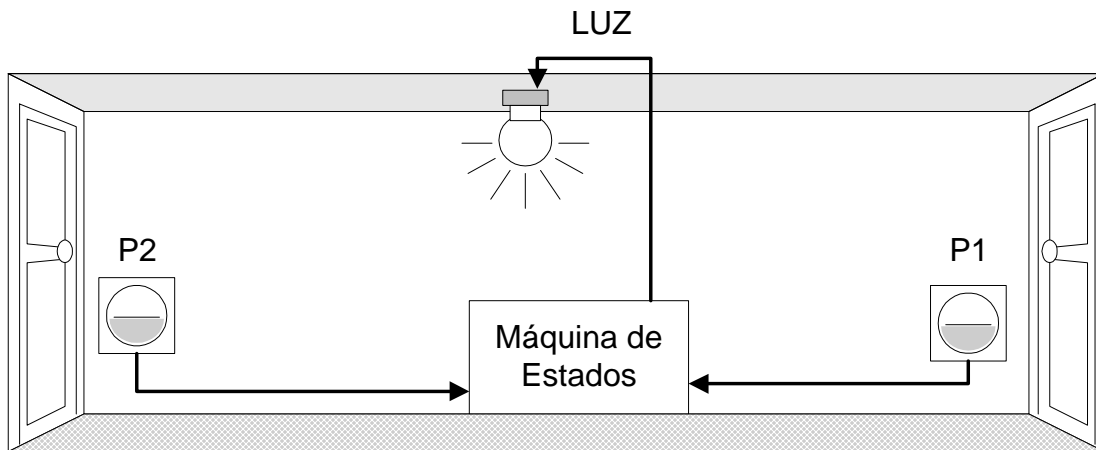


## Problema resuelto de Máquinas de estado.

Se desea diseñar un sistema de iluminación para un pasillo, de manera que cumpla con las siguientes especificaciones:

- El diseño estará basado en una máquina de estados síncrona.
- El pasillo dispone de dos pulsadores, uno al lado de cada puerta, de manera que se pueda encender y apagar la luz desde cada extremo. Cada pulsador produce un '1' lógico mientras está pulsado, y un '0' lógico cuando no lo está.
- Se desea que, cada vez que se pulse cualquier pulsador, la luz cambie de estado: si está apagada se debe encender, y viceversa.
- Se debe tener en cuenta el caso en el que, mientras se pulsa un interruptor, se pulse el otro. Por ejemplo, si estando apagada la luz, alguien pulsa P1 se enciende la luz. Pero si mientras está pulsado P1 alguien pulsa P2, entonces se apagará nuevamente la luz.
- Sin embargo, se puede considerar que la frecuencia del reloj es lo suficientemente alta como para que sea imposible un cambio simultáneo de los dos pulsadores (en el mismo ciclo de reloj).

En la siguiente figura se muestra un esquema de la instalación:



Como se observa, la máquina de estados tendrá dos entradas, P1 y P2, y una única salida, LUZ, que hará que se la luz se encienda mientras valga '1', y que permanezca apagada cuando valga '0'.

Se pide:

1. Diseño de la máquina de estados descrita. Implementación de la tabla de transición de estados. **(3 puntos)**
2. Reducción, si es posible de la máquina. Obtención de las ecuaciones de excitación de los biestables que compongan la máquina, así como de la ecuación de salida. Realizar la máquina con biestables tipo D, y puertas de dos entradas. **(4 puntos)**
3. Dibujar el esquema de la máquina de estados, incluyendo la salida. **(1 punto)**
4. Calcular el camino crítico y la máxima frecuencia de trabajo, suponiendo que los retrasos a considerar son:  $t_{PD}=20\text{ns}$ ;  $t_{SETUP}D=15\text{ns}$ ;  $t_{PROP}D=25\text{ns}$ ; **(2 puntos)**

**Basado en el Examen de Septiembre 2001. TIEMPO: 1h 30min**

## Solución al problema

Lo primero que me piden es la implementación de la máquina de estados. Veamos cómo resulta dicha máquina:

ESTADO	P1-P2				LUZ
	0 0	0 1	1 1	1 0	
0: OFF.	(0)	1	--	2	0
1: P2. ON	4	(1)	3	--	1
2: P1. ON	4	--	3	(2)	1
3: P1P2:OFF	--	5	(3)	6	0
4: ON	(4)	5	--	6	1
5: P2. OFF	0	(5)	7	--	0
6: P1. OFF	0	--	7	(6)	0
7: P1P2:ON	--	1	(7)	2	1

A continuación, vamos a tratar de explicar la tabla de transición descrita:

**Estado 0)** OFF. Luces apagadas, y pulsadores en reposo. Si se pulsa P2 voy al estado 1. Si se pulsa P1 voy al estado 2. No se puede pasar de 00 a 11 en un ciclo por lo que 11 no es posible en este estado.

**Estado 1)** P2.ON: se pulsó P2, para encender la luz. Mientras permanezca pulsado, sigo en este estado. Si se suelta (P2=0) voy al estado 4, y si se pulsa P1 voy al estado 3. La luz se enciende.

**Estado 2)** P1.ON Se pulsó P1 para encender la luz. Mientras permanezca pulsado, sigo en este estado. Si se suelta (P1=0) voy al estado 4, y si se pulsa P2 voy al estado 3. La luz se enciende.

**Estado 3)** P1P2.OFF. Se han pulsado P1 y P2, desde la posición de reposo, por lo cual la luz debe apagarse. Si se suelta algún pulsador, iré a los estados 5 ó 6, dependiendo de cuál quede pulsado. No es posible la entrada 00.

**Estado 4)** ON. Estado estable de luz encendida y pulsadores en reposo. Permaneceré así hasta que se pulse P1 o P2, en cuyo caso iré a 6 ó 5 respectivamente.

**Estado 5)** P2.OFF. Se ha pulsado P2, y la luz estaba encendida: apagar la luz. Al igual que pasaba en el estado 1, pueden pulsar P1 o soltar P2. Si pulsan P1, con ambos pulsados iré al estado 7, mientras que si sueltan P2, iré al estado 0 que representa la luz apagada y los pulsadores en reposo.

**Estado 6)** P1. OFF. Se ha pulsado P1, y la luz estaba encendida: apagar la luz. Al igual que pasaba en el estado 2, pueden pulsar P2 o soltar P1. Si pulsan P2, con ambos pulsados iré al estado 7, mientras que si sueltan P1, iré al estado 0 que representa la luz apagada y los pulsadores en reposo.

**Estado 7)** P1P2.ON: se han pulsado P1 y P2 estando la luz encendida, por lo que ha debido apagarse y volverse a encender. Al igual que pasaba con el estado 3, si se suelta alguno de los pulsadores iré al estado correspondiente, 1 ó 2, según cuál quede pulsado.

Ahora trataré de simplificar la tabla de transición de estados. Basta observar que los estados 1 y 2 tienen la misma salida, y que sus transiciones son compatibles. Lo mismo pasa con 5 y 6, pudiendo quedar entonces la tabla reducida a:

ESTADO	P1-P2				LUZ
	0 0	0 1	1 1	1 0	
0: OFF.	(0)	1	--	1	0
1: P2 o P2 ON	3	(1)	2	(1)	1
2: P1P2:OFF	--	4	(2)	4	0
3: ON	(3)	4	--	4	1
4: P2 o P1. OFF	0	(4)	5	(4)	0
5: P1P2:ON	--	1	(5)	1	1

Realizamos la asignación de estados haciendo corresponder a cada estado su código binario. De esta manera, es fácil observar que la salida, LUZ, será igual a Q0, el bit menos significativo. Con ello, podemos escribir ya la tabla de excitación de la máquina:

Q2-Q1-Q0	P1-P2	D2-D1-D0
0 0 0	0 0	0 0 0
0 0 0	0 1	0 0 1
0 0 0	1 1	-----
0 0 0	1 0	0 0 1
0 0 1	0 0	0 1 1
0 0 1	0 1	0 0 1
0 0 1	1 1	0 1 0
0 0 1	1 0	0 0 1
0 1 0	0 0	-----
0 1 0	0 1	1 0 0
0 1 0	1 1	0 1 0
0 1 0	1 0	1 0 0
0 1 1	0 0	0 1 1
0 1 1	0 1	1 0 0
0 1 1	1 1	-----
0 1 1	1 0	1 0 0
1 0 0	0 0	0 0 0
1 0 0	0 1	1 0 0
1 0 0	1 1	1 0 1
1 0 0	1 0	1 0 0
1 0 1	0 0	-----
1 0 1	0 1	0 0 1
1 0 1	1 1	1 0 1
1 0 1	1 0	0 0 1

A partir de esta tabla ya es posible realizar las tablas de Karnaugh para cada uno de los biestables. Dado que tenemos cinco variables, separamos en dos tablas de 4 variables, usando Q2 para separar ambas tablas:

Q2=0				
Q1Q0	P1-P2			
	00	01	11	10
00	0	0	--	0
01	0	0	0	0
11	0	1	--	1
10	--	1	0	1

Q2

Q2=1				
Q1Q0	P1-P2			
	00	01	11	10
00	0	1	1	1
01	--	0	1	0
11	--	--	--	--
10	--	--	--	--

$$Q2 = Q1 * (P1 \oplus P2) + P1 * P2 * Q2 + Q2 * /Q0 * (P1 + P2)$$

Q2=0				
Q1Q0	P1-P2			
	00	01	11	10
00	0	0	--	0
01	1	0	1	0
11	1	0	--	0
10	--	0	1	0

Q1

Q2=1				
Q1Q0	P1-P2			
	00	01	11	10
00	0	0	0	0
01	--	0	0	0
11	--	--	--	--
10	--	--	--	--

$$Q1 = /Q2 * P1 * P2 + /P1 * /P2 * Q0$$

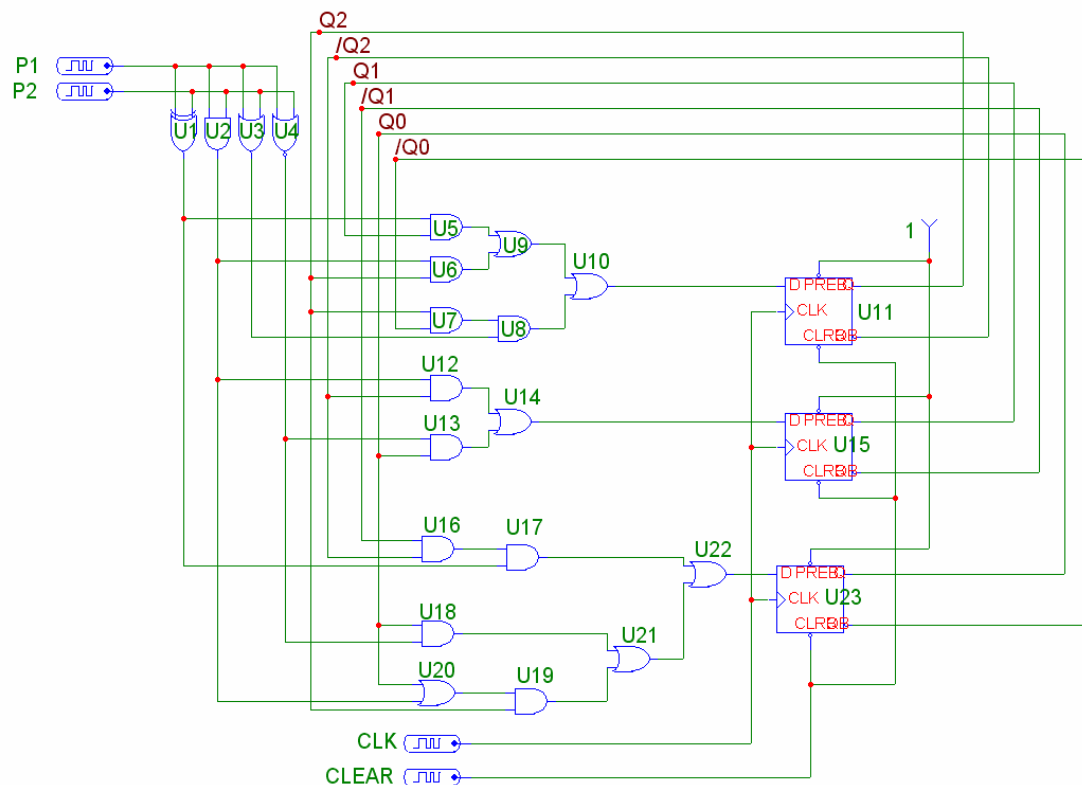
Q2=0				
Q1Q0	P1-P2			
	00	01	11	10
00	0	1	--	1
01	1	1	0	1
11	1	0	--	0
10	--	0	0	0

Q0

Q2=1				
Q1Q0	P1-P2			
	00	01	11	10
00	0	0	1	0
01	--	1	1	1
11	--	--	--	--
10	--	--	--	--

$$Q0 = /Q1 * /Q2 * (P1 \oplus P2) + /P1 * /P2 * Q0 + Q2 * (Q0 + P1 * P2)$$

Expresado de esta forma, se observa que las entradas siempre aparecen agrupadas de forma “simétrica”, lo cual resulta bastante explicable: no hay ninguna diferencia entre una y otra entrada, de manera que se debe poder cambiar P1 por P2 sin que cambie nada. Se observa en efecto que siempre aparecen  $P1 * P2$ ,  $/P1 * /P2$ ,  $P1 + P2$  ó  $P1 \oplus P2$ . Esto sugiere usar primero estas cuatro puertas, y tras esto generar ya las ecuaciones definitivas. En la imagen se observa el circuito ya resuelto:



Para calcular el camino crítico, basta observar cuál es el más largo que recorre una señal desde la salida de un biestable hasta la entrada de otro biestable. Es fácil ver que en este caso será el camino formado por las puertas U20, U19, U21 y U22, que la señal Q0 tiene que atravesar desde la salida a la entrada de su propio biestable.

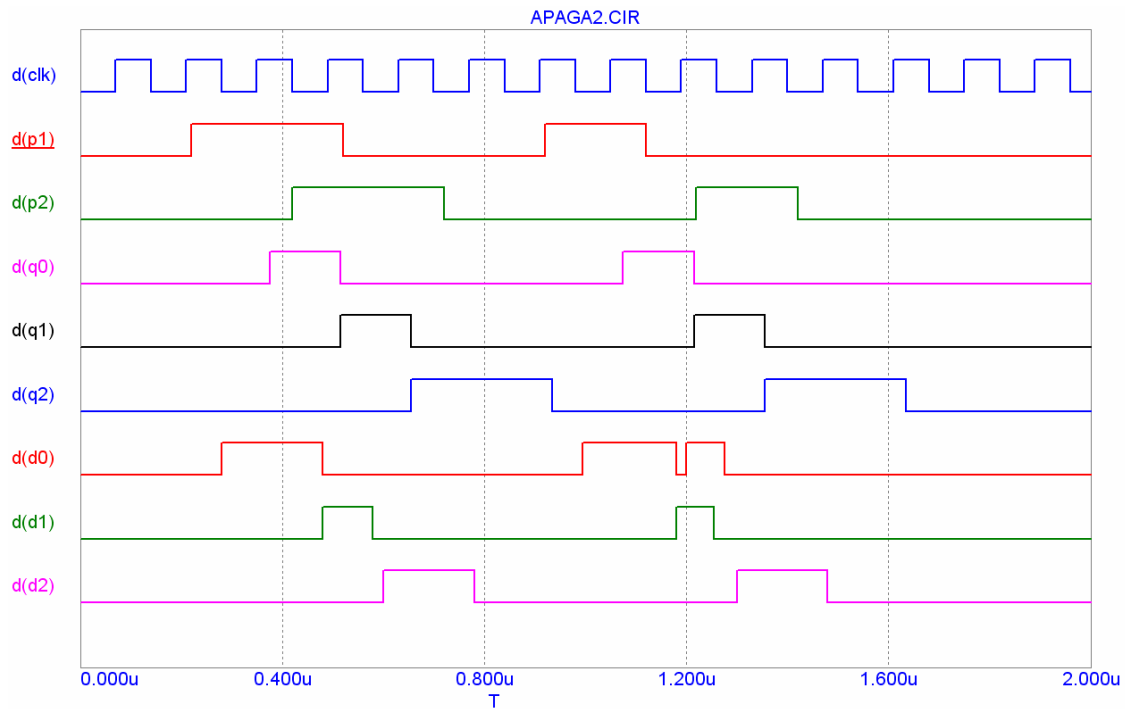
Por tanto, el retraso del camino crítico sería:

$$T_{\text{CRIT}} = 4 \cdot t_D + t_{\text{SETUP}} = 4 \cdot 20\text{ns} + 15\text{ns} + 25\text{ns} = 120\text{ns}$$

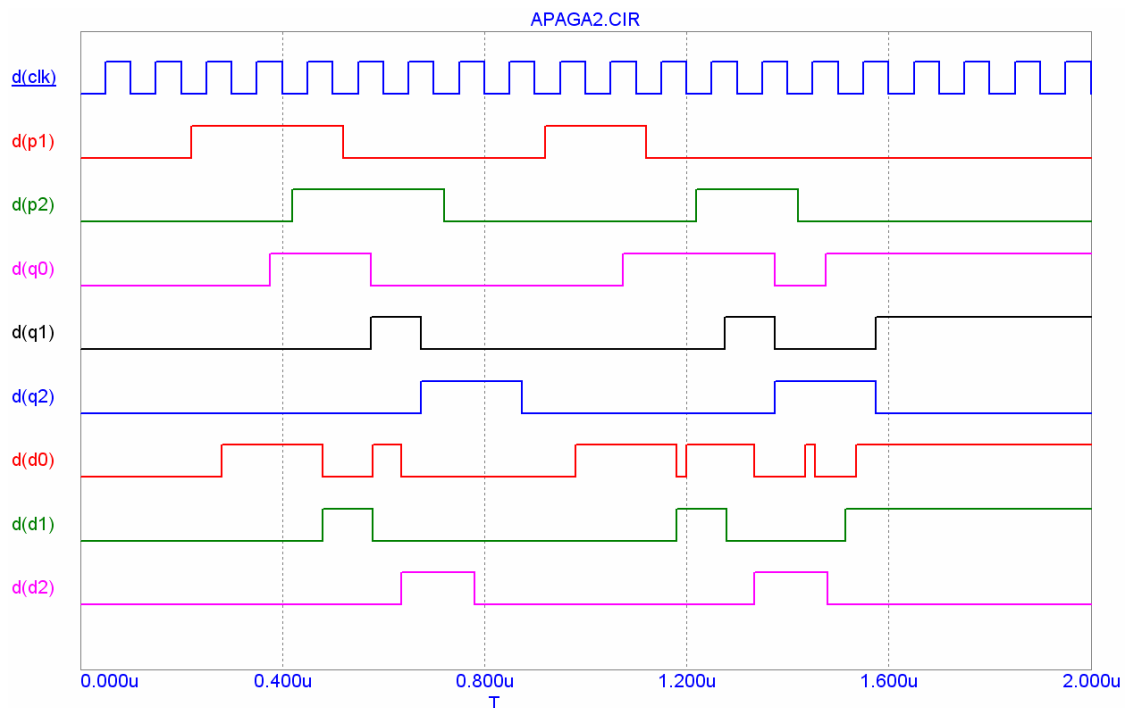
Por lo que la máxima frecuencia de funcionamiento del circuito será de:

$$F_{\text{MAX}} = 1/T_{\text{CRIT}} = 8.3\text{MHz}$$

Por último, vamos a presentar unas simulaciones del circuito. En la primera simulación, vemos el funcionamiento del circuito con una frecuencia de 140ns, comprobando que funciona correctamente, dado que la línea Q0 (que es la salida), se apaga y se enciende correctamente. Aun así, se observa que la entrada D0 presenta glitches provocados por el retraso de la señal, que aunque no causan problemas todavía, sí generan una cierta intranquilidad. Sin embargo, en la siguiente gráfica se observa que al subir a 10MHz (100ns) la frecuencia, el circuito empieza a funcionar mal, y la salida no responde como debiera.



Simulación 1: Circuito con 140ns de periodo de reloj.

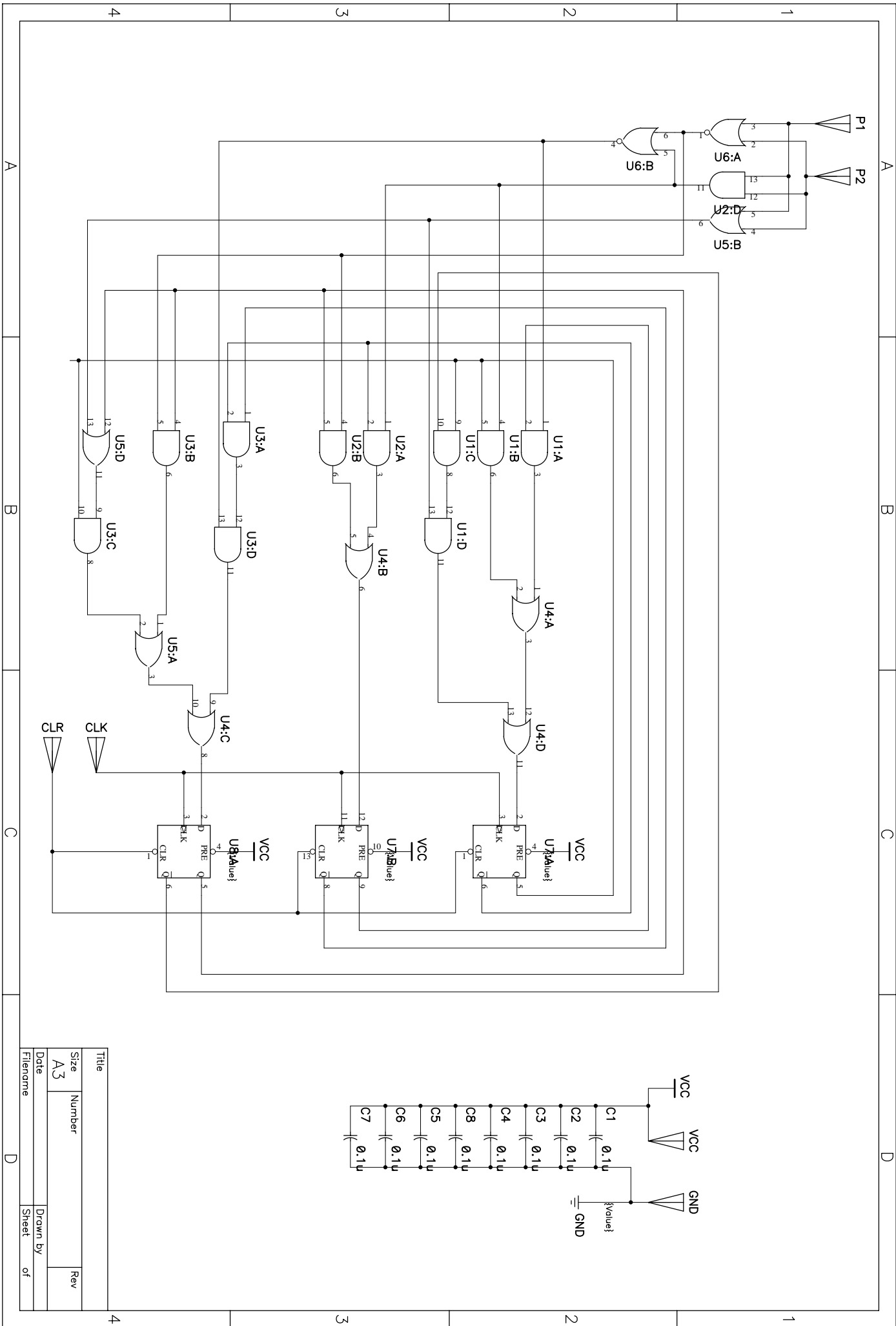


Simulación 2: Circuito con 100ns de periodo de reloj.

Para implementar este circuito con componentes discretos, sería conveniente hacer alguna transformación, para evitar tener que usar puertas AND, OR, NAND, NOR y XOR. Para ello, el circuito de las entradas se puede modificar, sabiendo que la puerta Xor la puedo expresar como:

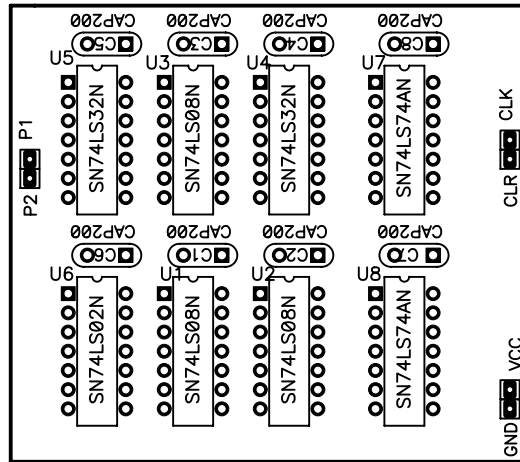
$$A \oplus B = ((A \cdot B) + (A + B))$$

esto es, con una AND y dos NOR. Por tanto, el circuito se puede diseñar, usando un programa de diseño de circuitos, como se muestra en el plano adjunto. Utilizando una herramienta de rutado automático de placas, en pocos minutos tenemos el diseño realizado, como se muestra en la última página.

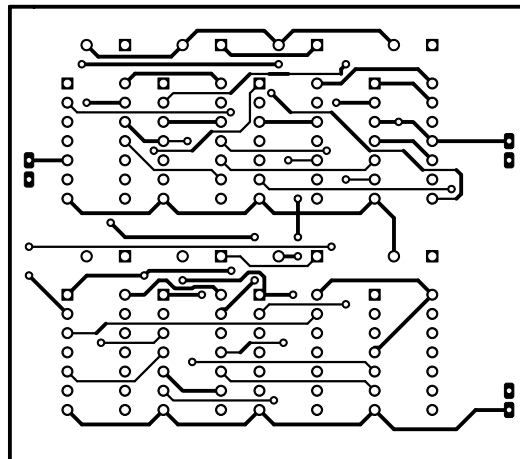


Title		
Size	Number	Rev
A3		
Date	Drawn by	
Filename	Sheet	of

### Serigrafía de Componentes:



### Pistas en la cara superior (componentes)



### Pistas en la cara inferior (soldaduras)

