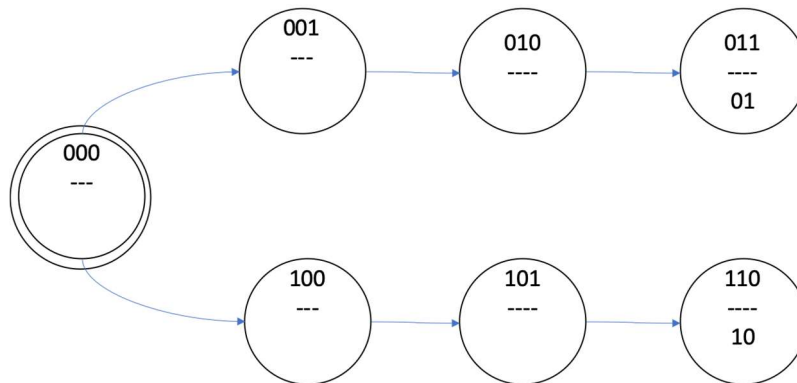


Ejercicio 3 (1,5 puntos)

Queremos implementar un circuito que detecte si el número entre 0's y 1's de la entrada (x) se desequilibra demasiado. Se considera desequilibrado si llegan 3 ceros o más que unos o 3 unos o mas que ceros. Una vez detectado el desequilibrio se dará una señal de alarma durante un ciclo codificada de la siguiente forma: W=10 si hay más ceros que unos y W=01 si hay mas unos que ceros. Una vez dada la alarma se volverá a iniciar la comprobación del equilibrio teniendo ya en cuenta la entrada (x) en el ciclo que hemos dado la señal de alarma. Mientras no demos alarma, el valor de W ha de ser 00. Completad el grafo del circuito (no olvidéis la leyenda). Por ejemplo, la secuencia 010111011 tiene 3 ceros y 6 unos por lo que al llegar el último uno debería activar la alarma de demasiados unos y volver a iniciar la comprobación a partir de siguiente valor de x, el que ha llegado en ciclo en que se ha activado la alarma.

**Ejercicio 4 (0,75 puntos)**

Dada la tabla de verdad correspondiente a una función que calcula w, responde a las siguientes preguntas.

- a) Dibujad el mapa de Karnaugh marcando claramente las agrupaciones de unos adecuadas para obtener la expresión mínima en suma de productos de la función w. Escribid la expresión mínima en suma de productos de w. (0.3puntos)

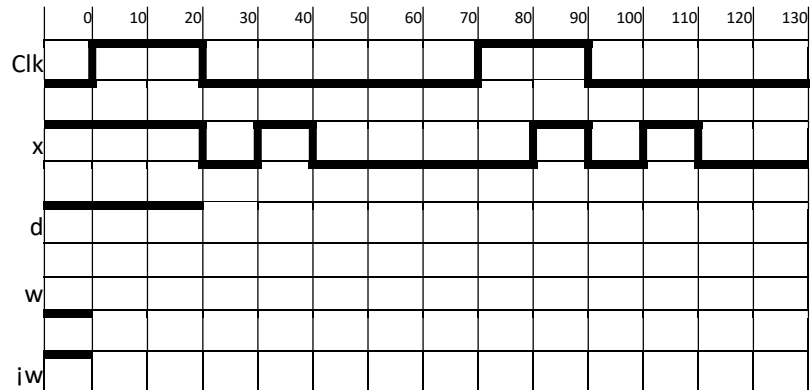
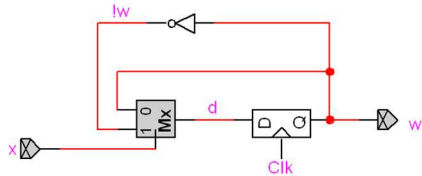
x	y	z	w
0	0	0	1
0	0	1	X
0	1	0	X
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

- b) Si implementamos **directamente** la expresión (**sin optimizar**) en suma de minterms de la función w considerando las x como 0, ¿Cuántas puertas And y OR y con cuantas entradas hacen falta? (0.1 puntos)

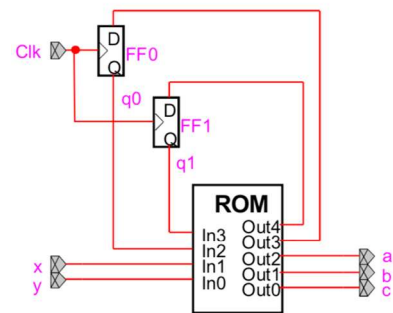
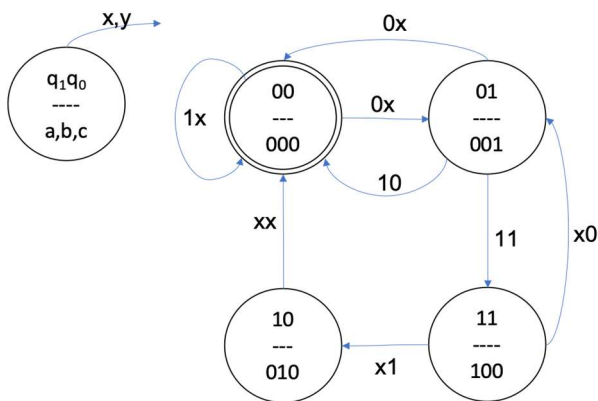
- c) Si implementamos la función w con una ROM. ¿Cuántas palabras y cuántos bits por palabra tiene la ROM? (0.1 puntos)

Ejercicio 5 (0,75 puntos)

Completad el siguiente cronograma de las señales del esquema lógico considerando que los tiempos de propagación son: $T_p(\text{Not})=10\text{u.t.}$, $T_p(\text{Mx})=20\text{t.}$, $T_p(\text{biestable})=30\text{t.}$

**Ejercicio 6 (0,75 puntos)**

Si se quiere implementar el grafo de estados siguiente con un circuito secuencial y una sola ROM tal y como se muestra en la figura.



Indicad, en hexadecimal, el contenido de las siguientes direcciones de la ROM.

ROM[0x2]= 0x

ROM[0x7]= 0x

ROM[0xF]= 0x

Ejercicio 7 (0,5 puntos)

Completa la siguiente tabla ensamblando las instrucciones en ensamblador SISA o desensamblando las instrucciones en lenguaje máquina según sea necesario. Indica poniendo NA en la casilla aquellos casos en los que la instrucción no corresponda al lenguaje SISA.

Lenguaje máquina SISA	Lenguaje ensamblador SISA
	MOVHI R3,0xAA
0x71C6	
	ADDI R4, R5, -1

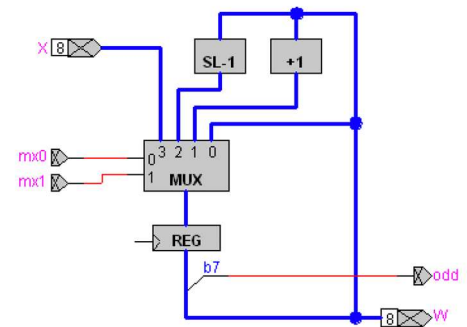
Ejercicio 8 (1,5 punto)

Queremos implementar un PPE que nos ayude a encriptar una entrada de 8 bits que llega por el bus X en el mismo ciclo que se activa la señal ini=1. La encriptación que haremos, muy sencilla, rotará dos bits hacia la izquierda, es decir, que si la entrada es $b_7b_6b_5b_4b_3b_2b_1b_0$ la salida será $b_5b_4b_3b_2b_1b_0b_7b_6$. Un ejemplo con números es el siguiente: si $X=10001010$ entonces W debería ser 00101010 .

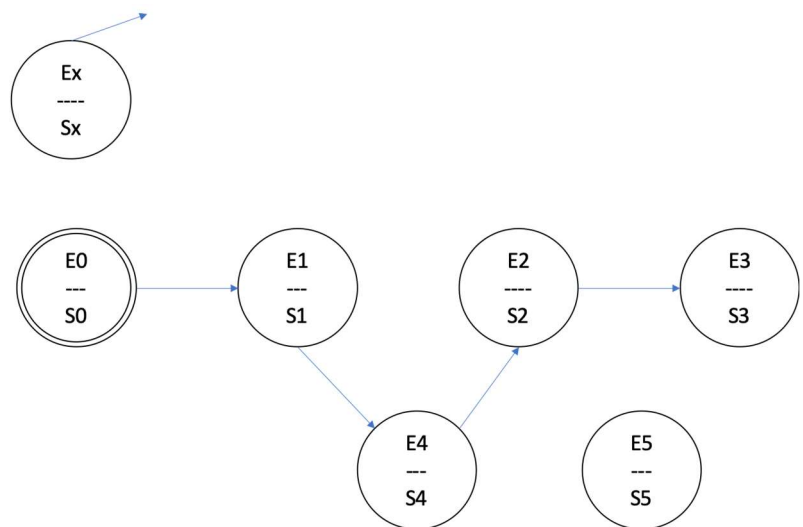
Una vez tengamos la salida disponible en el bus W, activaremos la señal de fi.

Durante la encriptación de una palabra se ignorará la señal ini, pero sí se tendrá en cuenta en el ciclo en que fi valga 1.

Os pedimos que completéis el grafo de la UC que implemente este circuito utilizando la siguiente UP. No os olvidéis de completar la leyenda.



	mx1	mx0	fi
S0			
S1			
S2			
S3			1
S4	0		
S5	0		

**Ejercicio 9 (0,75 puntos)**

Calcula el camino crítico (si hay más de uno, basta con uno) y el tiempo mínimo de ciclo del circuito del Ejercicio 8 suponiendo que la UC se ha implementado con una ROM y un multiplexor de buses. Asume los siguientes tiempos de propagación: $T_p(ADD)=660\text{u.t.}$, $T_p(MUX)=50\text{u.t.}$ (cualquier multiplexor de cualquier medida), $T_p(SHL-1) = 0\text{u.t.}$, $T_p(Reg)=100\text{u.t.}$, $T_p(ROM)=60\text{u.t.}$, todas las entradas necesitan 100u.t. para estabilizarse y las salidas deben estar 50u.t. estables antes del final del ciclo.

Camino crítico:

Tiempo mínimo de ciclo:

Ejercicio 10 (1,5 puntos)

Completad el fragmento de programa en lenguaje ensamblador SISA para que el procesador formado por la unidad de control de propósito general (UCG) junto con la UPG realice la funcionalidad descrita en el fragmento de código en C. El código SISA ya escrito **siempre utiliza el registro R7 para valores temporales**. Todos los datos son **naturales**. Rellenad la parte subrayada que falta. **Utilizad las etiquetas que os hemos puesto siempre que podáis.**

```

while (R2==R5) {
    if (R2 <= R3) {
        R5 = R5 - R2
    } else {
        R2 = R2 + 0xff
    }
}
MEMb[R2+10] = 10

```

```

.text
while: _____ R7, R2, R5
        _____ R7, while_end
        CMPLTU R7, _____
        _____ R7, else

then:   _____
        _____ R7, while

else:   _____
        _____
        _____
        _____ R7, while

while_end: _____
        _____ 10(_____), _____

.end

```