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
```

Example

```
connectmodule voltage_to_logic(inout electrical vin, logic lout);
    parameter real vth = 2.5;
   analog begin
   lout = (V(vin) > vth) ? 1'b1 : 1'b0;
  end
endconnectmodule
```

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