# Switching Circuits \＆ Logic Design 

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


Fall 2014

## §12 Registers and Counters



Drawing Hands
M．C．Escher， 1948

http：／／img106．imageshack．us／

## Outline

$\square$ Registers and register transfers
$\square$ Shift registers
$\square$ Design of binary counters
$\square$ Counters for other sequences
-Counter design using S-R and J-K flipflops
-Derivation of flip-flop input equations

## Registers and Register Transfers

$\square$ Several D flip-flops may be grouped together with a common clock to form a register

Registers and Register Transfers 4-Bit D Flip-Flop Register

## Using gated clock data out



With clock enable


Symbol


## Registers and Register Transfers <br> Data Transfer between Registers

Register A
$=$ FFs $A_{1}$ and $A_{2}$
Register B
$=$ FFs $B_{1}$ and $B_{2}$
Register Q
$=F F s Q_{1}$ and $Q_{2}$
$\left\{\begin{array}{l}\text { If } \mathrm{En}=1 \text { and } \operatorname{Load}=1, \mathrm{Q}=\mathrm{A} \\ \text { If } \mathrm{En}=0 \text { and Load=1, Q=B }\end{array}\right.$


Registers and Register Transfers 8-Bit Register with Tri-State Output


Different from clock enable

Registers and Register Transfers
Data Transfer Using a Tri-State Bus


## Registers and Register Transfers Parallel Adder with Accumulator

$\square$ Accumulator:
A register of FFs that can store one number and add a second number to it, leaving the result stored in it


## Registers and Register Transfers <br> Parallel Adder with Accumulator (cont'd)

- Implementation of adder cells (module-based design)
E.g., using Verilog

Adder cell without MUX


## Shift Registers

$\square$ A shift register is a register where binary data can be stored and shifted to the left/right when a shift signal is applied


## Shift Registers

## Right-Shift Register



I nitial state:

$\mathrm{Q}_{3} \mathrm{Q}_{2} \mathrm{Q}_{1} \mathrm{Q}_{0}=0101$
SI sequence:
1,1,0,1
Register states:
0101
1010
1101
0110
1011

## Shift Registers

Serial-In, Serial-Out Shift Register

## $\square$ Serial in

$\square$ Data is shifted into the first flip-flop one bit at a time
$\square$ Serial out

- Data can only be read out of the last flip-flop one bit at a time


## Shift Registers

8-Bit Serial-In, Serial-Out Shift Register
Block diagram


Timing diagram


## Shift Registers <br> Parallel-In, Parallel-Out Shift Register

## Block diagram



Application: conversion between parallel and serial data

## Shift Registers <br> Parallel-In, Parallel-Out Shift Register

## Logic diagram

(implementation using FFs and MUXes)


Shift register operation

| Inputs |  | Next State | Action | $\mathrm{Q}_{3}{ }^{+}=\mathrm{Sh} \cdot \mathrm{L} \cdot \mathrm{Q} \cdot \mathrm{Q}_{3}+\mathrm{Sh} \cdot \mathrm{L} \cdot \mathrm{D} \mathrm{D}_{3}+\mathrm{Sh} \cdot \mathrm{SI}$ |
| :---: | :---: | :---: | :---: | :---: |
| Sh (Shift) | L (Load) | $\mathrm{Q}_{3}{ }^{+} \mathrm{Q}_{2}{ }^{+} \mathrm{Q}_{1}{ }^{+} \mathrm{Q}_{0}{ }^{+}$ |  |  |
| 0 | 0 | $\mathrm{Q}_{3} \mathrm{Q}_{2} \mathrm{Q}_{1} \mathrm{Q}_{0}$ | no change | $\mathrm{Q}_{2}{ }^{+}=S h \cdot L \cdot \mathrm{Q}_{2}+S \mathrm{Sh} \cdot \mathrm{L} \cdot \mathrm{D}_{2}+\mathrm{Sh} \cdot \mathrm{Q}_{3}$ |
| 0 | 1 | $\mathrm{D}_{3} \mathrm{D}_{2} \mathrm{D}_{1} \mathrm{D}_{0}$ | load | $\mathrm{Q}_{1}^{+}=\mathrm{Sh} \cdot \mathrm{L} \cdot \mathrm{Q}_{1}+\mathrm{Sh} \cdot \mathrm{L} \cdot \mathrm{D}_{1}+\mathrm{Sh} \cdot \mathrm{O}_{3}$ |
| 1 | X | SI $\mathrm{Q}_{3} \mathrm{Q}_{2} \mathrm{Q}_{1}$ | right shift | $\mathrm{Q}_{0}{ }^{+}=S h^{\prime} \cdot \mathrm{L} \cdot \mathrm{Q}_{0}+S \mathrm{Sh}^{\prime} \cdot \mathrm{L} \cdot \mathrm{D}_{0}+\mathrm{Sh} \cdot \mathrm{Q}_{1}$ |

## Shift Registers <br> Parallel-In, Parallel-Out Shift Register

Timing diagram
(assume $\mathrm{D}_{3} \mathrm{D}_{2} \mathrm{D}_{1} \mathrm{D}_{0}=1011$ initially and 0000 afterwards)


## Shift Registers

Shift Register with Inverted Feedback
$\square$ A circuit that cycles through a fixed sequence of states is called a counter
$\square$ A shift register with inverted feedback is often called a Johnson counter


Flip-flop connections


State graph

## Design of Binary Counters

$\square$ We focus on the synchronous counter, where flipflops are synchronized by a common clock pulse

## State table

| Present State |  |  |  | Next State |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| C | B | A | $\mathrm{C}^{+}$ | $\mathrm{B}^{+}$ | $\mathrm{A}^{+}$ |  |
| 0 | 0 | 0 | 0 | 0 | 1 |  |
| 0 | 0 | 1 | 0 | 1 | 0 |  |
| 0 | 1 | 0 | 0 | 1 | 1 |  |
| 0 | 1 | 1 | 1 | 0 | 0 |  |
| 1 | 0 | 0 | 1 | 0 | 1 |  |
| 1 | 0 | 1 | 1 | 1 | 0 |  |
| 1 | 1 | 0 | 1 | 1 | 1 |  |
| 1 | 1 | 1 | 0 | 0 | 0 |  |

## Design of Binary Counters <br> D Flip-Flop Implementation

$\square$ State table

| Present State |  |  | Next State |  |  | Flip-Flop Inputs |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| C | B | A | $\mathrm{C}^{+}$ | $\mathrm{B}^{+}$ | $\mathrm{A}^{+}$ | $\mathrm{D}_{\mathrm{c}}$ | $\mathrm{D}_{\mathrm{B}}$ | $\mathrm{D}_{\mathrm{A}}$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

FF inputs are next-state functions in terms of $A, B, C$

## Design of Binary Counters <br> D Flip-Flop Implementation

$\square$ Karnaugh maps for D flip-flops


| $)^{C} 0 \quad 1$ |  |  |
| :---: | :---: | :---: |
| BA | 0 | 0 |
| 01 | 1 | 1 |
| 11 | 0 | 0 |
| 10 | 1 | 1 |


Binary counter with D flip-flops


## Design of Binary Counters <br> T Flip-Flop Implementation

-State table

| Present State |  |  | Next State |  |  | Flip-Flop Inputs |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| C | B | A | $\mathrm{C}^{+}$ | $\mathrm{B}^{+}$ | $\mathrm{A}^{+}$ | $\mathrm{T}_{\mathrm{C}}$ | $\mathrm{T}_{\mathrm{B}}$ | $\mathrm{T}_{\mathrm{A}}$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

FF inputs are next-state functions in terms of $A, B, C$

## Design of Binary Counters T Flip-Flop Implementation

- Karnaugh maps for D flip-flops


| $)^{C} 01$ |  |  |
| :---: | :---: | :---: |
| BA | 0 | 0 |
| 01 | 1 | 1 |
| 11 | 1 | 1 |
| 10 | 0 | 0 |


$\square$ Binary counter with D flip-flops


## Design of Binary Counters

 Up-Down Counter$\square$ State graph and table for up-down counter

$\mathrm{U}=1, \mathrm{D}=0$ count up
$U=0, D=1$ count down

| C B A | $\mathrm{C}^{+} \mathrm{B}^{+} \mathrm{A}^{+}$ |  |
| :---: | :---: | :---: |
|  | U | D |
| 000 | 001 | 111 |
| 001 | 010 | 000 |
| 010 | 011 | 001 |
| 011 | 100 | 010 |
| 100 | 101 | 011 |
| 101 | 110 | 100 |
| 110 | 111 | 101 |
| 111 | 000 | 110 |

## Design of Binary Counters Up-Down Counter

$\square$ The up-down counter can be implemented using
D FFs and gates through the logic equations
$D_{A}=A^{+}=A \oplus(U+D)$
$\mathrm{D}_{\mathrm{B}}=\mathrm{B}^{+}=\mathrm{B} \oplus\left(\mathrm{UA}+\mathrm{DA}^{\prime}\right)$
$\mathrm{D}_{\mathrm{C}}=\mathrm{C}^{+}=\mathrm{C} \oplus\left(\mathrm{UBA}+\mathrm{DB}^{\prime} \mathrm{A}^{\prime}\right)$

- When $\mathrm{U}=1, \mathrm{D}=0$, they reduce to binary up counter
- When $U=0, D=1$, they reduce to
$D_{A}=A^{+}=A \oplus 1=A^{\prime}$
$\mathrm{D}_{\mathrm{B}}=\mathrm{B}^{+}=\mathrm{B} \oplus \mathrm{A}^{\prime}$
$\mathrm{D}_{\mathrm{C}}=\mathrm{C}^{+}=\mathrm{C} \oplus \mathrm{B}^{\prime} \mathrm{A}^{\prime}$


## Design of Binary Counters Up-Down Counter

$$
\begin{aligned}
& D_{A}=A^{+}=A \oplus(U+D) \\
& D_{B}=B^{+}=B \oplus\left(U A+D A^{\prime}\right) \\
& D_{C}=C^{+}=C \oplus\left(U B A+D B^{\prime} A^{\prime}\right)
\end{aligned}
$$



# Design of Binary Counters Loadable Counter with Count Enable 



$$
\begin{aligned}
& \mathrm{A}^{+}=\mathrm{D}_{\mathrm{A}}=\left(\mathrm{Ld} \cdot \mathrm{~A}+\mathrm{Ld} \cdot \mathrm{D}_{\text {Ain }}\right) \oplus \mathrm{Ld} \cdot \mathrm{Ct} \\
& \mathrm{~B}^{+}=\mathrm{D}_{\mathrm{B}}=\left(\mathrm{Ld} \cdot \mathrm{CL}+\mathrm{Ld} \cdot \mathrm{D}_{\mathrm{Bin}}\right) \oplus \mathrm{Ldd}^{\prime} \cdot \mathrm{Ct} \cdot \mathrm{~A} \\
& \mathrm{C}^{+}=\mathrm{D}_{\mathrm{C}}=\left(\mathrm{Ld} \cdot \mathrm{C}+\mathrm{Ld} \cdot \mathrm{D}_{\mathrm{cin}}\right) \oplus \mathrm{Ld}^{\prime} \cdot \mathrm{Ct} \cdot \mathrm{~B} \cdot \mathrm{~A}
\end{aligned}
$$

Note that CIrN does not appear in these equations. Why not?

## Design of Binary Counters <br> Loadable Counter with Count Enable

$$
\begin{aligned}
& \mathrm{A}^{+}=\mathrm{D}_{\mathrm{A}}=\left(\mathrm{Ld} \cdot \mathrm{~A}+\mathrm{Ad} \cdot \mathrm{D}_{\text {Ain }}\right) \oplus \mathrm{Ld}^{\prime} \cdot \mathrm{Ct} \\
& \mathrm{~B}^{+}=\mathrm{D}_{\mathrm{B}}=\left(\mathrm{Ld} \cdot \mathrm{~B}+\mathrm{Ld} \cdot \mathrm{D}_{\text {Bin }}\right) \oplus \mathrm{Ld}^{\prime} \cdot \mathrm{Ct} \cdot \mathrm{~A} \\
& \mathrm{C}^{+}=\mathrm{D}_{\mathrm{C}}=\left(\mathrm{Ld} \cdot \mathrm{C}+\mathrm{Ld} \cdot \mathrm{D}_{\text {Cin }}\right) \oplus \mathrm{Ld}^{\prime} \cdot \mathrm{Ct} \cdot \mathrm{~B} \cdot \mathrm{~A}
\end{aligned}
$$



## Counters for Other Sequences

State graph


State table

| $C$ | $B$ | $A$ | $C^{+}$ | $B^{+}$ | $A^{+}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | - | - | - |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | - | - | - |
| 1 | 1 | 0 | - | - | - |
| 1 | 1 | 1 | 0 | 1 | 0 |

## Counters for Other Sequences <br> Counter Design Using T Flip-Flops

| State table |  | T FF input |  |  |
| :---: | :---: | :---: | :---: | :---: |
| C B A | $C^{+} \mathrm{B}^{+} \mathrm{A}^{+}$ | $\mathrm{T}_{\mathrm{C}} \mathrm{T}_{\mathrm{B}} \mathrm{T}_{\mathrm{A}}$ |  |  |
| 000 | 100 | 100 | $\mathrm{QQ}^{+}$ | T |
| 001 | - - - | - - - | 00 | 0 |
| 010 | $\begin{array}{lll}0 & 1\end{array}$ | 001 | 01 | 1 |
| $0 \begin{array}{lll}0 & 1 & 1\end{array}$ | 000 | 0111 | 10 | 1 |
| 100 | 111 | 011 | 11 | 0 |
| 101 | - - - | - - - |  |  |
| 110 | - - - | - - - | = |  |
| $1 \begin{array}{lll}1 & 1\end{array}$ | 010 | 101 |  |  |

## Counters for Other Sequences Counter Design Using T Flip-Flops



## Counters for Other Sequences <br> Counter Design Using T Flip-Flops

-For T flip-flop implementation
■ If the $\mathrm{Q}^{+}$map has a don't care in some square, the $T_{Q}$ map will have a don't care in the corresponding square
Divide the $\mathrm{Q}^{+}$map into two halves corresponding to $\mathrm{Q}=0$ and $\mathrm{Q}=1$, and transform each half of the map $\square W h e n e v e r \mathrm{Q}=0, \mathrm{~T}=\mathrm{Q}^{+}$

- Copy the half for which $\mathrm{Q}=0$
$\square W h e n e v e r \mathrm{Q}=1, \mathrm{~T}=\left(\mathrm{Q}^{+}\right)^{\prime}$
- Complement the half for which $\mathrm{Q}=1$


## Counters for Other Sequences Counter Design Using T Flip-Flops

$\square$ T flip-flop realization


## Counters for Other Sequences Counter Design Using T Flip-Flops

$\square$ Timing diagram

- Negative-edge triggered



## Counters for Other Sequences Counter Design Using T Flip-Flops

$\square$ In the process of completing the circuit design, the transitions of the original don't care sates will be specified

- Don't care states need to be checked to make sure they eventually lead into the main counting sequence unless a power-up reset is provided
$\square$ When the power in a circuit is first turned on, the initial states of the flip-flops may be unpredictable (can be in an arbitrary state)


What are the transitions for states 001, 101, 110 by the realization in Slide 33?

$$
\begin{aligned}
& T_{C}=C^{\prime} B^{\prime}+C B \\
& T_{B}=C^{\prime} A+C B^{\prime} \\
& T_{A}=C+B
\end{aligned}
$$

# Counters for Other Sequences Counter Design Using T Flip-Flops 

$\square$ Example (cont'd)
If FFs are powered up at $(C B A)=001$, then $T_{C}=T_{B}=1$ and $T_{A}=0$ leads to next state 111


$$
\begin{aligned}
& \mathrm{T}_{\mathrm{C}}=\mathrm{C}^{\prime} \mathrm{B}^{\prime}+C B \\
& \mathrm{~T}_{\mathrm{B}}=\mathrm{C}^{\prime} \mathrm{A}+\mathrm{CB} \\
& \mathrm{~T}_{\mathrm{A}}=\mathrm{C}+\mathrm{B}
\end{aligned}
$$

# Counters for Other Sequences <br> Counter Design Using T Flip-Flops 

$\square$ Example
Without power-up reset, the following Johnson counter may be incorrect when powered up at states 010 and 101


Flip-flop connections


State graph

## Counters for Other Sequences Counter Design Using D Flip-Flops

## State table D FF input

State graph


## Counters for Other Sequences <br> Counter Design Using D Flip-Flops

$\mathrm{D}_{\mathrm{C}}=\mathrm{C}^{+}=\mathrm{B}^{\prime}$
$D_{B}=B^{+}=C+B A^{\prime}$
$D_{A}=A^{+}=C A^{\prime}+B A^{\prime}=A^{\prime}(C+B)$


## Counter Design Using S-R and J-K FFs Counter Design Using S-R Flip-Flops

םS-R flip-flop inputs

| $S \mathrm{R} Q$ | $\mathrm{Q}^{+}$ |
| :---: | :---: |
| 000 | 0 |
| 0001 | 1 |
| 010 | 0 |
| $\begin{array}{llll}0 & 1 & 1\end{array}$ | 0 |
| 100 | 1 |
| $\begin{array}{lll}1 & 0 & 1\end{array}$ | 1 |
| $1 \begin{array}{lll}1 & 1 & 0\end{array}$ |  |
| $1 \begin{array}{lll}1 & 1\end{array}$ |  |



## Counter Design Using S-R and J-K FFs Counter Design Using S-R Flip-Flops

State graph


|  |  |  |  | C |  | B |  | A |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| C | B A | $\mathrm{C}^{+}$ | $\mathrm{B}^{+} \mathrm{A}^{+}$ | $\mathrm{S}_{\mathrm{C}}$ | $\mathrm{R}_{\mathrm{C}}$ | $\mathrm{S}_{B}$ | $\mathrm{R}_{\mathrm{B}}$ | $\mathrm{S}_{\text {A }}$ | $\mathrm{R}_{\mathrm{A}}$ |
| 0 | 00 | 1 | 00 | 1 | 0 | 0 | X | 0 | X |
| 0 | 01 | - | - | X | X | X | X | X | X |
| 0 | 10 | 0 | 11 | 0 | X | X | 0 | 1 | 0 |
| 0 | 11 | 0 | 00 | 0 | X | 0 | 1 | 0 | 1 |
| 1 | 00 | 1 | 11 | X | 0 | 1 | 0 | 1 | 0 |
| 1 | 01 | - | - - | X | X | X | X | X | X |
| 1 | 10 | - | - - | X | X | X | X | X | X |
| 1 | 11 | 0 | 10 | 0 | 1 | X | 0 | 0 | 1 |


| Q | $\mathrm{Q}^{+}$ | S | R |
| :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | 0 |

## Counter Design Using S-R and J-K FFs Counter Design Using S-R Flip-Flops

Next-state maps

| ${ }^{C} 0$ |  | 1 |
| :---: | :---: | :---: |
| BA |  |  |
| 00 | 1 | 1 |
| 01 | X | X |
| 11 | 0 | 0 |
| 10 | 0 | X |
|  |  |  |




+

$S_{A}$
$R_{A}$
$R_{A}=A$
$S_{A}=C^{\prime}+A^{\prime}$
$=A^{\prime}(C+B)$
$R_{B}=C^{\prime} A$
$S_{B}$

## Counter Design Using S-R and J-K FFs Counter Design Using S-R Flip-Flops

$\square S$-R flip-flop realization

(feedback lines omitted)

## Counter Design Using S-R and J-K FFs Counter Design Using J-K Flip-Flops

ㅁ.K flip-flop inputs


## Counter Design Using S-R and J-K FFs Counter Design Using J-K Flip-Flops

## State graph



|  |  |  | $\overbrace{}^{\text {C }}$ |  |  |  |  | A |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| C | B A | $\mathrm{C}^{+}$ | $\mathrm{B}^{+} \mathrm{A}^{+}$ | $J_{C}$ | $\mathrm{K}_{\mathrm{c}}$ | $J_{B}$ | $\mathrm{K}_{\mathrm{B}}$ | $\mathrm{J}_{\mathrm{A}}$ | $\mathrm{K}_{\mathrm{A}}$ |
| 0 | 00 | 1 | 00 | 1 | X | 0 | X | 0 | X |
| 0 | 01 | - | - | X | X | X | X | X | X |
| 0 | 10 | 0 | 11 | 0 | X | X | 0 | 1 | X |
| 0 | 11 | 0 | 00 | 0 | X | X | 1 | X | 1 |
| 1 | 00 | 1 | 11 | X | 0 | 1 | X | 1 | X |
| 1 | 01 | - | - - | X | X | X | X | X | X |
| 1 | 10 | - | - | X | X | X | X | X | X |
| 1 | 11 | 0 | 10 | X | 1 | X | 0 | X | 1 |


| Q | $\mathrm{Q}^{+}$ | J | K |
| :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

## Counter Design Using S-R and J-K FFs Counter Design Using J-K Flip-Flops

Next-state maps

## Counter Design Using S-R and J-K FFs Counter Design Using J-K Flip-Flops

## ㄱJ-K flip-flop realization


(feedback lines omitted)

## Derivation of Flip-Flop Input Equations

$\square$ Determination of flip-flop input equations from next-state equations using Karnaugh maps

| Type of Flip-Flop | Input | $\mathrm{Q}=0$ |  | $\mathrm{Q}=1$ |  | Rules for Forming Input Map From Next-State Map* |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | $\mathrm{Q}^{+}=0$ | $\mathrm{Q}^{+}=1$ | $\mathrm{Q}^{+}=0$ | $\mathrm{Q}^{+}=1$ | $\begin{gathered} \mathrm{Q}=0 \text { Half of } \\ \text { Map } \end{gathered}$ | $\begin{gathered} \mathrm{Q}=1 \text { Half of } \\ \text { Map } \end{gathered}$ |
| Delay | D | 0 | 1 | 0 | 1 | No change | No change |
| Trigger | T | 0 | 1 | 1 | 0 | No change | Complement |
| Set-Reset | S | 0 | 1 | 0 | X | No change | Replace 1's with X's** |
|  | R | X | 0 | 1 | 0 | Replace 0's with X's** | Complement |
| J-K | J | 0 | 1 | X | X | No change | Fill in with X 's |
|  | K | X | X | 1 | 0 | Fill in with X's | Complement |

[^0]** Fill in the remaining squares with 0 's

## Derivation of Flip-Flop Input Equations Flip-Flop Input Tables

| Q | $\mathrm{Q}^{+}$ | D |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |


| Q | $\mathrm{Q}^{+}$ | T |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |


| Q | $\mathrm{Q}^{+}$ | S | R |
| :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | 0 |


| Q | $\mathrm{Q}^{+}$ | J | K |
| :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

Realization using J-K flip flops usually yields lower cost implementation

## Derivation of Flip-Flop Input Equations Example


Next-state map

$D=Q^{\prime} A^{\prime} B+Q B^{\prime}+A B^{\prime}$
D input map

$T=A^{\prime} B+A B^{\prime}+Q B$
T input map


## Derivation of Flip-Flop Input Equations Example (1/3)

$\square$ Derivation of $\mathrm{Q}_{1}$ (T flip-flop) input equation using 4 -variable maps

|  |  | 01 | 11 | 10 | $B C Q^{2}$ |  | 01 | 11 | 10 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 00 | 0 | 1 | 0 | 1 | 00 | 0 | 1 | 1 | 0 |
| 01 | x | 1 | 1 | 0 | 01 | X | 1 | 0 | 1 |
| 11 | 1 | X | X | 1 | 11 | 1 | X | X | 0 |
| 10 | 0 | 0 | 0 | X | 10 | 0 | 0 | 1 | X |
| $\begin{gathered} \mathrm{Q}_{1}=0 \\ \text { half } \end{gathered}$ |  |  |  |  |  | T |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

## Derivation of Flip-Flop Input Equations Example (2/3)

$\square$ Derivation of $\mathrm{Q}_{2}$ (S-R flip-flop) input equations using 4 -variable maps

| $\mathrm{CQ}_{2}{ }^{\mathrm{AB}}$ |  |  |  |  | $\mathrm{CQ}_{2}{ }^{\mathrm{AB}}$ |  |  | 11 | 10 | $\mathrm{CQ}_{2}{ }^{\mathrm{AB}}$ |  | 01 | 11 | 10 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| -half $\longrightarrow 00$ | 1 | X | 1 | 0 | 00 | 0 | X | 0 | X | 00 | 1 | X | 1 | 0 |
| $\mathrm{Q}_{2}=1 \quad 01$ | 0 | 0 | X | 1 | 01 | 1 | 1 | X | 0 | 01 | 0 | 0 | X | X |
| half 11 | 1 | 0 | x | 1 | 11 | 0 | 1 | X | 0 | 11 | X | 0 | X | X |
| $\rightarrow 10$ | X | 0 | 0 | 1 | 10 | X | X | X | 0 | 10 | X | 0 | 0 | 1 |
| $\mathrm{Q}_{2}{ }^{+}$ |  |  |  |  | $\mathrm{R}_{2}$ |  |  |  |  | $\mathrm{S}_{2}$ |  |  |  |  |

## Derivation of Flip-Flop Input Equations Example (3/3)

$\square$ Derivation of $\mathrm{Q}_{3}$ (J-K flip-flop) input equations using 4 -variable maps

| $\mathrm{Q}_{3} C^{A E}$ |  |  |  |  |  |  |  |  | 10 |  | 00 | 01 | 11 | 10 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\bigcirc 0$ | 0 | 0 | 1 | X | 00 | 0 | 0 | 1 | X | 0 | X | X | X | X |
| 01 | 0 | 1 | X | 1 | 01 | 0 | 1 | X | 1 | 1 | X | X | X | X |
| ¢11 | X | x | 0 | 0 | 11 | X | x | X | X | 11 | x | X | 1 | 1 |
| 10 | 1 | 1 | 1 | 0 | 10 | X | X | X | X | 10 | 0 | 0 | 0 | 1 |
|  | $\mathrm{Q}_{3}{ }^{+}$ |  |  |  |  | $J_{3}=A+B C$ |  |  |  |  | ${ }_{3}$ |  |  |  |

## Derivation of Flip-Flop Input Equations Summary

-To implement a counter of N states, we need at least $\log _{2} \mathrm{~N}$ flip-flops
-Derive input equations depending on the target implementation using D, T, SR, or JK flip-flops
$\square$ Pay attention to don't care states for power-up conditions. Sometimes reset may be needed


[^0]:    *Always copy $X$ 's from the next-state map onto the input maps first

