

- Mínimo de aprobación: un problema y un ejercicio
- Cada hoja debe tener Nombre y CI.
- Utilice solo un lado de las hojas
- Deben estar numeradas y la primer hoja debe decir el total de hojas
- Incluya un solo problema por hoja
- Sea prolijo

Ejercicio 1

a) Codificar en Hamming de 7bits: m3 m2 m1 m0 = 1101

| | | | | | | |
|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| P1 | P2 | m3 | P3 | m2 | m1 | m0 |

b) La siguiente es una palabra de 4 bits codificada en código Hamming de 7 bits: 0000011
Indicar cual era la palabra de 4 bits original.

Ejercicio 2

a) Determinar la frecuencia máxima de funcionamiento del circuito. Suponer la entrada X1X0 constante. Justificar claramente con un diagrama de tiempo, indicando a qué FF corresponden los parámetros utilizados.

b) Indicar cuál es el intervalo de tiempo durante el cual debe permanecer constante X1 para asegurar un correcto funcionamiento. Tomar como referencia el flanco de subida de clock. (X0 permanece constante)

Datos:

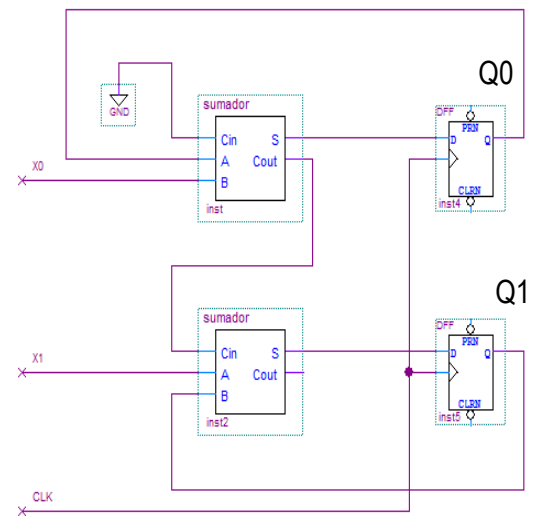
Tiempo de setup de FF: $t_{setupmin} < t_{setup}$

Tiempo de hold de FF: $t_{hmin} < t_h$

Tiempo de propagación de FF: $t_{pmin} < t_p < t_{pmax}$

Tiempo de prop. en sumadores entradas-salida s: $t_{summin} < t_{sum} < t_{summax}$

Tiempo de prop. en sumadores entradas-salida Cout: $t_{cmin} < t_c < t_{cmax}$



Problema 1

Se tienen 2 cintas transportadoras y un intercambiador que permite que los productos que viajan por la cinta 1 puedan pasar a la cinta 2 cuando un operario presiona un botón P=1.

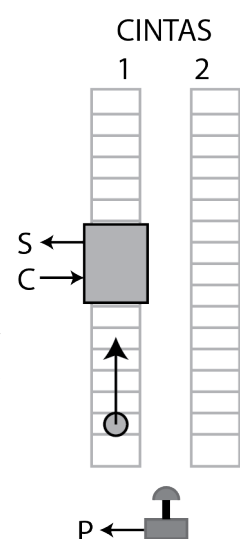
Si el operario presiona P y hay una pieza en el intercambiador, se debe activar la salida C hasta que la pieza haya salido del intercambiador. Si el operario suelta P la salida C se deberá mantener activa hasta que la pieza haya salido.

La salida C deberá activarse solo cuando se presiona P. Si llega una pieza y P está presionado, C NO debe activarse. Pero P podría soltarse y apretarse nuevamente y si la pieza está aún presente, entonces sí debe activarse.

El sensor S se pone a 1 durante todo el tiempo en que un producto está en el intercambiador.

Se supone que la distancia entre piezas es suficiente para que S valga 0 entre llegadas de nuevas piezas.

Diseñar un circuito modo nivel libre de carreras, espurios y azares, que reciba P y S y genere la señal C.



Problema 2

Se reciben por una señal S datos en formato serie. La cantidad n de bits serie de una palabra de datos es siempre múltiplo de 4 bits.

La señal S se encuentra sincronizada con el reloj del sistema y cada bit dura 1 período de reloj. Su formato es el siguiente:

- 1 bit de arranque: S=0.
- n bits de datos, con $n \geq 4$. Se comienza por el LSB (bit menos significativo).
- bandera de fin: 4 bits en 1
- se garantiza que los datos no contienen la secuencia "1111"
- en estado de reposo: S = 1
- pueden venir datos contiguos, es decir que en el período de reloj siguiente a la bandera de fin de dato puede venir un nuevo arranque

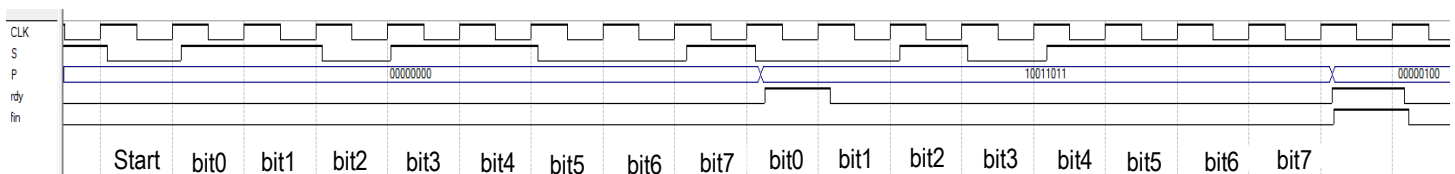
Se debe diseñar un circuito RTL, que tenga una entrada S, tal que sea capaz de recibir los datos por S, convertirlos a formato paralelo y sacarlos por su salida P[7..0] en paquetes de bytes (8 bits). Si la cantidad de bits no es múltiplo de 8, el último byte de P[7..0] se deberá ocupar con los bits de datos en los 4 lugares menos significativos y completarse el byte con "0000" en los 4 lugares más significativos.

La escritura en P se deberá hacer inmediatamente después del flanco en que se completó la recepción de 1 byte (ver figura); y se deberá actualizar cada vez que se reciba un nuevo byte, es decir cada 8 Tclk. Además, cada vez que se escribe un nuevo byte en la salida P se deberá activar la salida rdy = 1 durante 1 Tclk.

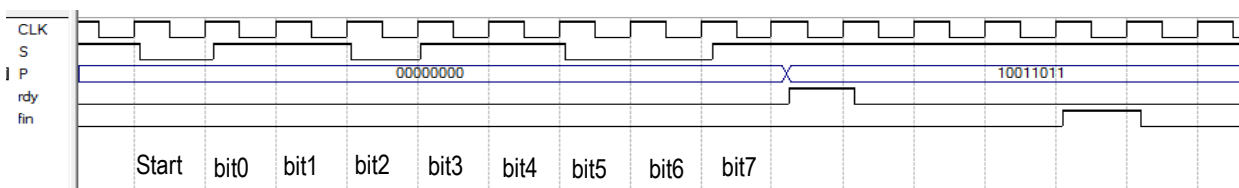
Se deberá generar una salida adicional fin=1, de 1 Tclk de duración, que indique si en P se está mostrando el último byte de una palabra de datos.

Luego de un reset la salida inicial del circuito debe ser P = 0000 0000.

En la figura 1 se muestra un ejemplo donde el dato es de 12 bits, y su valor es: 010010011011. Se presenta la salida en 2 bytes: 10011011 y 00000100 (en el segundo y último byte hay 4 bits de datos y se completa con 4 ceros).



En la figura 2 se muestra un ejemplo donde el dato es de 8 bits, y su valor es: 10011011. Se presenta la salida en 1 bytes: 10011011, no es necesario completar con ceros y fin recién se activará después de llegar el último 1 de la secuencia de parada 1111.



Se pide secuencia, bloque de datos y bloque de control del diseño RTL.