

## **3 ТИПОВЫЕ СТРУКТУРНО-АРХИТЕКТУРНЫЕ РЕШЕНИЯ ПОДСИСТЕМЫ ПИТАНИЯ МК ОБЩЕГО НАЗНАЧЕНИЯ**

<b>3.1 Типовой состав напряжений питания МК. Схемы подключения питания.....</b>	<b>2</b>
<b>3.2 Согласование внешних устройств с МК по уровням сигналов.....</b>	<b>9</b>
<b>3.3 Запуск МК при включении питания. Восстановления работы МК при сбоях по питанию .....</b>	<b>11</b>
<b>3.3.1. Общий сброс МК при включении питания и при сбоях по питанию .....</b>	<b>12</b>
<b>3.3.2. Применение энергонезависимой памяти для восстановления работоспособности МК при сбоях по питанию .....</b>	<b>18</b>
<b>3.3.3. Домен с резервированием питания (ДРП) .....</b>	<b>18</b>
<b>3.4 Минимизация энергопотребления МК. Энергосберегающие режимы работы.....</b>	<b>33</b>
<b>3.4.1. Общие принципы реализации энергосберегающих режимов ...</b>	<b>33</b>
<b>3.4.2. Минимизация энергопотребления и энергосберегающие режимы МК семейства AVR.....</b>	<b>37</b>
<b>3.4.3. Минимизация энергопотребления и энергосберегающие режимы МК семейства ARM Cortex-Mx.....</b>	<b>41</b>
<b>3.5 Выводы по разделу 3.....</b>	<b>49</b>

Как указано в подразделе 1.2, подсистема питания входит в состав функциональных блоков / подсистем МК, минимально необходимых для его практического применения (что очевидно).

Основными элементами архитектуры подсистемы питания МК являются:

- состав напряжений питания и схемы их подключения;
- допустимые уровни входных сигналов МК в зависимости от напряжений питания;
- реализация корректного запуска МК при включении питания и восстановления работы МК при сбоях по питанию, в том числе после его отключения (аварийного или преднамеренного);
- способы минимизации энергопотребления и реализация энергосберегающих режимов работы МК.

Данные архитектурные элементы подсистемы питания подлежат рассмотрению в настоящем разделе.

### **3.1 Типовой состав напряжений питания МК. Схемы подключения питания**

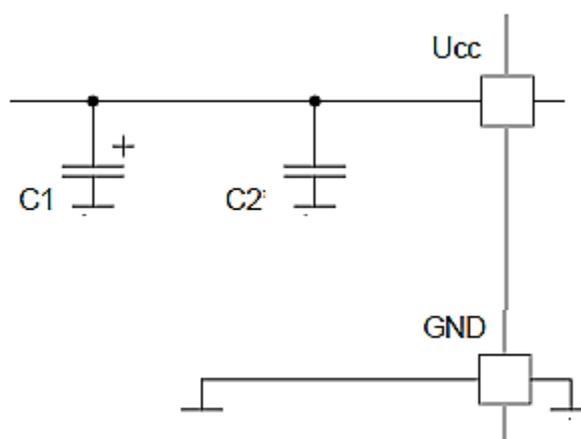
Состав напряжений питания МК определяется следующими основными факторами:

- классом, к которому принадлежит МК;
- наличием или отсутствием аналоговых / аналого-цифровых функциональных блоков в составе МК;
- производительностью ЦП;
- структурой и архитектурой подсистемы программирования памяти.

**Наиболее простым** является состав напряжений питания моделей МК класса «*cost-sensitive*», в структуре которых отсутствуют аналоговые или аналого-цифровые функциональные узлы и блоки (см., например, рис. 1.1 и 1.2). У таких МК имеется только одно напряжение питания, обычно обозначаемое  $U_{CC}, V_{CC}$  или  $V_{DD}$ , и подключаемое относительно общего вывода МК, обозначаемого  $V_{SS}, 0V$  или  $GND$ . **Примечание.** Последнее обозначение происходит от традиционно принятого английского термина *GROUND* – «Земля», хотя реально этот вывод не заземлен и его заземление **недопустимо**; он подключается к нулевому выводу источника питания (общей шине).

Специфических особенностей подключения при этом не имеется. Необходимо только:

- обеспечение допустимых пределов напряжения питания, оговариваемых технической документацией на МК;
- подключение блокировочных конденсаторов между выводом питания и общей шиной (см. рис. 3.1), осуществляющих фильтрацию (сглаживание) бросков и помех по питанию [20]; конденсаторы должны устанавливаться на минимально возможном расстоянии от вывода питания и общего вывода МК, а их номиналы оговариваются документацией на МК (см., например, [11]).

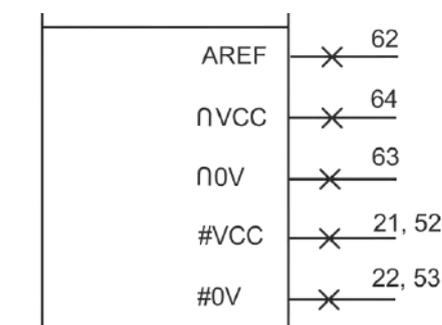


**Рис. 3.1.** Пример подключения блокировочных конденсаторов по питанию [11]

Существенно отличается от рассмотренного выше состав напряжений питания МК классов «*cost-sensitive*» и «*mainstream*», в структуру которых входят **аналоговые или / и аналого-цифровые** функциональные узлы и блоки. Типовым примером такого МК является 1887BE7T, аналог МК *ATmega128* (см. рис. 1.3). Фрагмент цоколевки («распиновки») БИС этого МК, содержащий выводы питания, представлен на рис. 3.2.

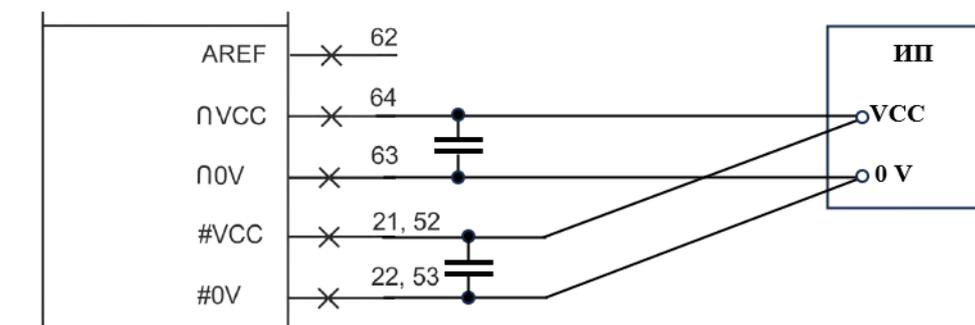
Основной отличительной особенностью подсистемы питания данной категории МК является наличие двух отдельных общих шин и двух отдельных шин питания – аналоговой и цифровой (см. рис. 3.2), как правило – при **одинаковых значениях** напряжений питания аналоговых и цифровых функциональных блоков и при **одном и том же источнике** их питания. Такое техническое решение широко распространено в системах питания устройств, содержащих как аналоговую, так и цифровую часть. Его целью является

минимизация влияния помех, возникающих в шине питания и общей шине цифровой части при переключениях ее компонентов, на чувствительную аналоговую часть (подробнее – см. [20]). При этом подключение выводов питания и общих выводов аналоговой и цифровой части к источнику питания осуществляется отдельными проводниками, соединяемыми между собой только на соответствующих выводах источника питания (см. рис. 3.3). Блокировочные конденсаторы обязательны, и должны подключаться на минимально возможном расстоянии от вывода питания и общего вывода МК (см. также рис. 3.1 и пояснения к нему).



A0V – общий вывод аналоговой части  
 #0V – общий вывод цифровой части  
 AVCC – вывод питания аналоговой части  
 #VCC – вывод питания цифровой части  
 AREF – вывод опорного напряжения АЦП

**Рис. 3.2.** Фрагмент цоколевки БИС МК 1887BE7Т [8]

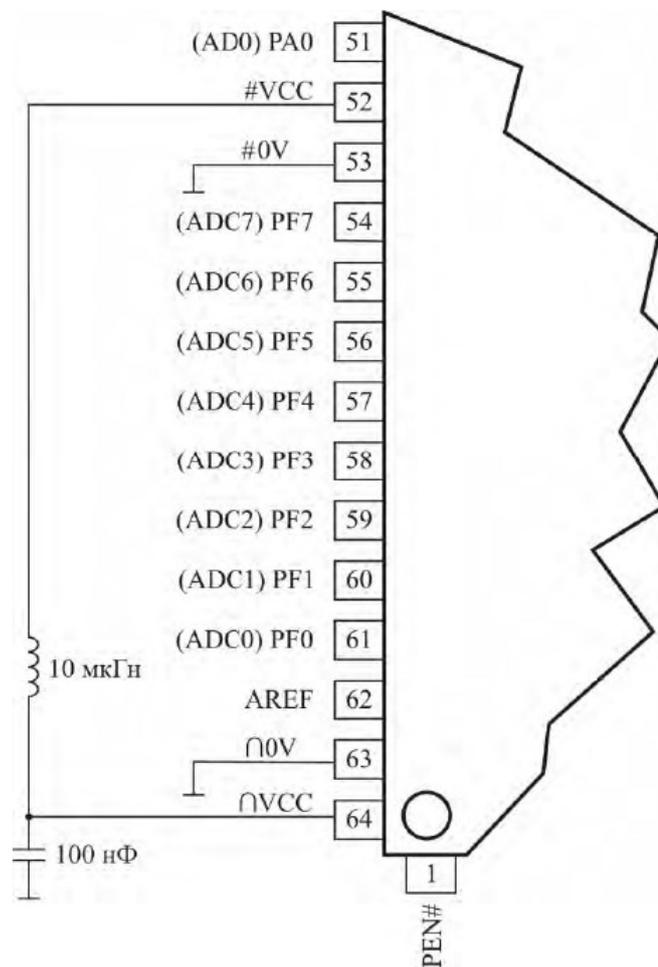


ИП – источник питания

**Рис. 3.3.** Раздельное подключение аналоговой и цифровой части МК к источнику питания (на примере БИС 1887BE7Т)

Другим возможным вариантом подключения питания к БИС МК, содержащей как аналоговые, так и цифровые функциональные

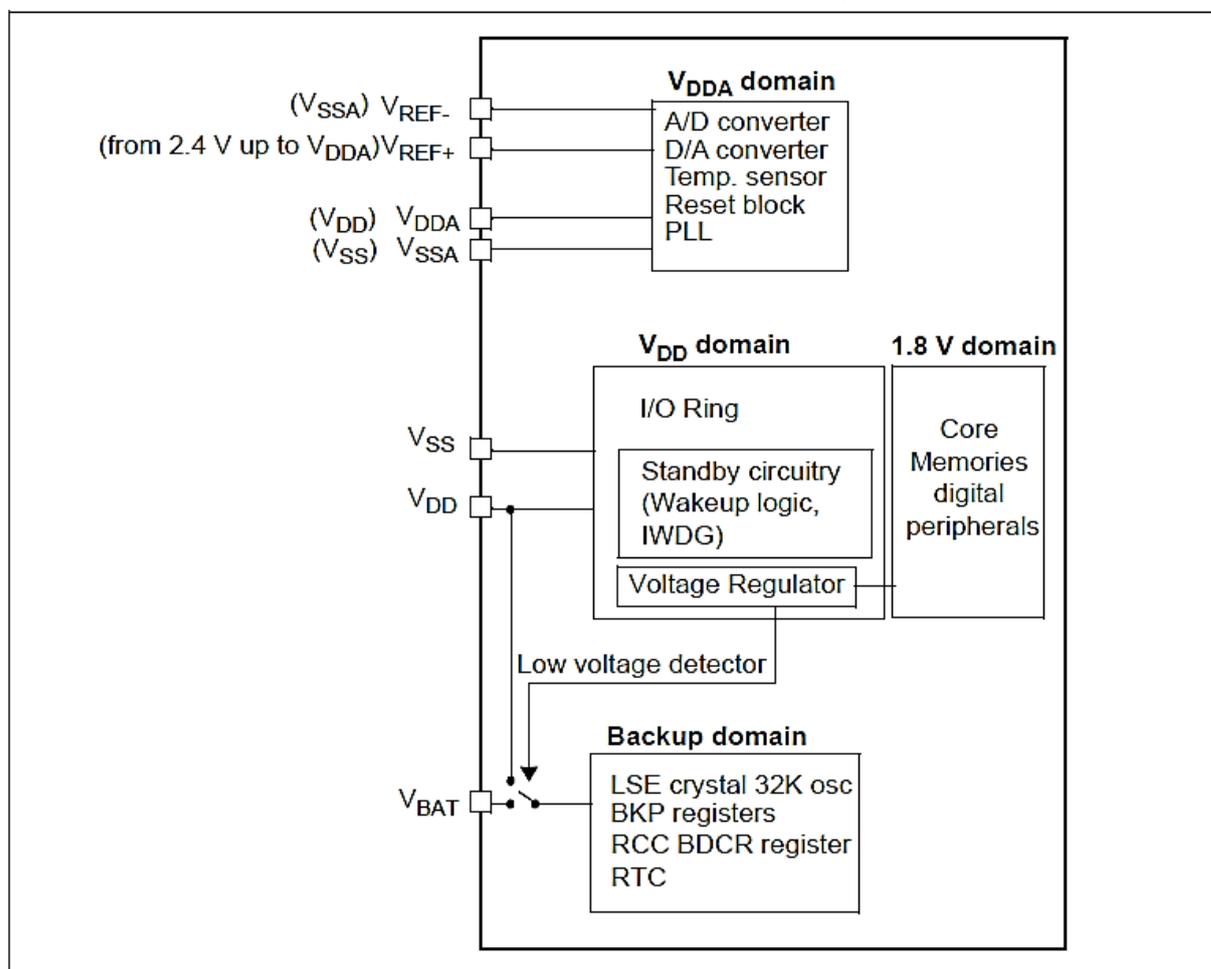
блоки, является схема, рекомендуемая технической документацией на БИС 1887BE7T [8] и представленная на рис. 3.4. Аналогичные схемы рекомендуются и для других МК подсемейства *ATmega* [6]. В данной схеме отдельное подключение выводов питания аналоговой и цифровой части МК не используется, а минимизация влияния помех, создаваемых цифровой частью, на аналоговую осуществляется *LC*-фильтром нижних частот (см. рис. 3.4). При этом отдельное подключение общих выводов аналоговой и цифровой части к источнику питания, а также включение блокировочного конденсатора между выводами #0V и #VCC (на схеме не показан) остается необходимым.



**Рис. 3.4.** Схема подключения питания, рекомендуемая технической документацией на БИС МК 1887BE7T [8]

По мнению автора, приведенная на рис. 3.4 схема менее предпочтительна, чем представленная на рис. 3.3, т. к. требует дополнительных компонентов, а степень подавления помех зависит от конкретных значений индуктивности дросселя и емкости конденсатора *LC*-фильтра.

Наиболее сложной является подсистема питания МК класса «*high-performance*», в частности, семейства *ARM Cortex-Mx*. На рис. 3.5 в качестве типового примера представлена структурная схема подсистемы питания МК модельного ряда *STM32F10xxx* [13], принадлежащего к подсемейству *ARM Cortex-M3*.



**Рис. 3.5.** Структурная схема подсистемы питания МК семейства *ARM Cortex-Mx* (на примере модельного ряда *STM32F10xxx* [13]).  
См. пояснения в тексте

На рис. 3.5:  
 $V_{SSA}$  – общий вывод аналоговой части МК;  
 $V_{DDA}$  – напряжение питания аналоговой части МК;  
 $V_{REF+}$ ,  $V_{REF-}$  – выводы подключения опорного напряжения АЦП и ЦАП ( $V_{REF-}$  соединяется с  $V_{SSA}$ );  
 $V_{SS}$  – общий вывод цифровой части МК;  
 $V_{DD}$  – основное напряжение питания цифровой части МК;  
 $V_{BAT}$  – напряжение резервного (батарейного) питания;

*V<sub>DDA</sub> domain* – аналоговая часть МК;  
*V<sub>DD</sub> domain* – домен цифровой части МК, питаемый напряжением *V<sub>DD</sub>*;  
*1.8 V domain* – домен цифровой части МК, питаемый напряжением 1,8 В;  
*Backup domain* – домен с резервированием питания (ДРП);  
*A/D converter* и *D/A converter* – АЦП и ЦАП соответственно;  
*Temp. sensor* – датчик температуры;  
*Reset block* – блок сброса;  
*PLL (Phase Locked Loop)* – блок фазовой автоподстройки частоты (ФАПЧ);  
*I/O Ring* – блоки ввода / вывода;  
*Standby circuitry (Wakeup logic, IWDG)* – блоки ожидания (имеется в виду – ожидания выхода из энергосберегающего режима, см. пункт 3.4.3);  
*Wakeup logic* – логика «пробуждения» (см. пункт 3.4.3);  
*IWDT* – независимый сторожевой таймер МК (см. пункт 5.4.4);  
*Voltage Regulator* – понижающий стабилизатор напряжения;  
*Low Voltage Detector* – выходной сигнал детектора понижения основного напряжения питания;  
*Core* – «ядро» (ЦП);  
*Memories* – резидентная память;  
*Digital peripherals* – цифровые ПУ;  
*LSE crystal 32K osc* – генератор тактовых импульсов частотой 32,768 кГц с внешним пьезоэлектрическим резонатором, ПЭР (*Low Speed External, LSE*, см. пункт 4.2.3);  
*BKP registers* – регистры для сохранения информации, не подлежащей потере при сбое / отказе основного питания (*Backup Data Registers*);  
*RCC BDCR register* – регистр управления ДРП;  
*RTC* – «часы» (корректнее – счетчик) реального времени (*Real Time Clock*).

Основными **особенностями** подсистемы питания МК семейства *ARM Cortex-Mx* являются (см. рис. 3.5):

- питание цифрового ядра МК (в состав которого входят ЦП, модули памяти и цифровые периферийные устройства, за исключением узлов сопряжения с внешними выводами МК) напряжением, пониженным по сравнению с основным;

- наличие так называемого домена с резервированием питания (ДРП);

- как и в МК ранее рассмотренной категории – отдельные общие шины и шины питания аналоговой и цифровой части (как правило, при **одинаковых значениях** и при **одном том же источнике напряжений** их питания).

Питание цифрового ядра МК пониженным напряжением (1,8 В при допустимом значении основного напряжения питания цифровой части, находящемся в пределах от 2.0 до 3,6 В [10]) позволяет существенно снизить потребляемую мощность и уровень помех создаваемых цифровым ядром, которое включает в себя основные энергопотребляющие узлы и блоки МК, являющиеся при этом и основными источниками помех. Напряжение питания цифрового ядра вырабатывается понижающим стабилизатором, обозначенным на рис. 3.5 как *Voltage Regulator*.

ДРП представляет собой группу узлов и блоков МК, функционирование которых необходимо для поддержания работоспособности МК в любом из энергосберегающих режимов (см. пункт 3.4.3), а также при отключении (аварийном или преднамеренном) основного источника питания. В их состав входят (см. рис. 3.5 и пояснения к нему): генератор тактовых импульсов частотой 32 кГц; регистры для сохранения данных, не подлежащих потере при сбое / отказе основного питания; регистр управления ДРП; счетчик реального времени; а также ряд функциональных узлов, не показанных на рис. 3.5.

При питании МК от основного источника работа функциональных узлов ДРП не прекращается ни в каком из энергосберегающих режимов, в том числе в режиме минимального энергопотребления (*Standby*) (см. пункт 3.4.3).

Если существует вероятность аварийного отключения основного источника питания или планируется намеренное его отключение, необходимо подать питание от резервного источника (батареи) на вывод  $V_{BAT}$  МК. При уменьшении основного напряжения питания ниже предельно допустимого срабатывает блок переключения батарейного питания (см. рис. 3.5), отключая шину питания батарейного домена от выхода основного источника и соединяя ее с выходом резервного.

Более подробное описание ДРП приведено в пункте 3.3.3.

Использование резервного источника питания является опциональным [13]. Если разработчик не считает его необходимым, вывод  $V_{BAT}$  должен быть объединен с выводом  $V_{DD}$ , с подключением блокировочного конденсатора емкостью 100 нФ к объединенным выводам.

Шины питания и общие шины аналоговых и цифровых функциональных блоков МК семейства *ARM Cortex-Mx* являются отдельными. При этом у ряда моделей МК, например, K1986VE92F11 [11], предусмотрены отдельные шины и выводы для подачи питания на блоки АЦП и ЦАП (более чувствительные к помехам) и на блоки компаратора и ФАПЧ (см. рис. 3.5). Подключение питания к аналоговой и к цифровой части МК (если это предусмотрено – также к более и к менее чувствительным узлам и блокам аналоговой части) должно осуществляться отдельными парами проводников, объединяемыми только на соответствующих выводах источника питания, по аналогии с рис. 3.3. Необходимо также применение блокировочных конденсаторов (см. рис. 14 в [10]), устанавливаемых на минимально возможном расстоянии от корпуса БИС МК.

Следует также отметить, что ряд подсемейств / моделей МК требует применения **дополнительных источников** напряжения (как правило, более высокого, чем напряжение питания МК) при программировании энергонезависимых модулей памяти. Например, МК подсемейства *ATmega* семейства *AVR*, в том числе 1887VE7T, в параллельном режиме программирования памяти требуют подключения дополнительного напряжения плюс 12 В к выводу *RESET* (кроме «штатных» напряжений питания  $\cap V_{CC}$  и  $\#V_{CC}$ , см. рис. 3.2) [8]. Более подробно данные вопросы рассмотрены в разделе 12, посвященном программированию памяти МК.

### **3.2 Согласование внешних устройств с МК по уровням сигналов**

Весьма важными параметрами при практическом применении МК являются возможные диапазоны выходных напряжений, а также допустимые пределы входных напряжений БИС МК. Очевидно, они определяются напряжением питания, а также схемотехникой портов ввода / вывода МК.

Практически все современные семейства МК общего назначения реализуются по технологии КМОП [20], для которой характерны следующие основные требования к входным напряжениям и следующие основные характеристики выходных.

**3.2.1.** Устойчивой тенденцией развития схемотехники и технологии МК является снижение напряжения питания [20]. В настоящее время его номинальное значение у большинства серийно выпускаемых МК равно 3 или 3,3 В; несколько менее распространены МК с напряжением питания, равным 5 В; выпускаются и МК с напряжением питания 1,6 В (см., например, [21]).

**3.2.2.** Допустимый диапазон входных напряжений КМОП-устройств, в том числе МК – от нуля до напряжения питания соответствующего функционального блока [20]. Выход напряжения за пределы данного диапазона может привести к необратимой потере работоспособности или деградации параметров данного блока. Исключение составляют имеющиеся у ряда семейств МК с номинальным напряжением питания, равным 3 или 3,3 В (в частности, *ARM Cortex-Mx*) выводы со статусом «*5 V tolerant I/O*». Максимально допустимый уровень входного напряжения на данных выводах равен 5 В, за счет применения специальных схем согласования с низковольтными внутренними узлами и блоками МК. Благодаря этому на выводы с указанным статусом могут подаваться выходные сигналы пока еще достаточно распространенных ИС и БИС с напряжением питания, равным 5 В. При этом минимально допустимое входное напряжение также равно нулю.

**3.2.3.** Из пункта 3.2.2 следует, что подача входного напряжения на КМОП-устройства, в том числе на БИС МК при отключенном питании **недопустима**.

**3.2.4.** При использовании вывода КМОП-МК в качестве цифрового входа, уровни напряжения от нуля до примерно  $U_{cc}/3$  воспринимаются как логический ноль, а от примерно  $2U_{cc}/3$  до  $U_{cc}$  – как логическая единица [20]. Конкретные значения верхнего предела напряжения логического нуля и нижнего предела напряжения логической единицы могут отличаться от вышеприведенных и указываются в технической документации на соответствующие модели МК.

**3.2.5.** Максимально допустимое значение опорного напряжения встроенных АЦП и ЦАП МК равно напряжению питания аналоговой части МК.

**3.2.6.** При использовании вывода КМОП-МК в качестве цифрового выхода, оговариваемые технической документацией на МК максимальное напряжение логического нуля и минимальное напряжение логической единицы равны примерно  $0,1V_{DD}$  и  $0,9V_{DD}$  соответственно [10]. На практике, при отсутствии перегрузки выхода, напряжения логического нуля и логической единицы примерно равны нулю и  $V_{DD}$  соответственно.

### **3.3 Запуск МК при включении питания. Восстановления работы МК при сбоях по питанию**

Для практического применения БИС МК весьма важно соблюдение следующих требований:

- корректного запуска МК по включении питания, т. е. запуска прикладной программы с определенного, заранее известного и предсказуемого адреса, при известных и предсказуемых состояниях функциональных узлов и блоков МК;

- корректного перезапуска МК при выходе из энергосберегающих режимов (см. подраздел 3.4) и при сбоях по питанию (уменьшении напряжения питания ниже допустимого значения, в том числе отключении питания), т. е. перезапуска также с определенного адреса, при предсказуемых состояниях узлов и блоков МК и без потерь программного кода и наиболее важных данных.

Для удовлетворения вышеперечисленных требований в различных семействах МК применяются следующие структурно-архитектурные решения:

- автоматический общий сброс МК при включении питания (*Power-On Reset, POR*) и при уменьшении напряжения питания ниже минимально допустимого уровня, называемый *Power-Down Reset, (PDR)* или *Brown Out Reset (BOR)*; у ряда семейств / подсемейств МК, например, *ARM* и *Cortex-M4*, существуют несколько различающиеся между собой функции *PDR* и *BOR* (см. далее);

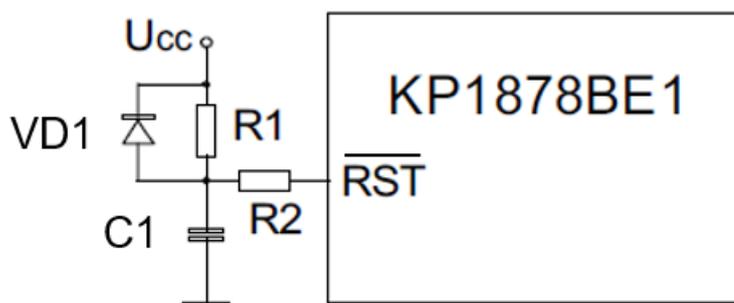
- использование энергонезависимых (и / или питаемых от резервных источников) модулей памяти МК для хранения кодов программ, а также не подлежащих потере данных;

- автоматический перевод группы «жизненно важных» функциональных узлов МК на питание от резервного источника при отключении основного источника питания или при уменьшении его напряжения до уровня ниже минимально допустимого (см. рис. 3.5).

### 3.3.1. Общий сброс МК при включении питания и при сбоях по питанию

Общий сброс является рациональным средством корректного запуска / перезапуска МК. При нем МК переходит к подпрограмме обслуживания прерывания по сбросу, начальный адрес которой указывается в технической документации на МК и по которому необходимо расположить начальную команду прикладной программы МК. Все функциональные узлы МК при сбросе устанавливаются в определенные состояния, также оговариваемые документацией на МК.

Автоматический общий сброс МК по включении питания характерен для большинства современных МК, за исключением некоторых моделей класса «*cost-sensitive*», в частности, КР1878ВЕ1 [5]. Такие МК должны снабжаться цепью автоматического сброса по включении питания, типовой пример которой представлен на рис. 3.6 [5].



**Рис. 3.6.** Пример внешней схемы сброса МК по включении питания [5]

Принцип работы данной цепи следующий. По включении питания напряжение на входе сброса МК ( $\overline{RST}$ ) достигнет максимального значения, воспринимаемого МК как логический ноль ( $U_{cc}/3$ ), только через интервал времени с длительностью, примерно равной  $0,4R1C1$ . Следовательно, в течение данного интервала вход  $\overline{RST}$  МК будет находиться в активном (нулевом) состоянии, что

обеспечит общий сброс МК по включении питания. Резистор  $R2$  защищает вход МК по току. Диод  $VD1$ , называемый форсирующим диодом, служит для быстрого разряда конденсатора  $C1$  при выключении питания, что предохраняет МК от подачи напряжения на его вход сброса в отсутствие питания (см. также пункт 3.2.3). Типовые сопротивления резисторов, при условии, что МК реализован по КМОП-технологии:  $R1$  - порядка 36 – 43 кОм,  $R2$  – порядка 1 кОм. Емкость конденсатора  $C1$  выбирается, исходя из условия  $0,4R1C1 > t_{RST min}$ , где  $t_{RST min}$  – нормируемая технической документацией на МК минимальная длительность импульса сброса.

Следует отметить, что таков же принцип работы и внутренних схем автоматизированного сброса МК по включении питания.

Общий сброс **при сбое по питанию** реализован практически во всех семействах / подсемействах МК классов «*mainstream*» и «*high performance*». В ряде подсемейств / моделей класса «*cost-sensitive*» отсутствует, но может быть осуществлен посредством внешних цепей (см., например, [4]).

Принцип реализации данной функции, в целом, одинаков у всех семейств МК. Основой блока сброса при сбое по питанию является аналоговый компаратор напряжений. Он осуществляет непрерывное сравнение напряжения питания с пороговым уровнем, который, в зависимости от семейства / подсемейства, к которому принадлежит МК, может быть:

- фиксированным (например, у МК семейства *ARM Cortex-Mx* при выполнении функции *PDR* [13]);

- задаваемым при программировании слов конфигурации МК (например, у семейства *AVR* [8]) или так называемых *Option Bytes* (например, у МК подсемейства *ARM Cortex-M4* при выполнении функции *BOR* [14]).

При уменьшении напряжения питания ниже порогового уровня на время, превышающее максимально допустимое, сигнал сброса МК переводится в активное состояние. При восстановлении напряжения питания до уровня выше порогового происходит возврат данного сигнала в пассивное состояние. Во избежание многократных паразитных переключений выхода компаратора, уровни напряжения, при которых происходит установка сигнала сброса в активное и в пассивное состояние, разнесены между собой, т. е.

вводится гистерезис (запаздывание) срабатывания компаратора (см. далее рис. 3.7 и 3.8).

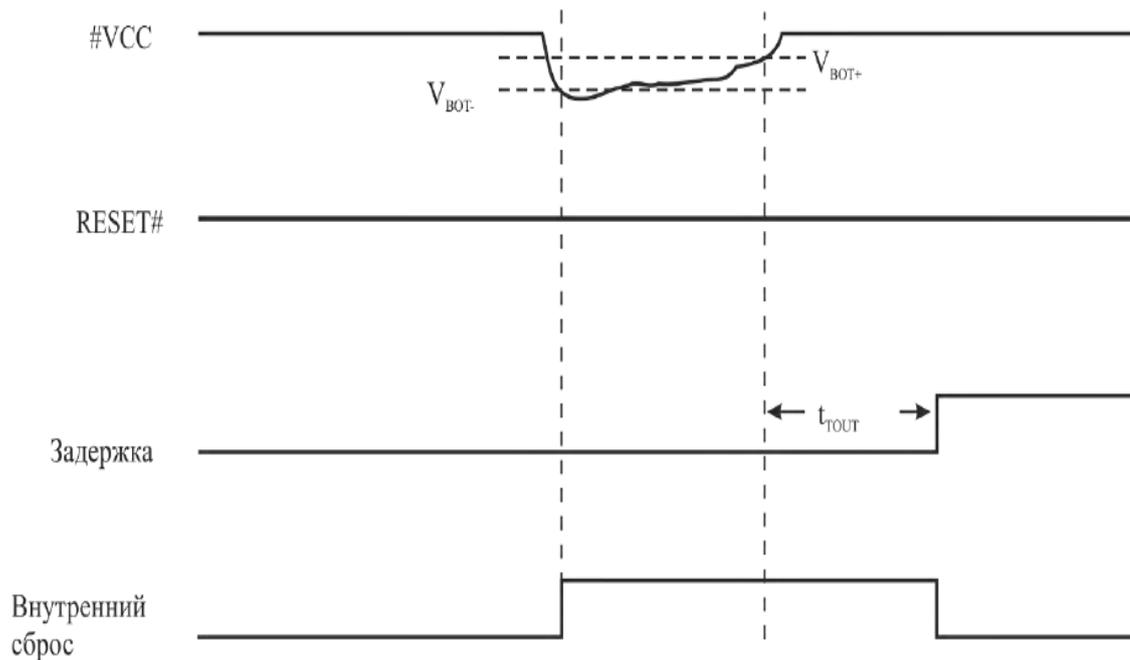
Рассмотрим подробнее принцип реализации сброса при сбое по питанию на примерах МК подсемейства *ATmega* семейства *AVR* и МК семейства *ARM Cortex-Mx*.

Временные диаграммы формирования импульса сброса МК подсемейства *ATmega* при сбое по питанию представлены на рис. 3.7 [8].

При уменьшении напряжения питания ниже уровня  $V_{BOT-}$ , на время, превышающее максимально допустимое (типовое значение которого – 2 мкс) сигнал внутреннего сброса МК переводится в активное (единичное) состояние. При восстановлении напряжения питания до уровня  $V_{BOT+}$  сигнал внутреннего сброса переводится в пассивное (нулевое) состояние, с задержкой  $t_{TOUT}$ . Значение напряжения  $V_{BOT-}$  равно  $V_{BOT} - (V_{HYST}/2)$ , а напряжение  $V_{BOT+}$  равно  $V_{BOT} + (V_{HYST}/2)$ , где  $V_{BOT}$  – порог срабатывания компаратора, а  $V_{HYST}$  – напряжение гистерезиса (разность уровней  $V_{BOT+}$  и  $V_{BOT-}$ ). Значение  $V_{BOT}$  задается битом *BODLEVEL* младшего конфигурационного байта МК, при нулевом состоянии которого данное напряжение равно  $4 \text{ В} \pm 500 \text{ мВ}$ , а при единичном –  $2,7 \text{ В} \pm 300 \text{ мВ}$ .  $V_{HYST}$  оговаривается технической документацией на МК, его типовое значение равно 100 мВ [8]. Длительность интервала  $t_{TOUT}$  равна минимальной длительности внутреннего импульса сброса МК, необходимой для установления напряжения питания МК, а также для входа генератора синхроимпульсов МК в рабочий режим (см. пункт 4.4.1).

Сброс при сбое по питанию может быть запрещен установкой в единичное состояние бита *BODEN* младшего конфигурационного байта.

Необходимо отметить, что конфигурационные байты МК семейства *AVR*, в том числе подсемейства *ATmega*, доступны для записи только в режиме программирования памяти (см. раздел 12).



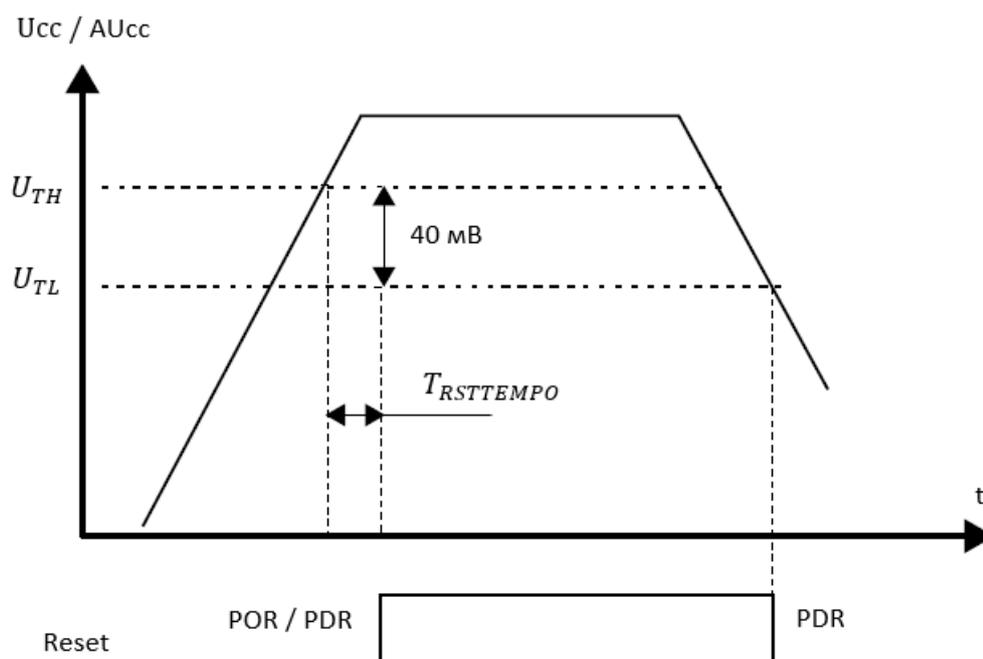
$V_{BOT-}$  и  $V_{BOT+}$  - соответственно нижний и верхний пороги срабатывания компаратора

$t_{TOUT}$  – минимально необходимая длительность внутреннего импульса сброса МК

**Рис. 3.7.** Временные диаграммы формирования импульса сброса МК подсемейства *ATmega* при сбое по питанию [8]

Во всех МК семейства *ARM Cortex-Mx* имеется функция *Power-Down Reset (PDR)*, совмещенная с функцией *Power-On Reset (POR)* и реализуемая одним и тем же функциональным узлом, называемым *POR/PDR circuitry*. Он же управляет переключением питания батарейного домена с основного источника на резервный и обратно (см. рис. 3.5). Временные диаграммы формирования сигнала сброса при реализации функции *POR/PDR* представлены на рис. 3.8.

Активное состояние сигнала сброса – нулевое. При включении напряжения питания или при его восстановлении после сбоя сигнал сброса находится в активном состоянии (соответственно *POR* или *PDR*) до достижения верхнего порогового уровня,  $U_{TH}$ . Перевод его в пассивное (единичное) состояние по достижении данного уровня происходит с задержкой (см. рис. 3.8), для гарантированного установления напряжения питания до минимально допустимого уровня.



$U_{TH}$ ,  $U_{TL}$  – соответственно верхний и нижний пороговые уровни напряжения питания

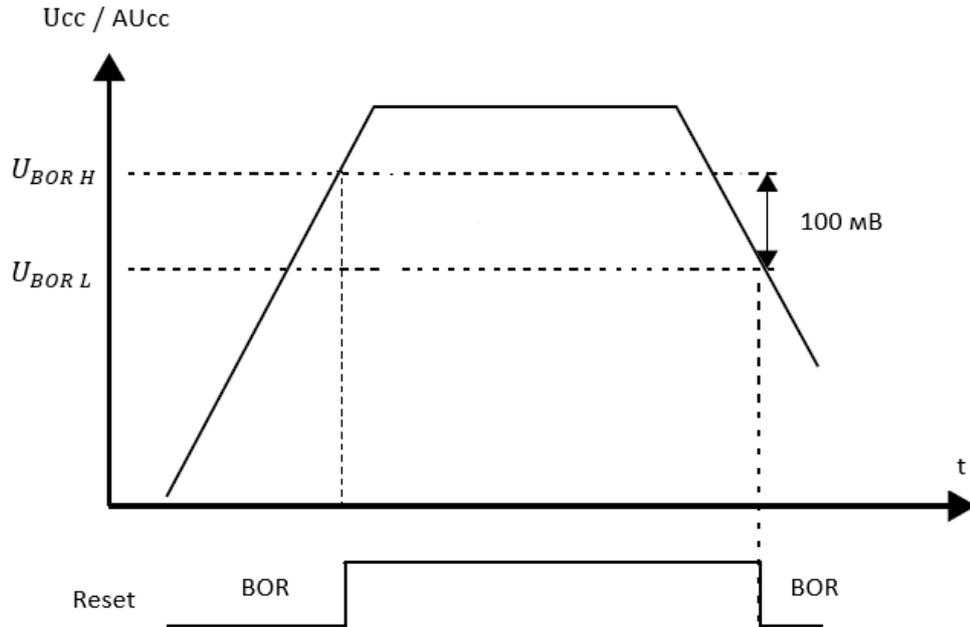
$T_{RSTTEMPO}$  – время задержки перевода сигнала сброса в пассивное состояние

**Рис. 3.8.** Временные диаграммы формирования сигнала сброса при реализации функции *POR/PDR* в МК семейства *ARM Cortex-Mx* [13]

При уменьшении напряжения питания до значения ниже уровня  $U_{TL}$  сигнал сброса практически без задержки переводится в активное состояние (*PDR*). При наличии резервного источника питания происходит подключение к нему батарейного домена, с отключением последнего от основного источника питания. Обратное переключение происходит по восстановлении напряжения основного источника питания до верхнего порогового уровня, после перевода сигнала сброса в пассивное состояние (см. рис. 3.8).

Значения напряжений  $U_{TH}$  и  $U_{TL}$ , а также время задержки  $T_{RSTTEMPO}$  нормируются технической документацией на конкретную модель МК. У большинства моделей  $U_{TH} \approx 2$  В, а  $T_{RSTTEMPO} \approx 4$  мс [10].

В ряде моделей МК семейства *ARM Cortex-Mx*, в частности, подсемейства *ARM Cortex-M4*, также реализована дополнительная функция *Brown Out Reset (BOR)*. Временные диаграммы формирования сигнала *BOR* представлены на рис. 3.9.



$U_{BORL}$  и  $U_{BORH}$  – пороговые уровни напряжения питания при установке сигнала сброса в активное и в пассивное состояние соответственно

**Рис. 3.9.** Временные диаграммы формирования сигнала сброса при реализации функции *BOR* в МК подсемейства *ARM Cortex-M4* [14]

Функция *BOR* МК семейства *ARM Cortex-Mx* отличается от *POR/PDR* следующими особенностями:

- напряжения  $U_{BORL}$  и  $U_{BORH}$  (см. рис. 3.9) не являются фиксированными, а задаются специально выделенными битами в одном из слов конфигурации (*User Option Byte*); их значения (в зависимости от состояний указанных битов) оговариваются технической документацией на конкретные модели МК;

- разность напряжений  $U_{BORL}$  и  $U_{BORH}$  (гистерезис) равна 100 мВ;

- при восстановлении напряжения питания до верхнего порогового уровня сигнал *Reset* устанавливается в пассивное состояние без задержки (сравните рис. 3.8 и 3.9);

- *BOR* не влияет на переключение источника питания батарейного домена;

- выполнение функции *BOR* может быть запрещено определенной битовой комбинацией в *User Option Byte*.

Более подробно вопросы реализации функции *BOR* освещены, например, в [14].

Общий сброс ряда подсемейств / моделей МК также осуществляется при выводе МК из режимов «глубокого сна», т. е. из вариантов энергосберегающего режима, характеризующихся наименьшим количеством работающих функциональных узлов и минимальным энергопотреблением (более подробно – см. подраздел 3.4).

Типовые структурно-архитектурные решения подсистемы сброса МК более подробно освещены в разделе 5.

### **3.3.2. Применение энергонезависимой памяти для восстановления работоспособности МК при сбоях по питанию**

Энергонезависимая память является очевидным и естественным средством сохранения кодов ПО МК и не подлежащих потере данных при выключении питания, сбоях по питанию и при переводе МК в энергосберегающий режим работы. Поэтому:

- ПП практически всех современных семейств МК общего назначения является энергонезависимой;

- структура и архитектура большинства семейств / подсемейств / моделей МК, за исключением наиболее простых моделей класса «*cost-sensitive*», характеризуется или наличием энергонезависимых модулей ПД для сохранения данных, не подлежащих потере, или возможностью записи таких данных в незадействованную под ПО область ПП.

Подробное рассмотрение архитектуры и применения энергонезависимых модулей памяти МК выходит за рамки настоящего раздела, так как типовые структурно-архитектурные решения памяти МК рассмотрены в разделе 2, а вопросы программирования энергонезависимой памяти – в разделе 12

### **3.3.3. Домен с резервированием питания (ДРП)**

**3.3.3.1.** Наличие ДРП (*Backup domain* на рис. 3.5) в структуре МК является эффективным средством поддержания его работоспособности и сохранения данных, не подлежащих потере, в

энергосберегающих режимах, а также при аварийном или преднамеренном отключении основного источника питания. ДРП характерен, в основном, для МК класса «*high-performance*», например, для большинства подсемейств *ARM Cortex-Mx* [11, 13]. Как указано ранее, он содержит группу узлов и блоков МК, работа которых не прекращается ни в одном из энергосберегающих режимов, а питание может осуществляться как от основного, так и от резервного источника питания, и функционирование которых минимально необходимо для поддержания работоспособности МК в энергосберегающих режимах и при отключении основного источника питания. Принцип переключения ДРП с основного источника на резервный описан в пояснениях к рис. 3.5.

После сброса МК регистры блоков, входящих в состав ДРП (см. рис. 3.5) **недоступны** для записи и чтения, во избежание случайного изменения или потери их содержимого. Для получения доступа к ним необходимо [13]:

- установить в единичное состояние бит разрешения тактирования интерфейса подсистемы питания МК (*PWREN, Power interface clock enable*) и бит разрешения тактирования ДРП (*BKPEN, Backup interface clock enable*) в регистре разрешения тактирования устройств *APB1* – домена (*RCC\_APB1ENR*);

- установить в единичное состояние бит *DBP (Disable backup domain write protection)* запрета защиты от записи регистров ДРП, расположенный в регистре управления подсистемой питания (*Power control register, PWR\_CR*).

**Сброс** регистров ДРП (*Backup domain reset*) осуществляется только по одному из следующих событий:

- установке в единицу бита *BDRST (Backup domain reset)* в регистре *RCC\_BDCR* подсистемы управления сбросом и синхронизацией (*Software reset*);

- включении напряжения  $V_{DD}$  или  $V_{BAT}$ , при условии, что до данного события оба этих напряжения были отключены.

**Основные функции**, выполняемые ДРП, следующие:

- сохранение данных, потеря которых недопустима;
- обеспечение бесперебойного отсчета времени, т. е. непрерывной работы «часов» и «календаря» МК.

Соответственно, основными функциональными блоками ДРП являются блок **счетчика реального времени (RTC)** и блок **ВКР-регистров**.

**3.3.3.2.** Как указано в пояснениях к рис. 3.5, **ВКР-регистры** предназначены для сохранения информации, не подлежащей потере при сбое / отказе основного питания.

Блок **ВКР-регистров** МК модельного ряда *STM32F10xxx* включает в себя [13]:

- регистры данных (*Backup data registers, ВКР\_DR*), число которых зависит от конкретной модели МК; у моделей групп *medium-density* и *low-density devices* – 10 регистров, групп *high-density*, *XL-density* и *connectivity line devices* – 42;

- регистр управления калибровкой счетчика реального времени (*RTC clock calibration register, ВКР\_RTCCR*);

- регистр управления ДРП (*Backup control register, ВКР\_CR*);

- регистр управления / статуса ДРП (*Backup control/status register, ВКР\_CSR*).

Основное назначение регистров данных ДРП – хранение данных, потеря которых недопустима, в т. ч. при переходе в какой-либо из энергосберегающих режимов (см. табл. 3.2) или переключении на резервное питание. Например, к таким данным относятся константы, используемые в процессе работы программы МК. После получения доступа к ДРП (см. подпункт 3.3.3.1) запись в эти регистры и чтение их содержимого осуществляются «стандартными» командами, например:

*ВКР->DR1 = xmin;*

(запись значения переменной *xmin* в 1-й регистр данных ДРП);

*xmin = ВКР->DR1;*

(присвоение переменной *xmin* значения, равного содержимому 1-го регистра данных ДРП).

При этом запись в регистры данных ДРП проще с точки зрения программной реализации и занимает значительно меньше времени, чем запись в энергонезависимую память. Поэтому единичные слова данных (не массивы) рационально сохранять именно в регистрах ДРП.

Содержимое регистров данных ДРП **не сбрасывается** ни при выходе из режима *Standby* (см. табл. 3.2), ни при системном сбросе (*System reset*) МК, см. рис. 5.1. Их обнуление может быть осуществлено **только**:

- по одному из событий, вызывающих *Backup domain reset* (см. подпункт 3.3.3.1);

- активным уровнем сигнала на выводе **TAMPER** (в дословном переводе – «Вмешательство») МК.

Функция **TAMPER** закреплена как альтернативная (см. пункт 6.5.2) за одним из выводов («пинов») определенного ПВВ МК; например, у МК модельного ряда *STM32F103xx* – за 13-м выводом ПВВ С [10]. Выполнение данным выводом функции **TAMPER** разрешается установкой в единицу 0-го бита, *TPE (TAMPER pin enable)* регистра *BKP\_CR*. Выбор активного уровня сигнала **TAMPER** производится 1-м битом, *TPAL (TAMPER pin active level)* регистра *BKP\_CR*: при нулевом состоянии данного бита активным является «высокий» уровень сигнала **TAMPER**, при единичном – «низкий». Имеется ряд особенностей использования данного сигнала, см. пункт 6.3.1 Руководства [13]. Отметим, что в регистре **BKP\_CR** задействованы только 0-й и 1-й биты.

В регистре **BKP\_CSR** задействованы следующие биты:

- 9-й (*TIF, Tamper interrupt flag*) – признак прерывания по активному уровню сигнала **TAMPER**;

- 8-й (*TEF, Tamper event flag*) – признак обнаружения активного уровня сигнала **TAMPER**;

- 2-й (*TPIE, TAMPER pin interrupt enable*) – разрешение (единичным состоянием данного бита) прерываний по активному уровню сигнала **TAMPER**;

- 1-й (*CTI, Clear tamper interrupt*) – сброс признака *TIF*, который производится записью единицы в бит *CTI*;

- 0-й (*CTE: Clear tamper event*) – сброс признака *TEF*, осуществляемый записью единицы в бит *CTE*.

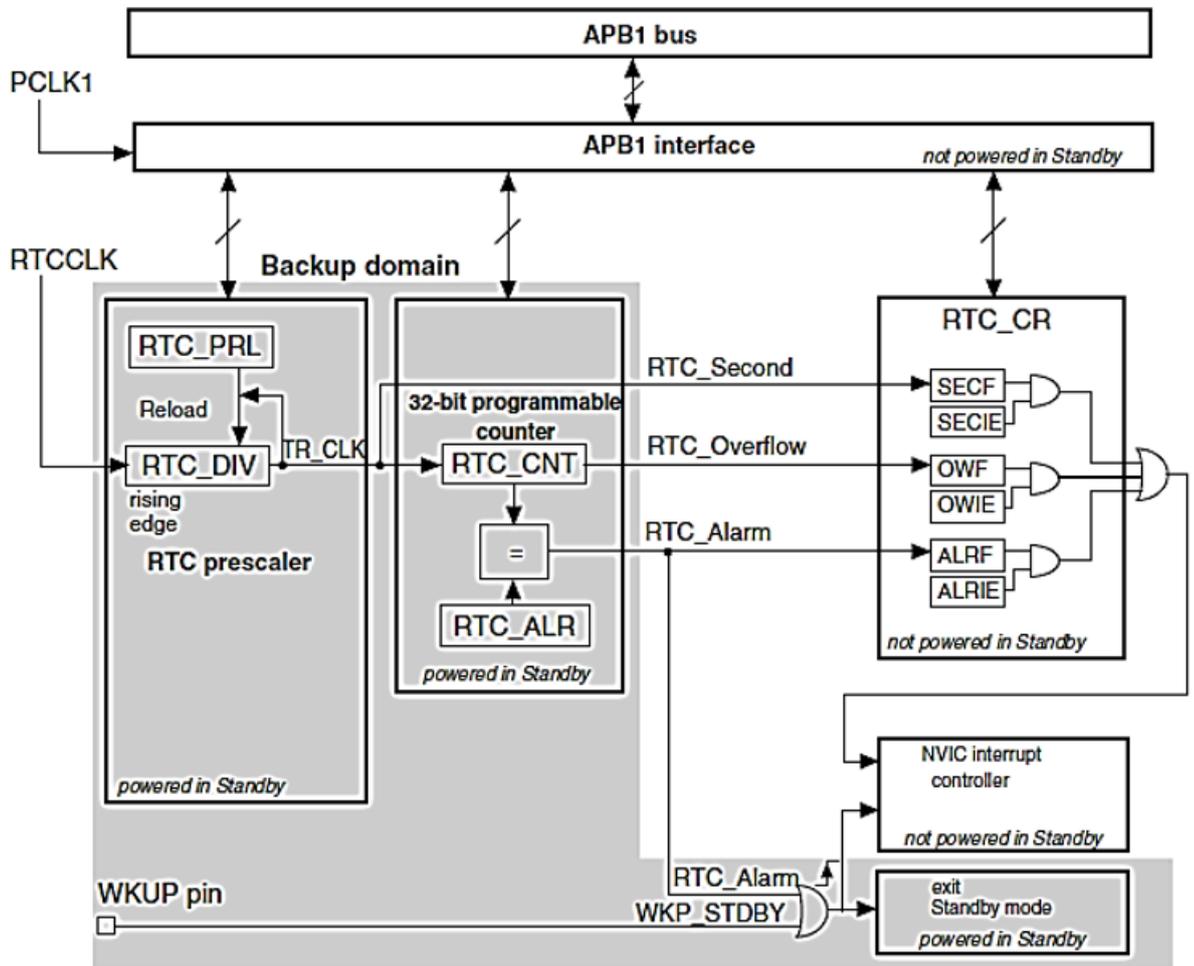
При этом биты *CTI* и *CTE* доступны только для записи; *TIF* и *TEF* – только для чтения. Бит *TPIE* доступен как для чтения, так и для записи; он сбрасывается только при системном сбросе МК (см. рис. 5.1) и при выходе из режима *Standby* (см. табл. 3.2). Прерывание по активному уровню сигнала **TAMPER** не выводит ядро МК из энергосберегающих режимов (см. табл. 3.2).

Регистр **BKP\_RTCCR** функционально относится к счетчику реального времени (*RTC*), и описан в подпункте 3.3.3.3, посвященном структуре и архитектуре *RTC*.

**3.3.3.3.** Вторым из основных блоков ДРП является блок счетчика реального времени (*RTC*). Как указано в подпункте 3.3.3.1, основное назначение блока *RTC* – обеспечение бесперебойного отсчета времени, т. е. непрерывной работы «часов» и «календаря»

МК, в т. ч. в энергосберегающих режимах и при переключении на резервное питание.

На рис. 3.10 в качестве типового примера представлена упрощенная структурная схема блока *RTC* МК модельного ряда *STM32F10xxx* [13].



**Рис. 3.10.** Упрощенная структурная схема блока *RTC* МК модельного ряда *STM32F10xxx* [13]

На рис. 3.10:

- *APB1 bus*, *APB1 interface* – соответственно шина (магистраль) и интерфейс домена *APB1* (см. рис. 1.7 и пояснения к нему);
- *PCLK1* – сигнал тактирования домена *APB1*;
- *RTCCLK* – тактовый сигнал блока *RTC*;
- *RTC\_PRL* – регистр коэффициента деления тактовой частоты счетчика реального времени;
- *RTC\_DIV* – предварительный делитель частоты сигнала *RTCCLK*;

- *TR\_CLK* – сигнал тактирования счетчика реального времени;
- *RTC\_CNT* – счетчик реального времени;
- = – цифровой компаратор;
- *RTC\_ALR* – регистр уставки, задающий значение содержимого счетчика, по достижении которого формируется сигнал *RTC\_Alarm*;
- *RTC\_Second* – сигнал, оповещающий о переполнении предварительного делителя, он же – тактовый сигнал счетчика реального времени, фактически совпадающий с сигналом *TR\_CLK* (название «*RTC\_Second*» происходит от того, что в большинстве практических случаев период данного сигнала задается равным одной секунде);
- *RTC\_Overflow* – сигнал, оповещающий о переполнении счетчика реального времени;
- *RTC\_Alarm* – сигнал «тревоги» (фактически – сигнал, оповещающий о совпадении содержимых счетчика и регистра *RTC\_ALR*);
- *RTC\_CR* – управляющий регистр блока *RTC*;
- *SECF*, *OWF*, *ALRF* – биты признаков соответственно переполнения предварительного делителя, переполнения *RTC*-счетчика и достижения им верхнего предела счета; активное состояние данных битов – единичное;
- *SECIE*, *OWIE*, *ALRIE* – биты разрешения прерываний соответственно по переполнении предварительного делителя, переполнении *RTC*-счетчика и достижении им верхнего предела счета; активное состояние перечисленных битов – единичное;
- *NVIC* – контроллер прерываний (см. пункт 7.3.2);
- *exit Standby mode* – выход из энергосберегающего режима *Standby* (см. табл. 3.2);
- *WKUP pin* – вывод БИС МК, который может быть программно сконфигурирован на функцию вывода МК из режима *Standby*;
- *WKP\_STDBY* – сигнал вывода МК из режима *Standby*;
- *powered in Standby / not powered in Standby* – напряжение питания поступает / не поступает в режиме *Standby* на соответствующий функциональный узел.

Источником тактового сигнала (*RTCCLK*) блока *RTC*, в принципе, может служить:

- генератор *LSE* (см. пояснения к рис. 3.5);
- внутренний низкочастотный RC-генератор МК (*LSI*), см. пункт 4.2.1;

- высокочастотный тактовый генератор с внешним ПЭР (*HSE*), см. пункт 4.2.3.

Выбор источника сигнала *RTCCLK* осуществляется битовым полем *RTCSEL* регистра *RCC\_BDCR* (см. рис. 3.5), подробнее – см. пункт 7.3.9 Руководства [13].

Если в процессе работы МК планируются переходы в энергосберегающие режимы (особенно в режим *Standby*) или существует вероятность сбоев / отключения основного питания, тактирование блока *RTC* рационально осуществлять генератором *LSE*, т. к. он входит в состав ДПП, и его работа не останавливается ни в одном из энергосберегающих режимов, а его питание может осуществляться от резервного источника. Принципы реализации генераторов тактовых импульсов с пьезоэлектрическим резонатором (ПЭР) изложены в пункте 4.2.3. Как правило, в *LSE*-генераторе используется ПЭР с резонансной частотой, равной 32768 Гц, т. е.  $2^{15}$  Гц. Следовательно, результатом деления частоты выходного сигнала *LSE* в  $2^{15}$  раз будет импульсный сигнал с частотой 1 Гц и с периодом, равным 1 секунде, что удобно для отсчета реального времени.

Основным функциональным узлом блока *RTC* является **32-битовый двоичный счетчик**, обозначенный на рис. 3.10 как *RTC\_CNT*. Он реализует собственно отсчет времени, т. е. основную функцию блока *RTC*. Его тактирование осуществляется выходным сигналом предварительного делителя *RTC\_DIV* (см. рис. 3.10), коэффициент деления которого задается 20-битовым регистром *RTC\_PRL*. Частота тактовых импульсов *TR\_CLK* счетчика *RTC\_CNT*, равна [13]:

$$f_{TR\_CLK} = \frac{f_{RTCCLK}}{PRL + 1};$$

где  $f_{RTCCLK}$  – частота сигнала тактирования блока *RTC*, *PRL* – содержимое регистра *RTC\_PRL*. В принципе, может быть установлено любое значение частоты в диапазоне от  $f_{RTCCLK}/2$  до  $f_{RTCCLK}/2^{20}$ ; не рекомендуется только нулевое содержимое регистра *RTC\_PRL* [13]. На практике, как правило,  $f_{TR\_CLK}$  устанавливается равной 1 Гц. Если в качестве источника сигнала *RTCCLK* выбран *LSE*-генератор с частотой 32768 Гц, для равенства частоты сигнала *TR\_CLK* одному Герцу в регистр *RTC\_PRL* должно быть записано число 32767 (7FFF в шестнадцатеричной системе счисления).

Блок *RTC* может служить источником запроса на прерывание по следующим событиям (см. рис. 3.10):

- генерация очередного импульса тактирования счетчика *RTC\_CNT* (*RTC\_Second*);
- переполнение счетчика (*RTC\_Overflow*), т. е. достижение им верхнего предела счета, равного  $0xFFFFFFFF$ ;
- совпадение содержимых счетчика и регистра *RTC\_ALR* (*RTC\_Alarm*);

при условиях, что прерывания по данным событиям разрешены единичными состояниями соответственно битов *SECIE*, *OWIE* и *ALRIE* регистра *RTC\_CR*. Запрос на прерывание по каждому из этих событий вызовет переход к одной и той же подпрограмме обслуживания прерывания (см. подпункт 7.3.2.15). Проверка, каким конкретно из перечисленных событий вызвано прерывание, осуществляется на программном уровне, по состояниям битов признаков соответствующих событий (*SECF*, *OWF* и *ALRF*).

Событие *RTC\_Alarm* также инициирует выход МК из энергосберегающего режима *Standby* (см. рис. 3.10 и табл. 3.2). В целом, данное событие является основным инструментом формирования интервалов времени заданной длительности, наиболее часто используемым при практическом применении блока *RTC*. Длительность отсчитываемого счетчиком *RTC\_CNT* интервала времени от его запуска до события *RTC\_Alarm* равна  $(ALR + 1)/f_{TR\_CLK}$ , где *ALR* – содержимое регистра *RTC\_ALR*. Поскольку разрядность данного регистра равна 32-м битам, при  $f_{TR\_CLK} = 1$  Гц посредством счетчика *RTC\_CNT* могут формироваться интервалы длительностью от 1 до  $2^{32}$  секунд, т. е. примерно до 34-х лет. Естественно, задачи формирования интервалов такой длительности практически не встречаются.

Типовой пример использования события *RTC\_Alarm* представлен в подпункте 3.3.3.8.

**3.3.3.4.** Управление блоком *RTC* и контроль его состояния осуществляются посредством следующих программно-доступных регистров:

- *RTC\_CNT* и *RTC\_DIV*, фактически являющихся счетчиками, но обладающими и статусом программно-доступных регистров ПУ с теми же именами, содержимое которых доступно для считывания ПО, а *RTC\_CNT* – также и для записи (загрузки);
- *RTC\_PRL* и *RTC\_ALR*, доступных только для записи;

- регистра управления блоком *RTC* (*RTC\_CR*);
- регистра управления калибровкой счетчика реального времени (*BKP\_RTCCR*), формально входящего в блок *BKP*-регистров [13], но функционально относящегося к блоку *RTC*.

Назначение счетчиков-регистров *RTC\_CNT* и *RTC\_DIV*, а также регистров *RTC\_PRL* и *RTC\_ALR* понятно из вышесказанного. Необходимо только заметить, что каждый из них, будучи 32-битовым, физически разделен на «старший» и «младший» 16-битовые регистры (счетчики-регистры), с различными именами, образованными добавлением суффикса «*H*» (*High*, старший) или «*L*» (*Low*, младший) к имени 32-битового регистра, например, *RTC\_CNTH* и *RTC\_CNTHL*. Каждому из этих 16-битовых регистров назначается отдельный адрес, обращение к ним осуществляется по их именам.

Аналогичным образом, регистр управления блоком *RTC* (*RTC\_CR*) разделен на «старший» и «младший» 16-битовые регистры (соответственно *RTC\_CRH* и *RTC\_CRL*). В регистре *RTC\_CRH* задействованы только 3 младших бита - *SECIE*, *ALRIE* и *OWIE* (см. пояснения к рис. 3.10), доступные как для чтения, так и для записи. В регистре *RTC\_CRL* используются 6 младших битов. Биты с 0-го по 2-й являются признаками событий *RTC\_Second*, *RTC\_Alarm* и *RTC\_Overflow* (см. пояснения к рис. 3.10). Данные биты доступны для чтения и для сброса записью нуля; запись в них единицы не вызывает эффекта. Бит 3-й регистра *RTC\_CRL* (*RSF*, *Registers synchronized flag*) служит признаком синхронизации блока *RTC* и домена *APB1*; как и биты с 0-го по 2-й, он доступен для чтения и для сброса записью в него нуля. Более подробно назначение бита *RSF* пояснено в описании особенностей процедур чтения содержимого регистров блока *RTC* (см. подпункт 3.3.3.5). 4-й и 5-й биты регистра *RTC\_CRL*, соответственно *CNF* (*Configuration flag*, доступен для чтения и для записи) и *RTOFF* (*RTC operation OFF*, доступен только для чтения) используются в процедуре записи регистров *RTC\_CNT*, *RTC\_PRL* и *RTC\_ALR* (см. подпункт 3.3.3.5).

В регистре *BKP\_RTCCR* задействованы биты с 0-го по 9-й. Биты с 7-го по 9-й (*CCO*, *ASOE* и *ASOS*) используются для управления контролем частоты тактовых импульсов счетчика *RTC\_CNT*; биты с 0-го по 6-й – для хранения калибровочного числа; подробнее – см. подпункт 3.3.3.6.

**3.3.3.5.** Необходимо подробнее остановиться на процедурах получения доступа к перечисленным в подпункте 3.3.3.4 регистрам для чтения и записи (в т. ч. на особенностях сброса их содержимого).

В первую очередь, необходимо отметить, что доступ к регистрам блока *RTC* для сброса, считывания и записи **усложнен** по сравнению с доступом к большинству регистров «ядра» и ПУ МК, ввиду недопустимости случайного повреждения или потери содержимого регистров блока *RTC*. Заметим, что по той же причине усложнен доступ и к *BKP*-регистрам (см. подпункт 3.3.3.2).

**Сброс** регистров *RTC\_CNT*, *RTC\_DIV*, *RTC\_PRL* и *RTC\_ALR* может быть осуществлен **только** по одному из событий, вызывающих *Backup domain reset* (см. подпункт 3.3.3.1). Установка в начальное состояние регистра управления блоком *RTC* (*RTC\_CR*) происходит при системном сбросе (*System reset*) МК, см. рис. 5.1. Биты *ASOS* и *ASOE* регистра *BKP\_RTCCR* обнуляются установкой в единицу бита *BDRST*; бит *CCO* – при отключении основного напряжения питания МК.

При **считывании** содержимого регистров блока *RTC* необходимо учитывать ряд особенностей. Программно-управляемый доступ к ним осуществляется через интерфейс домена *APB1* (см. рис. 3.10). Однако, тактирование данного домена, с одной стороны, и «ядра» блока *RTC*, включающего в себя *RTC\_CNT*, *RTC\_DIV*, *RTC\_PRL* и *RTC\_ALR*, с другой стороны, осуществляется от различных источников синхроимпульсов. Поэтому для корректного считывания содержимого перечисленных регистров, должна быть выполнена взаимная синхронизация домена *APB1* и ядра блока *RTC*. Она осуществляется на аппаратном уровне, ее признаком является единичное состояние бита *RSF* регистра *RTC\_CRL* (см. описание данного регистра в подпункте 3.3.3.4). При системном сбросе МК (см. рис. 5.1) или при его переходе в энергосберегающие режимы *Standby* или *Stop* (см. табл. 3.2) тактирование домена *APB1* останавливается, и синхронизация нарушается. Для корректного считывания содержимого регистров блока *RTC* после сброса или выхода из режимов *Standby* или *Stop*, должны быть выполнены следующие действия.

1. Программный сброс бита *RSF*;
2. Ожидание его установки в единичное состояние.

После выполнения данных действий может производиться считывание содержимого регистров блока *RTC*.

**Запись** содержимого регистров *RTC\_CNT*, *RTC\_DIV*, *RTC\_PRL* и *RTC\_ALR* может быть выполнена только после процедуры получения доступа к ДРП (см. подпункт 3.3.3.1) и перехода в режим **конфигурации** блока *RTC*. Типовой алгоритм записи следующий [13].

1. Ожидание установки в единичное состояние бита *RTOFF* (*RTC operation OFF*) регистра *RTC\_CRL*, являющееся признаком завершения предыдущей процедуры записи.

2. Вход в режим конфигурации блока *RTC*, выполняемый установкой в единицу бита *CNF* (*Configuration flag*) регистра *RTC\_CRL*.

3. Выполнение команд записи содержимого регистров *RTC\_CNT* или / и *RTC\_DIV* или / и *RTC\_PRL*.

4. Выход из режима конфигурации записью нуля в бит *CNF*.

5. Ожидание установки в единичное состояние бита *RTOFF* регистра *RTC\_CRL*, т. е. завершения процедуры записи.

**3.3.3.6.** Архитектура блока *RTC* предоставляет возможность **калибровки** источника тактирования, т. е. компенсации отклонения частоты сигнала *RTCCLK* (см. рис. 3.10) от ее номинального значения. Данная функция реализуется удалением («маскированием») некоторого количества импульсов из каждых  $2^{20}$  (1048576-ти), поступающих на вход тактирования (*RTCCLK*) блока *RTC*. Тем самым частота сигнала *RTCCLK* может быть скорректирована в сторону снижения (см. приведенный далее **Пример 1**). Коррекция частоты в сторону повышения осуществляется комбинированием коэффициента деления предварительного делителя *RTC\_DIV* (см. рис. 3.10) и числа удаляемых импульсов (см. **Пример 2**). При коррекции частоты как в сторону снижения, так и повышения число удаляемых импульсов задается 7-битовым полем *CAL[6:0]* регистра *VKP\_RTCCR*, и определяется по результатам измерения реального значения частоты синхросигнала *RTCCLK*. Для возможности ее измерения необходимо установить в единичное состояние бит *CCO* (*Calibration clock output*) регистра *VKP\_RTCCR*. При этом сигнал *RTCCLK*, через делитель частоты на 64, выдается на пин *TAMPER* МК (см. подпункт 3.3.3.2). Во избежание несанкционированного сброса регистров данных ДРП, перед установкой в единицу бита *CCO* необходимо обнулить бит *TPE* регистра *VKP\_CR* (см. подпункт 3.3.3.2).

Поскольку число импульсов, удаляемых из каждых 1048576-ти, задается 7-битовым полем, его значение находится в пределах от 0 до  $2^7 - 1$ , т. е. 127-ми. Поэтому данным удалением может быть скорректировано относительное отклонение частоты до

$(127/1048576) \times 100\%$ , т. е. до 0,0121% (до 121-й миллионной доли, *ppt, part per million*). Как правило, коррекция в таких пределах достаточна при использовании генераторов с ПЭР, в первую очередь – генератора *LSE*, предпочтительного для использования в качестве источника тактирования блока *RTC* (см. подпункт 3.3.3.3).

**Примечание.** Вообще говоря, во многих практических случаях такие отклонения могут считаться пренебрежимо малыми. Однако, при отсчете реального времени, особенно достаточно длительных временных интервалов (порядка часов, суток и более) отклонения даже порядка единиц – десятков *ppt* должны быть скорректированы.

Принцип коррекции тактовой частоты блока *RTC* удобно пояснить 2-мя примерами.

**Пример 1.** Коррекция частоты тактирования блока *RTC* в сторону снижения.

Пусть тактирование блока *RTC* осуществляется по предпочтительному варианту – *LSE*-генератором на основе ПЭР с номинальным значением резонансной частоты 32768 Гц.

В результате измерения поделенной в 64 раза частоты выходного сигнала генератора на выводе *TAMPER* МК получено значение 512,02 Гц, при номинальном значении  $32768 \text{ Гц} / 64 = 512 \text{ Гц}$ . Следовательно, отклонение частоты тактирования от номинальной равно  $0,02 \text{ Гц} / 512 \text{ Гц} \approx 39 \text{ ppt}$ , а реальная частота выходного сигнала тактового генератора составляет  $512,02 \text{ Гц} \times 64 = 32769,28 \text{ Гц}$ . Необходима коррекция в сторону снижения частоты на 39 *ppt*. Для нее из каждых 1048576-ти выходных импульсов *LSE*-генератора необходимо удалить  $1048576 \times 0,000039$ , т. е., с округлением до ближайшего целого, 41 импульс. Соответственно, в битовое поле *CAL[6:0]* регистра *BKP\_RTCCR* должно быть записано число 41 (29 в шестнадцатеричной системе). Тогда:

- с выхода *LSE*-генератора 1048576 импульсов поступят за интервал времени длительностью  $1048576 / 32769,28 \text{ Гц} \approx 31,99875 \text{ с}$ ;

- за каждый интервал времени с такой длительностью на вход тактирования блока *RTC* реально поступит  $1048576 - 41 = 1048535$  импульсов;

- следовательно, **средняя** частота тактовых импульсов блока *RTC* будет равна  $1048535 / 31,99875 \text{ с} \approx 32767,9987 \text{ Гц}$ , т. е. ее отклонение от номинального значения составит

$-0,0013 \text{ Гц}/32768 \text{ Гц} \approx 0,04 \text{ ppm}$  (в данном примере – в 1000 раз меньше, чем в отсутствие коррекции).

См. также **Примечание** после Примера 2.

**Пример 2.** Коррекция частоты тактирования блока *RTC* (точнее – счетчика *RTC\_CNT*) в сторону повышения.

Как и в Примере 1, тактирование блока *RTC* осуществляется по предпочтительному варианту – *LSE*-генератором на основе ПЭР с номинальным значением резонансной частоты 32768 Гц.

Поскольку при коррекции частоты тактирования в сторону повышения задействуется не только битовое поле *CAL[6:0]* регистра *VKP\_RTCCR*, но и коэффициент деления частоты тактирования счетчика *RTC\_CNT* (см. первый абзац подпункта 3.3.3.6), коррекция возможна только при известной и фиксированной частоте тактирования счетчика *RTC\_CNT*, необходимой для решения конкретной задачи. В данном примере положим, что она равна 1 Гц (что характерно для большинства задач практического применения блока *RTC*).

Пусть в результате измерения поделенной в 64 раза частоты выходного сигнала *LSE*-генератора получено значение 511,98 Гц (при номинальном значении 512 Гц, как и в Примере 1). Следовательно, в данном примере отклонение частоты от номинальной равно  $-0,02 \text{ Гц}/512 \text{ Гц} \approx -39 \text{ ppm}$ , а реальная частота выходного сигнала тактового генератора составляет  $511,98 \text{ Гц} \times 64 = 32766,72 \text{ Гц}$ . Необходима коррекция в сторону повышения частоты на 39 ppm.

Удаление импульсов из выходного сигнала тактового генератора блока *RTC* (см. Пример 1), очевидно, позволит корректировать частоту только в сторону снижения (причем частоту тактирования блока *RTC* в целом). Поэтому в сторону повышения может быть выполнена только коррекция тактовой частоты счетчика *RTC\_CNT*, реализуемая следующим способом.

1. Коэффициент деления предварительного делителя *RTC\_DIV* (см. рис. 3.10) задается (посредством регистра *RTC\_PRL*) таким, чтобы отклонение частоты тактирования счетчика *RTC\_CNT* от требуемого значения (в данном примере – от 1 Гц) находилось в пределах от 0 до +121 ppm при **реально измеренном** значении частоты выходного сигнала генератора тактовых импульсов;

2. Определяется число *CAL[6:0]*, снижающее среднюю частоту тактирования блока *RTC* до значения, при котором отклонение

средней частоты тактовых импульсов счетчика  $RTC\_CNT$  от требуемого значения не выходит за пределы  $\pm 1 \text{ ppm}$  (при заданном в соответствии с пунктом 1 коэффициенте деления предварительного делителя).

В рассматриваемом примере коэффициент деления предварительного делителя, удовлетворяющий пункту 1, равен 32766, ему соответствует содержимое регистра  $RTC\_PRL$ , равное 32765 ( $0x7FFD$ ). При этом частота тактового сигнала счетчика, в отсутствие коррекции, равна  $32766,72 \text{ Гц}/32766 \approx 1,000022 \text{ Гц}$ , т. е. ее отклонение от требуемой равно  $+22 \text{ ppm}$ . Следовательно, для достижения требуемой средней частоты тактирования счетчика (1 Гц) из каждых 1048576-ти выходных импульсов  $LSE$ -генератора необходимо удалить  $1048576 \times 0,000022$ , т. е., с округлением до ближайшего целого, 23 импульса. В битовое поле  $CAL[6:0]$  регистра  $VKP\_RTCCR$  при этом должно быть записано число 23 (17 в шестнадцатеричной системе). Тогда:

- с выхода  $LSE$ -генератора 1048576 импульсов поступят за интервал времени длительностью  $1048576 / 32766,72 \text{ Гц} \approx 32,00125 \text{ с}$ ;

- за каждый интервал времени с такой длительностью на вход тактирования блока  $RTC$  реально поступит  $1048576 - 23 = 1048553$  импульса;

- **средняя** частота тактовых импульсов блока  $RTC$  будет равна  $1048553 / 32,00125 \text{ с} \approx 32766,00133 \text{ Гц}$ ;

- **средняя** частота тактового сигнала счетчика  $RTC\_CNT$  при этом будет равна  $32766,00133 \text{ Гц}/32766 \approx 1,00000004 \text{ Гц}$ ; ее отклонение от номинального значения составит  $0,04 \text{ ppm}$ .

**Примечание.** Как следует из Примеров 1 и 2, удаление задаваемого битовым полем  $CAL[6:0]$  числа импульсов из выходного сигнала тактового генератора блока  $RTC$  позволяет корректировать только **среднюю** частоту тактирования за интервал времени длительностью 1048576 периодов выходного сигнала генератора. Поэтому данная коррекция имеет смысл только при использовании блока  $RTC$  для отсчета времени в течение интервалов с длительностью, на порядок и более превышающей  $2^{20} / f_{RTCCLK}$  (где  $f_{RTCCLK}$  – частота выходного сигнала тактового генератора блока  $RTC$ ). При отсчете интервалов меньшей длительности в коррекции нет необходимости; более того, она может даже отрицательно повлиять на точность формирования интервалов времени. Отметим,

что коррекция частоты тактирования, как правило, необходима именно при отсчете достаточно длительных интервалов времени.

**3.3.3.7.** Отдельно необходимо отметить, что функционально относящийся к ДРП регистр *RCC\_BDCR* (см. рис. 3.5), как и другие регистры ДРП, не сбрасывается при системном сбросе МК. Его установка в начальное состояние возможна только по одному из событий, вызывающих *Backup domain reset* (см. подпункт 3.3.3.1). После сброса МК регистр *RCC\_BDCR* недоступен для записи. Изменение его содержимого возможно только после процедуры получения доступа, описанной в подпункте 3.3.3.1.

**3.3.3.8.** Приведем типовой пример практического применения блока *RTC*. Пусть МК должен в фиксированные моменты времени, с интервалами в несколько десятков минут или несколько часов, получать команды от хост-компьютера и, в соответствии с ними, подавать управляющие сигналы на некоторые исполнительные устройства. При этом основное питание МК осуществляется от аккумулятора (батареи), поэтому желательно, чтобы в интервалах времени между сеансами связи с хостом МК работал в энергосберегающем режиме с минимальным потреблением мощности. Такая задача характерна, например, для систем «Умный дом».

Предполагается, что используется МК модельного ряда *STM32F10xxx*.

Для решения данной задачи может быть предложен следующий упрощенный алгоритм.

1. Битовым полем *RTCSEL* регистра *RCC\_BDCR* выбрать генератор *LSE* в качестве источника тактового сигнала блока *RTC*. Выполнить процедуру коррекции частоты тактирования счетчика *RTC\_CNT* (см. Примеры 1 и 2).

2. Запрограммировать регистры *RTC\_PRL* и *RTC\_ALR* таким образом, чтобы совпадение содержимого *RTC\_CNT* и *RTC\_ALR* произошло по истечении заданного интервала времени между сеансами связи. Например, если интервал времени между сеансами связи с хостом должен быть равен одному часу, а частота  $f_{TR\_CLK}$  равна 1 Гц, в регистр *RTC\_ALR* следует записать число 3599 (*E0F* в шестнадцатеричной системе).

3. Сбросить счетчик *RTC\_CNT*.

4. Синхронизировать с хостом начало отсчета времени.

5. Запустить *RTC\_CNT* и перевести МК в энергосберегающий режим работы *Standby*, характеризуемый минимальным энергопотреблением (см. пункт 3.4.3).

6. По выходе МК из режима *Standby*, который происходит по совпадении содержимого *RTC\_CNT* и *RTC\_ALR* - остановить и сбросить счетчик, провести сеанс связи с хостом и перейти к пункту 5.

Напомним, что при переходе в любой из энергосберегающих режимов и при переключении на питание от резервного источника питания счетчиков *RTC\_DIV* и *RTC\_CNT*, регистров *RTC\_PRL* и *RTC\_ALR*, а также регистра *RCC\_BDCR* не отключается, а их содержимое сохраняется.

**3.3.3.9.** Следует также отметить, что в режиме *Standby* ток потребления ДРП, например, МК *STM32F103C8* не превышает 2,2 мкА [10]. Поэтому его питание, например, от батареи распространенной модели *CR2032* емкостью 200 мА/ч, может осуществляться в течение примерно 8-и лет (при разряде батареи до уровня 20% от первоначальной емкости).

**3.3.3.10.** Структурно-архитектурные решения ДРП других моделей / семейств МК, в целом, аналогичны вышеописанному, но могут отличаться от него рядом особенностей. С ними можно ознакомиться по технической документации на соответствующие модели / семейства МК.

## **3.4 Минимизация энергопотребления МК. Энергосберегающие режимы работы**

### **3.4.1. Общие принципы реализации энергосберегающих режимов**

Во многих практических применениях МК критичной является мощность, потребляемая от источника питания устройством на базе МК. Особенно важен данный параметр при питании от батареи / аккумулятора. Поэтому практически во всех семействах / подсемействах МК предусмотрены возможности [4, 8, 9, 13 – 15]:

- минимизации энергопотребления путем рационального конфигурирования функциональных узлов и блоков МК;
- программно-управляемого перевода МК в режимы пониженного потребления мощности на интервалы времени, в

течение которых МК, в соответствии с реализуемым алгоритмом, не должен выполнять никаких действий.

В целом, принципы минимизации энергопотребления современных МК и реализации энергосберегающих режимов основываются на следующих базовых свойствах КМОП-технологии, по которой реализуются практически все современные семейства МК общего назначения:

- ток, потребляемый логическими элементами КМОП в статическом режиме, т. е. в отсутствие переключений и нахождении в некотором фиксированном состоянии (нулевом или единичном) весьма мал; например, ток, потребляемый БИС МК K1986BE92F1I в статическом режиме, не превышает 1,5 мА [11], а БИС *STM32F103C8* – 370 мкА [10];

- в динамическом режиме потребляемый ток увеличивается, возрастая с ростом частоты переключения; например, ток, потребляемый БИС *STM32F103C8*, равен порядка 9 мА при работе на тактовой частоте 8 МГц и порядка 50 мА – на частоте 72 МГц [10].

Из вышесказанного следует, что минимизация энергопотребления МК, реализованных по технологии КМОП, может быть достигнута следующими **основными способами**:

- снижение тактовых частот как ядра МК, так и задействованных периферийных устройств, до значений, допустимых с точки зрения выполняемых задач;

- перевод в статический режим функциональных узлов и блоков МК, не задействованных при выполнении реализуемой МК задачи или на некоторых ее этапах.

Следует также отметить, что снижение энергопотребления МК, казалось бы, очевидным способом отключения питания не используемых функциональных узлов и блоков **недопустимо**, т. к. при этом будет существовать вероятность подачи сигналов от работающих узлов / блоков МК на обесточенные, что приведет к выводу их из строя (см. пункт 3.2.3).

**Первый** из вышеперечисленных способов минимизации энергопотребления МК реализуется программным управлением частотами тактирования как ядра МК, так и каждого из периферийных устройств (ПУ). Технически данное управление реализуется посредством предделителей (*Prescalers*) тактовой

частоты, входящих в состав как подсистемы синхронизации, так и каждого из ПУ [4, 8, 13 – 15]. Их коэффициенты деления задаются программно, индивидуально для каждого из них. Для минимизации энергопотребления данные коэффициенты следует выбирать таким образом, чтобы тактовые частоты ядра МК и задействованных ПУ были минимально необходимыми для удовлетворения требований по производительности конкретного устройства на МК. Естественно, если требуется, коэффициенты деления каких-либо из предделителей могут устанавливаться равными единице (отсутствие деления частоты).

Более подробно вопросы применения предделителей освещены в пунктах 3.4.2 и 3.4.3, а также в подразделах 4.3 и 4.4.

**Второй** из вышеуказанных способов минимизации энергопотребления (перевод в статический режим не задействованных узлов и блоков МК) реализуется следующим образом.

Во-первых, практически во всех семействах и моделях МК существует возможность программного разрешения работы только узлов и блоков МК, необходимых для реализации конкретного устройства на его основе, что является наиболее простым и очевидным способом снижения потребляемой мощности. Поэтому в каком-либо из регистров управления практически всех узлов и блоков МК имеется бит разрешения (*Enable*) его работы [4, 8, 13 – 15]. По умолчанию, данный бит находится в пассивном (как правило, нулевом) состоянии. При этом питание соответствующего узла / блока не отключено, но его работа заблокирована, и, фактически, он находится в статическом режиме. Для его перевода в рабочий режим бит разрешения необходимо программно установить в активное состояние.

Во-вторых, архитектура практически всех современных семейств МК также предоставляет возможность программного перевода МК в какой-либо из **энергосберегающих режимов (ЭСР)**, также называемых «режимами сна». У простейших МК класса «*cost-sensitive*» существует только один ЭСР. Все ЭСР всех семейств МК характеризуются одним общим свойством – отключением тактирования основного потребителя мощности – ЦП, и, как следствие, его остановкой и переводом в статический режим, т. е. в режим минимального энергопотребления. В большинстве ЭСР

отключается тактирование и ряда других блоков / подсистем МК, за исключением тех, функционирование которых минимально необходимо для поддержания работоспособности МК в соответствующем ЭСР (см., например, рис. 3.10) и для вывода МК из него («пробуждения»). Различные ЭСР отличаются между собой составом функциональных блоков МК, тактирование которых отключается, а также составом событий, инициирующих «пробуждение» МК (см. пункты 3.4.2 и 3.4.3). При этом, чем большее число функциональных блоков МК находится в статическом режиме, тем меньше энергопотребление МК, но тем длительнее процесс его «пробуждения» и тем меньше событий могут его инициировать. Типовые структурно-архитектурные решения по реализации ЭСР на примере семейств МК *AVR* и *ARM Cortex-Mx* представлены в пунктах 3.4.2 и 3.4.3.

Естественно, перевод МК в какой-либо из ЭСР является допустимым только на этапах выполнения реализуемого алгоритма, на которых возможна работа МК в соответствующем режиме (см., например, алгоритм, приведенный в подпункте 3.3.3.8). Также, для минимизации энергопотребления в выбранном ЭСР, перед переходом в него рекомендуется [8, 9, 13 – 15]:

- запретить работу (в том числе тактирование) узлов и блоков МК, активное состояние которых не требуется для функционирования МК в выбранном ЭСР и для выхода из него;

- установить минимально необходимую частоту тактирования узлов и блоков МК, остающихся в активном состоянии в выбранном ЭСР.

Кроме вышеописанных основных способов энергосбережения, существует и ряд дополнительных, например, программная установка режима работы не задействованных выводов БИС МК, обеспечивающего минимальное энергопотребление (см. пункты 3.4.2 и 3.4.3).

Рассмотрим типовые структурно-архитектурные решения по минимизации энергопотребления и реализации энергосберегающих режимов МК на примере семейств *AVR* и *ARM Cortex-Mx*.

### 3.4.2. Минимизация энергопотребления и энергосберегающие режимы МК семейства AVR

Структурно-архитектурные решения по минимизации энергопотребления и реализации энергосберегающих режимов МК семейства AVR будем рассматривать на примере МК *ATmega128* (1887BE7T) [8].

Архитектура МК семейства AVR предоставляет разработчику следующие основные **способы** минимизации энергопотребления.

**3.4.2.1. Программное разрешение работы** только узлов и блоков, необходимых для функционирования реализуемого устройства. Осуществляется установкой в активное состояние бита разрешения в регистре управления соответствующим устройством. Исключения составляют таймеры МК семейства AVR, запуск и остановка которых производятся битами *CSx* выбора частоты тактирования в одном из управляющих регистров таймера; остановка таймера осуществляется записью нулей во все данные биты.

Необходимо отметить, что в МК семейства AVR, в отличие от *ARM Cortex-Mx* (см. пункт 3.4.3), тактирование остановленных узлов и блоков, в общем случае, не отключается.

**3.4.2.2. Программное управление тактовой частотой МК.** Реализуется посредством делителя частоты генератора тактовых импульсов (ГТИ) МК. Коэффициент деления задается программно-доступным регистром управления делителем. В МК *ATmega128* (1887BE7T) данному регистру присвоено имя *XDIV*, его разрядность – 8 бит, из которых старший (*XDIVEN*) служит битом разрешения деления, а младшие 7 битов (*XDIV0...XDIV6*) задают коэффициент деления. При единичном состоянии бита *XDIVEN* тактовая частота МК, в том числе его ЦП, определяется по выражению [8]:

$$f_{CLK} = \frac{f_G}{129 - XDIV}; \quad (3.1)$$

где  $f_G$  – частота основного ГТИ МК (см. пункт 4.4.1), *XDIV* – число, записанное в битовое поле *XDIV0...XDIV6*. При нулевом состоянии бита *XDIVEN* деление отсутствует, содержимое данного поля игнорируется. Однако, только при *XDIVEN = 0* биты *XDIV0...XDIV6* доступны для записи.

С точки зрения минимизации энергопотребления, рационален выбор значения  $f_{CLK}$ , не превышающего (или незначительно превышающего) минимально необходимое для удовлетворения требований по производительности конкретного устройства на МК.

Возможность предварительного деления частоты ГТИ МК имеется практически во всех МК семейства AVR. Имя и формат регистра управления делителем зависят от конкретной модели МК и могут отличаться от вышеописанных. В частности, архитектура ряда моделей МК предоставляет возможность установки только нескольких фиксированных значений коэффициента деления, являющихся целыми степенями 2-х [6, 7].

**3.4.2.3. Программное управление тактовой частотой ПУ,** реализуемое посредством предделителей, входящих в состав каждого из них (за исключением портов ввода / вывода). В качестве входного сигнала всех предделителей служит тактовый сигнал МК частотой  $f_{CLK}$  (см. выражение (3.1)). Коэффициенты деления предделителей задаются программно, индивидуально для каждого из них. Для этого в каком-либо из регистров управления каждым из ПУ имеется доступное для записи битовое поле, определяющее коэффициент деления его предделителя. Например, тактовая частота АЦП МК *ATmega128* (1887BE7T) задается младшими 3-мя битами (*ADPS2–0*) регистра *A* статуса и управления АЦП (*ADCSRA*) [8]. При содержимом данного битового поля, равном 000 или 001 тактовая частота АЦП равна  $f_{CLK}/2$ , коду 010 соответствует частота  $f_{CLK}/4$ , коду 011 -  $f_{CLK}/8$  и т. д.

Выбор тактовой частоты каждого из задействованных ПУ должен осуществляться, исходя из обеспечения приемлемого сочетания производительности и энергопотребления.

**3.4.2.4. Программная установка режима работы не задействованных выводов БИС МК,** при котором обеспечивается минимальное энергопотребление. Технической документацией на большинство моделей МК семейства AVR для минимального энергопотребления и максимальной помехоустойчивости рекомендуется не задействованные выводы портов МК устанавливать в режим работы входов с включенными подтягивающими резисторами (подробнее – см. пункт 6.3.2).

**3.4.2.5. Перевод МК в один из ЭСР** на этапах выполнения реализуемого алгоритма, допускающих работу МК в соответствующем ЭСР.

Архитектура МК большинства МК подсемейства *ATmega*, предоставляет возможность реализации 6-и ЭСР. Их характеристики представлены в табл. 3.1.

Таблица 3.1

*Энергосберегающие режимы МК ATmega128 / 1887BE7T [8]*

Вариант ЭСР	Краткие характеристики ЭСР		
	Работающие ГТИ <sup>2)</sup>	Тактируемые блоки МК <sup>4)</sup>	События, инициирующие выход МК из ЭСР <sup>5)</sup>
1	2	3	4
Режим холостого хода ( <i>Idle</i> ); $SM^{1)} = 000$	Основной ГТИ, тактовый генератор 0-го таймера <sup>3)</sup>	Все ПУ	Внешние прерывания; обращение к МК по интерфейсу $I^2C$ ; прерывания по готовности загрузки ПП / ЭСПЗУ; прерывание от любого из ПУ
Режим аналого-цифрового преобразования с пониженным уровнем шумов ( <i>ADC Noise Reduction</i> ); $SM^{1)} = 001$	Основной ГТИ, тактовый генератор 0-го таймера <sup>3)</sup>	АЦП, 0-й таймер	Внешние прерывания; обращение к МК по интерфейсу $I^2C$ ; прерывания по готовности загрузки ПП / ЭСПЗУ; прерывания от АЦП или от 0-го таймера
Режим хранения ( <i>Power-down</i> ); $SM^{1)} = 010$	- <sup>2)</sup>	- <sup>4)</sup>	Внешние прерывания; обращение к МК по интерфейсу $I^2C$
Режим микропотребления ( <i>Power-save</i> ); $SM^{1)} = 011$	Тактовый генератор 0-го таймера <sup>3)</sup>	0-й таймер <sup>3)</sup>	Внешние прерывания; обращение к МК по интерфейсу $I^2C$ ; прерывания от 0-го таймера <sup>3)</sup>
Режим ожидания ( <i>Standby</i> ) <sup>6)</sup> ; $SM^{1)} = 110$	Основной ГТИ	- <sup>4)</sup>	Внешние прерывания; обращение к МК по интерфейсу $I^2C$

### Окончание таблицы 3.1

1	2	3	4
Режим ожидания ( <i>Extended Standby</i> ) <sup>6)</sup> ; <i>SM</i> <sup>1)</sup> = 111	Основной ГТИ, тактовый генератор 0-го таймера <sup>3)</sup>	0-й таймер <sup>3)</sup>	Внешние прерывания; обращение к МК по интерфейсу <i>I<sup>2</sup>C</i> ; прерывания от 0-го таймера <sup>3)</sup>
<p><b>Примечания.</b></p> <p><sup>1)</sup> <i>SM</i> – содержимое битового поля <i>Sleep Mode</i> регистра <i>MCUCR</i> (кодовые комбинации 100 и 101 зарезервированы и не используются).</p> <p><sup>2)</sup> ГТИ сторожевого таймера функционирует во всех ЭСР (при условии, что его работа разрешена).</p> <p><sup>3)</sup> При условии, что бит разрешения асинхронного тактирования 0-го таймера (бит <i>AS0</i> регистра <i>ASSR</i>) установлен в активное (единичное) состояние.</p> <p><sup>4)</sup> Тактирование подсистемы сброса и портов ввода-вывода, служащих приемниками запросов внешних прерываний, осуществляется во всех ЭСР.</p> <p><sup>5)</sup> Во всех ЭСР «пробуждение» МК также осуществляется при внешнем сбросе, при сбросе, вызванном сбоем по питанию и при сбросе от сторожевого таймера (при условии, что его работа разрешена) (см. раздел 5).</p> <p><sup>6)</sup> Данные варианты ЭСР возможны только при использовании встроенного ГТИ с внешним пьезоэлектрическим резонатором в качестве основного ГТИ МК.</p>			

Задание варианта ЭСР МК *Atmega128* (1887BE7T) осуществляется битами со 2-го по 4-й регистра управления МК (*MCUCR*) [8]. Данное битовое поле имеет название *Sleep Mode (SM)*. Перевод МК в ЭСР осуществляется командой *SLEEP* (см. табл. 2.7) при активном (единичном) состоянии 5-го бита (*SE, Sleep Enable*) регистра *MCUCR*. Во избежание незапланированных переходов МК в ЭСР рекомендуется устанавливать данный бит в единицу только непосредственно перед командой *SLEEP* и сбрасывать в нулевое состояние сразу после выхода МК из ЭСР [8].

Корректной является следующая последовательность действий по переводу МК в ЭСР [8]:

- запретить работу узлов и блоков МК, активное состояние которых не требуется для функционирования МК в выбранном ЭСР и для выхода из него;

- установить минимально необходимую частоту тактирования узлов и блоков МК, остающихся в активном состоянии в выбранном ЭСР;

- записать в битовое поле *SM* код выбранного варианта ЭСР;
- установить в активное (единичное) состояние бит *SE MCUCR*;
- командой *SLEEP* перевести МК в выбранный ЭСР.

Из представленных в табл. 3.1 ЭСР минимальным энергопотреблением характеризуется режим хранения, максимальным – режим холостого хода. Согласно [8], в первом из данных ЭСР ток, потребляемый МК 1887BE7T от источника питания, не превышает 200 мкА, а во втором (при тактовой частоте МК, равной 8 МГц) – 22 мА. В активном режиме при той же тактовой частоте максимальный ток, потребляемый МК, равен 50 мА. Необходимо также заметить, что чем «глубже сон» МК в ЭСР, тем длительнее процесс его «пробуждения» и тем меньше событий могут его инициировать.

Состав и характеристики ЭСР МК *Atmega128* (1887BE7T) являются типовыми для всех МК подсемейства *Atmega* семейства *AVR*, за исключением незначительных отличий, обусловленных архитектурой конкретных моделей МК и отражаемых в технической документации на них. Принципы и структурно-архитектурные решения реализации ЭСР МК подсемейства *Attiny*, в целом, аналогичны таковым МК подсемейства *Atmega*, однако, архитектура большинства моделей *Attiny* предоставляет возможность реализации только 3-х ЭСР (режимы холостого хода, аналого-цифрового преобразования с пониженным уровнем шумов и хранения) [7].

Следует отметить, что ни в одном из ЭСР МК семейства *AVR* не отключается питание ни ЦП, ни энергозависимой памяти, ни каких-либо других функциональных блоков МК. Поэтому при переходах в ЭСР и при выходах из них (за исключением выходов путем сброса) содержимое РОН и РСФ, а также энергозависимой ПД остается неизменным.

Особенности минимизации энергопотребления конкретных моделей МК семейства *AVR* отражены в их технических описаниях и руководствах по применению.

### **3.4.3. Минимизация энергопотребления и энергосберегающие режимы МК семейства *ARM Cortex-Mx***

Структурно-архитектурные решения по минимизации энергопотребления и реализации энергосберегающих режимов МК

семейства *ARM Cortex-Mx* будем рассматривать их на примере МК модельного ряда *STM32F10xxx* [13].

Архитектура МК семейства *ARM Cortex-Mx*, как и МК семейства *AVR*, предоставляет возможность применения следующих основных способов минимизации энергопотребления.

**3.4.3.1. Программное разрешение работы** только узлов и блоков, необходимых для функционирования реализуемого устройства. Ввиду большого их количества, для минимизации энергопотребления в МК семейства *ARM Cortex-Mx* разрешение тактирования осуществляется индивидуально для каждого из функциональных блоков (в отличие от МК семейства *AVR*). Поэтому для запуска какого-либо из блоков МК семейства *ARM Cortex-Mx* необходимо разрешить не только его работу, но и его тактирование. Например, для запуска некоторого таймера (допустим, 3-го) МК модельного ряда *STM32F10xxx* [13] подсемейства *ARM Cortex-M3* необходимо:

- разрешить его тактирование установкой в активное (единичное) состояние бита *TIM3EN* регистра *RCC\_APB1ENR* разрешения тактирования периферийных устройств, подключенных к шине *APB1* (*Advanced Peripheral Bus 1*) (см. рис. 1.7);

- сконфигурировать таймер и разрешить его работу установкой в единицу 0-го бита (*CEN*) 1-го регистра управления 3-м таймером (*TIM3\_CR1*).

У некоторых моделей МК семейства *ARM Cortex-Mx* процедура разрешения работы функционального блока несколько сложнее. Например, для запуска 1-го таймера МК *K1986BE92F11* необходимо [11]:

- разрешить тактирование 1-го таймера установкой в единицу 14-го бита (*TIMER1*) регистра управления тактированием периферийных устройств (*MDR\_RST\_CLK->PER\_CLOCK*);

- задать частоту тактового сигнала 1-го таймера кодом, записываемым в биты с 0-го по 7-й регистра управления тактированием таймеров (*MDR\_RST\_CLK->TIM\_CLOCK*) и разрешить подачу тактовых импульсов на таймер установкой в единицу 24-го бита (*TIM1CLKEN*) того же регистра;

- сконфигурировать таймер и разрешить его работу установкой в единицу 0-го бита (*CNT\_EN*) регистра управления 1-м таймером (*MDR\_TIMER1->CNTRL*).

**3.4.3.2. Программное управление тактовой частотой** доменов (см. рис. 1.7) и частотами тактирования функциональных блоков МК, возможность которого имеется практически во всех моделях МК семейства *ARM Cortex-Mx*. Например, в МК модельного ряда *STM32F10xxx* [13]:

- частота тактирования ЦП, а также *AHB*-домена в целом задается битами с 4-го по 7-й (*HPRE[3:0]*) регистра конфигурации системы тактирования (*RCC\_CFGR*) и, в зависимости от состояния данных битов может быть равна от 1/2 до 1/512 частоты основного ГТИ (выбор которого осуществляется 0-м и 1-м битами того же регистра, подробнее – см. пункт 4.4.2);

- частота тактирования АЦП определяется битами 14-м и 15-м (*ADCPRE[1:0]*) регистра *RCC\_CFGR*, а также битами 11...13 (*PPRE[2:0]*) того же регистра, задающими частоту тактирования домена APB2, к которому принадлежат АЦП (см. рис. 1.4);

и т. п.

Выбор тактовой частоты ЦП и каждого из задействованных функциональных блоков осуществляется, исходя из обеспечения приемлемого сочетания производительности и энергопотребления.

**3.4.3.3.** Как и в МК семейства *AVR*, в МК семейства *ARM Cortex-Mx* имеется возможность программной установки **режима работы не задействованных выводов**, при котором обеспечивается минимальное энергопотребление. Аналогично незадействованным выводам БИС МК семейства *AVR*, для минимального энергопотребления и максимальной помехоустойчивости рекомендуется устанавливать их в режим работы входов с подключенными подтягивающими резисторами, МК (подробнее – см. пункт 6.3.2).

**3.4.3.4.** Перевод МК в **один из ЭСР** на этапах выполнения реализуемого алгоритма, допускающих работу МК в соответствующем ЭСР.

Архитектура МК семейства *ARM Cortex-Mx* предоставляет возможность реализации 3-х ЭСР [9, 13 – 15]:

- «Сон» (*Sleep*);
- «Останов» (*Stop*) или «Глубокий сон» (*Deep Sleep*);
- «Ожидание» (*Standby*).

Их основные характеристики представлены в табл. 3.2 и в примечаниях к ней.

Таблица 3.2

## Энергосберегающие режимы МК семейства ARM Cortex-Mx [13 – 15]

Вариант ЭСР	Останавливаемые ГТИ	Узлы и блоки МК, тактирование которых отключено	Состояние регулятора напряжения 1,8 В	Процедура входа в ЭСР	События, инициирующие выход МК из ЭСР
1	2	3	4	5	6
«Сон» ( <i>Sleep</i> ) (вариант с ожиданием прерывания) <sup>1)</sup>	Не останавливается ни один из ГТИ	ЦП, резидентная ПП и ПД	Работает в нормальном режиме	Команда <i>WFI</i> (см. табл. 2.8) при <i>SLEEPDEEP</i> <sup>3)</sup> = 0	Любой разрешенный запрос на прерывание <sup>5)</sup>
«Сон» ( <i>Sleep</i> ) (вариант с ожиданием события) <sup>1)</sup>				Команда <i>WFE</i> (см. табл. 2.8) при <i>SLEEPDEEP</i> = 0	Любое событие, инициирующее «пробуждение» <sup>6)</sup>
Останов ( <i>Stop</i> )	<i>HSE</i> и <i>HSI</i>	Все узлы и блоки, питаемые напряжением 1,8 В	Работает в нормальном режиме или в режиме пониженной мощности <sup>2)</sup>	Команда <i>WFI</i> или <i>WFE</i> при <i>SLEEPDEEP</i> = 1 и <i>PDDS</i> <sup>4)</sup> = 0	Запрос на прерывание или на событие <sup>7)</sup> по любой из линий внешних запросов ( <i>EXTI</i> ), по которым разрешена генерация запроса на прерывание / событие <sup>8)</sup>

Продолжение таблицы 3.2

1	2	3	4	5	6
Ожидание ( <i>Standby</i> )	<i>HSE</i> и <i>HSI</i>	Все узлы и блоки, питаемые напряжением 1,8 В <sup>9)</sup>	Отключен <sup>10)</sup>	Команда <i>WFI</i> или <i>WFE</i> при <i>SLEEPDEEP</i> = 1, <i>PDDS</i> = 1 и <i>WUF</i> <sup>11)</sup> = 0 См. также сноску <sup>12)</sup>	Сброс по выводу <i>NRST</i> <sup>13)</sup> , сброс от <i>IWDT</i> <sup>14)</sup> , перепад из 0 в 1 на каком- либо из активизированных входов <i>WKUPx</i> <sup>15)</sup> , <i>ALARM RTC</i> <sup>16)</sup> . См. также сноску <sup>17)</sup>

**Примечания.**

<sup>1)</sup> Для каждого из данных вариантов существует два режима перехода в «сон»: *Sleep now* (переход непосредственно по получении команды *WFI* или *WFE*) и *Sleep-on-exit* (переход по завершении обработки всех ранее полученных запросов на прерывания), в зависимости от состояния бита *SLEEPONEXIT* регистра *SCR* (*System Control Register*).

<sup>2)</sup> При условии, что перед переходом в ЭСП установлен в единицу бит *LPDS* регистра управления питанием (*PWR\_CR*). У ряда моделей семейства, в т. ч. K1986BE92F1I, данная опция отсутствует.

<sup>3)</sup> *SLEEPDEEP* – состояние бита *SLEEPDEEP* регистра *SCR* (см. сноску<sup>1)</sup>). После сброса равно нулю. Для перехода в режим останова или ожидания должно быть установлено в единицу перед командой *WFI* или *WFE*.

<sup>4)</sup> *PDDS* – состояние бита *PDDS* (*Power Down Deepsleep*) регистра управления питанием (*PWR\_CR*). У ряда моделей семейства, в т. ч. K1986BE92F1I, данный бит отсутствует; для выбора режима останова необходимо только установить в единицу бит *SLEEPDEEP* перед командой *WFI* или *WFE*.

<sup>5)</sup> С уровнем приоритета выше, чем указанный в регистре *BASEPRI* (см. рис. 2.30 и пояснения к нему).

<sup>6)</sup> См. пояснения в тексте.

<sup>7)</sup> В зависимости от того, какой командой осуществлялся переход в режим останова, *WFI* или *WFE*.

<sup>8)</sup> См. также пункт 7.3.2.

<sup>9)</sup> В ряде моделей семейства, в т. ч. K1986BE92F1I – все, кроме входящих в состав ДПП, а также генератора *LSI* [11].

### Окончание таблицы 3.2

1	2	3	4	5	6
<p><sup>10)</sup> При этом теряются все данные, хранящиеся в регистрах СОЗУ (кроме регистров ДРП) и в резидентном статическом ОЗУ данных. После выхода из режима ожидания программа МК запускается с начального адреса, как при сбросе.</p> <p><sup>11)</sup> <i>WUF</i> – <i>Wakeup Flag</i> (бит признака «пробуждения») в регистре состояния подсистемы питания (<i>PWR_CSR</i>). Перед командой <i>WFI</i> или <i>WFE</i> должен быть сброшен в ноль записью единицы в бит <i>CWUF</i> (<i>Clear Wakeup Flag</i>) регистра управления питанием (<i>PWR_CR</i>).</p> <p><sup>12)</sup> Ряд моделей семейства переводится режим ожидания установкой в единицу бита <i>STANDBY</i> в одном из регистров управления ДРП, например, в МК К1986ВЕ92F1I – в регистре <i>BKP_REG_0x0F</i> [11] (см. рис. 3.10).</p> <p><sup>13)</sup> <i>NRST</i> – вывод БИС МК для подачи внешнего сигнала сброса; у ряда моделей МК может быть обозначен как <i>RESET</i> (см. рис. 5.4).</p> <p><sup>14)</sup> <i>IWDT</i> – независимый сторожевой таймер МК (см. пункт 5.4.4).</p> <p><sup>15)</sup> <i>WKUPx</i> – входы БИС МК, которые программно могут быть сконфигурированы для подачи внешнего сигнала «пробуждения». У ряда моделей семейства, в т. ч. у К1986ВЕ92F1I, имеется только один вход <i>WKUP</i> (см. рис. 3.10).</p> <p><sup>16)</sup> <i>ALARM RTC</i> – совпадение содержимых счетчика <i>RTC_CNT</i> и регистра <i>RTC_ALR</i> (см. сигнал <i>ALRF</i> на рис. 3.10).</p> <p><sup>17)</sup> В моделях МК, у которых в режиме ожидания отключается тактирование всех узлов и блоков, кроме входящих в состав ДРП (в т. ч. в К1986ВЕ92F1I) выход из режима ожидания осуществляется только сигналом <i>WKUP</i> или по событию <i>ALARM RTC</i> (см. рис. 3.10).</p>					

Более подробно необходимо остановиться на специфике команд *WFI* (*Wait For Interrupt*, «Ожидание прерывания») и *WFE* (*Wait For Event*, «Ожидание события»). Первая из них переводит МК в ЭСР до возникновения **прерывания**, вторая – до возникновения **события**. Понятия прерывания и события рассмотрены в разделе 7, посвященном типовым структурно-архитектурным решениям подсистемы обработки прерываний. Однако, для понимания особенностей ЭСР МК семейства *ARM Cortex-Mx*, вкратце смысл понятий прерывания и события необходимо рассмотреть в данном разделе.

Напомним, что **прерыванием** называют приостановку выполнения основной программы по запросу от какого-либо ПУ или от ядра МК, с выполнением подпрограммы обслуживания соответствующего запроса [3, 9].

В свою очередь, **событием** называют некоторое явление, имевшее место в каком-либо функциональном узле или блоке МК, распознаваемое ЦП и / или ПУ МК и могущее вызвать определенные действия с их стороны, в том числе генерацию запроса на прерывание [9]. Например, перезагрузка счетчика некоторого таймера МК (допустим, 3-го) – это событие, которое, при определенных условиях, может вызвать прерывание по перезагрузке 3-го счетчика.

При этом:

- прерывания без событий невозможны; именно события инициируют запросы на прерывания (при условии, что формирование соответствующего запроса не запрещено, см. подраздел 7.1);

- события без прерываний (в том числе при запрете прерываний по соответствующим событиям) вполне возможны, и могут использоваться для управления узлами и блоками МК, в частности, для вывода МК из ЭСР (см. далее);

- генерация запросов на обслуживание ряда событий (например, изменения уровня сигнала на линиях запросов от внешних источников, *EXTI*) может быть программно запрещена или разрешена (подробнее – см. пункт 7.3.2).

При переводе МК в ЭСР «*Sleep*» командой *WFI* выход из ЭСР осуществляется запросами на прерывание, обслуживание которых разрешено (см. также сноску<sup>5</sup>) в табл. 3.2).

Если перевод МК в ЭСР «*Sleep*» реализуется командой *WFE*, то, в зависимости от состояния бита *SEVEONPEND* (*Send Event on Pending bit*) регистра *SCR* (см. сноску<sup>1)</sup> в табл. 3.2), вывод МК из ЭСР осуществляется:

- при нулевом состоянии бита *SEVEONPEND* – только прерываниями и событиями, обслуживание которых разрешено;
- при единичном состоянии бита *SEVEONPEND* – всеми разрешенными для обслуживания событиями и всеми прерываниями, как разрешенными, так запрещенными для обслуживания.

Вышесказанное справедливо и для режима «Останов», за исключением того, что в данном режиме «пробуждающим» действием обладают только прерывания и события, генерируемые по линиям внешних запросов (*EXTI*).

Необходимо также отметить, что при генерации какого-либо события устанавливается в единичное состояние определенный бит в регистре событий (программно не доступном). Если при вызове команды *WFE* этот бит установлен, ЦП «посчитает», что событие пробуждения уже произошло, сбросит данный бит и выйдет из ЭСР. Нормальный вход в ЭСР по команде *WFE* возможен только при нулевом состоянии указанного бита. Поэтому для гарантированного перехода в ЭСР следует вызывать команду *WFE* в цикле [9].

На основе табл. 3.2 нетрудно сделать вывод, что из 3-х ЭСР, реализуемых в МК семейства *ARM Cortex-Mx*, наименьшим энергопотреблением характеризуется режим ожидания, наибольшим – режим «Сон». Например, ток, потребляемый МК модельного ряда *STM32F103xx* в режиме ожидания – не более 5 мкА, в режиме останова – не более 370 мкА, в режиме «Сон» - не более 32 мА, в нормальном (рабочем) режиме, при тактовой частоте ЦП 72 МГц – не более 50 мА [10]. С другой стороны, верна закономерность, характерная для всех семейств МК (см. пункт 3.4.1), в том числе для семейства *AVR* (см. табл. 3.1): чем меньшим энергопотреблением характеризуется какой-либо ЭСР МК, тем длительнее процесс его «пробуждения» и тем меньше событий могут его инициировать.

Перед переводом в ЭСР энергосбережение МК семейства *ARM Cortex-Mx*, как и любого другого семейства, рекомендуется минимизировать посредством [9]:

- запрета работы (в том числе тактирования) узлов и блоков МК, активное состояние которых не требуется для функционирования МК в выбранном ЭСР и для выхода из него;

- установки минимально необходимой частоты тактирования узлов и блоков МК, остающихся в активном состоянии в выбранном ЭСР.

Особенности минимизации энергопотребления конкретных моделей МК семейства *ARM Cortex-Mx* отражены в их технических описаниях и руководствах по применению.

### **3.5 Выводы по разделу 3**

**3.5.1.** Подсистема питания входит в состав функциональных блоков / подсистем МК, минимально необходимых для его практического применения.

**3.5.2.** Основными элементами архитектуры подсистемы питания МК являются:

- состав напряжений питания и схемы их подключения;
- допустимые уровни входных сигналов МК в зависимости от напряжений питания;
- реализация корректного запуска МК при включении питания и восстановления работы МК при сбоях по питанию, в том числе после его отключения (аварийного или преднамеренного);
- способы минимизации энергопотребления и реализация энергосберегающих режимов работы МК.

**3.5.3.** Для современных МК общего назначения характерно питание от одного источника напряжения, типовое значение которого – 3 или 3,3 В. Наблюдается тенденция к снижению напряжения питания МК, в частности, выпускаются модели МК с номинальным значением напряжения питания, равным 1.6 В. Преобладавшее в прошлом напряжение питания 5 В в настоящее время используется, в основном, в моделях МК относительно давней разработки.

**3.5.4.** При наличии в структуре МК аналоговых и / или аналого-цифровых функциональных блоков (АЦП, ЦАП, компараторов) подключение аналоговой и цифровой части МК к источнику питания должно осуществляться отдельными парами проводников (см. рис. 3.3 и 3.5, а также комментарии к рис. 3.2). Для этого в структуре МК

предусматриваются отдельные общие шины и шины питания аналоговых и цифровых функциональных блоков, а также отдельные выводы БИС для подключения указанных шин к источнику питания (см. рис. 3.2). В ряде моделей МК класса «*high-performance*» предусмотрена отдельная подача напряжения питания на аналоговую, аналого-цифровую и цифровую часть МК (см. рис. 3.5).

**3.5.5.** Необходимо подключение блокировочных конденсаторов между выводами питания и общей шиной (см. рис. 3.1, 3.3 и 3.5), осуществляющих фильтрацию (сглаживание) бросков и помех по питанию; конденсаторы должны устанавливаться на минимально возможном расстоянии от вывода питания и общего вывода МК, а их номиналы оговариваются технической документацией на МК .

**3.5.6.** Ряд подсемейств / моделей МК требует применения дополнительных источников напряжения (как правило, более высокого, чем напряжение питания МК) при программировании энергонезависимых модулей памяти, что будет рассмотрено в разделе 12, посвященном программированию памяти МК

**3.5.7.** КМОП-технология, по которой реализуются практически все современные МК общего назначения, обуславливает следующие основные требования к входным напряжениям БИС МК и следующие основные характеристики выходных напряжений:

- допустимый диапазон входных напряжений МК – от нуля до напряжения питания; исключения составляют выводы со статусом «*5 V tolerant I/O*»; на них могут подаваться выходные сигналы ИС и БИС с напряжением питания, равным 5 В при минимально допустимом входном напряжении также равном нулю;

- максимально допустимое значение опорного напряжения встроенных АЦП и ЦАП МК равно напряжению питания аналоговой части МК;

- подача входного напряжения на БИС МК при отключенном питании **недопустима**;

- при использовании вывода КМОП-МК в качестве цифрового входа, уровни напряжения от нуля до примерно  $U_{cc}/3$  воспринимаются как логический ноль, а от примерно  $2U_{cc}/3$  до  $U_{cc}$  – как логическая единица (где  $U_{cc}$  – напряжение питания); конкретные значения пороговых значений входных напряжений логического нуля и логической единицы указываются в технической документации на соответствующие модели МК;

- при использовании вывода КМОП-МК в качестве цифрового выхода, максимальное напряжение логического нуля и минимальное напряжение логической единицы равны примерно  $0,1U_{cc}$  и  $0,9U_{cc}$  соответственно; при отсутствии перегрузки выхода они примерно равны нулю и  $U_{cc}$  соответственно.

**3.5.8.** Для корректного запуска МК по включении питания, корректного перезапуска МК при выходе из энергосберегающих режимов и при сбоях по питанию, без потерь программного кода и наиболее важных данных применяются следующие структурно-архитектурные решения:

- автоматический общий сброс МК при включении питания и при уменьшении напряжения питания ниже минимально допустимого уровня (см. рис. 3.7 – 3.9); при отсутствии внутренней цепи автоматического сброса МК необходимо использовать внешнюю цепь (см., например, рис. 3.6);

- использование энергонезависимых (и / или питаемых от резервных источников) модулей памяти МК для хранения кодов программ, а также не подлежащих потере данных;

- автоматический перевод группы «жизненно важных» функциональных узлов МК на питание от резервного источника при отключении основного источника питания или при уменьшении его напряжения до уровня ниже минимально допустимого (см. рис. 3.5).

**3.5.9.** Наличие домена с резервированием питания (ДРП, *Backup domain* на рис. 3.5) в структуре МК является эффективным средством поддержания его работоспособности и сохранения данных, не подлежащих потере, в энергосберегающих режимах, а также при аварийном или преднамеренном отключении основного источника питания. ДРП характерен, в основном, для МК класса «*high-performance*», например, для ряда подсемейств *ARM Cortex-Mx* [11, 13].

Основными функциями ДРП являются:

- сохранение данных, потеря которых недопустима (см. подпункт 3.3.3.2);

- обеспечение бесперебойного отсчета времени, т. е. непрерывной работы «часов» и «календаря» МК (см. подпункты 3.3.3.3 – 3.3.3.8).

**3.5.10.** Минимизация энергопотребления МК, реализованных по технологии КМОП, может быть достигнута следующими основными способами:

- снижение тактовых частот как ядра МК, так и задействованных периферийных устройств, до значений, допустимых с точки зрения выполняемых задач;

- запрет работы, в том числе тактирования, функциональных узлов и блоков МК, не задействованных при выполнении реализуемой задачи или на некоторых ее этапах;

- перевод МК в какой-либо из ЭСР (см. табл. 3.1 и 3.2 и комментарии к ним) на этапах выполнения реализуемого алгоритма, на которых возможна работа МК в соответствующем режиме.

**3.5.11.** Все ЭСР всех семейств МК характеризуются одним общим свойством – отключением тактирования основного потребителя мощности – ЦП, его остановкой и переводом в режим минимального энергопотребления. В большинстве ЭСР отключается тактирование и ряда других блоков / подсистем МК, за исключением тех, функционирование которых минимально необходимо для поддержания работоспособности МК в соответствующем ЭСР и для вывода МК из него. При этом питание ЦП, памяти и ПУ МК, как правило, не отключается, а данные, хранящиеся в энергозависимой памяти МК, не теряются. Исключением являются режимы «глубокого сна» некоторых семейств МК, например, режим ожидания МК семейства *ARM Cortex-Mx* (см. табл. 3.2). Различные ЭСР отличаются между собой составом функциональных блоков МК, тактирование которых отключается, а также составом событий, инициирующих «пробуждение» МК (см. табл. 3.1 и 3.2). Чем большее число функциональных блоков МК находится в статическом режиме, тем меньше энергопотребление МК, но тем длительнее процесс его «пробуждения» и тем меньше событий могут его инициировать.

**3.5.12.** Описанные в разделе 3 примеры структурно-архитектурных решений подсистемы питания являются **типовыми** для большинства современных семейств МК общего назначения.