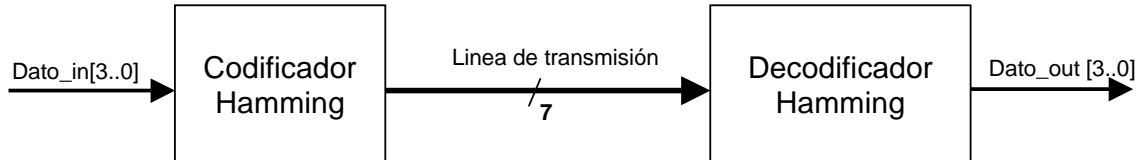


- Mínimo de aprobación: un problema y un ejercicio
- Cada hoja debe tener Nombre y CI.
- Utilice solo un lado de las hojas
- Deben estar numeradas y la primer hoja debe decir el total de hojas
- Incluya un solo problema por hoja
- Sea prolijo

### Ejercicio 1

Para transmitir datos por una línea de transmisión se utiliza el código de Hamming para poder corregir la información que llega al receptor.



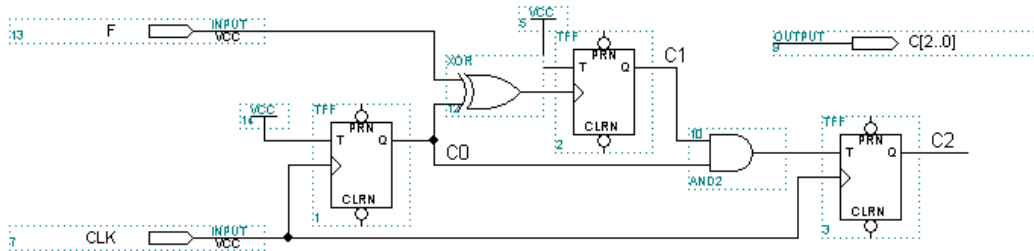
Realizar el circuito del **decodificador Hamming** de la figura de tal forma que se pueda asegurar que  $Dato\_in[3..0] = Dato\_out[3..0]$ . Utilizar el orden de los bits de la tabla siendo  $m_i$  los lugares de datos, y  $P_i$  las paridades.

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

Se pueden utilizar compuertas lógicas y bloques MSI (no se pide minimizar).

Se debe suponer que a lo sumo existe un único error por palabra de 7 bits transmitida.

### Ejercicio 2



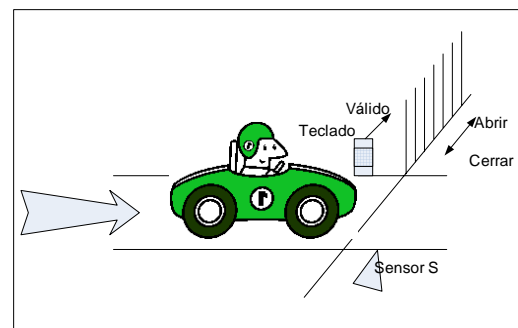
- Suponiendo que la entrada "F" permanece constante, dar una expresión de la frecuencia máxima de funcionamiento. Justificar con un diagrama de tiempos.
- Indicar en un diagrama de tiempos el intervalo en que la entrada "F" debe permanecer constante respecto al flanco para que el funcionamiento esté determinado.

Datos:

- tiempo de propagación en los FF:  $t_{p_{min}} < t_p < t_{p_{max}}$
- tiempo de setup de los FF:  $t_s$
- tiempo de hold de los FF:  $t_h$
- tiempo de retardo en la compuerta:  $t_{d_{min}} < t_d < t_{d_{max}}$

### Problema 1

Se desea diseñar un circuito modo nivel que controle la apertura de una reja de seguridad. El sistema debe funcionar de la siguiente manera: Cuando el auto se encuentra frente a la reja, el conductor ingresa su clave en un teclado que se halla junto a la ventanilla del conductor. La reja debe abrirse si el código ingresado fue correcto y mantenerse abierta solo hasta que el auto termina de ingresar.



Se dispone de un teclado cuya salida **válido** es **1** si la clave ingresada es correcta (coincide con la clave de autenticación). La señal **válido** vuelve a **0** en algún momento entre que se da el comando de **abrir** y antes que la reja termine de cerrarse.

Para detectar cuándo termina de pasar el auto se cuenta con un **sensor S** cuya salida es **1** cuando tiene un objeto delante y **0** en caso contrario.

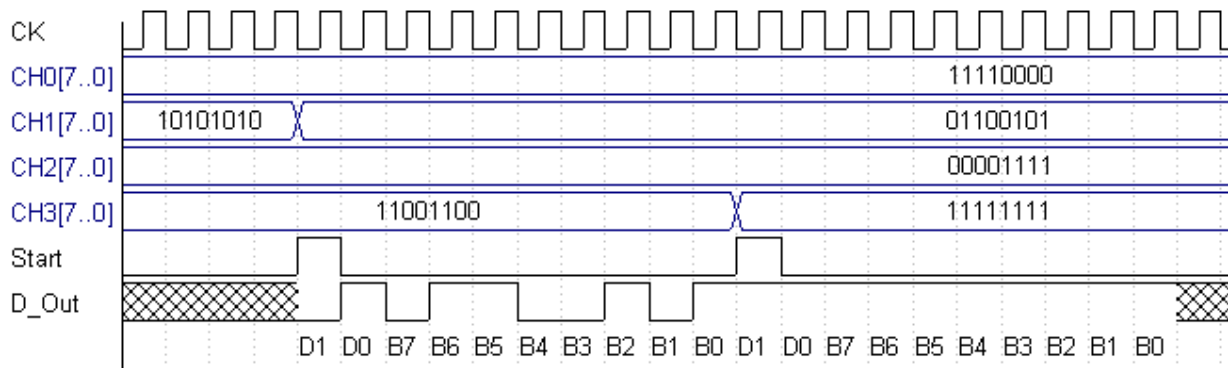
La reja comienza a abrirse con un flanco de subida en la señal **abrir** y a cerrarse con un flanco de subida en la señal **cerrar**. La reja se detiene sola una vez que queda completamente abierta o completamente cerrada.

## Problema 2

Se debe diseñar un circuito RTL que tenga como entradas 4 canales  $CH0[8]$ ,  $CH1[8]$ ,  $CH2[8]$  y  $CH3[8]$  y como salida  $D\_out$  y  $Start$  (de 1 bit).

Los canales  $CHi[8]$  cambian de valor en forma sincronizada con el reloj del sistema, con una separación no menor de 10 períodos de reloj. Es decir, que si  $CHi[8]$  cambió en este instante,  $CHj[8]$  cambiará de valor a lo sumo dentro de 10 períodos de reloj (cuales quiera sean  $i$  y  $j$ ).

El circuito a diseñar debe detectar cuando se produce un cambio en una de sus entradas. En ese momento debe poner en 1 la salida  $Start$  durante 1 período de reloj y simultáneamente comenzar a transmitir en forma serial el número de canal que cambió  $D[1..0]$  y el nuevo valor  $B[7..0]$  que adquirió. En la figura 1 se presenta un ejemplo en el que cambia  $CH1$  y luego de 10 períodos de reloj,  $CH3$ .



**Figura 1.** Cambio en  $CH1$  y  $CH3$  espaciados 10 períodos de reloj.

$Start$  solo toma el valor "1" en el momento que se transmite  $D1$ , el resto del tiempo debe ser "0".  $D\_Out$  no esta especificada fuera del tiempo de transmisión (desde  $D1$  a  $B0$ ).

Luego de un reset los canales permanecen constantes al menos por 10 períodos de reloj y pueden tener cualquier valor.

Se pide diseñar un circuito RTL (secuencia, bloque de datos y bloque de control) que cumpla con la especificación anterior luego de un reset.

Nota: Se pueden utilizar los bloques de lógica habituales. Para el caso de los comparadores, utilizar la notación  $Comp(A,B)$  (que es "1" si  $A=B$  y "0" en caso contrario).