

Apellidos y Nombre:

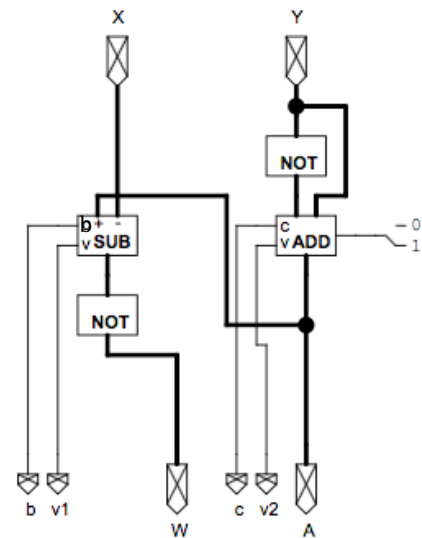
Grupo: DNI:

Examen E2 (temas 4, 5, 6 i 7)

Duración: 1 hora 45 minutos. La solución de cada ejercicio se tiene que escribir en el espacio reservado para ello en el propio enunciado. No podéis utilizar calculadora, apuntes, etc. La solución del examen se publicará en Atenea mañana por la tarde y las notas antes de una semana.

1. (Objetivos 2.4, 4.9, 5.6 y 5.11) (1.5 puntos)

Dado el esquema del CLC a bloques de la derecha, completad la siguiente tabla que indica el valor de las salidas del circuito para cada uno de los cuatro casos concretos de valores de las entradas (un caso por fila). As y Au es el valor que representa el vector de bits A según se interprete como número natural representado en binario o entero representado en complemento a dos, respectivamente:

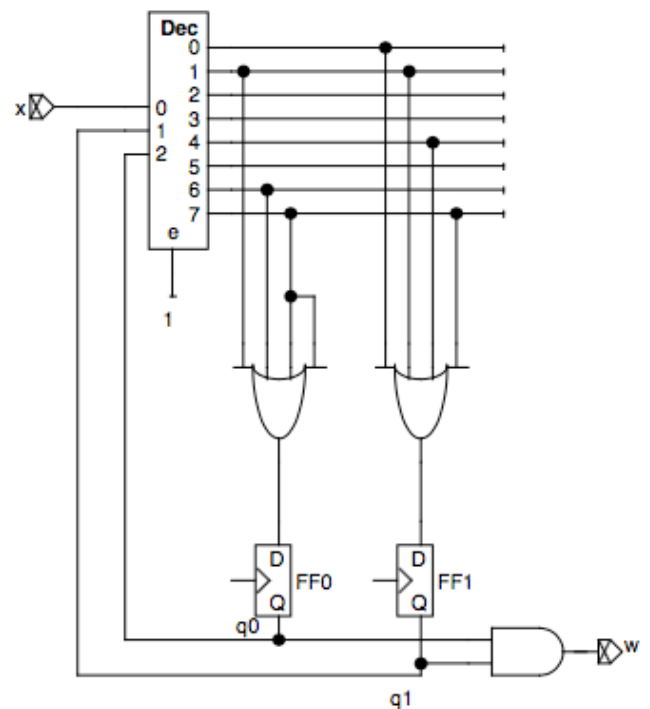


X	Y	b	v1	W	A	c	v2
0000 0000	1111 1111						
1000 0001	1000 0001						
1000 0000	0111 1111						
1111 1111	1000 0000						

2. (objetivos 6.14 6.10, 6.13 y 6.15) (3.5 puntos)

Dado el circuito siguiente

a. Construir el grafo de estados asumiendo que el estado inicial es el q1q0=00. **(1.5 puntos)**



Apellidos y Nombre:

.....Grupo:.....DNI:.....

b. Completad el cronograma simplificado. (0.5 puntos)

Estado	00	10									
x	0	1	1	0	0	1	0	1	0	0	1
w	0	0									

c. Si cambiamos la implementación por una ROM y un multiplexor de buses. (0.5 puntos)

La ROM tendrá palabras de bits por palabra.

d. Calculad un camino crítico del circuito asumiendo $T_p(\text{And-2})=20$, $T_p(\text{Or-4})=40$, $T_p(\text{Biestable})=100$, y $T_p(\text{Dec})=50$, y que la entrada x está estable pasadas 110 u.t. después del flanco de reloj de inicio de ciclo y que la salida w debe estar estable 0 u.t. antes del final de ciclo. (1 punto)

Camino crítico:

Tciclo:

3. (objetivos 5.3) (0.5 puntos)

¿Cuántos bits necesito, como mínimo, para representar los siguientes números enteros en complemento a dos?

-23:

2:

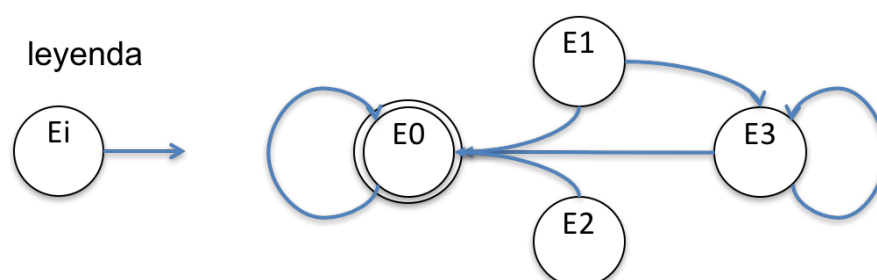
-3:

4. (objetivos 6.11) (2.0 puntos)

Queremos implementar un CLS con dos entradas, x e v, y una salida, Err, que nos ayude a detectar la estabilidad de señal de entrada x. La idea es que mientras la señal v valga 0, el circuito no hace nada y retorna un 0 en Err. Cuando v pasa a valer 1, se memoriza el valor de x y se comprueba que no cambie. Si el valor de x cambia mientras v vale 1, a partir del ciclo siguiente Err valdrá 1 y se quedará con ese valor hasta que v valga 0. Estemos en el estado que estemos, cuando v pase a valer 0, volvemos al estado inicial inactivo.

El cronograma simplificado muestra un ejemplo de entradas y salida. Completad el grafo que implementa este circuito añadiendo los arcos y etiquetas necesarias.

Ciclo	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x	0	1	1	1	1	0	0	1	1	1	0	0	0	1	0	0	0
v	0	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0
Err	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0



Apellidos y Nombre:

Grupo: DNI:

5. (objetivos 7.6) (2.5 puntos)

Queremos diseñar un PPE que controle la apertura de una caja fuerte protegida por una contraseña de 3 números. El circuito tiene las siguientes entradas:

- set: bit que si vale 1 indica que se quiere cambiar la contraseña
- try: bit que si vale 1 indica que se quiere introducir la contraseña y si es correcta abrir la puerta
- X: bus de 8 bits por donde entran los números (1 por ciclo) tanto para cambiar la contraseña, como para comprobar si la introducida por el usuario es correcta.

Y las siguientes salidas:

- open: bit que si vale 1 abre la puerta de la caja fuerte

El funcionamiento es el siguiente. Cuando llega un 1 en la entrada set, durante los 3 ciclos siguientes llegan los 3 números a través del bus X (1 por ciclo) que indican la nueva contraseña.

Cuando try vale 1, indica que, en los 3 ciclos siguientes, llegarán los 3 números indicados por el usuario a través del bus X. Si los 3 números coinciden, se dará la señal de abrir la puerta durante 1 ciclo (open = 1), en caso contrario a la que un número sea diferente se aborta la comprobación.

Una vez iniciadas las secuencias de set o try, no se interrumpirán hasta que finalicen.

En el caso en que lleguen set y try a la vez, se ignorará el try y se hará la secuencia de set.

Podemos asumir que una vez finalizada una secuencia de try o set, siempre pasan algunos ciclos hasta que llega una petición de try o set nueva. No tenemos que controlarlo, seguro que será así.

Diseña el grafo de estados que implementa la unidad de control de este circuito, sin olvidar la leyenda

