

INF245: Arquitectura de Computadores

Tarea 3

Profesor Mauricio Solar
Ayudantes Joaquín Montes
 Sebastián Alvarado
 Felipe González
 Javier Rojas

1. Reglas Generales

Para esta tarea se deberá utilizar el software Logisim¹, en el que deben construir los circuitos planteados en la sección de enunciado, siempre procurando usar la menor cantidad de compuertas lógicas posible.

Además deberá confeccionar un informe detallando el desarrollo de su tarea, utilizando tablas y mapas de Karnaugh. El informe debe entregarse en formato PDF y contener las secciones Portada, Resumen, Introducción, Desarrollo, Resultados, Análisis y Conclusión. Se recomienda utilizar L^AT_EX para construir el informe, y organizar el texto en secciones y subsecciones para mantener el orden.

2. Enunciado

Tras la implementación del display hexadecimal en su tarea anterior ahora se le pide crear una memoria primitiva para la consola de *OcularTM*. Para ello dispone nuevamente del software Logisim, el que deberá usar para desarrollar los circuitos necesarios para este encargo.

Su tarea consiste en implementar un circuito que reciba tres entradas, M, X y CLK, de 2 bits, 8 bits y 1 bit respectivamente (es decir, 11 bits en total), mientras que la salida S será de 8 bits. La tercera entrada CLK corresponde a un reloj que sincroniza todo el circuito.

El circuito debe contener una memoria interna compuesta de flip-flops D, y debe trabajar usando la entrada X de 8 bits y la pseudoentrada P(X) de 8 bits, donde P(X) es el valor anterior de X almacenado en la memoria interna.

El modo de operación del circuito dependerá del valor de la entrada M según la tabla 1, y el valor de la salida S dependerá tanto de la entrada X como del valor almacenado en memoria P(X).

Gracias a su tarea anterior ya sabe como implementar un sumador binario, pero esto de las memorias es nuevo para usted y por eso prefiere diseñarlo manualmente. Además, un colega le enseña un truco para realizar resta binaria: la resta $A - B$ es equivalente a la suma $A + \overline{B} + 1$.

Valor de M	Valor de S	Operación de S
00	Suma de X y P(X)	$X + P(X)$
01	Resta de X y P(X)	$X - P(X)$
10	Operación OR Bitwise	$X \text{ or } P(X)$
11	Operación AND Bitwise	$X \text{ and } P(X)$

Cuadro 1: Modos de Operación

¹<http://www.cburch.com/logisim/download.html>

3. Entrada y Salida de Datos

La entrada de datos será mediante pines (de la cantidad de bits que prefiera), mientras que la salida será a través de 8 LEDs en una línea horizontal, con el LED de la izquierda representando el bit más significativo de S, y el de la derecha el menos significativo, tal como lo muestra la figura 1.

La salida debe procesarse tras un ciclo completo del reloj CLK, por lo que se recomienda establecer una estructura de memoria interna con dos capas de flip-flops que almacenen los 8 bits de la entrada X y su estado anterior P(X), similar a un flip-flop maestro-esclavo como lo muestra la figura 2.



Figura 1: LEDs de salida

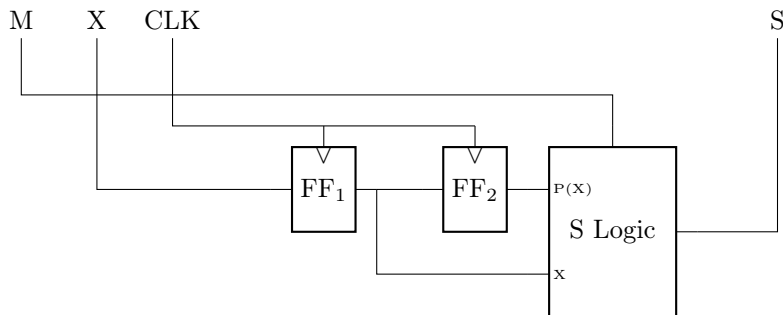


Figura 2: Estructura con memoria interna de dos capas

4. Datos de Ejemplo

A continuación se presenta una serie de entradas y sus respectivas salidas tras ser procesadas por el circuito. Cabe notar que la entrada P(X) no es externa, sino que solo representa los datos guardados en la memoria interna del circuito.

M	X (Entradas)	P(X) (Estado anterior)	S (Salida)
00	00110011	00000000	00110011
00	00001111	01101001	01111000
01	00010000	00001010	00000110
01	00001010	00010000	11111010
10	00001111	00110011	00111111
10	11111111	00000000	11111111
11	00001111	00110011	00000011
11	11111111	00000000	00000000

5. Consideraciones

- La tarea debe realizarse individualmente. Ante cualquier sospecha de copia o trabajo colaborativo se informará a las autoridades correspondientes.
- La tarea debe realizarse usando el software Logisim.
- Queda totalmente prohibido el uso de cualquier componente dentro de las carpetas *Arithmetic* y *Memory* de Logisim.
- La tarea se debe entregar via Aula en un solo archivo comprimido en formato `.zip` de nombre `T3_APELLIDO_ROL.zip` que incluya los siguientes archivos:
 - Un solo archivo `README.txt` con el nombre y ROL USM del estudiante, además de cualquier aclaración que sea necesaria.
 - Un solo archivo `.circ` que contenga los circuitos implementados. Haga uso de subcircuitos para ordenarlos sin necesitar más de un archivo `.circ`.
 - Un solo archivo `.pdf` con el informe completo del desarrollo de la tarea. Se recomienda utilizar \LaTeX (en Overleaf por ejemplo) u otra variante de \TeX para redactar la tarea.
- El informe debe contener las siguientes secciones, cada una ordenada y con toda la información necesaria:
 - Portada, incluyendo el nombre y ROL USM del estudiante, además de un título descriptivo.
 - Resumen, donde describa brevemente el desarrollo y resultados de la tarea.
 - Introducción, dejando claro el objetivo de la tarea y cualquier algoritmo que utilice.
 - Desarrollo, explicando detalladamente la resolución de la tarea.
 - Resultados, con todos los valores que haya obtenido durante el desarrollo de la tarea. Incluya extractos de cualquier prueba que haga con su tarea.
 - Análisis, donde discuta los resultados de la sección anterior y cualquier complicación con la que se haya encontrado.
 - Conclusión, comentando el nivel de finalización de la tarea. Además explique en profundidad la utilidad de los circuitos secuenciales.
- **Todas las preguntas respecto a la tarea deben hacerse a través del foro de consultas en Aula. No se responderán dudas durante las 48 horas previas a la entrega.**
- **La fecha límite de entrega de la tarea es el viernes 25 de junio de 2021 a las 23:59.**
- **Solo se recibirán tareas con un retraso máximo de 72 horas desde la fecha límite de entrega, las que podrán optar a una nota máxima de 75. Cualquier retraso por sobre este periodo será evaluado con nota 0.**