

Síntesis de Circuitos Integrados

Gabriel Fernando Araya Mora B80525

Escuela de Ingeniería Eléctrica, Universidad de Costa Rica

Circuitos Digitales II (IE-0523)

1. Tiempo de ejecución de la tarea:

- **Buscar información:** Siempre antes de empezar cualquier trabajo es importante tomarse un tiempo prudente para buscar información e investigar acerca del tema. Esto es especialmente cierto cuando se trata de programar. Para la búsqueda de información, revisé la documentación de yosys acerca de la síntesis y cuales son las mejores prácticas para obtener las descripciones estructurales para los diseños solicitados. Aproximadamente 1 hora me tomó la búsqueda de información.
- **Usando la información:** El problema de esta tarea no es diseñar nada ya que todos los módulos se tienen de tareas anteriores, el problema nace en aprender y entender lo que sea que hace yosys para generar las descripciones estructurales. Este proceso de acomodo del código y la obtención de la síntesis hecha por yosys me tomó aproximadamente 2 horas.
- **Elaboración del reporte:** La elaboración del reporte me tomó aproximadamente una hora.
- **Total:** De lo anterior se saca que el tiempo total para la realización de la tarea es de 4h.

2. Descripción arquitectónica o diagrama del circuito.

2.1. *NAND*

Para la compuerta *AND* se toman:

- $C_L = 15pF$
- $T_A = -40^{\circ}C$ a $125^{\circ}C$
- $t_{PD} \rightarrow MIN = 1ns, Typ = 4.75ns, MAX = 8.5ns$
- $C_{pd} = 9.5pF$
- Precio: \$0.187 por unidad.

switching characteristics over recommended operating free-air temperature range, $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ (unless otherwise noted) (see Figure 1)

| PARAMETER | FROM (INPUT) | TO (OUTPUT) | LOAD CAPACITANCE | T _A = 25°C | | | T _A = -40°C TO 125°C | | UNIT |
|------------------|-----------------|----------------|------------------------|-----------------------|-----|-----|------------------------------------|-----|------|
| | | | | MIN | TYP | MAX | MIN | MAX | |
| t _{PLH} | A or B | Y | C _L = 15 pF | 3.7 | 5.5 | 1 | 8.5 | ns | |
| t _{PHL} | | | | 3.7 | 5.5 | 1 | 8.5 | | |
| t _{PLH} | A or B | Y | C _L = 50 pF | 5.2 | 7.5 | 1 | 10.5 | ns | |
| t _{PHL} | | | | 5.2 | 7.5 | 1 | 10.5 | | |

operating characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$

| PARAMETER | TEST CONDITIONS | TYP | UNIT |
|--|-----------------------------|-----|------|
| C_{pd} Power dissipation capacitance | No load, $f = 1\text{ MHz}$ | 9.5 | pF |

Figura 1: Tiempos de retardo para la compuerta NAND. (Creación Propia)

2.2. NOR

Para la compuerta OR se toman:

- $C_L = 50\text{ pF}$
- $T_A = -40^\circ\text{C}$ a 125°C
- $t_{PLH} = t_{PHL} \rightarrow MIN = 1, Typ = 4.75\text{ ns}, MAX = 8.5$
- $C_{pd} = 15\text{ pF}$
- Precio: \$1.61 por unidad.

Switching Characteristics

over operating free-air temperature range, $V_{CC} = 5 \pm 0.5\text{ V}$ (unless otherwise noted) (see Figure 1)

| PARAMETER | FROM (INPUT) | TO (OUTPUT) | LOAD CAPACITANCE | T _A = 25°C | | | T _A = -55°C TO 125°C | | UNIT |
|------------------|-----------------|----------------|------------------------|-----------------------|-----|-----|---------------------------------|-----|------|
| | | | | MIN | TYP | MAX | MIN | MAX | |
| t _{PLH} | A or B | Y | C _L = 50 pF | 5.1 | 7.5 | 1 | 8.5 | ns | |
| t _{PHL} | | | | 5.1 | 7.5 | 1 | 8.5 | | |

Operating Characteristics

$V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$

| PARAMETER | TEST CONDITIONS | TYP | UNIT |
|--|-----------------------------|-----|------|
| C_{pd} Power dissipation capacitance | No load, $f = 1\text{ MHz}$ | 15 | pF |

Figura 2: Tiempos de retardo para la compuerta NOR. (Creación Propia)

2.3. Flip Flop

Para la compuerta FLIP FLOP se toman:

- $C_L = 30\text{ pF}$
- $T_A = -40^\circ\text{C}$ a 125°C

- $t_{PD} \rightarrow MIN = 1ns, Typ = 3ns, MAX = 5ns$
- $C_{pd} = 30pF$
- Precio: \$0.215 por unidad.

6.10 Switching Characteristics: $C_L = 30\text{ pF}$ or 50 pF , $T_A = -40^\circ\text{C}$ to $+125^\circ\text{C}$

over recommended operating free-air temperature range, $C_L = 30\text{ pF}$ or 50 pF (unless otherwise noted) (see Figure 4)

| PARAMETER | FROM (INPUT) | TO (OUTPUT) | T _A = -40°C to +125°C | | | | | | | | UNIT |
|------------------|-----------------|----------------|-------------------------------------|-----|------------------------------------|-----|------------------------------------|-----|----------------------------------|-----|------|
| | | | V _{CC} = 1.8 V ± 0.15 V | | V _{CC} = 2.5 V ± 0.2 V | | V _{CC} = 3.3 V ± 0.3 V | | V _{CC} = 5 V ± 0.5 V | | |
| | | | MIN | MAX | MIN | MAX | MIN | MAX | MIN | MAX | |
| f _{max} | | | 160 | | 160 | | 160 | | 160 | | MHz |
| t _{pd} | CLK | Q | 3.9 | 12 | 2 | 8.5 | 1.7 | 6 | 1 | 5 | ns |

6.11 Operating Characteristics

$T_A = 25^\circ\text{C}$

| PARAMETER | TEST CONDITIONS | $V_{CC} = 1.8\text{ V}$ | $V_{CC} = 2.5\text{ V}$ | $V_{CC} = 3.3\text{ V}$ | $V_{CC} = 5\text{ V}$ | UNIT |
|--|---------------------|-------------------------|-------------------------|-------------------------|-----------------------|------|
| | | TYP | TYP | TYP | TYP | |
| C_{pd} Power dissipation capacitance | $f = 10\text{ MHz}$ | 26 | 26 | 27 | 30 | pF |

Figura 3: Tiempos de retardo para la compuerta Flip-Flop. (Creación Propia)

2.4. Síntesis Estructural:

En este paso se busca obtener el código verilog de la descripción estructural, pero en lugar de hacerlo a mano, se hará usando el programa de síntesis automática Yosys, el cual utiliza la descripción conductual. Primeramente yosys devuelve un modelo estructural genérico, seguidamente si se le adjunta a los comandos yosys una biblioteca con las especificaciones de una arquitectura especial, el programa va a sintetizar el código estructural con los componentes disponibles en esa arquitectura dada. En este caso se utiliza la más común CMOS. Es importante mencionar que primeramente sin hacer ningún cambio a las los archivos de *cmos_cells.v*, la descripción se debe comportar exactamente igual a la conductual ya que aunque use las compuertas de la arquitectura, estas no tienen retardos. Después se le agregan retardos al archivo y yosys hace la síntesis con dichos retardos.

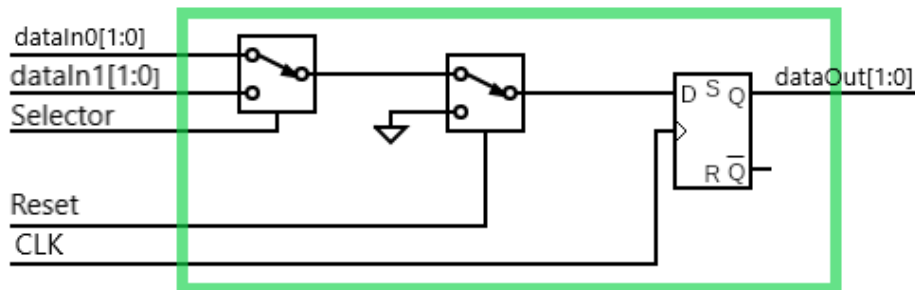


Figura 4: Esquemático de Diseño para la tarea #2 ahora de forma estructural. (Creación Propia)

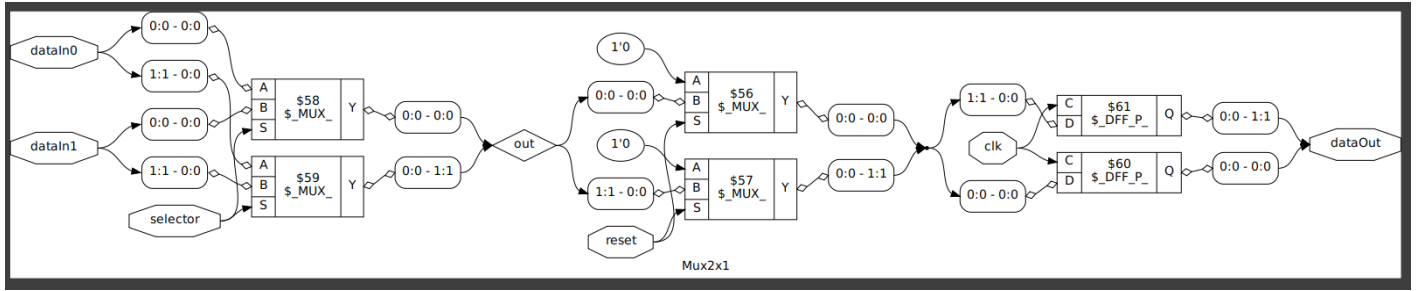


Figura 5: Modelo estructural sintetizado por yosys usando su propia biblioteca. (Creación Propia)

3. Plan de pruebas

Todos los cambios en las señales de entrada se hicieron con base en el archivo mostrado en la tarea #2 dada por el profesor, esto debido a que hace muchos casos posibles y combinaciones de los mismos, lo cual asegura que las descripciones funcionen de manera correcta. En esta tarea no hace falta probar entrada por entrada y revisar salida por salida, ya que como se trabaja con resultados obtenidos de la tarea anterior, basta con compararlos y ver que sean iguales.

1. Primeramente se compara el modulo estructural generado por yosys con el estructural que se sabe que sirve de la tarea anterior, pero sin retardos, esto con el fin de poner en igualdad de condiciones a ambos circuitos.
2. Seguidamente para la segunda descripción que arroja yosys se prueba de igual manera, se compara con el modelo conductual y debería de ser exactamente igual la salida ya que aun no se han agregado retardos.
3. Por último en otro banco de pruebas se jalen las descripciones hechas a mano para poder compararlas con las generadas por yosys, pero ahora agregando retardos al archivo de cmos.

4. Instrucciones para ejecutar el programa.

Estando dentro del directorio **src**. Este makefile fue pensando para ir de la mano con el plan de pruebas por lo que no basta con hacer únicamente make.

1. Si se desea ejecutar el punto 1 del plan de pruebas entonces se debe utilizar el comando:

```
make parte1
```

Este comando debe mostrar en gtk wave cómo el modelo estructural generado por yosys se comporta exactamente igual que el conductual de la tarea #2. Además va a mostrar el esquemático que arroja yosys.

2. Si se desea ejecutar el punto 2 del plan de pruebas entonces se debe utilizar el comando:

```
make parte2
```

Este comando debe mostrar en gtk wave cómo el modelo estructural sin retardos utilizando la arquitectura CMOS generado por yosys se comporta exactamente igual que el conductual de la tarea #2. Además va a mostrar el esquemático que arroja yosys. (Favor cuando abra el gtkwave no escoger la señal con el nombre de dataOutMan, esta se usa en la siguiente sección.)

3. Si se desea ejecutar el punto 3 del plan de pruebas entonces se debe utilizar el comando:

```
make parte3
```

Este comando debe mostrar en gtk wave cómo el modelo estructural con retardos utilizando la arquitectura CMOS generado por yosys se atrasa comparado al conductual de la tarea #2 y se compara la salida obtenida de la descripción estructural en la tarea #3 y #4 con la dada por yosys(dataOutMan). Además va a mostrar el esquemático que arroja yosys.

```
all: parte1 parte2 parte3
    echo fin

parte1:
    \$(MAKE) build1 -C YOSYS
    sed -i 's/Mux2x1/Mux2x1_Estructural_Sintetizado/' sintetizado.v
    iverilog BancodePruebas.v
    ./a.out
    rm a.out
    gtkwave tarea05.vcd

parte2:
    \$(MAKE) build2 -C YOSYS
    sed -i 's/Mux2x1/Mux2x1_Estructural_Sintetizado/' sintetizadoReal.v
    iverilog -T typ BancodePruebasReal.v YOSYS/cmos_cells2.v
    ./a.out
    rm a.out
    gtkwave tarea05.vcd

parte3:
    \$(MAKE) build2 -C YOSYS
    sed -i 's/Mux2x1/Mux2x1_Estructural_Sintetizado/' sintetizadoReal.v
    iverilog -T typ BancodePruebasReal.v YOSYS/cmos_cells.v
    ./a.out
    rm a.out
    gtkwave tarea05.vcd
```

5. Ejemplos de los Resultados.

5.1. Inciso A y B:

En la siguiente figura se ve como claramente el modelo que da yosys bajo su biblioteca estándar es identico al estructural que se quería en la tarea #3 y #4 a un nivel un poco mayor de abstracción ya que no se ven compuertas sino los muxes y

flops, pero si se ven los dos muxes en paralelo para formar el bus de 2 bits.

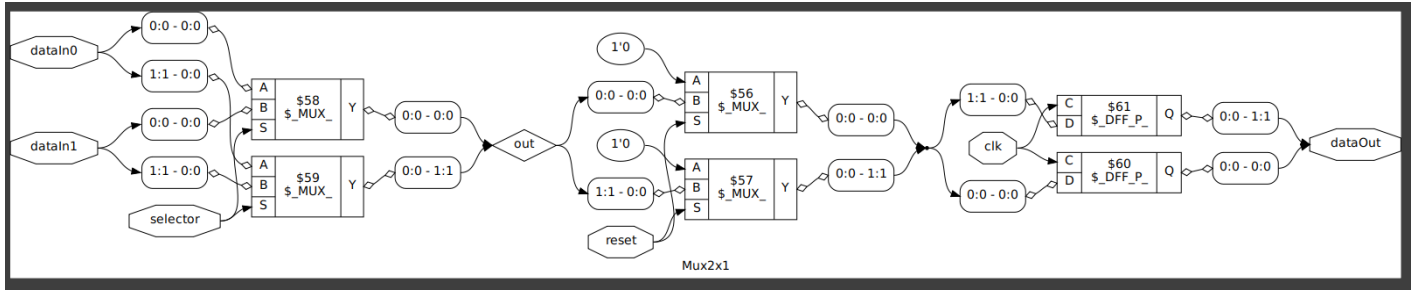


Figura 6: Modelo estructural sintetizado por yosys usando su propia biblioteca. (Creación Propia)

Como era de esperarse, si se compara la descripción estructural dada por yosys y la biblioteca estándar con la conductual hecha para la tarea #2, son iguales ya que no hay retardos y además quiere decir que la síntesis genérica se logro de manera correcta.

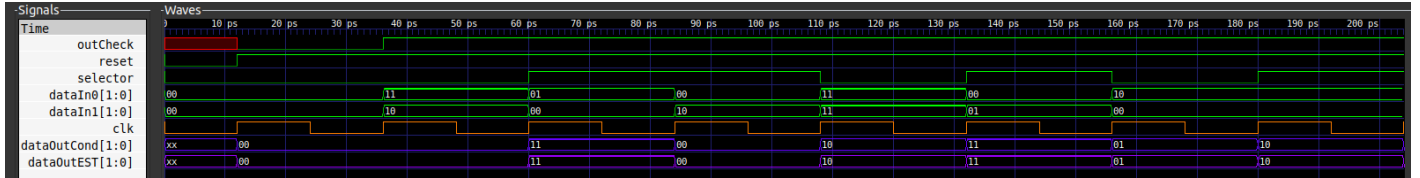


Figura 7: Prueba #1 (Creación Propia)

5.2. Inciso C y D:

En la siguiente figura se muestra como la síntesis hecha en CMOS es correcta, como se ve la descripción conductual es igual a la descripción en CMOS sin retardos. Por lo que entonces ya se pueden tomar en cuenta los retardos para la aproximación del sistema real.



Figura 8: Prueba #2 (Creación Propia)

En la siguiente figura se muestra como yosys hace la síntesis usando la librería de CMOS, la cual como se ve en la figura solo tiene compuertas de tipo *NOR*, *NAND*, *BUF* y *DFE*.

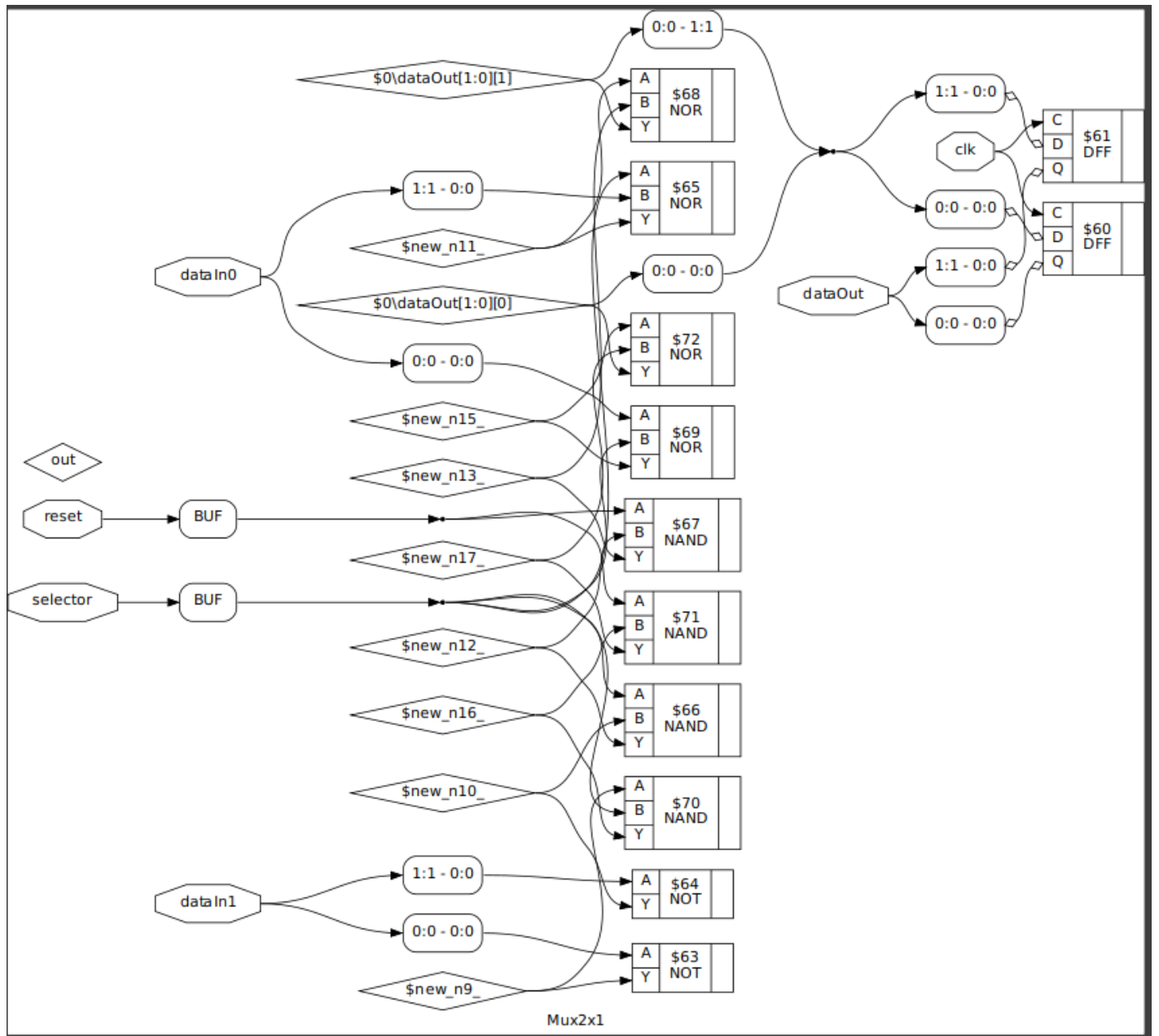


Figura 9: Modelo estructural sintetizado por yosys usando arquitectura CMOS. (Creación Propia)

5.3. Inciso E y H:

En la siguiente figura se resume todo el proyecto hasta este momento, ya que se puede ver el modelo conductual de color azul y se ve observa cómo se ajusta a los flancos positivos del reloj. De color morado se tiene la salida estructural generada por yosys con arquitectura CMOS con retardos, lo que indica que está bien es que su salida es la misma que la conductual pero un poco atrasada. Por último de color rosado se tiene la descripción estructural hecha a mano en la tarea #2 y es aún más lenta que la generada por yosys. Esto lo que indica es que la descripción generada por yosys soporta una mayor frecuencia antes de quebrarse. Esta comparación se hace flanco a flanco, es decir se sabe de antemano que la salida

del conductual va a estar alineada con el flanco del reloj, pero desde ese instante la descripción estructural tiene hasta un instante antes del próximo flanco positivo de reloj para llegar, de no ser así la frecuencia es muy alta y se quiebra el sistema.

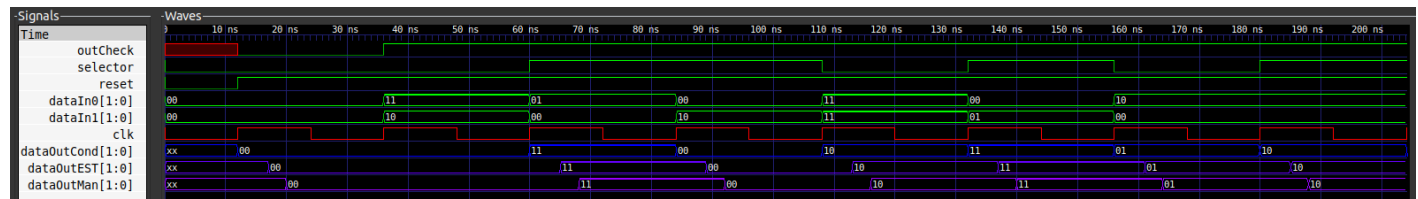


Figura 10: Prueba #3 (Creación Propia)

5.4. Inciso F:

| | Cantidad de compuertas | Precio |
|--------|------------------------|---------|
| NAND | 4 | \$0.187 |
| NOR | 4 | \$1.61 |
| NOT | 2 | \$0.187 |
| FLOP | 2 | \$0.215 |
| Total: | 12 | \$8.0 |

En la siguiente figura se observa como a la frecuencia de $87MHz$ el sistema se quiebra y a las salidas ya no les da tiempo de alcanzar el flanco positivo, además al no ser iguales indica que algo dentro del bloque estructural no le está dando el debido tiempo de asegurar la señal correcta.

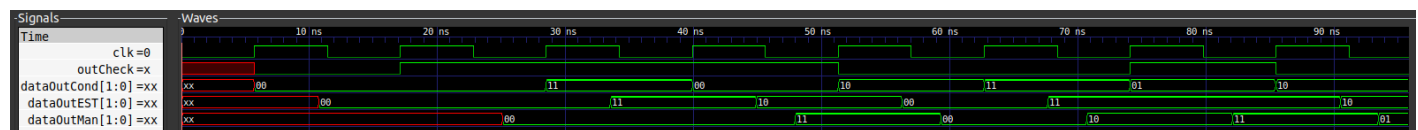


Figura 11: Prueba en frecuencia más alta para quebrar el sistema. (Creación Propia)

6. Análisis y conclusiones:

Esta tarea pone en evidencia lo poderoso que puede llegar a ser un software de síntesis como yosys, esto debido a que se ahorra mucho tiempo si se tiene el modelo conductual correcto. No es necesario ir a buscar todas las compuertas lógicas que forman el sistema, sino que con solo las que se encuentran disponibles en la arquitectura deseada. Por lo que trabajar con yosys no solo es mas eficiente en cuanto a tiempo invertido, sino que el se ve claramente en las pruebas que el módulo sintetizado por yosys es mucho más eficiente que el que se hizo a mano.