

Новая архитектура среднего семейства контроллеров Microchip

Многие годы компания Microchip с успехом выпускает 8-разрядные PIC-контроллеры, и продолжает вкладывать средства в их развитие для поддержания широкой номенклатуры продукции, которая удовлетворит потребностям существующих и будущих клиентов. Новая усовершенствованная архитектура среднего семейства контроллеров PIC12 и PIC16 взяла все самое лучшее от существующей архитектуры и получила дополнительные возможности, сохранив при этом преемственность и легкий переход на новые микроконтроллеры.

Новая улучшенная архитектура 8-битных PIC-микроконтроллеров среднего семейства PIC12 и PIC16 имеет:

- увеличенный объем памяти программ и данных (до 32 кбайт инструкций и более 4 кбайт памяти данных);
- улучшенный аппаратный стек (увеличен до 16 уровней, программный доступ);
- модернизированный метод переключения между страницами памяти и банками данных;
- дополнительные источники сброса;
- 14 дополнительных программных инструкций, включая команды для более эффективной работы C-компилятора;
- расширенную периферию;
- уменьшенное время входа в прерывание;
- увеличенную тактовую частоту ядра микроконтроллера.



Ядро контроллеров среднего семейства новой улучшенной архитектуры

Новые команды

Микроконтроллеры нового семейства в дополнение к 35 командам существующего среднего семейства получили 14 новых инструкций. Новые команды (табл. 1) позволяют получить более быстрый и более компактный код программы и предоставляют эффективный доступ к данным.

Таблица 1. Новые команды

Новые команды	
Мнемоника	Описание
Новые арифметические команды и команды сдвигов	
ADDWFC	Сложить значение рабочего регистра W с регистром F с учетом флага переноса
SUBWFB	Вычесть из регистра F значение рабочего регистра W с учетом флага заема
LSLF	Логический сдвиг влево
LSRF	Логический сдвиг вправо
ASRF	Арифметический сдвиг вправо
Одноцикловое переключение страниц и банков памяти	
MOVLP	Загрузить константу в регистр PCLATH
MOVLB	Загрузить константу в регистр BSR
Относительный переход	
BRA	Относительный переход (знаковый)
BRW	Переход PC +W (беззнаковый)
Вызов подпрограммы с адресом в W	
CALLW	Вызов PCLATH:W
Программный сброс	
RESET	Сброс программы и периферии
Дополнительные команды косвенной адресации	
ADDFSR	Добавить константу к FSR (знаковая команда)
MOVIW	Копировать косвенно в W
MOVWI	копировать W по указателю

К существующим арифметическим командам сложения и вычитания добавились команды сложения и вычитания с учетом флага переноса/заема. Данные команды улучшают эффективность кода для многобайтной арифметики. К командам циклического сдвига добавились команды логического и арифметического сдвигов. Конечно же, логический сдвиг можно сделать через циклический сдвиг, но для этого понадобятся дополнительные команды.

Примечание.
Внимательный читатель может задать вопрос: «а почему нет команды арифметического сдвига влево?»
Ответ: арифметический сдвиг влево эквивалентен логическому сдвигу влево. Для арифметического сдвига влево вы можете использовать псевдо-команду ассемблера ASLF.

В новых контроллерах с улучшенной архитектурой среднего семейства увеличено адресуемое пространство памяти программ и ОЗУ. Теперь не нужно устанавливать биты RP0 и RP1 в регистре STATUS для того, чтобы переключить банк памяти, достаточно загрузить адрес в регистр переключения банков BSR. Аналогично и с переключением страниц памяти – добавлена команда загрузки регистра PCLATH.

Новые команды позволяют использовать меньше команд для переключения банков ОЗУ и страниц памяти программ. Для обеспечения совместимости кода между контроллерами нового поколения и старой архитектуры лучше использовать специальные макросы ассемблера PAGESEL и BANKSEL (см. Табл.2), которые в зависимости от выбранного контроллера будут соответствующим образом интерпретированы компилятором.

Таблица 2. Примеры кода переключения банков ОЗУ и страниц памяти программ

Ядро Mid-Range	Ядро Enhanced Mid-Range	Пример совместимого кода
Вызов функции по любому адресу памяти программ		
movlw high My_Function	movlp high My_Function	PAGESEL My_Function
movwf PCLATH	call My_Function	call My_Function
call My_Function		
Доступ к произвольной переменной (с переключением банков)		
bsf STATUS, RP0	movlb HIGH Var1	BANKSEL Var1
bcf STATUS, RP1	addwf Var1	addwf Var1
addwf Var1		

Основное достоинство применения команд относительных переходов это уменьшение размера кода, связанное с отсутствием необходимости отслеживания переходов через границы страниц памяти программ. Иногда, на старом ядре, простой цикл может перестать работать, если сдвинуть программу так, что цикл попадет на границу страницы памяти программ. Таких сюрпризов можно избежать при использовании команды

относительных переходов. Также многие знакомы с организацией вычисляемых переходов на стандартном ядре Mid-Range, где нужно либо усложнять код, либо программист должен следить, чтобы таблицы данных не попали на границы блоков в 256 байт. Команда BRW осуществляет переход относительно текущего значения счетчика команд на беззнаковое значение, находящееся в рабочем регистре W. С дополнительной командой BRW не нужно следить за размещением таблиц и усложнять код – операции табличного чтения становятся намного проще:

```
constants
    brw
    retlw DATA1
    retlw DATA2
    retlw DATA3
    retlw DATA4
my_function
    ;... код программы ...
    movlw DATA_INDEX
    call constants
    ;... константа DATAx в W регистре
```

Команда BRA аналогична команде BRW, только осуществляет переход относительно текущего значения счетчика команд на 9-и битную константу с учетом знака (переход в пределах -256...+255 команд).

С командой CALLW становятся доступен вызов процедуры по адресу, указанному в W регистре, что может пригодиться для быстрого поиска по таблицам, организации машины состояний и вызова функций по указателю. Следующие три команды расширяют возможности косвенной адресации.

Команда ADDFSR производит знаковое сложение константы с указателем FSR, значение смещения должно лежать в диапазоне -32 до +31. Команды MOVIW и MOVWI более сложные. Обе эти команды могут выполняться с пре- и пост- инкрементом и декрементом указателя, а так же со смещением указателя.

Рассмотрим более подробно синтаксис команд MOVWI (команда MOVIW имеет тот же синтаксис, только работает в обратном порядке – пересылает данные из косвенно-адресуемого регистра в рабочий регистр W):

```
MOVWI 0[INDF0]
```

— перенос значения из W в косвенно адресуемый регистр. Эта операция может быть выполнена так же старыми командами MOVWF INDF0 или MOVF INDF0,W. Нет никаких преимуществ между тем или иным способом. Команда MOVWI 0[INDF0] не меняет значение указателя.

```
MOVWI ++INDF0
```

— перенос значения из W в косвенно адресуемый регистр с преинкрементом. Если перед INDF0 или INDF1 написать ++, то данная команда произведет инкремент указателя FSR до того как значение из регистра W будет помещено в косвенно адресуемый регистр. Подобные команды позволяют обновить указатель до записи или чтения.

```
MOVWI INDF0++
```

— перенос значения из W в косвенно адресуемый регистр с постинкрементом. Аналогично и команды с пре- и пост-инкрементом:

```
MOVWI --INDF0
MOVWI INDF0--
```

Комбинация преинкремента и постдекремента это быстрый способ программного построения стека или циклических буферов данных.

```
MOVWI k[INDF0]
```

— перенос значения из W в косвенно адресуемый регистр со смещением относительно указателя. Вы можете обратиться к регистру со смещением относительно указателя. Данная команда не изменяет значение самого указателя. Данная команда востребована для загрузки элемента структуры в рабочий регистр W или получении доступа к регистрам специального назначения в различных банках. Значение смещения может лежать в диапазоне -32 до +31 относительно адреса, загруженного в указатель FSR.

Память данных контроллеров

В новых контроллерах с улучшенной архитектурой среднего семейства существенно расширена память данных. Теперь они имеют 32 банка памяти общим размером до 4Кб. Как и прежде память данных разделена на регистры специального назначения (SFR - Special Function Registers) – данные регистры служат для настройки прерываний, периферии и пр.; регистры общего назначения (GPR – General Purpose Registers) – данные регистры служат для хранения пользовательских данных.

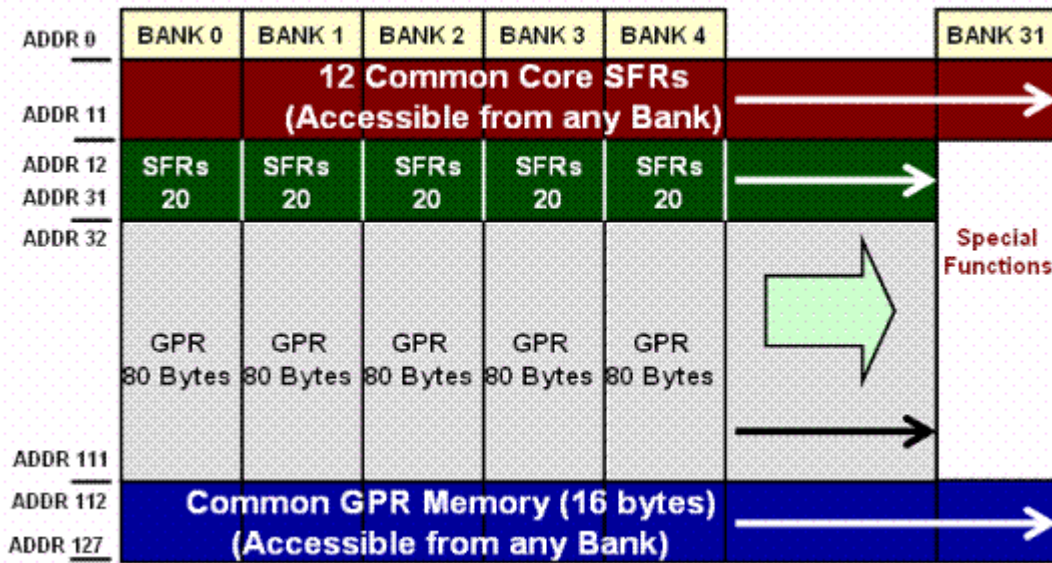


Рис. 1. Память данных нового ядра PIC16

Регистры GPR банков памяти с 0 по 30 могут использоваться по усмотрению программиста, банк 31 служит для специальных функций, назначение этого банка рассмотрим чуть позже.

Таблица 3. Первые 12 регистров специальных функций каждого банка.

	Адрес	Регистр	Функция
	0x00	INDF0	Регистр косвенной адресации 0
New!	0x01	INDF1	Регистр косвенной адресации 1
	0x02	PCL	Счетчик Команд (младший байт)
	0x03	STATUS	Регистр Статуса
New!	0x04	FSR0 Low	Указатель косвенной адресации 0 (младший байт)
New!	0x05	FSR0 High	Указатель косвенной адресации 0 (старший байт)
New!	0x06	FSR1 Low	Указатель косвенной адресации 1 (младший байт)
New!	0x07	FSR1 High	Указатель косвенной адресации 1 (старший байт)
New!	0x08	BSR	Регистр выбора банка
New!	0x09	WREG	Рабочий регистр
	0x0A	PCLATH	Защелка Счетчика Команд (старший байт)
	0x0B	INTCON	Регистр контроля прерываний

В каждый банк памяти отображаются 12 общих регистров специального назначения (Табл. 3), таким образом, можно получить доступ к этим регистрам из любого банка данных. Еще одна особенность новых контроллеров в том, что выделенные в Табл. 3 регистры автоматически сохраняются при возникновении любого прерывания. Сохраненные при входе в прерывание регистры доступны программисту в 31-м банке.

Память программ контроллеров. Новые свойства указателя FSR

В новых контроллерах с улучшенной архитектурой среднего семейства теперь два регистра FSR (FSR – File Select Register), которые к тому же стали 16-битными. Теперь через регистры указателей FSR программист получает доступ ко всей памяти микроконтроллера – памяти данных и памяти программ! Доступ к константам в памяти программ абсолютно такой же, как и к регистрам в ОЗУ – как только вы загрузили указатель, следующей инструкцией вы можете считать значение из памяти программ.

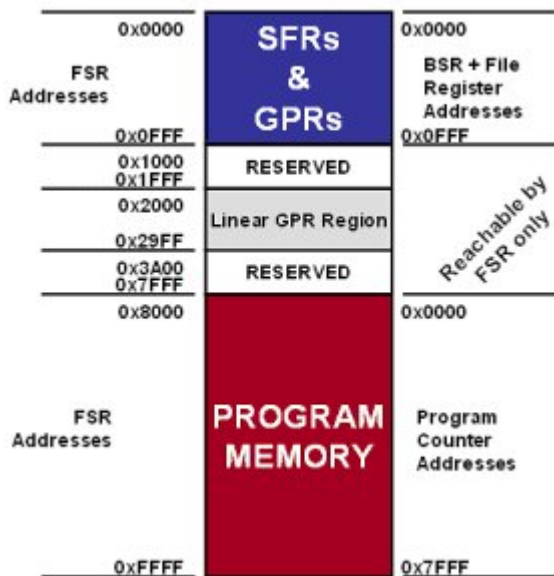


Рис. 2. Организация памяти новых контроллеров. Области адресации памяти данных, программ и доступ при косвенной адресации

Интересная новая возможность контроллеров, это линейная адресация всех регистров общего назначения. Как видно из рисунка 1, регистры общего назначения GPR имеют адреса 20h-6Fh, A0h-EFh, 120h-16Fh и т.д. Такое «дырчатое» распределение регистров общего назначения не удобно для организации больших массивов данных и их адресации. В новых контроллерах добавлен режим линейной адресации памяти, позволяющий при установке указателя FSR на адреса 2000h – 29AFh адресовать всю область регистров общего назначения без «дыр». Новый режим адресации позволяет упростить создание больших массивов в памяти данных.

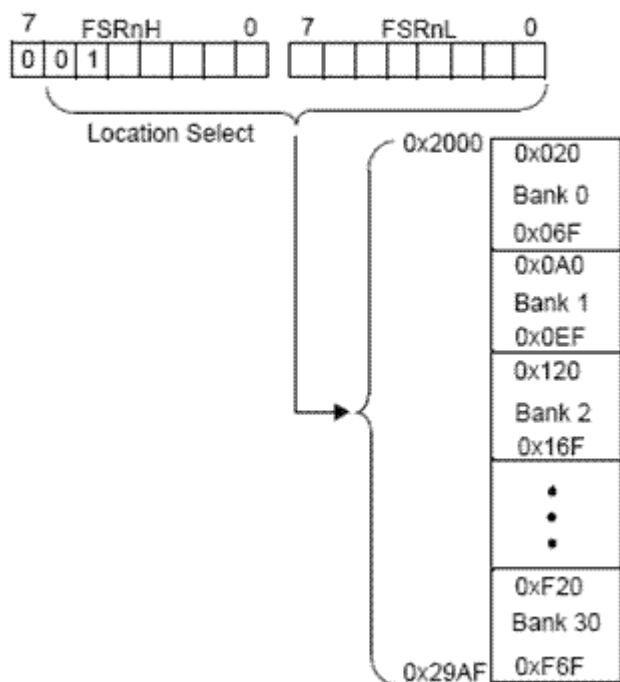


Рис. 3. Линейное отображение регистров общего назначения

Если старший бит регистра FSR установлен в «1», то это означает, что регистр FSR будет указывать на область памяти программ.

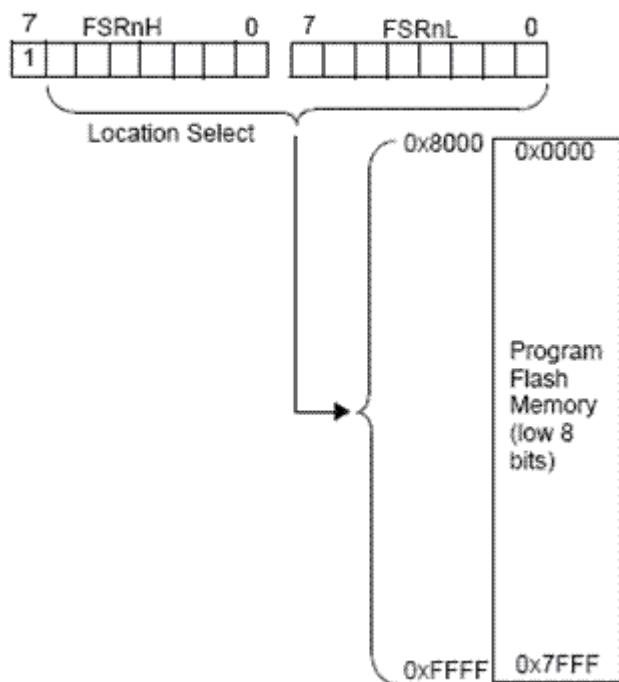


Рис.4. Адресация памяти программ через регистры косвенной адресации

Стек контроллеров улучшенной архитектуры

Контроллеры среднего семейства PIC16 (табл. 4) имеют кольцевой 8-уровневый стек. Если происходит переполнение стека, то происходит «затирание» самого первого адреса входа. В новых контроллерах стек стал больше – способен хранить 16 адресов. Кроме этого контроллеры получили возможность сброса при переполнении или опустошении стека. Более того, стек в новой архитектуре доступен программисту через регистры, находящиеся в 31-м банке – SPTR (указатель стека); текущее значение стека – состоит из 2-х байтов TOSH и TOSL. Программный доступ к стеку может пригодиться при создании операционных систем реального времени или безопасной отладке программ.

Для внутрисхемной отладки кода в старых контроллерах резервируется один уровень стека, в то время как новое ядро имеет отдельный стек для отладки. Это позволяет использовать весь стек в режиме отладки кода, при этом, когда включен режим отладки, сброс по переполнению стека становится точкой останова кода, что позволяет определить причину переполнения стека.

Таблица 4. Сравнение контроллеров среднего семейства (Mid-Range) и улучшенного среднего семейства (Enhanced Mid-Range)

Ядро	Mid-Range	Enhanced Mid-Range
Длина инструкций, бит	14	14
Адресуемая память программ	8 кбит инструкций	32 кбит инструкций
Максимальный объем ОЗУ и регистров, байт	446	>4096
Число инструкций	35	49
Аппаратный стек	8-уровневый	16-уровневый с дополнительными возможностями
Сохранение контекста	Программно	Аппаратно

Результаты улучшений

Новая архитектура контроллеров позволяет снизить размер кода на 40% и увеличить производительность микроконтроллера на 50% за счет расширенного набора команд, новых возможностей и увеличенной тактовой частоты.

На рис. 5, 6 приведены сравнительные данные по уменьшению числа циклов и снижению размера кода, требуемых для реализации различных алгоритмов.

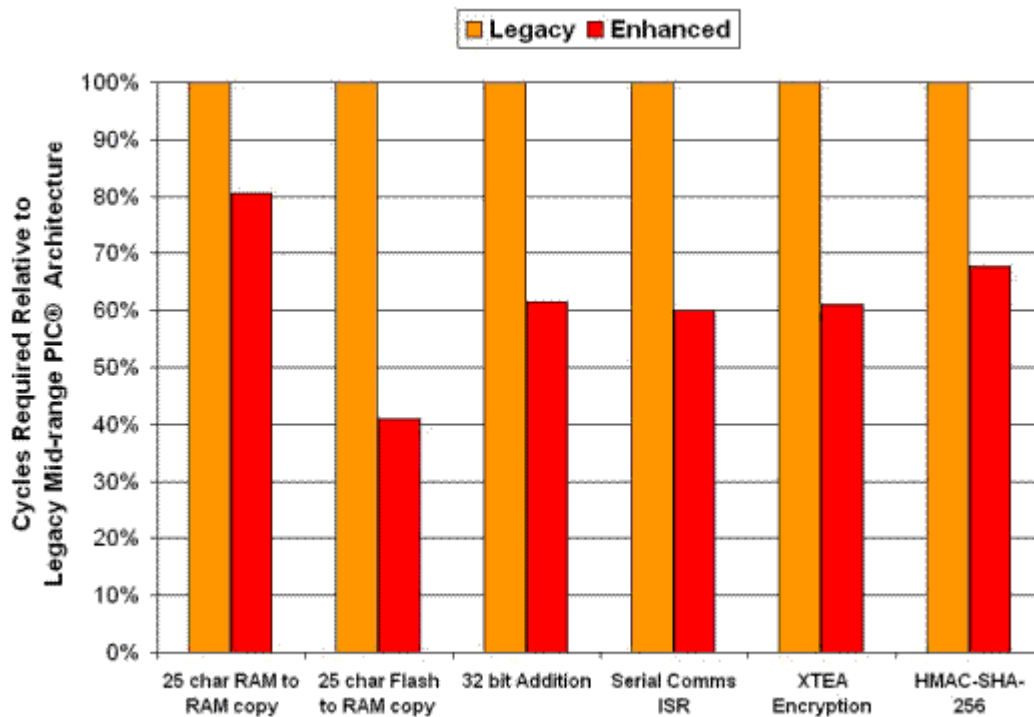


Рис. 5. Сравнение быстродействия контроллеров среднего семейства старой и улучшенной архитектуры

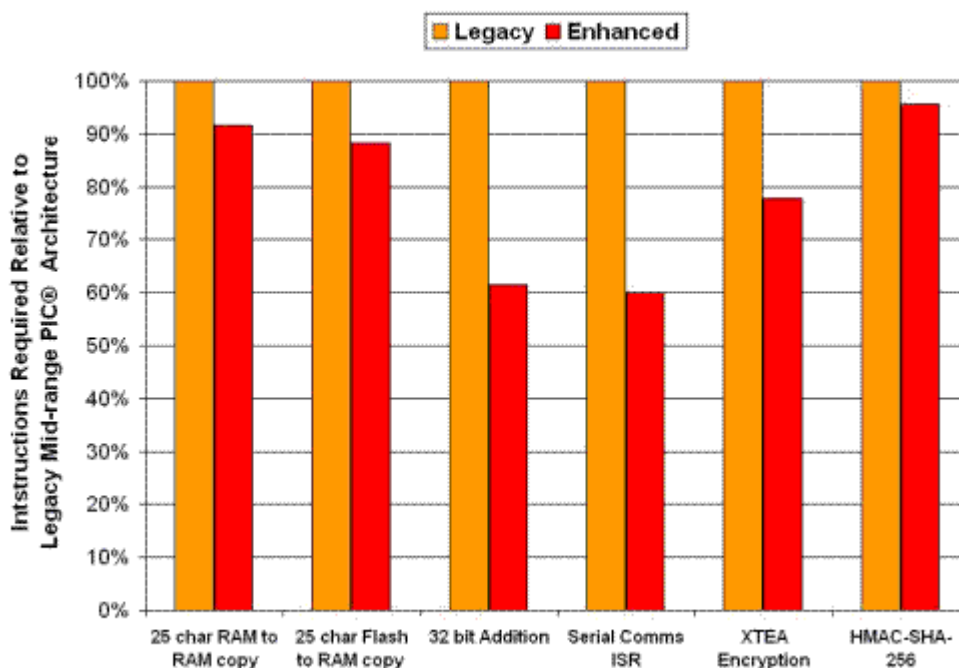


Рис. 6. Сравнение размера кода для контроллеров среднего семейства старой и улучшенной архитектуры

Результат тестирования улучшений производилось на следующих задачах:

- RAM to RAM copy: копирование строки между двумя областями GPR регистров. Данная задача эквивалентна команде strcpy() в Си. С использованием двух регистров косвенной адресации FSR, новых команд и улучшений в работе косвенной адресации, данная задача по копированию данных выполняется быстрее и занимает меньший размер памяти программ.
- Flash to RAM copy: Данный тест аналогичен предыдущему – осуществляет копирование строки между Flash памятью программ и регистрами общего назначения GPR. Подобная задача часто используется для чтения констант, сообщений и других данных из памяти программ микроконтроллера. Примечательно то, что в новых контроллерах функция копирования данных из Flash памяти программ и

ОЗУ будет иметь одинаковый программный код за счет возможности адресации данных и памяти программ с помощью регистров косвенной адресации, что позволит еще больше уменьшить размер программы.

- 32 bit addition: сложение 32-х разрядных чисел. Улучшенное ядро с новыми командами позволяет существенно сократить размер кода и ускорить арифметические операции с многобайтными данными.
- Serial ISR: подпрограмма обработки данных, получаемых с последовательного порта в прерываниях. Подпрограмма копирует данные из регистра UART в кольцевой буфер. Новые команды работы с регистрами косвенной адресации, а так же автоматическое сохранение контекста при входе в прерывание позволяют ускорить работу и сократить размер подобных подпрограмм.
- Алгоритм шифрования XTEA это простой и быстрый алгоритм, позволяющий осуществлять шифрацию и дешифрацию данных с помощью секретного пароля.
- Алгоритм HMAC-SHA-256 это алгоритм хеширования, который не осуществляет кодирование или декодирование данных, но осуществляет аутентификацию (подтверждение подлинности) и подтверждение целостности (подтверждение того что сообщение не было изменено). Оба алгоритма используют ключи, константы, 32- и 64-х битную арифметику и новое ядро с его улучшенной системой команд позволяет увеличить быстродействие и снизить требуемый размер кода.

Периферия

Новые контроллеры содержат множество разнообразных периферийных модулей:

- контроллер ЖК-индикаторов (в контроллерах PIC16F19xx, более подробно об этом модуле читайте в статье нашего каталога);
- модуль емкостных датчиков (CSM – Capacitive Sensing Module), поможет в построении сенсорных клавиатур;
- 10-и разрядный АЦП;
- источник опорного напряжения с программируемыми значениями напряжения 1,024, 2,048 и 4,096 В – может работать совместно с модулем АЦП, компараторами и 5-разрядным ЦАП;
- таймеры: четыре 8- и один 16-разрядный, с возможностью подключения кварца 32 кГц и построения микропотребляющих часов реального времени;
- модули захвата, сравнения и ШИМ (до 3-х стандартных и до 2-х расширенных, со специальными режимами управления электродвигателями);
- модуль синхронного последовательного порта (MSSP) – поддерживаются интерфейсы SPI, I2C, а так же поддержка совместимости с интерфейсами SMBUS и PMBUS;
- модуль Универсального синхронного-асинхронного приемника-передатчика (USART) – поддержка интерфейсов RS-232, RS-485 и LIN;
- RS-триггер;
- два аналоговых компаратора;
- Модулятор Данных.

Отдельно рассмотрим наиболее интересные нововведения в периферии контроллеров.

Тактовый генератор

Контроллеры PIC16F1xxx могут работать с тактовой частотой до 32МГц в диапазоне питающих напряжений от 2.3 до 5.5В и до 16МГц (от 1.8В до 2.3В).

Тактирование ядра может осуществляться как от внешнего генератора (основного и дополнительного низкочастотного), так и от внутреннего программируемого генератора.

Блок внутреннего генератора содержит 3 основных модуля:

- низкочастотный некалиброванный 31КГц генератор (может также служить генератором для сторожевого таймера, таймера начального запуска, обеспечивать детектирование пропадания внешнего

тактирования и др.) * среднечастотный калиброванный генератор 500КГц (обеспечивает получение тактовых частот 500, 125, 62.5, 31.25 КГц)

- высокочастотный калиброванный генератор 16МГц (обеспечивает получение тактовых частот 32, 16, 8, 4, 2 и 1МГц).

Частота высокочастотного генератора 16МГц получается из источника 500КГц с помощью умножителя частоты с ФАПЧ. При необходимости, для снижения потребления контроллера, можно переключаться на тактовые частоты менее 1МГц, при этом будет задействован среднечастотный генератор. Частота обоих высокочастотного и низкочастотного генератора калибрована на заводе-производителе, но может быть подстроена и программно. При запуске высокочастотного генератора программист, с помощью двух контрольных бит, может получать информацию о выходе петли ФАПЧ умножителя на точность 2% и 0.5%, что можно использовать для блокирования требовательных к точности частоты процессов, до выхода генератора на рабочий режим.

С помощью высокочастотного генератора (в режиме формирования 8МГц) и основного умножителя тактовой частоты можно получить тактовую частоту ядра 32МГц. Таким образом, блок внутреннего тактового RC-генератора позволяет получить сетку частот от 31КГц до 32МГц без применения внешних элементов.

Модуль опорных частот.

Модуль опорных частот позволяет выдать деленную тактовую частоту на вывод микроконтроллера или на блок модулятора. Модуль опорных частот позволяет выбрать частоту из множества вариантов и с заданной скважностью 0, 25, 50 или 75%.

Модуль модулятора сигналов.

Модуль модулятора сигналов (Data Signal Modulator) позволяет получить модулированный по амплитуде, частоте и фазе сигнал. Модулирующий сигнал выбирается от различных источников: внешний сигнал с вывода микроконтроллера, один из выходов встроенных компараторов, модулей сравнения/ШИМ, модулей синхронных интерфейсов (SPI или I2C), и модуля синхронного/асинхронного интерфейса USART. Модулируемым сигналом (несущей) может выступать выход модуля опорных частот, выходы модулей сравнения/ШИМ или внешние сигналы.

Переключение между различными источниками несущей частоты позволяет осуществить амплитудную и частотную манипуляцию, а выбор полярности несущей – фазовую манипуляцию.

Порты ввода-вывода.

Впервые в контроллерах PIC12/PIC16 появился регистр защелки порта LAT (как в старших контроллерах PIC18). Данный регистр необходим для работы с портами ввода-вывода с помощью команд типа чтение-модификация-запись. Еще одна необычная особенность портов ввода-вывода новых контроллеров среднего семейства – это отключаемая возможность затягивания фронтов выходного сигнала модуля опорной частоты и модулятора сигналов, что может пригодиться для минимизации электромагнитных излучений, при быстром переключении портов. Программист может выбрать нормальный режим или затянуть фронт переключения портов в 10 раз. Данная особенность может пригодиться, особенно на тактовых частотах выше 20МГц.

Модуль Сравнения/ШИМ

В новых контроллерах улучшенного среднего семейства к каждому из 4-х модулей ШИМ можно независимо подключить один из нескольких таймеров. Ранее, в предыдущих семействах, в качестве базы для всех модулей ШИМ можно было выбрать только один таймер.

Аналоговая периферия.

Микроконтроллеры серии PIC16F1xxx содержат всю, ставшую стандартной аналоговую периферию – 10-и разрядный АЦП, два аналоговых компаратора и программируемый источник опорного напряжения. Из нововведений следует отметить наличие встроенного источника фиксированного опорного напряжения. Для фиксированного источника опорного напряжения выбираются три значения напряжения 1.024В, 2.048 и 4.096В. Источник опорного напряжения может использоваться как база для программируемого источника опорного напряжения (ЦАП), может подключаться на вход АЦП и компараторов. Выход программируемого источника опорного напряжения может так же подключаться ко входу АЦП и компараторов, а также выводиться на вывод микроконтроллера для использования совместно с внешними цепями микроконтроллера. Микроконтроллеры

PIC16F1xxx имеют два аналоговых компаратора. Компараторы часто интегрируются в микроконтроллеры, так как они предоставляют некоторые полезные функции, независимые от исполняемого программного кода. Встроенные компараторы имеют входной мультиплексор, позволяющий подключать входы компараторов к тому или иному выводу микроконтроллера и источникам опорного напряжения. Выход компаратора может подключаться на выходной порт микроконтроллера, может быть источником прерывания, выводит микроконтроллер из режима энергосбережения Sleep, а так же разрешать работу модулей ШИМ микроконтроллера.

В зависимости от требований задачи, программист может выбрать либо большую скорость срабатывания компаратора, и как следствие большее потребление тока, либо снизить потребление за счет меньшей скорости срабатывания. Программисту так же доступно изменение гистерезиса срабатывания компараторов.

Модуль компараторов может работать совместно с интегрированным RS-триггером. Данная связка позволяет реализовывать сенсорные клавиатуры, импульсные преобразователи энергии, управление мощными светодиодами и многое другое.

mTouch, Capacitive Sensing Module

Для построения сенсорных клавиатур в новых контроллерах интегрирован специализированный модуль Capacitive Sensing Module (CSM). В типовом применении этот модуль подключается к контактным площадкам на печатной плате. При поднесении пальца к изолированной сенсорной контактной площадке вносится дополнительная емкость, что приводит к изменению частоты модуля CSM. Для построения сенсорных клавиатур не требуется дополнительных внешних элементов, необходима только программная обработка и выделенный таймер для фиксации изменения частоты модуля.

Номенклатура контроллеров среднего семейства новой улучшенной архитектуры

Компания Microchip начала производства нескольких микроконтроллеров среднего семейства с улучшенной архитектурой, контроллеры доступны для заказа образцов и серийных партий (табл. 5). Все контроллеры имеют максимальную тактовую частоту 32 МГц, при этом такую частоту можно получить от встроенного программируемого RC-генератора.

Контроллеры PIC16F1xxx выполнены по микропотребляющей технологии XLP и имеют ядро с максимальным напряжением питания 3,6 В, поэтому микроконтроллеры LF-версий, имеют диапазон питания 1,8–3,6 В, а контроллеры F-версий содержат встроенный стабилизатор питания ядра и могут работать в расширенном диапазоне питаний от 1,8 до 5,5 В.

Таблица 5. Номенклатура новых микроконтроллеров

Наименование	Память			Внутренний генератор	АЦП, каналов	MSSP		EUSART	ЖКИ, сегментов	Упит, В	Число портов	Число выводов
	Flash, кбайт	RAM, байт	EEPROM, байт			SPI	I2C					
PIC12F1822*	3,5	128	256	32 МГц, 32 кГц	4	1	1	1	-	1,8–5,5	6	8
PIC16F1823*	3,5	128	256	32 МГц, 32 кГц	8	1	1	1	-	1,8–5,5	12	14
PIC16F1826	3,5	256	256	32 МГц, 32 кГц	12	1	1	1	-	1,8–5,5	16	18
PIC16F1827	7	384	256	32 МГц, 32 кГц	12	2	2	1	-	1,8–5,5	16	18
PIC16F1828*	7	256	256	32 МГц, 32 кГц	12	1	1	1	-	1,8–5,5	16	18
PIC16F1829*	14	1024	256	32 МГц, 32 кГц	12	2	2	1	-	1,8–5,5	16	18
PIC16F1933	7	256	256	32 МГц, 32 кГц	11	1	1	1	60	1,8–5,5	25	28
PIC16F1934	7	256	256	32 МГц, 32 кГц	14	1	1	1	96	1,8–5,5	36	44
PIC16F1936	14	512	256	32 МГц, 32 кГц	11	1	1	1	60	1,8–5,5	25	28
PIC16F1937	14	512	256	32 МГц, 32 кГц	14	1	1	1	96	1,8–5,5	36	44

Наименование	Память			Внутренний генератор	АЦП, каналов	MSSP		EUSART	ЖКИ, сегментов	Упит, В	Число портов	Число выводов
	Flash, кбайт	RAM, байт	EEPROM, байт			SPI	I2C					
PIC16F1938*	14	1024	256	32 МГц, 32 кГц	11	1	1	1	60	1,8–5,5	25	28
PIC16F1939*	28	1024	256	32 МГц, 32 кГц	14	1	1	1	96	1,8–5,5	36	44
PIC16F1946*	14	512	256	32 МГц, 32 кГц	17	2	2	2	184	1,8–5,5	53	64
PIC16F1948*	28	1024	256	32 МГц, 32 кГц	17	2	2	2	184	1,8–5,5	53	64

Примечание. * контроллеры находятся на стадии подготовки к массовому производству.

Средства разработки для новых контроллеров

Новые микроконтроллеры поддерживаются бесплатной средой разработки MPLAB IDE, внутрисхемными программаторами-отладчиками PICkit3, ICD-3 и REAL ICE. Поддержку новых контроллеров осуществляют несколько производителей программного обеспечения: компании Hi-Tech, CCS и Byte Craft имеют Си-компиляторы для новых PIC16F1xxx, а компания microEngineering Labs поддерживает новое ядро в компиляторе PICBASIC PRO.

Литература

1. www.microchip.com/enhanced
2. PIC16F193X/LF193X Data Sheet
3. PIC12F1822/16F1823 Product Brief
4. PIC1x1xxx Software Migration