

16-битные микроконтроллеры Microchip

Александр БОРИСОВ
bor-alex@gamma.spb.ru

В 2004 году компания Microchip Technology Inc., известная, прежде всего, своими 8-битными микроконтроллерами PIC16 и PIC18, выпустила на рынок 16-битное семейство dsPIC30 с ядром цифровой обработки сигналов. Это был первый опыт компании в разработке 16-битного ядра, и нужно сказать, довольно успешный — цифровые сигнальные контроллеры dsPIC30 сразу завоевали нишу в таких областях, как управление двигателями, мощные источники питания, интеллектуальные датчики, счетчики энергии. Архитектура оказалась настолько удачной, что многие разработчики стали использовать линейку dsPIC30 в задачах, в которых не требуются алгоритмы ЦОС и в которых до этого применялись микроконтроллеры общего назначения.

В связи с этим компания Microchip приняла решение о расширении линейки 16-битных микроконтроллеров еще тремя семействами. Летом 2005 года были анонсированы dsPIC33, PIC24F и PIC24H, весной 2006-го началось их массовое производство. Следует заметить, что базовое семейство dsPIC30 постоянно расширяется, совсем недавно представлены новые микроконтроллеры с уникальным модулем Power PWM + SMPS.

Микроконтроллеры PIC24F/H не содержат DSP-ядра и позиционируются как контроллеры общего назначения, а семейство dsPIC33F является расширением dsPIC30. В таблице 1 приведены сводные характеристики всех семейств, основанных на 16-битной архитектуре Microchip.

Условно можно классифицировать 16-битные микроконтроллеры следующим образом: PIC24F является упрощенным вариантом dsPIC30 — отсутствует DSP-ядро, а PIC24H — упрощенным вариантом dsPIC33. Особое внимание следует обратить на тот факт, что все семейства основаны на одной архитектуре — единая система команд, способы адресации, карта памяти и т. д., а следовательно, взаимозаменяемы с точки зрения программного обеспечения.

Архитектура

Ядро dsPIC/PIC24 построено по модифицированной гарвардской архитектуре с рас-

ширенной системой команд. Архитектура dsPIC/PIC24 не является какой-либо модификацией привычных 8-битных ядер Microchip (Baseline, Midrange, PIC18) — это абсолютно новая разработка.

Микроконтроллеры dsPIC30/33 поддерживают выполнение специфических для алгоритмов цифровой обработки сигналов инструкций (умножение с накоплением), специальные методы адресации (модульная, бит-реверсивная). Все 16-битные контроллеры имеют векторную приоритетную систему прерываний, возможность отображения части программной памяти в нереализованную на кристалле область ОЗУ, знаковые вычисления с целыми числами и числами с фиксированной точкой. Более подробно особенности архитектуры 16-битных микроконтроллеров Microchip будут рассмотрены в этой части статьи.

В качестве примера рассмотрим структуру программной памяти, памяти данных, системы прерываний, АЛУ и набора инструкций микроконтроллеров dsPIC30, указав существенные отличия для других 16-битных семейств.

Организация памяти программ и памяти данных

Память программ. Карта памяти программ всех контроллеров dsPIC/PIC24 линейная и несегментированная. Все инструкции имеют фиксированную длину 24 бита; счетчик инструкций — 23-битный, младший бит

всегда равен 0 для обеспечения выравнивания данных при выборке инструкции. Таким образом, эффективное количество адресуемых инструкций равно $2^{22} \sim 4$ млн.

Карта программной памяти микроконтроллера dsPIC30F6014 изображена на рис. 1.

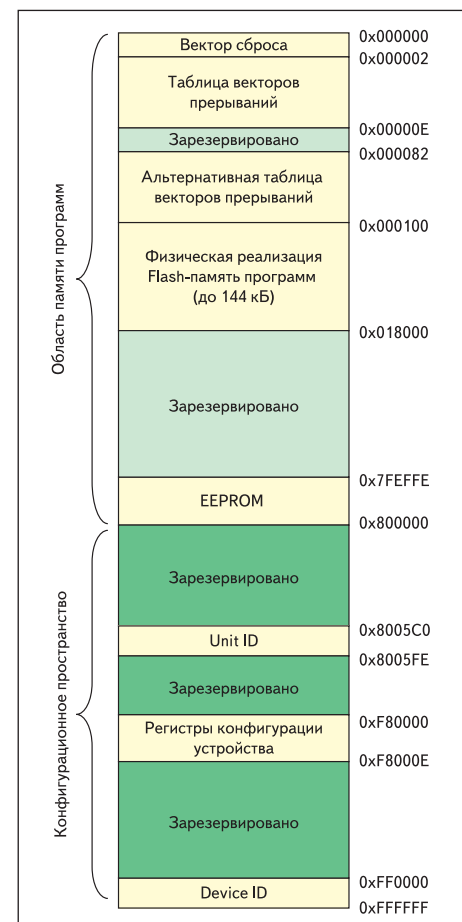


Рис. 1. Карта памяти программ микроконтроллера dsPIC30F6014 (не в реальном масштабе)

Таблица 1. Сводные характеристики микроконтроллеров Microchip с 16-битной архитектурой

Семейство	DSP ядро	Макс. производительность, MIPS	DMA	Макс. объем Flash-памяти, кбайт	Макс. объем ОЗУ, кбайт	EEPROM	Макс. кол-во выводов	Диапазон напряжений питания, В	JTAG
PIC24F	-	16	-	128	8	-	100	2,0-3,6	+
dsPIC30	+	30	-	144	8	+	80	2,5-5,5	-
PIC24H	-	40	+	256	16	-	100	3,0-3,3	+
dsPIC33	+	40	+	256	30	-	100	3,3-3,3	+

Адрес **0x00** является вектором сброса. Далее расположена основная и альтернативная таблица векторов прерываний — эти области программной памяти будут более подробно рассмотрены в соответствующем разделе статьи. Следующая область — область пользовательской программы, объем этого сектора зависит от объема интегрированной Flash-памяти.

Микроконтроллеры семейства dsPIC30 имеют интегрированную память EEPROM, которая отображена в соответствующем секторе программной памяти. Семейства dsPIC33, PIC24F/H не имеют EEPROM, что связано с изменением технологии изготовления кристаллов. Поэтому при необходимости наличия в системе энергонезависимой памяти малого объема с большим количеством циклов перепрограммирования рекомендуется использовать внешние микросхемы памяти EEPROM с последовательным интерфейсом.

Половину карты программной памяти занимает так называемое конфигурационное пространство. Во всех 16-битных микроконтроллерах физически реализована только небольшая часть данного сектора — это слова конфигурации, определяющие режим работы контроллера после сброса, и идентификационный код кристалла, предназначенный для определения контроллера аппаратными средствами разработки.

Физически программная память во всех контроллерах 16-битного семейства реализована в виде перепрограммируемой Flash-памяти. Все контроллеры поддерживают внутрисхемное программирование и программирование в ходе выполнения программы.

В ходе выполнения программы существует три способа доступа к программной памяти:

- Выборка инструкции в соответствии со значением командного счетчика — собственно само выполнение программы.
- Использование инструкций табличного чтения-записи, позволяющее получить доступ как к слову (16 бит), так и к байту программной памяти. Табличная запись осуществляется через буфер, не отображаемый в ОЗУ. Для контроллеров dsPIC30 минимальный объем записываемых данных равен 12 байтам (4 программных слова, 1/8 сектора Flash-памяти), для контроллеров dsPIC33 и PIC24F/H — 192 байта (64 программных слова — один сектор Flash-памяти).
- Чтение программной памяти с помощью механизма PSV (Program Space Visibility) — отображения сектора памяти программ в область ОЗУ. Механизм PSV позволяет отобразить любую часть программной памяти объемом 32 кбайт в верхнюю, не реализованную физически область ОЗУ (рис. 2). PSV предоставляет уникальную возможность обращаться к младшим 16 битам программного слова как к динамическим данным — любые инструкции, осуществляющие чтение из ОЗУ, могут использовать

программную память в качестве источника. Механизм PSV применяется, если алгоритм содержит большие массивы констант: текстовые строки, коэффициенты цифровых фильтров и т. п.

Существенное отличие DSC dsPIC30 от остальных 16-битных семейств (dsPIC33, PIC24F/H) заключается в технологии изготовления интегрированной Flash-памяти. Как следствие — различное количество циклов перепрограммирования. Для dsPIC30 оно составляет 100 тыс., для dsPIC33 и PIC24F/H — всего 1 тыс. Может показаться, что это серьезный недостаток новых контроллеров, однако, как показывает практика, 1 тыс. циклов перезаписи достаточно для большинства задач, в том числе и для реализации калибруемых устройств.

Различная технология изготовления кристалла определяет и различие в спецификации программирования — микроконтроллеры dsPIC33 и PIC24F/H не требуют подачи высокого напряжения (13 В) на кристалл во время программирования. Кроме того, значительно увеличилась скорость операций с памятью — для контроллеров с объемом Flash 64 кбайт полный цикл стирание/запись занимает менее 1 секунды.

Память данных. Шина адреса данных микроконтроллеров dsPIC30/33 и PIC24F/H позволяет адресовать до 64 кбайт памяти ОЗУ, которая физически выполнена в качестве статической памяти с возможностью байтового доступа. Почти все инструкции, работающие с ОЗУ, имеют спецификатор, указывающий, будет ли осуществляться доступ к слову (16 бит) или к байту. Карта памяти данных микроконтроллера dsPIC30F6014 приведена на рис. 2.

В начале ОЗУ находится область регистров специального назначения (SFR) объемом 2 кбайт. Внутри семейства все регистры SFR расположены статически по одним адресам.

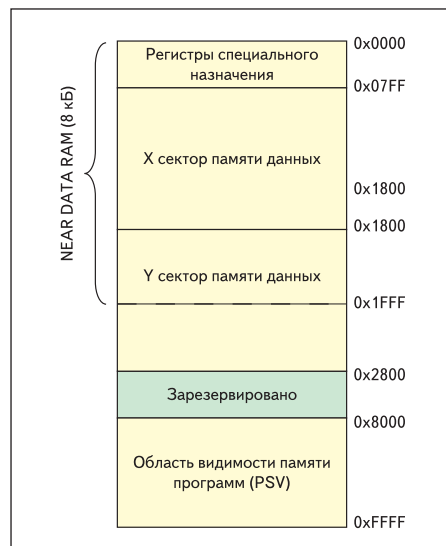


Рис. 2. Карта памяти данных микроконтроллера dsPIC30F6014 (не в реальном масштабе)

С адреса **0x800** начинается сектор ОЗУ общего назначения, максимальный объем которого составляет 30 кбайт; в разных контроллерах только часть этого сектора может быть реализована физически. Верхнюю половину ОЗУ занимает область, в которую отображается часть программной памяти при использовании механизма PSV.

В семействах с DSP-ядром (dsPIC30/33) реализовано два адресных генератора, что позволяет инструкциям DSP-ядра делать две выборки из ОЗУ за один командный такт. Это объясняет разделение области ОЗУ общего назначения на два сектора X и Y (рис. 2). Однако такое разделение не является сегментированием или ограничением — для всех инструкций CPU-ядра сектор ОЗУ общего назначения линейен.

Первые 8 кбайт ОЗУ (включая область SFR) называются пространством ближней памяти (Near Data Space). Прямая адресация возможна только к этому сегменту. Остальная часть ОЗУ может быть адресована косвенно.

16-битная архитектура Microchip предусматривает использование программного стека, указателем на который является один из рабочих (work) регистров. Стек растет с увеличением указателя, при этом возможен аппаратный контроль переполнения и опустошения стека. В случае переполнения или опустошения стека, ядром процессора генерируется аппаратное исключение — специальный вид прерывания, который будет рассмотрен в пункте «Система прерываний» данной статьи. Размер стека определяется программно.

При вызове подпрограммы в стек заносится адрес возврата и часть регистра статуса. Указатель на стек может быть изменен программно, что позволяет реализовывать гибкие системы реального времени.

В 16-битном ядре также присутствует LINK-регистр, который позволяет выделять в стеке фрейм для локальных переменных функций. Процедуры выделения и сброса фрейма выполняются программно за один командный такт.

АЛУ, умножитель, поддержка деления

16-битное арифметико-логическое устройство (АЛУ) позволяет выполнять за один командный такт следующие операции: сложение, вычитание, битовый сдвиг и поразрядные логические операции, включая инверсию. Результаты операций влияют на статусные флаги. АЛУ выполняет операции с 16-битными словами или байтами в зависимости от синтаксиса инструкции.

В состав ядра входит умножитель 17×17 бит, который позволяет выполнять операции умножения 16×16, 16×8 и 8×8 бит, знаковые, беззнаковые и смешанные за один командный такт. При знаковом и смешанном умножении применяется расширение знака. Операции умножения не влияют на статусные флаги. В качестве операндов используются рабочие регистры, результат умножения возвращается в указанную регистровую пару.

Аппаратная поддержка деления (итерационный метод) значительно сокращает время выполнения математических алгоритмов. Операции деления 32/16 и 16/16 бит (знаковые и беззнаковые) выполняются за 18 командных тактов. В качестве операндов используются рабочие регистры, после выполнения деления доступен результат и остаток.

Ядро содержит модуль сдвига (barrel shifter), благодаря которому за один командный такт выполняются операции сдвига и вращения на произвольное (до 15) количество бит. Сдвиг вправо может быть логическим (без расширения знака) или арифметическим (с расширением знака).

Ядро ЦОС

Ядро цифровой обработки сигналов (ЦОС) является независимым от CPU-ядра вычислительным модулем, который присутствует только в контроллерах семейства dsPIC30 и dsPIC33.

Умножитель dsPIC30/33 позволяет выполнять знаковые и беззнаковые операции с данными формата Q.31 (фиксированная точка). Тип операндов определяется битом в служебном регистре CORCON. При умножении чисел с фиксированной точкой производится сдвиг результата влево на один бит, что предоставляет возможность сохранить формат числа, при этом младший бит результата всегда обнуляется. Делитель микроконтроллеров с DSP ядром также разрешает выполнять знаковое деление чисел в формате Q.31.

Следует заметить, что умножитель и делитель могут использоваться как стандартными инструкциями умножения и деления, так и инструкциями DSP-ядра, операционный код которых отличается от стандартных. Так, инструкция MUL является стандартной инструкцией умножения CPU-ядра (для нее доступны функции умножителя, указанные в предыдущем разделе), а инструкция MPY — инструкцией умножения DSP-ядра, с помощью которой можно производить операции с Q.31 числами.

Результат инструкций DSP-ядра всегда сохраняется в один из 40-битных аккумуляторов: ACCA или ACCB. Аккумуляторы отображены в ОЗУ и могут быть доступны для любых инструкций. При представлении чисел в виде Q.31 диапазон значений аккумулятора составляет $[-256.0, +256.0 - 2^{-31}]$. Аккумуляторы имеют статусные флаги, сигнализирующие о переполнении, частичном переполнении (результат по модулю больше 1.0), опционально установка этих флагов может генерировать прерывание. Содержимое 40-битного аккумулятора может сохраняться в 16-битном слове ОЗУ, при этом используются различные режимы округления и усечения.

Инструкции, поддерживаемые ядром DSP, приведены в таблице 2.

В качестве примера рассмотрим одну из основных инструкций для ЦОС-алгоритмов — инструкцию MAC — умножение с накопле-

Таблица 2. Инструкции, поддерживаемые DSP-ядром

DSP инструкция	Алгебраический эквивалент	Выполняемая операция
MAC	$a = a + b \times c$	Умножение двух операндов, добавление результата к аккумулятору
MSC	$a = a - b \times c$	Умножение двух операндов, вычитание результата из аккумулятора
MPY	$a = b \times c$	Умножение двух операндов, сохранение результата в аккумуляторе
MPY.N	$a = -b \times c$	Умножение двух операндов, инверсия, сохранение результата в аккумуляторе
ED	$a = (b - c)^2$	Вычисление евклидовой метрики, сохранение результата в аккумуляторе
EDAC	$a = a + (b - c)^2$	Вычисление евклидовой метрики, добавление результата к аккумулятору

нием. Типовое использование MAC-инструкции (ассемблер ASM30):

```
MAC      W4*W5, A, [W8]+=2, W4, [W10]-=2, W5, [W13]+=2
```

Данная запись означает следующее:

- значение **W4** умножается на значение **W5**, результат умножения добавляется к аккумулятору **ACCA**;
- регистр **W4** загружается значением по указателю **W8** (указатель должен указывать на адрес в сегменте X ОЗУ);
- указатель **W8** увеличивается на 2;
- регистр **W5** загружается значением по указателю **W10** (указатель должен указывать на адрес в сегменте Y ОЗУ);
- указатель **W10** уменьшается на 2;
- значение аккумулятора **ACCB** сохраняется по указателю **W13**;
- указатель **W13** увеличивается на 2.

Все перечисленные операции с данными выполняются за один (!) командный такт. Одновременная выборка операндов в ходе выполнения умножения вместе с аппаратными циклами **DO** и **REPEAT** позволяет чрезвычайно эффективно реализовывать ЦОС-алгоритмы.

К ядру ЦОС можно отнести два аппаратных цикла, **DO** и **REPEAT**, хотя цикл **REPEAT** присутствует и в 16-битных контроллерах без DSP-ядра.

Цикл **REPEAT** позволяет аппаратно повторять (до 2^{14} раз) выполнение одной инструкции. Аппаратная вложенность не поддерживается, однако программно может быть реализовано любое количество уровней — при этом необходимо сохранять счетчик цикла в стеке. Цикл **REPEAT** может использоваться, например, совместно с инструкцией **MAC**, что позволяет всего двумя командами реализовать вычисление КИХ-фильтра.

Цикл **DO** предназначен для повторения блока кода объемом до 2^{16} инструкций. Произвольный блок может быть выполнен циклически до 2^{14} раз. Цикл **DO** аппаратно поддерживает 7 уровней вложенности, программно может быть реализовано произвольное количество уровней. Выполнение цикла может быть досрочно прервано в любой момент. Цикл **DO** можно использовать для выполне-

ния операций двумерной фильтрации, работы с векторами и матрицами и т. д.

Кроме ЦОС-инструкций DSP-ядро аппаратно поддерживает специфические для ЦОС-алгоритмов методы адресации. К таким относятся адресация по модулю и бит-реверсивная адресация.

Механизм адресации по модулю выполняется отдельно для областей ОЗУ X и Y и позволяет эффективно реализовывать циклические буферы. Суть адресации по модулю состоит в том, что контроль выхода указателя за пределы буфера и его возврат на начало осуществляются аппаратно. Причем максимально возможный размер буфера составляет 2^{15} слов ОЗУ, а размер буфера не обязательно должен быть кратен степени 2. Возможен опрос буфера как с увеличением, так и с уменьшением указателя.

Бит-реверсивная адресация является одной из основных операций алгоритма БПФ (быстрое преобразование Фурье) с прореживанием по времени или частоте. В DSP-ядре микроконтроллеров dsPIC30/33 реализована бит-реверсивная адресация, позволяющая выполнять БПФ с буфером, размер которого кратен степени 2 (radix-2 FFT). Суть метода заключается в зеркальном отражении части разрядов адреса для выборки следующего элемента из буфера, максимальный объем которого равен 2^{15} (32 768) 16-битных слов.

DMA

Контроллер DMA (Direct Memory Access — прямой доступ к памяти) является частью ядра семейств PIC24H и dsPIC33. В PIC24F и dsPIC30 контроллер DMA отсутствует. Назначение DMA — аппаратная поддержка обмена данными между периферией и областью ОЗУ общего назначения.

Контроллер DMA имеет 8 однонаправленных каналов, каждый из которых может обслуживать любой, поддерживающий аппаратный обмен периферийный модуль. К таким модулям относятся: модули захвата, модули сравнения, таймеры, SPI, UART, CAN, АЦП, DCI. Контроллер имеет 2 кбайт двухпортовой памяти ОЗУ, которая может использоваться в качестве буфера DMA.

Контроллер DMA и ядро микроконтроллера могут получать одновременный доступ как к двухпортовому буферу, так и к регистрам периферийного модуля. Единственное ограничение — DMA и CPU-ядро не должны одновременно сохранять данные по одному адресу. В этом случае приоритет имеет CPU-ядро, а контроллер DMA генерирует аппаратное исключение — специальный вид прерывания, который будет рассмотрен в пункте «Система прерываний» данной статьи.

Контроллер DMA поддерживает следующие режимы работы:

- **Однократный обмен (one-shot)**. В этом режиме контроллер DMA завершает работу после одной передачи данных, генерируя при этом прерывание. Для продолжения работы канал DMA нужно переинициализировать.

- **Обмен с автоматическим повтором (autorepeat)** подразумевает автоматическую переинициализацию канала DMA после завершения сессии обмена — указатель на буфер DMA сбрасывается, генерируется прерывание.
- **Прерывание при половинной передаче (half buffer transfer interrupt)**. В этом случае DMA-контроллер генерирует прерывание при половинном заполнении буфера. Это позволяет начать обработку данных еще до завершения сессии обмена. При заполнении буфера канал DMA переинициализируется автоматически, прерывание при этом не генерируется.
- **Режим ping-pong**. Канал DMA использует два буфера. При заполнении первого буфера указатель перемещается на начало второго, генерируется прерывание. После заполнения второго буфера указатель перемещается на начало первого, генерируется прерывание. Это позволяет CPU-ядру работать с данными в одном буфере, в то время как DMA-контроллер заполняет другой.
- **Режим косвенной адресации периферии**. В этом режиме указатель на буфер генерирует не DMA-контроллер, а непосредственно поддерживающий этот режим периферийный модуль (к таким относятся модули CAN и АЦП). Например, модуль АЦП может быть настроен так, чтобы отсчеты разных каналов АЦП сохранялись в определенных областях DMA-буфера, вне зависимости от последовательности преобразования.

Набор инструкций и методы адресации

16-битная архитектура Microchip имеет в своем составе блок из 16 рабочих регистров W0...W15. Все регистры ортогональны с точки зрения системы команд, то есть могут быть использованы в качестве операнда инструкции. Часть регистров имеет служебные функции: W15 — это указатель стека, W14 — указатель стекового фрейма. Некоторые инструкции могут использоваться в качестве операндов или регистров сохранения результата только определенный W-регистр или регистровую пару.

16-битные микроконтроллеры Microchip имеют расширенный набор инструкций, большинство из которых поддерживает операции типа «чтение-модификация-запись», что позволяет работать с данными напрямую в ОЗУ, не используя рабочие регистры. Большинство инструкций являются трехоперандными ($A = B + C$) и могут работать как с байтами, так и с 16-битными словами.

Семейства с DSP-ядром (dsPIC30/33) имеют 83 инструкции (включая инструкции DSP-ядра), семейства без DSP-ядра (PIC24F/H) — 76. Все инструкции выполняются за один командный такт за исключением:

- инструкций изменения программного потока — 2 или 3 такта,
- инструкций табличного чтения/записи — 2 такта,
- инструкции MOV.D — 2 такта,

- инструкции DO — 2 такта. Все инструкции можно условно разделить на несколько групп:
- MOVE — инструкции перемещения данных;
- MATH — инструкции выполнения математических операций;
- LOGIC — инструкции выполнения поразрядных логических операций;
- SHIFT/ROTATE — инструкции сдвига с переносом и без переноса;
- BIT — инструкции работы с битами;
- STACK — инструкции работы со стеком;
- PROGRAMM FLOW — инструкции изменения программного потока (переходы, вызовы, условные переходы);
- CONTROL — инструкции управления ядром (инициализация аппаратных циклов, программный сброс, перевод контроллера в энергосберегающий режим);
- DSP — инструкции DSP-ядра, присутствуют только в системе команд контроллеров dsPIC30/33.

Архитектура поддерживает большое количество методов адресации:

- прямая адресация, адрес операнда указывается непосредственно в операционном слове инструкции. Операнд должен быть расположен в области Near Space (первые 8 кбайт ОЗУ, рис. 2):

```
add    0x900, W0
; Сложение значения по адресу 0x900 с W0.
; Результат сохраняется в W0
```

- расширенный режим инструкции MOV — прямая адресация, но может адресоваться вся память ОЗУ

```
mov    0x2500, W7
; Сохранение значения по адресу 0x2500 в регистр W7
```

- прямая адресация регистров. В качестве операндов используются значения рабочих регистров:

```
ior    W0, W2, W5
; Поразрядное логическое ИЛИ регистров W0 и W2,
; сохранение результата в W5
```

- косвенная адресация — может использоваться в большинстве инструкций:

```
add    W4, [W5], [W6]
; Сложение W4 со словом по указателю W5,
; сохранение результата по указателю W6
```

- косвенная адресация с пре/пост инкрементом и декрементом, при этом учитываются правила адресной арифметики в зависимости от типа операнда (байт или слово):

```
mov    [++W0], [W1--]
; Увеличение указателя W0 на 2, перемещение значения
; по указателю W0 в ячейку, на которую указывает W1,
; уменьшение указателя W1 на 2.
```

- косвенная адресация со смещением:

```
mov    [W4+W5], [W6++]
; Получение указателя на ячейку операнда, путем сложения W4
; и W5, перемещение значения по полученному указателю
; по адресу W6, увеличение значения W6 на 2.
```

Поддержка различных методов адресации позволяет получать очень компактный код при использовании компиляторов с языком высокого уровня. Следует также отметить возможность вызова и перехода по значению в регистре (аналог вызова функции по указателю в Си) и инструкцию запрещения прерываний на определенное количество командных тактов (позволяет выполнять набор инструкций как одну атомарную).

Система прерываний

Микроконтроллеры dsPIC30/33 и PIC24F/H обладают векторной приоритетной системой прерываний. Каждый источник прерывания имеет собственный вектор в таблице, расположенной в программной памяти. Внутри таблицы прерывания имеют естественный приоритет: при одновременном возникновении двух прерываний приоритет имеет тот, чей вектор имеет меньший адрес. Вектор представляет собой 24-битное слово программы, в котором должна быть расположена команда перехода на сервис обработчика прерывания.

Пользователь может изменить естественный приоритет, назначив источнику прерывания искусственный приоритет. Для того чтобы прерывание было сгенерировано, необходимо, чтобы назначенный приоритет был выше текущего приоритета ядра (который та же можно изменять программно). При возникновении прерывания в программный счетчик записывается адрес вектора прерывания, а приоритет CPU-ядра становится равным приоритету возникшего прерывания. Это позволяет организовывать гибкую систему вложенных прерываний.

Возможно назначение приоритета каждому прерыванию, количество возможных уровней приоритета равно 8. CPU-ядро может иметь приоритет от 0 до 15, уровни 8–15 зарезервированы для аппаратных исключений.

Аппаратные исключения — это определенный вид немаскируемых прерываний, которые генерирует CPU-ядро. Условно исключения можно разделить на программные и аппаратные.

К программным относятся исключения, которые позволяют продолжить выполнение работ после программного сброса флага исключения. Программные исключения имеют приоритет от 8 до 12. К программным исключениям относятся: исключение АЛУ (которое генерируется ядром, например, при делении на 0) и ошибка стека.

При возникновении аппаратного исключения программа не может продолжить работу до тех пор, пока ошибка, которая вызва-

ла исключение, не будет устранена — флаг аппаратного исключения не может быть сброшен программно. Аппаратные исключения имеют приоритет от 13 до 15. К аппаратным исключениям относятся: ошибка адреса (не выровненный доступ к памяти программ или памяти данных) и ошибка тактового генератора (нестабильная работа PLL, отсутствие внешней тактовой частоты).

Система прерываний имеет фиксированное время реакции — 5 командных тактов. Возврат из прерывания осуществляется за 3 командных такта. Следует заметить, что при входе в прерывание часть статус-регистра **SR** автоматически сохраняется в стеке, это позволяет уменьшить объем кода и время выполнения обработчика прерывания. Все флаги прерываний устанавливаются вне зависимости от того, разрешено прерывание или нет.

В программной памяти расположены две идентичные таблицы прерываний (рис. 1) — основная и альтернативная. Используемая в текущий момент таблица прерываний может быть выбрана программно. Это удобно, например, для реализации внутрисхемного программного загрузчика (bootloader).

Микроконтроллеры семейства dsPIC30 имеют 54 вектора прерывания и 8 векторов исключений в каждой таблице. Семейства dsPIC33 и PIC24F/H имеют расширенные таблицы прерываний — 118 векторов прерываний и 8 векторов исключений. ■

Продолжение следует

Литература

1. DS39747C: PIC24FJ128GA Family Data Sheet
2. DS70165D: dsPIC33F Family Data Sheet
3. www.microchip.com/stellent/groups/dspic_sg/documents/devicedoc/en027621.zip
4. www.microchip.com/mplab
5. www.iar.com
6. www.piklab.sourceforge.net
7. www.microchip.com/c30
8. www.htsoft.com/products/dspiccompiler.php
9. www.iar.com
10. www.mikroe.com/en/compilers/mikroc/dspic
11. www.microchip.com/tcpip
12. www.cmx.com/microchip/index.htm
13. www.jacos.narod.ru
14. www.pumpkininc.com
15. www.micrium.com/microchip/index.html
16. www.benchmarks.caxapa.ru/?test=13
17. www.benchmarks.caxapa.ru/?test=23
18. www.techtrain.microchip.com/webseminars/Main.aspx
19. www.forum.microchip.com