

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE0523 – Circuitos Digitales II

I ciclo 2023

Tarea 01:

Sumador de 4 Bits

Andrés Chaves Vargas - B92198

Grupo 01

Profesor: Enrique Coen

Abril 2023

Índice

1. Descripción Arquitectónica.	1
2. Plan de Pruebas.	4
3. Instrucciones de Utilización de la Simulación.	7
4. Resultados.	8
4.1. Simulación Realizada al Sumador de 4 Bits.	8
4.2. Simulación Realizada al Sumador de 8 Bits.	8
4.3. Simulación Realizada al Sumador de 32 Bits.	9
5. Conclusiones.	10
6. Recomendaciones.	10

Resumen

En esta tarea se implementó mediante el uso del lenguaje de descripción de hardware (HDL) Verilog, el cuál permite realizar diseños que permitan realizar pruebas que simulan el comportamiento de un circuito digital, trabajando en conjunto con la herramienta GTKwave. La idea del diseño general es obtener un sumador de 32 bits instanciando sumadores más pequeños de 8 y 4 bits. Mediante la asignación de valores en las entradas A y B, permite obtener un resultado que depende del estado en que se encuentre la entrada MODO, la cuál es de gran importancia debido a que controla por completo el sumador utilizado.

Del diseño planteado se obtuvo un excelente comportamiento del sumador de 4 bits, mientras que respecto al sumador de 8 y 32 bits se generaron errores al realizar la operación resta en condiciones específicas. Como conclusiones de esta tarea se tiene que posiblemente existió un error humano al realizar el diseño, lo que provocó los errores mencionados anteriormente, además, es posible que en condiciones diferentes a las pruebas realizadas para el sumador de 32 bits, se tenga un error similar al obtenido para el de 8 bits.

1. Descripción Arquitectónica.

A continuación se muestran los diagramas utilizados para cada sumador:

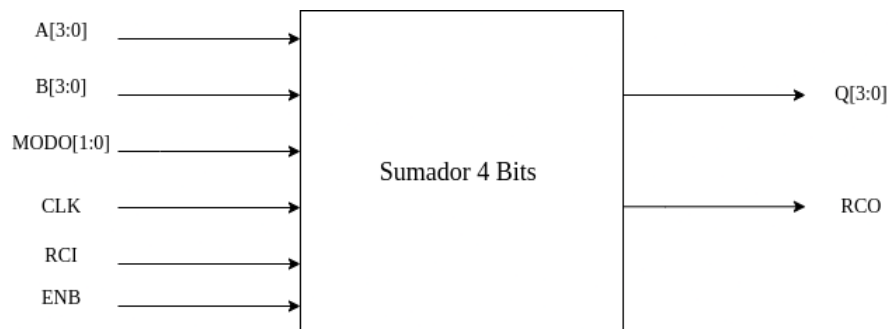


Figura 1: Diagrama Sumador 4 Bits.

En la figura anterior se muestra el bloque principal que permite realizar las operaciones esperadas.

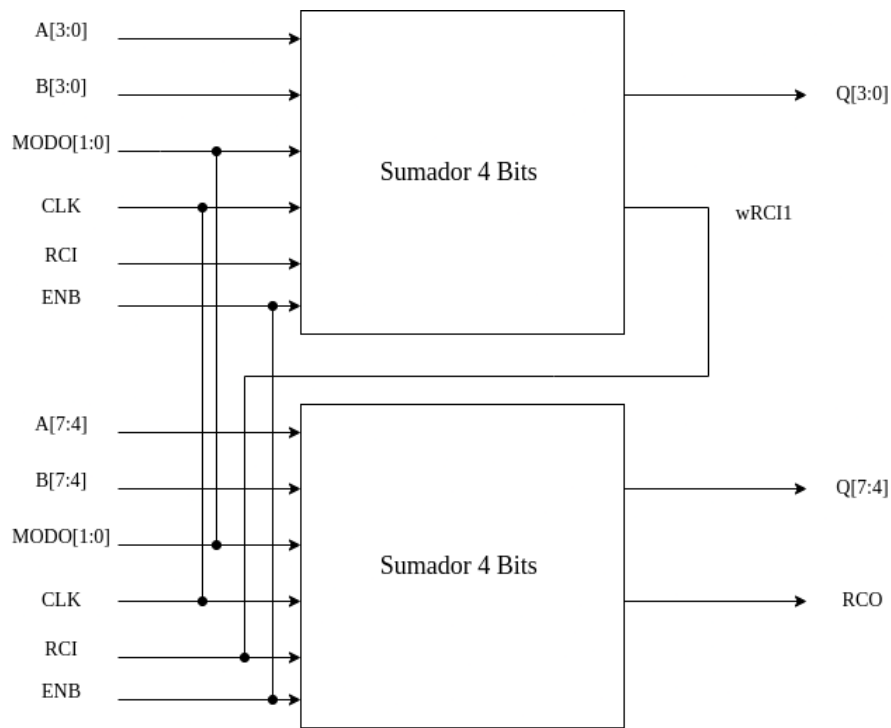


Figura 2: Diagrama Sumador 8 Bits.

En la figura anterior se muestra el diagrama utilizado para el simulador de 8 bits, se muestra que se crea instanciando dos veces el bloque de 4 bits mencionado anteriormente. Además, se puede observar que existe una conexión entre gracias al cable $wRCI1$ y que se tienen las entradas $MODO$, CLK y ENB conectadas entre sí.

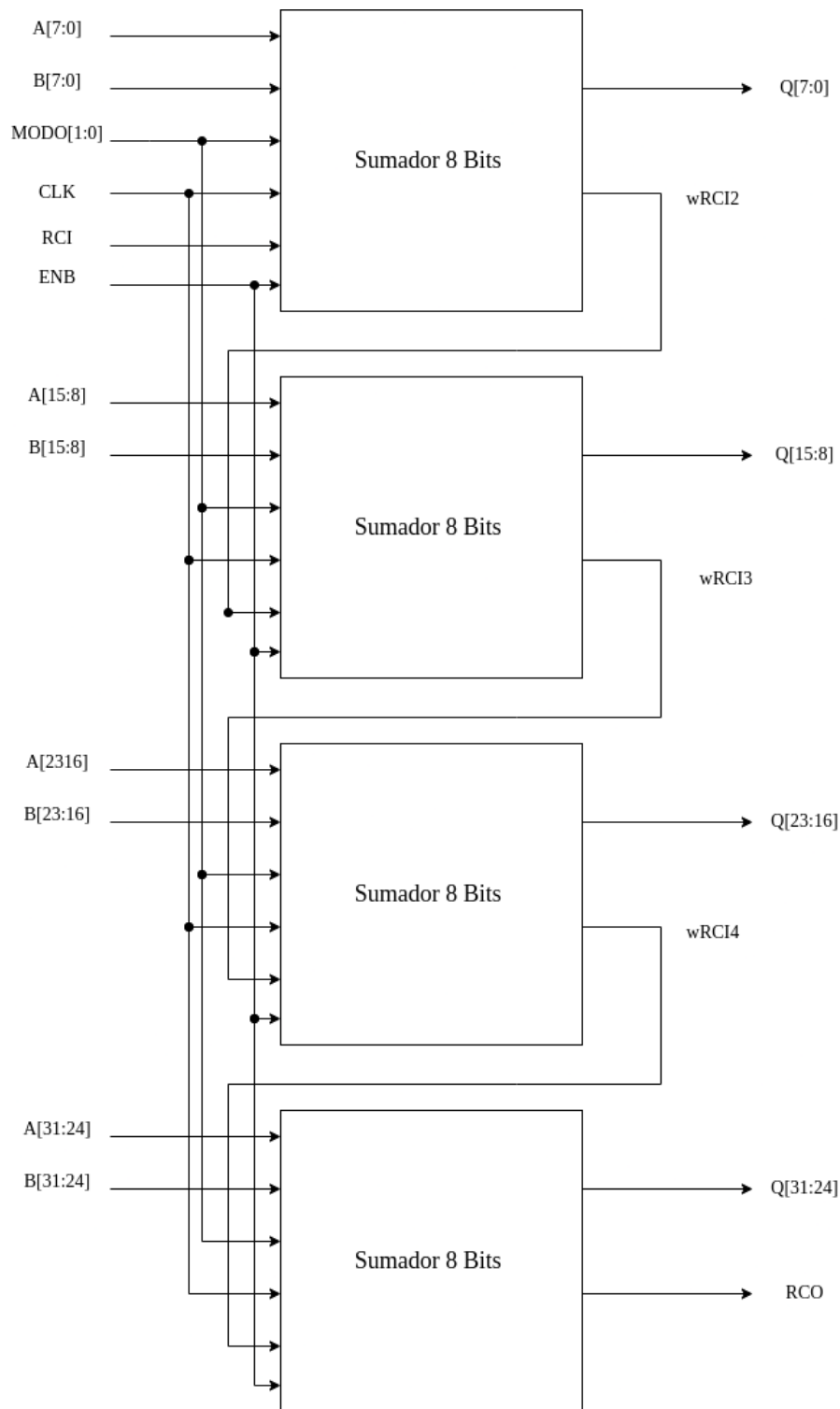


Figura 3: Diagrama Sumador 32 Bits.

Finalmente, en la figura anterior se muestra el bloque necesario para el sumador de 32 bits, en él se muestra que se encuentra compuesto por los diagramas de de 8 bits, por lo tanto, se instancia 4 veces este módulo. Al igual que el caso anterior, se tienen conexiones que permiten transmitir entre los módulos el acarreo de entrada al de salida.

Para los 3 sumadores se espera que se cumpla la siguiente tabla de transición de estados:

Estado Actual	Próximo Estado			
	MODO = 00	MODO = 01	MODO = 10	MODO = 11
MODO = 00	$Q = Q$	$Q = A + B + RCI$	$Q = A - B + RCI$	$Q = 0$
MODO = 01	$Q = A + B + RCI$	$Q = A + B + RCI$	$Q = A - B + RCI$	$Q = 0$
MODO = 10	$Q = A - B + RCI$	$Q = A + B + RCI$	$Q = A - B + RCI$	$Q = 0$
MODO = 11	$S \ Q = 0$	$Q = A + B + RCI$	$Q = A - B + RCI$	$Q = 0$

2. Plan de Pruebas.

A continuación se muestra el plan de pruebas a implementar:

■ Prueba 1: Suma de 4 Bits.

Esta prueba consiste en utilizar los archivos Sumador4Bits.v probador4Bits.v y test-bench4Bits.v para sumar dos números de 4 bits. Esta prueba se divide en dos casos, en el caso 1 se realizará la suma, con $RCI = 0$. En el caso 2 habilita el valor del $RCI = 1$. Para esta prueba fueron utilizados valores al azar.

El diseño implementado sí pasó esta prueba para ambos casos ya que proporcionó los valores esperados.

Pruebas realizadas:

- Caso: $A + B$ con $RCI = 0$: MODO = 2'b01;
 $A = 4'b0010$ $B = 4'b0001$
 $\#10$ $A = 4'b0111$ $B = 4'b0001$
- Caso: $A + B$ con $RCI = 1$:
 $\#5$ $RCI = 1$ $\#5$ $A = 4'b0111$ $B = 4'b0011$

■ Prueba 2: Resta de 4 Bits.

Para esta prueba fueron utilizados los archivos mencionados en la prueba anterior. Esta prueba se divide en tres casos, en el caso 1 se realizó la resta con $RCI = 0$ y siendo $A > B$. En el caso 2 se mantuvo el valor del $RCI = 0$ siendo $B > A$. Finalmente para la tercera prueba se habilitó $RCI = 1$ y utilizando $B > A$. La idea de utilizar $B > A$ es observar el comportamiento del diseño al tener resultados negativos. Al implementar la prueba el diseño pasó de forma correcta la prueba realizada.

- Caso: $A > B$: $\#10$ MODO = 2'b10
 $RCI = 0$ $\#5$ $A = 4'b0100$ $B = 4'b0001$
- Caso: $B > A$ con $RCI = 0$:
 $\#10$ $A = 4'b0001$ $B = 4'b0100$
- Caso: $B > A$ con $RCI = 1$:
 $\#5$ $RCI = 1$ $\#5$ $A = 4'b0001$ $B = 4'b0100$

■ **Prueba 3: Mantener el valor del MODO 00. Sumador 4 Bits.**

La idea de esta prueba es confirmar que al asignar un valor a $\text{MODO} = 00$ se mantenga el último valor asignado a Q hasta que exista un cambio en MODO. Para esta prueba no se habilitó el RCI. Al realizarla el diseño mantuvo el valor de Q sin ninguna interrupción.

- Caso 1. Mantener el valor del MODO 00.

#10 $\text{MODO} = 2'b01$

$\text{RCI} = 0$

#5 $A = 4'b0100$ $B = 4'b0010$ Se realiza el cambio de MODO:

#10 $\text{MODO} = 2'b00$

■ **Prueba 4: Mantener el valor cuando $\text{ENB} = 0$. Sumador de 4 Bits.**

Al igual que para la prueba anterior, se busca que se mantenga el valor de Q al deshabilitar el ENB. El diseño pasó la prueba sin problema. Manteniendo el valor de Q.

- Caso 1. Mantener el valor cuando $\text{ENB} = 0$.

#10 $\text{MODO} = 2'b10$

#5 $A = 4'b0100$ $B = 4'b0001$

Se realiza el cambio de ENB:

$\text{ENB} = 0$

■ **Prueba 5: Limpiar el Contador. Sumador de 4 Bits.**

Esta es la última prueba realizada directamente al sumador de 4 bits. En ella se asigna el valor 11 a MODO, de tal forma que $Q = 0$. Y que este valor se mantenga mientras el MODO continúe sin cambiar. La prueba se pasó fue pasada por el diseño implementado.

- Caso 1. Limpiar el contador.

#10 $\text{ENB} = 1$

$\text{MODO} = 2'b01$

$A = 4'b1000$ $B = 4'b0100$

Se realiza el cambio de MODO:

#10 $\text{MODO} = 2'b11$

■ **Prueba 6: Implementación Correcta Modos de Funcionamiento. Sumador de 8 Bits.**

Esta prueba fue aplicada al sumador de 8 bits, en ella se busca verificar que el diseño pueda realizar correctamente las acciones propuestas para cada asignación de MODO. Luego de realizar las pruebas correspondientes, el diseño no pasó la prueba, ya que al realizar la resta se tuvo un resultado erróneo.

A continuación se muestran los valores asignados para los casos realizados:

Valores iniciales para la simulación: $A = 8'b000000010$ $B = 8'b000000001$

$MODO = 2'b11$

$ENB = 1$ $CLK = 1$ $RCI = 0$

- Caso 1: $A + B$ con $RCI = 0$:

#10 $MODO = 2'b01$

#12 $A = 8'b01010010$ $B = 8'b00010001$

- Caso 2: $A \cdot B$:

#10 $MODO = 2'b10$

$RCI = 0$

#12 $A = 8'b01000000$ $B = 8'b00011001$

- Caso 3. Mantener el valor del $MODO$ 00.

#10 $MODO = 2'b01$

$RCI = 0$

#12 $A = 8'b01000000$ $B = 8'b00100000$

Se realiza el cambio de $MODO$:

#10 $MODO = 2'b00$

- Caso 4. Mantener el valor cuando $ENB = 0$.

#10 $MODO = 2'b10$

#12 $A = 8'b00101100$ $B = 8'b10000001$

Se realiza el cambio de ENB :

$ENB = 0$

#12 $A = 8'b00101100$ $B = 8'b10000001$

- Caso 5. Limpiar el contador.

10 $ENB = 1$

#10 $MODO = 2'b01$

#12 $A = 8'b00001000$ $B = 8'b00000100$

Se realiza el cambio de $MODO$:

#10 $MODO = 2'b11$

- **Prueba 7: Implementación Correcta Modos de Funcionamiento. Sumador de 32 Bits.** Al igual que para la prueba anterior, se busca cumplir con las asignaciones de $MODO$, pero en este caso para un sumador de 32 bits. El diseño utilizado superó la prueba propuesta sin ningún problema.

Los casos propuestos para esta prueba son los siguientes:

Valores Iniciales:

- #5 MODO = 2'b01
#6 A = 32'b00000000000000000000000000000000 B = 32'b00000000000000000000000000000001
- Caso 2. Resta de 4 Bits.
#5 MODO = 2'b10
#6 A = 32'b00000000000000000000000000000000 B = 32'b00000000000000000000000000000000
 - Caso 3. Mantener el valor del MODO 00.
#5 MODO = 2'b01;
#6 A = 32'b00000000000000000000000000000000 B = 32'b00000000000000000000000000000001
Se realiza el cambio de MODO:
#10 MODO = 2'b00
 - Caso 4. Mantener el valor cuando ENB = 0.
#5 MODO = 2'b10
#6 A = 32'b00000000000000000000000000000000 B = 32'b00000000000000000000000000000001
Se realiza el cambio de ENB:
ENB = 0
 - Caso 5. Limpiar el contador.
#10 ENB = 1
#5 MODO = 2'b01
A = 32'b00000000000000000000000000000000 B = 32'b00000000000000000000000000000001
Se realiza el cambio de MODO:
#5 MODO = 2'b11

4. Resultados.

4.1. Simulación Realizada al Sumador de 4 Bits.

La siguiente figura muestra la simulación realizada para el sumador de 4 bits:

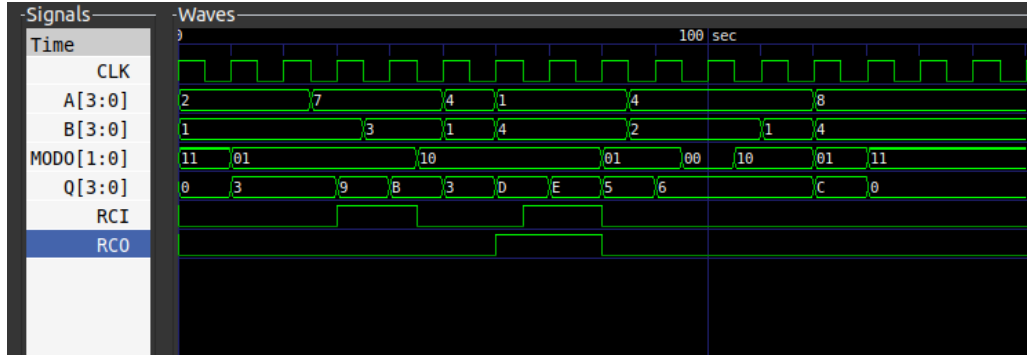


Figura 4: Simulación de las Pruebas Propuestas para un Sumador de 4 Bits.

Los resultados anteriores muestran el comportamiento del diseño para un sumador de 4 bits. Como se puede observar, se inicia poniendo MODO en 11, de tal forma que se haga un reset del sumador. Cuando llega el flanco positivo de reloj se procede a cambiar MODO a 01, de tal forma que se realicen las sumas correspondientes a la prueba 1. Se observa que se realizan las sumas sin ningún problema, sumando un 1 extra cuando $RCI = 1$.

Luego, se procede a asignar 10 a MODO de tal forma que se realice una resta, la cual corresponde a la prueba 2. Al igual que el caso anterior las restas las realiza correctamente, de nuevo, en el caso de que $RCI = 1$ suma un 1 extra, también al restar un $A \geq B$ se obtiene los valores esperados. De forma que sea sencillo de evidenciar el cambio, para la prueba 3 se realiza primero una suma asignando a MODO un 01 y luego un 00 para determinar que se mantiene el valor dado a Q en el flanco positivo de reloj anterior.

Para la prueba 4 se realiza un cambio de MODO a 10, y se deshabilita el ENB. Al realizar este cambio el valor de Q se mantiene, a pesar de que ENB es diferente de 1. Finalmente, para la prueba 5 se habilita el ENB nuevamente, y se asigna a MODO un valor de 01 para observar mejor el cambio. Luego, se procede a cambiar el valor de MODO a 00 y se observa que al realizar esta acción, en el mismo flanco positivo de reloj la señal de $Q = 0$. Por lo tanto, se puede decir que el sumador de 4 bits pasó todas las pruebas a las que fue sometido.

4.2. Simulación Realizada al Sumador de 8 Bits.

La siguiente figura muestra la simulación realizada para el sumador de 8 bits:

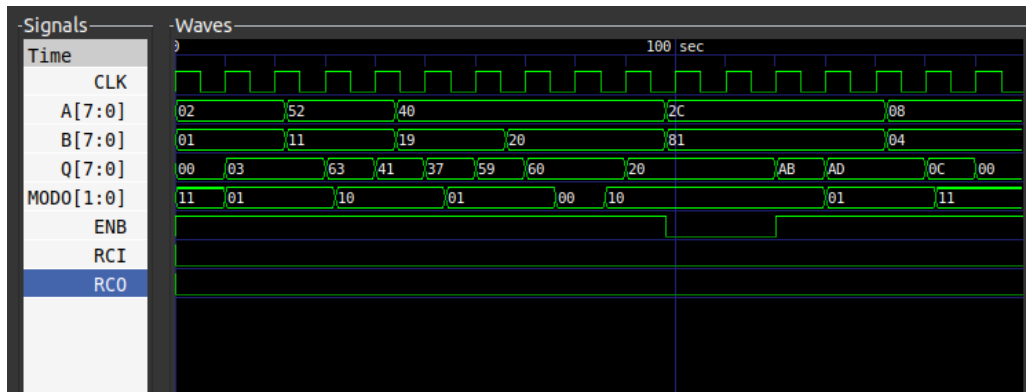


Figura 5: Simulación de las Pruebas Propuestas para un Sumador de 8 Bits.

En la simulación anterior se muestran los resultados de las pruebas básicas realizadas al sumador de 8 bits. Estas pruebas corresponden a los cambios de MODO. Para el primer caso, se pasa de el valor asignado a MODO = 11 a 01, de tal forma que se realice la suma esperada. La cual, el diseño realiza sin ningún problema.

Luego se procede a asignar el valor de 10 a MODO, de tal forma que se inicie la prueba de resta. En esta prueba se puede observar que inicialmente las restas se realizan correctamente, sin embargo, luego de hacer un cambio en los números A y B, específicamente cuando A = 0x40 y B = 0x19, existe un error en Q, ya que el resultado verdadero debería ser 0x27 y en cambio se obtiene 0x37. Este error puede ser provocado por un error al instanciar los módulos en el testbench posiblemente, y puede depender de los bits que se ingresen en A y B. ya que más adelante se realiza una resta, la cual resuelve correctamente.

Como caso siguiente se cambia el MODO a 00, de tal forma que se mantenga el valor de Q, esta prueba el diseño la pasa correctamente. Se procede a deshabilitar el ENB para la siguiente prueba, donde se observa que efectivamente se mantiene el valor correspondiente a Q. Finalmente se hace un cambio de MODO a 11 y se observa que inmediatamente el valor de Q se vuelve 0.

4.3. Simulación Realizada al Sumador de 32 Bits.

Finalmente, la siguiente figura muestra la simulación realizada para el sumador de 32 bits:

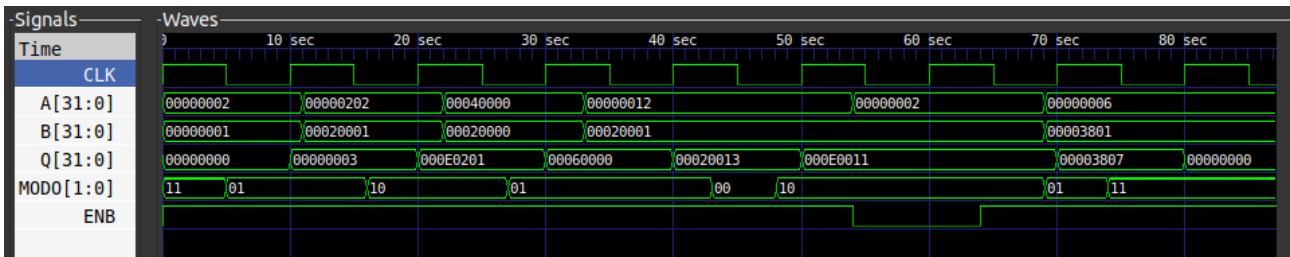


Figura 6: Simulación de las Pruebas Propuestas para un Sumador de 32 Bits.

Luego de realizar la simulación anterior, se observa que al igual que en los casos anteriores

se inicia la prueba poniendo MODO en 11. Este sumador se sometió a las mismas pruebas realizadas para el sumador de 8 bits, al observar la simulación sucedió un evento inesperado, ya que al realizar la prueba de resta de 32 bits, se obtuvieron los correctos de la resta, lo cual es interesante debido a la dependencia de este sumador con el sumador de 8 bits y al fallo de dicho sumador comentadas en la sección anterior. Este diseño pasó se forma efectiva todas las pruebas.

5. Conclusiones.

1. A partir del diseño propuesto se mostró que existe un funcionamiento correcto del sumador de 4 bits.
2. Al realizar la simulación del sumador de 8 bits en la prueba correspondiente de resta, se obtuvo que se asignan valores inesperados, dependiendo de las entradas asignadas en A y B.
3. Es posible que exista un error humano en el diseño realizado, que sea el causante de los errores en la prueba de 8 bits.
4. El sumador de 32 bits opera perfectamente bajo las condiciones propuestas, sin embargo, es necesario realizar pruebas más exhaustivas para determinar si su funcionamiento continúa siendo el correcto.

6. Recomendaciones.

- Es de gran ayuda realizar un diagrama de bloques o un diseño que permita comprender donde conectar las salidas o los cables utilizados, de tal forma que sea más sencillo instanciar los módulos.
- Puede ser buena idea brindar tiempos más largos a las pruebas de la simulación, de tal forma que no se pierdan valores esperados debido a que no logran asignarse correctamente antes del flanco positivo de reloj.
- Con respecto al tiempo invertido en el diseño, es importante que desde el inicio se cree un Makefile, de tal forma que se agilice el proceso de pruebas.