Chapter4 - Expressions

Introduction to Analog Operators in SystemVerilog AMS

- Expressions- Construct which combines operands with operator to produce result which is a function of values of operands and semantic meaning of operator
- System Verilog AMS extends System Verilog for analog circuit modeling.
- It enables behavioral descriptions for continuous-time signals.
- Analog operators are used to model electrical components and signal processing.

System-Verilog Operators

- Arithmetic -+ - * / % **
- Relational- == != < > <= >=
- Logical- && || !
- Bitwise- & | ^ ~
- Shift- << >> <<< >>>
- Concatenation & Replication- {} {<<} {>>}
- Reduction- & | ^ ~& ~| ~^
- Ternary- ?:
- Assignment- = += -= &= |= ^= <<= >>=

Operator Precedence in System Verilog

Precedence	**Operators**
Highest	+,-,!,~,&, ~&,|,~|,^,~^ (unary)
	** (power)
	*, /, % (multiplication, division, modulus)
	+ ,- (addition, subtraction)
	<< ,>> ,<<< ,>>> (shifts)
	<, <= ,> ,>= (relational)
	==, !=, ===, !== (equality)
	`&
	^,^~, ~^
	|
	`&&
	||
	? :
Lowest	{}, {{}} (concatenation, replication)

1) Contribution Operator (<+)

- The '<+'operator assigns analog equations to electrical nodes.
- Used to define voltage, current, and circuit behaviour.
- V(out) <+ 5.0; // Assigns 5V to node 'out'
- I(in) <+ V(in) / 1000; // Models a 1kΩ resistor.

2) Derivative (ddt) and Integral (idt) Operators

- ddt(expression)computes the time derivative.
- idt(expression, IC, time_constant) computes the integral over time.

  E.G.
- V(out) <+ ddt(V(in)); // Computes rate of change of input voltage
- V(cap) <+ idt(I(cap) / C, 0, 1); // Capacitor voltage equation

Laplace Transform and Noise Functions

- laplace_zd(expression, num_coeff, den_coeff) models frequency response.
- Noise functions ($rdist_uniform, $rdist_normal) generate random noise.
- Example:
- V(out) <+ laplace_zd(V(in), {1, 0}, {1, 2, 1}); // H(s) = (s+0)/(s²+2s+1)
- V(noise) <+ $rdist_uniform(seed, -0.1, 0.1); // Uniform noise generator

- Application of Analog Operators

- Used in SPICE-level simulation and mixed-signal design.
- Models analog filters, amplifiers, and power systems.
- Enables co-simulation of digital and analog circuits in Verilog AMS.

Verilog-AMS - Chapter 7: Mixed Signal

Overview

- Enables analog, digital, and mixed-signal modeling
- Mixed-signal blocks allow analog ↔ digital exchange
- Uses connect modules for interfacing

Domains

- Continuous (analog): voltages, currents
- Discrete (digital): logic values, registers
- Variables must match their domain context

Contexts

- Analog context: analog, analog initial
- Digital context: initial, always, assign
- Variables cannot cross contexts directly

Nets, Nodes, Ports

- Nets: wires/signals
- Nodes: connection points
- Ports: input/output/inout
- Types: Analog, Digital, Mixed

Behavioral Interactions

- Digital in analog: V(net), I(net)
- Analog in digital: cross(), transition(), idt()
- Enables bridging between domains

Connect Modules

- Bridge analog and digital ports
- Handle transitions, preserve timing
- Inserted manually or automatically

Connect Module Syntax

- Example:
- connectmodule d2a (in, out);

```
input in;
output out;
ddiscrete in;
electrical out;
...
endmodule
```

Connect Specification Statements

- connect ... split → one per port
- connect ... merged → group ports
- Example:
  - connect d2a merged input ttl, output electrical;

Auto-Insertion of Connect Modules

- Inserted post-elaboration
- Based on port disciplines
- Controlled by connect_mode, supply resolution

Signal Segmentation

- Digital segmented by connect modules
- Analog nodes remain unified
- Models loading and isolation

Driver-Receiver Segregation

- Prevents domain conflicts
- Groups receivers and drivers
- Ensures accurate signal flow

Summary

- Domain bridging via connect modules
- Behavioral interactions across contexts
- Critical for accurate AMS co-simulation