

PIC18CXX2 — принципиально новое семейство микроконтроллеров фирмы



MICROCHIP

с 10-разрядным АЦП

В конце 1998 года фирма MICROCHIP анонсировала новую серию микроконтроллеров/микропроцессоров **PIC18CXX2**, которая должна была превосходить по функциональной насыщенности, вычислительным возможностям и быстродействию все существующие **PICmicro**.

Совокупность рабочих характеристик, которые удалось достичь разработчикам

ков с собственными шинами доступа к ячейкам:

- памяти программ,
- памяти данных,
- памяти стека.

Память программ

Память программ адресуется с помощью 21-разрядного счетчика команд (PC). Младший бит счетчика команд всегда равен «0» в связи с тем, что команды имеют только четные

адреса. Поэтому при просмотре кодов после ассемблирования адрес перехода команды GOTO в дампе памяти будет иметь в два раза меньшее значение.

Для вычисления адреса перехода необходимо пользоваться простой формулой: ад-

пользуя команду TBLRD, или при зашивке кристалла, причем идентификационные регистры памяти доступны для чтения даже при установленном бите защиты.

В названии микроконтроллера вторая цифра в номере модели обозначает количество килобайт памяти на кристалле. Количество килобайт рассчитывается как $2^2 = 32$ Кбайт. Например, **PIC18CXX52** имеет $2^5 = 32$ Кбайт. **PIC18CXX42** имеет $2^4 = 16$ Кбайт или 8К однословных команд соответственно.

Память данных

Память данных организована в виде массива 8-разрядных регистров, каждый из которых имеет 12-разрядный адрес. Все адресное пространство 4096 байт разделено на 16 банков памяти объемом по 256 байт. Номер адресуемого банка содержится в четырех младших битах регистра выбора банка (BSR).

Перед разработчиками ядра архитектуры **PIC18CXX2** были поставлены следующие задачи:

- Достичь производительности 10MIPS при 10 МГц генератора.
- Сохранить совместимость снизу вверх с серией **PIC16CXX** и **PIC17CXX**.
- Получить эффективный код при компиляции программы на языке C.

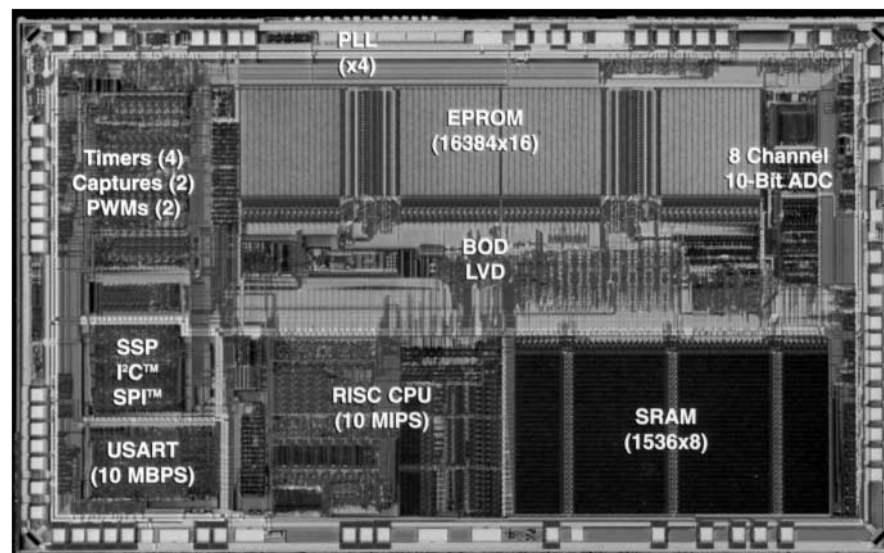
MICROCHIP, позволяет говорить о создании идеального микропроцессорного ядра для систем реального времени. Пиковая производительность составила 10MIPS@10МГц, а усовершенствования сделали работу с **PIC18CXX2** еще более удобной, сохранив возможности пользоваться наработанным программным обеспечением и конструктивными решениями. Сохранилась преемственность не только по коду, но и по выводам микроконтроллера и по периферийным модулям. Важнейшим усовершенствованием стала мгновенная реакция на прерывание (300 нс) с аппаратным сохранением контекста. Значительное повышение эффективности С-кода стало возможно благодаря линейному адресному пространству памяти программ и данных, трем указателям с пятью режимами индексации, удобству хранения глобальных переменных, программно доступному стеку, наличию 18-ти условий ветвления.

Практически все основные узлы нового ядра **PIC18CXX2** изменены и дополнены по сравнению с микроконтроллерами серии **PIC16CXX** и **PIC17CXX**. Рассмотрим важнейшие отличия модифицированных узлов.

Основные узлы архитектуры PIC18CXX2

Память

Память организована в виде трех бло-



рес перехода = 2^k , где k — это 21-битовая константа в коде команды GOTO или CALL. Наличие 21-разрядного адреса в коде команды позволяет адресоваться к любой ячейке памяти непосредственно без переключения страниц.

Область памяти выше 200000h отведена под идентификационные номера, регистры калибровки и конфигурации. Читать эту память можно или из программы, ис-

Память данных состоит из регистров общего назначения (GPR) и регистров специальных функций (SFR). Область SFR начинается с адреса 0xFFFFh и растет в сторону младших адресов. Область GPR начинается с 0x000h и растет в сторону старших адресов. Для выбора регистра в адресном пространстве памяти данных используют прямую или косвенную адресацию. Косвенная адресация использует

специальные регистры косвенной адресации (FSR). Каждый FSR содержит полный 12-разрядный адрес адресуемой ячейки памяти. Прямая адресация использует восемь младших разрядов из кода команды и четыре старших разряда адреса данных из регистра BSR.

Набор команд и особенности архитектуры позволяют непосредственно совершать операции с данными всего адресного пространства. Двухсловная команда MOVFF содержит в первом слове код команды и 12-разрядный адрес источника и код NOP с 12-разрядным адресом приемника во втором слове. Команда MOVFF пересылает содержимое любого регистра по произвольному адресу за два командных цикла.

Чтобы гарантировать быстрый доступ к наиболее часто используемым данным и регистрам специальных функций без использования BSR, применено оригинальное и интересное решение под условным названием «банк ускоренного доступа» (Access Bank). В банк ускоренного доступа включено 128 младших регистров GPR и 128 старших SFR. Банк ускоренного доступа очень полезен для оптимизации кода компилятора C. Техника оптимизации мо-

Пример программы очистки первого банка памяти данных

Ассемблер	Комментарий
LFSR 0x100, FSR0	Загрузить FSR0 12-разрядным адресом ff=00
NEXT: CLRF POSTINC0	Очистить INDF и +1 к указателю
BTFSS FSR0H,1	Закончить?
	Да , продолжить выполнение программы

Пример программы сохранения и восстановления регистров WREG, STATUS, BSR

Ассемблерный код	Комментарий
MOVWF W_T	WREG в виртуальный банк
MOVFF STATUS, STATUS_T	STATUS в виртуальный банк
MOVFF BSR, BSR_T	BSR в виртуальный банк
; Программа обработки прерывания	
MOVFF BSR_T, BSR	Восстановить BSR
MOVF W_T, W	Восстановить WREG
MOVFF STATUS_T, STATUS	Восстановить STATUS

жет быть удобна и при написании программы на ассемблере. Специальный «бит доступа» (a=0) в коде команды показывает, что обращение идет именно к банку ускоренного доступа, а содержимое регистра BSR игнорируется. Глобальные переменные помещаются именно в банк ускоренного доступа в так называемую область «AccessRAM».

Память стека

Объем стека был существенно расширен и составил массив из 31 ячейки памяти по 21 биту в каждой. Отличием архитектуры PIC18CXX2 в организации памяти стека является наличие 5-ти битного указателя стека (инициализируется как 00000b при сбросе). Во время выполнения команды CALL указатель стека STKPTR увеличивается на единицу, и ячейка стековой памяти, на которую указывает STKPTR, заполняется адресом возврата из счетчика команд PC (programm counter). При выполнении команды RETURN содержимое стека записывается в PC и значение указателя уменьшается. Специальные биты статуса устанавливаются, если указатель стека выходит за пределы 0–31.

Указатель аппаратного стека легко считать и перезаписать. Используя указатель стека и специальные регистры TOSU, TOSH, TOSL, можно считать и записать вершину стека. Это свойство позволяет разработчику в случае необходимости передавать параметры через аппаратный стек. Необходимо помнить, что во время процедуры перезаписи «в» и «из» стек(а) прерыва-

ния должны быть запрещены. Так как вершина стека (TOS) читаема и перезаписываема, то есть возможность командам PUSH и POP работать с данными стека без нарушения нормального хода программы. Команда PUSH работает аналогично команде CALL за исключением того, что вместо загрузки адреса подпрограммы она просто увеличивает программный счетчик на два. После выполнения команды PUSH вершина стека заполняется необходимыми данными с помощью команд пересылки. После этого вызывается подпрограмма, в которой используются данные из вершины стека. После возврата из подпрограммы новое содержимое вершины стека запоминается в отведенных ячейках, и выполняется команда POP. Ее действия аналогичны команде RETURN, но содержимое вершины стека не загружается в программный счетчик. Описанным выше способом происходит передача параметров через аппаратный стек.

Очень полезной и интересной особенностью архитектуры PIC18CXX2 является наличие теневого «быстрого» одноуровневого стека. Теневой стек используется программой прерывания высокого уровня для одномоментного запоминания содержимого регистров STATUS, WREG, BSR. Если прерывания не используются, то теневой стек можно использовать для сохранения STATUS, WREG, BSR при вызове подпрограммы. Достаточно указать признак FAST при вызове и возврате из подпрограммы:

```
CALL SUB1, FAST;
SUB1*;
RETURN FAST;
```

В результате нет необходимости запоминать в теле подпрограммы SUB1 содержимое регистров STATUS, WREG, BSR. Их значение восстановится автоматически при возврате из подпрограммы.

Как известно, работа микроконтроллера может быть прервана сигналом RESET. К знакомым условиям формирования этого сигнала POR, MCLR, WDT, BOR добавились два условия сброса при работе со стеком: стек полон и стек исчерпан.

Адресация

Прямая адресация

Необходимость адресации большого пространства памяти данных требует создания страничной организации памяти данных. Номер страницы требуется предварительно загрузить в BSR. Четыре бита из регистра BSR и восемь бит адреса из кода команды в результате образуют полный 12-битовый адрес регистра. Необходимо отметить, что без какого-либо переключения банков в процессе выполнения программы можно адресовать до 512 ячеек памяти. Для этого достаточно записать в BSR номер первого банка при начальной инициализации и не менять это значение. Все SFR и регистры GPR0-128 (то есть 256 адресов) будут доступны команде с установленным битом «а», и 256 регистров GPR из соответствующего банка будут доступны при сброшенном бите «а».

Косвенная адресация

Использование косвенной адресации предполагает, что в коде команды адрес данных не фиксируется, а указывается номер регистра косвенной адресации:

```
FSR0 = FSR0H: FSR0L
FSR1 = FSR1H: FSR1L
FSR2 = FSR2H: FSR2L
```

Так как регистр косвенной адресации (указатель) является ячейкой ОЗУ, то его содержимое легко модифицируется программой. Использование указателей упрощает работу с массивами данных (таблицами) и дает возможность организовывать программный стек.

Реальные регистры FSR0, FSR1, FSR2 в ассемблерном коде программы обозначаются по-разному в зависимости от режима адресации. Существует пять вариантов адресации:

- INDF n (изменение содержимого FSR n отсутствует).
- POSTDEC n (уменьшение содержимого FSR n после выполнения команды).

- POSTINC n (увеличение содержимого FSR n после выполнения команды).
- PREINC n (увеличение содержимого FSR n перед обращением к данным).
- PLUSW n (добавить содержимое WREG к содержимому FSR n при обращении к ячейке памяти).

Регистр статуса

В регистре STATUS появились два новых бита N — отрицательный результат операции и OV — переполнение (показывает изменение состояния седьмого бита значения при выполнении арифметических операций). Расширение условий ветвления дает дополнительную гибкость программирования и позволяет писать более эффективный код.

Система прерывания

Система прерывания имеет векторную приоритетную структуру.

Вектор 000008h соответствует прерыванию с высоким приоритетом; 000018h — с низким приоритетом; 000028h — внутрисхемному отладчику (TRAP vector) и, как обычно, 000000h — RESET. Каждому источнику прерывания соответствуют три бита (разрешение, флаг и бит назначения высокого или низкого приоритета). Для прерывания низкого уровня необходимо сохранять содержимое регистров WREG, STATUS, BSR, так как теневой стек должен использоваться только для прерывания высокого уровня.

Генератор тактовых импульсов

В схему генератора тактовых импульсов были внесены изменения, касающиеся выбора режима работы бита RA6. Бит RA6 можно использовать как выходной сигнал тактового генератора OSC2 или как вход/выход общего назначения. Например, для приложений, не требующих высокой точности опорной частоты, рекомендуется использовать тактовый генератор на основе времязадающей RC цепочки в режиме «RC» (на выходе RA6 формируется выходной сигнал тактовой частоты OSC2, деленный на 4) или в режиме «RCIO» (RA6 используется как вход/выход).

В случае использования внешнего источника тактовой частоты для уменьшения тока потребления в режимах «EC» и «ECIO» внутренняя обратная связь между выводами OSC1 и OSC2 разрывается. Внутренний таймер задержки (OST) при включении питания или при выходе из состояния «sleep» не используется, и вывод RA6 в режиме «ECIO» будет работать как порт ввода/вывода.

Новым устройством узла тактового генератора является схема умножения частоты кварцевого генератора на 4 (PLL). Устройство PLL может быть выбрано только при установке режима работы тактового генератора «HS». При использовании кварца на 10 МГц внутренняя частота достигнет 40 МГц. Это очень полезное свойство для тех пользователей, кто хотел бы уменьшить электромагнитное излучение при сохранении высокой производительности выполнения программы.

Для приборов с батарейным питанием важна способность PIC18CXX2 переключать системную тактовую частоту с основного тактового генератора на альтернативный низкочастотный генератор. Обычно в качестве низкочастотного генератора выбирают таймер реального времени T1 на 32 кГц. Системная частота переключается битом «s» в регистре управления тактовым генератором OSCCON. Только восемь тактов требуется для завершения процесса переключения. Специальная внутренняя схема устраняет все проблемы, связанные с переключением источников тактовой частоты.

Таймеры (T0, T1, T3)

Полностью устранены неудобства работы с 16-разрядными значениями. Достаточно настроить режим работы таймера на запись/чтение 16-разрядного значения (T1CON_7=1 для таймера T1), чтобы чтение младшего байта таймера автоматически сохраняло значение старшего байта в аппаратном буфере. Чте-

ние с использованием аппаратного буфера позволяет не беспокоиться о возможном переполнении младшего байта и потере информации в момент чтения. Содержимое буфера читается следующей командой программы.

Запись нового значения в таймер также начинается с записи данных в буферный регистр. Это значение автоматически переписывается в старший байт таймера при записи нового значения в младший байт. Таймер T0 имеет полный 16-разрядный режим, но с целью совместимости оставлен и обычный 8-разрядный. Таймер T1 используется как временная база для CCP1 или CCP1 и CCP2. Таймер T3 конфигурируется как временная база для CCP2 или CCP1 и CCP2.

Отметим, что выходной сигнал модуля CCP в режиме компаратора может быть настроен в режим переключения состояния. Если модуль CCP настроен на работу в режиме компаратора и формирования специального сигнала «special event trigger», то этот сигнал может сбрасывать T1 и автоматически запускать цикл АЦП. В режиме ШИМ временная база формируется таймером T2 модуля CCP.

Порты ввода/вывода

В порты ввода/вывода добавлены регистры защелки. В результате повысилась предсказуемость результата работы алгоритма. При выполнении операции чтение/изменение/запись данные при чтении берутся из регистра защелки. Результат записывается туда же. Выходной сигнал поступает из регистра защелки через буферный усилитель на вывод микроконтроллера. Состояние вывода может быть прочитано ко-

Пример выполнения двухсловной команды MOVFF

Ассемблер	
TSTFSZ REG1	; Состояние RAM ≠ 0
MOVFF REG1,REG2	; Нет , выполняется команда MOVFF
	; 2-й операнд содержит адрес REG2
ADDWF REG3	; Продолжение программы
Ассемблер	
TSTFSZ REG1	; Состояние RAM = 0
MOVFF REG1,REG2	; Да , первое слово пропускается
	; Второе слово команды
	; Выполняется как NOP
ADDWF REG3	; Продолжение программы

мандой чтения, хотя уровень выходного сигнала на выводе реально зависит от величины нагрузки.

Watchdog timer (WDT)

Полностью отделен от таймера T0. Режим работы задается при программировании битом WDTE.

При WDTE=0 watchdog таймер можно включать и выключать под управлением программы с помощью бита SWDTE в регистре WDTCON. Выключение WDT полезно для уменьшения энергопотребления в режиме SLEEP. При выходе из режима SLEEP WDT можно включить.

Система команд PIC18CXX2

Набор команд состоит из 75-ти 16-разрядных команд (одно слово) и четырех 32-разрядных команд (2 слова). Из двух слов состоят команды: MOVFF, CALL, GOTO, LFSR.

Второе слово в команде содержит комбинацию из четырех единиц 1111 в старших разрядах. Такой вид команды является специальным кодом инструкции NOP. Оставшиеся 12 разрядов содержат данные, относящиеся к коду команды. Если первое слово команды выполняется, то используются и данные из второго слова. В том случае если первое слово команды было пропущено после выполнения команды SKIP, то второе слово команды исполняется как NOP.

Добавлены две команды относительного ветвления (BRA label) и вызова подпрограммы (RCAL label), дающие возможность выполнить безусловный переход или вызвать подпрограмму на 1023 команды впереди или на 1024 команды сзади.

Основные данные по микроконтроллерам серии PIC18CXX2

- Набор команд оптимизирован под компилятор языка C.
- Исходный код совместим с сериями **PIC16CXX** и **PIC17CXX**.
- Линейная адресация программной памяти до 2 Мбайт.
- Линейная адресация памяти данных до 4 Кбайт.
- Производительность до 10 MIPS:
 - входная тактовая частота до 40 МГц (от 4 до 10 МГц; при включенном блоке PLL).
- 8-разрядные данные и 16-ти разрядные команды.
- Приоритетная векторная система прерываний.
- Аппаратный умножитель 8x8.

Периферийные устройства:

- ◆ Нагрузочная способность вх/вых 25мА;
- ◆ 3 входа внешних прерываний;
- ◆ Таймер 0 (8/16 бит с 8-ми разрядным делителем);
- ◆ Таймер 1 (16 разрядов);
- ◆ Таймер 2 (8 разрядов + регистр периода ШИМ);
- ◆ Таймер 3 (16 разрядов) с возможностью использования в качестве источника тактовых импульсов для микроконтроллера.
- ◆ Два модуля ССР конфигурируются следующим образом:
 - вход захвата с разрешением 6,25 нс (16 бит);
 - вход сравнения 16 бит с разрешением 100 нс;
 - выход ШИМ с разрешением 1...10 бит (156кГц при 8-ми разрядном разрешении).
- ◆ Последовательный интерфейс (MSSP) имеет два режима работы:
 - 3-проводный SPI (поддерживает все 4 режима);
 - I²C™ «master» и «slave» режимы.
- ◆ Универсальный синхронно-асинхронный интерфейс USART:
 - поддерживает прерывание по 9-му биту адреса.
- ◆ Параллельный интерфейс «Parallel Slave Port».
- ◆ 10-ти разрядный АЦП.
- ◆ PLVD — программируемый датчик уменьшения напряжения питания.
- ◆ BOR — программируемый сброс по падению напряжения питания.
- ◆ Дополнительные периферийные возможности:
 - Watchdog Timer (WDT), режим SLEEP.
- Модуль генератора тактовых импульсов имеет умножитель частоты с фазовой автоподстройкой (x4 при использовании кварцевого резонатора) и возможность переключения на второй источник (32 кГц тактовых импульсов).
- Микропотребление и широкий диапазон напряжения питания от 2.5 до 5.5 вольт.
- Индустриальный и расширенный диапазон рабочих температур.
- Совместимы по выводам с семейством **PIC16CXX** и **PIC17CXX**.

PICMicro	Память программ	Память данных	Количество выводов
PIC18C242	16К байт	512 байт	28
PIC18C252	32К байт	1536 байт	28
PIC18C442	16К байт	512 байт	40
PIC18C452	32К байт	1536 байт	40

Добавлено восемь команд условного ветвления (BC/BNC, BZ/BNZ, BOV/BN OV, BN/BNN). Для указания одного из четырех признаков ветвления (C,Z,OV,N) использованы два бита в коде команды. Соответственно глубина адресации в четыре раза уменьшилась и составляет +128 / - 127 команд.

Организация чтения данных из таблицы

Первый (традиционный для серии PIC16CXX) метод использует комбинацию команды ADDWF_PLCC и команды RETLW_0xnn. Смещение в таблице записано в WREG и показывает, какое число байт должен пропустить счетчик команд. Недостатком метода является то, что только один байт может быть записан в каждой команде, и то, что дополнительно используется один уровень стека.

Более совершенным способом чтения табличных данных является использование специальных регистров указателя TBLPTR (указывает на ячейку памяти) и TABLAT (содержит считываемый или записываемый байт данных). Две команды TBLRD (чтение из памяти программ) и TBLWT (запись в память программ) позволяют читать по два байта из каждой отведенной под команду ячейки памяти. Команды можно использовать с различными режимами автоматической индексации:

• TBLRD* (TBLWT*)	регистр TBLPTR не меняет своего значения
• TBLRD*+ (TBLWT*+)	регистр TBLPTR=TBLPTR+1 после чтения/записи
• TBLRD*- (TBLWT*-)	регистр TBLPTR=TBLPTR-1 после чтения/записи
• TBLRD*+ (TBLWT*+)	регистр TBLPTR=TBLPTR+1 до чтения/записи

Регистр TBLPTR имеет длину в 22 бита и состоит из трех регистров:

