שפת תכנון חומרה
ורילוג - Verilog

Verilog

Dr. Avihai Aharon

Verilog Code Template

- **timescale directive tells the simulator the base units and precision of the simulation**
  `timescale 1 ns / 10 ps
  module <name> (<input and outputs>);
- **parameter declarations**
  parameter <parameter_name> = <parameter value>;
- **Input output declarations**
  input <in1>;
  input <in2>; // single bit inputs
  input [<msb>:<lsb>] <in3>; // a bus input

Verilog Code Template (Cont.)

- **Internal signal register type declaration - register types (only *assigned within always statements).***
  reg <register variable 1>;
  reg [<msb>:<lsb>] <register variable 2>;
- **Internal signal. net type declaration - (only assigned outside *always statements)***
  wire <net variable 1>;
- **Hierarchy - instantiating another module**
  <reference name> <instance name> (
  .pin1 (net1),
  .pin2 (net2),
  .
  .
  .
  .pinn (netn)
  );

Verilog Code Template (Cont.)

- **Synchronous procedures**
  always @ (posedge <clock>)
  begin
  .
  .
  .
  end
- **Combinatinal procedures**
  always @ (signal1 or signal2 or signal3)
  begin
  .
  .
  .
  end
- *assign <net variable> = <combinational logic>;*
- *endmodule*

Verilog Code Template example

- **Example:**

```
module mux_2x1(a, b, sel, out);
    input a, a, sel;
    output out;
    always @(a or b or sel)
    begin
      if (sel == 1)
          out = a;
     else out = b;
    end
endmodule
```

Module

General definition

```
module module_name ( port_list );
    port declarations;
    …
    variable declaration;
    …
    description of behavior
endmodule
```

Example

```
module HalfAdder (A, B, Sum, Carry);
input A, B;
output Sum, Carry;
assign Sum = A ^ B;
//^ denotes XOR
assign Carry = A & B;
// & denotes AND
endmodule
```