

**Katedra Optoelektroniki i Systemów  
Elektronicznych**

**Wydział Elektroniki, Telekomunikacji  
i Informatyki  
Politechniki Gdańskiej**



## **LABORATORIUM MIKROSTEROWNIKI I MIKROSYSTEMY ROZPROSZONE**

**Dokumentacja mikrokontrolera PIC18F452  
firmy Microchip**

**Gdańsk 2008**

## 1 Główne cechy mikrokontrolera PIC 18F452:

- architektura zoptymalizowana do języka programowania C,
  - kod źródłowy zgodny z rodziną PIC16 i PIC17,
- pamięć programu 32Kbajtów,
- pamięć danych 1.5Kbajta,
- prędkość operacji:
  - DC – 40MHz – wejście zegarowe,
  - 4MHz – 10MHz – wejście zegarowe z aktywną pętlą PLL,
- instrukcje 16-bitowe, dane 8-bitowe,
- 100.000 cykli zapisu/odczytu pamięci FLASH,
- 1.000.000 cykli zapisu/odczytu pamięci EEPROM,
- przechowywanie danych w pamięci FLASH, EEPROM >40 lat,
- Power-on-reset (układ resetu po włączeniu zasilania) (POR),
- Power-up-timer (PWRT) oraz Oscillator Start-up Timer (OST),
- Watchdog Timer (WDT) z własnym oscylatorem RC,
- programowalne zabezpieczenie kodu przed odczytem,
- tryb zmniejszonego poboru mocy SLEEP,
- możliwość wyboru opcji oscylatora,
- możliwość programowania w systemie poprzez dwa piny (ICSP),

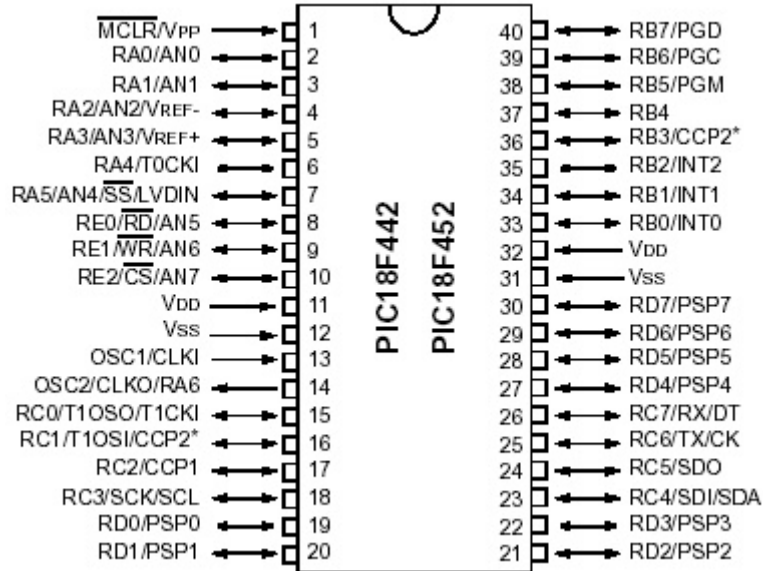
### Charakterystyka urządzeń peryferyjnych:

- trzy zewnętrzne wejścia przerwaniowe,
- Timer0 – 8-bitowy zegar / 16-bitowy licznik z 8-bitowym programowalnym preskalerem,
- Timer1 – 16-bitowy zegar/licznik,
- Timer2 – 8-bitowy zegar/licznik z 8-bitowym rejestrem okresu ( z przeznaczeniem dla PWM),
- Timer3 – 16-bitowy zegar/licznik,
- opcja drugiego oscylatora Timer1/Timer3,
- dwa moduły przechwytyjące/porównujące/PWM mogą być skonfigurowane jako:
  - przechwytywanie 16-bitowe o maksymalnej rozdzielczości 6.25ns,
  - porównywanie 16-bitowe o maksymalnej rozdzielczości 100ns,
  - PWM o rozdzielczości 1 do 10 bitów
    - ◆ maksymalna częstotliwość przy rozdzielczości 8-bitów – 156kHz,
    - ◆ maksymalna częstotliwość przy rozdzielczości 10-bitów – 39kHz,
- synchroniczny port szeregowy (Master Synchronous Serial Port) o dwóch trybach pracy:
  - 3-przewodowy interfejs SPI,
  - tryb I<sup>2</sup>C™ (tryb Master/Slave),
- tryb USART (wspomaga RS-465 i RS-232),
- port równoległy (PSP Parallel Slave Port),

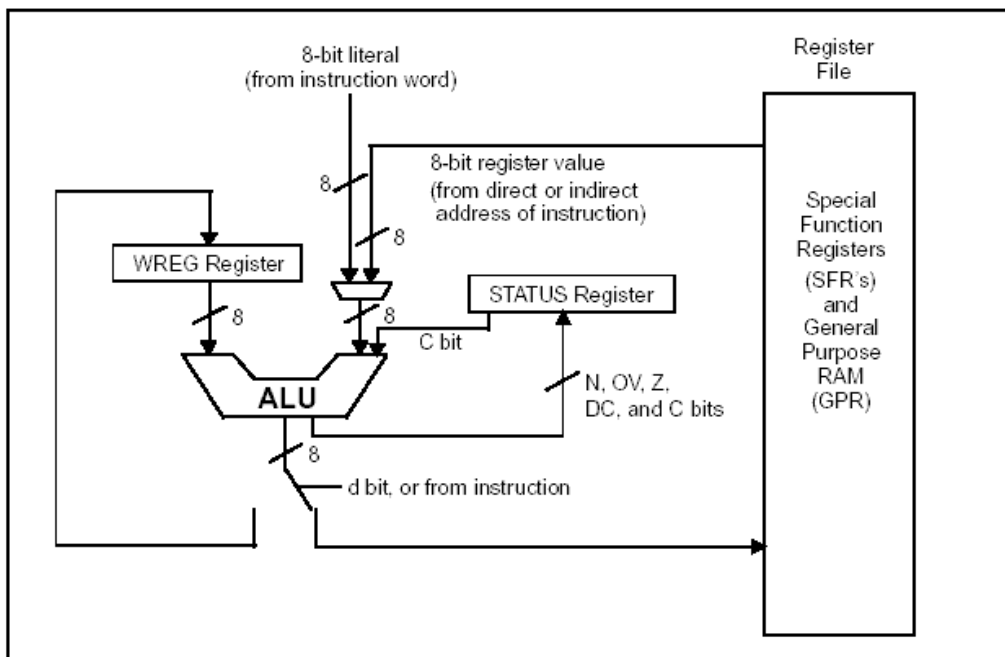
### Charakterystyka urządzeń analogowych:

- 10-bitowy przetwornik analogowo/cyfrowy:

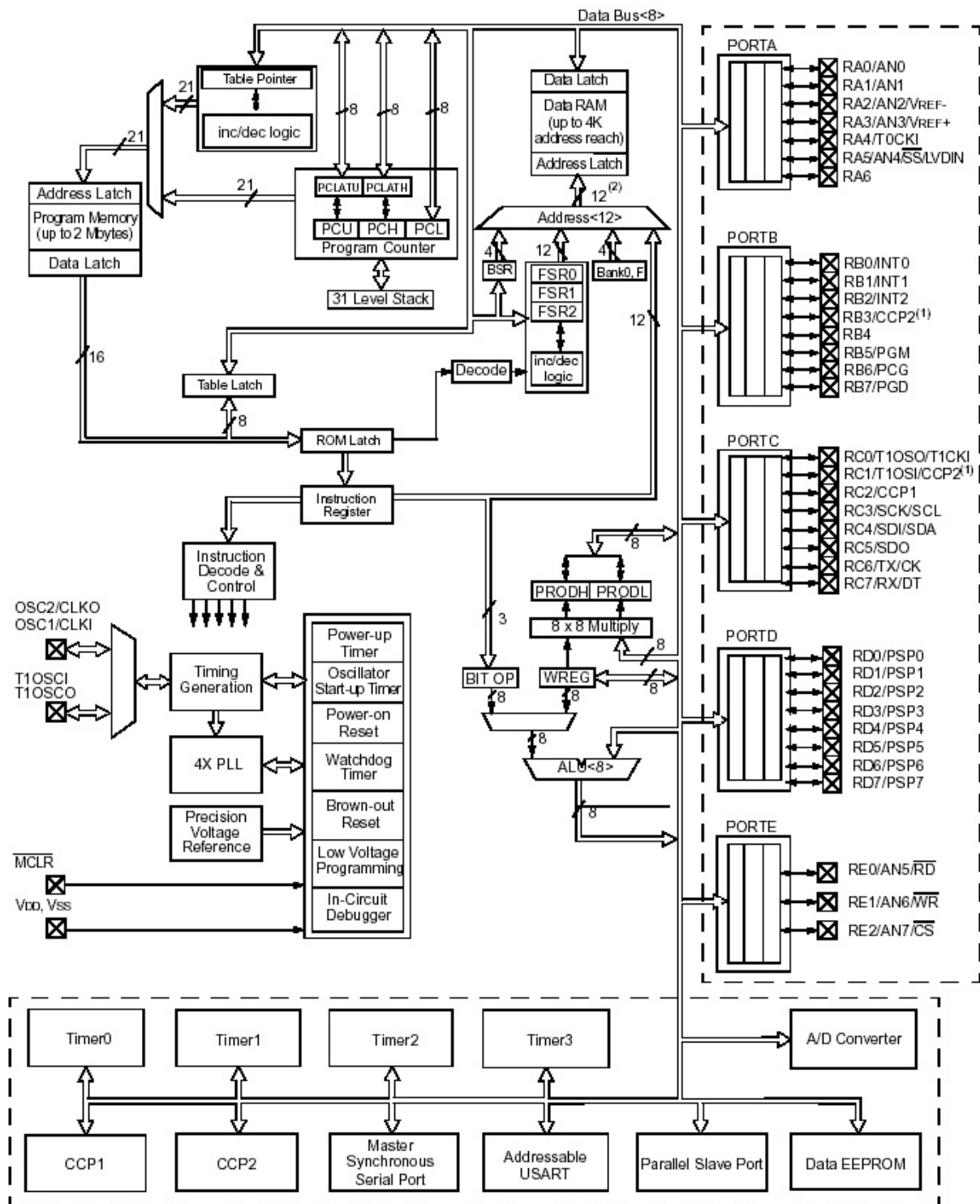
- konwersja AC możliwa w trybie SLEEP,
- liniowość  $\leq 1$  LSb,
- programowana detekcja niskiego napięcia (PLVD),



Rys. 1.1. Rozkład wyprowadzeń mikrokontrolera PIC18F452



Rys. 1.2. Jednostka arytmetyczno – logiczna ALU



Rys. 1.3. Schemat blokowy mikrokontrolera PIC18F452

## Opis wyprowadzeń mikrokontrolera PIC18F452

Nazwa pinu	Numer pinu	Wejście/ Wyjście	Typ bufora	Opis
$\overline{\text{MCLR}}/\text{Vpp}$ $\overline{\text{MCLR}}/\text{Vpp}$	1	we we	ST ST	Wejście zerowania, aktywne poziomem niskim. Wejście napięcia programującego.
NC	-	-	-	Nie podłączone.
OSC1/CLKI  OSC1 CLKI	13	we we	ST CMOS	Wejście oscylatora. Wejście zewnętrznego zegara.
OSC2/CLKO/RA6  OSC2 CLK0 RA6	14	we we we/wy	- - TTL	Wyjście oscylatora. Podłączone do oscylatora lub rezonatora w trybie <i>crystal</i> . W trybie RC, CLK0 jest wyjściem o częstotliwości $\frac{1}{4}$ OSC1 i oznacza czas wykonania instrukcji. Uniwersalny port we/wy.
<i>PORTA jest dwukierunkowym portem we / wy.</i>				
RA0/AN0 RA0 AN0	2	we/wy we	TTL Analog	we/wy cyfrowe. Wejście analogowe 0.
RA1/AN1 RA1 AN1	3	we/wy we	TTL Analog	we/wy cyfrowe. Wejście analogowe 1.
RA2/AN2/ $V_{\text{REF-}}$ RA2 AN2 $V_{\text{REF-}}$	4	we/wy we we	TTL Analog Analog	we/wy cyfrowe. Wejście analogowe 2. Źródło ujemnego napięcia referencyjnego.
RA3/AN3/ $V_{\text{REF+}}$ RA3 AN3 $V_{\text{REF+}}$	5	we/wy we we	TTL Analog Analog	we/wy cyfrowe. Wejście analogowe 3. Źródło dodatniego napięcia referencyjnego.
RA4/T0CKI RA4 T0CKI	6	we/wy we	SD/OD ST	We/wy cyfrowe, wyjście z otwartym drenem Wejście zegarowe modułu Timer0.
$\overline{\text{RA5}}/\text{AN4}/\text{SS}/\text{LVDI}$	7	we/wy	TTL	We/wy cyfrowe. Wejście analogowe 4.

<b>N</b> <b>RA5</b> <b>AN4</b> <b>SS</b> <b>LVDIN</b>		we we we	Analog ST Analog	Wybór układu Slave dla synchronicznego portu szeregowego. Detekcja niskiego napięcia wejściowego.  Patrz pin 14
<b>RA6</b>				
<i>PORTB jest dwukierunkowym portem we/wy, programowo mogą być dołączane wewnętrzne rezystory podciągające (pull-up).</i>				
<b>RB0/INT0</b> <b>RB0</b> <b>INT0</b>	33	we/wy we	TTL ST	We/wy cyfrowe. Przerwanie zewnętrzne 0.
<b>RB1/INT1</b> <b>RB1</b> <b>INT1</b>	34	we/wy we	TTL ST	We/wy cyfrowe. Przerwanie zewnętrzne 1.
<b>RB2/INT2</b> <b>RB2</b> <b>INT2</b>	35	we/wy we	TTL ST	We/wy cyfrowe. Przerwanie zewnętrzne 2.
<b>RB3/CCP2</b> <b>RB3</b> <b>CCP2</b>	36	we/wy we/wy	TTL ST	We/wy cyfrowe. Wejście modułu Capture2, wyjście modułów Compare2i PWM2.
<b>RB4</b>	37	we/wy		We/wy cyfrowe, źródło przerwania.
<b>RB5/PGM</b> <b>RB5</b> <b>PGM</b>	38	we/wy we/wy	TTL ST	We/wy cyfrowe, źródło przerwania. Wejście programowania niskonapięciowego (LVP).
<b>RB6/PGC</b> <b>RB6</b> <b>PGC</b>	39	we/wy we/wy	TTL ST	We/wy cyfrowe, źródło przerwania. Pin Debuggera, wejście sygnału zegarowego dla programowania szeregowego.
<b>RB7/PGD</b> <b>RB7</b> <b>PGD</b>	40	we/wy we/wy	TTL ST	We/wy cyfrowe, źródło przerwania. Pin Debuggera, wejście sygnału danych dla programowania szeregowego.

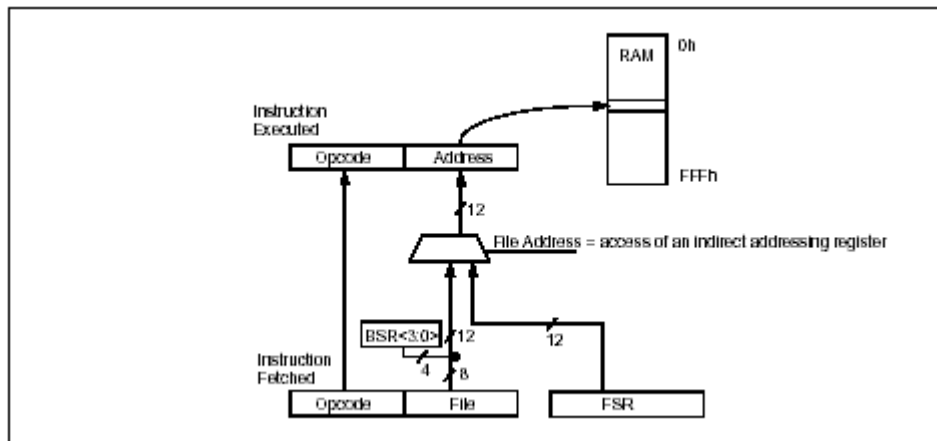
<i>PORTC jest dwukierunkowym portem we/wy.</i>				
<b>RC0/T1OSO/T1CK I RC0 T1OSO T1CKI</b>	15	we/wy wy we	ST - ST	We/wy cyfrowe. Wyjście oscylatora modułu Timer1. Wejście sygnału zegarowego modułów Timer1/Timer2.
<b>RC1/T1OSI/CCP2 RC1 T1OSI CCP2</b>	16	we/wy we we/wy	ST CMOS ST	We/wy cyfrowe. Wejście oscylatora modułu Timer1. Wejście modułu Capture2, wyjścia modułów Capture2 i PWM2.
<b>RC2/CCP1 RC2 CCP1</b>	17	we/wy we/wy	ST ST	We/wy cyfrowe. Wejście modułu Capture1, wyjścia modułów Compare1 i PWM1.
<b>RC3/SCK/SCL RC3 SCK SCL</b>	18	we/wy we/wy we/wy	ST ST ST	We/wy cyfrowe. We/wy zegarowe dla SPI. We/wy zegarowe dla I <sup>2</sup> C
<b>RC4/SDI/SDA RC4 SDI SDA</b>	23	we/wy we we/wy	ST ST ST	We/wy cyfrowe. Wejście danych dla SPI. We/wy danych dla I <sup>2</sup> C.
<b>RC5/SDO RC5 SDO</b>	24	we/wy wy	ST -	We/wy cyfrowe. Wyjście danych dla SPI.
<b>RC6/TX/CK RC6 TX CK</b>	25	we/wy wy we/wy	ST - ST	We/wy cyfrowe. Transmisja asynchroniczna USART. Sygnał zegarowy dla USART
<b>RC7/RX/DT RC7 RX DT</b>	26	we/wy we we/wy	ST ST ST	We/wy cyfrowe. Asynchroniczny odbiór danych USART Synchroniczna transmisja danych USART (patrz na związek sygnałów TX/CK).

<i>PORTD może być dwukierunkowym portem we/wy, lub portem równoległym, dla portu równoległego wszystkie wejścia są typu TTL.</i>				
<b>RD0/PSP0</b>	19	we/wy	ST TTL	We/wy cyfrowe. Dane portu równoległego.
<b>RD1/PSP1</b>	20	we/wy	ST TTL	We/wy cyfrowe. Dane portu równoległego.
<b>RD2/PSP2</b>	21	we/wy	ST TTL	We/wy cyfrowe. Dane portu równoległego.
<b>RD3/PSP3</b>	22	we/wy	ST TTL	We/wy cyfrowe. Dane portu równoległego.
<b>RD4/PSP4</b>	27	we/wy	ST TTL	We/wy cyfrowe. Dane portu równoległego.
<b>RD5/PSP5</b>	28	we/wy	ST TTL	We/wy cyfrowe. Dane portu równoległego.
<b>RD6/PSP6</b>	29	we/wy	ST TTL	We/wy cyfrowe. Dane portu równoległego.
<b>RD7/PSP7</b>	30	we/wy	ST TTL	We/wy cyfrowe. Dane portu równoległego.
<i>PORTE jest dwukierunkowym portem we/wy.</i>				
<b>RE0/RD/AN5</b> RE0 RD AN5	8	we/wy	ST TTL Analog	We/wy cyfrowe. Sterowanie odczytem dla danych portu równoległego. Analogowe wejście 5.
<b>RE1/WR/AN6</b> RE1 WR AN6	9	we/wy	ST TTL Analog	We/wy cyfrowe. Sterowanie odczytem dla danych portu równoległego. Analogowe wejście 6.
<b>RE2/CS/AN7</b> RE2 CS AN7	10	we/wy	ST TTL Analog	We/wy cyfrowe. Sygnał wyboru dla portu równoległego. Analogowe wejście 7.
<b>Vss</b>	12,31	P	-	Masa układów logicznych i linii we/wy
<b>VDD</b>	11,32	P	-	Plus napięcia zasilania układów logicznych i linii we/wy.



## 2 Adresowanie pośrednie, rejestry INDF i FSR

Adresowanie pośrednie jest sposobem adresowania pamięci danych, w którym adres komórki zawierającej daną zmienia się. Rejestr **FSR** (*File Select Register*) wskazuje miejsce, w którym dana ma być zapisana, lub z którego dana ma być odczytana. Mechanizm ten szczególnie jest użyteczny przy tworzeniu tablic danych oraz przy konstruowaniu programowego stosu.



Rys. 2.1. Adresowanie pośrednie

Adresowanie pośrednie wykorzystuje rejestr **INDF**. Nie jest on rejestrem fizycznym. Instrukcje wykonywane na **INDF** dotyczą komórki pamięci wskazywanej przez rejestr **FSR** (*File Select Register*). Przykładowo: wpisana dana do rejestru **INDF** zostanie zapisana w komórce o adresie wskazywanym przez **FSR**, odczytując daną, rejestrem **FSR** wskazujemy jej lokalizację, następnie poprzez rejestr **INDF** odczytujemy jej wartość. Istnieją trzy rejestry pośredniego adresowania.

Aby zapisać 12-bitowy adres każdy z tych rejestrów został podzielony na dwa rejestry 8-bitowe, mianowicie:

- ⇒ rejestr **FSR0** na który składają się rejestry **FSR0H** : **FSR0L**,
- ⇒ rejestr **FSR1** na który składają się rejestry **FSR1H** : **FSR1L**,
- ⇒ rejestr **FSR2** na który składają się rejestry **FSR2H** : **FSR2L**,

Dodatkowo istnieją trzy rejestry **INDF**: **INDF0**, **INDF1**, **INDF2**, które nie są rejestrami fizycznie zaimplementowanymi w strukturę mikrokontrolera.

### Operacje adresowania pośredniego

Gdy dostęp do danej poprzez rejestr **INDF** jest możliwy należy określić w jaki sposób rejestr **FSR** będzie modyfikowany.

Istnieje pięć sposobów zmiany rejestru **FSR**:

- ⇒ po odczytaniu/zapisaniu danej nie zmieniaj rejestru **FSR**,
- ⇒ po odczytaniu/zapisaniu danej zmniejsz wartość **FSR**, **POSTDEC** (*post-decrement*),
- ⇒ po odczytaniu/zapisaniu danej zwiększ wartość **FSR**, **POSTINC** (*post-increment*),

- ⇒ zwiększenie wartości **FSR**, następnie odczytanie/zapisanie danej, **PREINC** (*pre-increment*),
- ⇒ użycie zawartości rejestru **WREG** jako offsetu dla rejestru **FSR**, **PLUSW**,

Wykorzystując tryby *post-decrement* i *post-increment* efekt zmiany rejestru **FSR** nie jest odzwierciedlony w rejestrze **STATUS**. Przykładowo jeżeli rejestr **FSR** osiągnie wartość równą 0, bit **Z** rejestru **STATUS** nie zostanie ustawiony.

### 3 Rejestry związane z funkcjami CPU

#### Rejestr STATUS

Rejestr **STATUS** określa stan jednostki arytmetyczno-logicznej **ALU** (*Arithmetic-Logic Unit*). Bity rejestru **STATUS** są odpowiednio ustawiane, bądź zerowane zależnie od wyniku działania **ALU**.

Użytkownik ma możliwość zmiany wartości poszczególnych bitów. Wykonanie instrukcji **CLRF STATUS** spowoduje wyzerowanie trzech najstarszych bitów i ustawienie bitu **Z**. Rejestr **STATUS** będzie miał postać 000u u1uu (gdzie u - wartości niezmienione). Z tego powodu do zmiany rejestru **STATUS** zaleca się używanie instrukcji **BCF**, **BSF**, **SWAPF**, **MOVFF**, **MOVWF**. Instrukcje te nie zmieniają stanu bitów **Z**, **C**, **DC**, **OV**, oraz **N**.

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
-	-	-	<b>N</b>	<b>OV</b>	<b>Z</b>	<b>DC</b>	<b>C</b>
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bit 7-5     **bity niezaimplementowane**

bit 4       **N: bit negacji**

1 = wynik operacji jest ujemny

0 = wynik operacji jest dodatni

bit 3       **OV: bit przepełnienia**

1 = wystąpiło przepełnienie

0 = nie wystąpiło przepełnienie

bit 2       **Z: bit zera**

1 = wynik operacji arytmetycznej lub logicznej jest zero

0 = wynik operacji arytmetycznej lub logicznej jest różny od zera

bit 1       **DC: bit przeniesienia połówkowego / pożyczki**

dla instrukcji **ADDWF**, **ADDWL**, **SUBLW**, **SUBWF**

1 = wystąpiło przeniesienie z 3 na 4 bit

0 = nie wystąpiło przeniesienie z 3 na 4 bit

bit 0       **C: bit przeniesienia / pożyczki**

dla instrukcji **ADDWF**, **ADDWL**, **SUBLW**, **SUBWF**

1 = wystąpiło przeniesienie z najbardziej znaczącego bitu

0 = nie wystąpiło przeniesienie z najbardziej znaczącego bitu

## Rejestr RCON (*Reset Control*)

Rejestr **RCON** zawiera informacje o źródle powstania RESETU. Rejestr jest zapisywalny i odczytywalny.

R/W-0	U-0	U-0	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	-	-	~RI	~TO	~PD	~POR	~BOR
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **IPEN: priorytet przerwania**  
 1 = obsługa priorytetów przerwania (*low/high*)  
 0 = brak obsługi priorytetów przerwania (wszystkie przerwania traktowane są jako *high*)
- bit 6-5    **bity niezaimplementowane**
- bit 4      **~RI: flaga od wystąpienia instrukcji RESET**  
 1 = nie została wykonana instrukcja RESET  
 0 = została wykonana instrukcja RESET  
*(flaga ta musi być ustawiona programowo po wystąpieniu resetu Brown-out)*
- bit 3      **~TO: flaga resetu od licznika Watchdog**  
 1 = po włączeniu zasilania, po wykonaniu instrukcji CLRWDT, SLEEP  
 0 = reset od licznika Watchdog
- bit 2      **~PD: flaga zaniku napięcia zasilania**  
 1 = po zaniku napięcia zasilania, lub po wykonaniu instrukcji CLRWDT  
 0 = po wykonaniu instrukcji SLEEP
- bit 1      **~POR: reset po włączeniu napięcia zasilania**  
 1 = nie wystąpił reset po włączeniu zasilania  
 0 = wystąpił reset po włączeniu zasilania  
*(musi być ustawiony programowo po wystąpieniu ~POR)*
- bit 0      **~BOR: (Brown-out Reset) reset zaniku napięcia zasilania poniżej wartości progowej**  
 1 = nie wystąpił reset Brown-out  
 0 = wystąpił reset Brown-out  
*(musi być ustawiony programowo po wystąpieniu ~BOR)*

Legenda: R – bit odczytywalny,      W – bit zapisywalny,      U – bit niezaimplementowany, 0,1 – stan po włączeniu zasilania (x- stan nieokreślony),
--

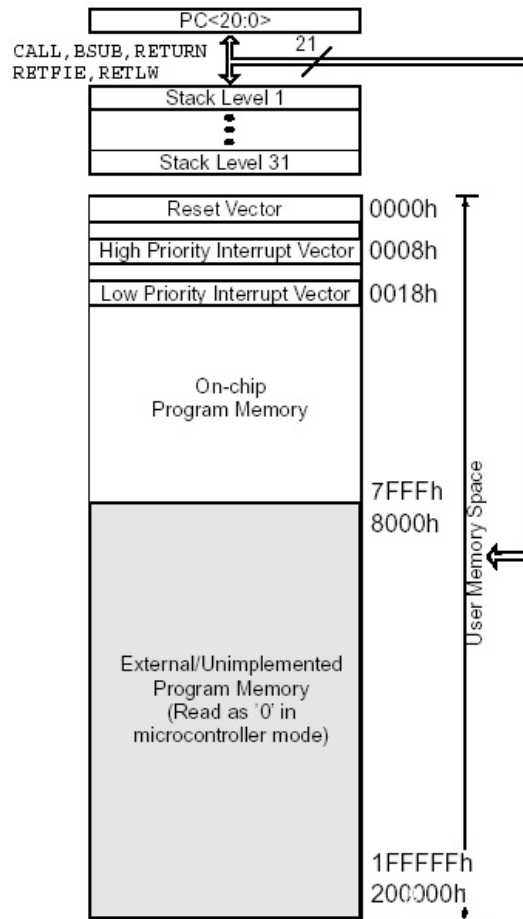
## 4 Organizacja pamięci

Pamięć mikroprocesora można podzielić na dwie odrębne części: pamięć programu oraz pamięć danych. W pamięci danych mogą być przechowywane dane pomiarowe, czy przejściowe wyniki obliczeń. Mikroprocesor może odczytywać oraz zapisywać dane. Niestety po zaniku napięcia zasilania dane te ulegają zanikowi. Wśród pamięci danych wyróżnić można również pamięć, w której raz zapisana informacja nie ulega utracie nawet po zaniku napięcia zasilania. W pamięciach tych EEPROM (*Electrically EPROM*) kasowanie zawartości odbywa się w sposób elektryczny. Służy do przechowywania ważnych danych takich jak np. nastawy urządzeń kalibracyjnych. Pamięć programu służy do przechowywania instrukcji wykonywanego programu.

### Pamięć programu

Pamięć programu jest zaimplementowana jako jeden niezależny blok, gdzie wektor **RESET** znajduje się pod adresem 0000h, natomiast wektory przerwań zostały umieszczone pod dwoma adresami 0008h i 0018h, którym odpowiadają wysoki i niski priorytet obsługi.

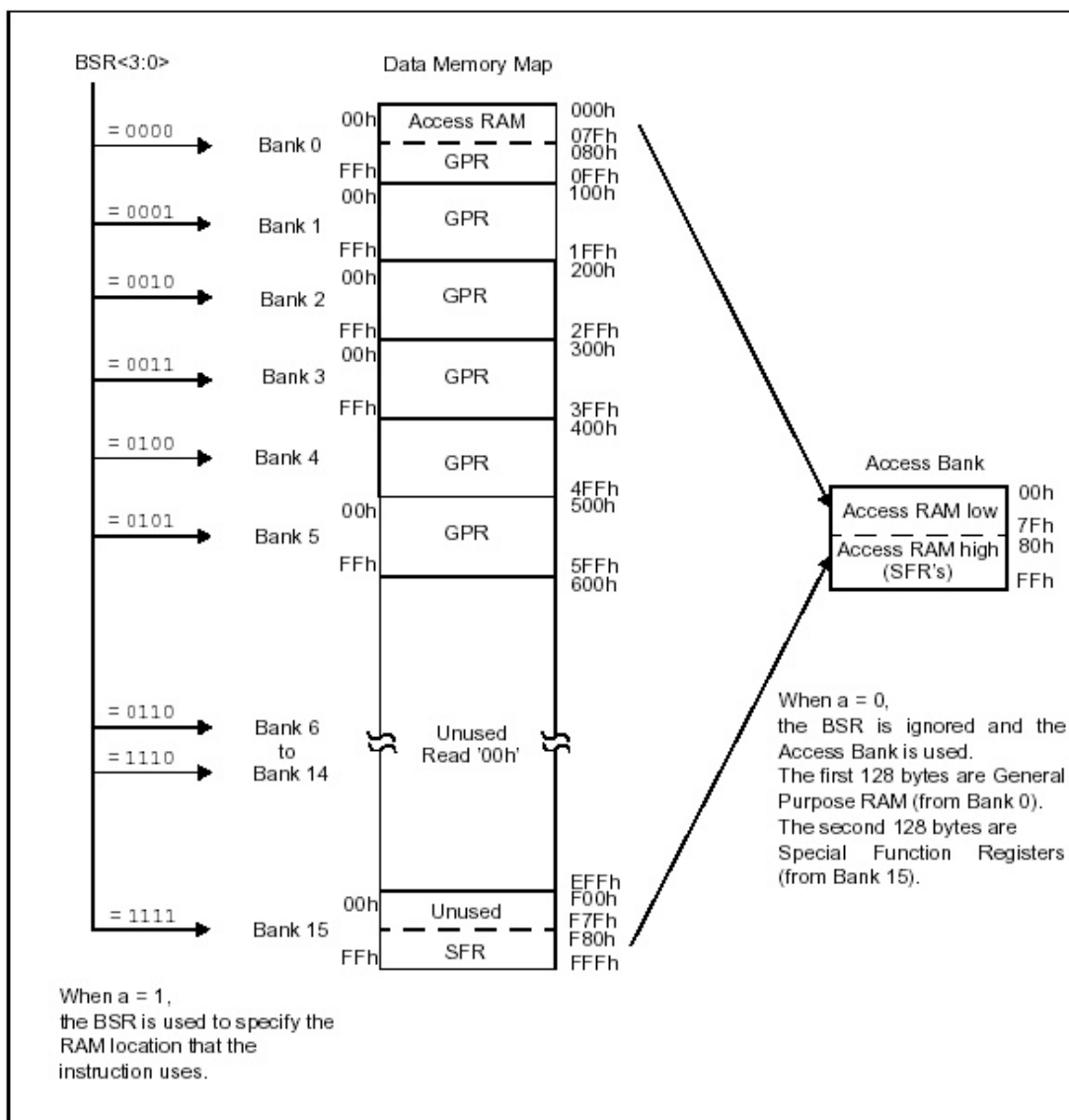
Instrukcje CALL i GOTO mogą adresować cały obszar mapy pamięci, natomiast BRA, i RCALL mają ograniczony zasięg (+1023 do -1024 komórek pamięci programu).



D3.1. Mapa pamięci programu i stosu

## Pamięć danych

Pamięć danych podzielona jest na 16 banków, z których każdy zawiera 256 bajtów. Wyboru banku dokonuje się poprzez rejestr **BSR** (*Bank Select Register*). W rejestrze tym cztery młodsze bity określają numer banku, pozostałe bity nie zostały zaimplementowane. W pamięci danych znajdują się rejestry funkcji specjalnych **SFR** (*Special Function Registers*) używane do kontroli stanu mikrokontrolera i urządzeń peryferyjnych, rozpoczynają się od ostatniej pozycji Banku 15 (0xFFF), oraz rejestry ogólnego przeznaczenia **GPR** (*General Purpose Registers*) stosowane do przechowywania danych zależnie od potrzeb użytkownika, rozpoczynają się od pierwszej pozycji Banku 0.



### D3.2. Mapa pamięci danych

Organizacji pamięci danych towarzyszy mechanizm banku dostępu (*Access Bank*). Składa się na niego starsza część 128 bajtów Banku 15, która stanowi zbiór rejestrów funkcji specjalnych **SFR**, oraz młodsza część 128 bajtów Banku 0, będąca rejestrami ogólnego przeznaczenia **GPR**. Taki obszar pamięci danych stosowany jest przy definiowaniu lokalnych zmiennych różnych podprogramów, szybkim zapisywaniu/przełączaniu kontekstu zmiennych, definiowaniu wspólnych zmiennych oraz szybkim przełączaniu rejestrów funkcji specjalnych **SFR**.

Aby móc korzystać z właściwości banku dostępu w kodzie instrukcji asemblera należy odpowiednio ustawić bit 'a' będącym jednym z argumentów instrukcji. Przykładowo instrukcja `ADDWF f, d, a` oznacza dodanie zawartości rejestru **W** do zawartości rejestru **f**. Bit 'd' określa, gdzie wynik operacji dodawania zostanie umieszczony, czy w rejestrze **f** (dla 'd'=1), czy też w rejestrze **W** (dla 'd'=0). Bit 'a', o którym była wcześniej mowa określa, który bank jest wykorzystywany do przeprowadzenia dodawania. Jeżeli 'a'=1, wykorzystywany jest bank wskazywany przez rejestr **BSR**, jeżeli 'a'=0, stosowany jest bank dostępu.

Zbiór rejestrów funkcji specjalnych **SFR**, stanowiących integralną część pamięci danych oraz banku dostępu przedstawiony jest na Rys.2.6.

Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDfH	INDF2 <sup>(2)</sup>	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2 <sup>(3)</sup>	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2 <sup>(3)</sup>	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2 <sup>(3)</sup>	FBCh	CCPR2H	F9Ch	—
FFBh	PCLATU	FDBh	PLUSW2 <sup>(3)</sup>	FBBh	CCPR2L	F9Bh	—
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	—
FF9h	PCL	FD9h	FSR2L	FB9h	—	F99h	—
FF8h	TBLPTRU	FD8h	STATUS	FB8h	—	F98h	—
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	—	F97h	—
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	—	F96h	TRISE <sup>(2)</sup>
FF5h	TABLAT	FD5h	T0CON	FB5h	—	F95h	TRISD <sup>(2)</sup>
FF4h	PRODH	FD4h	—	FB4h	—	F94h	TRISC
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	LVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	—
FF0h	INTCON3	FD0h	RCON	FB0h	—	F90h	—
FEFh	INDF0 <sup>(3)</sup>	FCFh	TMR1H	FAFh	SPBRG	F8Fh	—
FEEh	POSTINC0 <sup>(3)</sup>	FCEh	TMR1L	FAEh	RCREG	F8Eh	—
FEDh	POSTDEC0 <sup>(3)</sup>	FCDh	T1CON	FADh	TXREG	F8Dh	LATE <sup>(2)</sup>
FECh	PREINC0 <sup>(3)</sup>	FCCh	TMR2	FACH	TXSTA	F8Ch	LATD <sup>(2)</sup>
FEBh	PLUSW0 <sup>(3)</sup>	FC8h	PR2	FABh	RCSTA	F8Bh	LATC
FEAh	FSR0H	FCAh	T2CON	FAAh	—	F8Ah	LATB
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	—
FE7h	INDF1 <sup>(3)</sup>	FC7h	SSPSTAT	FA7h	EECON2	F87h	—
FE6h	POSTINC1 <sup>(3)</sup>	FC6h	SSPCON1	FA6h	EECON1	F86h	—
FE5h	POSTDEC1 <sup>(3)</sup>	FC5h	SSPCON2	FA5h	—	F85h	—
FE4h	PREINC1 <sup>(3)</sup>	FC4h	ADRESH	FA4h	—	F84h	PORTE <sup>(2)</sup>
FE3h	PLUSW1 <sup>(3)</sup>	FC3h	ADRESL	FA3h	—	F83h	PORTD <sup>(2)</sup>
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	—	FA0h	PIE2	F80h	PORTA

- Note 1: Unimplemented registers are read as '0'.  
 2: This register is not available on PIC18F2X2 devices.  
 3: This is not a physical register.

### D3.3. Rejestry funkcji specjalnych SFR

## Pamięć danych EEPROM

Przy obsłudze bloku pamięci EEPROM, rejestr **EEDATA** przechowuje 8-bitową daną przeznaczoną do zapisu/odczytu, zaś w rejestrze **EEADR** znajduje się adres dostępu do komórki pamięci. Użytkownik dysponuje 256 bajtami pamięci EEPROM o adresach od 00h-FFh.

Zapis do komórki wypełnionej spowoduje skasowanie jej zawartości i wpisanie nowej danej (*erase-before-write*). Czas wpisu zależy od napięcia zasilania zmieniającego się od  $V_{\text{MIN}} = 4.2\text{V}$  do  $V_{\text{DD}} = 5.5\text{V}$  oraz temperatury, jest kontrolowany przez wewnętrzny zegar i typowo wynosi  $T_{\text{DEW}} = 4\text{ms}$ .

Bit RD i WR rejestru **EECON1** rozpoczynają odpowiednio operację odczytu jak i zapisu. Mogą one być ustawiane programowo, zerowanie odbywa się sprzętowo stosownie do zakończenia odpowiedniej operacji. Brak możliwości programowego zerowania bitu WR zapobiega przypadkowemu lub przedwczesnemu zakończeniu zapisu.

Ustawienie bitu WREN jest jednoznaczne z umożliwieniem dokonania operacji zapisu. Bit ten po włączeniu zasilania jest ustawiony na '0'. Kiedy proces zapisu zostanie przerwany przez reset  $\sim\text{MCLR}$ , lub przez przepelnienie licznika Watch-dog (*WDT-reset*), bit WRERR jest ustawiony na '1'. W takim przypadku użytkownik odczytując stan bitu WRERR może wznowić proces zapisu, należy wówczas ponownie „załadować” daną i adres do rejestrów EEDATA i EEADR.

### UWAGA!

Kiedy zakończy się proces zapisu, ustawiana jest flaga EEIF w rejestrze **PIR2**.  
Flaga ta musi być programowo zerowana.



## Rejestr EECON1 (adres FA6h)

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	-	FREE	WRERR	WREN	WR	RD
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **EEPGD: wybór rodzaju pamięci FLASH/EEPROM**  
 1 = pamięć programu FLASH  
 0 = pamięć danych EEPROM
- bit 6      **CFGS: pamięć FLASH/pamięć EEPROM lub bity konfiguracyjne**  
 1 = dostęp do bitów konfiguracyjnych  
 0 = dostęp do pamięci FLASH lub pamięci EEPROM
- bit 5      **bit niezaimplementowany**
- bit 4      **FREE: bit zezwolenia skasowania komórki pamięci FLASH**  
 1 = skasowanie komórki pamięci wskazywanej przez rejestr TBLPTR przy następnej operacji zapisu WR  
*(bit kasowany po ukończeniu operacji zapisu)*  
 0 = wykonanie tylko zapisu
- bit 3      **WRERR: bit błędu zapisu do pamięci FLASH/EEPROM**  
 1 = operacja zapisu została przedwcześnie zakończona  
 0 = operacja zapisu zakończona powodzeniem
- bit 2      **WREN: bit zezwolenia zapisu do FLASH/EEPROM**  
 1 = zezwolenie zapisu  
 0 = zakaz zapisu do EEPROM
- bit 1      **WR: bit kontroli zapisu**  
 1 = rozpoczyna operację kasowania/zapisu do EEPROM lub do FLASH  
*(operacja jest kontrolowana przez wewnętrzny zegar, bit jest zerowany sprzętowo po zakończeniu zapisu, bit WR może być tylko ustawiany programowo)*  
 0 = zapis do EEPROM ukończony
- bit 0      **RD: bit kontroli odczytu**  
 1 = rozpoczyna operację odczytu EEPROM  
*(bit RD zerowany jest sprzętowo, bit RD może być tylko ustawiany programowo)*  
 0 = odczyt nie został rozpoczęty

Legenda:

R – bit odczytywalny,      W – bit zapisywalny,      U – bit niezaimplementowany,  
 0,1 – stan po włączeniu zasilania (x- stan nieokreślony),

## Odczyt pamięci danych EEPROM

Chcąc odczytać daną z pamięci EEPROM należy:

- ⇒ wpisać do rejestru **EEADR** adres spod którego dana ma być odczytana,
- ⇒ wybrać rodzaj pamięci przez wyzerowanie bitu **EEPGD** w rejestrze **EECON1**,
- ⇒ umożliwić dostęp do pamięci EEPROM przez wyzerowanie bitu **CFGS** w rejestrze **EECON1**,
- ⇒ rozpocząć operację odczytu ustawieniem bitu **RD** rejestru **EECON1**,
- ⇒ dana dostępna w rejestrze **EEDATA**,

```
MOVLW  ADRES_DANEJ
MOVWF  EEADR           ; adres danej
BCF    EECON1, EEPGD  ; wybór EEPROM
BCF    EECON1, CFGS   ; dostęp do EEPROM
BCF    EECON1, RD     ; odczyt EEPROM
MOVF   EEDATA, W      ; dana w rejestrze W
```

*Listing D1.1 Odczyt pamięci danych EEPROM*

## Zapis do pamięci danych EEPROM

Chcąc zapisać daną do pamięci EEPROM należy:

- ⇒ do rejestru **EEADR** wpisać adres wskazujący miejsce zapisu danej,
- ⇒ do rejestru **EEDATA** wpisać daną do zapisu w EEPROM-ie,
- ⇒ wybrać rodzaj pamięci przez wyzerowanie bitu **EEPGD** w rejestrze **EECON1**,
- ⇒ umożliwić dostęp do pamięci EEPROM przez wyzerowanie bitu **CFGS** w rejestrze **EECON1**,
- ⇒ zezwolić na operację zapisu ustawieniem bitu **WREN** rejestru **EECON1**,
- ⇒ zablokować przerwania zerując bit **GIE**(*Global Interrupt Enable*) rejestru **INTCON** (o ile przerwania są używane),
- ⇒ wykonać sekwencję instrukcji:
  - zapisać wartość 55h do rejestru **EECON2** w dwóch krokach (najpierw do rejestru **W**, później do rejestru **EECON2**),
  - zapisać wartość AAh do rejestru **EECON2** w dwóch krokach (najpierw do rejestru **W**, później do rejestru **EECON2**),
- ⇒ rozpocząć operację zapisu ustawieniem bitu **WR** rejestru **EECON1**,
- ⇒ odblokować przerwania ustawiając bit **GIE** (*Global Interrupt Enable*) rejestru **INTCON** (o ile przerwania są używane),

	MOVLW	ADRES_DANEJ	
	MOVWF	EEADR	; adres danej
	MOVLW	DANA_DO_ZAPISANIA	;dana do zapisania
	MOVWF	EEDATA	
	BCF	EECON1, EEPGD	; wybór EEPROM
	BCF	EECON1, CFGS	; dostęp do EEPROM
	BSF	EECON1, WREN	; zezwolenie na zapis
	BCF	INTCON, GIE	; blokada przerw
<b>wymagana</b>	MOVLW	55h	
<b>sekwencja</b>	MOVWF	EECON2	
<b>instrukcji</b>	MOVLW	AAh	
	MOVWF	EECON2	
	BSF	EECON1, WR	; rozpoczęcie zapisu
	BSF	INTCON, GIE	; odblokowanie ;przerw
	.		
	.		; kod użytkownika
	.		
	BCF	EECON1, WREN	; zakończenie zapisu, ; flaga przerwania ; EEIF ; rejestru <b>INTCON</b> ; ustawiona na 1

*Listing D3.1. Zapis do pamięci danych EEPROM*

Wspomniana wyżej wymagana sekwencja instrukcji musi być wykonywana dla każdego zapisywanego bajtu. Brak tych instrukcji uniemożliwi zapis do EEPROM. Zaleca się, aby przed wykonaniem instrukcji zapisu do rejestru **EECON2** zablokować przerwania. Bit WREN powinien być wyzerowany przez czas normalnej pracy programu, ustawiany jedynie przy zapisie do EEPROM. Zapobiega się w ten sposób przypadkowemu zapisowi do EEPROM np. nieskończona pętla. Rozpoczęcie zapisu będzie możliwe, jeżeli w poprzedniej instrukcji bit WREN został ustawiony na '1'. Po zakończeniu zapisu bit WR jest zerowany sprzętowo, flaga zakończenia zapisu EEIF rejestru **INTCON** zostanie ustawiona.

## 5 Mnożenie wspomagane sprzętowo

Jednostka arytmetyczno logiczna w swojej strukturze zawiera sprzętowy mechanizm wspomagający mnożenie. Umożliwia to wykonywanie mnożenia w jednym cyklu rozkazowym. Mnożąc dwie liczby 8-bitowe otrzymany wynik umieszczony jest w parze rejestrów **PRODH:PRODL**.

Zalety mnożenia sprzętowego:

- ⇒ zmniejszenie zajętości mikroprocesora (większa przepustowość),
- ⇒ zredukowanie wielkości kodu przy algorytmach mnożących,

Czas wykonywanych operacji pozwala na wykorzystanie urządzenia w aplikacjach wcześniej zarezerwowanych tylko dla Procesorów Sygnałowych.

Rodzaj mnożenia	Metoda mnożenia	Pamięć programu (słowa)	Cykle (max)	Czas		
				40MHz	10MHz	4MHz
8 x 8 unsigned	Mnożenie bez wspomagania sprzętowego	13	69	6.9µs	27.6µs	69µs
	Mnożenie sprzętowe	1	1	100ns	400ns	1µs
8 x 8 signed	Mnożenie bez wspomagania sprzętowego	33	91	9.1µs	36.4µs	91µs
	Mnożenie sprzętowe	6	6	600ns	2.4µs	6µs
16 x 16 unsigned	Mnożenie bez wspomagania sprzętowego	21	242	24.2µs	96.8µs	242µs
	Mnożenie sprzętowe	24	24	2.4µs	9.6µs	24µs
16 x 16 signed	Mnożenie bez wspomagania sprzętowego	52	254	25.4µs	102.6µs	254µs
	Mnożenie sprzętowe	36	36	3.6µs	14.4µs	36µs

*Tabela D4.1. Porównanie metod mnożenia*

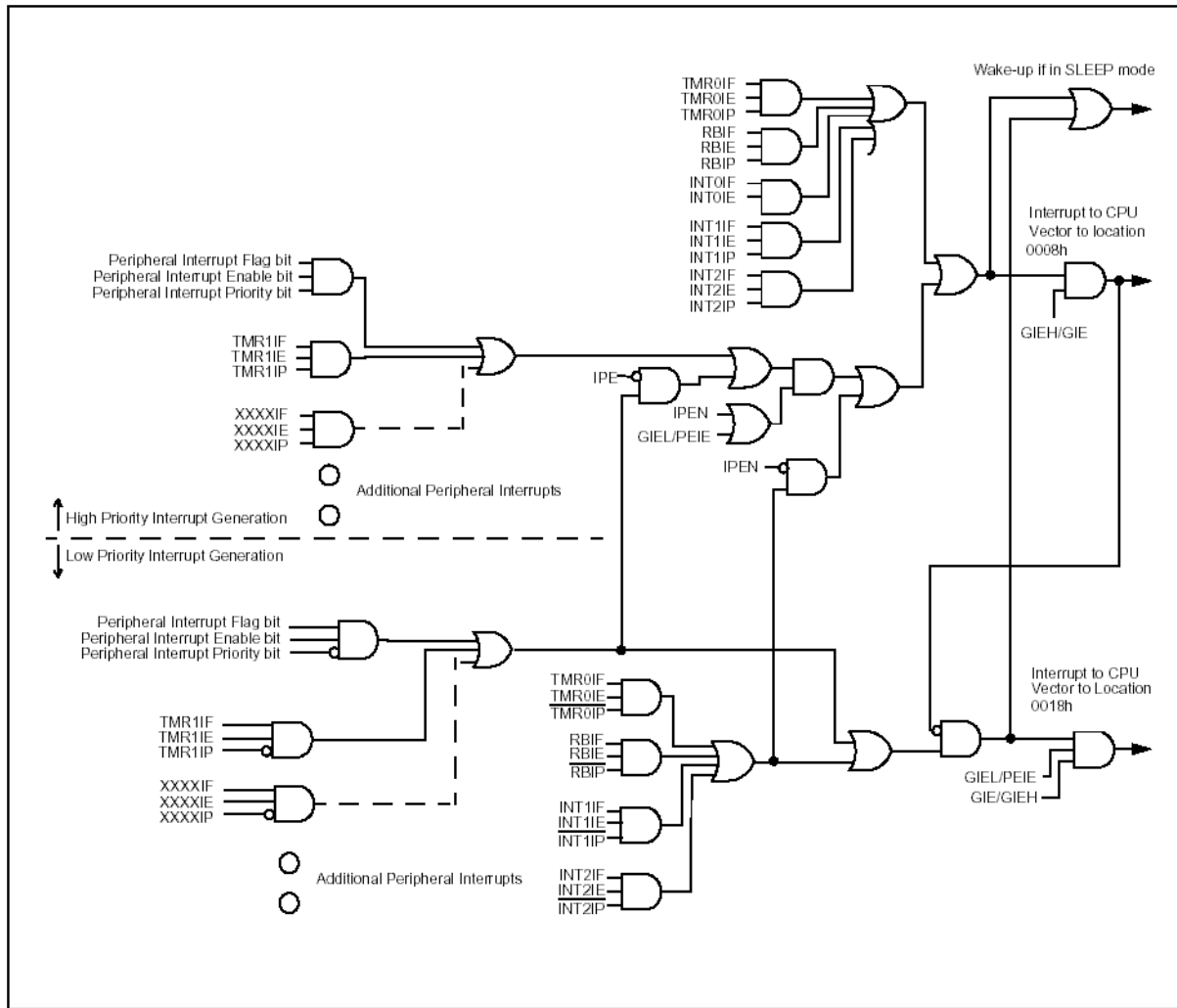
## 6 Przerwania

Zmieniając wartość bitu IPEN w rejestrze **RCON**, użytkownik może zdecydować, czy używać priorytetów przerwania, czy też wszystkie przerwania traktować jednakowo – jak przerwania o wysokim priorytecie. Chcąc wykorzystywać przerwania należy sterować bitami odpowiedzialnymi za globalne blokowanie/odblokowywanie przerwania. Bit GIEH rejestru **INTCON** odblokowuje przerwania wysokiego priorytetu, bit GIEL rejestru **INTCON** odblokowuje przerwania priorytetu niskiego.

Wykorzystując kod obsługi przerwania mikrokontrolerów rodziny PIC18Cxxx w mikrokontrolerach rodziny PIC<sup>®</sup> mid-range, ustawieniem bitu IPEN=0 rejestru **RCON** zapewniona jest zgodność obsługi przerwania. Wszystkie przerwania przekazywane są pod adres 000008h.

Pojawienie się przerwania powoduje ustawienie bitu GIE=0 uniemożliwiając tym samym przyjęcie następnego przerwania. Adres powrotu z obsługi przerwania jest umieszczany na stosie, do licznika rozkazów PC (*Program Counter*) umieszczany jest adres wektora przerwania (000008h lub 000018h). Źródło przerwania może być określone poprzez przeglądanie bitów flag przerwania. Bit flagi musi być zerowany programowo przed ponownym odblokowaniem przerwania. Zapobiega to powtarzaniu się przerwania.

Dla przerwania zewnętrznych (portu B), występuje opóźnienie przyjęcia przerwania równe trzem lub czterem cyklom rozkazowym.



D5.1. Układ przerwań

Rysunek D5.1 przedstawia układ przerwań z wszystkimi dostępnymi bitami stosowanymi przy obsłudze przerwań, przedstawia również podział urządzeń posiadających wysoki, oraz niski priorytet obsługi.

## Rejestr INTCON

( Rejestry **INTCON** - **INTCON3** są rejestrami odczytywalnymi/zapisywalnymi, zawierają bity zezwalania na wystąpienie przerwania, ustalania priorytetów oraz bity flag ).

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/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7	Bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **GIE/GIEH: bit globalnego zezwolenia na przerwanie:**  
gdy IPEN=0  
1 = odblokowanie wszystkich nie zamaskowanych przerw,   
0 = wszystkie przerwy zablokowane,  
gdy IPEN=1  
1 = odblokowanie przerw wysokiego priorytetu,   
0 = wszystkie przerwy zablokowane,
- bit 6      **PEIE/GIEL: zezwolenie na przerwanie od układów peryferyjnych:**  
gdy IPEN=0  
1 = odblokowanie wszystkich nie zamaskowanych przerw od układów peryferyjnych,   
0 = wszystkie przerwy od urządzeń peryferyjnych zablokowane,  
gdy IPEN=1  
1 = odblokowanie przerw niskiego priorytetu od układów peryferyjnych,   
0 = zablokowanie wszystkich przerw niskiego priorytetu od układów peryferyjnych,
- bit 5      **TMR0IE: zezwolenie na przerwanie od przepełnienia licznika TMR0:**  
1 = odblokowanie przerwania od przepełnienia licznika TMR0,   
0 = zablokowanie przerwania od przepełnienia licznika TMR0,
- bit 4      **INT0IE: zezwolenie na przerwanie zewnętrzne z linii RB0/INT0:**  
1 = odblokowanie przerwania zewnętrznego na linii RB0/INT0,   
0 = zablokowanie przerwania zewnętrznego na linii RB0/INT0,
- bit 3      **RBIE: zezwolenie na przerwanie pochodzące od zmiany stanu PORTuB:**  
1 = odblokowanie przerwania od zmiany stanu na liniach PORTuB,   
0 = zablokowanie przerwania od zmiany stanu na liniach PORTuB,
- bit 2      **TMR0IF: flaga przerwania od przepełnienia licznika 0 (TMR0):**  
1 = nastąpiło przepełnienie licznika TMR0 (*musi być zerowana programowo*),   
0 = nie nastąpiło przepełnienie licznika TMR0,
- bit 1      **INT0IF: flaga zewnętrznego przerwania z linii RB0/INT0:**  
1 = wystąpiło zewnętrzne przerwanie na linii RB0/INT0 (*musi być zerowana programowo*),   
0 = nie wystąpiło zewnętrzne przerwanie na linii RB0/INT0,
- bit 0      **RBIF: flaga przerwania od zmiany stanu PORTuB:**  
1 = pojawiła się przynajmniej jedna zmiana stanu na liniach RB4:RB7 (*musi być zerowana programowo*),   
0 = nie pojawiła się zmiana stanu na liniach RB4:RB7,

## Rejestr INTCON2

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
RBPU	INTEDG0	INTEDG1	INTEDG2	-	TMR0IP	-	RBIP
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **~RBPU: zezwolenie na podłączenie rezystorów pull-up do PORTuB:**  
 1 = rezystory pull-up odłączone,  
 0 = rezystory pull-up dołączone,
- bit 6      **INTEDG0: wybór zbocza przerwania zewnętrznego INTE0:**  
 1 = przerwanie na narastające zbocze,  
 0 = przerwanie na opadające zbocze,
- bit 5      **INTEDG1: wybór zbocza przerwania zewnętrznego INTE1:**  
 1 = przerwanie na narastające zbocze,  
 0 = przerwanie na opadające zbocze,
- bit 4      **INTEDG2: wybór zbocza przerwania zewnętrznego INTE2:**  
 1 = przerwanie na narastające zbocze ,  
 0 = przerwania na opadające zbocze,
- bit 3      **bit niezaimplementowany,**
- bit 2      **TMR0IP: wybór priorytetu przepelnienia licznika TMR0:**  
 1 = wysoki priorytet,  
 0 = niski priorytet,
- bit 1      **bit niezaimplementowany,**
- bit 0      **RBIP: wybór priorytetu zmiany stanu na liniach RB4:RB7:**  
 1 = wysoki priorytet,  
 0 = niski priorytet,

Legenda: R – bit odczytywalny,      W – bit zapisywalny,      U – bit niezaimplementowany, 0,1 – stan po włączeniu zasilania (x- stan nieokreślony),
--

## Rejestr INTCON3

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	-	INT2IE	INT1IE	-	INT2IF	INT1IF
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **INT2IP: wybór priorytetu przerwania zewnętrznego od INT2:**  
 1 = wysoki priorytet,  
 0 = niski priorytet,
- bit 6      **INT1IP: wybór priorytetu przerwania zewnętrznego od INT1:**  
 1 = wysoki priorytet,  
 0 = niski priorytet,
- bit 5      **bit niezaimplementowany,**
- bit 4      **INT2IE: zezwolenie na przerwanie zewnętrzne na INT2:**  
 1 = odblokowanie przerwania zewnętrznego na INT2,  
 0 = zablokowanie przerwania zewnętrznego na INT2,
- bit 3      **INT1IE: zezwolenie na przerwanie zewnętrzne na INT1:**  
 1 = odblokowanie przerwania zewnętrznego na INT1,  
 0 = zablokowanie przerwania zewnętrznego na INT1,
- bit 2      **bit niezaimplementowany,**
- bit 1      **INT2IF: flaga przerwania zewnętrznego od INT2:**  
 1 = wystąpiło zewnętrzne przerwanie na INT2  
 (*musi być zerowana programowo*),  
 0 = nie wystąpiło zewnętrzne przerwanie na INT2,
- bit 0      **INT1IF: flaga przerwania zewnętrznego od INT1:**  
 1 = wystąpiło zewnętrzne przerwanie na INT1  
 (*musi być zerowana programowo*),  
 0 = nie wystąpiło zewnętrzne przerwanie na INT1,

Legenda: R – bit odczytywalny,      W – bit zapisywalny,      U – bit niezaimplementowany, 0,1 – stan po włączeniu zasilania (x- stan nieokreślony),
--



## Rejestr PIR1

### PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

( Rejestry **PIR1**, **PIR2** zawierają flagi przerwania od urządzeń peryferyjnych. Poszczególne bity flag przerwania ustawiane są w momencie wystąpienia przerwania, niezależnie od stanu bitu **GIE** rejestru **INTCON**. Po obsłudze przerwania należy zerować flagę przerwania programowo ).

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>PSPIF</b>	<b>ADIF</b>	<b>RCIF</b>	<b>TXIF</b>	<b>SSPIF</b>	<b>CCP1IF</b>	<b>TMR2IF</b>	<b>TMR1IF</b>
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7     **PSPIF: flaga przerwania zapisu/odczytu portu równoległego (zarezerwowany dla układów rodziny PIC18F2x2):**  
1 = wystąpiła operacja zapisu/odczytu (*musi być zerowana programowo*),  
0 = nie wystąpiła operacja zapisu/odczytu,
- bit 6     **ADIF: flaga przerwania od przetwornika A/C:**  
1 = konwersja analogowo-cyfrowa zakończyła się (*musi być zerowana programowo*),  
0 = konwersja analogowo-cyfrowa nie zakończyła się,
- bit 5     **RCIF: flaga przerwania odbiornika USART:**  
1 = bufor odbiornika USART, rejestr **RCREG** jest pełny (*flaga jest zerowana przy odczycie rejestru RCREG*),  
0 = bufor odbiornika USART jest pusty,
- bit 4     **TXIF: flaga przerwania nadajnika USART:**  
1 = bufor nadajnika USART, rejestr **TXREG** jest pusty (*flaga jest zerowana przy zapisie do rejestru TXREG*),  
0 = bufor nadajnika USART jest pełny,
- bit 3     **SSPIF: flaga przerwania transmisji synchronicznej:**  
1 = transmisja/odbiór zakończony,  
0 = oczekiwanie na transmisję/odbiór,
- bit 2     **CCP1IF: flaga przerwania od układu CCP (Capture, Compare, PWM):**  
tryb Capture:  
1 = nastąpiło zatrzaśnięcie rejestru TMR1 (*flaga musi być zerowana programowo*),  
0 = nie wystąpiło zatrzaśnięcie rejestru TMR1,  
tryb Compare:  
1 = nastąpiło zrównanie rejestru TMR1 (*flaga musi być zerowana programowo*),  
0 = nie nastąpiło zrównanie rejestru TRM1,  
tryb PWM:  
nieużywany,
- bit 1     **TMR2IF: flaga przerwania równości rejestrów TMR2 i PR2:**  
1 = wystąpiło zrównanie wartości rejestrów **TMR2** i **PR2** (*flaga musi być zerowana programowo*),  
0 = nie wystąpiło zrównanie wartości rejestrów **TMR2** i **PR2**,
- bit 0     **TMR1IF: flaga przerwania przepełnienia rejestru TMR1:**  
1 = wystąpiło przepełnienie rejestru TMR1,  
0 = nie wystąpiło przepełnienie rejestru TMR1,

## Rejestr PIR2

### PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7-5      **bity niezaimplementowane,**
- bit 4      **EEIF: flaga przerwania zakończenia zapisu do pamięci EEPROM/FLASH:**  
1 = operacja zapisu zakończyła się (*flaga musi być zerowana programowo*),  
0 = operacja zapisu nie zakończyła się, lub nie została rozpoczęta,
- bit 3      **BCLIF: flaga przerwania kolizji magistrali:**  
1 = wystąpiła kolizja magistrali (*flaga musi być zerowana programowo*),  
0 = nie wystąpiła kolizja magistrali,
- bit 2      **LVDIF: flaga przerwania wykrycia spadku napięcia zasilania:**  
1 = wystąpił spadek napięcia (*flaga musi być zerowana programowo*),  
0 = napięcie zasilania jest wyższe od zaprogramowanego progu zadziałania,
- bit 1      **TMR3IF: flaga przerwania przepełnienia licznika TMR3:**  
1 = nastąpiło przepełnienie rejestru TMR3  
(*flaga musi być zerowana programowo*),  
0 = nie nastąpiło przepełnienie rejestru TMR3,
- bit 0      **CCP2IF: flaga przerwania od układu CCP (Capture, Compare, PWM):**  
tryb Capture:  
1 = nastąpiło zatrzaśnięcie rejestru TMR1  
(*flaga musi być zerowana programowo*),  
0 = nie wystąpiło zatrzaśnięcie rejestru TMR1,  
tryb Compare:  
1 = nastąpiło zrównanie rejestru TMR1  
(*flaga musi być zerowana programowo*),  
0 = nie nastąpiło zrównanie rejestru TRM1,  
tryb PWM:  
nieużywany,

Legenda:

R – bit odczytywalny,      W – bit zapisywalny,      U – bit niezaimplementowany,  
0,1 – stan po włączeniu zasilania (x- stan nieokreślony),

## Rejestr PIE1

### PERIPHERAL INTERRUPT ENABLE REGISTER 1

( Rejestry *PIE1*, *PIE2* zawierają bity zezwoleń na przerwania od poszczególnych układów peryferyjnych. Aby odblokować przerwania układów peryferyjnych bit *PEIE* rejestru *INTCON* musi być ustawiony).

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSP1E	AD1E	RC1E	TX1E	SSP1E	CCP11E	TMR21E	TMR11E
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **PSP1E: zezwolenie na przerwanie zapisu/odczytu portu równoległego (zarezerwowany dla układów rodziny PIC18F2x2):**  
1 = odblokowanie przerwania operacji zapisu/odczytu,  
0 = zablokowanie przerwania operacji zapisu/odczytu,
- bit 6      **AD1E: zezwolenie na przerwanie od przetwornika A/C:**  
1 = odblokowanie przerwania od przetwornika A/C,  
0 = zablokowanie przerwania od przetwornika A/C,
- bit 5      **RC1E: zezwolenie na przerwanie od odbiornika USART:**  
1 = odblokowanie przerwania od odbiornika USART,  
0 = zablokowanie przerwania od odbiornika USART,
- bit 4      **TX1E: zezwolenie na przerwanie od nadajnika USART:**  
1 = odblokowanie przerwania od nadajnika USART,  
0 = zablokowanie przerwania od nadajnika USART,
- bit 3      **SSP1E: zezwolenie na przerwanie od układu SSP (Synchronous Serial Port):**  
1 = odblokowanie przerwania od układu SSP,  
0 = zablokowanie przerwania od układu SSP,
- bit 2      **CCP11E: zezwolenie na przerwanie od układu CCP1:**  
1 = odblokowanie przerwania od układu CCP1,  
0 = zablokowanie przerwania od układu CCP1,
- bit 1      **TMR21E: zezwolenie na przerwanie od przepełnienia licznika 2 (TMR2):**  
1 = odblokowanie przerwania od przepełnienia licznika 2 (TMR2),  
0 = zablokowanie przerwania od przepełnienia licznika 2 (TMR2),
- bit 0      **TMR11E: zezwolenie na przerwanie od przepełnienia licznika 1 (TMR1):**  
1 = odblokowanie przerwania od przepełnienia licznika 1 (TMR1),  
0 = zablokowanie przerwania od przepełnienia licznika 1 (TMR1),

Legenda:

R – bit odczytywalny,      W – bit zapisywalny,      U – bit niezaimplementowany,  
0,1 – stan po włączeniu zasilania (x- stan nieokreślony),

## Rejestr PIE2

### PERIPHERAL INTERRUPT ENABLE REGISTER 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7-5      **bity niezaimplementowane,**
- bit 4      **EEIE: zezwolenie na przerwanie od zakończenia zapisu do pamięci EPROM/FLASH:**  
1 = odblokowanie,  
0 = zablokowanie,
- bit 3      **BCLIE: zezwolenie na przerwanie przy wystąpieniu kolizji na liniach magistrali:**  
1 = odblokowanie,  
0 = zablokowanie,
- bit 2      **LVDIE: zezwolenie na przerwanie przy wykryciu spadku napięcia zasilania:**  
1 = odblokowanie,  
0 = zablokowanie,
- bit 1      **TMR3IE: zezwolenie na przerwanie od przepełnienia licznika 3 (TMR3):**  
1 = odblokowanie przerwania od przepełnienia licznika 3 (TMR3),  
0 = zablokowanie przerwania od przepełnienia licznika 3 (TMR3),
- bit 0      **CCP2IE: zezwolenie na przerwanie od układu CCP2:**  
1 = odblokowanie przerwania od układu CCP2,  
0 = zablokowanie przerwania od układu CCP2,

Legenda: R – bit odczytywalny,      W – bit zapisywalny,      U – bit niezaimplementowany, 0,1 – stan po włączeniu zasilania (x- stan nieokreślony),
--

## Rejestr IPR1

### PERIPHERAL INTERRUPT PRIORITY REGISTER 1

( Rejestry **IPR1**, **IPR2** zawierają typ priorytetu przerwania urządzeń peryferyjnych. Obsługa poziomów priorytetów wymaga ustawienia bitu **IPEN** rejestru **RCON** ).

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **PSPIP: priorytet przerwania zapisu/odczytu portu równoległego (zarezerwowany dla układów rodziny PIC18F2x2):**  
1 = wysoki priorytet,  
0 = niski priorytet,
- bit 6      **ADIP: priorytet przerwania od przetwornika A/C:**  
1 = wysoki priorytet,  
0 = niski priorytet,
- bit 5      **RCIP: priorytet przerwania od odbiornika USART:**  
1 = wysoki priorytet,  
0 = niski priorytet,
- bit 4      **TXIP: priorytet przerwania od nadajnika USART:**  
1 = wysoki priorytet,  
0 = niski priorytet,
- bit 3      **SSPIP: priorytet przerwania od układu SSP (Synchronous Serial Port):**  
1 = wysoki priorytet,  
0 = niski priorytet,
- bit 2      **CCP1IP: priorytet przerwania od układu CCP1:**  
1 = wysoki priorytet,  
0 = niski priorytet,
- bit 1      **TMR2IP: priorytet przerwania od przepełnienia licznika 2 (TMR2):**  
1 = wysoki priorytet,  
0 = niski priorytet,
- bit 0      **TMR1IP: priorytet przerwania od przepełnienia licznika 1 (TMR1):**  
1 = wysoki priorytet,  
0 = niski priorytet,

Legenda: R – bit odczytywalny,      W – bit zapisywalny,      U – bit niezaimplementowany, 0,1 – stan po włączeniu zasilania (x- stan nieokreślony),
--

## Rejestr IPR2

### PERIPHERAL INTERRUPT PRIORITY REGISTER 2

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
-	-	-	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7-5     **bity niezaimplementowane,**  
bit 4       **EEIP: priorytet przerwania od zakończenia zapisu do pamięci EEPROM/FLASH:**  
              1 = wysoki priorytet,  
              0 = niski priorytet,  
bit 3       **BCLIP: priorytet przerwania od wystąpienia kolizji na liniach magistrali:**  
              1 = wysoki priorytet,  
              0 = niski priorytet,  
bit 2       **LVDIP: priorytet przerwania od wykrycia spadku napięcia zasilania:**  
              1 = wysoki priorytet,  
              0 = niski priorytet,  
bit 1       **TMR3IP: priorytet przerwania od przepelnienia licznika 3 (TMR3):**  
              1 = wysoki priorytet,  
              0 = niski priorytet,  
bit 0       **CCP2IP: priorytet przerwania od układu CCP2:**  
              1 = wysoki priorytet,  
              0 = niski priorytet,

<p>Legenda:  R – bit odczytywalny,           W – bit zapisywalny,           U – bit niezaimplementowany,  0,1 – stan po włączeniu zasilania (x- stan nieokreślony),</p>
---

## Przerwanie zewnętrzne

Zależnie od ustawienia bitów INTEDGx rejestru **INTCON2** przerwania zewnętrzne RB0/INT0, RB1/INT1, RB2/INT2 wyzwalane są zboczem narastającym, lub opadającym. Pojawiające się ważne zbocze na którymkolwiek z wejść powoduje ustawienie odpowiedniej flagi INTxIF w rejestrze **INTCON**. Flaga ta musi być zerowana programowo w procedurze obsługi przerwania. Przerwanie może być zablokowane bitem INTxIE rejestru **INTCON**. Wszystkie przerwania zewnętrzne (INT0, INT1, INT2) mogą "obudzić" procesor ze stanu uśpienia (*SLEEP*). Jeżeli bit GIE rejestru **INTCON** jest ustawiony, po wyjściu ze stanu uśpienia procesor przechodzi do obsługi przerwania.

Bity INT1IP i INT2IP rejestru **INTCON3** określają priorytet przerwania. Przerwanie INT0 nie ma przypisanego priorytetu, zawsze traktowane jest jako przerwanie o wysokim priorytecie.

## Przerwanie od zmiany stanu PORTB

Zmiana stanu na liniach RB4:RB7 ustawia flagę RBIR rejestru **INTCON**. Przerwanie może być zablokowane/odblokowane bitem RBIE rejestru **INTCON**. Priorytet przerwania wybierany jest bitem RBIP rejestru **INTCON2**.

## Zachowywanie kontekstu podczas obsługi przerwania

Przy obsłudze przerwania licznik rozkazów PC jest umieszczany na stosie. Dodatkowo rejestry **WREG**, **STATUS** i **BSR** są umieszczane na stosie „szybkiego powrotu” (*fast return stack*). Jeżeli nie jest używany mechanizm (*fast return stack*), istnieje potrzeba programowego zachowywania zawartości rejestrów **WREG**, **STATUS** i **BSR**. Poniższy przykład obrazuje sposób zapisywania i odtwarzania rejestrów.

```
MOVWF  W_TEMP          ; zachowanie WREG w rejestrze
                          ; tymczasowym W_TEMP
MOVFF  STATUS, STATUS_TEMP ; STATUS w rejestrze STATUS_TEMP
MOVFF  BSR,    BSR_TEMP    ; RSR w rejestrze BSR_TEMP
.
.                          ; obsługa przerwania
.
MOVFF  BSR_TEMP, BSR      ; odtworzenie BSR
MOVF   W_TEMP,   W        ; odtworzenie WREG
MOVFF  STATUS_TEMP, STATUS ; odtworzenie STATUS
```

*Listing D5.1. Zachowanie kontekstu podczas obsługi przerwania*

## 7 Porty wejścia/wyjścia

### PORTA

PortA jest 7-bitowym dwukierunkowym portem we/wy. Zmiana w rejestrze kierunku **TRISA** dowolnego bitu na '1' powoduje ustawienie linii jako wejście, natomiast zmiana bitu na '0' ustawia linię jako wyjście.

Odczytując rejestr **PORTA**, odczytuje się stan linii portu, informacja zapisywana do portu umieszczana jest w rejestrach zatraskowych portu.

Nazwa	Bit	Bufor	Funkcja
<b>RA0/AN0</b>	bit0	TTL	wejście/wyjście lub wejście analogowe,
<b>RA1/AN1</b>	bit1	TTL	wejście/wyjście lub wejście analogowe,
<b>RA2/AN2/V<sub>REF-</sub></b>	bit2	TTL	wejście/wyjście lub wejście analogowe lub V <sub>REF-</sub>
<b>RA3/AN3/V<sub>REF+</sub></b>	bit3	TTL	wejście/wyjście lub wejście analogowe lub V <sub>REF+</sub>
<b>RA4/T0CKI</b>	bit4	ST	wejście/wyjście lub wejście zewnętrznego zegara dla Timer0, wyjście jest typu otwarty dren
<b>RA5/~SS/AN4/LVDIN</b>	bit5	TTL	wejście/wyjście lub wybór trybu slave dla SSP lub wejście analogowe lub wejście detekcji niskiego napięcia,
<b>OSC2/CLKO/RA6</b>	bit6		OSC2 lub wyjście zegarowe lub linia we/wy,

Legenda: TTL = wejście typu TTL, ST = wejście z przerzutnikiem Schmitta

*Tabela D6.1. Funkcje linii PORTA*

Pin RA4 jest multipleksowany z wejściem licznika Timer0 i nazywany jest RA4/T0CKI. Wejście tego pinu zawiera przerzutnik Schmitta, wyjście natomiast jest typu otwarty dren. Poziomy napięcie wejściowych wszystkich pinów portu PORTA zawierają się w standardzie TTL, natomiast porty wyjściowe są typu CMOS.

Inne piny portu PORTA są multipleksowane z analogowymi wejściami, jak i analogowymi wejściami napięcia referencyjnego V<sub>REF+</sub> i V<sub>REF-</sub>. Funkcja każdego z pinów jest ustalana przez zerowanie/ustawianie bitów kontrolnych w rejestrze **ADCON1** (rejestr kontrolny przetwornika A/D).

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>PORTA</b>	-	RA6	RA5	RA4	RA3	RA2	RA1	RA0	-x0x 0000	-u0u 0000
<b>LATA</b>	-	Rejestr danych							-xxx xxxx	-uuu uuuu
<b>TRISA</b>	-	Rejestr kierunku portu A							-111 1111	-111 1111
<b>ADCON1</b>	ADFM	ADCS2	-	-	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	00-- 0000

*Tabela D6.2. Rejestry związane z PORTA*

Po resece piny RA5 i RA3:RA0 są skonfigurowane jako wejścia analogowe i odczytywane jako '0'. Piny RA6 i RA4 skonfigurowane są jako wejścia cyfrowe.



## PORTB

Każdy pin portu PORTB ma wewnętrzny rezystor podciągający (weak internal pull-up). Pojedynczym bitem można włączyć rezystory podciągające wszystkich pinów. Jest to wykonywane przez zerowanie bitu  $\sim$ RBPU (**INTCON2**<7>). Rezystory podciągające są automatycznie odłączane, gdy dany pin jest skonfigurowany jako wyjście. Po stanie Power-on Reset wewnętrzna polaryzacja jest wyłączona.

Po Power-on-Reset linie PORTB skonfigurowane są jako wejścia cyfrowe.

Cztery piny portu PORTB, RB7:RB4, posiadają możliwość generacji przerwania, gdy nastąpi zmiana poziomu na tych pinach. Jedynie te piny, które są skonfigurowane jako wejścia mogą spowodować wygenerowanie przerwania (gdy któryś z pinów RB7:RB4 jest skonfigurowany jako wyjście, nie ma możliwości wygenerowania przerwania). Wejścia tych pinów (RB7:RB4) są porównywane ze starymi wartościami, zatrzaśniętymi w ostatnim odczycie portu PORTB. „Błąd” pochodzący z pinów RB7:RB4 jest sumowany (funkcja OR) po to, aby wygenerować przerwanie za pomocą flagi RBIF (**INTCON**<0>) mówiące o zmianie stanu na jednym z tych pinów. Przerwanie to może obudzić układ ze stanu SLEEP.

Programista w procedurze obsługi przerwania może wyzerować to przerwanie w następujący sposób:

- ⇒ wykonać dowolny odczyt lub zapis do portu PORTB (oprócz instrukcji MOVFF), zakończy to warunek wystąpienia błędu,
- ⇒ wyczyścić flagę przerwania RBIF w rejestrze INTCON,

Błąd będzie powodował ciągle ustawianie flagi RBIF. Odczytanie portu PORTB zakończy warunek wystąpienia błędu, co spowoduje, że flaga RBIF będzie mogła zostać wyzerowana.

Przerwanie na zmianę poziomu jest zalecaną metodą do budzenia układu przez wciśnięcie klawisza i sytuacji, w której port PORTB jest użyty jedynie na potrzeby przerwania. Podczas używania portu PORTB jako źródła przerwania, nie jest zalecane jego przeglądanie.

Pin RB3 może być skonfigurowany poprzez bit CCP2MX=0 rejestru **CONFIG3H**<0> jako linia modułu peryferyjnego CCP2.

1. W trybie Low Voltage ICSP pin RB5 nie jest używany jak uniwersalna linia we/wy, w czasie normalnej pracy powinna być utrzymywana w stanie niskim, zapobiega to przypadkowemu przejściu w tryb ISCP.
2. Stosując programowanie Low Voltage ICSP (LVP) rezystory podciągające linii RB5 stają się nieaktywne. Jeżeli bit **TRISB**<5>='0' pin RB5 ustawiony jest jako wyjście, bit **LATB**<5> musi być również wyzerowany.

Nazwa	Bit	Bufor	Funkcja
<b>RB0/INT0</b>	bit0	TTL/ST <sup>(1)</sup>	wejście/wyjście lub wejście zewnętrznego przerwania 0, możliwość podłączenia rezystora podciągającego.
<b>RB1/INT1</b>	bit1	TTL/ST <sup>(1)</sup>	wejście/wyjście lub wejście zewnętrznego przerwania 1, możliwość podłączenia rezystora podciągającego.
<b>RB2/INT2</b>	bit2	TTL/ST <sup>(1)</sup>	wejście/wyjście lub wejście zewnętrznego przerwania 2, możliwość podłączenia rezystora podciągającego.
<b>RB3/CCP2<sup>(3)</sup></b>	bit3	TTL/ST <sup>(4)</sup>	Wejście/wyjście gdy bit CCP2MX='1' rejestru <b>CONFIG3H</b> to wejście Compare2/wyjście Compare2/ wyjście PWM, możliwość podłączenia rezystora podciągającego.
<b>RB4</b>	bit4	TTL	wejście/wyjście (przerwanie na zmianę poziomu), możliwość podłączenia rezystora podciągającego.
<b>RB5/ PGM<sup>(5)</sup></b>	bit5	TTL/ST <sup>(2)</sup>	wejście/wyjście (przerwanie na zmianę poziomu), możliwość podłączenia rezystora podciągającego, pin programujący w trybie LVP.
<b>RB6/PGC</b>	bit6	TTL/ST <sup>(2)</sup>	wejście/wyjście (przerwanie na zmianę poziomu), możliwość podłączenia rezystora podciągającego, zegar dla programowania szeregowego.
<b>RB7/PGD</b>	bit7	TTL/ST <sup>(2)</sup>	wejście/wyjście (przerwanie na zmianę poziomu), możliwość podłączenia rezystora podciągającego, dane dla programowania szeregowego.

Legenda: TTL = wejście typu TTL, ST = wejście z przerzutnikiem Schmitta

*Tabela D6.3. Funkcje linii PORTB*

- 1: Ten bufor jest przerzutnikiem Schmitta, kiedy jest skonfigurowany jako zewnętrzne przerwianie.
- 2: Ten bufor jest przerzutnikiem Schmitta, kiedy jest używany podczas trybu programowania szeregowego.
- 3: Bitem konfiguracyjnym CCP2MX='1' wybiera się pomiędzy we/wy lub trybem CCP2.
- 4: Bufor ten jest wejściem przerzutnikiem Schmitta, kiedy jest skonfigurowany jako wejście CCP2.
- 5: Low Voltage Programming (LVP) standardowo jest aktywne, co powoduje zablokowanie funkcji RB5. LVP musi być wyłączone, aby uaktywnić RB5 jako pin wejściowy lub wyjściowy i pozwolić na maksymalną kompatybilność dla pozostałych pinów kontrolera.

## PORTC

**PORTC** oprócz podstawowych funkcji we/wy daje możliwość korzystania z interfejsów komunikacyjnych SPI, I<sup>2</sup>C oraz USART. Przy konfiguracji interfejsu komunikacyjnego należy pamiętać o odpowiednim zdefiniowaniu bitów kierunku rejestru **TRISC**. Niektóre urządzenia wymagają, aby dany pin był skonfigurowany jako wejście, podczas, gdy inne wymagają ustalenia danego pinu jako wyjście. Ponadto **PORTC** daje możliwość podłączenia zewnętrznego oscylatora dla modułu TIMER1, może również pełnić rolę wejść dla modułu Compare1.

Linia RC1 konfigurowalna jest bitem CCP2MX jako urządzenie peryferyjne układu CCP2.

Po Power-on-Reset linie PORTC skonfigurowane są jako wejścia cyfrowe.

Nazwa	Bit	Bufor	Funkcja
-------	-----	-------	---------

<b>RC0/T1OSO/T1CKI</b>	bit0	ST	wejście/wyjście, wyjście oscylatora modułu Timer1/wejście zegara modułu Timer1.
<b>RC1/T1OSI/CCP2</b>	bit1	ST	wejście/wyjście, wejście oscylatora modułu Timer1 lub wejście modułu Capture2/wyjście modułu Compare2/wyjście modułu PWM2.
<b>RC2/CCP1</b>	bit2	ST	wejście/wyjście, wejście modułu Capture1/wyjście modułu Compare1/wyjście modułu PWM1.
<b>RC3/SCK/SCL</b>	bit3	ST	RC3 może także być zegarem dla modułów SPI i I <sup>2</sup> C.
<b>RC4/SDI/SDA</b>	bit4	ST	RC4 może także być pinem danych wejściowych modułu SPI (tryb SPI) lub danych wejściowych /wyjściowych modułu I <sup>2</sup> C (tryb I <sup>2</sup> C).
<b>RC5/SDO</b>	bit5	ST	wejście/wyjście, wyjście danych modułu SSP (Synchronous Serial Port).
<b>RC6/TX/CK</b>	bit6	ST	wejście/wyjście, zegar synchroniczny/ asynchroniczna transmisja modułu USART.
<b>RC7/RX/DT</b>	bit7	ST	wejście/wyjście, asynchroniczny odbiór lub Synchroniczne dane modułu USART.

Legenda: ST = wejście z przerzutnikiem Schmitta

*Tabela D6.4. Funkcje linii PORTC*

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>PORTC</b>	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
<b>LATC</b>	Rejestr danych wyjściowych								xxxx xxxx	uuuu uuuu
<b>TRISC</b>	Rejestr kierunku portu C								1111 1111	1111 1111

*Tabela D6.5. Rejestry związane z PORTC*

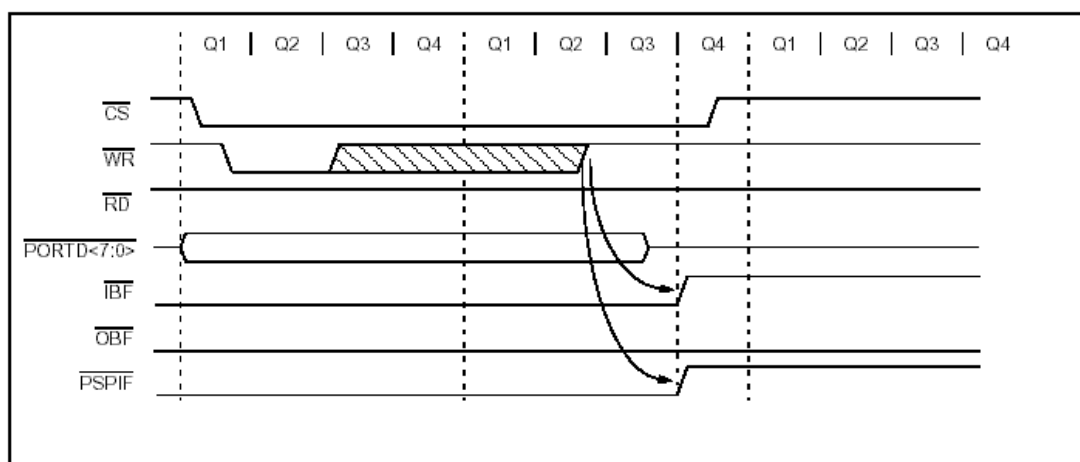
## PORTD

Zależnie od ustawienia bitu PSPMODE rejestru **TRISE<4>** PORTD pracuje jako 8-bitowy port równoległy (*Parallel Slave Port*), lub jego linie pełnią funkcje uniwersalnych linii we/wy. Jest on asynchronicznie odczytywalny i zapisywalny poprzez linie sterujące PORTE (RE0/~RD, RE1/~WR).

Moduł PSP może być bezpośrednio podłączony do 8-bitowej magistrali komputerowej. Zewnętrzny mikroprocesor może czytać i zapisywać do portu PORTD jak do 8-bitowego zatrasku. Ustawiając bit PSPMODE='1' linie PORTE stają się odpowiednio RE0/~RD – wejściem zapisu, RE1/~WR – wejściem odczytu, RE2/~CS – wejściem wyboru układu. Wejścia te pełnią rolę funkcji sterujących pracą portu równoległego. Dodatkowo bity w rejestrze kierunku **TRISE** oraz bity PCFG2:PCFG0 rejestru **ACCON1** muszą być skonfigurowane jako wejścia cyfrowe.

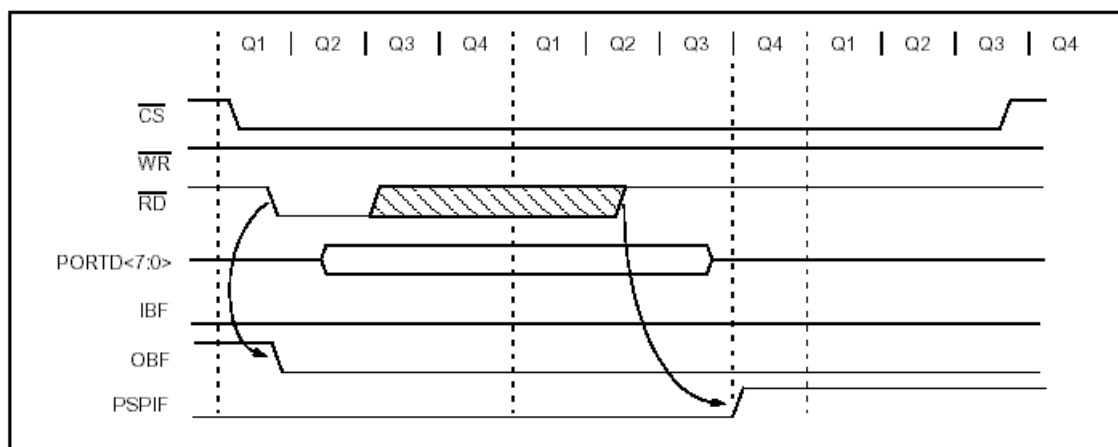
Zapis do modułu PSP następuje, gdy na liniach ~CS i ~WR wystąpi stan niski. Odczyt natomiast wystąpi, gdy na liniach ~CS i ~RD pojawi się również stan niski.

PORTE staje się portem uniwersalnych linii we/wy, gdy bit PSPMODE='0' rejestru **TRISE**, oraz bity kierunku TRISE2:TRISE0 ustawione są jako wejścia, rejestr **ADCON1** musi być również skonfigurowany do obsługi cyfrowych linii wejściowych.



Rys. 6.1. Przebiegi zapisu do PSP

Zapis danych do portu równoległego, Rys. 6.1. następuje w momencie, gdy obie linie  $\sim$ CS i  $\sim$ WR znajdują się w stanie niskim. Kiedy którakolwiek z linii  $\sim$ CS lub  $\sim$ WR znajdzie się w stanie wysokim, ustawiona zostanie flaga odebrania bajtu danych - IBF (**TRISE<7>**) oraz flaga przerwania PSPIF (**PIR1<7>**). Proces zapisu do portu równoległego trwa dwa cykle maszynowe. Flagą IBF może zostać wyczyszczona jedynie poprzez odczyt PORTD. W momencie gdy dana z bufora wejściowego nie zostanie odczytana, a nastąpi kolejna próba zapisu danej z portu, ustawiona zostanie flaga przepełnienia bufora wejściowego - IBOV (**TRISE<5>**).



Rys. 6.2. Przebiegi odczytu z PSP

Odczyt danych z portu równoległego, Rys. 6.2 następuje w momencie, gdy obie linie  $\sim$ CS i  $\sim$ RD znajdują się w stanie niskim. Flagą odczytu bufora wyjściowego - OBF (**TRISE<6>**) zostaje od razu wyczyszczona sygnalizując gotowość do odczytu danych z zatrząsków portu PORTD przez zewnętrzną szynę.

Kiedy którakolwiek z linii  $\sim$ CS lub  $\sim$ RD znajdzie się w stanie wysokim, ustawiona zostanie flaga przerwania PSPIF (**PIR1**<7>), oznaczając tym samym koniec operacji odczytu. Flaga OBF pozostanie w stanie niskim, aż do momentu pojawienia się nowej danej do odczytania.

W momencie, gdy moduł PSP nie jest aktywny bity IBF oraz OBF są wyzerowane. Bit IBOV musi zostać wyzerowany programowo, jeżeli poprzednio został ustawiony.

Przerwanie jest generowane i prezentowane poprzez flagę PSPIF po zakończeniu każdej operacji zapisu bądź odczytu. Flaga ta, musi zostać wyzerowana programowo. Przerwanie od modułu PSP może zostać zablokowane poprzez wyzerowanie bitu zezwolenia PSPIE (**PIE1**<7>).

Nazwa	Bit	Bufor	Funkcja
RD0/PSP0	bit0	ST/TTL <sup>(1)</sup>	wejście/wyjście lub pin portu równoległego bit0,
RD1/PSP1	bit1	ST/TTL <sup>(1)</sup>	wejście/wyjście lub pin portu równoległego bit1,
RD2/PSP2	bit2	ST/TTL <sup>(1)</sup>	wejście/wyjście lub pin portu równoległego bit2,
RD3/PSP3	bit3	ST/TTL <sup>(1)</sup>	wejście/wyjście lub pin portu równoległego bit3,
RD4/PSP4	bit4	ST/TTL <sup>(1)</sup>	wejście/wyjście lub pin portu równoległego bit4,
RD5/PSP5	bit5	ST/TTL <sup>(1)</sup>	wejście/wyjście lub pin portu równoległego bit5,
RD6/PSP6	bit6	ST/TTL <sup>(1)</sup>	wejście/wyjście lub pin portu równoległego bit6,
RD7/PSP7	bit7	ST/TTL <sup>(1)</sup>	wejście/wyjście lub pin portu równoległego bit7,

1: Bufor wejściowy jest przetrzutnikiem Schmitta, kiedy skonfigurowany jest jako uniwersalne linie we/wy, lub TTL w trybie portu równoległego.

Tabela D6.6. Funkcje linii PORTD

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
PORTD	RD7	RAD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
LATD	Rejestr danych wyjściowych								xxxx xxxx	uuuu uuuu
TRISD	Rejestr kierunku portu D								1111 1111	1111 1111
TRISE	IBF	OBF	IBOV	PSPMODE	-	Rejestr kierunku PORTE			0000 -111	0000 -111

Tabela D6.7. Rejestry związane z PORTD

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
PORTD	Zatrask danych portu przy zapisie, stan linii portu przy odczycie								xxxx xxxx	uuuu uuuu
LATD	Dane wyjściowe (stan zatrasków wyjściowych)								xxxx xxxx	uuuu uuuu
TRISD	Rejestr kierunku portu D								1111 1111	1111 1111
PORTE	-	-	-	-	-	RE2	RE1	RE0	---- -000	---- -000
LATE	-	-	-	-	-	Rejestr danych wyjściowych			---- -xxx	---- -uuu
TRISE	IBF	OBF	IBOV	PSPMODE	-	Rejestr kierunku portu E			0000 -111	0000 -111
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBF	0000 000x	0000 000u
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
ADCON1	ADFM	ADCS2	-	-	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	00-- 0000

Tabela D6.8. Rejestry związane z portem równoległym

## PORTE

Rejestr **TRISE** określa kierunek pinów, nawet kiedy są one używane jako analogowe wejścia. Programista musi zapewnić, by bity w rejestrze **TRISE** były ustawione (=1), kiedy piny zostaną użyte jako wejścia analogowe.

**PORTE** składa się z trzech linii (RE0/~RD/AN5, RE1/~WR/AN6, RE2/~CS/AN7) posiadających na wejściu przerzutniki Schmitta. Piny portu **PORTE** są multipleksowane z analogowymi wejściami, oraz pełnią rolę linii sterujących dla modułu PSP.

Po Power-on-Reset linie PORTE skonfigurowane są jako wejścia analogowe.

Nazwa	Bit	Bufor	Funkcja
RE0/~RD/AN5	bit0	ST/TTL	wejście/wyjście, linia kontrolna odczytu modułu PSP lub wejście analogowe:  ~RD 1 = operacja nie czytaj, 0 = operacja czytaj, czytaj rejestry PORTD (jeżeli układ jest wybrany),
RE1/~WR/AN6	bit1	ST/TTL	wejście/wyjście, linia kontrolna zapisu modułu PSP lub wejście analogowe:  ~WR 1 = operacja nie zapisuj, 0 = operacja zapisz, zapisz do rejestrów PORTD (jeżeli układ jest wybrany),
RE2/~CS/AN7	bit2	ST/TTL	wejście/wyjście, linia kontrolna wyboru układu modułu PSP lub wejście analogowe:  ~CS 1 = układ nie jest wybrany, 0 = układ jest wybrany,

*Tabela D6.9. Funkcje linii PORTE*

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>PORTE</b>	-	-	-	-	-	RE2	RE1	RE0	---- -000	---- -000
<b>LATE</b>	-	-	-	-	-	Rejestr danych wyjściowych			---- -xxx	---- -uuu
<b>TRISE</b>	IBF	OBF	IBOV	PSP MODE	-	Rejestr kierunku portu E			0000 -111	0000 -111
<b>ADCON1</b>	ADFM	ADCS2	-	-	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	00--0000

*Tabela D6.10. Rejestry związane z PORTE*

## Rejestr TRISE

R-0	R-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1
IBF	OBF	IBOV	PSPMODE	-	TRISE2	TRISE1	TRISE0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **IBF: bit statusu bufora wejściowego:**  
 1 = bajt został odebrany i oczekuje na odczyt przez CPU,  
 0 = nie odebrano bajtu,
- bit 6      **OBF: bit statusu bufora wyjściowego:**  
 1 = bufor wyjściowy zawiera poprzednio zapisane bajt,  
 0 = bufor wyjściowy został odczytany,
- bit 5      **IBOV: bit przepelnienie bufora wejściowego:**  
 1 = zapis nastąpił gdy poprzednie słowo nie zostało odczytane  
*(musi być zerowany programowo),*  
 0 = nie nastąpiło przepelnienie,
- bit 4      **PSPMODE: tryb portu równoległego:**  
 1 = tryb portu równoległego,  
 0 = linie pracują jako uniwersalne we/wy,
- bit 3      **bit niezaimplementowany**
- bit 2      **TRISE2: bit kierunku PORTE:**  
 1 = wejście,  
 0 = wyjście,
- bit 1      **TRISE1: bit kierunku PORTE:**  
 1 = wejście,  
 0 = wyjście,
- bit 0      **TRISE0: bit kierunku PORTE:**  
 1 = wejście,  
 0 = wyjście,

Legenda: R – bit odczytywalny,      W – bit zapisywalny,      U – bit niezaimplementowany, 0,1 – stan po włączeniu zasilania (x- stan nieokreślony),
--

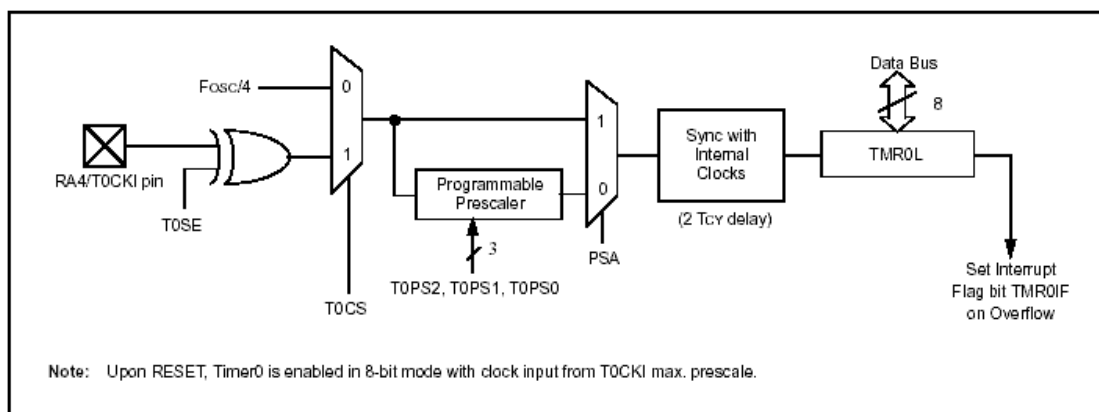
## 8 Moduły liczników

Jednym z podstawowych rodzajów urządzeń peryferyjnych umieszczanych wewnątrz struktury mikrokontrolerów są układy licznikowe. W przypadku mikrokontrolera PIC18F452 wyróżnia się cztery układy liczników TIMER0, TIMER1, TIMER2, TIMER3. Wszystkie moduły liczników oprócz licznika TIMER2 dają możliwość wyboru sygnały zegarowego pomiędzy sygnałem pochodzącym od zegara wewnętrznego, a sygnałem zewnętrznym.

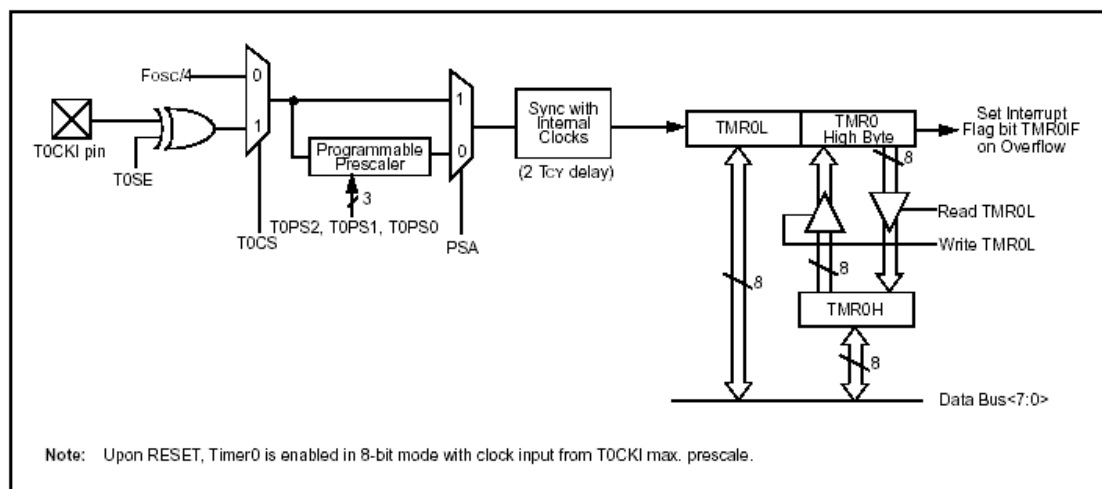
### TIMER0

Główne cechy licznika TIMER0

- ⇒ programowo dokonywany wybór między 8 lub 16 bitowym licznikiem/zegarem,
- ⇒ 8-bitowy dedykowany prescaler,
- ⇒ wybór zewnętrznego lub wewnętrznego zegara,
- ⇒ przerwanie przy przepełnieniu z FFh na 00h w trybie 8-bitowym i z FFFFh na 0000h w trybie 16-bitowym,
- ⇒ wybór zbocza zewnętrznego zegara,



Rys. 7.1. Schemat blokowy licznika TIMER0 w trybie 8-bitowym



Rys D7.1. Schemat blokowy licznika TIMER0 w trybie 16-bitowym



## Działanie licznika **TIMER0**

Tryb pracy licznika wybierany jest bitem **T0CS**. Wyzerowanie tego bitu powoduje przejście w tryb Timer (zliczanie impulsów wewnętrznych), rejestr licznika zostaje zwiększany w każdym cyklu instrukcji (bez preskalera). Jeżeli nastąpi zapis do rejestru **TMR0L** zliczanie jest wstrzymane na następne dwa cykle. Użytkownik może to obejść wpisując do **TMR0L** wartość zmodyfikowaną uwzględniającą zgubione cykle.

Ustawiając bit **T0CS=1** **TIMER0** przechodzi w tryb licznika. W trybie tym, zależnie od ustawienia bitu **T0SE**, rejestr **TMR0L** zwiększa się na każde narastające bądź opadające zbocze sygnału podanego na pin **RA4/T0CKI**.

## Preskaler

Bity **PSA** i **POPS2:POPS0** rejestru **T0CON** wyznaczają przydzielenie preskalera i jego współczynnik podziału. Wyzerowanie bitu **PSA** spowoduje przypisanie preskalera do modułu **TIMER0**, umożliwiając wybór współczynnika podziału. Przydzielenie preskalera do **TIMER0** powoduje, że wszystkie instrukcje zapisu do **TMR0L** (**CLRF TMR0**, **MOVWF TMR0**, **BSF TMR0...**) powodują wyczyszczenie preskalera.

Zapisywanie do **TMR0L**, kiedy prescaler jest przypisany do **Timer0**, wyczyści współczynnik prescalera ale nie zmieni jego przypisania.

Przypisywanie preskalera jest kontrolowane programowo i może być zmieniane podczas normalnej pracy programu.

## Przerwanie od **TIMER0**

Przerwanie od licznika **Timer0** jest generowane, kiedy rejestr **TMR0** przepełni się z **FFh** na **00h** w trybie 8-bitowym lub z **FFFFh** na **0000h**. Wystąpienie przepełnienia powoduje ustawienie flagi **TMR0IF** w rejestrze **INTCON**. Przerwanie może być zamaskowane przez wyzerowanie bitu **TMR0IE** w rejestrze **INTCON**. Bit **TMR0IF** musi być wyzerowany programowo przez procedurę obsługi przerwania przed ponownym zgłoszeniem przerwania. Przerwanie **TMR0** nie może obudzić procesora z trybu **SLEEP**, ponieważ w tym trybie licznik jest wyłączony.

## Tryb 16-bitowy, zapis i odczyt

Rejestr **TMR0H** nie jest starszym bajtem zegara/licznika w trybie 16-bitowym, ale buforem starszego bajtu modułu **TIMER0**. Starszy bajt nie ma możliwości bezpośredniego zapisu/odczytu. Rejestr **TMR0H** jest uaktualniany wartością starszego bajtu przy operacji czytania rejestru **TMR0L**. Daje to możliwość odczytu 16-bitów modułu **TIMER0** w jednym cyklu rozkazowym.

Zapis starszego bajtu do modułu **TIMER0** odbywa się również za pośrednictwem rejestru **TMR0H**. Starszy bajt jest uaktualniany zawartością rejestru **TMR0H** przy operacji zapisu do **TMR0L**. Daje to możliwość jednoczesnego uaktualnienia 16-bitów modułu **TIMER0**.

## Rejestr T0CON

(Stosowany jest do kontroli pracy licznika TIMER0)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7     **TMR0ON: włącz/wyłącz TIMER0:**  
 1 = odblokowanie TIMER0,  
 0 = zatrzymanie TIMER0,
- bit 6     **T08BIT: tryb pracy licznika 8-bitowy/16-bitowy:**  
 1 = licznik 8-bitowy,  
 0 = licznik 16-bitowy,
- bit 5     **T0CS: wybór źródła zegarowego:**  
 1 = zewnętrzne źródło sygnału na pinie T0CKI,  
 0 = wewnętrzne źródło sygnału (cykl rozkazowy),
- bit 4     **T0SE: wybór zbocza sygnału:**  
 1 = zwiększenie licznika na zbocze opadające sygnału podanego na pin T0CKI,  
 0 = zwiększenie licznika na zbocze narastające sygnału podanego na pin T0CKI,
- bit 3     **PSA: przyłączenie preskalera do licznika TIMER0:**  
 1 = prescaler nie jest podłączony do TIMER0, źródło sygnału zegarowego omija prescaler,  
 0 = prescaler podłączony do TIMER0, źródło sygnału zegarowego podawane jest na wejście preskalera,
- bit 2-0   **T0PS2: T0PS0 wybór wartości preskalera:**  
 111 = 1:256  
 110 = 1:128  
 101 = 1:64  
 100 = 1:32  
 011 = 1:16  
 010 = 1:8  
 001 = 1:4  
 000 = 1:2

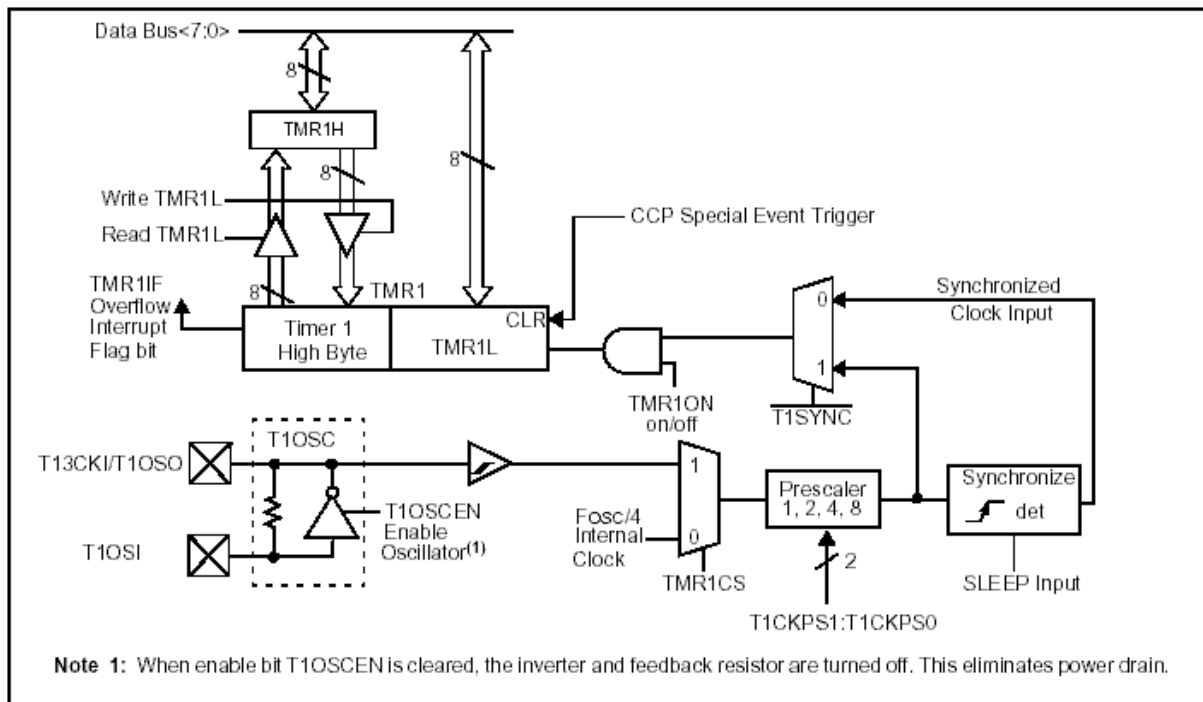
Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>TMR0L</b>	Rejestr młodszego bajtu modułu TIMER0								xxxx xxxx	uuuu uuuu
<b>TMR0H</b>	Rejestr starszego bajtu modułu TIMER0								0000 0000	0000 0000
<b>INTCON</b>	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>T0CON</b>	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	1111 1111
<b>TRISA</b>	-	Rejestr kierunku portu A							-111 1111	-111 1111

Tabela D7.1. Rejestry związane z TIMER0

## TIMER1

Główne cechy licznika TIMER1:

- ⇒ 16-bitowy licznik/zegar (dwa 8-bitowe rejestry **TMR1H** i **TMR1L**),
- ⇒ możliwość zapisu i odczytu obu rejestrów,
- ⇒ wybór zewnętrznego lub wewnętrznego zegara,
- ⇒ przerwanie przy przepełnieniu z FFFFh do 0000h,
- ⇒ RESET od modułu CCP (specjalny tryb wyzwalania),



Rys. 7.2. Schemat blokowy licznika TIMER1 w trybie 16-bitowego zapisu/odczytu

### Działanie licznika TIMER1

Licznik TIMER1 może pracować w jednym z trzech trybów:

- jako zegar,
- jako licznik synchroniczny,
- jako licznik asynchroniczny,

Tryb pracy licznika TIMER1 uzależniony jest od ustawionego bitu TMR1CS w rejestrze **TMR1CON**. Gdy TMR1CS='0' licznik zwiększa się za każdym cyklem zegarowym, gdy TMR1CS='1' licznik zwiększa się za każdym narastającym zboczem zewnętrznego sygnału zegarowego.

Kiedy zewnętrzny oscylator podłączony jest do wejść licznika Timer1 (bit T1OSCEN jest ustawiony), piny RC1/T1OSI/CCP2 oraz RC0/T1OSO/T1CKI stają się wejściami. To znaczy wartość TRISC<1:0> jest ignorowana i piny te czytane są jako '0'.

Timer1 posiada także wewnętrzne wejście RESET. Sygnał RESET może być wywołany przez jeden z dwóch modułów CCP.

Do modułu TIMER1 może zostać podłączony zewnętrzny oscylator. Przeznaczony jest on głównie do aplikacji wymagających utrzymania czasu rzeczywistego. Powszechnie w

tego typu aplikacjach stosowany jest rezonator kwarcowy o częstotliwości 32.768kHz zapewniający odpowiednią częstotliwość taktowania, oraz możliwie mały pobór mocy. TIMER1 zapewnia działanie licznika w czasie trybu uśpienia mikrokontrolera (SLEEP), przez co urządzenie może pracować przy obniżonym poborze mocy.

### Resetowanie licznika TIMER1 poprzez moduł CCP

Jeżeli moduł CCP jest skonfigurowany w trybie Compare zdolnym wygenerować zdarzenie specjalne (*special event trigger*), bity CCP1M3:CCP1M0='1011' rejestru **CCP1CON** powodują zresetowanie układu TIMER1, jeżeli przetwornik A/C jest odblokowany, rozpoczęcie konwersji A/C. Aby móc korzystać z możliwości resetowania licznika TIMER1 zdarzeniem specjalnym od modułu CCP, TIMER1 musi być skonfigurowany do pracy w trybie zegara, lub licznika synchronicznego. Jeżeli operacja zapisu do rejestrów TIMER1 zbiega się z wystąpieniem (*special event trigger*) od modułu CCP1, zapis zostanie wykonany jako pierwszy.

Wyzwolenie zdarzeniem specjalnym od modułu CCP1 nie ustawia bitu flagi przerwania TMR1IF w rejestrze **PIR1<0>**

### Przerwanie od TIMER1

Para rejestrów TMR1 (**TMR1H:TMR1L**) jest zwiększana od 0000h do FFFFh, po przepełnieniu zawiera 0000h. Przerwanie od licznika TIMEMR1, jeśli jest włączone, generowane jest podczas przepełnienia i sygnalizowane przez ustawienie flagi TMR1IF rejestru **PIR1<0>**. To przerwanie może być włączane/wyłączane przez ustawienie/wyzerowanie bitu maski przerwania TMR1IE rejestru **PIE1<0>**.

### TIMER1 - tryb 16-bitowy, zapis i odczyt

Jeżeli bit RD16 rejestru **T1CON<7>** jest ustawiony wówczas możliwy staje się zapis/odczyt w trybie 16-bitowym. Starszy bajt mapowany jest przez rejestr **TMR1H**, który pełni rolę bufora. Rejestr **TMR1H** jest uaktualniany wartością starszego bajtu przy operacji czytania rejestru TMR1L. Daje to możliwość odczytu 16-bitów modułu TIMER1 w jednym cyklu rozkazowym.

Zapis starszego bajtu do modułu TIMER1 odbywa się również za pośrednictwem rejestru **TMR1H**. Starszy bajt jest uaktualniany zawartością rejestru **TMR1H** przy operacji zapisu do **TMR1L**. Daje to możliwość jednoczesnego uaktualnienia 16-bitów modułu TIMER1.

## Rejestr T1CON

(Rejestr T1CON stosowany jest do kontroli pracy licznika TIMER1)

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	-	T1CKPS1	T1CKPS0	T1OSCEN	~T1SYNC	TMR1CS	TMR1ON
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **RD16: odblokowanie 16-bitowego trybu zapisu/odczytu:**  
 1=odblokowanie zapisu/odczytu rejestrów TIMER1 w jednej 16-bit operacji,  
 0=odblokowanie zapisu/odczytu rejestrów TIEMR1 w dwóch 8-bit operacjach,
- bit 6      **bit niezaimplementowany,**
- bit 5-4    **T1CKPS1: T1CKPS0: wybór wartości preskalera:**  
 11 = 1:8  
 10 = 1:4  
 01 = 1:2  
 00 = 1:1
- bit 3      **T1OSCEN: załączenie oscylatora dla TIMER1:**  
 1 = oscylator włączony,  
 0 = oscylator wyłączony (inwerter oscylatora jest wyłączony, aby  
 wyeliminować pobór mocy).
- bit 2      **~T1SYNC: bit synchronizacji Timer1 z zegarem zewnętrznym:**  
gdy TMR1CS=1:  
 1 = nie synchronizuje zewnętrznego wejścia zegarowego,  
 0 = synchronizuje zewnętrzne wejście zegarowe,  
gdy TMR1CS=0:  
 Bit jest ignorowany. Timer1 używa zegara wewnętrznego, gdy TRM1CS=0.
- bit 1      **TMR1CS: wybór zegara dla TIMER1:**  
 1 = zewnętrzne źródło zegara z pinu RC0/T1OSO/T13CKI  
 (na zboczu narastającym),  
 0 = zegar wewnętrzny (Fosc/4),
- bit 0      **TMR1ON: bit włączenia TIMER1:**  
 1 = TIMER1 włączony,  
 0 = TIMER1 zatrzymany,

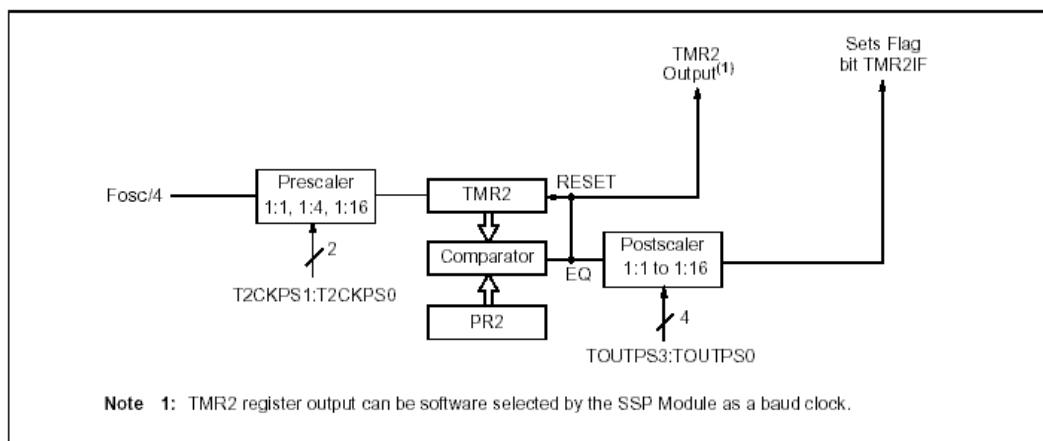
Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
TMR1L	Rejestr młodszego bajtu modułu TIMER1								xxxx xxxx	uuuu uuuu
TMR1H	Rejestr starszego bajtu modułu TIMER1								xxxx xxxx	uuuu uuuu
T1CON	RD16	-	T1CKP S1	T1CKP S0	T1OSC EN	~T1SY NC	TMR1C S	TMR1ON	0-00 0000	u-uu uuuu

Tabela D7.2. Rejestry związane z TIMER1

## TIMER2

Główne cechy licznika TIMER2:

- ⇒ 8-bitowy zegar, rejestr **TMR2**,
- ⇒ 8-bitowy rejestr okresu, **PR2**,
- ⇒ możliwość zapisu/odczytu,
- ⇒ programowalny preskaler (1:1, 1:4, 1:16),
- ⇒ programowalny postskaler (od 1:1 do 1:16),
- ⇒ przerwanie generowane na skutek zrównania rejestrów **TMR2** i **PR2**,
- ⇒ wykorzystanie wyjścia TMR2 w module SSP przy generacji przesunięcia sygnału zegarowego,



Rys. 7.3. Schemat blokowy TIMER2

### Działanie licznika TIMER2

TIMER2 może być używany przy generowaniu sygnału PWM. Rejestr **TMR2** jest odczytywalny/zapisywalny, oraz zerowany przy każdorazowym sygnale RESET urządzenia. Wejściowy sygnał zegarowy ( $F_{osc}/4$ ) jest dzielony w preskalerze przez współczynniki podziału 1:1, 1:4, 1:16, które są ustawiane bitami T2CKPS1:T2CKPS0 rejestru **T2CON**<1:0>. Sygnał z wyjścia TMR2 przechodzi przez 4-bitowy postskaler (współczynniki podziału zawierają się od 1:1 do 1:16) i generuje przerwanie (zatrzaszkowane w fladze TMR2IF (**PIR1**<1>)).

TIMER2 posiada 8-bitowe rejestry **PR2**, służący do wyznaczania okresu (*period register*) przebiegu PWM. Licznik się zwiększa od 00h do momentu zrównania się z wartością rejestru **PR2**, po czym następuje jego wyzerowanie. Rejestr **PR2** ma możliwość odczytu i zapisu. Po stanie RESET rejestr **PR2** inicjalizowany jest wartością FFh.

Wyjście TMR2 (przed postskalerem) dołączone jest do SSP, który może służyć opcjonalnie do generowania przesunięcia zegarowego. Rejestr

Zarówno preskaler, jak i postskaler, mogą być wyzerowane, gdy wystąpi:

- zapis do rejestru **TMR2**,
- zapis do rejestru **T2CON**,
- sygnał RESET (*Power-on Reset, ~MCLR Reset, Watchdog Timer Reset, Brown-out Reset*),

## Rejestr T2CON

(Rejestr **T2CON** stosowany jest do kontroli pracy licznika **TIMER2**)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	

- bit 7            **bit niezaimplementowany,**
- bit 6-3        **TOUTPS3:TOUTPS0: wybór wartości postskalera:**  
 0000 = 1:1  
 0001 = 1:2  
 .  
 .  
 .  
 1111 = 1:16
- bit 2            **TMR2ON: włączenie TIMER2:**  
 1 = TIMER2 włączony,  
 0 = TIMER2 wyłączony,
- bit 1-0        **T2CKPS1: T2CKPS0: wybór wartości preskalera:**  
 00 = 1:1  
 01 = 1:4  
 1x = 1:16

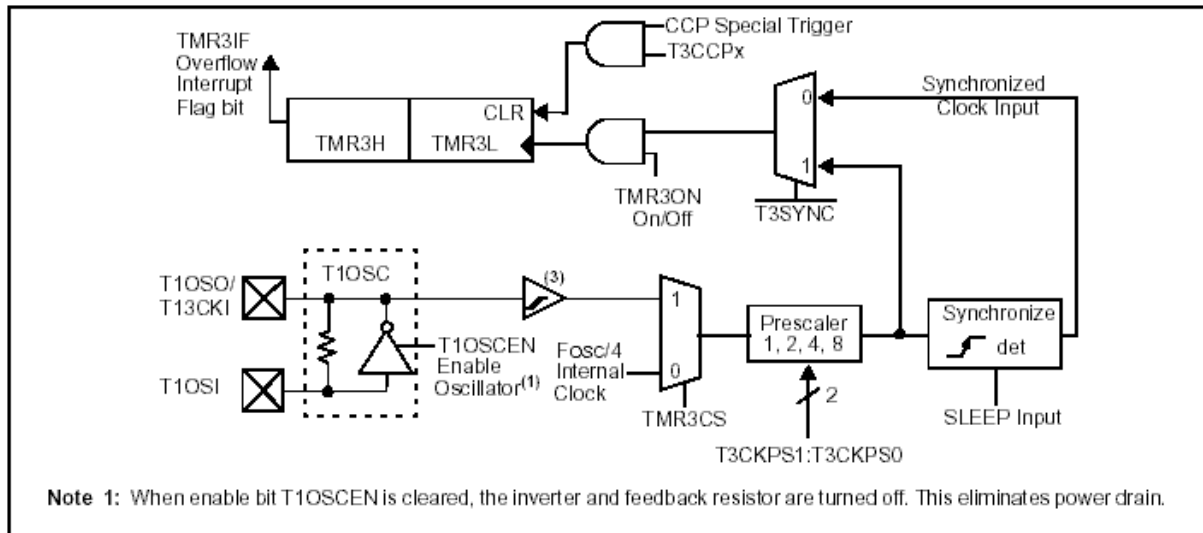
Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>INTCON</b>	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>TMR2</b>	Rejestr modułu TIMER2								0000 0000	0000 0000
<b>T2CON</b>	-	TOUT PS3	TOUT PS2	TOUT PS1	TOUT PS0	TMR2 ON	T2CK PS1	T2CK PS0	-000 0000	-000 0000
<b>PR2</b>	Rejestr okresu								1111 1111	1111 1111

Tabela D7.3. Rejestry związane z **TIMER2**

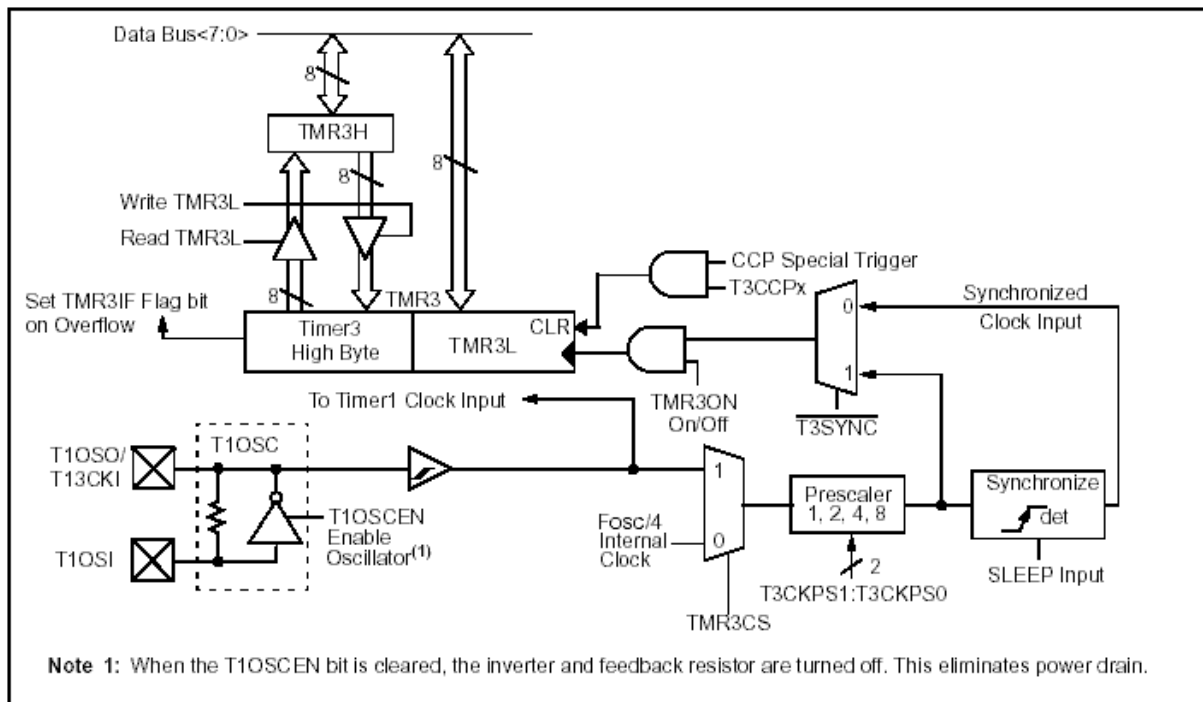
## TIMER3

Główne cechy licznika TIMER3:

- ⇒ 16-bitowy zegar/licznik (dwa rejestry 8-bitowe **TMR3H:TMR3L**),
- ⇒ możliwość czytania/zapisu obu rejestrów,
- ⇒ wybór zewnętrznego lub wewnętrznego źródła zegarowego,
- ⇒ przerwanie przy przepełnieniu licznika z FFFFh do 0000h,
- ⇒ RESET od układu CCP,



Rys. 7.4. Schemat blokowy TIMER3



Rys. 7.5. Schemat blokowy TIMER3 w konfiguracji 16-bitowej



## Działanie licznika TIMER3

Licznik TIMER3 może pracować w jednym z trzech trybów:

- jako zegar,
- jako licznik synchroniczny,
- jako licznik asynchroniczny,

Tryb pracy licznika TIMER3 uzależniony jest od ustawionego bitu TMR3CS w rejestrze **TMR3CON<1>**. Gdy TMR3CS='0' licznik zwiększa się za każdym cyklem zegarowym, gdy TMR3CS='1' licznik zwiększa się za każdym narastającym zboczem sygnału zegarowego podłączonego do licznika TIMER1, lub wewnętrznego oscylatora, jeżeli jest odblokowany.

Kiedy oscylator licznika TIMER1 jest aktywny (T1OSCEN='1') piny RC1/T1OSI/CCP2 oraz RC0/T1OSO/T1CKI stają się wejściami. Powoduje to, że wartości bitów **TRISC<1:0>** są ignorowane i piny te czytane są jako '0'.

Timer1 posiada także wewnętrzne wejście RESET. Sygnał RESET może być wywołany przez jeden z dwóch modułów CCP.

Oscylator licznika TIMER3 może być używany jako źródło sygnału zegarowego dla licznika TIMER3.

Para rejestrów **TMR3H:TMR3L** zwiększa się od 0000h do FFFFh. Przepelnienie licznika jest sygnalizowane ustawieniem flagi TMR3IF w rejestrze **PIR2<1>**. Pojawienie się przerwania może być zablokowane/odblokowane poprzez bit TMR3IE rejestru **PIE2<1>**

Jeżeli moduł CCP jest skonfigurowany w trybie Compare zdolnym wygenerować zdarzenie specjalne (*special event trigger*), bity CCP1M3:CCP1M0='1011' rejestru **CCP1CON** powodują zresetowanie układu TIMER3. Aby móc korzystać z możliwości resetowania licznika TIMER3 zdarzeniem specjalnym od modułu CCP, TIMER3 musi być skonfigurowany do pracy w trybie zegara, lub licznika synchronicznego. Jeżeli operacja zapisu do rejestrów TIMER3 zbiega się z wystąpieniem (*special event trigger*) od modułu CCP1, zapis zostanie wykonany jako pierwszy. W tym trybie pracy rejestry **CCPR1H:CCPR1L** stają się rejestrami okresu dla licznika TIMER3.

Wyzwolenie zdarzeniem specjalnym od modułu CCP1 nie ustawia bitu flagi przerwania TMR3IF w rejestrze **PIR1<0>**

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>INTCON</b>	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR2</b>	-	-	-	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	-- -0 0000	-- -0 0000
<b>PIE2</b>	-	-	-	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	-- -0 0000	-- -0 0000
<b>IPR2</b>	-	-	-	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	-- -1 1111	-- -1 1111
<b>TMR3L</b>	Rejestr młodszego bajtu licznika TIMER3								xxxx xxxx	uuuu uuuu
<b>TMR3H</b>	Rejestr starszego bajtu licznika TIMER3								xxxx xxxx	uuuu uuuu
<b>T1CON</b>	RD16	-	T1CKP S1	T1CKP S0	T1OSC EN	~T1 SYNC	TMR1 CS	TMR1ON	0-00 0000	u-uu uuuu
<b>T3CON</b>	RD16	T3CCP2	T3CKP S1	T3CKP S0	T3CCP1	~T3 SYNC	TMR3 CS	TMR3ON	0000 0000	uuuu uuuu

Tabela D7.4. Rejestry związane z TIMER3

## Rejestr T3CON

(Rejestr T3CON stosowany jest do kontroli pracy licznika TIMER3)

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	~T3SYNC	TMR3CS	TMR3ON
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **RD16: odblokowanie 16-bitowego trybu zapisu/odczytu:**  
1=odblokowanie zapisu/odczytu rejestrów TIMER3 w jednej 16-bit operacji,  
0=odblokowanie zapisu/odczytu rejestrów TIEMR3 w dwóch 8-bit operacjach,
- bit 6      **T3CCP2: przyporządkowanie TIMER1 i TIMER3 do modułów CCPx,**  
1x = TIMER3 jest źródłem sygnału zegarowego dla compare/capture modułów CCP,  
01 = TIMER3 jest źródłem sygnału zegarowego dla compare/capture modułu CCP2,  
      TIMER1 jest źródłem sygnału zegarowego dla compare/capture modułu CCP1,  
00 = TIMER1 jest źródłem sygnału zegarowego dla compare/capture modułów CCP,
- bit 5-4    **T3CKPS1: T3CKPS0: wybór wartości preskalera:**  
11 = 1:8  
10 = 1:4  
01 = 1:2  
00 = 1:1
- bit 3      **T3CCP1: załączenie oscylatora dla TIMER1:**  
1 = oscylator włączony,  
0 = oscylator wyłączony (inwerter oscylatora jest wyłączony, aby wyeliminować pobór mocy).
- bit 2      **~T3SYNC: bit synchronizacji Timer3 z zegarem zewnętrznym:**  
(bit nie jest używany gdy sygnał zegarowy pochodzi od TIMER1/TIMER3)  
gdy TMR3CS=1:  
1 = nie synchronizuje zewnętrznego wejścia zegarowego,  
0 = synchronizuje zewnętrzne wejście zegarowe,  
gdy TMR3CS=0:  
Bit jest ignorowany. Timer3 używa zegara wewnętrznego, gdy TRM3CS=0.
- bit 1      **TMR3CS: wybór zegara dla TIMER3:**  
1 = zewnętrzne źródło zegara od TIMER1 lub T1CKI  
      (na zboczu narastającym zaraz po pierwszym zboczu opadającym),  
0 = zegar wewnętrzny (Fosc/4),
- bit 0      **TMR3ON: bit włączenia TIMER3:**  
1 = TIMER3 włączony,  
0 = TIMER3 zatrzymany,

## 9 Moduły CAPTURE/COMPARE/PWM

Obydwa moduły CCP1 oraz CCP2 wykonują te same operacje, z wyjątkiem obsługi zdarzenia związanego z układem wywołania specjalnego zdarzenia (*special event trigger*), dlatego w dalszej części opisany będzie tylko moduł CCP1.

Rejestr **CCPR1** Capture/Compare/PWM zawiera dwa 8 bitowe rejestry: **CCPR1L** (młodszy bajt) i **CCPR1H** (starszy bajt). Bity związane z konfiguracją modułu CCP1 znajdują się w rejestrze **CCP1CON**.

Analogicznie dla modułu CCP2, rejestr **CCPR2** Capture/Compare/PWM zawiera dwa 8 bitowe rejestry: **CCPR2L** (młodszy bajt) i **CCPR2H** (starszy bajt). Bity związane z konfiguracją modułu CCP2 znajdują się w rejestrze **CCP2CON**

Tryb CCP	Źródło Timera
Capture	Timer1 lub Timer3
Compare	Timer1 lub Timer3
PWM	Timer2

Tabela D8.1. Źródła sygnałów doprowadzonych do modułu CCP

Tryb CCPx	Tryb CCPy	Oddziaływanie
Capture	Capture	TMR1 lub TMR3, podstawa czasu może być różna dla każdego z modułów CCP,
Capture	Compare	Tryb porównania może być skonfigurowany dla wywołania specjalnego zdarzenia, które wyzeruje TMR1 lub TMR3 zależnie od tego, który licznik jest używany jako podstawa czasu.
Compare	Compare	Tryby porównania mogą być skonfigurowane dla wywołania specjalnego zdarzenia, które wyzeruje TMR1 lub TMR3 zależnie od tego, który licznik jest używany jako podstawa czasu.
PWM	PWM	Moduły PWM będą miały taką samą częstotliwość i wartość (przerwanie TMR2)
PWM	Capture	Brak
PWM	Compare	Brak

Tabela D8.2. Zależności pomiędzy dwoma modułami CCP

## Rejestry CCP1CON/CCP2CON

(Rejestry *CCP1CON/CCP2CON* stosowane są do kontroli pracy modułów CCP1 i CCP2)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bit 7-6      **bity niezaimplementowane**

bit 5-4      **DCxB1: DCxB0: współczynnik wypełnienia PWM bit1 i bit0:**

tryb Compare:

nieużywany

tryb Capture:

nieużywany

tryb PWM:

bity te są dwoma najmniej znaczącymi bitami (bit1 i bit0) 10-bitowego współczynnika wypełnienia PWM. Starsze 8-bitów (**DCx9: DCx2**) współczynnika wypełnienia znajduje się w rejestrze **CCPRxL**.

bit 3-0      **CCPxM3: CCPxM0: wybór trybu pracy CCPx:**

0000 = wyłączenie modułu CCP (RESET modułów CCPx),

0001 = zarezerwowany,

0010 = tryb Compare, przełączenie wyjścia do porównania (ustawiana flaga CCPxIF),

0011 = zarezerwowany,

0100 = tryb Capture, wpis na zboczu opadającym,

0101 = tryb Capture, wpis na zboczu narastającym,

0110 = tryb Capture, wpis na co 4 zbocze narastające,

0111 = tryb Capture, wpis na co 16 zbocze narastające,

1000 = tryb Compare,

po inicjalizacji pin modułu CCP jest w stanie LOW, zrównanie ustawia pin modułu CCP w stan HIGH (flaga CCPIF ustawiona),

1001 = tryb Compare,

po inicjalizacji pin modułu CCP jest w stanie HIGH, zrównanie ustawia pin modułu CCP w stan LOW (flaga CCPIF ustawiona),

1010 = tryb Compare,

zrównanie generuje programowe przerwanie (ustawiona flaga CCPIF, pin CCP niezmienny),

1011 = tryb Compare, wywołanie specjalnego zdarzenia (flaga CCPIF ustawiona)

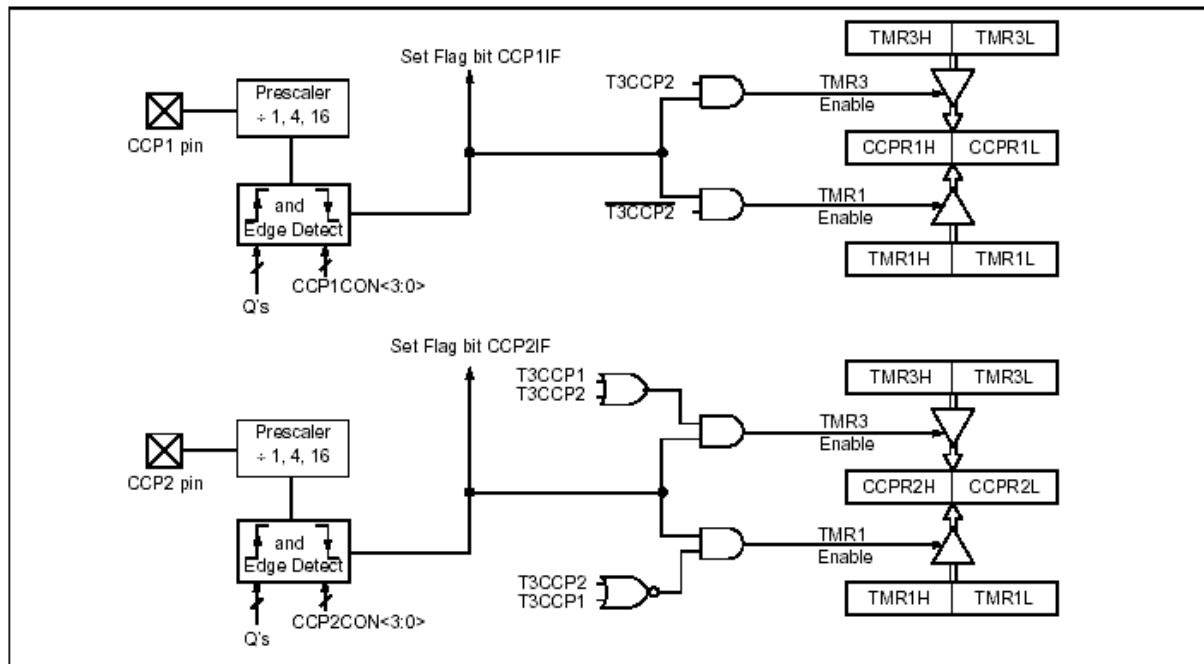
11xx = tryb PWM

## Tryb CAPTURE

W trybie Capture, rejestry **CCPR1H:CCPR1L** przechwytyją 16-bitową wartość z rejestru **TMR1** lub **TMR3** w momencie, gdy na pinie RC2/CCP1 wystąpi jakieś zdarzenie.

Może to być jedno z następujących zdarzeń:

- ⇒ każde opadające zbocze,
- ⇒ każde narastające zbocze,
- ⇒ każde co 4 narastające zbocze,
- ⇒ każde co 16 narastające zbocze.



Rys. 8.1. Schemat blokowy układu CCP w trybie CAPTURE

Typ zdarzenia wybierany jest przez bity kontrolne CCP1M3:CCP1M0 rejestru **CCP1CON<3:0>**. Kiedy nastąpi przechwycenie, ustawiony zostanie bit flagi przerwania CCP1IF rejestru **PIR1<2>**. Flaga przerwania musi być zerowana programowo. Jeżeli kolejne przechwycenie wystąpi przed odczytem wartości z rejestru **CCPR1**, poprzednio przechwycona wartość jest kasowana przez wpis nowej wartości.

W trybie przechwytywania (Capture) pin RC2/CCP1 powinien być skonfigurowany jako wejściowy przez ustawienie bitu w rejestrze kierunkowym **TRISC<2>**.

Jeżeli pin RC2/CCP1 jest skonfigurowany jako wyjściowy, zapis do portu może być warunkiem przechwytywania.

Liczniki używane w trybie Capture (TIMER1 i/lub TIMER3) muszą działać w trybie zegara, lub licznika synchronicznego. W trybie licznika asynchronicznego operacje Capture mogą nie działać. Wyboru licznika dla modułu CCP dokonuje się w rejestrze **T3CON**.

Kiedy tryb przechwytywania jest zmieniany, mogą być generowane fałszywe przerwania. Użytkownik powinien wyzerować bit zezwolenia na przerwanie z układu CCP1 - CCP1IE rejestru **PIE1<2>**, aby uniknąć fałszywych przerwań. Powinien również wyzerować bit (flagę przerwania) CCP1IF przed jakąkolwiek zmianą trybu pracy.

Cztery ustawienia preskalera są konfigurowane przez bity CCP1M3:CCP1M0. Ilekroć moduł CCP jest wyłączany, lub moduł ten nie jest w trybie przechwytywania, wówczas licznik preskalera jest wyzerowany. Jakikolwiek sygnał RESET zeruje licznik preskalera.

Przełączanie z jednego ustawienia preskalera do innego może generować przerwanie. Również licznik preskalera nie zostanie wyzerowany, dlatego pierwsza operacja przechwycenia może być z niezerowym licznikiem preskalera.

Poniższy przykład pokazuje zalecaną metodę przełączania pomiędzy wartościami preskalera. Przykład ten zeruje także licznik preskalera, aby nie generował on "fałszywych" przerw.

```
CLRF    CCP1CON,F           ; wyłącz moduły CCP
MOVLW  NEW_CAPT_PS         ; załadowanie rejestru
                               ; W nową wartość
                               ; preskalera i
                               ; włączenie modułu CCP
MOVWF  CCP1CON             ; załadowanie CCP1CON
                               ; przez tą wartość
```

*Listing D8.1. Sposób zmiany wartości preskalera*

## Tryb COMPARE

W trybie Compare 16-bitowa wartość rejestru **CCPR1** (**CCPR2**) jest stale porównywana z wartością rejestru **TMR1** lub rejestru **TMR3**. Kiedy wystąpi zrównanie tych wartości, pin RC2/CCP1 (RC2/CCP1) jest ustawiany:

- ⇒ w stan wysoki,
- ⇒ w stan niski,
- ⇒ zmienia stan wyjścia (z wysokiego na niski lub z niskiego na wysoki),
- ⇒ pozostaje bez zmian.

Tryb pracy tego pinu jest ustawiany za pomocą bitów kontrolnych CCP1M3:CCP1M0 (CCP2M3:CCP2M0), w tym samym czasie ustawiana jest flaga przerwania CCP1IF (CCP2IF).

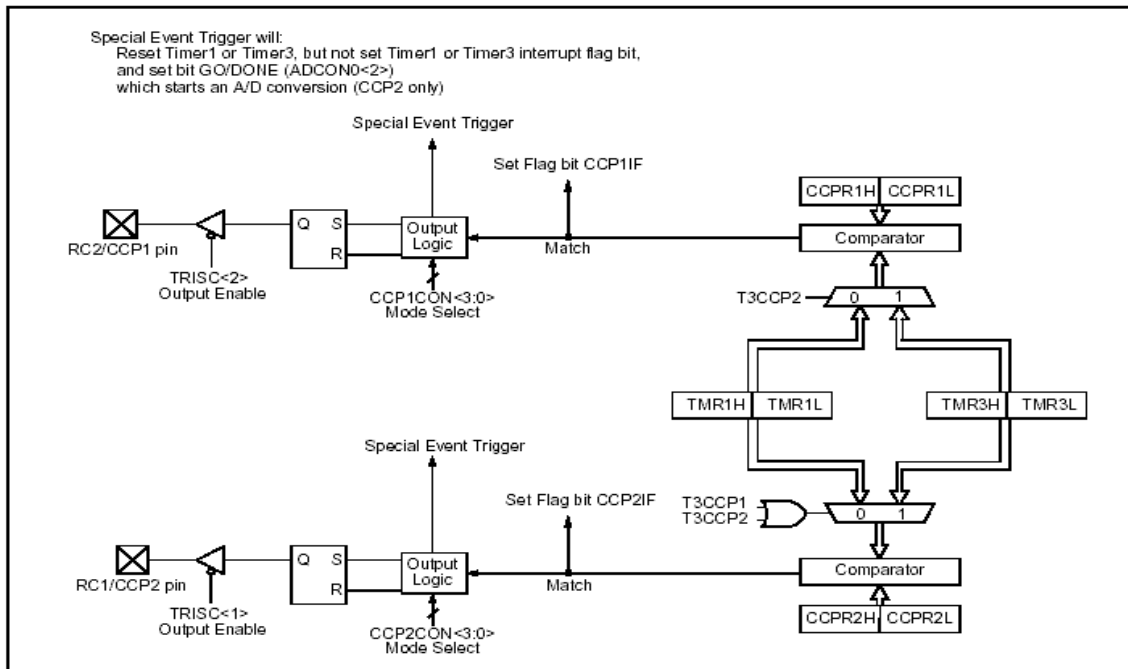
Użytkownik musi skonfigurować pin modułu CCPx jako wyjściowy poprzez wyzerowanie odpowiednich bitów w rejestrze kierunkowym **TRISC**.

Aby możliwa była praca modułu CPP w trybie porównania, **TIMER1** i/lub **TIMER3** muszą być uruchomione w trybie zegara lub w trybie licznika synchronicznego. W trybie asynchronicznego licznika operacja porównania może nie funkcjonować.

Kiedy wybrany jest tryb generowania przerwania programowo, pin CCP1 jest nie aktywny. Bit CCPIF jest ustawiony w skutek pojawienia się przerwania od CCP (jeżeli jest ono dostępne).

W trybie wywołania specjalnego zdarzenia (*Special Event Trigger*) generowane jest wewnętrzne wyzwolenie, które może zostać wykorzystane do zapoczątkowania jakiejś akcji.

Wyjście układu specjalnych zdarzeń układu CCP1 kasuje zawartość rejestrów **TMR1**. Pozwala to na efektywne wykorzystanie rejestru **CCPR1** jako 16-bitowego programowalnego rejestru okresu dla **Timer1**.



Rys. 8.2. Schemat blokowy układu CCP w trybie COMPARE

Wyjście układu specjalnych zdarzeń modułów CCPx kasuje zarówno rejestry **TMR1** jak i rejestry **TMR3**. Dodatkowo jeżeli moduł A/C jest odblokowany, zdarzenie specjalne od modułu CCP2 rozpoczyna konwersję A/C.

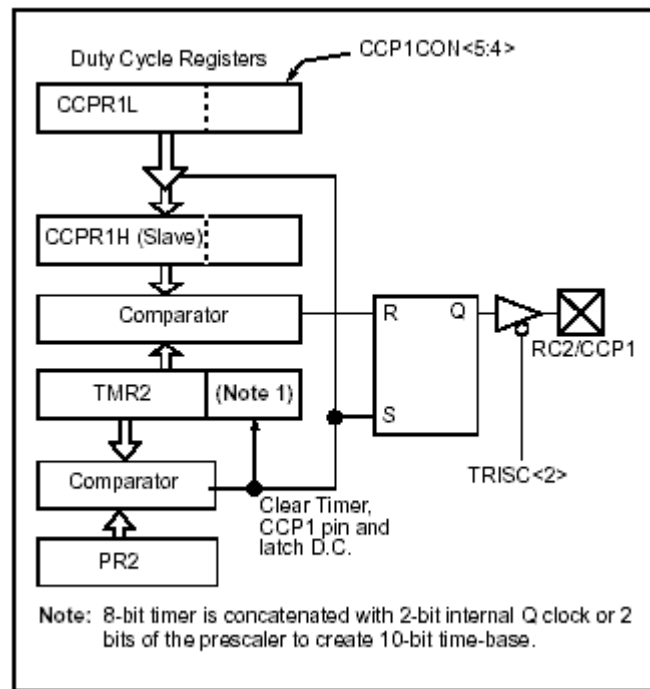
Zdarzenie specjalne od modułu CCP2 nie ustawia flag przerwania do TIMER1 i TIMER3

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>INTCON</b>	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>TRISC</b>	Rejestr kierunku danych PORTC								1111 1111	1111 1111
<b>TMR1L</b>	Rejestr młodszego bajtu licznika TIMER1								xxxx xxxx	uuuu uuuu
<b>TMR1H</b>	Rejestr starszego bajtu licznika TIMER1								xxxx xxxx	uuuu uuuu
<b>T1CON</b>	RD16	-	T1CKP S1	T1CKP S0	T1OSC EN	~T1 SYNC	TMR1 CS	TMR1ON	0-00 0000	u-uu uuuu
<b>CCPR1L</b>	Rejestr Capture/Compare/PWM (LSB)								xxxx xxxx	uuuu uuuu
<b>CCPR1H</b>	Rejestr Capture/Compare/PWM (MSB)								xxxx xxxx	uuuu uuuu
<b>CCP1CON</b>	-	-	DC1B1	DC1B0	CCP1 M3	CCP1 M2	CCP1 M1	CCP1 M0	- -00 0000	- -00 0000
<b>CCPR2L</b>	Rejestr Capture/Compare/PWM (LSB)								xxxx xxxx	uuuu uuuu
<b>CCPR2H</b>	Rejestr Capture/Compare/PWM (MSB)								xxxx xxxx	uuuu uuuu
<b>CCP2CON</b>	-	-	DC2B1	DC2B0	CCP2 M3	CCP2 M2	CCP2 M1	CCP2 M0	- -00 0000	- -00 0000
<b>PIR2</b>	-	-	-	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	-- -0 0000	-- -0 0000
<b>PIE2</b>	-	-	-	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	-- -0 0000	-- -0 0000
<b>IPR2</b>	-	-	-	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	-- -1 1111	-- -1 1111
<b>TMR3L</b>	Rejestr młodszego bajtu licznika TIMER3								xxxx xxxx	uuuu uuuu
<b>TMR3H</b>	Rejestr starszego bajtu licznika TIMER3								xxxx xxxx	uuuu uuuu
<b>T3CON</b>	RD16	T3CCP2	T3CKP S1	T3CKP S0	T3CCP1	~T3 SYNC	TMR3 CS	TMR3ON	0000 0000	uuuu uuuu

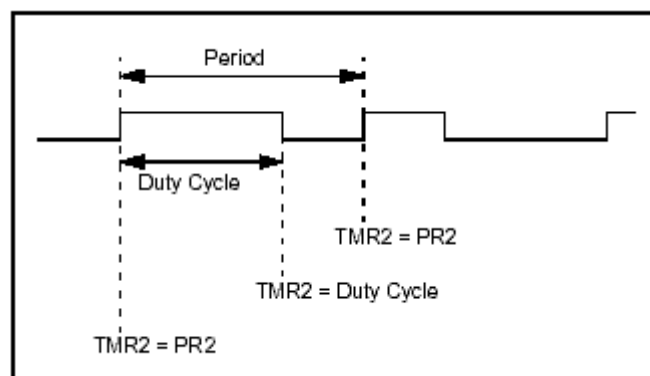
Tabela D8.3. Rejestry związane z trybami CAPTURE/COMPARE i licznikami

## Tryb PWM

W trybie modulacji szerokości impulsów pin CCPx jest źródłem sygnału PWM o 10-bitowej rozdzielczości. Ponieważ pin CCP1 jest multipleksowany z zatrzaskami danych PORTC, bit **TRISC<2>** musi być wyzerowany, aby móc ustawić pin CCP1 jako wyjście.



Rys. 8.3. Schemat blokowy układu CCP w trybie PWM



Rys. 8.4. Wyjście układu PWM

## Okres PWM



Okres PWM jest określony poprzez wpis do rejestru PR2. Okres PWM można obliczyć używając następującego wzoru:

$$\text{Okres PWM} = [(PR2) + 1] * 4 * T_{osc} * (\text{wartość preskalera TMR2})$$

Częstotliwość PWM określona jest jako  $1/[\text{okres PWM}]$ .

Kiedy TMR2 jest równy PR2, to w czasie następnego cyklu mają miejsce trzy następujące zdarzenia:

- TMR2 jest wyzerowany,
- pin CCP1 jest ustawiony (za wyjątkiem, gdy wypełnienie PWM jest 0% ),
- wypełnienie PWM jest przepisane z CCPR1L do CCPR1H.

### Wypełnienie PWM

Wypełnienie PWM jest określone przez wpis do rejestru CCPR1L i bitów CCP1CON<5:4>. Maksymalna rozdzielczość wynosi 10 bitów. CCPR1L zawiera osiem najbardziej znaczących bitów (MSB), a CCP1CON<5:4> zawiera dwa najmniej znaczące bity (LSB). Dziesięciobitowa wartość jest reprezentowana przez CCPR1L:CCP1CON<5:4>. Do określenia wypełnienia należy użyć następującego równania:

$$\text{Wypełnienie PWM} = (\text{CCPR1L:CCP1CON<5:4>}) * T_{osc} * (\text{wartość preskalera TMR2})$$

Wartość wypełnienia może być wpisana do rejestrów CCPR1L:CCP1CON<5:4> w dowolnym momencie. Zostanie ona przepisana do CCPR1H po zrównaniu się wartości PR2 i TMR2. W trybie PWM, CCPR1H jest rejestrem tylko do odczytu.

Maksymalna rozdzielczość dla danej częstotliwości PWM jest określona równaniem :

$$\text{Rozdzielczość} = \frac{\log\left(\frac{F_{osc}}{F_{pwm}}\right)}{\log(2)} \text{ bitów}$$

Jeśli wypełnienie (*duty cycle*) PWM jest dłuższe niż okres (*period*) PWM, to pin CCP1 nie będzie wyzerowany.

Aby skonfigurować moduł PWM należy wykonać następujące operacje:

1. Ustawić okres PWM przez zapis do rejestru PR2.
2. Ustawić wypełnienie przez zapis do rejestru CCPR1L i bitów CCP1CON<5:4>.
3. Skonfigurować pin CCP1 jako wyjście przez wyzerowanie bitu TRISC<2>.
4. Ustawić wartości preskalera i uaktywnić Timer 2 przez zapis do T2CON.
5. Skonfigurować moduł CCP1 do pracy w trybie PWM.

Częstotliwość PWM	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
prescaler Timera (1, 4, 16)	16	4	1	1	1	1
wartość PR2	0xFFh	0xFFh	0xFFh	0x3Fh	0x1Fh	0x17h
max. rozdzielczość (w bitach)	14	12	10	8	7	6.25

Tabela D8.4. Przykładowe częstotliwości PWM i rozdzielczości przy 40MHz

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>INTCON</b>	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>TRISC</b>	Rejestr kierunku danych PORTC								1111 1111	1111 1111
<b>TMR2</b>	Rejestr młodszego bajtu licznika TIMER1								0000 0000	0000 0000
<b>PR2</b>	Rejestr starszego bajtu licznika TIMER1								1111 1111	1111 1111
<b>T2CON</b>	-	TOUT PS3	TOUT PS2	TOUT PS1	TOUT PS0	TMR2 ON	T2CK PS1	T2CK PS0	-000 0000	-000 0000
<b>CCPR1L</b>	Rejestr Capture/Compare/PWM (LSB)								xxxx xxxx	uuuu uuuu
<b>CCPR1H</b>	Rejestr Capture/Compare/PWM (MSB)								xxxx xxxx	uuuu uuuu
<b>CCP1CON</b>	-	-	DC1B1	DC1B0	CCP1 M3	CCP1 M2	CCP1 M1	CCP1 M0	- -00 0000	- -00 0000
<b>CCPR2L</b>	Rejestr Capture/Compare/PWM (LSB)								xxxx xxxx	uuuu uuuu
<b>CCPR2H</b>	Rejestr Capture/Compare/PWM (MSB)								xxxx xxxx	uuuu uuuu
<b>CCP2CON</b>	-	-	DC2B1	DC2B0	CCP2 M3	CCP2 M2	CCP2 M1	CCP2 M0	- -00 0000	- -00 0000

Tabela D8.5. Rejestry związane z modułem PWM i TIMER2

## 10 Moduł synchronicznego portu szeregowego (MSSP)

( *Master Synchronous Serial Port* )

Moduł synchroniczny portu szeregowego (Master Synchronous Serial Port – MSSP) jest szeregowym interfejsem używanym do komunikacji z układami peryferyjnymi lub innymi mikrokontrolerami. Tymi urządzeniami mogą być szeregowy pamięci EEPROM, rejestry przesuwne, sterowniki wyświetlaczy, przetworniki A/D i inne. Moduł MSSP może pracować w jednym z dwóch trybów:

- ⇒ szeregowy interfejs SPI (Serial Peripheral Interface),
- ⇒ szeregowy interfejs I<sup>2</sup>C (Inter-Integrated Circuit).

Interfejs I<sup>2</sup>C sprzętowo obsługuje następujące tryby pracy:

- ⇒ tryb nadrzędny (*master mode*),
- ⇒ tryb (*multi-master mode*),
- ⇒ tryb podrzędny (*slave mode*),

Do obsługi modułu MSSP stosowane są trzy rejestry, rejestry statusu **SSPSTAT** i dwa rejestry kontrolne **SSPCON1** i **SSPCON2**. Rejestry te znajdują przeznaczenia zależnie od trybu pracy modułu MSSP (SPI czy I<sup>2</sup>C).

## Rejestr SSPSTAT (Status register – SPI Mode)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
<b>SMP</b>	<b>CKE</b>	<b>D/~A</b>	<b>P</b>	<b>S</b>	<b>R/~W</b>	<b>UA</b>	<b>BF</b>
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7     **SMP: wybór momentu próbkowania danych wejściowych:**  
Tryb SPI Master:  
 1 = dana wejściowa jest próbkowana na końcu czasu wyjścia danych,  
 0 = dana wejściowa jest próbkowana w środku czasu wyjścia danych.  
Tryb SPI Slave:  
 Bit SPM musi być wyzerowany, kiedy SPI jest używany w trybie Slave
- bit 6     **CKE: wybór zbocza sygnału zegarowego dla transmisji danych:**  
CKP=0:  
 1 = dana transmitowana przy narastającym zboczku SCK,  
 0 = dana transmitowana przy opadającym zboczku SCK.  
CKP=1:  
 1 = dana transmitowana przy opadającym zboczku SCK,  
 0 = dana transmitowana przy narastającym zboczku SCK.
- bit 5     **D/~A: dane/zanegowany adres ():**  
 używany tylko w trybie I<sup>2</sup>C
- bit 4     **P: bit STOP:**  
 używany tylko w trybie I<sup>2</sup>C, bit ten jest zerowany, kiedy moduł MSSP jest wyłączony, SSPEN jest zerowany,
- bit 3     **S: bit START:**  
 używany tylko w trybie I<sup>2</sup>C,
- bit 2     **R/~W: zapis/odczyt informacji:**  
 używany tylko w trybie I<sup>2</sup>C,
- bit 1     **UA: wskaźnik aktualizacji adresu:**  
 używany tylko w trybie I<sup>2</sup>C,
- bit 0     **BF: flaga zajętości bufora SSPBUF(tylko w trybie odbioru):**  
 1 = odbiór kompletny, SSPBUF jest pełny,  
 0 = odbiór nie jest kompletny, SSPBUF jest pusty.

## Rejestr SSPCON1 (Control Register1 – SPI Mode)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

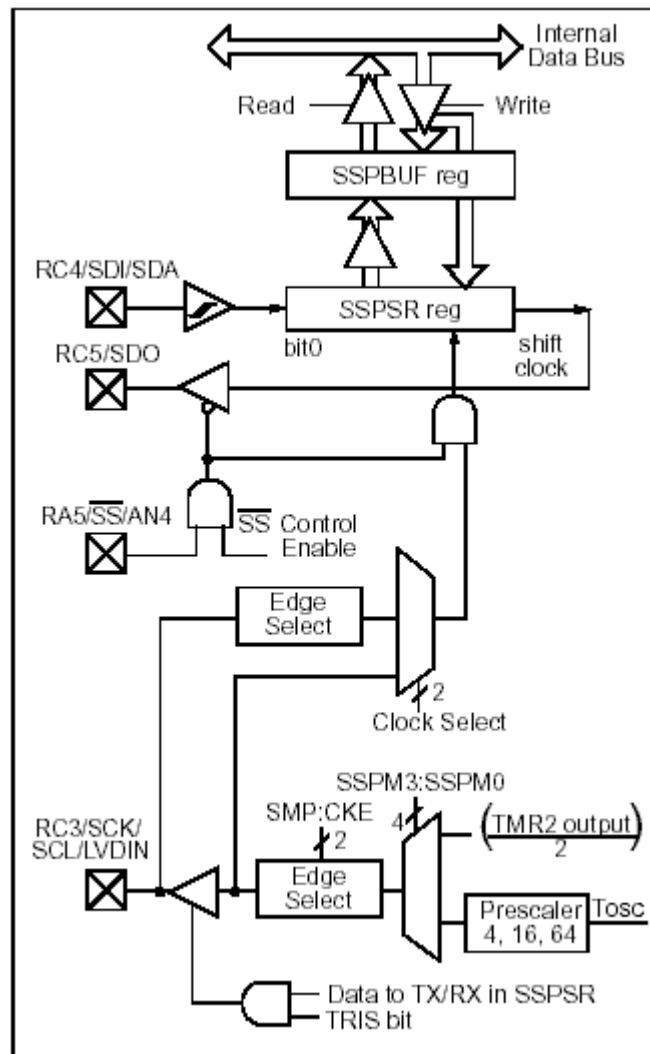
- bit 7      **WCOL: detekcja kolizji (tylko w trybie nadawania):**  
 1 = rejestr SSPBUF jest zapisywany podczas trwania transmisji poprzedniego słowa (musi być wyzerowany programowo),  
 0 = brak kolizji.
- bit 6      **SSPPOV: przepełnienie rejestru przy odbiorze:**  
tryb Slave SPI  
 1 = nowy bajt jest odbierany, podczas gdy rejestr SSPBUF nadal przechowuje poprzednią daną. W razie przepełnienia dana w SSPSR jest tracona. Przepełnienie może wystąpić jedynie w trybie Slave. Aby uniknąć przepełnienia użytkownik musi odczytać SSPBUF, nawet gdy tylko transmituje daną. (musi być zerowany programowo),  
 0 = brak przepełnienia,
- bit 5      **SSPEN: uaktywnienie modułu SSP:**  
 1 = uaktywnienie portu szeregowego i skonfigurowanie SCK, SDO, SDI i ~SS jako pinów portu szeregowego,  
 0 = zablokowanie portu szeregowego i skonfigurowanie tych pinów jako pinów portu we/wy.  
*Aktywne piny muszą być skonfigurowane odpowiednio jako wejścia lub wyjścia*
- bit 4      **CKP: bit polaryzacji zegara:**  
 1 = stan czuwania dla zegara jest poziomem wysokim,  
 0 = stan czuwania dla zegara jest poziomem niskim,
- bit 3-0    **SSPM3: SSPM0: wybór trybu pracy SSP:**  
 0101 = tryb SPI Slave, zegar = pin SCK, pin kontrolny ~SS nie aktywny, pin ~SS może być użyty jako pin we/wy,  
 0100 = tryb SPI Slave, zegar = pin SCK, pin kontrolny ~SS aktywny,  
 0011 = tryb SPI Master, zegar = wyjście TMR2/2,  
 0010 = tryb SPI Master, zegar = Fosc/64,  
 0001 = tryb SPI Master, zegar = Fosc/16,  
 0000 = tryb SPI Master, zegar = Fosc/4,  
*(bity które nie są wyszczególnione albo są zarezerwowane, albo używane w trybie I<sup>2</sup>C)*

## Tryb SPI

Tryb SPI pozwala synchronicznie szeregowo wysyłać i jednocześnie odbierać 8 bitów danych. Dostępne są wszystkie cztery tryby SPI.

Do komunikacji typowo używa się trzech linii:

- ⇒ SDO (*Serial Data Out*) – RC5/SDO,
- ⇒ SDI (*Serial Data In*) – RC4/SDI/SDA,
- ⇒ SCK (*Serial Clock*) – RC3/SCK/SCL/LVDIN,
- ⇒  $\sim$ SS (*Slave Select*) – RA5/ $\sim$ SS/AN4 – dodatkowy pin używany w trybie Slave,



Rys. 9.1. Schemat blokowy MSSP (tryb SPI)

Aby skonfigurować interfejs SPI należy określić kilka opcji. Dokonuje się tego przez zaprogramowanie odpowiednich bitów kontrolnych rejestru **SSPCON1<5:0>** oraz bitów statusu w rejestrze **SSPSTAT<7:6>** Bity te umożliwiają określenie:

- ⇒ trybu Master (SCK jest wyjściem zegarowym),
- ⇒ trybu Slave (SCK jest wejściem zegarowym),
- ⇒ polaryzację sygnału zegarowego,
- ⇒ próbkowanie danych wejściowych(pośrodku, lub na końcu danych wyjściowych),
- ⇒ wyboru zbocza sygnału zegarowego(dane wyjściowe na narastające/opadające zbocze sygnału SCK),
- ⇒ zakres pracy sygnału zegarowego (tylko w trybie Master),
- ⇒ tryb wyboru układu Slave ( tylko w trybie Slave),

Układ MSSP zawiera rejestr przesuwany nadawania/odbioru **SSPSR** i rejestr będący buforem **SSPBUF**. Rejestr **SSPSR** przesuwa dane do/z urządzenia począwszy od najbardziej znaczącego bitu (MSb). Bufor **SSPBUF** natomiast zawiera dane, które zostały zapisane w rejestrze **SSPSR**. Otrzymana 8-bitowa dana jest automatycznie przepisywana do rejestru **SSPBUF**. Kiedy bufor jest pełen, ustawiany jest bit BF w rejestrze **SSPSTAT<0>**, jednocześnie ustawiana jest flaga przerwania SSPIF. Podwójne buforowanie danych wejściowych pozwala na przyjęcie następnego bajtu przed odczytem danych, które właśnie zostały odebrane. Jakikolwiek zapis do rejestru **SSPBUF** podczas nadawania/odbioru będzie ignorowany, bit kolizji zapisu WCOL rejestru **SSPCON1<7>** zostanie ustawiony. Programista musi zapewnić wyzerowanie bitu WCOL.

Kiedy aplikacja oczekuje na odbiór ważnych danych, rejestr **SSPBUF** powinien być odczytany zanim następny bajt danych do nadania jest zapisany do **SSPBUF**. Bit przepełnienia BF rejestru **SSPSTAT<0>** wskazuje, kiedy rejestr **SSPBUF** został zapełniony danymi przychodzącymi. Kiedy rejestr **SSPBUF** jest odczytany bit BF jest zerowany. Przerwania układu MSSP używane są do określenia kiedy operacja nadawania/odbioru została zakończona. Rejestr **SSPBUF** musi być odczytany i/lub zapisany. Jeżeli przerwania nie są używane, wówczas do określenia czy wystąpiła kolizja zapisu stosuje się przeglądanie programowe. Poniższy przykład przedstawia zapisywanie do rejestru **SSPBUF(SSPSR)** danymi do nadania.

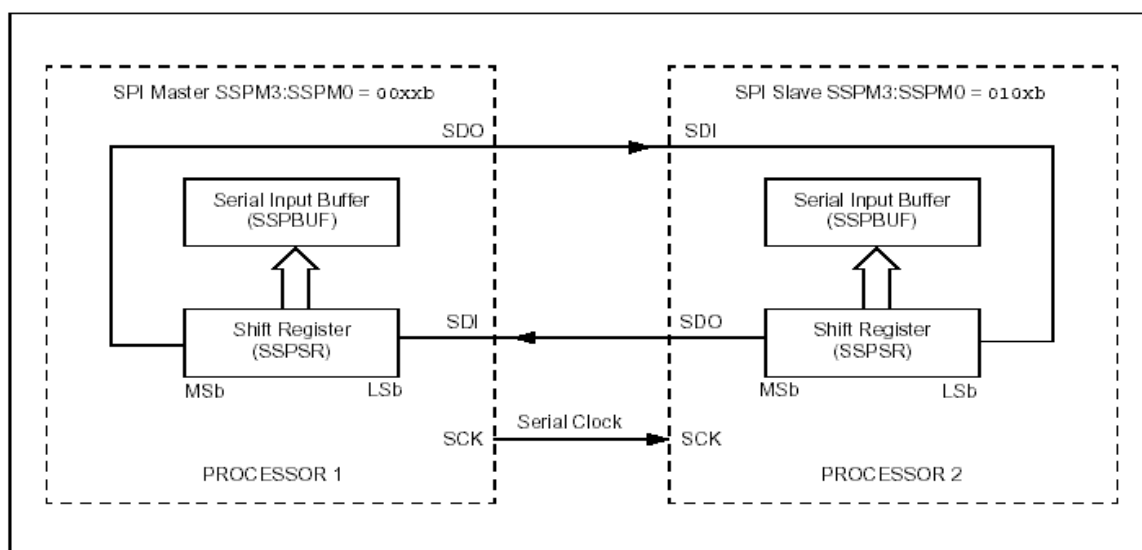
LOOP	BTFSS	SSPSTAT, BF	; czy odbiór/nadawanie
			; zostało zakończone?
	BRA	LOOP	; nie
	MOVF	SSPBUF, W	; rejestr WREG=rejestr SSPBUF
	MOVWF	RXDATA	; zapis w pamięci RAM -
			; jeżeli jest taka potrzeba
	MOVF	TXDATA, W	; rejestr WREG=rejestr TXDATA
	MOVWF	SSPBUF	; nowa dana do nadania

*Listing D9.1. Zapis danych do rejestru SSPBUF (SSPSR)*

By uaktywnić port szeregowy, należy ustawić bit aktywacji SSPEN w rejestrze **SSPCON<5>**. By zresetować albo przekonfigurować tryb SPI, należy wyzerować bit SSPEN, reinicjalizować rejestr **SSPCON** i ponownie ustawić bit SSPEN. Skonfiguruje to piny SDI, SDO, SCK i ~SS jako piny portu szeregowego. Aby piny zachowały funkcje portu szeregowego, muszą być odpowiednio skonfigurowane (w rejestrze kierunkowym **TRISC**).

Oznacza to:

- SDI jest automatycznie kontrolowany przez moduł SPI,
- SDO musi mieć **TRISC<5>** wyzerowany,
- SCK (w trybie Master) musi mieć **TRISC<3>** wyzerowany,
- SCK (w trybie Slave) musi mieć **TRISC<3>** ustawiony,
- ~SS musi mieć **TRISA<5>** ustawiony.



Rys. 9.2. Połączenie SPI Master/Slave

Rysunek D9.2 przedstawia typowe połączenie pomiędzy dwoma mikroprocesorami. Kontroler Master (*Processor1*) inicjalizuje transmisję danych przez wysyłanie sygnału SCK. Dane są przesuwane w obu rejestrach przesuwanych, na zaprogramowane przez użytkownika zbocze zegarowe i zatraskiwane na przeciwnym zboczku zegarowym. Oba procesory powinny pracować z taką samą polaryzacją sygnału zegarowego (CKP), wówczas oba kontrolery są w stanie wysyłać i odbierać dane w tym samym czasie.

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>INTCON</b>	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>TRISC</b>	Rejestr kierunku danych PORTC								1111 1111	1111 1111
<b>SSPBUF</b>	Bufor(rejestr) odbioru/nadawania danej modułu SSP								xxxx xxxx	uuuu uuuu
<b>SSPCON1</b>	WCOL	SSPPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
<b>TRISA</b>	Rejestr kierunku danych PORTA								-111 1111	-111 1111
<b>SSPSTAT</b>	SMP	CKE	D/-A	P	S	R/-W	UA	BF	0000 0000	0000 0000

Tabela. D9.1. Rejestry związane z interfejsem SPI

## Tryb Master układu SPI

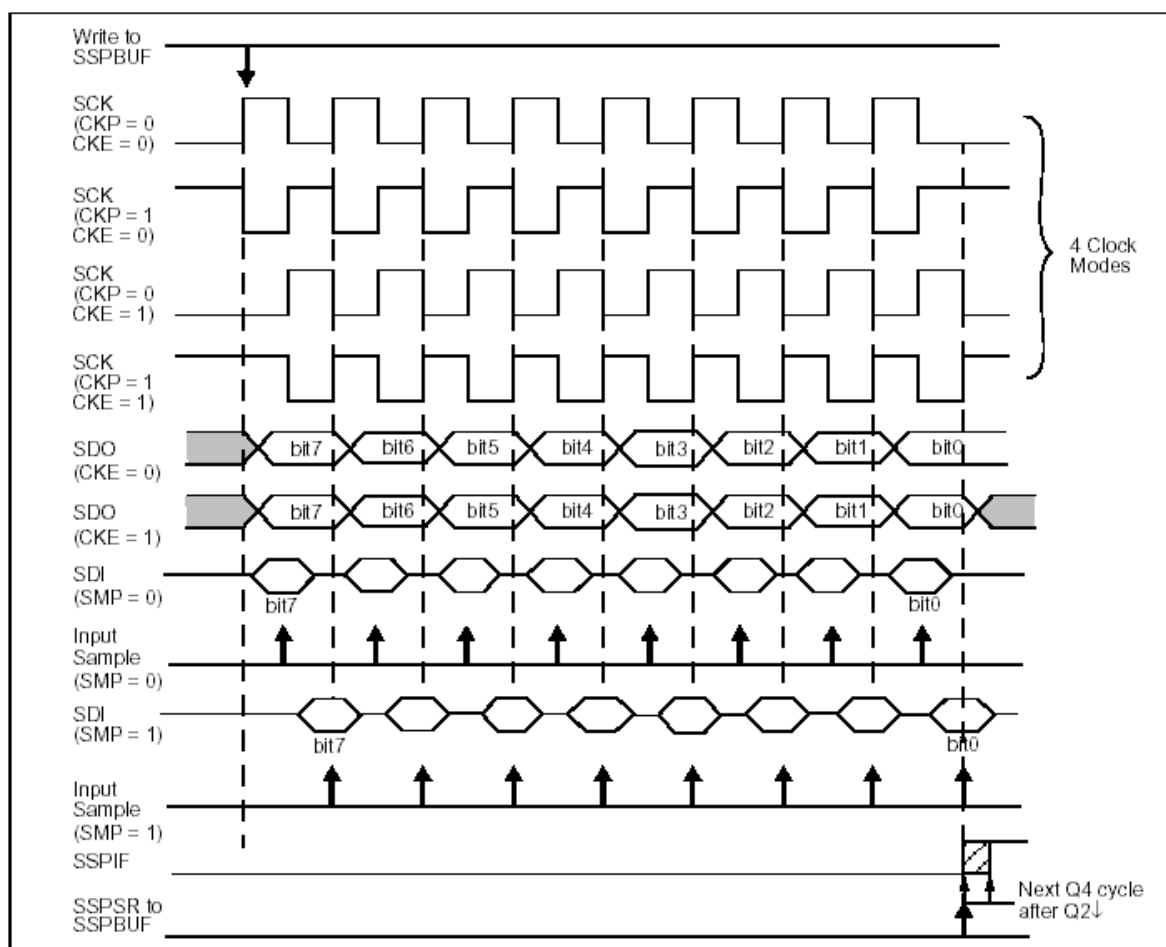
W tym trybie układ SPI może zainicjować transmisję w dowolnej chwili, ponieważ kontroluje on sygnał na linii SCK. Proces wysłania lub odbierania danej przez interfejs jest uruchamiany przez wprowadzenie jej do rejestru SSPBUF. Jeżeli interfejs SPI ma tylko odbierać dane, linia SDO może być nieaktywna (ustawiona jako wejście). Dana z rejestru SSPSR bit po bicie jest wystawiana na linii SDI w takt sygnału zegarowego. Każdy odebrany bajt jest umieszczany w rejestrze SSPBUF, ponadto ustawiane są odpowiednie bity statusu.

Polaryzacja sygnału zegarowego jest wybierana przez bit CKP (SSPCON<4>). W trybie Master podział sygnału zegarowego jest ustawiany przez użytkownika. Możliwe są następujące podziały sygnału zegarowego:

- FOSC/4 (lub TCY),
- FOSC/16 (lub 4 • TCY),
- FOSC/64 (lub 16 • TCY),
- Timer2 output/2.

Pozwala to na uzyskanie prędkości transmisji przy częstotliwości zegara równej 40MHz równej 10Mbps (*bits per second*).

Na rysunku poniżej pokazano przebiegi czasowe dla trybu Master. Kiedy CKE = 1, dane na pinie SDO są ważne przed pojawieniem się zbocza sygnału zegarowego na pinie SCK. Zmianę próbkowania sygnału zegarowego dokonuje się za pomocą bitu SMP.



Rys. 9.3. Przebiegi czasowe SPI w trybie Master

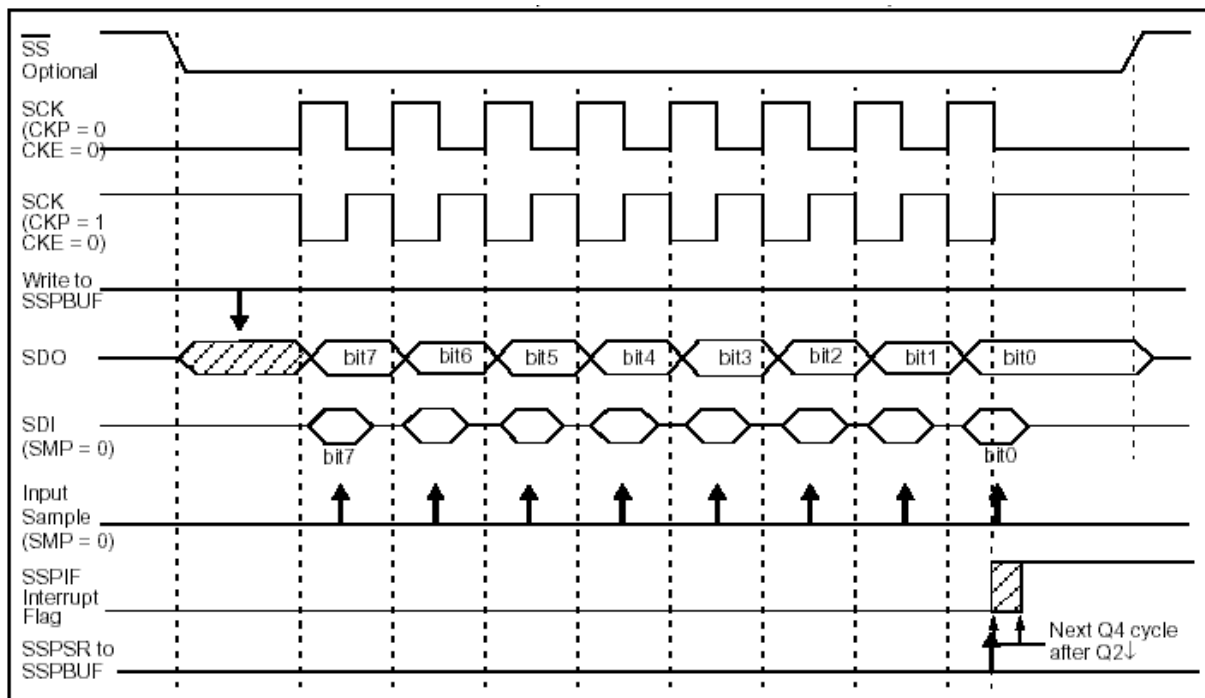


## Tryb Slave układu SPI

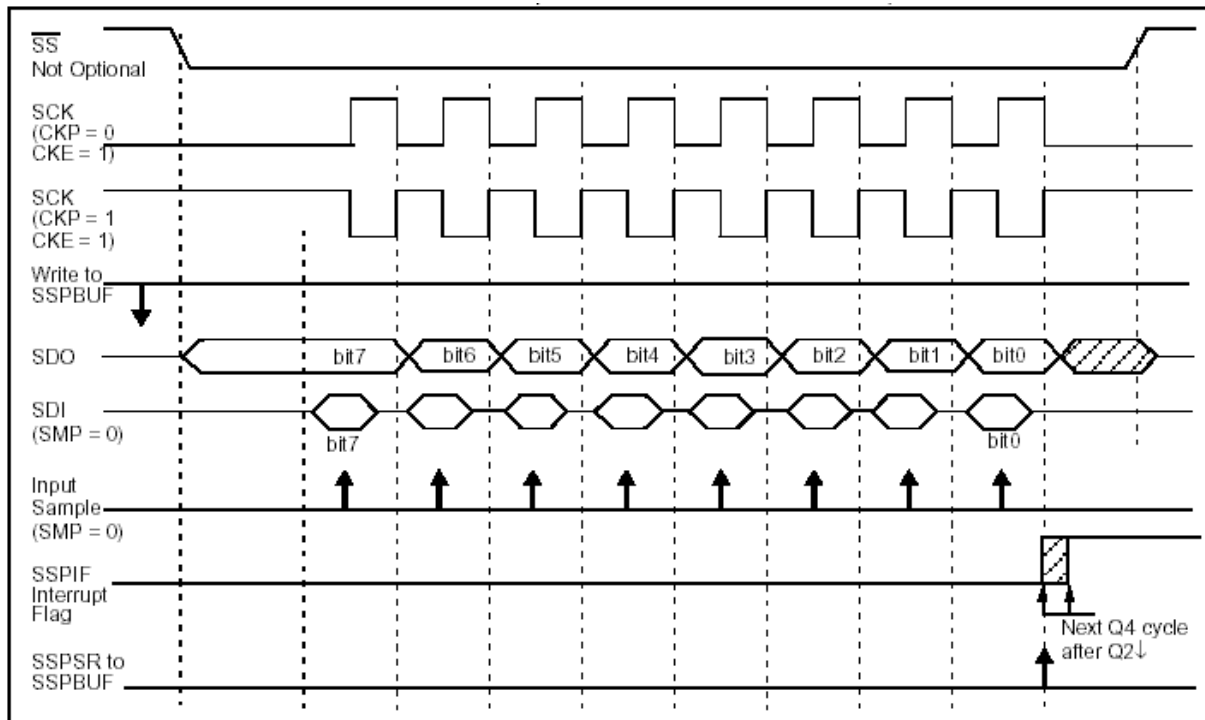
W trybie Slave dane są wysyłane i odbierane, gdy na wejściu SCK pojawiają się impulsy zegarowe generowane przez zewnętrzne urządzenie pracujące w trybie Master. Po wprowadzeniu ostatniego bitu danej rejestru **SSPSR**, całe odebrane słowo zatrzymywane jest w rejestrze **SSPBUF** i ustawiana jest flaga przerwania **SSPIF** w rejestrze **PIR1<3>**. Parametry sygnału zegarowego dostarczanego do układu SPI muszą spełniać odpowiednie parametry czasowe zgodnie ze specyfikacją mikrokontrolera. W stanie uśpienia mikrokontrolera interfejs SPI może nadawać/odbierać dane. Gdy bajt zostanie odebrany, mikrokontroler wychodzi ze stanu uśpienia.

Linia  $\sim$ SS umożliwia pracę mikrokontrolera w trybie Slave. Aby umożliwić pracę w tym trybie linia  $\sim$ SS musi być odblokowana **SSPCON1<3:0>=04h**. Gdy linia  $\sim$ SS jest w stanie niskim nadawanie i odbiór jest możliwy.

1. Kiedy moduł SPI pracuje w trybie Slave z ustawioną kontrolą za pomocą pinu  $\sim$ SS (**SSPCON<3:0> = 0100**), to zostaje on resetowany, jeśli sygnał na pinie SS przyjmie wartość VDD.
2. Jeśli układ SPI jest używany w trybie Slave z ustawionym bitem **CKE = '1'**, to kontrola za pomocą pinu  $\sim$ SS musi być włączona.



Rys. 9.4. Przebiegi czasowe SPI w trybie Slave, CKE = 0



Rys. 9.5. Przebiegi czasowe SPI w trybie Slave, CKE = 1

## 11 Interfejs USART

Moduł uniwersalnego synchronicznego-asynchronicznego odbiornika-nadajnika (USART) jest jednym z dwóch szeregowych modułów wejściowo-wyjściowych. USART jest także znany jako interfejs komunikacji szeregowej lub SCI (*Serial Communications Interface*). USART może być konfigurowany jako w pełni dwukierunkowy interfejs asynchroniczny, który może komunikować się z urządzeniami peryferyjnymi, takimi jak terminale CRT i komputery osobiste. Może on być również konfigurowany jako jednokierunkowy synchroniczny interfejs, służący do współpracy z urządzeniami peryferyjnymi takimi jak: przetworniki A/D i / D/A, szeregowy pamięci EEPROM, itd.

Układ USART może pracować w następujących trybach:

- ⇒ asynchronicznym (dwukierunkowym – full duplex),
- ⇒ synchronicznym – master (jednokierunkowym – half duplex),
- ⇒ synchronicznym – slave (jednokierunkowym – half duplex).

Aby linie RC6/TX/CK i RC7/RX/DT były skonfigurowane do obsługi układu USART należy ustawić:

- ⇒ bit SPEN (**RCSTA**<7>) = 1,
- ⇒ bit **TRISC**<6> = 0,
- ⇒ bit **TRISC**<7> = 1,

## Baud Rate Generator

Układ BRG wspomaga oba tryby pracy interfejsu USART – asynchroniczny i synchroniczny. Jest to dedykowany 8-bitowy generator szybkości transmisji. Rejestr **SPBRG** kontroluje okres swobodnej pracy 8-bitowego zegara. W trybie asynchronicznym bit BRGH (**TXTA<2>**) kontroluje również szybkość transmisji. W trybie synchronicznym bit BRGH jest ignorowany. Poniższa tabela pokazuje wzory potrzebne do obliczenia szybkości transmisji dla różnych trybów USART (tylko dla pracy jako Master).

SYNC	BGRH=0 (mała prędkość)	BRGH=1 (duża prędkość)
0	(Asynchroniczny) szybkość transmisji = $F_{osc}/(64(X+1))$	szybkość transmisji = $F_{osc}/(16(X+1))$
1	(Synchroniczny) szybkość transmisji = $F_{osc}/(4(X+1))$	nie oblicza się

*Tabela D10.1. Zależności określające prędkość transmisji*

Mając podaną żądaną szybkość transmisji i  $F_{osc}$ , najbliższa wartość całkowita dla rejestru SPBRG może być obliczona przy użyciu wzorów z powyższej tabeli.

Poniższy przykład obrazuje sposób obliczania błędu prędkości transmisji przy następujących warunkach:

- $F_{osc} = 16\text{MHz}$ ,
- Pożądana prędkość transmisji = 9600,
- BRGH = 0,
- SYNC = 0,

Pożądana prędkość transmisji	= $F_{osc} / (64(X+1))$
Rozwiązując względem X:	
X	= $((F_{osc} / \text{Pożądana prędkość transmisji}) / 64) - 1$
X	= $((16000000/9600) / 64) - 1$
X	= $[25.042] = 25$
Obliczona prędkość transmisji	= $16000000 / (64(25+1))$ = 9615
Błąd	= $(\text{Obliczona prędkość transmisji} - \text{Pożądana prędkość transmisji}) /$ $\text{Pożądana prędkość transmisji}$
	= $(9615 - 9600) / 9600$
	= 0.16%

*Przykład D10.1. Obliczanie błędu prędkości transmisji*

Korzystne może być użycie wysokiej szybkości transmisji (BRGH=1), nawet dla wolniejszych zegarów transmisji. Dzieje się tak dlatego, gdyż równanie  $F_{osc}/(16(X+1))$  może w niektórych przypadkach zredukować błąd transmisji.

Wpisanie nowej wartości do rejestru SPBRG powoduje zresetowanie timera BRG.

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>TXSTA</b>	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
<b>RCSTA</b>	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
<b>SPBRG</b>	Rejestr generatora szybkości transmisji								0000 0000	0000 0000

Tabela D10.2. Rejestry związane z generatorem BRG układu USART

## Transmisja Asynchroniczna

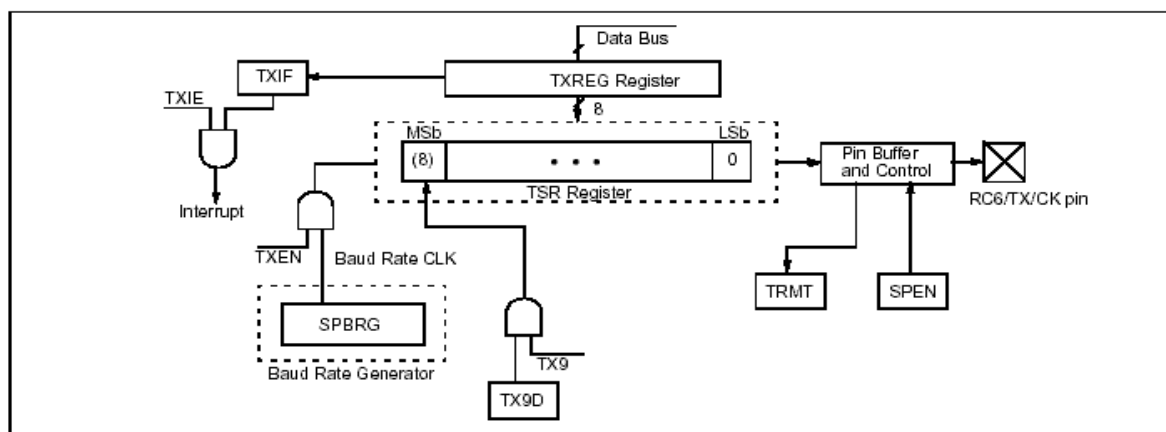
W tym trybie układ USART używa standardowego formatu bez powrotu do zera (NRZ) (jeden bit START, 8 lub 9 bitów danych i jeden bit STOP). Najbardziej popularnym formatem danych jest format 8-bitowy (8N1). Transmisja danych zaczyna się od bitu najmniej znaczącego - LSB. Nadajnik i odbiornik USART są funkcjonalnie niezależne, ale używają tego samego formatu danych i tej samej szybkości transmisji. Generator szybkości transmisji generuje zegar o szybkości x16 albo x64 przesunięcia transmisji bitów, zależnie od bitu BRGH (TXSTA <2>). Parzystość nie jest wspomagana sprzętowo, ale może być implementowana programowo (i zapisana jako 9-ty bit danych). Tryb asynchroniczny jest zatrzymany podczas trybu SLEEP mikrokontrolera. Tryb asynchroniczny jest wybierany przez wyczyszczenie bitu SYNC (TXSTA<4>).

W tym trybie układ USART składa się z elementów:

- Generators szybkości transmisji
- Układu próbkującego.
- Nadajnika asynchronicznego.
- Odbiornika asynchronicznego.

## Układ nadajnika transmisji asynchronicznej – USART

Schemat blokowy nadajnika USART pokazany jest poniżej. Sercem nadajnika jest rejestr przesuwany nadajnika (TSR). Rejestr przesuwany otrzymuje dane z bufora transmisji TXREG (można do niego zapisać i odczytać zawartość). Do rejestru TXREG dane są wprowadzane przez program użytkownika.



Rys. 10.1. Schemat blokowy układu nadajnika transmisji asynchronicznej

Rejestr **TSR** jest załadowany nową daną z rejestru **TXREG** pod warunkiem wykrycia bitu STOP w poprzedniej transmisji (zakończony nadawanie). Przepisanie danych z rejestru **TXREG** do **TSR** powoduje ustawienie flagi TXIF (**PIR1<4>**) sygnalizując opróżnienie rejestru TXREG. Przerwanie może być aktywne/nie aktywne przez ustawienie/wyzerowanie bitu TXIE (**PIE1<4>**). Flaga bitu TXIF będzie ustawiona nie zważając na stan bitu TXIE i nie może być wyzerowana programowo. Bit ten będzie wyzerowany dopiero wtedy, kiedy nowe dane będą załadowane do rejestru **TXREG**. Podczas gdy flaga TXIF wskazuje na status rejestru **TXREG**, inny bit TRMT (**TXSTA <1>**) pokazuje status rejestru **TSR**.

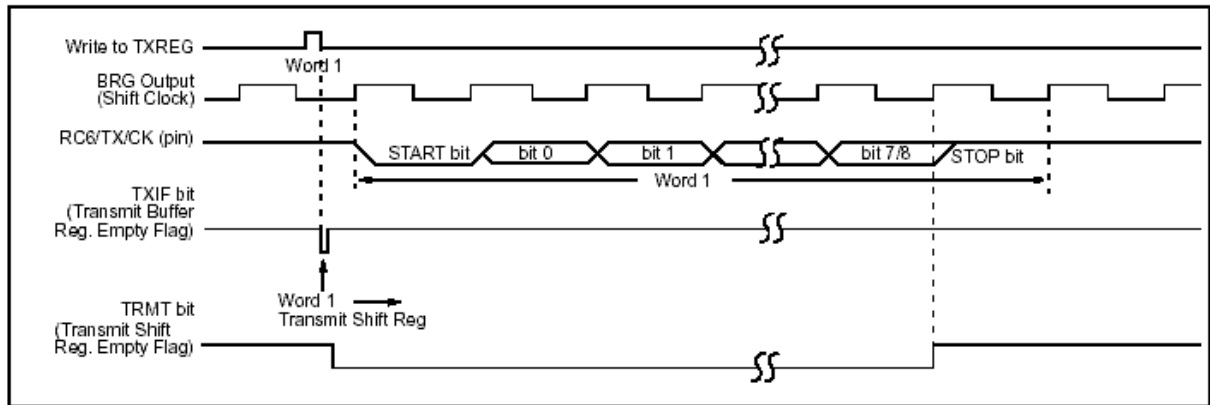
1. Rejestr **TSR** nie jest mapowany w pamięci danych, więc nie jest dostępny dla użytkownika.
2. Flaga TXIF jest ustawiona, kiedy ustawiony jest bit TXEN.

Układ USART jest włączany (uaktywniony) przez ustawienie bitu TXEN (**TXSTA<5>**). Transmisja rozpoczyna się, gdy do rejestru **TXREG** wprowadzimy daną. Oczywiście, generator BRG musi generować sygnał zegarowy o właściwej częstotliwości. Normalnie, gdy zaczyna się transmisja, rejestr **TSR** jest pusty. W tym momencie transfer do rejestru **TXREG** spowoduje natychmiastowy transfer do rejestru **TSR** i opróżnienie rejestru **TXREG**. Zerując bit TXEN podczas transmisji spowodujemy przerwanie transmisji i zresetowanie układu nadajnika. W rezultacie pin RC6/TX/CK przejdzie w stan wysokiej impedancji.

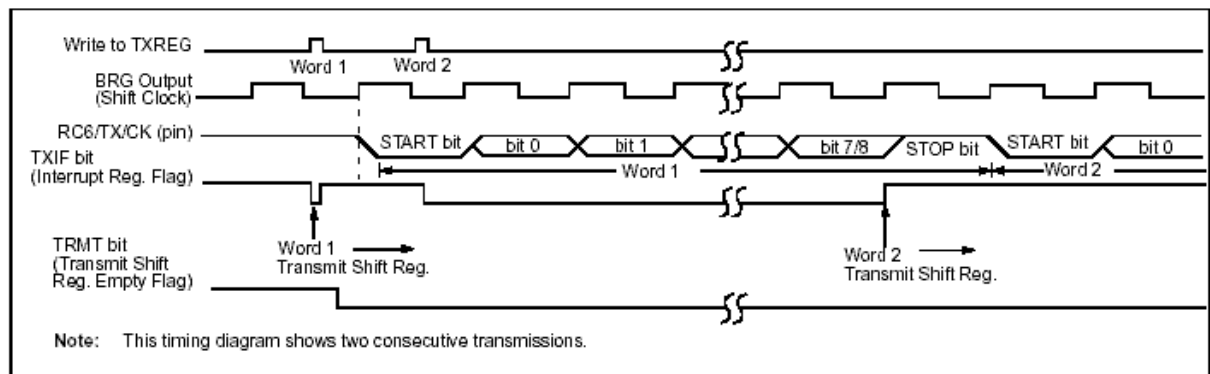
Jeśli zostanie wybrana transmisja 9-bitowa, bit transmisji TX9 (**TXSTA<6>**) powinien być ustawiony, dziewiąty bit powinien być zapisany do TX9D (**TXSTA<0>**). Dziewiąty bit musi być zapisany przed zapisem ośmiu bitów danych do rejestru **TXREG**. Taka sytuacja ma miejsce, ponieważ dane zapisane do rejestru **TXREG** mogą spowodować natychmiastowe przeniesienie danych do rejestru **TSR** (jeśli **TSR** jest pusty). W takim przypadku nieprawidłowy dziewiąty bit może być zapisany do rejestru **TSR**.

Aby ustawić transmisję asynchroniczną należy wykonać następujące kroki:

- Uaktywnić rejestr **SPBRG** wpisując odpowiednią wartość szybkości transmisji. W przypadku chęci uzyskania szybkiej transmisji, należy ustawić bit BRGH (**SPBRG<2>**).
- Uaktywnić asynchroniczny port szeregowy przez wyzerowanie bitu SYNC i ustawienie bitu SPEN.
- Jeśli przerwania będą wykorzystane, należy ustawić bit TXIE.
- Jeśli wykorzystana będzie transmisja 9-bitowa, należy ustawić bit TX9.
- Uaktywnić transmisję przez ustawienie bitu TXEN, który ustawi bit TXIF.
- Jeśli została wybrana transmisja 9-bitowa, bit dziewiąty powinien być załadowany do bitu TX9D.
- Załadować dane do rejestru **TXREG** (spowoduje to rozpoczęcie transmisji).



Rys. 10.2 Przebiegi czasowe z nadajnika asynchronicznego USART – jeden bajt



Rys. 10.3. Przebiegi czasowe z nadajnika asynchronicznego USART – dwa bajty

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>INTCON</b>	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>RCSTA</b>	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
<b>TXREG</b>	Rejestr układu transmisji USART								0000 0000	0000 0000
<b>TXSTA</b>	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
<b>SPBRG</b>	Rejestr generatora BRG								0000 0000	0000 0000

Tabela D10.3. Rejestry związane z nadajnikiem asynchronicznym układu USART

## Rejestr TXSTA

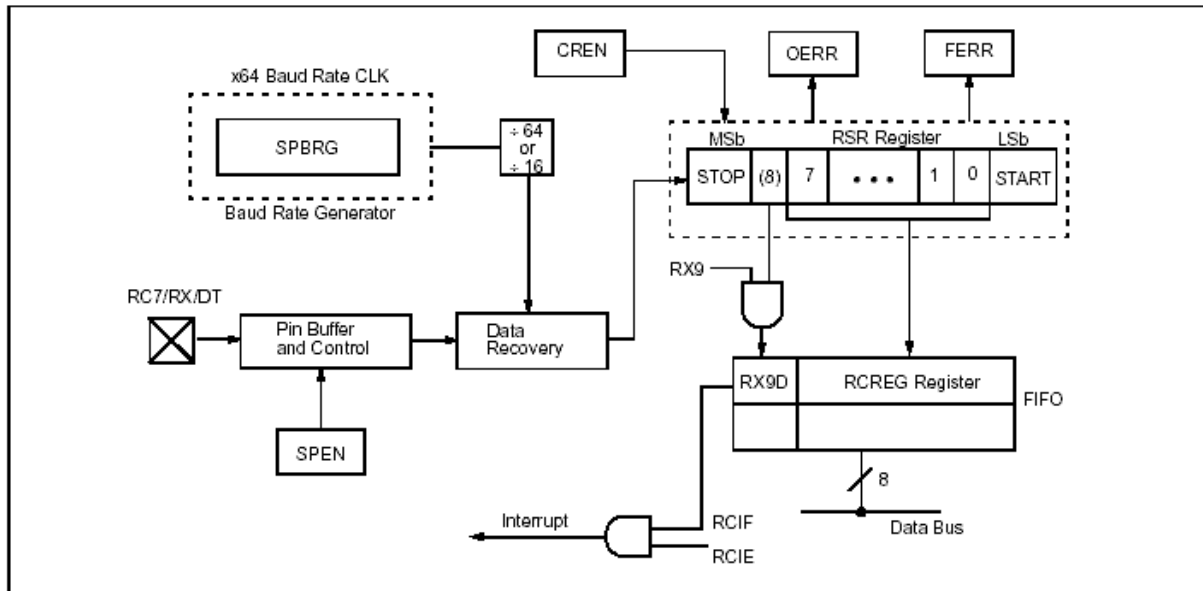
(Rejestr TXSTA jest rejestrem statusu i rejestrem kontrolnym nadajnika modułu USART)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D
bit 7	Bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **CSRC: detekcja kolizji (tylko w trybie nadawania):**  
tryb Asynchroniczny:  
bez znaczenia,  
tryb Synchroniczny:  
1 = tryb Master (sygnał zegara generowany wewnątrz z BRG),  
0 = tryb Slave (sygnał zegara z źródła zewnętrznego).
- bit 6      **TX9: bit włączenia transmisji 9-bitowej:**  
1 = transmisja 9-bitowa,  
0 = transmisja 8-bitowa.
- bit 5      **TXEN: bit włączający transmisję:**  
1 = transmisja możliwa,  
0 = transmisja niemożliwa.  
**Uwaga:** SREN/CREN jest nadrzędny w stosunku do TXEN w trybie SYNC.
- bit 4      **SYNC: bit wyboru trybu pracy USART:**  
1 = tryb synchroniczny,  
0 = tryb asynchroniczny,
- bit 3      **bit niezaimplementowany**
- bit 2      **BRGH: Bit wyboru szybkości transmisji:**  
tryb Asynchroniczny:  
1 = duża prędkość transmisji,  
0 = niska prędkość transmisji.  
tryb Synchroniczny:  
Nie używany w tym trybie.
- bit 1      **TRMT: Bit statusu rejestru przesuwającego TSR w nadawaniu:**  
1 = rejestr TSR pusty,  
0 = rejestr TSR pełny.
- bit 0      **TX9D: 9-ty bit danych transmitowanych:**  
może być bitem adresu/danych lub parzystości,

## Układ odbiornika transmisji asynchronicznej - USART

Schemat blokowy odbiornika jest pokazany na *Rys. 11*. Dane są czytane na pinie RC7/RX/DT i są przesyłane do bloku odbioru danych (*Data Recovery Block*). Blok odbioru danych jest szybkim rejestrem przesuwającym (prędkość generatora BRG x16), podczas gdy główny rejestr odbiornika szeregowego pracuje z częstotliwością  $F_{osc}$ .



*Rys. 10.4. Schemat blokowy układu odbiornika transmisji asynchronicznej*

Najważniejszą częścią odbiornika jest szeregowy rejestr przesuwany **RSR**. Po wykryciu bitu STOP, odebrana dana w rejestrze **RSR** jest przesyłana do rejestru **RCREG** (jeżeli jest on pusty). Po zakończeniu przesyłania zostaje ustawiona flaga RCIF (**PIR1<5>**). Przerwanie może być odblokowane/zablokowane poprzez ustawienie/wyzerowanie bitu RCIE w rejestrze **PIE1<5>**. Flagą RCIF jest bitem tylko do odczytu i może być wyzerowana tylko przez kontroler. Zostaje ona wyzerowana, gdy zostanie odczytany rejestr **RCREG** i jest on pusty. Rejestr **RCREG** jest podwójnie buforowany (dwustopniowy FIFO).

Jest możliwy odbiór i przesłanie dwóch bajtów danych do **RCREG** FIFO i zapoczątkowanie przesuwania trzeciego bajtu do rejestru **RSR**. Po wykryciu bitu STOP w trakcie wysyłania trzeciego bajtu, jeżeli rejestr **RCREG** jest ciągle pełny, to ustawiany jest bit przepełnienia OERR (**RCSTA<1>**). Dane w **RSR** zostaną utracone. Aby odzyskać dwa bajty znajdujące się w FIFO, należy dwukrotnie odczytać rejestr **RCREG**. Bit przepełnienia OERR musi być wyzerowany programowo. Robi się to poprzez wyzerowanie i ustawienie bitu CREN. Gdy bit OERR jest ustawiony, przesyłanie z rejestru **RSR** do **RCREG** jest wstrzymane, a kolejne dane nie są odbierane. Dlatego niezbędne jest wyzerowanie bitu błędu OERR. Bit błędu ramki FERR (**RCSTA<2>**) zostaje ustawiony, gdy bit STOP nie zostaje wykryty (stan wysoki). Bit FERR i 9-ty odebrany bit są buforowane tak, jak odbierana dana. Odczyt **RCREG** załaduje bity RX9D i FERR nowymi wartościami, dlatego konieczny jest odczyt **RCSTA**, zanim odczytany zostanie rejestr **RCREG**, aby nie stracić informacji o starych wartościach bitów RX9D i FERR.

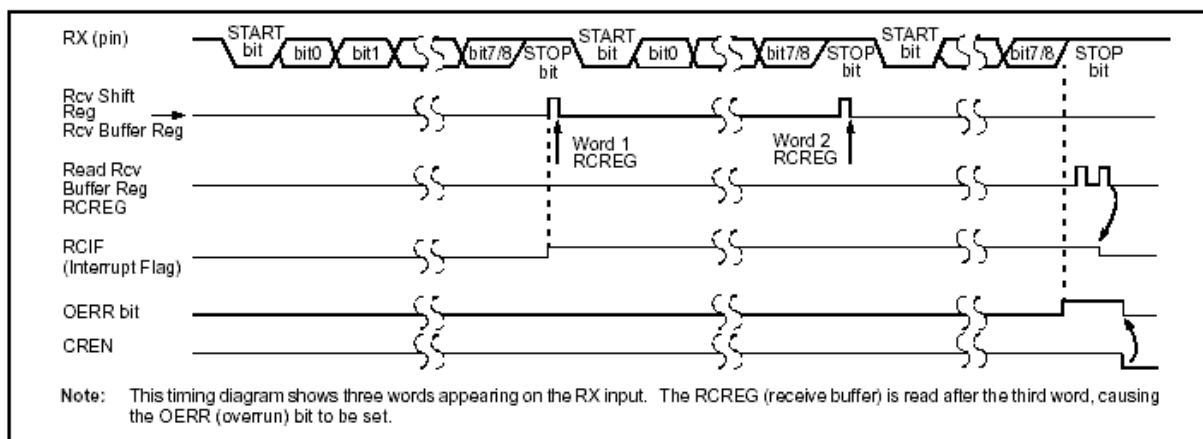


Aby ustawić tryb odbioru asynchronicznego należy:

1. Ustawić w rejestrze SPBRG zadaną szybkość transmisji. Jeżeli zadana jest duża szybkość transmisji, należy ustawić bit BRGH.
2. Uaktywnić asynchroniczny port szeregowy, zerując bit SYNC i ustawiając SPEN.
3. Jeśli przerwania będą wykorzystane, należy ustawić bit RCIE.
4. Jeśli wykorzystana będzie transmisja 9-bitowa, należy ustawić bit RX9.
5. Uaktywnić transmisję przez ustawienie bitu CREN.
6. Flaga RCIF zostanie ustawiona, gdy dane zostaną odebrane i zostanie wygenerowane przerwanie, jeżeli bit RCIE jest ustawiony.
7. Należy odczytać rejestr RCSTA, aby uzyskać dziewiąty bit (jeżeli dostępny) oraz określić, czy nie wystąpił błąd podczas transmisji.
8. Odczytać 8-bitów danych z rejestru RCREG.
9. W przypadku wystąpienia błędu, wyczyścić bit CREN.
10. Jeśli używane są przerwania, należy upewnić się, że bity GIE i PIE (bity 7 i 6) w rejestrze INTCON są ustawione.

Aby ustawić tryb odbioru asynchronicznego z detekcją adresu należy:

1. Ustawić w rejestrze SPBRG zadaną szybkość transmisji. Jeżeli zadana jest duża szybkość transmisji, należy ustawić bit BRGH.
2. Uaktywnić asynchroniczny port szeregowy, zerując bit SYNC i ustawiając SPEN.
3. Jeśli przerwania będą wykorzystane, należy ustawić bit RCEN i odpowiedni bit priorytetu RCIP.
4. Ustawić bit RX9, by odblokować 9-bitowy odbiór.
5. Ustawić bit ADDEN, by odblokować detekcję adresu.
6. Odblokować odbiór ustawieniem bitu CREN.
7. Flaga RCIF zostanie ustawiona, gdy dane zostaną odebrane. Przerwania zostaną wygenerowane, jeżeli bity RCIE i GIE są ustawione.
8. Należy odczytać rejestr **RCSTA** by określić, czy wystąpił błąd podczas odbioru, oraz odczytać 9 bit danej.
9. Należy odczytać rejestr **RCREG** by określić, czy urządzenie zostało zaadresowane.
10. Jeżeli wystąpił jakikolwiek błąd należy wyzerować bit CREN.
11. Jeżeli urządzenie zostało zaadresowane, należy wyzerować bit ADDEN by dane odebrane zostały umieszczone w buforze odbioru i mogło być wygenerowane przerwanie.



Rys. 10.5 Przebiegi czasowe odbiornika asynchronicznego USART

## Rejestr RCSTA

(Rejestr **RCSTA** jest rejestrem statusu i rejestrem kontrolnym odbiornika modułu USART)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7      **SPEN: bit uaktywniania portu szeregowego:**  
1 = port szeregowy aktywny (konfiguruje piny RC7/RX/DT i RC6/TX/CK jako piny portu szeregowego),  
0 = port szeregowy nieaktywny.
- bit 6      **RX9: bit włączenia odbioru 9-bitowego:**  
1 = ramka 9-bitowa,  
0 = ramka 8-bitowa.
- bit 5      **SREN: bit włączający jednokrotny odbiór:**  
tryb Asynchroniczny:  
bez znaczenia,  
tryb Synchroniczny – Master:  
1 = włącza jednokrotny odbiór,  
0 = wyłącza jednokrotny odbiór,  
(bit jest zerowany po zakończeniu odbioru),  
Tryb Synchroniczny – Slave:  
bez znaczenia,
- bit 4      **CREN: bit aktywowania odbioru ciągłego:**  
tryb Asynchroniczny:  
1 = włącza ciągły odbiór,  
0 = wyłącza ciągły odbiór,  
tryb Synchroniczny:  
1 = ciągły odbiór, dopóki bit CREN nie jest wyzerowany (CREN jest nadrzędny w stosunku do SREN),  
0 = wyłącza ciągły odbiór,
- bit 3      **ADDEN: bit aktywowania wykrywania adresu:**  
tryb Asynchroniczny 9-bitowy (RX9 = 1):  
1 = uaktywnia wykrywanie adresu, uaktywnia przerwanie i zapisuje bufor odczytu, kiedy RSR<8> jest ustawiony,  
0 = dezaktywuje wykrywanie adresu, wszystkie bajty są odczytywane i dziewiąty bit może być wykorzystany jako bit parzystości..
- bit 2      **FERR: bit błędu odbioru ramki:**  
1 = błąd odbioru ramki (może być zmieniony przez odczyt rejestru RCREG i odbiór następnego bajtu),  
0 = brak błędu odbioru ramki.
- bit 1      **OERR: bit błędu przepełnienia:**  
1 = błąd przepełnienia (kasowanie poprzez wyzerowanie bitu CREN),  
0 = brak błędu przepełnienia.
- bit 0      **RX9D: 9-ty bit danych odebranych:**  
może być bitem adresu/danych lub parzystości, musi być wyliczony przez oprogramowanie użytkownika,

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>INTCON</b>	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>RCSTA</b>	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
<b>RCREG</b>	Rejestr układu odbiorczego USART								0000 0000	0000 0000
<b>TXSTA</b>	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
<b>SPBRG</b>	Rejestr generatora BRG								0000 0000	0000 0000

Tabela D10.4. Rejestry związane z odbiornikiem asynchronicznym układu USART

## Transmisja Synchroniczna –Master

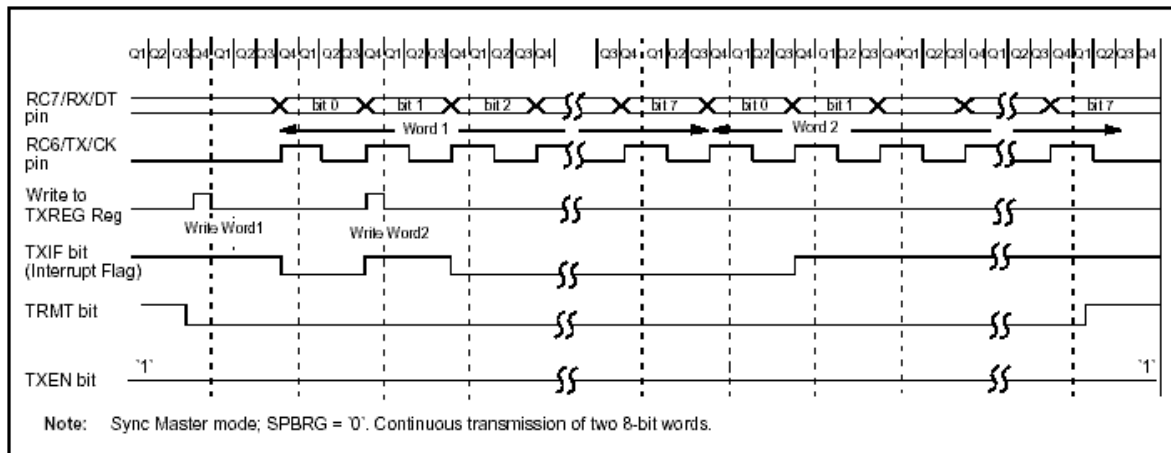
W synchronicznym trybie Master, dana jest przekazywana sposobem half-duplex (nadawanie i odbiór nie następują w tym samym momencie). Podczas nadawania odbiór jest wstrzymywany i na odwrót. Wejście w tryb synchroniczny następuje poprzez ustawienie bitu SYNC (rejestr **TXSTA**<4>). Dodatkowo ustawienie bitu SPEN (**RCSTA**<7>) przyporządkowuje odpowiednio pinom RC6/TX/CK i RC7/RX/DT linii CK (zegar) i DT (dane). Tryb Master oznacza, że procesor nadaje sygnał zegara na linii CK. Wejście w tryb Master następuje poprzez ustawienie bitu CSRC (**TXSTA**<7>).

## Układ nadajnika

Najważniejszym elementem nadajnika jest szeregowy rejestr przesuwany **TSR**. Rejestr przesuwany otrzymuje dane z rejestru buforowego **TXREG**. Do rejestru **TXREG** informacja jest ładowana programowo. Rejestr **TSR** nie zostanie załadowany kolejną daną, dopóki ostatni bit poprzedniej danej nie zostanie wysłany. Gdy tylko ostatni bit zostanie nadany, rejestr **TSR** jest ładowany nową daną z **TXREG** (jeżeli jest dostępna). Kiedy rejestr **TXREG** prześle dane do rejestru **TSR** (następuje to w jednym cyklu zegarowym), **TXREG** jest pusty i zostaje ustawiona flaga przerwania **TXIF** (**PIR1**<4>). Przerwanie może być odblokowane/zablokowane poprzez ustawienie/wyzerowanie bitu **TXIE** (**PIE1**<4>). Flaga **TXIF** będzie ustawiona niezależnie od stanu bitu **TXIE** (**PIE**<4>) i nie może być wyzerowana programowo. Zostanie skasowana tylko wtedy, kiedy nowa dana będzie załadowana do rejestru **TXREG**. Podczas gdy flaga **TXIF** pokazuje stan rejestru **TXREG**, inny bit **TRMT** (w **TXSTA**<1>) pokazuje stan rejestru **TSR**. **TRMT** jest bitem tylko do odczytu, który jest ustawiany, kiedy **TSR** jest pusty. Z tym bitem nie jest związane żadne przerwanie, zatem aby określić czy rejestr **TSR** jest pusty, musi to być każdorazowo sprawdzone przez użytkownika. Rejestr **TSR** nie jest mapowany w pamięci danych, zatem nie jest on dostępny dla użytkownika.

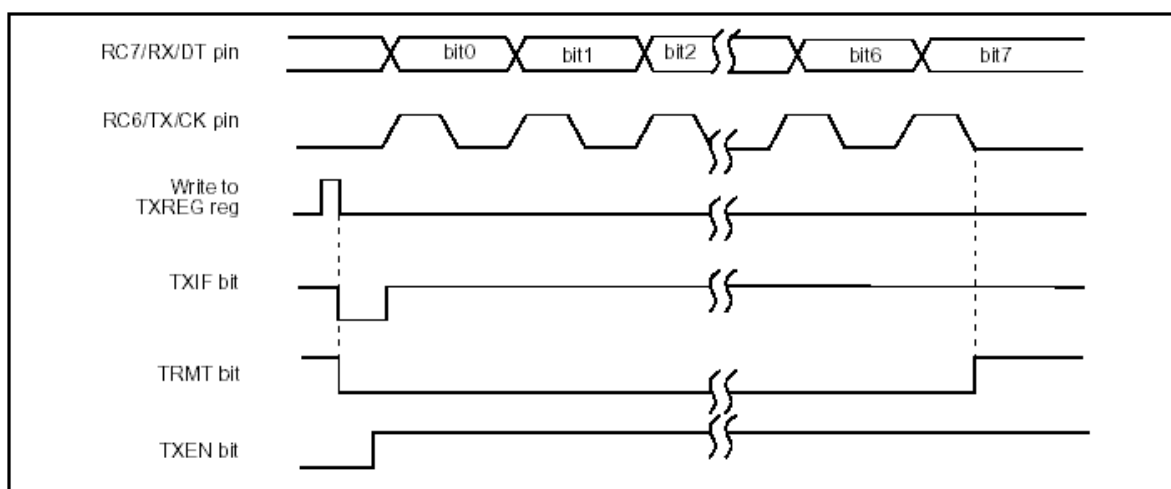
Umożliwienie transmisji następuje poprzez ustawienie bitu **TXEN** (**TXSTA**<5>). Transmisja nie nastąpi dopóki rejestr **TXREG** nie zostanie załadowany danymi. Pierwszy bit danej zostanie wysłany przy kolejnym narastającym zboczku zegarowym na linii CK. Dana wyjściowa jest stabilna przy opadającym zboczku zegarowym. Transmisja może być także zapoczątkowana poprzez pierwsze załadowanie rejestru **TXREG**, a następnie ustawienie bitu

TXEN. Ustawienie bitu TXEN zapoczątkuje pracę układu BRG, który bezzwłocznie generuje sygnał zegara. Normalnie, kiedy transmisja jest zapoczątkowana, rejestr **TSR** jest pusty, więc przesłanie danej do rejestru TXREG spowoduje natychmiastowe przesłanie jej do **TSR**, powodując wyczyszczenie **TXREG**.



Rys. 10.6. Transmisja Synchroniczna – ciągła transmisja dwóch słów 8-bit

Wyzerowanie bitu TXEN podczas transmisji spowoduje jej przerwanie oraz reset nadajnika. Piny DT i CK przejdą w stan wysokiej impedancji. Jeżeli w czasie trwania transmisji bit CREN lub bit SREN będzie ustawiony, to także zostanie ona przerwana i pin DT przejdzie w stan wysokiej impedancji (dla odbioru). Pin CK pozostanie ustawiony jako wyjście, jeżeli bit CSRC będzie ustawiony (wewnętrzny zegar). Logika nadajnika jednak nie zostanie skasowana, choć nie jest połączona z pinami. Aby zresetować nadajnik, użytkownik musi wyczyścić bit TXEN. Jeżeli bit SREN jest ustawiony (by przerwać trwającą transmisję i odebrać pojedyncze słowo), wówczas po odebraniu pojedynczego słowa, bit SREN będzie wyzerowany i port szeregowy powróci do nadawania do momentu, gdy bit TXEN będzie ciągle ustawiony. Linia DT będzie natychmiastowo przełączona ze stanu wysokiej impedancji trybu odbioru do nadawania i zacznie pracować. Aby tego uniknąć, bit TXEN powinien zostać wyzerowany.



Rys. 10.7. Transmisja Synchroniczna – rozpoczęcie poprzez bit TXEN

W celu wybrania 9-cio bitowej transmisji, bit TX9 (**TXSTA<6>**) powinien być ustawiony, a 9-ty bit powinien być wpisywany do TX9D (**TXSTA<0>**). Dziewiąty bit musi

być wpisany zanim zostanie wpisana 8-mio bitowa dana do rejestru **TXREG**. Jest tak dlatego, ponieważ dana wpisana do **TXREG** może spowodować natychmiastowe przesyłanie danej do rejestru **TSR** (jeśli jest on pusty). Jeżeli **TSR** był pusty i **TXREG** został zapisany zanim wpisano nową wartość TX9D, to będzie załadowana wartość dotychczasowa TX9D.

Aby ustawić transmisję synchroniczną w trybie Master należy:

1. Zainicjalizować rejestr **SPBRG** dla odpowiedniej prędkości transmisji.
2. Uaktywnić synchroniczny port szeregowy w trybie Master ustawiając bity SYNC, SPEN i CSRC.
3. Jeśli przerwania będą używane, należy ustawić bit TXIE.
4. Dla transmisji 9-cio bitowej danej należy ustawić bit TX9.
5. Umożliwić transmisję ustawiając bit TXEN.
6. Jeżeli wybrano transmisję 9-cio bitową, 9-ty bit powinien być załadowany do TX9D.
7. Zapoczątkować transmisję poprzez załadowanie danej do rejestru **TXREG**.

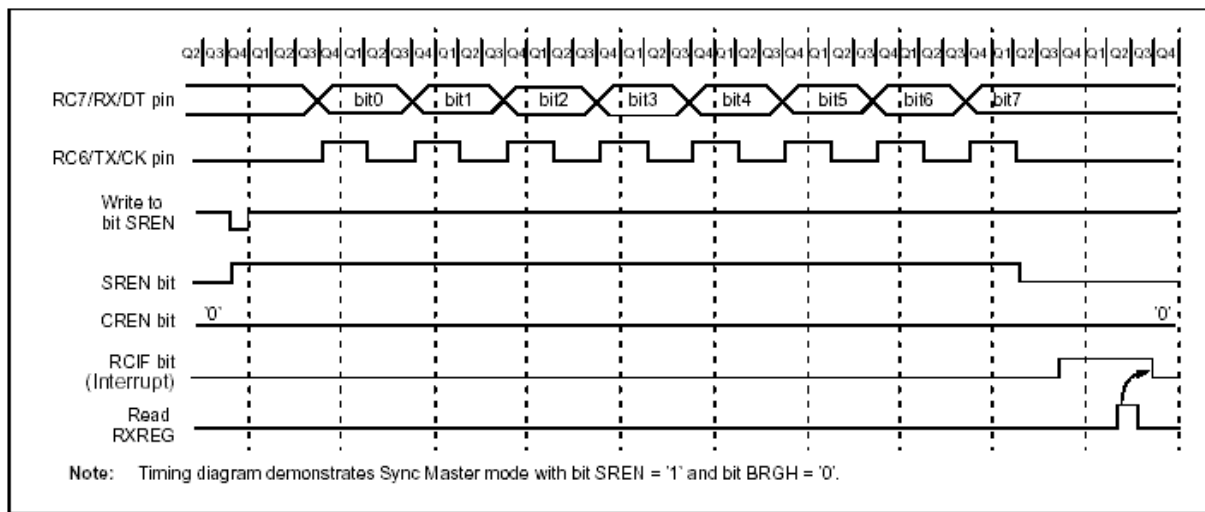
Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>INTCON</b>	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>RCSTA</b>	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
<b>RCREG</b>	Rejestr układu odbiorczego USART								0000 0000	0000 0000
<b>TXSTA</b>	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
<b>SPBRG</b>	Rejestr generatora BRG								0000 0000	0000 0000

Tabela D10.5. Rejestry związane z nadajnikiem synchronicznym układu USART – tryb Master

## Układ odbiornika

Gdy tryb synchroniczny jest wybrany, odbiór jest możliwy poprzez ustawienie bitu SREN (**RCSTA<5>**) lub bitu CREN (**RCSTA<4>**). Dana jest próbkowana na pinie RC7/RX/DT na opadającym zboczu zegara. Jeżeli jest ustawiony bit SREN, wówczas jest odbierane tylko pojedyncze słowo. Jeżeli bit CREN jest ustawiony, to odbiór będzie kontynuowany, aż bit CREN zostanie wyzerowany. Jeżeli oba bity są ustawione, pierwszeństwo ma CREN. Po ostatnim taktie zegara odebrana dana w rejestrze RSR (*Receive Shift Register*) jest przesyłana do rejestru **RCREG** (jeżeli jest on pusty). Kiedy przesłanie jest zakończone, flaga przerwania RCIF (**PIR1<5>**) zostaje ustawiona. Przerwanie może być odblokowane/zablokowane poprzez ustawienie/wyzerowanie bitu RCIE (**PIE1<5>**). Flaga RCIF jest bitem tylko do odczytu, która może być wyzerowana sprzętowo. W tym przypadku będzie ona skasowana, gdy rejestr **RCREG** zostanie odczytany i będzie on pusty. **RCREG** jest rejestrem podwójnie buforowanym (dwupoziomowy FIFO). Możliwe jest odebranie dwóch bajtów i przesłanie ich do RCRREG FIFO i rozpoczęcie przyjmowania trzeciego bajtu do rejestru **RSR**. Jeżeli podczas odbierania ostatniego bitu trzeciego bajtu **RCREG** jest wciąż pełny, to zostanie ustawiony bit błędu przepełnienia OERR (**RCSTA<1>**). Słowo w **RSR** zostanie utracone. Rejestr **RCREG** może być odczytany dwukrotnie, by odzyskać dwa bajty z FIFO. Bit OERR musi być wyczyszczony programowo (poprzez wyczyszczenie CREN). Jeżeli bit OERR jest ustawiony, przesyłanie z **RSR** do **RCREG** jest zablokowane, więc niezbędne jest, aby go wyzerować. Dziewiąty bit odbieranej danej jest buforowany w ten sam

sposób, jak odbierana dana. Odczytanie rejestru **RCREG** załaduje bit **RX9D** nową wartością, dlatego jest konieczne, by użytkownik odczytał rejestr **RCSTA** zanim odczyta **RCREG**, aby nie utracić aktualnej informacji w **RX9D**.



*Rys. D10.8. Odbiór synchroniczny pojedynczego słowa (SREN) w trybie Master*

Aby ustawić odbiór synchroniczny w trybie Master należy:

1. Ustawić w rejestrze **SPBRG** zadaną prędkość transmisji.
2. Uaktywnić synchroniczny port szeregowy ustawiając bity **SYNC**, **SPEN** i **CSRC**.
3. Upewnić się, że bity **CREN** i **SREN** są wyczyszczone.
4. Jeśli przerwania będą używane, należy ustawić bit **RCIE**.
5. Dla odbioru 9-cio bitowej danej ustawić bit **RX9**.
6. Dla odbioru pojedynczego słowa ustawić bit **SREN**. Dla odbioru ciągłego ustawić bit **CREN**.
7. Flaga **RCIF** zostanie ustawiona, gdy dana zostanie odebrana i zostanie wygenerowane przerwanie, pod warunkiem, że bit **RCIE** był ustawiony.
8. 9-ty bit danej należy odczytać z rejestru **RCSTA** (jeżeli używany) oraz określić, czy nie wystąpił błąd podczas odbioru.
9. Odczytać 8-mio bitową daną z rejestru **RCREG**.
10. W przypadku wystąpienia błędu wyczyścić bit **CREN**.
11. Dla obsługi przerwań upewnić się, że bity **GIE** i **PEIE** (bity 7 i 6) w rejestrze **INTCON** są ustawione.

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany pozostałych RESET-ach
<b>INTCON</b>	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>RCSTA</b>	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
<b>RCREG</b>	Rejestr układu odbiorczego USART								0000 0000	0000 0000
<b>TXSTA</b>	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
<b>SPBRG</b>	Rejestr generatora BRG								0000 0000	0000 0000

Tabela D10.6. Rejestry związane z odbiornikiem synchronicznym USART – tryb Master

## Transmisja Synchroniczna - Slave

Synchroniczny tryb Slave różni się od trybu Master tym, że sygnał zegara jest dostarczany zewnątrz przez pin RC6/TX/CK (a nie wewnątrz, jak w trybie Master). Pozwala to urządzeniu na przesyłanie lub odbiór danych podczas przebywania w stanie SLEEP. Tryb ten jest ustawiany przez wyzerowanie bitu CSRC (**TXSTA<7>**).

### Układ nadajnika

Operacje w synchronicznych trybach Master i Slave przebiegają identycznie z wyjątkiem stanu SLEEP.

Jeśli dwa słowa zostaną wpisane do **TXREG** i następnie wykonana zostanie instrukcja SLEEP wówczas wykonają się następujące operacje:

- Pierwsze słowo zostanie natychmiast przeniesione do rejestru **TSR** i wysłane.
- Drugie słowo pozostanie w rejestrze **TXREG**.
- Flaga TXIF nie zostanie ustawiona.
- Kiedy pierwsze słowo zostanie wysłane przez przesuwany rejestr **TSR**, rejestr **TXREG** przeniesie drugie słowo do rejestru **TSR** i flaga TXIF zostanie ustawiona.
- Jeśli bit TXIE jest ustawiony, przerwanie „obudzi” układ ze stanu SLEEP i jeżeli przerwania są globalnie odblokowane, program przejdzie do wykonywania instrukcji umieszczonych pod wektorem przerwania.

Aby ustawić transmisję synchroniczną w trybie Slave należy:

- Ustawić bity SYNC i SPEN i wyzerować bit CSRC – uaktywni to synchroniczny port szeregowy w trybie Slave.
- Wyzerować bity CREN i SREN.
- Jeśli przerwania są wykorzystywane, ustawić bit TXIE.
- Jeśli wybrana została transmisja 9-bitowa, ustawić bit TX9.
- Ustawić bit TXEN zezwalający na nadawanie.
- Jeśli została wybrana transmisja 9-bitowa, dziewiąty bit powinien być wpisany do bitu TX9D.
- Rozpoczęcie transmisji następuje poprzez wpisanie danej do rejestru **TXREG**.
- Jeśli przerwania są używane, upewnić się, że bity GIE i PEIE (bity 6 i 7) w rejestrze **INTCON** są ustawione.

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany pozostałe ch RESET-ach
<b>INTCON</b>	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>RCSTA</b>	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
<b>TXREG</b>	Rejestr układu transmisji USART								0000 0000	0000 0000
<b>TXSTA</b>	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
<b>SPBRG</b>	Rejestr generatora BRG								0000 0000	0000 0000

Tabela D10.7.. Rejestry związane z nadajnikiem synchronicznym USART – tryb Slave

## Układ odbiornika

Jeżeli przed wykonaniem instrukcji SLEEP został ustawiony bit CREN zezwalający na odbiór, wówczas dana może zostać odebrana podczas przebywania w stanie SLEEP. Po odebraniu słowa, rejestr **RSR** prześle daną do rejestru **RCREG** i jeśli bit zezwolenia RCIE jest ustawiony, wygenerowane zostanie przerwanie „obudzi” układ ze stanu SLEEP.

Aby ustawić odbiór synchroniczny w trybie Slave należy:

1. Ustawić bity SYNC i SPEN i wyzerować bit CSRC – uaktywni to synchroniczny port szeregowy w trybie Slave.
2. Jeśli przerwania są wykorzystywane, ustawić bit RCIE.
3. Jeśli wybrana została transmisja 9-bitowa, ustawić bit RX9.
4. Ustaw bit CREN zezwalający na odbiór.
5. Gdy odbiór zostanie zakończony, flaga RCIF zostanie ustawiona. Jeśli bit RCIE został ustawiony, to wygenerowane zostanie przerwanie.
6. Odczytaj rejestr RCSTA w celu odczytania dziewiątego bitu i sprawdzenia, czy podczas transmisji nie pojawił się błąd.
7. Odczytaj odebraną 8-bitową daną znajdującą się w rejestrze RCREG.
8. Jeżeli pojawił się błąd, wyzeruj bit CREN.
9. Jeśli przerwania są używane, upewnij się, że bity GIE i PEIE (bity 7 i 6) w rejestrze INTCON są ustawione.

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany pozostałe ch RESET-ach
<b>INTCON</b>	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>RCSTA</b>	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
<b>RCREG</b>	Rejestr układu odbioru USART								0000 0000	0000 0000
<b>TXSTA</b>	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
<b>SPBRG</b>	Rejestr generatora BRG								0000 0000	0000 0000

Tabela D10.8.. Rejestry związane z odbiornikiem asynchronicznym USART – tryb Slave



## 12 Moduł 10-bitowego przetwornika analogowo-cyforwego

Moduł przetwornika analogowo-cyforwego mikrokontrolera PIC18F452 posiada osiem wejść analogowych (AN0:AN7).

Moduł przetwornika A/D posiada cztery rejestry:

- **ADRESH** rejestr wyniku przetwarzania – starszy bajt,
- **ADRESL** rejestr wyniku przetwarzania – młodszy bajt,
- **ADCON0** rejestr kontrolny 0,
- **ADCON1** rejestr kontrolny 1.

### Rejestr ADCON0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/~DONE	-	ADON
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bit 7-6      **ADCS1:ADCS0: bit wyboru zegara przetwornika A/D:**

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Prędkość konwersji
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC wewnętrzny oscylator RC
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC wewnętrzny oscylator RC

bit 5-3      **CHS2:CHS0: bity wyboru kanału analogowego:**

000 = kanał 0 (RA0/AN0),  
 001 = kanał 1 (RA1/AN1),  
 010 = kanał 2 (RA2/AN2),  
 011 = kanał 3 (RA3/AN3),  
 100 = kanał 4 (RA5/AN4),  
 101 = kanał 5 (RE0/AN5),  
 110 = kanał 6 (RE1/AN6),  
 111 = kanał 7 (RE2/AN7),

bit 2      **GO/~DONE: bit statusu konwersji:**

Jeżeli ADON=1:

1 = trwa konwersja (ustawiając ten bit rozpoczyna się konwersję, który jest automatycznie zerowany sprzętowo, gdy zakończy się proces konwersji),  
 0 = układ nie jest w trakcie konwersji,

bit 1      **bit niezaimplementowany,**

bit 0      **ADON: uaktywnienie modułu przetwornika A/D:**

1 = przetwornik jest aktywny,  
 0 = przetwornik jest wyłączony i nie pobiera prądu,

## Rejestr ADCON1

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	-	-	PCFG3	PCFG2	PCFG1	PCFG0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bit 7 **ADFM: bit wyboru formatu wyniku przetwarzania,**  
 1 = justowanie w prawo. 6 najbardziej znaczących bitów ADRESH czytane jest jako '0',  
 0 = justowanie w lewo. 6 najmniej znaczących bitów ADRESL czytane jest jako '0'.

bit 6 **ADCS2: bit wyboru zegara przetwornika A/D:**

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Prędkość konwersji
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC wewnętrzny oscylator RC
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC wewnętrzny oscylator RC

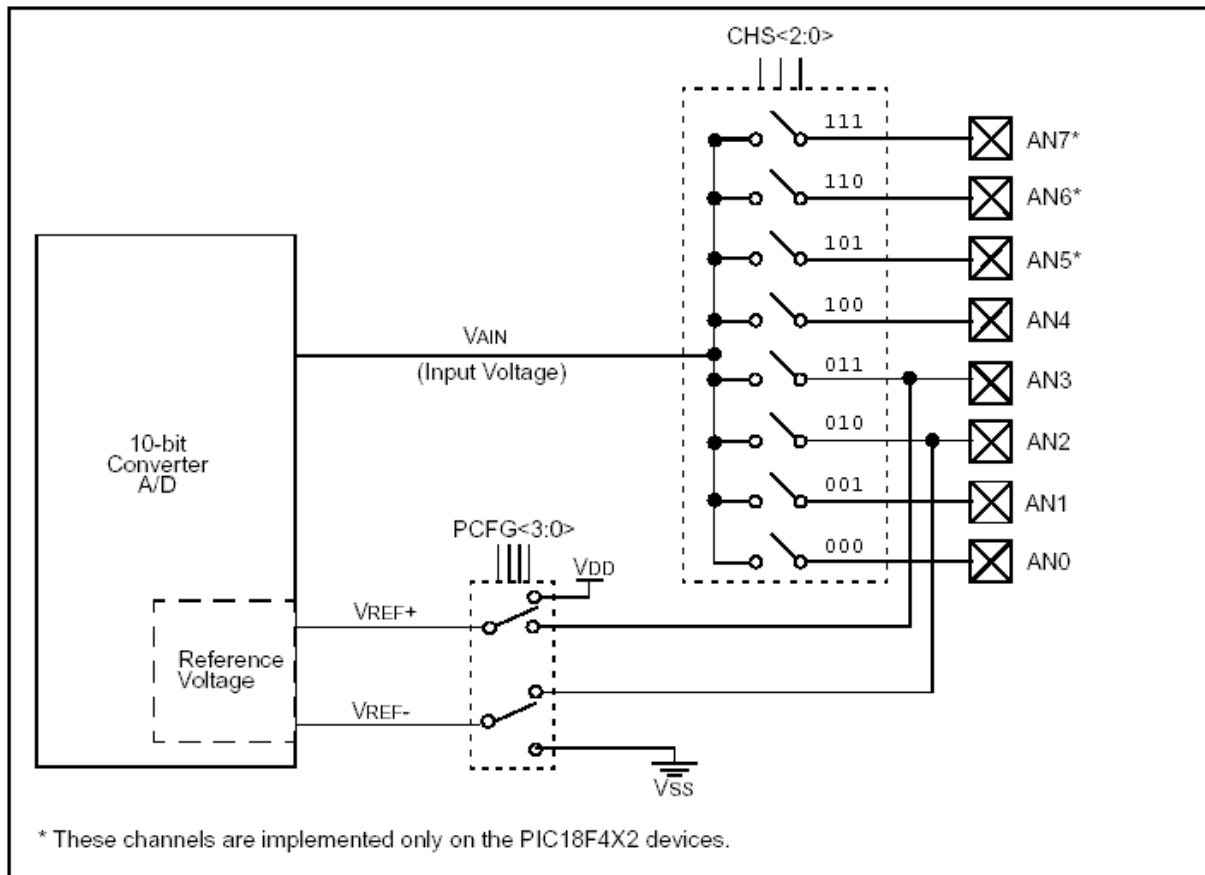
bit 5-4 **bity niezaimplementowane,**  
 bit 3-0 **PCFG3:PCFG0: bity odpowiedzialne za konfigurację linii portu przetwornika A/D:**

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	V <sub>REF+</sub>	V <sub>REF-</sub>	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8 / 0
0001	A	A	A	A	V <sub>REF+</sub>	A	A	A	AN3	VSS	7 / 1
0010	C	C	C	A	A	A	A	A	VDD	VSS	5 / 0
0011	C	C	C	A	V <sub>REF+</sub>	A	A	A	AN3	VSS	4 / 1
0100	C	C	C	C	A	C	A	A	VDD	VSS	3 / 0
0101	C	C	C	C	V <sub>REF+</sub>	C	A	A	AN3	VSS	2 / 1
011x	C	C	C	C	C	C	C	C	-	VSS	0 / 0
1000	A	A	A	A	V <sub>REF+</sub>	V <sub>REF-</sub>	A	A	AN3	AN2	6 / 2
1001	C	C	A	A	A	A	A	A	VDD	VSS	6 / 0
1010	C	C	A	A	V <sub>REF+</sub>	A	A	A	AN3	VSS	5 / 1
1011	C	C	A	A	V <sub>REF+</sub>	V <sub>REF-</sub>	A	A	AN3	AN2	4 / 2
1100	C	C	C	A	V <sub>REF+</sub>	V <sub>REF-</sub>	A	A	AN3	AN2	3 / 2
1101	C	C	C	C	V <sub>REF+</sub>	V <sub>REF-</sub>	A	A	AN3	AN2	2 / 2
1110	C	C	C	C	C	C	C	A	VDD	VSS	1 / 0
1111	C	C	C	C	V <sub>REF+</sub>	V <sub>REF-</sub>	C	A	AN3	AN2	1 / 2

A = wejście analogowe C = wejście/wyjście cyfrowe

1: Kolumna ta zawiera liczbę kanałów analogowych dostępnych jako wejścia przetwornika A/D i liczbę kanałów analogowych dostępnych jako wejścia napięcia referencyjnego.

Rejestr **ADCON0** kontroluje pracę przetwornika A/D, natomiast Rejestr **ADCON1** konfiguruje funkcje poszczególnych pinów portu. Piny portu mogą pełnić rolę wejść analogowych (RA3 może również być wykorzystany jako wejście napięcia referencyjnego) lub być liniami cyfrowymi wejścia/wyjścia.



Rys. 11.1. Schemat blokowy przetwornika A/C

Napięcie referencyjne (dodatnie lub ujemne) wybierane jest programowo. Napięcie referencyjne może również być podłączone do zewnętrznych pinów RA3/AN3/Vref+ i RA2/AN2/Vref-.

Przetwornik A/C ma możliwość pracy podczas trybu uśpienia SLEEP. Aby był to możliwe, sygnał zegarowy musi pochodzić z wewnętrznego generatora RC.

Rejestry **ADRESH** i **ADRESL** zawierają 10-bitowy wynik konwersji analogowo-cyfrowej. Kiedy zakończy się konwersja wynik jest przesyłany do tej pary rejestrów, bit GO/~DONE (**ADCON0<2>**) jest zerowany i flaga przerwania od przetwornika A/D jest ustawiana.

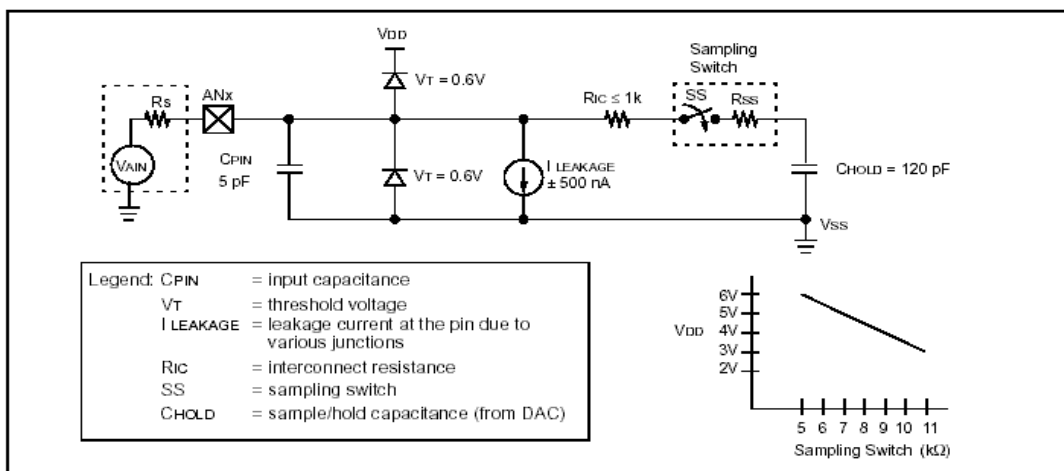
Po skonfigurowaniu przetwornika A/D należy wybrać kanał przetwarzania. Linie wejść analogowych muszą zostać odpowiednio ustawione jako wejścia w rejestrach kierunkowych TRIS portów A i E.

Aby dokonać konwersji analogowo-cyfrowej, należy wykonać następujące czynności:

1. Skonfiguruj moduł przetwornika A/D:
  - Skonfiguruj wejścia analogowe/napięcie referencyjne i cyfrowe we/wy (**ADCON1**).
  - Wybierz kanał przetwornika A/D (**ADCON0**).
  - Wybierz zegar przetwornika A/D (**ADCON0**).
  - Włącz moduł przetwornika A/D (**ADCON0**).
2. Skonfiguruj przerwanie przetwornika A/D (jeśli potrzebne):
  - Wyzeruj bit ADIF.
  - Ustaw bit ADIE.
  - Ustaw bit GIE.
  - Ustaw bit PEIE.
3. Odczekaj czas akwizycji.
4. Rozpocznij konwersję ustawiając bit GO/~DONE (**ADCON0**).
5. Odczekaj na zakończenie konwersji przez sprawdzanie:
  - czy wyzerowany został bit GO/~DONE (gdy przerwania są zablokowane),
  - oczekiwanie na przerwanie z przetwornika A/D,
6. Odczytaj parę rejestrów z wynikiem przetwarzania (**ADRESH:ADRESL**), wyzeruj bit ADIF – jeśli jest to wymagane.
7. Aby dokonać następnej konwersji przejdź do kroku 1 lub 2 w zależności od potrzeb. Przed rozpoczęciem kolejnej konwersji należy odczekać czas równy co najmniej  $2T_{AD}$  ( $T_{AD}$  – czas konwersji dla jednego bitu).

### Wymagania związane z przetwarzaniem A/C

Aby przetwornik A/C osiągał swoją typową dokładność, kondensator układu próbkująco-pamiętającego ( $C_{HOLD}$ ) musi mieć możliwość pełnego naładowania do poziomu napięcia wejściowego. Model analogowego wejścia pokazany jest na Rys.33. Impedancja źródła ( $R_s$ ) i impedancja wewnętrznego klucza próbkującego ( $R_{ss}$ ) bezpośrednio wpływają na czas naładowania kondensatora  $C_{HOLD}$ . Impedancja klucza ( $R_{ss}$ ) zmienia się wraz z napięciem zasilania ( $V_{DD}$ ). **Maksymalna rekomendowana impedancja dla źródła analogowego wynosi 2.5k $\Omega$ .**



Rys. 11.2. Model wejścia analogowego

Aby obliczyć minimalny czas akwizycji, należy się posłużyć następującym równaniem:

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF} = 2\mu s + T_C + [(Temperatura - 25^\circ C)(0.05\mu s / ^\circ C)]$$

$$\begin{aligned} T_C &= - C_{HOLD} (R_{IC} + R_{SS} + R_S) \ln(1/2048) \\ &= - 120\text{pF} (1\text{k}\Omega + 7\text{k}\Omega + 2.5\text{k}\Omega) \ln(0.0004883) \\ &= - 120\text{pF}(10.5\text{k}\Omega) \ln(0.0004883) \\ &= - 1.26\mu s (- 7.6246) \\ &= 9.61\mu s \end{aligned}$$

$$\begin{aligned} T_{ACQ} &= 2\mu s + 9.61\mu s + [(50^\circ C - 25^\circ C)(0.05\mu s / ^\circ C)] \\ &= 11.61\mu s + 1.25\mu s \\ &= 12.86\mu s \end{aligned}$$

$T_{AMP}$  = czas ustalenia się czasu na wejściu wzmacniacza

$T_C$  = czas przeładowania pojemności w układzie próbkująco – pamiętającym

$T_{COEFF}$  = czas wynikający z pracy w określonej temperaturze

### *Przykład D11.1 Obliczanie minimalnego czasu akwizycji*

Przykład został obliczony przy następujących założeniach:

- $C_{HOLD} = 120\text{pf}$ ,
- $R_S = 2.5\text{k}\Omega$ ,
- błąd konwersji  $\leq \frac{1}{2}$  LSB,
- $V_{dd} = 5\text{V} \rightarrow R_{SS} = 7\text{k}\Omega$ ,
- Temperatura =  $50^\circ\text{C}$

### **Wybór zegara do przetwarzania A/C**

Czas przetwarzania jednego bitu zdefiniowany jest jako  $T_{AD}$ . Przetwarzanie 10-bitowe wymaga minimum czasu równego  $12 T_{AD}$ . Źródło zegara dla konwersji jest wybierane programowo. Istnieje siedem możliwości do wyboru dla  $T_{AD}$ :

- $2T_{OSC}$ ,
- $4T_{OSC}$ ,
- $8T_{OSC}$ ,
- $16T_{OSC}$ ,
- $32T_{OSC}$ ,
- $64T_{OSC}$ ,
- wewnętrzny oscylator RC ( $2\text{--}6\mu s$ ).

Dla przetwarzania analogowo–cyfrowego zegar przetwarzania A/D musi być tak wybrany, aby zapewnić minimalny czas  $T_{AD}$  równy  $1,6\mu s$ .

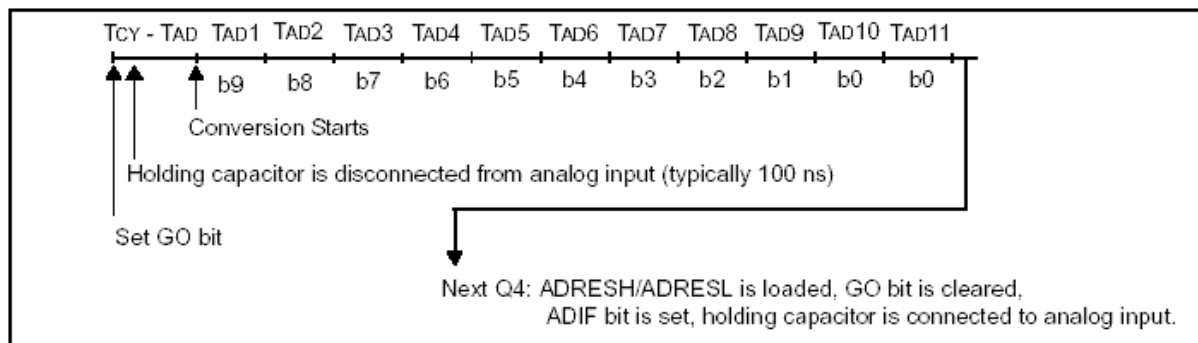
Źródło sygnału zegara modułu A/D ( $T_{AD}$ )		Maksymalna Częstotliwość Układu	
Opcja	ADCS1:ADCS0	PIC18FXX2	PIC18LFXX2
2 $T_{OSC}$	000	1.25 MHz	666kHz
4 $T_{OSC}$	100	2.5 MHz	1.33 MHz
8 $T_{OSC}$	001	5 MHz	2.67 MHz
16 $T_{OSC}$	101	10 MHz	5.33 MHz
32 $T_{OSC}$	010	20 MHz	10.67 MHz
64 $T_{OSC}$	110	40 MHz	21.33 MHz
RC	011	-	-

Tabela D11.1 Częstotliwości pracy przetwornika A/C

## Konwersja A/C

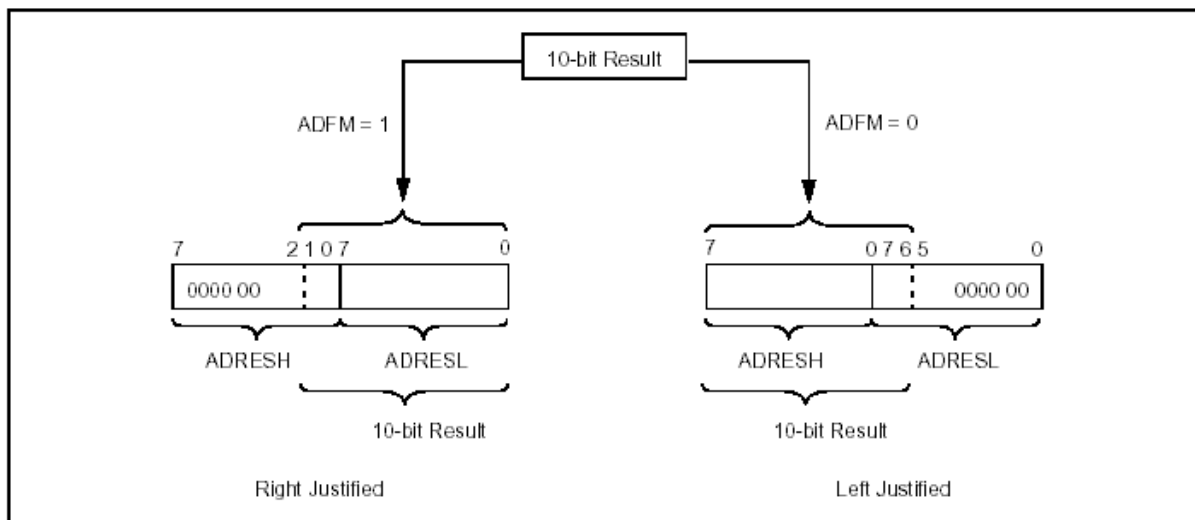
Wyzerowanie bitu GO/ $\sim$ DONE podczas przetwarzania spowoduje przerwanie konwersji. Para rejestrów ADRESH:ADRESL nie zostanie uaktualniona nowymi danymi, ale będzie zawierać dane z ostatniej konwersji (lub wartość ostatnio zapisaną do rejestrów ADRESH:ADRESL). Po tym, jak przetwarzanie zostało przerwane, należy odczekać czas 2TAD przed rozpoczęciem kolejnego przetwarzania. Po tym czasie przetwarzanie na wybranym kanale jest automatycznie rozpoczęte.

Bit GO/ $\sim$ DONE nie powinien być ustawiany w tej samej instrukcji, która uaktywnia (włącza) przetwornik A/D.



Rys. 11.3 Cykle konwersji A/C

W rejestrach ADRESH:ADRESL zapisywany jest 10-bitowy wynik przetwarzania A/D po zakończeniu konwersji. Para rejestrów ma wielkość 16 bitów. Moduł przetwornika daje możliwość justowania 10-bitowego wyniku w 16-bitowym obszarze. Bit ADFM kontroluje justowanie. Rysunek D11.6 pokazuje, w jaki sposób wykonywane jest justowanie. Bity dodatkowe są zapisywane jako '0'. Kiedy moduł przetwornika jest nieaktywny, rejestry te mogą być używane jako dwa 8-bitowe rejestry ogólnego przeznaczenia.



Rys. 11.6. Justowanie wyniku przetwarzania A/C

### Wyzwalanie przetwornika A/C od modułu CCP2

Przetwarzanie A/C może być rozpoczęte poprzez zdarzenie specjalne od modułu CCP2. Wymaga to, aby bity CCP2M3:CCP2M0 rejestru **CCP2CON<3:0>** = 1011, oraz aby przetwornik A/C był odblokowany ADON=1. Kiedy zdarzenie wystąpi bit GO/~DONE zostanie ustawiony rozpoczynając konwersję A/C, zerując tym samym stan licznika TMIER1 (lub TIMER3).

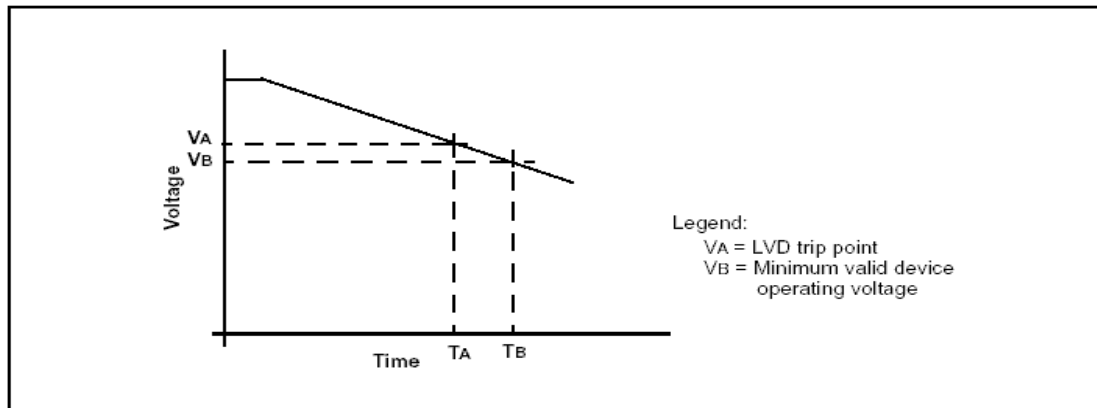
Jeżeli moduł A/C nie jest odblokowany ADON=0, wyzwolenie przetwornika A/C zdarzeniem specjalnym zostanie zignorowane, natomiast zerowanie licznika TIMER1 (lub TIMER2) będzie prawidłowo realizowane.

Nazwa	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Stany po: POR, BOR	Stany po pozostałych RESET-ach
<b>INTCON</b>	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
<b>PIR1</b>	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
<b>PIE1</b>	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
<b>IPR1</b>	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
<b>PIR2</b>	-	-	-	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	-- -0 0000	-- -0 0000
<b>PIE2</b>	-	-	-	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	-- -0 0000	-- -0 0000
<b>IPR2</b>	-	-	-	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	-- -1 1111	-- -1 0000
<b>ADRESH</b>	Rejestr wyniku przetwarzani A/D – starszy bajt								xxxx xxxx	uuuu uuuu
<b>ADRESL</b>	Rejestr wyniku przetwarzani A/D – młodszy bajt								xxxx xxxx	uuuu uuuu
<b>ADCON0</b>	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/ ~DONE	-	ADON	0000 00-0	0000 00-0
<b>ADCON1</b>	ADFM	ADCS2	-	-	PCFG3	PCFG2	PCFG1	PCFG0	0000 -000	0000 -000
<b>PORTA</b>	-	RA6	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
<b>TRISA</b>	-	Rejestr kierunkowy PORTA							--11 1111	--11 1111
<b>PORTE</b>	-	-	-	-	-	RE2	RE1	RE0	---- -000	---- -000
<b>LATE</b>	-	-	-	-	-	LATE2	LATE1	LATE0	---- -xxx	---- -uuu
<b>TRISE</b>	IBF	OBF	IBOV	PSP MODE	-	Rejestr kierunku PORTE			0000 -111	0000 -111

Tabela D11.2. Rejestry związane z modułem przetwornika A/C

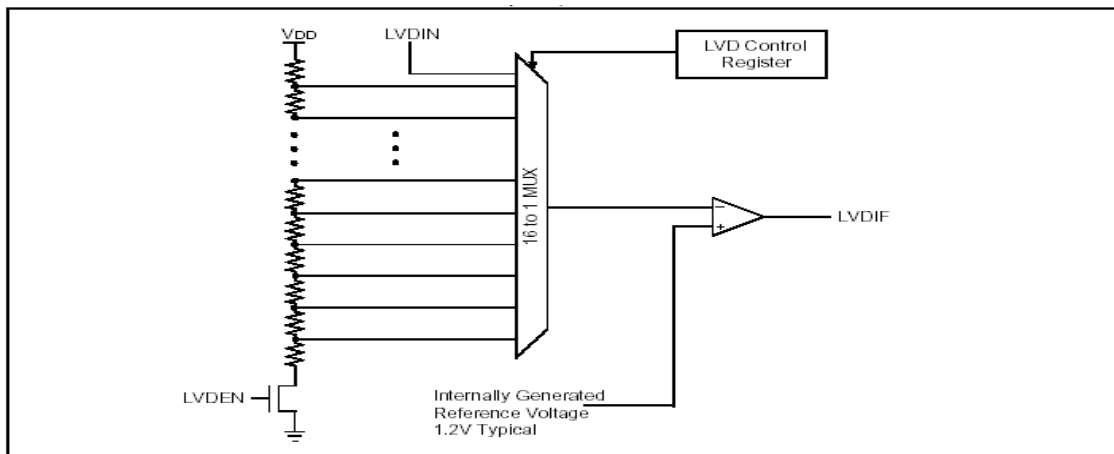
### 13 Detekcja zaniku napięcia LVD (Low Voltage Detect)

W wielu aplikacjach istnieje potrzeba określenia faktu spadku napięcia zasilania. Kontrolę taką umożliwia moduł LVD (*Low Voltage Detect*). Jest to programowalny obwód, w którym określa się poziom napięcia poniżej którego nastąpi ustawienie flagi przerwania. Moduł ten jest kontrolowany programowo.



Rys. 12.1. Sposób działania modułu LVD

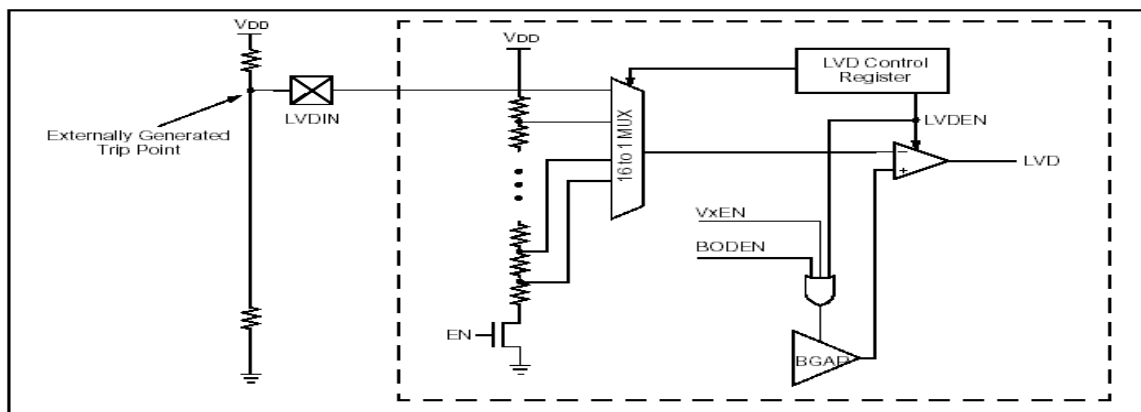
Gdy napięcie osiągnie poziom VA – moduł LVD ustawi flagę przerwania. Realizowana aplikacja ma jeszcze czas na dokończenie zaczętych operacji, zanim napięcie zasilania spadnie poniżej wartości umożliwiającej prawidłową pracę systemu - VB.



Rys. 12.2. Schemat blokowy układu LVD

Istnieje możliwość podłączenia zewnętrznego napięcia odniesienia do modułu LVD. Tryb ten jest dostępny przy ustawieniu bitów LVDL3:LVDL0 = 1111. Zewnętrzne napięcie podłączone jest do pinu LVDIN. Umożliwia to skonfigurowanie LVD na dowolne napięcie.





Rys. 12.3. Moduł LVD z zewnętrznym wejściem

## Rejestr LVDCON

(Rejestr *LVDCON* jest rejestrem kontrolnym modułu LVD)

U-0	U-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1
-	-	IRVST	LV DEN	LV DL3	LV DL2	LV DL1	LV DL0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bit 7-6

**bity niezaimplementowane,**

bit 5

**IRVST: flaga stabilności wewnętrznego napięcia:**

1 = sygnalizuje, że moduł LVD wygeneruje przerwanie przy ustawionym zakresie napięcia,

0 = sygnalizuje, że moduł LVD nie wygeneruje przerwania przy ustawionym zakresie napięcia, dodatkowo przerwanie od LVD nie powinno być odblokowane,

bit 4

**LV DEN: bit zasilania modułu LVD:**

1 = odblokowanie modułu LVD, zasilanie obwodu LVD,

0 = zablokowanie modułu LVD, odłączenie obwodu zasilania modułu LVD,

bit 3-0

**LV DL3: LV DL0: bity zakresów detekcji napięcia:**

1111 = używane zewnętrzne wejście analogowe (pin LVDIN),

1110 = 4.5V – 4.77V,

1101 = 4.2V – 4.45V,

1011 = 4.0V – 4.24V,

1011 = 3.8V – 4.03V,

1001 = 3.6V – 3.82V,

1010 = 3.5V – 3.71V,

1000 = 3.3V – 3.50V,

0111 = 3.0V – 3.18V,

0110 = 2.8V – 2.97V,

0101 = 2.7V – 2.86V,

0100 = 2.5V – 2.65V,

0011 = 2.4V – 2.54V,

0010 = 2.2V – 2.33V,

0001 = 2.0V – 2.12V,

0000 = zarezerwowane.