21 HDL Coding Laws for Hardware Designer's

1. *Reset all flip-flops*
2. *Use active-low asynchronous resets*
3. *Signals that cross clock domains must be double-sampled*
4. *Use as few clock domains as possible*
5. *Do NOT use clocks or reset as data or enable signals*
6. *Use named notation when instantiating modules*
7. *Code one module per file, with the file name matching the module name*
8. *Keep the same signal name throughout the design hierarchy*
9. *Suffix names with _a for asynchronous signals, _n for active low signals*
10. *All busses must be of form* [n-1:0], *with MSB on the left*
11. *Avoid latches*

C/L = Combinational Logic

VLSI Excellence

21 HDL Coding Laws for Hardware Designer's

12. *Avoid using the negative edge of the clock**
13. *Include all input signals in a C/L block sensitivity list*
14. *Ensure variables are assigned in all branches of a C/L block*
15. *In RTL, Never initialize registers in the decleration*
16. *Write FSMs in 2 to 3 seperate always blocks*
17. *Use non-blocking (<=) for registers and blocking (=) in C/L*
18. *Avoid long if-then-else statements*
19. *The same test-bench should be used for RTL and gate-level*
20. *The test-bench must never change inputs on the clock edge*
21. *The test-bench should only check outputs @(posedge) clock*

C/L = Combinational Logic

VLSIExcellence

Best Free VLSI Content

1. Verilog HDL Crash Course – Link
2. Static Timing Analysis (STA) – Theory Concepts – Link
3. Static Timing Analysis (STA) – Practice/Interview Questions – Link
4. Low Power VLSI Design – Theory Concepts – Link
5. Low Power VLSI Design (LPVLSI) – Practice/Interview Questions – Link
6. Digital ASIC Design Verilog Projects – Link

Please Like, Comment, Share & Subscribe My Channel in Order to Reach Out the Content to a Larger Audience.

Thanks !!

VLSI Excellence