

Switching Circuits & Logic Design

Jie-Hong Roland Jiang
江介宏

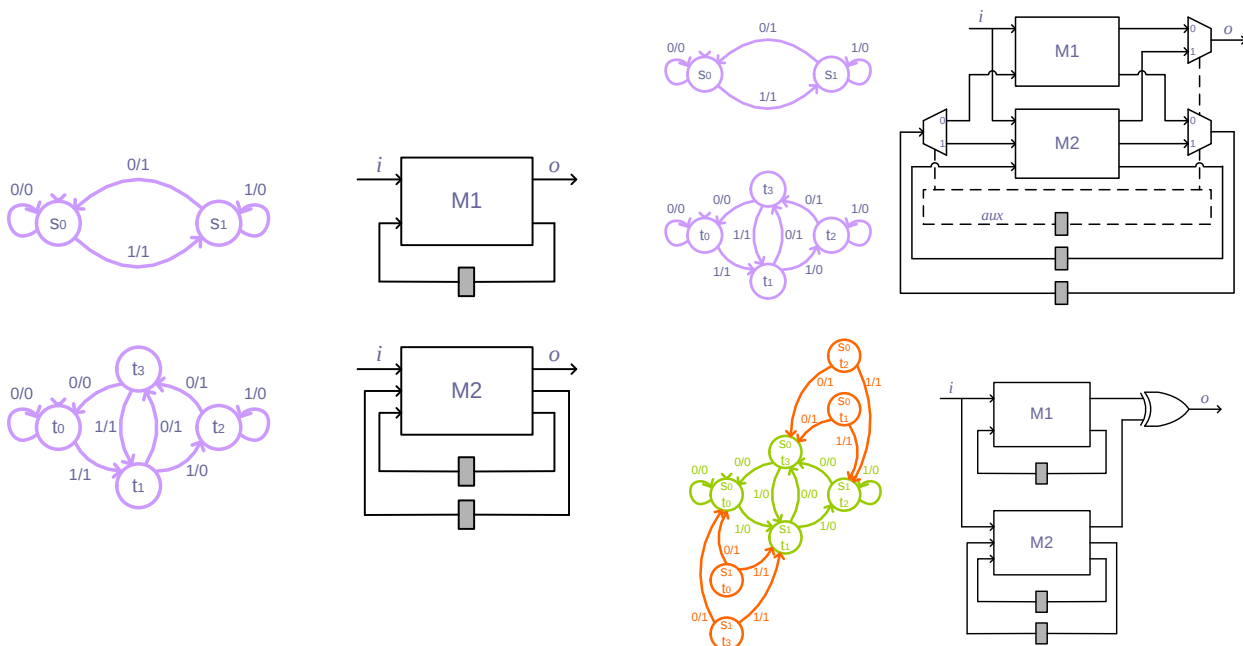
Department of Electrical Engineering
National Taiwan University



Fall 2012

1

§15 Reduction of State Tables, State Assignment



2

Outline

- Elimination of redundant states
- Equivalent states

Not in exam:

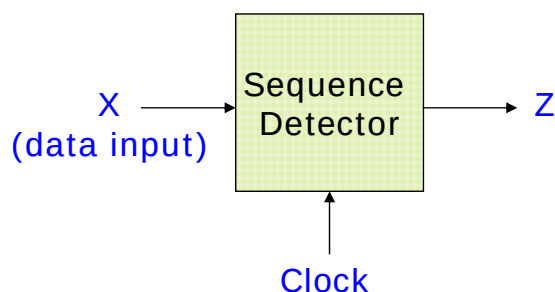
- Determination of state equivalence using an implication table
- Equivalent sequential circuits
- State assignment

3

Elimination of Redundant States

□ Example (§14.3)

Block diagram



$Z=1 \Leftrightarrow$ input sequence 0101 or 1001 occurs

The circuit examines groups of 4 consecutive inputs, and resets after every 4 inputs

Input/output sequence example

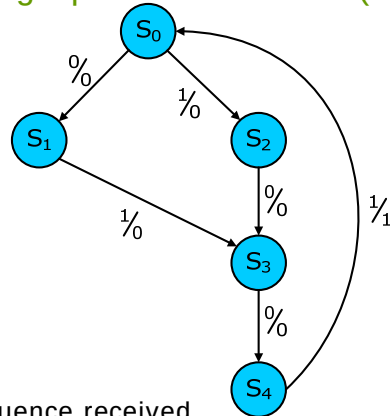
X =	0101	0010	1001	0100
Z =	0001	0000	0001	0000

4

Elimination of Redundant States

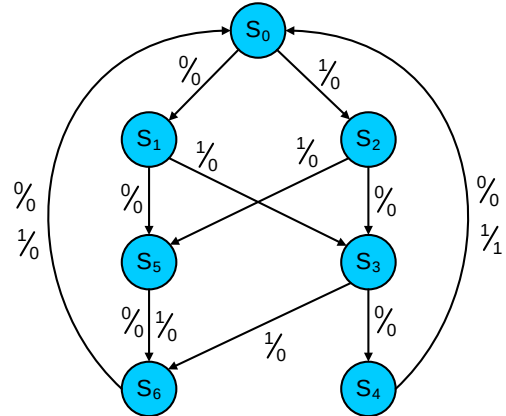
Mealy machine implementation (recap)

(1) Partial graph



State	Sequence received
S ₀	reset
S ₁	0
S ₂	1
S ₃	01 or 10
S ₄	010 or 100

(2) Complete state graph



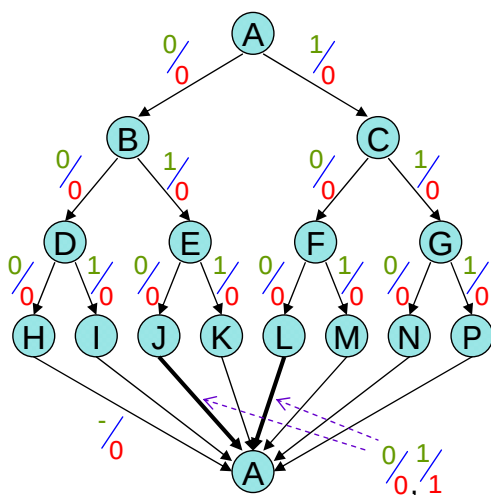
State	Sequence received
S ₅	two inputs received, no 1 output is possible
S ₆	three inputs received, no 1 output is possible

5

Elimination of Redundant States

State table for {0101, 1001} sequence detector

Consider all possible input sequences of length four



Input Sequence	Present State	Next State		Present Output	
		X=0	X=1	X=0	X=1
reset	A	B	C	0	0
0	B	D	E	0	0
1	C	F	G	0	0
00	D	H	I	0	0
01	E	J	K	0	0
10	F	L	M	0	0
11	G	N	P	0	0
000	H	A	A	0	0
001	I	A	A	0	0
010	J	A	A	0	1
011	K	A	A	0	0
100	L	A	A	0	1
101	M	A	A	0	0
110	N	A	A	0	0
111	P	A	A	0	0

6

Elimination of Redundant States

Input Sequence	Present State	Next State		Present Output	
		X=0	X=1	X=0	X=1
reset	A	B	C	0	0
0	B	D	E	0	0
1	C	F	G	0	0
00	D	H	H	0	0
01	E	J	K	0	0
10	F	J	M	0	0
11	G	N	P	0	0
000	H	A	A	0	0
001	I	A	A	0	0
010	J	A	A	0	1
011	K	A	A	0	0
100	L	A	A	0	1
101	M	A	A	0	0
110	N	A	A	0	0
111	P	A	A	0	0

- If two states have the same next state and the same output under every possible input, then they are equivalent states (**the converse is not true!**)

- {H,I,K,M,N,P} and {J,L} are equivalent state sets

- For every equivalent state set, we can take any of its states as the representative and replace the other states with this representative

- E.g., take H for {H,I,K,M,N,P} and take J for {J,L}

7

Elimination of Redundant States

Input Sequence	Present State	Next State		Present Output	
		X=0	X=1	X=0	X=1
reset	A	B	C	0	0
0	B	D	E	0	0
1	C	F	G	0	0
00	D	H	H	0	0
01	E	J	K	0	0
10	F	J	M	0	0
11	G	N	P	0	0
000	H	A	A	0	0
001	I	A	A	0	0
010	J	A	A	0	1
011	K	A	A	0	0
100	L	A	A	0	1
101	M	A	A	0	0
110	N	A	A	0	0
111	P	A	A	0	0

- After substituting H for I,K,M,N,P, and substituting J for L, we see that {D,G} and {E,F} are again equivalent state sets

- I.e., having the same next state and the same output under every possible input

- Taking D as the representative for equivalent state set {D,G} and E for {E,F}, we can eliminate rows of G and F

8

Elimination of Redundant States

Input Sequence	Present State	Next State		Present Output	
		X=0	X=1	X=0	X=1
reset	A	B	C	0	0
0	B	D	E	0	0
1	C	F E	G D	0	0
00	<u>D</u>	H	I H	0	0
01	<u>E</u>	J	K H	0	0
10	F	L J	M H	0	0
11	G	N H	P H	0	0
000	<u>H</u>	A	A	0	0
001	I	A	A	0	0
010	<u>J</u>	A	A	0	1
011	K	A	A	0	0
100	L	A	A	0	1
101	M	A	A	0	0
110	N	A	A	0	0
111	P	A	A	0	0

At the end of the above procedure, known as **row matching**, we have 7 states A,B,C,D,E,H,J left

These 7 states may or may not be equivalent

Their equivalences need to be further determined by the method of §15.2 and §15.3

In this example, the 7 states happen to be inequivalent

Row matching is not sufficient to find all equivalent states (**why?**)

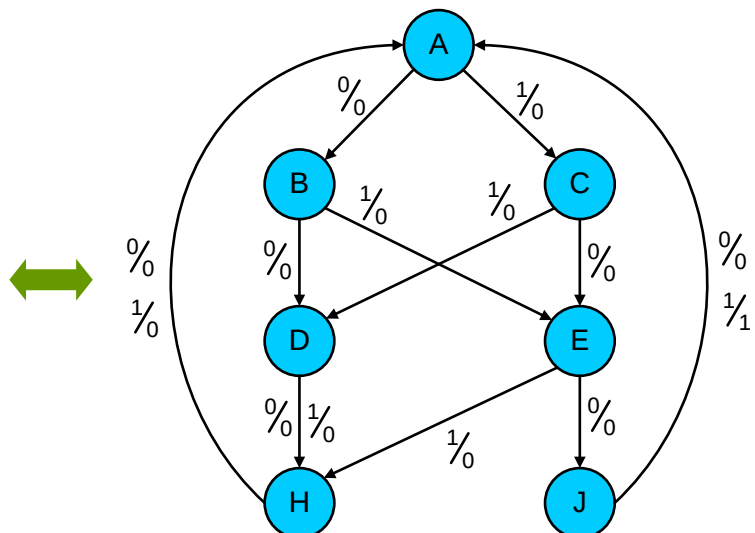
It works however in the special case where the circuit resets to the starting state after receiving a fixed number of inputs (**why?**)

9

Elimination of Redundant States

Reduced state table and state graph

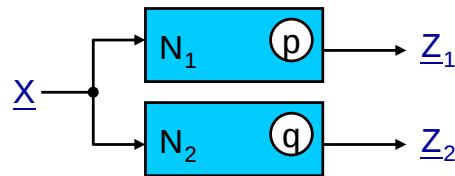
Present State	Next State		Output	
	X=0	X=1	X=0	X=1
A	B	C	0	0
B	D	E	0	0
C	E	D	0	0
D	H	H	0	0
E	J	H	0	0
H	A	A	0	0
J	A	A	0	1



10

Equivalent States

- Two states are **equivalent** if there is no way of telling them apart through **observation of the circuit inputs and outputs**
- Consider two sequential circuits N_1 and N_2 (they may be different circuits or two copies of the same circuit), one starting in state p and one in state q
 - If the output sequences $\underline{Z}_1$ and $\underline{Z}_2$ are the same (**different**) for every (**some**) input sequence $\underline{X}$, then states p and q are equivalent (**inequivalent**)
 - we write $\underline{Z}_1 = \lambda_1(p, \underline{X})$ and $\underline{Z}_2 = \lambda_2(q, \underline{X})$
(because the output sequence is a function of the initial state and the input sequence)



11

Equivalent States

- **Definition 15.1**
Let N_1 and N_2 be sequential circuits (not necessarily different). Let $\underline{X}$ represent a sequence of inputs of arbitrary length. Then state p in N_1 is **equivalent** to state q , denoted $p \equiv q$, in N_2 iff $\lambda_1(p, \underline{X}) = \lambda_2(q, \underline{X})$ for every possible input sequence $\underline{X}$.
 - Symbol " $\equiv$ " here is different from XNOR
- **Theorem 15.1** (proof in Appendix C)
Two states p and q of a sequential circuit are equivalent iff for every single input X , the outputs are the same and the next states are equivalent, i.e.,

$$\lambda(p, X) = \lambda(q, X) \text{ and } \delta(p, X) \equiv \delta(q, X)$$
 where $\lambda(p, X)$ and $\delta(p, X)$ are the output and the next state, respectively, given the present state p and input X .
 - Note that the next states don't need to be the same ($=$) (used in row matching), but just equivalent ($\equiv$)
 - E.g., $D \equiv G$ in the table of Slide 6, but their next states (H and N for $X=0$, and I and P for $X=1$) are not equal
 - Row matching is a special case of Theorem 15.1

12

Equivalent States

□ Example (Table 13.4)

- Show no equivalent states

Present State	Next State				Present Output (Z_1Z_2)			
	$X_1X_2 = 00$	01	10	11	$X_1X_2 = 00$	01	10	11
S_0	S_3	S_2	S_1	S_0	00	10	11	01
S_1	S_0	S_1	S_2	S_3	10	10	11	11
S_2	S_3	S_0	S_1	S_1	00	10	11	01
S_3	S_2	S_2	S_1	S_0	00	00	01	01

- From the outputs, we know only S_0 and S_2 can possibly be equivalent. Moreover,

$$S_0 \equiv S_2 \text{ iff } S_3 \equiv S_3, S_2 \equiv S_0, S_1 \equiv S_1, \text{ and } S_0 \equiv S_1$$

But $S_0 \neq S_1$ (because the outputs differ), so $S_0 \neq S_2$

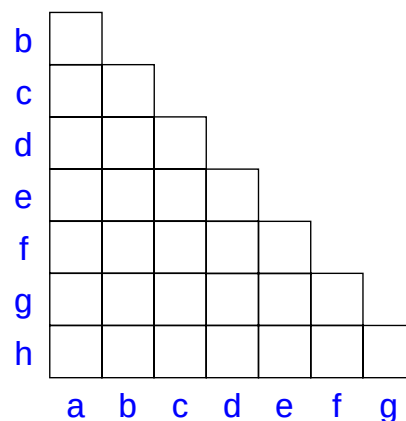
13

Determination of State Equivalence (Not in Exam)

- Use an **implication table** (a **pair chart**) to check each pair of states for possible equivalence

- Non-equivalent pairs are systematically eliminated until only the equivalent pairs remain
- This chart has a square for every possible states; a square in column i and row j corresponds to state pair $i-j$

Present State	Next State		Present Output
	$X = 0$	1	
a	d	c	0
b	f	h	0
c	e	d	1
d	a	e	0
e	c	a	1
f	f	b	1
g	b	h	0
h	c	g	1



14

Determination of State Equivalence

Example

Present State	Next State		Present Output
	X = 0	1	
a	d	c	0
b	f	h	0
c	e	d	1
d	a	e	0
e	c	a	1
f	f	b	1
g	b	h	0
h	c	g	1

b	d-f c-h						
c	X	X					
d	a-d c-e	a-f e-h	X				
e	X	X	c-e a-d	X			
f	X	X	e-f b-d	X	c-f a-b		
g	b-d c-h	b-f	X	a-b e-h	X	X	
h	X	X	c-e d-g	X	a-g	c-f b-g	X
	a	b	c	d	e	f	g

$a \equiv b$ iff $d \equiv f$ and $c \equiv h$
 $b \not\equiv c$ since outputs differ

15

Determination of State Equivalence

Example (cont'd)

b	d-f c-h						
c	X	X					
d	c-e	a-f e-h	X				
e	X	X	a-d	X			
f	X	X	e-f b-d	X	c-f a-b		
g	b-d c-h	b-f	X	a-b e-h	X	X	
h	X	X	c-e d-g	X	a-g	c-f b-g	X
	a	b	c	d	e	f	g

16

Determination of State Equivalence

Example (cont'd)

b	d-f c-h						
c	X	X					
d	c-e	a-f e-h	X				
e	X	X	a-d	X			
f	X	X	e-f b-d	X	c-f a-b		
g	b-d c-h	b-f	X	a-b e-h	X	X	
h	X	X	c-e d-g	X	a-g	c-f b-g	X
	a	b	c	d	e	f	g

Reduced State Table

Present State	Next State		Present Output
	X = 0	1	
a	a	c	0
b	f	h	0
c	c	a	1
f	f	b	1
g	b	h	0
h	c	g	1

17

Equivalent Sequential Circuits (Not in Exam)

Definition 15.2

Sequential circuit N_1 is **equivalent** to circuit N_2 if for each state p in N_1 , there is a state q in N_2 such that $p \equiv q$, and conversely, for each state s in N_2 , there is a state t in N_1 such that $s \equiv t$

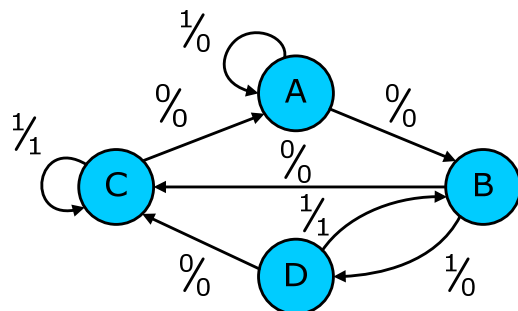
- If both N_1 and N_2 have a minimum number of states (i.e., state minimized) and $N_1 \equiv N_2$, then N_1 and N_2 must have the same number of states

18

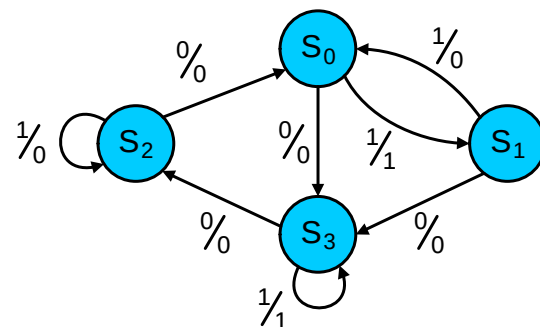
Equivalent Sequential Circuits

Example

		N ₁		
	X=0	1	X=0	1
A	B	A	0	0
B	C	D	0	1
C	A	C	0	1
D	C	B	0	0



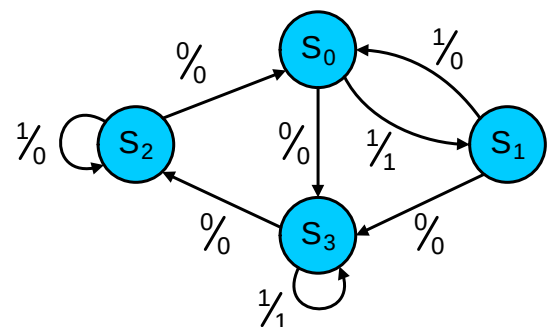
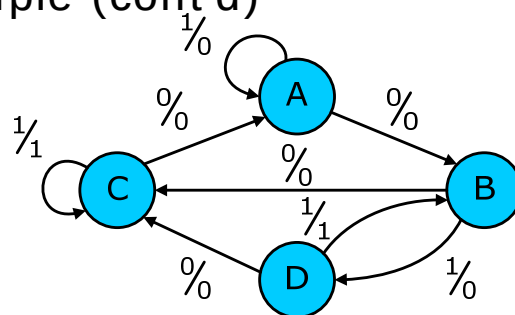
		N ₂		
	X=0	1	X=0	1
S ₀	S ₃	S ₁	0	1
S ₁	S ₃	S ₀	0	0
S ₂	S ₀	S ₂	0	0
S ₃	S ₂	S ₃	0	1



19

Equivalent Sequential Circuits

Example (cont'd)



S ₀	X	C-S ₃ D-S ₁	A-S ₃ C-S ₁	X
S ₁	B-S ₃ A-S ₀	X	X	C-S ₃ B-S ₀
S ₂	B-S ₀ A-S ₂	X	X	C-S ₀ B-S ₂
S ₃	X	C-S ₂ D-S ₃	A-S ₂ C-S ₃	X
	A	B	C	D

S ₀	X	C-S ₃ D-S ₁	A-S₃ C-S₁	X
S ₁	B-S₃ A-S₀	X	X	C-S ₃ B-S ₀
S ₂	B-S ₀ A-S ₂	X	X	C-S₀ B-S₂
S ₃	X	C-S₂ D-S₃	A-S ₂ C-S ₃	X
	A	B	C	D

20

State Assignment

(Not in Exam)

- After the number of states in a state table has been reduced, the flip-flop input equations can be derived as follows
 1. Perform **state assignment** (assign flip-flop state values to correspond to the states in the reduced table)
 - The cost of the logic required to realize a sequential circuit is strongly dependent on the way this state assignment is made (subject of §15.7 ~ §15.9)
 2. Construct a transition table which gives the next states of the flip-flops as a function of the present states and inputs
 3. Derive the next-state maps from the transition table
 4. Find flip-flop maps from the next-state maps using the techniques of §12 and find the flip-flop input equations from the maps