Organización y Arquitectura de Computadoras
Practica 5: Lógica Secuencial

Profesor: José Galaviz Casas
Ayudante de Laboratorio: Ricardo Enrique Pérez Villanueva

22 de marzo de 2023

1. Objetivos

1.1. Generales

El alumno aprenderá a analizar, diseñar y simular circuitos secuenciales.

1.2. Particulares

Al finalizar la practica el alumno estará familiarizado con:

- El diseño de maquinas de estados finitos.
- El funcionamiento de un banco de registros y su interacción con la unidad aritmético lógica.

2. Requerimientos

2.1. Conocimientos Previos:

- Funciones de conmutación.
- Minimización de funciones de conmutación por medio de manipulación algebraica y mapas de Karnaugh.
- Los componentes básicos del diseño de circuitos combinacionales: transistores y compuertas AND, OR y NOT.
- Conceptos de circuitos secuenciales: estado, memoria, reloj, etc.
- Funcionamiento de: biestables, latches y flip-flops.
- Las funciones de una unidad aritmético lógica. Se puede consultar los temas en [**Mano**] y [**Patterson**].

2.2. Tiempo de realización sugerido: 5 horas

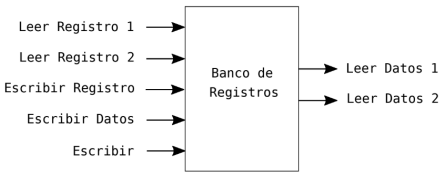
2.3. Numero de colaboradoras: Parejas (Equipos de 2 personas)

2.4. Software a utilizar.

- Java Runtime Environment versión 5 o superior.
- El paquete Logisim.

3. Planteamiento

Un registro es un circuito con la capacidad de almacenar datos, los cuales pueden ser leídos o escritos según sea necesario; son utilizados para guardar tanto operandos como resultados de las operaciones realizadas por el procesador, por lo que es común contar con un arreglo de registros junto con puertos dedicados para lectura y escritura, a esta unidad se le conoce como banco de registros o register file. Por la naturaleza de estos componentes, la implementación puede llevarse a cabo por medio de circuitos secuenciales. Nuestra tarea para esta practica sera simular un banco de registros, el cual interconectaremos con el sumador/restador de 8 bits elaborada en la practica anterior.

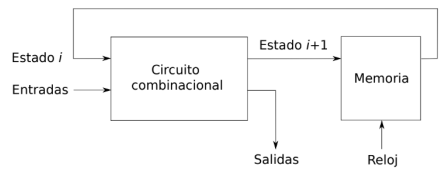


El banco de registro esta conformado por:

- Cuatro registros de 8 bits.
- Dos puertos de lectura. El banco de registros debe incluir dos entradas: **Leer registro 1** y **Leer registro 2** para indicar los registros que serán consultados y dos salidas: **Leer Datos 1** y **Leer Datos 2** en donde se colocaran los datos almacenados en los registros indicados.
- Un puerto de escritura. Para realizar la escritura de datos, se contara con tres entradas: **Escribir registro**, para indicar el registro en el que escribirán los datos; **Escribir Datos**, en donde se colocaran los datos a escribir; y **Escribir**, cuando esta señal de entrada cambie a 1, los datos serán escritos en el registro indicado.

4. Desarrollo:

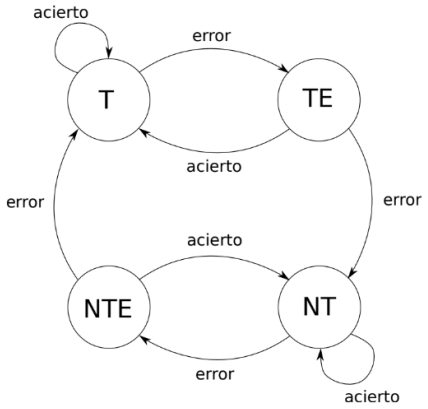
El diseño de circuitos secuenciales es similar al diseño de circuitos combinacionales, lo único distinto es que debemos considerar el estado en el que se encuentra el circuito para calcular el siguiente estado. Recordando el proceso de diseño de las practicas anteriores, el primer paso es hallar las variables de entrada para el circuito, como es necesario considerar el estado del circuito para calcular el siguiente, agregaremos las variables necesarias para representar el estado actual del circuito. A las funciones de salida le agregaremos las que representen el estado al que debe cambiar el circuito. El resto del proceso continua intacto hasta el momento de la simulación. El diseño que hemos obtenido hasta ahora es el de un circuito combinacional, agregando elementos de memoria para guardar el estado y un reloj para controlar la transición de estados, convertirá a nuestro circuito en uno secuencial, como se muestra en la figura 2. Los componentes que usaremos como memoria serán flip-flops.



4.1. Maquinas de estado finito

Hasta ahora, cada circuito que se presentaba era un circuito combinatorio. Eso significa que su salida depende solo de sus entradas actuales. Las entradas anteriores para ese tipo de circuitos no tienen efecto en la salida. Sin embargo, hay muchas aplicaciones donde es necesario que nuestros circuitos tengan "memoria"; recordar entradas anteriores y calcular sus salidas de acuerdo con ellas.

Un circuito cuya salida depende no solo de la entrada presente sino también de la historia de la entrada se llama circuito secuencial. Podemos ver que un circuito secuencial como un autómata, donde el estado actual de memoria es modificado por un input, el cual nos va a conducir un nuevo estado. Podemos modelar este comportamiento con flip flops, conectando la salida de un flip-flop con las entradas o el reset de otro flip-flop o inclusive de si mismo. Entonces, podemos ver que flip-flops es posible construir, en hardware, maquinas de estados finitos. Esos modelos de computo restringidos de acuerdo con el tamaño y las carácterısticas de acceso a su memoria. Los flip-flops son el medio que necesitamos para recordar el estado en que nos encontramos. Usaremos las funciones caracterısticas para llevar a cabo las transiciones del autómata. [1]
Para el ejemplo de esta practica, vamos a analizar el siguiente autómata y su tabla de valores.



Recordemos que, S_O y S_L con nuestros flip-flops, **In** es el bit de Input, **Next**(S_O) y **Next**(S_L) con los estados siguientes dado un estado actual y un input, S_OS y S_LS son el source de nuestro flip-flop y S_OR y S_LR con el reset del flip-flop.[2]

S_O	S_L	In	Next(S_O)	Next(S_L)	S_OS	S_0R	S_LS	S_LR
0	0	1	0	0	0	X	0	X
0	0	0	0	1	0	X	1	0
0	1	1	0	0	0	X	0	1
0	1	0	1	0	1	0	0	1
1	0	1	1	0	X	0	0	X
1	0	0	1	1	X	0	1	0
1	1	1	1	0	X	0	0	1
1	1	0	0	0	0	1	0	1

4.2. Banco de datos

El circuito principal a desarrollarse en esta practica, además de otros ejercicios, es la interconexión entre un sumador de 8 bits, desarrollado en la practica anterior, y el banco de registros, dicho circuito recibirá como entrada de 16 bits, la entrada representara una operación que deberá llevar a cabo el circuito. La decodificación de la operación sera como se muestra en la tabla, comenzando desde el bit mas significativo. Los códigos de operación del sumador son los mismos que la practica anterior.

No. de bits	Uso
1	Código de operación
1	Selector de segundo operando: 0 para registro, 1 para inmediato
2	Número de registro en el que se almacenará el resultado
2	Número de registro para el primer operando
2	Número de registro para el segundo operando
8	Inmediato entero para el segundo operando

Tabla 1: Decodificación de la instrucción.

[1] Para ver mas ejemplos sobre las maquinas de estado finito, consulta la pagina 35 de la bibliografía del curso. Puedes encontrar el capitulo sobre circuitos en este link `https://drive.google.com/file/d/1BdCwuwFcSar5W5nPxA98FVNfTxZfd6yg/view`

[2] Puedes aprender mas sobre estos terminos en la clase del 21 de marzo, donde explico mas afondo como se usan esta tabla.

5. Procedimiento

Deberás entregar un archivo de Logisim con las soluciones de los ejercicios, un archivo de código fuente escrito en el lenguaje de programación C con la función que simula el funcionamiento de una ALU y un documento PDF con las respuesta a las preguntas planteadas. Para la simulación de Logisim solo puedes hacer uso de compuertas lógicas **AND, OR, NOT y XOR**, multiplexores, separadores y pines de entrada y salida. Recuerda etiquetar las entradas y salidas de cada uno de los subcircuitos.
Deberas entregar un solo archivo de Logisim con las soluciones de los ejercicios y un documento con las respuesta a las preguntas planteadas. Solamente puedes hacer uso de compuertas lógicas AND, OR y NOT; flip-flops tipo D, JK y SR; multiplexores, demultiplexores; separadores; y pines de entrada y salida. Recuerda etiquetar las entradas y salidas de cada uno de los subcircuitos.

6. Ejercicios

1. Diseña un circuito secuencial en Logisim que simule el autómata descrito arriba, donde tendrás que minimizar las tablas para cada Source y Reset usando Mapas de Karnaugh o álgebra booleana, recuerda que los resultados de los 2 estados de memoria deben ir conectados a el cable 0 y 1, de un separador de 4 bits conectado a un LED Hexadecimal (Hex Display, en la parte de Input/Output de Logisim), de tal forma que despliegue la succión de números 0,1,2,3 o la sucesion de letras C,d,E,F en el LED Para guardar el estado, deberás usar flip-flops tipo SR.[3]

2. Diseña un circuito para simular un registro, solo puedes usar flip-flops tipo D.

3. Diseña un circuito para simular el banco de registros descrito en el planteamiento. Utiliza el registro diseñado en el ejercicio anterior.

4. Interconecta el sumador/restador de 8 bits elaborada en la practica anterior con el banco de registros del ejercicio anterior y de acuerdo a las especificaciones de las seccion de banco de datos.

7. Preguntas

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

2. ¿Como se decide que tipo se usara en el circuito?

3. ¿Como se ve un flip-flop SR usando solo compuertas lógicas básicas?¿Y un flip-flip JK?

4. Existen 2 tipos de maquinas de estado finito que usan Flip-Flops. ¿Cuales son?¿En que se diferencian?

5. ¿Porque es mejor utilizar flip-flops de tipo JK en vez de tipo SR para realizar maquinas de estado finito?

6. 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, ¿como se implementa dicho circuito? ¿Como podrıamos simular su funcionamiento con las operaciones que se tienen en la ALU de 8 bits?

[3] Vimos como se hacia en clase, por favor vean la clase o esto se les hará imposible :P