David González Fabián

Ingenieria electrónica industrial y automática.

*P4 eLECTRONICA DIGITAL Y MICROPROCESADORES*

*Resolución de ejemplos de aplicaciones, diseñando las características delhardware, la programación, ensamblado, linkado y simulación, con la iniciacióna las herramientas de desarrollo*

## En base al valor contenido en los dos bits menos significativos de la posición de memoria 2000h de memoria de datos externa, realizar las operaciones señaladas en la tabla 2, teniendo en cuenta que los operandos se encuentran en las posiciones señaladas.

## OPERANDOS Posición de memoria de datos interna

## Operando 1 **50h** operando 2, (sustraendo,divisor) **51h**

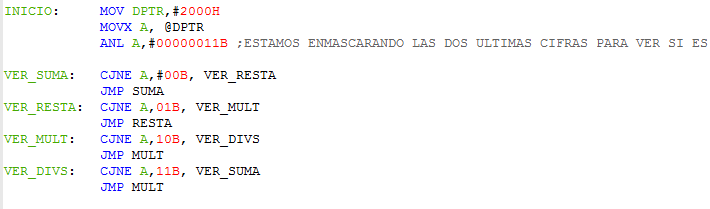
## Valor de los dos bits menos significativos de la posición 2000h de memoria de datos externa Operación a realizar

## **00 Suma 01 Resta 10 Multiplicación 11 División**

## El resultado se deberá dar en formato BCD y a partir en las posiciones 60h (unidades), 61h (decenas) y 62h (centenas) de memoria de datos interna. El resultado será correcto incluso con existencia de acarreo en la suma. En el caso de resta, se considera que los valores de los operandos siempre son positivos. Si el resultado es negativo se deberá colocar un 01 en la posición 63h.

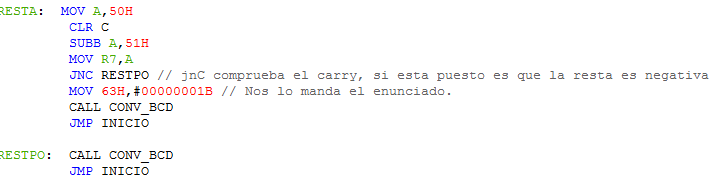
El primer paso será colocar el data pointer en la posición 2000H y mover lo que haya al acumulador. Ahí enmascaramos con un AND lógico las ultimas dos cifras que son las que nos interesan.

Despues lo pasamos un CJNE compara y salta si no es igual, si nuestra A acabase en 10, se saltaría la suma, y la resta, pero en el cjne pasaría de línea lo que le llevaría a Multiplicacion



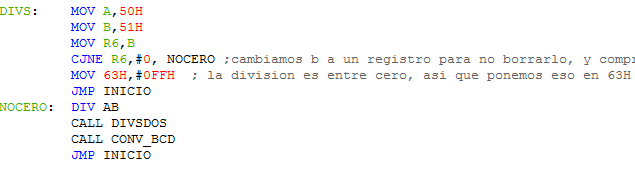
La suma y la multiplicación son muy sencillas, solamente movemos las posiciones 50 a A y a B la 51 (en el caso de la suma ni haría falta) y guardamos el resultado en R7. Los resultados tienen que estar en BCD asi que llamamos a una subrutina para que lo acabe.

Pero para la resta hay que marcar un 01 en la posición 63h en caso de que sea positivo, asi que llamamos a una subrutina en caso de que el carry no este ahí, JNC , jump if not carry , pero si es negativa, ira a la siguiente línea que mueve a la posición 63H el 01Binario

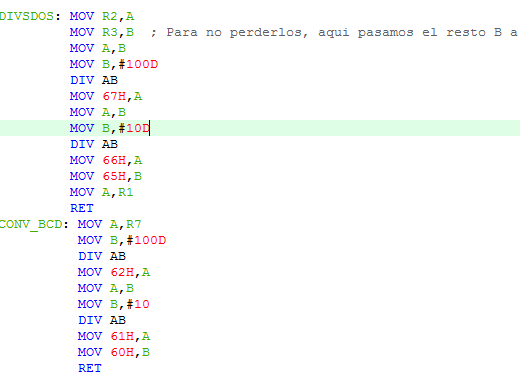


Para la división tenemos que hacer dos cosas

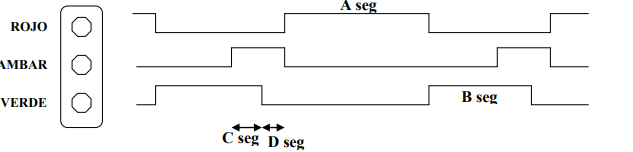
primero asegurarnos que B no sea cero, para ello hacemos lo mismo que en la reste pero en vez de usar JNC usamos CJNE(compara y salta si no son iguales con el valor B (guardado en R antes) y el cero, si son iguales en el 63h dejamos un 0ff y volvemos a empezar.



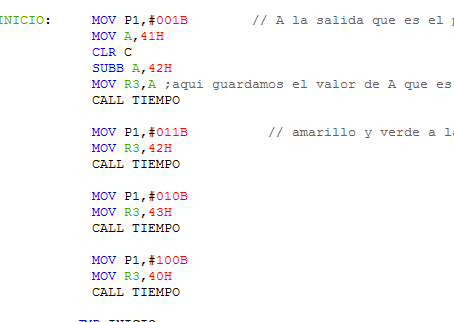
Y segundo mover el resto de forma BCD a las líneas 66, 65 y 67. Para ponerlo en BCD dividimos primero entre 100, sacando las centenas, depues entre 10 y sacamos las decenas en un lado y las unidades en otro ,( El comando DIV nos deja resto en B y cociente en A).

Para hacer el CONV BCD hacemos lo mismo que en el resto,dividr entre 100 y luego entre 10. 

# Realizar un sistema que permita controlar el funcionamiento de un semáforo. Éste deberá repetir el siguiente ciclo: Los valores A, B, C y D se encuentran en las posiciones 40, 41, 42 y 43h de memoria de datos interna



Primero identificamos las posiciones de los leds del semáforo que existen.

Rojo, dura el valor de 40H ( que definiremos como 10 segs x ejemplo ) pero el ámbar tiene un problema, ya que dura C+D segundos y además se enciende a la vez que el Verde (B esta encendido. Asi que creamos 4 estados, Verde, ámbar rojo y ámbar y verde a la vez ( C segs) 

Este de aquí es el verde solo, que es 41-42 ( B-C)

Hemos supuesto que empieza el ciclo en verde.

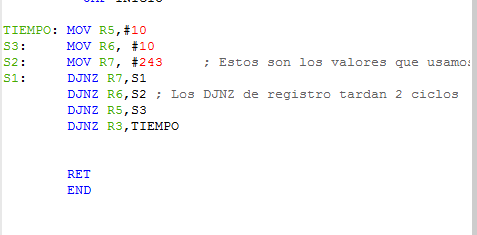
Llamamos a tiempo para que haga subsecuencias durante 41-42H segundos ( A )

011, encendidos p1.1 y p1.2 , amar y verde durante 42H (C segs)

Ámbar solo D segs

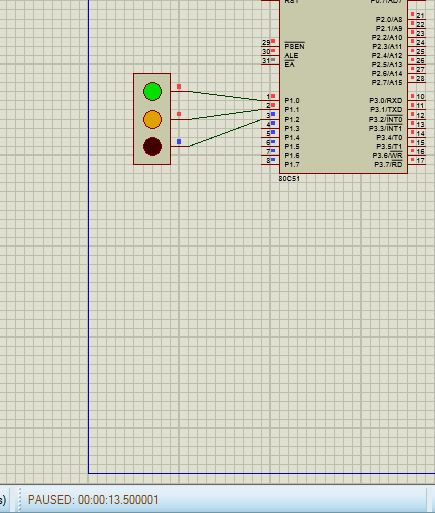
Rojo , A segs y empieza B de nuevo

Para que la secuencia dure 1 segundo tendríamos que hace 1Millon de ciclos maquina, y como cada DJNC consume 2 ciclos máquina, tendríamos que hacer una instrucción 500000 veces. Eso es lo mismo que 213+213\*235+213\*235\*9 = 500700.



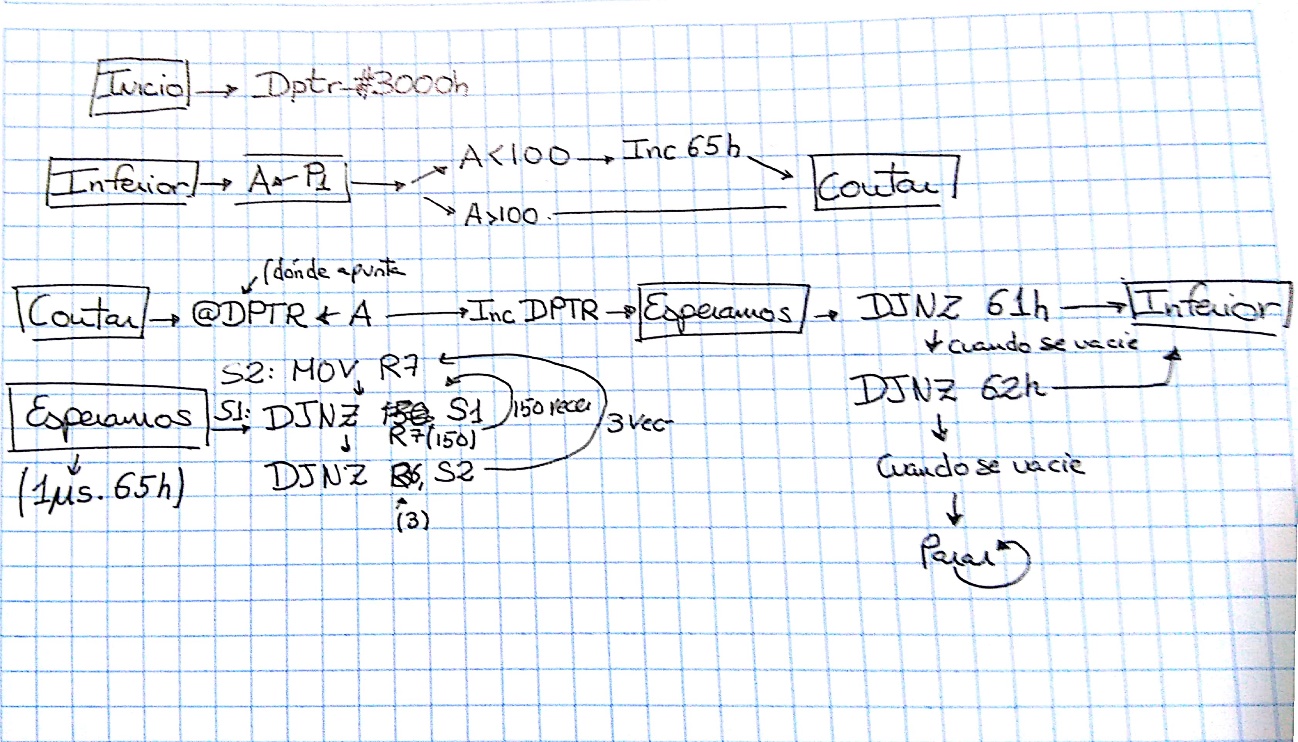
La razón por la cual están cambiados aquí los valores del “contador de tiempo” es por que esos valores en Proteus no dejaban avanzar el semáforo, asi que los invertí y el semáforo iba muy rápido, avanzando frame a frame vi cuanto duraba la secuencia y calcule para que el semáforo fuese a una velocidad adecuada.

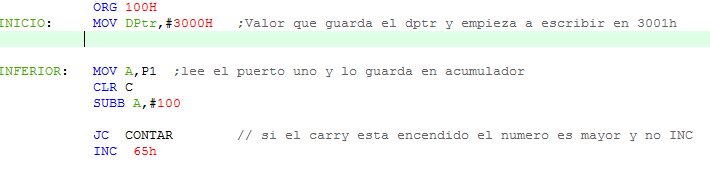
De todas maneras el funcionamiento del “contador de tiempo” es muy sencillo, con los DJNZ , que (es salta si no es cero) empiezan contando el valor del que salta a S1 y lo hace tantas veces hasta que se vacié, entonces va a línea de debajo, la cual le lleva a dos líneas encima suyo donde está la instrucción de llenar el primer registro otra vez con el valor original, lo que lo hace volver a contar. Si solo hubiera 2 instrucciones, el segundo valor que corresponde a S2 hubiera llenado X veces la primera instrucción, volveria al principio. En nuestro caso hay S1,S2,S3.

.

Para la simulación en proteus hay que crear un archivo .hex, colocar el microcontralor 80c51 y en edit properties buscar el archivo .hex, para el semáforo buscar “Street light” en librerías.

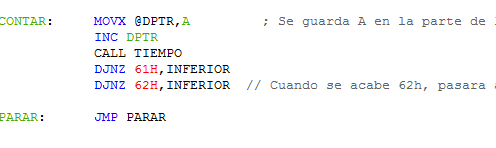
# Realizar un programa que permita la lectura del puerto 1 a intervalos de tantos milisegundos como se indique en la posición 60h de memoria de datos interna, y tantas muestras como se indique en un valor de 16 bits contenido en las posiciones 62h (parte alta) y 61h (parte baja) (sabemos que el número de muestras a guardar no superará 2000). Guardar estos datos a partir 3000h. En la posición 65h se deberá encontrar cuántos datos de los recibidos hasta ese momento, son inferiores a 100d.

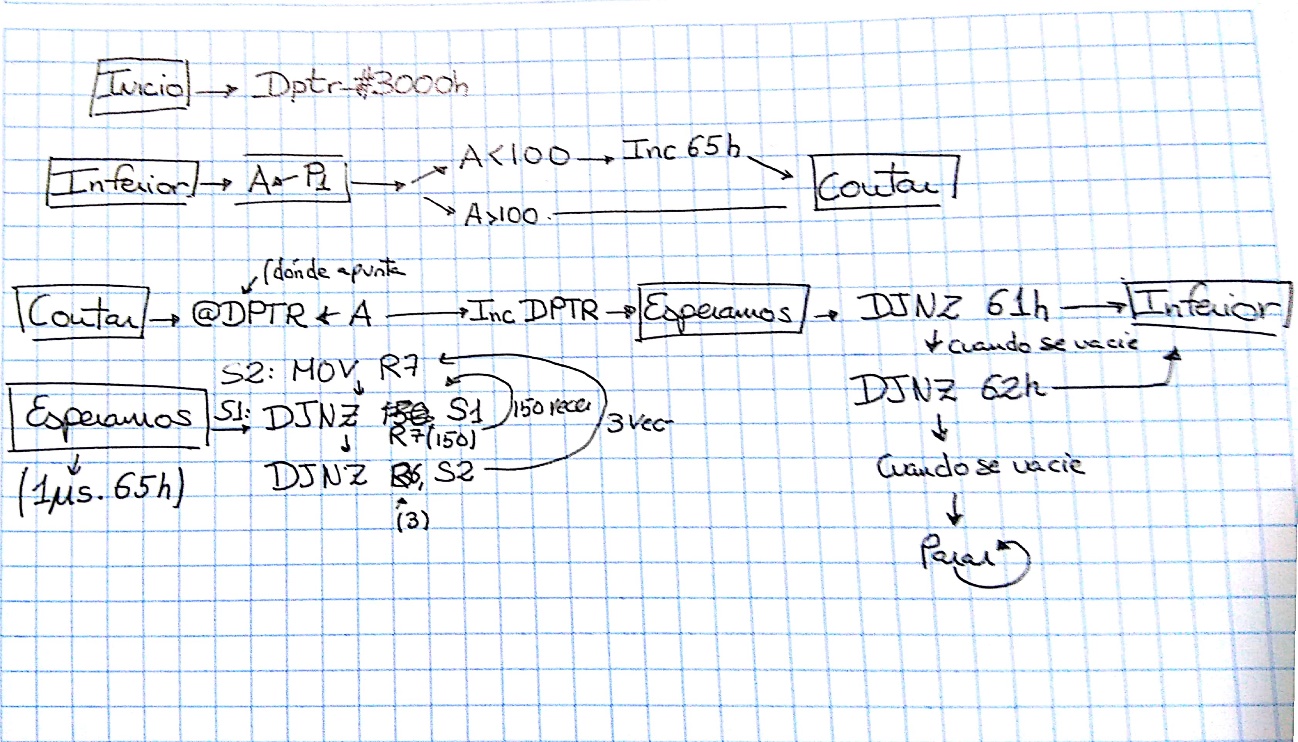




Aquí el dptr empieza en 3000 como dice el enunciado y guarda en acumulador lo que entra por el puerto 1

Después mira si el numero que ha entrado es menor que 100, para ello lo resta, y si el carry es 1 es que la resta ha dado negativo y aumenta +1 el valor de la posición 65h. la siguiente línea es contar aumente 65h o no.



Aquí nos dice que A se guarda en la dirección en la cual el DPTR esta apuntando, y después apunta a la siguiente posición con el INC dptr. Como hay que esperar (60H)milisegs, llamamos a tiempo para que cuente, tras acabar el valor de 61/62h nos dice cuantos datos vamos a contar, así que le bajamos un dato, primero a 61H y cuando 61H esta vacío se lo bajamos a 62H. Después de eso entra en bucle sin hacer nada. Mientras se esta vaciando vuelve a llevarlo al origen donde A va cambiando