(de 1 Bit).12Código del módulo *fullAdder.v* (de 1 Bit).figure.caption.1

EIE
Escuela de
Ingeniería Eléctrica

UNIVERSIDAD DE COSTA RICA

Universidad de Costa Rica

Escuela de Ingeniería Eléctrica

Proyecto Final
(Parte Grupal)

Circuitos Digitales I

Estudiantes:
Royer Méndez Ramírez - A43333
Franco Castro Chaves - C01886
Carmen Garita Víquez - B93225

Profesor:
Rafael Esteban Badilla Alvarado

Circuitos Digitales I
IE-0323

San José,
1 de marzo de 2022

1. Diseñe e implemente un sumador completo de 4 bits, basado en el sumador completo de 1 bit anteriormente descrito.

En la figura 1 se puede observar el código del sumador completo de 1 bit implementado en la sección anterior. El código de la figura 1 puede ser compilado mediante el comando: *iverilog fullAdder.v*.

Figura 1: Código del módulo *fullAdder.v* (de 1 Bit).

```
module fullAdder(
    input inA,
    input inB,
    input carryIn,
    output sum,
    output carryOut);

assign carryOut =
    (~inA & inB & carryIn) |
    (inA & ~inB & carryIn) |
    (inA & inB & ~carryIn) |
    (inA & inB & carryIn);

assign sum =
    (~inA & ~inB & carryIn) |
    (~inA & inB & ~carryIn) |
    (inA & ~inB & ~carryIn) |
    (inA & inB & carryIn);

endmodule
```

Elaboración propia

Una manera de implementar un sumador completo de 4 bits es por medio de la unión de cuatro módulos del sumador completo de 1 bit. Esta fué la implementación utilizada en la Figura 2 para la programación del sumador completo de 4 bits. En este caso los módulos conectados para formar el sumador completo de 4 bits tiene el nombre de *fullAdder FA0, fullAdder FA1, FullAdder FA2, FullAdder FA3*.

Cada módulo del sumador completo de 1 bit se encuentra conectado por medio de un ”cable”, los cuales reciben el nombre de *wire carryOut0, carryOut1, carry Out2*. Las entradas a recibir serán de 4 bits por lo que se definen las entradas como números de [3:0] y el acarreo de entrada *carryIn* de un bit. Se puede observar que el módulo *fullAdder FA0* al ser el primer sumador completo de un bit del sumador completo de 4 bits recibe de acarreo entrada *carryIn* y produce un acarreo de salida con el nombre de *carryOut0*. El segundo sumador de un bit que conforma el sumador de 4 bits, *fullAdder FA1*, recibe como acarreo de entrada el acarreo de salida del *fullAdder FA0* es decir *carryOut0* y produce un acarreo de salida *carryout1*. Así sucesivamente los demás sumadores de 1 bit recibirán como acarreo de entrada el acarreo de salida del sumador de bit anterior y producirán un acarreo de salida (Figura 2).

Figura 2: Código del módulo *fullAdder4Bits.v*.

```
//FullAdder4Bits
`include "fullAdder.v"
module fullAdder4Bits(
    input [3:0] inA,
    input [3:0] inB,
    input carryIn,
    output [3:0] sum,
    output carryOut3);

wire carryOut0, carryOut1, carryOut2;

fullAdder FA0 (
    .inA(inA[0]),
    .inB(inB[0]),
    .carryIn(carryIn),
    .sum(sum[0]),
    .carryOut(carryOut0)
);

fullAdder FA1 (
    .inA(inA[1]),
    .inB(inB[1]),
    .carryIn(carryOut0),
    .sum(sum[1]),
    .carryOut(carryOut1)
);

fullAdder FA2 (
    .inA(inA[2]),
    .inB(inB[2]),
    .carryIn(carryOut1),
    .sum(sum[2]),
    .carryOut(carryOut2)
);

fullAdder FA3 (
    .inA(inA[3]),
    .inB(inB[3]),
    .carryIn(carryOut2),
    .sum(sum[3]),
    .carryOut(carryOut3)
);

endmodule
```

Elaboración propia.

2. Diseñe e implemente un banco de pruebas para el sumador completo de 4 bits del punto anterior.

En la Figura 3 se puede observar cómo en el código del banco de pruebas se incluye el sumador completo de 4 bits descrito anteriormente. Se crea el módulo del banco de pruebas en donde se crean las variables de entrada *A_tb, B_tb* las cuales tienen un tamaño de 4 bits, así también; se crean las variables de salida *Sumtb* de un bit y *CarryOut3_tb* de 4 bit.

Figura 3: Módulo del banco de pruebas para el sumador completo de 4 bits (Creación de las Variables).

fullAdder4Bits_tb.v

C: > Users > DELL > Desktop > fullAdder4Bits_tb.v

```
1  `include "fullAdder4Bits.v"
2
3  //Selección de las variables y salidas del modulo de pruebas
4  module fullAdder4Bits_tb;
5
6  //Entradas
7  reg [3:0] inA_tb;
8  reg [3:0] inB_tb;
9  reg carryIn_tb;
10
11 //Salidas
12 wire [3:0] sum_tb;
13 wire carryOut3_tb;
14
```

Elaboración propia

Se hace la instancia al módulo *fullAdder4Bits*, en la Figura 4 a partir de la línea 17 de código se puede observar esta instanciación, la cual tiene por nombre DUT. La instanciación se hizo de esta forma: el *fullAdder4Bits* tiene 4 pines llamados inA, esos cuatro pines van a ser cableados a los respectivas entradas inA_tb del módulo de pruebas, el fullAdder4Bits tiene 4 pines llamados inB, esos cuatro pines van a ser cableados a las respectivas entradas del módulo de prueba inB_tb, en el caso del carryIn es el único pin que se encuentra cableado a la única entrada del módulo de pruebas de carryIn_tb.

Figura 4: Módulo del banco de pruebas para el sumador completo de 4 bits (Instanciación).

```
15
16 //instanciacion al módulo fullAdder4Bits
17 fullAdder4Bits DUT (
18     .inA(inA_tb),
19     .inB(inB_tb),
20     .carryIn(carryIn_tb),
21     .sum(sum_tb),
22     .carryOut3(carryOut3_tb)
23 );
24
```

Elaboración propia

Posteriormente, se realizan las variaciones en el banco de pruebas correspondientes a las variaciones que pueden tomar las entradas A,B y carryIn para posteriormente verificae los valores de las salidas con respecto a cada una de las variaciones de las entradas en el GTKwave (Figura 5).

Únicamente se pueden realizar modificaciones a los valores de las entradas en el módulo de testing debido a que no es posible modificar las salidas, debido a que las salidas depende de las entradas; es por esto que la modificación de las salidas se encarga el módulo *fullAdder4Bits*. Para llevar a cabo esa revisión se le da la orden a verilog que cree un archivo *"test4Bits.vcd"* y que incluya todas las permutaciones realizadas en el test de las variables de entrada en el mismo archivo, el cual después va a ser leído por GTKwave.

La Figura 5 no corresponde al código completo utilizado para el banco de pruebas, contiene únicamente algunas cuantas lineas que muestran la estructura del mismo. El código completo contiene las líneas de código necesarias para ejecutar las 512 permutaciones posibles. El código completo utilizado para el banco de pruebas del sumador completo de 4 bits puede encontrarse en la sección de anexos (Figuras de la 22 a la 29).

Figura 5: Módulo del banco de pruebas para el sumador completo de 4 bits

```
24
25  //Creación del archivo .vcd que será utilizado por GTKwave
26  initial begin
27      $dumpfile("test4Bits.vcd");
28      $dumpvars;
29      inA_tb = 4'b0000; inB_tb = 4'b0000; carryIn_tb = 1'b0;
30      {inA_tb, inB_tb, carryIn_tb} <= 9'b000000000; #1;
31      {inA_tb, inB_tb, carryIn_tb} <= 9'b000000010; #1;
32      {inA_tb, inB_tb, carryIn_tb} <= 9'b000000100; #1;
33      {inA_tb, inB_tb, carryIn_tb} <= 9'b000000110; #1;
34      {inA_tb, inB_tb, carryIn_tb} <= 9'b000001000; #1;
35      {inA_tb, inB_tb, carryIn_tb} <= 9'b000001010; #1;
36      {inA_tb, inB_tb, carryIn_tb} <= 9'b000001100; #1;
37      {inA_tb, inB_tb, carryIn_tb} <= 9'b000001110; #1;
38      {inA_tb, inB_tb, carryIn_tb} <= 9'b000010000; #1;
39      {inA_tb, inB_tb, carryIn_tb} <= 9'b000010010; #1;
40      {inA_tb, inB_tb, carryIn_tb} <= 9'b000010100; #1;
41      {inA_tb, inB_tb, carryIn_tb} <= 9'b000010110; #1;
42      {inA_tb, inB_tb, carryIn_tb} <= 9'b000011000; #1;
43      {inA_tb, inB_tb, carryIn_tb} <= 9'b000011010; #1;
44      {inA_tb, inB_tb, carryIn_tb} <= 9'b000011100; #1;
45      {inA_tb, inB_tb, carryIn_tb} <= 9'b000011110; #1;
46      {inA_tb, inB_tb, carryIn_tb} <= 9'b000100000; #1;
47      {inA_tb, inB_tb, carryIn_tb} <= 9'b000100010; #1;
48      {inA_tb, inB_tb, carryIn_tb} <= 9'b000100100; #1;
49      {inA_tb, inB_tb, carryIn_tb} <= 9'b000100110; #1;
50      {inA_tb, inB_tb, carryIn_tb} <= 9'b000101000; #1;
51      {inA_tb, inB_tb, carryIn_tb} <= 9'b000101010; #1;
52      {inA_tb, inB_tb, carryIn_tb} <= 9'b000101100; #1;
53      {inA_tb, inB_tb, carryIn_tb} <= 9'b000101110; #1;
54      {inA_tb, inB_tb, carryIn_tb} <= 9'b000110000; #1;
55      {inA_tb, inB_tb, carryIn_tb} <= 9'b000110010; #1;
56      {inA_tb, inB_tb, carryIn_tb} <= 9'b000110100; #1;
57      {inA_tb, inB_tb, carryIn_tb} <= 9'b000110110; #1;
58      {inA_tb, inB_tb, carryIn_tb} <= 9'b000111000; #1;
59      {inA_tb, inB_tb, carryIn_tb} <= 9'b000111010; #1;
60      {inA_tb, inB_tb, carryIn_tb} <= 9'b000111100; #1;
```

Elaboración propia

Para ejecutar el sumador completo de 4 bits,en su terminal diríjase a la ubicación que contiene los archivos extraídos del .zip.Posteriormente, ejecute los siguiente comando en en su terminal en el siguiente orden.

1. El archivo con el nombre fullAdder.v, debe compilar sin problemas bajo el comando: iverilog fullAdder.v

2. El archivo con el nombre fullAdder4Bits.v debe complilar sin problemas bajo el comando: iverilog fullAdder4Bits.v

3. El archivo con el nombre fullAdder4Bits debe compilar sin problemas bajo el siguiente comando: iverilog -o fullAdder4Bits.o fullAdder4Bits.v

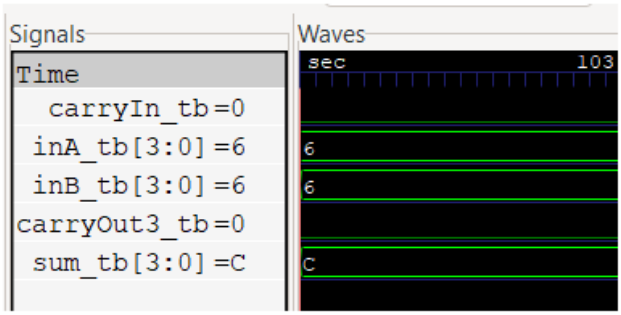
4. El archivo con el nombre fullAdder4Bits_tb.v debe compilar sin problemas con siguinte comando: iverilog -o fullAdder4Bits_tb.o fullAdder4Bits_tb.v

5. Al ejecutar el comando del punto anterior, se obtendrá el archivo fullAdder4Bits_tb.o, el cual es el ejecutable que se ha creado, para ejecutarlo se utiliza en siguiente comando: vvp fullAdder4Bits_tb.o

6. Al ejecutar el comando anterior se obtendrá un archivo llamado *test4Bits.vcd*, el cual se puede abrir con gtkwave y contendrá las señales del circuito. gtkwave test4Bits.vcd

Al ejecutar el último comando *gtkwave test4Bits.vcd* se abrirá una ventana en GTKWave en donde se podrá verificar el funcionamiento del sumador completo de 4 bits. A continuación algunos ejemplos del funcionamiento de sumador completo de 4 bits con sus respectivas comprobaciones:

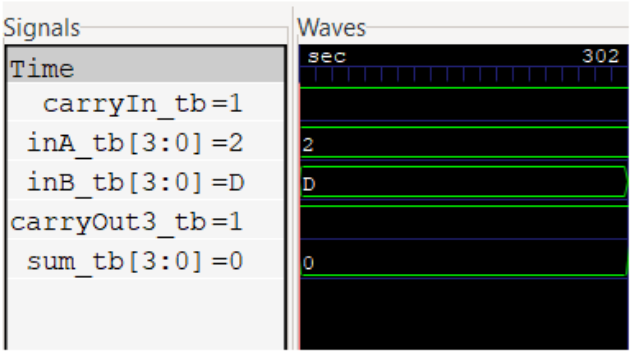
Figura 6: Funcionamiento de sumador completo de 4 bits(1)



	carryOut	(llevos internos)			carryIn
	0	1	1		0
inA		0	1	1	0
inB		0	1	1	0
sum	0	1	1	0	0

Elaboración propia

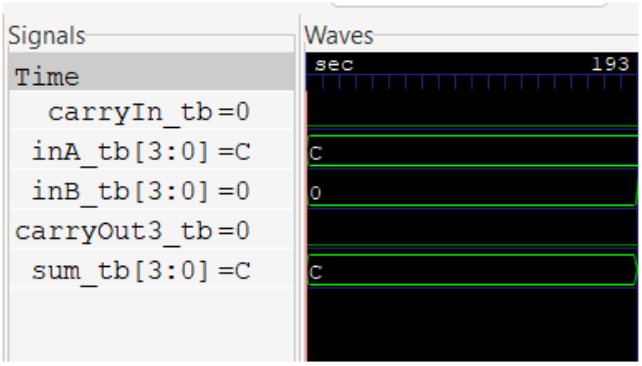
Figura 7: Funcionamiento de sumador completo de 4 bits(2)



	carryOut	(llevos internos)			carryIn
	1	1	1	1	1
inA		0	0	1	0
inB		1	1	0	1
sum	1	0	0	0	0

Elaboración propia

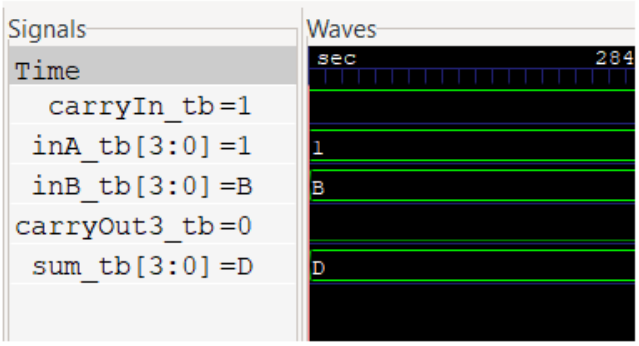
Figura 8: Funcionamiento de sumador completo de 4 bits(3)



	carryOut	(llevos internos)			carryIn
	0				0
inA		1	1	0	0
inB		0	0	0	0
sum	0	1	1	0	0

Elaboración propia

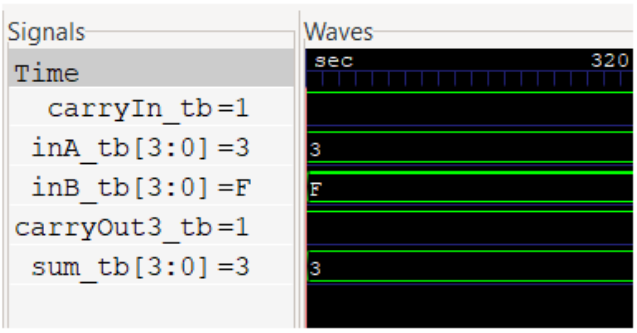
Figura 9: Funcionamiento de sumador completo de 4 bits(4)



	carryOut	(llevos internos)			carryIn
	0		1	1	1
inA		0	0	0	1
inB		1	0	1	1
sum	0	1	1	0	1

Elaboración propia

Figura 10: Funcionamiento de sumador completo de 4 bits(5)



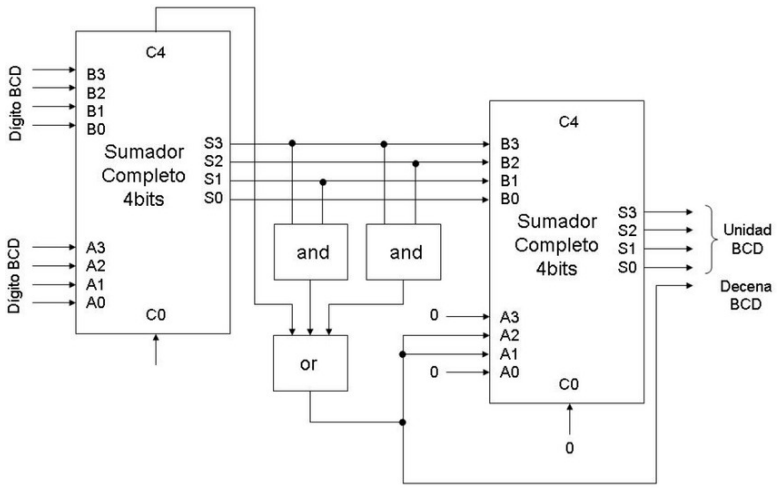
	carryOut	(llevos internos)			carryIn
	1	1	1	1	1
inA		0	0	1	1
inB		1	1	1	1
sum	1	0	0	1	1

Elaboración propia

3. Diseñe e implemente un sumador BCD de un dígito a partir del sumador de 4 bits previamente desarrollado. Agregue una señal de salida llamada sumVal que indique si la suma es válida.

Para la implementación de un sumador BCD de un dígito, se toma como referencia el diagrama de bloques del mismo:

Figura 11: Diagrama de bloques de un sumador BCD de dos dígitos.



Montoya,E.Ortega,S.(2011).ResearchGate(https://www.researchgate.net/figure/FIGURA-6-Diagrama-de-bloques-de-un-sumador-de-2-digitos-BCD$_{f}ig3_{2}$*60778529).CreativeCommonsAttribution−NonCommercial*3,0.

Con base en la Figura 11 se escribe el siguiente código:

Figura 12: Entradas, salidas del sumador BCD y función sumVal.

```
`include "fullAdder4Bits.v"

module BCDnivel1(
    input [3:0] inA,
    input [3:0] inB,
    input carryIn,
    output [3:0] sumBCD,
    output acarreoBCD,
    output decenaBCD,
    output sumVal);


wire [3:0] conectores;
wire [3:0] entradaB;
wire acarreo, carryinnivel2;

assign sumVal = (inA > 9) ? 0 : ((inB > 9) ? 0 : 1);
```

Elaboración propia

En el código se incluye el sumador completo de cuatro bits, el cual se demostrará su uso posteriormente. Se declaran tres variables de entrada, los vectores binarios de cuatro bits *InA* e *InB*, que representan los dígitos en sistema BCD a sumar y un acarreo inicial *carryIn*. De forma similar, se declaran cuatro salidas. La salida *sumBCD* es un vector de 4 bits que denota el valor de la unidad que

resulta de la suma de las entradas en sistema BCD, la salida *acarreoBCD* es un bit en 0 necesario por la construcción del sumador completo del cuatro bits, la salida *decenaBCD* simboliza si se genera una decena y finalmente *sumVal* muestra si la suma es valida o no (Figura 12).

Asímismo, se declara un vector de nets denominado *conectores*, el cual cablea las salidas del primer sumador de cuatro bits con el segundo sumador de cuatro bits, otro vector de nets llamado e*ntradasB*, el cual funciona como el encargado de recibir el dígito que corrige el resultado de la suma BCD en caso de ser necesario y finalmente, dos nets declaradas como *acarreo* y *carryinnivel2* y la función *sumVal*.

Para la net *sumVal* se declara un *continuos assignment* y utilizando el operador ternario se modela el valor de *sumVal* de modo que devuelva 1 si la suma es válida, es decir, los vectores de entrada en sistema BCD representan un número menor o igual a nueve y devuelvan un 0 en caso contrario (Figura 12).

Seguidamente, se procede a completar el modulo, mediante la instanciación de los dos sumadores completos de 4 bits y generando los *continuos assignment* necesarios completar la lógica combinacional.

Figura 13: Par de sumadores de cuatro bits en cascada para el sumador BCD.

```
fullAdder4Bits nivel1(
    .inA(inA),
    .inB(inB),
    .carryIn(carryIn),
    .sum(conectores),
    .carryOut3(acarreo)
);

assign salidaOr =
    acarreo |
    (conectores[3] & conectores[2] & conectores[1]) |
    (conectores[3] & conectores[2]) |
    (conectores[3] & conectores[1]);


assign entradaB[0] = 0;
assign entradaB[3] = 0;
assign entradaB[1] = salidaOr;
assign entradaB[2] = salidaOr;
assign carryinnivel2 = 0;
assign decenaBCD = salidaOr;


fullAdder4Bits nive2(
    .inA(conectores),
    .inB(entradaB),
    .carryIn(carryinnivel2),
    .sum(sumBCD),
    .carryOut3(acarreoBCD)
);

endmodule
```

Elaboración propia

A continuación se procederá a explicar el funcionamiento del par de sumadores de cuatro bits y los assignments mostrados en la Figura 13.

En primer lugar, se instancia el primer sumador de cuatro bits el cual se denomina como nivel 1 y recibe como entradas a los vectores *inA*, *inB* y al bit *carryIn*. Su funcionalidad es realizar una suma de cuatro bits de forma común y dirigir el resultado de la suma a la net *conectores* y en caso de que se produzca un acarreo, producir una salida en 1 con el mismo nombre.

Ahora, en caso de que el nibble *conectores* sea mayor a 9 (en sistema BCD) o que *acarreo* sea 1, se debe corregir el resultado, para ello, se declara un *continuos assignment* el cual cumple la función de una compuerta OR de tres entradas, cuyo resultado se conecta a las entradas 1 y 2 del vector *entradasB*, el cual a su vez tiene conectadas sus entradas 0 y 3 a tierra, pues si la compuerta OR produce un 1 lógico en su salida, se generará un 6 en BCD (0110) y en caso contrario se generará un 0 en BCD (0000).

Ahora, para que la compuerta OR de como resultado un alto se debe satisfacer una de tres condiciones, la primera es que *acarreo* sea 1, la segunda y tercera condición tienen que ver con el valor de *conectores*. Note que el numero 9 en BCD está dado por la expresión 1001, por lo tanto cualquier numero en *conectores* que tengan la entradas 3 y 2 en 1, las entradas 3 y 1 en 1, o las entradas 3 y 2 y 1 en 1 (este último caso no se muestra en el diagrama de la Figura 11 ya que es un caso redundante que es como una composicion de las dos compuertas AND anteriores, aún así, debido a que es un caso que tambien ocurre, tambien lo agregamos como una tercera compuerta AND), será un número mayor que 9. Estas son las tres condiciones faltantes mencionadas, las cuales se representan mediante compuertas AND y se conectan con la OR. Con esto se obtiene toda la lógica combinacional necesaria para corregir el resultado en *conectores* en caso de ser necesario. De igual forma, la salida *OR* se conecta a la net *decenaBCD* la cual indicará si la suma ha producido un número de un dígito (si su resultado es) o si la suma ha producido un número de dos dígitos (si su resultado es 1).

Finalmente, una vez planteada la lógica combinacional de la corrección del resultado, se instancia un segundo sumador de cuatro bits en cascada el cual recibe como primer nibble a *conectores* y como segundo nibble a **entradaB** y un acarreo que siempre será 0 denominado *carryInnivel2*. Con lo anterior, realiza la suma de cuatro bits cuyo resultado se denomina *sumBCD* y es el valor final que se desea obtener (Figura 13).

4. Diseñe e implemente un banco de pruebas para el sumador BCD de un dígito del punto anterior.

Considere la siguiente figura:

Figura 14: Modulo del banco de pruebas para el sumador BCD (1)

```
`include "BCDnivel1.v"

module BCDnivel1_tb;

reg [3:0] inA_tb;
reg [3:0] inB_tb;
reg carryIn_tb;

wire [3:0] sumBCD_tb;
wire acarreoBCD_tb;
wire cable_tb, carryinnivel2_tb, decenaBCD_tb;
wire sumVal_tb;

BCDnivel1 BCDnivel1_inst (
    .inA(inA_tb),
    .inB(inB_tb),
    .carryIn(carryIn_tb),
    .sumBCD(sumBCD_tb),
    .acarreoBCD(acarreoBCD_tb),
    .decenaBCD(decenaBCD_tb),
    .sumVal(sumVal_tb)
);
```

Elaboración propia

Para este banco de pruebas se declaran las variables necesarias mostradas en la Figura 14 y se instancia el modulo del sumador BCD. Seguidamente, se declara un bloque *initial* el cual sigue la siguiente estructura:

Figura 15: Modulo del banco de pruebas para el sumador BCD (2)

```
initial begin
    $dumpfile("testBCD.vcd");
    $dumpvars;
    inA_tb = 4'b0000; inB_tb = 4'b0000; carryIn_tb = 1'b0;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000010; #1;
.
.
.
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111111101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111111111; #1;


    $monitor($time,,, "inA: %4b, inB: %4b, carryIn_tb: %1b",inA_tb, inB_tb, carryIn_tb);
    #600
    $finish();

end
```

Elaboración propia

La Figura 15 no es el código completo utilizado para el banco de pruebas, contiene únicamente

unas cuantas lineas que muestran cual es la estructura que sigue el código final. El código completo contiene las lineas de código necesarias para ejecutar las 512 posibles permutaciones posibles con las entradas del sumador BCD y se puede ver en detalle en la sección de *Anexos* (Figuras 30 a la 35).

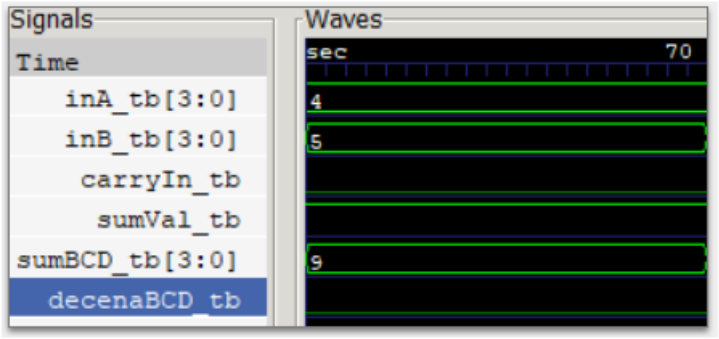
Ahora, para ejecutar el sumador BCD, en su terminal diríjase a la ubicación donde contiene los archivos extraídos del .zip y digite las siguientes lineas de código:

1. El archivo con el nombre fullAdder debe compilar sin problemas bajo el comando: iverilog fullAdder.v
2. El archivo con el nombre fullAdder4Bits debe compilar sin problemas bajo el comando:iverilog fullAdder4Bits.v
3. El archivo con el nombre BCDnivel1 debe compilar sin problemas bajo el comando: iverilog -o BCDnivel1.o BCDnivel1.v
4. El archivo con el nombre BCDnivel1_tb debe compilar sin problemas bajo el comando: iverilog -o BCDnivel1_tb.o BCDnivel1_tb.v
5. Al ejectuar el comando del punto anterior, obtendrá el archivo BCDnivel1_tb.o, el cual es el ejecutable que se ha creado, para ejecutarlo, utilice el siguiente comando: vvp BCDnivel1_tb.o
6. Al ejecutar el comando anterior, obtendrá un archivo llamado "testBCD.vcd", el cual podrá abrir en GTKWave para mostrar las señales del circuito. Para ello, utilice el siguiente comando: gtkwave testBCD.vcd

Al ejecutar la ultima línea de código se desplegará una ventana en GTKWave donde podrá verificar el funcionamiento del sumador BCD.

A continuación se muestran unos cuantos ejemplos y sus respectivas comprobaciones:

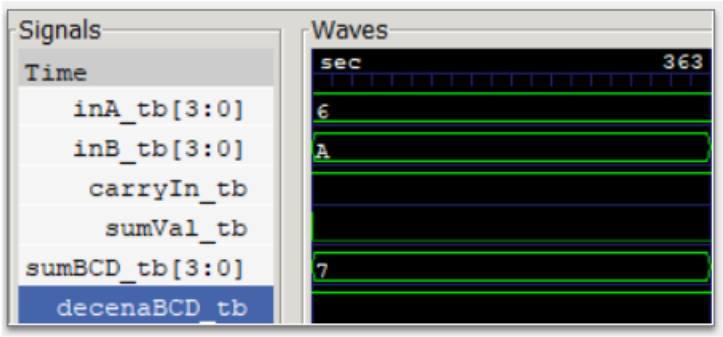
Figura 16: Funcionamiento del sumador BCD (1)



1er Nivel		Dec.BCD	(llevos internos)			carryIn		sumVal	
		0	1			0		No valida = 0	Válida = 1
	inA		0	1	0	0	→→→		
	inB		0	1	0	1			
	sumBCD	0	1	0	0	1			1

Elaboración propia

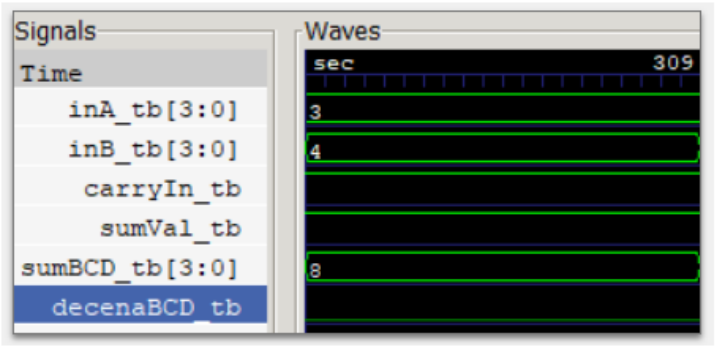
Figura 17: Funcionamiento del sumador BCD (2)



		Dec.BCD	(llevos internos)			carryIn		sumVal	
		1	1	1		1	→→→	No valida = 0	Válida = 1
1er Nivel	inA		0	1	1	0		0	
	inB		1	0	1	0			
	sumBCD	1	0	0	0	1			
2do Nivel	+6		0	1	1	0			
	sumBCD	1	0	1	1	1			

Elaboración propia

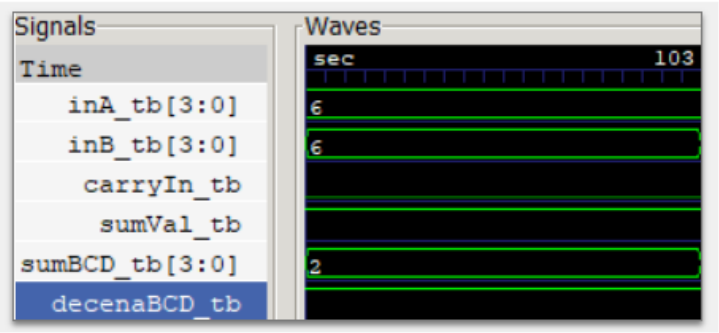
Figura 18: Funcionamiento del sumador BCD (3)



		Dec.BCD	(llevos internos)			carryIn		sumVal	
		0	1	1	1	1	→→→	No valida = 0	Válida = 1
1er Nivel	inA		0	0	1	1			1
	inB		0	1	0	0			
	sumBCD	0	1	0	0	0			

Elaboración propia

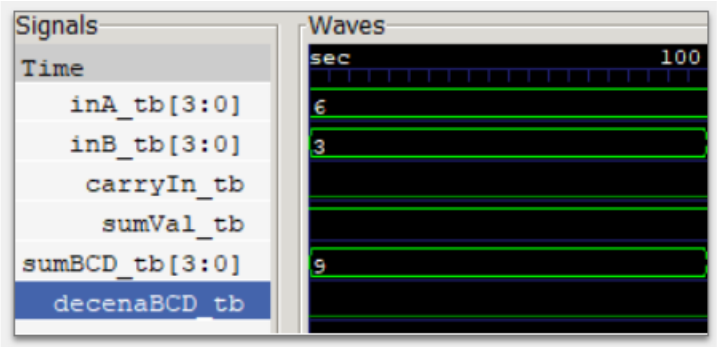
Figura 19: Funcionamiento del sumador BCD (4)



		Dec.BCD	(llevos internos)			carryIn		sumVal	
		1	1	1		0	→→→	No valida = 0	Válida = 1
1er Nivel	inA		0	1	1	0			1
	inB		0	1	1	0			
	sumBCD		1	1	0	0			
2do Nivel	+6		0	1	1	0			
	sumBCD	1	0	0	1	0			

Elaboración propia

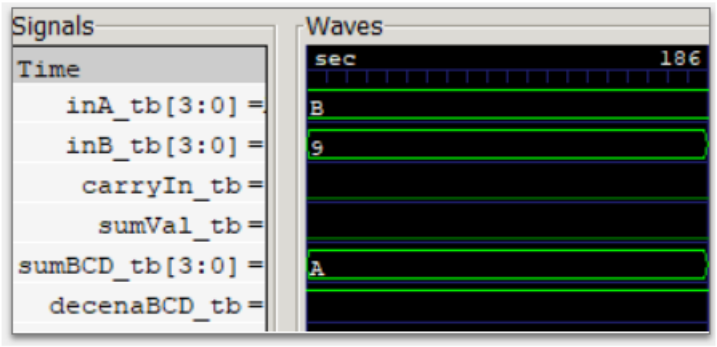
Figura 20: Funcionamiento del sumador BCD (5)



		Dec.BCD	(llevos internos)			carryIn		sumVal	
		0	1	1		0	→→→	No valida = 0	Válida = 1
1er Nivel	inA		0	1	1	0			1
	inB		0	0	1	1			
	sumBCD	0	1	0	0	1			

Elaboración propia

Figura 21: Funcionamiento del sumador BCD (6)



		Dec.BCD	(llevos internos)			carryIn		sumVal	
		1		1	1	0	→→→	No valida = 0	Válida = 1
1er Nivel	inA		1	0	1	1		0	
	inB		1	0	0	1			
	sumBCD	1	1 0	1	0	0			
2do Nivel	+6		0	1	1	0			
	sumBCD	1	1	0	1	0			

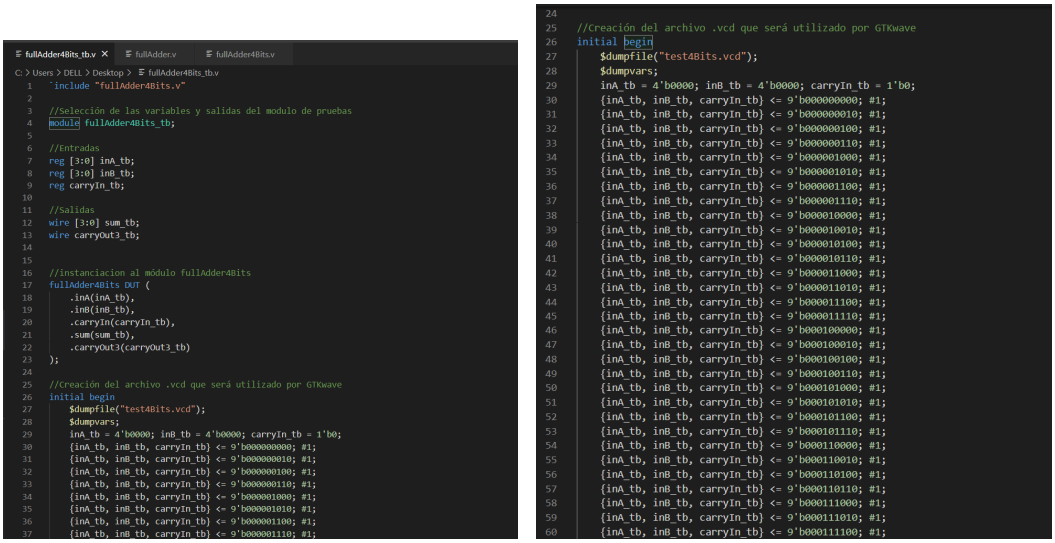
Elaboración propia

En todos los casos expuestos anteriormente se puede verificar que funciona la salida *sumVal_tb* cuando las entradas *inA_tb* o *inB_tb* son valores mayores a 9 (A=10, B=11 en este caso), se cumple que *sumVal_tb*=0, mientras que se obtiene que *sumVal*=1 para casos donde las entradas son menores que 9. Por otra parte, en todos los casos en los cuales la suma es valida se cumple que *sumBCD_tb* corresponde al valor de la cifra menos significativa de la suma. De igual manera, se obtiene que *decenasBCD_tb*=1 cuando la suma resulta en un valor de dos cifras. Con esto se verifica el correcto funcionamiento del sumador BCD.

4.1. Anexos

De las Figuras 22 a la 29 se puede observar el código completo correspondiente al banco de pruebas para el sumador completo de 4 bits.

Figura 22: Módulo del banco de pruebas para el fullAdder de 4 Bits (Parte 1).

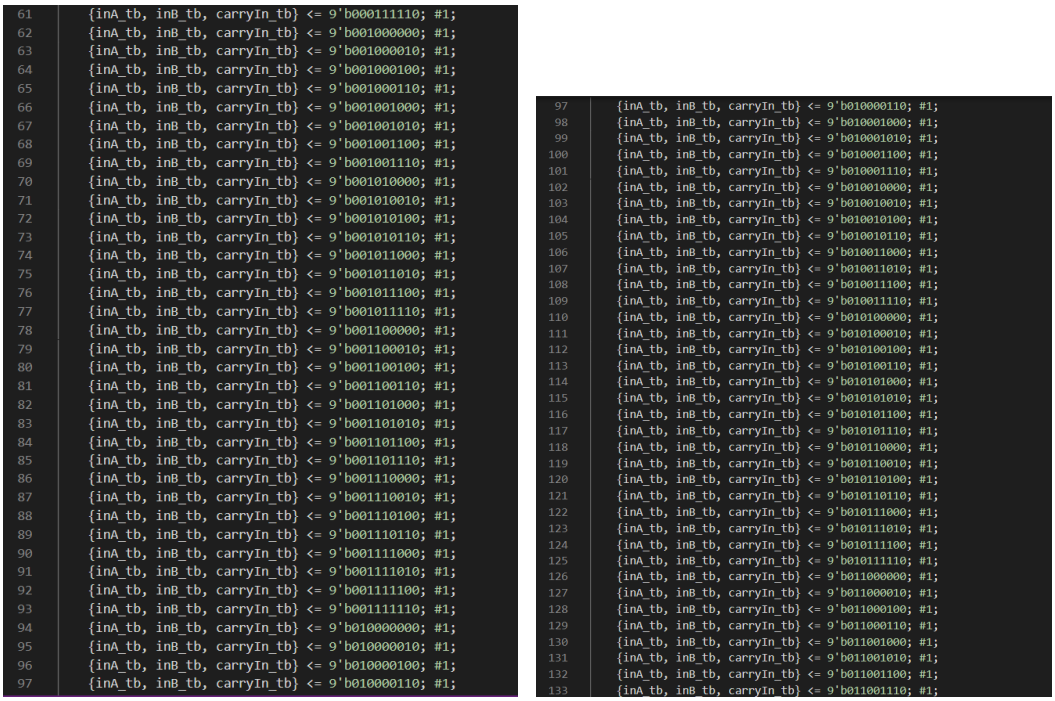


```
fullAdder4Bits_tb.v | fullAdder.v | fullAdder4Bits.v
C: > Users > DELL > Desktop > fullAdder4Bits_tb.v
`include "fullAdder4Bits.v"

//Selección de las variables y salidas del modulo de pruebas
module fullAdder4Bits_tb;

//Entradas
reg [3:0] inA_tb;
reg [3:0] inB_tb;
reg carryIn_tb;

//Salidas
wire [3:0] sum_tb;
wire carryOut3_tb;


//instanciacion al módulo fullAdder4Bits
fullAdder4Bits DUT (
    .inA(inA_tb),
    .inB(inB_tb),
    .carryIn(carryIn_tb),
    .sum(sum_tb),
    .carryOut3(carryOut3_tb)
);

//Creación del archivo .vcd que será utilizado por GTKwave
initial begin
    $dumpfile("test4Bits.vcd");
    $dumpvars;
    inA_tb = 4'b0000; inB_tb = 4'b0000; carryIn_tb = 1'b0;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001110; #1;
```

```
//Creación del archivo .vcd que será utilizado por GTKwave
initial begin
    $dumpfile("test4Bits.vcd");
    $dumpvars;
    inA_tb = 4'b0000; inB_tb = 4'b0000; carryIn_tb = 1'b0;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000010000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000010010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000010100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000010110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000011000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000011010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000011100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000011110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000100000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000100010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000100100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000100110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000101000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000101010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000101100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000101110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000110000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000110010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000110100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000110110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000111000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000111010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000111100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000111110; #1;
```

Elaboración propia.

Figura 23: Módulo del banco de pruebas para el fullAdder de 4 Bits (Parte 2).

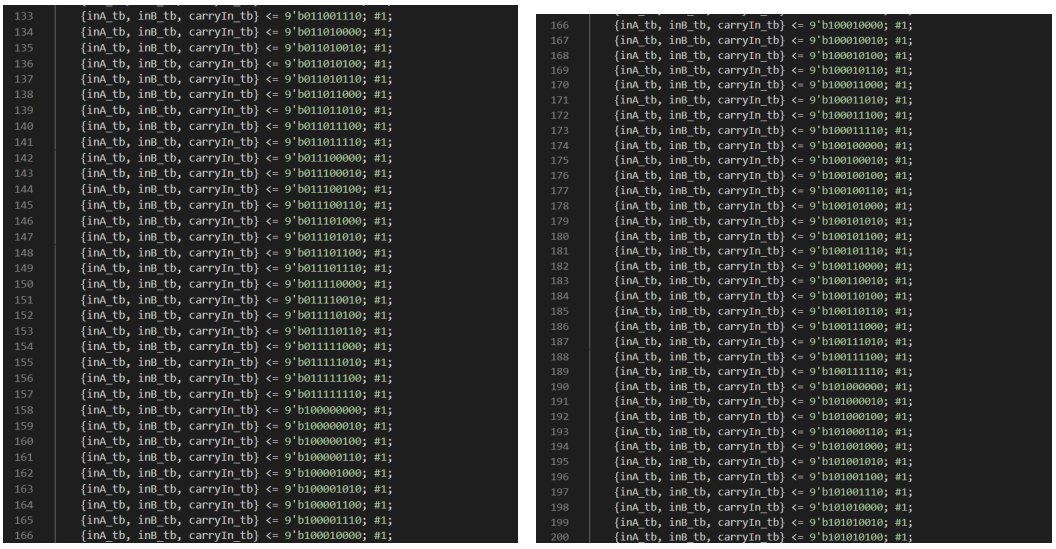


```
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000111110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001000000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001000010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001000100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001000110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001001000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001001010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001001100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001001110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001010000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001010010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001010100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001010110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001011000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001011010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001011100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001011110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001100000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001100010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001100100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001100110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001101000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001101010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001101100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001101110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001110000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001110010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001110100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001110110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001111000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001111010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001111100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001111110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010000000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010000010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010000100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010000110; #1;
```

```
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010000110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010001000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010001010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010001100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010001110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010010000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010010010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010010100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010010110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010011000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010011010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010011100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010011110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010100000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010100010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010100100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010100110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010101000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010101010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010101100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010101110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010110000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010110010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010110100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010110110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010111000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010111010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010111100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010111110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b011000000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b011000010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b011000100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b011000110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b011001000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b011001010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b011001100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b011001110; #1;
```

Elaboración propia.

Figura 24: Módulo del banco de pruebas para el fullAdder de 4 Bits (Parte 3).

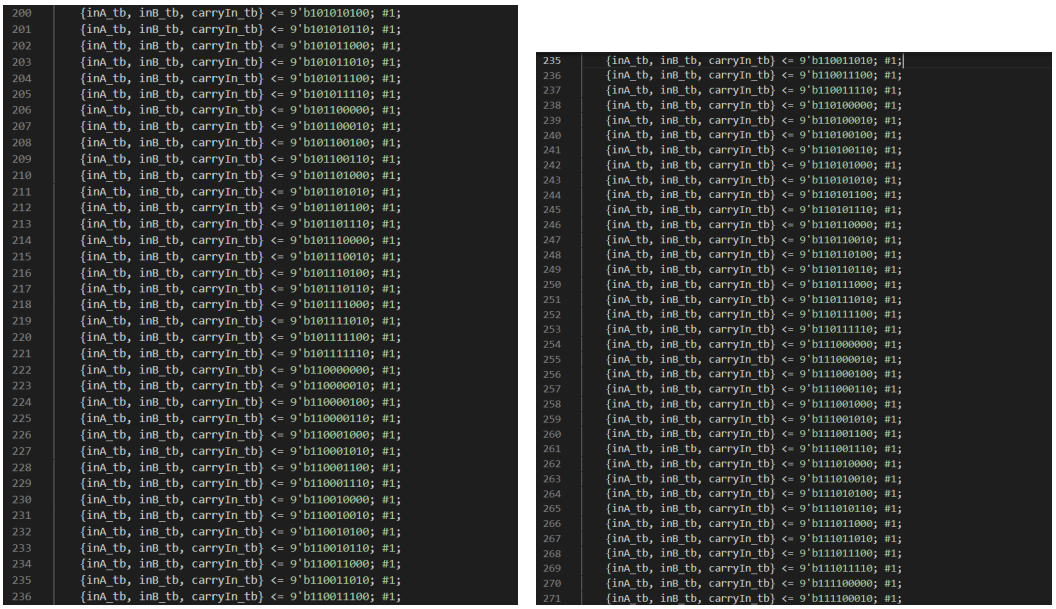


```
133     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001110; #1;
134     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010000; #1;
135     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010010; #1;
136     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010100; #1;
137     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010110; #1;
138     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011000; #1;
139     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011010; #1;
140     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011100; #1;
141     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011110; #1;
142     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100000; #1;
143     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100010; #1;
144     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100100; #1;
145     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100110; #1;
146     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101000; #1;
147     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101010; #1;
148     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101100; #1;
149     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101110; #1;
150     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110000; #1;
151     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110010; #1;
152     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110100; #1;
153     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110110; #1;
154     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111000; #1;
155     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111010; #1;
156     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111100; #1;
157     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111110; #1;
158     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000000; #1;
159     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000010; #1;
160     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000100; #1;
161     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000110; #1;
162     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001000; #1;
163     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001010; #1;
164     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001100; #1;
165     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001110; #1;
166     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010000; #1;
```

```
166     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010000; #1;
167     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010010; #1;
168     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010100; #1;
169     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010110; #1;
170     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011000; #1;
171     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011010; #1;
172     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011100; #1;
173     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011110; #1;
174     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100000; #1;
175     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100010; #1;
176     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100100; #1;
177     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100110; #1;
178     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101000; #1;
179     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101010; #1;
180     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101100; #1;
181     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101110; #1;
182     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110000; #1;
183     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110010; #1;
184     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110100; #1;
185     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110110; #1;
186     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111000; #1;
187     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111010; #1;
188     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111100; #1;
189     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111110; #1;
190     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000000; #1;
191     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000010; #1;
192     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000100; #1;
193     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000110; #1;
194     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001000; #1;
195     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001010; #1;
196     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001100; #1;
197     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001110; #1;
198     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010000; #1;
199     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010010; #1;
200     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010100; #1;
```

Elaboración propia.

Figura 25: Módulo del banco de pruebas para el fullAdder de 4 Bits (Parte 4).

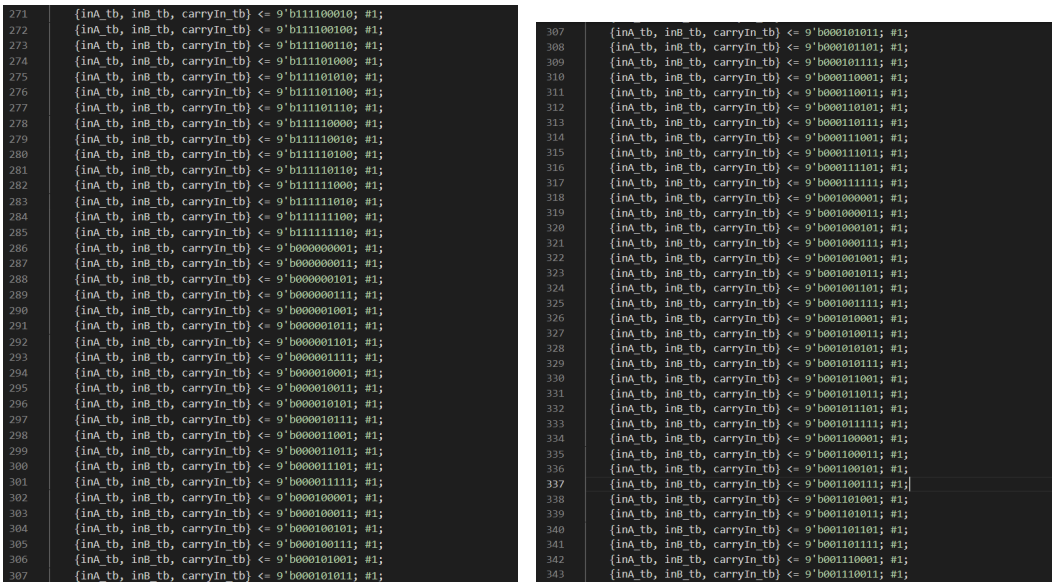


```
200     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010100; #1;
201     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010110; #1;
202     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011000; #1;
203     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011010; #1;
204     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011100; #1;
205     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011110; #1;
206     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100000; #1;
207     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100010; #1;
208     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100100; #1;
209     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100110; #1;
210     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101000; #1;
211     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101010; #1;
212     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101100; #1;
213     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101110; #1;
214     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110000; #1;
215     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110010; #1;
216     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110100; #1;
217     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110110; #1;
218     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111000; #1;
219     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111010; #1;
220     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111100; #1;
221     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111110; #1;
222     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000000; #1;
223     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000010; #1;
224     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000100; #1;
225     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000110; #1;
226     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001000; #1;
227     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001010; #1;
228     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001100; #1;
229     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001110; #1;
230     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010000; #1;
231     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010010; #1;
232     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010100; #1;
233     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010110; #1;
234     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011000; #1;
235     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011010; #1;
236     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011100; #1;
```

```
235     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011010; #1;
236     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011100; #1;
237     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011110; #1;
238     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100000; #1;
239     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100010; #1;
240     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100100; #1;
241     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100110; #1;
242     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101000; #1;
243     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101010; #1;
244     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101100; #1;
245     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101110; #1;
246     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110000; #1;
247     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110010; #1;
248     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110100; #1;
249     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110110; #1;
250     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111000; #1;
251     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111010; #1;
252     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111100; #1;
253     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111110; #1;
254     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000000; #1;
255     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000010; #1;
256     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000100; #1;
257     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000110; #1;
258     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001000; #1;
259     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001010; #1;
260     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001100; #1;
261     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001110; #1;
262     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010000; #1;
263     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010010; #1;
264     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010100; #1;
265     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010110; #1;
266     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011000; #1;
267     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011010; #1;
268     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011100; #1;
269     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011110; #1;
270     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100000; #1;
271     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100010; #1;
```

Elaboración propia.

Figura 26: Módulo del banco de pruebas para el fullAdder de 4 Bits (Parte 5).

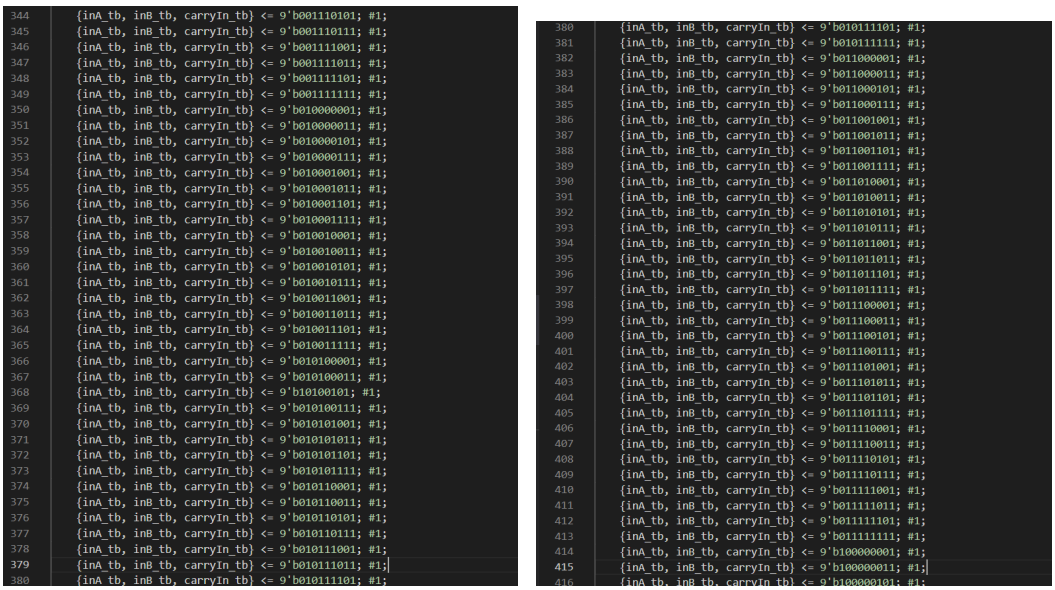


```
271     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100010; #1;
272     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100100; #1;
273     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100110; #1;
274     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101000; #1;
275     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101010; #1;
276     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101100; #1;
277     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101110; #1;
278     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110000; #1;
279     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110010; #1;
280     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110100; #1;
281     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110110; #1;
282     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111000; #1;
283     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111010; #1;
284     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111100; #1;
285     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111110; #1;
286     {inA_tb, inB_tb, carryIn_tb} <= 9'b000000001; #1;
287     {inA_tb, inB_tb, carryIn_tb} <= 9'b000000011; #1;
288     {inA_tb, inB_tb, carryIn_tb} <= 9'b000000101; #1;
289     {inA_tb, inB_tb, carryIn_tb} <= 9'b000000111; #1;
290     {inA_tb, inB_tb, carryIn_tb} <= 9'b000001001; #1;
291     {inA_tb, inB_tb, carryIn_tb} <= 9'b000001011; #1;
292     {inA_tb, inB_tb, carryIn_tb} <= 9'b000001101; #1;
293     {inA_tb, inB_tb, carryIn_tb} <= 9'b000001111; #1;
294     {inA_tb, inB_tb, carryIn_tb} <= 9'b000010001; #1;
295     {inA_tb, inB_tb, carryIn_tb} <= 9'b000010011; #1;
296     {inA_tb, inB_tb, carryIn_tb} <= 9'b000010101; #1;
297     {inA_tb, inB_tb, carryIn_tb} <= 9'b000010111; #1;
298     {inA_tb, inB_tb, carryIn_tb} <= 9'b000011001; #1;
299     {inA_tb, inB_tb, carryIn_tb} <= 9'b000011011; #1;
300     {inA_tb, inB_tb, carryIn_tb} <= 9'b000011101; #1;
301     {inA_tb, inB_tb, carryIn_tb} <= 9'b000011111; #1;
302     {inA_tb, inB_tb, carryIn_tb} <= 9'b000100001; #1;
303     {inA_tb, inB_tb, carryIn_tb} <= 9'b000100011; #1;
304     {inA_tb, inB_tb, carryIn_tb} <= 9'b000100101; #1;
305     {inA_tb, inB_tb, carryIn_tb} <= 9'b000100111; #1;
306     {inA_tb, inB_tb, carryIn_tb} <= 9'b000101001; #1;
307     {inA_tb, inB_tb, carryIn_tb} <= 9'b000101011; #1;
```

```
307     {inA_tb, inB_tb, carryIn_tb} <= 9'b000101011; #1;
308     {inA_tb, inB_tb, carryIn_tb} <= 9'b000101101; #1;
309     {inA_tb, inB_tb, carryIn_tb} <= 9'b000101111; #1;
310     {inA_tb, inB_tb, carryIn_tb} <= 9'b000110001; #1;
311     {inA_tb, inB_tb, carryIn_tb} <= 9'b000110011; #1;
312     {inA_tb, inB_tb, carryIn_tb} <= 9'b000110101; #1;
313     {inA_tb, inB_tb, carryIn_tb} <= 9'b000110111; #1;
314     {inA_tb, inB_tb, carryIn_tb} <= 9'b000111001; #1;
315     {inA_tb, inB_tb, carryIn_tb} <= 9'b000111011; #1;
316     {inA_tb, inB_tb, carryIn_tb} <= 9'b000111101; #1;
317     {inA_tb, inB_tb, carryIn_tb} <= 9'b000111111; #1;
318     {inA_tb, inB_tb, carryIn_tb} <= 9'b001000001; #1;
319     {inA_tb, inB_tb, carryIn_tb} <= 9'b001000011; #1;
320     {inA_tb, inB_tb, carryIn_tb} <= 9'b001000101; #1;
321     {inA_tb, inB_tb, carryIn_tb} <= 9'b001000111; #1;
322     {inA_tb, inB_tb, carryIn_tb} <= 9'b001001001; #1;
323     {inA_tb, inB_tb, carryIn_tb} <= 9'b001001011; #1;
324     {inA_tb, inB_tb, carryIn_tb} <= 9'b001001101; #1;
325     {inA_tb, inB_tb, carryIn_tb} <= 9'b001001111; #1;
326     {inA_tb, inB_tb, carryIn_tb} <= 9'b001010001; #1;
327     {inA_tb, inB_tb, carryIn_tb} <= 9'b001010011; #1;
328     {inA_tb, inB_tb, carryIn_tb} <= 9'b001010101; #1;
329     {inA_tb, inB_tb, carryIn_tb} <= 9'b001010111; #1;
330     {inA_tb, inB_tb, carryIn_tb} <= 9'b001011001; #1;
331     {inA_tb, inB_tb, carryIn_tb} <= 9'b001011011; #1;
332     {inA_tb, inB_tb, carryIn_tb} <= 9'b001011101; #1;
333     {inA_tb, inB_tb, carryIn_tb} <= 9'b001011111; #1;
334     {inA_tb, inB_tb, carryIn_tb} <= 9'b001100001; #1;
335     {inA_tb, inB_tb, carryIn_tb} <= 9'b001100011; #1;
336     {inA_tb, inB_tb, carryIn_tb} <= 9'b001100101; #1;
337     {inA_tb, inB_tb, carryIn_tb} <= 9'b001100111; #1;
338     {inA_tb, inB_tb, carryIn_tb} <= 9'b001101001; #1;
339     {inA_tb, inB_tb, carryIn_tb} <= 9'b001101011; #1;
340     {inA_tb, inB_tb, carryIn_tb} <= 9'b001101101; #1;
341     {inA_tb, inB_tb, carryIn_tb} <= 9'b001101111; #1;
342     {inA_tb, inB_tb, carryIn_tb} <= 9'b001110001; #1;
343     {inA_tb, inB_tb, carryIn_tb} <= 9'b001110011; #1;
```

Elaboración propia.

Figura 27: Módulo del banco de pruebas para el fullAdder de 4 Bits (Parte 6).

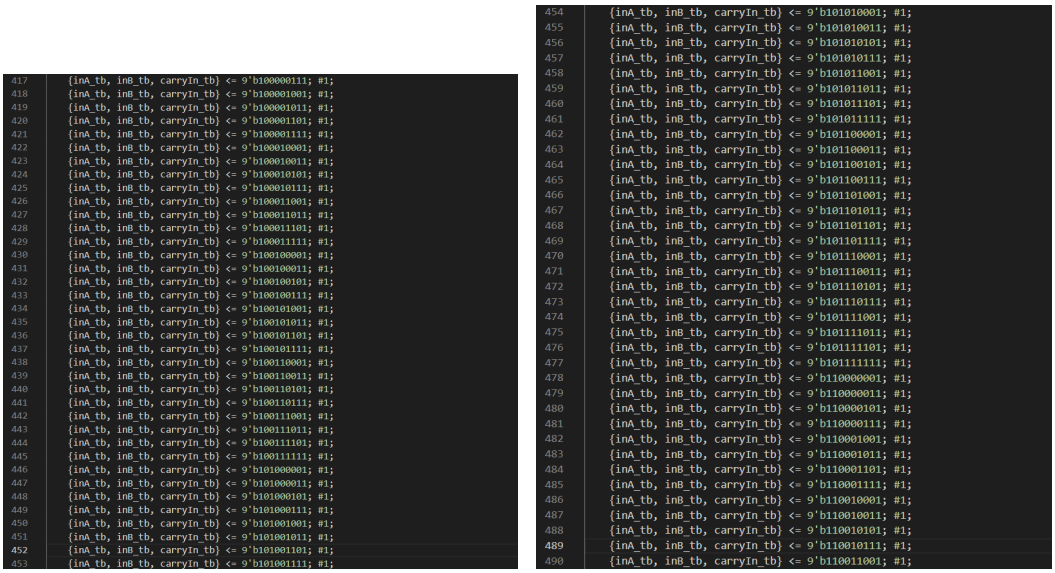


```
344     {inA_tb, inB_tb, carryIn_tb} <= 9'b001110101; #1;
345     {inA_tb, inB_tb, carryIn_tb} <= 9'b001110111; #1;
346     {inA_tb, inB_tb, carryIn_tb} <= 9'b001111001; #1;
347     {inA_tb, inB_tb, carryIn_tb} <= 9'b001111011; #1;
348     {inA_tb, inB_tb, carryIn_tb} <= 9'b001111101; #1;
349     {inA_tb, inB_tb, carryIn_tb} <= 9'b001111111; #1;
350     {inA_tb, inB_tb, carryIn_tb} <= 9'b010000001; #1;
351     {inA_tb, inB_tb, carryIn_tb} <= 9'b010000011; #1;
352     {inA_tb, inB_tb, carryIn_tb} <= 9'b010000101; #1;
353     {inA_tb, inB_tb, carryIn_tb} <= 9'b010000111; #1;
354     {inA_tb, inB_tb, carryIn_tb} <= 9'b010001001; #1;
355     {inA_tb, inB_tb, carryIn_tb} <= 9'b010001011; #1;
356     {inA_tb, inB_tb, carryIn_tb} <= 9'b010001101; #1;
357     {inA_tb, inB_tb, carryIn_tb} <= 9'b010001111; #1;
358     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010001; #1;
359     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010011; #1;
360     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010101; #1;
361     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010111; #1;
362     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011001; #1;
363     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011011; #1;
364     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011101; #1;
365     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011111; #1;
366     {inA_tb, inB_tb, carryIn_tb} <= 9'b010100001; #1;
367     {inA_tb, inB_tb, carryIn_tb} <= 9'b010100011; #1;
368     {inA_tb, inB_tb, carryIn_tb} <= 9'b10100101; #1;
369     {inA_tb, inB_tb, carryIn_tb} <= 9'b010100111; #1;
370     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101001; #1;
371     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101011; #1;
372     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101101; #1;
373     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101111; #1;
374     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110001; #1;
375     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110011; #1;
376     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110101; #1;
377     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110111; #1;
378     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111001; #1;
379     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111011; #1;
380     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111101; #1;
```

```
380     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111101; #1;
381     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111111; #1;
382     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000001; #1;
383     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000011; #1;
384     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000101; #1;
385     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000111; #1;
386     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001001; #1;
387     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001011; #1;
388     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001101; #1;
389     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001111; #1;
390     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010001; #1;
391     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010011; #1;
392     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010101; #1;
393     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010111; #1;
394     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011001; #1;
395     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011011; #1;
396     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011101; #1;
397     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011111; #1;
398     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100001; #1;
399     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100011; #1;
400     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100101; #1;
401     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100111; #1;
402     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101001; #1;
403     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101011; #1;
404     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101101; #1;
405     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101111; #1;
406     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110001; #1;
407     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110011; #1;
408     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110101; #1;
409     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110111; #1;
410     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111001; #1;
411     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111011; #1;
412     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111101; #1;
413     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111111; #1;
414     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000001; #1;
415     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000011; #1;
416     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000101; #1;
```

Elaboración propia.

Figura 28: Módulo del banco de pruebas para el fullAdder de 4 Bits (Parte 7).

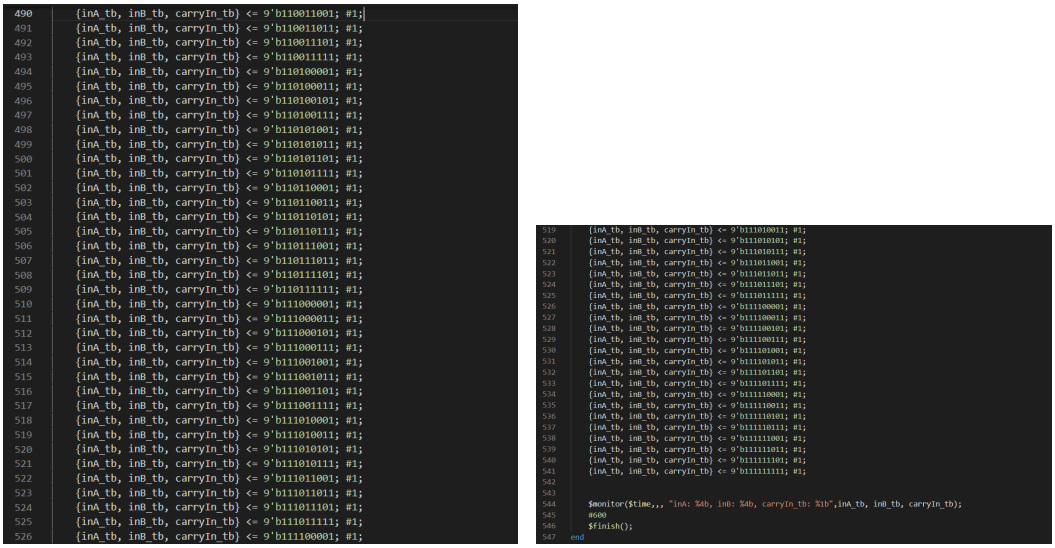


```
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100000111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100001001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100001011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100001101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100001111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100010001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100010011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100010101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100010111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100011001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100011011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100011101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100011111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100100001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100100011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100100101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100100111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100101001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100101011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100101101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100101111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100110001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100110011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100110101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100110111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100111001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100111011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100111101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b100111111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101000001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101000011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101000101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101000111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101001001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101001011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101001101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101001111; #1;
```

```
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101010001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101010011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101010101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101010111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101011001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101011011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101011101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101011111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101100001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101100011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101100101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101100111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101101001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101101011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101101101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101101111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101110001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101110011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101110101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101110111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101111001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101111011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101111101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b101111111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110000001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110000011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110000101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110000111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110001001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110001011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110001101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110001111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110010001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110010011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110010101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110010111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110011001; #1;
```

Elaboración propia.

Figura 29: Módulo del banco de pruebas para el fullAdder de 4 Bits (Parte 8).



```
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110011001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110011011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110011101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110011111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110100001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110100011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110100101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110100111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110101001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110101011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110101101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110101111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110110001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110110011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110110101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110110111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110111001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110111011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110111101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b110111111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111000001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111000011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111000101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111000111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111001001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111001011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111001101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111001111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111010001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111010011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111010101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111010111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111011001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111011011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111011101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111011111; #1;
```

```
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111100001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111100011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111100101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111100111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111101001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111101011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111101101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111101111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111110001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111110011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111110101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111110111; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111111001; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111111011; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111111101; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b111111111; #1;


    $monitor($time,,, "inA: %4b, inB: %4b, carryIn_tb: %1b",inA_tb, inB_tb, carryIn_tb);
    #600
    $finish();
end
```

Elaboración propia.

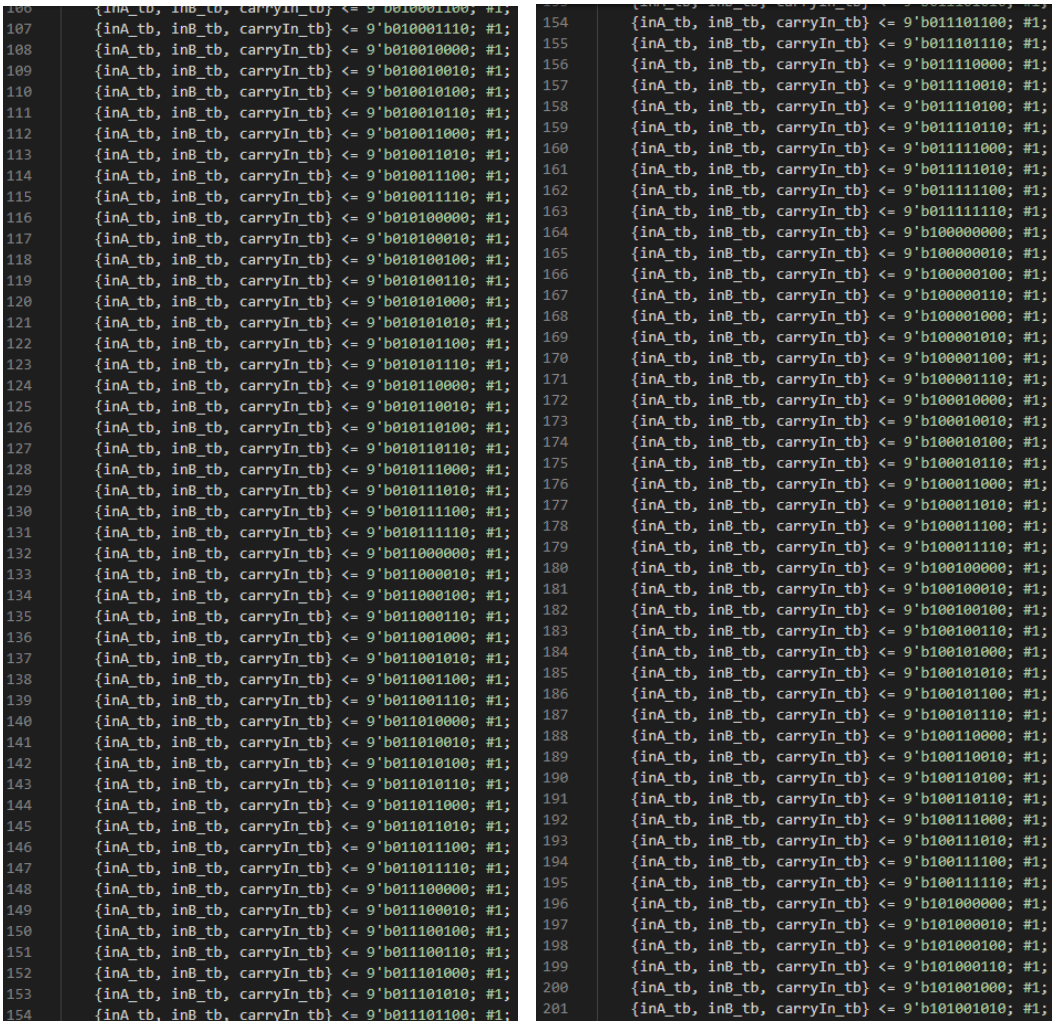
De las Figuras 30 a la 35 se adjunta el código completo del banco de pruebas del sumador BCD.

Figura 30: Módulo del banco de pruebas para el sumador BCD (Parte 1).

```
//Creación del archivo .vcd que será utilizado por GTKwave
initial begin
    $dumpfile("testBCD.vcd");
    $dumpvars;
    inA_tb = 4'b0000; inB_tb = 4'b0000; carryIn_tb = 1'b0;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000000110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000001110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000010000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000010010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000010100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000010110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000011000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000011010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000011100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000011110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000100000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000100010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000100100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000100110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000101000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000101010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000101100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000101110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000110000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000110010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000110100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000110110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000111000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000111010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000111100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b000111110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001000000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001000010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001000100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001000110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001001000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001001010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001001100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001001110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001010000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001010010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001010100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001010110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001011000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001011010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001011100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001011110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001100000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001100010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001100100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001100110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001101000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001101010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001101100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001101110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001110000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001110010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001110100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001110110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001111000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001111010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001111100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b001111110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010000000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010000010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010000100; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010000110; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010001000; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010001010; #1;
    {inA_tb, inB_tb, carryIn_tb} <= 9'b010001100; #1;
```

Elaboración propia.

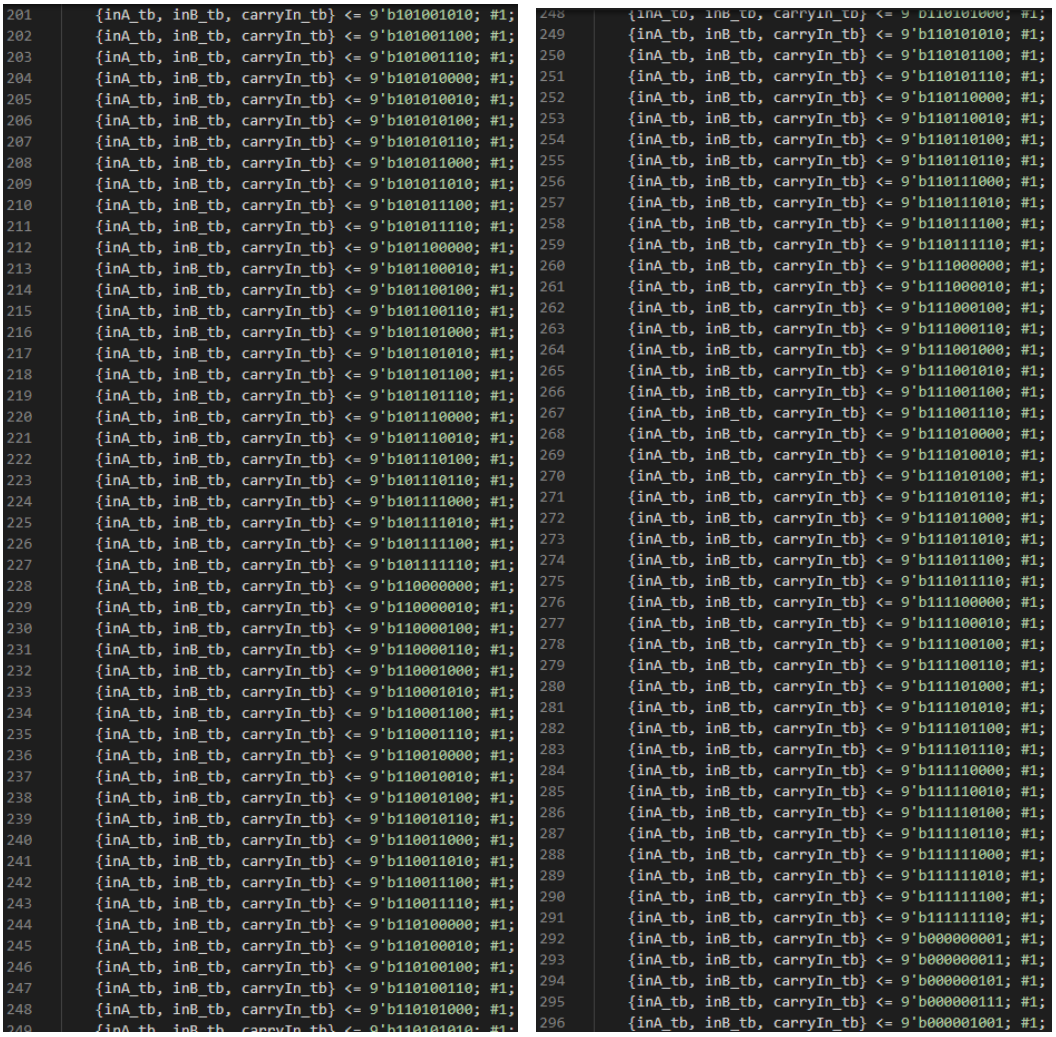
Figura 31: Módulo del banco de pruebas para el sumador BCD(Parte 2).



```
107     {inA_tb, inB_tb, carryIn_tb} <= 9'b010001110; #1;
108     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010000; #1;
109     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010010; #1;
110     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010100; #1;
111     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010110; #1;
112     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011000; #1;
113     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011010; #1;
114     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011100; #1;
115     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011110; #1;
116     {inA_tb, inB_tb, carryIn_tb} <= 9'b010100000; #1;
117     {inA_tb, inB_tb, carryIn_tb} <= 9'b010100010; #1;
118     {inA_tb, inB_tb, carryIn_tb} <= 9'b010100100; #1;
119     {inA_tb, inB_tb, carryIn_tb} <= 9'b010100110; #1;
120     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101000; #1;
121     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101010; #1;
122     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101100; #1;
123     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101110; #1;
124     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110000; #1;
125     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110010; #1;
126     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110100; #1;
127     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110110; #1;
128     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111000; #1;
129     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111010; #1;
130     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111100; #1;
131     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111110; #1;
132     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000000; #1;
133     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000010; #1;
134     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000100; #1;
135     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000110; #1;
136     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001000; #1;
137     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001010; #1;
138     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001100; #1;
139     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001110; #1;
140     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010000; #1;
141     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010010; #1;
142     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010100; #1;
143     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010110; #1;
144     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011000; #1;
145     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011010; #1;
146     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011100; #1;
147     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011110; #1;
148     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100000; #1;
149     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100010; #1;
150     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100100; #1;
151     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100110; #1;
152     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101000; #1;
153     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101010; #1;
154     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101100; #1;
154     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101100; #1;
155     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101110; #1;
156     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110000; #1;
157     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110010; #1;
158     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110100; #1;
159     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110110; #1;
160     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111000; #1;
161     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111010; #1;
162     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111100; #1;
163     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111110; #1;
164     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000000; #1;
165     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000010; #1;
166     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000100; #1;
167     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000110; #1;
168     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001000; #1;
169     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001010; #1;
170     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001100; #1;
171     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001110; #1;
172     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010000; #1;
173     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010010; #1;
174     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010100; #1;
175     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010110; #1;
176     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011000; #1;
177     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011010; #1;
178     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011100; #1;
179     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011110; #1;
180     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100000; #1;
181     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100010; #1;
182     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100100; #1;
183     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100110; #1;
184     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101000; #1;
185     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101010; #1;
186     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101100; #1;
187     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101110; #1;
188     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110000; #1;
189     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110010; #1;
190     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110100; #1;
191     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110110; #1;
192     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111000; #1;
193     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111010; #1;
194     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111100; #1;
195     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111110; #1;
196     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000000; #1;
197     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000010; #1;
198     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000100; #1;
199     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000110; #1;
200     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001000; #1;
201     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001010; #1;
```

Elaboración propia.

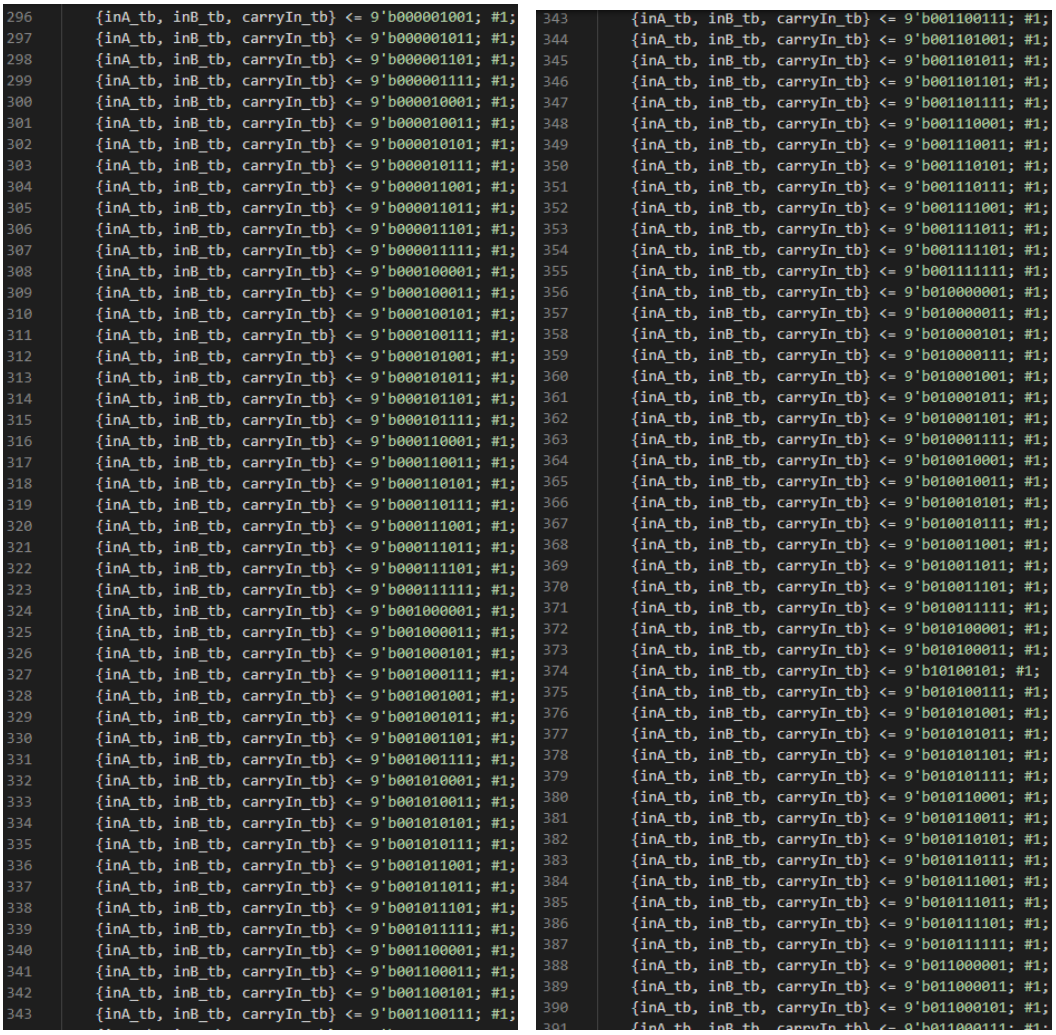
Figura 32: Módulo del banco de pruebas para el sumador BCD(Parte 3).



```
201     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001010; #1;
202     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001100; #1;
203     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001110; #1;
204     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010000; #1;
205     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010010; #1;
206     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010100; #1;
207     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010110; #1;
208     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011000; #1;
209     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011010; #1;
210     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011100; #1;
211     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011110; #1;
212     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100000; #1;
213     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100010; #1;
214     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100100; #1;
215     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100110; #1;
216     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101000; #1;
217     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101010; #1;
218     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101100; #1;
219     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101110; #1;
220     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110000; #1;
221     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110010; #1;
222     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110100; #1;
223     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110110; #1;
224     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111000; #1;
225     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111010; #1;
226     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111100; #1;
227     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111110; #1;
228     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000000; #1;
229     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000010; #1;
230     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000100; #1;
231     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000110; #1;
232     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001000; #1;
233     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001010; #1;
234     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001100; #1;
235     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001110; #1;
236     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010000; #1;
237     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010010; #1;
238     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010100; #1;
239     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010110; #1;
240     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011000; #1;
241     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011010; #1;
242     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011100; #1;
243     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011110; #1;
244     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100000; #1;
245     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100010; #1;
246     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100100; #1;
247     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100110; #1;
248     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101000; #1;
249     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101010; #1;
250     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101100; #1;
251     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101110; #1;
252     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110000; #1;
253     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110010; #1;
254     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110100; #1;
255     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110110; #1;
256     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111000; #1;
257     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111010; #1;
258     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111100; #1;
259     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111110; #1;
260     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000000; #1;
261     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000010; #1;
262     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000100; #1;
263     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000110; #1;
264     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001000; #1;
265     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001010; #1;
266     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001100; #1;
267     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001110; #1;
268     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010000; #1;
269     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010010; #1;
270     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010100; #1;
271     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010110; #1;
272     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011000; #1;
273     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011010; #1;
274     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011100; #1;
275     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011110; #1;
276     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100000; #1;
277     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100010; #1;
278     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100100; #1;
279     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100110; #1;
280     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101000; #1;
281     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101010; #1;
282     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101100; #1;
283     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101110; #1;
284     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110000; #1;
285     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110010; #1;
286     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110100; #1;
287     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110110; #1;
288     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111000; #1;
289     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111010; #1;
290     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111100; #1;
291     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111110; #1;
292     {inA_tb, inB_tb, carryIn_tb} <= 9'b000000001; #1;
293     {inA_tb, inB_tb, carryIn_tb} <= 9'b000000011; #1;
294     {inA_tb, inB_tb, carryIn_tb} <= 9'b000000101; #1;
295     {inA_tb, inB_tb, carryIn_tb} <= 9'b000000111; #1;
296     {inA_tb, inB_tb, carryIn_tb} <= 9'b000001001; #1;
```

Elaboración propia.

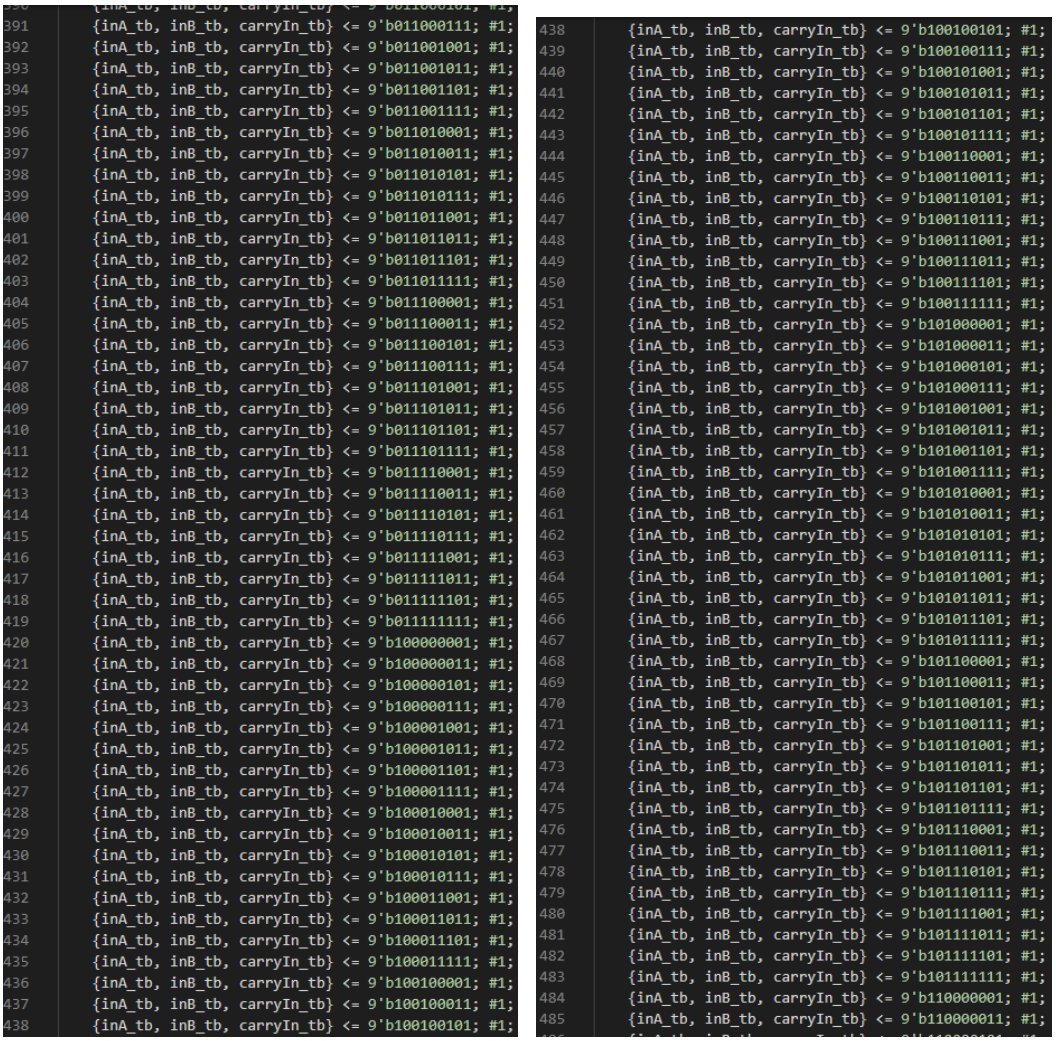
Figura 33: Módulo del banco de pruebas para el sumador BCD(Parte 4).



```
296     {inA_tb, inB_tb, carryIn_tb} <= 9'b000001001; #1;
297     {inA_tb, inB_tb, carryIn_tb} <= 9'b000001011; #1;
298     {inA_tb, inB_tb, carryIn_tb} <= 9'b000001101; #1;
299     {inA_tb, inB_tb, carryIn_tb} <= 9'b000001111; #1;
300     {inA_tb, inB_tb, carryIn_tb} <= 9'b000010001; #1;
301     {inA_tb, inB_tb, carryIn_tb} <= 9'b000010011; #1;
302     {inA_tb, inB_tb, carryIn_tb} <= 9'b000010101; #1;
303     {inA_tb, inB_tb, carryIn_tb} <= 9'b000010111; #1;
304     {inA_tb, inB_tb, carryIn_tb} <= 9'b000011001; #1;
305     {inA_tb, inB_tb, carryIn_tb} <= 9'b000011011; #1;
306     {inA_tb, inB_tb, carryIn_tb} <= 9'b000011101; #1;
307     {inA_tb, inB_tb, carryIn_tb} <= 9'b000011111; #1;
308     {inA_tb, inB_tb, carryIn_tb} <= 9'b000100001; #1;
309     {inA_tb, inB_tb, carryIn_tb} <= 9'b000100011; #1;
310     {inA_tb, inB_tb, carryIn_tb} <= 9'b000100101; #1;
311     {inA_tb, inB_tb, carryIn_tb} <= 9'b000100111; #1;
312     {inA_tb, inB_tb, carryIn_tb} <= 9'b000101001; #1;
313     {inA_tb, inB_tb, carryIn_tb} <= 9'b000101011; #1;
314     {inA_tb, inB_tb, carryIn_tb} <= 9'b000101101; #1;
315     {inA_tb, inB_tb, carryIn_tb} <= 9'b000101111; #1;
316     {inA_tb, inB_tb, carryIn_tb} <= 9'b000110001; #1;
317     {inA_tb, inB_tb, carryIn_tb} <= 9'b000110011; #1;
318     {inA_tb, inB_tb, carryIn_tb} <= 9'b000110101; #1;
319     {inA_tb, inB_tb, carryIn_tb} <= 9'b000110111; #1;
320     {inA_tb, inB_tb, carryIn_tb} <= 9'b000111001; #1;
321     {inA_tb, inB_tb, carryIn_tb} <= 9'b000111011; #1;
322     {inA_tb, inB_tb, carryIn_tb} <= 9'b000111101; #1;
323     {inA_tb, inB_tb, carryIn_tb} <= 9'b000111111; #1;
324     {inA_tb, inB_tb, carryIn_tb} <= 9'b001000001; #1;
325     {inA_tb, inB_tb, carryIn_tb} <= 9'b001000011; #1;
326     {inA_tb, inB_tb, carryIn_tb} <= 9'b001000101; #1;
327     {inA_tb, inB_tb, carryIn_tb} <= 9'b001000111; #1;
328     {inA_tb, inB_tb, carryIn_tb} <= 9'b001001001; #1;
329     {inA_tb, inB_tb, carryIn_tb} <= 9'b001001011; #1;
330     {inA_tb, inB_tb, carryIn_tb} <= 9'b001001101; #1;
331     {inA_tb, inB_tb, carryIn_tb} <= 9'b001001111; #1;
332     {inA_tb, inB_tb, carryIn_tb} <= 9'b001010001; #1;
333     {inA_tb, inB_tb, carryIn_tb} <= 9'b001010011; #1;
334     {inA_tb, inB_tb, carryIn_tb} <= 9'b001010101; #1;
335     {inA_tb, inB_tb, carryIn_tb} <= 9'b001010111; #1;
336     {inA_tb, inB_tb, carryIn_tb} <= 9'b001011001; #1;
337     {inA_tb, inB_tb, carryIn_tb} <= 9'b001011011; #1;
338     {inA_tb, inB_tb, carryIn_tb} <= 9'b001011101; #1;
339     {inA_tb, inB_tb, carryIn_tb} <= 9'b001011111; #1;
340     {inA_tb, inB_tb, carryIn_tb} <= 9'b001100001; #1;
341     {inA_tb, inB_tb, carryIn_tb} <= 9'b001100011; #1;
342     {inA_tb, inB_tb, carryIn_tb} <= 9'b001100101; #1;
343     {inA_tb, inB_tb, carryIn_tb} <= 9'b001100111; #1;
344     {inA_tb, inB_tb, carryIn_tb} <= 9'b001101001; #1;
345     {inA_tb, inB_tb, carryIn_tb} <= 9'b001101011; #1;
346     {inA_tb, inB_tb, carryIn_tb} <= 9'b001101101; #1;
347     {inA_tb, inB_tb, carryIn_tb} <= 9'b001101111; #1;
348     {inA_tb, inB_tb, carryIn_tb} <= 9'b001110001; #1;
349     {inA_tb, inB_tb, carryIn_tb} <= 9'b001110011; #1;
350     {inA_tb, inB_tb, carryIn_tb} <= 9'b001110101; #1;
351     {inA_tb, inB_tb, carryIn_tb} <= 9'b001110111; #1;
352     {inA_tb, inB_tb, carryIn_tb} <= 9'b001111001; #1;
353     {inA_tb, inB_tb, carryIn_tb} <= 9'b001111011; #1;
354     {inA_tb, inB_tb, carryIn_tb} <= 9'b001111101; #1;
355     {inA_tb, inB_tb, carryIn_tb} <= 9'b001111111; #1;
356     {inA_tb, inB_tb, carryIn_tb} <= 9'b010000001; #1;
357     {inA_tb, inB_tb, carryIn_tb} <= 9'b010000011; #1;
358     {inA_tb, inB_tb, carryIn_tb} <= 9'b010000101; #1;
359     {inA_tb, inB_tb, carryIn_tb} <= 9'b010000111; #1;
360     {inA_tb, inB_tb, carryIn_tb} <= 9'b010001001; #1;
361     {inA_tb, inB_tb, carryIn_tb} <= 9'b010001011; #1;
362     {inA_tb, inB_tb, carryIn_tb} <= 9'b010001101; #1;
363     {inA_tb, inB_tb, carryIn_tb} <= 9'b010001111; #1;
364     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010001; #1;
365     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010011; #1;
366     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010101; #1;
367     {inA_tb, inB_tb, carryIn_tb} <= 9'b010010111; #1;
368     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011001; #1;
369     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011011; #1;
370     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011101; #1;
371     {inA_tb, inB_tb, carryIn_tb} <= 9'b010011111; #1;
372     {inA_tb, inB_tb, carryIn_tb} <= 9'b010100001; #1;
373     {inA_tb, inB_tb, carryIn_tb} <= 9'b010100011; #1;
374     {inA_tb, inB_tb, carryIn_tb} <= 9'b10100101; #1;
375     {inA_tb, inB_tb, carryIn_tb} <= 9'b010100111; #1;
376     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101001; #1;
377     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101011; #1;
378     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101101; #1;
379     {inA_tb, inB_tb, carryIn_tb} <= 9'b010101111; #1;
380     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110001; #1;
381     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110011; #1;
382     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110101; #1;
383     {inA_tb, inB_tb, carryIn_tb} <= 9'b010110111; #1;
384     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111001; #1;
385     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111011; #1;
386     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111101; #1;
387     {inA_tb, inB_tb, carryIn_tb} <= 9'b010111111; #1;
388     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000001; #1;
389     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000011; #1;
390     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000101; #1;
```

Elaboración propia.

Figura 34: Módulo del banco de pruebas para el sumador BCD(Parte 5).

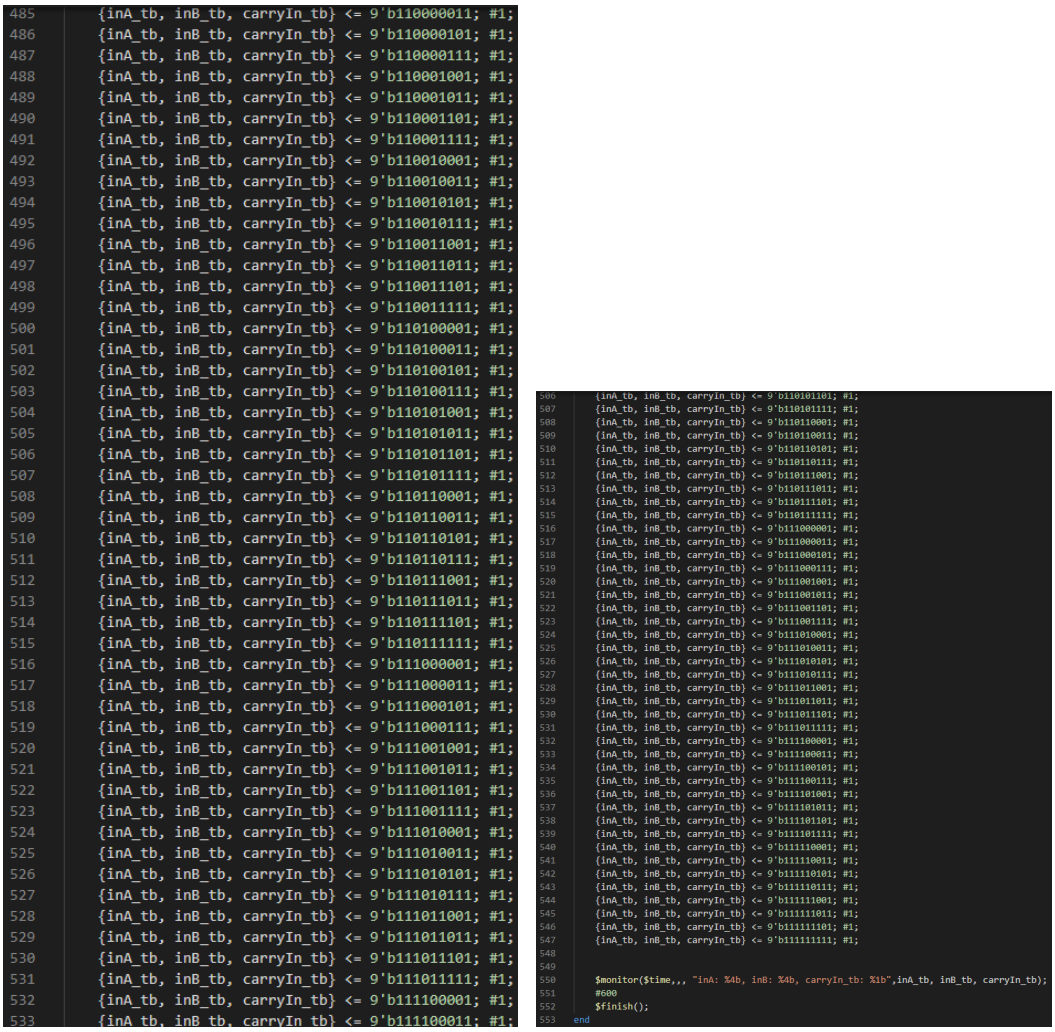


```
391     {inA_tb, inB_tb, carryIn_tb} <= 9'b011000111; #1;
392     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001001; #1;
393     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001011; #1;
394     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001101; #1;
395     {inA_tb, inB_tb, carryIn_tb} <= 9'b011001111; #1;
396     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010001; #1;
397     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010011; #1;
398     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010101; #1;
399     {inA_tb, inB_tb, carryIn_tb} <= 9'b011010111; #1;
400     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011001; #1;
401     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011011; #1;
402     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011101; #1;
403     {inA_tb, inB_tb, carryIn_tb} <= 9'b011011111; #1;
404     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100001; #1;
405     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100011; #1;
406     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100101; #1;
407     {inA_tb, inB_tb, carryIn_tb} <= 9'b011100111; #1;
408     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101001; #1;
409     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101011; #1;
410     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101101; #1;
411     {inA_tb, inB_tb, carryIn_tb} <= 9'b011101111; #1;
412     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110001; #1;
413     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110011; #1;
414     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110101; #1;
415     {inA_tb, inB_tb, carryIn_tb} <= 9'b011110111; #1;
416     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111001; #1;
417     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111011; #1;
418     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111101; #1;
419     {inA_tb, inB_tb, carryIn_tb} <= 9'b011111111; #1;
420     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000001; #1;
421     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000011; #1;
422     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000101; #1;
423     {inA_tb, inB_tb, carryIn_tb} <= 9'b100000111; #1;
424     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001001; #1;
425     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001011; #1;
426     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001101; #1;
427     {inA_tb, inB_tb, carryIn_tb} <= 9'b100001111; #1;
428     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010001; #1;
429     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010011; #1;
430     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010101; #1;
431     {inA_tb, inB_tb, carryIn_tb} <= 9'b100010111; #1;
432     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011001; #1;
433     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011011; #1;
434     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011101; #1;
435     {inA_tb, inB_tb, carryIn_tb} <= 9'b100011111; #1;
436     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100001; #1;
437     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100011; #1;
438     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100101; #1;
```

```
438     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100101; #1;
439     {inA_tb, inB_tb, carryIn_tb} <= 9'b100100111; #1;
440     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101001; #1;
441     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101011; #1;
442     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101101; #1;
443     {inA_tb, inB_tb, carryIn_tb} <= 9'b100101111; #1;
444     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110001; #1;
445     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110011; #1;
446     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110101; #1;
447     {inA_tb, inB_tb, carryIn_tb} <= 9'b100110111; #1;
448     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111001; #1;
449     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111011; #1;
450     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111101; #1;
451     {inA_tb, inB_tb, carryIn_tb} <= 9'b100111111; #1;
452     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000001; #1;
453     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000011; #1;
454     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000101; #1;
455     {inA_tb, inB_tb, carryIn_tb} <= 9'b101000111; #1;
456     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001001; #1;
457     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001011; #1;
458     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001101; #1;
459     {inA_tb, inB_tb, carryIn_tb} <= 9'b101001111; #1;
460     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010001; #1;
461     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010011; #1;
462     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010101; #1;
463     {inA_tb, inB_tb, carryIn_tb} <= 9'b101010111; #1;
464     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011001; #1;
465     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011011; #1;
466     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011101; #1;
467     {inA_tb, inB_tb, carryIn_tb} <= 9'b101011111; #1;
468     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100001; #1;
469     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100011; #1;
470     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100101; #1;
471     {inA_tb, inB_tb, carryIn_tb} <= 9'b101100111; #1;
472     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101001; #1;
473     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101011; #1;
474     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101101; #1;
475     {inA_tb, inB_tb, carryIn_tb} <= 9'b101101111; #1;
476     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110001; #1;
477     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110011; #1;
478     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110101; #1;
479     {inA_tb, inB_tb, carryIn_tb} <= 9'b101110111; #1;
480     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111001; #1;
481     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111011; #1;
482     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111101; #1;
483     {inA_tb, inB_tb, carryIn_tb} <= 9'b101111111; #1;
484     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000001; #1;
485     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000011; #1;
```

Elaboración propia.

Figura 35: Módulo del banco de pruebas para el sumador BCD(Parte 6).



```
485     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000011; #1;
486     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000101; #1;
487     {inA_tb, inB_tb, carryIn_tb} <= 9'b110000111; #1;
488     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001001; #1;
489     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001011; #1;
490     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001101; #1;
491     {inA_tb, inB_tb, carryIn_tb} <= 9'b110001111; #1;
492     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010001; #1;
493     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010011; #1;
494     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010101; #1;
495     {inA_tb, inB_tb, carryIn_tb} <= 9'b110010111; #1;
496     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011001; #1;
497     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011011; #1;
498     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011101; #1;
499     {inA_tb, inB_tb, carryIn_tb} <= 9'b110011111; #1;
500     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100001; #1;
501     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100011; #1;
502     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100101; #1;
503     {inA_tb, inB_tb, carryIn_tb} <= 9'b110100111; #1;
504     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101001; #1;
505     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101011; #1;
506     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101101; #1;
507     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101111; #1;
508     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110001; #1;
509     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110011; #1;
510     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110101; #1;
511     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110111; #1;
512     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111001; #1;
513     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111011; #1;
514     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111101; #1;
515     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111111; #1;
516     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000001; #1;
517     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000011; #1;
518     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000101; #1;
519     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000111; #1;
520     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001001; #1;
521     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001011; #1;
522     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001101; #1;
523     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001111; #1;
524     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010001; #1;
525     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010011; #1;
526     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010101; #1;
527     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010111; #1;
528     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011001; #1;
529     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011011; #1;
530     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011101; #1;
531     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011111; #1;
532     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100001; #1;
533     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100011; #1;
```

```
506     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101101; #1;
507     {inA_tb, inB_tb, carryIn_tb} <= 9'b110101111; #1;
508     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110001; #1;
509     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110011; #1;
510     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110101; #1;
511     {inA_tb, inB_tb, carryIn_tb} <= 9'b110110111; #1;
512     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111001; #1;
513     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111011; #1;
514     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111101; #1;
515     {inA_tb, inB_tb, carryIn_tb} <= 9'b110111111; #1;
516     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000001; #1;
517     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000011; #1;
518     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000101; #1;
519     {inA_tb, inB_tb, carryIn_tb} <= 9'b111000111; #1;
520     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001001; #1;
521     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001011; #1;
522     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001101; #1;
523     {inA_tb, inB_tb, carryIn_tb} <= 9'b111001111; #1;
524     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010001; #1;
525     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010011; #1;
526     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010101; #1;
527     {inA_tb, inB_tb, carryIn_tb} <= 9'b111010111; #1;
528     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011001; #1;
529     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011011; #1;
530     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011101; #1;
531     {inA_tb, inB_tb, carryIn_tb} <= 9'b111011111; #1;
532     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100001; #1;
533     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100011; #1;
534     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100101; #1;
535     {inA_tb, inB_tb, carryIn_tb} <= 9'b111100111; #1;
536     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101001; #1;
537     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101011; #1;
538     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101101; #1;
539     {inA_tb, inB_tb, carryIn_tb} <= 9'b111101111; #1;
540     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110001; #1;
541     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110011; #1;
542     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110101; #1;
543     {inA_tb, inB_tb, carryIn_tb} <= 9'b111110111; #1;
544     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111001; #1;
545     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111011; #1;
546     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111101; #1;
547     {inA_tb, inB_tb, carryIn_tb} <= 9'b111111111; #1;
548
549
550     $monitor($time,,, "inA: %4b, inB: %4b, carryIn_tb: %1b",inA_tb, inB_tb, carryIn_tb);
551     #600
552     $finish();
553 end
```

Elaboración propia.