```
1    `timescale 1ns/10ps
2    `define CYCLE 50.0
3    `define DATA_NUM 100
4    `define FILE_A "./a.txt"
5    `define FILE_B "./b.txt"
6    `define FILE_D "./d.txt"
```

宣告 parameter

CYCLE 設為 50

Testing data 100 筆

FILE_A 為 input a 的 txt 檔案路徑

FILE_B 為 input b 的 txt 檔案路徑

FILE_D 為 outcome 寫回 txt 檔案的檔案路徑

```
module testbench();
    integer file_a, file_b, file_d;

    reg CLK = 0;
    reg RST = 0;
    reg [31:0] data_a [0:`DATA_NUM - 1];
    reg [31:0] data_b [0:`DATA_NUM - 1];
    reg [31:0] input_a;
    reg [31:0] input_b;
    wire [31:0] outcome;
    reg [31:0] answer;

    FXP_adder test_module( .a(input_a), .b(input_b), .d(outcome));

    //Post_sim 使用，在nc_post_syn.f 中define SDF_FILE與SDF_FILE路徑
    `ifdef SDF_FILE
        initial $sdf_annotate(`SDF_FILE, test_module);
    `endif
```

宣告陣列 data_a, data_b 來存放 input

Outcome 為電路之輸出

Answer 為正確答案，用來檢查 outcome 是否正確

SDF_FILE 用於 Post_sim

```
// 設定clock訊號
always begin #(`CYCLE/2) CLK = ~CLK; end

integer i, flag=0, error=0, garbage;

//將file_a與file_b中資料存入data_a與data_b陣列中
initial begin
    file_a = $fopen(    `FILE_A , "r");
    file_b = $fopen(    `FILE_B , "r");
    file_d = $fopen(    `FILE_D , "w");
    for (i = 0; i < `DATA_NUM; i=i+1)
    begin
        garbage = $fscanf(file_a, "%X", data_a[i]);
        garbage = $fscanf(file_b, "%X", data_b[i]);
    end
end
```

設定 Clock 訊號

將 a.txt 檔中的輸入讀入 data_a 陣列，b.txt 檔中的輸入讀入 data_b 陣列

讀檔驗證方法

```
initial begin
    //File approach
    $display("-------------------------------------\n");
    $display("-          FXP_ADDER_USE_FILE         -\n");
    $display("-------------------------------------\n");

    CLK = 0;
    RST = 1;
    #(`CYCLE*2);
    RST = 0;
    for(i=0; i<`DATA_NUM; i=i+1)
    begin
        input_a = data_a[i];
        input_b = data_b[i];
        answer = $signed(input_a) + $signed(input_b);
        #(`CYCLE);
        $fwrite(file_d, "%X\n", outcome);
        if(answer !== outcome)
        begin
            error = error+1;
            if(1 || flag==0)
            begin
                $display("-------------------------------------\n");
                $display("Output error at #%d\n", i+1);
                $display("The input A is     : %X\n", input_a);
                $display("The input B is     : %X\n", input_b);
                $display("The answer is      : %X\n", answer);
                $display("Your module output: %X\n", outcome);
                $display("-------------------------------------\n");
                flag = 1;
            end //if flag
        end //if
    end //for
    if(flag==1)//if wrong
    begin
        $display("Total %4d error in %4d testdata.\n", error, i);
        $display("-------------------------------------\n");
    end//if
    else
    begin//if right
        $display("-------------------------------------\n");
        $display("All testdata correct!\n");
        $display("-------------------------------------\n");
    end//else
```

將 data_a 與 data_b 的資料傳進電路，testbench 算答案存在 answer 用於之後跟 outcome 做比對，如果有錯，會印出錯誤的 outcome 與正確答案。
並將 outcome 寫入 d.txt 檔案中。

```
//random approach
$display("------------------------------------\n");
$display("-         FXP_ADDER_USE_RANDOM         -\n");
$display("------------------------------------\n");

flag=0;
error=0;
CLK = 0;
RST = 1;
#(`CYCLE*2);
RST = 0;
for(i=0; i<`DATA_NUM; i=i+1)
begin
    input_a = $random % 2147483647;//產生介於 -(2^31)-1到(2^31)-1的數
    input_b = $random % 2147483647;
    answer = $signed(input_a) + $signed(input_b);
    #(`CYCLE);
    if(answer !== outcome)
    begin
        error = error+1;
        if(1 || flag==0)
        begin
                $display("------------------------------------\n");
                $display("Output error at #%d\n", i+1);
                $display("The input A is    : %X\n", input_a);
                $display("The input B is    : %X\n", input_b);
                $display("The answer is     : %X\n", answer);
                $display("Your module output: %X\n", outcome);
                $display("------------------------------------\n");
                flag = 1;
        end //if flag
    end //if
end //for
if(flag==1)//if wrong
begin
    $display("Total %4d error in %4d testdata.\n", error, i);
    $display("------------------------------------\n");
end//if
else
begin//if right
    $display("------------------------------------\n");
    $display("All testdata correct!\n");
    $display("------------------------------------\n");
end//else
```

Random 產生測資驗證方法一致，差別在於 input_a 與 input_b 使用$random 函數產生介於 -(2^31)-1 到(2^31)-1 的數