Organización y Arquitectura de Computadoras 2019-2

Práctica 5: Lógica Secuencial

Sandra del Mar Soto Corderi Edgar Quiroz Castañeda

13 de marzo del 2019

1. Minimización del autómata

La tabla de verdad del autómata se vería como sigue:

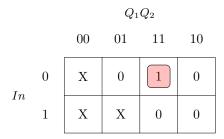
Cuadro 1: Tabla de verdad del autómata

In	E		E_{+1}		S_1	R_1	S_2	R_2
0	0	0	0	1	0	X	1	0
1	0	0	0	0	0	X	0	X
0	0	1	1	0	1	0	0	1
1	0	1	0	0	0	X	0	1
0	1	0	1	1	X	0	1	0
1	1	0	1	0	X	0	0	X
0	1	1	0	0	0	1	0	1
1	1	1	1	0	X	0	0	1

$$\bullet \ S_1 = \overline{Q_1}Q_2 \overline{In}$$

$$\bullet$$
 $S_2 = \overline{Q_2} \overline{In}$

 $R_1 = Q_1 Q_2 \quad \overline{In}$



 $\blacksquare R_2 = Q_2$

		Q_1Q_2							
		00	01	11	10				
In	0	0	1	1	0				
	1	X	1	1	X				

2. Preguntas

1. ¿En qué difieren los distintos tipos de flip-flops?

En cómo están configuradas sus entradas. Esto surge de que al tener entradadas de set y reset separadas (que son operaciones opuestas) no puedes garantizar que no vayan en entrar en conflicto.

El flip-flop SR simplemente asigna como *indefinido* cuando ambos estén encendidos, y sin cambios cuando ambos estén apagados.

El flip-flop JK es igual, sólo que cuando abmos están encendidos, se invierte el valor actual del estado.

Y el flip-flop D quita ambas entradas y las remplaza con una sólo, que cuando está encendida marca que hay que invertir el estado actual, y cuando está apagado indica que no va a cambiar nada.

¿Cómo se decide qué tipo se usará en el circuito?

Al final, los tres tienen un comportamiento equivalente de almacenar y modificar un bit, por lo que todo lo que puedes hacer con uno se puede hacer con los otro.

Pero dependiendo de la aplicación específica, es posible que los circuitos sean más pequeños usando cierto tipo. Esto es porque los diferentes flip-flops tienen diferentes "valores comodín" en sus tablas de transición que se pueden usar a favor al minimizar los circuitos deseados.

2. Un registro de desplazamiento es un circuito secuencial que desplaza a la izquierda o a la derecha la información contenida en el. Considerando el desplazamiento de 1 bit a la izquierda, ¿cómo se implementa dicho circuito?

De forma abstracta, desplazar todo a la izquierda un bit significa agregar un bit a la derecha, que significa que todo se multiplicó por la base, es decir por dos. Esto es análogo al sistema decimal, donde agregar un cero a la derecha equivale a multiplicar el número por 10, que es la base.

¿Cómo podríamos simular su funcionamiento con las operaciones que se tienen en la ALU de 8 bits?

Multiplicar un número por $n \in \mathbb{N}$ es equivalente a sumar ese número consigo mismo n-1 veces.

Entonces, para multiplicarlo por 2 basta sumar el número consigo mismo una vez.

Y la suma ya está implemetada en la ALU de 8 bits.