Model Checking - II

Virendra Singh
Associate Professor
Computer Architecture and Dependable Systems Lab.
Dept. of Electrical Engineering
Indian Institute of Technology
Bombay
http://www.ee.iitb.ac.in/~viren
viren@ee.iitb.ac.in

EE 709: Testing & Verification of VLSI Circuits
Lecture – 25 (Mar 01, 2012)
The Model Checking Problem

The Model Checking Problem (CE81):

- Let $M$ be a Kripke structure (i.e., state-transition graph).
- Let $f$ be a formula of temporal logic (i.e., the specification).
- Find all states $s$ of $M$ such that $M, s \models f$
"It is never possible to have a green light for both N-S and E-W."

Mar 01, 2012
EE-709@IITB
Finite State Machine (FSM)

- **I**: input alphabet
- **S**: finite, non-empty set of states
- \( \delta \): \( S \times I \rightarrow S \), next-state function
- \( S^0 \subseteq S \): set of initial (reset) states
- **O**: output alphabet
- \( \lambda \): \( S \times I \rightarrow O \), output function

**Mealy FSM**: \( \langle I, S, \delta, S^0, O, \lambda \rangle \)

**State Transition Table**

<table>
<thead>
<tr>
<th>( x = 0 )</th>
<th>( x = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S1</strong></td>
<td>S1,0</td>
</tr>
<tr>
<td><strong>S2</strong></td>
<td>S1,0</td>
</tr>
<tr>
<td><strong>S3</strong></td>
<td>S3,0</td>
</tr>
</tbody>
</table>

**State Transition Graph**

Mar 01, 2012 EE-709@IITB
3 Step Process

- **Formal Specification**
  - Precise statement and property
  - Environment constraint
  - Logic: Temporal logic
  - Automata, Labeled transition system

- **Models**
  - Flexible model generation to specify design
  - Fairness
  - Transition system

- **Formal Verification**
  - Checking that model satisfy the property
Semantic of Finite State System

- Semantic associated with behaviour
- Branching Time Semantics
  - The tree of states obtained by unwinding the state machine transition graph
  - Possible choices are explicitly represented
- Linear Time Semantics
  - The set of all possible runs of the system
  - The set of infinite paths in SM
Formal Specification

- Describe unambiguously and precisely the expected behaviour of the design
- In general, a list of properties
- Includes, environmental constraints
Classification of Properties

- **Safety Property**
  - \((\text{un})\) desirable things always (never) happen
    - A bus arbiter never grants the requests to two masters
    - Message received is message sent

- **Liveness (Progress) Property**
  - Desirable state eventually reached
    - Every bus request is eventually granted
    - A car at a traffic light is eventually allowed to pass

- **Fairness Property**
  - Desirable state repeatedly reached
    - A request state and a grant state for each client must be visited infinitely often
Example: traffic light controller

- Guarantee no collisions
- Guarantee eventual service
Property Specification

Properties for traffic light controller

- $P_1 = (s_1 \oplus w_1) + (s_2 \oplus w_2)$
- Sequence $R, G, Y, R, G, Y, \ldots$
Thank you