Universidad de Costa Rica

Escuela de Ingenería Eléctrica

IE-0523 Circuitos Digitales II

Tarea 2

Profesor:
Enrique Coen Alfaro

Alumno:
Diego Alfaro Segura C20259

19 de septiembre de 2024

1. Resumen

En el presente trabajo se estudia el proceso de síntesis por medio de la herramienta de síntesis Yosis, usando como módulo el controlador diseñado en la Tarea 1. En el proceso se verificó la capacidad de síntesis de dicho modelo al igual que su funcionamiento ante las pruebas originalmente impuestas en la Tarea 1. Se realizaron 3 procesos de síntesis distintos; síntesis con las celdas disponibles de forma predeterminada en la herramienta Yosis, síntesis con celdas CMOS indicadas en el enunciado, y síntesis con las celdas CMOS agregando retardos a las mismas. Se logró hacer el proceso de síntesis en cada caso y se obtuvieron los diagramas que muestran el resultado final. Además, se realizaron nuevamente las pruebas de la tarea 1 y las simulaciones verificaron que los módulos sintetizados cumplen con lo esperado. Finalmente, se evidenció y verificó el efecto de los retardos escogidos en las simulaciones y posibles consecuencias de retardos grandes en los diseños.

2. Descripción Arquitectónica

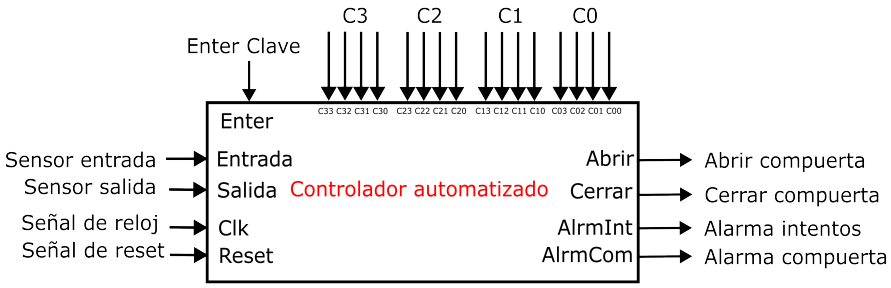

Figura 1: Diagrama de bloques del controlador diseñado.

En la Figura 1 se muestra el diagrama de bloques del controlador, este es el mismo diseño empleado en la Tarea 1 ya que ninguno de los cambios realizados para esta tarea modificaron la estructura original. Esto indica que el diseño original fue diseñado de forma correcta, o al menos no requiere cambios para ser sintetizado. En términos de síntesis, el módulo de controlador en sí solo se modificó para asesorar el uso de parámetros por defecto, ya que el sintetizador Yosys no considera los parámetros de este tipo, se cambió a un parámetro fijo y se le colocó el valor de los últimos 4 dígitos del carné (0259). En la sección 5 se explica el otro cambio realizado (equivalente al diseño original) para optimización donde se agrega un cable interno notClk para manejar el flanco decreciente de los vectores de entrada. Para cumplir el objetivo de sintetización de la tarea se emplean 3 conjuntos de instrucciones de Yosys distintos según las 3 formas de sintesis a probar:

2.1. Síntesis por descripción estructural genérica (RTLIL)

En esta se realiza la sintesis según los módulos/compuertas disponibles internamente en Yosys, esta incluye las compuertas estándar, muxes, and-or-invert, etc [2]. Para realizar esta síntesis se emplean los comandos:

```
read_verilog Controlador.v;
synth; opt; fsm; opt -full;
show; clean; write_verilog ControladorSynth.v;
```

Synth se encarga de hacer el proceso de síntesis general, mientras que fsm identifica formas de máquinas de estado en el módulo. Se agrega opt para optimizar y limpiar el diseño constantemente y al final se escribe a “ControladorSynth.v”. Este proceso genera una descripción estructural del módulo Controlador, empleando todas las celdas disponibles en Yosys por defecto.

2.2. Síntesis por mapeo tecnológico a CMOS

Este segundo paso implica utilizar las celdas especificadas en el enunciado para el diseño, estas solo incluyen Flip-Flops tipo D, buffers, AND, NOT, NAND y NOR. Esto restringe el diseño estructural pues no se pueden emplear Muxes o demás componentes. Se emplean los comandos:

```
read_liberty -lib ./lib/cmos_cells.lib; read_verilog Controlador.v;
synth; opt; fsm; opt -full;
dfflibmap -liberty ./lib/cmos_cells.lib; abc -liberty ./lib/cmos_cells.lib;
share; opt -full; clean; show; write_verilog ControladorSynth.v;
```

La primera línea carga el archivo e indica cuales son las entradas y salidas de las celdas (esto evita un problema encontrado al usar la librería CMOS, que todas las celdas son marcadas como que solo tienen inputs). Luego se realiza la síntesis y se asignan las celdas de la librería CMOS (línea 3). El resultado del proceso es un módulo estructural construido solamente a base de las celdas indicadas en "cmos_cells.lib" y con el comportamiento descrito por "cmos_cells.v".

2.3. Síntesis por mapeo tecnológico a CMOS con tiempos de retardo

Es el mismo proceso de síntesis de la sección 2.2, pero se usan archivos de celdas distintos ("cmos_cells_retardos.lib" y "cmos_cells_retardos.v") para incluir el efecto de los retardos en el diseño. En términos del módulo obtenido, la diferencia principal es que las celdas usadas serán las que poseen los retardos dados, hecho el cual se evidencia en los resultados de las simulaciones correspondientes. Comandos usados:

```
read_liberty -lib ./lib/cmos_cells_retardos.lib; read_verilog Controlador.v;
synth; opt; fsm; opt -full;
dfflibmap -liberty ./lib/cmos_cells_retardos.lib; abc -liberty ./lib/cmos_cells_retardos.lib;
share; opt -full; clean; show; write_verilog ControladorSynth.v;
```

Los retardos empleados se obtuvieron al analizar distintos datasheets de celdas CMOS reales, se tomaron al azar de libros de texto [3] y de proveedores de chips [1].

Cuadro 1: Asignación de retardos

Celda	Retardo (ns)
BUF	0.001
NOT	0.0117
NAND	0.05
NOR	0.07
DFF	0.8

3. Plan de pruebas

Del proceso de síntesis se espera que no hayan mensajes de durante el proceso y que se obtengan diagramas de la estructura de celdas/señales de los módulos obtenidos. En estos diagramas se espera identificar las principales partes de la máquina de estados definida en la Tarea 1. Para verificar el funcionamiento típico se emplearon cuatro tipos de pruebas que cubren las situaciones generales que pueden darse durante su operación normal, estas son las mismas que la Tarea 1, por lo que solo se muestran las figuras esperadas, no la explicación de cada una. Los únicos cambios realizados al archivo "tester.v" fueron la prolongación de la señal reset al inicio (se agregaron 10 unidades de tiempo para resolver problemas de que el módulo estructural no registraba la señal por suficiente tiempo para iniciar correctamente las pruebas) y se aumentó la resolución de la simulación (para ver los retardos). Para poder simular los módulos sintetizados con el tester se emplea el mismo testbench usado en la Tarea 1, con la excepción de que el módulo del controlador no maneja el parámetro de la clave correcta y se incluyen las librerías respectivas de las celdas CMOS.

3.1. Prueba #1, funcionamiento normal básico

Se espera encontrar la siguiente figura (prolongada por la ampliación del reset):

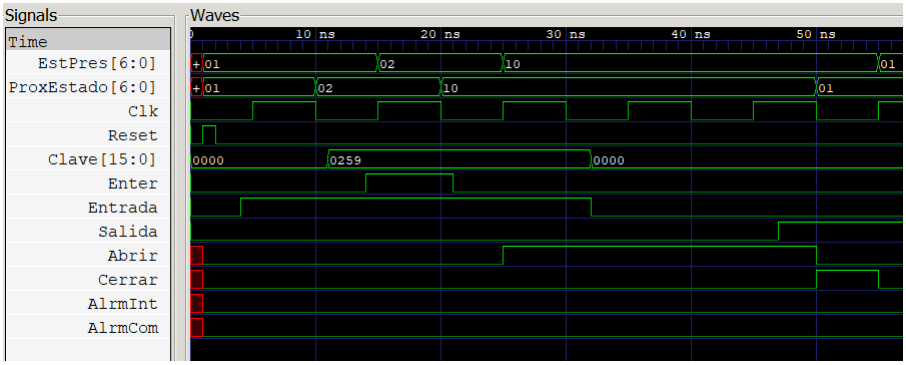

Figura 2: Diagrama de tiempo de prueba #1 de Tarea 1.

3.2. Prueba #2, ingreso de pin incorrecto menos de 3 veces

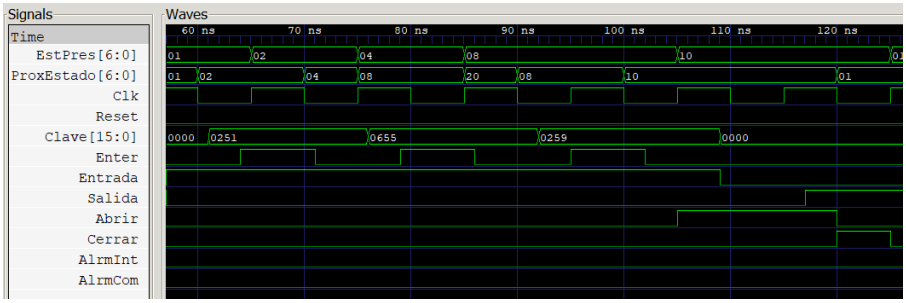

Figura 3: Diagrama de tiempo de prueba #2 de Tarea 1.

3.3. Prueba #3, ingreso de pin incorrecto 3 veces

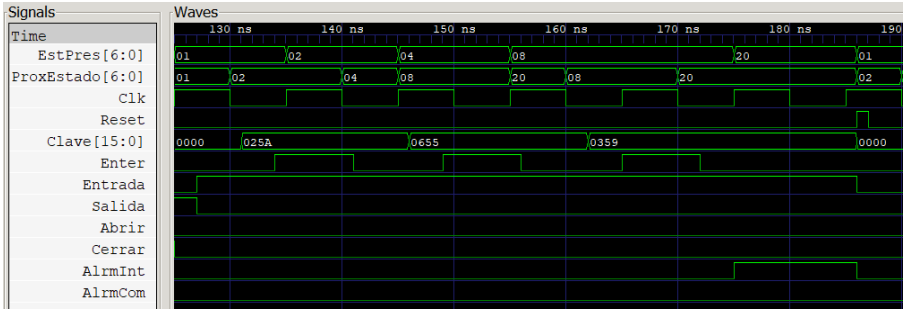

Figura 4: Diagrama de tiempo de prueba #3 de Tarea 1.

3.4. Prueba #4, alarma de bloqueo

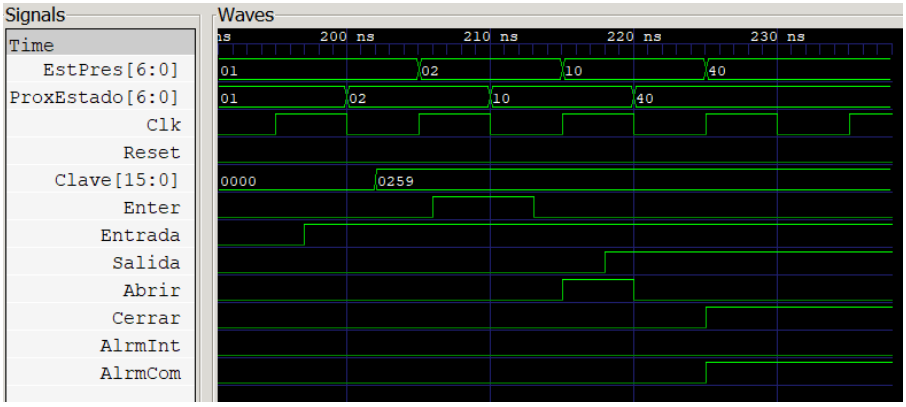

Figura 5: Diagrama de tiempo de prueba #4 de Tarea 1.

4. Instrucciones de utilización de la simulación

Para repetir las simulaciones utilizadas se incluye un archivo de Makefile que permite correr los comandos necesarios para realizarlas. Se debe tener instalado las herramientas de software Icarus Verilog (compilador de verilog), GTKWave (visualizador de ondas), Yosys (sintetizador de código Verilog) y GNU-make o mingw Make (comandos para ejecutar un Makefile). Además, se deben tener los archivos incluidos del diseño (tester.v, testbench.v y Controlador.v), el archivo Makefile en el mismo directorio que estos archivos y los archivos de la librería CMOS (cmos_cells.lib, cmos_cells.v, cmos_cells_retardos.lib y cmos_cells_retardos.v) en un subdirectorio llamado "lib". Se incluyen 4 comandos, notar que si se emplea la versión de make de mingw se debe escribir "mingw32-make" en vez de solo "make". **En este caso solo se puede emplear el GNU-make por las restricciones de Yosys**.

Comandos incluidos en el Makefile:

- make RTLIL Realiza la síntesis con las celdas por defecto de Yosys, corre la simulación al compilar el archivo y muestra los resultados en GTKWave, está configurado para mostrar todas las ondas importantes según se muestra en los resultados.
- make CMOS Realiza la síntesis con las celdas de cmos_cells, corre la simulación al compilar el archivo y muestra los resultados en GTKWave, está configurado para mostrar todas las ondas importantes según se muestra en los resultados.
- make CMOSretardos Realiza la síntesis con las celdas de cmos_cells_retardos (la versión de celdas CMOS con retardos), corre la simulación al compilar el archivo y muestra los resultados en GTKWave, está configurado para mostrar todas las ondas importantes según se muestra en los resultados.
- make clean Elimina los archivos generados para limpiar el directorio (no elimina los de código ni el Makefile).

5. Ejemplos de los resultados

A través de las simulaciones se obtuvieron los resultados mostrados en los siguientes diagramas de estructura sintetizada y diagramas de tiempo. Se comparan los diagramas de tiempo con las figuras de la sección 3.

5.1. Resultados síntesis RTLIL

Del proceso de síntesis se obtuvo el diagrama visto en la Figura 6. En esta se evidencian las partes principales del diseño conductual siendo la asignación de flip-flops para las entradas Clave, Entrada y Salida en la sección izquierda baja, también los flip-flops del vector de estado presente. Acompañado al vector de estado presente se encuentra la lógica combinacional de este y de la verificación de la clave unificada en la señal BCD. Se nota que las entradas y salidas del sistema se encuentran todas conectadas. Se pueden identificar varios tipos de celdas, algunas distintas a las típicas como OR_NOT o AND_NOT, esto verifica que se emplean las celdas incluidas por defecto en Yosys.

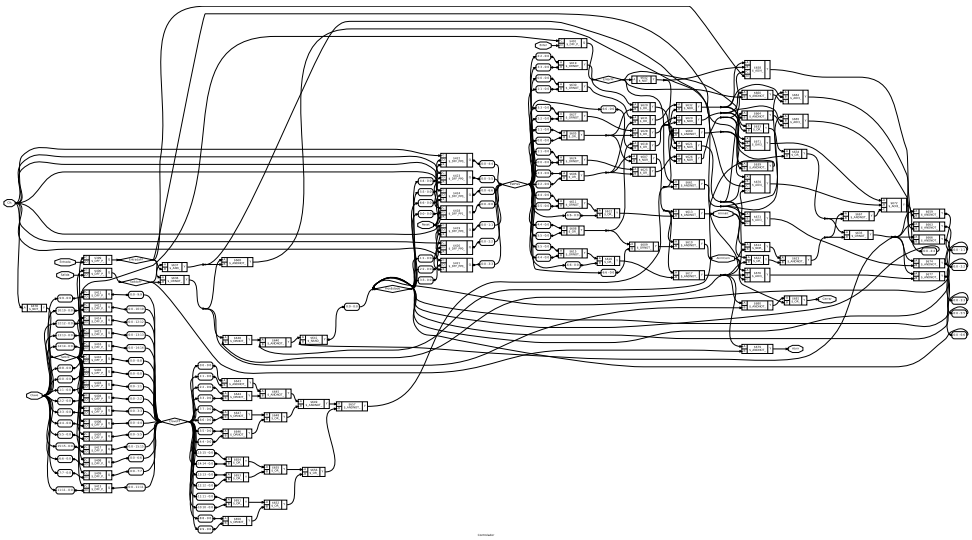

Figura 6: Diagrama resultante de síntesis con RTLIL.

5.1.1. Resultados prueba #1 Y #2

Al realizar la simulación completa se obtuvieron los siguientes resultados correspondientes a las pruebas 1 y 2. Tomar en cuenta la prolongación de la señal Reset y de la nueva resolución de la simulación.

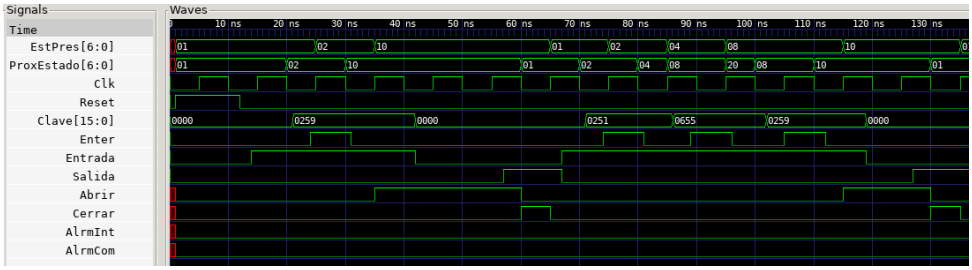

Figura 7: Diagrama de tiempo de pruebas 1 y 2 con módulo sintetizado (RTLIL).

De la figura se nota que durante ambas pruebas las salidas y sus tiempos son idénticos a los del diseño conductual, indicando que el módulo sintetizado cumple con estas pruebas.

5.1.2. Resultados prueba #3 Y #4

Al igual que en las pruebas 1 y 2, las pruebas 3 y 4 muestran el mismo comportamiento en el módulo sintetizado que en el conductual, evidenciado por la siguiente figura:

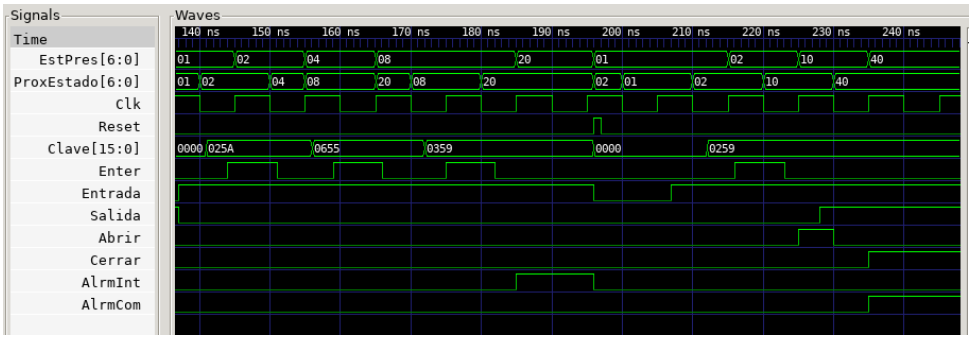

Figura 8: Diagrama de tiempo de pruebas 3 y 4 con módulo sintetizado (RTLIL).

5.2. Resultados síntesis CMOS

En la Figura 9 se muestra el diagrama resultante de la síntesis al emplear las celdas CMOS indicadas. Se nota que la estructura/posición del diagrama es distinto al visto en la Figura 6. Las principales diferencias

son las celdas usadas, puesto que solo se presentan las celdas disponibles en la librería CMOS. Se nota que se mantienen las principales partes del diagrama (como los flip-flops de los vectores de entrada y de estados), pero la lógica combinacional es distinta. De importancia es que para optimizar la cantidad de celdas usadas en el diseño se agregó un wire interno (notClk) para la señal de reloj negada y se cambió el cambio de entradas a el flanco positivo de notClk. **Esto es equivalente al diseño original**, pero se hace pues la librería CMOS solo contiene flip-flops de flanco creciente y el sintetizador creará celdas NOT para cada flip-flop encargado en la lectura de entradas. Este cambio reduce fuertemente la cantidad de celdas NOT sintetizadas.

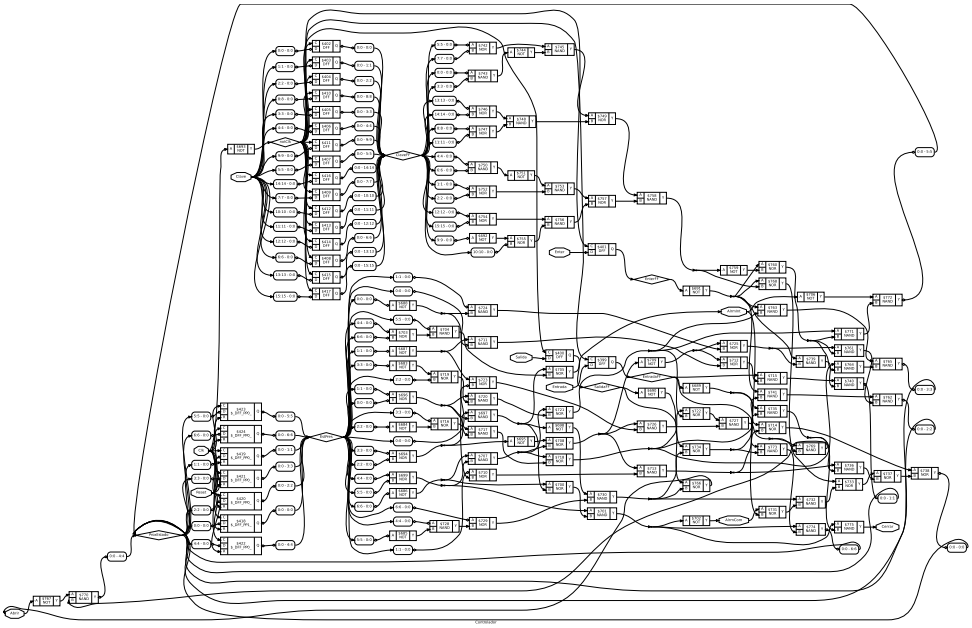

Figura 9: Diagrama resultante de síntesis con CMOS.

5.2.1. Resultados prueba #1 Y #2

De los resultados de la simulación del módulo sintetizado con CMOS se muestran básicamente los mismos resultados que los obtenidos en la simulación de la síntesis con RTLIL. Esto indica que el módulo diseñado no solo es sintetizable, sino que también es compatible a una síntesis con las celdas limitadas de la librería dada sin tener que hacer cambios que alteren el cambio del módulo.

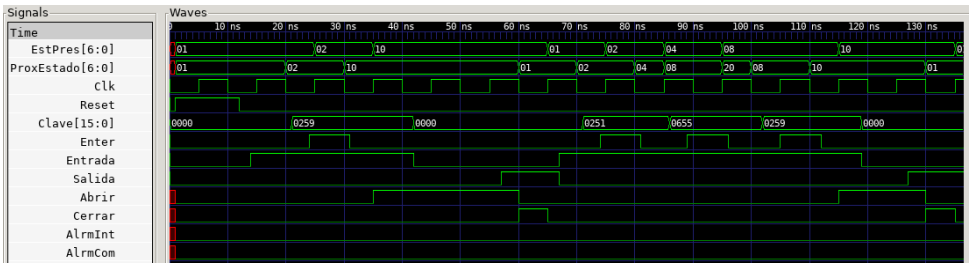

Figura 10: Diagrama de tiempo de pruebas 1 y 2 con módulo sintetizado (CMOS).

5.2.2. Resultados prueba #3 Y #4

Lo mismo ocurre con las pruebas 3 y 4:

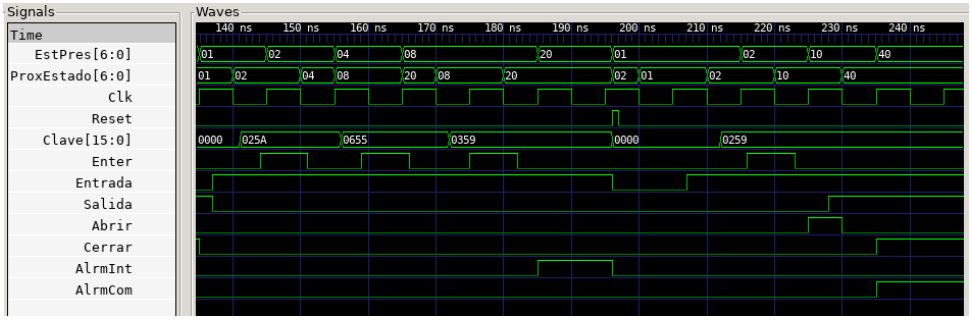

Figura 11: Diagrama de tiempo de pruebas 3 y 4 con módulo sintetizado (CMOS).

5.3. Resultados síntesis CMOS con retardos

Finalmente se tienen los resultados de la sintetización con la librería CMOS al agregarle los retardos, este diagrama es idéntico al realizado en la síntesis de CMOS excepto que los nombres de cada celda se le agrega "_R".

5.4. Evidencias de retardos

Los resultados de las pruebas con los retardos asignados en el Cuadro 1 generan un retardo en las salidas notable según se ve en la Figura 12. En esta se nota que hay una diferencia entre flanco inactivo de reloj y la salida asignada al inicio retardado del estado actual. Esta salida se debió iniciar en el mismo punto que el marcador rojo y se nota una diferencia de 1.101 ns entre ambos (usando el cursor). Esto indica que la lógica combinacional que calcula las salidas y la lectura de entradas/estados es afectada por los retardos de cada celda.

Figura 12: Evidencia de retardo entre marcador rojo y cambio en la salida.

Este mismo se puede notar si se colocan las dos ondas de resultados en Inkscape, con los resultados de la síntesis CMOS y la de CMOS con retardos (semi-transparente):

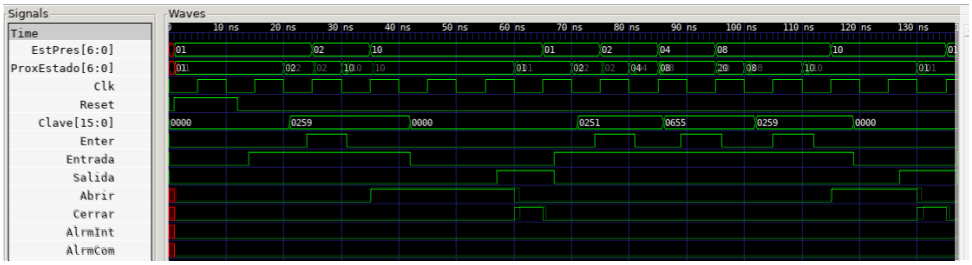

Figura 13: Comparación entre resultados de síntesis CMOS y CMOS con retardos.

Con esto se nota que (aparte del retraso de todas las salidas) el vector de próximo estado también se vio afectado por los retardos, esto pues la lógica combinacional para calcularlo y los flip-flops que actualizan las entradas presentan retardos.

Se notó que si se aumentan los retardos drásticamente se pueden dar dos escenarios: el sistema no es capaz de calcular el próximo estado de forma correcta, causando que se mantenga en un estado por mucho tiempo y no se registren ingresos de la clave; que el sistema sea tan lento que desde el primer reset dado se indefine el próximo estado y no se logre establecer el estado de ninguna señal (como se ve en la Figura 14)

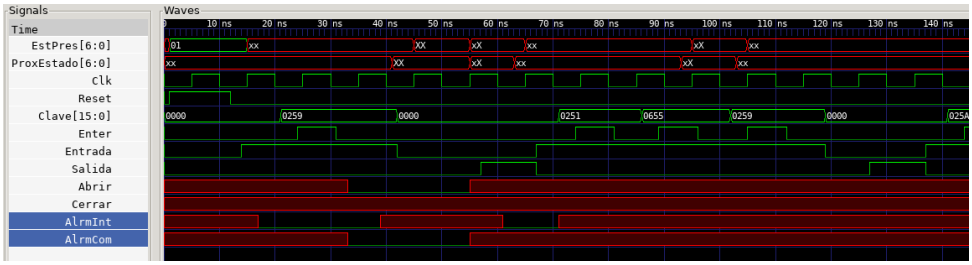

Figura 14: Evidencia de indefinición del sistema causada por retardos altos.

5.4.1. Resultados prueba #1 Y #2

Aún con los retardos escogidos, se lograron pasar las pruebas dadas, esto debido a que los retardos no fueron lo suficientemente altos para afectar las pruebas.

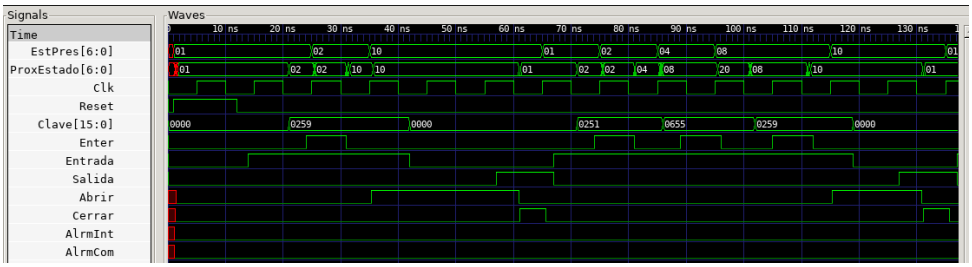

Figura 15: Diagrama de tiempo de pruebas 1 y 2 con módulo sintetizado (CMOS con retardos).

5.4.2. Resultados prueba #3 Y #4

También se cumplen las pruebas 3 y 4, indicando que con los retardos asignados el módulo es capaz de funcionar correctamente. En ambas pruebas se nota que el retardo en el cálculo de proximo estados no afectó, pero que las salidas se encuentran desplazadas y esto podría afectar el funcionamiento real del controlador (posiblemente no mucho).

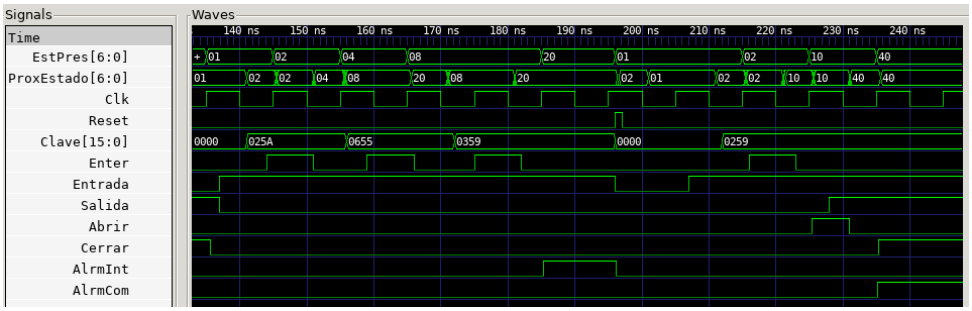

Figura 16: Diagrama de tiempo de pruebas 3 y 4 con módulo sintetizado (CMOS con retardos).

5.5. Componentes finales

Se presena una tabla con el resultado final de componentes usados, cada uno con los retardos asignados:

Cuadro 2: Componentes de síntesis final.

Celda	Cantidad	Retardo (ns)
NOT	20	0.0117
NAND	39	0.05
NOR	34	0.07
DFF	26	0.8

6. Conclusiones y recomendaciones

Durante el proceso de diseño se logró obtener 3 distintos módulos estructurales del diseño conductual original a través de las herramientas de síntesis. Por medio de este proceso de síntesis se concluye que Yosys: no puede sintetizar parámeteros por defecto/cambios de parámetros en módulos; es capaz de obtener diseños estructurales y optimizados partiendo de un diseño conductual, y estos se pueden realizar con la librería estándar o con celdas específicas y personalizables; se puede simular el efecto de los retardos de las celdas dadas a Yosys por medio de la definición de las mismas. Además al no tener que hacer cambios fundamentales al comportamiento del módulo se puede concluir que el diseño realizado en la Tarea 1 fue robusto y sintetizable. Por medio de las pruebas se pudo concluir que los diseños obtenidos cumplen con el comportamiento original del módulo y que al agregar retardos realistas este es capaz de cumplir su función sin problemas. Finalmente se pudo verificar el efecto de los retardos en las máquinas de estado al modificarlos y estudiar sus efectos en las simulaciones.

Referencias

[1] Texas Instruments. *SN74AUC1G79: Low-Voltage Single Positive-Edge-Triggered D-Type Flip-Flop*. Accessed: 2024-09-19. 2023. URL: https://www.ti.com/lit/ds/symlink/sn74auc1g79.pdf?ts=1726760783776&ref_url=https%253A%252F%252Fwww.google.com%252F.

[2] Yosys Development Team. *Cell Library Documentation*. https://yosyshq.readthedocs.io/projects/yosys/en/latest/yosys_internals/formats/cell_library.html. Accessed: 2024-09-18. 2024.

[3] Neil HE Weste y David Harris. *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India, 2015.