



Curso: Taller de Diseño Digital

Laboratorio 2: Cuestionario Previo

PROFESOR

Roberto Carlos Molina Robles

Grupo #1

ESTUDIANTES

Oscar David Conejo Cantón 2020234423

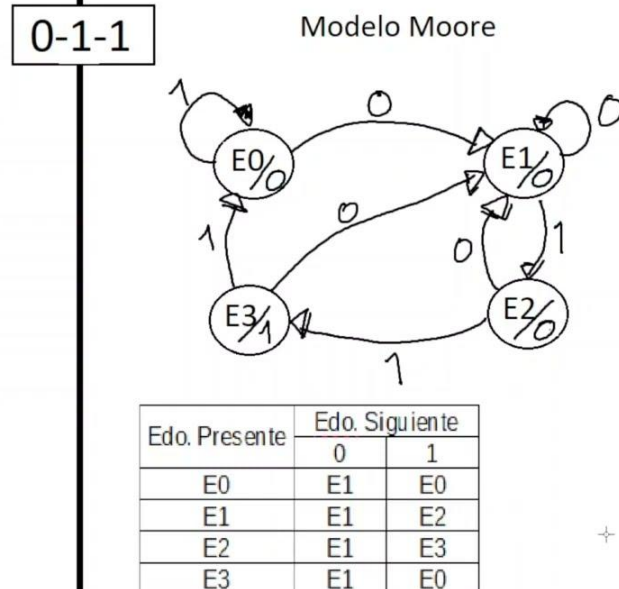
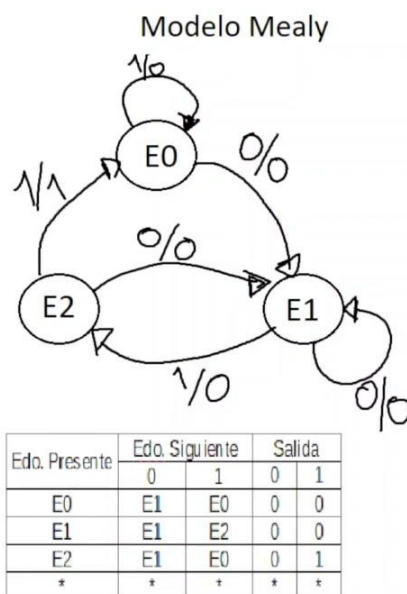
Katherine Daniela Salazar Martínez 2014160591

Juan Pablo Solano Solano 2020425831

I semestre 2025

Cuestionario Previo:

- Investigue cual es el funcionamiento de las máquinas de estado finitos. Explique la diferencia entre una máquina de Moore y una de Mealy, y muestre la diferencia por medio de diagramas de estados y señales.
- Las FSM son un modelo de comportamiento secuencial que cambia de estado en respuesta a entradas y sigue una lógica determinada. Estos estados son diferentes situaciones en las que puede estar el sistema. Estos estados son determinados mediante entradas o datos internos de la máquina. También tienen reglas que determinan como y cuando un estado cambia al otro. Cada estado puede estar asociado a una señal de salida de la máquina. Y siempre tienen un estado inicial en el que la maquina comienza. El sistema avanza de un estado a otro según las entradas actuales y la lógica de transición, y genera salidas en cada ciclo de reloj o evento.
 - En la maquina de Moore la salida depende solamente del estado actual, lo que significa que las salidas no cambian inmediatamente con la entrada. Sino que cambian cuando se hace la transición a un nuevo estado. Son fáciles de diseñar y de probar. Normalmente se implementan cuando se necesita una salida estable.
 - En la maquina de Mealy la salida depende del estado actual y de la entrada del sistema. La salida puede cambiar inmediatamente cuando cambia la entrada. Esto sin necesidad de cambiar de estado. Lo que generar que utilice menos estados que una maquina de Moore. Debido a esto es más rápida en respuesta a los estímulos externos.



2. Explique los conceptos de setup time y hold time. ¿Qué importancia tienen en el diseño de sistemas digitales?
 - El setup time es el tiempo mínimo antes del flanco activo del reloj durante el cual la entrada de un flip-flop debe mantenerse estable para que el dato sea correctamente capturado. Si el dato cambia muy cerca del flanco del reloj, el flip-flop podría capturar un valor incorrecto o quedar en estado metaestable.
 - El hold time es el tiempo mínimo después del flanco del reloj durante el cual la entrada debe permanecer sin cambiar para garantizar que el dato capturado sea válido. Si el dato cambia demasiado rápido después del reloj, el valor podría corromperse.
 - Estos conceptos son de suma importancia porque Evitan errores de sincronización como metaestabilidad o captura de datos incorrectos, afectan el rendimiento del sistema: el setup time contribuye al tiempo mínimo de ciclo de reloj, fundamentales en análisis de timing y guían decisiones de diseño, como agregar retardos, ajustar el reloj, o cambiar la lógica para cumplir los requisitos temporales.
3. Explique los conceptos de tiempos de propagación y tiempos de contaminación en circuitos combinacionales. Investigue sobre la ruta crítica y cómo esta afecta en el diseño de sistemas digitales complejos; por ejemplo, un procesador con pipeline. Investigue su relación con la frecuencia máxima de operación de un circuito.

En conceptos de circuitos combinacionales se pueden definir las siguientes definiciones de tiempos

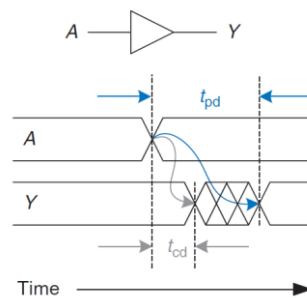
Tiempos de propagación

Un tiempo de propagación se refiere al tiempo que transcurre entre una señal de entrada y una señal de salida. Dentro de un circuito digital, amplificador o en este caso, uno combinacional, el tiempo de propagación corresponde al “delay” de una señal que tarda en ser procesada por el circuito. Un ejemplo de este tiempo lo define M Floyd en su libro al mencionar que “an RC circuit causes a **propagation delay** from input to output, thus creating a phase shift between the input signal and the output signal.” (2017). Lo cual también crea un cambio de fase. También lo definen David y Sarah Harris como “s the maximum time from when any input changes until the output or outputs reach their final value.” (2016), lo que se refiere al tiempo para un cambio de estado completo. De ahí el uso de clocks comunes en los circuitos combinacionales para permitir un funcionamiento apropiado según el

tiempo de procesamiento o tiempo de propagaciones de las señales de entrada hasta sus salidas

Tiempos de contaminación

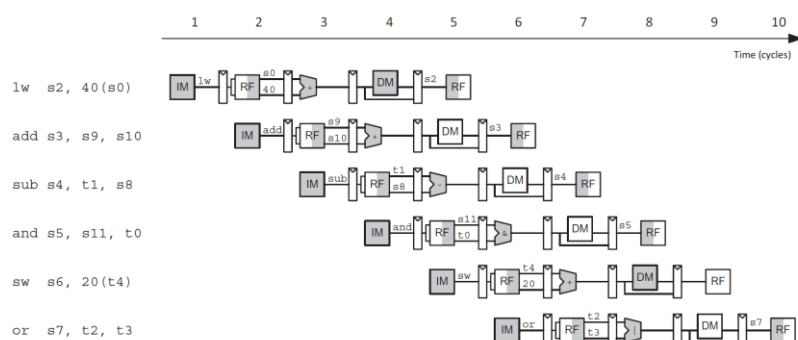
El tiempo de contaminación está relacionado al tiempo de propagación, el tiempo de propagación se refería al tiempo máximo para un cambio completo de estado de la salida en función de la entrada hasta que alcance su estado final, mientras que el tiempo de contaminación se el tiempo intermedio. Según lo detallan David y Sarah Harris en su libro “is the minimum time from when any input changes until any output starts to change its value” (2016). Se refiere al tiempo en el que la salida empieza a cambiar de estado. Esto por como su nombre lo indica se refiere a un lapso donde la salida puede presentar un error si se trata de leer mientras está en estado de transición. La siguiente imagen tomada del Libro de David y Sarah Harris (2016) muestra gráficamente el tiempo de contaminación de una señal mostrada con el nombre de t_{pd} .



Tiempo de contaminación de una señal

Ruta crítica

La ruta crítica se refiere al camino, cantidad de módulos o componentes por los que pasa un dato o una instrucción hasta se completamente procesada, en el caso del procesador Pipeline según David y Sarah Harris (2016) lo definen como el siguiente procesador



Instrucciones procesador pipeline

Según esta imagen se procesa una instrucción por cada ciclo de reloj, es importante tener en cuenta los tiempos de propagación y contaminación, ya que una de las desventajas de este procesador es que puede llegar a procesar un dato que todavía no está listo, a este error se le denomina Hazard y se trata de mitigar con un controlador de Hazard especial para este procesador. Todo esto demuestra la importancia del tiempo para este procesador, porque no solamente representa un atraso o no en la ejecución de sus instrucciones, sino en su funcionamiento general

4. Investigue sobre las mejores prácticas para la asignación de relojes y división de frecuencia en FPGA's. En este apartado haga énfasis en el uso de las entradas habilitadoras de reloj (clock enables) presentes en las celdas de la FPGA, para lograr tener tiempos de ejecución diferentes a lo largo del sistema mientras se utiliza un solo reloj.

En caso de las mejores prácticas para la asignación de relojes en FPGA's primero que nada hay que tener en cuenta cual FPGA se quiere utilizar. Dependiendo del modelo, cada FPGA tiene distintas frecuencias a las que puede operar, ya sea 50 MHz, 100 MHz, etc. Con esto en cuenta, se pueden resaltar las siguientes buenas prácticas.

Mejores prácticas para la asignación de relojes

Uso de un solo reloj:

O también llamado un clock global, es de alta importancia para que no haya errores de metástasis en los datos, también recordar que la frecuencia de este reloj debe estar dentro de las capacidades de la FPGA.

Uso Enables para el reloj:

Esto se refiere a que, dentro de la simulación, que exista una entrada capaz de controlar si el reloj está encendido o apagado. Esta práctica es muy útil si tratamos con Flip-Flops para saber si es conveniente o no capturar los datos que estén entrando. Esta práctica también da la posibilidad de crear múltiples "frecuencias virtuales" sin cambiar de dominio de reloj

5. Investigue sobre el fenómeno de rebotes y ruido en pulsadores e interruptores. Defina qué técnicas digitales y analógicas (circuitos) se utilizan para cancelar este fenómeno. Además, investigue sobre los problemas de metastabilidad cuando se tienen entradas asíncronas en circuitos digitales. Finalmente, presente circuitos que permitan la sincronización de entradas como pulsadores e interruptores.

Fenómeno de rebotes y ruido en pulsadores e interruptores

Cuando se utiliza un pulsador o interruptor mecánico, al presionar o liberar el contacto no se realiza de forma instantánea. En lugar de cerrar o abrir el circuito de manera limpia, los contactos rebotan físicamente, generando múltiples transiciones entre los estados “abierto” y “cerrado” durante unos pocos milisegundos. Este fenómeno se conoce como **rebote (bounce)**.

Adicionalmente, los pulsadores pueden estar sujetos a **ruido eléctrico**, como interferencia electromagnética (EMI), lo cual también puede provocar lecturas erróneas o múltiples activaciones en un sistema digital.

Técnicas para cancelar rebotes y ruido

Técnicas digitales:

1. **Retardo por software:** Se detecta el cambio de estado del pulsador, se espera un retardo breve (5–20 ms), y se verifica si la señal permanece estable antes de aceptarla como válida.
2. **Filtro por contador:** Se requiere que el mismo estado se mantenga durante un número fijo de ciclos antes de registrar el cambio.
3. **Filtro digital por promedio o umbral:** En sistemas más complejos, se puede usar lógica secuencial o filtros digitales.

Técnicas analógicas:

1. **Filtro RC:** Un circuito con una resistencia y un condensador conectado al pulsador suaviza los cambios rápidos, filtrando rebotes.
2. **Schmitt Trigger:** Un inversor con histéresis (como el 74HC14) convierte la señal analógica filtrada en una señal digital con transiciones definidas.
3. **Flip-Flops tipo SR o D:** Usados para registrar un único cambio de estado, se pueden combinar con señales de reloj para evitar múltiples transiciones.

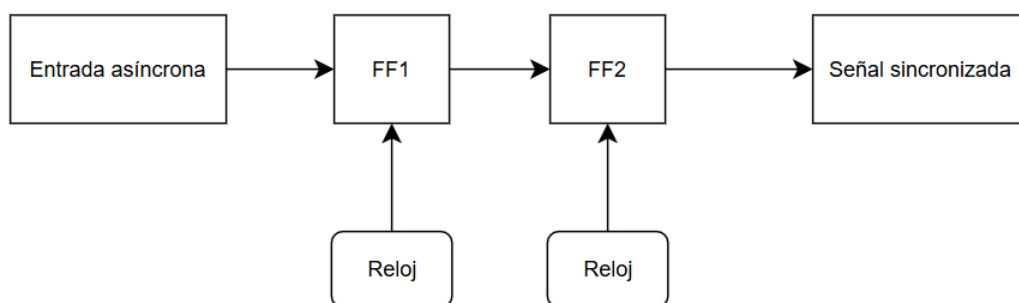
Problemas de metastabilidad en entradas asíncronas

Cuando una señal asíncrona (como la de un pulsador externo) se conecta directamente a un sistema digital síncrono (por ejemplo, un flip-flop que opera con reloj), puede presentarse **metastabilidad** si la transición ocurre muy cerca del borde del reloj. En este estado, el flip-flop puede quedarse en un nivel intermedio o producir una salida impredecible durante un tiempo corto, afectando la lógica del sistema.

Circuitos para sincronizar entradas como pulsadores e interruptores

1. Sincronizador doble Flip-Flop:

Consiste en dos flip-flops tipo D en cascada, ambos activados por el mismo reloj del sistema.



Este esquema reduce la probabilidad de metastabilidad en la salida del segundo flip-flop.

2. Circuito completo de depuración:

- **Filtro RC:** Suaviza la señal del pulsador.
- **Schmitt Trigger:** Genera transiciones digitales limpias.
- **Doble flip-flop:** Sincroniza la señal y elimina metastabilidad.

6. Investigue sobre el concepto de IP-Core. Revise la documentación relativa al uso de las herramientas de IP-Core en Vivado, en particular sobre el

Clocking-Wizard, y los IPs de verificación física: ILA (Integrated Logic Analyzer) y VIO (Virtual Input/Output). Sobre estos IPs, resume para qué se utilizan, cómo configurarlos y cómo utilizarlos en su proyecto.

Concepto de IP-Core

Un **IP-Core (Intellectual Property Core)** es un bloque de lógica reutilizable, predefinido y probado, que se puede integrar en sistemas digitales diseñados en plataformas como FPGAs o ASICs. Los IP-Cores pueden realizar funciones desde simples (como sumadores o temporizadores) hasta complejas (como controladores de memoria DDR o procesadores embebidos). En Vivado, los IPs se gestionan mediante el **IP Catalog**, que permite instanciar, configurar e integrar estos bloques en un diseño.

Herramientas de IP-Core en Vivado

Vivado Design Suite de Xilinx proporciona un entorno gráfico para la creación, configuración y uso de IP-Cores. El flujo de trabajo general para usar un IP en Vivado es:

1. Abrir el **IP Catalog**.
2. Buscar e instanciar el IP deseado.
3. Configurarlos mediante un asistente (Wizard).
4. Generar el bloque IP.
5. Integrarlo en un diseño (block design o HDL).
6. Conectarlo con los demás módulos y realizar síntesis e implementación.

Clocking Wizard

El **Clocking Wizard** es un IP que permite generar señales de reloj derivadas de una fuente de reloj externa. Puede usar componentes como **MMCM** o **PLL** internos del FPGA para generar relojes con distintas frecuencias, fases o características.

Utilidad:

- Derivar múltiples señales de reloj desde una fuente primaria.
- Ajustar la frecuencia, el desfase o la precisión del reloj.
- Asegurar señales de reloj adecuadas para diferentes módulos en el diseño.

Configuración:

1. Se selecciona el tipo de fuente (entrada).
2. Se define la frecuencia deseada para la salida.
3. Se puede activar la generación de señales adicionales como locked, reset, etc.

Uso:

- Se instancia en el diseño (por ejemplo, en el **Block Design**).
- Se conecta la señal de reloj externa a la entrada.
- Se usan las señales de salida como relojes para los demás módulos.

ILA (Integrated Logic Analyzer)

El **ILA** es un IP que permite **monitorear señales internas del FPGA en tiempo real**, como si fuera un osciloscopio digital embebido en el diseño.

Utilidad:

- Depuración y verificación de señales lógicas internas.
- Observación de buses, señales de control, FSMs, etc.
- Captura de eventos en funcionamiento real del hardware.

Configuración:

1. Se define el número de probes (entradas de monitoreo).
2. Se configura el ancho de cada señal a observar.
3. Se puede activar la compresión, capturas por trigger, etc.

Uso:

- Se conecta en el diseño a las señales que se desean observar.
- Se genera el bitstream e implementa el diseño.
- Desde **Vivado Hardware Manager**, se visualiza la ILA, se configura el trigger y se capturan señales en tiempo real.

VIO (Virtual Input/Output)

El **VIO** permite tanto **monitorear** como **controlar** señales internas del FPGA en tiempo real desde el entorno de Vivado.

Utilidad:

- Visualizar estados de señales internas.
- Forzar valores de señales internas para pruebas.
- Útil para validar FSMs, buses o activar secuencias específicas.

Configuración:

1. Se definen las señales de entrada (del FPGA al host) y salida (del host al FPGA).
2. Se especifica el ancho de bits de cada señal.

Uso:

- Se conecta en el diseño HDL o block design.
- Se implementa y carga en el FPGA.
- Desde el **Hardware Manager**, se visualizan y modifican las señales en tiempo real.

Bibliografía

Floyd, T. L. (2017). *Dispositivos electrónicos* (10ª ed.). Pearson / Editorial Limusa. ISBN 978-9681851170

Harris, D., & Harris, S. (2016). *Diseño digital y arquitectura de computadores* (2ª ed., Ed. Reverté). ISBN: 978-8429174207.