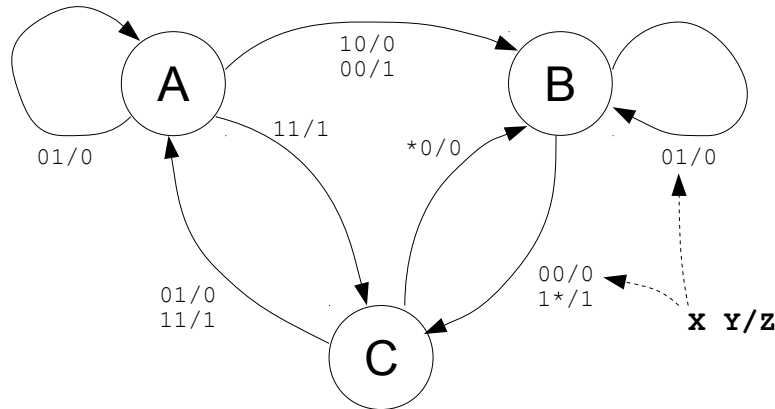
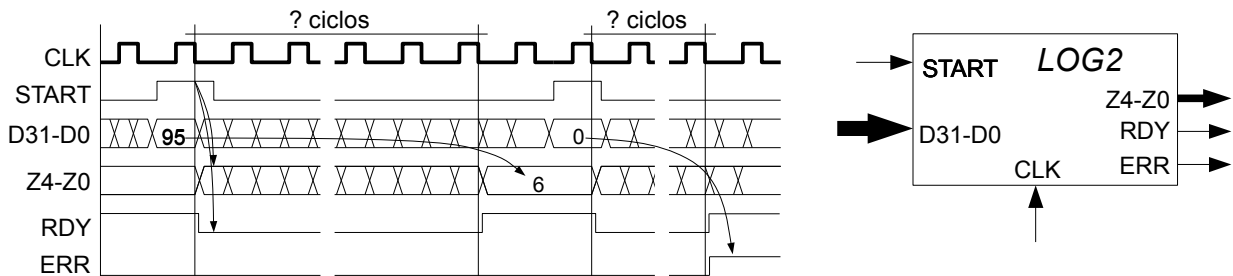


### Pregunta 1

Un circuito secuencial tiene entradas **X**, **Y**, y salida **Z**. El siguiente diagrama de estados especifica su comportamiento. Utilice la metodología vista en clases para implementar este circuito. Haga la codificación de estados, tablas de verdad, mapas de karnaugh y obtenga las fórmulas algebraicas.



### Pregunta 2



El circuito LOG2 de la figura calcula la parte entera del logaritmo en base 2 de un número entero binario de 32 bits. Esto equivale al peso (o posición) del bit más significativo de ese número. Es decir que si el valor aportado por ese bit es  $2^k$ , entonces su peso o posición es k. Por ejemplo  $\log_2(1)=\log(0b1)=0$ ,  $\log_2(2)=\log(0b10)=1$ ,  $\log_2(3)=\log(0b11)=1$ ,  $\log_2(4)=\log(0b100)=2$ , ...,  $\log_2(95)=\log(0b1011111)=6$ , ..., etc. Cuando START está en 1 durante el pulso de bajada del reloj, el circuito LOG2 lee el número binario que aparece en D31-D0, y comienza a calcular el logaritmo. Después de una cierta cantidad de ciclos se indica el término del cálculo colocando un 1 en RDY y el resultado aparece en Z4-Z0. Solo en el caso en que D31-D0 sea el número 0 las líneas ERR y RDY se colocan en 1 y Z4-Z0 queda indeterminado.

Implemente el circuito LOG2 usando diseño modular, recurriendo a las componentes vistas en clases, como multiplexores, registros, sumadores, restadores, desplazadores, etc. Si necesita un circuito secuencial o combinacional, solo especifique el diagrama de estados o la tabla de verdad, sin implementarlo. Para calcular el resultado Ud. *debe* usar el siguiente algoritmo:

```

int log2(unsigned d, int *perr) { // d es de 32 bits
    if (d==0) {
        *perr= 1;
        return GARBAGE;
    }
    *perr= 0;
    int z= 31;
    while (d&80000000==0) { // Se examina el bit más significativo de d
        z--;
        d= d<<1;
    } // Cada iteración es un ciclo del reloj
    return z;
}
    
```