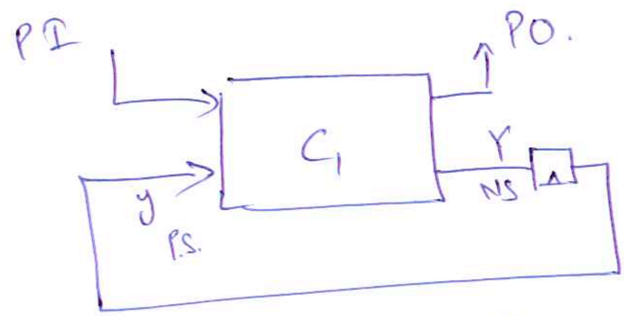
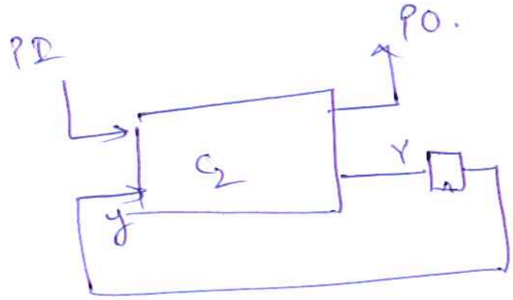
Sequential Circuit Verification

Given two sequential circuits, starting from their respective initial states, do they have the same I-O-response? i.e., for all possible input sequences, do they produce the same output sequence?



seq. ckt. S1

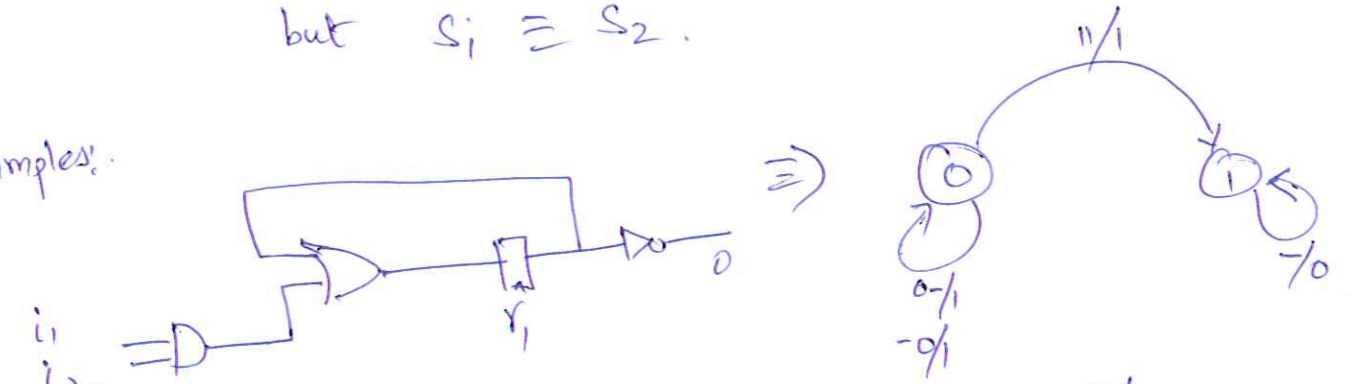


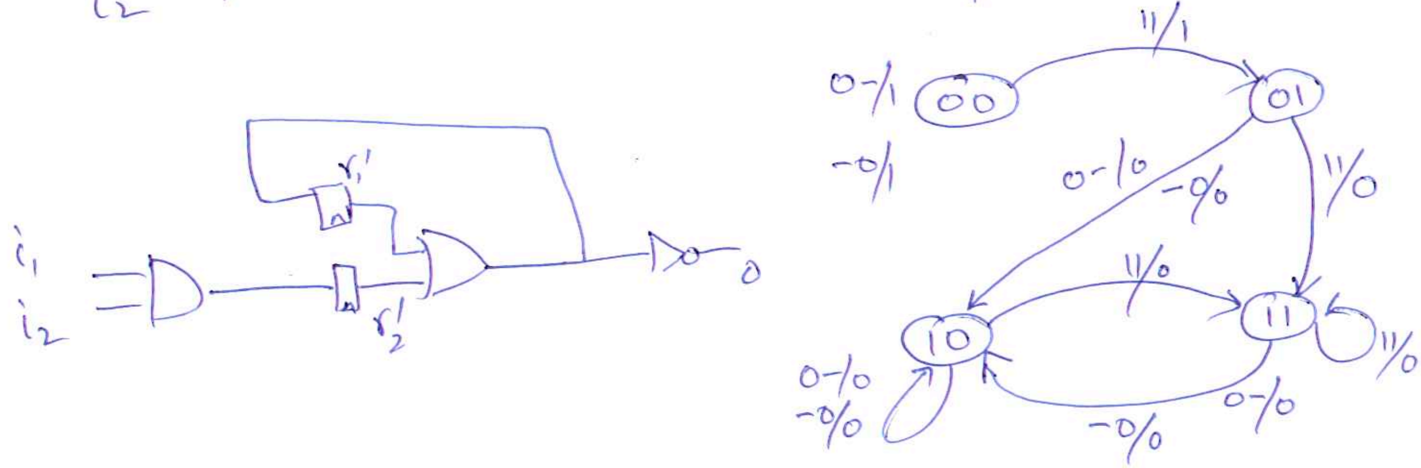
seq. ckt. S2.

C_1 may not be equivalent to C_2,

but $S_1 \equiv S_2$.

Examples:





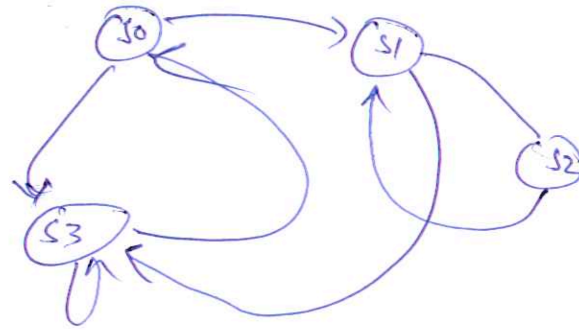
Sequential Equivalence required!-

1 Sequential optimizations

- retiming changes the # of D-FFs.
- code-reassignment

Eg Code I

FSM



S0 = 00
S1 = 01
S2 = 10
S3 = 11

code II

S0 = 11
S1 = 10
S2 = 00
S3 = 01

but I-O behaviour is the same.

but combinational logic ≠ same.

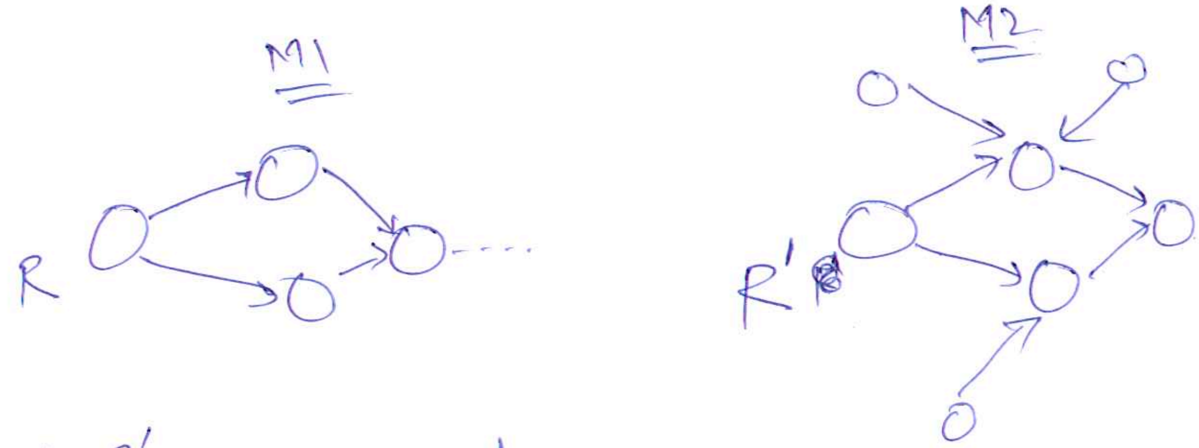
2 New problems.

equivalence of symbolic C++ model

≡ RTL (clocked Verilog)

This problem slightly different from "testing" (sequential ATPG).

Seq. verif ≡ state space analysis.



R & R' ≡ corresponding initial states.

Machine M1 ≡ M2 if

1. Both are identical
2. Identical states, but different encodings
3. $M_1 \subseteq M_2$ or $M_1 \supseteq M_2$
4. Different reachable states, but same distinguishable states.
5. Different unreachable states, but unreachable states = Don't care.

our problem:-

1. given two FSMs (i.e. given their state tables or graphs)

   M_1 & M_2

   $M_1 \equiv M_2$?

2. Given two circuits, but not their state tables, are they equivalent?

1 → "easier"

2 → extract the underlying FSMs, & then prove equivalence.

Do this efficiently as a CAD Solution.

Requires:- "Implicit state enumeration."

- Product FSM
- Implicit state enumeration, BFS-traversal
- BDDs.
- equivalence check.

FSM defined as:

$$M = (\Sigma, O, S, S^0, \Delta, \Lambda)$$

Σ = input label

O = output label

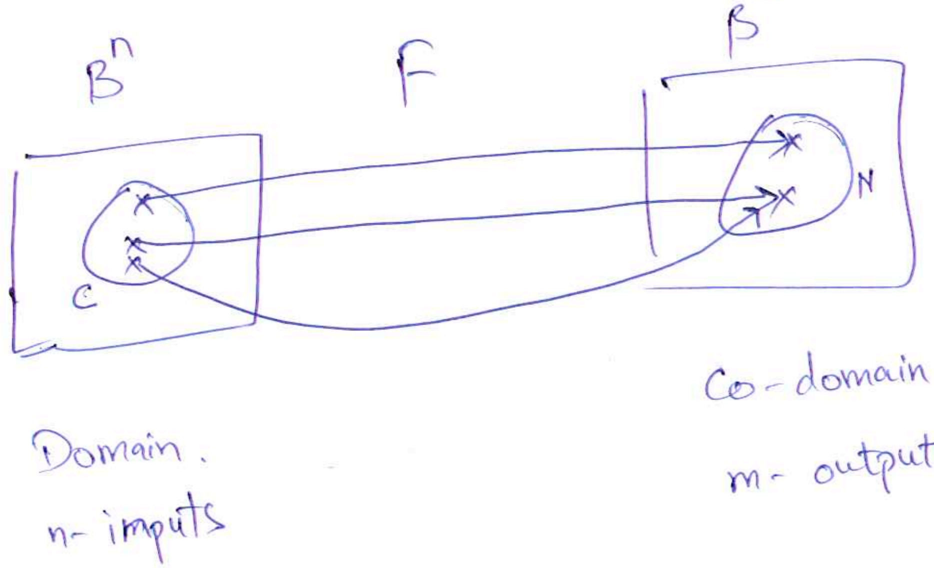
S = set of states

$S^0 \subset S$ = initial (reset) state(s)

$\Delta : S \times \Sigma \rightarrow S$, next state transition function.

$\Lambda : S \times \Sigma \rightarrow O$, output function.

For FSM traversal, we will use image computation.



N = Image of F under C = Image(F, C).

First we will understand how to ~~store~~ solve $M_1 \equiv M_2$ assuming STGs are given.

Then, we will see how to apply those concepts on a circuit.

Product Machine

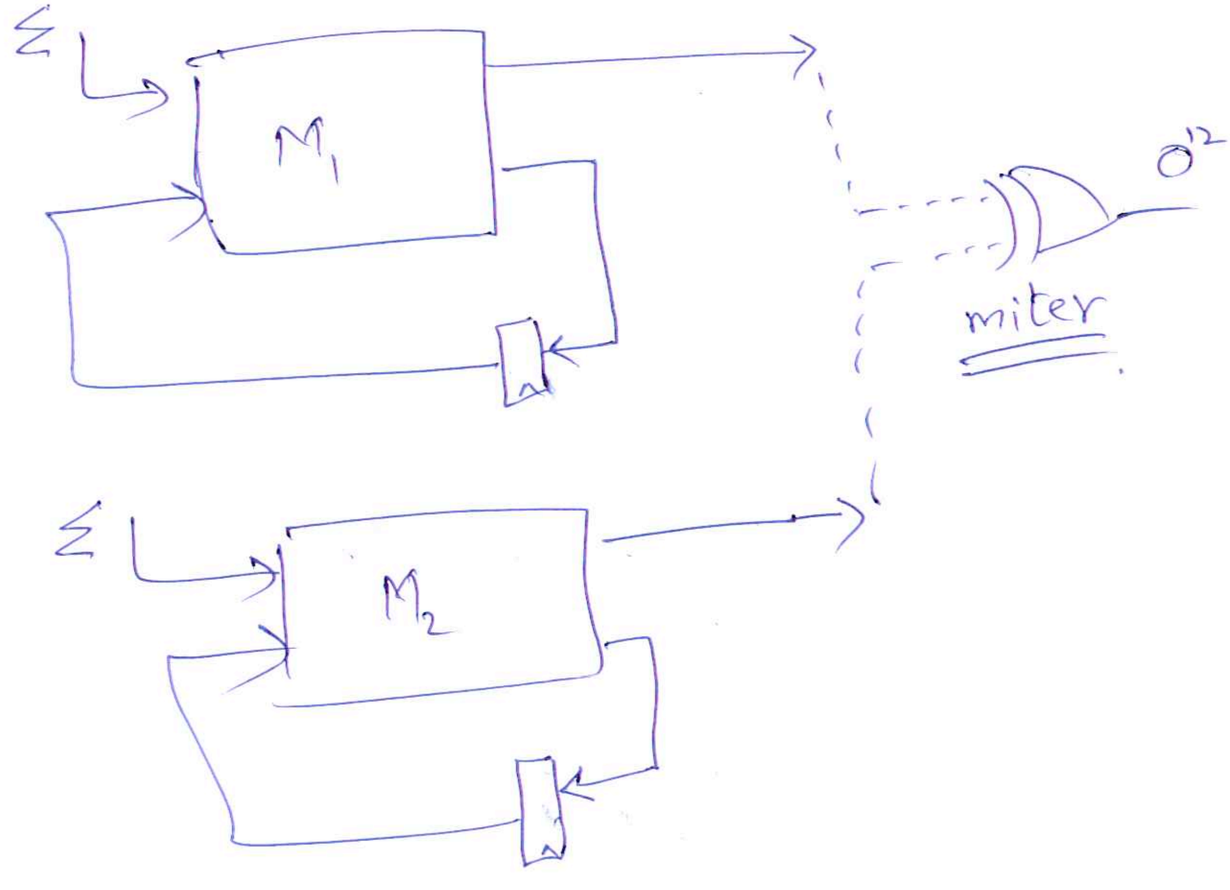
$M_1 = \{\Sigma, O, S^1, S^{01}, \Delta^1, \Lambda^1\}$

$M_2 = \{\Sigma, O, S^2, S^{02}, \Delta^2, \Lambda^2\}$

$$\boxed{M_{12} = M_1 \times M_2}$$

$M_{12} = \{\Sigma, O^{12}, S^{12}, S_0^{12}, \Delta^{12}, \Lambda^{12}\}$

Product machine.



miter

$s^1 \in S^1$, $s^2 \in S^2$

Product of states $S^{12} = S^1 \times S^2 = (S^1, S^2)$

(concatenation).

$\Delta^1(s^1, x)$ & $\Delta^2(s^2, x)$

($x \in \Sigma$ = input)

$$\Delta^{12} = \Delta^{12}(s^{12}, x) : (S^1 \times S^2) \times \Sigma \to (S^1 \times S^2)$$

$$\Rightarrow [(S^1 \times \Sigma) \to S^1]\ [(S^2 \times \Sigma) \to S^2]$$

$$\Rightarrow [\Delta^1(S^1, x), \Delta^2(S^2, x)]$$

$\Rightarrow$ conjunction of transition relations.

Output $\Lambda^{12} = \Lambda^{12}(s, x) : (S^1 \times S^2) \times \Sigma \to \{0, 1\}$.

$$\Lambda^{12} = 1 \text{ if } \lambda^1(S^1, x) = \lambda^2(S^2, x)$$

$= 0$ otherwise.