Overview of the extensions

• Support for constraints (Assumptions)
• Enhancing the expressiveness of the language
• Broaden the scope of assertion usage within SystemVerilog
Support for Constraints

- Assume directive
  - `assume`
  - Immediate assume
  - Same level as `assert` and `cover`
- Intended for both formal and simulation
- `dist` operator is now allowed inside properties
  - Support random distributions
Example: Assume

```verilog
property pr1;
  @(posedge clk) reset|-> !req;  //when in reset, req must be low
endproperty

property pr2;
  @(posedge clk) ack|=> !req;    //one cycle after ack, deassert req
endproperty

property pa1;
  @(posedge clk) reset|| !req |-> !ack;  //ack must be low on reset or no req
endproperty

property pa2;
  @(posedge clk) ack |=> !ack;      //ack asserted for one clock only
endproperty

assume property (pr1);
assume property (pr2);

assert property (pa1)
  else $display("\n  ack asserted while req is still deasserted");
assert property (pa2)
  else $display("\n  ack is extended over more than one cycle");
```

Prove pa1 and pa2 while assuming pr1 and pr2
Enhancing Expressiveness

- Extend local variable usage
  - Accessing local variables via ended and matched functions
  - Attaching local variable assignments to a sequence expression
- Accessing sampled values using \( \textit{\$sampled} \)
- Dynamic and recursive calls to a property
- Expand power of property expressions
  - Allow Boolean conjunction/disjunction of properties as a property expression
  - Allow if-else combination of properties as a property expression
  - Allow a property instance as a property expression
  - Allow Boolean negation of a property as a property expression
Some Examples

Two properties share the same antecedent:

\[(s1 \Rightarrow (s2 \underline{##} s3)) \text{ and } (s1 \Rightarrow (s4 \Rightarrow s5))\]
\[(s1 \Rightarrow (s2 \underline{##} s3)) \text{ or } (s1 \Rightarrow (s4 \Rightarrow s5))\]

Can now be written as:

\[(s1 \Rightarrow ((s2 \underline{##} s3) \text{ and } (s4 \Rightarrow s5)))\]
\[(s1 \Rightarrow ((s2 \underline{##} s3) \text{ or } (s4 \Rightarrow s5)))\]

//nested implication

property p1;
  s1 \Rightarrow (s2 \Rightarrow s3);
endproperty

If "s2 \Rightarrow s3" is shared with other properties:

property p0;
  s2 \Rightarrow s3;
endproperty

property p1;
  s1 \Rightarrow p0;
endproperty

Properties can now be used in RHS of implications

Implications nested inside implications
Broadening the Scope

- Allow assertions to be used within functions in procedural scope
- Support parameters for assertions, extend to allow sequence as parameter
- Support for non-blocking
- Extend $past function to support gated clocks
Other Extensions Beyond 3.1a

- Templates
- Assertions on transaction level
- Complete integration with Classes
- Additional temporal operators