



SRI KRISHNA INSTITUTE OF TECHNOLOGY

(Accredited by NAAC, Approved by A.I.C.T.E. New Delhi, Recognised by Govt. of Karnataka & Affiliated to V.T U., Belagavi)
#29, Chimney Hills, Hesaraghatta Main Road, Chikkabanavara Post, Bengaluru- 560090

Department of EE-VLSI

Subject Name: Verilog HDL

Subject Code: BVL304

SEM: DIV: III A

Faculty: Dr. N C Patil

Module-5 NOTES

-Chapter 9: Useful Modeling Techniques

9.1 Procedural Continuous Assignments

- Use `assign` inside procedural blocks with keywords:
 - o `assign` and `deassign`: for controlling variables within procedural blocks
 - o `assign a = b;`
`deassign a;`
 - o `force` and `release`: override net values temporarily
 - o `force a = 1;`
`release a;`

9.2 Overriding Parameters

- Default parameter values in modules can be overridden at instantiation
- `module delay #(parameter WIDTH = 8) (...);`
`delay #(16) D1 (...);` // Overrides WIDTH to 16

9.3 Conditional Compilation and Execution

- Controlled using compiler directives

```
0 `ifdef` / `ifndef` / `else` / `endif`
```
- ``define DEBUG`
- ``ifdef DEBUG`
- `$display("Debug mode");`
- ``endif`
- Useful for enabling/disabling code blocks in simulation or synthesis

9.4 Useful System Tasks

- `$display`, `$monitor`, `$time`, `$strobe`: for displaying values
- `$finish`, `$stop`: control simulation end
- `$dumpfile`, `$dumpvars`: generate waveform files for GTKWave/VCD



Chapter 14: Logic Synthesis with Verilog

14.1 Logic Synthesis

- Translates Verilog RTL code into gate-level netlist
- Must follow synthesizable coding rules

14.2 Impact of Logic Synthesis

- Transforms high-level constructs to actual hardware implementation
- Optimizations:
 - o Gate minimization
 - o Resource sharing
 - o Pipelining (optional by tool)

14.3 Verilog HDL Synthesis Subset

- Only a subset of Verilog is synthesizable:
 - o `always @(posedge clk) blocks`
 - o `assign` statements for combinational logic
 - o No delays (`#`), no `fork/join`, or `initial` blocks in synthesizable design

14.4 Synthesis Design Flow

1. Write RTL (Register Transfer Level) code
2. Perform simulation and verification
3. Synthesize using a synthesis tool (e.g., Synopsys Design Compiler, Xilinx Vivado)
4. Generate gate-level netlist
5. Proceed to place & route

14.5 Verification of Gate-Level Netlist

- Re-run simulation using the gate-level netlist
- Timing delays included (SDF back-annotation)
- Compares output from RTL and gate-level simulation for equivalence

Faculty Signature