Test bench design example

```
// n-bit adder with loop.
module nbit_adder_for( x, y, c_in, sum, c_out);

parameter n = 4;
input     [n-1:0] x, y;  // 4-bit array
input     c_in;
output reg [n-1:0] sum; // a 4-bit array
output reg c_out;
reg       co;
integer i;

// specify the function of an n-bit adder with loop.
always @(x or y or c_in)
  begin
  co = c_in;
  for (i = 0; i < n; i = i + 1)
     {co, sum[i]} = x[i] + y[i] + co;
  c_out = co;
  end
endmodule
```

1. test bench ex.0 : Exhaustive test

```
// Test bench /Ex. 0: Exhaustive test.
`timescale 1 ns / 100 ps
module nbit_adder_for_tb;

parameter N = 3;
reg  [N-1:0] x, y;
reg          c_in;
wire [N-1:0] sum;
wire c_out;

// Unit Under Test port map
nbit_adder_for UUT (.x(x), .y(y), .c_in(c_in), .sum(sum), .c_out(c_out));
reg [2*N-1:0] i;
initial
    for (i = 0; i <= 2**(2*N)-1; i = i + 1)
        begin
        x[N-1:0] = i[2*N-1:N];
        y[N-1:0] = i[N-1:0];
        c_in =1'b0;
        #20;
        end
initial
     #800 $finish;
initial
    $monitor($realtime,"ns %h %h %h %h", x, y, c_in, {c_out, sum});
endmodule
```

2. Test bench ex.1 : Random test

```
// Test bench Ex.1: Random test
`timescale 1 ns / 100 ps
module nbit_adder_for_tb1;

parameter N = 4;
reg  [N-1:0] x, y;
reg          c_in;
wire [N-1:0] sum;
wire c_out;

// Unit Under Test port map
nbit_adder_for UUT (.x(x), .y(y), .c_in(c_in), .sum(sum), .c_out(c_out));

integer i;
reg [N:0] test_sum;

initial
    for (i = 0; i <= 2*N ; i = i +  1)
          begin
          x = $random % 2**N;
          y = $random % 2**N;
          c_in =1'b0;
          test_sum = x +  y;
          #15;
          if (test_sum != {c_out, sum})
            $display("Error iteration %h", i);
          #5;
          end

initial
     #200 $finish;

initial
    $monitor($realtime,"ns %h %h %h %h", x, y, c_in, {c_out, sum});

endmodule
```

3. Test bench ex.2 : Golden vectors

```
// Test bench Ex.2 - Golden vectors
`timescale 1 ns / 100 ps
module nbit_adder_for_tb2;

// internal signals declarations:
parameter N = 4;
parameter M = 8;
reg  [N-1:0] x, y;
reg  c_in;
wire [N-1:0] sum;
wire c_out;

// Unit Under Test port map
nbit_adder_for UUT (.x(x),      .y(y),    .c_in(c_in),      .sum(sum),      .c_out(c_out));

integer i;
reg [N-1:0] x_array [M-1:0];
reg [N-1:0] y_array [M-1:0];
reg [N:0] expected_sum_array [M-1:0];

initial
  begin  // reading verification vector files
    $readmemh("inputx.txt", x_array);
    $readmemh("inputy.txt", y_array);
    $readmemh("sum.txt", expected_sum_array);
  end

initial
    for (i = 0; i <= M - 1 ; i = i + 1)
        begin
        x = x_array[i];
        y = y_array[i];
        c_in =1'b0;
        #15;
        if (expected_sum_array[i] !=  {c_out, sum})
           $display("Error iteration %h", i);
        #5;
        end

initial
     #200 $finish;

initial
    $monitor($realtime,"ns %h %h %h %h", x, y, c_in, {c_out, sum});

endmodule
```

inputx.txt

```
4
9
d
5
1
6
d
9
```

inputy.txt

```
1
3
d
2
d
d
c
6
```

sum.txt

```
05
0c
1a
07
0e
13
19
0f
```