

Universidad Nacional Autónoma de México

Facultad de Ciencias

Organización y Arquitectura de Computadoras
2025-2

Práctica 05

Docentes:

José Galaviz Ricardo Pérez Ximena Lezama

Autores:

Fernanda Ramírez Juárez Ianluck Rojo Peña

Fecha de entrega: Jueves 20 de marzo de 2025



Ejercicios.

Mapas de Karnaugh para el Ejercicio 1.

Mapas de Karnaugh

$s_0 s_1$	00	01	11	10
s_2	0	0	1	0
s_3	1	0	0	1

$s_0 s_1$	00	01	11	10
s_2	0	1	0	0
s_3	1	1	0	0

$s_0 s_1$	00	01	11	10
s_2	0	1	0	0
s_3	1	0	0	0

$s_0 s_1$	00	01	11	10
s_2	0	0	1	1
s_3	1	1	1	1

Preguntas.

1. En el contexto de los biestables: ¿Qué es el problema de la meta estabilidad? ¿Cómo puedes reducir la meta estabilidad en sistemas reales?

Un biestable es un circuito multivibrador con dos estados estables, que puede cambiar de estado mediante señales de control. Su principal función es almacenar un bit de datos binarios y se emplea en aplicaciones como contadores, registros y divisores de frecuencia.

La metaestabilidad ocurre cuando un biestable entra en un estado indefinido, la señal de entrada no cumple con los tiempos de establecimiento (*setup time*) o retención (*hold time*), lo que impide que el biestable alcance un estado estable de manera confiable; almacenando un nivel lógico no válido. Este fenómeno puede manifestarse de dos formas:

Metaestabilidad analógica: La salida se mantiene en un voltaje indefinido, cercano al umbral de conmutación.

Metaestabilidad oscilatoria: La salida alterna repetidamente entre los dos niveles lógicos.

La metaestabilidad puede provocar fallos en el funcionamiento del circuito digital y, en algunos casos, daños en el sistema. Su resolución es teóricamente ilimitada, dependiendo de la temporización de entrada y de ciertos parámetros del circuito.

Si bien la metaestabilidad es inevitable en dispositivos biestables, se pueden implementar estrategias para mitigar sus efectos, tales como:

- **Uso de circuitos sincronizadores:**
Consisten en biestables en cascada para filtrar estados metaestables antes de su uso en el sistema.
- **Circuitos de handshake:**
Los cuales aseguran la correcta sincronización de señales asíncronas.
- **Sincronizadores de múltiples etapas con duplicadores de reloj:**
Permiten estabilizar la señal de salida reduciendo el tiempo de resolución.
- **Selección de dispositivos con alto producto ganancia-ancho de banda:**
Lo que minimizan el tiempo de recuperación del biestable.
- **Optimización del diseño del sistema:**
Mantener una alimentación estable y controlar la temperatura del chip, ya que la degradación de la tensión de alimentación y el aumento de temperatura pueden agravar la metaestabilidad.

Estas medidas permiten mejorar la fiabilidad del sistema y reducir el impacto de la metaestabilidad en circuitos digitales críticos.

2. ¿Qué es el **Sesgo de Reloj** (Clock Skew)? ¿Este fenómeno impacta la frecuencia máxima de un sistema?

El sesgo de reloj, o *clock skew*, es la diferencia en el tiempo de llegada de la señal de reloj a registros secuenciales adyacentes dentro de un circuito síncrono, lo que provoca problemas de sincronización y afectar el rendimiento del sistema digital. Este desfase puede ser:

Positivo: cuando la señal de reloj llega más tarde al registro receptor que al registro transmisor.

Negativo: cuando la señal de reloj llega primero al registro receptor.

Este fenómeno es un desafío crítico en el diseño digital, ya que puede provocar violaciones en las restricciones de temporización, afectando el correcto funcionamiento del circuito.

El sesgo de reloj se origina principalmente por diferencias en el retraso de la red de distribución del reloj. Entre sus principales causas se incluyen:

- **Diferencias en $Y = 2025 - 1971 =$ la longitud de las interconexiones:**

Si las rutas de entrega del reloj a registros secuenciales tienen longitudes distintas, la señal llegará en momentos diferentes.

- **Variaciones en los retardos de interconexión:**

Factores como la resistencia, la capacitancia y el acoplamiento inductivo pueden generar diferencias de propagación en rutas de igual longitud.

- **Desigualdad en los retardos de las rutas lógicas:**

La presencia de puertas lógicas en la distribución del reloj puede introducir retardos adicionales si no están balanceadas.

- **Factores ambientales y de fabricación:**

Variaciones en la temperatura, diferencias en los procesos de fabricación e imperfecciones en los materiales pueden afectar la sincronización del reloj.

El sesgo de reloj influye directamente en la frecuencia máxima a la que puede operar un sistema. Un alto desfase puede reducir el margen de tiempo disponible para la propagación de datos entre registros, limitando la frecuencia de operación. Si el sesgo de reloj no se gestiona adecuadamente, puede llevar a fallos de temporización y errores en la ejecución del sistema digital.

Para mitigar este efecto, se emplean técnicas como el **clock balancing** (*equilibrio del reloj*), el diseño simétrico de rutas de reloj y el uso de buffers para compensar diferencias de propagación.

3. ¿Qué es **Circuito Secuencial Tolerante a Fallos** (Fault Tolerant Sequential Circuit)?
¿Podemos hacer Tolerante a Fallos el Contador y el Banco de Registros que construimos?

Un **Circuito Secuencial Tolerante a Fallos** es un tipo de circuito digital diseñado específicamente para seguir funcionando correctamente incluso cuando ocurren errores o fallas en alguno de sus componentes. Es como un sistema que puede *sobrevivir* a los problemas y continuar realizando sus tareas sin interrupción graves.

Existen varias técnicas para lograr la tolerancia a fallos en circuitos secuenciales, algunas de las más comunes son:

- **Redundancia de Hardware:**

Esta técnica implica duplicar o triplicar los componentes críticos del circuito. Si un componente falla, los otros pueden tomar el control y mantener el sistema en funcionamiento.

- **Detección y Corrección de Errores:**

Se utilizan códigos especiales y circuitos adicionales para detectar errores en los datos y corregirlos automáticamente. Esto asegura que la información procesada por el circuito sea precisa y confiable.

- **Reconfiguración Dinámica:**

En sistemas más avanzados, el circuito puede reconfigurarse automáticamente para evitar componentes defectuosos. Si se detecta una falla, el sistema puede cambiar su configuración para utilizar componentes alternativos.

- **Recuperación de fallos:**

Esta técnica consiste en que el sistema pueda regresar a un estado de operación estable y correcto, después de que ocurra una falla.

Estos circuitos son esenciales en una amplia gama de aplicaciones críticas, incluyendo:

- **Sistemas aeroespaciales:**

Aviones, satélites y naves espaciales.

- **Equipos médicos:**

Dispositivos de monitoreo, máquinas de diagnóstico y sistemas de soporte vital.

- **Sistemas de control industrial:**

Plantas de energía, fábricas y sistemas de transporte.

- **Sistemas de telecomunicaciones:**

Redes de comunicación y equipos de transmisión.

- **Sistemas informáticos críticos:**

Servidores, sistemas de almacenamiento de datos y aplicaciones financieras

Y sí, sí es posible implementar tolerancia a fallos tanto en el banco de registros como en el contador.

En el caso del contador, al ser un circuito secuencial que sigue una secuencia lógica predefinida, será relativamente sencillo detectar errores si su valor se desvía del patrón esperado. Para ello podemos:

- Implementar múltiples contadores en paralelo y comparar sus resultados para detectar discrepancias.
- Incorporar circuitos adicionales que validen la transición correcta entre estados.

Por otro lado, el banco de registros, lo podemos hacer tolerante a fallos implementando:

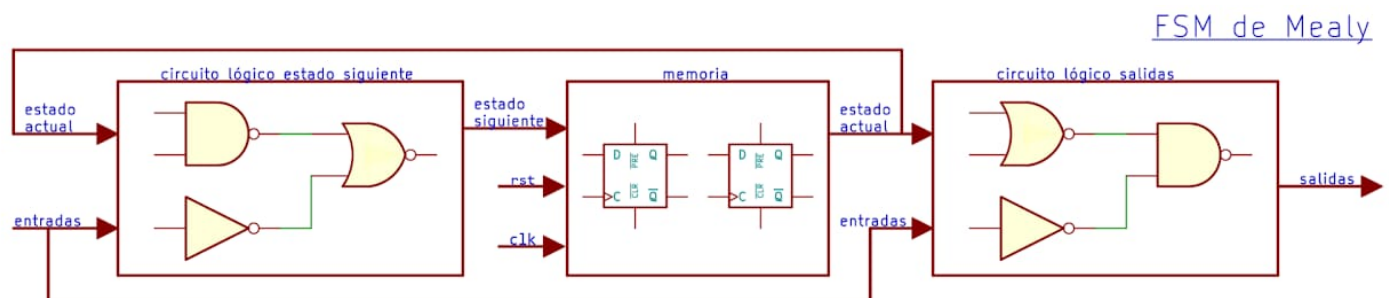
- Mantener múltiples copias de cada valor almacenado para su recuperación en caso de error.
- Bits adicionales para corrección de errores.
- Verificación de coherencia.

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

Las máquinas de estado finito que utilizan flip-flops se dividen:

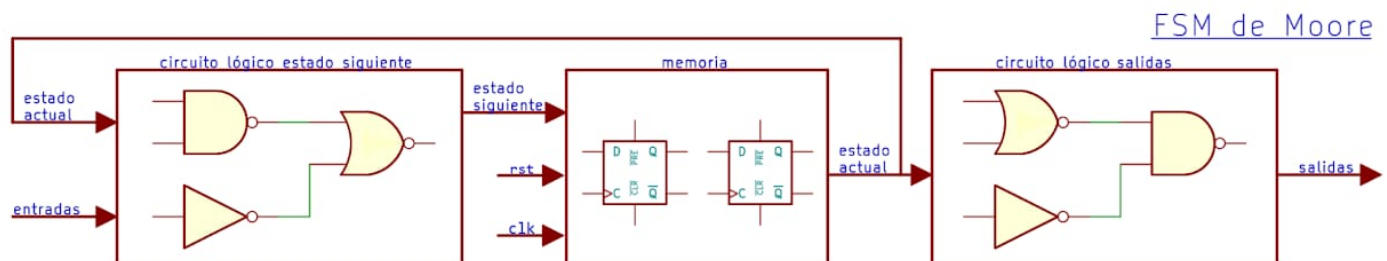
1) Máquina de Mealy.

- La salida depende tanto del estado actual como de las entradas actuales.
- Su circuito combinacional genera la salida considerando el estado y las entradas en ese instante.
- Dado que las salidas pueden cambiar en respuesta inmediata a una entrada, las máquinas de Mealy suelen reaccionar más rápido a cambios en la entrada.
- En su representación gráfica, las salidas se escriben en las transiciones entre estados, junto con las entradas correspondientes.



2) Máquina de Moore.

- La salida depende únicamente del estado actual, sin considerar las entradas directamente.
- El circuito combinacional que genera la salida solo toma en cuenta el estado de la máquina.
- Generalmente requiere más estados que una máquina de Mealy equivalente, ya que cada estado debe reflejar las posibles salidas.
- En su representación, la salida se asocia al estado mismo, mientras que las entradas se indican en las transiciones.



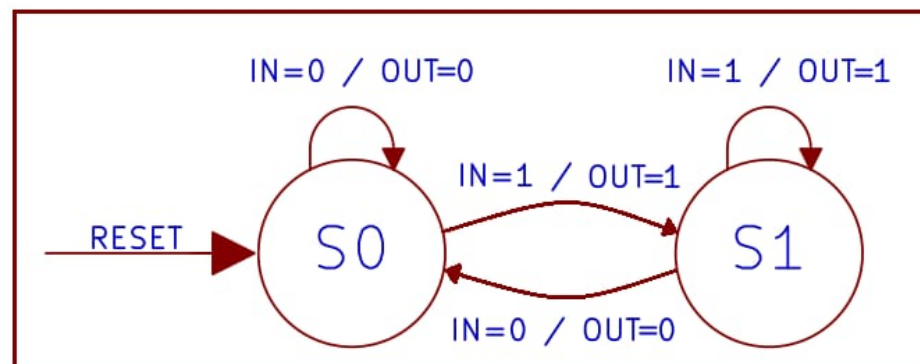
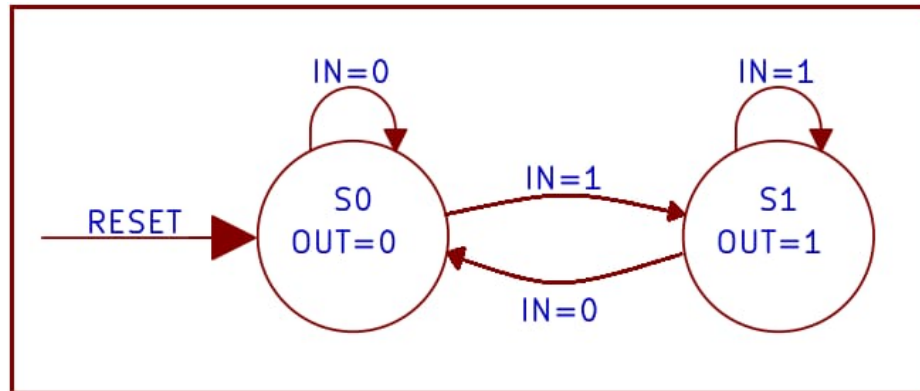
La principal diferencia radica en cómo se determinan las salidas:

En una máquina de Mealy, la salida cambia inmediatamente en función de la entrada y el

estado.

En una máquina de Moore, la salida solo cambia cuando el estado se actualiza.

Debido a estas diferencias, las máquinas de Mealy pueden ser más rápidas en respuesta, mientras que las máquinas de Moore ofrecen mayor estabilidad en las salidas, evitando cambios bruscos ante variaciones momentáneas en la entrada.



Punto Extra.

Link al video para el punto extra: [Registro de 4 bits con Latch tipo D](#)