# Switching Circuits \＆ Logic Design 

Jie－Hong Roland Jiang
江介宏
Department of Electrical Engineering National Taiwan University


Fall 2014

## §15 Reduction of State Tables，State Assignment



## Outline

-Elimination of redundant states
-Equivalent states

Not in exam:
-Determination of state equivalence using an implication table
-Equivalent sequential circuits
-State assignment

## Elimination of Redundant States

-Example (§14.3)
Block diagram


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

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

I nput/ output sequence example

| $X=0101$ | 0010 | 1001 | 0100 |
| :--- | :--- | :--- | :--- |
| $Z=0001$ | 0000 | 0001 | 0000 |

## Elimination of Redundant States

Mealy machine implementation (recap)
(1) Partial graph

(2) Complete state graph

$\mathrm{S}_{1} \quad 0$
$\mathrm{S}_{2} 1$
$\mathrm{S}_{3} \quad 01$ or 10

| $\mathrm{S}_{4}$ | 010 or 100 |
| :--- | :--- |

## Elimination of Redundant States

$\square$ State table for $\{0101,1001\}$ sequence detector
Consider all possible input sequences of length four


## Elimination of Redundant States

| Input Sequence | Present State | Next State |  | Present <br> Output |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | $\mathrm{X}=0$ | $\mathrm{X}=1$ | $\mathrm{X}=0$ | $\mathrm{X}=1$ |
| reset | A | B | C | 0 | 0 |
| 0 | B | D | E | 0 | 0 |
| 1 | C | F | G | 0 | 0 |
| 00 | D | H | $\times \mathrm{H}$ | 0 | 0 |
| 01 | E | J | K H | 0 | 0 |
| 10 | F | $\pm J$ | M H | 0 | 0 |
| 11 | G | NH | RH | 0 | 0 |
| 000 | (H) | A | A | 0 | 0 |
| -001 |  | A | A | 0 | 0 |
| 010 | (1) | A | A | 0 | 1 |
| -011 | , |  | A | 0 | 0 |
| 100 | L |  | 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 $\{\mathrm{H}, \mathrm{I}, \mathrm{K}, \mathrm{M}, \mathrm{N}, \mathrm{P}\}$ and take J for $\{J, L\}$


## Elimination of Redundant States

| Input Sequence | Present State | Next State |  | Present Output |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | X=0 | X=1 | $\mathrm{X}=0$ | $\mathrm{X}=1$ |
| reset | A | B | C | 0 | 0 |
| 0 | B | D | E | 0 | 0 |
| 1 | C | KE | GD | 0 | 0 |
| 00 | D | H | XH | 0 | 0 |
| 01 | (E) | J | KH | 0 | 0 |
| 10 | F |  | MH |  | 0 |
| 11 | G | N 1 | $\cdots$ | 0 | 0 |
| 000 | (H) | A | A | 0 | 0 |
| -001 |  |  | A | 0 | 0 |
| 010 | (1) | A | A | 0 |  |
| -011 | $K$ | A | A | 0 | 0 |
| 100 | L | A | A |  | 1 |
| 101 | M | A | A | 0 | 0 |
| $\underline{110}$ | N | A | A | 0 | 0 |
| $\underline{111}$ | p | A | A | 0 | 0 |

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

- I.e., having the same next state and the same output under every possible inputTaking D as the representative for equivalent state $\operatorname{set}\{\mathrm{D}, \mathrm{G}\}$ and E for $\{E, F\}$, we can eliminate rows of $G$ and $F$


## Elimination of Redundant States

| Input Sequence | Present State | Next State |  | Present Output |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | X=0 | X=1 | $\mathrm{X}=0$ | X=1 |
| reset | A | B | C | 0 | 0 |
| 0 | B | D | E | 0 | 0 |
| 1 | C | KE | $G D$ | 0 | 0 |
| 00 | (D) | H | XH | 0 | 0 |
| 01 | (E) | J | K H | 0 | 0 |
| 10 | + | * | M H |  | 0 |
| 11 | $\checkmark$ | N ${ }^{\text {N }}$ | $\cdots$ | 0 | 0 |
| 000 | (H) | A | A | 0 | 0 |
| -001 |  | A | A | 0 | 0 |
| 010 | (1) | A | A | 0 |  |
| -011 | K | A | A | 0 | 0 |
| 100 | L | A | A | 0 | 1 |
| 101 | M | A | A | 0 | 0 |
| 1110 | N | A | A | 0 | 0 |
| 1111 | P | A | A | 0 | 0 |

$\square$ 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?)


## Elimination of Redundant States

$\square$ Reduced state table and state graph

| Present <br> 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 |



## Equivalent States

$\square$ Two states are equivalent if there is no way of telling them apart through observation of the circuit inputs and outputs
$\square$ Consider two sequential circuits $\mathrm{N}_{1}$ and $\mathrm{N}_{2}$ (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)
$\square$ we write $\underline{Z}_{1}=\lambda_{1}(\mathrm{p}, \underline{\mathrm{X}})$ and $\underline{Z}_{2}=\lambda_{2}(\mathrm{q}, \underline{\mathrm{X}})$ (because the output sequence is a function of the initial state and the input sequence)



## Equivalent States

$\square$ Definition 15.1
Let $\mathrm{N}_{1}$ and $\mathrm{N}_{2}$ be sequential circuits ( not necessarily different). Let $X$ represent a sequence of inputs of arbitrary length. Then state $p$ in $\mathrm{N}_{1}$ is equivalent to state q , denoted $\mathrm{p} \equiv \mathrm{q}$, in $\mathrm{N}_{2}$ iff $\lambda_{1}(\mathrm{p}, \underline{X})=$ $\lambda_{2}(q, \underline{X})$ for every possible input sequence $\underline{X}$.

- Symbol " $\equiv "$ here is different from XNOR
$\square$ Theorem 15.1 (proof in Appendix D)
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 ( $\equiv$ )
$\square$ 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
$\square$ Row matching is a special case of Theorem 15.1


## Equivalent States

$\square$ Example (Table 13.4)
Show no equivalent states

|  | Next State |  |  |  |  | Present Output $\left(\mathrm{Z}_{1} \mathrm{Z}_{2}\right)$ |  |  |  |
| :---: | ---: | :--- | :--- | :--- | ---: | :--- | :--- | :--- | :---: |
| State | $\mathrm{X}_{1} \mathrm{X}_{2}=00$ | 01 | 10 | 11 | $\mathrm{X}_{1} \mathrm{X}_{2}=00$ | 01 | 10 | 11 |  |
| $\mathrm{~S}_{0}$ | $\mathrm{~S}_{3}$ | $\mathrm{~S}_{2}$ | $\mathrm{~S}_{1}$ | $\mathrm{~S}_{0}$ | 00 | 10 | 11 | 01 |  |
| $\mathrm{~S}_{1}$ | $\mathrm{~S}_{0}$ | $\mathrm{~S}_{1}$ | $\mathrm{~S}_{2}$ | $\mathrm{~S}_{3}$ | 10 | 10 | 11 | 11 |  |
| $\mathrm{~S}_{2}$ | $\mathrm{~S}_{3}$ | $\mathrm{~S}_{0}$ | $\mathrm{~S}_{1}$ | $\mathrm{~S}_{1}$ | 00 | 10 | 11 | 01 |  |
| $\mathrm{~S}_{3}$ | $\mathrm{~S}_{2}$ | $\mathrm{~S}_{2}$ | $\mathrm{~S}_{1}$ | $\mathrm{~S}_{0}$ | 00 | 00 | 01 | 01 |  |

$\square$ From the outputs, we know only $\mathrm{S}_{0}$ and $\mathrm{S}_{2}$ can possibly be equivalent. Moreover,

$$
\mathrm{S}_{0} \equiv \mathrm{~S}_{2} \text { iff } \mathrm{S}_{3} \equiv \mathrm{~S}_{3}, \mathrm{~S}_{2} \equiv \mathrm{~S}_{0}, \mathrm{~S}_{1} \equiv \mathrm{~S}_{1} \text {, and } \mathrm{S}_{0} \equiv \mathrm{~S}_{1}
$$

But $\mathrm{S}_{0} \neq \mathrm{S}_{1}$ (because the outputs differ), so $\mathrm{S}_{0} \neq \mathrm{S}_{2}$

## Determination of State Equivalence (Not in Exam)

$\square$ 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 |
| :---: | :---: | :---: | :---: |
|  | $\mathrm{X}=0$ | 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 |



## Determination of State Equivalence

| $\square$ Example |  |  |  | bcd | $\begin{aligned} & \hline \text { d-f } \\ & c-h \end{aligned}$ | $\longleftarrow a \equiv b$ iff $d \equiv f$ and $c \equiv h$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Present State | Next State |  | Present Output |  | $X$ | $X$ | $\qquad$ | $b \neq c \text { si }$ | ince o | utputs differ |
|  |  |  | $\bar{a}-\alpha$ c-e |  | $\begin{aligned} & \text { a-f } \\ & \text { e-h } \end{aligned}$ | $X$ |  |  |  |
|  | $\mathrm{X}=0$ | 1 |  | e |  | $x$ | ree | $X$ |  |  |
| a | d | n |  | 0 | e |  |  | $a-d$ |  |  |  |
| c | e | h | 0 1 | f |  |  | e-f | $X$ | c-f |  |
| c |  | ${ }_{\text {d }}{ }^{\text {d }}$ | 1 | $f$ |  | $\Delta$ | b-d | $x$ | $a-b$ |  |
| e | a | e | 1 | 9 | $\begin{aligned} & \text { b-d } \\ & c-h \end{aligned}$ | b-f | $X$ | $\begin{aligned} & \text { a-b } \\ & e-b \end{aligned}$ |  |  |
| ¢ |  | b |  | h |  |  | c-e |  | a-g | c-f |
| h | c | g | 1 |  |  |  | d-g | , | a-g | b-g |

## Determination of State Equivalence

Example (cont'd)


## Determination of State Equivalence

$\square$ Example (cont'd)


## Equivalent Sequential Circuits (Not in Exam)

$\square$ 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 $\mathrm{N}_{1} \equiv \mathrm{~N}_{2}$, then $\mathrm{N}_{1}$ and $\mathrm{N}_{2}$ must have the same number of states


## Equivalent Sequential Circuits

$\square$ Example

|  |  | $\mathrm{N}_{1}$ |  |  |
| :---: | :---: | :---: | :---: | :---: |
|  | $\mathrm{X}=0$ | 1 | $\mathrm{X}=0$ | 1 |
| A | B | A | 0 | 0 |
| B | C | D | 0 | 1 |
| C | A | C | 0 | 1 |
| D | C | B | 0 | 0 |


|  |  | $\mathrm{N}_{2}$ |  |  |
| :---: | :---: | :---: | :---: | :---: |
|  | $\mathrm{X}=0$ | 1 | $\mathrm{X}=0$ | 1 |
| $\mathrm{~S}_{0}$ | $\mathrm{~S}_{3}$ | $\mathrm{~S}_{1}$ | 0 | 1 |
| $\mathrm{~S}_{1}$ | $\mathrm{~S}_{3}$ | $\mathrm{~S}_{0}$ | 0 | 0 |
| $\mathrm{~S}_{2}$ | $\mathrm{~S}_{0}$ | $\mathrm{~S}_{2}$ | 0 | 0 |
| $\mathrm{~S}_{3}$ | $\mathrm{~S}_{2}$ | $\mathrm{~S}_{3}$ | 0 | 1 |




## Equivalent Sequential Circuits

Example (cont'd)


## State Assignment (Not in Exam)

$\square$ 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 $\$ 15.7 \sim \$ 15.9$ )

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
