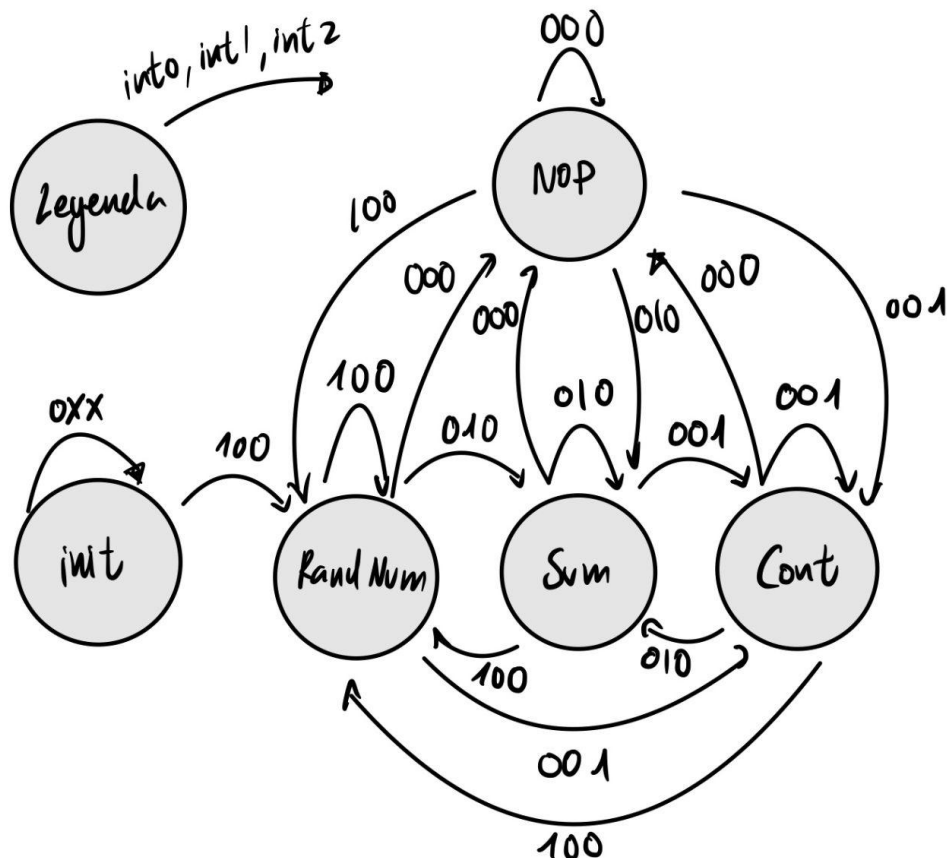


QÜESTIONARI-L3 (B) Display 7 segments

Nom i Cognoms: Sergio Shmyhelsky & Songhe Wang

Grup LAB: 31

1) Dibuixa un diagrama de flux amb els estats i les transicions del programa descrit a l'apartat 5. Considerarem que solo se puede pulsar un botón al mismo tiempo.



2) Indiqueu el contingut dels següents registres (en binari) just després d'haver saltat el hardware i el software breakpoint en INT0 (segons us demanem a la secció 5.6 i 5.7 d'aquest enunciat).

Hardware breakpoint:

INTCON = 0x92

INTCON2= 0xE5

INTCON3 = 0xC0 / 0xD8

Software breakpoint:

INTCON = 0x12

INTCON2= 0xE5

INTCON3 = 0xC0 / 0xD8

Dependiendo de si es la primera vez que se ejecuta INT0, tenemos dos valores posibles de INTCON3.

3) Quin és el elapsed time que us indica Proteus (temps d'execució entre dos breakpoints consecutius, indicat en la barra inferior), des de l'instant en que salta el hardware breakpoint al apretar el botó associat a INT1 fins al software breakpoint en la primera línia de la RSI? Justifica aquest retard.

1.9555 microsegons

Este es el tiempo que necesita el procesador para saltar a la rutina de ISR, normalmente tardamos unos 12 a 20 ciclos de reloj, dependiendo de la última instrucción que se ha ejecutado antes de la interrupción. Si consideramos que la simulación se hace con la micro a 8MHz, entonces multiplicando podemos ver que efectivamente el tiempo estimado es de entre 1.5 a 2.5 microsegundos.

4) Quin és el temps que es triga a pintar un número de 4 xifres als displays de 7-segments?
8.0680ms (8MHz)

5) Quin és el temps que triga el vostre codi de la RSI de la INT0?

1.1220ms (8MHz)

Un client ens pregunta quin és el límit de funcionament del nostre sistema. Li hem de contestar de forma raonada estressant el sistema tal i com es menciona al punt 5.7. Decidim dividir la resposta en dues parts, la primera consisteix en simular quan la informació dels displays deixa de ser útil. La segona consisteix en trobar el punt on ja no hi ha manera de que el codi evolucioni de forma correcta.

6) Quin és el límit de funcionament (freqüència o temps de la senyal d'estrès) on deixen de funcionar els displays.

Después de numerosas pruebas, al final hemos detectado que cuando la frecuencia de entrada es superior a 1MHz, el display deja de funcionar, es decir, aparecen números sueltos sin ningún criterio.

7) Quin és el límit de funcionament (freqüència o temps de la senyal d'estrès) on deixa d'evolucionar el programa.

Esta frecuencia curiosamente coincide con la anterior, hasta 999.9kHz el display funciona correctamente. Una vez lo ponemos a 1MHz el display deja de funcionar y al cabo de 3 segundos aproximadamente el simulador deja de funcionar también, que lo entendemos en el sentido de que el programa ha dejado de evolucionar.

8) Escriu una breu explicació que justifiqui aquests valors tenint en compte que segurament tindreu dues parts clarament diferenciades al vostre codi, una part al bucle infinit que gestiona el display i fa servir delays, i una altra a la RSI o interrupció que gestiona el funcionament del conjunt.

Una posible respuesta al porqué existe este límite de funcionamiento podría ser debido al hecho de llamar tantas veces a la rutina de interrupción, se llega hasta el punto dónde impide el funcionamiento correcto del código del main. Los cuatro números del display se ven al mismo tiempo gracias a que hacemos iteraciones y encendemos los leds secuencialmente para que dé la sensación visual, pero si entre iteración e iteración hay muchas interrupciones que ralentizan la tasa de refresco el display dejará de funcionar correctamente.

9) Com podries millorar els límits de funcionament?

Durante la realización del trabajo, como era en parejas, hemos hecho dos versiones del mismo trabajo, donde en un uno se dividían las interrupciones en high y low y en otra versión solo con high. Hemos realizado las mismas pruebas y hemos observado que en el código donde solo habían rutinas high tiene un límite de funcionamiento más alto que el otro.

Este hecho es debido a que en el código solo hay una rutina, de manera que en una sola rutina podemos tratar más de una interrupción (por cláusulas seguidas de if). Esto permite que cuando haya una frecuencia elevada de interrupciones se puedan tratar la mayoría (en el código de las rutinas una para cada tipo de prioridades, si durante un ejecución de una rutina high entra una interrupción considerada como low, ésta última se ignorará).

Otra solución podría ser aumentar el tiempo de reloj, ya que así conseguimos que las interrupciones tengan un impacto menor en el funcionamiento correcto del display. También podría ser buena idea intentar reducir el código, ya sea mejorar la eficiencia o bien sustituir parte de la RSI con código de máquina,

Si nuestro objetivo es solamente mejorar el límite de funcionamiento, entonces también podría ser buena idea ignorar voluntariamente interrupciones para que el sistema no se sature.