Bangladesh University of Engineering and Technology
Department of Electrical and Electronic Engineering
EEE 303: Digital Electronics (July 2022)

Assignment 1 (Replaces Class Test 2)

Time: 20 min **Due Date:** 01 Feb 2023 **Total Marks:** 20 **Term**: July 2022

Student Name: Md. Touhidul Islam

Student ID 1 8 0 6 1 7 7

Your code solutions should be your own work. Partial marks may be given for incomplete efforts, however, plagiarized or copied solutions will result in ZERO in this assignment.

This file is to be used as a template for submitting your solution.

Q1. Calculate the Parameters X, Y, T. Write your answers in your script. These parameters are to be used in Q2 and Q3

X = 32 bit number with your student ID in BCD
(eg if your student ID is 1706001, X=01706001$|_{16}$ = 24141825$|_{10}$)
Y = 8 bit number, with each byte of X (eg if your student ID is 1706001, Y=(01+70+60+01)$|_{16}$ = D2$|_{16}$)
T = ((X*3+5) mod 16) (eg if your student ID is 1706001, T = 8)

Ans:

X = 01806177$|_{16}$ = (0000 0001 1000 0000 0110 0001 0111 0111)$_2$ = (25190775)$_{10}$

(01 + 80 + 61 + 77)$|_{16}$ = 159$|_{16}$ = (1 0101 1001)$_2$; So Y is 9 bit. Trimming the MSB, Y = 0101 1001 $|_2$ = 89$|_{10}$

T = (1806177*3 + 5) mod 16 = 5418536 mod 16 = 8$|_{10}$

Q2. A Verilog function has input [4:0]A, Clk, I and S, and output [7:0]B. The system has a memory [3:0]M. If S=0, the output will be in high-impedance state, regardless of the state of the Clk. If S is changed from 1 to 0, output will be immediately high-impedance state (S is Asynchronous, or Clk independent).

If S=1, the output will change only at the positive edge of Clk (0 to 1 transition):
If I=1, the M will be initialized to 4'b0000 (M must be initialized by your test bench). B will be 8'x0
If I =0, then M and B will be updated in the following manner:
- **If (M=0): B will be the lowest (0th) byte of X (eg. 01 of the 01706001)**
- **If (M=1): B will be the 1st byte of X (eg. 60 of the 01706001)**
- **If (M=2): B will be the 2nd byte of X (eg. 70 of the 01706001)**
- **If (M=3): B will be the upper (3rd) byte of X (eg. 01 of the 01706001)**
- **If (M>3 and M < 16): B = Fibb(M-2) + Y + A**
- **M will be incremented by 1**

Ans:

Explain your logic how you will solve the problem:
We will need the Fibb function in our Verilog module. So first we will make the fibb function. This can be done simply by case statements as there are limited number of possible inputs in the Fibb function(only 16) for all practical purpose.

Then in the main module, we will have to create memory M as reg, as it will have to hold its value. Then we need an always block that will trigger at posedge of clk. Inside this always block we can obtain the necessary functionality for S =1 using case statements.

We will need another always block that will trigger for all value change of S. We will need this, as S is clk independent. Whenever S becomes zero, this block will make B = 8'bz.

Verilog Code:

// Your Verilog code goes here for the module.

```
// Fibb function [FINAL]
// Code your design here
function [7:0] fibb_res(input [3:0] M);
    case(M)
      0: fibb_res = 0;
      1: fibb_res = 1;
      2: fibb_res = 1;
      3: fibb_res = 2;
      4: fibb_res = 3;
      5: fibb_res = 5;
      6: fibb_res = 8;
      7: fibb_res = 13;
      8: fibb_res = 21;
      9: fibb_res = 34;
      10: fibb_res = 55;
      11: fibb_res = 89;
      12: fibb_res = 144;
      13: fibb_res = 233;
      14: fibb_res = 8'b0111_1001; //trimming off the MSB to keep it
8 bit
      15: fibb_res = 8'b0110_0010; // trimming off MSB
      default: fibb_res = 0;
    endcase
endfunction

// Code your design here
module myMod(input [4:0] A, input Clk, I, S, output reg [7:0] B);
  reg [3:0] M;
  reg [31:0] X = 32'h01806177;
  reg [7:0] Y = 8'b0101_1001;
  //reg [7:0] fibb_res;
  reg [3:0] M_next;

  always @(posedge Clk) begin
    if(S==1) begin
      if(I==1) begin
        B <= 8'h0;
      end else begin
        case(M)
          0: B <= X[7:0];
          1: B <= X[15:8];
          2: B <= X[23:16];
          3: B <= X[31:24];
          default: begin
            //fibb_res <= Fibb(M-2);
            B <= fibb_res(M-2) + Y + A;
          end
```

```
        endcase
        M_next = M + 1;
        if(M_next>=16) M_next = 0;
        M <= M_next;
      end
    end
  end

  always@(S)
    begin
      if (S==0) begin
        B = 8'bz;
      end
    end
endmodule
```

Q3. Design an appropriate test bench in SystemVerilog to test the module in Q2.
Your test bench should show output for the function for representative test cases of the input, with time-step of T between each successive steps. Show waveform for your solution. Waveform should include both binary and Hexadecimal value of the required variables, and should be clearly legible in a printout. Use EDAPlayground for this part.

Verilog Code:
// Your testbench code goes here for the module.
Verilog testbench for representative test cases of input:

```
// Code your testbench here
// or browse Examples
module main_tb();
  reg [4:0] A;
  reg Clk, I, S;
  wire [7:0] B;
  int count;

  myMod DUT(A, Clk, I, S, B);

  // initializing memory M of the DUT
  initial
    begin
      Clk = 0;
      DUT.M = 4'b0000;
    end

  //checking test cases
  initial
    begin

      $display("checking all possible combinations");
      for(count = 0; count<32; count = count+1) begin
```

```
        // using time step = 8 as T = 8
        A = count; S = 0; I = 0; #8;
        S = 0; I = 1; #8;
        S = 1; I = 0; #8;
        S = 1; I = 1; #8;
      end

      $finish;
    end

  // Clock generator
  always #1 Clk = ~Clk;

  // console monitoring
  initial
    begin
      $monitor("Clk = %b, A = %d, S = %b, I = %b, B = %h",Clk, A, S,
I, B);
    end

  // Waveform generation
  initial begin
    $dumpfile("myTestbench.vcd");
    $dumpvars;
  end

endmodule
```

Output:

// Paste the output here

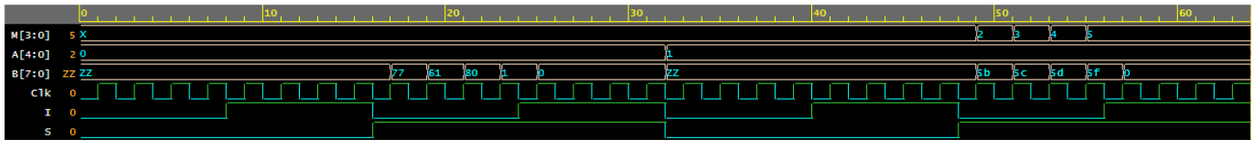
```
[2023-01-28 12:06:24 EST] iverilog '-Wall' '-g2012' design.sv testbench.sv  && unbuffer vvp
a.out
checking all possible combinations
VCD info: dumpfile myTestbench.vcd opened for output.
Clk = 0, A =  0, S = 0, I = 0, B = zz
Clk = 1, A =  0, S = 0, I = 0, B = zz
Clk = 0, A =  0, S = 0, I = 0, B = zz
Clk = 1, A =  0, S = 0, I = 0, B = zz
Clk = 0, A =  0, S = 0, I = 0, B = zz
Clk = 1, A =  0, S = 0, I = 1, B = zz
Clk = 0, A =  0, S = 0, I = 1, B = zz
Clk = 1, A =  0, S = 0, I = 1, B = zz
Clk = 0, A =  0, S = 0, I = 1, B = zz
Clk = 1, A =  0, S = 0, I = 1, B = zz
Clk = 0, A =  0, S = 1, I = 0, B = zz
Clk = 1, A =  0, S = 1, I = 0, B = 77
Clk = 0, A =  0, S = 1, I = 0, B = 77
Clk = 1, A =  0, S = 1, I = 0, B = 61
Clk = 0, A =  0, S = 1, I = 0, B = 61
Clk = 1, A =  0, S = 1, I = 1, B = 00
Clk = 0, A =  0, S = 1, I = 1, B = 00
Clk = 1, A =  0, S = 1, I = 1, B = 00
Clk = 0, A =  0, S = 1, I = 1, B = 00
Clk = 1, A =  0, S = 1, I = 1, B = 00
Clk = 0, A =  1, S = 0, I = 0, B = zz
Clk = 1, A =  1, S = 0, I = 0, B = zz
Clk = 0, A =  1, S = 0, I = 0, B = zz
```

```
Clk = 1, A =  1, S = 0, I = 0, B = zz
Clk = 0, A =  1, S = 0, I = 0, B = zz
Clk = 1, A =  1, S = 0, I = 1, B = zz
Clk = 0, A =  1, S = 0, I = 1, B = zz
Clk = 1, A =  1, S = 0, I = 1, B = zz
Clk = 0, A =  1, S = 0, I = 1, B = zz
Clk = 1, A =  1, S = 0, I = 1, B = zz
Clk = 0, A =  1, S = 1, I = 0, B = zz
Clk = 1, A =  1, S = 1, I = 0, B = 80
Clk = 0, A =  1, S = 1, I = 0, B = 80
Clk = 1, A =  1, S = 1, I = 0, B = 01
Clk = 0, A =  1, S = 1, I = 0, B = 01
Clk = 1, A =  1, S = 1, I = 1, B = 00
Clk = 0, A =  1, S = 1, I = 1, B = 00
Clk = 1, A =  1, S = 1, I = 1, B = 00
Clk = 0, A =  1, S = 1, I = 1, B = 00
Clk = 1, A =  1, S = 1, I = 1, B = 00
Clk = 0, A =  2, S = 0, I = 0, B = zz
Clk = 1, A =  2, S = 0, I = 0, B = zz
Clk = 0, A =  2, S = 0, I = 0, B = zz
Clk = 1, A =  2, S = 0, I = 0, B = zz
Clk = 0, A =  2, S = 0, I = 0, B = zz
Clk = 1, A =  2, S = 0, I = 1, B = zz
Clk = 0, A =  2, S = 0, I = 1, B = zz
Clk = 1, A =  2, S = 0, I = 1, B = zz
Clk = 0, A =  2, S = 0, I = 1, B = zz
Clk = 1, A =  2, S = 0, I = 1, B = zz
Clk = 0, A =  2, S = 1, I = 0, B = zz
Clk = 1, A =  2, S = 1, I = 0, B = 5c
Clk = 0, A =  2, S = 1, I = 0, B = 5c
Clk = 1, A =  2, S = 1, I = 0, B = 5d
Clk = 0, A =  2, S = 1, I = 0, B = 5d
Clk = 1, A =  2, S = 1, I = 1, B = 00
```

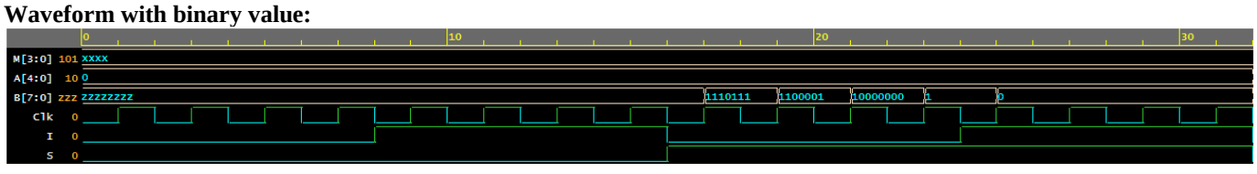
….

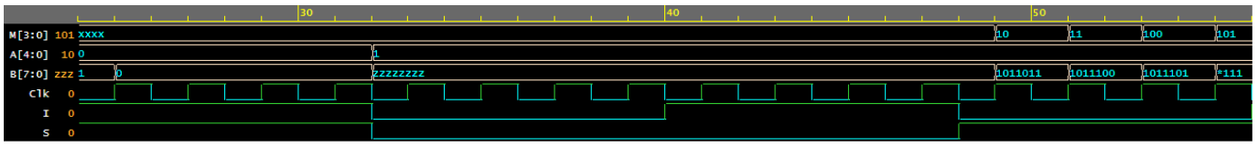
Paste wave forms from GTKWave here

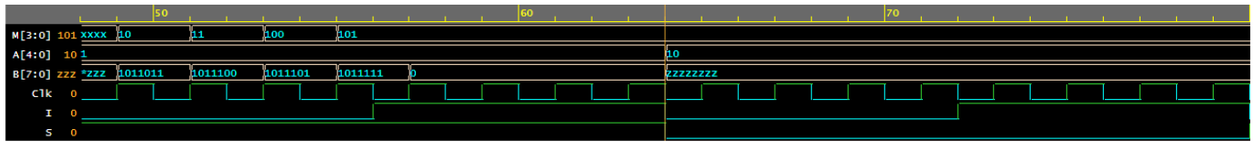
Waveform with hexadecimal value:



Waveform with binary value:







Q4. Modify the testbench in Q3, so that the Verilog module is given the digits of X sequentially in each clock pulse in a repetitive manner.

Verilog Code:

// Your testbench code goes here for the module.

```
// Code your testbench here
// or browse Examples
module main_tb();
  reg [4:0] A;
```

```
reg Clk, I, S;

reg [31:0] X_main = 32'h01806177;
reg [31:0] X_temp = 32'h01806177;
reg [31:0] X_in = 0;

reg [1:0] i = 0;
reg [10:0] a, b;
wire [7:0] B;
int  count;

myMod DUT(A, Clk, I, S, B);

// initializing memory M of the DUT
initial
  begin
    Clk = 0;
    DUT.M = 4'b0000;
    DUT.X = 32'b0;
  end

//checking test cases
initial
  begin

    $display("checking all possible combinations");
    for(count = 0; count<32; count = count+1) begin
      // using time step = 8 as T = 8
      A = count; S = 0; I = 0; #8;
      S = 0; I = 1; #8;
      S = 1; I = 0; #8;
      S = 1; I = 1; #8;
    end

    $finish;
  end

// Clock generator
always begin
  #1 Clk = ~Clk;

  // sequentially giving digits of X to the design module at every clock pulse
  X_in[3:0] = X_temp[3:0];
  DUT.X = X_in;

  $display("X input going to module = %h", X_in);

  X_in = X_in << 4;
  X_temp = X_temp >> 4;
  if (X_temp == 0) X_temp = X_main;

end


// console monitoring
initial
  begin
    $monitor("Clk = %b,  A = %d, S = %b, I = %b, B = %h",Clk,A, S, I, B);
  end

// Waveform generation
initial begin
```

```
    $dumpfile("myTestbench.vcd");
    $dumpvars;
  end

endmodule
```

Output:

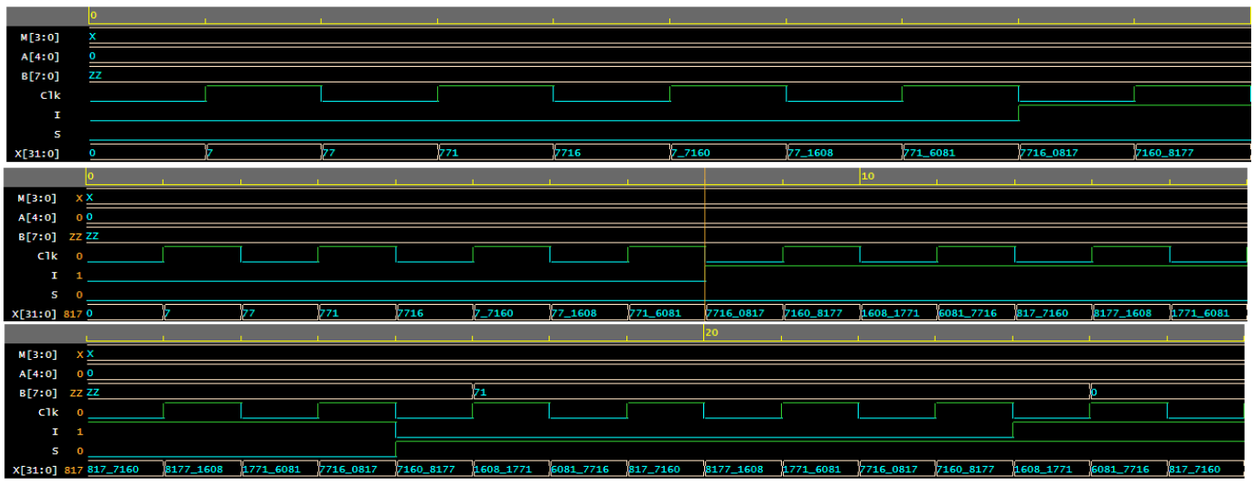
// Paste the output here

```
[2023-01-28 12:22:15 EST] iverilog '-Wall' '-g2012' design.sv testbench.sv  && unbuffer vvp a.out
checking all possible combinations
VCD info: dumpfile myTestbench.vcd opened for output.
Clk = 0,  A =  0, S = 0, I = 0, B = zz
X input going to module = 00000007
Clk = 1,  A =  0, S = 0, I = 0, B = zz
X input going to module = 00000077
Clk = 0,  A =  0, S = 0, I = 0, B = zz
X input going to module = 00000771
Clk = 1,  A =  0, S = 0, I = 0, B = zz
X input going to module = 00007716
Clk = 0,  A =  0, S = 0, I = 0, B = zz
X input going to module = 00077160
Clk = 1,  A =  0, S = 0, I = 1, B = zz
X input going to module = 00771608
Clk = 0,  A =  0, S = 0, I = 1, B = zz
X input going to module = 07716081
Clk = 1,  A =  0, S = 0, I = 1, B = zz
X input going to module = 77160817
Clk = 0,  A =  0, S = 0, I = 1, B = zz
X input going to module = 71608177
Clk = 1,  A =  0, S = 0, I = 1, B = zz
X input going to module = 16081771
Clk = 0,  A =  0, S = 1, I = 0, B = zz
X input going to module = 60817716
X received by module = 60817716
Clk = 1,  A =  0, S = 1, I = 0, B = 16
X input going to module = 08177160
Clk = 0,  A =  0, S = 1, I = 0, B = 16
X input going to module = 81771608
X received by module = 81771608
Clk = 1,  A =  0, S = 1, I = 0, B = 16
X input going to module = 17716081
Clk = 0,  A =  0, S = 1, I = 0, B = 16
X input going to module = 77160817
Clk = 1,  A =  0, S = 1, I = 1, B = 00
X input going to module = 71608177
Clk = 0,  A =  0, S = 1, I = 1, B = 00
X input going to module = 16081771
Clk = 1,  A =  0, S = 1, I = 1, B = 00
X input going to module = 60817716
Clk = 0,  A =  0, S = 1, I = 1, B = 00
X input going to module = 08177160
Clk = 1,  A =  0, S = 1, I = 1, B = 00
X input going to module = 81771608
Clk = 0,  A =  1, S = 0, I = 0, B = zz
X input going to module = 17716081
Clk = 1,  A =  1, S = 0, I = 0, B = zz
X input going to module = 77160817
Clk = 0,  A =  1, S = 0, I = 0, B = zz
X input going to module = 71608177
Clk = 1,  A =  1, S = 0, I = 0, B = zz
X input going to module = 16081771
Clk = 0,  A =  1, S = 0, I = 0, B = zz
X input going to module = 60817716
Clk = 1,  A =  1, S = 0, I = 1, B = zz
X input going to module = 08177160
Clk = 0,  A =  1, S = 0, I = 1, B = zz
X input going to module = 81771608
Clk = 1,  A =  1, S = 0, I = 1, B = zz
X input going to module = 17716081
Clk = 0,  A =  1, S = 0, I = 1, B = zz
X input going to module = 77160817
Clk = 1,  A =  1, S = 0, I = 1, B = zz
X input going to module = 71608177
Clk = 0,  A =  1, S = 1, I = 0, B = zz
X input going to module = 16081771
X received by module = 16081771
Clk = 1,  A =  1, S = 1, I = 0, B = 08
X input going to module = 60817716
Clk = 0,  A =  1, S = 1, I = 0, B = 08
```

```
X input going to module = 08177160
X received by module = 08177160
Clk = 1,  A =  1, S = 1, I = 0, B = 08
X input going to module = 81771608
Clk = 0,  A =  1, S = 1, I = 0, B = 08
X input going to module = 17716081
Clk = 1,  A =  1, S = 1, I = 1, B = 00
X input going to module = 77160817
Clk = 0,  A =  1, S = 1, I = 1, B = 00
X input going to module = 71608177
Clk = 1,  A =  1, S = 1, I = 1, B = 00
X input going to module = 16081771
Clk = 0,  A =  1, S = 1, I = 1, B = 00
X input going to module = 60817716
Clk = 1,  A =  1, S = 1, I = 1, B = 00
X input going to module = 08177160
Clk = 0,  A =  2, S = 0, I = 0, B = zz
X input going to module = 81771608
Clk = 1,  A =  2, S = 0, I = 0, B = zz
X input going to module = 17716081
Clk = 0,  A =  2, S = 0, I = 0, B = zz
X input going to module = 77160817
Clk = 1,  A =  2, S = 0, I = 0, B = zz
X input going to module = 71608177
Clk = 0,  A =  2, S = 0, I = 0, B = zz
X input going to module = 16081771
Clk = 1,  A =  2, S = 0, I = 1, B = zz
X input going to module = 60817716
Clk = 0,  A =  2, S = 0, I = 1, B = zz
X input going to module = 08177160
Clk = 1,  A =  2, S = 0, I = 1, B = zz
X input going to module = 81771608
Clk = 0,  A =  2, S = 0, I = 1, B = zz
X input going to module = 17716081
Clk = 1,  A =  2, S = 0, I = 1, B = zz
X input going to module = 77160817
Clk = 0,  A =  2, S = 1, I = 0, B = zz
X input going to module = 71608177
X received by module = 71608177
Clk = 1,  A =  2, S = 1, I = 0, B = 5c
X input going to module = 16081771
Clk = 0,  A =  2, S = 1, I = 0, B = 5c
```

…..

Paste wave forms from GTKWave here



Q5. Prepare a short video presentation where you explain your solution and demonstrate the code execution. Use Zoom with screen sharing to record your video. Keep the recording under 5 minutes. Paste a link to the video (Microsoft Stream Link preferred, OneDrive/Google Drive/YouTube is also acceptable. The video should be "unlisted" or not publicly searchable in all cases, with link sharing open to public. The video should be available for view until the result is published for this semester).

Write the link to your video below:

Ans:

https://youtu.be/wH6NPUshUzg

(Time step used on testbench should be 8, as T = 8 in my case. In the video, time step was kept 5. The code has been updated later and #8 was used in place of #5. Nonetheless the output remains same.)