STATE REDUCTION AND STATE ASSIGNMENT

If one is able to reduce the total number of states, one may be able to save on the number of flip-flops required for a design. This is the optimal situation. For example if a finite state machine drops from 7 states to 4 states and compact state assignments are used, the design drops from three flip-flops to two flip-flops. A sub optimal situation is when the number of states is reduced, but the number of flip-flops is not. This does add don't cares to the combinational logic that generates the next state equations. This will most likely drop the over all cost of the finite state machine.

Once the number of states is at a minimum, then a judicious assignment of states may further reduce the cost of the next state equations and/or the cost of the output equations. A set of heuristic rules is proposed where each rule is directed toward the reduction of the Combinational logic in the finite state machine design.

As opposed to compact state assignments, one may propose a one-hot state assignment. One-hot is a set of state assignments in which a unique bit is one in the assignment for each state. This often leads to a reduction in the logic cost for the outputs, because in one and only one state a given output is asserted.

A. STATE REDUCTION

The three main methods of state reduction include: **row matching, implication charts**, and **Successive partitioning**. Row matching, this is the easiest of the three, works well for state transition tables which have an obvious next state and output equivalences for each of the present states. This method will generally not give the most simplified state machine available, but its ease of use and consistently fair results is a good reason to pursue the method. The implication chart uses a graphical grid to help find any implications or equivalences and is a great systematic approach to reducing state machines. Successive partitioning is almost a cross between row matching and implication chart where both a graphical table and equivalent matching is used. Each of these methods will, in most cases, reduce a state machine into a smaller number of states.

Keep in mind that one method may result in a simpler state machine than another. Experience will help in understanding and determining the best method to use in any particular situation.