CSE 453

Verilog HDL
Writing Modular Code

Managing a Large Project

- Tasks
- Functions
- Modules

Tasks

- Can enable other tasks & functions
- May execute in non-zero simulation time
- May contain delay, event, or timing control statements
- May have zero or more arguments
  - Type
    - Input
    - Output
    - Inout
- Can pass values, but do NOT return a value
- Syntax
  - Definition

```
task Task_Name;
  Input, Output, & Local Variable List
  begin
     Body
  end
endtask
```

  - Call

```
Task_Name(Argument_List);
```

Functions

- Can enable another function
  - Not a task
- Execute in zero simulation time
- Can NOT contain
  - A delay event
  - Timing control statements
- Must have at least one input argument
- Always return a single value
  - Output & InOut arguments NOT allowed

- Syntax
  - Definition

```
function Function_Name; // Note name may represent a vector
  Input & Local Variable List
  begin
     Body - Note the output must be assigned
            a value
  end
endfunction
```

  - Call

```
Function_Name(Argument_List);
```

Tasks & Functions

- Local to the module
- They can contain
  - Local variables
  - Registers
  - Time Variables
  - Integers
  - Reals
  - Events
- Cannot contain
  - Wires
  - Always Blocks
  - Initial Blocks
- Behavioral only!

Modules

- Instantiating modules can help make code easier to write, modify, read, and debug

Strobe Example

- Description
  - The number ‘12’ is displayed on the middle two digits of the 7-segment display of the BASYS2 FPGA Development Board by Digilent
  - LED blinks at the frequency used to strobe the 7-segment display
  - Reset
    - Switch 0
    - When low, the 7-segment display is disabled
    - When high, ‘12’ is displayed on the 7-segment display
- Pin Assignments
  - Reset (P11)
  - Clock (B8)
  - LD0 (M5)
  - Anodes 0…3 (F12, J12, K14, M13)
  - Cathodes A…G (L14, H12, N14, N11, P12, L13, M12)

```
module strobe_eg(led, clk, cathodes, anodes, rst);

   input clk, rst;
   output [6:0] cathodes;
   output led;
   output [3:0] anodes;
   reg [6:0] cathodes;
   reg [3:0] anodes;
   reg [3:0] dig;
   reg led;
   reg slow_clock;
   integer count;

   always @(posedge clk)
      create_slow_clock(clk, slow_clock);

always @(posedge slow_clock)
  begin
    led=~led;
    if (rst == 0) anodes=4'b 1111;
    else
      begin
       case (anodes)
          4'b 1101: anodes=4'b 1011;
          4'b 1011: anodes=4'b 1101;
          4'b 1111: anodes=4'b 1011;
          default: anodes=1111;
       endcase
       case (anodes)
          4'b 1011: dig=1;
          4'b 1101: dig=2;
       endcase
       cathodes=calc_cathode_value(dig);
      end
  end

   function [6:0] calc_cathode_value;
     input [3:0] dig;
     begin
        case (dig)
           1: calc_cathode_value = 7'b 1111001;
           2: calc_cathode_value = 7'b 0100100;
        endcase
     end
   endfunction
```

```
  task create_slow_clock;
    input clock;
    inout slow_clock;
    integer count;

      begin
        if (count > 250000)
        begin
           count=0;
           slow_clock = ~slow_clock;
        end
        count = count+1;
     end

   endtask
endmodule
```

References

- Donald E. Thomas and Philip R. Moorby, *The Verilog Hardware Description Language*, Kluwer Academic Publishers, 1998
- Samir Palnitkar, *Verilog HDL A Guide to Digital Design and Synthesis*, Prentice Hall, Inc., 4th Edition, 1996
- David R. Smith and Paul D. Franzon, *Verilog Styles of Digital Systems*, Prentice Hall, Inc., 2000
- Michael D. Ciletti, *Advanced Digital Design with the Verilog HDL*, Pearson Education, Inc. (Prentice Hall), 2003
- *Digilent Basys 2 Board Reference Manual*, Digilent, Inc., May 25, 2009
- *Digilent BASYS 2 System Board Schematics*, Digilent, Inc., December 12, 2008