Sequential Equivalence Checking - III

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

EE 709: Testing & Verification of VLSI Circuits

Lecture – 18 (Feb 09, 2012)

How to Represent States ?

- Not practical to represent individual states
- Represent s set of states symbolically
- OBDD encodes boolean functions
  - Code elements S
- Represent a subset T as boolean function f_T
- $f_T = \{0,1\}^n \rightarrow \{0.1\}$

How to Represent States ?

- Computation uses symbolic BFS approach to all reachable states by shortest path
- Key step is image computation
  - Img(δ(s,x), C(s))
- BFS allows to deal multiple states simultaneously
- BDD is used to represent TF
- Let $t_i = \delta_i(s,x)$ $i = 1,2,\ldots n$
- C(s) is a symbolic state set

How to Represent TF ?

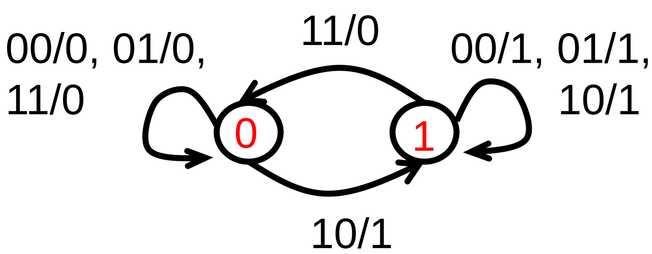
- Given a deterministic transition function (s,x) the corresponding transition relation is defined by
  - $T(s,x,t) = \Pi\ (t_i = \delta_i(s,x))$
- T(s,x,t) = 1 denotes a set of encoded tripples (s,x,t) , each representing a transition in the FST of a given FSM
- Straight forward to compute image
- Need new boolean operation
  - Existential Abstraction
  - $\exists_{xi}.f = f_{xi} + f_{x'i}$
  - f_{xi}- smallest (fewest minterm) function that contains all minterms of f and independent of xi

How to Represent TF ?

- Given $f(s,x) = f(s, .. s_n, x_1, ...x_m)$ the existential abstraction w.r.t a set of variables is defined as
  - $\exists_x .f(s,x,) = \exists_{x1}(\exists_{x2}(.. \exists_{xm}(f(s,x)))$

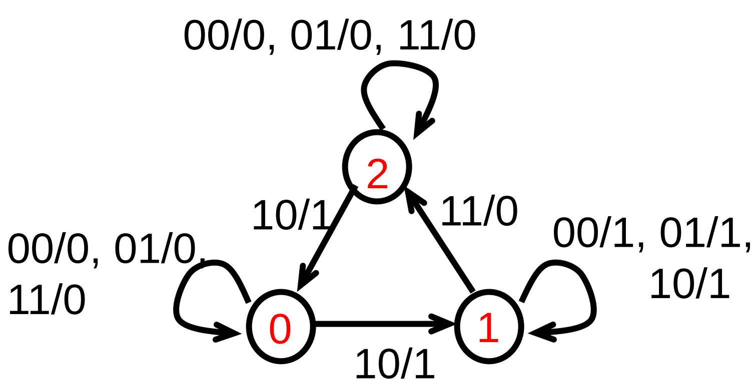
- **Procedure**
  - Compute TF, T(s,x,t)
  - Compute conjunction of T and C
  - Existentially abstract all s variable and all x variable - provides I(t)
  - I(t) is the smallest function independent of s and x which contains all the tripples in f(s,x,t)

State Reachability in Product FSM



M$_1$

- $t_1 = \delta^1{}_1 = s'_1 x_1 x_2' + s_1(x'_1 + x'_2)$
- $\lambda^1 = \delta^1{}_1$



M$_2$

- 0=00, 1=01, 2 =10
- $t_2 = \delta^2{}_2 = s_3 x_1 x_2 + s_2(x'_1 + x_2)$
- $t_3 = \delta^2{}_3 = s'_2 x_1 x'_2 + s_3(x'_1 + x_2)$
- $\lambda^2 = s_3 x'_1 + x_1 x'_2$

Symbolic FSM Traversal

Transition relation of the product machine

- $T(s, x, t) = (t_1 \equiv \delta^1{}_1).\ (t_2 \equiv \delta^2{}_2).\ (t_3 \equiv \delta^2{}_3)$

- Initial State is $s'_1 s'_2 s'_3$
- $T(s,x,t).C(s) = T(s,x,t).s'_1 s'_2 s'_3$

  $= (t_1 \equiv s'_1 x_1 x_2).(t_2 \equiv 0).(t_3 \equiv s'_2 x_1 x'_2).\ s'_1 s'_2 s'_3$

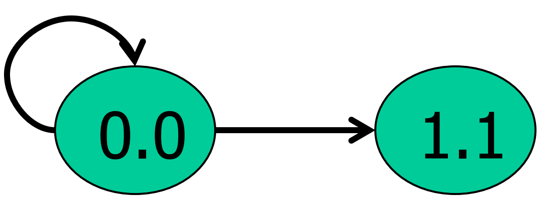
- Since this conjunction evaluate to 1 for just one s-minterm $(s'_1 s'_2 s'_3)$

Symbolic FSM Traversal

$G(x,t) = \exists_s(T(s,x,t).C(s))$

$= (t_1 \equiv x_1x'_2).(t_2 \equiv 0).(t_3 \equiv x_1x'_2)$

- Since this conjunction evaluate to 1 for just one s-minterm $(s'_1s'_2s'_3)$
- $g_{x1x'2} = (t_1 \equiv 1).(t_2 \equiv 0).(t_3 \equiv 1)$
- $g_{x'1x2} = g_{x'1x'2} = g_{x1x2} = (t_1 \equiv 0).(t_2 \equiv 0).(t_3 \equiv 0)$
- Img $(T,C) = g\ (x,t) = t'_1t'_2t'_3 + t_1t'_2t_3$



Symbolic FSM Traversal

- Implicit representation
- Graphs and their traversal are converted to Boolean functions and Boolean operations
- BDD can be use for symbolic computation

Symbolic FSM Representation

- M (Q, Σ, δ, q_0, F)
- Characteristic Function

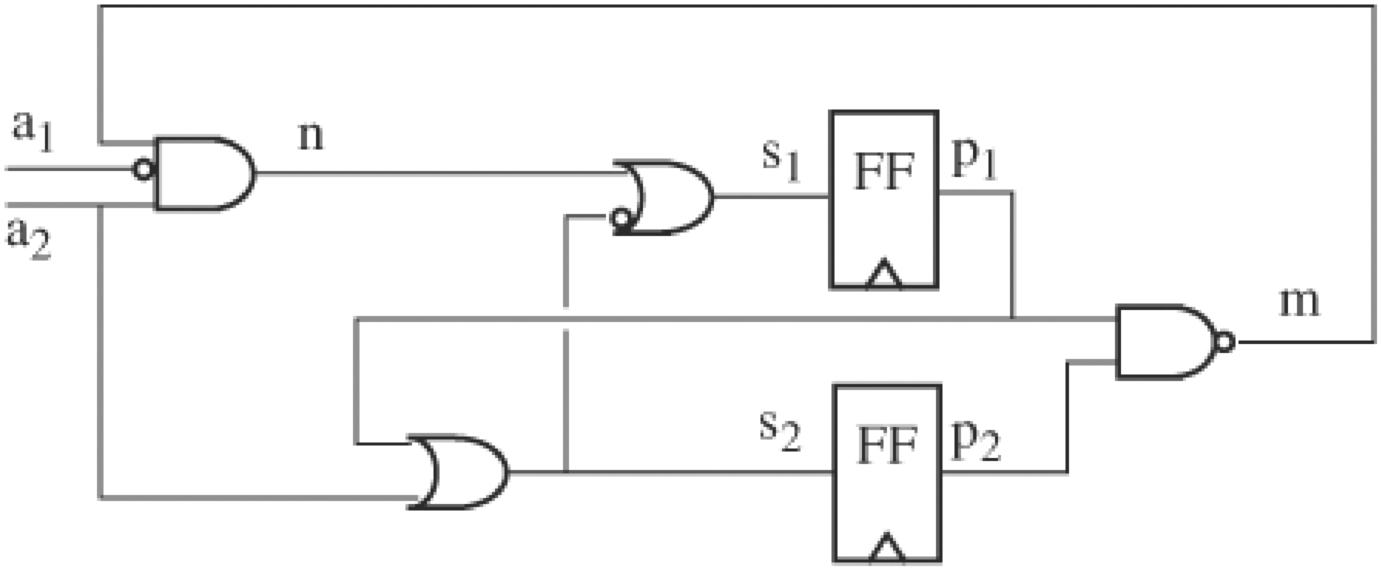
$$Q(r) = \begin{cases} 1, r \in S \\ 0, r \notin S \end{cases}$$

- Transition Function

$$T(p, n, a) = \begin{cases} 1, n = \delta(p, a) \\ 0, otherwise \end{cases}$$

$$N(s) = \begin{cases} 1, \exists(p, a)(T(p, s, a) \cdot P(p) = 1) \\ 0, otherwise \end{cases}$$

Symbolic FSM Traversal



Relational representation of transition function

$$s_1 = s'_2 + a'_1 a_2 (p_1 p_2)'$$

$$s_2 = p_1 + a_2$$

Symbolic FSM Traversal

Transition relation

$$T(p_1, p_2, s_1, s_2, a_1, a_2) = \overline{(s_1 \oplus (\overline{s_2} + \overline{a_1}a_2(\overline{p_1 p_2})))}\,\overline{(s_2 \oplus (p_1 + a_2))}$$

Next state

Present state: 00, Input: 10

$T(0,0,s_1,s_2,0,1) = S_1 . S_2$

11

Set of all next state for all possible inputs

$$N(s_1, s_2) = \exists(a_1, a_2, p_1, p_2)T(p_1, p_2, s_1, s_2, a_1, a_2)P(p_1, p_2)$$

Symbolic FSM Traversal

- set of all next states if the present state is either 00 or 11

characteristic function

$$P(p_1, p_2) = \bar{p}_1\bar{p}_2 + p_1p_2$$

Next state

$$N(s_1, s_2) = \exists(a_1, a_2, p_1, p_2)T(p_1, p_2, s_1, s_2, a_1, a_2)P(p_1, p_2)$$

$$\begin{aligned}
&\exists(a_1, a_2, p_1, p_2)T(p_1, p_2, s_1, s_2, a_1, a_2)P(p_1, p_2) \\
&= (T(p_1, p_2, s_1, s_2, 0, 0) + T(p_1, p_2, s_1, s_2, 0, 1) \\
&\quad + T(p_1, p_2, s_1, s_2, 1, 0) + T(p_1, p_2, s_1, s_2, 1, 1))P(p_1, p_2)
\end{aligned}$$

Symbolic FSM Traversal

$N(S_1 , S_2) = S_1 + S_2$

Next Sates: 01, 10, 11

Forward Reachability

Forward Reachable States by Symbolic Computation

Input: transition relation $T(p, s, a)$ and initial state $I(s)$

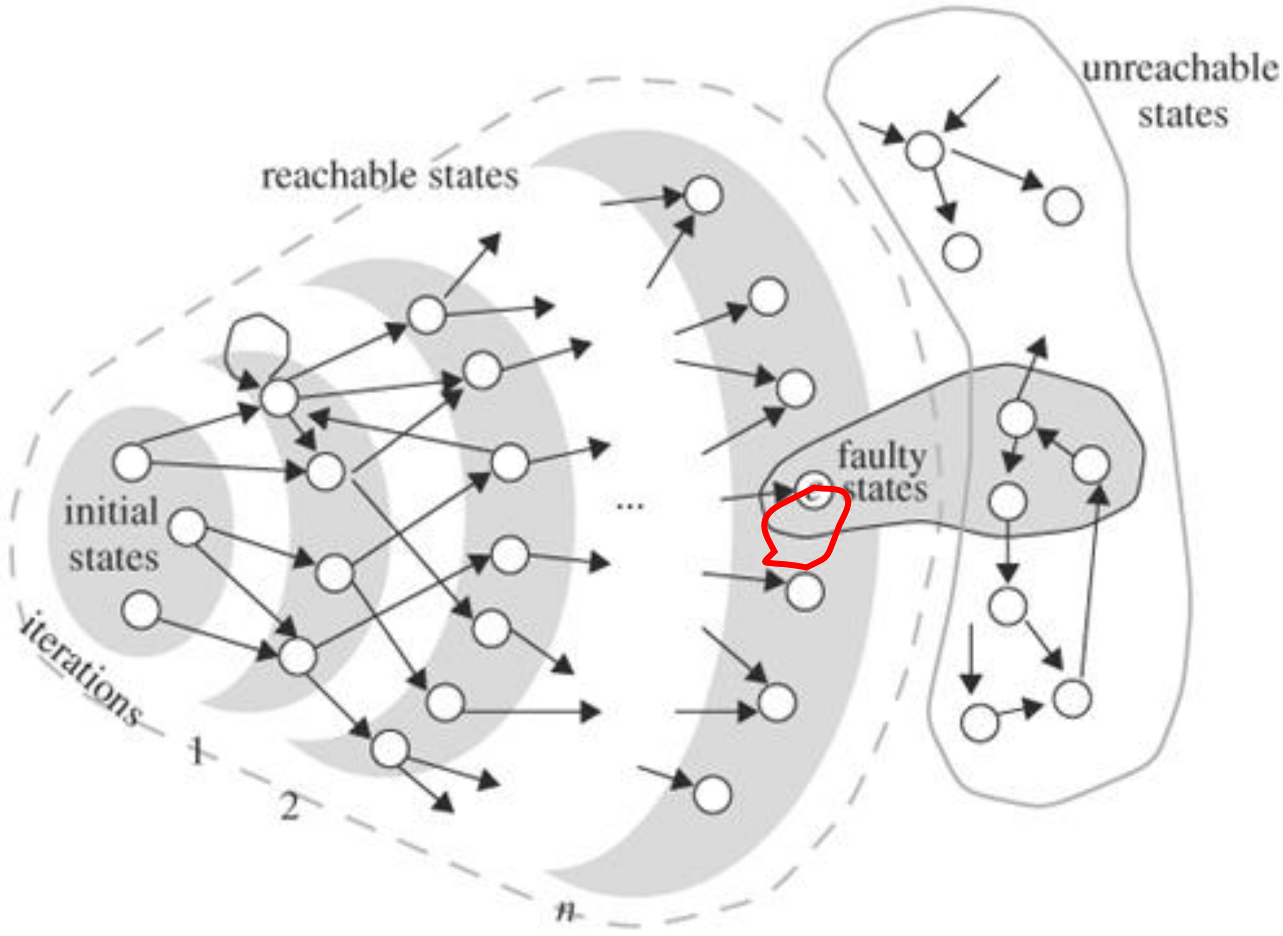
Output: a characteristic function $R(s)$ of all reachable states

ReachableState(T, I):

1. Set $S = I$
2. Compute $N(s) = \exists\, (p,a)(T(p,s,a) \cdot S(p))$
3. $R = S + N$
4. If $R \neq S$, set $S = R$ and repeat steps 2 and 3; otherwise, return R.

Forward Reachability

- BDD is used



Thank you