Sequential Circuits: State Minimization

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

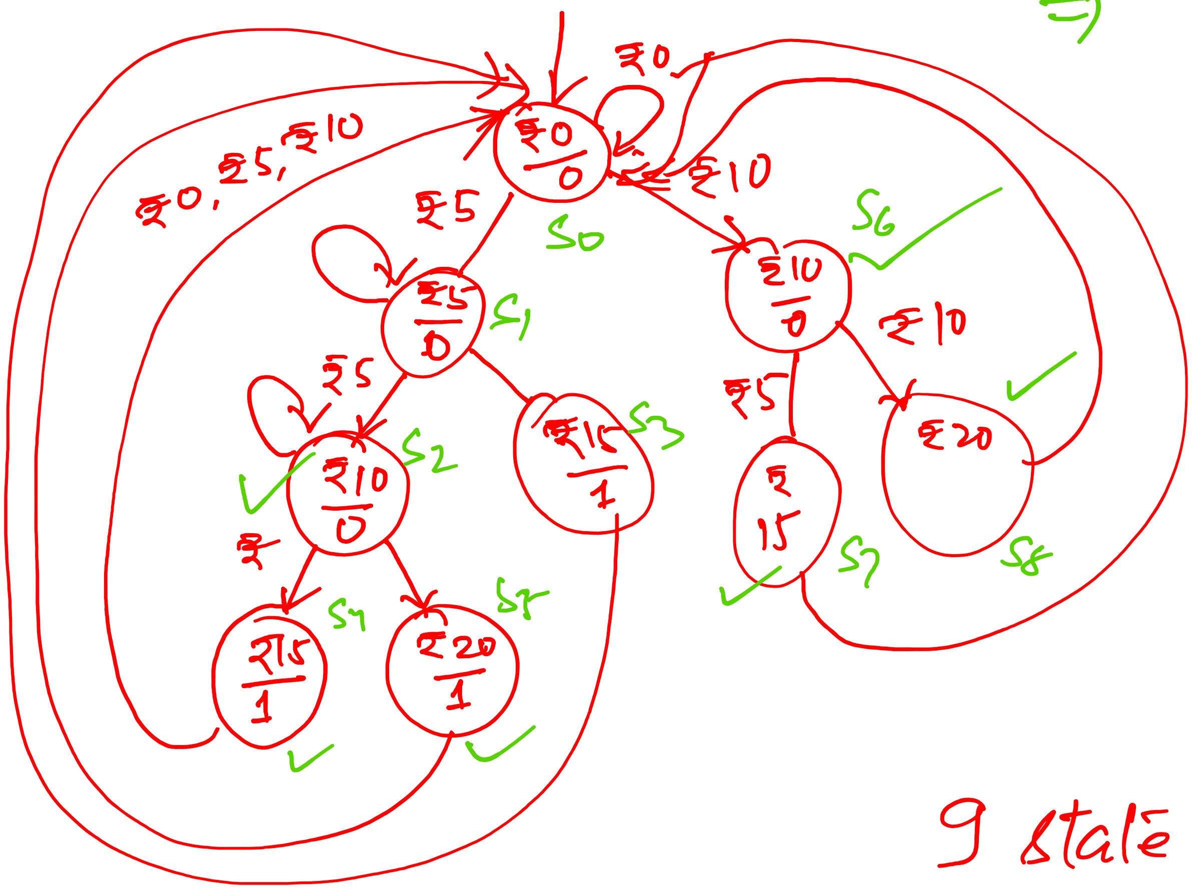
CS-226: Digital Logic Design

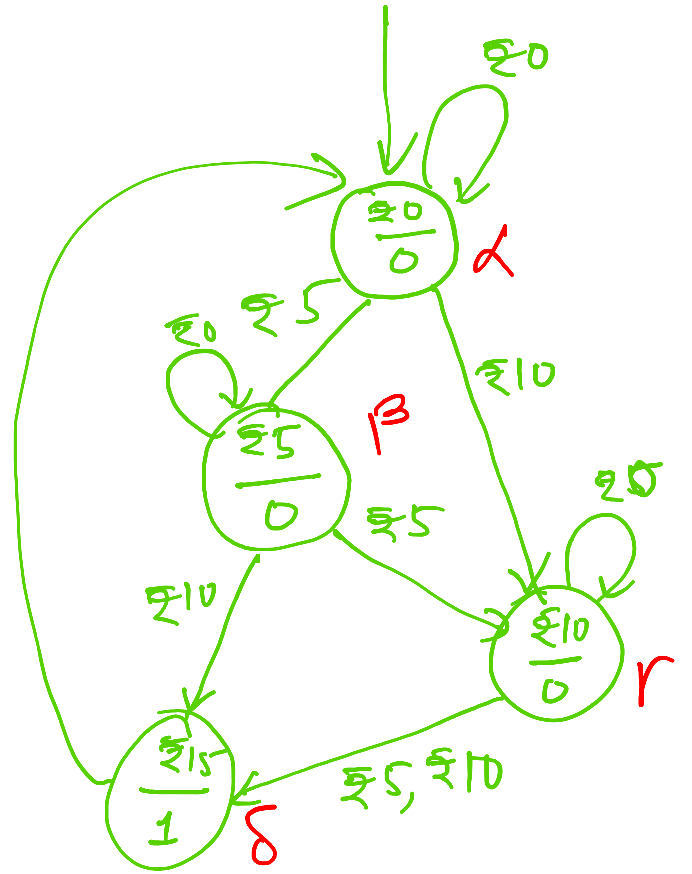
Lecture 23: 30 March 2021

Finite State Machine

$M = (I, O, S, S_0, \delta, \lambda)$

Vending Machine





9 state → 4 states

4 variable

2 variable.

State Minimization

X-Successor – If an input sequence X takes a machine from state S_i to state S_j, then S_j is said to be the X-successor of S_j

Strongly connected:- If for every pair of states (S_i, S_j) of a machine M there exists an input sequence which takes M from state S_i to S_j, then M is said to be strongly connected

State Equivalence

- Two states S_i and S_j of machine M are distinguishable if and only if there exists at least one finite input sequence which, when applied to M, causes different output sequences, depending on whether S_i or S_j is the initial state
- The sequence which distinguishes these states is called a distinguishing sequence of the pair (S_i, S_j)
- If there exists for pair (S_i, S_j) a distinguishing sequence of length k, the states in (S_i, S_j) are said to be k-distinguishable

State Equivalence

Machine M1

PS	NS, z	
	X = 0	X = 1
A	E, 0	D, 1
B	F, 0	D, 0
C	E, 0	B, 1
D	F, 0	B, 0
E	C, 0	F, 1
F	B, 0	C, 0

(A, B) – 1 Distinguishable

$A \xrightarrow{0} E/0$

$A \xrightarrow{1} D/1$

$B \xrightarrow{0} F/0$

$B \xrightarrow{1} D/0$

distinguishable

X = 1

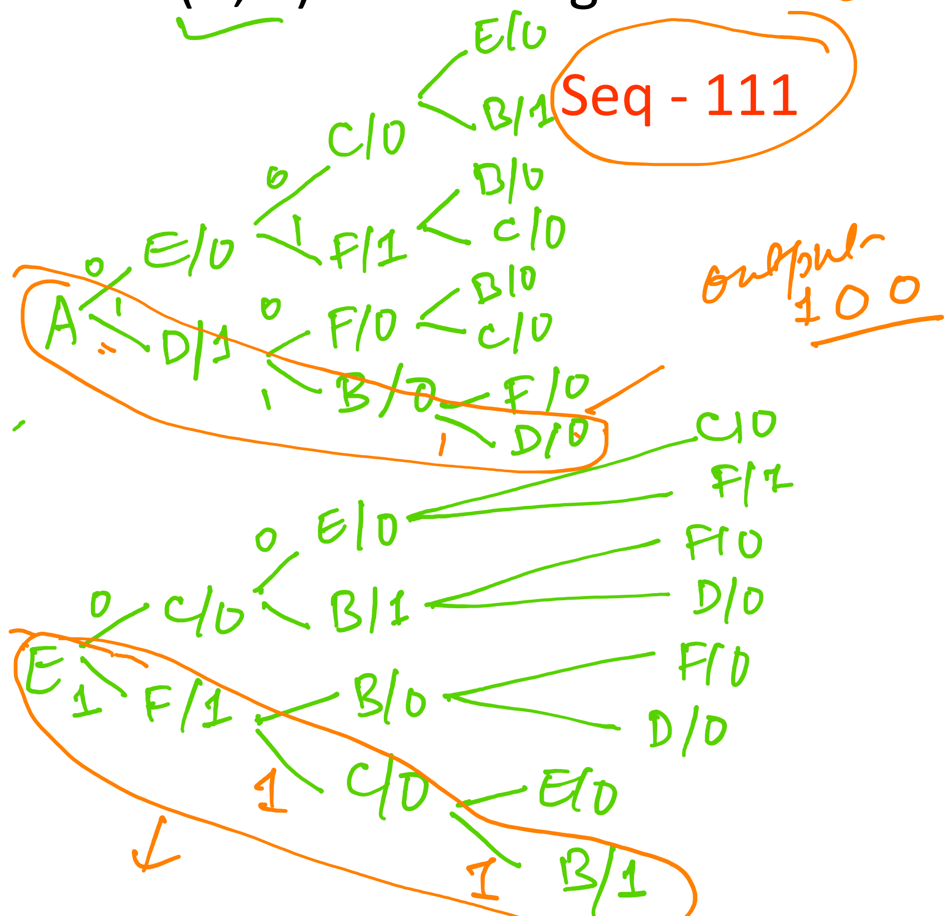
State Equivalence

2- Indistinguishable.

Machine M1

PS	NS, z	
	X = 0	X = 1
A	E, 0	D, 1
B	F, 0	D, 0
C	E, 0	B, 1
D	F, 0	B, 0
E	C, 0	F, 1
F	B, 0	C, 0

(A, E) – 3 Distinguishable



output 1 0 1

State Equivalence

Machine M1

PS	NS, z	
	X = 0	X = 1
A	E, 0	D, 1
B	F, 0	D, 0
C	E, 0	B, 1
D	F, 0	B, 0
E	C, 0	F, 1
F	B, 0	C, 0

(A, B) – 1 Distinguishable

(A, E) – 3 Distinguishable

2- equivalent

1- equivalent

Seq - 111

k-equivalent – The states that are not *k*-distinguishable are said to be k-equivalent

Also *r-equivalent* r<k

Distiguishable States

K- distinguishable

K-1 equivalent

State Equivalence

- States S_i and S_j of machine **M** are said to be equivalent if and only if, for every possible input sequence, the same output sequence will be produced regardless of whether S_i or S_j is the initial state

- States that are k-equivalent for all k < n-1, are equivalent

- $S_i = S_j$, and $S_j = S_k$, then $S_i = S_k$

(S_i, S_j, S_k)
equivalence set

State Equivalence

- The set of states of a machine M can be partitioned into disjoint subsets, known as equivalence classes
- Two states are in the same equivalence class if and only if they are equivalent, and are in different classes if and only if they are distinguishable

Property: If S_i and S_j are equivalent states, their corresponding X-successors, for all X, are also equivalent

S_p, S_q

State Minimization Procedure

1. Partition the states of M into subsets s.t. all states in same subset are *1-equivalent*
2. Two states are 2-equivalent iff they are 1-equivalent and their I_i successors, for all possible I_i, are also 1-equivalent

PS	NS, z	
	X = 0	X = 1
A	E, 0	D, 1
B	F, 0	D, 0
C	E, 0	B, 1
D	F, 0	B, 0
E	C, 0	F, 1
F	B, 0	C, 0

6

n

P0 = (ABCDEF) – 0-equivalent

P1 = (ACE), (BDF) – 1-equivalent

P2 = (ACE), (BD), (F) – 2-equivalent

P3 = (AC), (E), (BD), (F) – 3-equivalent

P4 = (AC), (E), (BD), (F) – 4-equivalent

(AC) → E, (BD)

$(ACE) \xrightarrow{0} (E, C)$, $\xrightarrow{1} (B, D, F)$

$(BDF) \xrightarrow{0} (F, B)$, $\xrightarrow{1} (B, D), (C)$

(BD) – F, (BD)

Machine Equivalence

- Two machines M1, M2 are said to be equivalent if and only if, for every state in M1, there is corresponding equivalent state in M2
- If one machine can be obtained from the other by relabeling its states they are said to be isomorphic to each other

PS	NS, z	
	X = 0	X = 1
AC - α	β, 0	γ, 1
E - β	α, 0	δ, 1
BD - γ	δ, 0	γ, 0
F - δ	γ, 0	α, 0

State Equivalence - Example

Machine M2

PS	NS, z	
	X = 0	X = 1
A	E, 0	C, 0
B	C, 0	A, 0
C	B, 0	G, 0
D	G, 0	A, 0
E	F, 1	B, 0
F	E, 0	D, 0
G	D, 0	G, 0

P0 = (ABCDEFG)

1-equivalent

P1 = (ABCDFG) (E)

2-equivalent

P2 = (AF) (BCDG) (E)

3-equivalent

P3 = (AF) (BD) (CG) (E)

P4 = (A) (F) (BD) (CG) (E)

P5 = (A) (F) (BD) (CG) (E)

BD, CG, A

CG, BD, GD.

AF, E, (C,D)

(BCDG), CBGD, AG

Finite State Machine

K- equivalence classes

If successors are also K - equivalent

then States are (K+1) equivalent

Partition the states until

- get no more partition
- n times

n state machine

Thank You