

MICROCONTROLADORES

LENGUAJES DE PROGRAMACIÓN

- LENGUAJE ENSAMBLADOR
- LENGUAJE C
- LENGUAJE BASIC

LENGUAJE BASIC

Código simbólico de instrucciones de propósito general para principiantes.

Creado por John George Kemeny y Thomas Eugene Kurtz en el año 1964, se basaron en otros dos lenguajes de programación como el FORTRAN II y el Algol 60, haciéndolo que este lenguaje fuese adecuado para el uso del computador a tiempo compartido y para la aritmética de matrices

LENGUAJE BASIC

Ventajas:

- Es un lenguaje muy simple y con instrucciones fácilmente legibles, incluso por no expertos.

Desventajas:

- Nunca se va a tener el control del programa en cuanto tiempos de ejecución y control de registros bit a bit.
- Es muy complicado el manejo de interrupciones simultáneas en este lenguaje.
- Tiene limitaciones cuando genera el archivo .hex, no optimiza el tamaño.
- La mayoría de compiladores para este lenguaje pueden utilizarse únicamente bajo ambiente Windows.

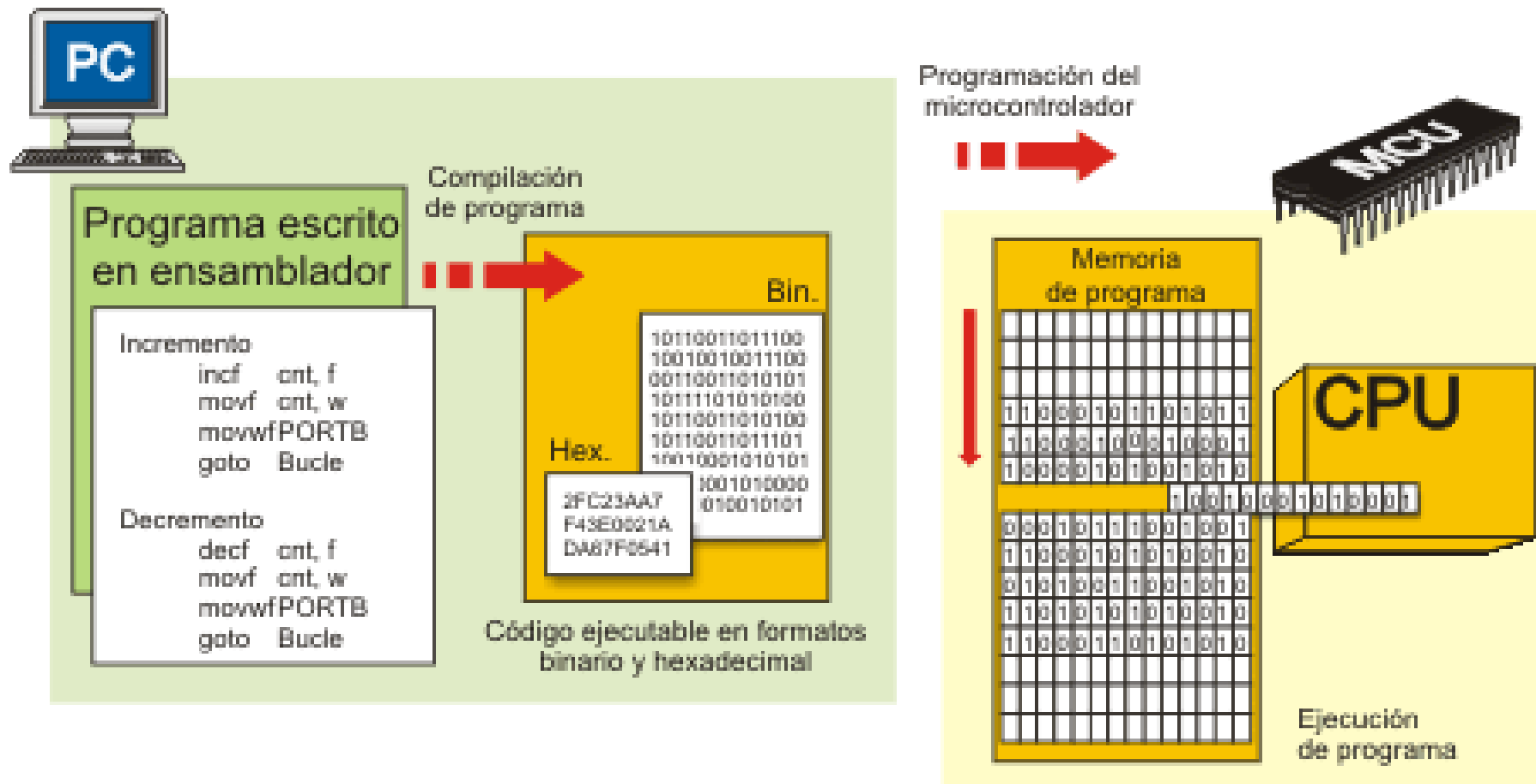
LENGUAJE ASEMBLER

Uno de los primeros lenguajes de programación, facil de escribir y entender.

Las introducciones en ensamblador son abreviaturas con significado y a cada una le corresponde una localidad de memoria.

Los programas denominados ensamblador compila o traduce las instrucciones del lengua a código maquinal.

LENGUAJE ENSAMBLADOR



LENGUAJE ASEMBLER

Ventajas:

- Es el lenguaje de bajo nivel natural de la línea PIC tanto para gama baja, media o alta.
- Con el se tiene un aprovechamiento eficiente de los recursos del PIC.
- Se pueden crear macros con este lenguaje, para después simplificar el código en diferentes desarrollos.
- Con el se pueden controlar los tiempos y los registros bit a bit.
- Excelente para manejar interrupciones simultáneas.
- Cuando se genera el archivo .hex éste es completamente optimizado.

LENGUAJE ASEMBLER

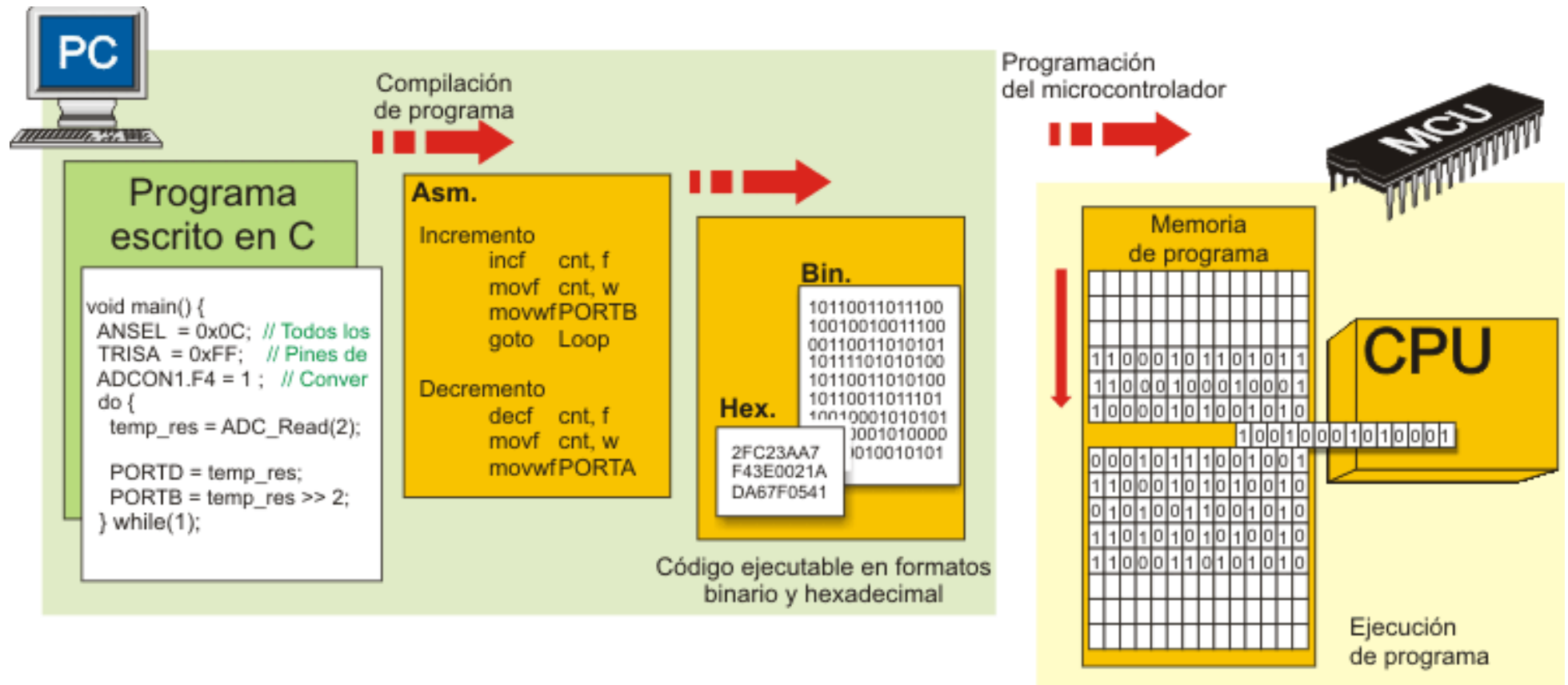
Desventajas:

- Tiempos de desarrollo lentos para algunas rutinas en comparación con los otros lenguajes.
- Incluso una sola operación en el programa escrito en ensamblador consiste en muchas instrucciones, haciéndolo muy largo y difícil de manejar.
- Un programador tiene que conocer el hardware del microcontrolador para escribir un programa.
- Cada tipo de microcontrolador tiene su propio conjunto de instrucciones que un programador tiene que conocer para escribir un programa

LENGUAJE C

El lenguaje C dispone de todas las ventajas de un lenguaje de programación de alto nivel y permite realizar algunas operaciones tanto sobre los bytes como sobre los bits. Las características de C pueden ser muy útiles al programar los microcontroladores. Además, C está estandarizado (el estándar ANSI), es muy portable, así que el mismo código se puede utilizar muchas veces en diferentes proyectos

LENGUAJE C



Ventajas:

- Es un lenguaje de alto nivel mas cercano a la máquina.
- Se puede construir rutinas matemáticas fácilmente.
- Se puede combinar con ensamblador.
- Se pueden crear macros con este lenguaje, para después simplificar el código en diferentes desarrollos.

LENGUAJES C

Desventajas:

- Los programas al compilarlos pueden resultar un poco extensos y pesados por ello debe tenerse en cuenta la capacidad de memoria de programa del PIC a utilizar.
- Con este lenguaje no se puede controlar los tiempos.

PIC16F887

Arquitectura RISC

El microcontrolador cuenta con solo 35 instrucciones diferentes

Todas las instrucciones son uni-ciclo excepto por las de ramificación

Frecuencia de operación 0-20 MHz

Oscilador interno de alta precisión

Calibrado de fábrica

Rango de frecuencia de 8MHz a 31KHz seleccionado por software

Voltaje de la fuente de alimentación de 2.0V a 5.5V

Consumo: 220uA (2.0V, 4MHz), 11uA (2.0 V, 32 KHz), 50nA

PIC16F887

Brown-out Reset (BOR) con opción para controlar por software

35 pines de entrada/salida

Alta corriente de fuente y de drenador para manejo de LED

Resistencias pull-up programables

individualmente por software

Interrupción al cambiar el estado del pin

Memoria ROM de 8K con tecnología FLASH

El chip se puede re-programar hasta 100.000 veces

PIC16F887

Opción de programación serial en el circuito

256 bytes de memoria EEPROM

Los datos se pueden grabar más de 1.000.000 veces

368 bytes de memoria RAM

Convertidor A/D:

14 canales

Resolución de 10 bits

PIC16F887

3 temporizadores/contadores independientes

Temporizador perro guardián

Módulo comparador analógico con

Dos comparadores analógicos

Referencia de voltaje fija (0.6V)

Referencia de voltaje programable en el chip

Módulo PWM incorporado

Módulo USART mejorado

Soporta las comunicaciones seriales RS-485, RS-232 y LIN2.0

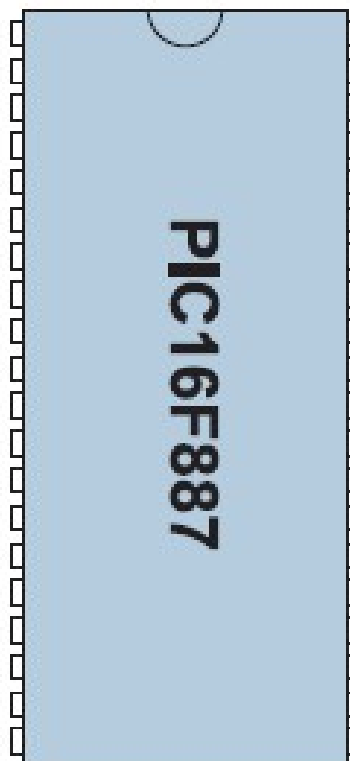
Auto detección de baudios

Puerto Serie Síncrono Maestro (MSSP)

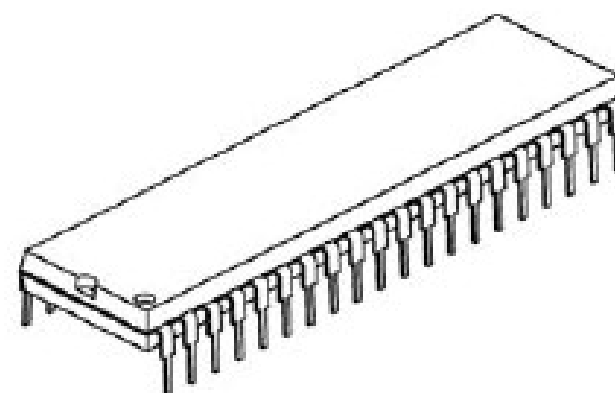
Soporta los modos SPI e I2C

PIC16F887

RE3/MCLR/Vpp
 RA0/AN0/ULPWU/C12IN0-
 RA1/AN1/C12IN1-
 RA2/AN2/Vref-/CVref/C2IN+
 RA3/AN3/Vref+/C1IN+
 RA4/T0CK/C1OUT
 RA5/AN4/SS/C2OUT
 RE0/AN5
 RE1/AN6
 RE2/AN7
 Vdd
 Vss
 RA7/OSC1/CLKIN
 RA6/OSC2/CLKOUT
 RC0/T1OSO/T1CKI
 RC1/T1OSI/CCP2
 RC2/P1A/CCP1
 RC3/SCK/SCL
 RD0
 RD1

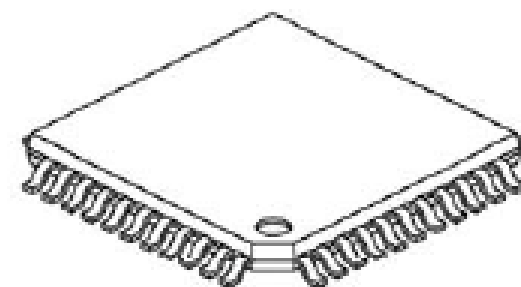
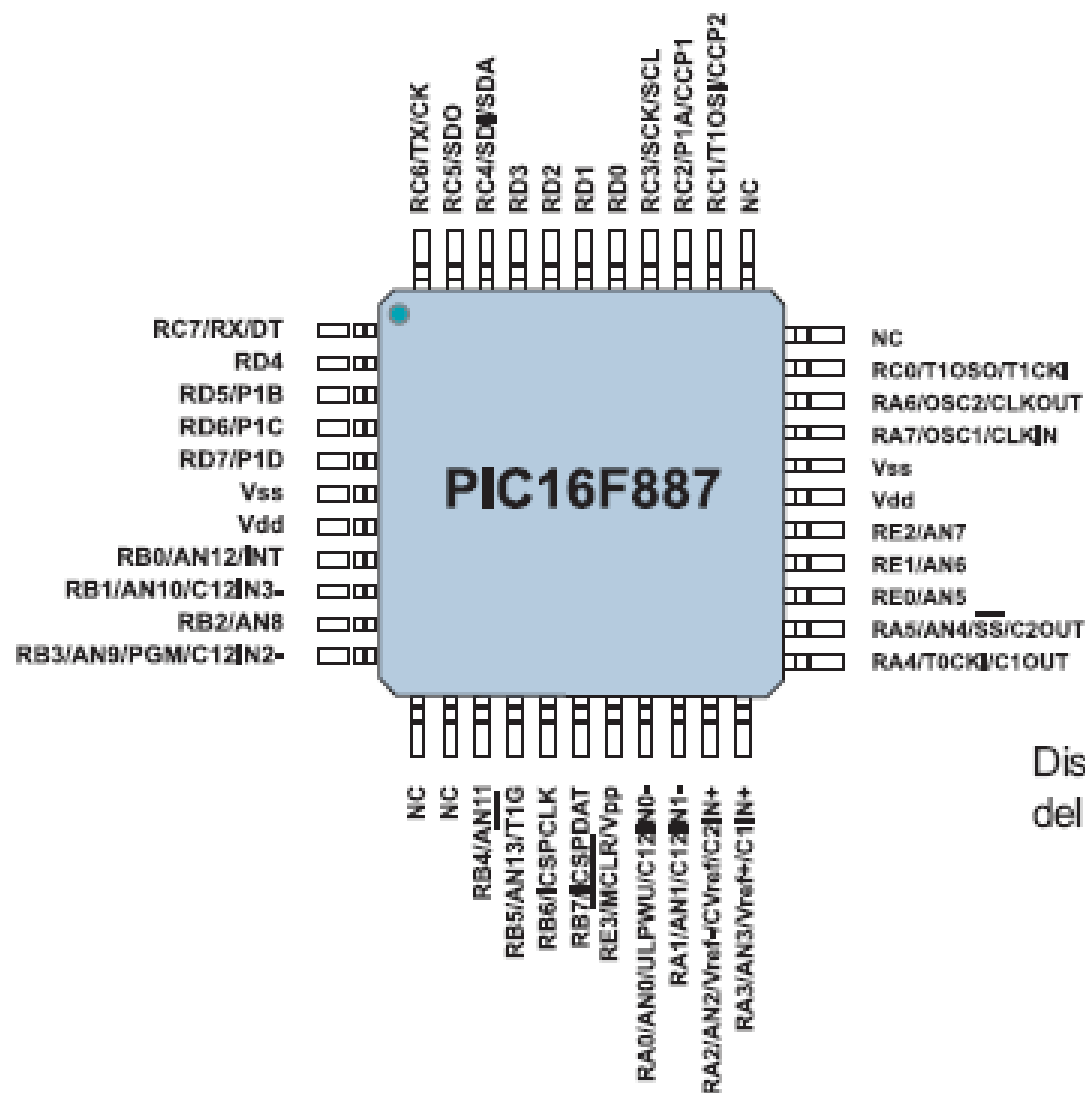


RB7/CSPDAT
 RB6/CSPCLK
 RB5/AN13/T1G
 RB4/AN11
 RB3/AN9/PGM/C12IN2-
 RB2/AN8
 RB1/AN10/C12IN3-
 RB0/AN12/INT
 Vdd
 Vss
 RD7/P1D
 RD6/P1C
 RD5/P1B
 RD4
 RC7/RX/DT
 RC6/TX/CK
 RC5/SDO
 RC4/SDI/SDA
 RD3
 RD2



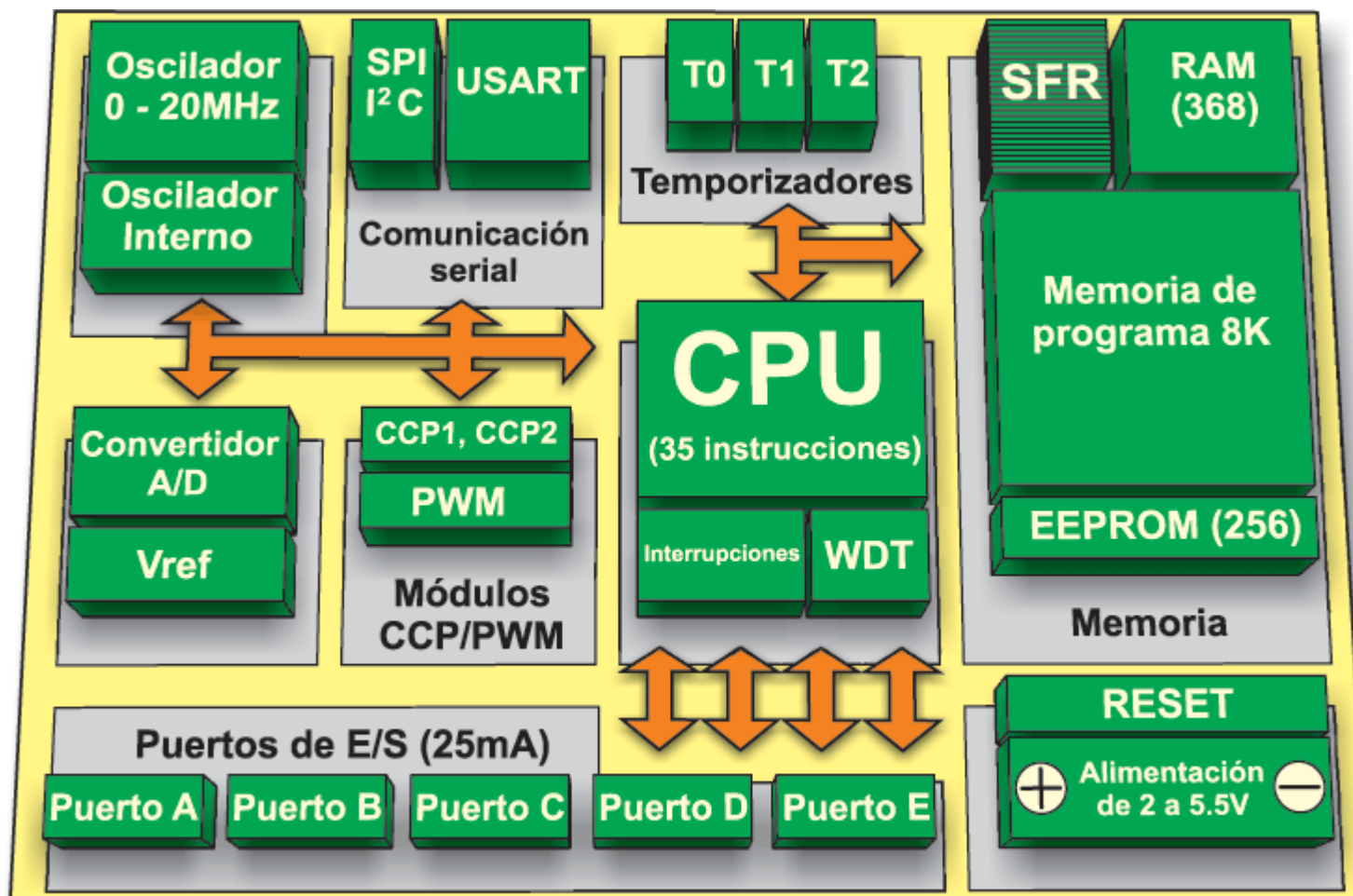
Disposición de pines en el encapsulado DIP40 del microcontrolador PIC16F887

PIC16F887



Disposición de pines en el encapsulado QFN44 del microcontrolador PIC16F887

PIC16F887



PIC16F887

Nombre	Número (DIP 40)	Función	Descripción
RE3/MCLR/Vpp	1	RE3	Entrada de propósito general en el puerto PORTE
		MCLR	Pin de reinicio. El nivel lógico bajo en este pin reinicia al microcontrolador
		Vpp	Voltaje de programación
RA0/AN0/ULPWU/C12IN0-	2	RA0	E/S de propósito general en el puerto PORTA
		AN0	Entrada del canal 0 del convertidor A/D
		ULPWU	Entrada de desactivar el modo de espera
		C12IN0-	Entrada negativa del comparador C1 o C2
RA1/AN1/C12IN1-	3	RA1	E/S de propósito general en el puerto A
		AN1	Canal 1 del convertidor A/D
		C12IN1-	Entrada negativa del comparador C1 o C2
RA2/AN2/Vref-/CVref/C2IN+	4	RA2	E/S de propósito general en el puerto PORTA
		AN2	Canal 2 del convertidor A/D
		Vref-	Entrada de referencia negativa de voltaje del convertidor A/D
		CVref	Salida de referencia de voltaje del comparador
		C2IN+	Entrada positiva del comparador C2
RA3/AN3/Vref+/C1IN+	5	RA3	E/S de propósito general en el puerto PORTA
		AN3	Canal 3 del convertidor A/D
		Vref+	Entrada de referencia positiva de voltaje del convertidor A/D
		C1IN+	Entrada positiva del comparador C1

PIC16F887

RA4/T0CKI/C1OUT	6	RA4	E/S de propósito general en el puerto PORTA
		T0CKI	Entrada de reloj del temporizador T0
		C1OUT	Salida del comparador C1
RA5/AN4/SS/C2OUT	7	RA5	E/S de propósito general en el puerto PORTA
		AN4	Canal 4 del convertidor A/D
		SS	Entrada del módulo SPI (<i>Selección del esclavo</i>)
		C2OUT	Salida del comparador C2
RE0/AN5	8	RE0	E/S de propósito general en el puerto PORTE
		AN5	Canal 5 del convertidor A/D
RE1/AN6	9	RE1	E/S de propósito general en el puerto PORTE
		AN6	Canal 6 del convertidor A/D
RE2/AN7	10	RE2	E/S de propósito general en el puerto PORTE
		AN7	Canal 7 del convertidor A/D
Vdd	11	+	Suministro de voltaje positivo
Vss	12	-	Tierra (ground - GND)

PIC16F887

Nombre	Número (DIP 40)	Función	Descripción
RA7/OSC1/CLKIN <small>oscilador C2 verificador A/D</small>	13	RA7	E/S de propósito general en el puerto PORTA
		OSC1	Entrada del oscilador de cristal
		CLKIN	Entrada del reloj externo
RA6/OSC2/CLKOUT <small>inversor A/D verificador A/D</small>	14	OSC2	Salida del oscilador del cristal
		CLKO	Salida en la que se presenta la señal Fosc/4
		RA6	E/S de propósito general en el puerto PORTA
RC0/T1OSO/T1CKI <small>GND</small>	15	RC0	E/S de propósito general en el puerto PORTC
		T1OSO	Salida del oscilador del temporizador 1
		T1CKI	Entrada de reloj del temporizador 1
RC1/T1OSO/T1CKI	16	RC1	E/S de propósito general en el puerto PORTC
		T1OSI	Entrada del oscilador del temporizador 1
		CCP2	E/S de los módulos CCP1 y PWM1
RC2/P1A/CCP1	17	RC2	E/S de propósito general en el puerto PORTC
		P1A	Salida del módulo PWM
		CCP1	E/S de los módulos CCP1 y PWM1

PIC16F887

RC3/SCK/SCL	18	RC3	E/S de propósito general en el puerto PORTC
		SCK	E/S de reloj del módulo MSSP en el modo SPI
		SCL	E/S de reloj del módulo MSSP en el modo I ² C
RD0	19	RD0	E/S de propósito general en el puerto PORTD
RD1	20	RD1	E/S de propósito general en el puerto PORTD
RD2	21	RD2	E/S de propósito general en el puerto PORTD
RD3	22	RD3	E/S de propósito general en el puerto PORTD
RC4/SDI/SDA	23	RC4	E/S de propósito general en el puerto PORTC
		SDI	Entrada <i>Data</i> del módulo MSSP en el modo SPI
		SDA	E/S <i>Data</i> del módulo MSSP en el modo I ² C
RC5/SDO	24	RC5	E/S de propósito general en el puerto PORTC
		SDO	Salida <i>Data</i> del módulo MSSP en el modo SPI
RC6/TX/CK	25	RC6	E/S de propósito general en el puerto PORTC
		TX	Salida asíncrona del módulo USART
		CK	Reloj síncrono del módulo USART
RC7/RX/DT	26	RC7	E/S de propósito general en el puerto PORTC
		RX	Entrada asíncrona del módulo USART
		DT	Datos del módulo USART en modo síncrono

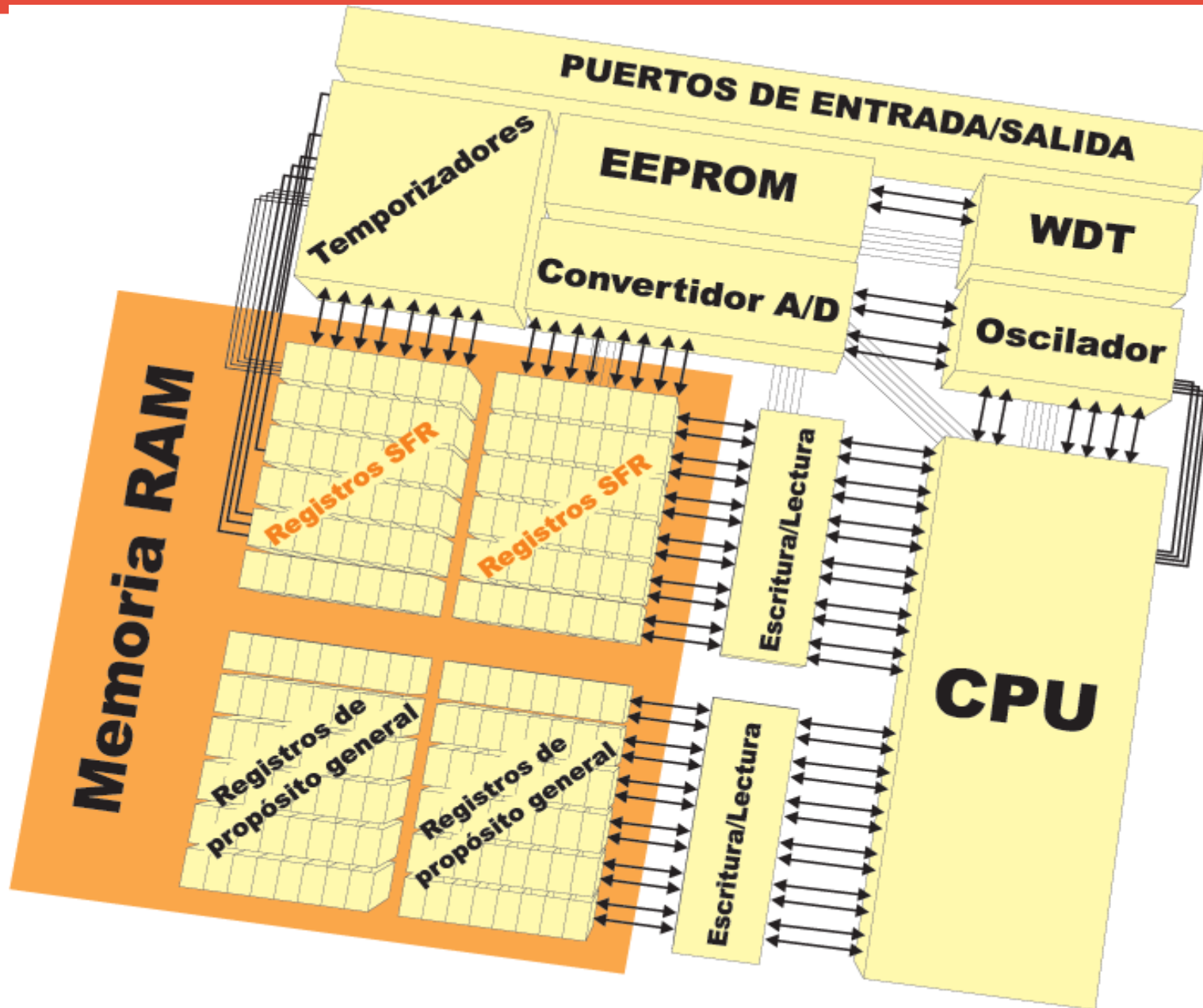
PIC16F887

Nombre	Número (DIP 40)	Función	Descripción
RD4	27	RD4	E/S de propósito general en el puerto PORTD
RD5/P1B	28	RD5	E/S de propósito general en el puerto PORTD
		P1B	Salida del módulo PWM
RD6/P1C	29	RD6	E/S de propósito general en el puerto PORTD
		P1C	Salida del módulo PWM
RD7/P1D	30	RD7	E/S de propósito general en el puerto PORTD
		P1D	Salida del módulo PWM
Vss	31	-	Tierra (GND)
Vdd	32	+	Suministro de voltaje positivo
RB0/AN12/INT	33	RB0	E/S de propósito general en el puerto PORTB
		AN12	Canal 12 del convertidor A/D
		INT	Interrupción externa
RB1/AN10/C12INT3-	34	RB1	E/S de propósito general en el puerto PORTB
		AN10	Canal 10 del convertidor A/D
		C12INT3-	Entrada negativa de los comparadores C1 o C2

PIC16F887

RB2/AN8	35	RB2	E/S de propósito general en el puerto PORTB
		AN8	Canal 8 del convertidor A/D
RB3/AN9/PGM/C12IN2-	36	RB3	E/S de propósito general en el puerto PORTB
		AN9	Canal 9 del convertidor A/D
		PGM	Habilita la programación del chip
		C12IN2-	Entrada negativa de los comparadores C1 o C2
RB4/AN11	37	RB4	E/S de propósito general en el puerto PORTB
		AN11	Canal 11 del convertidor A/D
RB5/AN13/T1G	38	RB5	E/S de propósito general en el puerto PORTB
		AN13	Canal 13 del convertidor A/D
		T1G	Entrada externa del temporizador 11
RB6/ICSPCLK	39	RB6	E/S de propósito general en el puerto PORTB
		ICSPCLK	Entrada de reloj de programación serial
RB7/ICSPDAT	40	RB7	E/S de propósito general en el puerto PORTB
		ICSPDAT	Pin de E/S para introducir los datos durante la programación ICSP™

PIC16F887



PIC16F887-CPU

- La CPU cuenta con sólo 35 instrucciones simples. Cabe decir que para poder programar otros microcontroladores en lenguaje ensamblador es necesario saber más de 200 instrucciones.
- El tiempo de ejecución es igual para casi todas las instrucciones y tarda 4 ciclos de reloj. La frecuencia del oscilador se estabiliza por un cristal de cuarzo. Las instrucciones de salto y de ramificación tardan ocho ciclos de reloj en ejecutarse. Esto significa que si la velocidad de operación del microcontrolador es 20 MHz, el tiempo de ejecución de cada instrucción será 200nS.

PIC16F887-RAM

Es la tercera y la más compleja parte de la memoria del microcontrolador. En este caso consiste en dos partes: en registros de propósito general y en los registros de funciones especiales (los SFR). Todos estos registros se dividen en cuatro bancos de memoria.

PIC16F887-REGISTROS DE PROPÓSITO GENERAL

Los registros de propósito general se utilizan para almacenar los datos temporales y los resultados creados durante el funcionamiento, es necesario especificar la dirección de un registro de propósito general y asignarle una función.

Pag 23-30

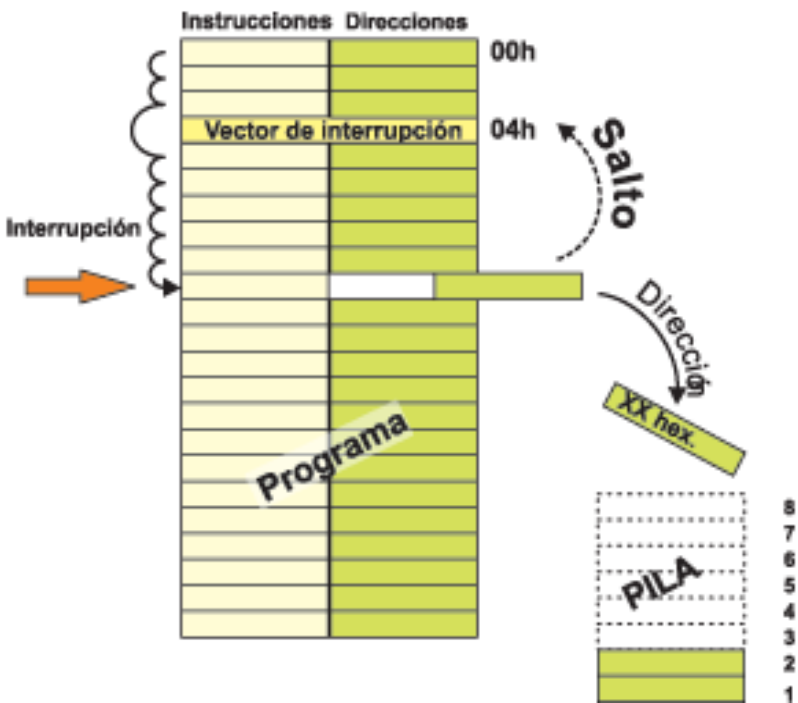
PIC16F887-REGISTROS DE FUNCIONES ESPECIALES (SFR)

Los registros de funciones especiales son también parte de la memoria RAM. A diferencia de los registros de propósito general, su propósito es predeterminado durante el proceso de fabricación y no se pueden cambiar. Como los bits están conectados a los circuitos particulares en el chip (convertidor A/D, módulo de comunicación serial, etc), cualquier cambio de su contenido afecta directamente al funcionamiento del microcontrolador o de alguno de sus módulos.

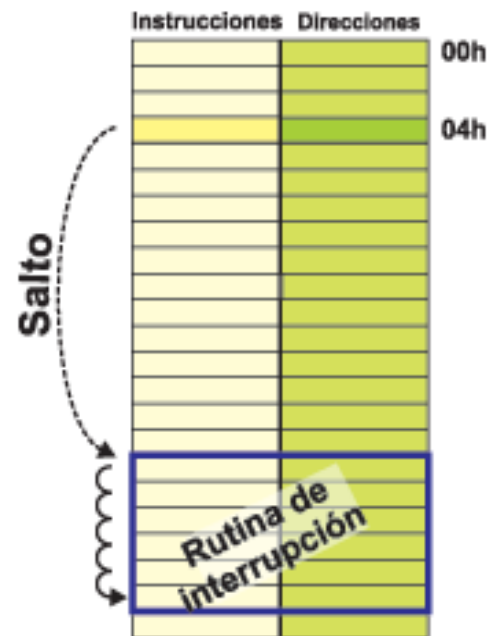
Otra característica de estas localidades de memoria es que tienen nombres (tanto los registros como sus bits), lo que simplifica considerablemente el proceso de escribir un programa.

PIC16F887-INTERRUPCIONES

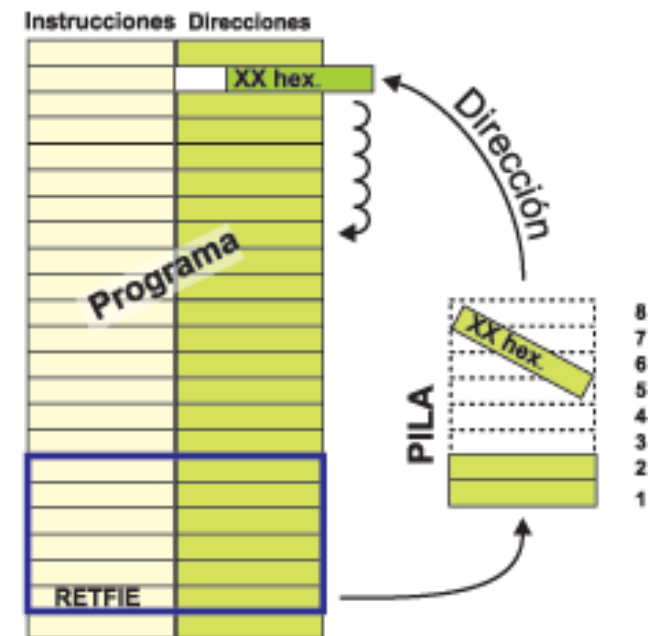
1



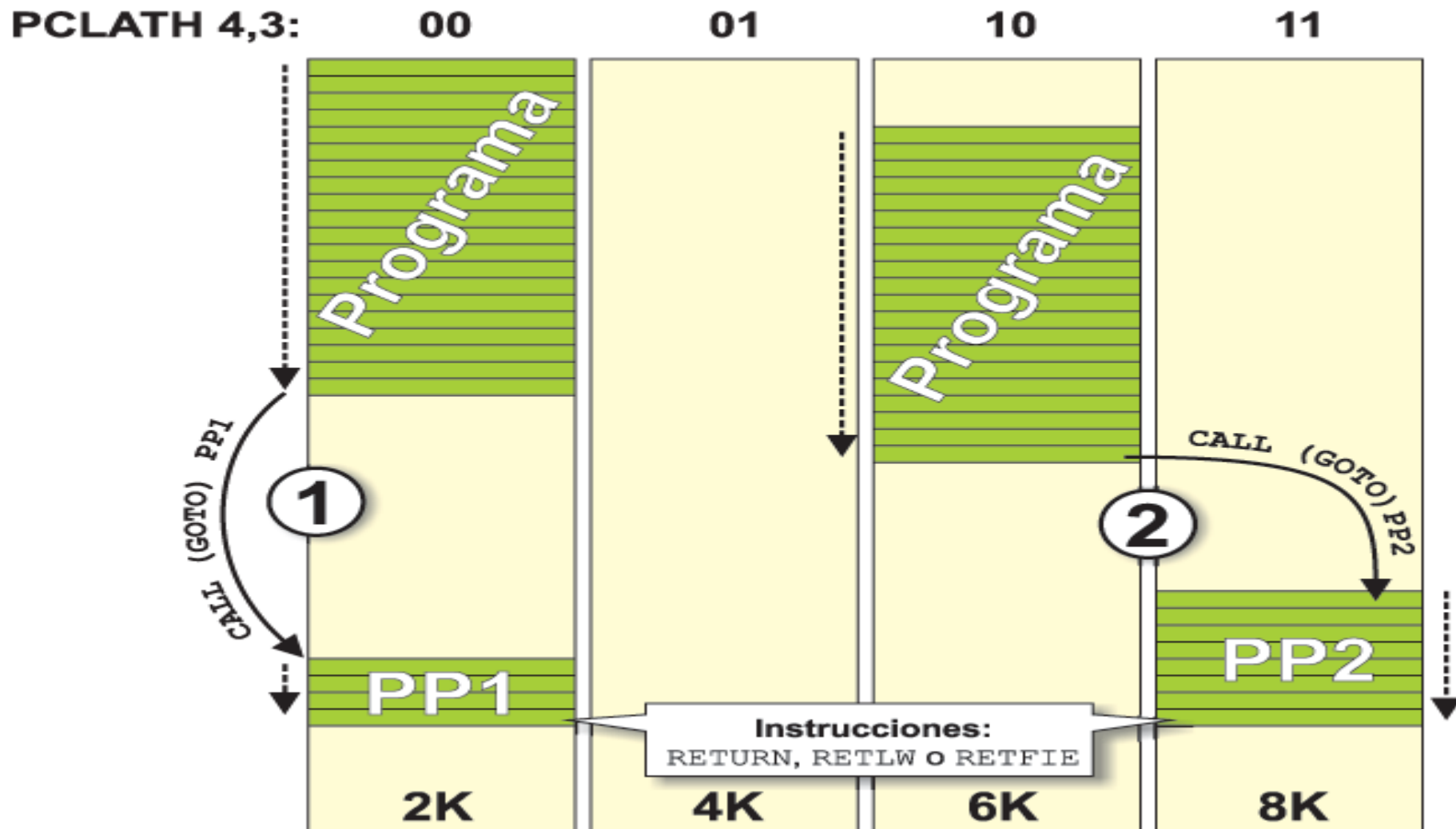
2



3

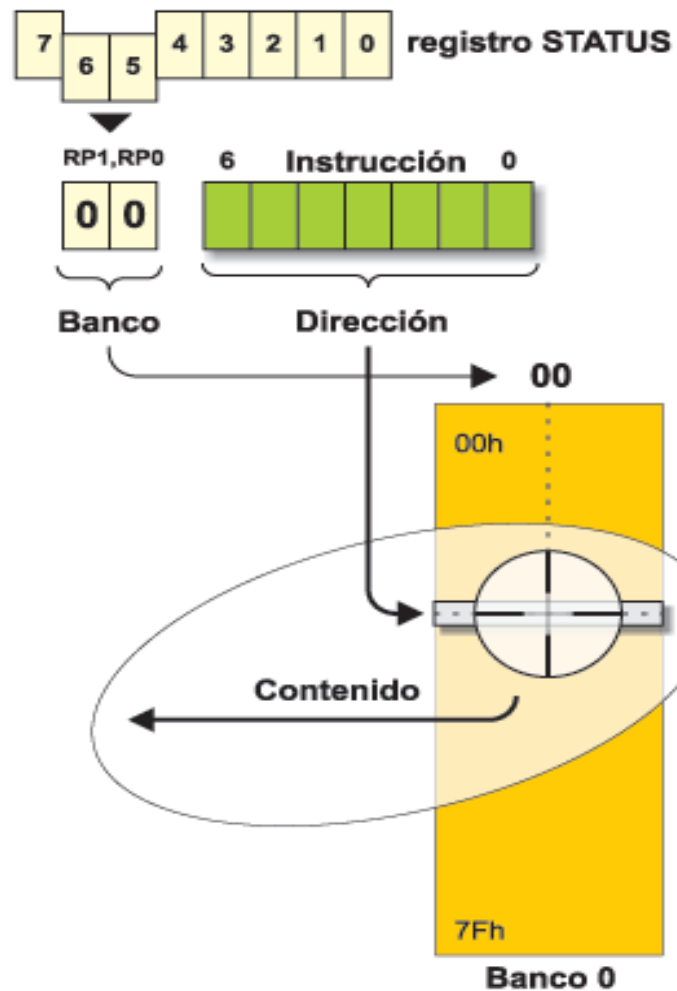


PIC16F887

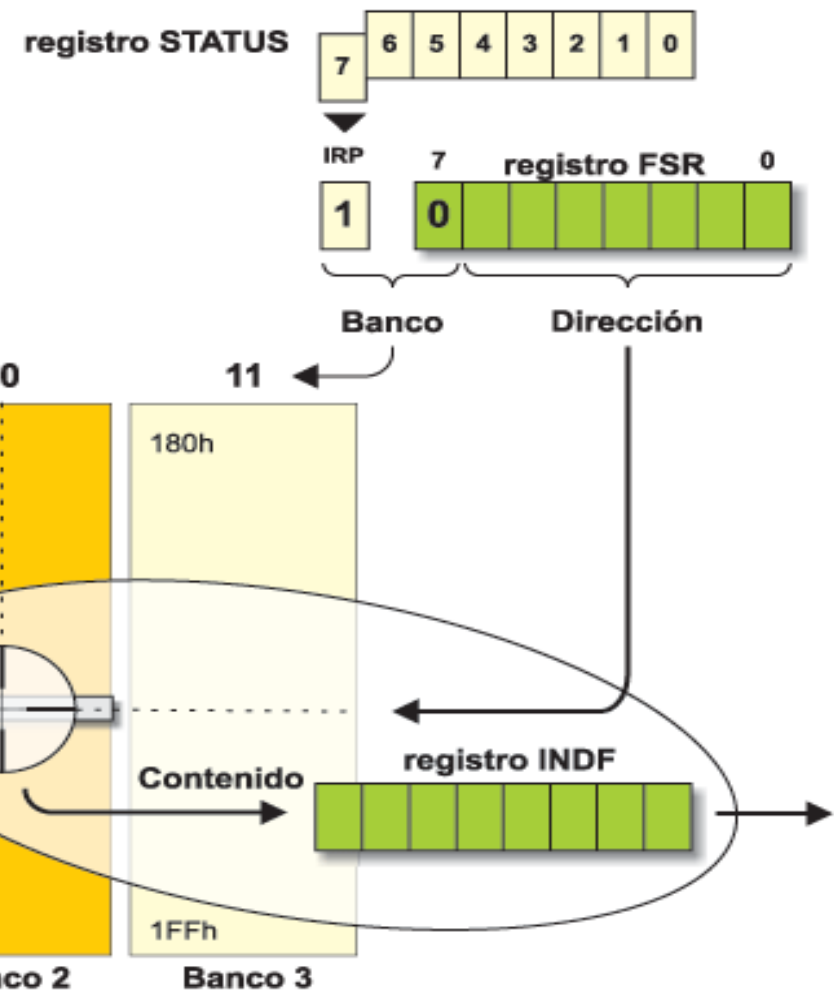


PIC16F887

Direccionamiento directo



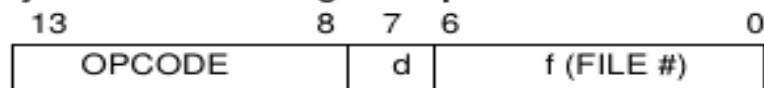
Direccionamiento indirecto



PIC16F887-INSTRUCCIONES

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
\overline{TO}	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
\overline{PD}	Power-down bit

Byte-oriented file register operations

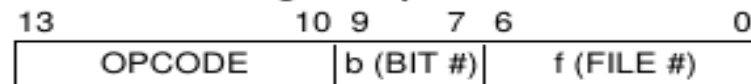


d = 0 for destination W

d = 1 for destination f

f = 7-bit file register address

Bit-oriented file register operations

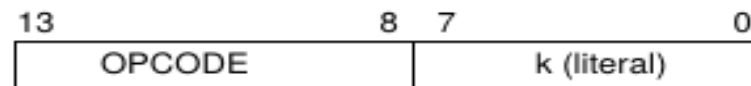


b = 3-bit bit address

f = 7-bit file register address

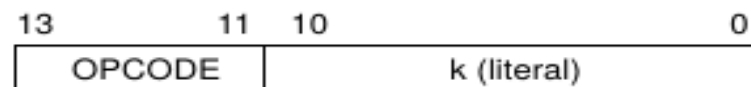
Literal and control operations

General



k = 8-bit immediate value

CALL and GOTO instructions only



k = 11-bit immediate value

PIC16F887-INSTRUCCIONES

INSTRUCCIÓN	DESCRIPCIÓN	OPERACIÓN	BANDERA	CLK	*
Instrucciones para la transmisión de datos					
MOVLW k	Mover literal a W	k -> w		1	
MOVWF f	Mover el contenido de W a f	W -> f		1	
MOVF f,d	Mover el contenido de f a d	f -> d	Z	1	1, 2
CLRW	Borrar el contenido de W	0 -> W	Z	1	
CLRF f	Borrar el contenido de f	0 -> f	Z	1	2
SWAPF f,d	Intercambiar de nibbles en f	f(7:4),(3:0) -> f(3:0),(7:4)		1	1, 2

PIC16F887-INSTRUCCIONES

Instrucciones aritmético – lógicas					
ADDLW k	Sumar literal a W	$W+k \rightarrow W$	C, DC, Z	1	
ADDWF f,d	Sumar el contenido de W y f	$W+f \rightarrow d$	C, DC, Z	1	1, 2
SUBLW k	Restar W de literal	$k-W \rightarrow W$	C, DC, Z	1	
SUBWF f,d	Restar W de f	$f-W \rightarrow d$	C, DC, Z	1	1, 2
ANDLW k	AND W con literal	$W \text{ AND } k \rightarrow W$	Z	1	
ANDWF f,d	AND W con f	$W \text{ AND } f \rightarrow d$	Z	1	1, 2
IORLW k	OR inclusivo de W con literal	$W \text{ OR } k \rightarrow W$	Z	1	
IORWF f,d	OR inclusivo de W con f	$W \text{ OR } f \rightarrow d$	Z	1	1, 2
XORWF f,d	OR exclusivo de W con literal	$W \text{ XOR } k \rightarrow W$	Z	1	1, 2

PIC16F887-INSTRUCCIONES

XORLW k	OR exclusivo de W con f	W XOR f -> d	Z	1	
INCF f,d	Sumar 1 a f	f+1 -> f	Z	1	1, 2
DECF f,d	Restar 1 a f	f-1 -> f	Z	1	1, 2
RLF f,d	Rotar F a la izquierda a través del bit de Acarreo		C	1	1, 2
RRF f,d	Rotar F a la derecha a través del bit de Acarreo		C	1	1, 2
COMF f,d	Complementar f	f -> d	Z	1	1, 2

PIC16F887-INSTRUCCIONES

Instrucciones orientadas a bit					
BCF f,b	Poner a 0 el bit b del registro f	0 -> f(b)	1	1,	2
BSF f,b	Poner a 1 el bit b del registro f	1 -> f(b)	1	1,	2

PIC16F887-INSTRUCCIONES

Instrucciones de control de programa				
BTFSC f,b	Saltar si bit b de registro f es 0	Skip if f(b) = 0	1 (2)	3
BTFSS f,b	Saltar si bit b de reg. f es 1	Skip if f(b) = 1	1 (2)	3
DECFSZ f,d	Disminuir f en 1. Saltar si el resultado es 0	f-1 -> d skip if Z = 1	1 (2)	1, 2, 3
INCFSZ f,d	Incrementar f en 1. Saltar si el resultado es 1	f+1 -> d skip if Z = 0	1 (2)	1, 2, 3
GOTO k	Saltar a una dirección	k -> PC	2	
CALL k	Llamar a una subrutina	PC -> TOS, k -> PC	2	
RETURN	Retornar de una subrutina	TOS -> PC	2	
RETLW k	Retornar con literal en W	k -> W, TOS -> PC	2	
RETFIE	Retornar de una interrupción	TOS -> PC, 1 -> GIE	2	

PIC16F887-INSTRUCCIONES

Otras instrucciones				
NOP	No operación	TOS -> PC, 1 -> GIE		1
CLRWDT	Reiniciar el temporizador perro guardián	0 -> WDT, 1 -> TO, 1 -> PD	TO, PD	1
SLEEP	Poner en estado de reposo	0 -> WDT, 1 -> TO, 0 -> PD	TO, PD	1