

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

IE0411 - Microelectrónica: Sistemas en Silicio I

Tarea 3

Prof. Javier Pacheco Brito

Por:
Marco Chacon Soto, B61868

2 de noviembre de 2020

Índice

1. Introducción	3
2. Resultados Obtenidos	3
2.1. Pruebas con retardos default y diseño original (2 input XOR)	3
2.2. Pruebas con retardos = 1 y diseño original (2 input XOR)	7
2.3. Pruebas con retardos default y diseño modificado (3 input XOR)	10
3. Repositorio	13
4. Conclusiones y recomendaciones	13

Índice de figuras

1.	Funcionamiento del sumador de rizado, con retardos default y diseño original (2 input XOR)	3
2.	Resultados de las pruebas para A=\$00 y B=\$00 con retardos default y diseño original (2 input XOR)	5
3.	Resultados de las pruebas para A=\$00 y B=\$01 con retardos default y diseño original (2 input XOR)	6
4.	Resultados de las pruebas para A=\$FF y B=\$01 con retardos default y diseño original (2 input XOR)	6
5.	Resultados de las pruebas para A=\$00 y B=\$00 con retardos = 1 y diseño original (2 input XOR)	9
6.	Resultados de las pruebas para A=\$00 y B=\$01 con retardos = 1 y diseño original (2 input XOR)	9
7.	Resultados de las pruebas para A=\$FF y B=\$01 con retardos = 1 y diseño original (2 input XOR)	10
8.	Resultados de las pruebas para A=\$00 y B=\$00 con retardos default y diseño modificado (3 input XOR)	12
9.	Resultados de las pruebas para A=\$00 y B=\$01 con retardos default y diseño modificado (3 input XOR)	12
10.	Resultados de las pruebas para A=\$FF y B=\$01 con retardos default y diseño modificado (3 input XOR)	13

1. Introducción

El siguiente reporte contiene los resultados obtenidos de la implementación de un sumador completo y un sumador rizado. Dichos resultados se comparan con los obtenidos para sumadores Lógicos y Lookahead con el fin de analizar las características de los diseños y cómo al cambiar las semillas y la cantidad de pruebas, cambia PwrC, el cuál es un indicador de cuántas veces se tiene un 1 en la salida de algún componente de los sumadores al realizar una suma.

Se incluyen también las observaciones de los resultados obtenidos, el repositorio del git donde se encuentran los archivos y las recomendaciones.

2. Resultados Obtenidos

Las pruebas realizadas se corrieron con 3 semillas: 10, 20 y 40. Los casos particulares (A=\$00 B=\$00, A=\$00 B=\$01 y A=\$FF B=\$01) se utilizaron como semilla inicial al menos una vez para todos los distintos diseños utilizados como se muestra a continuación.

El funcionamiento del sumador de rizado se muestra en la siguiente figura



Figura 1: Funcionamiento del sumador de rizado, con retardos default y diseño original (2 input XOR)

2.1. Pruebas con retardos default y diseño original (2 input XOR)

Para esta sección en las 3 pruebas con 500 sumas se utilizaron como semillas iniciales A=\$00 B=\$00, para las 3 pruebas con 1000 sumas se utilizaron los valores A=\$00 B=\$01 y para las 3 pruebas con 2000 sumas se utilizaron A=\$FF B=\$01.

En la siguiente tabla se muestran los resultados para las primeras pruebas

Número de sumas	Sumador	Seed	Potencia (PwrC)
500	Lógico	10	585060
500	Lógico	20	583920
500	Lógico	40	583920
500	Lookahead	10	509520
500	Lookahead	20	515400
500	Lookahead	40	514140
500	Rizado	10	501240
500	Rizado	20	513000
500	Rizado	40	510300
1000	Lógico	10	1174860
1000	Lógico	20	1164540
1000	Lógico	40	1162740
1000	Lookahead	10	1023480
1000	Lookahead	20	1010400
1000	Lookahead	40	1019400
1000	Rizado	10	1022460
1000	Rizado	20	1018500
1000	Rizado	40	1017960
2000	Lógico	10	2341260
2000	Lógico	20	2313720
2000	Lógico	40	2319840
2000	Lookahead	10	2041200
2000	Lookahead	20	2002920
2000	Lookahead	40	2035200
2000	Rizado	10	2040420
2000	Rizado	20	2014320
2000	Rizado	40	2026620
5000	Lógico	10	5821620
5000	Lógico	20	5773140
5000	Lógico	40	5771940
5000	Lookahead	10	5030520
5000	Lookahead	20	4959000
5000	Lookahead	40	5010540
5000	Rizado	10	5034300
5000	Rizado	20	5010660
5000	Rizado	40	5036460

Tabla 1: Resultados de las pruebas con retardos default y diseño original (2 input XOR)

En la tabla se observa que el sumador Lógico es el que más potencia (PwrC) consume, esto está directamente relacionado con la cantidad de compuertas que este sumador tiene. Si se observan los diseños de los tres sumadores utilizados, el sumador lógico es el que utiliza la mayor cantidad de compuertas y como la potencia (PwrC) es un indicador de cuántos componentes ponen en 1 su salida al realizar una suma, es de esperar que este sumador utilice más componentes para una suma que

los demás.

También se observa que la potencia (PwrC) aumenta conforme aumentan la cantidad de sumas realizadas, esto es de esperar por supuesto ya que al incrementar la cantidad de sumas, se incrementa la cantidad de veces que se hace uso de los componentes de cada sumador.

A continuación se muestran capturas de los tres casos particulares (A=\$00 B=\$00, A=\$00 B=\$01 y A=\$FF B=\$01)

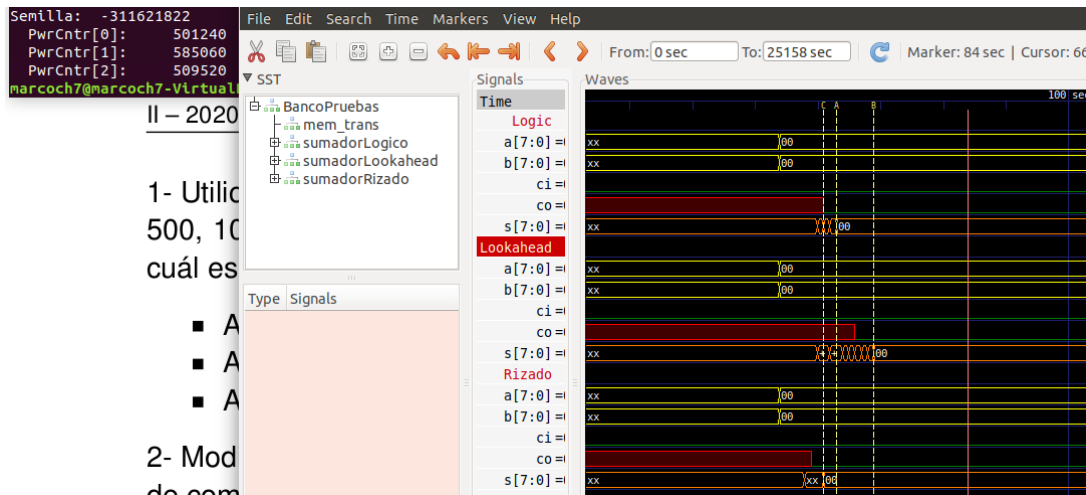


Figura 2: Resultados de las pruebas para A=\$00 y B=\$00 con retardos default y diseño original (2 input XOR)

Para el caso particular con A=\$00 y B=\$00 se espera obtener el menor retardo ya que se espera que los sumadores utilicen la menor cantidad de componentes para realizar esta suma.

Los tiempos que toman cada sumador para realizar esta suma fueron los siguientes

- **Lógico:** 63 s
- **Lookahead:** 69 s
- **Rizado:** 61 s

El sumador con mayor retardo es el de Lookahead, esto se puede justificar con el hecho de que el diseño de este sumador hace uso de compuertas con mayor retardo que los otros como lo son xor3_p, or4_p y or5_p. También se espera que el sumador de Rizado tenga el menor retardo ya que utiliza pocas compuertas y las mismas tienen retardos pequeños.

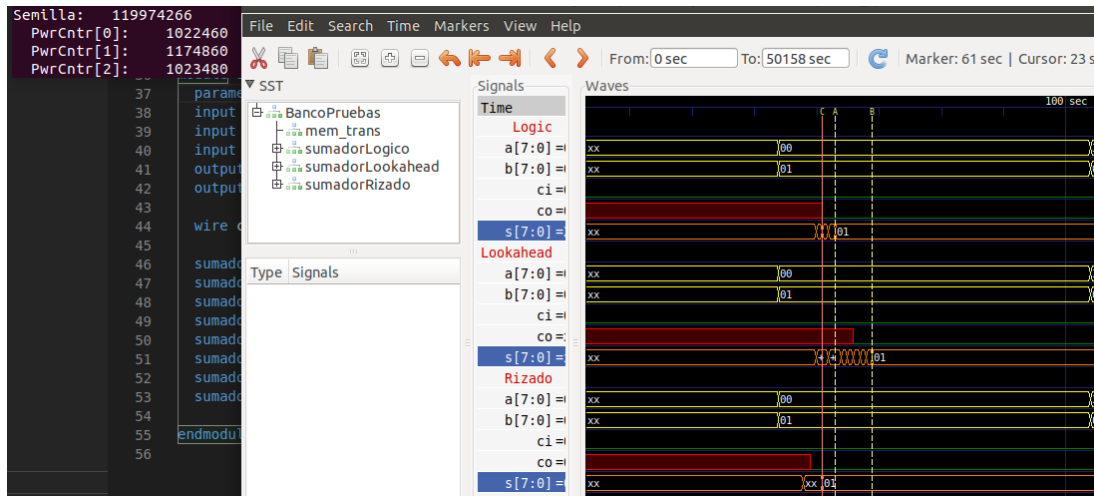


Figura 3: Resultados de las pruebas para A=\$00 y B=\$01 con retardos default y diseño original (2 input XOR)

Nuevamente se utilizan valores iniciales muy pequeños, por lo cual se espera que el retardo obtenido sea muy pequeño, similar al obtenido en la figura 8.

Los tiempos que toman cada sumador para realizar esta suma fueron los siguientes

- **Lógico:** 63 s
- **Lookahead:** 69 s
- **Rizado:** 61 s

Se observa que los retardos obtenidos son los mismos que para el caso anterior, esto se debe a que el único cambio que hubo fue el de la entrada B que ahora tiene un valor de 1, sin embargo este cambio no fue muy significativo en cuanto al uso de compuertas en los sumadores y por lo tanto los retardos se mantuvieron iguales.

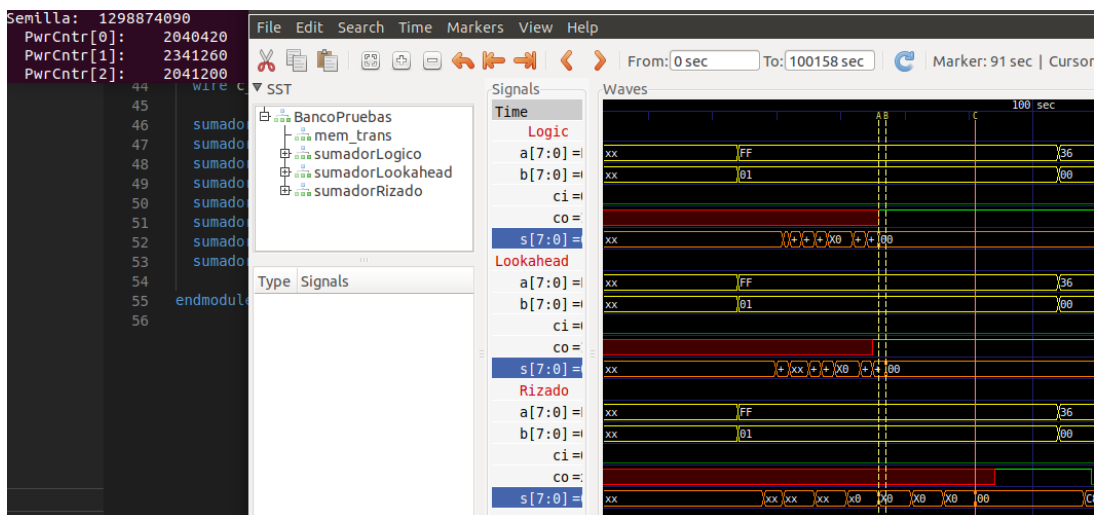


Figura 4: Resultados de las pruebas para A=\$FF y B=\$01 con retardos default y diseño original (2 input XOR)

Finalmente se obtienen los retardos para un caso mucho peor que los anteriores, ya que esta vez se tiene un valor de FF en la entrada A, por ende se espera que los sumadores utilicen todos sus componentes para realizar esta suma y por lo tanto haya un retardo mayor en cada uno. Los tiempos que toman cada sumador para realizar esta suma fueron los siguientes

- **Lógico:** 76 s
- **Lookahead:** 77 s
- **Rizado:** 91 s

Se observa que para este caso particular el sumador de Rizado tiene el mayor retardo, lo cuál indica que cada componente del sumador completo utilizado para realizar el sumador de Rizado es usado de manera exhaustiva lo cuál incrementa tanto su retardo.

2.2. Pruebas con retardos = 1 y diseño original (2 input XOR)

Para esta sección se realizaron las mismas pruebas pero esta vez modificando el archivo de definiciones para utilizar el mismo retardo en todas las compuertas (1).

Número de sumas	Sumador	Seed	Potencia (PwrC)
500	Lógico	10	631500
500	Lógico	20	629640
500	Lógico	40	627780
500	Lookahead	10	554640
500	Lookahead	20	565980
500	Lookahead	40	564960
500	Rizado	10	501240
500	Rizado	20	513000
500	Rizado	40	510300
1000	Lógico	10	1270980
1000	Lógico	20	1253460
1000	Lógico	40	1257300
1000	Lookahead	10	1119240
1000	Lookahead	20	1109820
1000	Lookahead	40	1116780
1000	Rizado	10	1022460
1000	Rizado	20	1018500
1000	Rizado	40	1017960
2000	Lógico	10	2536620
2000	Lógico	20	2500800
2000	Lógico	40	2508840
2000	Lookahead	10	2237580
2000	Lookahead	20	2201160
2000	Lookahead	40	2227140
2000	Rizado	10	2040420
2000	Rizado	20	2014320
2000	Rizado	40	2026620
5000	Lógico	10	6291120
5000	Lógico	20	6253560
5000	Lógico	40	6266700
5000	Lookahead	10	5506080
5000	Lookahead	20	5447400
5000	Lookahead	40	5475660
5000	Rizado	10	5034300
5000	Rizado	20	5010660
5000	Rizado	40	5036460

Tabla 2: Resultados de las pruebas con retardos = 1 y diseño original (2 input XOR)

Al cambiar los retardos de las compuertas a 1 se observa que el sumador de rizado presenta los mismos valores de PwrC, esto se debe a que cuando se realiza una suma, sin importar los retardos, se hacen la misma cantidad de transiciones (cada sumador completo realiza la suma de un bit y el bit de acarreo se pasa cuando esta suma haya terminado). Por otra parte, el PwrC para los sumadores de Lookahead y Lógico aumenta significativamente, esto puede ser que ocurra debido a un aumento

en la cantidad de transiciones al disminuir el retardo en las compuertas.

A continuación se muestran las capturas de los resultados para los casos particulares

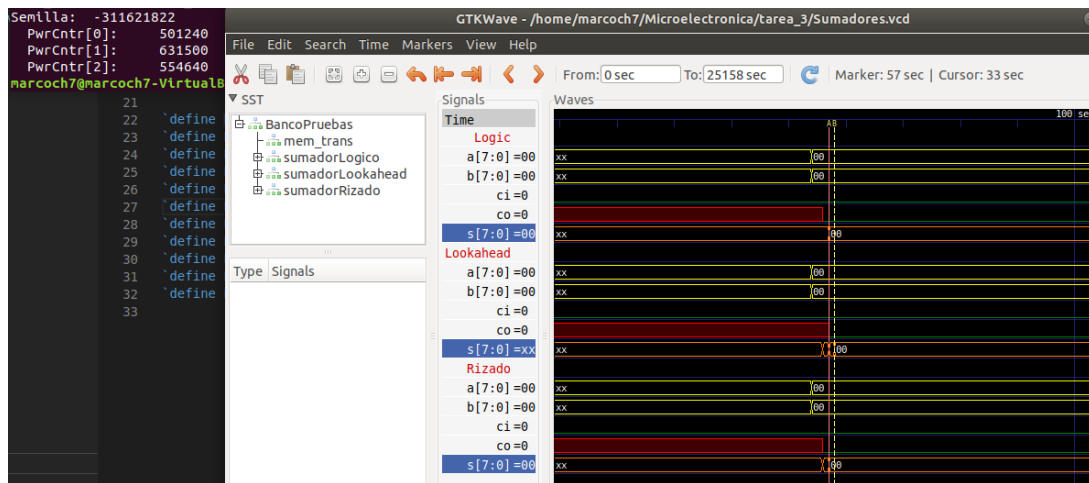


Figura 5: Resultados de las pruebas para A=\$00 y B=\$00 con retardos = 1 y diseño original (2 input XOR)

Los tiempos que toman cada sumador para realizar esta suma fueron los siguientes

- **Lógico:** 57 s
- **Lookahead:** 58 s
- **Rizado:** 57 s

Lo primero que se observa es que el retardo de todos los sumadores mejora respecto a la prueba anterior. También se observa que el sumador Lógico alcanza el mismo retardo que el sumador de rizado al reducir el delay en las compuertas.

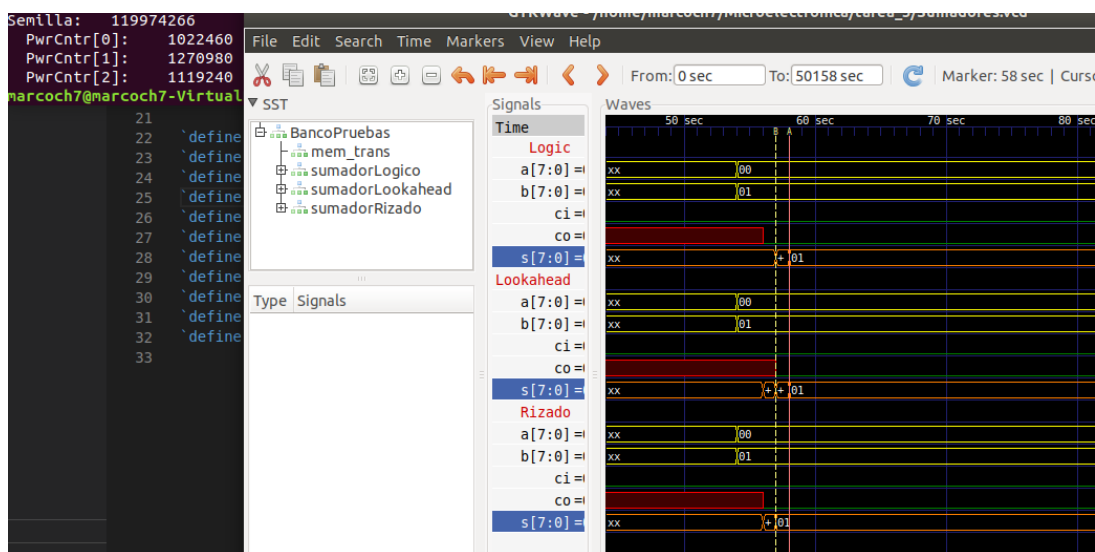


Figura 6: Resultados de las pruebas para A=\$00 y B=\$01 con retardos = 1 y diseño original (2 input XOR)

Los tiempos que toman cada sumador para realizar esta suma fueron los siguientes

- **Lógico:** 58 s
- **Lookahead:** 58 s
- **Rizado:** 57 s

Se observa que al utilizar un dato mayor en la entrada B se obtiene un mayor retardo en el sumador Lógico respecto al anterior, sin embargo, los valores obtenidos son muy buenos ya que presentan una reducción de al menos 4 s con respecto a los obtenidos en la prueba 1 (Rizado 61 s -> 57 s)

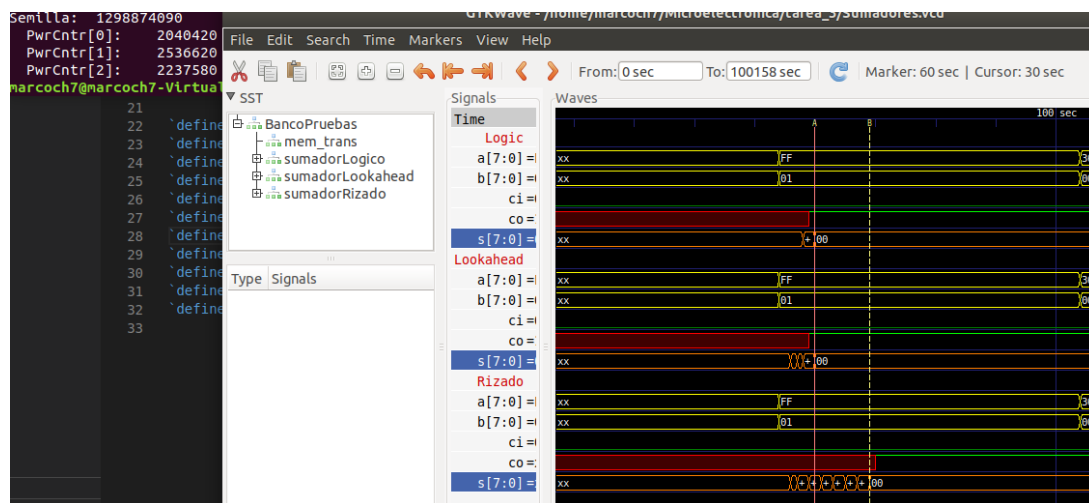


Figura 7: Resultados de las pruebas para A=\$FF y B=\$01 con retardos = 1 y diseño original (2 input XOR)

Los tiempos que toman cada sumador para realizar esta suma fueron los siguientes

- **Lógico:** 60 s
- **Lookahead:** 60 s
- **Rizado:** 69 s

Se esperan retardos mayores para este caso ya que se utilizan los componentes que llevan el acarreo. Se observa que los retardos de todos los sumadores disminuyen significativamente, teniendo el mayor cambio en el sumador de Rizado con 22 s menos que en la prueba 1. Se nota también que el sumador de Rizado se mantiene como el más lento mientras que los de Lookahead y Lógico presentan un mejor rendimiento para sumas con acarreo.

2.3. Pruebas con retardos default y diseño modificado (3 input XOR)

Para esta sección se realizaron las mismas pruebas pero esta vez modificando el archivo sumador completo al sustituir las dos XOR de dos entradas por una de tres entradas.

La siguiente tabla contiene los resultados obtenidos para el PwrC

Número de sumas	Sumador	Seed	Potencia (PwrC)
500	Lógico	10	585060
500	Lógico	20	583920
500	Lógico	40	583920
500	Lookahead	10	509520
500	Lookahead	20	515400
500	Lookahead	40	514140
500	Rizado	10	569160
500	Rizado	20	581400
500	Rizado	40	579240
1000	Lógico	10	1174860
1000	Lógico	20	1164540
1000	Lógico	40	1162740
1000	Lookahead	10	1023480
1000	Lookahead	20	1010400
1000	Lookahead	40	1019400
1000	Rizado	10	1159800
1000	Rizado	20	1154520
1000	Rizado	40	1154760
2000	Lógico	10	2341260
2000	Lógico	20	2313720
2000	Lógico	40	2319840
2000	Lookahead	10	2041200
2000	Lookahead	20	2002920
2000	Lookahead	40	2035200
2000	Rizado	10	2316360
2000	Rizado	20	2284440
2000	Rizado	40	2298240
5000	Lógico	10	5821620
5000	Lógico	20	5773140
5000	Lógico	40	5771940
5000	Lookahead	10	5030520
5000	Lookahead	20	4959000
5000	Lookahead	40	5010540
5000	Rizado	10	5715480
5000	Rizado	20	5683800
5000	Rizado	40	5710800

Tabla 3: Resultados de las pruebas con retardos default y diseño modificado (3 input XOR)

Lo primero que se observa es que al hacer la sustitución de compuertas en el sumador de Rizado, el PwrC aumenta respecto a las pruebas anteriores. Este cambio de compuertas puede presentar un beneficio en términos de área y velocidad en el diseño, ya que se están eliminando 8 compuertas en el sumador de rizado, sin embargo en términos de potencia este se puede comparar con el sumador Lógico, mientras que el sumador Lookahead ahora presenta el menor consumo de potencia. Como es

de esperar, el PwrC de los sumadores Lookahead y Lógico se mantiene igual que en la prueba 1 ya que se utilizan las mismas semillas y no se hacen modificaciones en los diseños de estos.

A continuación se muestran capturas de los casos particulares

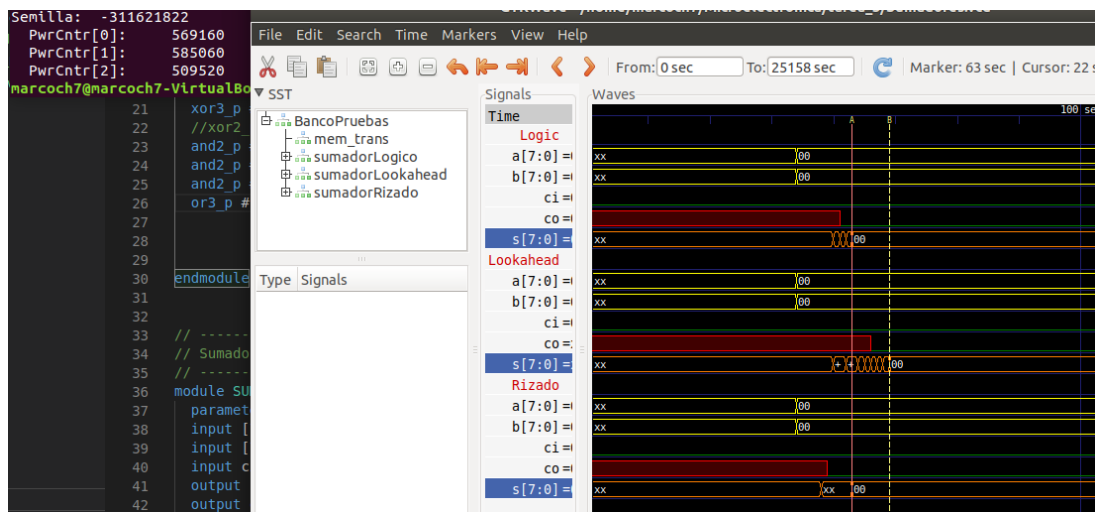


Figura 8: Resultados de las pruebas para A=\$00 y B=\$00 con retardos default y diseño modificado (3 input XOR)

Los tiempos que toman cada sumador para realizar esta suma fueron los siguientes

- **Lógico:** 63 s
- **Lookahead:** 69 s
- **Rizado:** 63 s

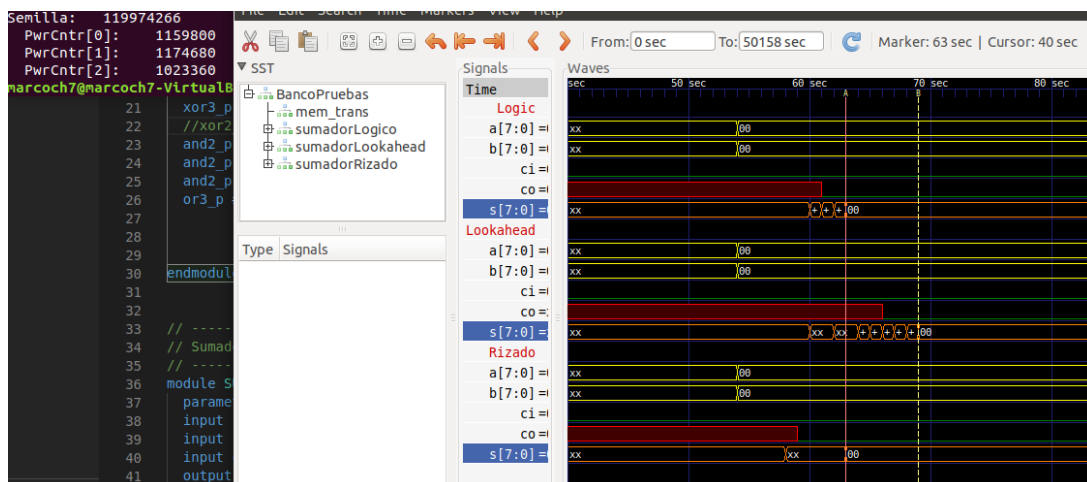


Figura 9: Resultados de las pruebas para A=\$00 y B=\$01 con retardos default y diseño modificado (3 input XOR)

Los tiempos que toman cada sumador para realizar esta suma fueron los siguientes

- **Lógico:** 63 s

- Lookahead: 69 s
- Rizado: 63 s

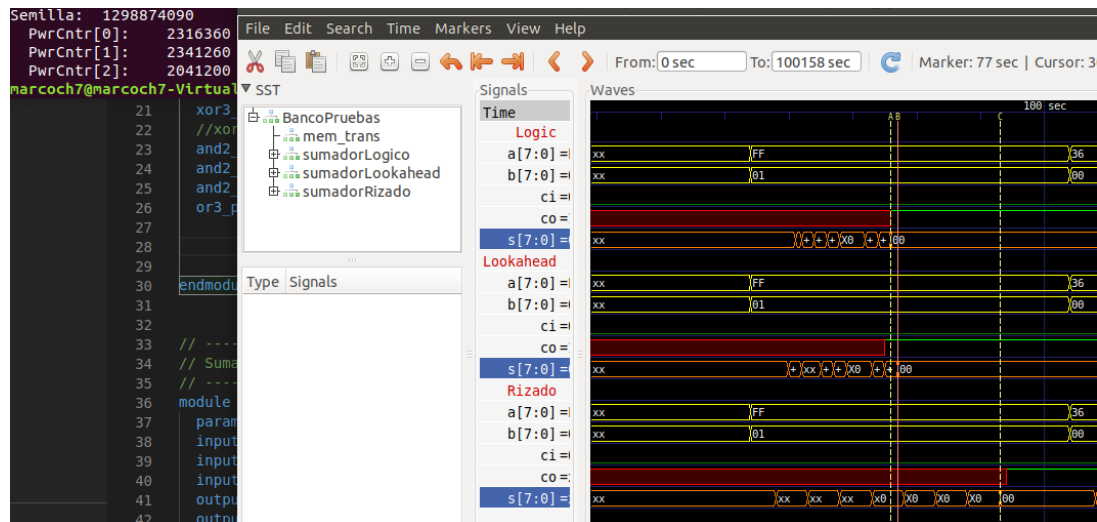


Figura 10: Resultados de las pruebas para $A=\$FF$ y $B=\$01$ con retardos default y diseño modificado (3 input XOR)

Los tiempos que toman cada sumador para realizar esta suma fueron los siguientes

- Lógico: 76 s
- Lookahead: 77 s
- Rizado: 93 s

Se observa que al modificar el diseño del sumador de Rizado se empeora el tiempo de retardo para las sumas con y sin acarreo, por lo cuál el único posible beneficio que se obtiene de realizar este cambio sería una disminución en el área del dispositivo.

3. Repositorio

El firmware del trabajo se encuentra disponible en el siguiente [repositorio](#).

4. Conclusiones y recomendaciones

A partir del análisis de resultados se llega a las siguientes conclusiones

- Se logra comparar el consumo de potencia de tres contadores distintos ante diferentes semillas y con distintos retardos de compuertas
- Reducir el retardo en las compuertas de los sumadores disminuye los retardos de los contadores al permitir que las transiciones se realicen con mayor rapidez

- Eliminar compuertas en el diseño de los sumadores puede mejorar el área del dispositivo, pero a la vez puede ocasionar que haya un mayor consumo de potencia y que la frecuencia del dispositivo disminuya.
- El sumador de Lookahead tiene un retardo similar para sumas con y sin acarreo, dicho retardo es mayor que el de los otros sumadores para sumas sin acarreo y es el segundo menor para sumas con acarreo. Además tiene un consumo de potencia bajo similar al del sumador de Rizado original.
- El sumador Lógico tiene un retardo pequeño tanto para sumas con acarreo como para sumas sin acarreo, sin embargo presenta el mayor consumo de potencia de los tres sumadores.
- El sumador de Rizado con diseño original presenta el menor retardo para sumas sin acarreo pero tiene el mayor retardo para sumas con acarreo. En cuanto al consumo de potencia, este sumador presenta el consumo más bajo respecto a los otros.
- El sumador de Rizado con XOR de tres entradas presenta un consumo de potencia muy elevado así como un retardo un poco mayor al del diseño original. Este cambio implica una reducción en el área del dispositivo.

También se recomienda lo siguiente

- Contar con archivos de texto para realizar la recolección de información después de cada prueba.
- Trabajar de forma ordenada mediante un repositorio de git
- Utilizar distintos archivos para las pruebas con retardos y diseños modificados
- Automatizar las pruebas mediante un makefile