

Unidad 4 Caso de Estudio

Micro-Controlador PIC

Técnicas Digitales II UTN-FRM

Ver. 1.1- JUN-02

técnicas digitales 2

Introducción

técnicas digitales 2

- Se presenta la familia de micro-controladores PIC.
 - Familia PIC
 - Características Generales
 - Organización de Memoria
 - Direccionamiento, Registros y Formato de Instrucción
 - Conjunto de Instrucciones
 - Ejemplo Nro 1: Circuito de Led
 - Ejemplo Nro 2: Manejo de Timers e Interrupciones
 - Timers, Compara y Captura
 - Comunicaciones Serie - UART
 - Conversión A/D
- Bibliografía
 - "Design with PIC Microcontrollers" John B. Peatman - Prentice Hall
 - "PICmicro Mid-Range MCU Family Reference Manual" MicroChip, 1997
- Autor:
 - Ing. Gustavo Mercado (gmercado@frm.utn.edu.ar)

Junio 02

UNIDAD 4 TD II FRM UTN

2

La familia PIC

ROM EPROM EEPROM

Performance ↑	PIC18CXX		PIC18C242		Next Gen 16 - Bit Instruction
	PIC17CXX		PIC17C4X		High-End 16 - Bit Instruction
	PIC16CXX	PIC16CR6X PIC16CR8X	PIC16C6X PIC16C7X PIC 16C6XX	PIC16C8X	Mid Range 14 - Bit Instruction
	PIC16C5X	PIC16CR5X PIC16CR5X A	PIC16C5X PIC16C5XA	PICSEE	BaseLine 12 - Bit Instruction
		Onboard Memory Tech →			

Junio 02
UNIDAD 4 TD II FRM UTN
3

La familia PIC con t

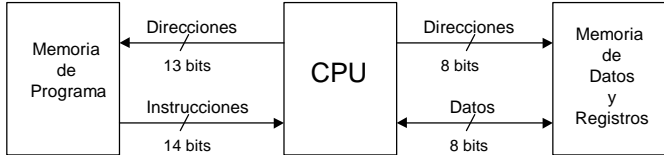
técnicas digitales 2

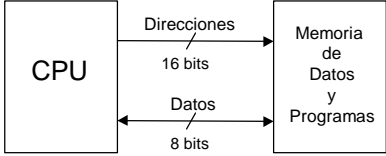
PIC 16C5X BASE LINE 12 BIT											
Frec. Oper.	Eprom	RAM	I/O Pin	Instruc	INT	Peripheral					
						Timer	Com	RV	PWM	Serial	A/D
20 Mhz	512 4K	24 80	12 20	33	X	1	2	Si	x	x	x
PIC 16C6X MID RANGE 14 BIT											
20 Mhz	2k 16K	128 368	22 33	35	1 12	3	2	X	2	2	8 to12 Bit
PIC 17CXX HIGH PERFORMANCE 16 BIT											
25Mhz	2k 128k	232 902	33	55 58	18	4	2	X	2	Si	10 Bit
PIC 18CXX 16 BIT											
40Mhz	2M	4K	34	75	14	4	x	Si	2	I2C SPI Usart	10 Bit

Junio 02
UNIDAD 4 TD II FRM UTN
4

PIC Arquitectura

técnicas digitales 2

- Arquitectura Harvard**


Arquitectura Harvard
- Arquitectura Von Neumann**


Arquitectura Von Neumann

Junio 02 UNIDAD 4 TD II FRM UTN 5

PIC Arquitectura con t

técnicas digitales 2

- Palabra de Instrucción larga (LIW)**
 - Instrucciones de 12, 14 y 16 bits
 - Comparar con instrucciones de 8 bits del Z80
- Palabra de Instrucción Sencilla**
 - Una instrucción por palabra de Memoria
 - Búsqueda de instrucción más rápida
 - Comparar con los hasta 4 bytes por instrucción del Z80
- Un ciclo de ejecución por instrucción**
 - La instrucción es buscada en un solo ciclo de máquina
 - Los saltos tiene un ciclo adicional

Junio 02 UNIDAD 4 TD II FRM UTN 6

PIC Arquitectura con ́t

técnicas digitales 2

- **Tubería de instrucciones**
 - Tubería de dos etapas
 - Ejecución de instrucciones en un solo ciclo de máquina

	Tcy0	Tcy1	Tcy2	Tcy3	Tcy4	Tcy5
1. MOVLW 55h	Fetch 1	Execute 1				
2. MOVWF PORTB		Fetch 2	Execute 2			
3. CALL SUB_1			Fetch 3	Execute 3		
4. .BSF PORTA, BIT3 (Forced NOP)				Fetch 4	Flush	
5. Intruccion @ address SUB_1					Fetch SUB_1	Execute Sub_1
						Fetch SUB_1+1

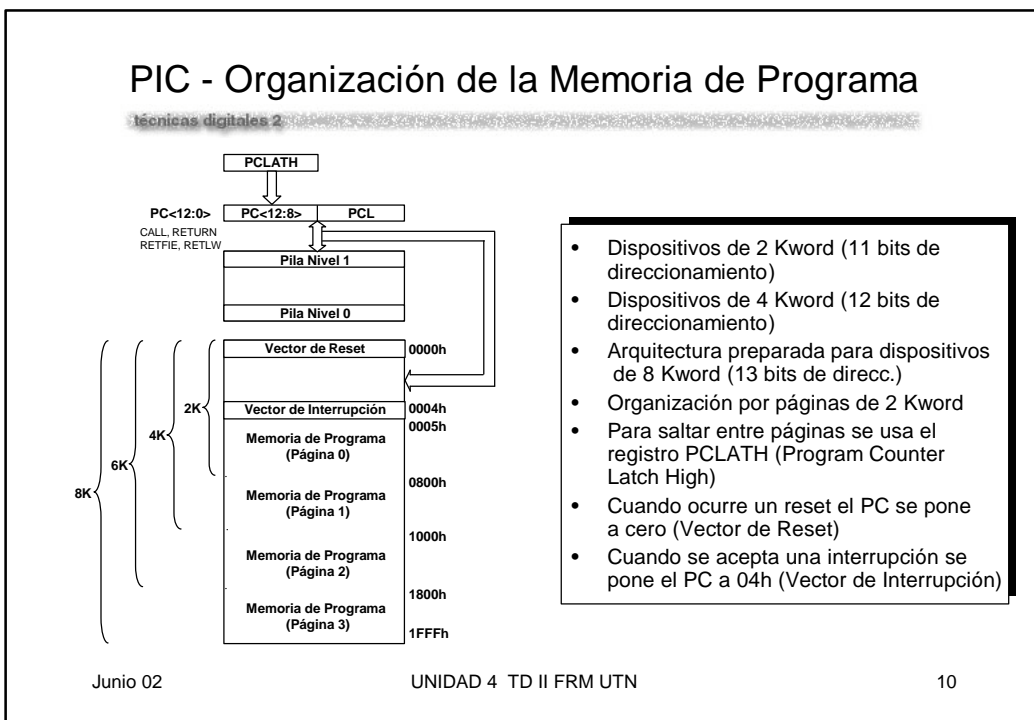
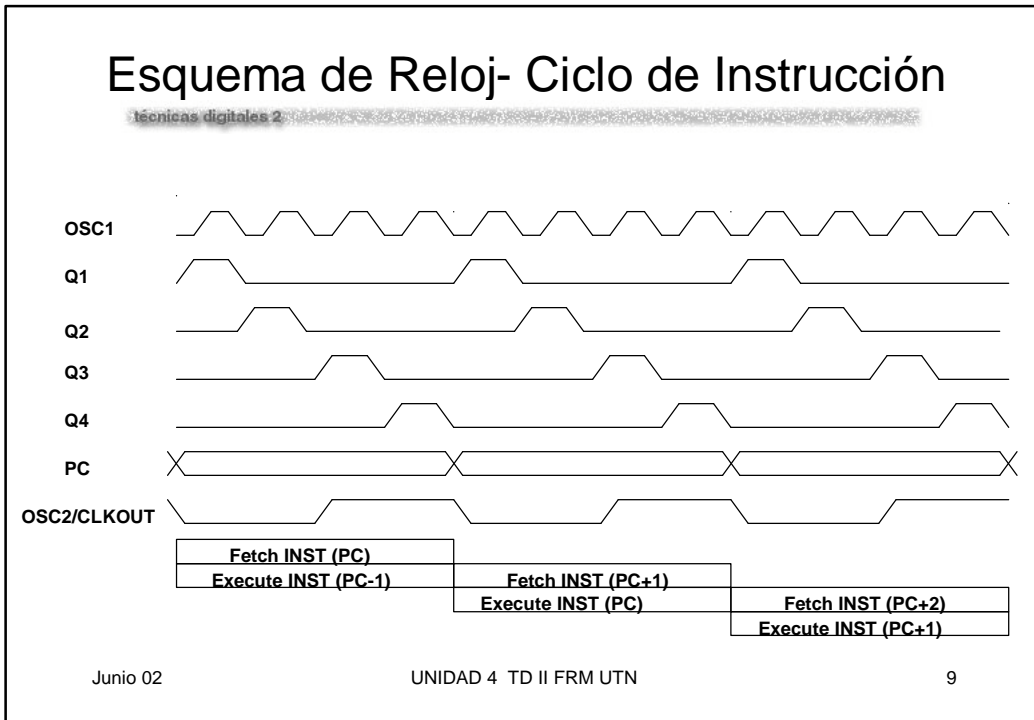
Junio 02
UNIDAD 4 TD II FRM UTN
7

PIC Arquitectura con ́t

técnicas digitales 2

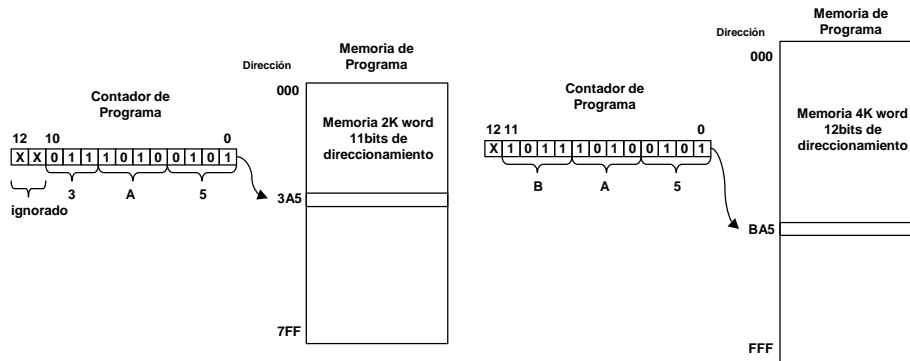
- **Conjunto de instrucciones reducido (RISC)**
 - Reduced Instruction Set Computer
 - Solo 33 a 55 instrucciones según modelo
 - Comparar con 200 del Motorola 6805, más de 100 del Z80
 - Unidad de control más rápida
 - No es RISC completo (concepto Load-Store)
- **Arquitectura con conjunto de registros**
 - Register File
 - Direccionamiento Directo e Indirecto
 - Registros direccionados en memoria
- **Instrucciones ortogonales**
 - Operaciones simétricas
 - Palabra de instrucción constante
 - Todas la instrucciones con cualquier Registro con cualquier modo de direccionamiento

Junio 02
UNIDAD 4 TD II FRM UTN
8



PIC - Organización de la Memoria de Programa

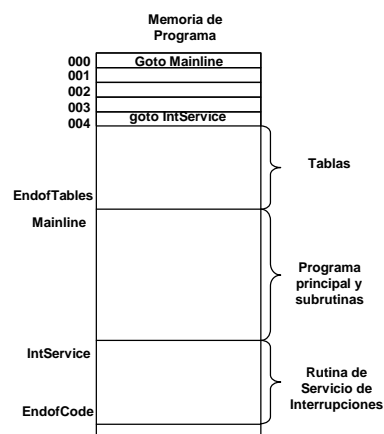
técnicas digitales 2



- El Contador de Programa apunta a la próxima instrucción a ser ejecutada
- La parte baja del PC se denomina PCL. Se puede leer y escribir
- La parte alta se denomina PCH y solo se modifica por medio del PCLATH

PIC Organización de la Memoria de Programa cont´

técnicas digitales 2



```

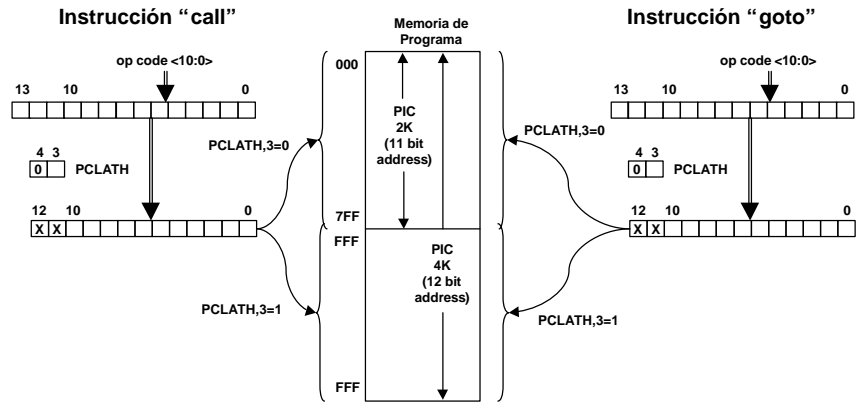
Mainline
call Initial ; Inicialización

MainLoop
call Task1
call Task2
.
.
.
call LoopTime ; ajuste tiempo
goto MainLoop ; repetición
    
```

- EndofTables ? D´255 = H´0FF
- EndofCode ? D´2047 = H´7FF

PIC Organización de la Memoria de Programa cont´

técnicas digitales 2



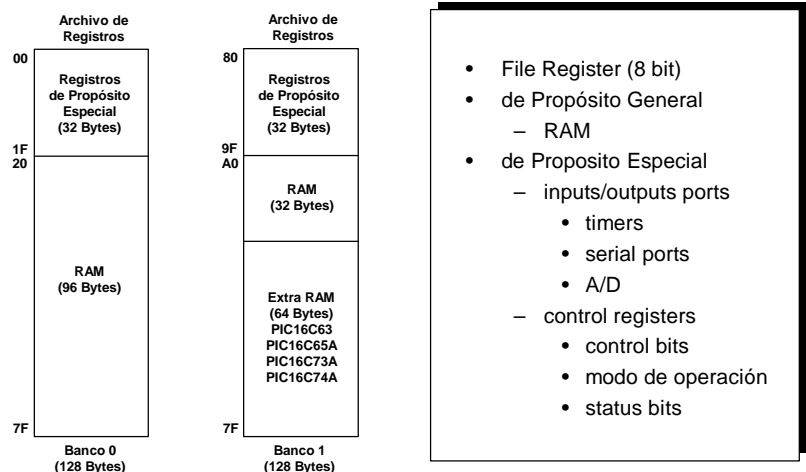
Junio 02

UNIDAD 4 TD II FRM UTN

13

Pic Register File

técnicas digitales 2



- File Register (8 bit)
- de Propósito General
 - RAM
- de Propósito Especial
 - inputs/outputs ports
 - timers
 - serial ports
 - A/D
 - control registers
 - control bits
 - modo de operación
 - status bits

Junio 02

UNIDAD 4 TD II FRM UTN

14

PIC Direccionamiento directo

técnicas digitales 2

- Memoria de datos organizado en bancos.
- Existen 2 (4) bancos de 128 bytes.
- Para cambiar de banco se usa RP0 (RP1) del Status Register

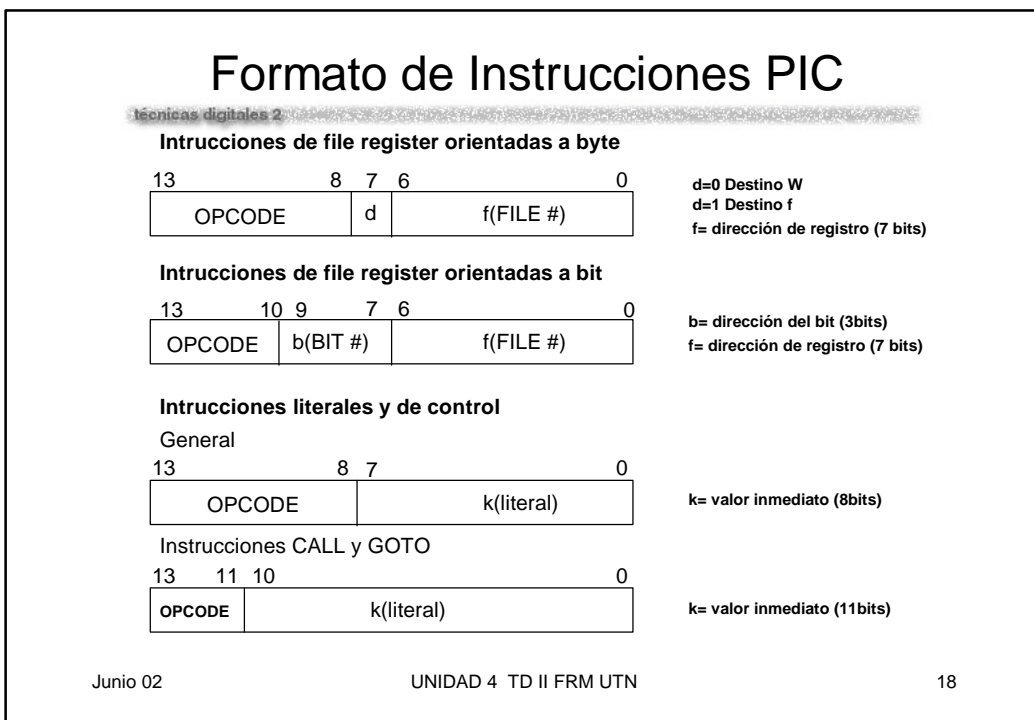
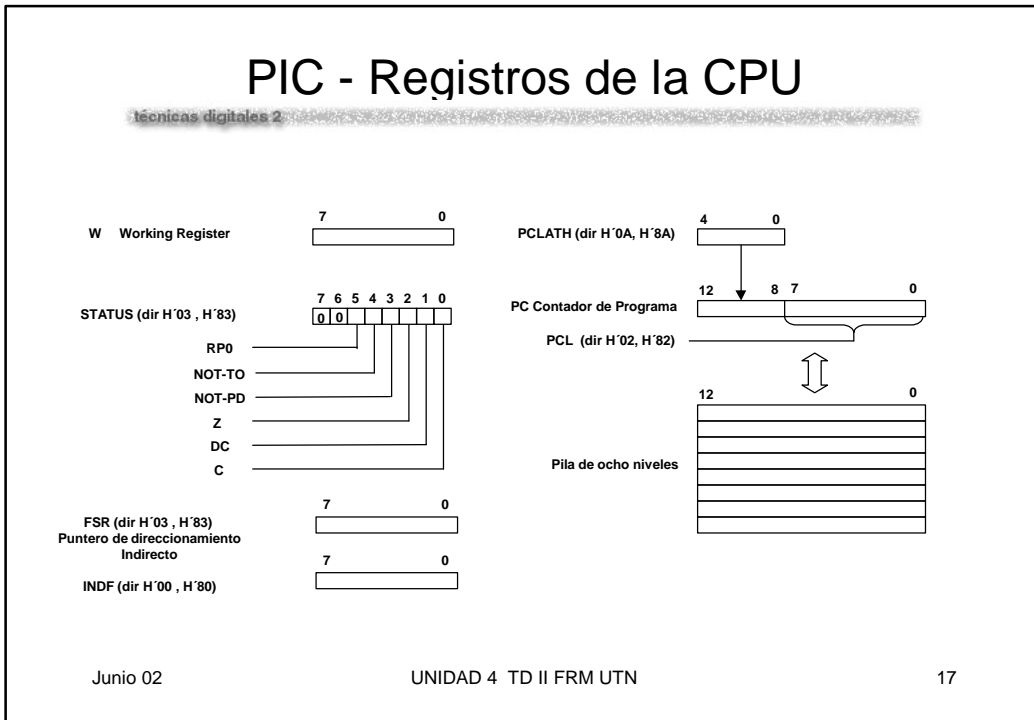
Junio 02
UNIDAD 4 TD II FRM UTN
15

PIC Direccionamiento indirecto

técnicas digitales 2

- Direccionamiento indirecto usa un registro como puntero de dirección.
- FSR (File Select Register) es el puntero de indirección
- Direccionamiento útil para el manejo de tablas

Junio 02
UNIDAD 4 TD II FRM UTN
16



PIC Conjunto de Instrucciones (16CXX)

técnicas digitales 2

Mnemónicos	Op	Descripción	Ciclos	Status
OPERACIONES DE REGISTROS ORIENTADAS A BYTE				
ADDWF	f, d	Add W and f	1	C,DC,Z
ANDWF	f, d	AND W with f	1	Z
CLRF	F	Clear f	1	Z
CLRWF	-	Clear W	1	Z
COMF	f,d	Complement f	1	Z
DECF	f, d	Decrement f	1	Z
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	
INCF	f, d	Increment f	1	Z
INCFSZ	f, d	Increment f, Skip if 0	1(2)	
IORWF	f, d	Inclusive OR W with f	1	Z
MOVF	f, d	Move f	1	Z
MOVWF	f	Move W to f	1	
NOP	-	No Operation	1	
RLF	f,d	Rotate Left f through Carry	1	C
RRF	f, d	Rotate Right f through Carry	1	C
SUBWF	f, d	Subtract W from f	1	C,DC,Z
SWAPF	f, d	Swap nibbles in f	1	
XORWF	f, d	Exclusive OR W with f	1	Z

Junio 02

UNIDAD 4 TD II FRM UTN

19

PIC Conjunto de Instrucciones (16CXX)

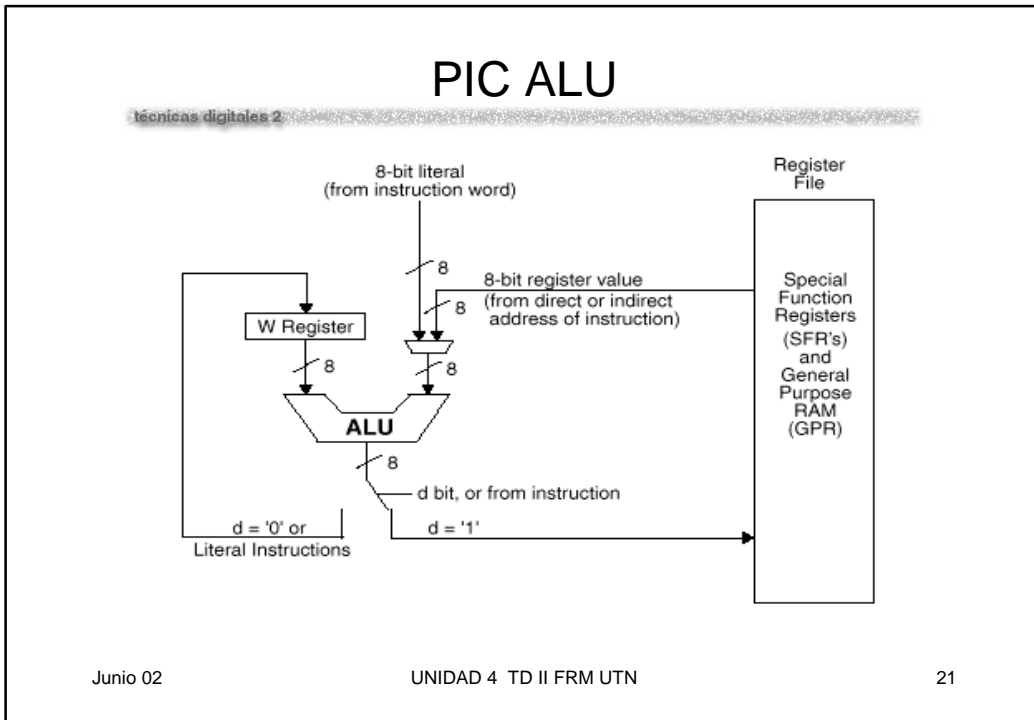
técnicas digitales 2

Mnemónicos	Op	Descripción	Ciclos	Status
OPERACIONES DE REGISTROS ORIENTADAS A BIT				
BCF	f ,b	Bit Clear f	1	1,2
BSF	f,b	Bit Set f	1	1,2
BTFSF	f,b	Bit Test f, Skip if Clear	1 (2)	3
BTFSN	f,b	Bit Test f, Skip if Set	1 (2)	3
OPERACIONES LITERALES Y DE CONTROL				
ADDLW	k	Add literal and W	1	C,DC,Z
ANDLW	k	AND literal with W	1	Z
CALL	k	Call subroutine	2	
CLRWDT	-	Clear Watchdog Timer	1	TO,PD
GOTO	k	Go to address	2	
IORLW	k	Inclusive OR literal with W	1	Z
MOVLW	k	Move literal to W	1	
RETFIE	-	Return from interrupt	2	
RETLW	k	Return with literal in W	2	
RETURN	-	Return from Subroutine	2	
SLEEP	-	Go into standby mode	1	TO,PD
SUBLW	k	Subtract W from literal	1	C,DC,Z
XORLW	k	Exclusive OR literal with W	1	Z

Junio 02

UNIDAD 4 TD II FRM UTN

20



PIC Ejemplo Nro1

técnicas digitales 2

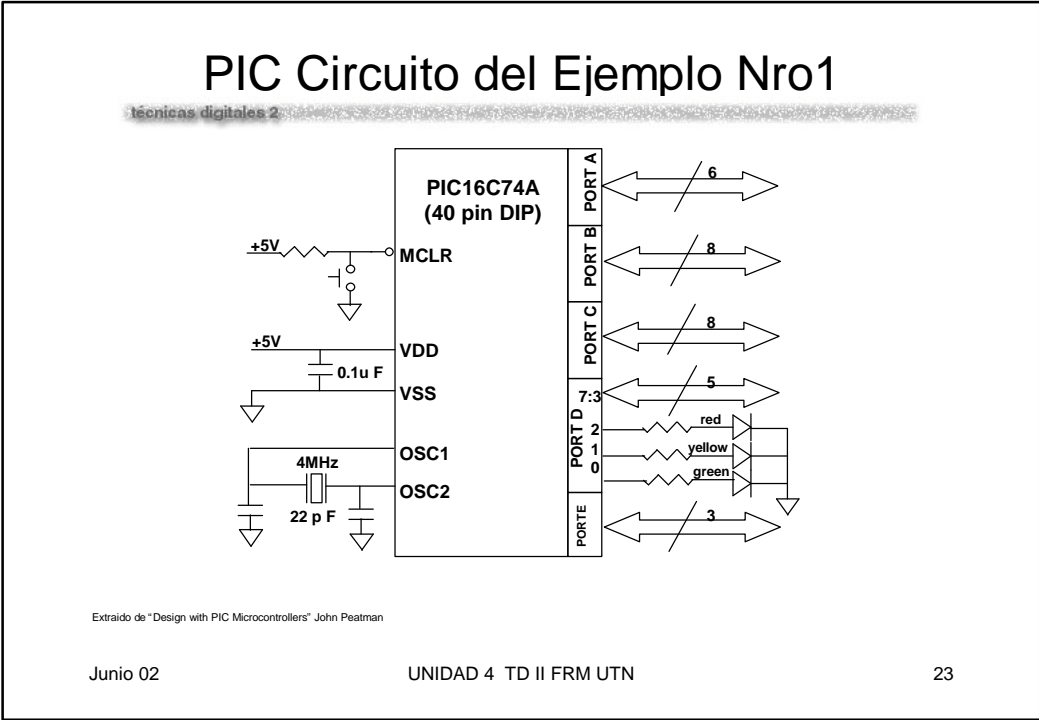
LED DRIVER

- Encender y apagar el LED Verde cada segundo
- Circuito muy sencillo
- Se presenta primer programa
- Se usan subrutinas
- Se presenta el uso del MPASM

- Code Profile
 - Encabezamiento y jerarquía de programa
 - Directivas al ensamblador
 - Definición de Constantes (equ)
 - Definición de Variables
 - Vectores de Interrupción y Reset
 - Código principal (Mainline) y sus subrutinas
 - Código de inicialización

Extraido de "Design with PIC Microcontrollers" John Peatman

Junio 02
UNIDAD 4 TD II FRM UTN
22



PIC Programa Nro1 del Ejemplo Nro1

tecnicas digitales 2

```

#####
; TECNICAS DIGITALES II - UTN FRM
; Programación en PIC
; Ejercicio Nro1
; Toggle the green led every half second
; Count cycles to obtain timing
; Use 4 MHz crystal for 1 microsecond internal
  clock period
;
; Extraido de "Design with PIC microcontrollers" J.
  Peatman, Prentice Hall
; Capitulo 2 Pl.ASM
#####

list P=PICC74A, F=INH8M, C160, N=8, ST=OFF,
  MM=OFF, R=DEC
include "C:\MPLAB\P16C74A.INC"
_config (_CP_OFF & _PWRITE_ON & _XT_OSC &
  _WTD_OFF & _BODEN_OFF)
errorlevel -302 ; Ignorar "error" cuando se
  almacana en BANK1

#####;equates#####
Bank0RAM equ          H'20'
MaxCount equ           50
  
```

```

Green          equ          B0'00000001'
TenMsh         equ          13
TenMsl         equ          250
#####;variables#####
cblock Bank0RAM
BLNKCNT
COUNTH
COUNTL
endc
#####;vectores#####
org H'000'
goto MainLine
org H'004'
Stop
goto Stop
#####;MainLine Program#####
MainLine
call Initial
MainLoop
call Blink
call TenMs
goto MainLoop
  
```

Junio 02 UNIDAD 4 TD II FRM UTN 24

Programa Nro1 cont'

técnicas digitales 2

```

;;;;;;;;;Initial Subrutine;;;;;;;;;
Initial
movlw MaxCount
movwf BLKCNT
movlw Green
movwf PORTD
bsf STATUS, RP0
clrf TRISD
bcf STATUS, RP0
return
;;;;;;;;;Blink Subrutine;;;;;;;;;
Blink
decfsz BLNKLCNT,F
goto BlinkEnd
movlw MaxCount
movwf BLNKLCNT
movlw Green
xorwf PORTD,F
BlinkEnd
return

;;;;;;;;;TenMs Subrutine;;;;;;;;;
TenMs
nop
movlw TenMsH
movwf COUNTH
movlw TenMsL
movwf COUNTL
Ten_1
decfsz COUNTL,F
goto Ten_1
decfsz COUNTH,F
goto Ten_1
return
end
    
```

PIC Ejemplo Nro. 2

técnicas digitales 2

- Idem anterior
 - Encender en secuencia los tres LEDs con una duración de 0,5 seg. cada uno
- Mismo circuito
- Para rutina de tiempo
 - Se usa Timmer2
 - Se usa Interrupciones

- Code Profile
 - Encabezamiento y jerarquía de programa
 - Directivas al ensamblador
 - Definición de Constantes (equ)
 - Definición de Variables
 - Vectores de Interrupción y Reset
 - Una Tabla
 - Código principal (Mainline) y sus subrutinas
 - Código de inicialización
 - Rutina de Servicio de Interrupciones
 - Resguardo de W y STATUS
 - Rutina de Polling
 - Subrutina específica de manejo de interrupciones
 - Control de Tiempo (Looptime)

Programa Nro 2

tecnicas digitales 2

Idem anterior, agregar

```

;;;;;;;;;BlinkTable subrutine ;;;;;;;;;
BlinkTable
    movf PORTD,W
    andlw B'00000111'
    addwf PCL,F
    retlw B'00000001' ;(000->001) reinitialize green
    retlw B'00000011' ;(001->011) green to yellow
    retlw B'00000110' ;(010->101) yellow to red
    retlw B'00000010' ;(011->001) reinitialize green
    retlw B'00000101' ;(100->001) red to green
    retlw B'00000100' ;(101->001) reinitialize green
    retlw B'00000111' ;(110->001) reinitialize green
    retlw B'00000110' ;(111->001) reinitialize green
    
```

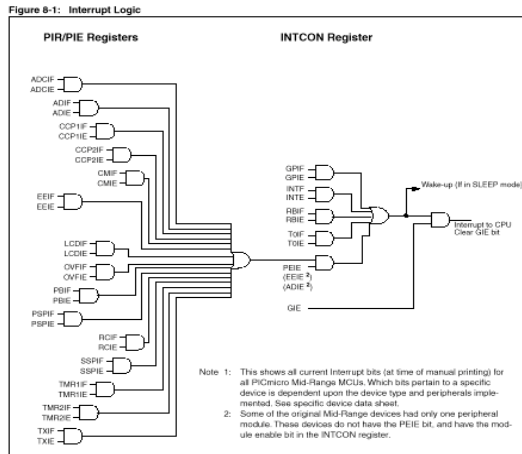
Cambiar

```

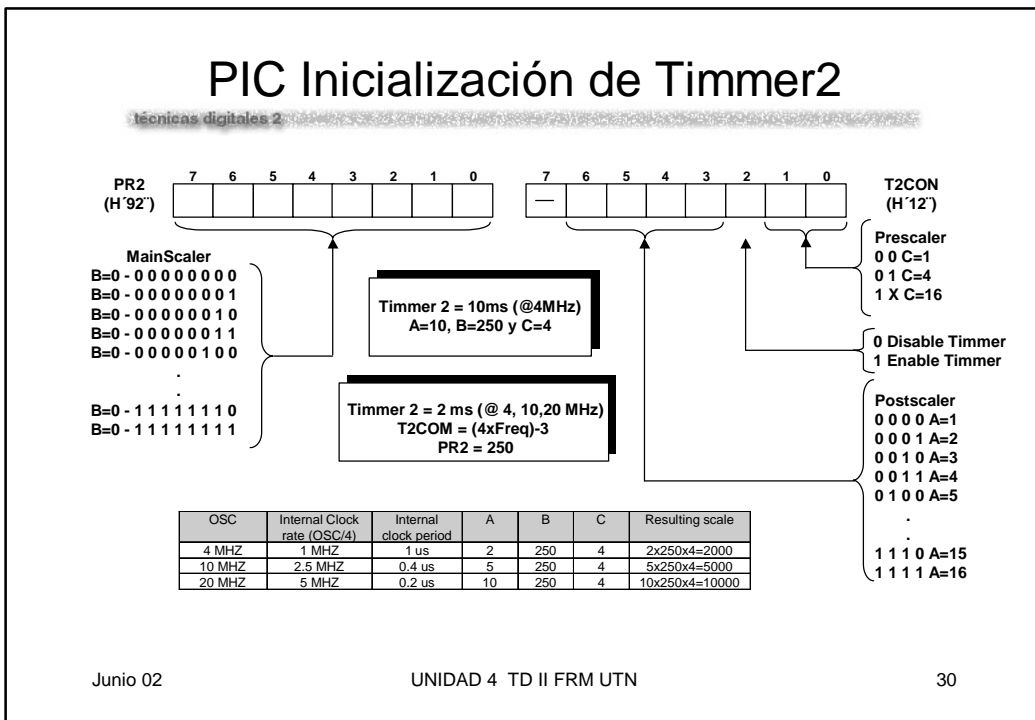
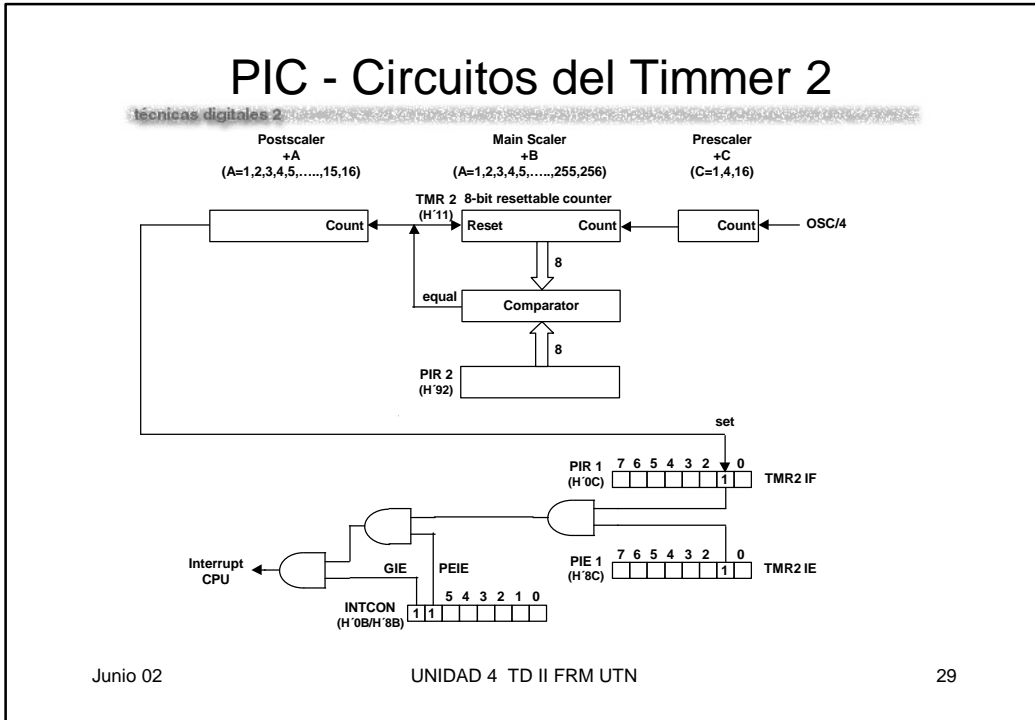
movlw Green por call BlinkTable
    
```

PIC Lógica de Interrupciones

tecnicas digitales 2



Extrado de "PICmicro Mid-Range MCU Family Reference Manual" MICROCHIP 1997



PIC Programa Nro 3

técnicas digitales 2

```

; P3
; Toggle the LEDs every half second in sequence: green, yellow, red,
; green,
; Use 4 MHz crystal for 1 microsecond internal clock period.
; Use Timer2 to obtain 10 millisecond loop time.
;
; Program hierarchy
;
; Mainline
; Initial
; Blink
; LoopTime
;
; IntService
; Timer2
;

list      P=PIC16C74A, F=INH8M, C=160, N=80, ST=OFF,
MM=OFF, R=DEC
include   "C:\MPLAB\PIC74A.INC"
__config (_CP_OFF & _PWRTE_ON & _XT_OSC & _WDT_OFF &
_BODEN_OFF)
errorlevel -302      ;ignore "error" when storing to Bank1
    
```

```

; Equates
Freq      equ 4          ;Crystal frequency in MHz (4, 10, or 20)
Bank0RAM  equ H'20'      ;Start of Bank 0 RAM area
MaxCount  equ 50         ;Number of loops in half a
second
Green     equ B'00000001' ;PORTD mask for green LED
TenMsH    equ 13         ;Initial value of TenMs
subroutine's counter
TenMsL    equ 250
    
```

```

; Variables
cblock    Bank0RAM
W_TEMP   ;Temporary storage for W during
interrupts
STATUS_TEMP ;Temporary storage for STATUS during
; interrupts
SCALER   ;Scale-of-5 scaler used by LoopTime
; subroutine
BLNKCNT  ;LED loop counter
endc
    
```

```

; Vectors
org       H'000'        ;Reset vector
goto     Mainline      ;Branch past tables
org       H'004'        ;Interrupt vector
goto     IntService    ;Branch to interrupt service routine
    
```

Junio 02

UNIDAD 4 TD II FRM UTN

31

PIC Programa Nro 3 con t

técnicas digitales 2

```

; BlinkTable subroutine
; This subroutine reads PORTD and retains only the LED bits. It uses
; them
; to access table. It returns in W the bits of PORTD to be toggled.

BlinkTable
movf     PORTD,W      ;copy present state of LEDs into W
andlw   B'00000111'  ; and keep only LED bits
addwf   PCL,F        ;change PC with PCLATH and offset in
;W
retlw   B'00000001'  ;(000 -> 001) reinitialize to green
retlw   B'00000011' ; 001 -> 010 green to yellow
retlw   B'00000110' ; 010 -> 100 yellow to red
retlw   B'00000010' ;(011 -> 001) reinitialize to green
retlw   B'00000101' ; 100 -> 001 red to green
retlw   B'00000100' ;(101 -> 001) reinitialize to green
retlw   B'00000111' ;(110 -> 001) reinitialize to green
retlw   B'00000110' ;(111 -> 001) reinitialize to green

; End of Tables

; Mainline program

Mainline
call    Initial      ;Initialize everything
    
```

```

MainLoop
call    Blink        ;Blink LED
call    LoopTime     ;Force loop time to be ten
; milliseconds
goto   MainLoop
    
```

```

; Initial subroutine
; This subroutine performs all initializations of variables and
; registers.

Initial
movlw   MaxCount     ;Initialize first blink of LED
movwf   BLNKCNT      ; to last about 0.50 seconds
movlw   Green        ;Turn on green LED; turn off
; others
movwf   PORTD
movlw   (4*Freq)-3   ;Set up Timer2
movwf   T2CON
movlw   4             ;Set up Timer2 subroutine
movwf   SCALER
bsf     INTCON,PEIE  ;Enable Timer2 interrupt source
    
```

Junio 02

UNIDAD 4 TD II FRM UTN

32

PIC Programa Nro 3 con't

técnicas digitales 2

```

bsf STATUS,RP0 ;Set register access to bank 1
cirf TRISD ;Set up all bits of PORTD as
; outputs
movlw 249 ;Set up Timer2
movwf PR2
bsf PIE1,TMR2IE ;Enable interrupt path
bcf STATUS,RP0 ;Set register access back to
; bank 0
bsf INTCON,GIE ;Enable global interrupts
return

;::::: Blink subroutine ;:::::
;
; This subroutine blinks a new LED every 0.5 second.

Blink
decfsz BLNKCNT,F ;Decrement loop counter and
; return if not zero

goto BlinkEnd
movlw MaxCount ;Reinitialize BLNKCNT
movwf BLNKCNT
call BlinkTable ;get bits to change into w
xorwf PORTD,F ;and toggle them into PORTD

BlinkEnd
return

;::::: LoopTime subroutine ;:::::
;
; This subroutine first waits, for however long is necessary, for
; Timer2 and the Timer2 subroutine in IntService to indicate
; that ten milliseconds have passed since the last time that
; they indicated this.

LoopTime
btfss SCALER,7 ;Wait for 00000000 to 11111111
change
goto LoopTime
movlw 5 ;Add 5 to SCALER
addwf SCALER,F
return

;::::: IntService interrupt service routine ;:::::
;
; This interrupt service routine fields all interrupts. It first sets
; aside W and STATUS. It assumes that direct addressing will
; not be used in the mainline code to access Bank 1 addresses
    
```

Junio 02

UNIDAD 4 TD II FRM UTN

33

PIC Programa Nro 3 con't

técnicas digitales 2

```

; (once the Initial subroutine has been executed and interrupts
; enabled). It polls each possible interrupt source to determine
; whether it needs service.

IntService
; Set aside W and STATUS
movwf W_TEMP ;Copy W to RAM
swapf STATUS,W ;Move STATUS to W without
; affecting Z bit
movwf STATUS_TEMP;Copy to RAM (with nibbles
; swapped)

; Execute polling routine
btfsc PIR1,TMR2IF ;Check for Timer2 interrupt
call Timer2 ;If ready, service it
; btfsc ... ;Check another interrupt source
; call ... ;If ready, service it
; btfsc ... ;Check another interrupt source
; call ... ;If ready, service it

; Restore STATUS and W and return from interrupt
swapf STATUS_TEMP,W ;Restore STATUS bits
; (unswapping nibbles)
movwf STATUS ; without affecting Z bit

swapf W_TEMP,F ; Swap W_TEMP
swapf W_TEMP,W ; and swap again into W without
; affecting Z bit
retfie ; Return to mainline code
; reenale interrupts

;::::: Timer2 subroutine ;:::::
;
; This subroutine, called from within IntService, clears the Timer2
; flag, decrements SCALER for use by LoopTime subroutine in
; the mainline code, and returns.

Timer2
bcf PIR1,TMR2IF ;Clear interrupt flag (Bank 0)
decf SCALER,F
return

end
    
```

Junio 02

UNIDAD 4 TD II FRM UTN

34

PIC Interrupciones Externas

técnicas digitales 2

Inicialización de PORT B con tres salidas y cinco entradas de alta impedancia

PIC	PORT B (H'06)	7	6	5	4	3	2	1	0
	TRISB (H'86)	1	1	1	1	0	0	0	1
	OPTION_REG (H'81)	1							

↑ NOT_RBPU=1 (valor por defecto)

Circuito de Teclado con resistores de pullup

Inicialización de PORT B con resistores de pullup

PIC	PORT B (H'06)	7	6	5	4	3	2	1	0
	TRISB (H'86)	1	1	1	1	0	0	0	1
	OPTION_REG (H'81)	0							

↑ NOT_RBPU=0

Junio 02

UNIDAD 4 TD II FRM UTN

35

PIC Inicialización de Interrupciones

técnicas digitales 2

PORT B (H'06)	7	6	5	4	3	2	1
TRISB (H'86)							1

↑ RB0/INT

OPTION_REG (H'81)	INTEDG
-------------------	--------

↑ 1=Input
0=Output

INTCON (H'0B/H'8B)

1		1					
---	--	---	--	--	--	--	--

INTF { set by selected edge on RB0/INT
cleared by selected edge on RB0/INT

INTE { 1: Habilita las interrupciones de RB0/INT
0: Deshabilita las interrupciones de RB0/INT

GIE { Habilitación global de interrupciones
Deshabilitación global de interrupciones

Interrupción a la CPU

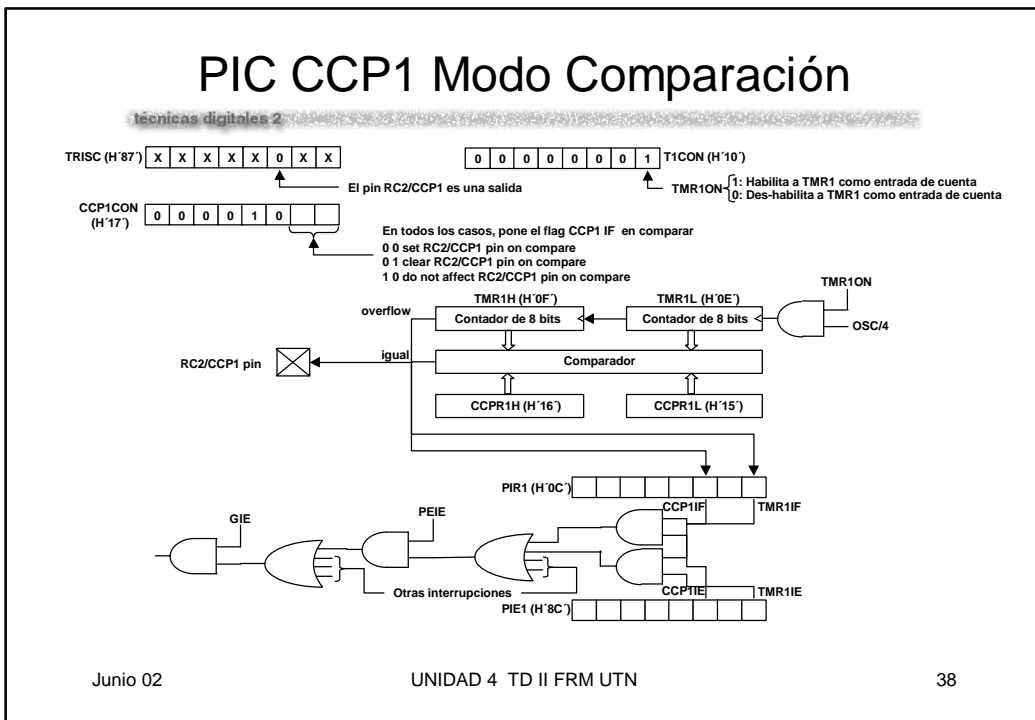
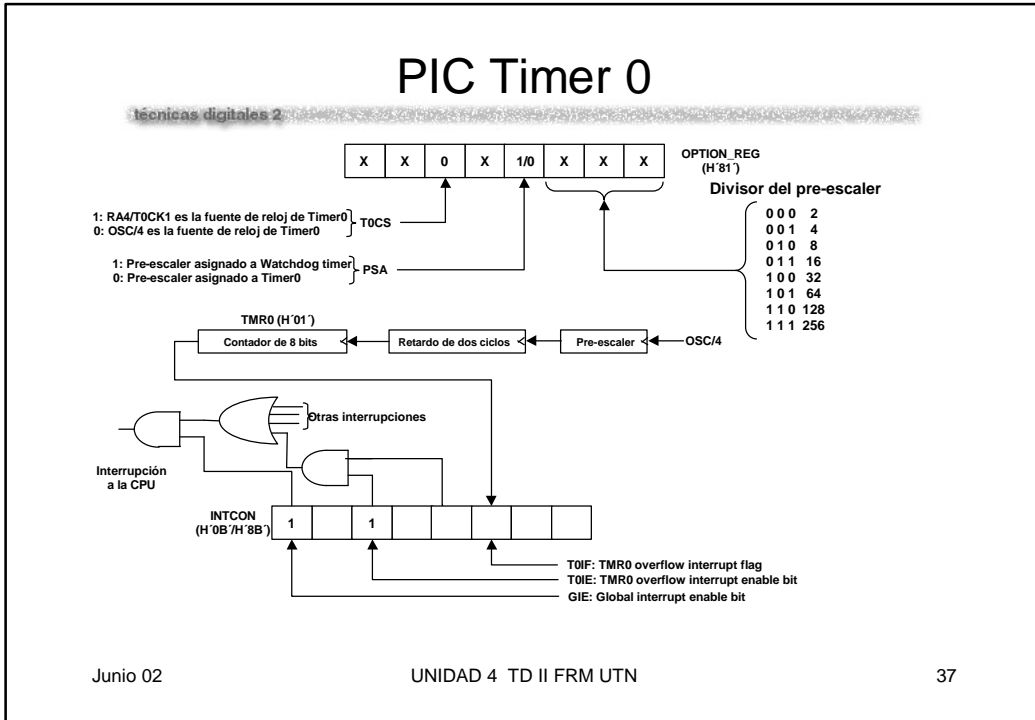
Otras interrupciones

INTEDG { 1: Interrupción en el flanco de subida de RB0/INT
0: Interrupción en el flanco de bajada de RB0/INT

Junio 02

UNIDAD 4 TD II FRM UTN

36



PIC CCP1 Modo Captura

técnicas digitales 2

TRISC (H'87) X X X X X 1 X X T1CON (H'10) 0 0 0 0 0 0 0 1

CCP1CON (H'17) 0 0 0 0 0 1

El pin RC2/CCP1 es una entrada

TMR1ON: $\bar{1}$: Habilita a TMR1 como entrada de cuenta
 0: Des-habilita a TMR1 como entrada de cuenta

En todos los casos, pone el flag CCP1IF en comparador

- 0 0 capture time of every falling edge on RC2/CCP1 pin
- 0 1 capture time of every rising edge on RC2/CCP1 pin
- 1 0 capture time of every 4th rising edge on RC2/CCP1 pin
- 1 1 capture time of every 16th rising edge on RC2/CCP1 pin

CCPR1H (H'16) CCPR1L (H'15)

Transfer when selected edge occurs on RC2/CCP1 pin

Contador de 8 bits TMR1H (H'0F) Contador de 8 bits TMR1L (H'0E)

overflow

CCP1 divider and edge selector $\div 1, \div 4 \text{ o } \div 16$

TMR1ON OSC/4

Interrupción CPU GIE

PEIE

Otras interrupciones

PIE1 (H'8C)

PIR1 (H'0C)

CCP1IF TMR1IF

CCP1IE TMR1IE

Junio 02 UNIDAD 4 TD II FRM UTN 39

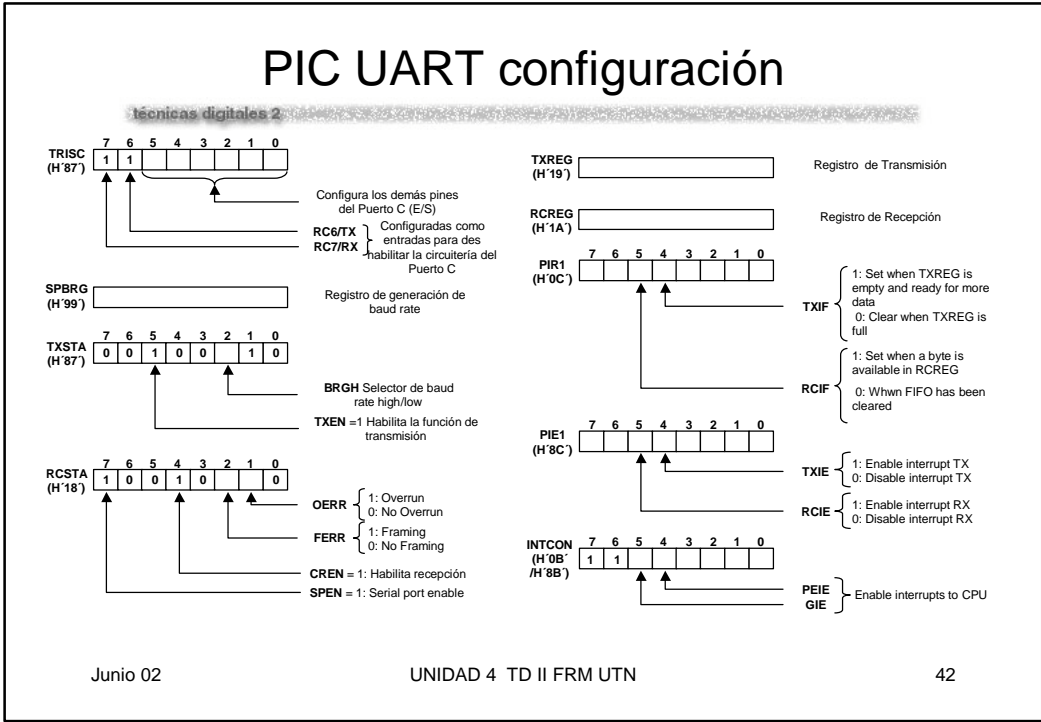
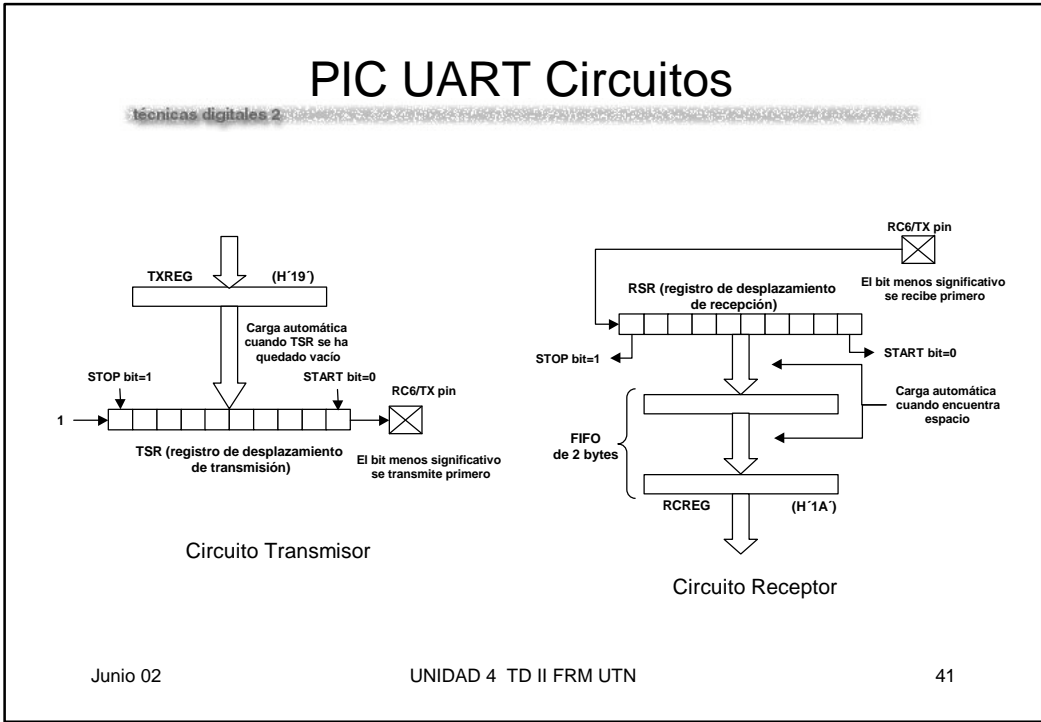
PIC UART

técnicas digitales 2

- **Universal Asynchronous Receiver Transmitter**
- Modo asincrono
- Baud Rate vs Bit Rate
- $BRGH = 1$ Baud Rate = $Osc/16(SPBRG+1)$ (HIGH SPEED)
- $BRGH = 0$ Baud Rate = $Osc/64(SPBRG+1)$ (LOW SPEED)

Nominal Baud Rate	OSC 4mHz			OSC 10 mHz			OSC 20 Mhz		
	BRGH	SPBRG	%error	BRGH	SPBRG	%error	BRGH	SPBRG	%error
9600Baud	1(high)	25	+0.16	1(high)	64	+0.16	1(high)	129	+0.16
19200 Baud	1(high)	12	+0.16	1(high)	32	-1.4	1(high)	64	+0.16

Junio 02 UNIDAD 4 TD II FRM UTN 40



PIC Conversión Analógica a Digital

técnicas digitales 2

- Familia PIC16C7x
- De cinco a ocho canales de entrada
- Multiplexor analógico
- Circuito de "track and hold"
- Reloj de conversión alternativo
- Tasa de muestreo autónomo ajustable
- Referencia de Tensión interna o externa
- Conversión de 8 bits
- Fin de conversión maneja interrupciones

Junio 02

UNIDAD 4 TD II FRM UTN

43

Conclusiones

técnicas digitales 2

- Se ha presentado la familia de micro controladores PIC de MicroChip
- Se ha presentado el modelo estructural
- Se ha presentado el modelo funcional
- Se ha mostrado algunos ejemplos.

Junio 02

UNIDAD 4 TD II FRM UTN

44