

SEGUNDA ACTIVIDAD EVALUABLE

1: *Asignatura:* **FUNDAMENTOS de SISTEMAS DIGITALES**

2: *Título de la Actividad:* **Diseño, Implementación y Simulación de un Circuito en Lógica Secuencial**

3: *Datos personales:*

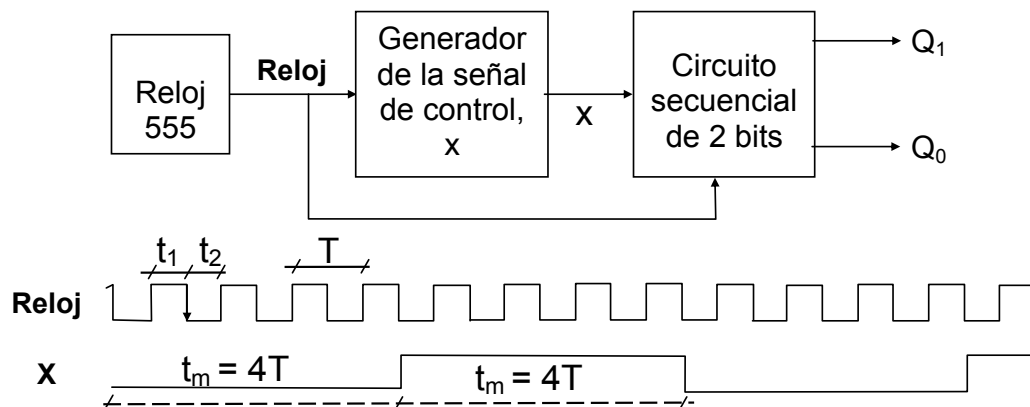
- Nombre y Apellidos:
- DNI:
- Centro asociado: **Cádiz**

4: *Código de la actividad que le ha correspondido realizar:* **A-E-2-097.doc**

5: *Enunciado.*

Enunciado A-E-2-097.doc

Esta actividad consiste en el diseño e implementación en el un circuito secuencial que consta de tres bloques funcionales que realizan funciones claramente diferentes. El primer bloque consiste en construir un reloj con el circuito de tiempo 555 funcionando como astable. El segundo bloque consiste en obtener, a partir de este reloj, una onda cuadrada de mayor periodo y, por ultimo, el tercer bloque consiste en construir un autómata finito controlado por el tren de pulsos anterior y que actúe de una forma u otra en función de si este pulso está en alta o en baja. Por tanto, el esquema a nivel de diagrama de bloques del circuito a diseñar y simular es el de la siguiente figura:



Diseñe el sistema lógico secuencial de la figura para que el circuito secuencial de 2 bits realice las siguientes funciones:

- Si $x=0$ no cambia de estado. Es decir, permanece en el mismo estado con independencia del estado en el que se encuentre.
- Si $x=1$ y está en el estado 00 ó en el estado 01 pasa al estado 10 y si está en el estado 10 ó en el estado 11 pasa al estado 01.

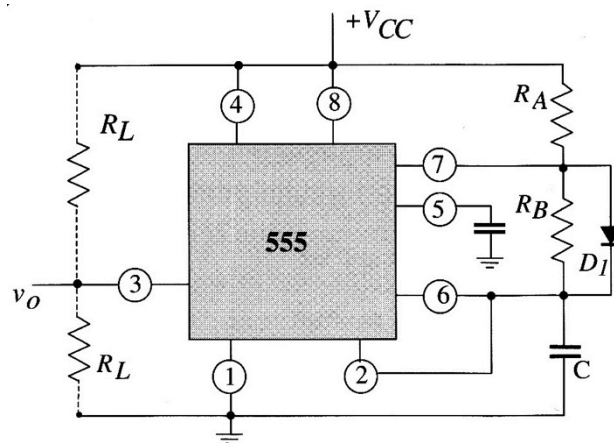
Esta práctica consiste en el diseño de un circuito en lógica secuencial y su posterior implementación, el cual constará de 3 bloques los cuales se de diseñarán, implementarán y se verificaran con el simulador por separado antes de unirlos para formar el esquema completo de la actividad y su posterior verificación.

En el primer bloque se creará un circuito de tiempo basado en el 555 funcionando en modo astable en el que tendremos que conseguir una onda cuadrada en el que el tiempo en alta será igual al tiempo en baja, para ello tomaremos por ejemplo un periodo $T=2\text{ms}$ y por tanto los periodos de la salida serán $t_1=t_2=1\text{ms}$.

El segundo bloque consisten en obtener una onda cuadrada de mayor periodo, utilizando un contador integrado SN74393, concretamente $t_m = 4T$ (el periodo anterior), partiendo de la salida del reloj del bloque anterior.

Para terminar tenemos que diseñar y construir el tercer bloque, un autómata finito controlado por el tren de pulsos generado en el bloque 2 y que actuará en función de si el pulso está en alta o baja según el enunciado del problema.

Comenzamos por el primer bloque, configurando un circuito 555 en modo astable, como muestra la figura siguiente.



Esquema de un 555 en modo astable

Primero miramos la alimentación que requiere esté circuito, para ello miramos su datasheet para no pasarnos de los límites. Como la salida de este circuito se va a conectar al contador integrado SN74393 miramos en datasheet del 74393 que valores de tensión admite, viendo que la tensión nominal del contador es 5v nuestro reloj estará alimentado también por 5 voltios.

Una vez que tenemos la alimentación tenemos que asegurarnos que los valores de tensión en la salida V_0 (pin 3) son lo más cercanas a 5v y a 0v, lo que constituiría el "1" y el "0" lógico respectivamente. Para ello tenemos que poner las resistencias R_L (R_1 y R_2 en el diseño) a un valor alto, en este caso 1000 k Ω .

La patilla 5 del integrado se usa como entrada de señal de control permitiendo cambiar las tensiones de referencia de los comparadores internos. Ya que esta patilla no se usará para tal fin, hemos de colocarle un condensador del orden de 0,01 μ F para filtrar el ruido de la fuente y evitar pulsos de ruido, por ello colocamos el valor del condensador C_1 a 10 nF.

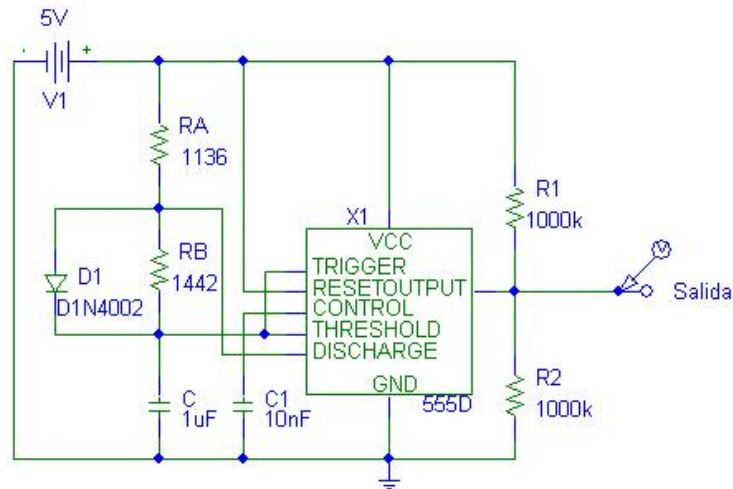
Los valores de R_A , R_B y C los tenemos que calcular mediante las ecuaciones de la página 555 del libro de teoría $t_1 \approx 0,69R_A \cdot C$ y $t_2 \approx 0,69R_B \cdot C$. Como sólo disponemos de 2 ecuaciones y tres incógnitas, hemos de dar un valor arbitrario a una de ellas, decantándonos por el condensador, porque en la practica un condensador de cierto valor no estándar puede ser complicado y caro de conseguir mientras que el valor de las resistencias se puede conseguir de muchas formas, por ejemplo con un potenciómetro. Al condensador C le daremos un valor de 1 μ F, pasando a calcular R_A y R_B :

$$t_1 \approx 0,69R_A \cdot C \quad \Rightarrow \quad R_A = \frac{t_1}{0,69 \cdot C} = \frac{1 \text{ ms}}{0,69 \cdot 1 \mu\text{F}} \approx 1442 \Omega$$

$$t_2 \approx 0,69R_B \cdot C \quad \Rightarrow \quad R_B = \frac{t_1}{0,69 \cdot C} = \frac{1 \text{ ms}}{0,69 \cdot 1 \mu\text{F}} \approx 1442 \Omega$$

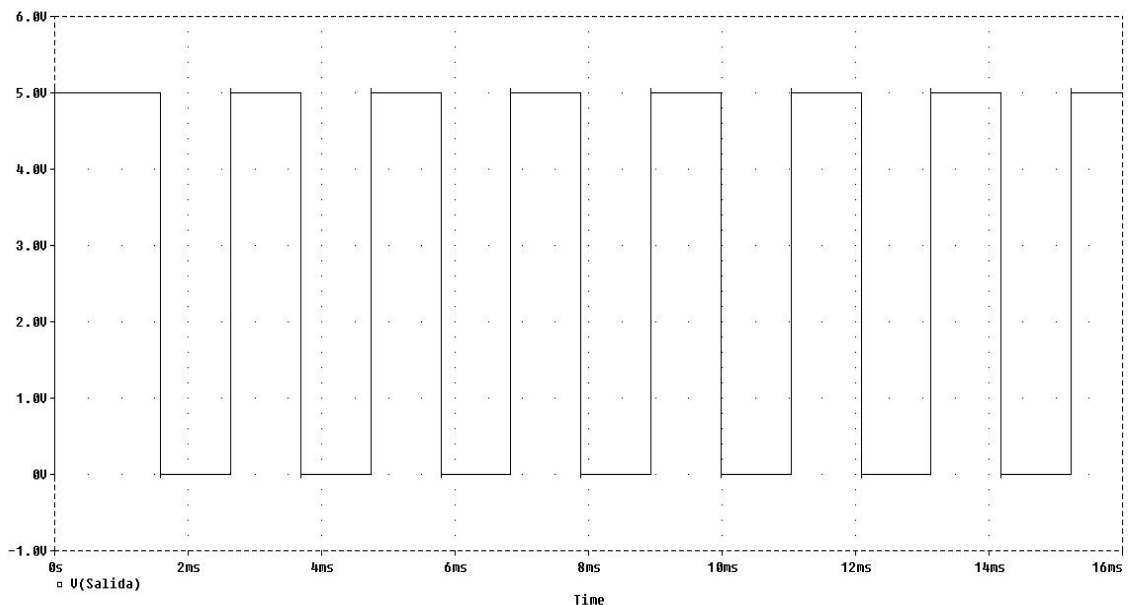
En teoría según el libro (pag 555), al ser iguales t_1 y t_2 y haber colocado el diodo D_1 para mejorar el máximo ciclo de uso, R_A y R_B deberían ser iguales, cosa que en el simulador no sucede, diferenciándose t_1 de t_2 y pudiéndose deber a la resistencia que ofrece el diodo D_1 . Realizando pequeños ajustes en el valor de R_B y comprobando la simulación se puede llegar a decir que con $R_B = 1136 \Omega$ se obtiene $t_1 \approx t_2 \approx 1 \text{ ms}$.

Una vez realizados los cálculos procedemos a la realización en el simulador del circuito para su posterior verificación, quedando como sigue:



Esquema del bloque 1: reloj

Una vez finalizado el esquema, colocamos una punta de verificación en la salida del circuito y configuramos el "print step" a 2ms y "final time" a 16ms en la opción transient y "flip flop initialization" a "All 0" del menú "Digital Setup" para preparar los parámetros de simulación. Una vez hecho esto, pulsamos "simulate" para realizar obtener el cronograma del circuito, mostrándose a continuación.



Cronograma del reloj 555

Ya que el bloque se ha verificado y cumple con las especificaciones del problema, pasamos al siguiente bloque funcional.

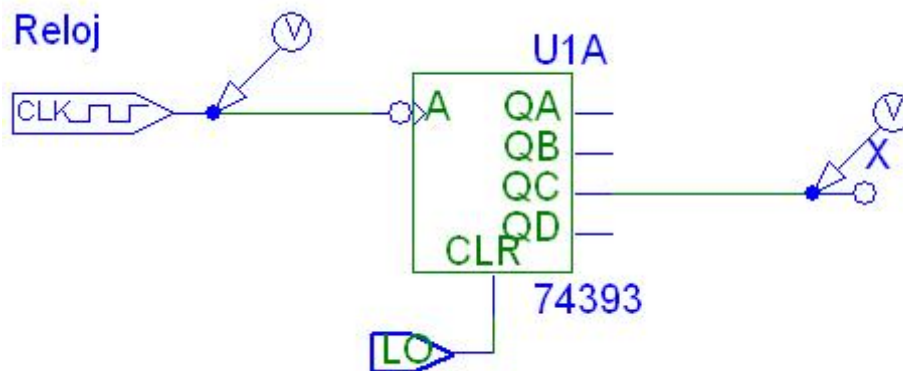
El segundo bloque es bastante sencillo, ya que para el generador de la señal de control tenemos que usar, según las recomendaciones de la guía de estudio, un comparador integrado SN74393, el cual tiene una entrada "A", una patilla de clear y cuatro salidas QD, QC, QB, QA.

La patilla de entrada "A" la conectaremos para realizar las pruebas a un digclock, más adelante cuando hagamos la integración de los bloques, las salida del reloj la conectaremos directamente aquí.

La patilla clear la tenemos que conectar a nivel bajo para activar el contador.

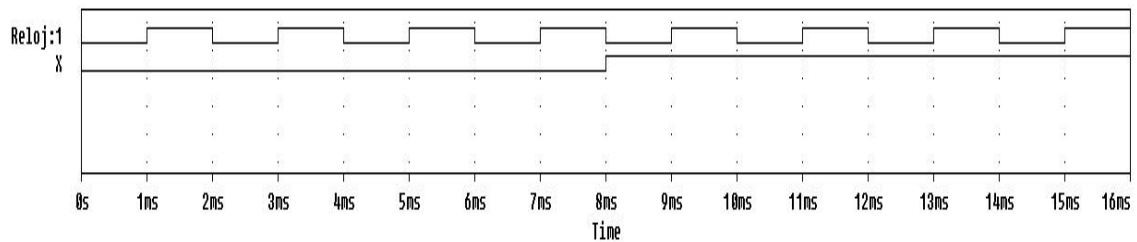
Las patillas de salida QD, QC, QB, y QA indican cuanto contar, por lo que nos interesa que sea 4 veces el periodo de la entrada de pulsos en A, en decimal 4 sería 0100 con lo que nuestra salida se encontraría en QC, según su datasheet.

El circuito del bloque quedaría de la siguiente manera:



Esquema del generador de señal de control

Antes de hacer la simulación hay que configurar los parámetros de la misma, colocamos una punta de verificación en la salida del circuito y otra en la entrada "A" y a continuación configuramos el "print step" a 1ms y "final time" a 16ms en la opción transient y "flip flop initialization" a "All 0" del menú "Digital Setup". Una vez hecho esto, pulsamos "simulate" para realizar obtener el cronograma del circuito, mostrándose a continuación.



Cronograma del bloque 2: generador de señal de control.

Como se puede apreciar en el intervalo 0s – 8ms se han producido los cuatro pulsos de la señal de entrada y la salida se ha mantenido a nivel bajo, teniendo la salida 4 veces el periodo de la entrada. En el intervalo 8ms – 16ms también se puede comprobar que esta la salida en alta mientras están los 4 pulsos de entrada, con lo que la verificación del bloque 2 es correcta.

El último bloque es el circuito secuencial en sí que tenemos que diseñar, siendo síncrono con la señal de reloj del bloque 1 y como variable de entrada la salida del bloque 2.

Para ello vamos a proceder a identificar la variables y los estados, y a partir de ellos y las especificaciones del problema en lenguaje natural, obtener el diagrama de estados y posteriormente las matrices de transición y funcional y las funciones de excitación de los biestables D y su posterior implementación.

Como se comentaba, del lenguaje natural hemos obtenido el siguiente diagrama de estados que muestra las transiciones entre estados dependiendo de su estado actual y la entrada X.

Del enunciado también se obtiene los 4 estados posibles para los pares Q1 Q0, S0 para 00, S1 para 01, S2 para 10 y S3 para 11.

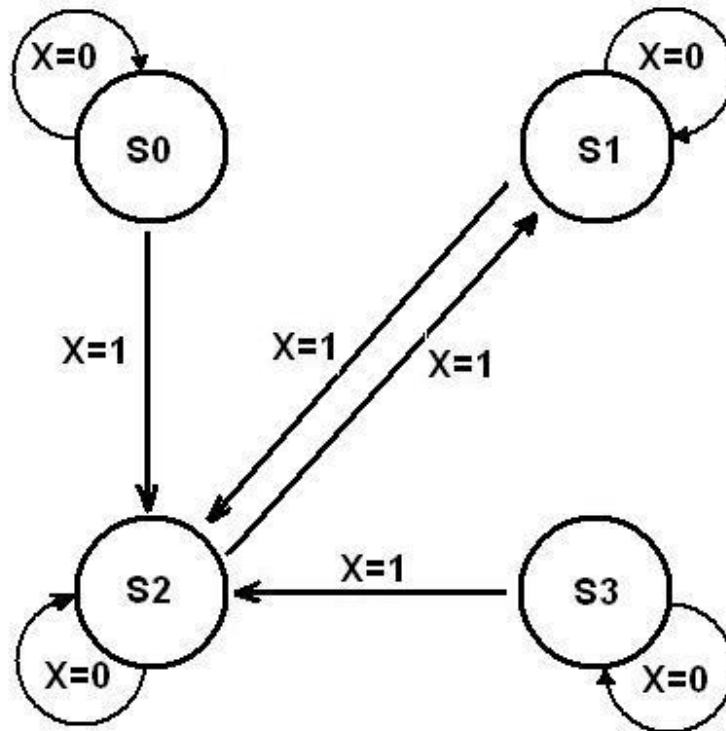


Diagrama de transición de estados

Una vez obtenida el diagrama de transición de estados seguimos las transiciones y las plasmamos en dos matrices de transición (una para x=0 y otra para x=1) marcamos en la tabla un 1 donde cruce el estado actual (primera columna) con la columna cuyo estado sea el siguiente. (Ej. de S1 -> S1, 3ª fila (S0) 3ª columna (S1)).

Obtenemos la matriz de transición para x=0

	S0	S1	S2	S3
X=0	Q1Q0 00	Q1Q0 01	Q1Q0 10	Q1Q0 11
S0	1	0	0	0
S1	0	1	0	0
S2	0	0	1	0
S3	0	0	0	1

Obtenemos también la matriz de transición para $x=1$

X=1	S0	S1	S2	S3
	Q1Q0 00	Q1Q0 01	Q1Q0 10	Q1Q0 11
S0	0	0	1	0
S1	0	0	1	0
S2	0	1	0	0
S3	0	0	1	0

Para obtener la matriz funcional combinamos las dos matrices de transición colocando los 1 de la matriz de transición $x=0$ como \bar{X} y los 1 de la matriz de transición $x=1$ como X. Quedando como se muestra a continuación.

	S0	S1	S2	S3
	Q1Q0 00	Q1Q0 01	Q1Q0 10	Q1Q0 11
S0	\bar{X}	0	X	0
S1	0	\bar{X}	X	0
S2	0	X	\bar{X}	0
S3	0	0	X	\bar{X}

Para calcular las funciones de excitación D_x debemos observar las columnas de la matriz funcional cuya Q_x sea 1 (para D1 -> Q1 columnas S2 y S3, para D0 -> Q0 columnas S1 y S3) y obtener por minterms multiplicando para cada fila de esas columnas los valores de Q0 y Q1 (complementados o no, según sean 0 ó 1) por la X ó \bar{X} de cada fila. Así se obtiene las siguientes dos funciones:

$$D1 = \bar{Q}1 \cdot \bar{Q}0 \cdot X + \bar{Q}1 \cdot Q0 \cdot X + Q1 \cdot \bar{Q}0 \cdot \bar{X} + Q1 \cdot Q0 \cdot X + Q1 \cdot Q0 \cdot \bar{X}$$

$$D0 = \bar{Q}1 \cdot Q0 \cdot \bar{X} + Q1 \cdot \bar{Q}0 \cdot X + Q1 \cdot Q0 \cdot \bar{X}$$

Pasamos a simplificar las dos funciones mediante mapas de Karnaugh para hacerlas mínimas.

Simplificación de $D1$

$\begin{matrix} Q1 \\ X \backslash Q0 \end{matrix}$	00	01	11	10
0	0	0	1	1
1	1	1	1	0

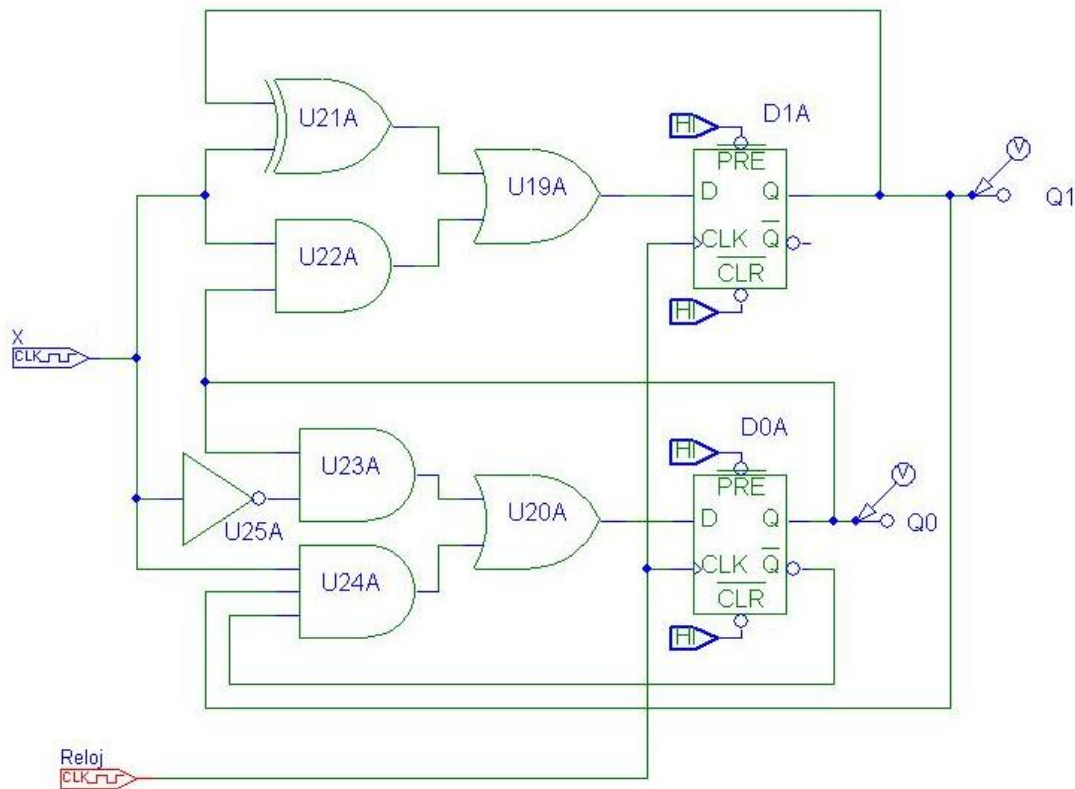
$$D1 = \overline{Q1} \cdot X + Q1 \cdot \overline{X} + Q0 \cdot X = (Q1 \oplus X) + Q0 \cdot X$$

Simplificación de $D0$

$\begin{matrix} Q1 \\ X \backslash Q0 \end{matrix}$	00	01	11	10
0	0	1	1	0
1	0	0	0	1

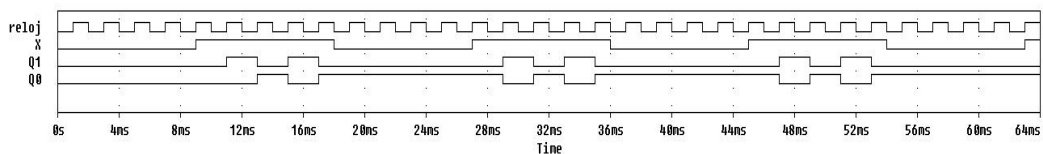
$$D0 = Q0 \cdot \overline{X} + Q1 \cdot \overline{Q0} \cdot X$$

Con las funciones ya minimizadas, el siguiente paso es obtener el esquema con dos biestables y las puertas necesarias del autómata finito para su comprobación. El esquema resultante es el que se muestra a continuación.



Una vez finalizado el esquema, colocamos una punta de verificación en las salidas de los biestables del circuito y en los dos digclock que harán las funciones de x (a 7ms) y el reloj (a 1ms) del sistema para comprobaciones.

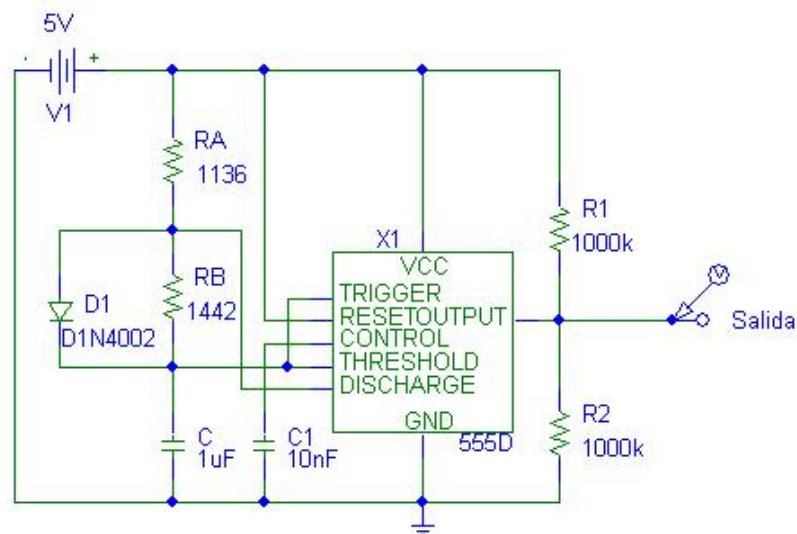
Configuramos el "print step" a 2ms y "final time" a 64ms en la opción transient y "flip flop initialization" a "All 0" del menú "Digital Setup" para preparar los parámetros de simulación. El cronograma de la simulación es el siguiente:



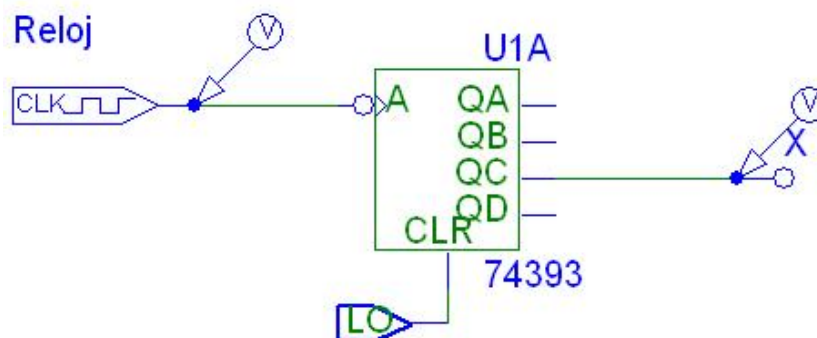
Se comprueba que cuando X toma un valor de 0 las variables Q1 y Q0 del autómata no cambia y cuando X toma el valor 1 cambia hasta que se estabiliza en una alternancia entre 10 y 01 hasta el siguiente cambio de X. Solo queda formar el sistema completo conectando los tres bloques funcionales.

7: Esquemas capturados del simulador, primero presente los esquemas de cada uno de los bloques funcionales por separado y después presente el del circuito completo.

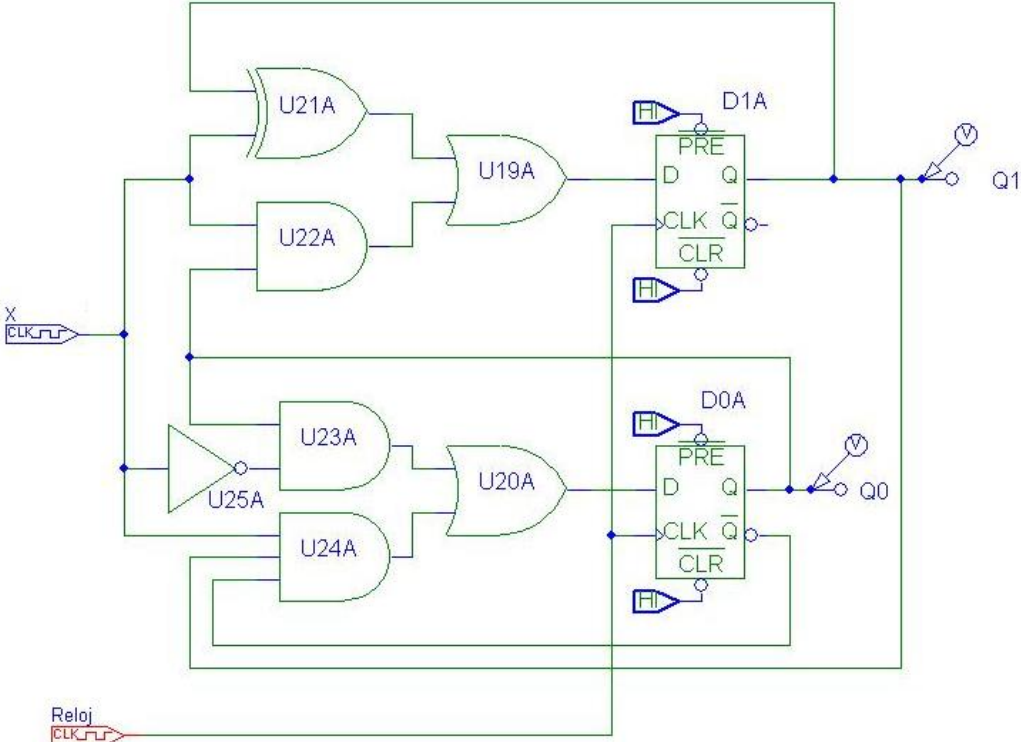
El esquema del bloque del reloj queda de la siguiente manera:



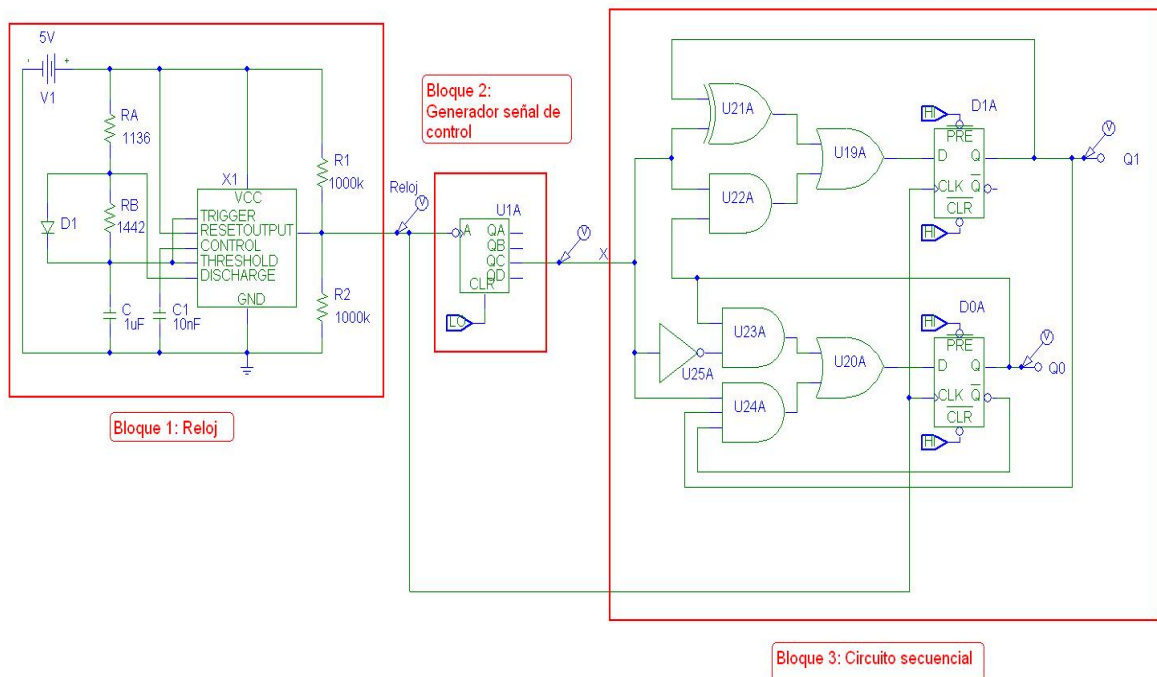
El esquema del generador de la señal de control es el que se muestra en la siguiente ilustración :



El último bloque tiene el siguiente esquema:



El circuito completo en el que se integran todos los bloques funcionales es el siguiente:



8: Descripción de los parámetros de los componentes utilizados.

Alimentación 5V.

Resistencias:

$$R1=R2=1000K\Omega$$

$$Ra=1136\Omega$$

$$Rb=1442\Omega$$

Condensadores:

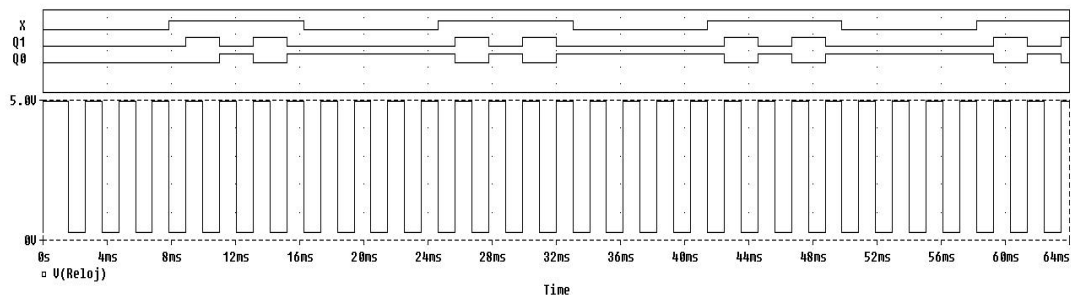
$$C=1\mu F$$

$$C1=10nF$$

Configuramos el dentro de "Transient" "print step" a 2ms y "final time" a 64ms y "flip flop initialization" a "All 0" del menú "Digital Setup" para preparar los parámetros de simulación.

9: Cronograma de todas las señales (entrada, control y salida).

El cronograma de la simulación del circuito completo muestra las señales del reloj, la señal X del generador de señal y las dos señales Q1 y Q0 del circuito secuencial.



10: Explicación del funcionamiento y verificación de que el circuito funciona de acuerdo con las especificaciones del diseño.

El funcionamiento del sistema es el siguiente, el reloj genera una señal cuadrada de aproximadamente 2ms de periodo que atraviesa un generador de señal que modifica esta hasta hacer un periodo por cada cuatro de entrada. Esta última señal es la que marca los cambios en el circuito secuencial.

Con la señal X en baja, las señales Q1 y Q0 se mantienen en sus valores actuales y no varían como se puede apreciar en el cronograma de la simulación. Cuando la señal X cambia a un estado de alta, se tiene que producir la transición de estado de Q1 y Q0 a 10 o 01 y alternar entre ambas como indica el enunciado del problema, se observa claramente como en esos periodos cambian las señales Q1 y Q0 a los valores correctos.

11: Explicación de los problemas/dificultades encontrados y explicación de la forma y el medio por el que se han resuelto.

He encontrado algunas dificultades con el calculo de las resistencias, en el libro de texto (página 555) se indica que al haber colocado un diodo en paralelo con R_B las ecuaciones de $t_1 \approx 0,69R_A \cdot C$, $t_2 \approx 0,69R_B \cdot C$, lo que a la restricción $t_1=t_2$ hace que las dos resistencias sean iguales, lo cual no parece ser cierto en el simulador ya que t_1 dista mucho de ser igual a t_2 . La solución ha sido modificar el valor de R_B poco ha poco hasta dar con los valores aproximados como se indicó en el foro.

Otro problema que he encontrado ha sido la imposibilidad de probar todos los estados del autómata. Por una lado las especificaciones del problema generan dos nodos a los que no se puede acceder desde otros estados, lo cual fuerza a que sean estados iniciales y sólo se pueden dar en el momento de inicio de la simulación y no simultáneamente en la misma simulación. En el caso del estado S0 es fácil ya que colocando en la configuración del simulador a que todos los biestables partan del valor 0 se puede probar sin problema alguno y en el caso del estado S3 habría que inicializar los biestables a 1.