

# Справочник по среднему семейству микроконтроллеров PICmicro™

## Раздел 8. Прерывания

Перевод основывается на технической документации DS33023A  
компании Microchip Technology Incorporated, USA.

© ООО «Микро-Чип»  
Москва - 2002

Распространяется бесплатно.  
Полное или частичное воспроизведение материала допускается только с письменного разрешения  
ООО «Микро-Чип»  
тел. (095) 737-7545  
[www.microchip.ru](http://www.microchip.ru)

# PICmicro™ Mid-Range MCU Family Reference Manual

“All rights reserved. Copyright © 1997, Microchip Technology Incorporated, USA. Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.”

## **Trademarks**

The Microchip name, logo, PIC, KEELOQ, PICMASTER, PICSTART, PRO MATE, and SEEVAL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

MPLAB, PICmicro, ICSP and In-Circuit Serial Programming are trademarks of Microchip Technology Incorporated.

Serialized Quick-Turn Production is a Service Mark of Microchip Technology Incorporated.

All other trademarks mentioned herein are property of their respective companies.

## Содержание

|  |    |
|--|----|
| 8.1 Введение .....                               | 4  |
| 8.2 Регистры управления .....                    | 6  |
| 8.2.1 Регистр <i>INTCON</i> .....                | 6  |
| 8.2.2 Регистры <i>PIE</i> .....                  | 7  |
| 8.2.3 Регистры <i>PIR</i> .....                  | 9  |
| 8.3 Время перехода на обработку прерываний ..... | 11 |
| 8.4 Внешние прерывание INT .....                 | 11 |
| 8.5 Сохранение контекста .....                   | 12 |
| 8.6 Инициализация .....                          | 15 |
| 8.7 Ответы на часто задаваемые вопросы .....     | 17 |
| 8.8 Дополнительная литература .....              | 18 |

## 8.1 Введение

Микроконтроллеры PICmicro среднего семейства могут иметь несколько источников прерываний. Для каждого периферийного модуля назначен отдельный источник прерываний, хотя некоторые периферийные модули содержат несколько источников прерываний (например, модуль USART).

Возможные источники прерываний в микроконтроллерах PICmicro среднего семейства:

- Внешний источник прерываний INT;
- Переполнение таймера TMR0;
- Изменение уровня сигнала на входах PORTB (выводы RB7:RB4);
- Изменение выходного уровня компаратора;
- Прерывание от ведомого параллельного порта;
- Прерывания от USART;
- Прерывание от приемника;
- Прерывание от передатчика;
- Завершение преобразования АЦП;
- Прерывания от LCD;
- Завершение цикла записи в EEPROM память данных;
- Переполнение таймера TMR1;
- Переполнение таймера TMR2;
- Прерывания от модуля CCP;
- Прерывания от модуля SSP.

В микроконтроллерах среднего семейства присутствует как минимум один регистр, управляющий прерываниями. Это регистр:

- INTCON

Если в микроконтроллере есть дополнительные периферийные модули, то в нем будут реализованы регистры для управления прерываниями от периферийных модулей (регистр маски, чтобы разрешить/запретить прерывания; регистр флагов прерываний, указывающий на возникшее прерывание). В зависимости от типа микроконтроллера в нем могут быть реализованы регистры:

- PIE1
- PIR1
- PIE2
- PIR2

В этой документации мы будем обращаться к этим регистрам как PIR и PIE. Если новые микроконтроллеры будут содержать больше источников прерываний, то будут реализованы регистры PIR3 и PIE3.

Регистр управления прерываниями INTCON содержит индивидуальные биты флагов прерываний для ядра микроконтроллера, биты маски разрешения прерываний, а также бит глобального разрешения прерываний.

Если бит глобального разрешения прерываний GIE (INTCON<7>) установлен в '1', то разрешены все немаскированные прерывания. Все прерывания запрещены, если GIE (INTCON<7>) сброшен в '0'. Прерывания индивидуально запрещены сбросом соответствующего бита в регистре INTCON. При сбросе микроконтроллера бит GIE сбрасывается в '0'.

Возврат из обработки прерываний выполняется по команде RETFIE, при этом происходит установка бита GIE в '1', что позволяет обработать любое отложенное прерывание.

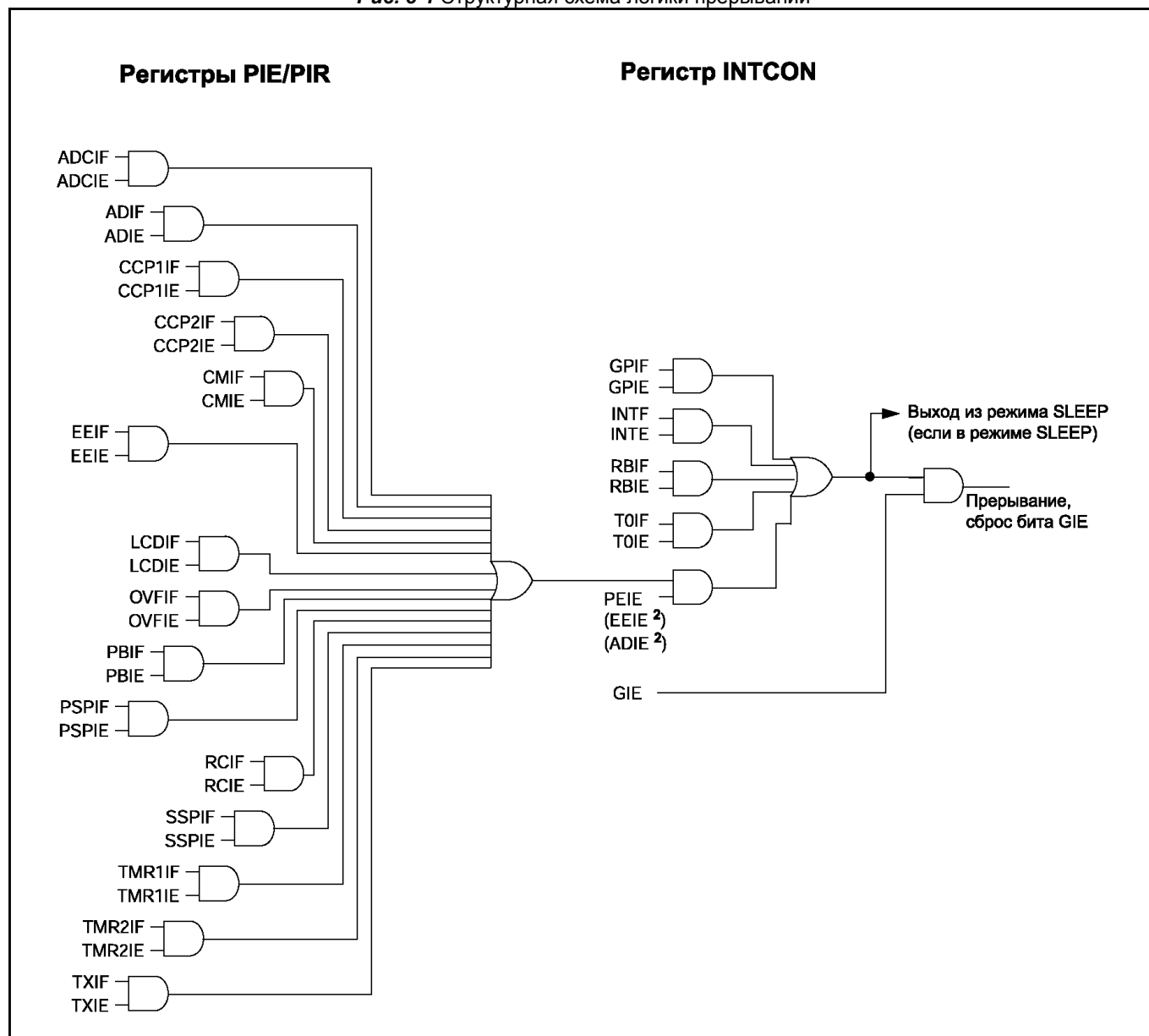
Регистр INCON содержит биты управления следующими прерываниями: внешнее прерывание INT; изменение сигнала на входах RB7:RB4; пополнение TMR0. В регистре INCON также расположен бит разрешения прерываний от периферийных модулей PEIE. Если PEIE=1, то разрешен переход по вектору прерываний при возникновении периферийного прерывания.

При обработке прерываний бит GIE=0, чтобы предотвратить повторную загрузку счетчика команд PC в стек и запись в PC адреса вектора прерываний 0004h. В обработчике прерываний источник прерываний может быть идентифицирован проверкой флагов прерываний. Как правило флаги прерываний должны быть сброшены в обработчике прерываний перед разрешением прерываний в системе, чтобы предотвратить повторный переход на обработку прерываний. Индивидуальные флаги прерываний устанавливаются независимо от состояния бита общего разрешения прерываний GIE и соответствующих битов маски.

**Примечание 1.** Индивидуальные флаги прерываний устанавливаются независимо от состояния бита общего разрешения прерываний GIE и соответствующих битов маски.

**Примечание 2.** При выполнении команды, сбрасывающей бит GIE в '0', любое прерывание, ожидающее выполнения в следующем машинном цикле, игнорируется. Микроконтроллер выполнит пустой цикл NOP после команды, сбрасывающей бит GIE в '0'. Игнорированные прерывания ставятся в ожидание выполнения, пока бит GIE не будет установлен в '1'.

Рис. 8-1 Структурная схема логики прерываний



## Примечания:

1. На рисунке показаны все возможные источники прерываний для микроконтроллеров PICmicro среднего семейства. Наличие управляющих битов в микроконтроллере зависит от реализованных периферийных модулей. Смотрите техническую документацию на микроконтроллер.
2. Часть микроконтроллеров среднего семейства имеют только один периферийный модуль. В этих микроконтроллерах нет бита PEIE, а реализован бит разрешения прерываний от периферийного модуля в регистре INTCON.

## 8.2 Регистры управления

Как правило микроконтроллеры среднего семейства имеют три регистра управления прерываниями. Регистр INTCON содержит бит глобального разрешения прерываний GIE, а также бит разрешения прерываний от периферийных модулей PEIE. В паре регистров PIE/PIR размещаются индивидуальные биты разрешения прерываний от периферийных модулей и флаги возникшего прерывания.

### 8.2.1 Регистр INTCON

Регистр INTCON доступен для записи и чтения. В этом регистре содержатся различные биты разрешений и флагов прерываний.

**Примечание.** Флаги прерываний устанавливаются при возникновении условий прерываний вне зависимости от соответствующих битов разрешения и бита общего разрешения прерываний GIE (INTCON<7>). Это позволяет выполнять программный контроль возникновения условия прерываний.

#### Регистр INTCON

| R/W-0  | R/W-0               | R/W-0 | R/W-0               | R/W-0                 | R/W-0 | R/W-0               | R/W-x                 |
|--|---------------------|-------|---------------------|-----------------------|-------|---------------------|-----------------------|
| GIE  | PEIE <sup>(3)</sup> | TOIE  | INTE <sup>(2)</sup> | RBIE <sup>(1,2)</sup> | TOIF  | INTF <sup>(2)</sup> | RBIF <sup>(1,2)</sup> |
| Бит 7  |                     |       |                     |                       |       |                     | Бит 0                 |
| <p>R – чтение бита<br/>W – запись бита<br/>U – не реализовано, читается как 0<br/>-n – значение после POR<br/>-x – неизвестное значение после POR</p>  |                     |       |                     |                       |       |                     |                       |
| <p>бит 7: <b>GIE</b>: Глобальное разрешение прерываний<br/>1 = разрешены все немаскированные прерывания<br/>0 = все прерывания запрещены</p> <p>бит 6: <b>PEIE</b>: Разрешение прерываний от периферийных модулей<br/>1 = разрешены все немаскированные прерывания периферийных модулей<br/>0 = прерывания от периферийных модулей запрещены</p> <p>бит 5: <b>TOIE</b>: Разрешение прерывания по переполнению TMR0<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p>бит 4: <b>INTE</b>: Разрешение внешнего прерывания INT<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p>бит 3: <b>RBIE<sup>(1)</sup></b>: Разрешение прерывания по изменению сигнала на входах RB7:RB4 PORTB<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p>бит 2: <b>TOIF</b>: Флаг прерывания по переполнению TMR0<br/>1 = произошло переполнение TMR0 (сбрасывается программно)<br/>0 = переполнения TMR0 не было</p> <p>бит 1: <b>INTF</b>: Флаг внешнего прерывания INT<br/>1 = выполнено условие внешнего прерывания на выводе RB0/INT (сбрасывается программно)<br/>0 = внешнего прерывания не было</p> <p>бит 0: <b>RBIF<sup>(1)</sup></b>: Флаг прерывания по изменению уровня сигнала на входах RB7:RB4 PORTB<br/>1 = зафиксировано изменение уровня сигнала на одном из входов RB7:RB4 (сбрасывается программно)<br/>0 = не было изменения уровня сигнала ни на одном из входов RB7:RB4</p> |                     |       |                     |                       |       |                     |                       |

**Примечание 1.** В некоторых микроконтроллерах бит RBIE может быть заменен битом GPIE, а бит RBIF может быть заменен битом GPIF.

**Примечание 2.** Некоторые микроконтроллеры могут не содержать эту функцию.

**Примечание 3.** В микроконтроллерах с одним периферийным модулем этот бит может быть EEIE или ADIE.

## 8.2.2 Регистры PIE

В зависимости от числа источников прерываний периферийных модулей в микроконтроллере могут содержаться регистры разрешения периферийных прерываний PIE1 и PIE2. В этих регистрах располагаются индивидуальные биты разрешения прерываний от периферийных модулей. В данной документации эти регистры будут обозначаться как PIE. Если в микроконтроллере присутствует регистр PIE, то необходимо установить в '1' бит PEIE для разрешения прерываний от периферийных модулей.

**Примечание.** Чтобы разрешить любое периферийное прерывание необходимо установить в '1' бит PEIE(INTCON<6>).

Расположение битов в регистрах PIE стандартизовано, однако во вновь разрабатываемых микроконтроллерах оно может измениться. Не будет возникать проблем с разрядным размещением битов в управляющих регистрах, если Вы будете использовать дополнительный файл от Microchip Inc. с символьным обозначением битов. Это позволит ассемблеру выполнить компиляцию исходного текста программы с правильным указанием адреса регистра и номера бита.

## Регистры PIE

| R/W-0          |  |
|----------------|--|
| (Примечание 1) |  |
| Бит 7          | Бит 0  |
| бит:           | <p><b>TMR1IE:</b> Разрешение прерывания по переполнению TMR1<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>TMR2IE:</b> Разрешение прерывания по переполнению TMR2<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>CCP1IE:</b> Разрешение прерывания от модуля CCP1<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>CCP2IE:</b> Разрешение прерывания от модуля CCP2<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>SSPIE:</b> Разрешение прерывания от модуля синхронного последовательного порта<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>RCIE:</b> Разрешение прерывания от приемника USART<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>TXIE:</b> Разрешение прерывания от передатчика USART<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>ADIE:</b> Разрешение прерывания по окончании преобразования АЦП<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>ADCIE:</b> Разрешение прерывания по окончании преобразования АЦП<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>OVFIE:</b> Разрешение прерывания по переполнению таймера АЦП<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>PSPIE:</b> Разрешение прерывания записи/чтения ведомого параллельного порта<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>EEIE:</b> Разрешение прерывания по окончании записи в EEPROM данных<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>LCDIE:</b> Разрешение прерывания от модуля LCD<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> <p><b>CMIE:</b> Разрешение прерывания от модуля компараторов<br/>1 = прерывание разрешено<br/>0 = прерывание запрещено</p> |

R – чтение бита  
W – запись бита  
U – не реализовано,  
читается как 0  
-n – значение после POR  
-x – неизвестное  
значение после POR

**Примечание 1.** Размещение битов в управляющих регистрах смотрите в технической документации на микроконтроллер.



### 8.2.3 Регистры PIR

В зависимости от числа источников прерываний периферийных модулей в микроконтроллере могут содержаться регистры флагов прерываний от периферийных модулей PIR1 и PIR2. В этих регистрах располагаются индивидуальные биты флагов прерываний от периферийных модулей. В данной документации эти регистры будут обозначаться как PIR.

**Примечание 1.** Флаги прерываний устанавливаются при возникновении условий прерываний вне зависимости от соответствующих битов разрешения и бита общего разрешения прерываний GIE (INTCON<7>).

**Примечание 2.** Программное обеспечение пользователя должно сбрасывать соответствующие флаги прерываний в '0' перед разрешением прерывания и в подпрограмме обработки прерывания.

Расположение битов в регистрах PIR стандартизовано, однако во вновь разрабатываемых микроконтроллерах оно может измениться. Не будет возникать проблем с разрядным размещением битов в управляющих регистрах, если Вы будете использовать дополнительный файл от Microchip Inc. с символьным обозначением битов. Это позволит ассемблеру выполнить компиляцию исходного текста программы с правильным указанием адреса регистра и номера бита.

#### Регистры PIR

| R/W-0  |   |
|--|---|
| (Примечание 1)   |   |
| Бит 7  | Бит 0   |
| <p>бит: <b>TMR1IF</b>: Флаг прерывания по переполнению TMR1<br/>           1 = произошло переполнение TMR1 (сбрасывается программно)<br/>           0 = переполнения TMR1 не было</p> <p>бит: <b>TMR2IF</b>: Флаг прерывания по переполнению TMR2<br/>           1 = произошло переполнение TMR2 (сбрасывается программно)<br/>           0 = переполнения TMR2 не было</p> <p>бит: <b>CCP1IF</b>: Флаг прерывания от модуля CCP1<br/> <u>Режим захвата</u><br/>           1 = выполнен захват значения TMR1 (сбрасывается программно)<br/>           0 = захвата значения TMR1 не происходило<br/> <u>Режим сравнения</u><br/>           1 = значение TMR1 достигло указанного в регистрах CCPR1H:CCPR1L(сбрасывается программно)<br/>           0 = значение TMR1 не достигло указанного в регистрах CCPR1H:CCPR1L<br/> <u>ШИМ режим</u><br/>           Не используется</p> <p>бит: <b>CCP2IF</b>: Флаг прерывания от модуля CCP2<br/> <u>Режим захвата</u><br/>           1 = выполнен захват значения TMR1 (сбрасывается программно)<br/>           0 = захвата значения TMR1 не происходило<br/> <u>Режим сравнения</u><br/>           1 = значение TMR1 достигло указанного в регистрах CCPR2H:CCPR2L(сбрасывается программно)<br/>           0 = значение TMR1 не достигло указанного в регистрах CCPR2H:CCPR2L<br/> <u>ШИМ режим</u><br/>           Не используется</p> <p>бит: <b>SSPIF</b>: Флаг прерываний от модуля SSP<br/>           1 = выполнено условие возникновения прерывания от модуля SSP (сбрасывается программно)<br/>           0 = условие возникновения прерывания от модуля SSP не выполнено</p> | <p>R – чтение бита<br/>           W – запись бита<br/>           U – не реализовано,<br/>           читается как 0<br/>           -n – значение после POR<br/>           -x – неизвестное<br/>           значение после POR</p> |

**Примечание 1.** Размещение битов в управляющих регистрах смотрите в технической документации на микроконтроллер.

## Регистры PIR (продолжение)

| R/W-0          |  |
|----------------|--|
| (Примечание 1) |  |
| Бит 7          | Бит 0  |
| бит:           | <p><b>RCIF:</b> Флаг прерывания от приемника USART<br/>1 = буфер приемника USART полон (сбрасывается чтением регистра RCREG)<br/>0 = буфер приемника USART пуст</p> <p><b>TXIF:</b> Флаг прерывания от передатчика USART<br/>1 = буфер передатчика USART пуст (сбрасывается записью в регистр TXREG)<br/>0 = буфер передатчика USART полон</p> <p><b>ADIF:</b> Флаг прерывания от модуля АЦП<br/>1 = преобразование АЦП завершено (сбрасывается программно)<br/>0 = преобразование АЦП не завершено</p> <p><b>ADCIF:</b> Флаг прерывания от модуля АЦП<br/>1 = преобразование АЦП завершено (сбрасывается программно)<br/>0 = преобразование АЦП не завершено</p> <p><b>OVIF:</b> Флаг прерывания по переполнению таймера АЦП<br/>1 = произошло переполнение таймера АЦП (сбрасывается программно)<br/>0 = переполнение таймера АЦП не происходило</p> <p><b>PSPIF:</b> Флаг прерывания от ведомого параллельного порта<br/>1 = произошла операция чтения или записи (сбрасывается программно)<br/>0 = операции чтения или записи не происходило</p> <p><b>EEIF:</b> Флаг прерывания по окончании записи в EEPROM данных<br/>1 = запись в EEPROM данных завершена (сбрасывается программно)<br/>0 = запись в EEPROM данных не завершена или не была начата</p> <p><b>LCDIF:</b> Флаг прерывания от модуля LCD<br/>1 = возникло прерывание от модуля LCD (сбрасывается программно)<br/>0 = прерывания от модуля LCD не было</p> <p><b>CMIF:</b> Флаг прерывания от модуля компараторов<br/>1 = возникло прерывание от модуля компараторов (сбрасывается программно)<br/>0 = прерывания от модуля компараторов не было</p> |

R – чтение бита  
W – запись бита  
U – не реализовано,  
читается как 0  
-n – значение после POR  
-x – неизвестное  
значение после POR

**Примечание 1.** Размещение битов в управляющих регистрах смотрите в технической документации на микроконтроллер.

### 8.3 Время перехода на обработку прерываний

Временем переход на обработку прерываний считается интервал времени от установки флага возникшего прерывания до момента начала выполнения команды по адресу 0004h в памяти программ (если прерывание разрешено).

Для синхронных прерываний (внутренних) время перехода равно  $3T_{CY}$ .

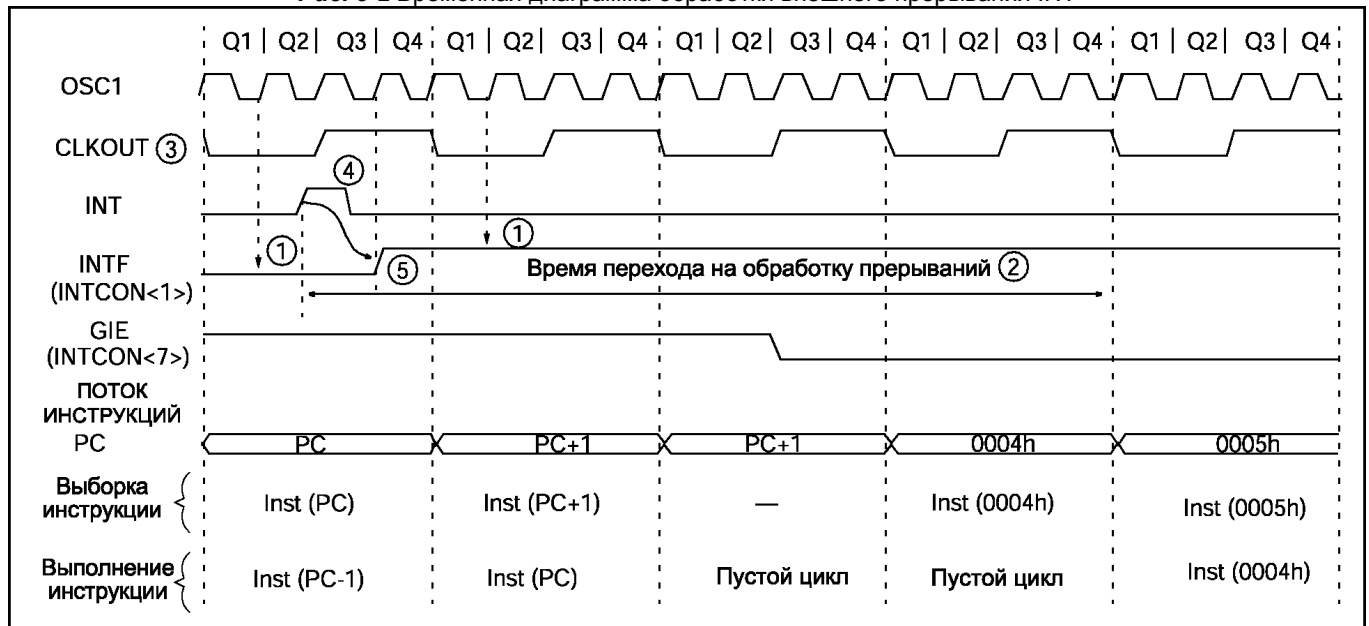
Для асинхронных прерываний (внешних), например внешнее прерывание INT или изменение уровня сигнала на входах RB7:RB4, время перехода на обработку прерываний будет составлять 3 -  $3.75T_{CY}$ . Точное время перехода на обработку прерываний зависит от момента возникновения прерывания (см. рисунок 8-2).

Время перехода на обработку прерываний одинаково для одно и двух цикловых команд.

### 8.4 Внешние прерывание INT

Внешнее прерывание с входа RB0/INT происходит: по переднему фронту сигнала, если бит INTEDG (OPTION\_REG<6>) установлен в '1'; по заднему фронту сигнала, если бит INTEDG сброшен в '0'. Когда активный фронт сигнала появляется на входе RB0/INT, бит INTF (INTCON<1>) устанавливается в '1'. Прерывание может быть запрещено сбросом бита INTE (INTCON<4>) в '0'. Флаг прерывания INTF должен быть сброшен программно в подпрограмме обработки прерываний. Прерывание INT может вывести микроконтроллер из режима SLEEP, если бит INTE=1 до перехода в режим SLEEP. Состояние бита GIE определяет: переходить ли на подпрограмму обработки прерываний после выхода из режима SLEEP. Дополнительную информацию смотрите в разделах, описывающих сторожевой таймер WDT и режим энергосбережения SLEEP.

Рис. 8-2 Временная диаграмма обработки внешнего прерывания INT



Примечания:

1. Флаг INTF проверяется в такте Q1.
2. Время перехода на обработку прерываний: не синхронизированный сигнал  $3-4T_{CY}$ ; синхронизированный сигнал  $3T_{CY}$ . Время перехода не зависит от выполняемой инструкции (одно или двух цикловая команда).
3. CLKOUT доступен только в RC режиме генератора.
4. Минимальную длительность импульса INT смотрите в разделе "Электрические характеристики".
5. Флаг INTF может быть установлен в любой момент (Q1-Q4).

## 8.5 Сохранение контекста

При переходе на подпрограмму обработки прерываний в стеке аппаратно сохраняется только адрес возврата. Как правило, дополнительно необходимо сохранять ключевые регистры (например W, STATUS), что выполняется программно.

Операция сохранения значения регистров обычно обозначается "PUSH", а восстановление значения регистров обозначается "POP". Обратите внимание, что PUSH, POP не являются мнемоникой команд, а лишь обозначают действие, которое может быть выполнено последовательностью команд. Для упрощения текста программы можно эти сегменты кода программы представить в виде макросов (описание использования макрокоманд смотрите в документации "Руководство пользователя MPASM").

В примере 8-1 показано восстановление регистров STATUS, W для микроконтроллеров с общим ОЗУ (например, PIC16C77). Регистр W\_TEMP должен быть определен во всех банках памяти с одинаковым смещением относительно начала банка. Регистр STATUS\_TEMP может быть определен в одном банке памяти данных. В примере 8-1 регистр STATUS\_TEMP определен в банке 0.

Последовательность операций примера 8-1:

1. Сохранить регистр W независимо от текущего банка памяти.
2. Сохранить регистр STATUS в банке 0.
3. Выполнить подпрограмму обработки прерываний.
4. Восстановить регистр STATUS и текущий банк памяти данных.
5. Восстановить регистр W.

Если необходимо сохранить и другие регистры, то сохранение нужно выполнять после сохранения регистра STATUS (шаг 2), а восстановление перед восстановлением STATUS (шаг 4).

**Пример 8-1** Сохранение регистров STATUS, W в ОЗУ (для микроконтроллеров с общим ОЗУ)

```

MOVWF    W_TEMP          ; Копировать W во временный регистр
                          ; независимо от текущего банка
SWAPF    STATUS, W       ; Обменять полубайты в регистре STATUS
                          ; и записать в W
MOVWF    STATUS_TEMP      ; Сохранить STATUS во временном регистре
                          ; банка 0
:
: (Выполнить код подпрограммы обработки прерываний)
:
SWAPF    STATUS_TEMP, W   ; Обменять полубайты оригинального значения STATUS
                          ; и записать в W (восстановить текущий банк)
MOVWF    STATUS           ; Восстановить значение STATUS
                          ; из регистра W
SWAPF    W_TEMP, F       ; Обменять полубайты в регистре W_TEMP и сохранить
                          ; результат в W_TEMP
SWAPF    W_TEMP, W       ; Обменять полубайты в регистре W_TEMP и восстановить
                          ; оригинальное значение W без воздействия на STATUS

```

В примере 8-2 показано восстановление регистров STATUS, W для микроконтроллеров без общего ОЗУ (например PIC16C74A). Регистр W\_TEMP должен быть определен во всех банках памяти с одинаковым смещением относительно начала банка. Регистр STATUS\_TEMP может быть определен в одном банке памяти данных. В примере 8-2 регистр STATUS\_TEMP определен в банке 0.

Последовательность операций примера 8-2:

1. Сохранить регистр W независимо от текущего банка памяти.
2. Сохранить регистр STATUS в банке 0.
3. Выполнить подпрограмму обработки прерываний.
4. Восстановить регистр STATUS и текущий банк памяти данных.
5. Восстановить регистр W.

Если необходимо сохранить и другие регистры, то сохранение нужно выполнять после сохранения регистра STATUS (шаг 2), а восстановление перед восстановлением STATUS (шаг 4).

**Пример 8-2** Сохранение регистров STATUS, W в ОЗУ (для микроконтроллеров без общего ОЗУ)

```

MOVWF    W_TEMP                ; Копировать W во временный регистр
                                ; независимо от текущего банка
SWAPF    STATUS, W              ; Обменять полубайты в регистре STATUS
                                ; и записать в W
BCF       STATUS, RPO           ; Выбрать банк 0
MOVWF    STATUS_TEMP            ; Сохранить STATUS во временном регистре
                                ; банка 0
:
: (Выполнить код подпрограммы обработки прерываний)
:
SWAPF    STATUS_TEMP, W         ; Обменять полубайты оригинального значения STATUS
                                ; и записать в W (восстановить текущий банк)
MOVWF    STATUS                 ; Восстановить значение STATUS
                                ; из регистра W
SWAPF    W_TEMP, F              ; Обменять полубайты в регистре W_TEMP и сохранить
                                ; результат в W_TEMP
SWAPF    W_TEMP, W              ; Обменять полубайты в регистре W_TEMP и восстановить
                                ; оригинальное значение W без воздействия на STATUS

```

В примере 8-3 показано восстановление регистров STATUS, W для микроконтроллеров с универсальным ОЗУ только в банке 0 (например PIC16C620). Банк памяти должен быть проверен перед сохранением любого регистра. Регистр W\_TEMP должен быть определен во всех банках памяти с одинаковым смещением относительно начала банка. Регистр STATUS\_TEMP может быть определен в одном банке памяти данных. В примере 8-3 регистр STATUS\_TEMP определен в банке 0.

Последовательность операций примера 8-3:

1. Проверить текущий банк.
2. Сохранить регистр W независимо от текущего банка памяти.
3. Сохранить регистр STATUS в банке 0.
4. Выполнить подпрограмму обработки прерываний.
5. Восстановить регистр STATUS и текущий банк памяти данных.
6. Восстановить регистр W.

Если необходимо сохранить и другие регистры, то сохранение нужно выполнять после сохранения регистра STATUS (шаг 3), а восстановление перед восстановлением STATUS (шаг 5).

**Пример 8-3** Сохранение регистров STATUS, W в ОЗУ (для микроконтроллеров с универсальным ОЗУ, только в банке 0)

```

Push
    BTFSS    STATUS, RP0           ; В банке 0?
    GOTO    RP0CLEAR           ; Да
    BCF     STATUS, RP0         ; Нет
    MOVWF   W_TEMP             ; Сохранить регистр W
    SWAPF   STATUS, W           ; Обменять полубайты в регистре STATUS
    MOVWF   STATUS_TEMP        ; и записать в STATUS_TEMP
    BSF     STATUS_TEMP, 5      ; Установить бит RP0 в сохраненном значении STATUS
    GOTO    ISR_Code           ; Сохранение регистров завершено
RP0CLEAR
    MOVWF   W_TEMP             ; Сохранить регистр W
    SWAPF   STATUS, W           ; Обменять полубайты в регистре STATUS
    MOVWF   STATUS_TEMP        ; и записать в STATUS_TEMP
                                ;
ISR_Code
:
: (Выполнить код подпрограммы обработки прерываний)
:
Pop
    SWAPF   STATUS_TEMP, W      ; Восстановить значение STATUS
    MOVWF   STATUS             ;
    BTFSS   STATUS, RP0         ; Банк 1?
    GOTO    Restore_WREG       ; Нет
    BCF     STATUS, RP0         ; Да
    SWAPF   W_TEMP, F          ; Восстановить значение регистра W
    SWAPF   W_TEMP, W          ;
    BSF     STATUS, RP0         ; Восстановить банк 1
    RETFIE  ; Восстановление регистров завершено
Restore_WREG
    SWAPF   W_TEMP, F          ; Восстановить значение регистра W
    SWAPF   W_TEMP, W          ;
    RETFIE  ; Восстановление регистров завершено

```

## 8.6 Инициализация

В примере 8-4 показана инициализация прерываний, где PIE1\_MASK - значение, записываемое в регистр маски периферийных прерываний.

Создание макрокоманд сохранения/восстановления значений регистров показано в примере 8-5. Макрокоманды должны быть определены прежде, чем они будут использоваться. Для простоты отладки текста программы макрокоманды рекомендуется помещать в отдельные файлы, включаемые в исходный файл программы, до применения макрокоманды. Рекомендуется включать файлы с макрокомандами в начале исходного файла (см. пример 8-6).

В примере 8-7 представлена типовая структура проверки возникшего прерывания. В этом примере используются макрокоманды для сохранения значения регистров перед выполнением кода обработки прерываний.

### Пример 8-4 Инициализация прерываний

```

PIE1_MASK1      EQU B'01101010'          ; Значение для регистра
:                                                       ; маски прерываний
:
CLRF            STATUS                    ; Банк 0
CLRF            INTCON                    ; Выключить прерывания и сбросить флаги
CLRF            PIR1                      ; Сбросить все флаги
BSF             STATUS, RP0                ; Банк 1
MOVLW          PIE1_MASK1                 ; Записать маску прерываний в регистр PIE1
MOVWF          PIE1                       ;
BCF             STATUS, RP0                ; Банк 0
BSF             INTCON, GIE                ; Включить прерывания

```

### Пример 8-5 Макрокоманды сохранения/восстановления значения регистров

```

PUSH_MACRO      MACRO                    ; Макрос сохранения регистров
MOVWF           W_TEMP                    ; Копировать W во временный регистр
; независимо от текущего банка
SWAPF           STATUS, W                 ; Обменять полубайты в регистре STATUS
; и записать в W
MOVWF           STATUS_TEMP               ; Сохранить STATUS во временном регистре
; банка 0
ENDM            ; Конец макроса
;
POP_MACRO        MACRO                    ; Макрос восстановления регистров
SWAPF           STATUS_TEMP, W            ; Обменять полубайты оригинального значения STATUS
; и записать в W (восстановить текущий банк)
MOVWF           STATUS                    ; Восстановить значение STATUS
; из регистра W
SWAPF           W_TEMP, F                 ; Обменять полубайты в регистре W_TEMP и сохранить
; результат в W_TEMP
SWAPF           W_TEMP, W                 ; Обменять полубайты в регистре W_TEMP и восстановить
; оригинальное значение W без воздействия на STATUS
ENDM            ; Конец макроса

```

**Пример 8-6** Шаблон исходного файла

```

LIST      p = p16C77          ; Список директив
;
;
; #INCLUDE <P16C77.INC>      ; Дополнительный файл к микроконтроллеру
;
; #INCLUDE <MY_STD.MAC>      ; Подключить файл стандартных макрокоманд
; #INCLUDE <APP.MAC>         ; подключить файл специальных макрокоманд
;                             ; для этого приложения
;
; Определение битов конфигурации
__CONFIG _XT_OSC & _PWRTE_ON & _BODEN_OFF & _CP_OFF & _WDT_ON
;
org 0x00          ; Начало памяти программ
RESET_ADDR :     ; Первая выполняемая инструкция после сброса
end

```

**Пример 8-7** Типовая обработка прерываний

```

org      ISR_ADDR          ;
PUSH_MACRO          ; Макрокоманда сохранения регистров,
                    ; или другой код
CLRF      STATUS          ; Банк 0
BTFSF    PIR1, TMR1IF     ; Прерывание от TMR1?
GOTO     T1_INT           ; Да
BTFSF    PIR1, ADIF        ; Нет, прерывание от АЦП?
GOTO     AD_INT           ; Да, от АЦП
:          ; Нет, проверка других источников прерываний
:          ;
BTFSF    PIR1, LCDIF       ; Нет, прерывание от LCD?
GOTO     LCD_INT          ; Да, прерывание от LCD
BTFSF    INTCON, RBIF      ; Нет, прерывание по изменению сигнала на RB7:RB6?
GOTO     PORTB_INT        ; Да, прерывание по изменению сигнала на RB7:RB6
INT_ERROR_LP1      ; Нет, процедура восстановления при ошибке
GOTO     INT_ERROR_LP1    ; Здесь должна располагаться процедура
                    ; обработки возникновения неожиданного
                    ; прерывания
T1_INT          ; Обработка прерываний от TMR1
:          ;
BCF      PIR1, TMR1IF      ; Сброс флага прерывания от TMR1
GOTO     END_ISR          ; Завершение обработки прерываний
AD_INT          ; Обработка прерываний от АЦП
:          ;
BCF      PIR1, ADIF        ; Сброс флага прерывания от АЦП
GOTO     END_ISR          ; Завершение обработки прерываний
LCD_INT          ; Обработка прерываний от LCD
:          ;
BCF      PIR1, LCDIF       ; Сброс флага прерывания от LCD
GOTO     END_ISR          ; Завершение обработки прерываний
PORTB_INT        ; Обработка прерываний по изменению сигнала на RB7:RB6
:          ;
END_ISR          ;
POP_MACRO          ; Макрокоманда восстановления значения регистров
                    ; или другой код
RETFIE        ; Возвращение из обработки прерываний,
                    ; разрешение прерываний

```



## 8.7 Ответы на часто задаваемые вопросы

Если вы не найдете ответа на Ваш вопрос в этой главе раздела, задайте его, написав нам письмо по адресу [support@microchip.ru](mailto:support@microchip.ru).

**Вопрос 1:** Алгоритм программы дает неправильные результаты.

**Ответ 1:**

При разрешенных прерываниях во время выполнения алгоритма необходимо гарантировать, что регистры, используемые алгоритмом, сохраняются и восстанавливаются в подпрограмме обработки прерываний. Проверьте подпрограмму обработки прерываний, т.к. некоторые регистры могут быть изменены.

**Вопрос 2:** Выполнение программы прекращается, что может быть причиной?

**Ответ 2:**

Если в программе используются прерывания, то необходимо следить за тем, чтобы перед выходом из обработки прерываний (выполнения команды RETFIE) был сброшен флаг источника прерываний. Если флаг прерывания останется установленным, то после исполнения команды RETFIE выполнение программы опять перейдет по вектору прерываний и останется невыполненным разрешенное прерывание.

## 8.8 Дополнительная литература

Дополнительная литература и примеры применения, связанные с этим разделом документации. Примеры применения не могут использоваться для всех микроконтроллеров среднего семейства (PIC16CXXX). Как правило примеры применения написаны для конкретной группы микроконтроллеров, но принципы примеров могут использоваться, сделав незначительные изменения (с учетом существующих ограничений).

Документы, связанные с прерываниями в микроконтроллерах PICmicro MCU:

| Документ   | Номер |
|--|-------|
| Using the PortB Interrupt On Change as an External Interrupt<br>Применение внешнего прерывания и прерывания по изменению сигнала на входах PORTB | AN566 |

## Уважаемые господа!

ООО «Микро-Чип» поставляет полную номенклатуру комплектующих фирмы **Microchip Technology Inc** и осуществляет качественную техническую поддержку на русском языке.

С техническими вопросами Вы можете обращаться по адресу [support@microchip.ru](mailto:support@microchip.ru)

По вопросам поставок комплектующих Вы можете обращаться к нам по телефонам:  
**(095) 963-9601**  
**(095) 737-7545**  
и адресу [sales@microchip.ru](mailto:sales@microchip.ru)

На сайте  
[www.microchip.ru](http://www.microchip.ru)

Вы можете узнать последние новости нашей фирмы, найти техническую документацию и информацию по наличию комплектующих на складе.