

04. Ngắt (Interrupts)

GIỚI THIỆU NGẮT (1)

- **Ngắt** là sự xảy ra một điều kiện (sự kiện) làm cho VĐK treo tạm thời chương trình chính để thực hiện một chương trình con khác

Sau khi thực hiện xong chương trình con này thì VĐK lại tiếp tục quay lại thực hiện chương trình chính

- **Interrupt** is the signal sent to the processor to mark the event that requires immediate attention

Interrupt is “asking” the processor to stop to perform the current program and to “make time” to execute a special code

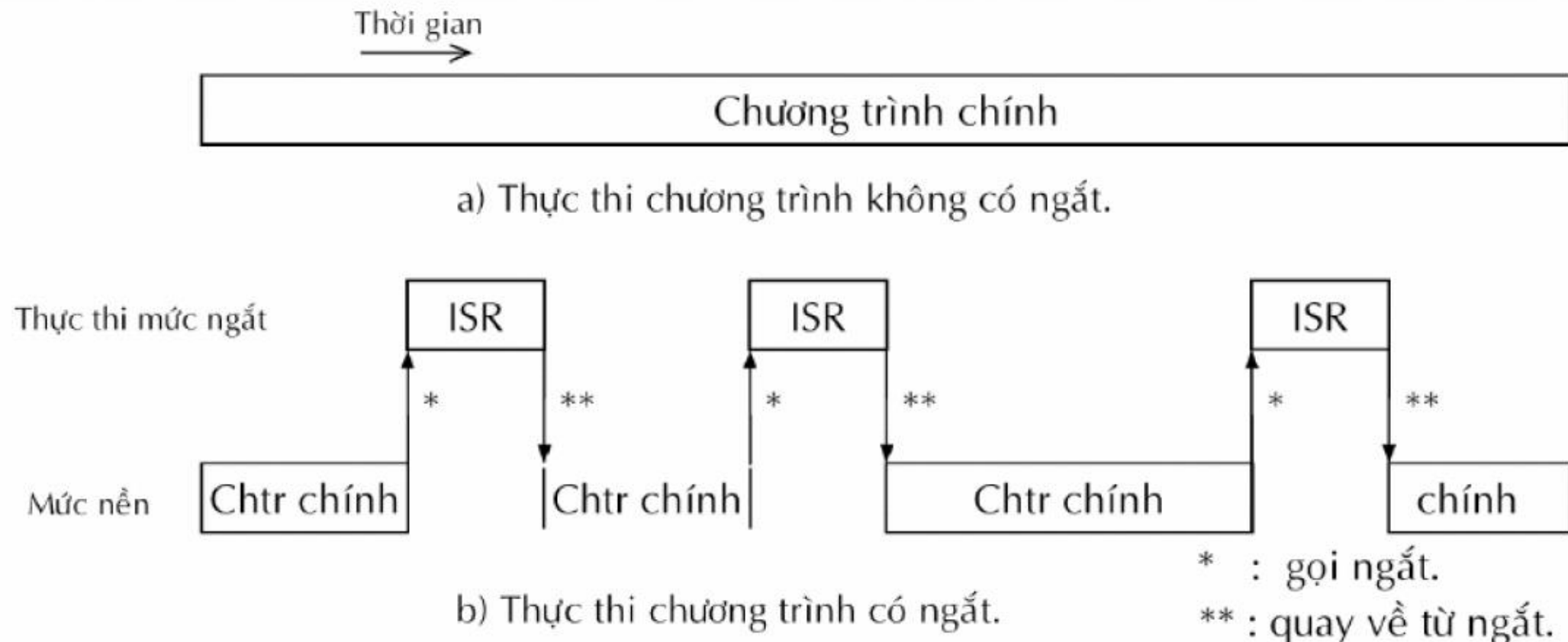
GIỚI THIỆU NGẮT (2)

- Các ngắt đóng vai trò quan trọng trong việc thiết kế và hiện thực các ứng dụng của bộ vi điều khiển
- Các ngắt cho phép hệ thống đáp ứng một sự kiện theo cách không đồng bộ và xử lý sự kiện trong khi một chương trình khác đang thực thi
- Một hệ thống được điều khiển bởi ngắt cho ta ảo tưởng đang làm nhiều công việc đồng thời

THỰC THI CHƯƠNG TRÌNH (1)

- Chương trình xử lý một ngắt được gọi là **trình phục vụ ngắt ISR** (Interrupt Service Routine) hay **quản lý ngắt** (Interrupt Handler)
- **ISR** được thực thi nhằm đáp ứng một ngắt và trong trường hợp tổng quát thực hiện việc xuất nhập đối với một thiết bị
- Khi một ngắt xuất hiện, việc thực thi chương trình chính tạm thời bị dừng và **CPU** thực hiện việc rẽ nhánh đến trình phục vụ ngắt **ISR**
- **CPU** thực thi **ISR** để thực hiện 1 công việc và sau khi kết thúc việc thực thi **ISR** thì chương trình chính lại được tiếp tục tại nơi bị tạm dừng

THỰC THI CHƯƠNG TRÌNH (2)



NGUYÊN NHÂN NGẮT (1)

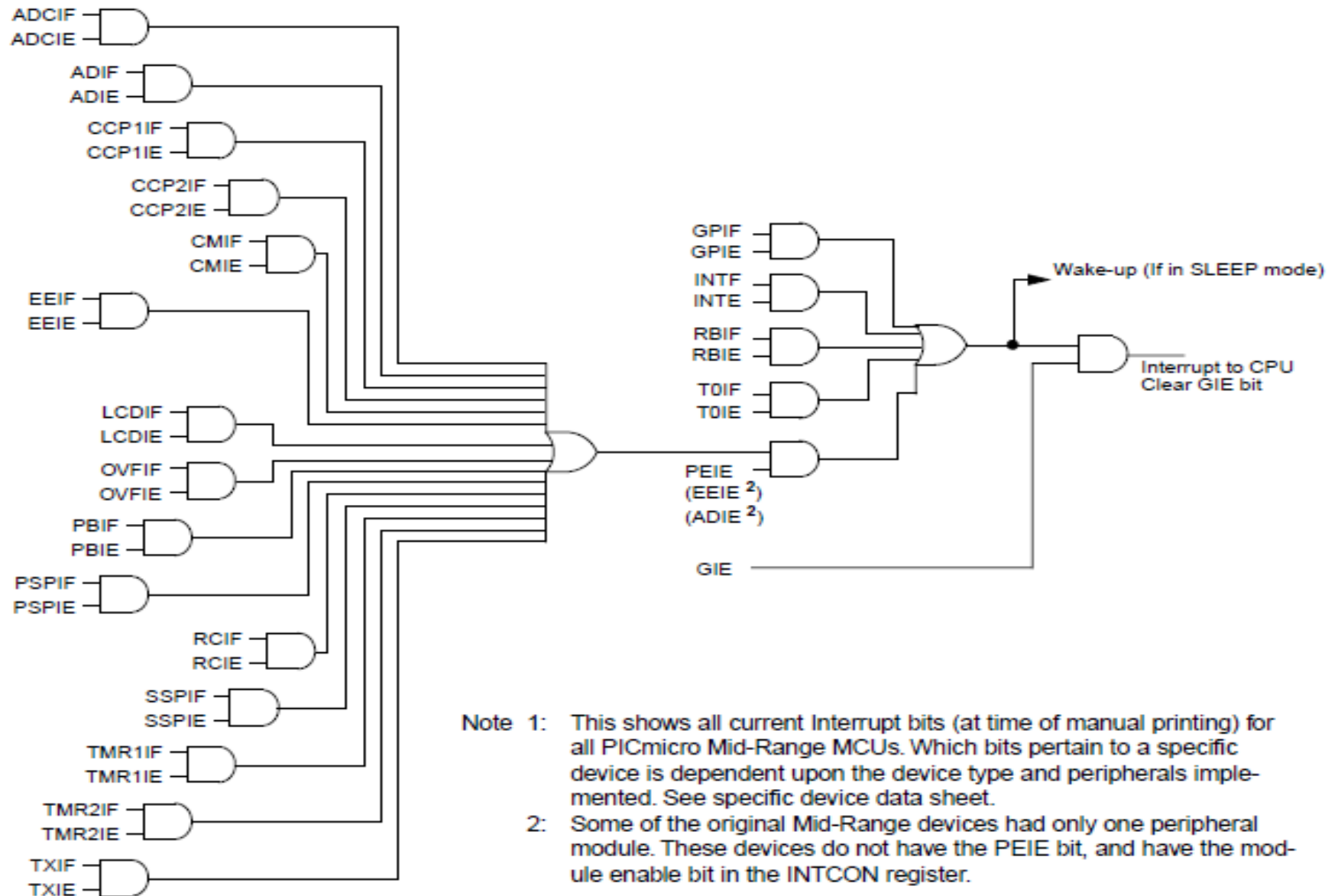
- **Ngắt nội:** Tín hiệu ngắt đến từ các thiết bị bên trong **VĐK** (VD: bit cờ ngắt được set lên 1)
- **Ngắt ngoại:** Tín hiệu ngắt đến từ các thiết bị bên ngoài **VĐK** (VD: khi chân ngắt chuyển từ mức cao về mức thấp)

NGUYÊN NHÂN NGẮT (2)

Thông thường có 14 loại ngắt:

1. **EEIF** - EEPROM Write Operation Interrupt Flag bit
2. **PSPIF** - Parallel Slave Port Read/Write Interrupt Flag bit
3. **ADIF** - A/D Converter Interrupt Flag bit
4. **RCIF** - USART Receive Interrupt Flag bit
5. **TXIF** - USART Transmit Interrupt Flag bit
6. **SSPIF** - Synchronous Serial Port (SSP) Interrupt Flag bit
7. **CCP1IF** - CCP1 Interrupt Flag bit
8. **TMR2IF** - TMR2 to PR2 Match Interrupt Flag bit
9. **TMR1IF** - TMR1 Overflow Interrupt Flag bit
10. **CCP2IF** - CCP2 Interrupt Flag bit
11. **BCLIF** - Bus Collision Interrupt Flag bit
12. **T0IF** - Timer 0 interrupt flag
13. **INTF** - RB0/INT External Interrupt Flag bit
14. **RBIF** - RB Port Change Interrupt Flag bit

NGUYÊN NHÂN NGẮT (3)



SỬ DỤNG NGẮT (1)

- **XXXIF:** là bit cờ ngắt thể hiện trạng thái của một ngắt, mang giá trị 0 là bình thường, mang giá trị 1 là ngắt xảy ra
- **XXXIE:** là bit cho phép ngắt, có thể cho phép hay ngăn không cho phép một ngắt xảy ra
- Chúng ta có thể cho phép toàn bộ hệ thống VĐK có ngắt hay không bằng cách gán giá trị cho bit cho phép ngắt toàn cục **Global Interrupt Enable bit **GIE****

SỬ DỤNG NGẮT (2)

Để có thể thực hiện một ngắt, chúng ta làm 2 bước sau:

- Bước 1: Kích khởi bit cho phép ngắt toàn cục GIE
- Bước 2: Kích khởi bit cho phép loại ngắt mà chúng ta muốn sử dụng

SỬ DỤNG NGẮT (3)

Khi có một ngắt xuất hiện và được CPU chấp nhận, chương trình chính bị ngắt và các thao tác sau đây xảy ra:

- Hoàn tất việc thực thi lệnh hiện hành
- Bộ đếm chương trình PC được cất vào stack
- Trạng thái của ngắt hiện hành được lưu giữ lại
- Các ngắt được chặn lại ở mức ngắt
- Bộ đếm chương trình PC được nạp địa chỉ vector của chương trình phục vụ ngắt ISR
- ISR được thực thi

THANH GHI INTCON (1)

REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TOIE	INTE	RBIE ^(1,3)	TOIF ⁽²⁾	INTF	RBIF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

- bit 7

GIE: Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts
- bit 6

PEIE: Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
- bit 5

TOIE: Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 interrupt

THANH GHI INTCON (2)

bit 4	INTE: INT External Interrupt Enable bit 1 = Enables the INT external interrupt 0 = Disables the INT external interrupt
bit 3	RBIE: PORTB Change Interrupt Enable bit ^(1,3) 1 = Enables the PORTB change interrupt 0 = Disables the PORTB change interrupt
bit 2	TOIF: Timer0 Overflow Interrupt Flag bit ⁽²⁾ 1 = TMR0 register has overflowed (must be cleared in software) 0 = TMR0 register did not overflow
bit 1	INTF: INT External Interrupt Flag bit 1 = The INT external interrupt occurred (must be cleared in software) 0 = The INT external interrupt did not occur
bit 0	RBIF: PORTB Change Interrupt Flag bit 1 = When at least one of the PORTB general purpose I/O pins changed state (must be cleared in software) 0 = None of the PORTB general purpose I/O pins have changed state

THANH GHI PIE1 (1)

REGISTER 2-4: **PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1**

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

- bit 7

Unimplemented: Read as ‘0’
- bit 6

ADIE: A/D Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt
- bit 5

RCIE: EUSART Receive Interrupt Enable bit
1 = Enables the EUSART receive interrupt
0 = Disables the EUSART receive interrupt

THANH GHI PIE1 (2)

bit 4	TXIE: EUSART Transmit Interrupt Enable bit 1 = Enables the EUSART transmit interrupt 0 = Disables the EUSART transmit interrupt
bit 3	SSPIE: Master Synchronous Serial Port (MSSP) Interrupt Enable bit 1 = Enables the MSSP interrupt 0 = Disables the MSSP interrupt
bit 2	CCP1IE: CCP1 Interrupt Enable bit 1 = Enables the CCP1 interrupt 0 = Disables the CCP1 interrupt
bit 1	TMR2IE: Timer2 to PR2 Match Interrupt Enable bit 1 = Enables the Timer2 to PR2 match interrupt 0 = Disables the Timer2 to PR2 match interrupt
bit 0	TMR1IE: Timer1 Overflow Interrupt Enable bit 1 = Enables the Timer1 overflow interrupt 0 = Disables the Timer1 overflow interrupt

THANH GHI PIE2 (1)

REGISTER 2-5: **PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
OSFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE	—	CCP2IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7	OSFIE: Oscillator Fail Interrupt Enable bit 1 = Enables oscillator fail interrupt 0 = Disables oscillator fail interrupt
bit 6	C2IE: Comparator C2 Interrupt Enable bit 1 = Enables Comparator C2 interrupt 0 = Disables Comparator C2 interrupt
bit 5	C1IE: Comparator C1 Interrupt Enable bit 1 = Enables Comparator C1 interrupt 0 = Disables Comparator C1 interrupt

THANH GHI PIE2 (2)

bit 4	EEIE: EEPROM Write Operation Interrupt Enable bit 1 = Enables EEPROM write operation interrupt 0 = Disables EEPROM write operation interrupt
bit 3	BCLIE: Bus Collision Interrupt Enable bit 1 = Enables Bus Collision interrupt 0 = Disables Bus Collision interrupt
bit 2	ULPWUIE: Ultra Low-Power Wake-up Interrupt Enable bit 1 = Enables Ultra Low-Power Wake-up interrupt 0 = Disables Ultra Low-Power Wake-up interrupt
bit 1	Unimplemented: Read as '0'
bit 0	CCP2IE: CCP2 Interrupt Enable bit 1 = Enables CCP2 interrupt 0 = Disables CCP2 interrupt

THANH GHI PIR1 (1)

REGISTER 2-6: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7

Unimplemented: Read as '0'
- bit 6

ADIF: A/D Converter Interrupt Flag bit

1 = A/D conversion complete (must be cleared in software)

0 = A/D conversion has not completed or has not been started
- bit 5

RCIF: EUSART Receive Interrupt Flag bit

1 = The EUSART receive buffer is full (cleared by reading RCREG)

0 = The EUSART receive buffer is not full
- HCM City Univ. of Technology, Faculty of Mechanical Engineering

Phung Tri Cong

THANH GHI PIR1 (2)

bit 4

TXIF: EUSART Transmit Interrupt Flag bit

1 = The EUSART transmit buffer is empty (cleared by writing to TXREG)

0 = The EUSART transmit buffer is full

bit 3

SSPIF: Master Synchronous Serial Port (MSSP) Interrupt Flag bit

1 = The MSSP interrupt condition has occurred, and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:

SPI

A transmission/reception has taken place

I²C Slave/Master

A transmission/reception has taken place

I²C Master

The initiated Start condition was completed by the MSSP module

The initiated Stop condition was completed by the MSSP module

The initiated restart condition was completed by the MSSP module

The initiated Acknowledge condition was completed by the MSSP module

A Start condition occurred while the MSSP module was idle (Multi-master system)

A Stop condition occurred while the MSSP module was idle (Multi-master system)

0 = No MSSP interrupt condition has occurred

THANH GHI PIR1 (3)

bit 2

CCP1IF: CCP1 Interrupt Flag bit

Capture mode:

1 = A TMR1 register capture occurred (must be cleared in software)

0 = No TMR1 register capture occurred

Compare mode:

1 = A TMR1 register compare match occurred (must be cleared in software)

0 = No TMR1 register compare match occurred

PWM mode:

Unused in this mode

bit 1

TMR2IF: Timer2 to PR2 Interrupt Flag bit

1 = A Timer2 to PR2 match occurred (must be cleared in software)

0 = No Timer2 to PR2 match occurred

bit 0

TMR1IF: Timer1 Overflow Interrupt Flag bit

1 = The TMR1 register overflowed (must be cleared in software)

0 = The TMR1 register did not overflow

THANH GHI PIR2 (1)

REGISTER 2-7: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	—	CCP2IF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7	OSFIF: Oscillator Fail Interrupt Flag bit 1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software) 0 = System clock operating
bit 6	C2IF: Comparator C2 Interrupt Flag bit 1 = Comparator output (C2OUT bit) has changed (must be cleared in software) 0 = Comparator output (C2OUT bit) has not changed
bit 5	C1IF: Comparator C1 Interrupt Flag bit 1 = Comparator output (C1OUT bit) has changed (must be cleared in software) 0 = Comparator output (C1OUT bit) has not changed

THANH GHI PIR2 (2)

bit 4	EEIF: EE Write Operation Interrupt Flag bit 1 = Write operation completed (must be cleared in software) 0 = Write operation has not completed or has not started
bit 3	BCLIF: Bus Collision Interrupt Flag bit 1 = A bus collision has occurred in the MSSP when configured for I ² C Master mode 0 = No bus collision has occurred
bit 2	ULPWUIF: Ultra Low-Power Wake-up Interrupt Flag bit 1 = Wake-up condition has occurred (must be cleared in software) 0 = No Wake-up condition has occurred
bit 1	Unimplemented: Read as '0'
bit 0	CCP2IF: CCP2 Interrupt Flag bit <u>Capture mode:</u> 1 = A TMR1 register capture occurred (must be cleared in software) 0 = No TMR1 register capture occurred <u>Compare mode:</u> 1 = A TMR1 register compare match occurred (must be cleared in software) 0 = No TMR1 register compare match occurred <u>PWM mode:</u> Unused in this mode

CÁC LỆNH SỬ DỤNG NGẮT

Chương trình con

#INT_EXT

#INT_RB

Khai báo ngắt

enable_interrupts(GLOBAL);

enable_interrupts(INT_EXT);

enable_interrupts(INT_RB);

Ext_int_edge(H_to_L)

VÍ DỤ 1: SỬ DỤNG NGẮT NGOÀI

Viết chương trình đếm số lần nhấn nút dừng ngắt ngoài và hiển thị kết quả ra LED 7 đoạn?

VÍ DỤ 2: SỬ DỤNG NGẮT NGOÀI

Viết chương trình sử dụng ngắt ngoài bằng cách tăng giá trị của PORTB mỗi lần 1 đơn vị và hiển thị kết quả ra LED 7 đoạn?

VÍ DỤ 3: SỬ DỤNG NGẮT PORTB

Viết chương trình sử dụng ngắt PORTB và hiển thị kết quả ra PORTD?