

Nombre:.....

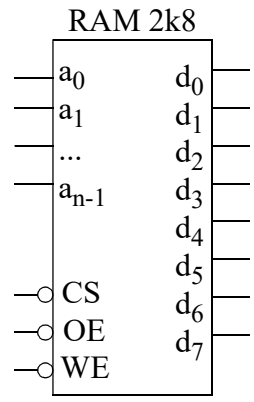
--	--	--	--

TEORÍA (Cada pregunta vale 1 punto)

- 1.- Defina clara y brevemente los siguientes conceptos:
 - a) Condensador de desacoplo.
 - b) Fan-out: Definición y expresiones.
 - c) Rangos de tensión.
 - d) Encapsulado: qué es, para que sirve y algunos tipos.

- 2.- Explique qué es un codificador de prioridad y su funcionamiento. Dé el símbolo lógico y la tabla de funcionamiento de un codificador 4 a 2 con dos entradas de habilitación y salidas GS (Group Select) y EO (Enable Output). Comente todas las posibles situaciones conflictivas que se pueden dar en un codificador sin salidas GS y EO.

- 3.- Suponiendo que dispone de chips de RAM de 2Kx8, con el funcionamiento que se muestra en la tabla adjunta, obtenga un módulo de memoria RAM de 8Kx8 con las mismas señales de control. Dimensione y justifique el bus de direcciones de la RAM 2K8. Si lo necesita, puede utilizar elementos adicionales como puertas lógicas y/o decodificadores.



CS#	OE#	WE#	$D_{0..m-1}$	Modo
1	X	X	HiZ	Reposo
0	1	1	HiZ	Sal. deshabilitada
0	0	1	Salida	Lectura
0	-	0	Entrada	Escritura

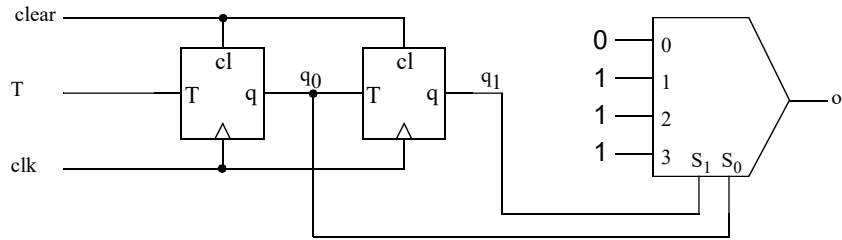
- 4.- Para el CS3, escriba las micro-operaciones para ejecutar las instrucciones LD Rd, (Rb) y LDI Rd, dir (siendo dir un número cualquiera de 8 bits). Debe dar tanto transferencias de registros como las señales de control que se activan.

--	--

PROBLEMAS (Cada pregunta vale 3 puntos)

1.- Analice el circuito de la figura:

- a) Suponiendo ideales todos los componentes, obtenga la tabla de estados/salida (sin tener en cuenta la entrada de clear asíncrona) y asigne los estados de la forma: $q_1q_0 = "00"$ estado A, $q_1q_0 = "01"$ estado B, $q_1q_0 = "10"$ estado C, $q_1q_0 = "11"$ estado D. A partir de la tabla de estados obtenga la evolución de la salida para las entradas que se proponen.
- b) Suponiendo que todos los dispositivos tienen un retraso Δ , dibuje la evolución de la salida y de las señales intermedias.
- c) Comente qué ocurriría si el tiempo de retraso Δ fuese mayor que el periodo de reloj menos el tiempo de setup de los biestables.



SOLUCIÓN

a) Análisis para la obtención de la tabla de estados.

El proceso de análisis es el siguiente: Obtención de las ecuaciones de las entradas de los biestables, Tabla de excitación/salida, tabla de transición/salida y tabla de estados.

$q_1q_0 \backslash T$	0	1	
00	00	01	0
01	10	11	1
11	10	11	1
10	00	01	1
	T_1T_0	O_1	

Tabla de excitación/salida

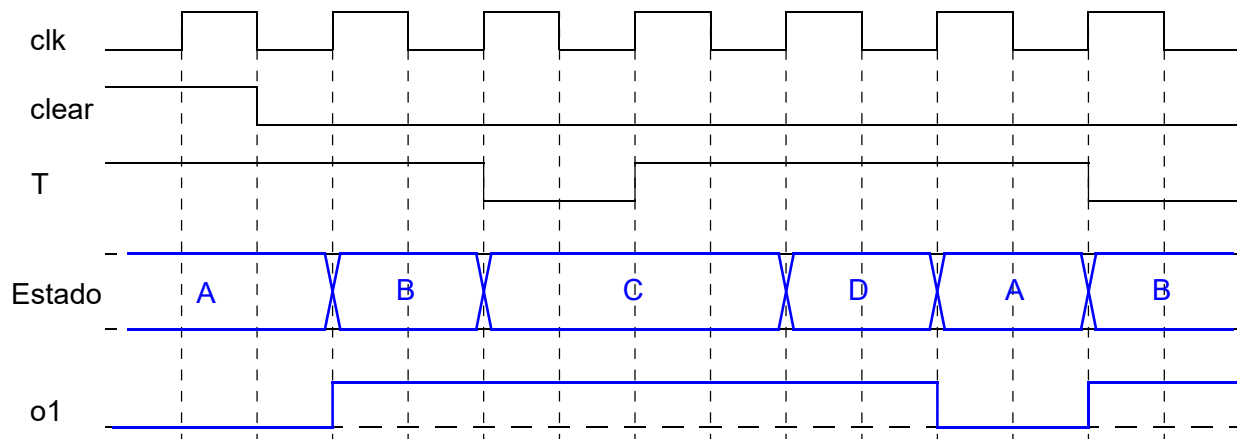
$q_1q_0 \backslash T$	0	1	
00	00	01	0
01	11	10	1
11	01	00	1
10	10	11	1
	Q_1Q_0	O_1	

Tabla de transición/salida

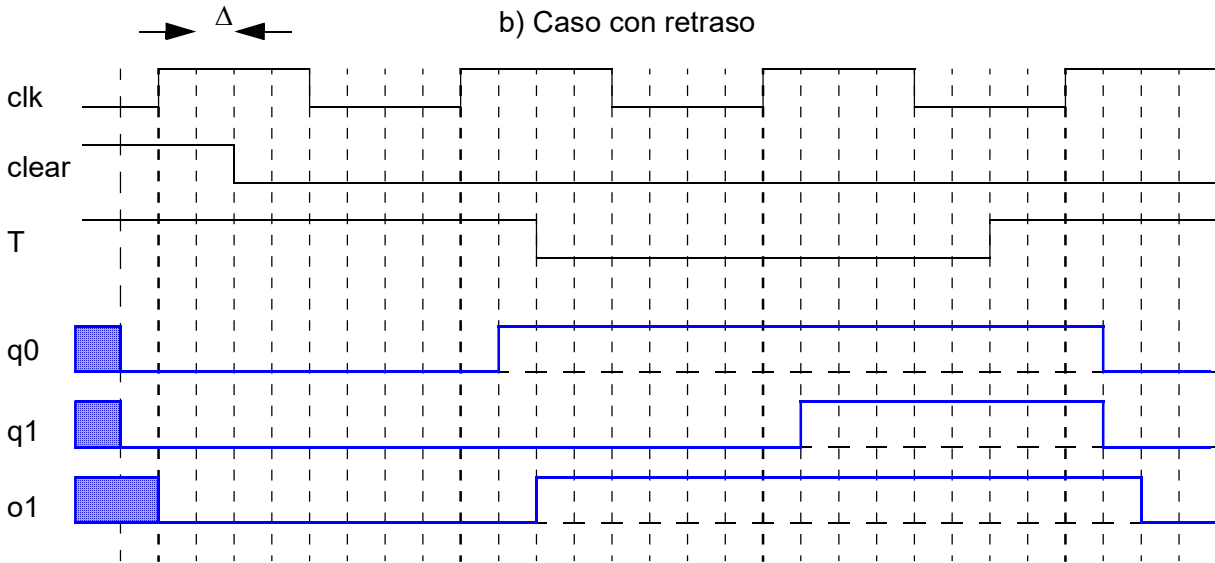
$s \backslash T$	0	1	
A	A	B	0
B	D	C	1
D	B	A	1
C	C	D	1
	NS	O_1	

Tabla de estados/salida

Caso ideal:



Caso con retraso:



- 2.- El protocolo WS2812 permite programar el color de un LED RGB en base a 24 bits que se cargan en el led de forma serie. Para programar en el led un '1' lógico hay que poner su entrada a '1' durante 800 ns y después a '0' durante 450 ns. Para programar un '0' lógico hay que poner su entrada a '1' durante 450 ns y después a '0' durante 800 ns. Gráficamente:



Se desea diseñar un circuito que controle la programación de los 24 bits de un LED RGB. Para ello se dispone de una unidad de datos con los siguientes elementos:

- Un registro paralelo-serie (A) de 24 bits, que contiene desde el inicio los 24 bits a programar en el LED. Este registro tiene una entrada SHL que cuando esté a '1' producirá un desplazamiento a la izquierda de este registro. La salida del bit más significativo se llama A23.
 - Un contador descendente, C1, módulo 256 con entrada de carga en paralelo síncrona activa en alta (LD1) y salida de Borrow (BW1).
 - Un contador descendente, C2, módulo 64, con entrada de carga en paralelo síncrona activa en alta (LD2) y salida de Borrow (BW2).
- a) Si en la unidad de datos necesita algún componente más, indíquelo y describa su funcionalidad a nivel RT.
- b) Dibuje la carta ASM de la unidad de datos y de la unidad de control del controlador que genere la señal para programar los 24 bits del LED RGB de la forma que se ha mostrado en la figura anterior. Los valores a programar están en el registro A. El controlador cuenta con una entrada Xs que indica el inicio de una programación. Una vez terminada la programación debe poner una salida FIN a uno. La salida con la señal para programar el led ha de llamarse "prog_led". La señal de reloj tiene una frecuencia de 100 MHz.

SOLUCIÓN

Hay distintas alternativas para solucionar este problema. La más sencilla consiste en utilizar el contador C1 para contar tanto el tiempo que tiene que estar a '1' prog_led como el tiempo que tiene que estar a '0'. Como el reloj es de 100 MHz, su periodo es de 10 ns. Esto significa que 800 ns son 80 ciclos de reloj y 450 ns son 45 ciclos de reloj. Como a C1 vamos a tener que cargar 80 ó 45, vamos a necesitar un multiplexor 2a1 (uno por cada bit). Cuando su entrada de selección (s) esté a '1' conectará la entrada con el valor 80 a la salida. Cuando esté a '0' cargará el valor 45. El contador C2 lo vamos a utilizar para contar el número de bits (24). Vamos a necesitar modificar ligeramente la

descripción porque se necesita que tenga una entrada de inhibición. Proponemos una entrada INH2 activa en baja.

La descripción del funcionamiento va a ser el siguiente:

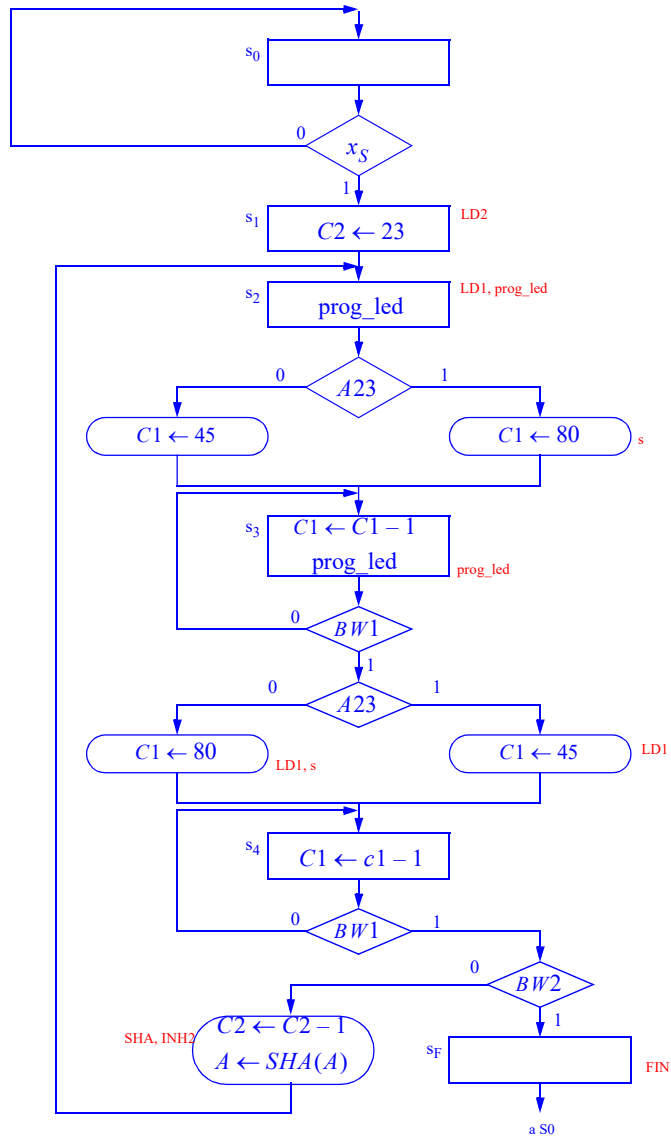
Cuando se activa la entrada Xs pasamos a un estado en el que cargamos C2 a 23.

Pasamos a otro estado en el que ponemos prog_led a '1'. Preguntamos por A23, si vale '0' cargamos C1 con 45 y si vale '0' cargamos C1 con 80.

Pasamos a otro estado en el que vamos decrementando C1. Cuando C1 valga cero (se activa BW1), volvemos a preguntar cuanto vale A23. Si vale '0' cargamos C1 con 80 y si vale '1' lo cargamos con 45.

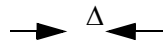
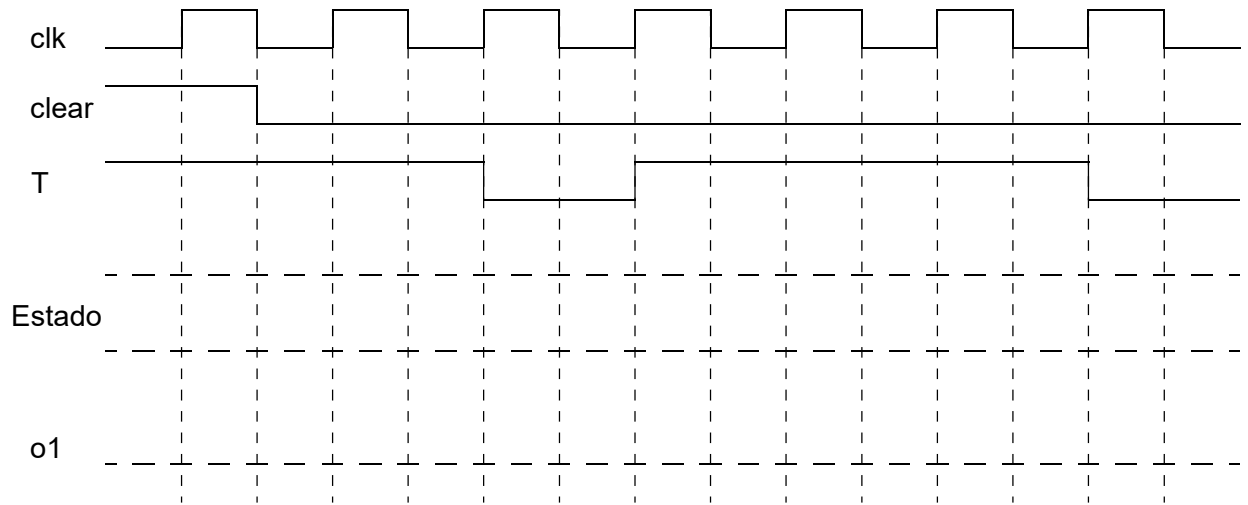
Pasamos a otro estado en el que ponemos prog_led a '0' y vamos decrementando C1. Cuando sea cero preguntamos por C2. Si no es cero (BW2 = '0') decrementamos C2 y desplazamos A. Si C2 es cero entonces se ha terminado.

La carta ASM es:



Problema 1

a) Caso ideal



b) Caso con retraso

