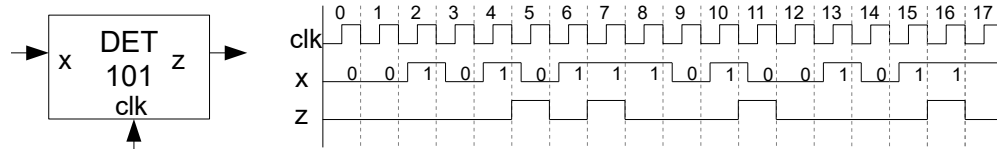


Pregunta 1

Diseñe el diagrama de estados para el circuito detector de secuencias 101, con nombre DET 101. Este circuito entrega un 1 en su salida z cuando la secuencia de bits vista en su entrada x en los últimos 3 ciclos es 101. La figura muestra a la izquierda las entradas y salidas del circuito, a la derecha un diagrama de tiempo que corresponde a un ejemplo de uso.



Sólo interesa el valor de las salidas justo antes del pulso de bajada del reloj. No se preocupe por las transiciones que ocurran en medio de un ciclo. **Ud. debe resolver el problema usando un diagrama de a lo más 4 estados.** Cuidado: en el ciclo 6 la salida es 0 porque la secuencia en los 3 ciclos anteriores fue 010, pero en el ciclo 7, la salida sí debe ser 1 porque la secuencia fue 101. Es decir que si en un ciclo no se detecta la secuencia 101, no significa que hay que esperar 3 ciclos para que pueda detectarse la secuencia pedida.

Pregunta 2

Parte a.- (2 puntos) Haga un mapa de Karnaugh para la siguiente fórmula booleana:

$$x\bar{y}zw + \bar{x}\bar{y}\bar{z}w + x y z w + \bar{x} y \bar{z} \bar{w} + x \bar{y} \bar{z} w + \bar{x} \bar{y} z w$$

Haga las simplificaciones en el mapa para reducir la fórmula a su mínima expresión, de acuerdo a la metodología enseñada.

Parte b.- (4 puntos) Justifique su resultado con álgebra de boole recurriendo específicamente a las 2 reglas de simplificación de fórmulas vistas en clase de cátedra y en clase auxiliar (como en la tarea 1).

Pregunta 3

Implemente el circuito *Bcd2Bin* de la figura de abajo a la izquierda. Este circuito recibe en su entrada x de 32 bits un número de 8 cifras en BCD (Binary Coded Decimal). Debe entregar en su salida z de 32 bits el mismo número representado en binario. El cálculo comienza cuando la entrada *start* se pone en 1 (por un solo ciclo del reloj) y termina cuando la salida *rdy* es 1 arrojando el resultado en z. Mientras se realiza el cálculo *rdy* debe permanecer en 0 y el valor de la salida z es irrelevante. La entrada x es solo válida cuando *start* es 1. Debe mantener las salidas *rdy* y z constantes hasta que *start* vuelva a 1. Para calcular z debe usar el algoritmo que muestra el cuadro de la derecha. Observe que $(z \ll 3) + (z \ll 1)$ es $10 * z$. El diagrama de tiempo muestra el funcionamiento solicitado. En su circuito use túneles de logisim para señales como *start*, *rdy*, en y *clk*.

```
unsigned bcd2bin(unsigned x) {
    unsigned z = 0;
    for (int i = 8; i > 0; i--) {
        z = (x >> 28) +
            (z << 3) + (z << 1); // 10*z
        x = x << 4;
    }
    return z;
}
```

