Verilog 2001

Dr. Paul D. Franzon

Outline
1. Brief overview of new features

References
- Ciletti, Appendix I
Verilog “upgrades”

Verilog is an IEEE standard

- Verilog 2001 features mainly aimed at improving designer productivity
- To use on-campus:
  - See tutorial 1. Also:
  - add cadence
  - ncvverilog +access+r file1.v file2.v +gui

System Verilog

- Sometimes referred to as “Verilog2005”
- Adds numerous features to improve designer productivity
- Includes features for non-synthesizable system design and for verification
  - Embedded verification features again aimed at improving productivity further
New Features for Synthesizable Verilog

Sensitivity Lists
- **Commas:** `always@(a,b,c)`
- **For Combinational Logic:** `always@(*)`

Data Types
- Default is now variable width net type (`reg`)
- Can declare no default → forces all variables to be declared
  ```
  'default net_type none
  ```
- Net types can be signed and initialized

New Operators
- Signed shift `<<<` `>>>`; Exponent `**`

Port List in input / output declarations
```verilog
module test (    
    input [3:0] in1;  
    output wire [7:0] out1);
```
... New Features

Generate

- Generates multiple instances of a module using a for loop

```verbatim
genvar i;
genenerate
  for (i=0; i<4; i=i+1) begin: MEM
    memory U (read, write, data[i*8:i*8+7]);
  end
endgenerate
```

Multi-dimensional arrays

```verbatim
reg [7:0] Mem[0:255][0:255];
Readout = Mem[addr1][addr2];
```

Re-entrant (recursive) functions and tasks

- Unlikely to be used for synthesizable code

$\text{random is standardized}$
Parameter Passing

- Can pass parameters to modules, overwriting their local values

```verilog
module top;
Parameter RFsize1 = 64;
Parameter AddressSize1 = 6;

RegFile #( .size(RFsize1), .Asize(AddressSize1) U1 ( ... );
endmodule

module RegFile( ...
    input [Asize-1:0] WriteAddress, ReadAddress,
);
parameter Asize=5;
parameter size=32;
reg [15:0] Register [0:size-1];
```