независимо от того, находится ли МК в нормальном режиме функционирования или в каком-либо из энергосберегающих режимов.

Основным функциональным узлом таймера является вычитающий счетчик с предустановкой, при которой в него загружается некоторое число, задающее длительность

отсчитываемого интервала времени. После запуска или перезагрузки (т. е. очередной предустановки) счетчика он начинает отсчет «вниз», с уменьшением его содержимого на единицу по приходу каждого тактового импульса. Если счетчик не будет перезагружен до момента его антипереполнения (т. е. до момента, когда его содержимое, после обнуления, с приходом следующего тактового импульса примет максимальное значение, $0xFF$), выработается сигнал сброса МК, *IWDG reset* (см. рис. 5.10).



Reload register – регистр предустановки

Key register – регистр ключевого слова

Reload value – число, загружаемое в счетчик при предустановке

Downcounter – вычитающий счетчик

Рис. 5.10. Обобщенная структурная схема независимого сторожевого таймера МК семейства *ARM Cortex-Mx* [13 – 15]

Длительность интервала времени от момента запуска / перезагрузки счетчика до его антипереполнения рассчитывается по выражению [13 – 15]:

$$t_{IWDG} = \frac{PR \times (RLV + 1)}{f_{LSI}}; \quad (5.1)$$

где PR – коэффициент деления предделителя; RLV – число, загружаемое в счетчик при предустановке / перезагрузке; f_{LSI} – частота генератора *LSI* (номинальное значение – 40 кГц).

Значение коэффициента PR задается регистром *IWDG_PR*. У большинства моделей МК семейства *ARM Cortex-Mx* оно может принимать значения 4, 8, 16, 32, 64, 128 и 256. Число RLV , являясь 12-битовым, очевидно, находится в диапазоне от $0x000$ до $0xFFF$.

Следовательно, в соответствии с выражением (5.1), минимальная длительность интервала t_{IWDG} равна 0,1 мс (при $PR = 4$, $RLV = 0x000$), максимальная – примерно 26,2 с (при $PR = 256$, $RLV = 0xFFFF$). Соответственно, значения PR и RLV должны выбираться таким образом, чтобы длительность интервала t_{IWDG} превышала (с запасом порядка 5%) максимально допустимое время выполнения цикла, предохраняемого сторожевым таймером от «зависания».

Как при программировании сторожевого таймера МК семейства *AVR* (см. пункт 5.4.3), для защиты от случайных ошибок при конфигурировании, запуске и перезагрузке независимого сторожевого таймера МК семейства *ARM Cortex-Mx* в данные процедуры введены специальные операции. Они реализуются через регистр ключевого слова (*IWDG_KR* на рис. 5.10). Для выполнения какой-либо из вышеперечисленных процедур в данный регистр должно быть записано определенное ключевое слово, в том числе [13 – 15]:

- для включения таймера – $0xCCCC$;
- для изменения содержимого регистров *IWDG_PR* и *IWDG_RLR* – $0x5555$;
- для перезагрузки счетчика – $0xAAAA$.

По включении питания в счетчик таймера загружается число $0xFFFF$, коэффициент деления предделителя устанавливается равным 64-м. Это обеспечивает завершение описанной далее процедуры начального конфигурирования таймера до истечения интервала времени t_{IWDG} (см. выражение (5.1)).

Начальное конфигурирование независимого сторожевого таймера должно производиться следующей последовательностью операций [13 – 15].

1. Включить таймер записью слова $0xCCCC$ в регистр *IWDG_KR*. Его счетчик при этом начинает отсчет «вниз», от значения $0xFFFF$ (см. выше).

2. Разрешить доступ к регистрам *IWDG_PR* и *IWDG_RLR* записью слова $0x5555$ в регистр *IWDG_KR*.

3. Записать требуемые значения PR и RLV (см. выражение (5.1)) в регистры *IWDG_PR* и *IWDG_RLR*.

4. Дождаться завершения обновления содержимого указанных регистров, признаком которого является сброс в ноль битов *RVU* (*Watchdog counter reload value update*) и *PVU* (*Watchdog prescaler value update*) регистра статуса (*IWDG_SR* на рис. 5.10).

5. Загрузить содержимое регистра *IWDG_RLR* в счетчик таймера записью слова *0xAAAA* в регистр *IWDG_KR*, после чего начнется отсчет интервала времени t_{IWDG} (см. выражение (5.1)).

Перезагрузка независимого сторожевого таймера (т. е. установка его счетчика в начальное состояние, равное содержимому регистра *IWDG_RLR*) реализуется также записью слова *0xAAAA* в регистр *IWDG_KR*. В процессе работы МК перезагрузка независимого сторожевого таймера должна производиться с периодичностью, меньшей, чем длительность интервала времени t_{IWDG} . Если по истечении данного интервала перезагрузка таймера не будет выполнена, он произведет сброс МК.

5.4.4.2. Оконный сторожевой таймер МК семейства *ARM Cortex-Mx* реализуется по обобщенной структурной схеме, представленной на рис. 5.11 [13 – 15].

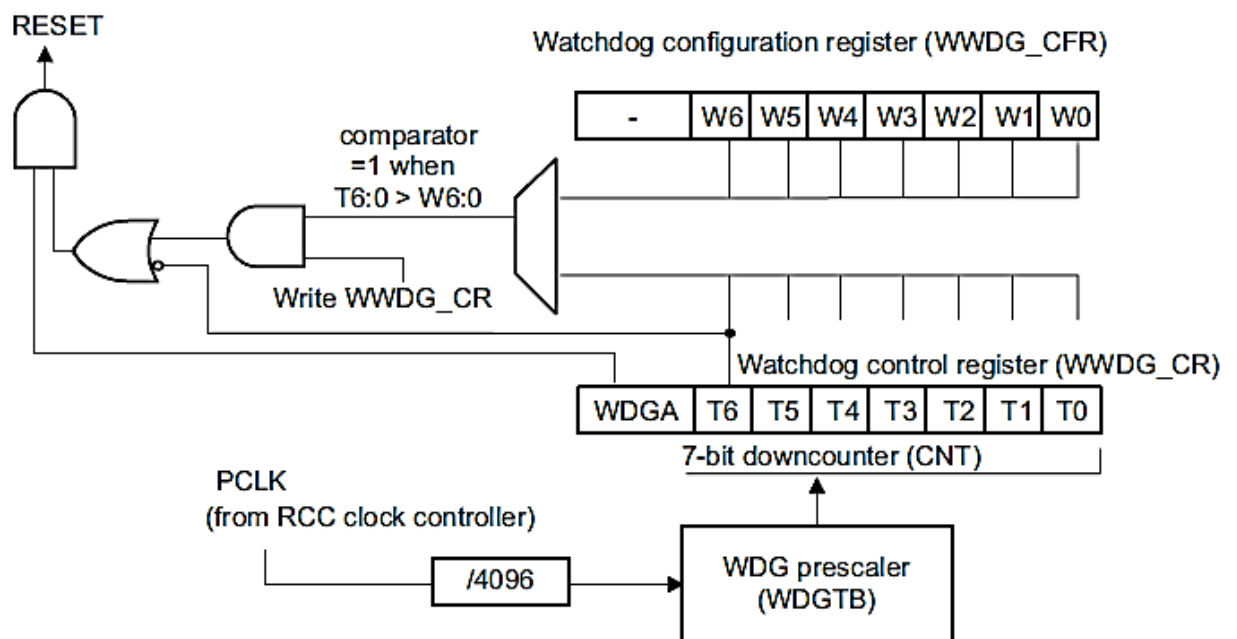


Рис. 5.11. Обобщенная структурная схема оконного сторожевого таймера МК семейства *ARM Cortex-Mx* [13 – 15]
(пояснения см. в тексте)

Оконный сторожевой таймер включает в себя следующие основные функциональные узлы (см. рис. 5.11):

- вычитающий счетчик (*downcounter*);
- регистр конфигурации *WWDG_CFR*, задающий верхнее пороговое значение содержимого счетчика, а также коэффициент

деления предделителя его тактовой частоты; формат данного регистра одинаков практически у всех моделей семейства, и представлен на рис. 5.12 [13 – 15];

- предделитель тактовой частоты счетчика, состоящий из 2-х каскадов; коэффициент деления первого из них равен 4096, второго (обозначенного как *WDG prescaler*) может быть принимать значения 1, 2, 4 или 8, и задается двухбитовым полем *WDGTB* регистра *WWDG_CFR* (см. рис. 5.12);

- компаратор, осуществляющий сравнение содержимых счетчика и регистра *WWDG_CFR*;

- схему формирования сигнала сброса на элементах 2И и элементе 2ИЛИ.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						EWI	WDGTB[1:0]		W[6:0]						
						rs	rw		rw						

W[6:0] – верхнее пороговое значение содержимого счетчика

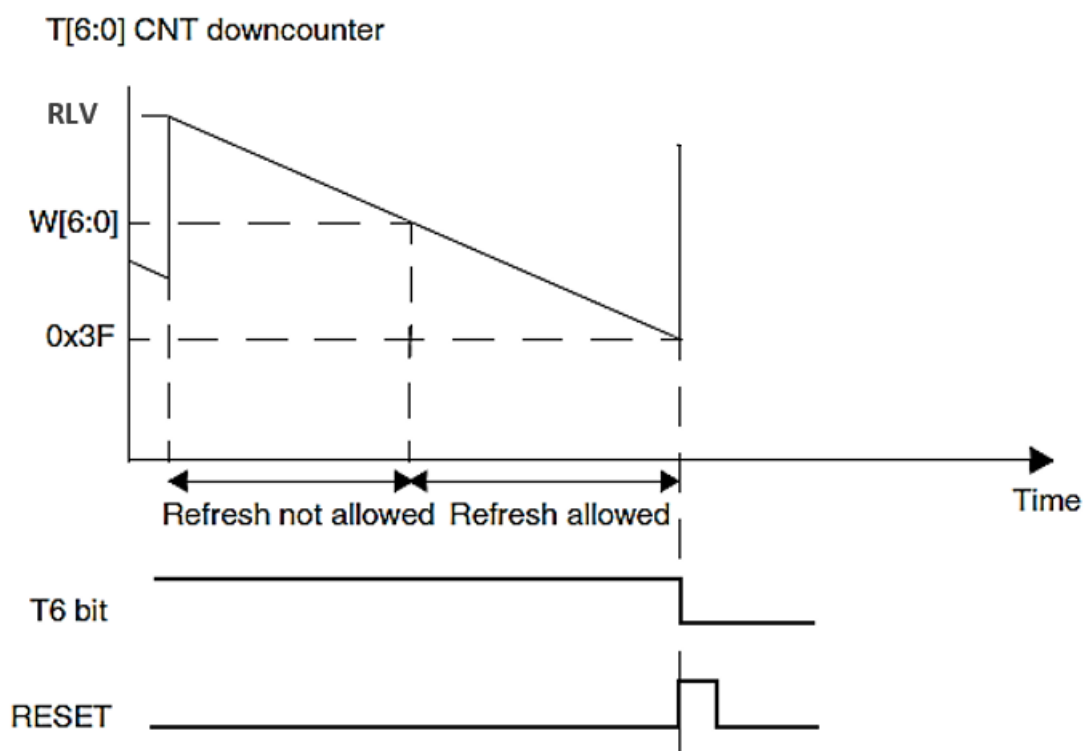
WDGTB[1:0] – биты, задающие коэффициент деления предделителя
(см. рис. 5.11)

EWI – бит разрешения прерываний от оконного сторожевого таймера
(доступен только для чтения и для программной установки в единицу, обнуляется аппаратно при сбросе МК)

Рис. 5.12. Формат регистра конфигурации оконного сторожевого таймера МК семейства *ARM Cortex-Mx* [13 – 15]

Временные диаграммы, поясняющие принцип работы оконного сторожевого таймера МК семейства *ARM Cortex-Mx*, приведены на рис. 5.13 [13 – 15].

Входным сигналом предделителей таймера является синхросигнал домена *APB* (см. рис. 4.1 и 5.11), поэтому оконный сторожевой таймер не является независимым: его тактирование прекращается в энергосберегающих режимах *Stop* и *Standby* (см. табл. 3.2). Если МК не находится одном из данных режимов, счетчик таймера работает постоянно, в вычитающем режиме; его содержимое уменьшается на единицу по приходу каждого тактового импульса.



$T[6:0]$ CNT – содержимое счетчика
 RLV – число, загружаемое в счетчик при перезагрузке
 $Refresh\ not\ allowed / Refresh\ allowed$ – обновление (т. е. перезагрузка счетчика) запрещено / обновление разрешено

Рис. 5.13. Временные диаграммы работы оконного сторожевого таймера МК семейства *ARM Cortex-Mx* [13 – 15]
(пояснения см. в тексте)

Генерирование сигнала сброса разрешено при единичном состоянии бита $WDGA$ регистра управления таймером, $WWDG_CR$ (см. рис. 5.11). После включения питания значение данного бита равно нулю. Его установка в единичное состояние осуществляется программно, а обнуление производится только при сбросе МК.

Если бит $WDGA$ находится в активном (единичном) состоянии, оконный сторожевой таймер генерирует сигнал сброса МК в следующих случаях (см. схему формирования сигнала сброса на рис. 5.11, а также рис. 5.13):

- при попытке перезагрузки счетчика (записи в регистр $WWDG_CR$) до достижения счетчиком верхнего порогового

значения ($W[6:0]$), т. е. до истечения минимально допустимого интервала времени с момента ближайшей перезагрузки;

- если перезагрузка счетчика не произошла до истечения максимально допустимого интервала времени с момента ближайшей перезагрузки, т. е. по достижении содержимым счетчика нижнего порогового уровня ($0x3F$), признаком которого служит обнуление бита $T6$.

Длительность минимально допустимого интервала времени между перезагрузками счетчика рассчитывается по выражению (см. рис. 5.13) [13 – 15]:

$$t_{min} = \frac{4096 \times PR \times (RLV - W[6:0] + 1)}{f_{PCLK}}; \quad (5.2)$$

где PR – коэффициент деления предделителя (см. рис. 5.11); RLV – число, загружаемое в счетчик при перезагрузке; f_{PCLK} – частота синхронизации домена APB .

В свою очередь, длительность максимально допустимого интервала времени между перезагрузками счетчика, рассчитывается по следующему выражению (также см. рис. 5.13) [13 – 15]:

$$t_{max} = \frac{4096 \times PR \times (RLV - 0x40 + 1)}{f_{PCLK}}; \quad (5.3)$$

Значения t_{min} и t_{max} представляют собой, по существу, минимальное и максимальное допустимое время («временное окно») выполнения цикла, предохраняемого оконным сторожевым таймером от «зависания». Если в пределах данного окна не будет произведена перезагрузка счетчика, таймер сгенерирует сигнал сброса МК. Процедура перезагрузки должна быть введена в тело цикла.

Значения PR и RLV выбираются, исходя из обеспечения требуемых, в соответствии с решаемой прикладной задачей, длительностей интервалов t_{min} и t_{max} . При этом допустимые значения RLV находятся в пределах от $0x40$ до $0x7F$ (от 64 до 127 в десятичной системе), а PR может быть равен 1, 2, 4 или 8.

Пусть, например, минимальное и максимальное допустимое время выполнения общего цикла прикладной программы МК равны 25 и 100 мс соответственно, а $f_{PCLK} = 8$ МГц.

1. Поскольку погрешность задания значений t_{min} и t_{max} равна $\pm (4096 \times PR)/f_{PCLK}$, рационально выбрать наименьшее значение коэффициента PR , обеспечивающее требуемую длительность интервала t_{max} при значении RLV , не превышающем максимально допустимое ($0x7F$). Из выражения (5.3) получаем, что данное требование удовлетворяется при PR , равном 4-м; период тактовых импульсов счетчика при этом равен 2,048 мс.

2. Рассчитываем значение RLV в соответствии с выражением (5.3) при PR , равном 4-м и $f_{PCLK} = 8$ МГц. Естественно, оно не может быть дробным, поэтому при расчетах округляем его до ближайшего большего целого (для обеспечения некоторого запаса по времени до генерации сигнала сброса). Получаем значение RLV , равное $0x70$ (112 в десятичной системе). Реальная длительность интервала t_{max} при этом составит 100,352 мс.

3. По выражению (5.2) рассчитываем верхнее пороговое значение содержимого счетчика ($W[6:0]$), обеспечивающее требуемую длительность интервала t_{min} при выбранных значениях PR и RLV . Оно также не может быть дробным; его округление рационально производить также до ближайшего большего целого, для предохранения от генерации сигнала сброса при времени выполнения цикла, не существенно меньшем минимального. Получаем $W[6:0] = 0x65$ (101 в десятичной системе); реальное значение t_{min} при этом будет равно 24,576 мс.

Оконный сторожевой таймер, в отличие от независимого, может быть источником запроса на прерывание (если установлен бит EWI в регистре конфигурации, см. рис. 5.12). Запрос вырабатывается по достижении содержимым счетчика значения $0x40$, т. е. за один такт до сброса. Данный запрос является своего рода «предупреждением» о предстоящем сбросе. Он должен быть обработан до сброса, т. е. в течение интервала времени, меньшего, чем один период тактового импульса счетчика, равный $(4096 \times PR)/f_{PCLK}$. Это требование вполне реализуемо, т. к. длительность этого периода равна минимум 4096-ти тактам ЦП. Типовой вариант подпрограммы обработки данного прерывания выполняет перезагрузку счетчика оконного сторожевого таймера, во избежание сброса МК. По выходе из подпрограммы ПО МК должно выработать сообщение о возникшей нештатной ситуации, а также, желательно, определить ее причину.

5.5 Сброс МК, инициируемый ПО

Реализуется в ряде семейств МК класса «*high performance*», в частности, в семействе *ARM Cortex-Mx*. В данном семействе инициируется установкой бита *SYSRESETREQ* в регистре управления прерываниями и сбросом (*Application Interrupt and Reset Control Register*), входящем в состав модуля памяти системного назначения (см. пункт 2.5.8).

5.6 Сброс МК по инициализации входа в энергосберегающие режимы и по выходу из них

Возможен не во всех семействах МК; в частности, реализуется в МК семейства *ARM Cortex-Mx*. В данном семействе [9, 13 – 15]:

- автоматический сброс МК происходит при выходе из энергосберегающего режима «Ожидание» (*Standby*) (см. табл. 3.2);
- может быть разрешен сброс МК по выполнении МК последовательности операций по переходу в ЭСР *Standby* и *Stop* (см. табл. 3.2); при этом вырабатывается сигнал *Low-power management reset* и, вместо перехода в ЭСР, происходит сброс МК (см. рис. 5.1).

Разрешение *Low-power management reset* осуществляется установкой в единичное состояние бита *nRST_STDBY* или, соответственно, *nRST_STOP* системного РСФ, называемого (в зависимости от конкретного подсемейства МК) *Option bytes* или *User and read protection option byte*. Детали реализации данного варианта сброса различны у различных моделей МК, и излагаются в технической документации на них.

5.7 Сброс МК через отладочный интерфейс

Реализован, в частности, в МК подсемейства *Atmega*, в которых возможен сброс через отладочный интерфейс *JTAG* (см. рис. 5.2). Производится командой *AVR_RESET*, поддерживаемой данным интерфейсом. Подробности реализации данной разновидности сброса изложены, например, в [8].

5.8 Формирование внутреннего сигнала сброса МК

Сигналы сброса, генерируемые всеми его источниками, объединяются по логическому ИЛИ (см. рис. 5.1 и 5.2) и поступают на блок формирования внутреннего сигнала сброса, с длительностью, гарантированно достаточной для установления напряжения питания и входа в рабочий режим всех ГТИ.

В МК семейства *ARM Cortex-Mx* (см. рис. 5.1) блок формирования внутреннего сигнала сброса (активный уровень - нулевой) включает в себя:

- одновибратор, генерирующий импульс с длительностью не менее 20 мкс по поступлении сигнала сброса от любого из его источников;
- инвертор выходного сигнала одновибратора, состоящий из управляемого им ключа на МОП-транзисторе и нагрузочного подтягивающего резистора;
- триггер Шмидта (пороговый элемент с гистерезисом [22]) и фильтр, в совокупности обеспечивающие помехоустойчивость формирования сигнала сброса (особенно от внешнего источника).

После сброса подсистема синхронизации МК семейства *ARM Cortex-Mx* автоматически конфигурируется под тактирование от внутреннего высокочастотного *RC*-ГТИ (*HSI*). Поэтому длительность их внутреннего импульса сброса постоянна (в отличие от МК семейства *AVR*, см. далее), и достаточна для установления рабочего режима именно *HSI*.

Блок формирования внутреннего сигнала сброса МК семейства *AVR*, в том числе подсемейства *ATmega* (см. рис. 5.2) состоит из счетчика задержки и *RS*-триггера, выход которого является выходом блока в целом. Активное состояние внутреннего сигнала сброса – единичное. Установка выхода *RS*-триггера и, соответственно, сигнала сброса в единичное состояние происходит по наступлению какого – либо из событий, вызывающих сброс, в том числе:

- по достижении напряжением питания порогового уровня (см. рис. 5.3);
- по достижении порогового уровня сигналом на входе внешнего сброса (см. рис. 5.5);
- по снижении напряжения питания ниже допустимого уровня при сбое по питанию (см. рис. 3.7);

- по переднему фронту импульса переполнения сторожевого таймера (см. рис. 5.8).

По переходу в неактивное состояние сигнала сброса от вызвавшего его источника, в том числе:

- по превышении напряжением питания порогового уровня (см. рис. 5.3);

- по переходу в единичное состояние сигнала на входе внешнего сброса (см. рис. 5.3 и 5.5);

- по возврату напряжения питания к нормальному значению после сбоя по питанию (см. рис. 3.7);

- после переполнения сторожевого таймера (см. рис. 5.8);

происходит сброс счетчика задержки, и начинается отсчет интервала времени t_{TOUT} , минимально необходимого для установления напряжения питания и входа источника тактирования МК в рабочий режим (см. пункт 4.2.6). По окончании интервала t_{TOUT} , т. е. по переполнении счетчика задержки, происходит сброс RS-триггера и, соответственно, обнуление внутреннего сигнала сброса МК (см. рис. 3.7, 5.3, 5.5 и 5.8).

В отличие от МК семейства *ARM Cortex-Mx*, источник тактирования МК семейства *AVR* задается младшим конфигурационным байтом, доступным для записи только в режиме программирования энергонезависимой памяти (см. пункт 4.4.1). Соответственно, длительность интервала t_{TOUT} должна быть достаточна для установления рабочего режима именно выбранного источника тактирования. Поэтому, она не постоянна, как минимальная длительность внутреннего импульса МК семейства *ARM Cortex-Mx*, а задается битами *CSEL[3:0]* и *SUT[1:0]* младшего конфигурационного байта (см. пункт 4.4.1).

5.9 Индивидуальный сброс функциональных блоков МК

Архитектура ряда семейств МК, в т. ч. *ARM Cortex-Mx*, предоставляет возможность индивидуального сброса функциональных блоков МК, т. е. установки в состояние «по умолчанию» (*Reset Value*) всех компонентов блока, в т. ч. его регистров, без общего сброса МК [13 – 15].

В МК семейства *ARM Cortex-Mx* данная операция осуществляется посредством входящих в подсистему сброса и синхронизации (*RCC*) регистров сброса периферийных устройств. В

качестве примера на рис. 5.14 представлен формат регистра сброса устройств домена *AHB* (см. подраздел 1.3) МК модельного ряда *STM32F030xx*, относящихся подсемейству *ARM Cortex-M0*.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOPF RST	Res.	IOPD RST	IOPC RST	IOPB RST	IOPA RST	Res.
									rw		rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Рис. 5.14. Формат регистра *AHB peripheral reset register* (*RCC_AHBRSTR*) МК модельного ряда *STM32F030xx* [15]

Сброс функционального блока МК осуществляется записью единицы в соответствующий ему разряд регистра сброса. Например, сброс ПБВ *A* МК модельного ряда *STM32F030xx* производится установкой в единичное состояние 17-го бита (*IOPARST*) регистра *RCC_AHBRSTR* (см. рис. 5.14), со сбросом (после временной задержки) данного бита в ноль. Данная процедура может быть реализована следующим программным фрагментом *CMSIS* -уровня (см. также подпункт 2.6.1.14 и Приложение В):

```
//Установка в единицу бита сброса ПБВ A
RCC->AHBRSTR |= 0x00020000;
```

```
//Временная задержка
for(i=0;i<1000;i++);
```

```
//Сброс в ноль всех битов регистра RCC_AHBRSTR
RCC->AHBRSTR = 0x00000000;
```

5.10 Выводы по разделу 5

5.10.1. Подсистема сброса, наряду с подсистемами питания и синхронизации и с портами ввода / вывода, входит в состав функциональных блоков / подсистем МК, минимально необходимых для его практического применения. Ее основное назначение – установка всех энергозависимых ЗУ МК (в 1-ю очередь – РОН и РСФ) в некоторые определенные и предсказуемые состояния при включении питания, при штатной перезагрузке МК или при

возникновении нештатных ситуаций, например, сбоях по питанию, невыполнении условий выхода из цикла и т. п.

5.10.2. Состояния, в которые устанавливаются регистры МК при сбросе, называются *Reset value*, *Initial value* или *Default value*, и всегда приводятся в технической документации на МК.

5.10.3. Типовая подсистема сброса МК включает в себя следующие основные функциональные узлы (см. рис. 5.1 и 5.2):

- источники сигналов сброса;
- схему объединения сигналов сброса по логическому ИЛИ;
- блок формирования внутреннего (системного) сигнала сброса.

5.10.4. В абсолютном большинстве семейств МК всех классов основными событиями, вызывающими сброс, являются:

- включение питания;
- активный уровень входа внешнего сброса, как правило, формируемый механическим переключателем;
- окончание интервала времени ожидания, формируемого сторожевым таймером.

Кроме вышеперечисленных, в большинстве семейств МК классов «*mainstream*» и «*high performance*» существует ряд дополнительных источников сброса:

- сброс по уменьшению напряжения питания ниже допустимого уровня (*Brown Out Reset*, *BOR*);
- сброс от оконного сторожевого таймера, при времени ожидания некоторого события, как большем некоторого максимально допустимого, так и меньшем заданного минимального;
- сброс при выходе из энергосберегающего режима (ЭСР), а также по выполнении МК последовательности операций по переходу в ЭСР;
- сброс, инициируемый ПО (*Software reset*);
- сброс через отладочный интерфейс.

5.10.5. Сигналы сброса от источников, имеющих в составе конкретной модели МК, объединяются по логическому ИЛИ (см. рис. 5.1 и 5.2) и поступают на формирователь внутреннего импульса сброса МК. Основное назначение данного формирователя – генерация импульса сброса с длительностью, достаточной для установления напряжения питания МК, а также для входа ГТИ, служащего источником синхронизации МК, в рабочий режим (см. рис. 3.7, 5.1, 5.3, 5.5 и 5.8).

5.10.6. С точки зрения архитектуры МК процедура сброса является прерыванием (см. раздел 7). Прерывание по сбросу не может быть запрещено (замаскировано). В ряде моделей МК класса «*cost-sensitive*», в частности, во многих моделях *PIC*-МК младшего подсемейства, сброс является единственным источником прерывания.

5.10.7. При каждом сбросе МК в некотором РСФ (например, в *MCUCSR* МК подсемейства *ATmega*, *RCC_CSR* МК семейства *ARM Cortex-Mx*) фиксируется информация о факте сброса и об его источнике. Биты – индикаторы сброса и его источников доступны для чтения и для обнуления прикладным ПО. Фиксация событий сброса и его источников позволяет прикладному ПО, в частности, определять причины нештатных ситуаций и выдавать сообщения о них.

5.10.8. Примеры структурно-архитектурных решений функциональных узлов и блоков подсистемы сброса описаны в подразделах 5.2 – 5.9. В целом, они являются типовыми для большинства семейств МК общего назначения.