Logic Models
Four valued logic: "don't know" and "high impedance"
Can add further to simulate charge holding elements etc

Delays
Pure delay: delayed by a time constant
Inertial delay: capacitive delay modelled

Simulation
Simple model: scan netlist and update everything
Delta simulation: only notify gates when on input change

Behavioural Verilog
Models combinatorial logic to construct a state machine

Structural Verilog
Hierarchic netlist form sometimes generated by a Verilog compiler

Assignment
Continuous (assign x = …)
Blocking (x = …)
Non-blocking (x <= …)

QM Minim.
Find all prime implicants:

1. Produce a truth table and extract minterms
2. Compare pairs of terms that differ by 1 bit to produce a new term where the difference becomes x
3. Tick those terms that have been covered by the new term
4. Repeat with the new set of terms until no more terms can be produced
5. Unticked terms are prime implicants

Select smallest set of prime implicants to cover function:

1. Prepare chart showing how implicants cover the minterms
2. Select essential prime implicants (i.e. those that cover unique minterms)
3. Select other prime implicants as necessary to cover

Don't cares important for minimisation so you should use "x" a lot in Verilog

Other Minim.
Wire minimisation: mainly done by place + route, but synthesis may help by duplicating logic and hinting
State minimisation: remove duplicate/unreachable states
State assignment: can be done automatically

Faults
Logical: stuck at, stuck open, bridging faults
Parametric: low/high voltage, gate and path delay
Equivalence: remove test for least significant fault
Dominance: if even test for 1 detects 2 then 1 dominates 2
Undetectable: no test exists
Redundant: undetectable and doesn't affect operation

Testability
Controllability
Observability

D-Algorithm
~D: 0 in good, 1 in bad
D: 1 in good, 0 in bad

1. Assign each checkpoint D or ~D
2. Propagate D forward
3. From a fault detecting primary output work backwards to determine inputs to make the fault visible

Scan Paths
Make all the D latches scannable (solve state problem) with multiplexers
Aids ATPG, latches have sliced circuit into combinatorial logic