INTERRUPCIONES PARA PIC18F46K22

Definición

- La ejecución de un programa se realiza de manera secuencial. Una instrucción tras otra
- De ser necesario, se utilizan estructuras para controlar el flujo del programa (if, if-else, for, while, switch-case)
- Una interrupción es una "señal" interna o externa de hardware (en algunos dispositivos puede ser también por software) que ocasiona que el flujo de un programa se redireccione a una función específica para ejecutar un conjunto de instrucciones.
- Es un tipo de control del flujo de programa ejercido no por las estructuras de programación convencionales si no para el manejo de "un evento específico" ocurrido dentro o fuera del microcontrolador.

Términos utilizados

- A los “entes” que pueden generar una interrupción se les denomina “fuentes de interrupción”.
- Cuando ocurre una interrupción, el programa salta a una dirección fija específica de la memoria de programa. A esa dirección de salto se le denomina “vector de interrupción”.
- La dirección del vector de interrupción no se puede cambiar; es algo definido por el fabricante del dispositivo. Típicamente en el vector de interrupción se aloja una instrucción de salto para redirigirse a las instrucciones que se ejecutarán cuando ocurre una interrupción.
- A la función (o procedimiento) que se ejecuta al ocurrir una interrupción se le denomina “rutina de servicio a la interrupción” (en inglés ISR).
- Se tienen prioridades de atención a las interrupciones: prioridad alta y prioridad baja

Fuentes de Interrupción

- El microcontrolador PIC18F46K22 posee las siguientes fuentes de interrupción (las mas usadas):
  - Interrupción externa por INTx.
  - Interrupción por cambio de nivel lógico en RB4 -RB7.
  - Interrupción por desborde de los timers.
  - Interrupción del transmisor del modulo USART.
  - Interrupción del receptor del modulo USART.
  - Interrupción del modulo CPP (comparador).
  - Interrupción del ADC.

Control de Interrupciones

- Para controlar toda la lógica de interrupciones, se tienen disponibles 19 registros. No todos se configuran al mismo tiempo; todo depende de las fuentes de interrupción que se desee controlar.
- Los registros de configuración son:
  - RCON
  - INTCON, INTCON2, INTCON3
  - PIR1, PIR2, PIR3, PIR4, PIR5
  - PIE1, PIE2, PIE3, PIE4, PIE5
  - IPR1, IPR2, IPR3, IPR4, IPR5

Mapa de Memoria

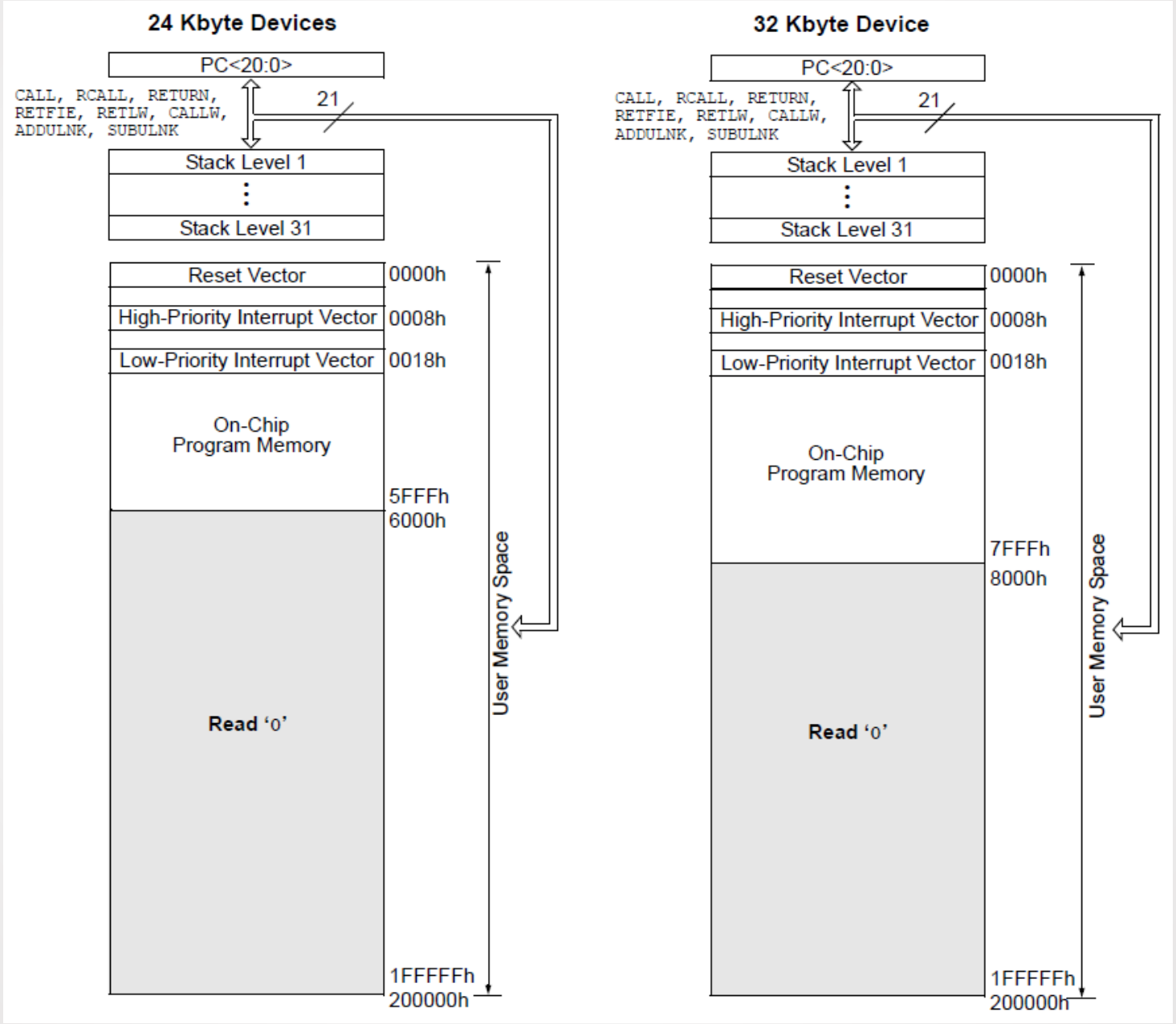
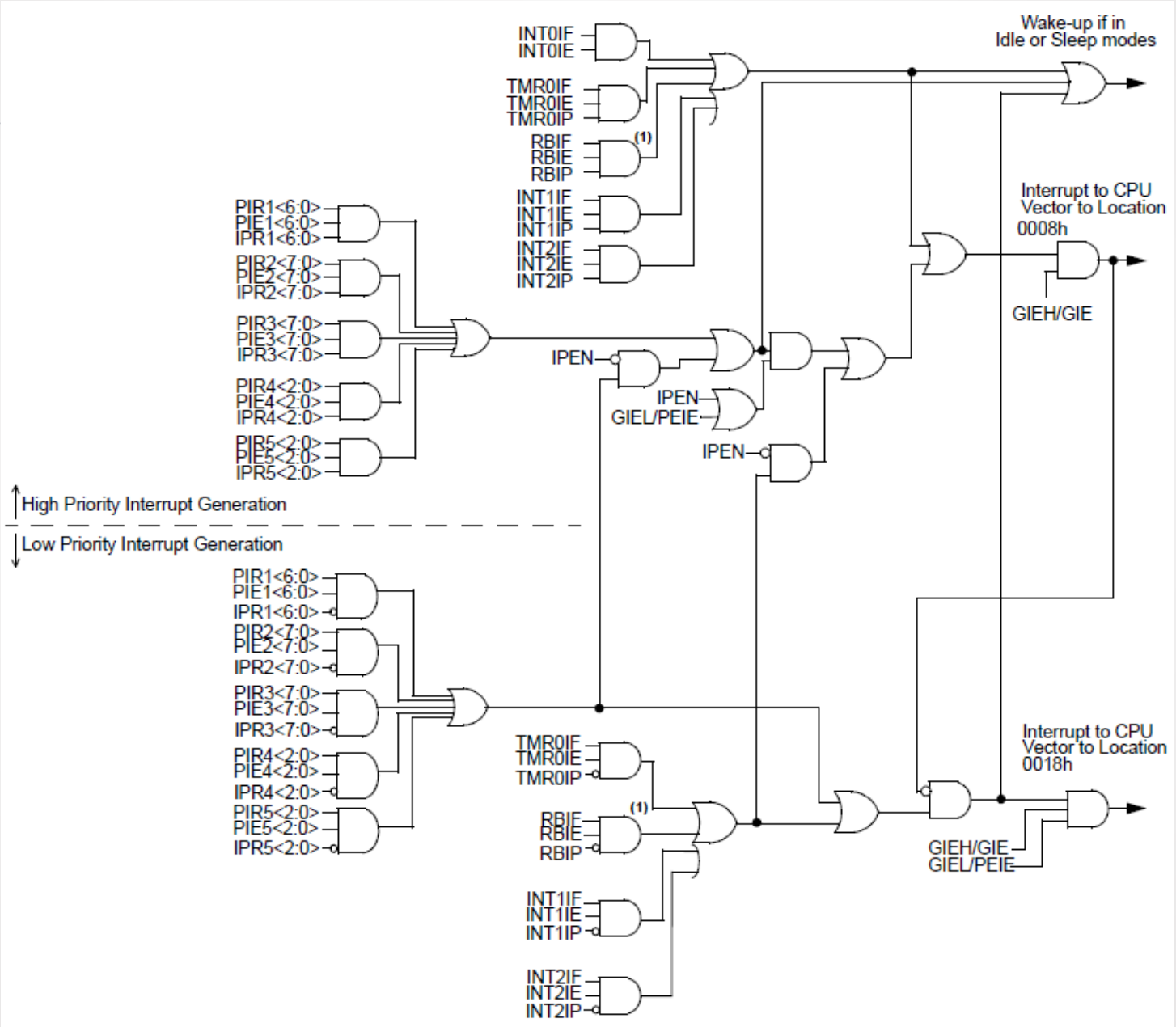


Diagrama lógico de interrupciones



Manejo de Interrupciones

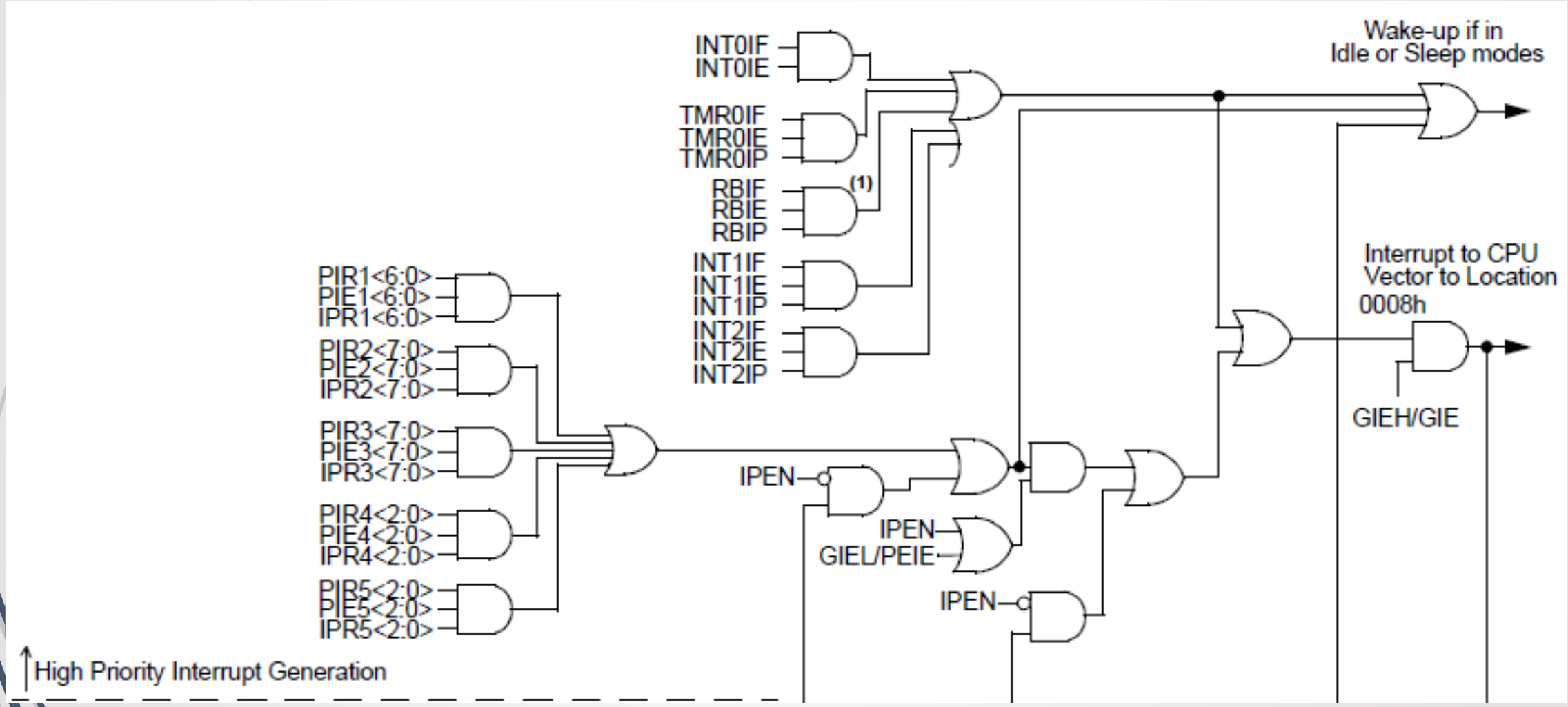
- Se deben configurar bits independientes dentro de los registros de configuración para determinar lo que se hará cuando sucede una interrupción
- La rutina de manejo de la interrupción debe identificar qué evento interrumpió el programa para ejecutar las instrucciones correspondientes.
- Cuando la rutina de manejo de la interrupción finaliza (RETFIE) el programa retorna al punto donde se interrumpió el flujo del programa.

Manejo de Interrupciones

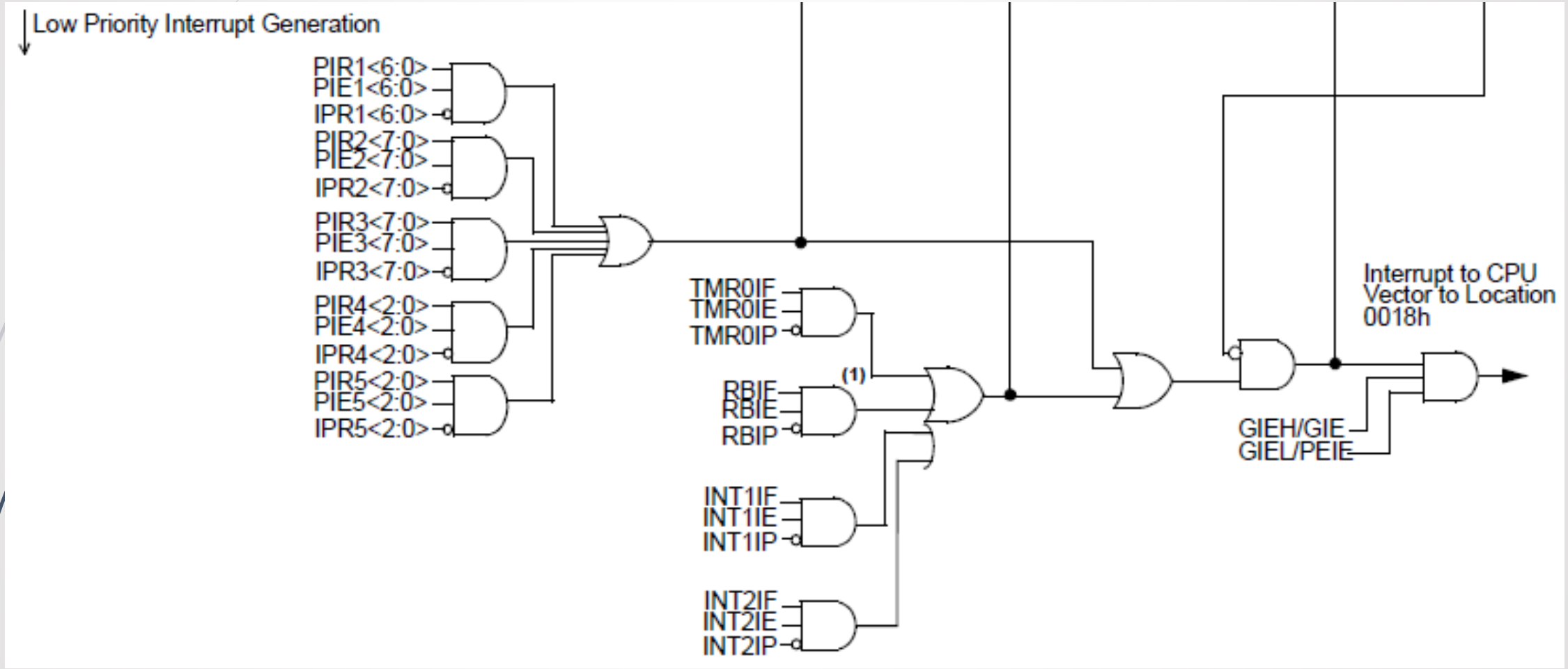
Cada fuente de interrupción tiene asociados tres bits dentro de los registros de control de interrupciones, esto son:

Bit bandera (**Flag-bit**)	• indica si el evento de interrupción asociado a este bit ha ocurrido.
Bit de habilitación (**Enable-bit**):	• Permite la habilitación de una interrupción específica
Bit de prioridad (**Priority-bit**)	• Define la prioridad de la interrupción

Un poco de más atención



Un poco de más atención



Descripción de Registros
INTCON (Registro de control de interrupciones)

REGISTER 9-1: 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/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7							bit 0

bit 7 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts including peripherals
When IPEN = 1:
1 = Enables all high priority interrupts
0 = Disables all interrupts including low priority

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1:
1 = Enables all low priority interrupts
0 = Disables all low priority interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 **RBIE:** Port B Interrupt-On-Change (IOCx) Interrupt Enable bit(2)
1 = Enables the IOCx port change interrupt
0 = Disables the IOCx port change interrupt

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared by software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared by software)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** Port B Interrupt-On-Change (IOCx) Interrupt Flag bit(1)
1 = At least one of the IOC<3:0> (RB<7:4>) pins changed state (must be cleared by software)
0 = None of the IOC<3:0> (RB<7:4>) pins have changed state

Note 1: A mismatch condition will continue to set the RBIF bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.
2: RB port change interrupts also require the individual pin IOCB enables.

Descripción de Registros
INTCONT2 (Registro de control de interrupciones 2)

REGISTER 9-2: INTCON2: INTERRUPT CONTROL 2 REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
bit 7							bit 0

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled provided that the pin is an input and the corresponding WPUB bit is set.

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **Unimplemented:** Read as '0'

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **Unimplemented:** Read as '0'

bit 0 **RBIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

Descripción de Registros
INTCONT3 (Registro de control de interrupciones 3)

REGISTER 9-3: INTCON3: INTERRUPT CONTROL 3 REGISTER

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 0

bit 7 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **Unimplemented:** Read as '0'

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

bit 2 **Unimplemented:** Read as '0'

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared by software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared by software)
0 = The INT1 external interrupt did not occur

Descripción de Registros RCON (Control de reset)

REGISTER 4-1: RCON: RESET CONTROL REGISTER

R/W-0/0	R/W-q/u	U-0	R/W-1/q	R-1/q	R-1/q	R/W-q/u	R/W-0/q
IPEN	SBOREN(1)	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$(2)	$\overline{BOR}$
bit 7							bit 0

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)
For details of bit operation, see Register 4-1.

bit 5 **Unimplemented:** Read as '0'

bit 4 **$\overline{RI}$:** RESET Instruction Flag bit
For details of bit operation, see Register 4-1.

bit 3 **$\overline{TO}$:** Watchdog Time-out Flag bit
For details of bit operation, see Register 4-1.

bit 2 **$\overline{PD}$:** Power-Down Detection Flag bit
For details of bit operation, see Register 4-1.

bit 1 **$\overline{POR}$:** Power-on Reset Status bit(2)
For details of bit operation, see Register 4-1.

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
For details of bit operation, see Register 4-1.

Descripción de Registros para la configuración de interrupciones

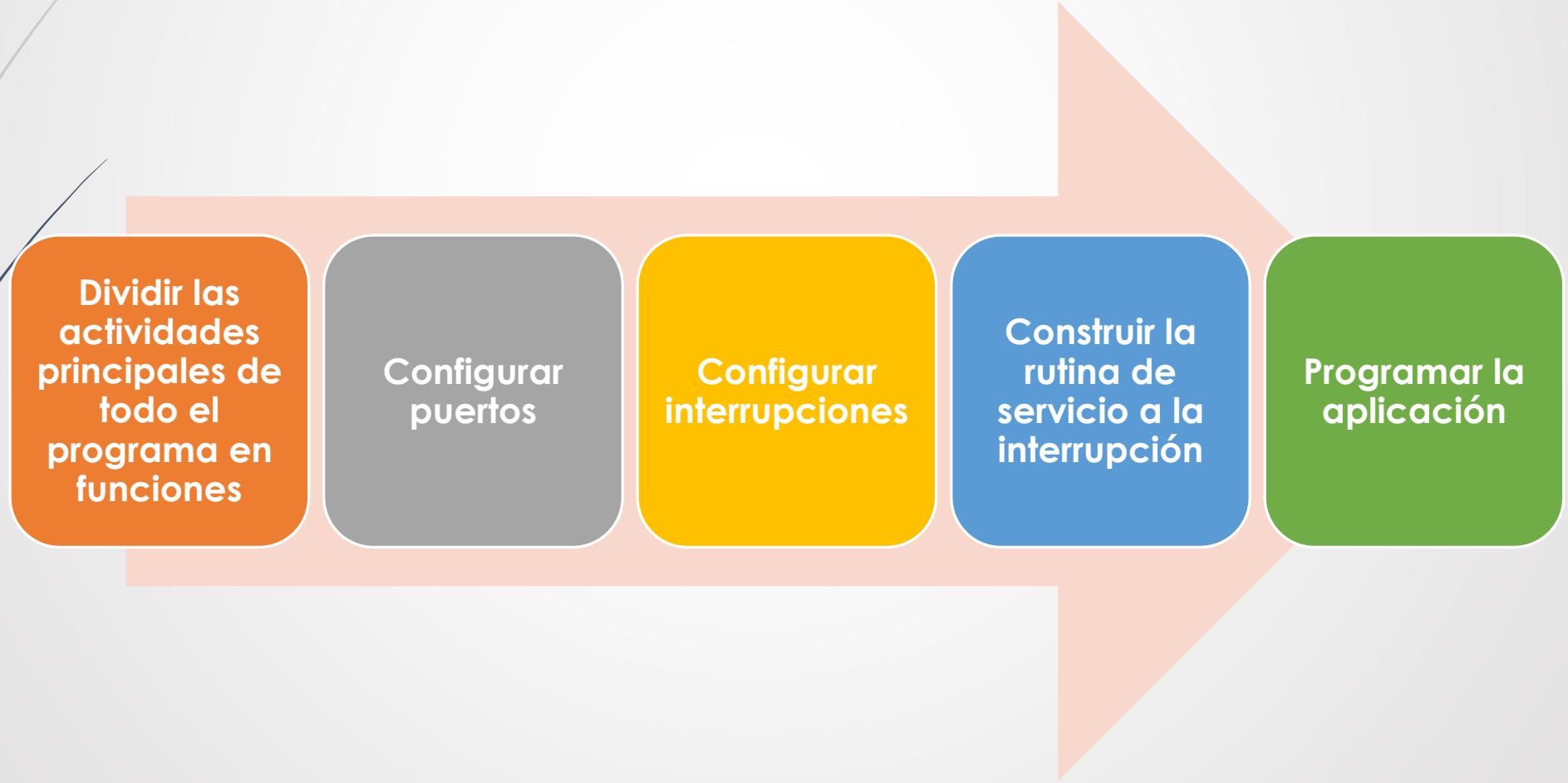
Pero faltan varios registros más por explicar…. Para ser exactos faltan por describir 15 registros¡¡¡

- PIR1, PIR2, PIR3, PIR4, PIR5
- PIE1, PIE2, PIE3, PIE4, PIE5
- IPR1, IPR2, IPR3, IPR4, IPR5

La descripción de estos registros se abordaran en temas posteriores cuando se utilicen los periféricos internos del microcontrolador.

¿CÓMO PROGRAMAR UNA INTERRUPCIÓN?

Algunos tips para programar interrupciones



¿CÓMO PROGRAMAR UNA ISR?

```
void __interrupt () nombre_de_rutina(void)
{

    INSTRUCCIONES A EJECUTAR CUANDO OCURRE UNA INTERRUPCIÓN

}
```

La rutina de interrupción no necesita tener un prototipo de función como ocurre con una función “típica” de programación.

La rutina de interrupción puede considerarse una función “especial” dentro de todo el conjunto de funciones que tenga el programa y tiene casi las mismas funcionalidades que una función “típica” de programación.