

# MICROCONTROLADORES

*ASSEMBLER (ASM) Y PIC16F887*

*Ing Marlon Moreno*

# PIC16F887

*Arquitectura RISC:* El microcontrolador cuenta con solo 35 instrucciones diferentes, todas las instrucciones son un 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: 2.0V a 5.5V*

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



# PIC16F887

- *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

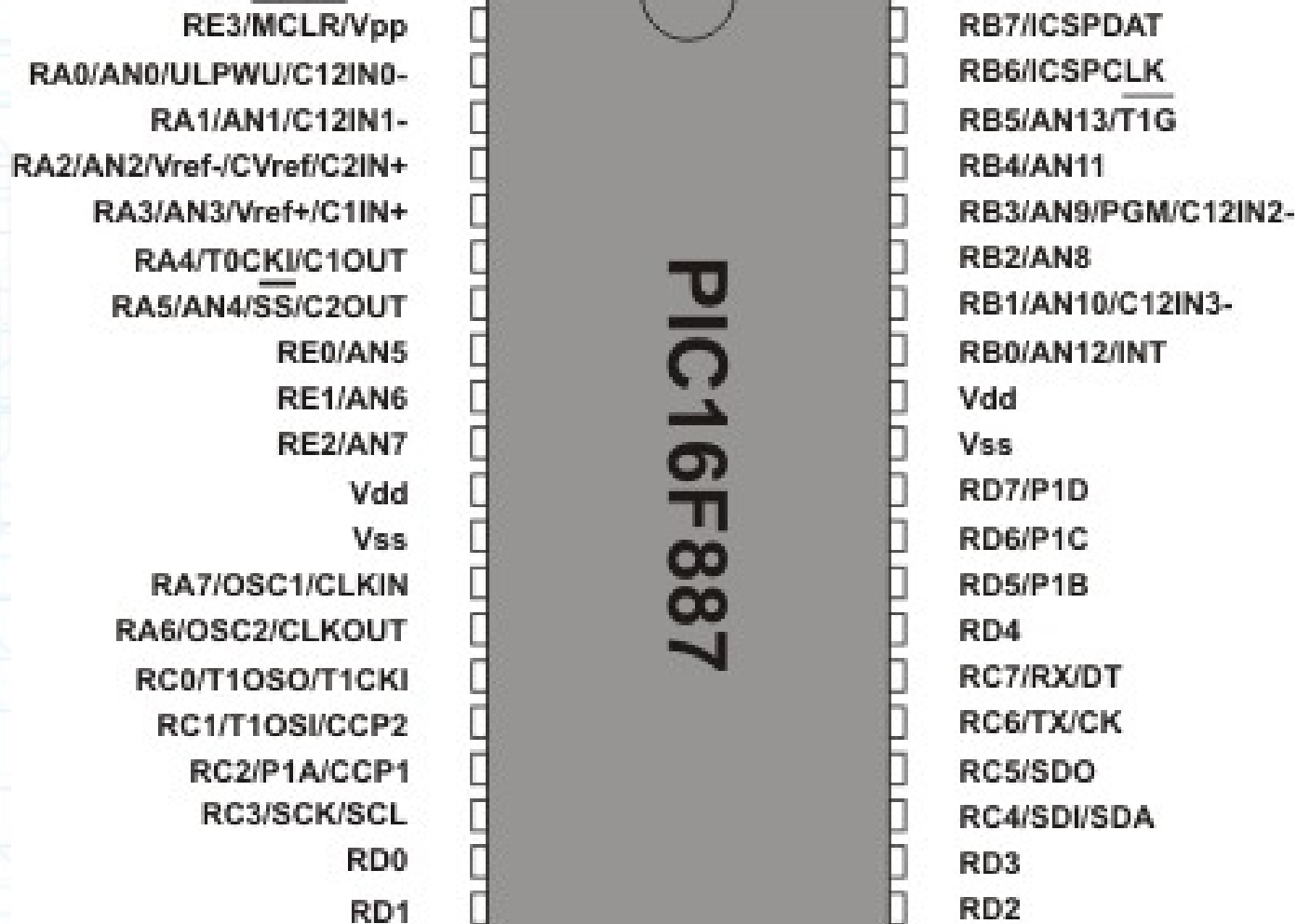
- *Opción de programación in-circuit.*
- *256 bytes de memoria EEPROM*
  - *Los datos se pueden grabar más de 1.000.000 veces*
- *368 bytes de memoria RAM*
- *Convertidor A/D:*
  - *14 canales*
  - *Resolución de 10 bits*



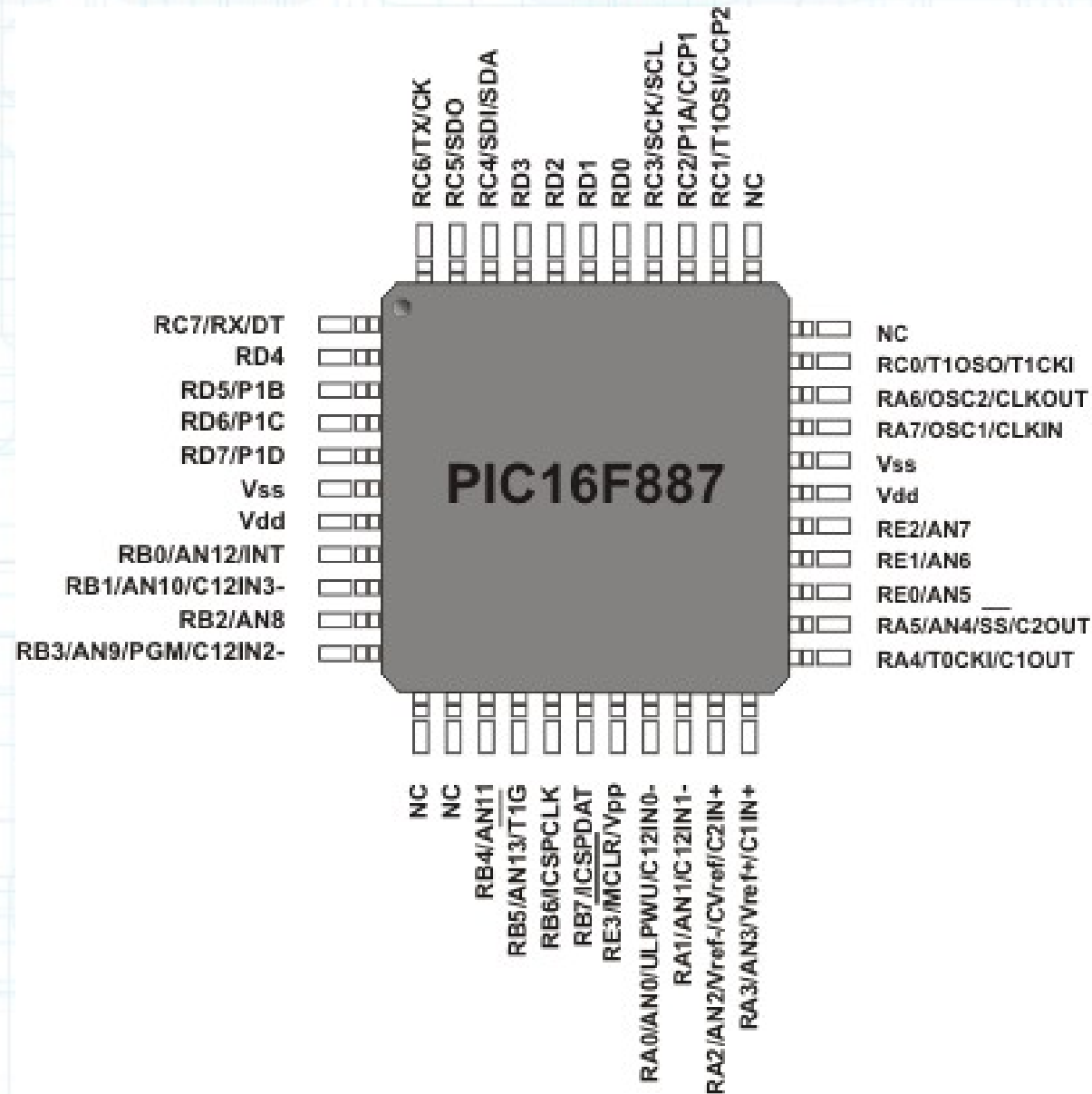
# PIC16F887

- *3 temporizadores/contadores independientes*
  - *Temporizador (Watchdog) perro guardián.*
- *Módulo comparador analógico:*
- *Dos comparadores analógicos*
- *Módulo PWM incorporado*
- *Módulo USART mejorado*
  - *Soporta las comunicaciones seriales RS-485, RS-232 y LIN2.0*
- *Puerto Serie Síncrono Maestro (MSSP)*
  - *Soporta los modos SPI e I2C*

# PIC16F887 -PINES

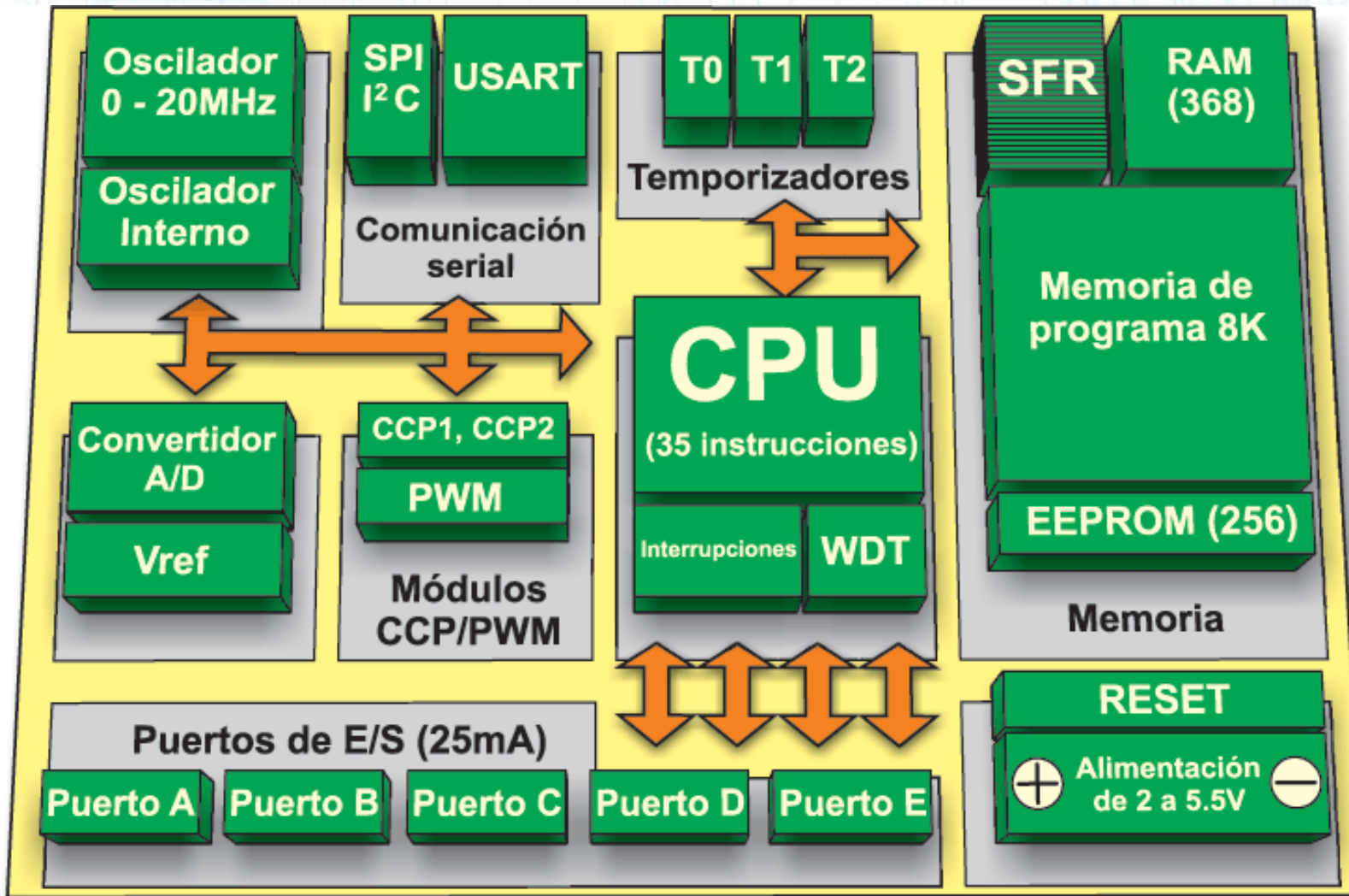


# PIC16F887 - PINES





# PIC16F887 - ORGANIZACIÓN





# PIC16F887 - CPU

- *El tiempo de ejecución de las instrucciones tarda 4 ciclos de reloj (periodos de la señal de reloj).*
- *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 más compleja memoria del microcontrolador. Esta constituida de dos partes: Registros de propósito general (GPRs) y en los registros de funciones especiales (los SFR). Todos los registros estan distibuidos en cuatro bancos de memoria.

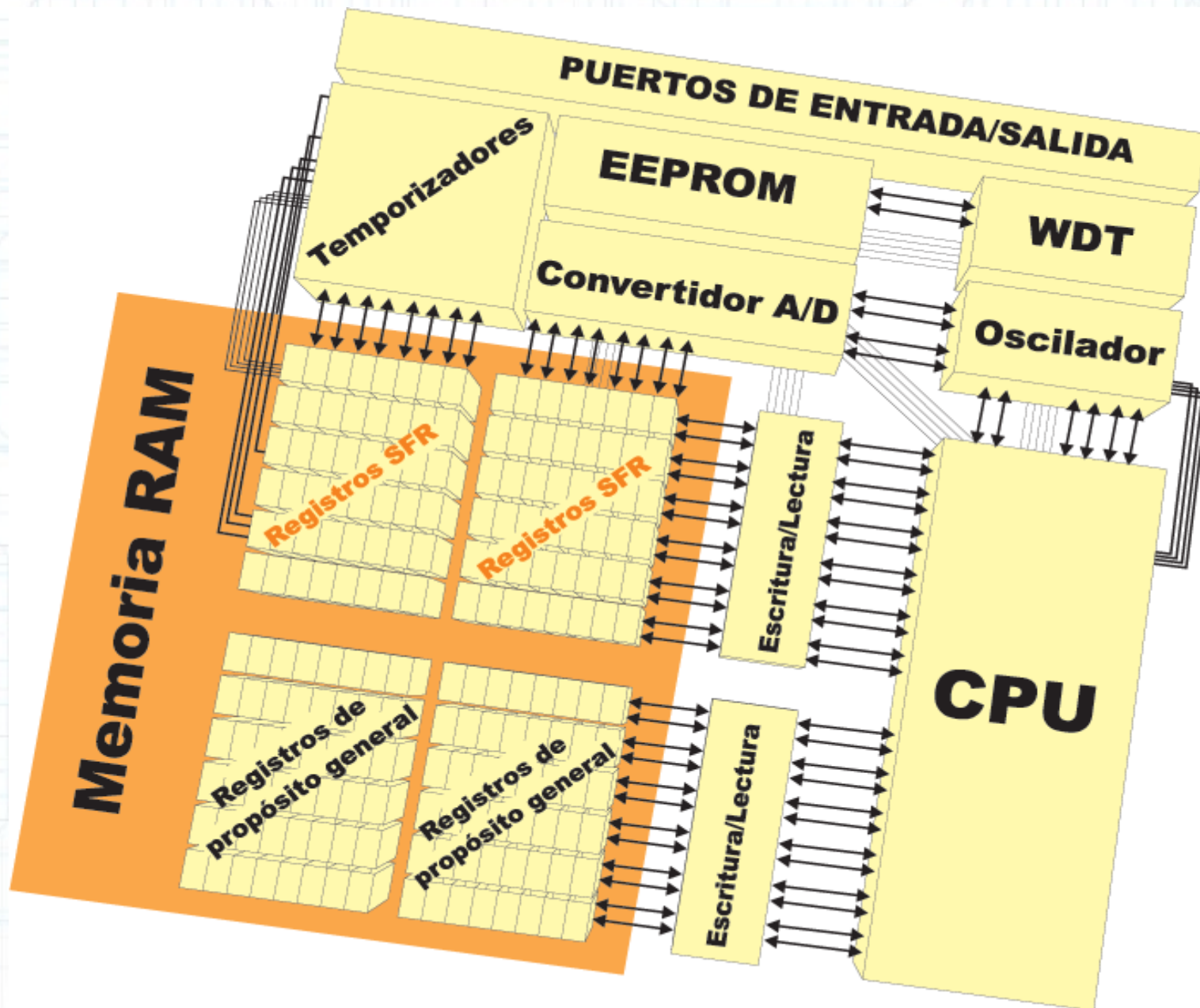


# PIC16F887 - RAM

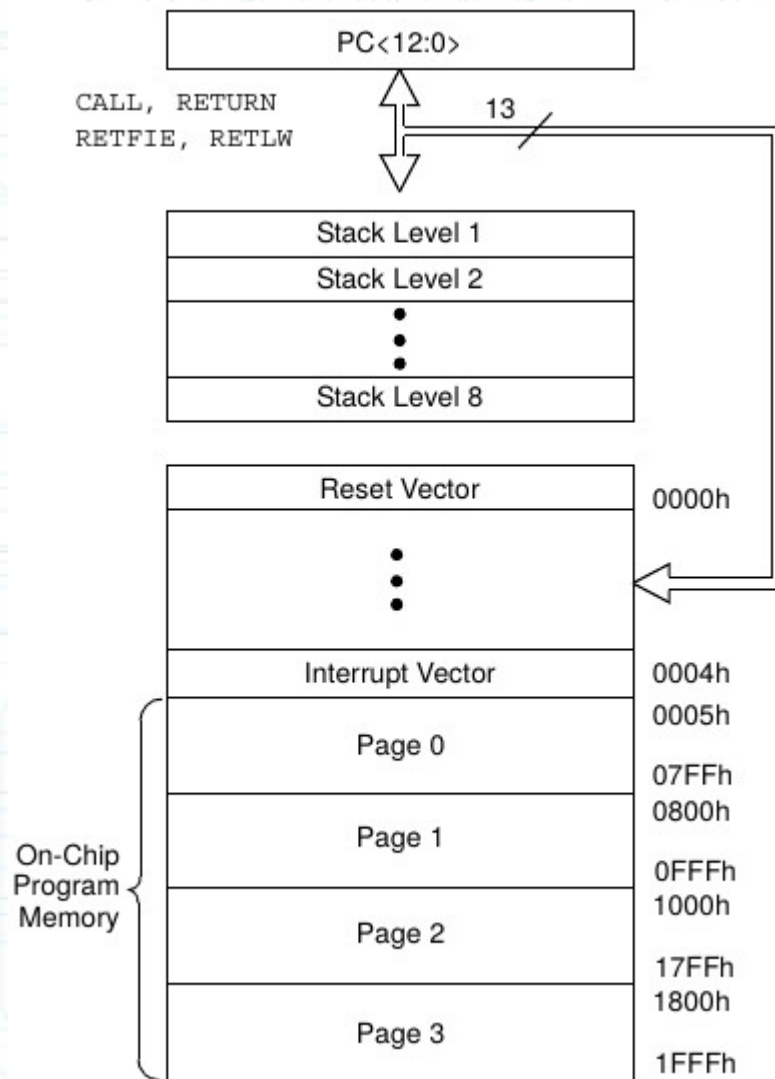
- **GPRs:** *Se utilizan para almacenar los datos temporales y los resultados creados durante el funcionamiento, es necesario especificar la dirección de un registro de propósito general y asignarle una función.*
- **SFR:** *su propósito es predeterminado durante el proceso de fabricación y no se pueden cambiar. La modificación de estos cambia el funcionamiento del microcontrolador o del modulo al cual se encuentre asociados. Pag 25-30.*



# PIC16F887 - ORGANIZACIÓN



# MEMORIA DE PROGRAMA Y PILA



- Contador de programa de 13 bits.
- 8 Niveles de pila.
- Inicio de programa de usuario desde la posición 0x00
- Vector de interrupciones desde la posición 0x04



# ASSEMBLER

*Lenguaje de bajo nivel, constituido por una lista de mnemónicos\* que representan las instrucciones básicas del microprocesador, microcontrolador o otro circuito integrado programable.*

*Representación simbólica de los códigos máquina binarios y variables necesarias para la programación de una arquitectura específica de CPU. Es específico para cada dispositivo.*

*\*Palabra que sustituye un código de operación.*



# GRUPO DE INSTRUCCIONES

- **OPERACIONES CON ENTEROS:** Trabajo con datos de 8,16,32,64 bits de acuerdo con la arquitectura de la CPU.
  - Aritmeticas: suma, resta....
  - Booleanas: And, Or....
  - Bits: Rotar
  - Comparaciones....
- **OPERACIONES CON NUMEROS REALES.** Requieren coprocesador.

# GRUPO DE INSTRUCCIONES

- **OPERACIONES DE MOVIMIENTOS DE DATOS:**
  - *Movimiento entre registros.*
  - *Movimiento de pila: PUSH, POP*
  - *Movimiento con puertos.*
- **OPERACIONES DE CONTROL DE FLUJO:**
  - *Salto condicionales e incondicionales.*
  - *Llamado y retorno de subrutinas.*
  - *Llamado y retorno de interrupciones.*



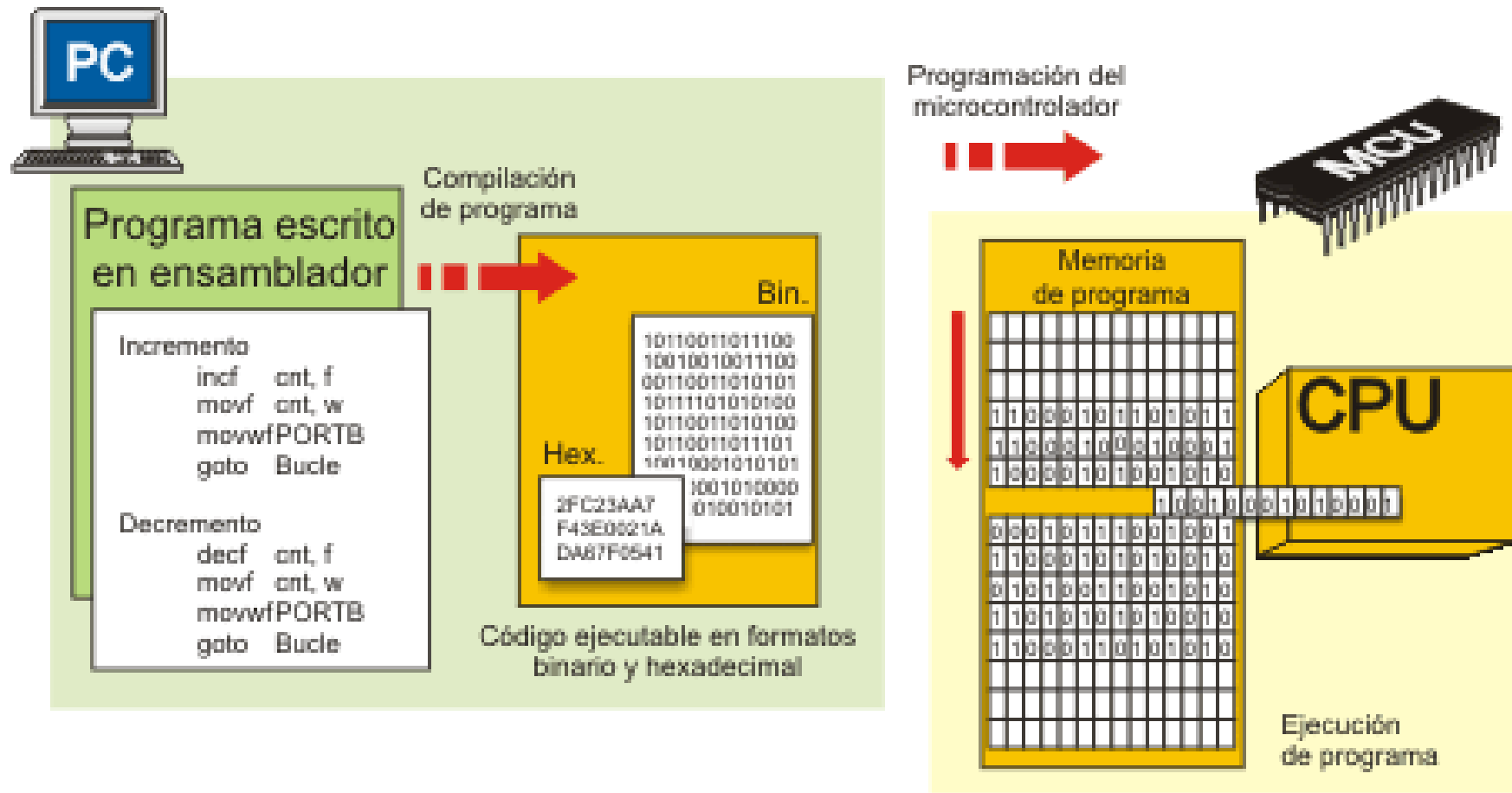
# ENSAMBLADO

- *Proceso de transformación de los mnemónicos usados en el lenguaje ensamblador a código máquina (binario).*

*La transformación suele ser uno a uno. Para cada línea en ensamblador existe una única representación en código máquina a excepciones cuando se usan pseudo instrucciones.*



# ENSAMBLADO



# PROGRAMACIÓN EN ASM

- *Estudiar la organización y arquitectura del microcontrolador. Pag 16-17.*
- *Comprender la distribución de los registros en los bancos de trabajo. Pag 25-30*
- *Identificar y determinar las funciones de los SFR.*
- *Estudiar el set de instrucciones y reconocer la sintaxis propia de la CPU.*

# SFR STATUS

STATUS	R/W (0)	R/W (0)	R/W (0)	R (1)	R (1)	R/W (x)	R/W (x)	R/W (x)	Características
	IRP	RP1	RP0	TO	PD	Z	DC	C	Nombre de bit
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

- *IRP*: Selección de banco.
- *TO*: Bit de tiempo fuera del perro guardian.
- *PD*: Bit de apagado.
- *Z* : Bit de cero.
- *DC*: Bit de acarreo o prestamo. 4<sup>th</sup> low-order.
- *C*: Carry, cuando es mayor a 255 o menor a 0.



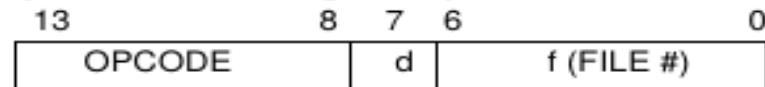
# SFR WORKING

- *Registro de trabajo usado para almacenar temporalmente los resultados de ALU (unidad logico aritmetica) antes de ser movidos a los registros de uso general o otros SFR. Tambien se usa para almacenar uno de los operandos de las operaciones.*

# SINTAXIS

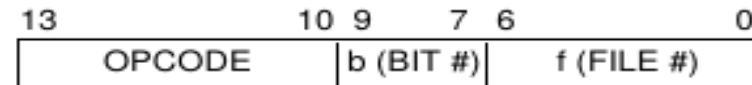
Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility 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

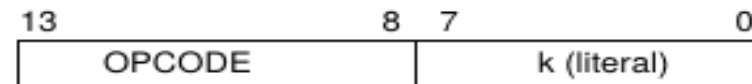
## Bit-oriented file register operations



b = 3-bit bit address  
f = 7-bit file register address

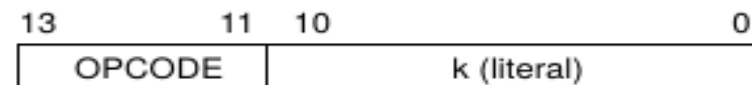
## Literal and control operations

### General



k = 8-bit immediate value

### CALL and GOTO instructions only



k = 11-bit immediate value

# INSTRUCCIONES

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



# INSTRUCCIONES

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

# INSTRUCCIONES

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

# INSTRUCCIONES

## Instrucciones orientadas a bit

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

## Otras instrucciones

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



# INSTRUCCIONES

## Instrucciones de control de programa

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

# ASIGNACIÓN DE DIRECCIONES DE MEMORIA A SFR

STATUS
FSR
PORTA
PORTB
PORTC
PORTD <sup>(2)</sup>
PORTE
PCLATH
INTCON

- 03h • STATUS EQU H'0003'
- 04h • FSR EQU H'0004'
- 05h • PORTA EQU H'0005'
- 06h • PORTB EQU H'0006'
- 07h • PORTC EQU H'0007'
- 08h • PORTD EQU H'0008'
- 09h • PORTE EQU H'0009'
- 0Ah • PCLATCH EQU H'000A'
- 0Bh • INTCON EQU H'000B'