TUTORÍA #2

Brando Miguel Palacios Mogollon

Operación: **MOVC**
Función: Mover el byte de código al acumulador
Sintaxis: MOVC A, *registro* @ A +

Instrucciones	OpCode	Bytes	Banderas
MOVC A, @ A + DPTR	0x93	1	Ninguna
MOVC A, @ A + PC	0x83	1	Ninguna

Descripción: MOVC mueve un byte de la Memoria de código al Acumulador. La dirección de la memoria de código desde la que se moverá el byte se calcula sumando el valor del acumulador con DPTR o el contador de programa (PC). En el caso del Contador de programas, la PC primero se incrementa en 1 antes de sumarse con el acumulador.

Ver también: MOV , MOVX

@R0: Memoria apuntada por R0.

#30H: valor de memoria

#=!@

Brando Miguel Palacios Mogollon

EJM:

```
org 0000h
Mov R0,#30h
Mov A,#1
Mov @R0,A       ;es lo mismo que decir mov 30h,A
Add A,#30h
Lcall sndchr
end
```

Brando Miguel Palacios Mogollon

TABLA

Resultado:

Posiciones de memorias llenas del 30h al 3Eh com los valores de la subrutina obtengo_valor:

```
                org 0000h
                mov R0,#30h
                mov R4,#16
                mov R5,#0
lazo:
                mov A,R5
                lcall obtengo_valor
                mov @R0,A
                inc R5
                inc R0
                djnz R4,lazo
                sjmp $
obtengo_valor:
                inc A
                movc A,@A+PC
                ret
db 2Bh,7Fh,19h,88h,08h,1Dh,77h,55h,33h,9Fh,0CCh,0DEh,44h,0B0h,33h
END
```

Brando Miguel Palacios Mogollon

CÁLCULOS

- Modo 0 : 2^13 =8192
- Modo 1: 2^16 = 65536
- Modo 2: 2^8 = 256

Caso 1:

Kx2^nx(12/11.0592x10^6)s=1/f s donde f: frecuencia

Cuando el K es menor a 1 siendo un valor poco acotado

Caso 2:

(2^n)x(12/11.0592x10^6)s=0.033s

Nota: el resultado de K y j se almacenan en la posición TH0 como prioridad, se sobrar se lo coloca en TL0 como un numero mas.

Brando Miguel Palacios Mogollon

K VS J

El valor de K es un valor multiplicativo, es decir son las veces que el TF0 debe detectar un overflow por lo q no es necesario iterar con bastante información. Es decir con el TH0 esto mas se utiliza cuando el timer se encuentra en modo 0.

El valor J es una cota es decir el programa va a empezar desde J hasta que sucesa el overflow.

Brando Miguel Palacios Mogollon

POR EJM: TIMER 0 EN MODO 1 FREC 15HZ

periodo : T=1/15=0.066

Medio periodo : T/2 = 0.033

Caso 1:

Kx2^16x(12/11.0592x10^6)s=0.033 → k: 0.46 no se puede

Caso 2:

(65536-j)x(12/11.0592x10^6)s=0.033s 35123.2>>35124 >> 89 33h

>> TH0: 89h

>> TL0: 33h

Brando Miguel Palacios Mogollon

EJM CON K

Brando Miguel Palacios Mogollon

UN SEGUNDO REAL EN TIMER 0 MODO 0 Y MODO 1

periodo=1/1hz=1s semiperiodo=0.5s

k*2^13*(12/11.0592*10^6)=0.5s ;k=56

```
        org 0000h
        mov TMOD,#20h
        setb TR0
segundo:
        mov R6,#56
loop:
        jnb TF0,$
        clr TF0
        cpl P1.0
        djnz R6,loop
        end
```

periodo=1/1hz=1s semiperiodo=0.5s

k*2^16*(12/11.0592*10^6)=0.5s ;k=7

```
        org 0000h
        mov TMOD,#21h
        setb TR0
segundo:
        mov R6,#7
loop:
        jnb TF0,$
        clr TF0
        cpl P1.0
        djnz R6,loop
        end
```

Brando Miguel Palacios Mogollon

EJM CON J

Brando Miguel Palacios Mogollon

Hacer un programa que genere una frecuencia de 15hertz en el pin P1.0 del Puerto 1. Seleccionar en que modo quiero trabajar modo 1 del timer0

periodo = 1/15hz=0.066s semiperiodo=0.033s (65536-j)*(12/11.059*10^6)s=0.033s

tenemos que j=34816 necesitamos q el microcontrolador cuente desde 34816 hasta 65536

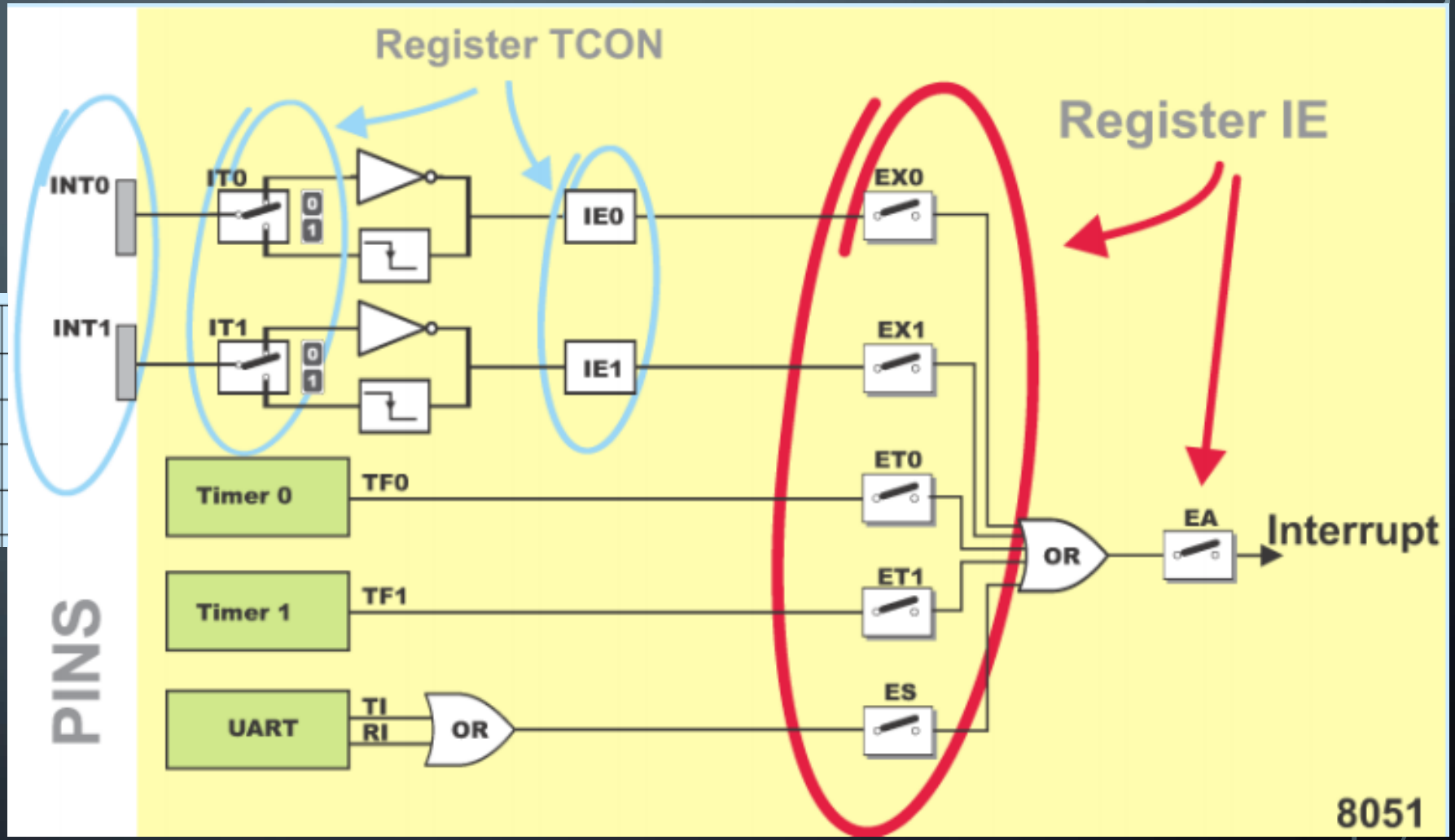
```
        org 8000h
        mov TMOD,#21h;
otra_vez:
        mov TH0,#88h;regitro interno
        mov TL0,#0h
        setb TR0;inicio el temp
        jnb TF0,$
        clr TF0
        cpl P1.0
        clr TR0
        sjmp otra_vez
        end
```

EJM 1

Brando Miguel Palacios Mogollon

INTERRUPCIONES:

Interrupción	Bandera
Externo 0	IE0
Externo 1	IE1
Timer 0	TF0
Timer 1	TF1



Brando Miguel Palacios Mogollon

SUBRUTINA SETINTVEC (SETINVEC EQU 145H)

Permite el salto de la memoria a la interrupción necesaria se clasifica de la siguiente manera:

- Mov A,#0 ;interrupción externa 0
- Mov A,#2 ; interrupción externa 1
- Mov A,#1 ;interrupción interna 0 (timer 0)
- Mov A,#3 ; interrupción interna 1 (timer 1)

Brando Miguel Palacios Mogollon

INSTALACIÓN DE INTERRUPCIONES

```
        org 8000h
        mov A,#0 ; Selecciono interrupción  externa 0
        mov dptr,#rutina_interrup_ext0
        lcall setinvec
        setb EX0 ; Habilita la interrupción externa 0
        setb EA  ; Habilita interrupciones globales
        setb P3.2   ;Instalo la interrupción para el botón P3.2
        ;Código siguiente a la instalación
        sjmp $                  ;ljmp  2f0h
rutina_interrup_ext0:
        clr EX0
        ;La interrupción que deseas colocar
        setb EX0
        reti
        end
```

Brando Miguel Palacios Mogollon

```
        org 8000h
        setb IT0 ; Configuro para considerar el flanco de bajada
        mov A,#0
        mov dptr,#rutina_interrup_ext0
        lcall setinvec
        setb EX0
        setb EA
        setb P3.2
        setb P1.0
        sjmp $
rutina_interrup_ext0:
        clr EX0
        mov A,#100
        lcall delay
        cpl P1.0
        mov A,#200
        lcall delay
        setb EX0
        reti
        end
```

EJM: BOTÓN DE LED

PARA OPERACIONES CON BOTONES EN UNA INTERRUPCIÓN SE UTILIZA IT0 PARA QUE EL MICROCONTROLADOR PUEDA DETECTAR LAS PULSACIONES CUANDO EL BIT SEA 1.

CABE RECALCAR QUE DADO QUE EL BOTÓN REALIZA REBOTE ES BUENO PONER UN RETARDO.

Brando Miguel Palacios Mogollon