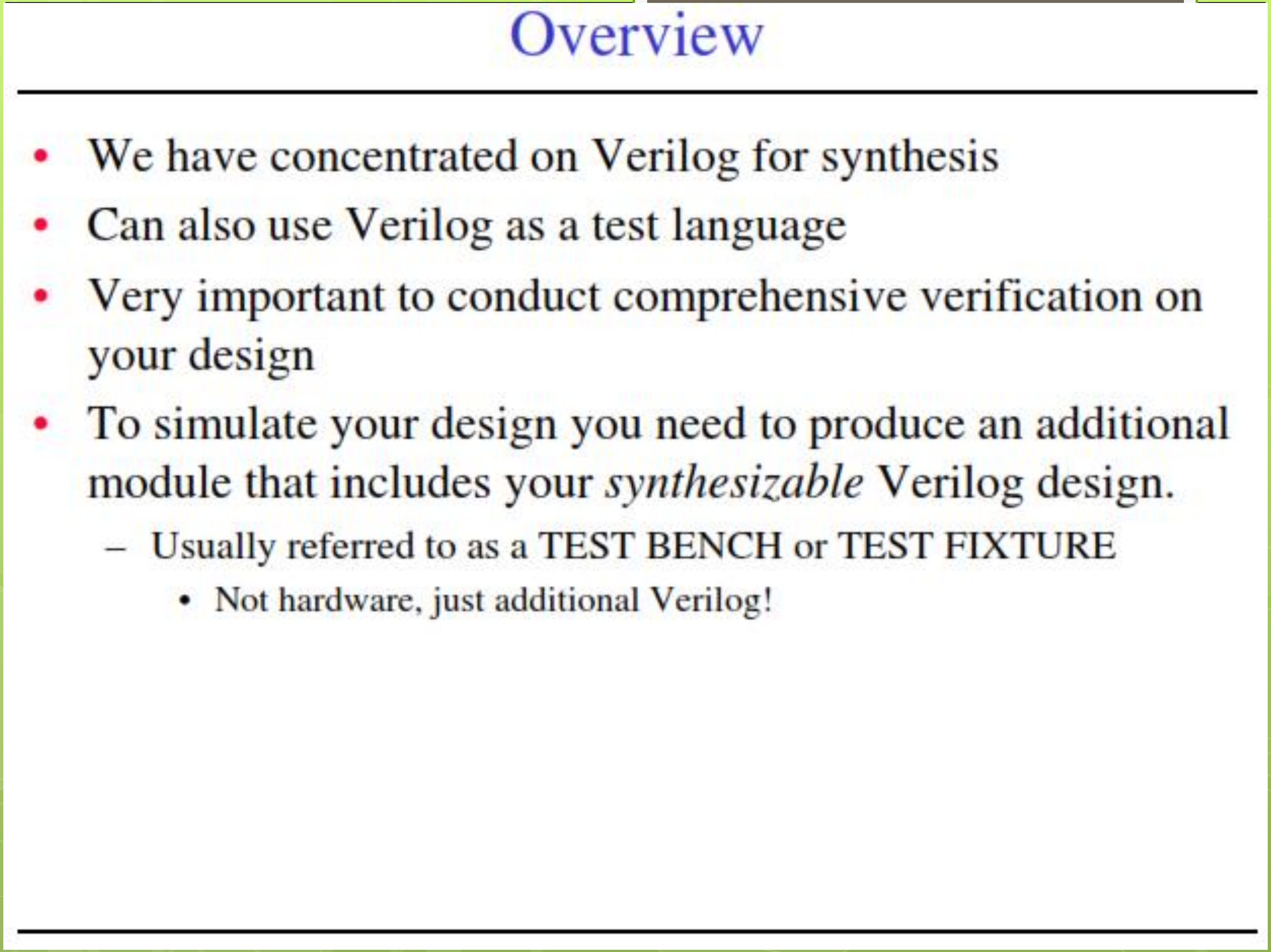
Introduction to HDL - TestBench

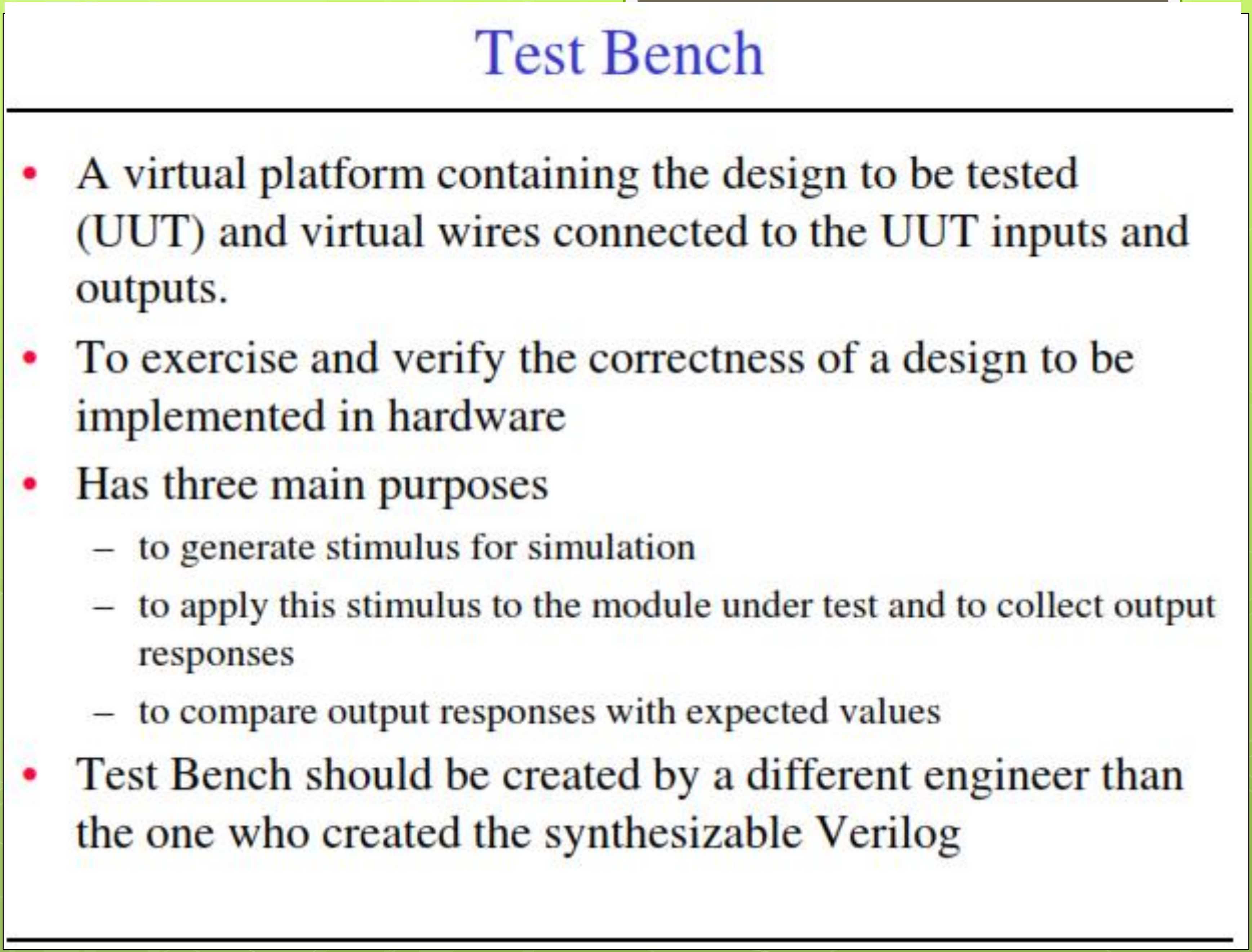
Credit to: Dr. MD Rizal Othman
Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

Ext: 6036



Overview

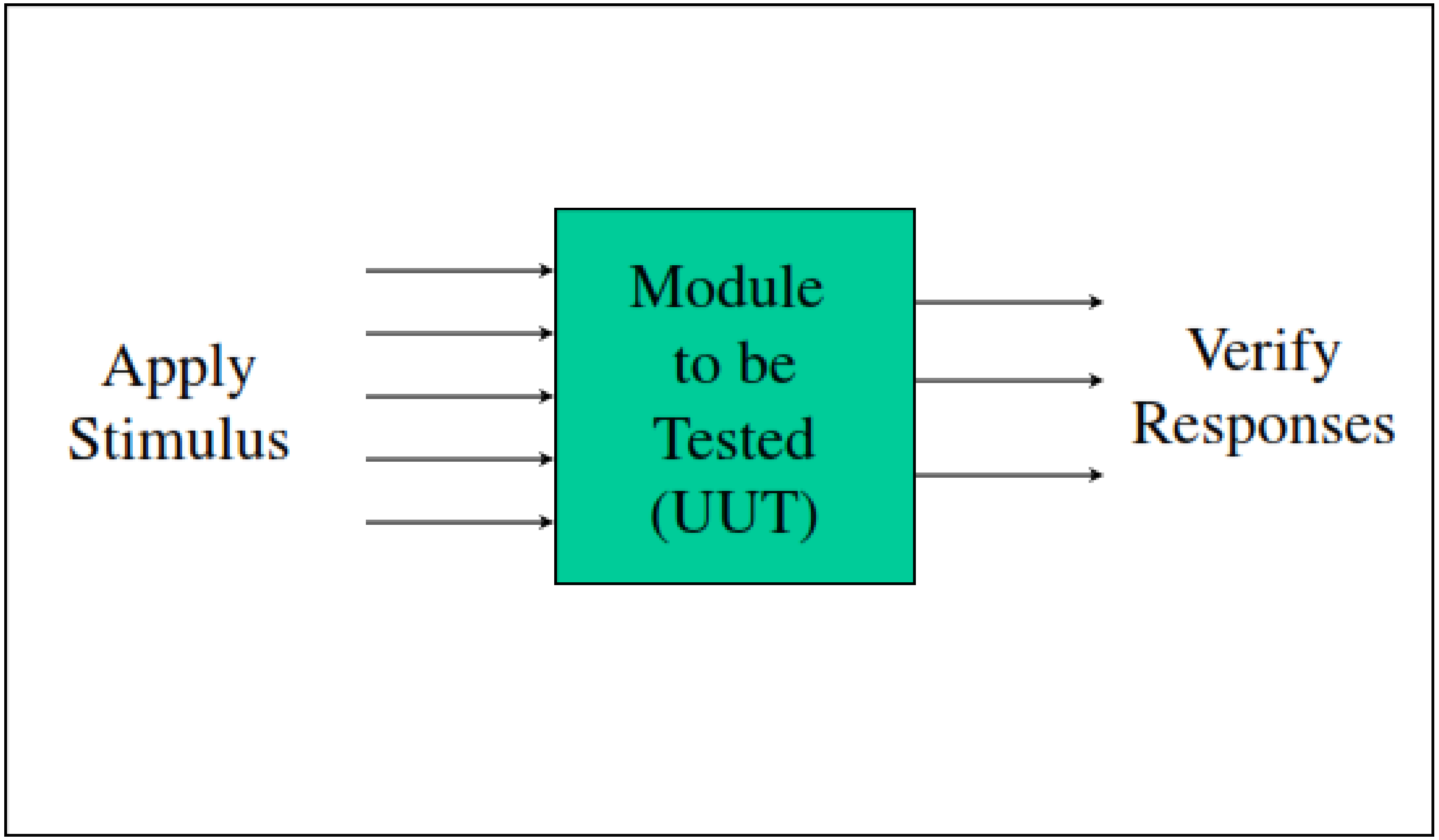
- We have concentrated on Verilog for synthesis
- Can also use Verilog as a test language
- Very important to conduct comprehensive verification on your design
- To simulate your design you need to produce an additional module that includes your *synthesizable* Verilog design.
  - Usually referred to as a TEST BENCH or TEST FIXTURE
    - Not hardware, just additional Verilog!



Test Bench

- A virtual platform containing the design to be tested (UUT) and virtual wires connected to the UUT inputs and outputs.
- To exercise and verify the correctness of a design to be implemented in hardware
- Has three main purposes
  - to generate stimulus for simulation
  - to apply this stimulus to the module under test and to collect output responses
  - to compare output responses with expected values
- Test Bench should be created by a different engineer than the one who created the synthesizable Verilog

Test Bench – Virtual Platform

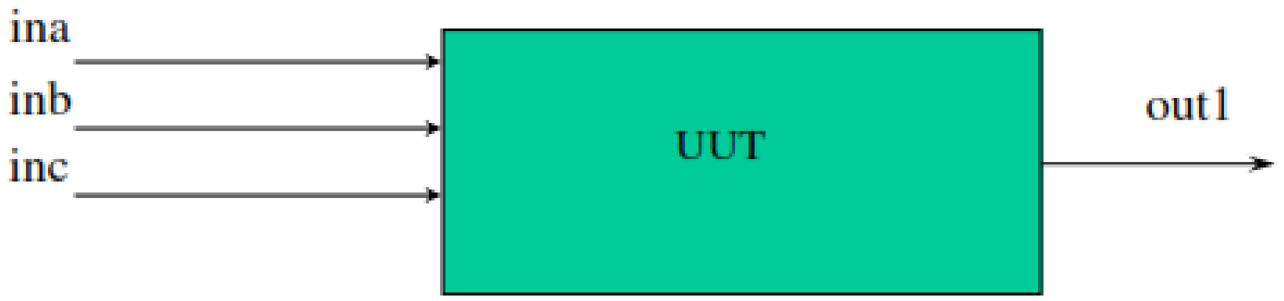


Test Bench Overview

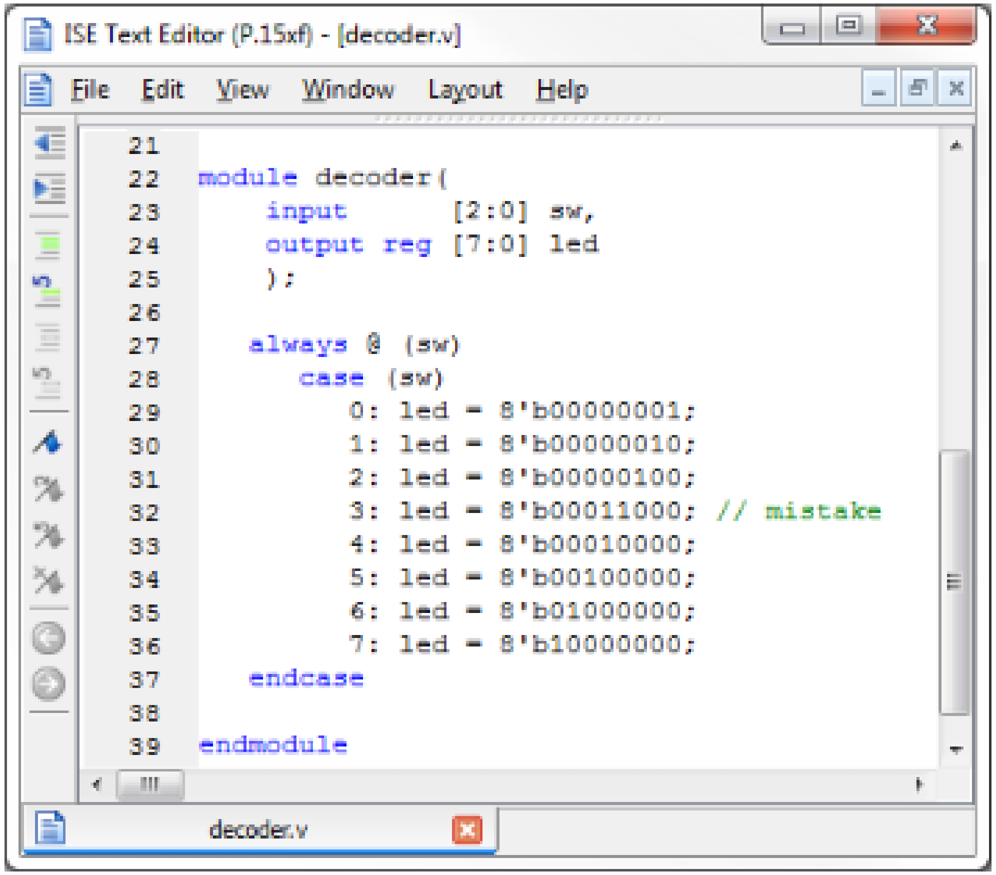
- A Test Bench module consists of
  - Port list has NO ports
  - Instantiate module to be tested (UUT)
  - Declare internal signals to wire to UUT inputs and outputs
  - Verilog statements to provide stimulus and verify UUT responses
  - Designing a test bench that has good coverage can be a very involved project!

Test Bench - general structure

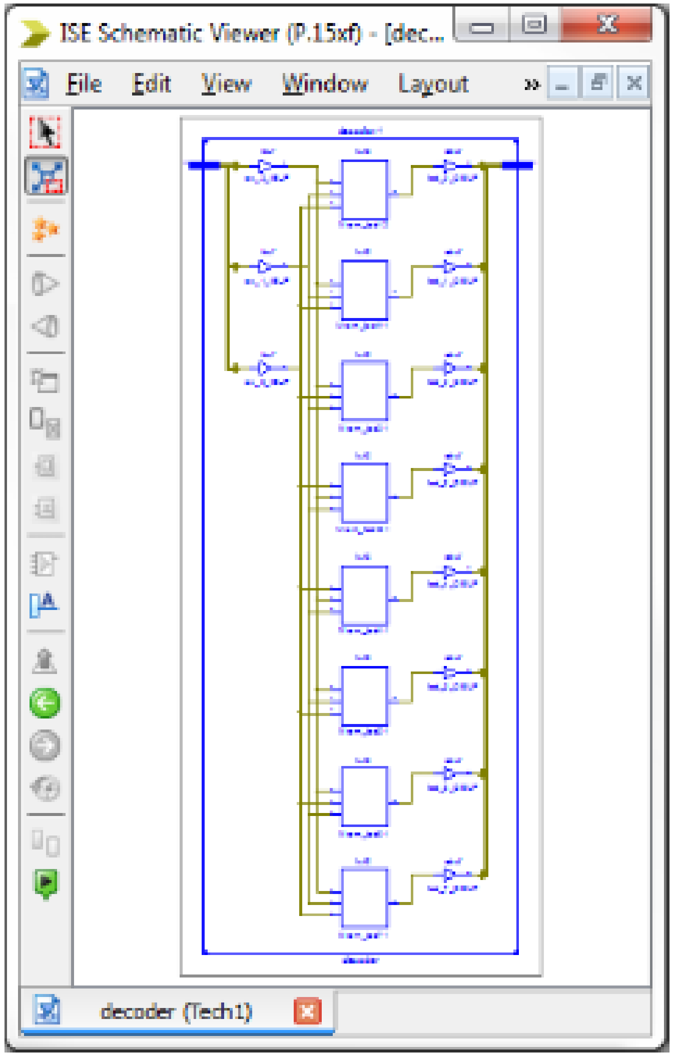
```
module decoder_tf;

        internal signal declarations

        UUT: test_component instantiation

        signals to generate stimulus

        statements to verify responses

endmodule
```



Example Decoder – is the design correct?

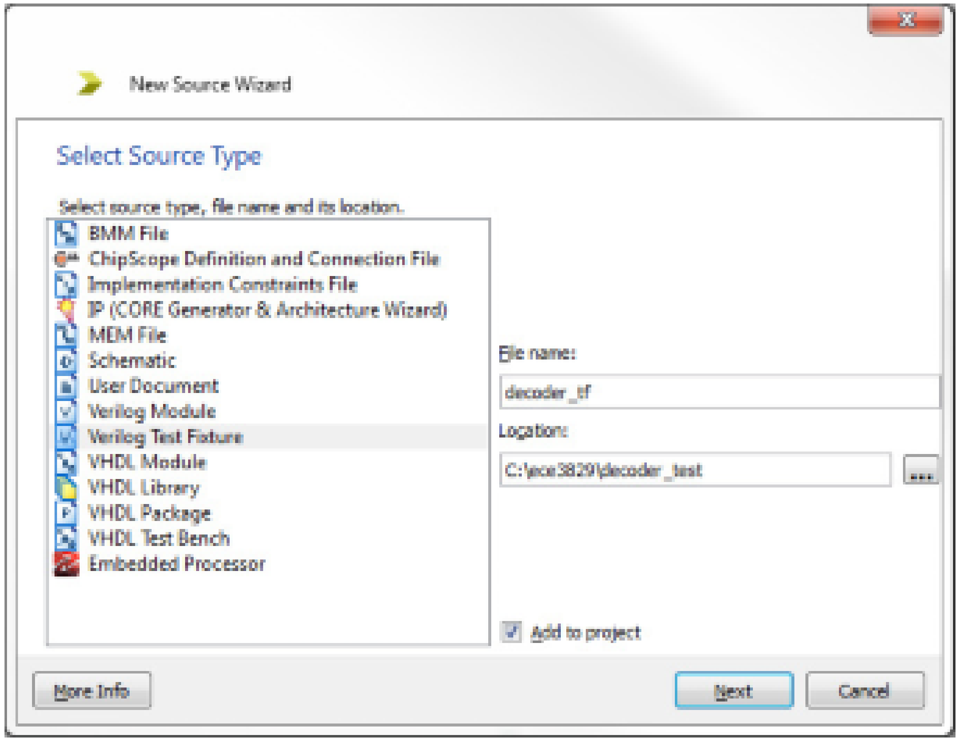

ISE Text Editor (P.15xf) - [decoder.v]
File Edit View Window Layout Help
decoder.v

```
module decoder(
    input       [2:0] sw,
    output reg  [7:0] led
    );

  always @ (sw)
     case (sw)
        0: led = 8'b00000001;
        1: led = 8'b00000010;
        2: led = 8'b00000100;
        3: led = 8'b00011000; // mistake
        4: led = 8'b00010000;
        5: led = 8'b00100000;
        6: led = 8'b01000000;
        7: led = 8'b10000000;
  endcase

endmodule
```


ISE Schematic Viewer (P.15xf) - [dec...
File Edit View Window Layout
decoder (Tech1)

Create Test Fixture

- Select Simulation View
- Select Project => New Source



Select source type, file name and its location.

- BMM File
- ChipScope Definition and Connection File
- Implementation Constraints File
- IP (CORE Generator & Architecture Wizard)
- MEM File
- Schematic
- User Document
- Verilog Module
- Verilog Test Fixture
- VHDL Module
- VHDL Library
- VHDL Package
- VHDL Test Bench
- Embedded Processor

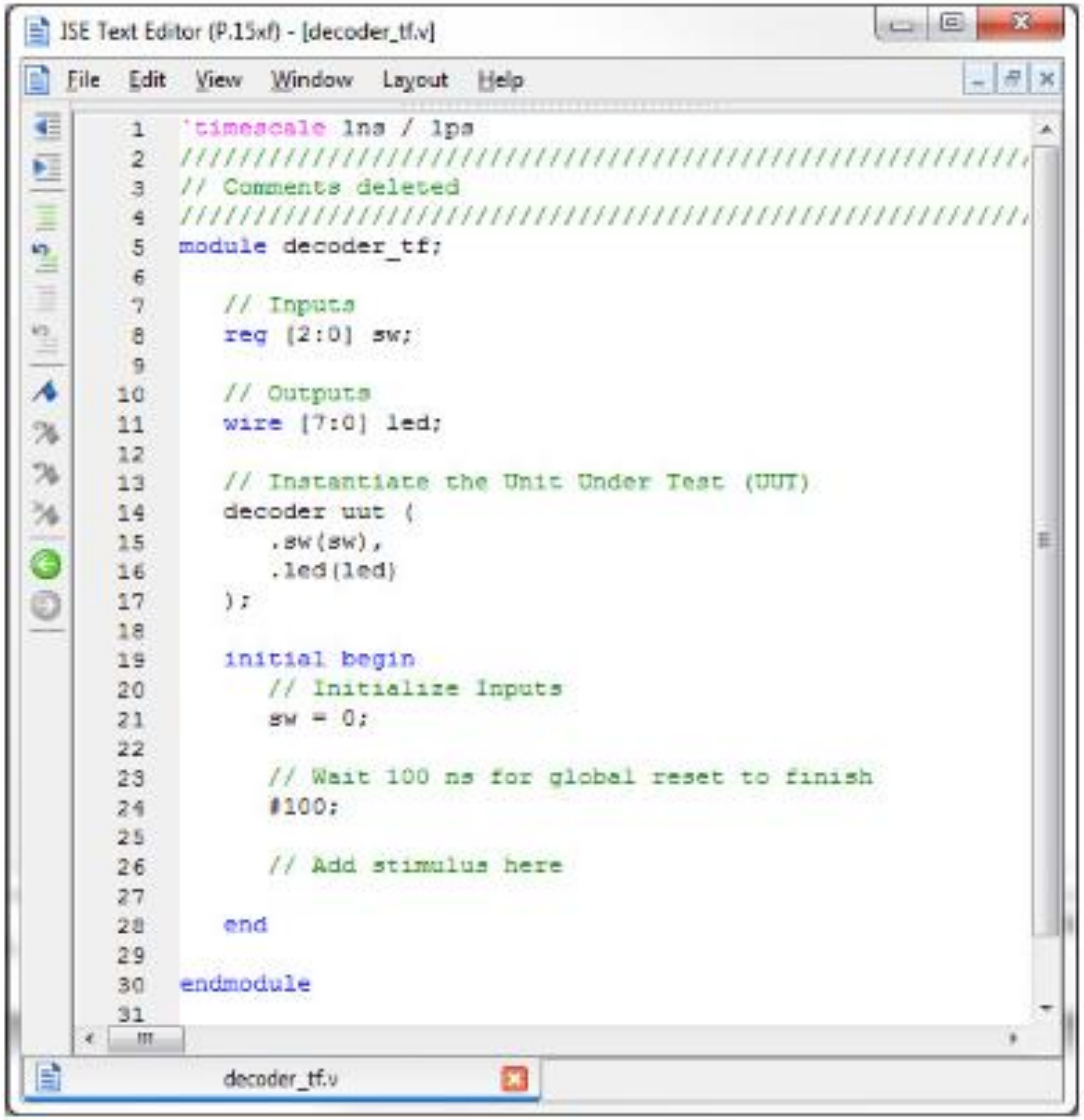
File name: decoder_tf

Location: C:\ece3829\decoder_test

[✓] Add to project

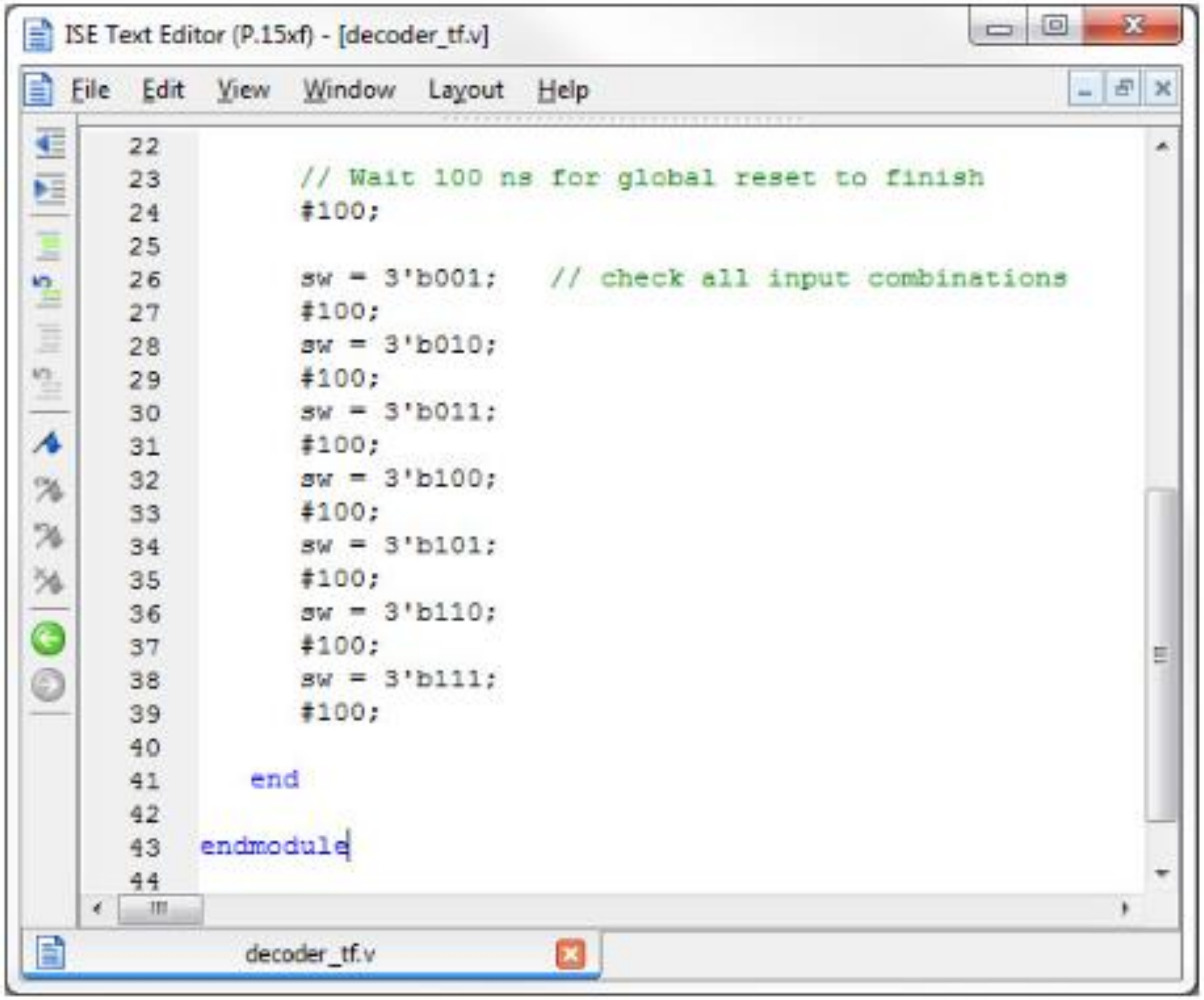
- Select decoder source to associate with test fixture

Outline of Test Fixture produced



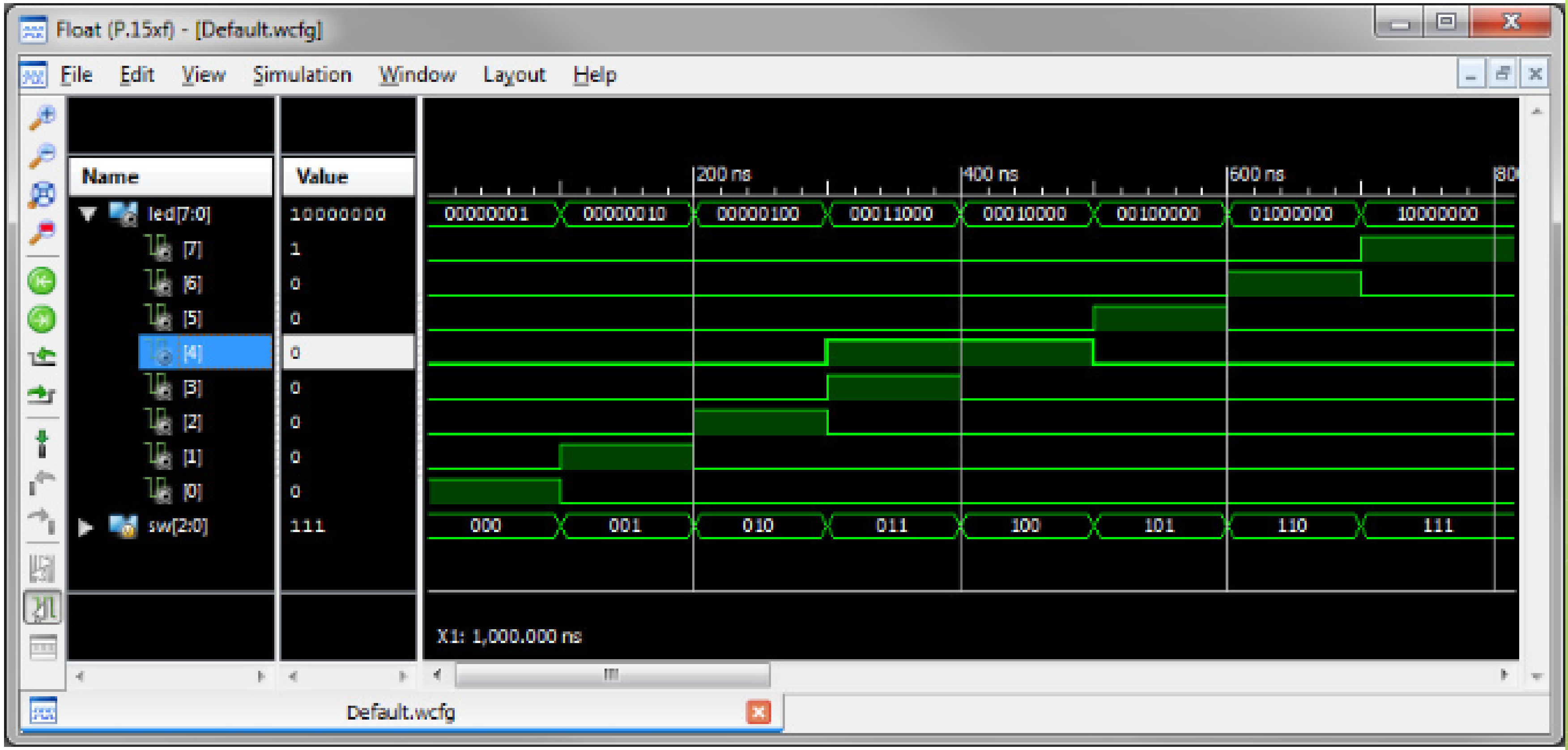
```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////
// Comments deleted
////////////////////////////////////////////////////////////////////
module decoder_tf;

   // Inputs
   reg [2:0] sw;

   // Outputs
   wire [7:0] led;

   // Instantiate the Unit Under Test (UUT)
   decoder uut (
      .sw(sw),
      .led(led)
   );

   initial begin
      // Initialize Inputs
      sw = 0;

      // Wait 100 ns for global reset to finish
      #100;

      // Add stimulus here

   end

endmodule

```

Apply input stimulus


ISE Text Editor (P.15xf) - [decoder_tf.v]
File Edit View Window Layout Help
decoder_tf.v

```

        // Wait 100 ns for global reset to finish
        #100;

        sw = 3'b001;    // check all input combinations
        #100;
        sw = 3'b010;
        #100;
        sw = 3'b011;
        #100;
        sw = 3'b100;
        #100;
        sw = 3'b101;
        #100;
        sw = 3'b110;
        #100;
        sw = 3'b111;
        #100;

    end

endmodule

```

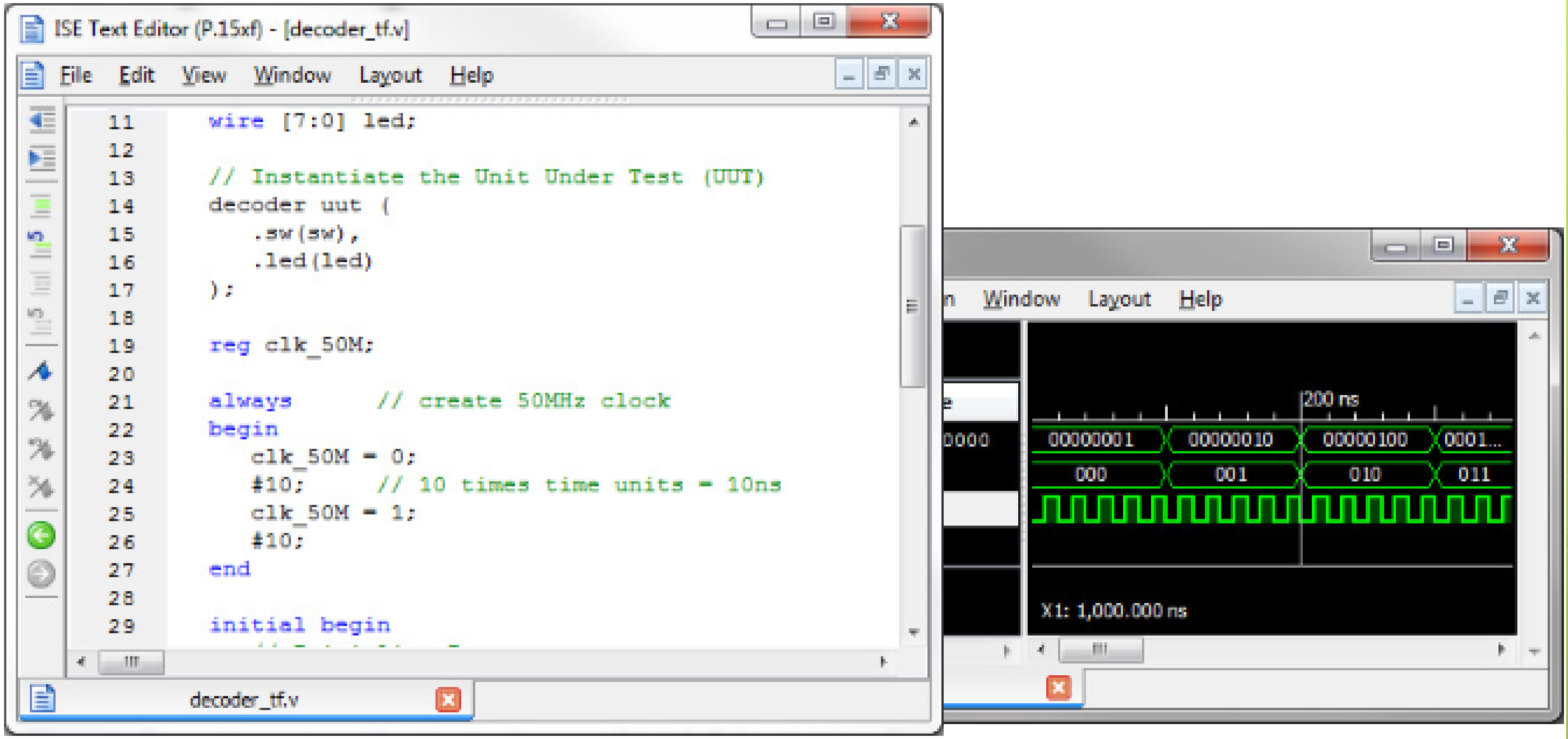
Simulate Behavioral Model


Float (P.15xf) - [Default.wcfg]
File Edit View Simulation Window Layout Help
Name
Value
200 ns
400 ns
600 ns
Default.wcfg

```
led[7:0]   10000000   00000001 | 00000010 | 00000100 | 00011000 | 00010000 | 00100000 | 01000000 | 10000000
  [7]      1
  [6]      0
  [5]      0
  [4]      0
  [3]      0
  [2]      0
  [1]      0
  [0]      0
sw[2:0]    111        000 | 001 | 010 | 011 | 100 | 101 | 110 | 111

X1: 1,000.000 ns
```

Testing sequential logic

- Creating clock signal


ISE Text Editor (P.15xf) - [decoder_tf.v]
File Edit View Window Layout Help
decoder_tf.v
Window Layout Help
200 ns
X1: 1,000.000 ns

```
wire [7:0] led;

// Instantiate the Unit Under Test (UUT)
decoder uut (
    .sw(sw),
    .led(led)
);

reg clk_50M;

always      // create 50MHz clock
begin
    clk_50M = 0;
    #10;      // 10 times time units = 10ns
    clk_50M = 1;
    #10;
end

initial begin
```

```
0000  00000001  00000010  00000100  0001...
      000       001       010       011
```

- Can also ‘restart’, ‘run’, and add internal signals to wave

Discussion 1

Testbench 4 bit ALU