

1. Reading Test Vectors from an external HEX file in Quartus and Questa

```
module half_adder (input a,b,
                    output [1:0] sum);

assign sum = a+b;

endmodule

module half_adder_tb_ReadFile ();

reg a,b;
wire [1:0] sum;

half_adder UUT (a, b, sum);

reg [1:0] expected_sum;
reg [3:0] test_data [0:5];
integer i;

initial
begin
    $readmemh ("adder_InputData.hex", test_data); // copy data from file into test_data array

    for (i=0; i<6; i=i+1)
    begin
        {a,b,expected_sum} = test_data[i];
        $display ("a = %b and b = %b => expected_sum = %b", a,b,expected_sum);
        #20 $display ("a = %b and b = %b => sum = %b", a,b,sum);

        if (sum != expected_sum)
            $display ("Test error at a = %b and b = %b", a,b);
    end
    $stop;
end

endmodule
```

Use the **for loop** to read the test vector **line by line**.

#20 delay was added to display the simulation results clearly in timing diagram.

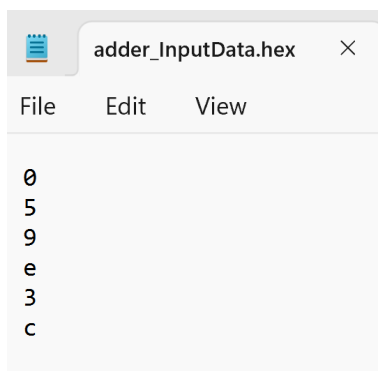
The expected_sum and the if statement are used to add a **self-checking** feature to the testbench. If the expected and actual values do not match, an error message is displayed in the transcript window.

The **adder_InputData.hex** file was created using **Notepad** and saved as a **HEX file**

Make sure to change the text file default to **All files** when you save the file in the project folder.
Verify the file type within the file explorer.

File name:	adder_InputData.hex
Save as type:	All files

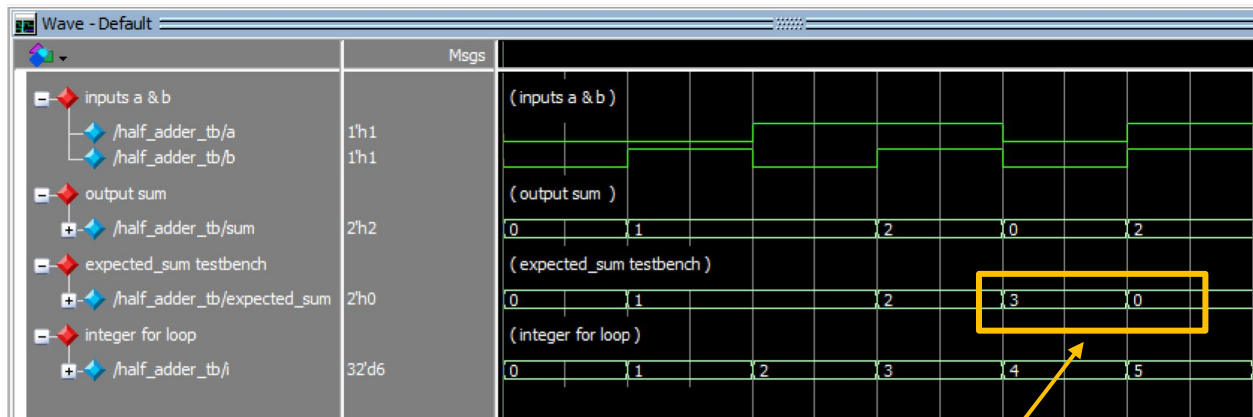
adder_InputData.hex	7/23/2024 2:57 AM	HEX File
c5_pin_model_dump	7/23/2024 3:25 AM	Text Document
half_adder	7/23/2024 2:23 AM	QPF File



a	b	expected_sum
0	0	00
0	1	01
1	0	01
0	0	11
1	1	00

2 wrong expected values inserted to verify the self-testing feature in this testbench

Simulation results in Questa



2 wrong values inserted to verify the self-testing feature in this testbench. The sum generated by the half adder has actually the correct values.

```

Transcript
work -L work -voptargs="+acc" half_adder_tb
# vsim -t lps -L altera_ver -L lpm_ver -L sgate_ver
-L altera_mf_ver -L altera_insim_ver -L cyclonev_ver
-L cyclonev_hssi_ver -L cyclonev_pcie_hip_ver -L rtl
_work -L work -voptargs="+acc" half_adder_tb
# Start time: 03:25:57 on Jul 23, 2024
# ** Note: (vsim-3812) Design is being optimized...
# Loading work.half_adder_tb(fast)
# Loading work.half_adder(fast)
#
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
# a = 0 and b = 0 => expected_sum = 00
# a = 0 and b = 0 => sum = 00
# a = 0 and b = 1 => expected_sum = 01
# a = 0 and b = 1 => sum = 01
# a = 1 and b = 0 => expected_sum = 01
# a = 1 and b = 0 => sum = 01
# a = 1 and b = 1 => expected_sum = 10
# a = 1 and b = 1 => sum = 10
# a = 0 and b = 0 => expected_sum = 11
# a = 0 and b = 0 => sum = 00
# Test error at a = 0 and b = 0
# a = 1 and b = 1 => expected_sum = 00
# a = 1 and b = 1 => sum = 10
# Test error at a = 1 and b = 1
    
```

The test vectors for the simulation data are read from the HEX file

adder_InputData.hex		
File	Edit	View
0	0_0_00=0	
5	0_1_01=5	
9	1_0_01=9	
e	1_1_10=e	
3	0_0_11=3	
c	1_1_00=c	

2. Modifying the testbench so that the test results can be saved in a HEX file

```

module half_adder_tb ();
reg      a,b;
wire [1:0] sum;

half_adder uut (a, b, sum);

reg [1:0] expected_sum;
reg [3:0] test_data [0:5];
integer i;
integer f;                                     // Numerical Identifier for the output file

initial
begin
    $readmemh ("adder_InputData.hex", test_data);           // copy data from file into test_data array

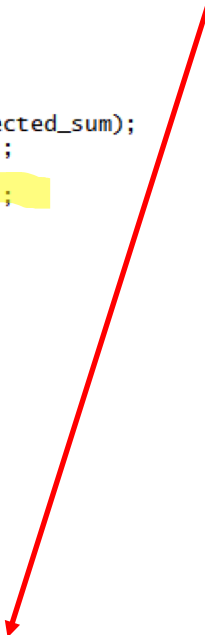
    // Copy simulation results in a simResults.hex file
    f = $fopen ("simResults.hex","w");                      // Appears in the simulation folder
    $fwrite (f, "a      b      sum      expected_sum \n");








    for (i=0; i<6; i=i+1)
    begin
        {a,b,expected_sum} = test_data[i];
        $display ("a = %b and b = %b => expected_sum = %b", a,b,expected_sum);
        #20 $display ("a = %b and b = %b => sum          = %b", a,b,sum);

        $fwrite (f, "%h      %h => %h      %h \n", a,b,sum, expected_sum);


        if (sum != expected_sum)
            $display ("Test error at a = %b and b = %b", a,b);
    end
    $fclose(f);
    $stop;
end
endmodule

```






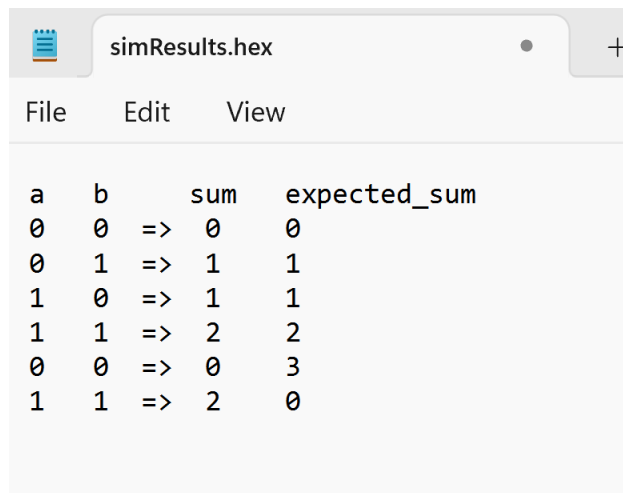
	db	7/23/2024 3:25 AM	File folder	
	incremental_db	7/23/2024 2:45 AM	File folder	
	output_files	7/23/2024 3:25 AM	File folder	
	<input checked="" type="checkbox"/> simulation	7/23/2024 2:46 AM	File folder	
	adder_InputData.hex	7/23/2024 2:57 AM	HEX File	1 KB
	c5_pin_model_dump	7/23/2024 3:25 AM	Text Document	5 KB
	half_adder	7/23/2024 2:23 AM	QPF File	2 KB

☐ Name

☒  questa

Project_folder>simulation>questa>simResults.hex

	msim_transcript	7/23/2024 3:26 AM	File	
	simResults.hex	7/23/2024 3:26 AM	HEX File	
	vsim.wlf	7/23/2024 3:26 AM	WLF File	



The image shows a text editor window with the title 'simResults.hex'. The window contains a table with four columns: 'a', 'b', 'sum', and 'expected_sum'. The table lists seven test cases, each on a new line. The first three cases show correct results, while the last two show mismatches between the actual sum and the expected sum.

a	b	sum	expected_sum
0	0	=> 0	0
0	1	=> 1	1
1	0	=> 1	1
1	1	=> 2	2
0	0	=> 0	3
1	1	=> 2	0

We can also add the test error message within the HEX file by using \$fwrite under the if statement for error detection.