



Chương 3

Vi điều khiển PIC 18F8722

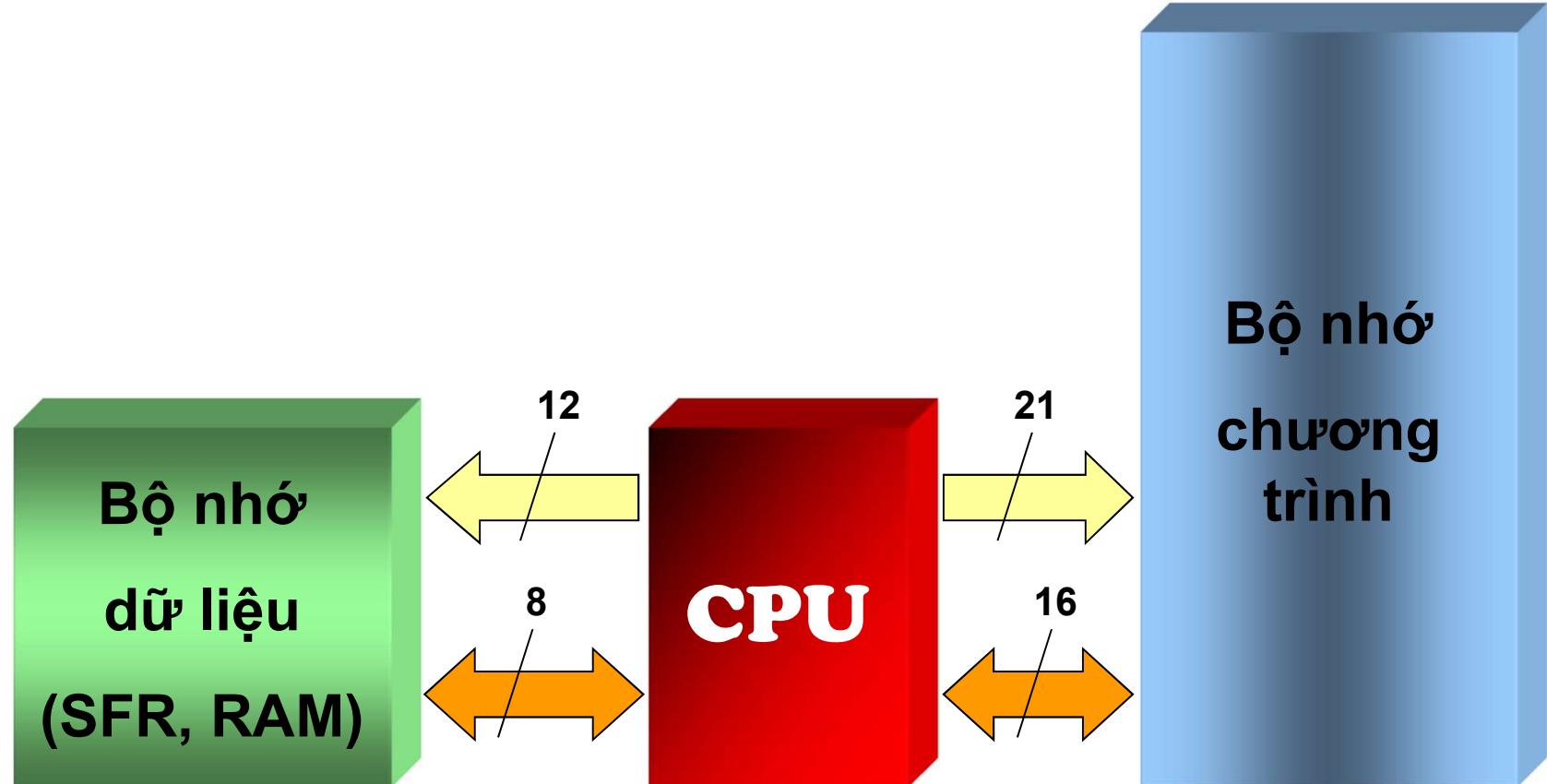
3.1 PIC family 18Fxxxx overview.

3.2 Features of PIC 18F8722.

3.3 Memory.



Kiến trúc Harvard

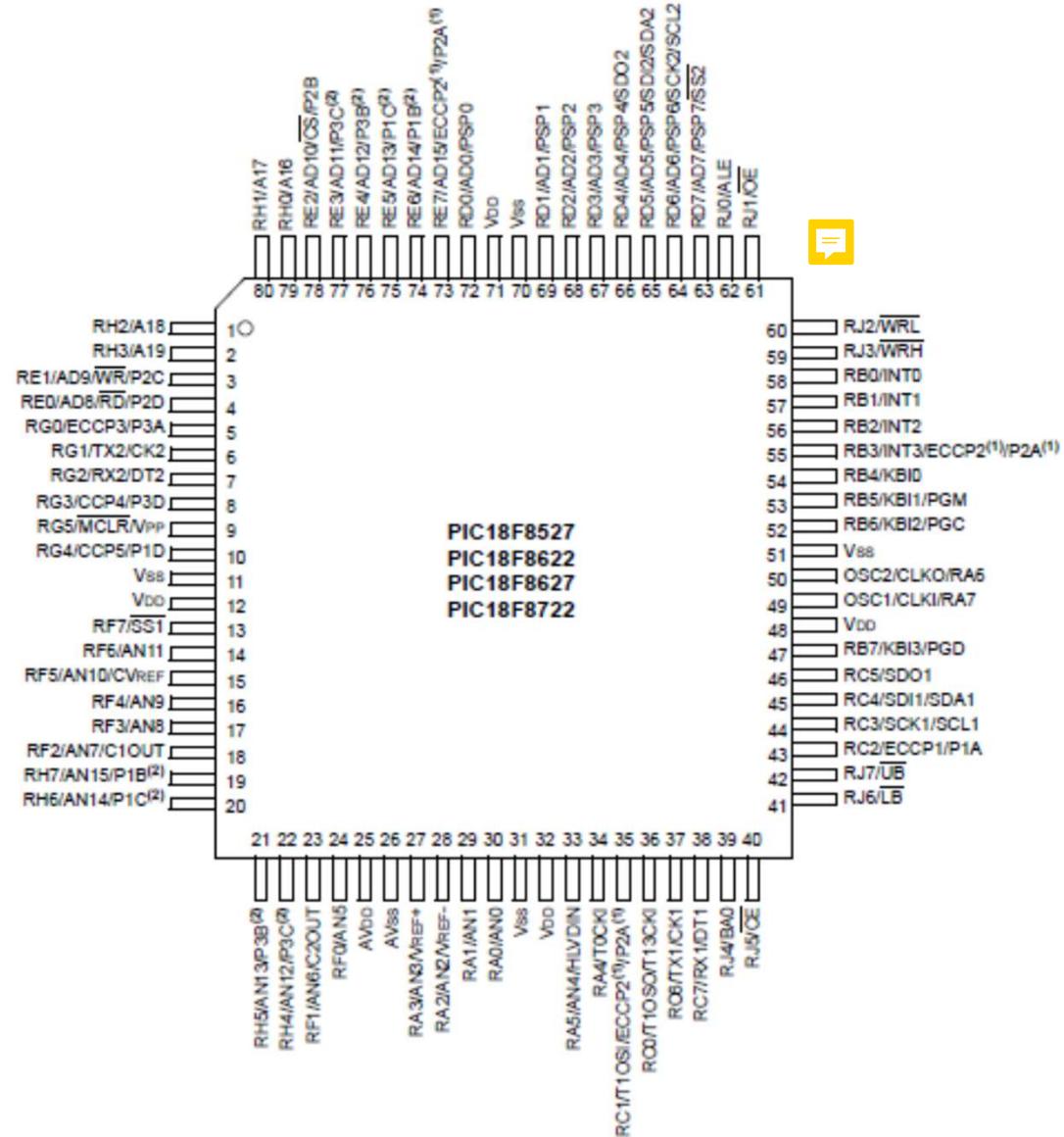


Sơ đồ khối PIC18F8722



Chân linh kiện MCU PIC18F8722

80 pin TQFD



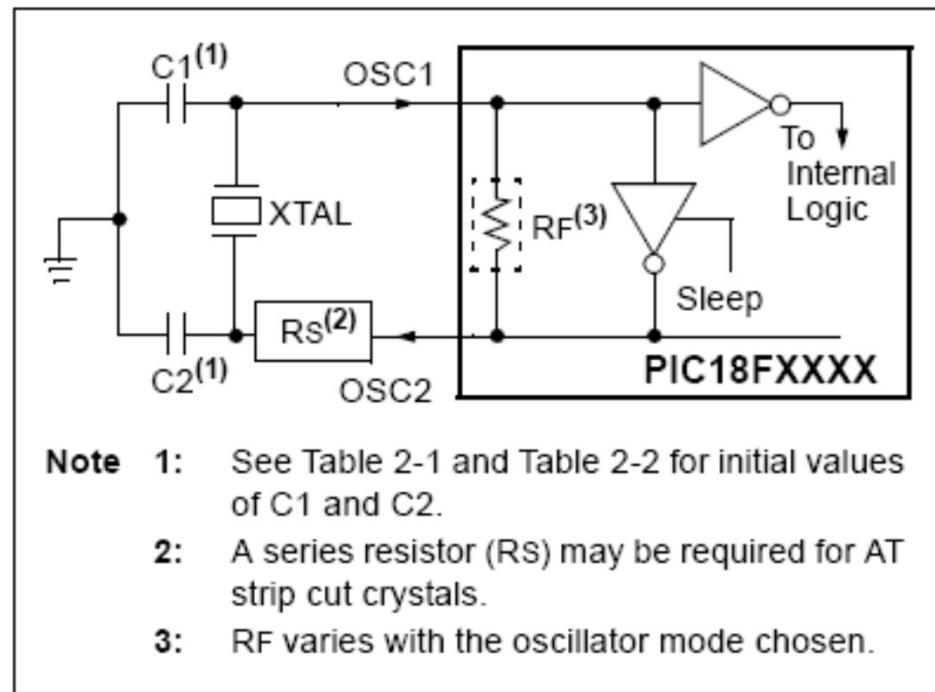
Đặc điểm PIC18F8722



Nguồn dao động

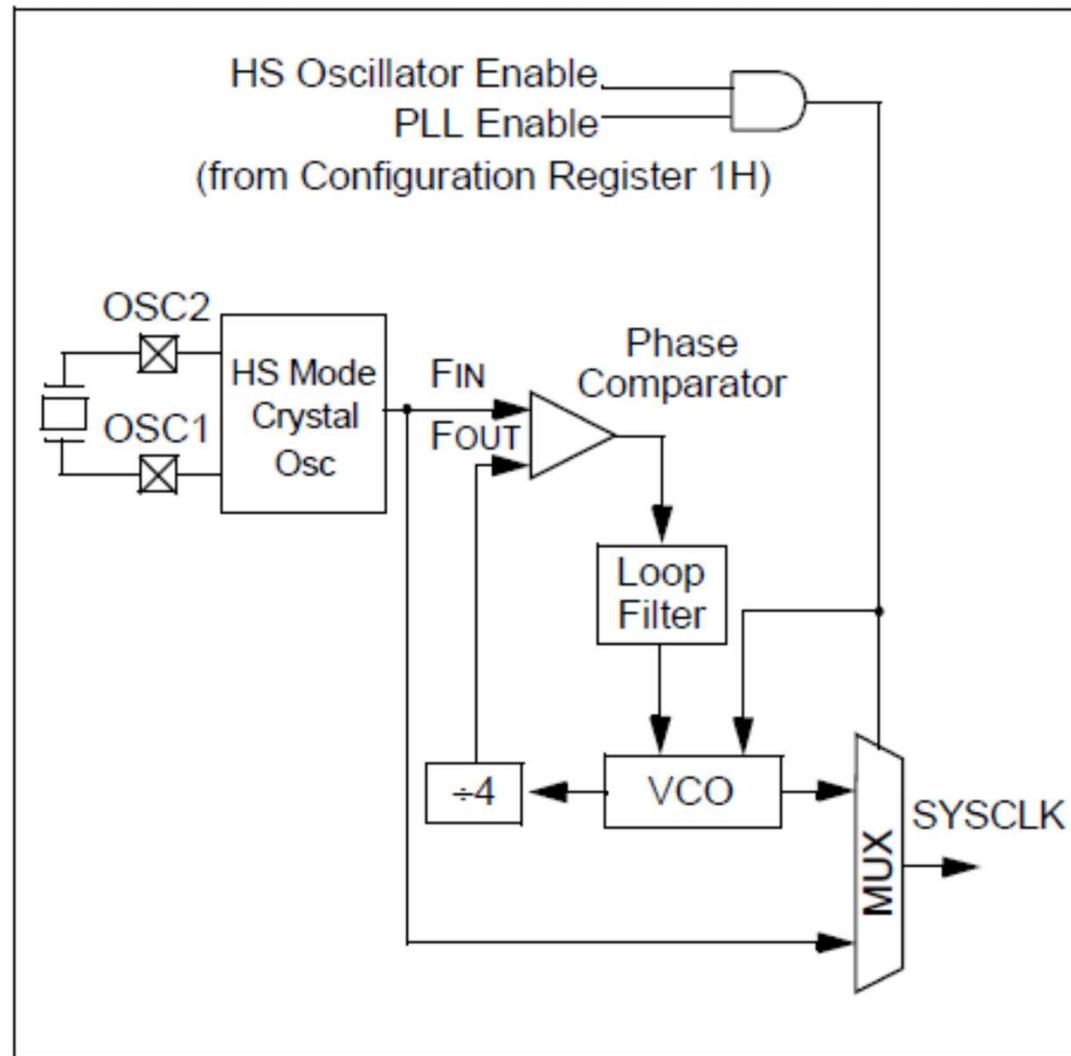
- PIC 18F8722 có 10 lựa chọn nguồn dao động
 1. LP Low-Power Crystal
 2. XT Crystal/Resonator
 3. HS High-Speed Crystal/Resonator
 4. HSPLL High-Speed Crystal/Resonator with PLL Enabled
 5. RC External Resistor/Capacitor with $F_{osc}/4$ Output on RA6
 6. RCIO External Resistor/Capacitor with I/O on RA6
 7. INTIO1 Internal Oscillator with $F_{osc}/4$ Output on RA6 and I/O on RA7
 8. INTIO2 Internal Oscillator with I/O on RA6 and RA7
 9. EC External Clock with $F_{osc}/4$ Output
 10. ECIO External Clock with I/O on RA6

The XT, LP, HS oscillators (bên trong)



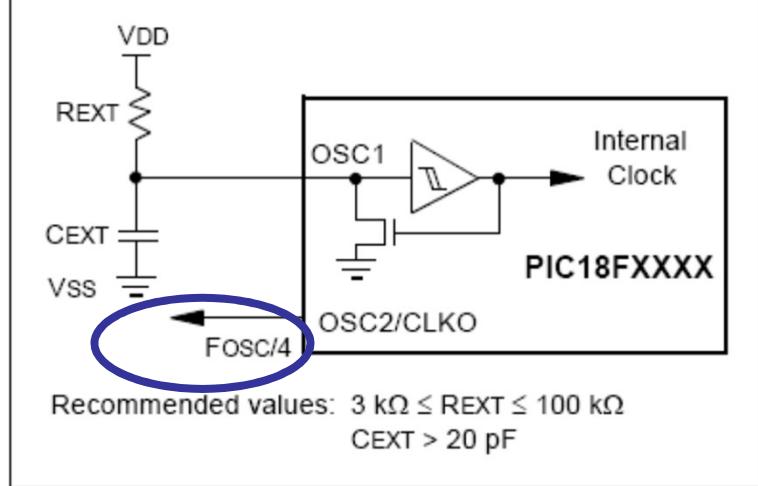
The HSPLL oscillators (nhân tần số bên trong)

FIGURE 2-7: HSPLL BLOCK DIAGRAM

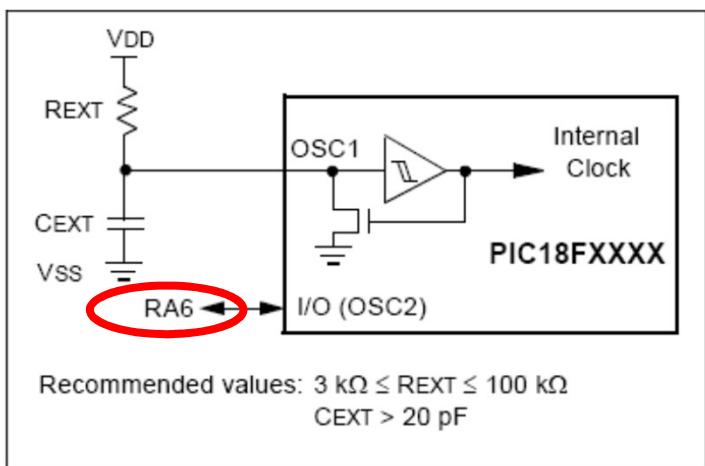


The RC and RCIO oscillators (bên trong)

RC

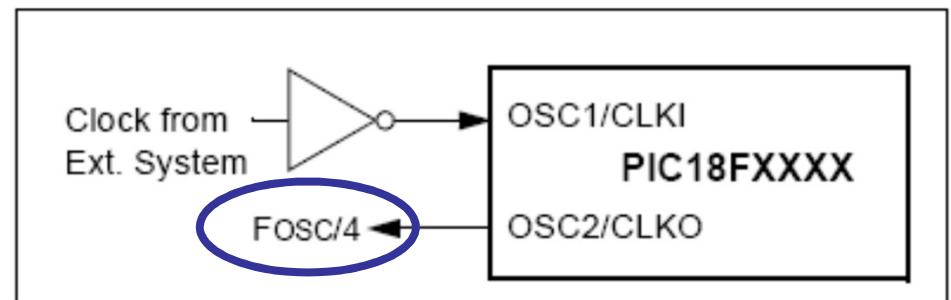


RCIO

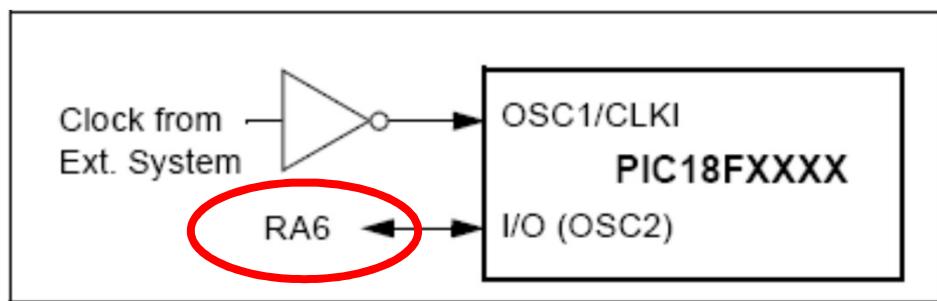


The EC and ECIO oscillators (bên ngoài)

EC

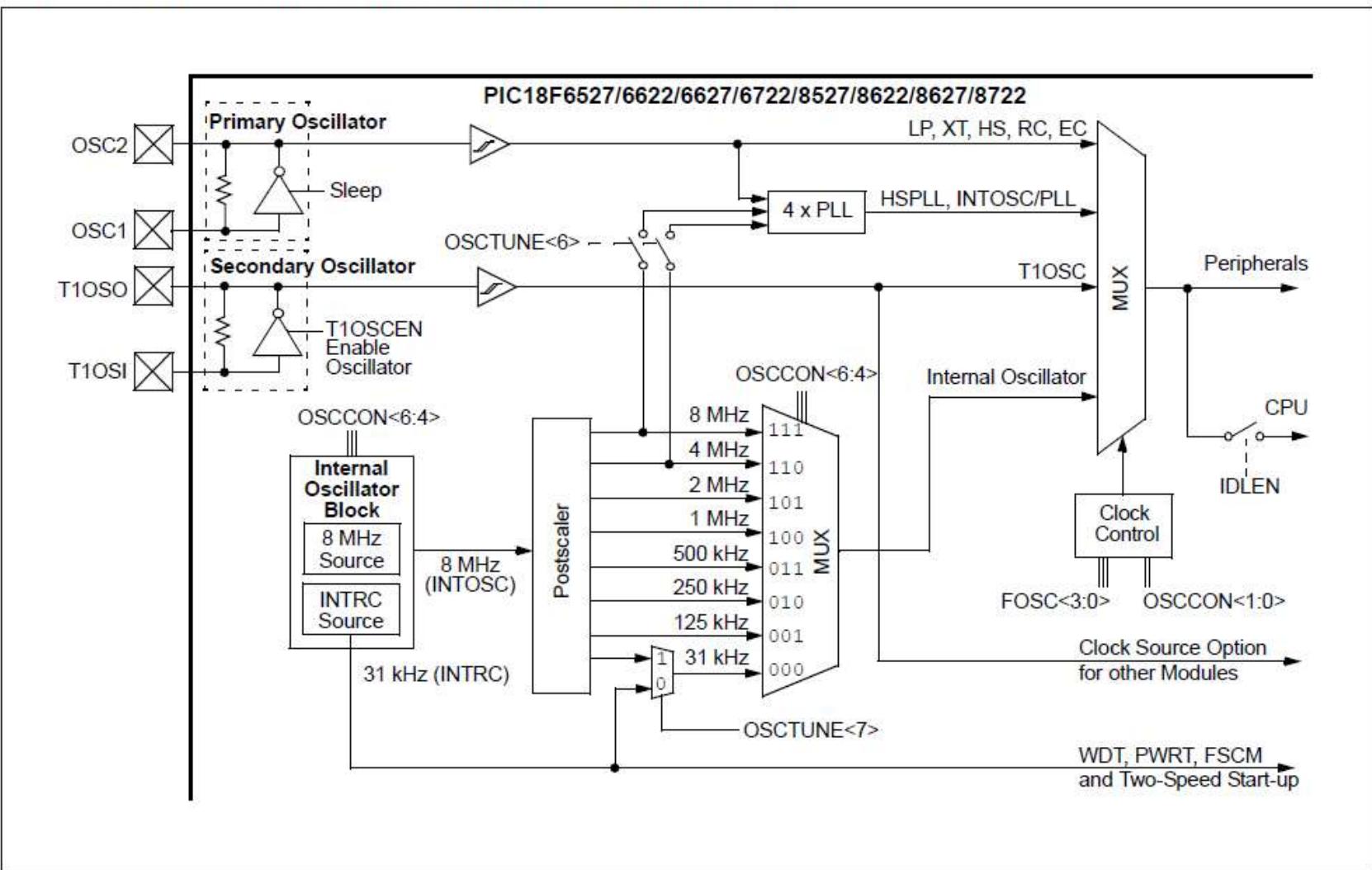


ECIO



Mạch tạo xung clock của PIC18F8722

FIGURE 2-11: PIC18F8722 FAMILY CLOCK DIAGRAM



Oscillator control register (OSCCON)

R/W-0	R/W-0	R/W-0	R/W-0	R ⁽¹⁾	R-0	R/W-0	R/W-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0

bit 7	IDLEN: Idle Enable bit 1 = Device enters an Idle mode when a SLEEP instruction is executed 0 = Device enters Sleep mode when a SLEEP instruction is executed
bit 6-4	IRCF<2:0>: Internal Oscillator Frequency Select bits ⁽⁵⁾ 111 = 8 MHz (INTOSC drives clock directly) 110 = 4 MHz 101 = 2 MHz 100 = 1 MHz ⁽³⁾ 011 = 500 kHz 010 = 250 kHz 001 = 125 kHz 000 = 31 kHz (from either INTOSC/256 or INTRC directly) ⁽²⁾
bit 3	OSTS: Oscillator Start-up Time-out Status bit ⁽¹⁾ 1 = Oscillator Start-up Timer (OST) time-out has expired; primary oscillator is running 0 = Oscillator Start-up Timer (OST) time-out is running; primary oscillator is not ready
bit 2	IOFS: INTOSC Frequency Stable bit 1 = INTOSC frequency is stable 0 = INTOSC frequency is not stable
bit 1-0	SCS<1:0>: System Clock Select bits ⁽⁴⁾ 1x = Internal oscillator block 01 = Secondary (Timer1) oscillator 00 = Primary oscillator

- Note 1:** Reset state depends on state of the IESO Configuration bit.
- 2:** Source selected by the INTSRC bit (OSCTUNE<7>), see text.
- 3:** Default output frequency of INTOSC on Reset.
- 4:** Modifying the SCS<1:0> bits will cause an immediate clock source switch.
- 5:** Modifying the IRCF<3:0> bits will cause an immediate clock frequency switch if the internal oscillator is providing the device clocks.

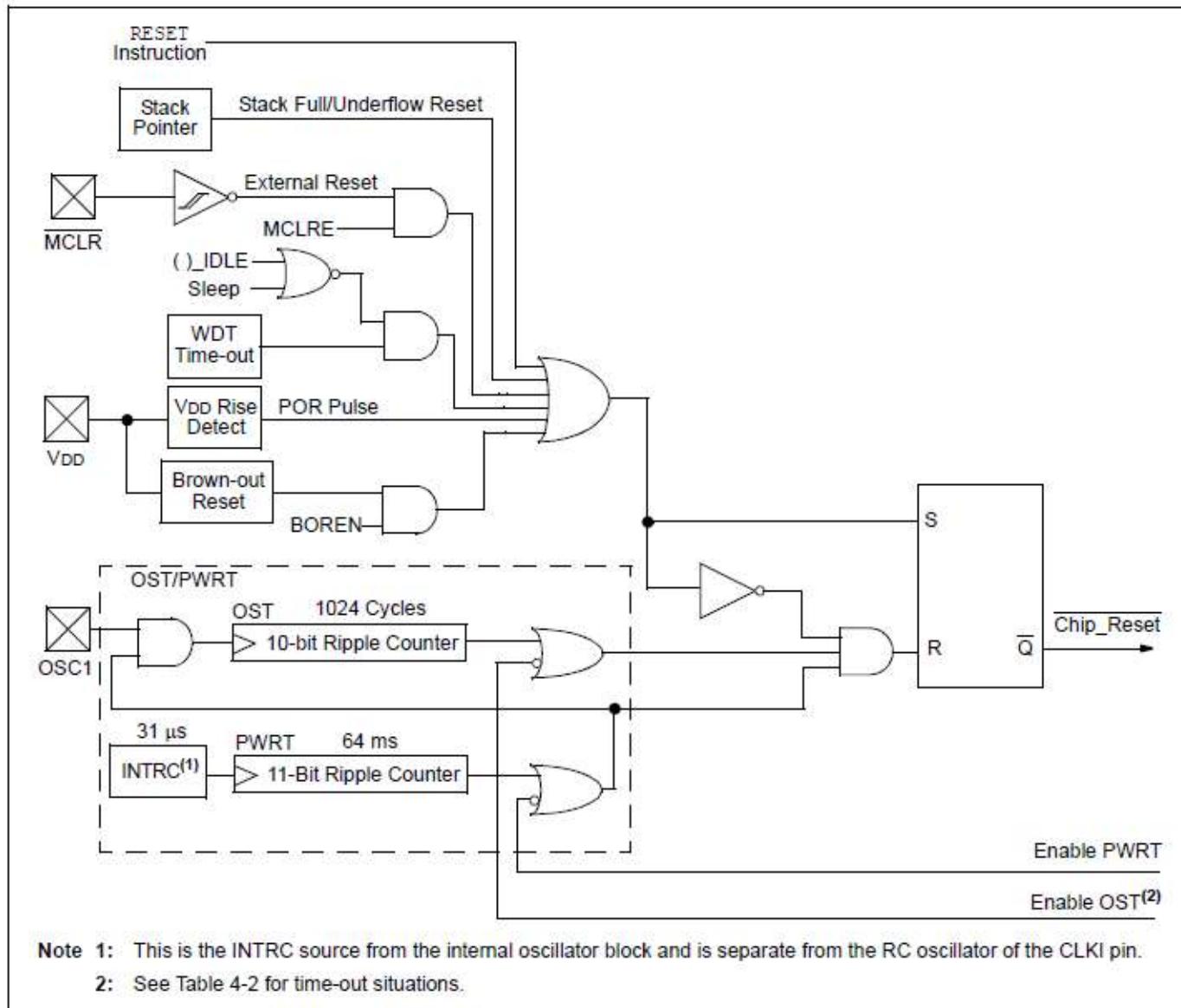
Lập trình chọn dao động

```
#include <xc.h>
#pragma config OSC=HS, WDT=OFF, LVP=OFF
// Che do dao dong High Speed, tan so = 10 MHz
```

Các tác nhân gây ra Reset

1. Power-on Reset (POR).
2. MCLR Reset during normal operation.
3. MCLR Reset during power-managed modes.
4. Watchdog Timer (WDT) Reset (during execution).
5. Programmable Brown-out Reset (BOR).
6. RESET Instruction.
7. Stack Full Reset.
8. Stack Underflow Reset.

Sơ đồ mạch reset



Thanh ghi RCON

REGISTER 4-1: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1 ⁽¹⁾	U-0	R/W-1	R-1	R-1	R/W-0 ⁽²⁾	R/W-0
IPEN	SBOREN	—	RI	TO	PD	POR	BOR
bit 7							

- bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)
- bit 6 **SBOREN:** BOR Software Enable bit⁽¹⁾
If BOREN<1:0> = 01:
1 = BOR is enabled
0 = BOR is disabled
If BOREN<1:0> = 00, 10 or 11:
Bit is disabled and read as '0'
- bit 5 **Unimplemented:** Read as '0'

Thanh ghi RCON

REGISTER 4-1: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1 ⁽¹⁾	U-0	R/W-1	R-1	R-1	R/W-0 ⁽²⁾	R/W-0
IPEN	SBOREN	—	<u>RI</u>	<u>TO</u>	<u>PD</u>	<u>POR</u>	<u>BOR</u>
bit 7							bit 0

bit 4

RI: RESET Instruction Flag bit

1 = The RESET instruction was not executed (set by firmware only)
 0 = The RESET instruction was executed causing a device Reset (must be set in software after a Brown-out Reset occurs)

bit 3

TO: Watchdog Time-out Flag bit

1 = Set by power-up, CLRWDT instruction or SLEEP instruction
 0 = A WDT time-out occurred

bit 2

PD: Power-down Detection Flag bit

1 = Set by power-up or by the CLRWDT instruction
 0 = Set by execution of the SLEEP instruction

bit 1

POR: Power-on Reset Status bit⁽²⁾

1 = A Power-on Reset has not occurred (set by firmware only)
 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0

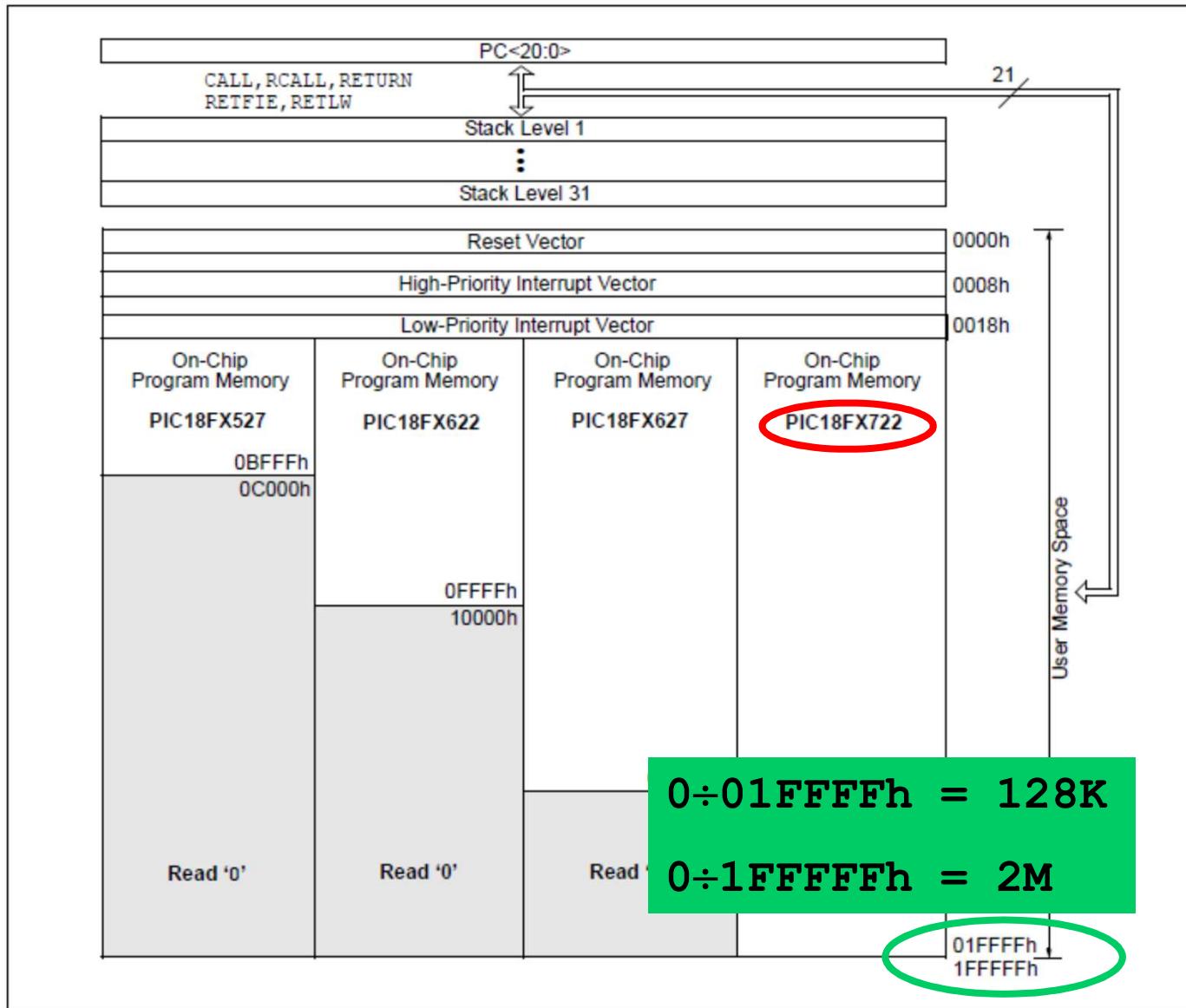
BOR: Brown-out Reset Status bit

1 = A Brown-out Reset has not occurred (set by firmware only)
 0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

Note 1: If SBOREN is enabled, its Reset state is '1'; otherwise, it is '0'.

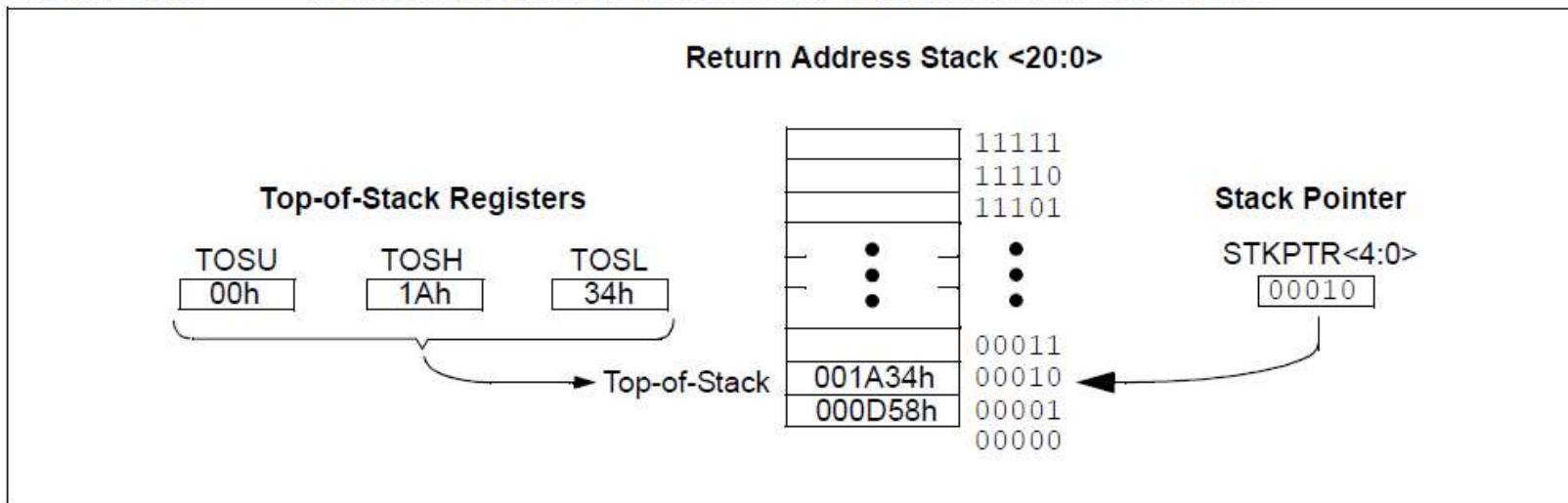
2: The actual Reset value of POR is determined by the type of device Reset. See the notes following this register and **Section 4.6 “Reset State of Registers”** for additional information.

Bản đồ bộ nhớ chương trình và Stack



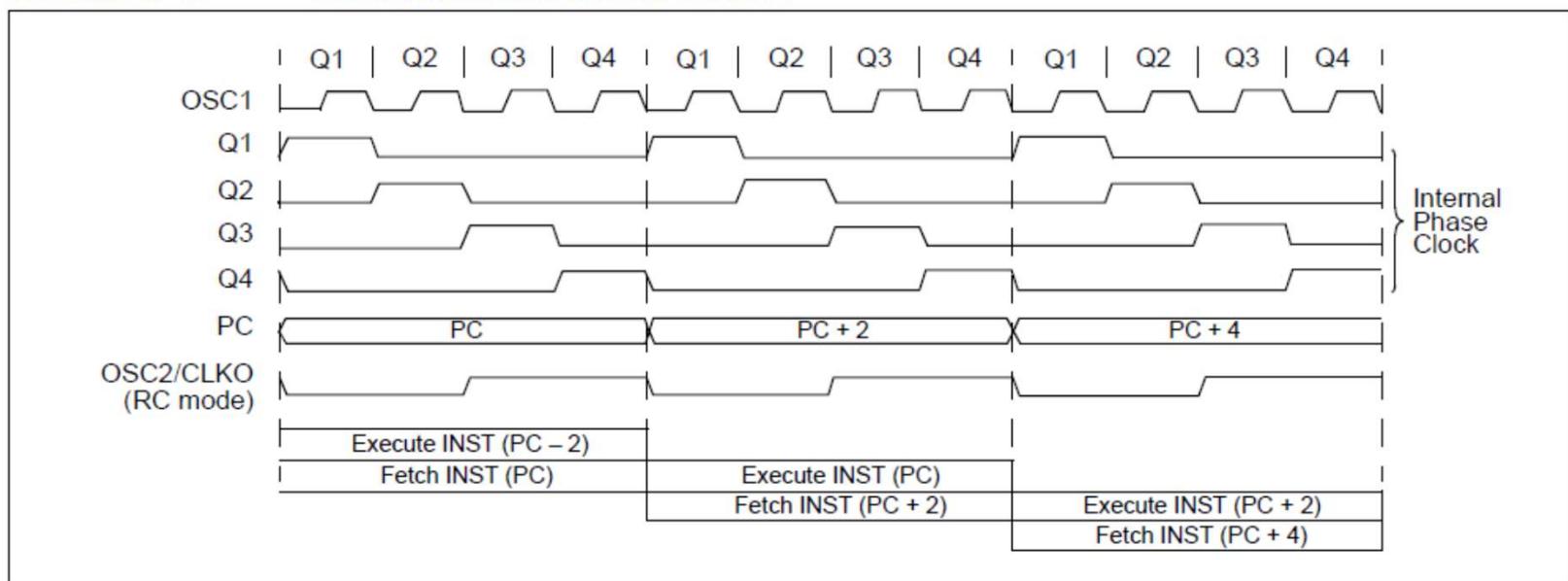
Thanh ghi TOS và STKPTR

FIGURE 5-3: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



Giản đồ xung lấy lệnh

FIGURE 5-4: CLOCK/INSTRUCTION CYCLE



Cơ chế lấy lệnh/thi hành lệnh Pipeline

EXAMPLE 5-3: INSTRUCTION PIPELINE FLOW

	TCY0	TCY1	TCY2	TCY3	TCY4	TCY5
1. MOVLW 55h	Fetch 1	Execute 1				
2. MOVWF PORTB		Fetch 2	Execute 2			
3. BRA SUB_1			Fetch 3	Execute 3		
4. BSF PORTA, BIT3 (Forced NOP)				Fetch 4	Flush (NOP)	
5. Instruction @ address SUB_1					Fetch SUB_1	Execute SUB_1

All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

Lưu trữ lệnh trong bộ nhớ chương trình

FIGURE 5-5: INSTRUCTIONS IN PROGRAM MEMORY

Program Memory Byte Locations →		LSB = 1	LSB = 0	Word Address ↓
Instruction 1:	MOVLW 055h			000000h
Instruction 2:	GOTO 0006h	0Fh	55h	000008h
Instruction 3:	MOVFF 123h, 456h	EFh	03h	0000Ah
		F0h	00h	0000Ch
		C1h	23h	0000Eh
		F4h	56h	000010h
				000012h
				000014h

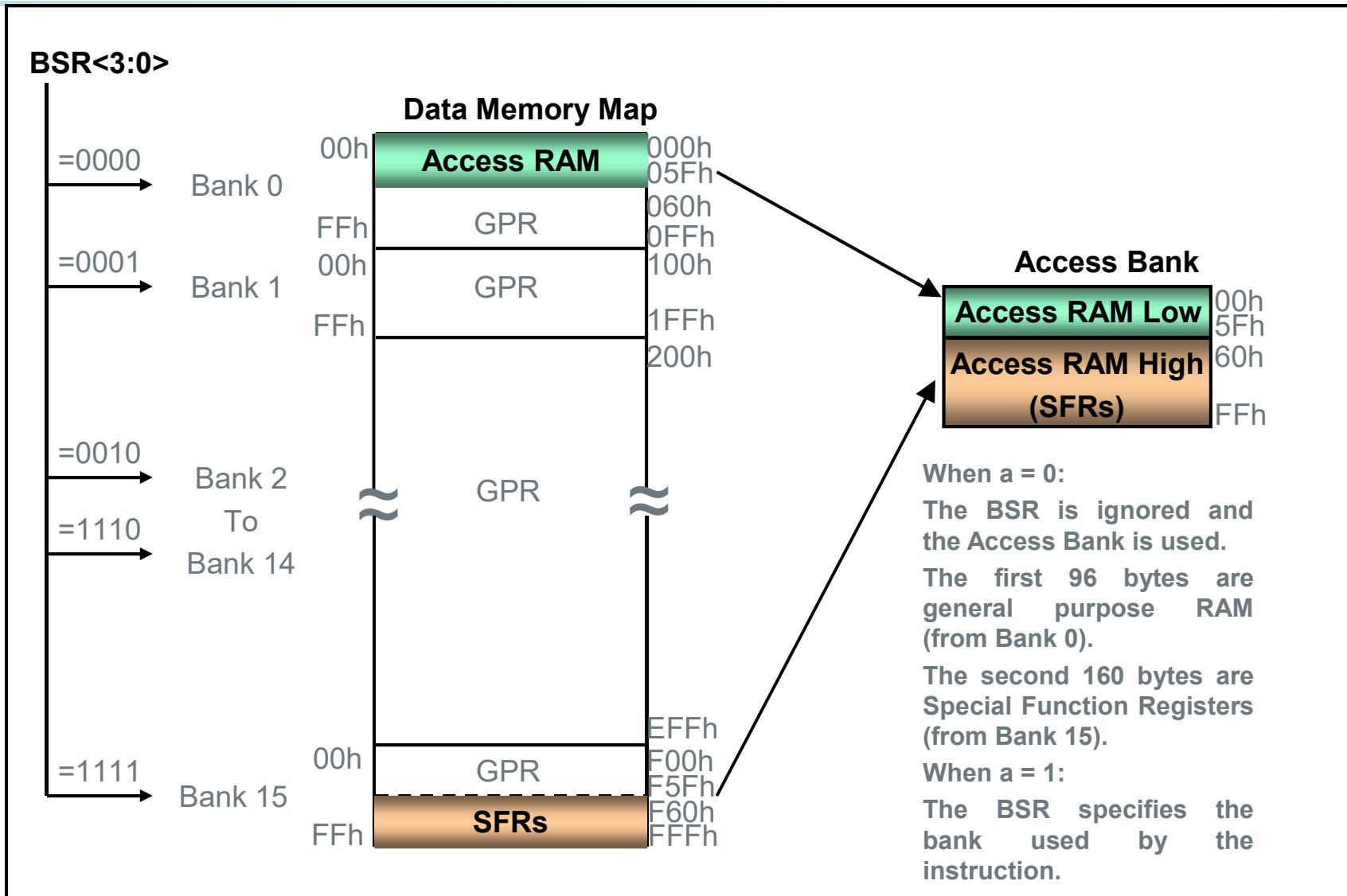
Lệnh 2 từ

EXAMPLE 5-4: TWO-WORD INSTRUCTIONS

CASE 1:			
Object Code	Source Code		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; No, skip this word
1111 0100 0101 0110			; Execute this word as a NOP
0010 0100 0000 0000	ADDWF	REG3	; continue code

CASE 2:			
Object Code	Source Code		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; Yes, execute this word
1111 0100 0101 0110			; 2nd word of instruction
0010 0100 0000 0000	ADDWF	REG3	; continue code

Bộ nhớ dữ liệu



Địa chỉ SFRs dùng chung với RAM

TABLE 5-2: SPECIAL FUNCTION REGISTER MAP FOR THE PIC18F8722 FAMILY OF DEVICES

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2 ⁽¹⁾	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	SPBRGH1
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	BAUDCON1
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	SPBRGH2
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	CCPR2H	F9Ch	MEMCON	F7Ch	BAUDCON2
FFBh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCPR2L	F9Bh	OSCTUNE	F7Bh	__(2)
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	TRISJ ⁽³⁾	F7Ah	__(2)
FF9h	PCL	FD9h	FSR2L	FB9h	CCPR3H	F99h	TRISH ⁽³⁾	F79h	ECCP1DEL
FF8h	TBLPTRU	FD8h	STATUS	FB8h	CCPR3L	F98h	TRISG	F78h	TMR4
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	CCP3CON	F97h	TRISF	F77h	PR4
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS	F96h	TRISE	F76h	T4CON
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD	F75h	CCPR4H
FF4h	PRODH	FD4h	__(2)	FB4h	CMCON	F94h	TRISC	F74h	CCPR4L
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB	F73h	CCP4CON
FF2h	INTCON	FD2h	HLVDCON	FB2h	TMR3L	F92h	TRISA	F72h	CCPR5H
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	LATJ ⁽³⁾	F71h	CCPR5L
FF0h	INTCON3	FD0h	RCON	FB0h	PSPCON	F90h	LATH ⁽³⁾	F70h	CCP5CON
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H	FAFh	SPBRG1	F8Fh	LATG	F6Fh	SPBRG2

Bộ xử lý ngắt quãng - Interrupts

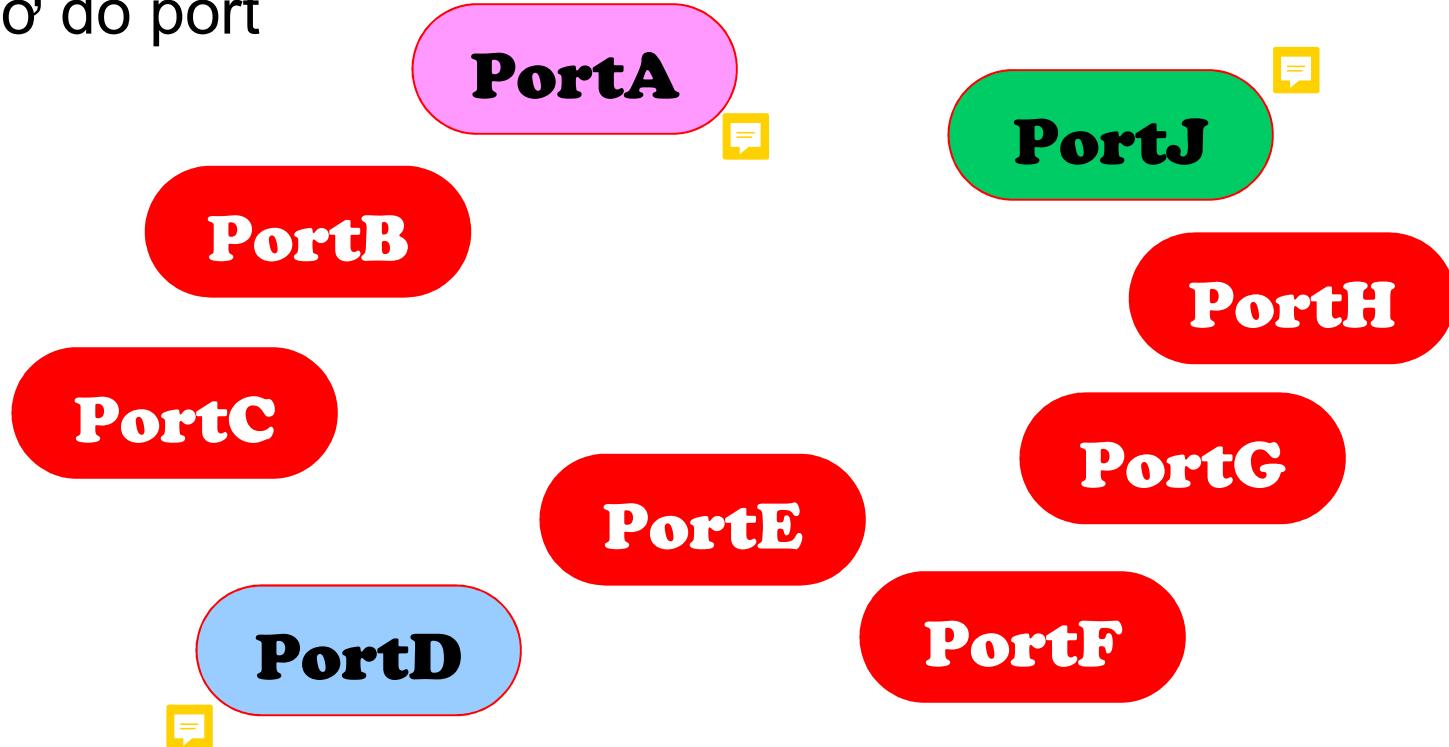
- Có 2 vector ngắt :
 - Vector ngắt ưu tiên cao dùng địa chỉ 000008h.
 - Vector ngắt ưu tiên thấp dùng địa chỉ 000018h..
- Khi 1 ngắt hợp lệ xảy ra, chương trình đang thực hiện sẽ tạm dừng và nhảy đến vector ngắt tương ứng và bit Global Interrupt Enable (GIE, GIEH hoặc GIEL) sẽ tự động xóa (cấm ngắt).

Sơ đồ nguồn ngắt



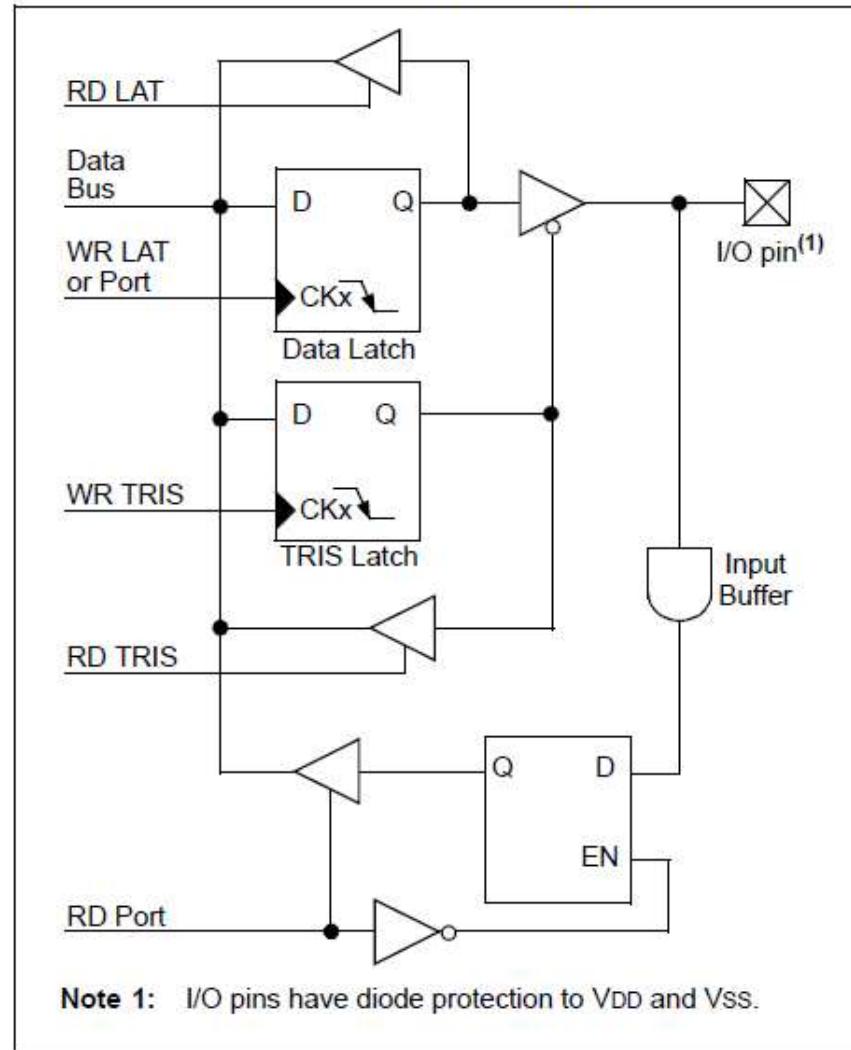
Cổng xuất/nhập

- PIC 18F8722 có các cổng đặt tên từ A đến J.
- Sơ đồ port



Mạch của 1 Port xuất/nhập tổng quát

FIGURE 11-1: GENERIC I/O PORT OPERATION

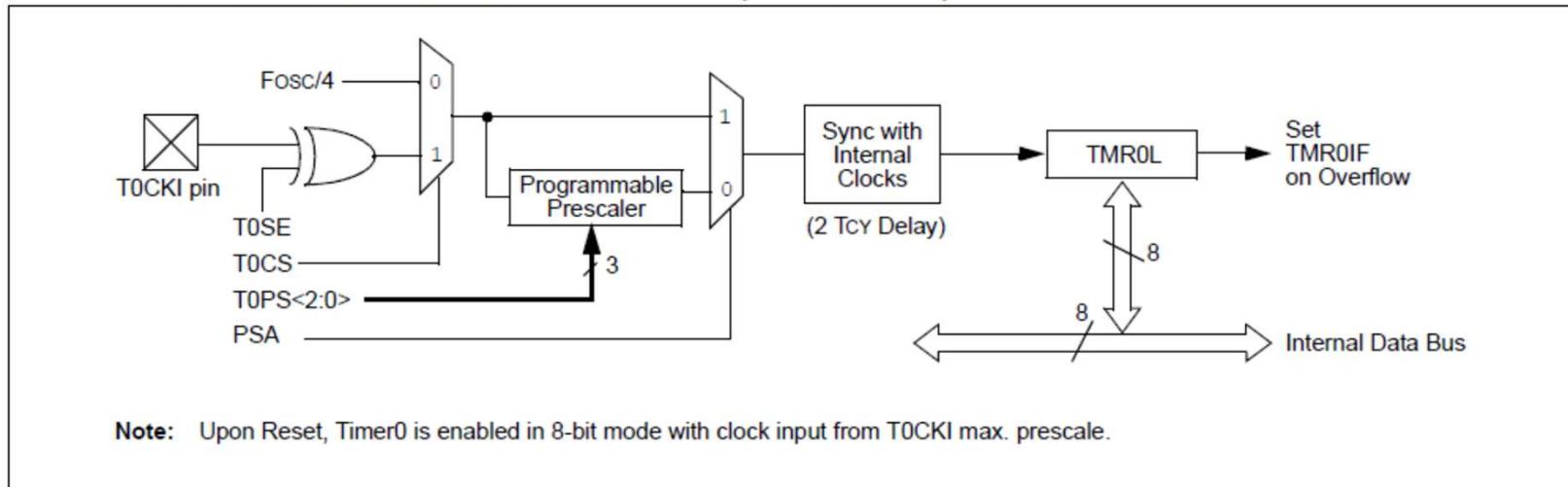


Bộ định thi

- Timer0 : dùng cho định thi trong chương trình.
- Timer1, Timer3 : dùng cho compare, capture. Có nguồn clock riêng dùng cho chế độ RTC (real time clock).
- Timer2, Timer4 : dùng cho PWM.

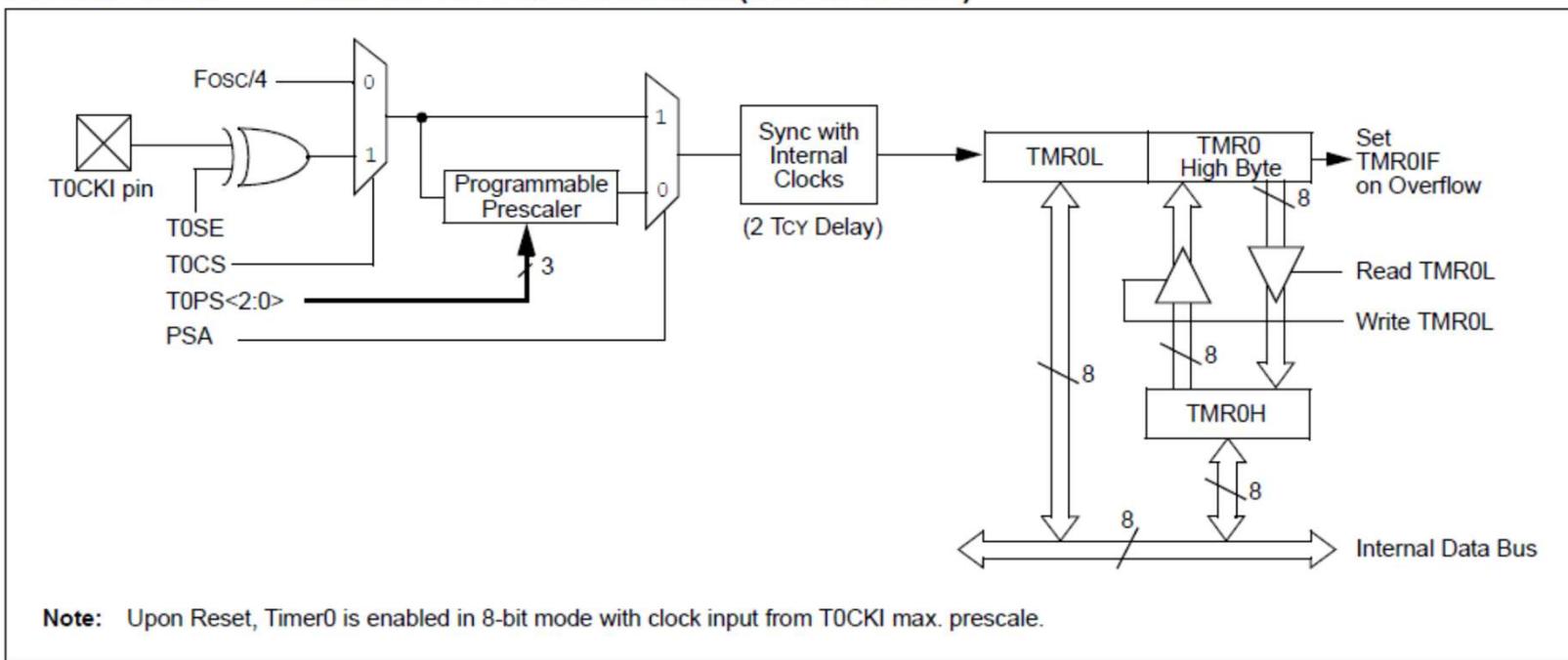
Chế độ 8 bit Timero

FIGURE 12-1: TIMER0 BLOCK DIAGRAM (8-BIT MODE)



Chế độ 16 bit Timer0

FIGURE 12-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)

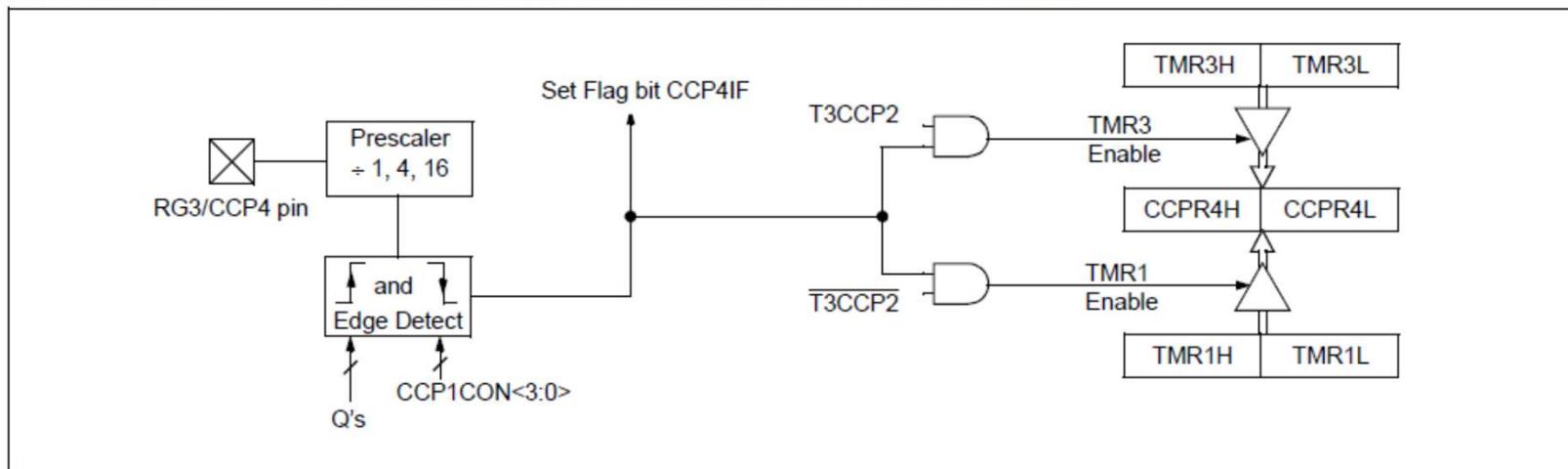


Module CCP (Capture/Compare/PWM)

- Thanh ghi điều khiển 8-bit (CCPxCON)
- Thanh ghi 16-bit (CCPRxH:CCPRxL) hoạt động như:
 - Thanh ghi Capture 16-bit.
 - Thanh ghi Compare 16-bit.
 - Thanh ghi 16-bit PWM Master/Slave duty cycle.

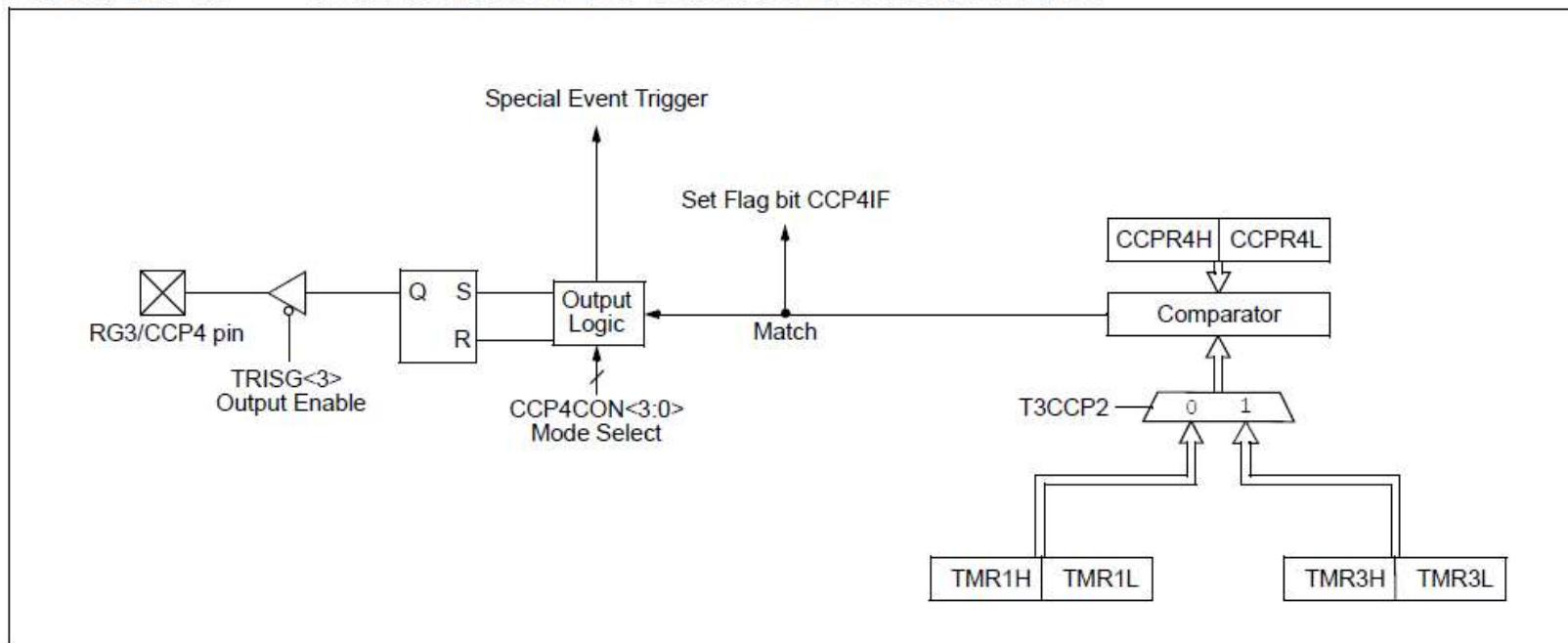
Chế độ Capture

FIGURE 17-2: CAPTURE MODE OPERATION BLOCK DIAGRAM

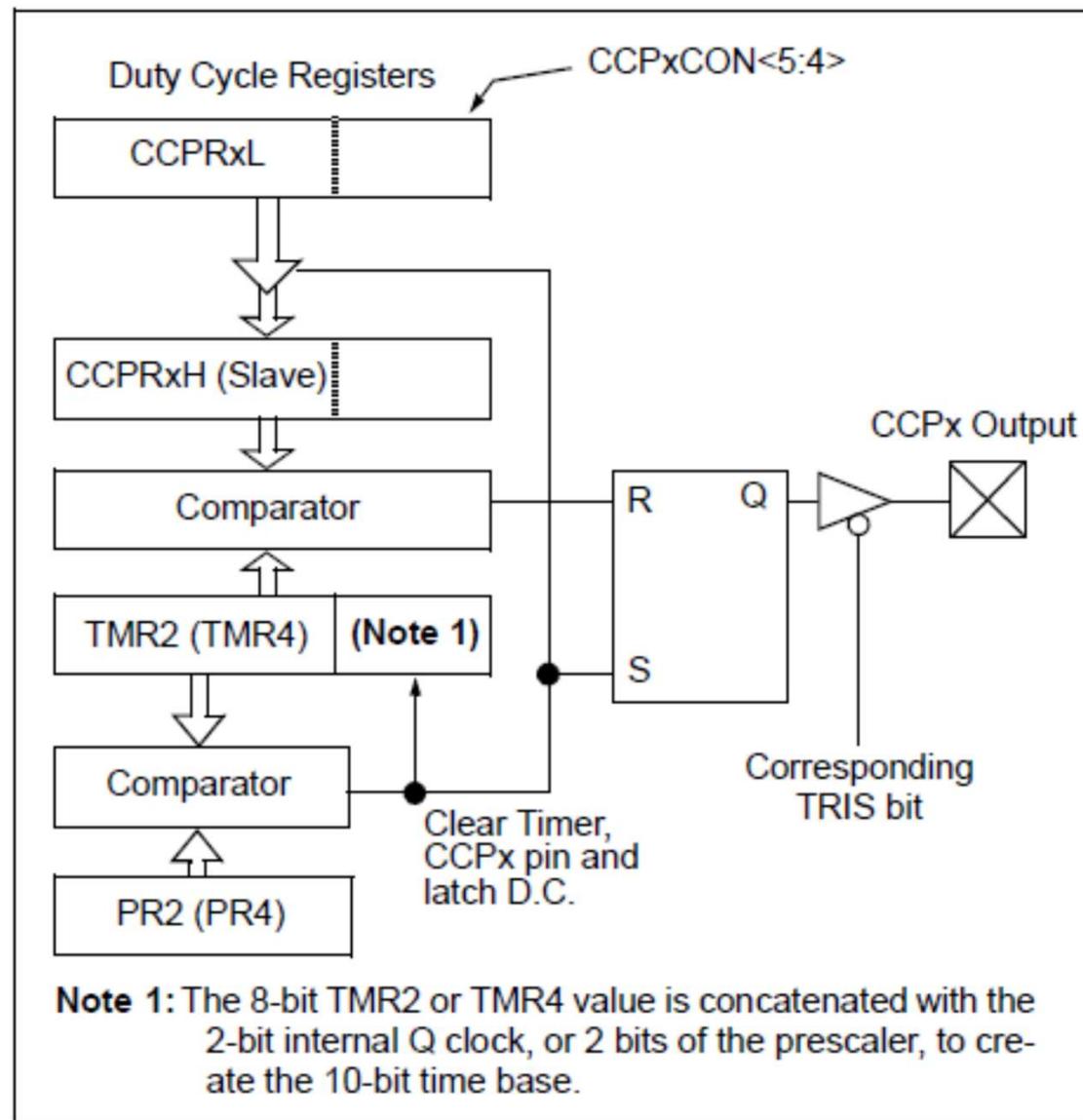


Chế độ Compare

FIGURE 17-3: COMPARE MODE OPERATION BLOCK DIAGRAM

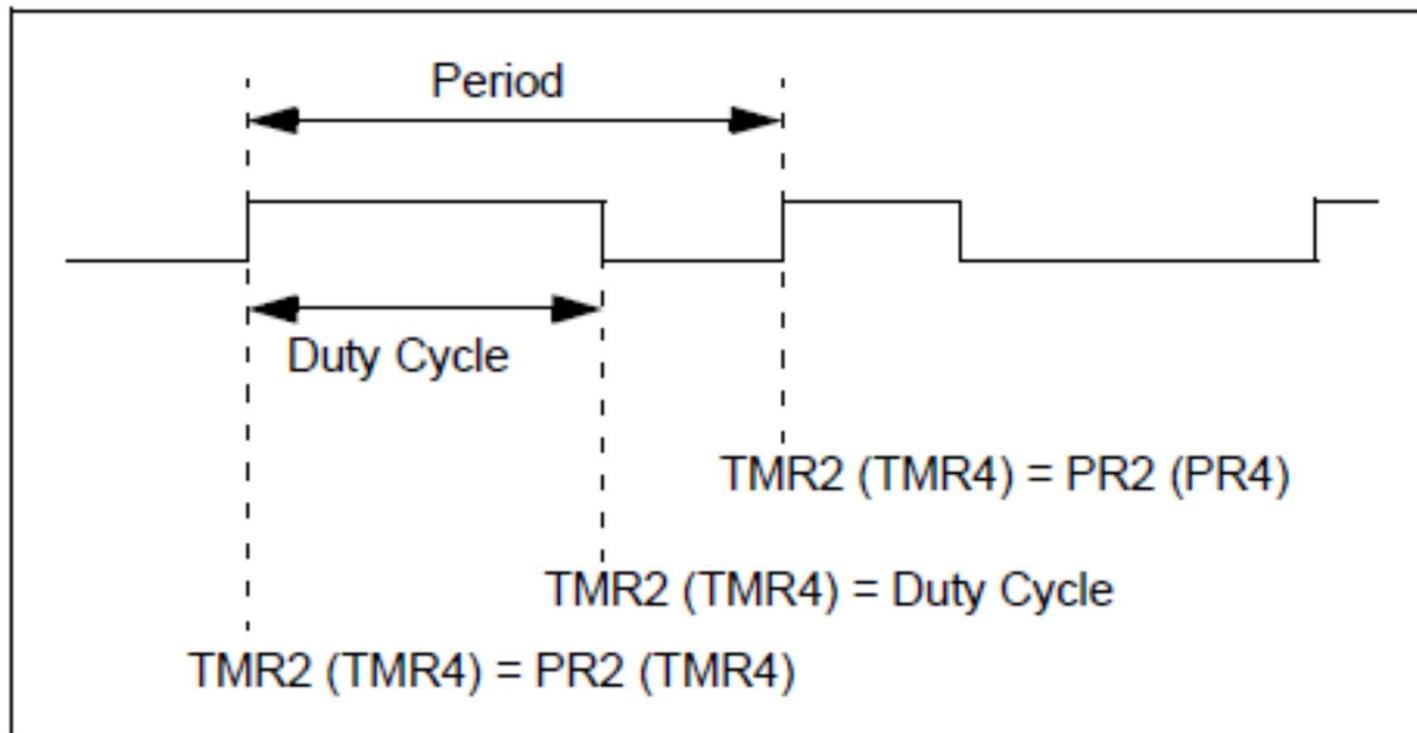


Sơ đồ khối mô hình hoạt động PWM



Dạng sóng PWM ra

FIGURE 17-5: PWM OUTPUT

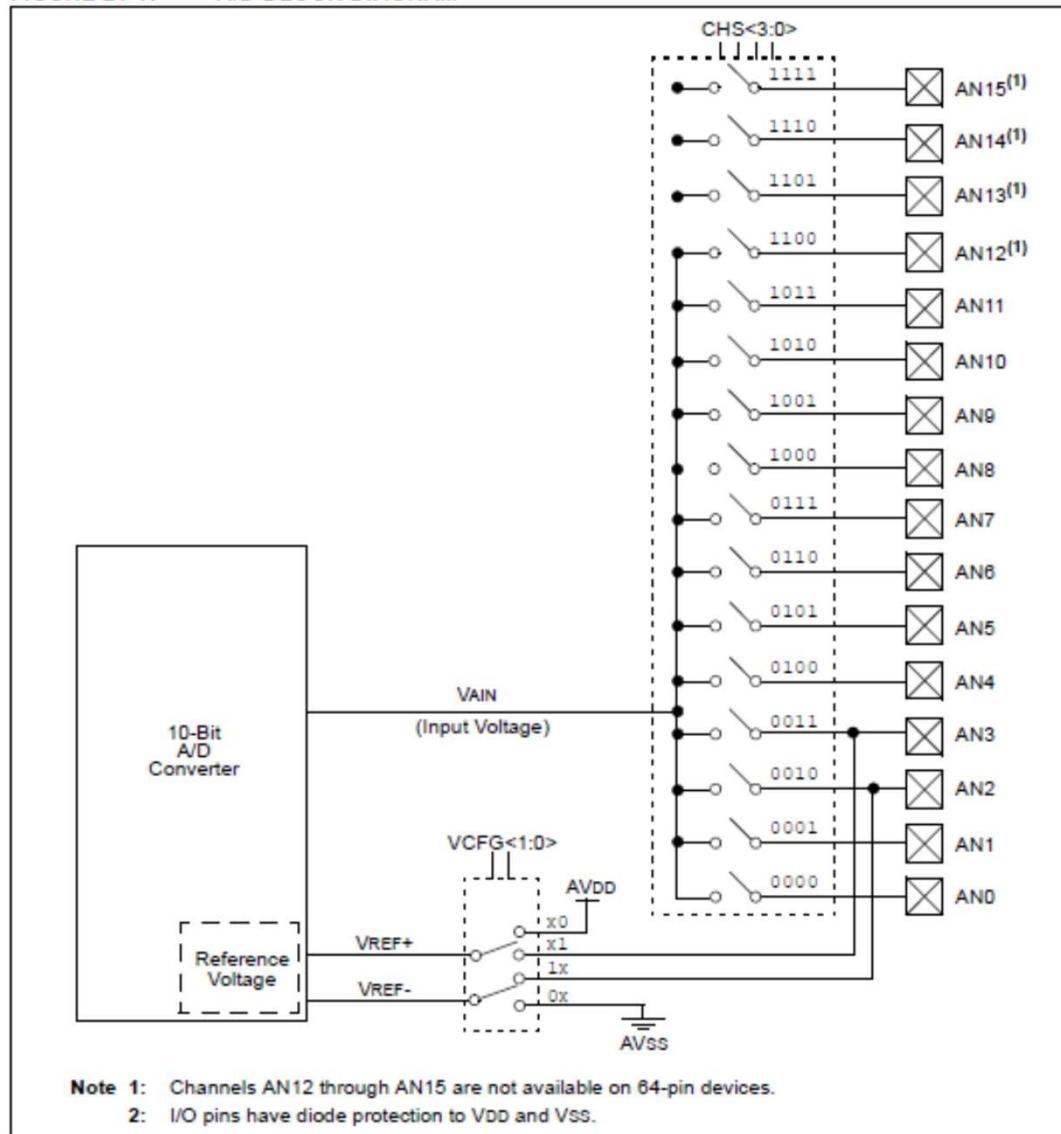


Bộ biến đổi A/D 10-bit

- Có thể cho phép đến 16 tín hiệu tương tự làm ngõ nhập (analog input).
- Cứ mỗi lần analog input nạp một mẫu và giữ trong tụ điện, sau đó output của mẫu này được dùng để làm input cho bộ biến đổi.
- Bộ biến đổi tạo ra tín hiệu số bằng cách xấp xỉ các mức analog.
- Kết quả của nó là tạo ra các kết quả số tương ứng với số 10-bit.

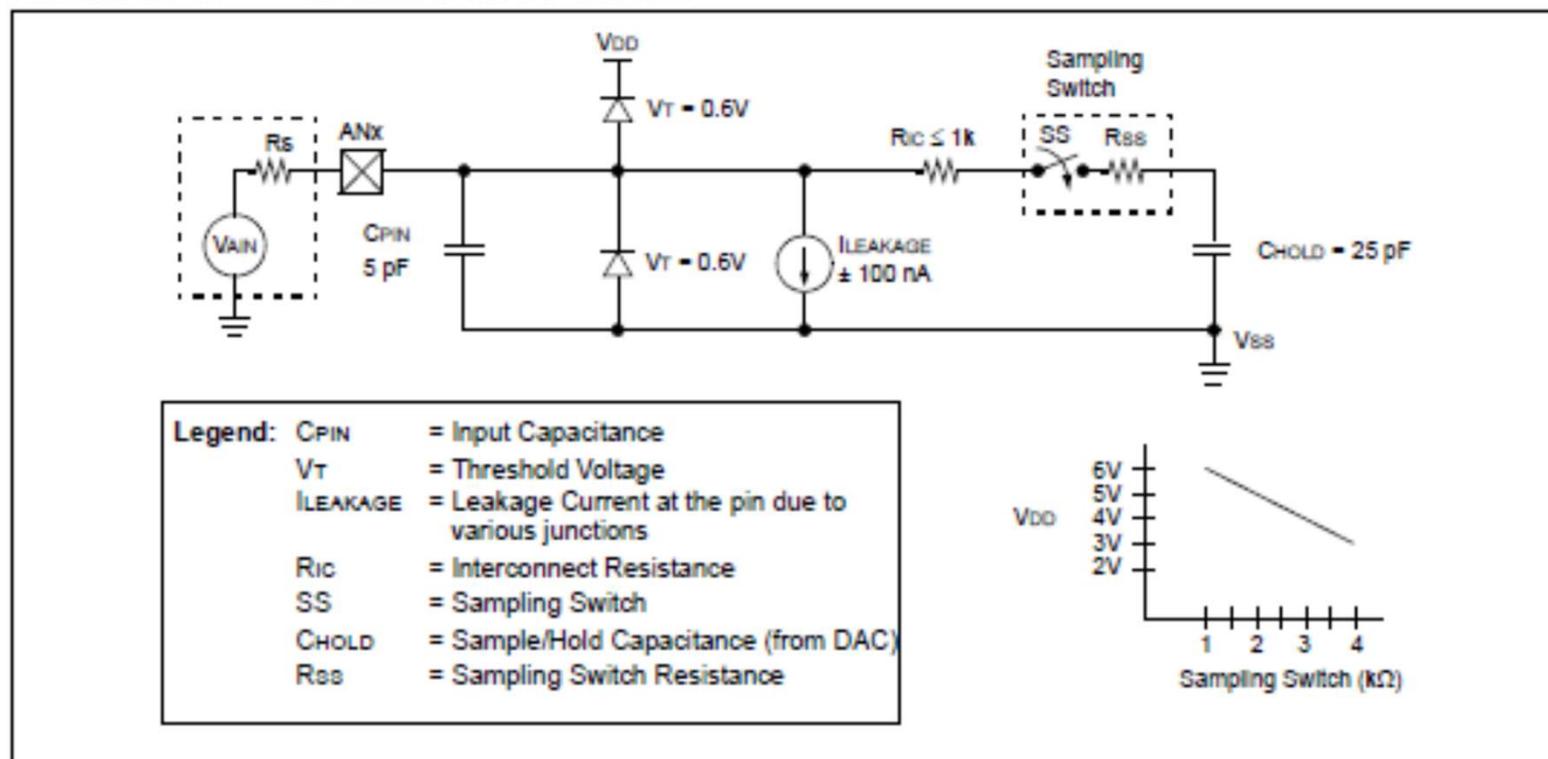
Sơ đồ khối

FIGURE 21-1: A/D BLOCK DIAGRAM

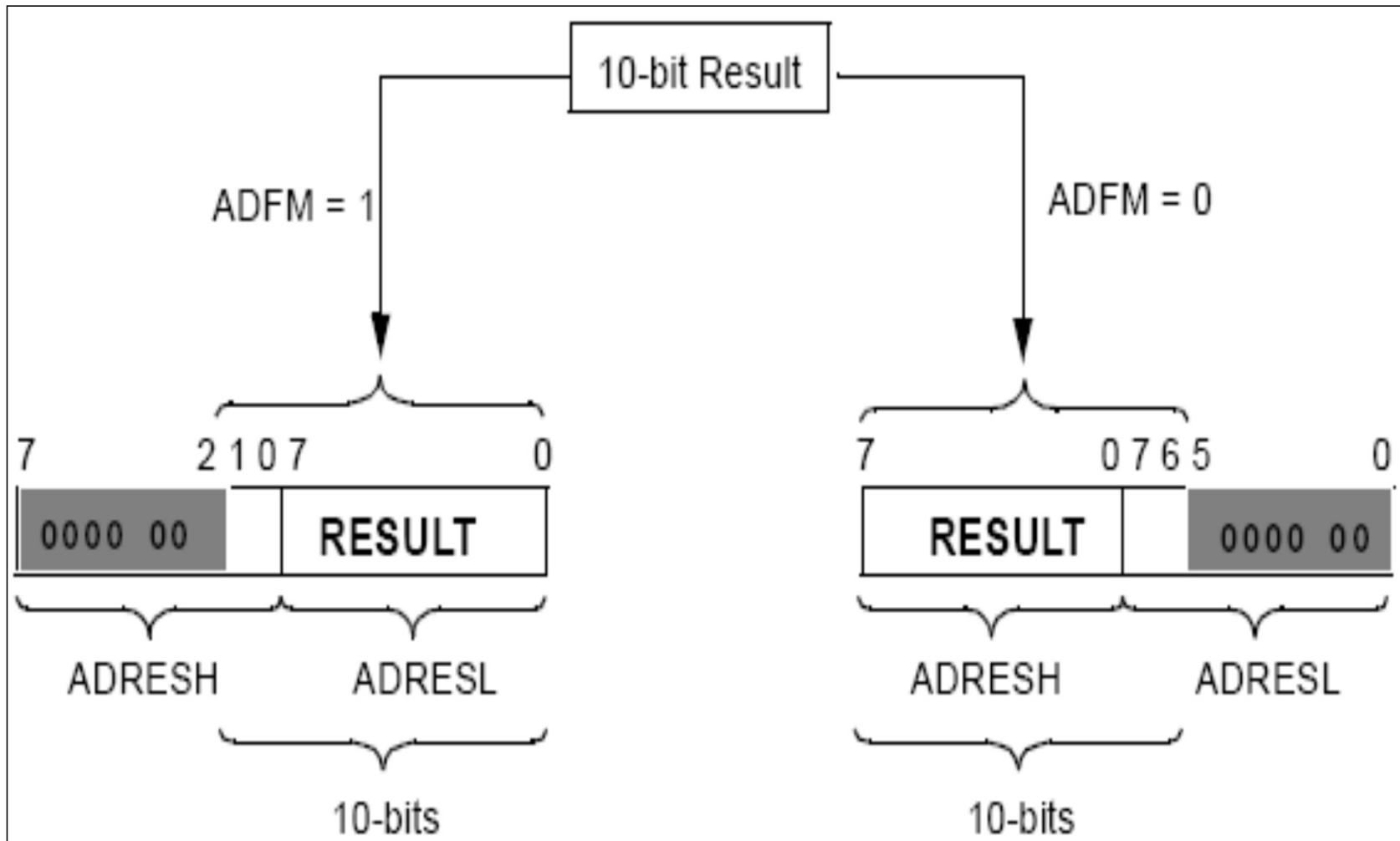


Mô hình Analog input

FIGURE 21-3: ANALOG INPUT MODEL

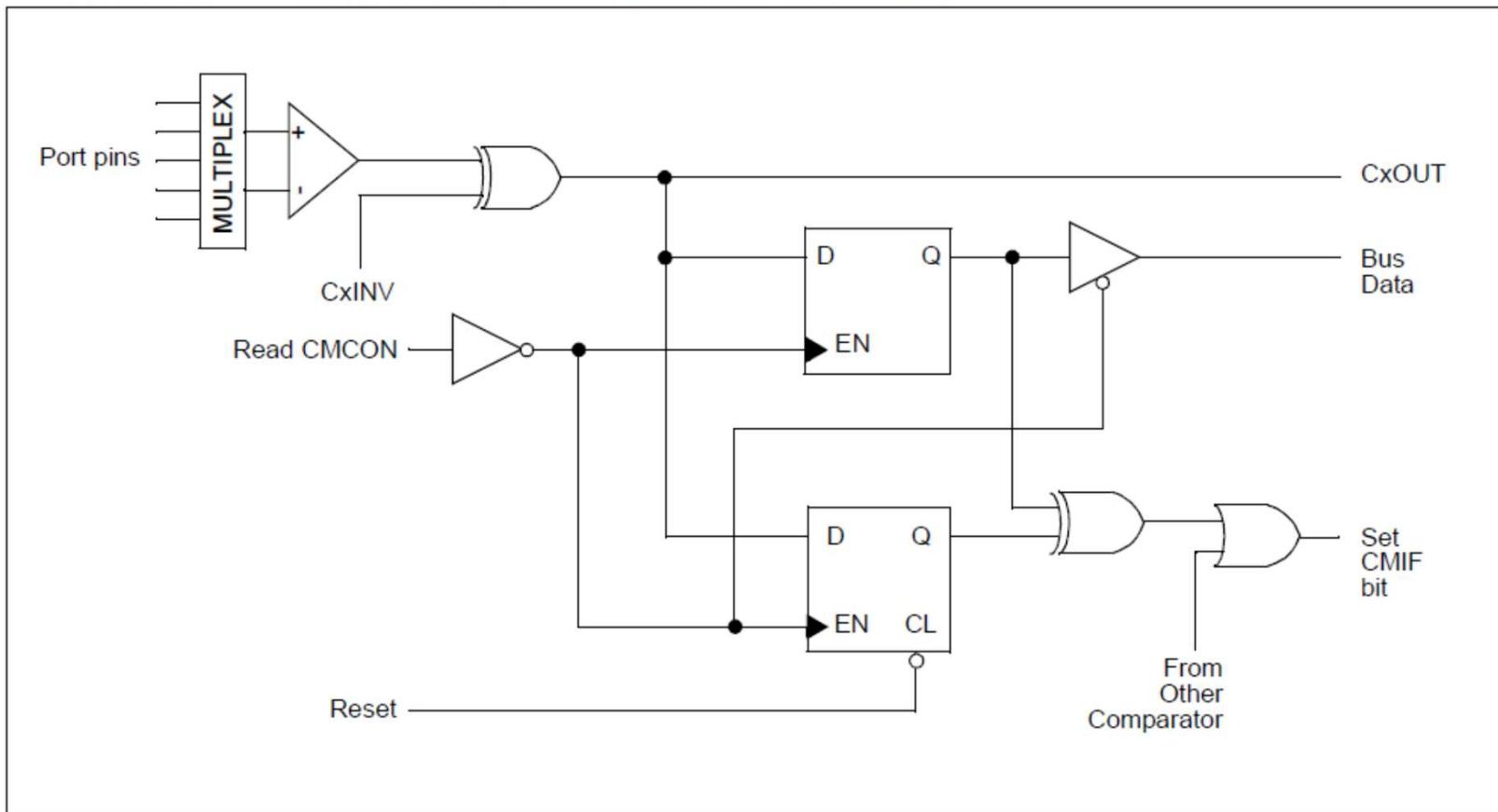


Kết quả A/D 10 bit



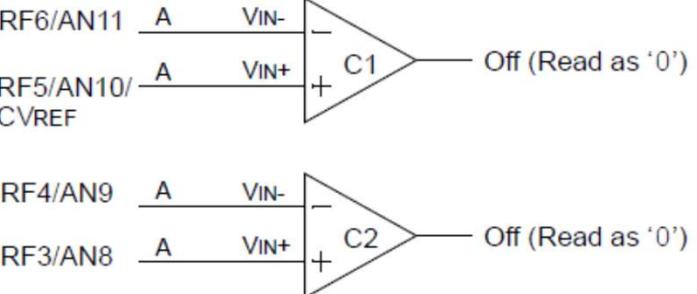
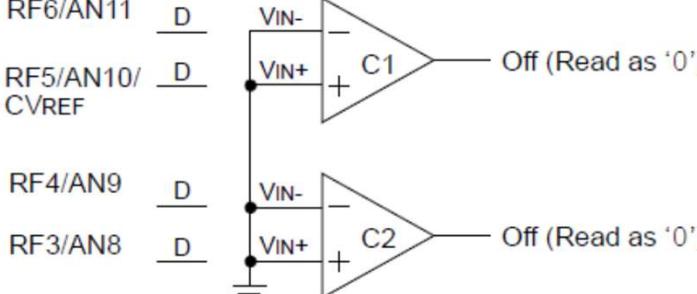
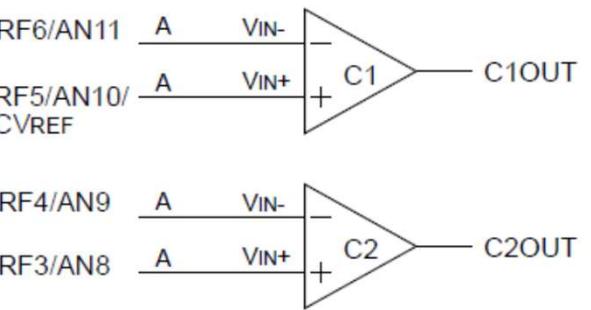
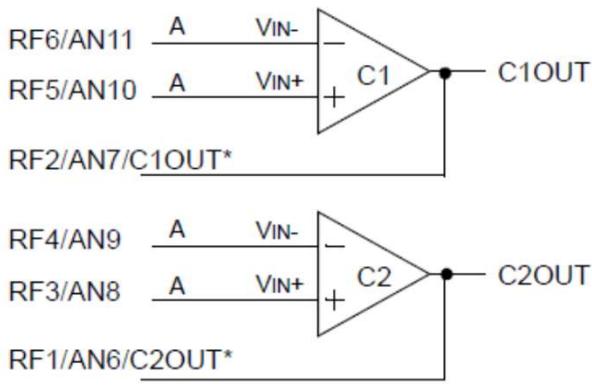
Khối so sánh tín hiệu

FIGURE 22-3: COMPARATOR OUTPUT BLOCK DIAGRAM



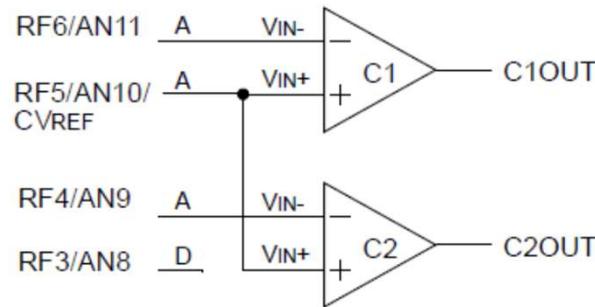
Chế độ Compare

FIGURE 22-1: COMPARATOR I/O OPERATING MODES

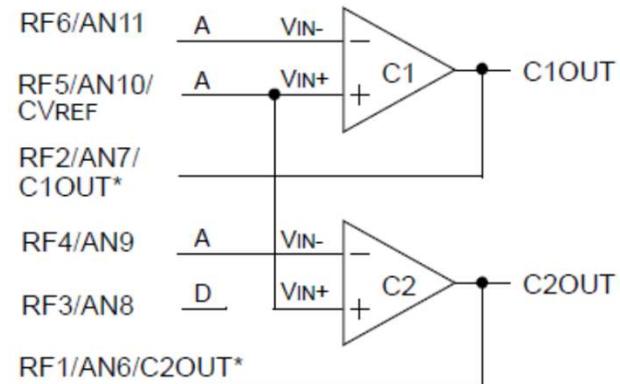
<p>Comparators Reset CM<2:0> = 000</p> 	<p>Comparators Off (POR Default Value) CM<2:0> = 111</p> 
<p>Two Independent Comparators CM<2:0> = 010</p> 	<p>Two Independent Comparators with Outputs CM<2:0> = 011</p> 

Chế độ Compare

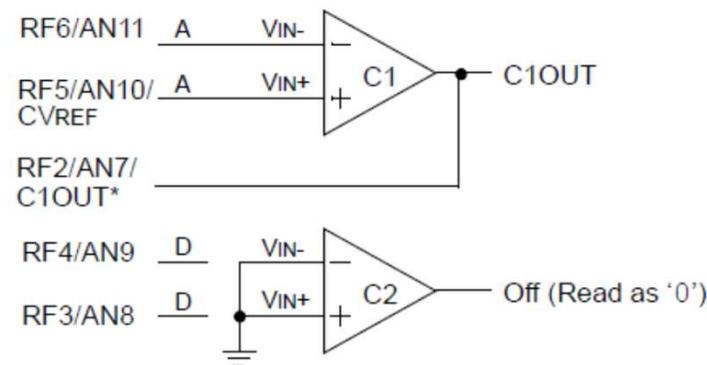
Two Common Reference Comparators
CM<2:0> = 100



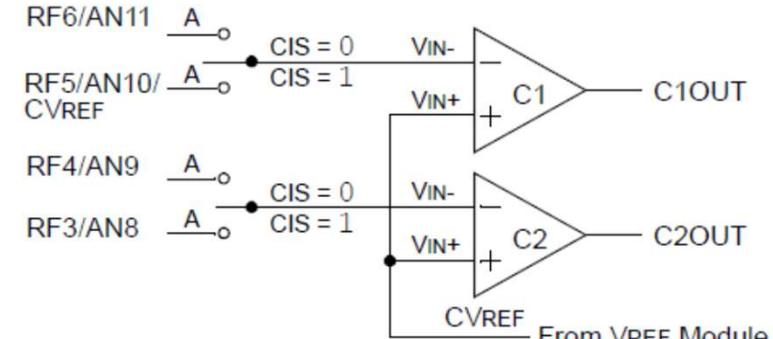
Two Common Reference Comparators with Outputs
CM<2:0> = 101



One Independent Comparator with Output
CM<2:0> = 001



Four Inputs Multiplexed to Two Comparators
CM<2:0> = 110



A = Analog Input, port reads zeros always

D = Digital Input

CIS (CMCON<3>) is the Comparator Input Switch

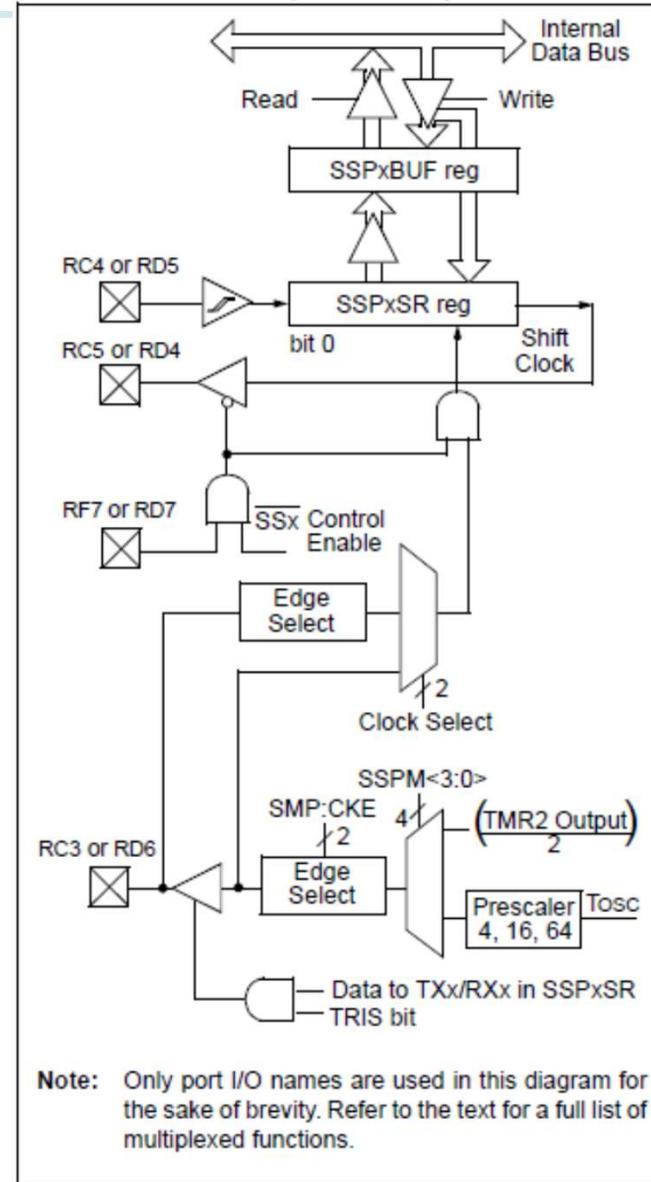
* Setting the TRISF<2:1> bits will disable the comparator outputs by configuring the pins as inputs.

Master Synchronous Serial Port (MSSP)

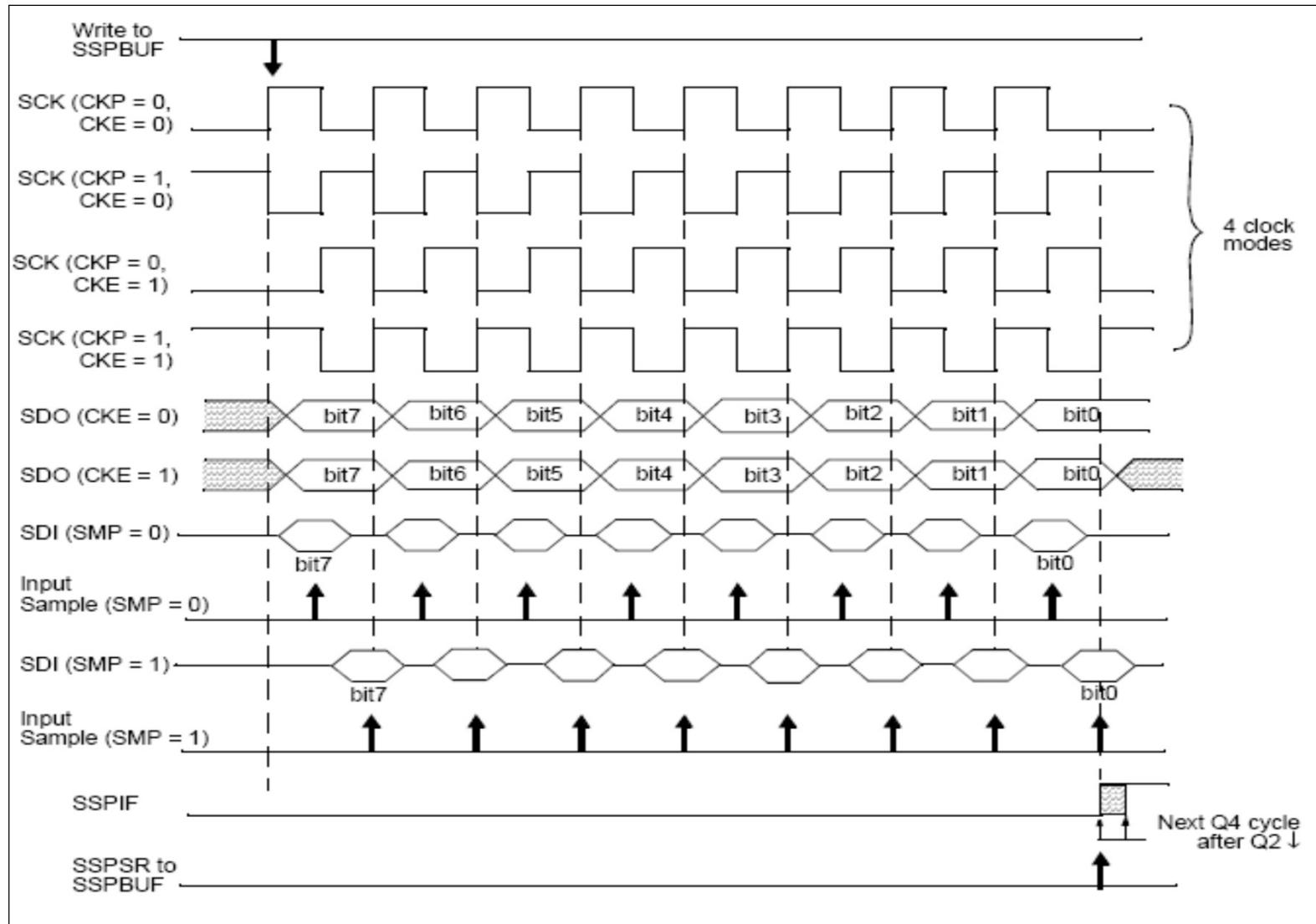
- Module MSSP có thể hoạt động một trong hai chế độ:
 - Giao tiếp ngoại vi tuần tự (SPI)
 - Mạch tích hợp bên trong (I2C TM)
- Chế độ chủ hoàn toàn (Full Master mode).
- Chế độ tớ (Slave mode).

Chế độ SPI

FIGURE 19-1: MSSP BLOCK DIAGRAM (SPI MODE)



Dạng sóng SPI, chế độ Master

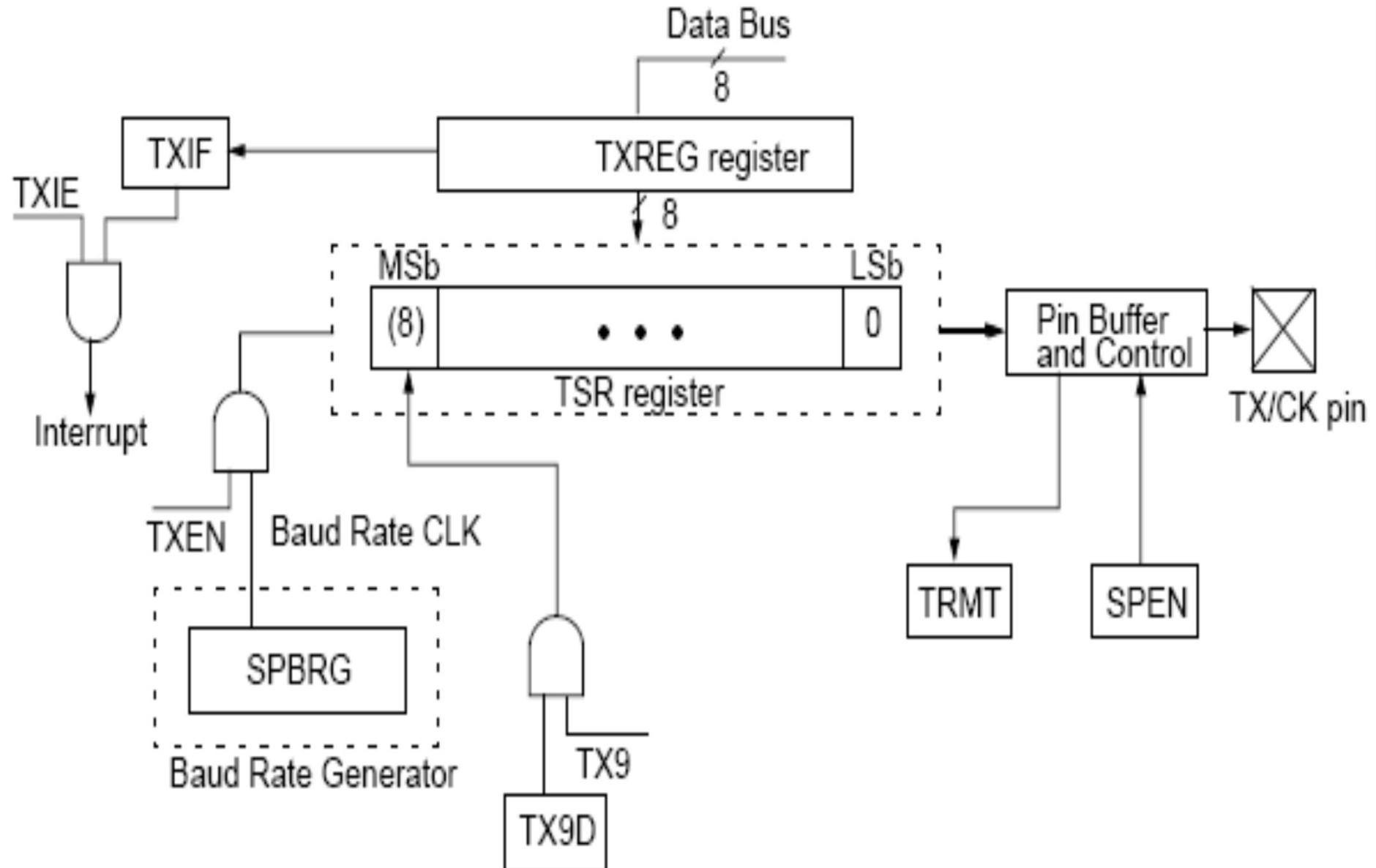


Giao tiếp USART

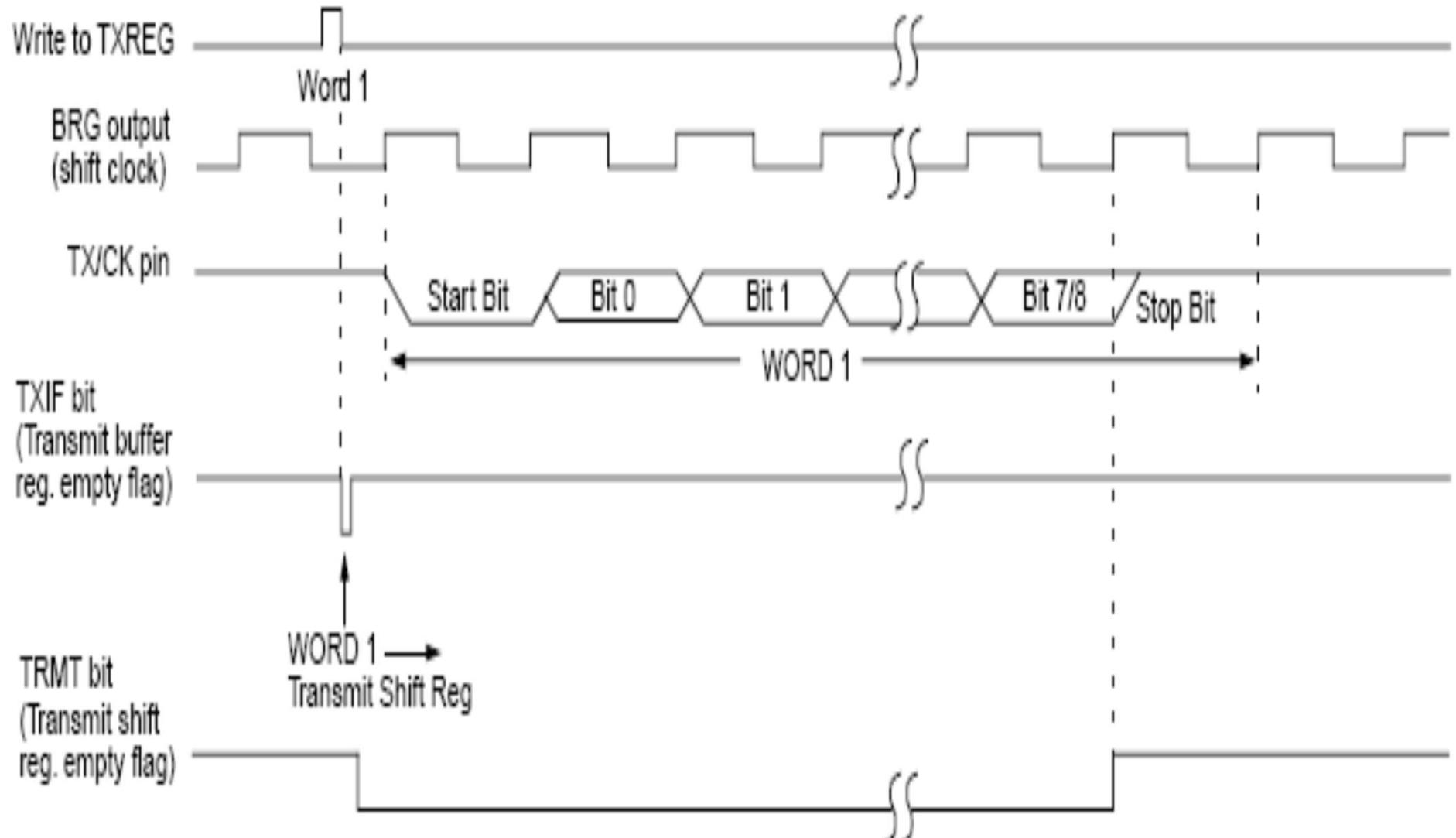
Bộ USART có thể được thiết lập một trong các chế độ sau:

- Bất đồng bộ - song công (full duplex)
- Đồng bộ - Master - bán song công (half duplex)
- Đồng bộ - Slave – bán song công (half duplex)

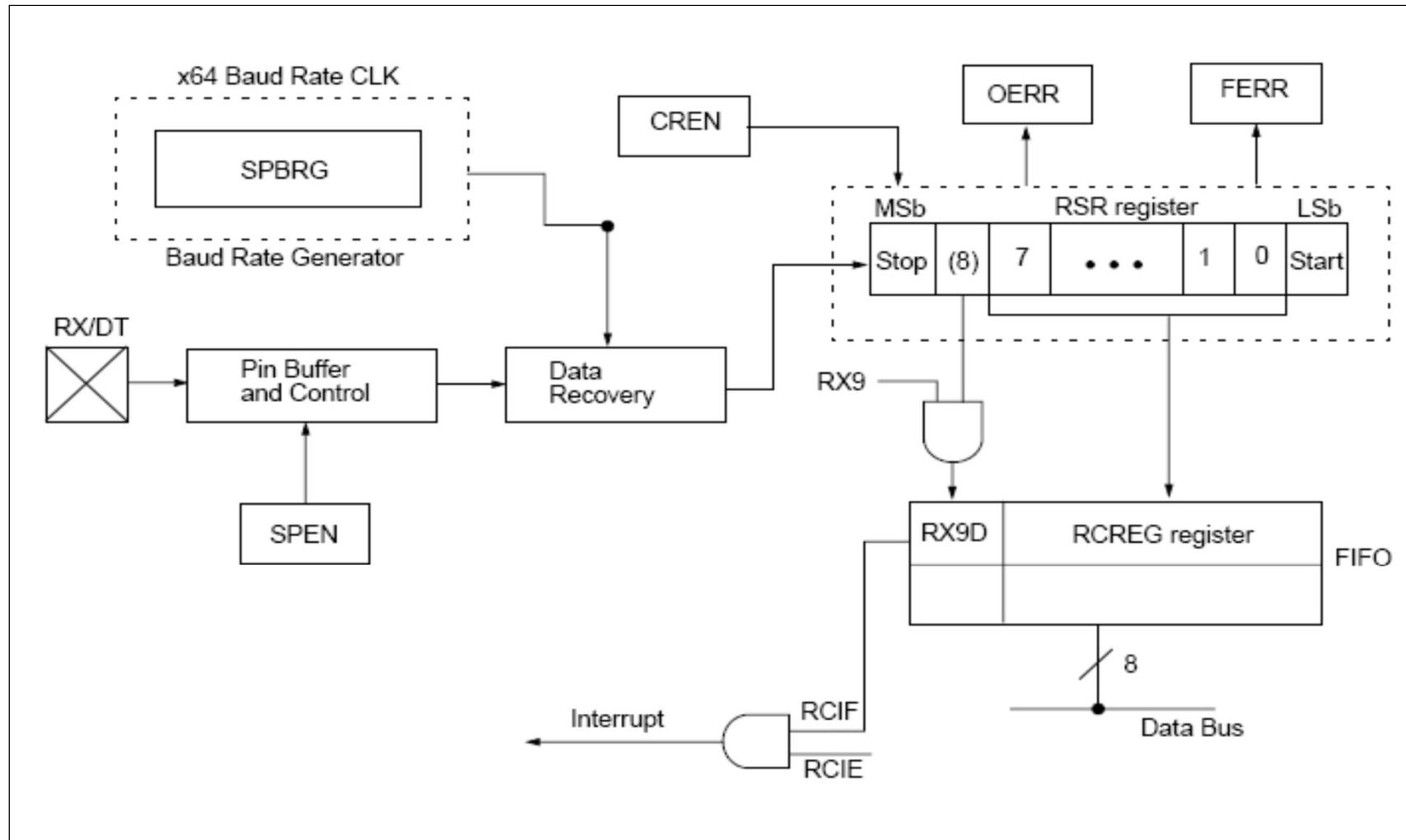
Sơ đồ khối truyền



Dạng xung truyền



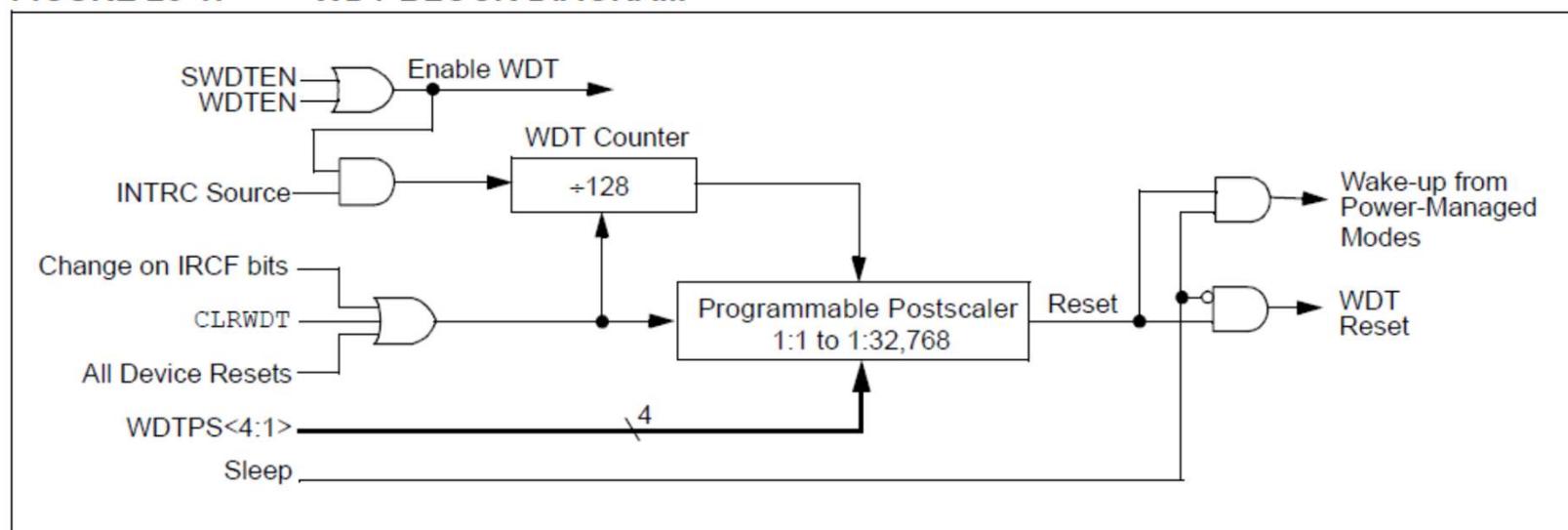
Sơ đồ khối bộ nhận USART



Watchdog Timer (WDT)

- WDT có thể được dùng để theo dõi hoạt động của CPU. Nếu trong thời gian lập trình nhất định ($4\text{ ms} \div 2.18\text{ phút}$) mà CPU không thực hiện lệnh SLEEP hoặc CLRWDT thì nó sẽ reset lại hệ thống.

FIGURE 25-1: WDT BLOCK DIAGRAM



Chế độ tiết kiệm năng lượng SLEEP

FIGURE 3-5: TRANSITION TIMING FOR ENTRY TO SLEEP MODE

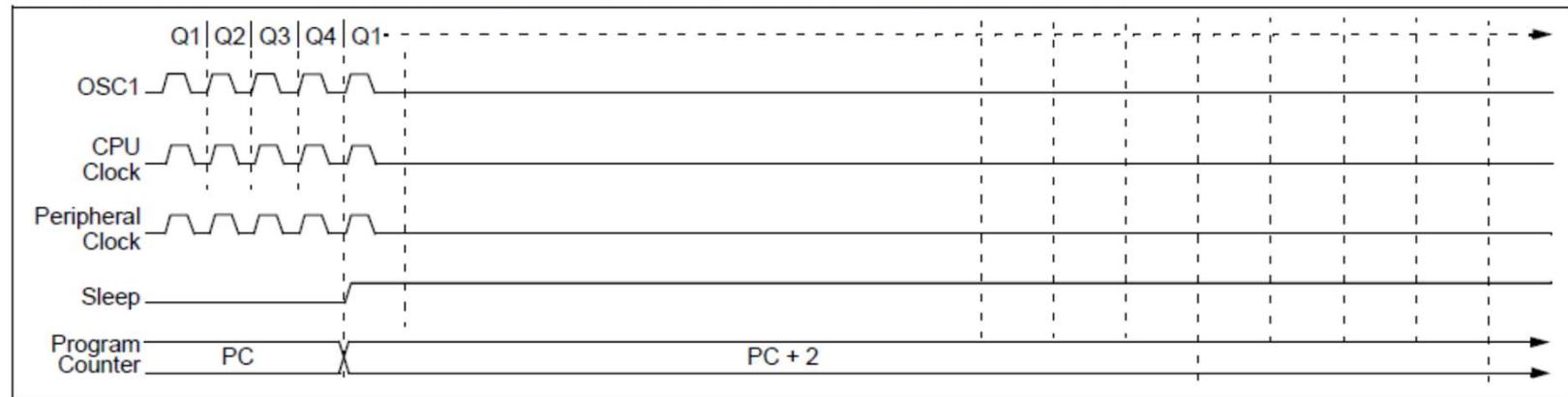
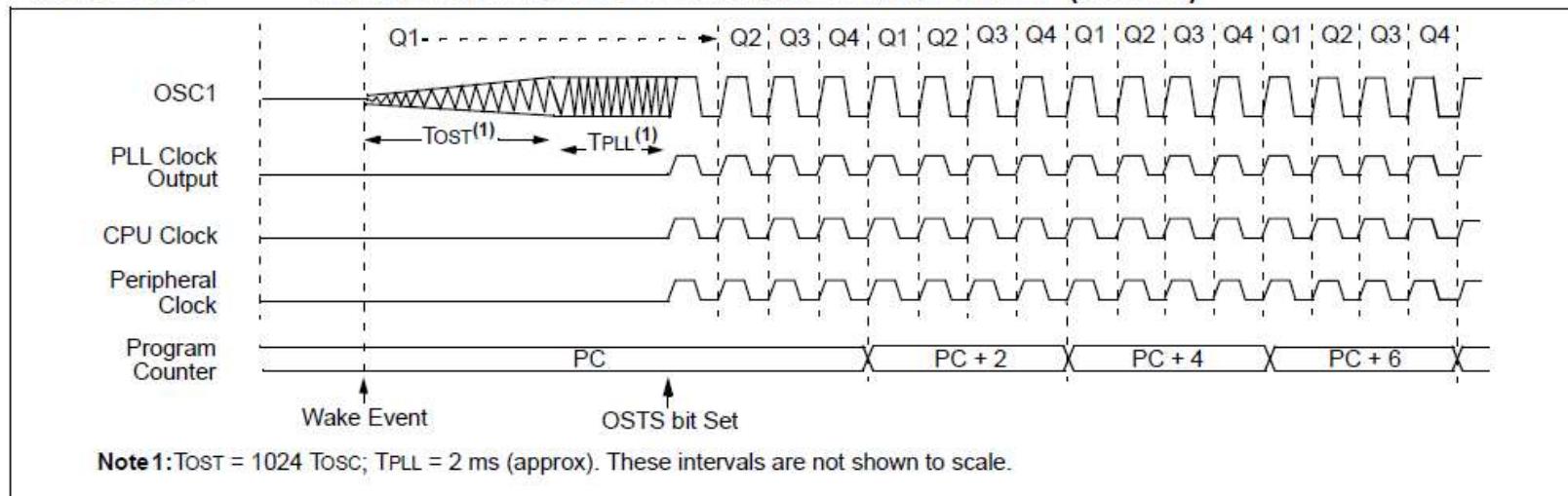


FIGURE 3-6: TRANSITION TIMING FOR WAKE FROM SLEEP (HSPLL)



Chế độ tiết kiệm năng lượng IDLE

FIGURE 3-7: TRANSITION TIMING FOR ENTRY TO IDLE MODE

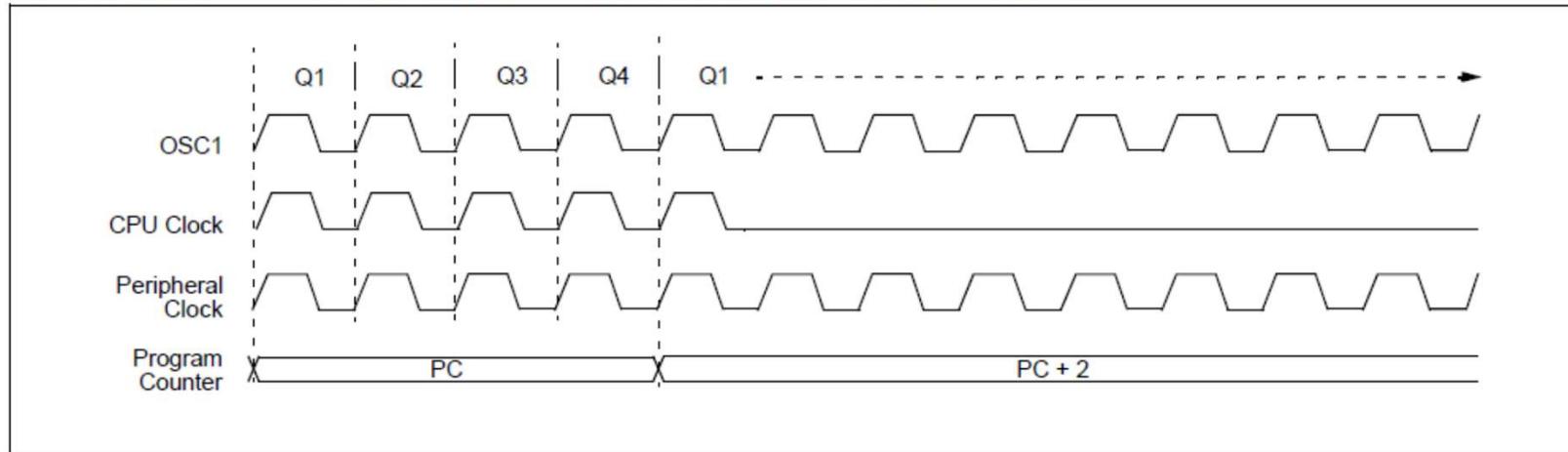
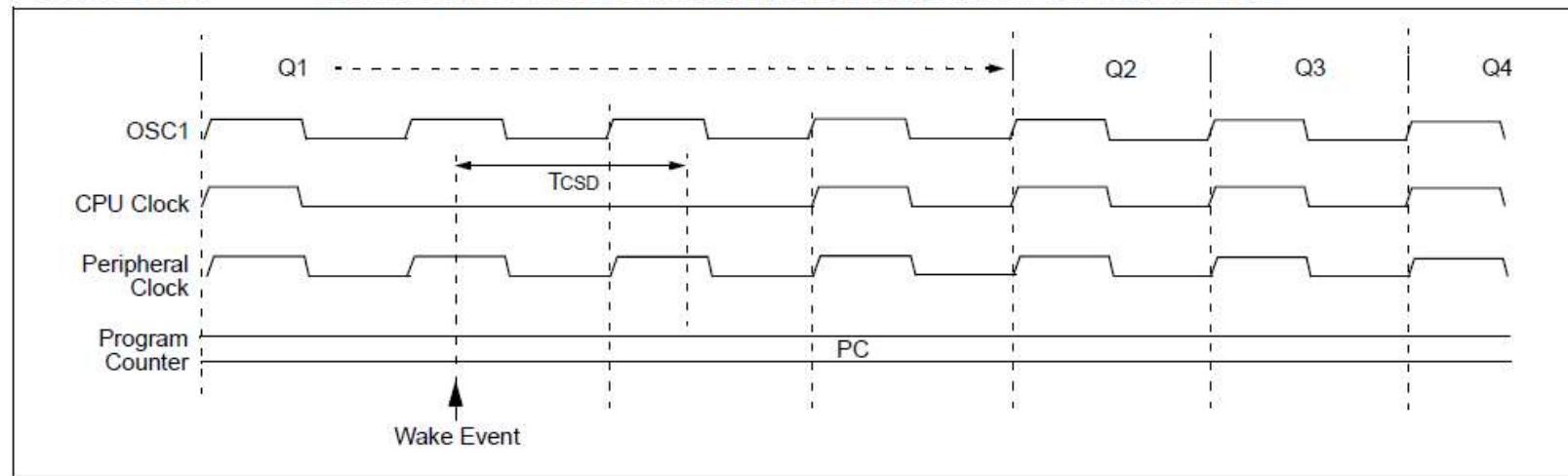


FIGURE 3-8: TRANSITION TIMING FOR WAKE FROM IDLE TO RUN MODE

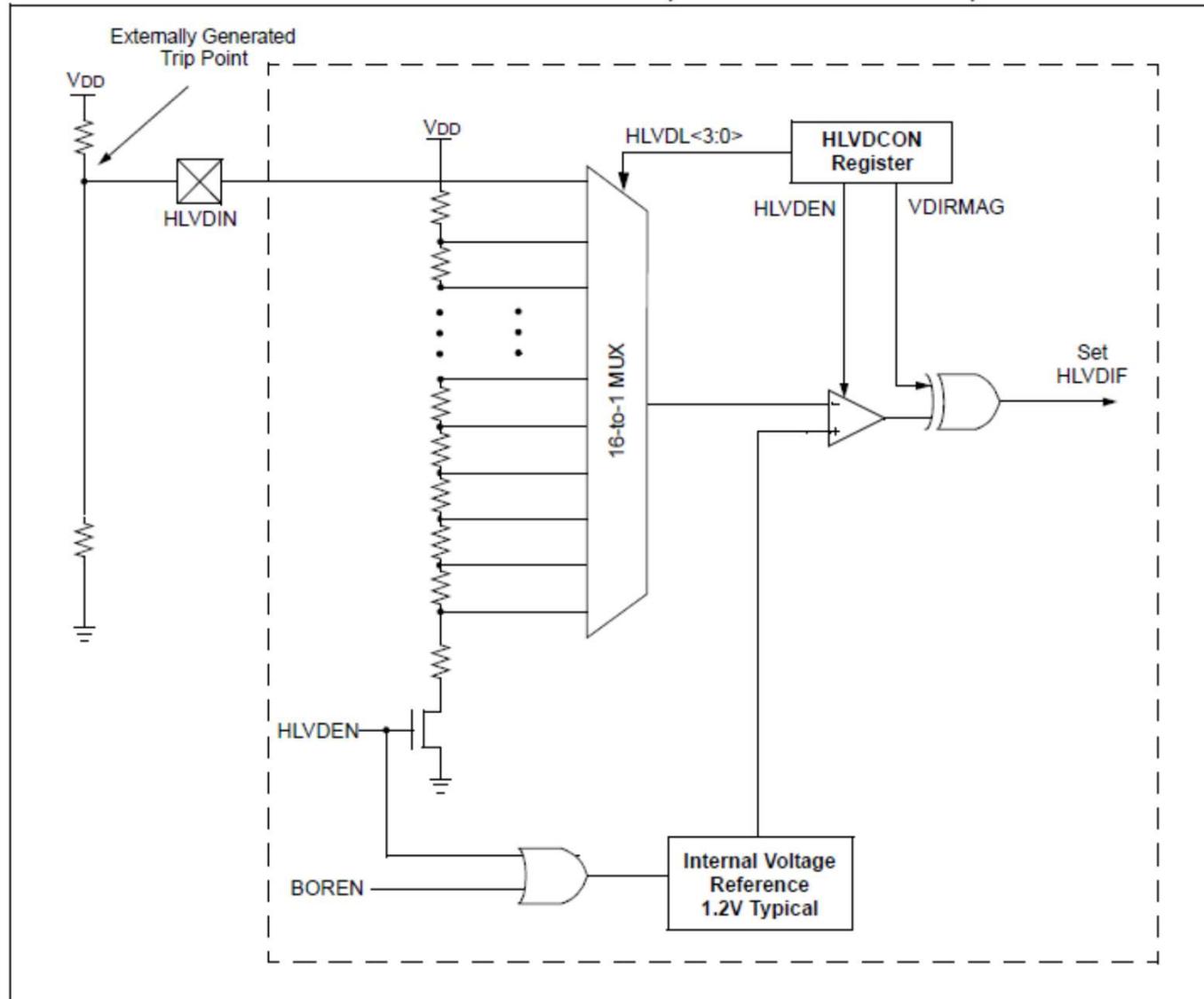


High/Low Voltage Detect (HLVD)

- Module lập trình được này cho phép phát hiện sự tăng hoặc giảm điện áp nguồn.
- Khi mức điện áp nguồn của vi điều khiển vượt ngưỡng đã chọn theo chiều tăng hoặc giảm thì sẽ có một ngắt quãng xảy ra.
- Chương trình xử lý ngắt có nhiệm vụ cảnh báo cho người sử dụng biết.

High/Low Voltage Detect (HLVD)

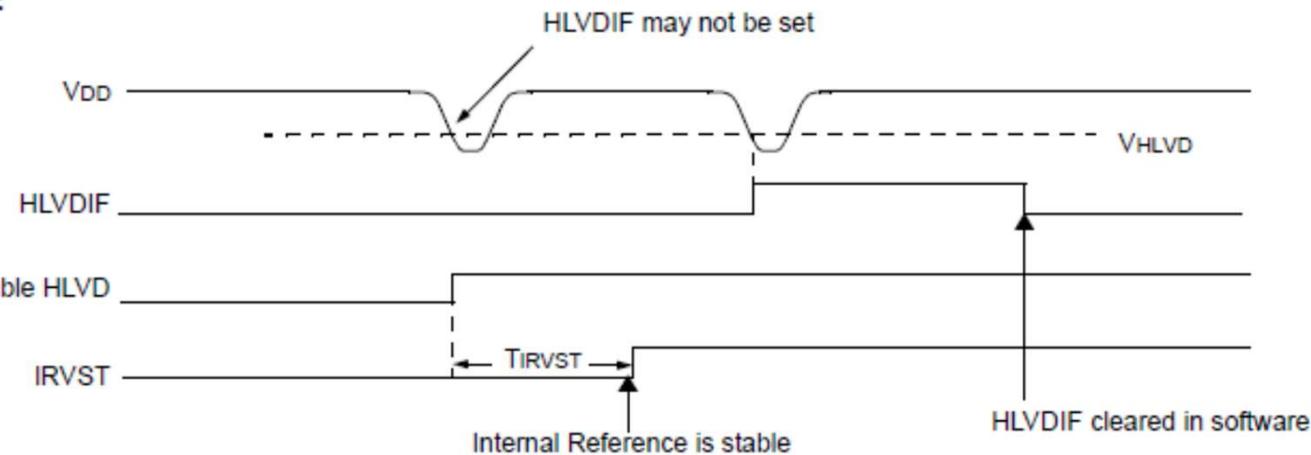
FIGURE 24-1: HLVD MODULE BLOCK DIAGRAM (WITH EXTERNAL INPUT)



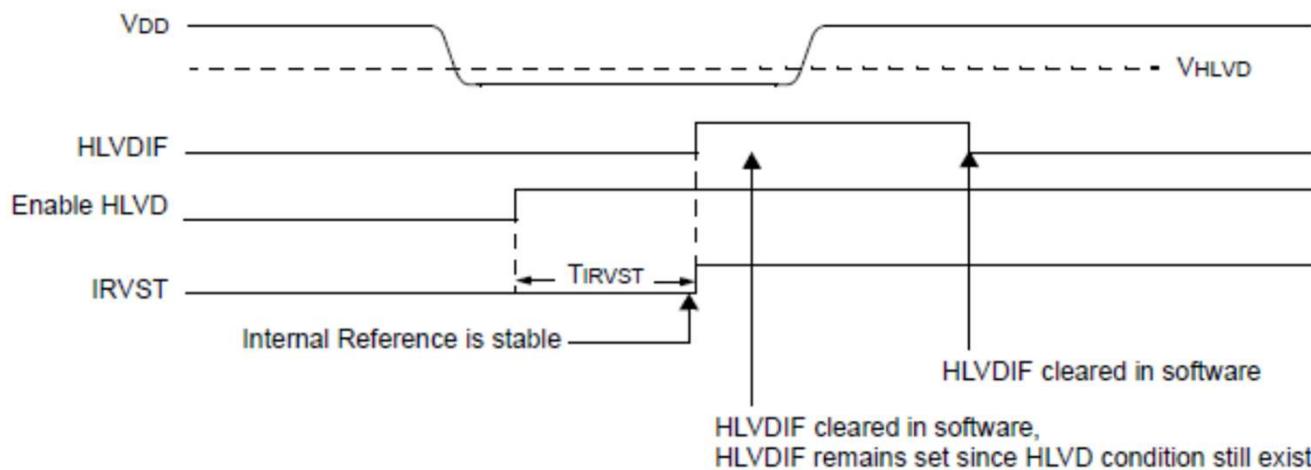
Sụt áp

FIGURE 24-2: LOW-VOLTAGE DETECT OPERATION (VDIRMMAG = 0)

CASE 1:

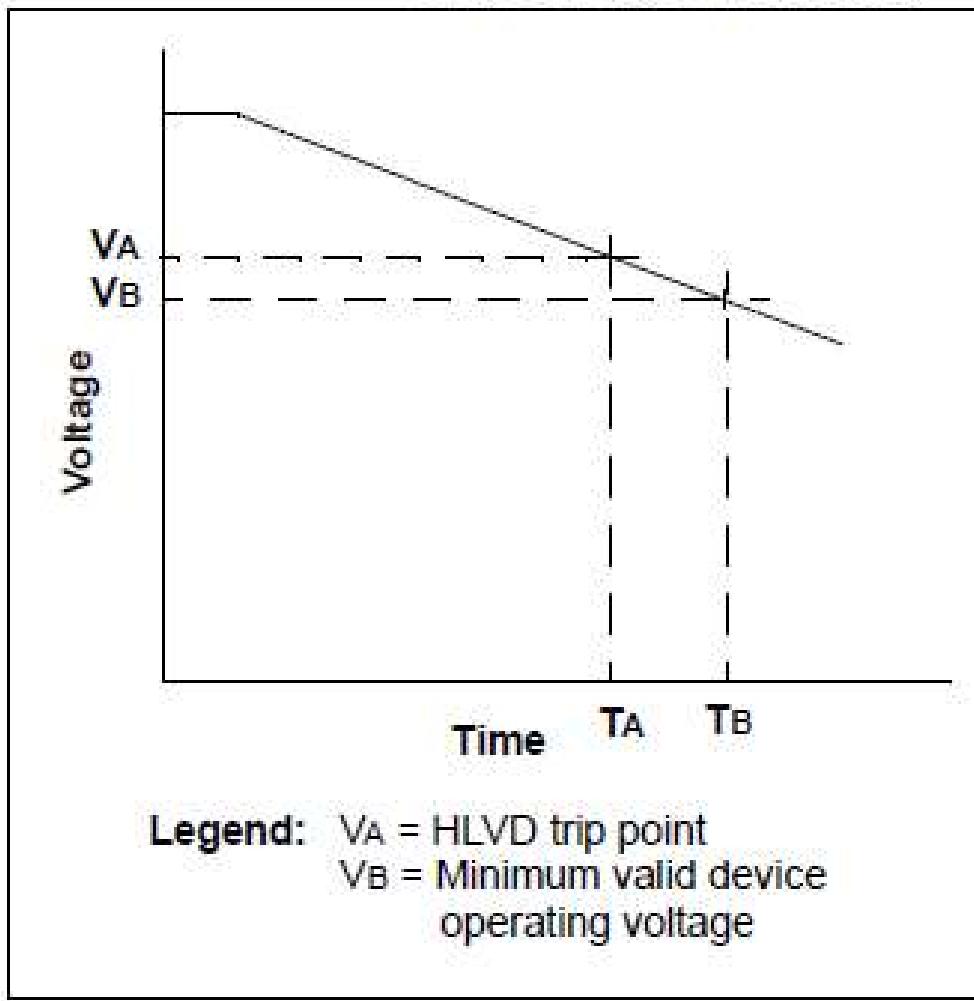


CASE 2:



Điện áp ngưỡng phát hiện sụt áp

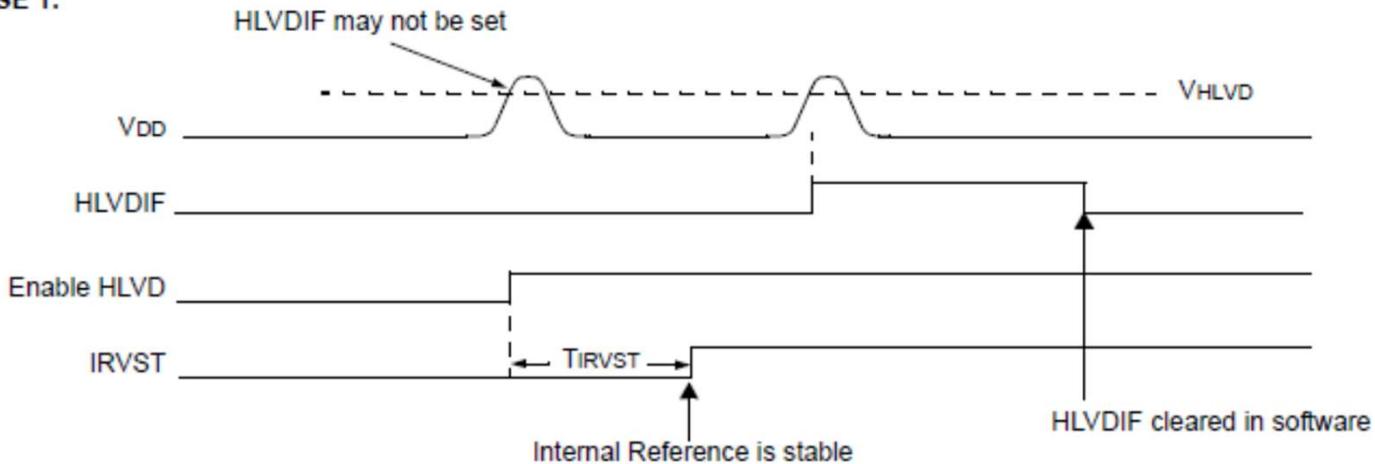
FIGURE 24-4: TYPICAL LOW-VOLTAGE DETECT APPLICATION



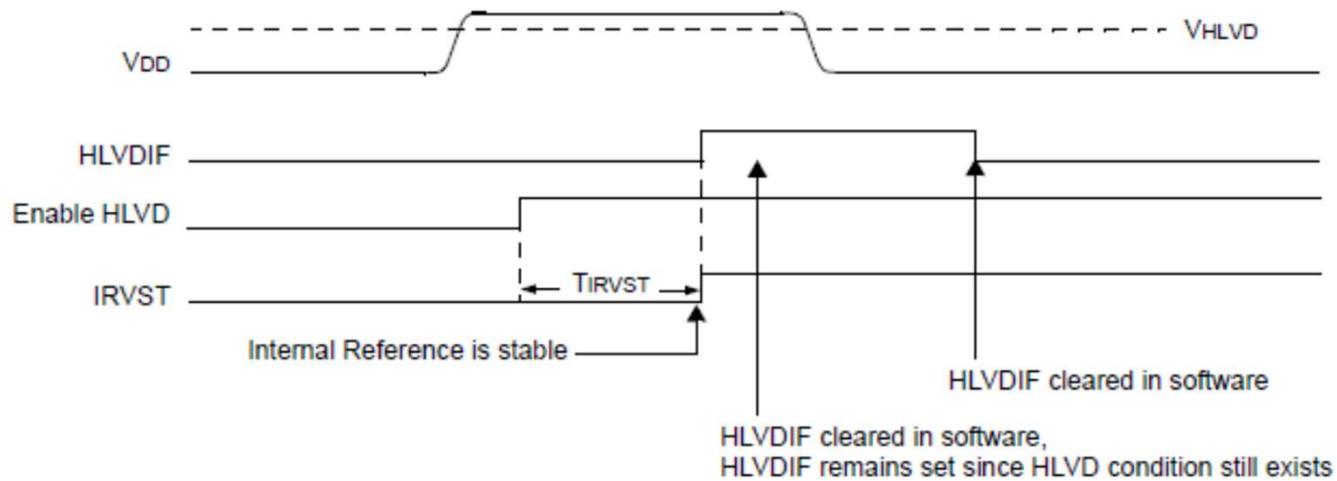
Tăng áp

FIGURE 24-3: HIGH-VOLTAGE DETECT OPERATION (VDIRMAG = 1)

CASE 1:



CASE 2:



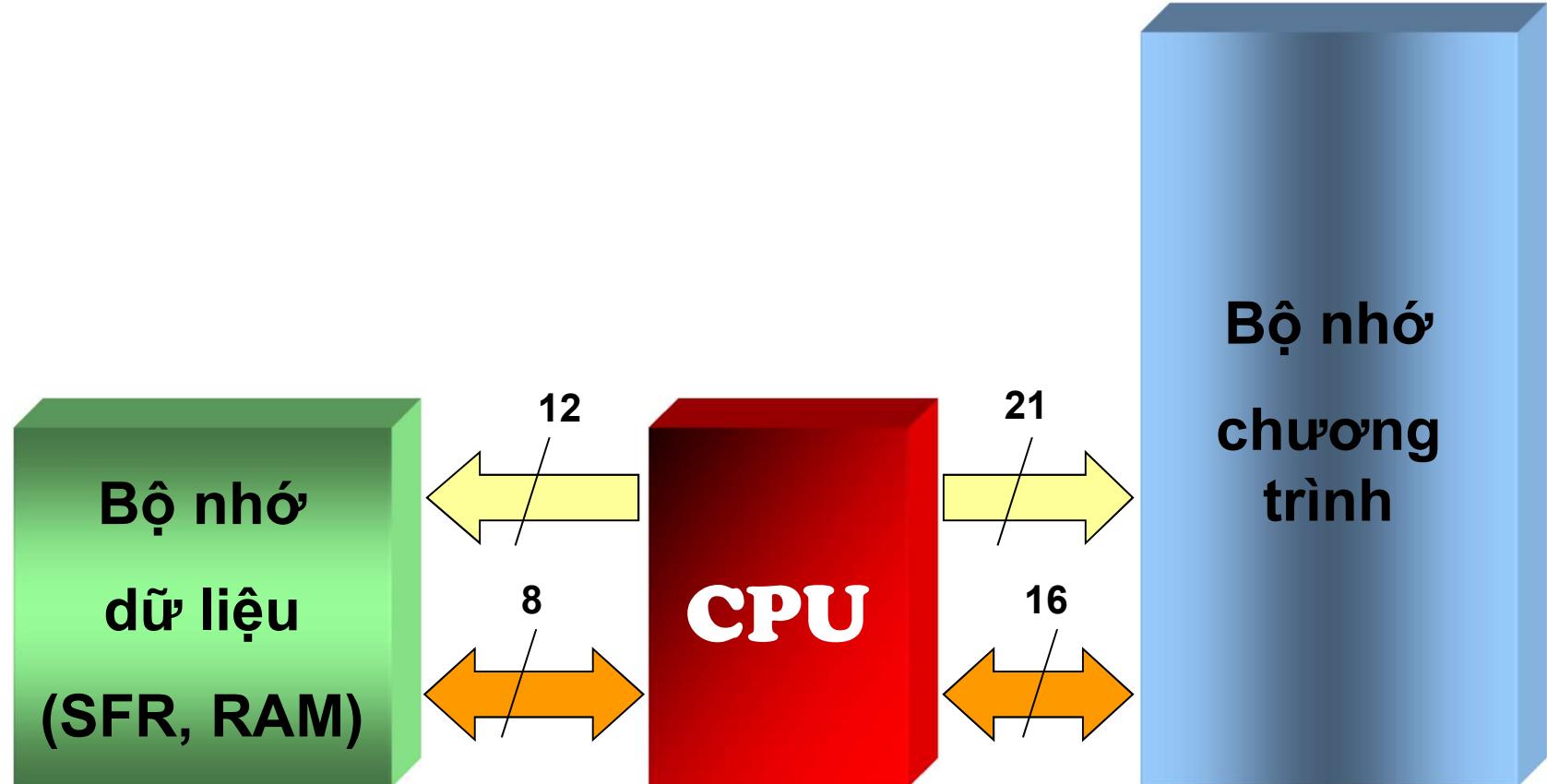


Bộ nhớ và thanh ghi

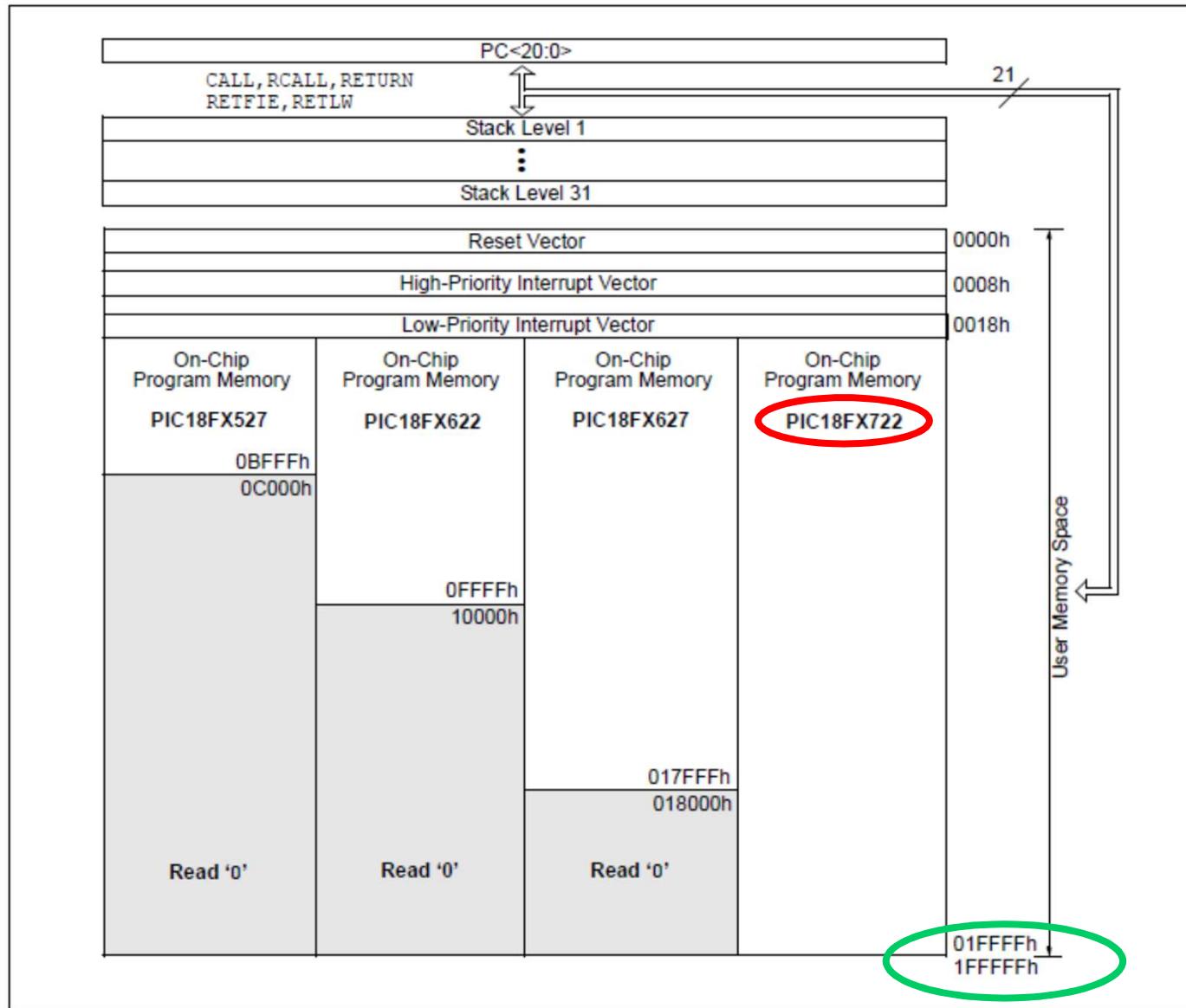
1. Tổ chức bộ nhớ
2. Thanh ghi chức năng đặc biệt SFR
3. Truy xuất



Kiến trúc Harvard



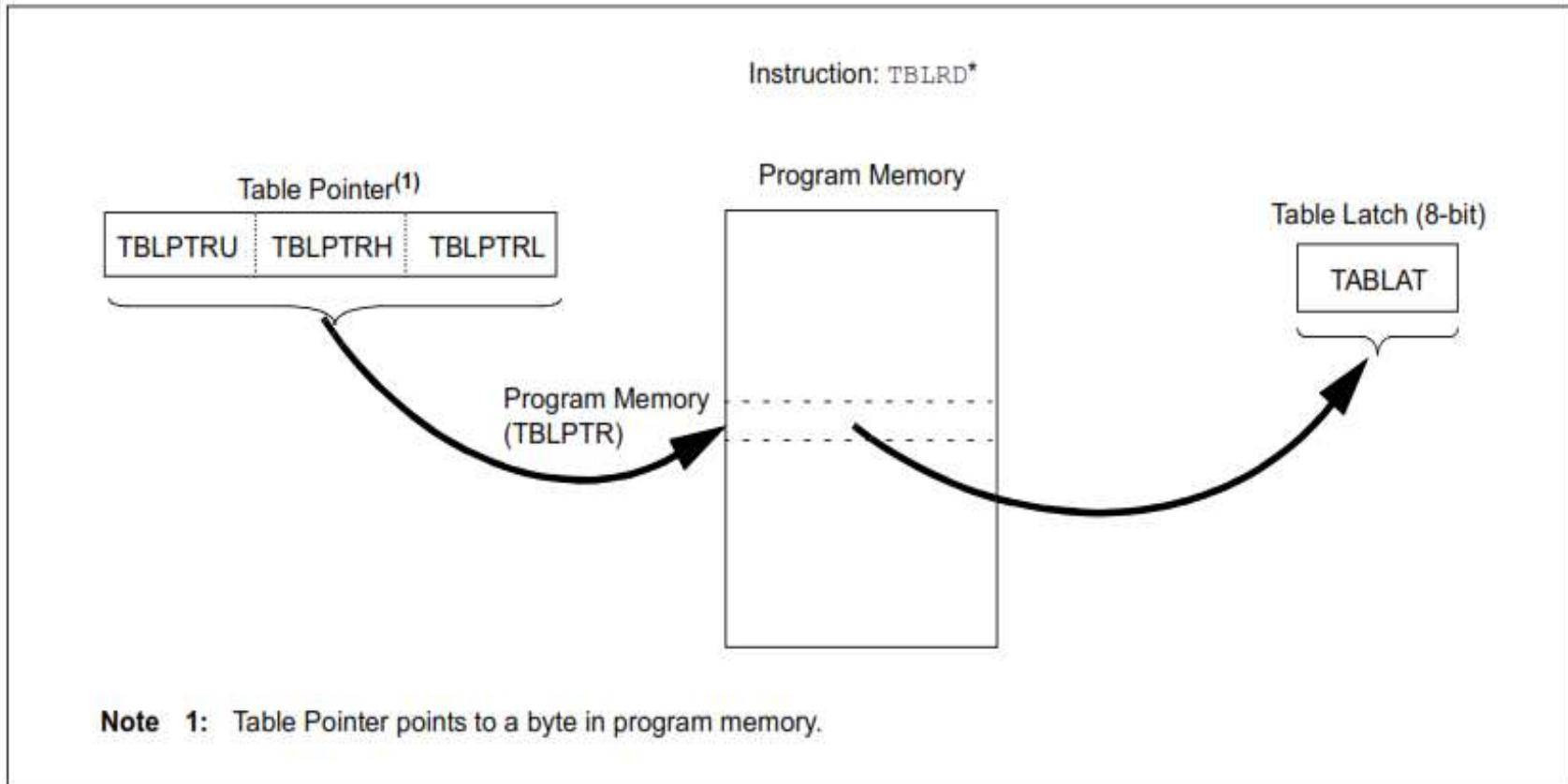
Bản đồ bộ nhớ chương trình và Stack



Bộ nhớ chương trình (Flash ROM)

- Chứa **chương trình** và các **bảng dữ liệu**.
- **Chương trình** do CPU truy xuất bằng thanh ghi PC.
- Người lập trình có thể truy xuất **bảng dữ liệu** bằng thanh ghi TBLPTR (Table pointer) 21 bits gồm:
 - TBLPTRU chứa 5 bit $b_{20} \div b_{16}$.
 - TBLPTRH chứa 8 bit $b_{15} \div b_8$.
 - TBLPTRL chứa 8 bit $b_7 \div b_0$.
- Chương trình thường được nạp vào Flash ROM từ bên ngoài.
- PIC 18F có khả năng ghi (xoá) Flash ROM khi đang chạy chương trình nhưng theo khối 8 bytes (64 bytes). Tuy nhiên, chức năng này ít được sử dụng.

Thao tác đọc bảng bằng TBLPTR



Trong C dùng biến non-auto với từ khóa const đứng trước

const unsigned char VALUE = 12; //dữ liệu trong ROM

Bộ nhớ EEPROM

- PIC 18F8722 có 1024 byte EEPROM có thể đọc/ghi khi đang chạy chương trình.
- Việc đọc/ghi EEPROM phải thực hiện thông qua 4 thanh ghi :
 - EECON1 - điều khiển qui trình đọc/ghi.
 - EECON2 - **thanh ghi ảo** dùng trong ghi/xóa EEPROM.
 - EEADRH, EEADR - chứa địa chỉ ô nhớ cần đọc/ghi (000-3FF).
 - và EEDATA – chứa dữ liệu đọc/ghi.
- Qui trình đọc/ghi EEPROM tham khảo tài liệu.

Đọc EEPROM

- Nạp địa chỉ vào EEADRH:EEADR.
- Xóa bit EEPGD và CFGS trong EECON1.
- Lập bit RD trong EECON1.
- Đọc dữ liệu từ EEDATA vào WREG.
- Code hợp ngữ như sau:

```
MOVlw  DATA_EE_ADDRH      ;  
MOVwf  EEADRH             ; Upper bits of Data Memory Address to read  
MOVlw  DATA_EE_ADDR       ;  
MOVwf  EEADR              ; Lower bits of Data Memory Address to read  
BCF    EECON1, EEPGD       ; Point to DATA memory  
BCF    EECON1, CFGS        ; Access EEPROM  
BSF    EECON1, RD          ; EEPROM Read  
MOVf   EEDATA, W           ; W = EEDATA
```

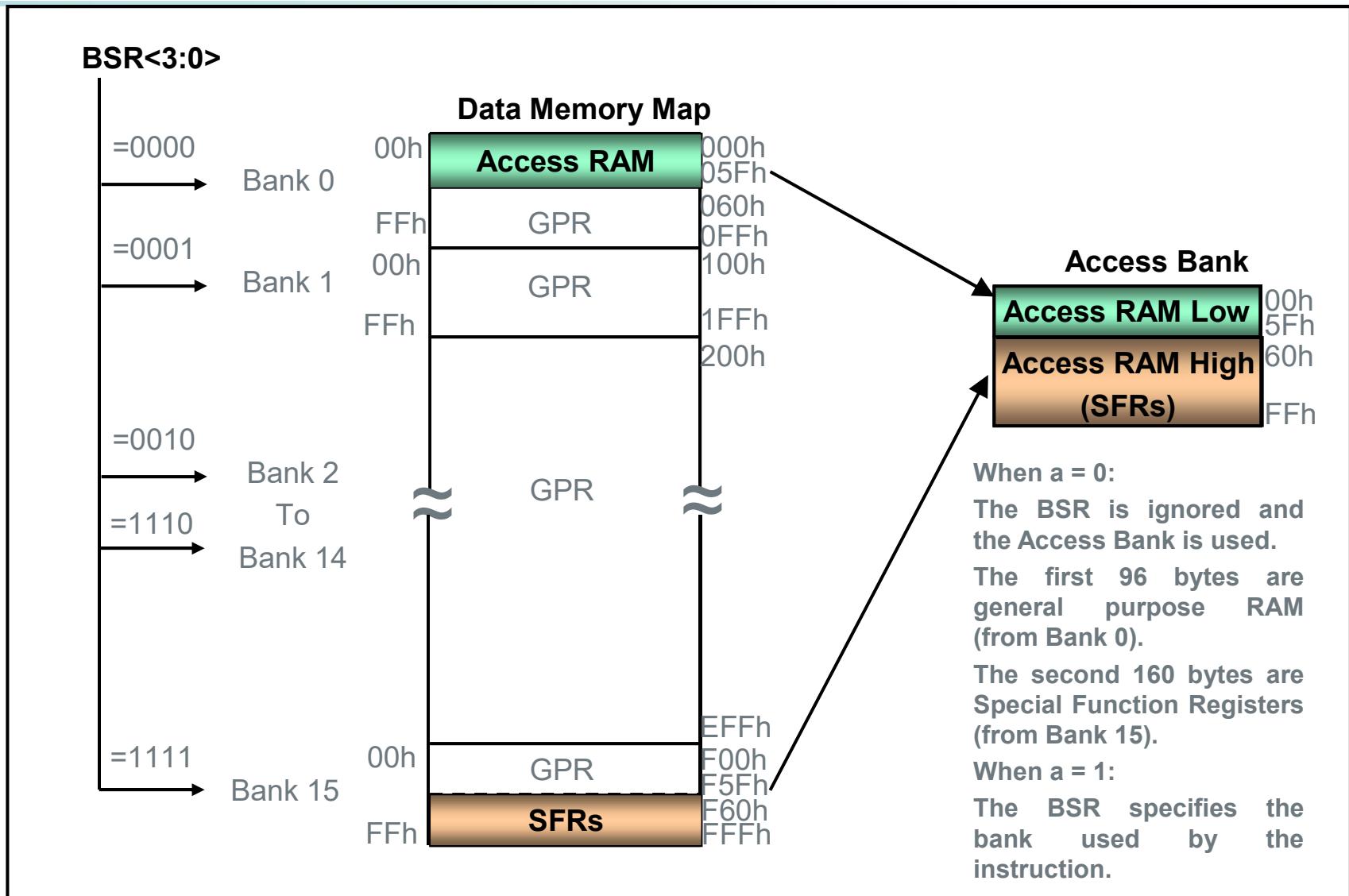
```
unsigned char eeprom_read(unsigned char address);
```

Ghi EEPROM

	MOVLW DATA_EE_ADDRH ;	
	MOVWF EEADRH ; Upper bits of Data Memory Address to write	
	MOVLW DATA_EE_ADDR ;	
	MOVWF EEADR ; Lower bits of Data Memory Address to write	
	MOVLW DATA_EE_DATA ;	
	MOVWF EEDATA ; Data Memory Value to write	
	BCF EECON1, EPGD ; Point to DATA memory	
	BCF EECON1, CFGS ; Access EEPROM	
	BSF EECON1, WREN ; Enable writes	
	BCF INTCON, GIE ; Disable Interrupts	
	MOVLW 55h ;	
Required Sequence	MOVWF EECON2 ; Write 55h	
	MOVLW 0AAh ;	
	MOVWF EECON2 ; Write 0AAh	
	BSF EECON1, WR ; Set WR bit to begin write	
	BSF INTCON, GIE ; Enable Interrupts	
		; User code execution
	BCF EECON1, WREN ; Disable writes on write complete (EIF set)	

```
void eeprom_write(unsigned char address, unsigned char value);
```

Bộ nhớ dữ liệu RAM



Truy xuất bộ nhớ dữ liệu RAM

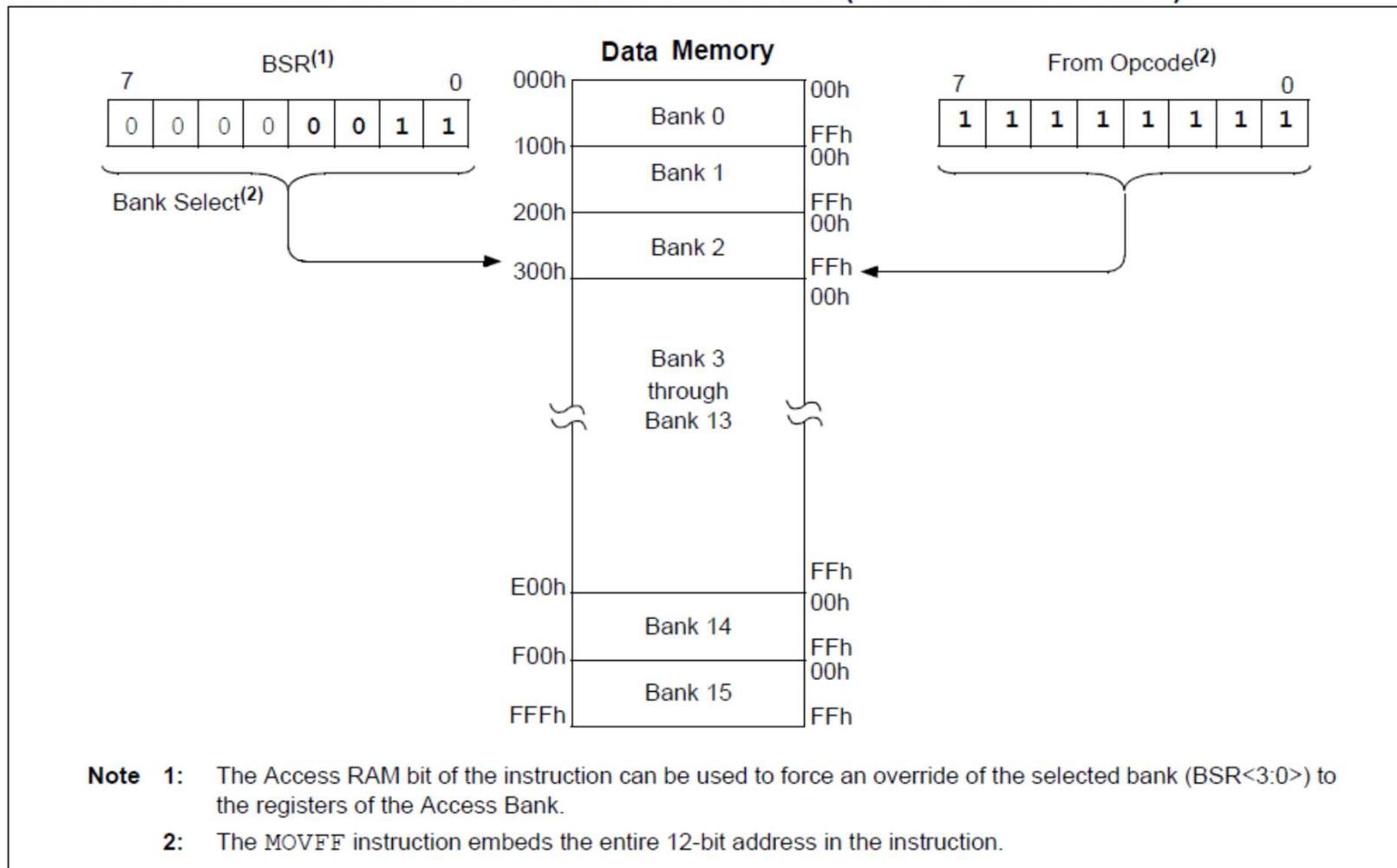
- Có 3 cách truy xuất bộ nhớ dữ liệu RAM (3936 bytes) :
 - Trực tiếp dùng Access Bank (địa chỉ 8 bit).
 - Trực tiếp qua BSR (địa chỉ 8 bit).
 - Gián tiếp qua thanh ghi FSRx (địa chỉ 12 bit).
- PIC 18F8722 có 3 thanh ghi FSRx (FSR0, FSR1 và FSR2).
- Để định nghĩa một biến RAM trong C, ta có thể dùng biến auto (cục bộ trong hàm) hay biến toàn cục volatile.

```
volatile unsigned char tam;
void main()
{
    unsigned char so;
    init();
    so=0;
    while(1)
```

[DS-p.74]

Truy xuất trực tiếp theo BSR

FIGURE 5-7: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)



Truy xuất gián tiếp theo FSRI

FIGURE 5-8: INDIRECT ADDRESSING

Using an instruction with one of the Indirect Addressing registers as the operand....

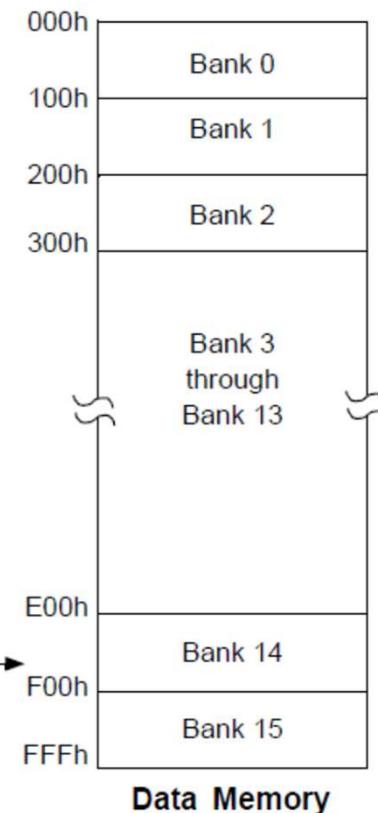
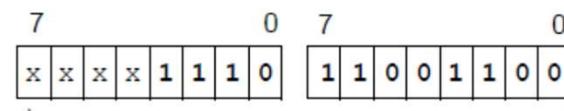
...uses the 12-bit address stored in the FSR pair associated with that register....

...to determine the data memory location to be used in that operation.

In this case, the FSR1 pair contains ECCh. This means the contents of location ECCh will be added to that of the W register and stored back in ECCh.

ADDWF, INDF1, 1

FSR1H:FSR1L



Địa chỉ SFRs dùng chung với RAM

TABLE 5-2: SPECIAL FUNCTION REGISTER MAP FOR THE PIC18F8722 FAMILY OF DEVICES

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2 ⁽¹⁾	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	SPBRGH1
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	BAUDCON1
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	SPBRGH2
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	CCPR2H	F9Ch	MEMCON	F7Ch	BAUDCON2
FFBh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCPR2L	F9Bh	OSCTUNE	F7Bh	— ⁽²⁾
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	TRISJ ⁽³⁾	F7Ah	— ⁽²⁾
FF9h	PCL	FD9h	FSR2L	FB9h	CCPR3H	F99h	TRISH ⁽³⁾	F79h	ECCP1DEL
FF8h	TBLPTRU	FD8h	STATUS	FB8h	CCPR3L	F98h	TRISG	F78h	TMR4
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	CCP3CON	F97h	TRISF	F77h	PR4
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS	F96h	TRISE	F76h	T4CON
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD	F75h	CCPR4H
FF4h	PRODH	FD4h	— ⁽²⁾	FB4h	CMCON	F94h	TRISC	F74h	CCPR4L
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB	F73h	CCP4CON
FF2h	INTCON	FD2h	HLVDCON	FB2h	TMR3L	F92h	TRISA	F72h	CCPR5H
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	LATJ ⁽³⁾	F71h	CCPR5L
FF0h	INTCON3	FD0h	RCON	FB0h	PSPCON	F90h	LATH ⁽³⁾	F70h	CCP5CON
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H	FAFh	SPBRG1	F8Fh	LATG	F6Fh	SPBRG2

Thanh ghi chức năng đặc biệt SFR

FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H	FAFh	SPBRG1	F8Fh	LATG	F6Fh	SPBRG2
FEEh	POSTINC0 ⁽¹⁾	FCEh	TMR1L	FAEh	RCREG1	F8Eh	LATF	F6Eh	RCREG2
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	FADh	TXREG1	F8Dh	LATE	F6Dh	TXREG2
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2	FACH	TXSTA1	F8Ch	LATD	F6Ch	TXSTA2
FEBh	PLUSW0 ⁽¹⁾	FCBh	PR2	FABh	RCSTA1	F8Bh	LATC	F6Bh	RCSTA2
FEAh	FSR0H	FCAh	T2CON	FAAh	EEADRH	F8Ah	LATB	F6Ah	ECCP3AS
FE9h	FSR0L	FC9h	SSP1BUF	FA9h	EEADR	F89h	LATA	F69h	ECCP3DEL
FE8h	WREG	FC8h	SSP1ADD	FA8h	EEDATA	F88h	PORTJ ⁽³⁾	F68h	ECCP2AS
FE7h	INDF1 ⁽¹⁾	FC7h	SSP1STAT	FA7h	EECON2 ⁽¹⁾	F87h	PORTH ⁽³⁾	F67h	ECCP2DEL
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSP1CON1	FA6h	EECON1	F86h	PORTG	F66h	SSP2BUF
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSP1CON2	FA5h	IPR3	F85h	PORTF	F65h	SSP2ADD
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE	F64h	SSP2STAT
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD	F63h	SSP2CON1
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	SSP2CON2
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	__(2)
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	__(2)

Note 1: This is not a physical register.

2: Unimplemented registers are read as '0'.

3: This register is not available on 64-pin devices.

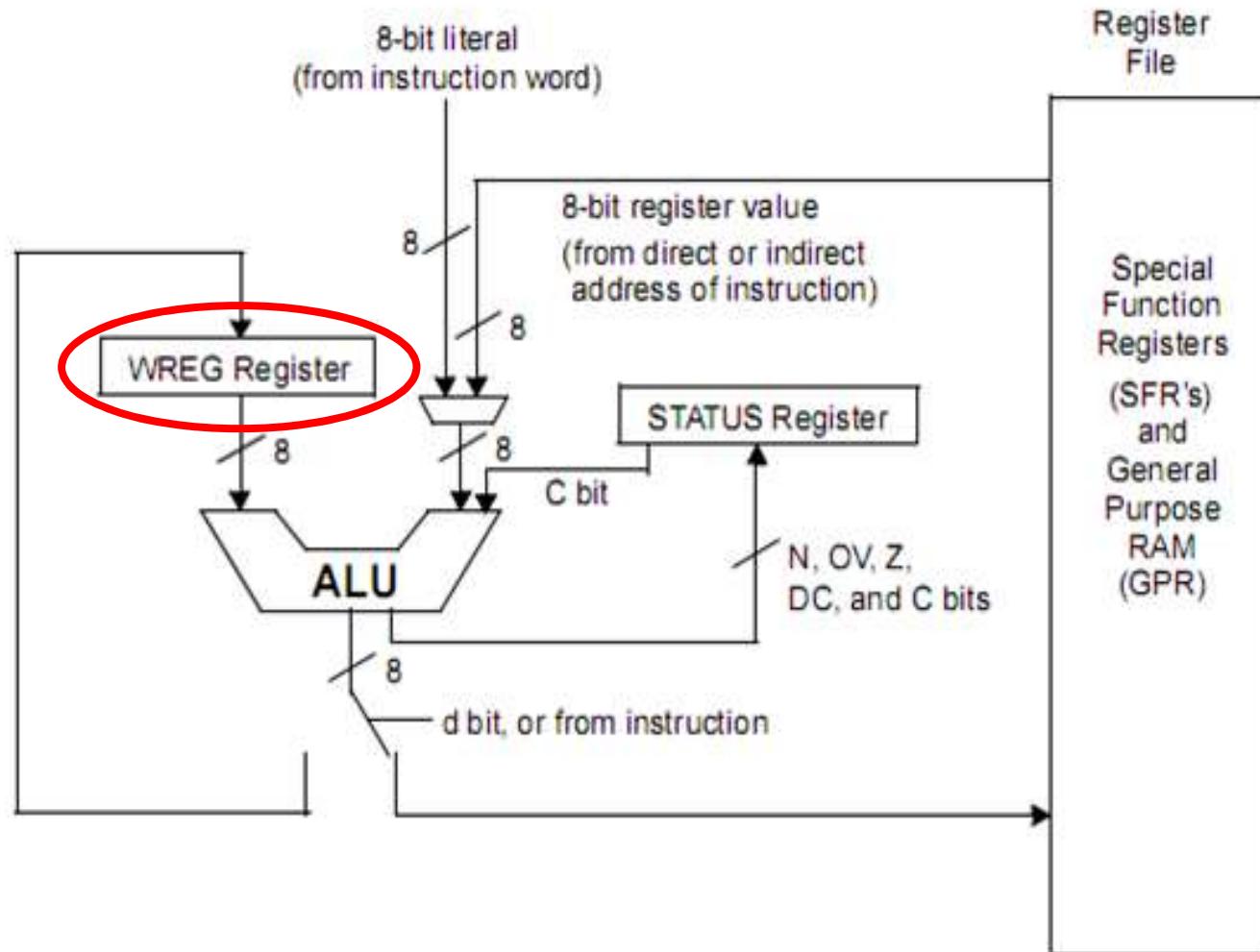
Định nghĩa địa chỉ trực tiếp trong C

- Dùng cú pháp sau cho cả RAM và SFR:
 - ❑ **volatile unsigned char <tên biến> @ <địa chỉ>;**
- Ví dụ:
 - ❑ **volatile unsigned char led @ 0xF8C;**
sẽ định nghĩa biến tên led có địa chỉ truy xuất là F8Ch (thanh ghi LATD).
 - ❑ **volatile unsigned char count @ 0x06;**
sẽ định nghĩa biến tên count là ô nhớ địa chỉ 0x06 trong RAM.

Phân nhóm thanh ghi

- Thường dùng, quan trọng
 - WREG - Working Reg. (FE8h)
 - STATUS (FD8h)
 - BSR - Bank Select Reg. Chỉ dùng 4 bit thấp.
- Có bộ thanh ghi phụ **Fast register stack** (lưu tạm STATUS, WREG và BSR) dùng cho gọi chương trình con:
 - CALL proc_name, FAST
 - RETURN, FAST
- Thanh ghi PC (program counter) 21 bit gồm: PCLATU(b_{20} - b_{16}), PCLATH (b_{15} - b_8) và PCL (b_7 - b_0).
- Thanh ghi dùng cho các chức năng đặc biệt (Timer, Interrupt, MSSP, ADC, USART, . . .)

Thanh ghi WREG



Thanh ghi STATUS

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	N	OV	Z	DC ⁽¹⁾	C ⁽²⁾
bit 7							bit 0

N (Negative) - **Còn âm** hay cò dấu cho biết kết quả là số âm

=1: Nếu kết quả là số âm

=0: Nếu kết quả là số dương

OV (Overflow) - **Còn tràn** phản ánh sự tràn ở bit dấu (MSB)

=1: Nếu kết quả tràn sang bit dấu

=0: Nếu kết quả không bị tràn

Z (Zero) - **Còn không** cho biết kết quả bằng 0

=1: Nếu kết quả bằng 0

=0: Nếu kết quả khác 0

DC (Digital Carry/Borrow) - **Còn nhớ nữa** báo tràn bit 4, dùng cho chỉnh dạng thập phân (BCD)

=1: Có nhớ ở bit 4

=0: Không nhớ bit 4

C (Carry/Borrow) - báo tràn dung lượng

=1: Có tràn về dung lượng bit

=0: Không tràn

Tra bảng dùng **CALL - ADDWF PCL - RETLW**

```
MAIN: ORG      0x0000
      MOVLW    0x00
      CALL     TABLE
      ...
      ORG      0x8000
TABLE: MOVF    PCL, F      ; A simple read of PCL will update PCLATH, PCLATU
      RLNCF    W, W      ; Multiply by 2 to get correct offset in table
      ADDWF    PCL      ; Add the modified offset to force jump into table
      RETLW    'A'
      RETLW    'B'
      RETLW    'C'
      RETLW    'D'
      RETLW    'E'
      END
```