

Е.М. Кравцунов, М.С. Михайлов, д.т.н., проф. С.В. Семенихин

(ЗАО «МЦСТ», ПАО «ИНЭУМ им. И.С. Брука)

E. Kravtsunov, M. Mikhailov, S. Semenikhin

**ИСПОЛЬЗОВАНИЕ ПРЕРЫВАНИЙ СИСТЕМНОГО КОНТРОЛЯ SCI ДЛЯ
УПРАВЛЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЕМ МИКРОПРОЦЕССОРОВ
СЕМЕЙСТВА «ЭЛЬБРУС»**

**APPLICATION OF SYSTEM CONTROL INTERRUPTS FOR ELBRUS
MICROPROCESSOR POWER MANAGEMENT**

Описан контроллер SPMC в составе разработанной в ЗАО «МЦСТ» микросхемы КПИИ-2. Предложен метод его использования для формирования прерываний системного контроля (SCI) с целью отслеживания событий энергопотребления и управления ими из операционной системы.

The article describes System Power Management Controller (SPMC), developed in ZAO «MCST». Authors suggest a software monitoring method for power consumption events, that take place in Elbrus microprocessor, and a special algorithm for handling such events in operating system.

Ключевые слова: управление энергопотреблением, прерывания SCI, ACPI, Эльбрус.

Keywords: Elbrus power management, SCI, ACPI, SPMC.

Введение

Характеристики энергопотребления микропроцессоров семейства «Эльбрус» являются одним из важнейших факторов при создании вычислительных комплексов (ВК) на их основе и имеют решающее значение при проектировании бортовой аппаратуры. Так, например, для соответствия климатическим требованиям, предъявляемым к ней в

ответственных применениях, микропроцессор должен сохранять работоспособность при постоянном воздействии температуры 80°C и кратковременном воздействии температуры 95°C в условиях кондуктивного охлаждения, когда в первичном контуре охлаждения не допускается установка вентиляторов, а используются лишь радиаторы. Для бортовых ВК на основе микропроцессоров семейства «Эльбрус» выполнение таких требований возможно только при управлении энергопотреблением из операционной системы (ОС) с использованием специальной аппаратной поддержки.

Это стало целью ряда проектов, в результате которых в работу микропроцессора были введены две функциональности. Одна из них обеспечивала постоянный мониторинг влияющих на энергопотребление состояний аппаратуры и переключение между ними, а вторая – непосредственное отслеживание состояний энергопотребления, требующих или позволяющих снизить его уровень (долгий простой процессорных ядер, переключение системы питания от сети на аккумулятор, недопустимое снижение заряда аккумулятора).

Ранее в рамках первой функциональности был разработан алгоритм логического и физического отключения ядер микропроцессора, и на макете микропроцессора реализованы переключение ядра из состояния простоя в состояние сна, отключение дешифрации команд, управление частотой и отключение и синхроимпульса на ядрах [1]. В статье приводятся результаты проектов, выполненных в течение прошлого года для обеспечения второй функциональности в части ее аппаратных средств (раздел 1) и программной реализации в ОС (раздел 2).

1. Отслеживание событий энергопотребления в аппаратуре

Для реализации функции отслеживания событий энергопотребления в составе контроллера периферийных интерфейсов КПИ-2 был спроектирован контроллер энергопотребления SPMC (System Power Management Controller), входные и выходные сигналы которого выводятся на корпус микросхемы КПИ-2 и ее плату. Линия прерываний

от SPMC, отнесенных к прерываниям системного управления SCI (system control interrupt), внутри этой микросхемы соединяется с контроллером IOAPIC, общим для всех устройств в ее составе.

Контроллер SPMC состоит из следующих логических блоков, связывающих сигналы с программно-доступными регистрами управления:

1) таймер PM (Power management) – устройство, отслеживающее событие длительного простоя. Для совместимости со стандартом ACPI [2] таймер PM работает на частоте 4,9152 МГц;

2) блок обработки события «Нажатие кнопки управления питанием»;

3) блок обработки события «Переключение источника питания с сети переменного тока на аккумулятор и обратно»;

4) блок обработки события «Недостаточный уровень заряда батареи»;

5) блок обработки события Wake up event;

6) блок связи управляющих регистров с сигналами типа Output.

Блоки 1–5 реализуют логику отслеживания соответствующих событий и генерации прерываний SCI в зависимости от значений программно-доступных регистров управления SPMC. Сигналы SPMC описаны в табл. 1.

Таблица 1

Физический интерфейс SPMC. Сигналы типа Input/Output

Название и тип сигнала	Назначение сигнала	Комментарий
sus_clk Input	Сигнал установлен в 1, если на таймер PM подается частота 4,9152 МГц	Источником частоты таймера PM является сигнал синхронизации, общий для SPMC и последовательных портов
pwr_btn_ Input	Сигнал предназначен для реализации кнопки управления питанием. Он используется с целью перевода микропроцессора из состояния нормального энергопотребления (G0) в состояние пониженного энергопотребления	Кратковременное нажатие кнопки управления питанием в состоянии G0 приводит к формированию прерывания SCI, по которому ОС может инициировать переход в состояние G1. Если данная кнопка нажата в течение 4 с, то питание ВК

	(G1) путем уменьшения тактовой частоты	автоматически выключается
ac_power_psnt Input	Сигнал присутствует, если питание системы осуществляется от сети переменного тока, и сбрасывается (устанавливается в 0), если источником питания является аккумулятор	Состояние сигнала отображается в программно-доступном регистре и может быть использовано программным обеспечением для перевода микропроцессора в состояние G1. При изменении сигнала в состоянии G0 формируется прерывание SCI
batlow_ Input	Сигнал является индикатором заряда батареи. Он сбрасывается при недостаточном уровне заряда аккумулятора	Значение сигнала отображается в программно-доступном регистре. Его изменение в состоянии G0 приводит к формированию прерывания SCI, что может быть использовано программным обеспечением для перевода микропроцессора в состояние G1. При недостаточном уровне заряда аккумулятора сигнал препятствует переводу микропроцессора в активное состояние G0
wake_up_ Input	Сигнал свидетельствует о событии wake up event, которое является запросом на переход микропроцессора из состояния G1 в состояние G0	Подача сигнала вызывает переход микропроцессора из состояния G1 в состояние G0 с выработкой прерывания SCI. Состояние сигнала отображается в программно-доступном регистре
sleep_s3_ Output	Сигнал переводит микропроцессор из состояния G0 в состояние G1 с минимальной глубиной сна S3	Биты управления сигналом содержатся в программно-доступном регистре
sleep_s4_ Output	Сигнал переводит микропроцессор из состояния G0 в состояние G1 со средней глубиной сна S4	Биты управления сигналом содержатся в программно-доступном регистре
sleep_s5_ Output	Сигнал переводит микропроцессор из состояния G0 в состояние G1 с максимальной глубиной сна S5	Биты управления сигналом содержатся в программно-доступном регистре

Программно-доступные регистры управления SPMC, управляющие таймером PM, событиями питания, нажатием кнопки и генерацией прерывания SCI, находятся в конфигурационном пространстве PCI контроллера SPMC. Разрядность регистров – 32 бита. Часть регистров имеет лишь 16 значащих битов, расположенных в младших разрядах – это регистры ACPI, описанные в стандарте [2]. Представленный в SPMC набор

регистров ACPI позволяет реализовать самый простой вариант управления энергопотреблением – ACPI Fixed [2]. Карта программно-доступных регистров приведена в табл. 2.

Регистр PM_TMR содержит текущее значение счетчика в таймере PM, он доступен для чтения драйвером SPMC. В регистре PM1_STS отображается статус работы блоков, его биты доступны драйверу SPMC для чтения. Программная запись 1 в эти биты приводит к их обнулению. Статус конкретных логических блоков устройства SPMC задается в битах PM1_STS, которые соответствуют номерам битов активности этих блоков в регистре PM1_EN. В регистре PM1_EN предусмотрен отдельный бит для каждого блока устройства SPMC. Эти биты доступны его драйверу для записи и служат для включения/выключения активности блоков. Регистр PM1_CNT предназначен для глобального включения/выключения режима SCI прерываний и управления сигналами типа Output, с помощью которых микропроцессор переводится из состояния G0 в состояние G1 и обратно.

Таблица 2

Программно-доступные регистры SPMC

№	Старшие 16 битов (31 - 16)	Младшие 16 битов (15 - 0)	Смещение относительно начала конфигурационного пространства PCI
1	Device ID (800Eh)	Vendor ID (1FFFh)	0x00
2	PM TMR [31:16]	PM TMR [15:0]	0x40 (ACPI)
3	Резерв	PM1_STS	0x44 (ACPI)
4	Резерв	PM1_EN	0x48 (ACPI)
5	Резерв	PM1_CNT	0x4C (ACPI)

Табл. 3, 4, 5 содержат описание битов регистров PM1_EN, PM1_STS, PM1_CNT соответственно. В поле «тип бита» используются следующие значения: R/W1C – бит доступен по чтению, запись в этот бит приводит к его обнулению; RO – бит доступен

только по чтению; WO – бит доступен только по записи; RW – бит доступен как по чтению, так и по записи.

Таблица 3

Регистр PM1_EN

Бит	Тип бита	Название	Описание
0	WO	TMR_EN	Если бит в состоянии 1, то при установке в 1 бита TMR_STS регистра PM1_STS вырабатывается прерывание SCI
1	RW	TMR_32	Бит для переключения разрядности таймера PM: 0 – 24 разряда; 1 – 32 разряда
2	WO	AC_PWR_EN	Если бит в состоянии 1, то при установке в 1 бита AC_PWR_STS регистра PM1_STS вырабатывается прерывание SCI
3	Резерв		
4	WO	BATLOW_EN	Если бит в состоянии 1, то при установке в 1 бита BATLOW_STS регистра PM1_STS вырабатывается прерывание SCI
7...5	Резерв		
8	WO	PWRBTN_EN	Если бит в состоянии 1, то при установке в 1 бита PWRBTN_STS регистра PM1_STS вырабатывается прерывание SCI
15...9	Резерв		

Таблица 4

Регистр PM1_STS

Бит	Тип бита	Название	Описание
0	R/W1C	TMR_STS	Устанавливается в 1 при переключении старшего бита PM_TMR (24-го или 31-го в зависимости от бита 1 регистра PM1_EN), из 0 в 1 или из 1 в 0. Если бит TMR_EN регистра PM1_EN тоже установлен, то вырабатывается внешнее прерывание SCI. Обработчик прерывания SCI гасит бит записью в него 1
1	RO	AC_PWR_STATE	Показывает состояние входа ac_power_psnt: питание от сети – 1, от батареи – 0
2	R/W1C	AC_PWR_STS	Устанавливается в 1 при любом изменении сигнала ac_power_psnt. Если бит AC_PWR_EN регистра PM1_EN тоже установлен, то возникает внешнее прерывание SCI. Обработчик прерывания SCI гасит бит записью в него 1
3	RO	BATLOW_STATE	Инверсия входного сигнала batlow: заряд батареи недостаточен – 1, в норме – 0
4	R/W1C	BATLOW_STS	Устанавливается в 1 при любом изменении сигнала ac_power_psnt. Если бит AC_PWR_EN регистра PM1_EN тоже взведен, то вырабатывается внешнее

			прерывание SCI. Обработчик прерывания SCI гасит бит записью в него 1
7...5	Резерв		
8	R/W1C	PWRBTN_STS	Состояние кнопки управления питанием. Бит устанавливается в 1 аппаратурой, если кнопка нажата. Для системы в состоянии G0: – если бит PWRBTN_STS и бит PWRBTN_EN регистра PM1_EN установлены в 1, то возникает прерывание SCI; – если кнопка находится в нажатом состоянии больше 4 с, то бит аппаратно гасится, для его повторной установки требуется, чтобы кнопка была отпущена, а потом нажата снова. Для системы в состоянии G1: – если PWRBTN_STS установлен в 1, то бит WAK_STS также устанавливается в 1. Бит гасится записью в него 1
14...9	Резерв		
15	R/W1C	WAK_STS	Бит устанавливается в 1 аппаратурой, когда система находится в состоянии G1 и происходит одно из событий wake up или нажатие кнопки управления питанием. При установке бита возникает прерывание SCI. Бит гасится записью в него 1

Таблица 5

Регистр PM1_CNT

Бит	Тип бита	Название	Описание
0	RW	SCI_EN	Бит осуществляет программное переключение режимов Legacy/ ACPI: 0 – режим Legacy (функции PMC не используются, записи в регистры PMC ни к чему не приводят); 1 – режим ACPI (функции PMC используются, записи в регистры PMC приводят к предусмотренным результатам). Значение этого бита выставляется ядром ОС при инициализации
9...11	Резерв		
12...10	RW	SLP_TYPx	3-битовое поле, определяющее состояние сна: 0x0 – S0 (G0) 0x1 – S0 (G0) 0x2 – S0 (G0) 0x3 – S3 (G1) 0x4 – S4 (G1) 0x5 – S5 (G1) 0x6 – S0 (G0) 0x7 – S0 (G0)
13	Write Only	SLP_EN	Установка бита в 1 приводит к переводу микропроцессора к состоянию, определенному в SLP_TYPx

Функционирование таймера PM с использованием программно-доступных регистров определено следующим образом.

Таймер инициализируется нулем при включении ВК и не обнуляется в процессе его работы. Таймер устанавливает в 1 бит TMR_STS регистра PM1_STS при переключении значения старшего бита таймера с 1 на 0 и с 0 на 1 (в зависимости от разрядности таймера это или 23-й, или 31-й бит). Если при этом бит TMR_EN регистра PM1_EN установлен в 1, то вырабатывается внешнее прерывание SCI. Таймер реализован с возможностью переключения разрядности из ядра ОС. Для этого используется бит TMR_32 регистра PM1_EN.

Помимо таймера PM, отслеживающего события длительного простоя, источником SCI прерывания могут быть другие логические блоки SPMC. Общее условие формирования SCI прерывания контроллером SPMC с учетом состояния всех логических блоков задается следующим образом:

```
SCI = SCI_EN & (
    (TMR_EN & TMR_STS) |
    (AC_PWR_EN & AC_PWR_STS) |
    (BATLOW_EN & BATLOW_STS) |
    (G0 & PWRBTN_STS & PWRBTN_EN) |
    (G1 & WAK_STS)
);
```

Для перевода микропроцессора из состояния G0 в состояние G1 и обратно используются бит SLP_TYPx и бит SLP_EN регистра PM1_CNT. Важно отметить, что записи в биты регистров ACPI (PM1_EN, PM1_STS, PM1_CNT) дают результат только в том случае, когда бит SCI_EN регистра PM1_CNT установлен драйвером устройства SPMC в 1.

3. Программная поддержка функций контроллера SPMC на уровне ядра ОС

Поддержка SPMC в ядре ОС состоит из драйвера этого контроллера и доработок, введенных в реализацию процесса idle.

Драйвер SPMC предоставляет интерфейсы взаимодействия с контроллером программы daemon из пространства пользователя, циклически опрашивающей устройства В/В. При инициализации драйвера выполняются четыре основных действия:

1) поиск устройства SPMC, при нахождении которого на шине PCI драйвер инициализирует для него ее конфигурационное пространство. Это позволяет программному обеспечению в дальнейшем иметь доступ к регистрам контроллера путем операций чтения/записи программно-доступных регистров конфигурационного пространства;

2) инициализация дескриптора прерывания SCI и регистрация обработчика этого прерывания в ядре ОС. При этом драйвер выделяет память для двух структур, необходимых при передаче дополнительных данных: служебной структуры, через которую информация о базовом адресе конфигурационного пространства PCI передается в обработчик прерывания, и структуры, через которую в пространство пользователя передается аргумент прерывания SCI, описывающий тип события;

3) создание и инициализация интерфейсов управления контроллером SPMC, которые предназначены для взаимодействия с программой daemon, принимающей решения о переводе микропроцессора в различные состояния энергопотребления. Интерфейсы располагаются в файловой системе sysfs и представляют собой файлы с правами чтения или записи, выставляемыми в зависимости от типа соответствующего бита (табл. 3, 4, 5).

4) создание и инициализация интерфейса, через который программа пространства пользователя информируется о событии энергопотребления, на которое следует реагировать. Такой интерфейс реализуется с помощью подсистемы input layer

ядра ОСРВ Эльбрус, основанного на ядре Linux-2.6.33. Интерфейс представляет собой файл в каталоге `/dev/input/`, для которого реализована файловая операция `poll`.

После завершения инициализации драйвера пользовательским процессам становятся доступны интерфейсы программного управления SPMC.

Доработка процесса `idle` предназначена для корректного отслеживания длительного простоя микропроцессора. С этой целью процесс `idle`, попадая на процессорное ядро, включает таймер PM путем установки в 1 бита `TMR_EN` регистра `PM1_TMR`. При уходе этого процесса с ядра он своим последним действием отключает таймер PM. Если время, в течение которого процесс `idle` находится на процессорном ядре, превышает время переполнения счетчика в таймере PM, то таймер срабатывает, и контроллер SPMC вырабатывает прерывание SCI, подлежащее обработке его драйвером.

Приведенные выше действия рассчитаны на одноядерный микропроцессор, в то время как многоядерность усложняет доработку процесса `idle`. В N -ядерном микропроцессоре длительный простой рассматривается как событие в том случае, когда процессы `idle` выполняются на всех N ядрах в течение длительного времени. Для отслеживания такого события в ядре ОС вводится дополнительная глобальная переменная – маска простоя ядер, единица в i -ом разряде которой означает, что в момент чтения маски на i -ом ядре микропроцессора выполняется процесс `idle`.

Таким образом, применительно к многоядерному микропроцессору функции `idle` меняются следующим образом. При попадании на i -ое ядро микропроцессора процесс `idle` устанавливает 1 в i -ый бит маски простоя ядер, считывает результат обновления маски и включает таймер PM только при условии, что в маске простоя все значимые биты находятся в состоянии 1. При уходе с процессорного ядра процесс `idle` обнуляет соответствующий номеру процессорного ядра бит в маске простоя и отключает таймер PM. В ядре ОС маска простоя является критической секцией и защищена от

одновременного доступа механизмом синхронизации `spin_lock_irqsave`.

Описанные аппаратная и программная поддержка прерываний SCI были использованы для создания программного метода управления энергопотреблением из операционной системы.

4. Метод управления энергопотреблением из ОС с использованием SCI

Метод реализован в виде пользовательской задачи `daemon`, в фоновом режиме выполняющей опрос интерфейсов SPMC, предоставляемых ядром ОС. При получении прерывания `daemon` анализирует поступивший с ним аргумент и в зависимости от его значения переводит микропроцессор в состояние пониженного энергопотребления или выключения части процессорных ядер с целью пассивного охлаждения. Эффект от первого выбора зависит от наличия соответствующей аппаратной поддержки в микропроцессоре. Варианты реакции на SCI-прерывание для различных моделей микропроцессоров Эльбрус приведены в табл. 6.

Таблица 6

Реакция на SCI – прерывание для различных моделей микропроцессора

Микропроцессор	Архитектура	Возможные реакции на SCI-прерывание
Эльбрус-2С+	Эльбрус	Выключение одного процессорного ядра микропроцессора путем отключения синхроимпульса
Эльбрус-4С	Эльбрус	1. Выключение от одного до трех процессорных ядер. 2. Перевод всех ядер микропроцессора в состояние глубокого сна путем отключения синхроимпульса. Выход из состояния глубокого сна происходит при приходе внешнего прерывания
МЦСТ R1000	Sparc V9	Выключение от одного до трех процессорных ядер микропроцессора путем отключения синхроимпульса
Процессор-2	Эльбрус	1. Снижение частоты синхроимпульса до минимально возможной. 2. Перевод ядра микропроцессора в состояние глубокого сна путем отключения синхроимпульса. Выход из состояния глубокого сна происходит при приходе внешнего прерывания. 3. Комбинация вариантов 1 и 2

Заключение

Данная статья заключает серию публикаций [1, 3, 4], посвященных исследованиям и разработкам в области управления энергопотреблением микропроцессоров семейства «Эльбрус» из операционной системы. Эти результаты позволяют создать систему адаптивного управления энергопотреблением в перспективных вычислительных комплексах на основе микропроцессоров типа «Процессор-2», микросхемы КПИ-2 и описанных в статье аппаратно-программных средств, которая отвечала бы современным требованиям к встраиваемым системам и условиям их промышленного выпуска.

Литература

1. Кравцунов Е.М., Семенихин С.В. Управление энергопотреблением системы на кристалле «Эльбрус-2С+» в состоянии простоя процессорного ядра. – «Вопросы радиоэлектроники», сер. ЭВТ, 2013, вып. 3.

2. Advanced Configuration and Power Interface Specification, Hewlett-Packard Corporation, Intel Corporation, Microsoft Corporation, Phoenix Technologies Ltd, Toshiba Corporation, Revision 4.0a, April 5, 2010.

3. Волин В.С., Кравцунов Е.М., Семенихин С.В., Фельдман В.М., Черепанов С.А. Управление энергопотреблением процессорных ядер из операционной системы для прототипа вычислительного комплекса на базе микропроцессоров семейства «Эльбрус» – «Вопросы радиоэлектроники», сер. ЭВТ, 2014, вып. 3.

4. Кравцунов Е.М. Пути реализации стандарта ACPI 4.0 (Advanced Configuration and Power Interface Specification) для многопроцессорных вычислительных комплексов на базе процессоров Эльбрус-E2S: Конференция молодых ученых и специалистов, посвященная 55-летию со дня образования НИИ автоматической аппаратуры им. В.С. Семенихина. – М.: НИИАА, 2011.