

Problema 1. Codificar en VHDL los siguientes elementos HW sencillos:

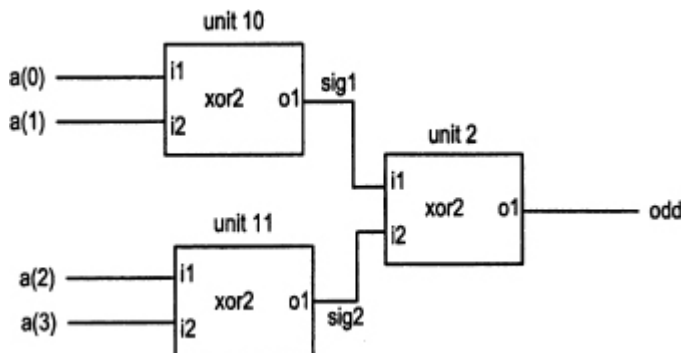
- Una XOR de dos números de 4 bits
- Un multiplexor 4 a 1
- Un decodificador 3 a 8
- Un codificador 8 a 3
- Un comparador de dos números de 4 bits con salida igual a 1 cuando son iguales

Problema 2. Codificar en VHDL el siguiente sistema combinacional:

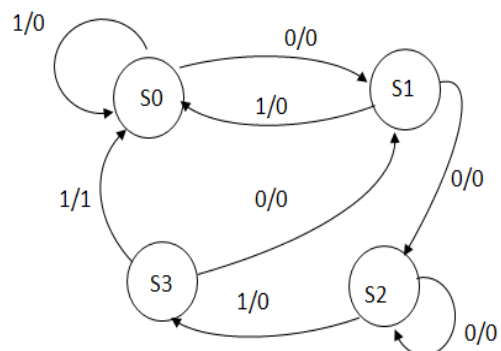
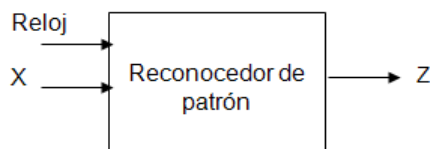
- Tiene como entrada un número positivo del 0 al 15
- La salida Z vale 1 si el número de la entrada cumple alguno de estos requisitos:
 - o Es un número primo
 - o Es menor de 4 y par (considerar el 0 como par)
 - o Es mayor de 8 e impar.

Problema 3. Codificar en VHDL un sistema combinacional cuya función es multiplicar por 3 un número X entero positivo en el rango {0 a 7}. La salida es un número Z entero positivo en el rango {0 a 15}. Además el sistema tiene un bit de salida D que indica si hay desbordamiento.

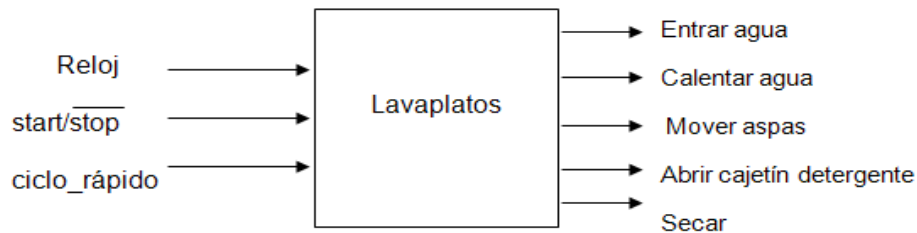
Problema 4. En este ejercicio se muestra el diagrama de un circuito formado por puertas XOR. Escribir, basado en el diagrama, el código VHDL que describe la arquitectura (*architecture*) del circuito.



Problema 5. Dado el diagrama de estados del sistema secuencial representado en la figura, escriba el código VHDL que lo describe.



Problema 6. Diseñe en VHDL un sistema secuencial para controlar el funcionamiento de un lavaplatos. El sistema tiene 2 entradas, la tecla **start/stop** (**asíncrona**), y la tecla de **ciclo_rápido** y 5 salidas como muestra la figura. En el estado inicial, todas las salidas valen 0. Desde cualquier estado se va inmediatamente al estado inicial siempre que la tecla **start/stop** vale 0, y allí se permanece hasta que **start/stop** vale 1, que comienza a funcionar desde el principio. Durante el funcionamiento el aparato pasa por 3 etapas: lavado (2 ó 4 ciclos dependiendo del valor de la tecla **ciclo_rápido**), aclarado (1 ó 2 ciclos dependiendo del valor de la tecla **ciclo_rápido**) y secado (1 ciclo). Después del secado se pasa siempre al estado inicial.



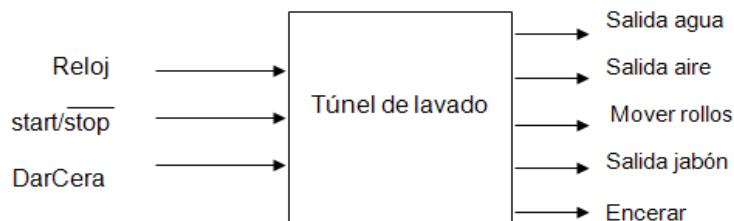
Durante el lavado entra agua durante el primer ciclo, y durante el mismo se calienta. En el segundo ciclo se abre el cajetín del detergente. Todos los ciclos del lavado se mueven las aspas.

Durante el aclarado entra agua el primer ciclo. Se mueven las aspas todos los ciclos del aclarado.

Durante el secado se activa la salida secar.

Problema 7. Diseñe en VHDL un sistema secuencial para controlar el funcionamiento de un túnel de lavado de coches. El sistema tiene 2 entradas, la tecla **start/stop** (**asíncrona**) y la tecla **DarCera** que permite la opción de encerar el coche, y 5 salidas como muestra la figura. En el estado inicial, todas las salidas valen 0. Desde cualquier estado se va inmediatamente al estado inicial siempre que la tecla **start/stop** vale 0. El túnel de lavado empieza a funcionar cuando **start/stop** vale 1.

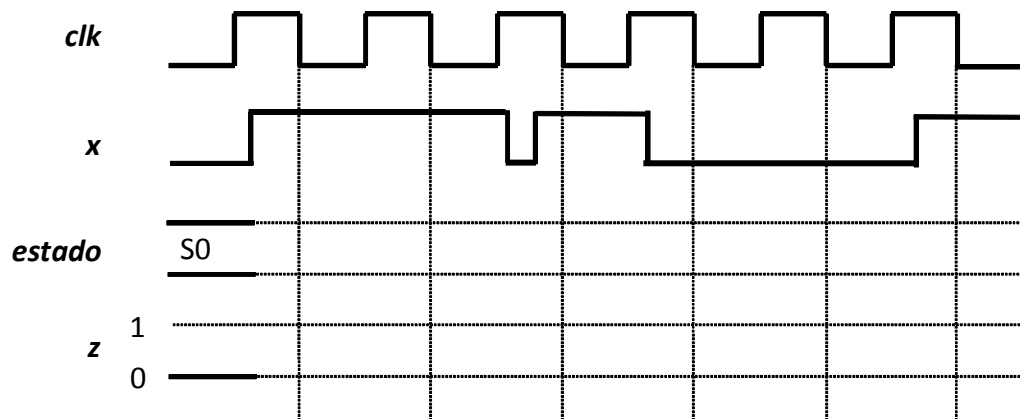
El funcionamiento es el siguiente: durante 1 ciclo rocía el coche con jabón, luego activa los rodillos durante 2 ciclos, después activa el agua para aclarar (1 ciclo) y el aire para secar (1 ciclo). Si la tecla **DarCera** está activada, antes de acabar está 2 ciclos dando cera. En caso contrario vuelve al estado inicial.



Problema 8. Sea el siguiente sistema secuencial:

$$z(t) = \begin{cases} 1 & x(t-2, t-1, t) = aaa \text{ ó } bbb \\ 0 & \text{en caso contrario} \end{cases}$$

- Diseñe dicho sistema en VHDL como máquina Mealy.
- Complete el siguiente cronograma:



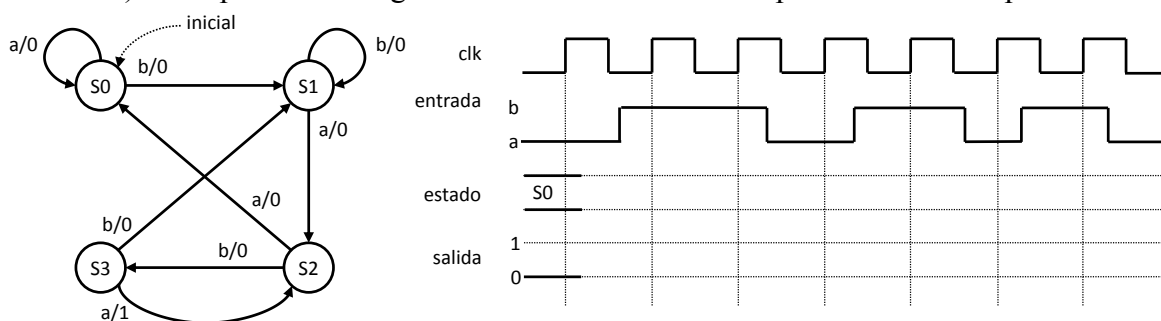
- c) Cree un fichero de test para simulación que se base en dicho cronograma y comparar resultados. Suponer que el periodo de reloj es de 100 ns.

Problema 9. Diseñe en VHDL el sistema de control de una muñeca interactiva (como máquina Moore). El sistema tiene 2 entradas y 2 salidas, todas ellas binarias. La entrada **R** valdrá 1 cuando haya ruido y la entrada **C** lo hará cuando haya un chupete en la boca de la muñeca. Por su parte, la salida **G** habilita un generador de sonidos que reproduce o bien un llanto (si **L** es igual a 1) o bien algunas palabras (si **L** es igual a 0).

Una vez encendida, la muñeca se encontrará en estado “tranquila” donde no habla, ni llora. Si se hace ruido, pasa al estado de “habla”. Si se le pone el chupete, dejará de hablar (si lo estuviera haciendo) y pasará al estado “dormida”. En el estado “dormida” no hace nada y permanecerá en él hasta que, sin tener el chupete puesto, se escuche un ruido. En ese caso llorará y pasará al estado “asustada”. En el estado “asustada” permanecerá llorando mientras el ruido se mantenga. Cuando el ruido desaparezca dejará de llorar y pasará a estar “dormida” o “tranquila” en función de si tiene o no el chupete puesto.

Problema 10. El diagrama de estados de la figura representa un reconocedor de patrón.

- ¿Qué patrón reconoce?
- Implemente dicho reconocedor en VHDL
- Complete el cronograma. Cree un fichero de test para simulación a partir de



dicho cronograma, suponiendo que la frecuencia de reloj es de 5 MHz.

Problema 11. Diseñe en VHDL un conversor serie-paralelo con 4 entradas de 1 bit (**Activa**, **reloj**, **rst** y **A**) y una salida **P** de 4 bits.

El circuito secuencial será capaz de recibir durante 4 ciclos de reloj una entrada de 4 bits. La entrada se recibe en serie, se lee 1 bit (**A**) en cada ciclo de reloj, empezando por

el bit menos significativo. Tras haber recibido todos los bits de entrada, el sistema lo transmite en paralelo, de forma que se puede visualizar la señal de 4 bits a la salida. El sistema posee una entrada asíncrona de reset, de modo que si **rst** = 1 la salida de todos los módulos secuenciales del sistema será 0. La señal Activa actúa como una señal de habilitación de forma que el sistema sólo funcionará si **Activa**=1. El sistema constará básicamente de un contador módulo 6 y un registro de desplazamiento de entrada serie y salida paralela de 4 bits. Su funcionamiento es el siguiente:

- Primer ciclo de reloj (la salida del contador Q=0): si **Activa**=1 se selecciona la entrada de control del registro (S) para realizar un desplazamiento y se habilita el funcionamiento del contador.
- Segundo, tercer y cuarto ciclo (Q=1, 2 y 3): el registro realiza desplazamientos.
- Quinto ciclo (Q=4): se realiza un desplazamiento y se codifica la señal S para que en el ciclo siguiente mantenga constante la salida.
- Sexto ciclo (Q=5): se activa el reset asíncrono del contador (Q=0).

Problema 12. El siguiente código VHDL calcula la paridad de una cadena de bits, pero no es sintetizable, nunca podrá ejecutarse en una FPGA. Realizar una máquina de estados y codificarla en VHDL que calcule la paridad de una cadena de bits.

```
architecture beh2_arch of even_detector is
begin
  process(a)
    variable sum, r: integer;
  begin
    sum := 0;
    for i in 2 downto 0 loop
      if a(i)='1' then
        sum := sum + 1;
      end if ;
    end loop ;
    r := sum mod 2;
    if (r=0) then
      even <= '1';
    else
      even <= '0';
    end if;
  end process;
end beh2_arch;
```

Problema 13. Se desea implementar un registro de desplazamiento a la izquierda de 8 bits con 3 bits de control, los bits de control indican la cantidad de bits que se tienen que desplazar. Teniendo en cuenta que el desplazamiento se tiene que realizar en un único ciclo de reloj, independientemente de la cantidad de bits a desplazar, diseñar el código VHDL describe la funcionalidad de este circuito. *NOTA no se puede utilizar la instrucción de VHDL SLL.*