Properties in Temporal Logic

Reachability Deadlock-Freeness Abstraction
Safety Fairness Liveness

System Verification

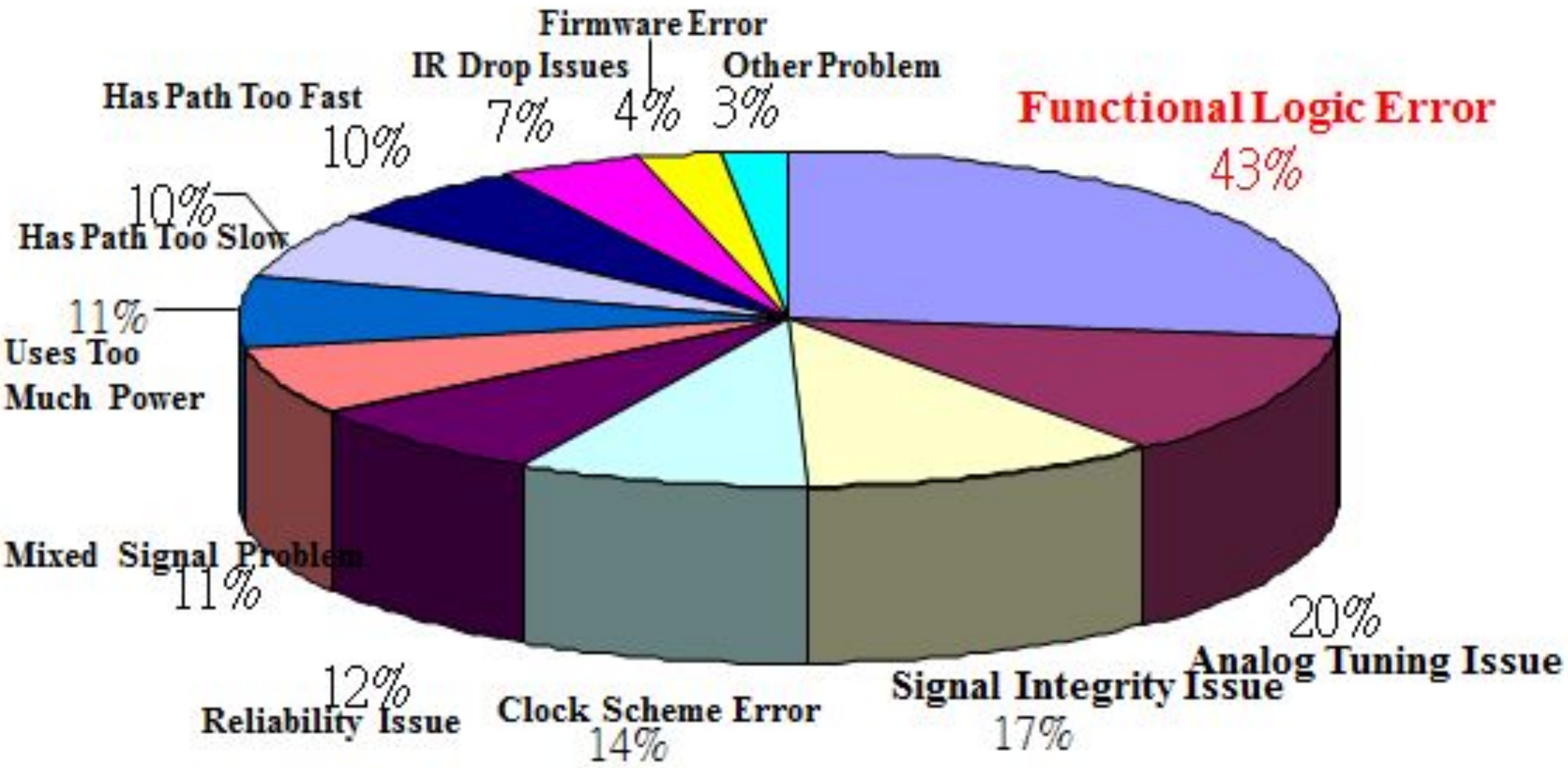
"System Verification is a set of actions used to check the correctness of any element, such as a system element, a system, a document, a service, a task, a requirement, etc."

Source: System Verification, Second Edition: Proving the Design Solution Satisfies the Requirements 2nd Edition by Jeffrey O. Grady

Importance of System Verification in Real Time Systems

PROBLEMS FOUND ON 1ST SPIN ICS/ASICS

- Overall 61% of New ICs/ASICs Require At Least One Re-Spin
- %43 due to functional error



Source: DeepChip, Aart de Geus, Chairman & CEO of Synopsys used it during Wed, 10 Sep 2003 Boston SNUG keynote address

Software Verification Techniques

Theorem Proving

- Subfield of automated reasoning and mathematical logic dealing with proving mathematical theorems by computer programs
- Limitation: Quite tedious for complex designs

SMT Solver

- Satisfiability Modulo Theories (SMT) Solver
- Provides the desired reachable path
- Limitation: Doesn’t provide the trace of counter example

Model Checking

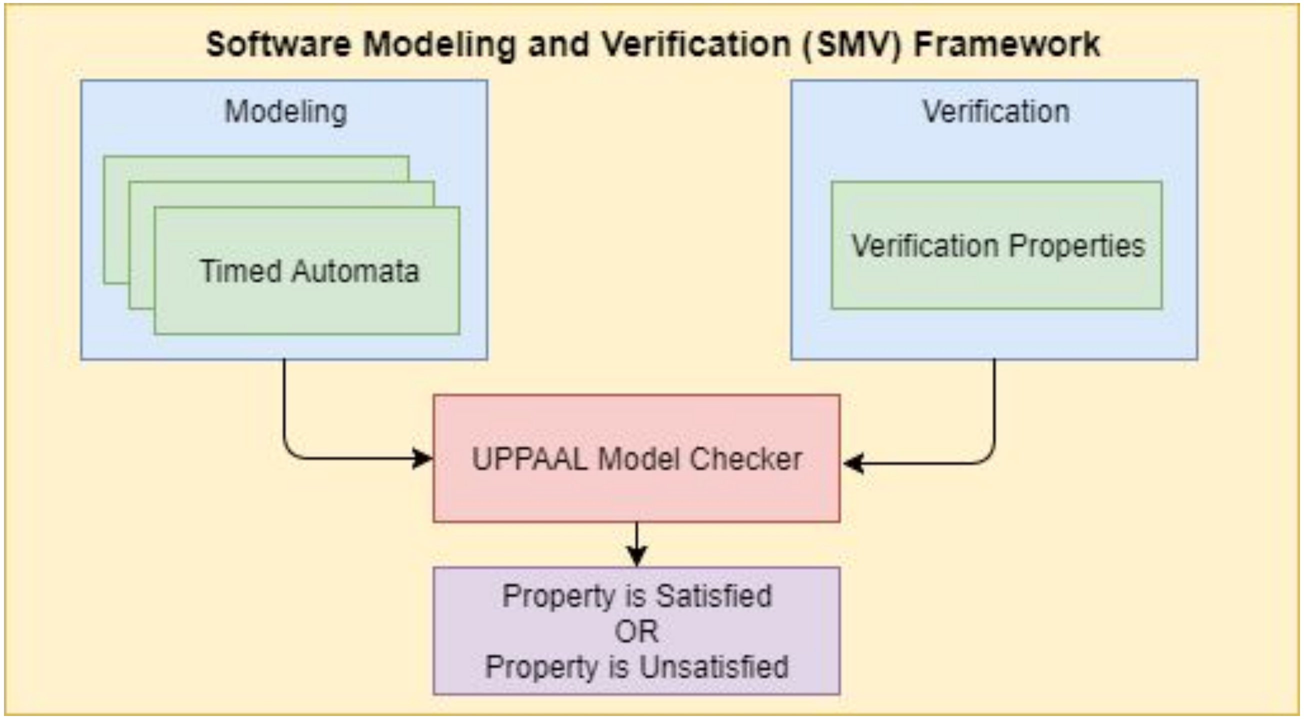
- A technique for automatically verifying correctness properties of finite-state systems
- Potentials: Provides extensive GUI and the trace of counter example

Model Checking

- Exhaustively exploring all states of the system is an extensive task.

- Model checking techniques result in more precise and accurate analysis.

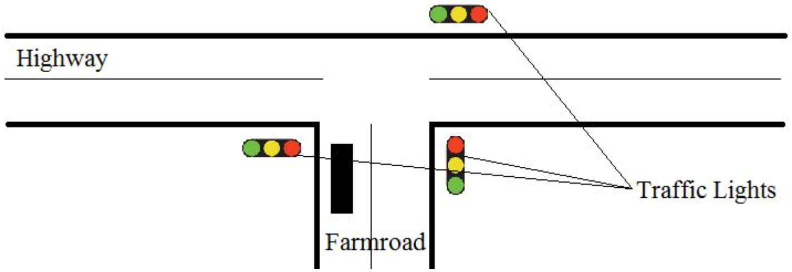
  - Lack of technical knowledge related to temporal logics and model checker usage is the main reason for not using it.


Software Modeling and Verification (SMV) Framework
Modeling
Timed Automata
Verification
Verification Properties
UPPAAL Model Checker
Property is Satisfied
OR
Property is Unsatisfied

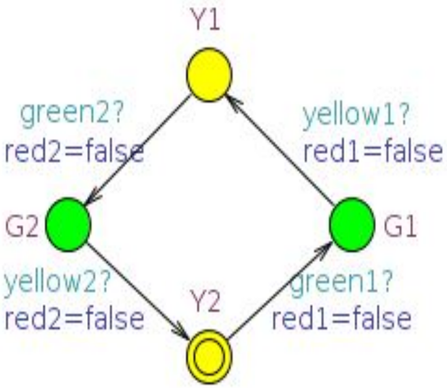
Case Studies

- **First Case Study – Highway Farm Traffic Light System [16]**

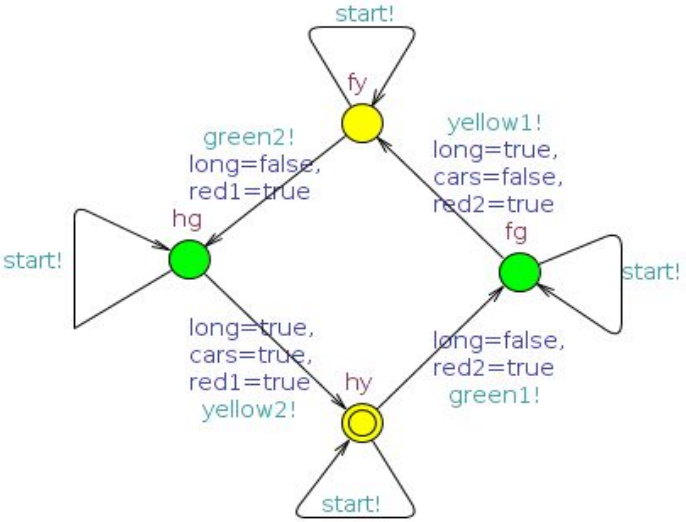
- Figure shows the Highway Farm Traffic Light (HFTL) system where there are two highway and farm road, and both facilitate 2-way traffic.
- The cross section in the center is the safety critical area and traffic light system is deployed to control the access of this shared resource.



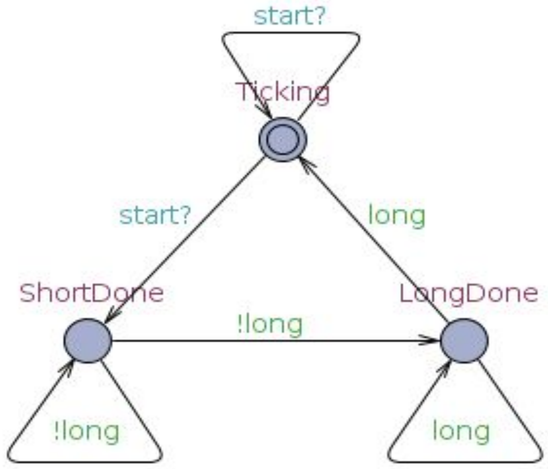
Modeling of HFTL



Light Model for HFTL System



Controller Model for HFTL System



Timer Model for HFTL System

Verification of HFTL system

Some basic real-time properties are verified in CTL as follows:

i. Safety-I Property:

The system is in safe mode when green signal of farm road and red signal of highway is on.

A[] (Light1.G1 == true imply red2==true)

ii. Safety-II Property:

The system is in safe state when highway traffic is moving and farm road traffic is waiting.

A[] (Light1.G1 == true imply Light1.G2==false)

iii. Reachability Property:

The highway light turns green infinitely often. E<>(Light1.G2)

Verification of HFTL system (Cont'd)

iv. Utility Property:

If car appears on the farm road, it eventually activates a green light.

A<>(cars == true imply Light1.G1)

v. Deadlock Freeness Property:

Deadlock freeness ensures that system is always in a progressive state.

A[] not deadlock

vi. Fairness Property:

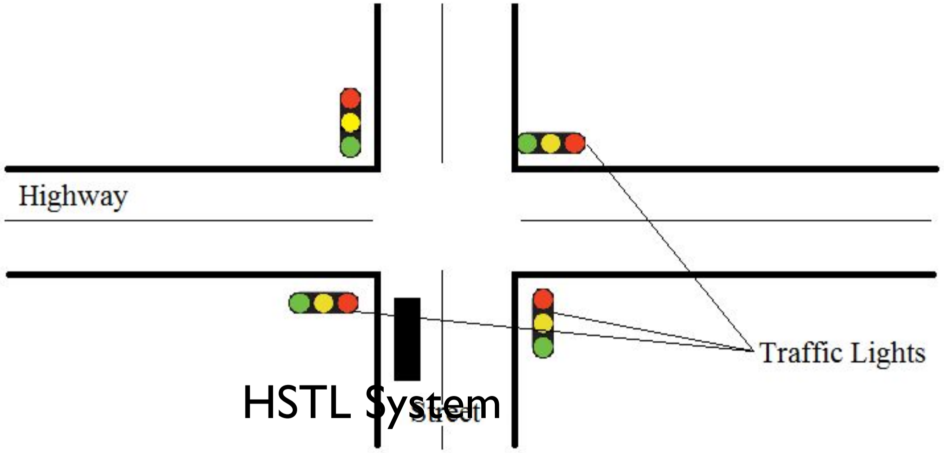
Either the farm road or the highway has a red light and both cannot be red at a same time.

A[]((red1==true | red2==true)&!(red1==true & red2==true))

Case Studies (Cont'd)

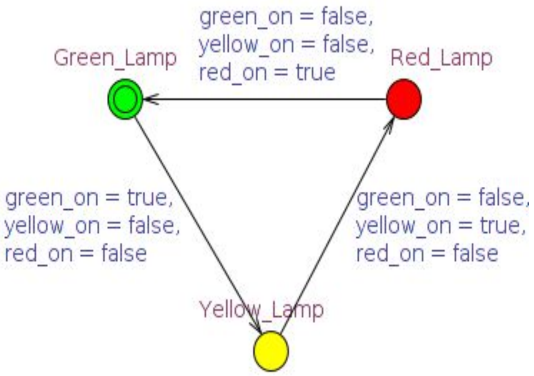
- **Second Case Study – Highway Street Traffic Light System [17]**

- Figure shows the Highway Street Traffic Light (HSTL) system where there are highway and street roads, and both facilitate 2-way traffic.
- The cross section in the center is the safety critical area and traffic light system is deployed to control the access of this shared resource.

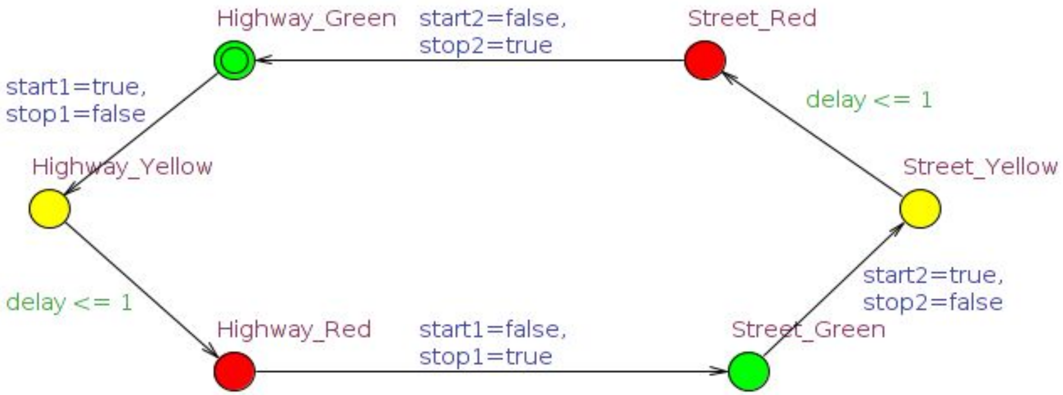


HSTL System

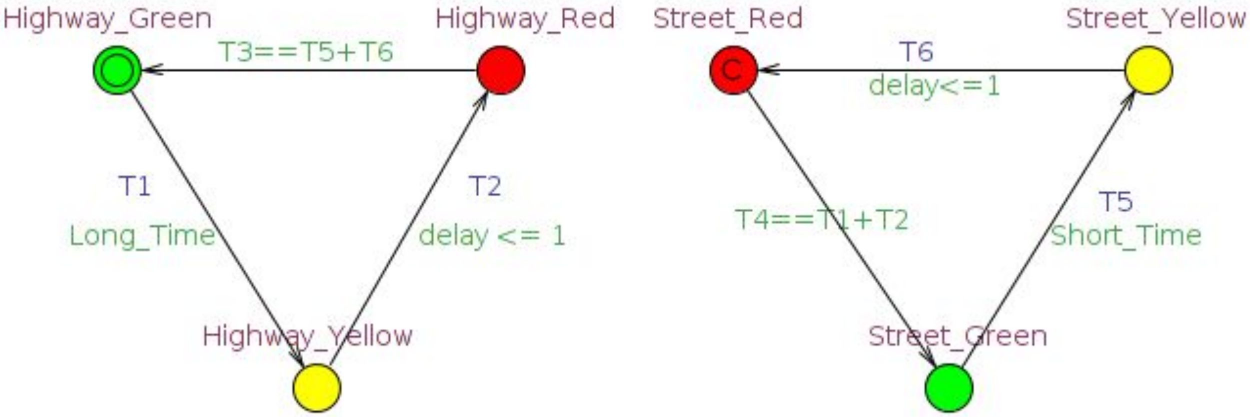
Modeling of HSTL system



Lamp Model for HSTL System



Controller Model for HSTL System



Timer Model for HSTL System

Verification of HSTL system

Some basic real-time properties are verified in CTL as follows:

i. Safety Property:

Safety property states that if the traffic is flowing through the highway road then the street traffic should be in waiting state.

A[] (Controller.start1 == true imply Controller.start2==false)

ii. Fairness Property:

Fairness deals with equal distribution and utilization of resources. In case of yellow states, the delay should be <=1 for all artifact.

A<> Controller.delay<=1

iii. Live-ness Property:

Either Highway or Street traffic should be flowing at a time to keep the system alive.

A<> (Controller.start1==true & Controller.stop1==false | Controller.start2==true & Controller.stop2==false) > (Lamp.green_on== true)

Verification of HSTL system (Cont'd)

iv. Deadlock Freeness Property:

System is in deadlock state when it remains in a same state and stopped progressing to next states.

A[] not deadlock

v. Reachability Property:

Reachability illustrates that some particular state is approachable and reachable. It is likely to reach each and every good state at least once and to verify that bad states are unreachable.

E<> (Controller.start1==true & Controller.stop1==false)