Chapter 6

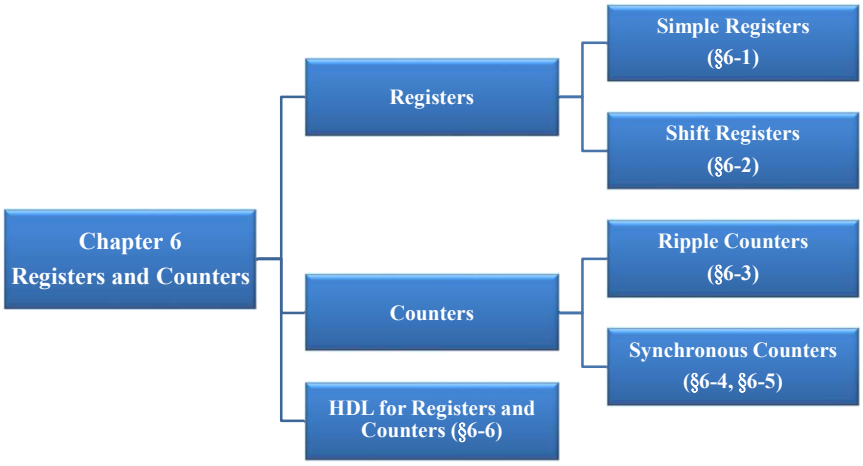
Registers and Counters (Sync. Seq. Ckts)

J.J. Shann

1

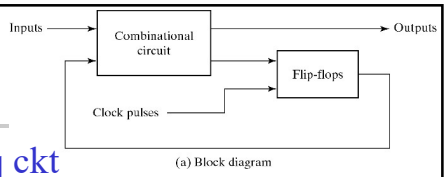
Chapter Overview

2



3

Overview



(a) Block diagram

- Clocked seq ckt: sync seq ckt
  - consists of a group of flip-flops and combinational gates connected to form a feedback path.

    Flip-flops + Combinational gates

    (essential) (optional)
- Register:
  - consists of a group of flip-flops capable of storing binary information and gates that determine how the information is transferred into the register.
- Counter:
  - is essentially a register that goes through a predetermined sequence of states.

J.J. Shann 6-4

4

Exercises in Textbook (6th ed.)

Sections	Exercises	Typical Ones
§6-1	6.1, 6.2	6.2
§6-2	6.3~6.10	6.6, 6.10
§6-3	6.11~6.16	6.12, 6.15*
§6-4	6.17~ 6.23	6.17*, 6.22
§6-5	6.24~6.30	6.24*, 6.29
HDL	6.31~6.59	

* : Answers to problems appear at the end of the text.

J.J. Shann 6-5

5

6-1

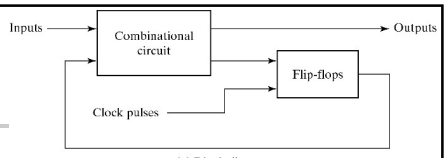
Registers

J.J. Shann

6

Registers

- Simple Register
- Register with Parallel Load
- Shift Register (§6-2)



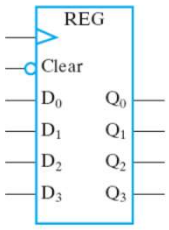
(a) Block diagram

J.J. Shann 6-7

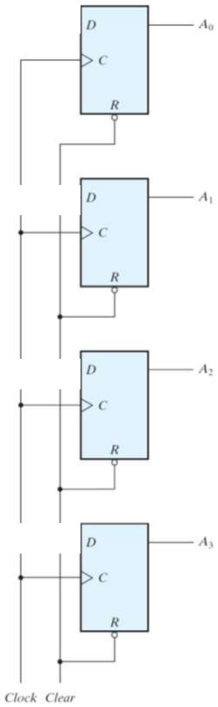
7

A. Simplest Register

- Simplest register:
  - consists of only flip-flops w/o any gates.
  - E.g.: a 4-bit register with asynchronous clear input



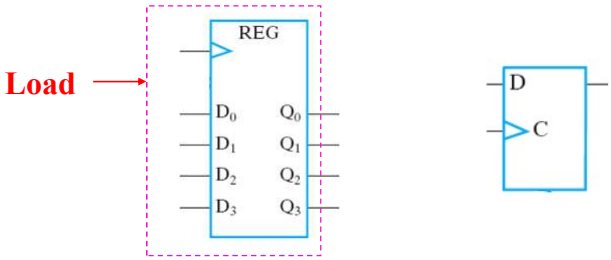
(b) Symbol



ın 6-8

8

B. Register w/ Parallel Load

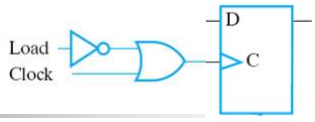


- Approach 1:
  Load control input through the *C* inputs of the f-fs
  → *clock gating*
- Approach 2:
  Load control input through the *D* inputs of the f-fs

J.J. Shann 6-9

9

Approach 1

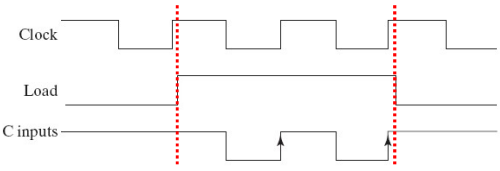


- Approach 1: Load control input through the *C* inputs of the f-fs → *clock gating*
  - prevent the clock from reaching the clock input to the ckt if the contents of the reg are to be left unchanged

Load = 1, C inputs = Clock
Load = 0, C inputs = 1

$$C\ inputs = Load \cdot Clock + \overline{Load} \cdot 1$$
$$= \overline{Load} + Clock$$

(c) Load control input

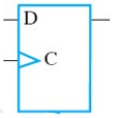


(d) Timing diagram

* Inserting gates in the clock pulse path produces different propagation delays b/t clock and the inputs of f-fs w/ and w/o clock gating. ⇒ clock skew

10

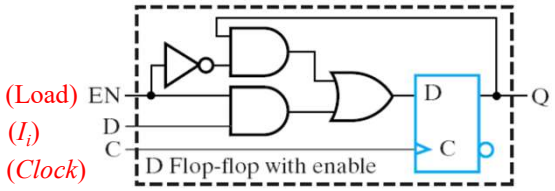
Approach 2



- Approach 2: Load control input through the *D* inputs of the f-fs

Load = 1, $D_i = I_i$
Load = 0, $D_i = Q_i$

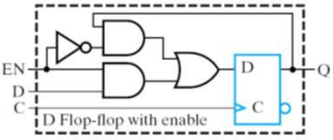
$$D_i = Load \cdot I_i + \overline{Load} \cdot Q_i$$



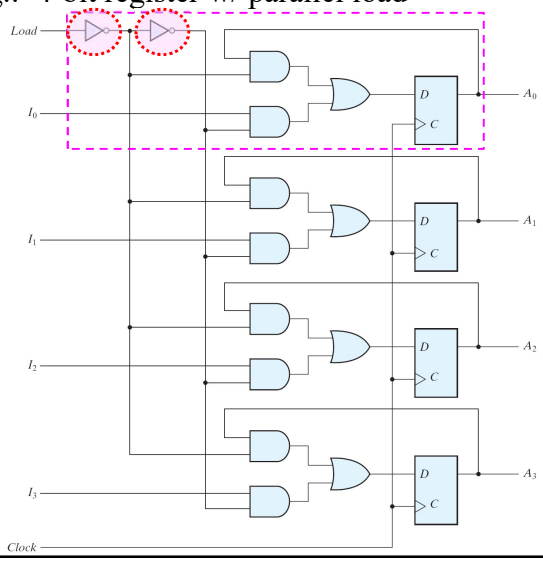
– E.g.: 4-bit register w/ parallel load

J.J. Shann 6-11

■11



– E.g.: 4-bit register w/ parallel load



J.J. Shann 6-12

■12

6-2

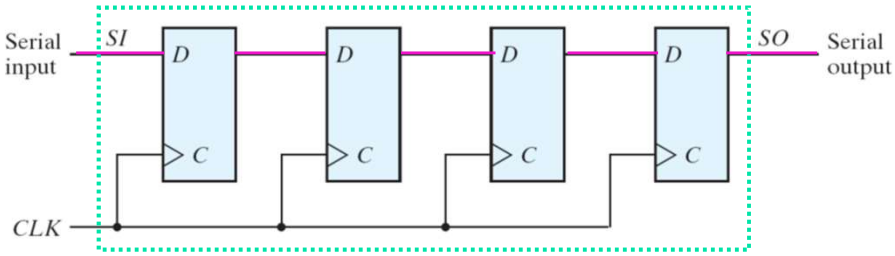
Shift Registers

J.J. Shann

13

Shift Registers

- Shift register:
  - a register capable of shifting its binary information in one or both direction
- Simplest shift register: unidirectional



J.J. Shann 6-14

14

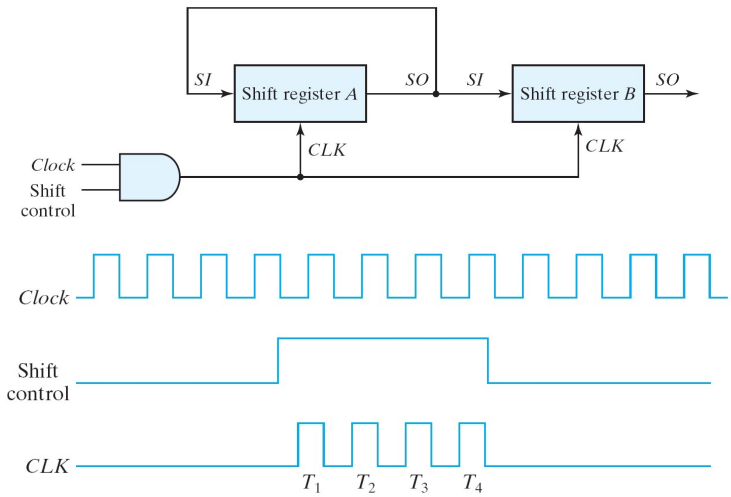
A. Serial Transfer

- Serial mode of a digital system:
  - Information is transferred and manipulated one bit at a time.
- Serial transfer vs. Parallel transfer
  - Serial transfer:
    - Information is transferred one bit at a time by shifting the bits out of the source register into the destination register.
  - Parallel transfer:
    - All the bits of the register are transferred at the same time.

J.J. Shann 6-15

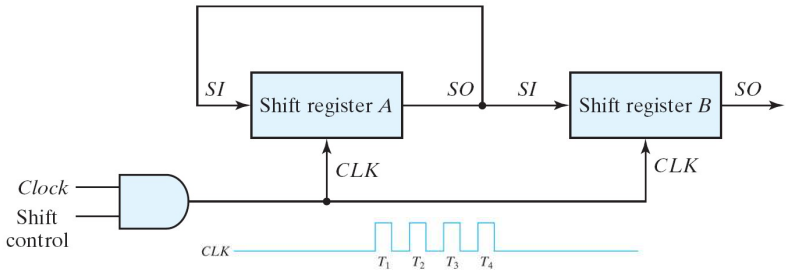
15

- Example: Serial transfer from reg A to reg B



6-16

16



Serial-Transfer Example

Timing Pulse	Shift Register *A*				Shift Register *B*			
Initial value	1	0	1	1	0	0	1	0
After T_1	1	1	0	1	1	0	0	1
After T_2	1	1	1	0	1	1	0	0
After T_3	0	1	1	1	0	1	1	0
After T_4	1	0	1	1	1	0	1	1

6-17

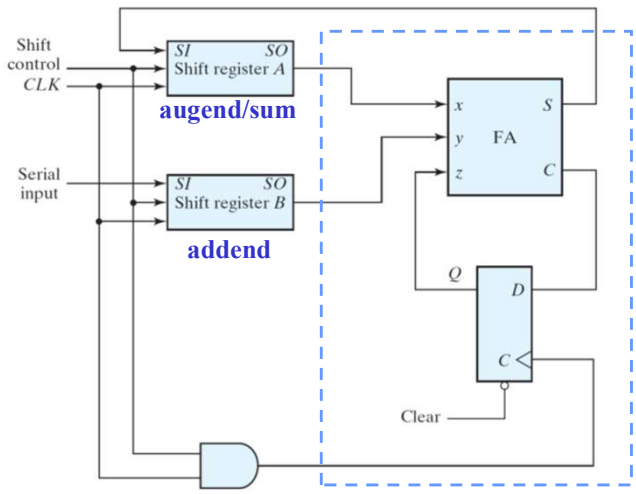
■17

B. Serial Addition

* Recall: Parallel Adder

$$\begin{array}{r} A_3\ A_2\ A_1\ A_0 \\ +\ B_3\ B_2\ B_1\ B_0 \\ \hline \end{array}$$

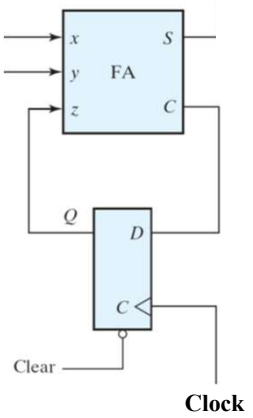
- Serial adder: D flip-flop



J.J. Shann 6-18

■18

— State table:



State Table for Serial Adder

Present State	Inputs		Next State	Output
Q	x	y	Q^+	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$D = xy + (x \oplus y)Q$$

$$S = x \oplus y \oplus Q$$

J.J. Shann 6-19

19

— JK flip-flop input equations and output equation:

State Table for Serial Adder

Present State	Inputs		Next State	Output	Flip-Flop Inputs	
Q	x	y	Q^+	S	J_Q	K_Q
0	0	0	0	0	0	X
0	0	1	0	1	0	X
0	1	0	0	1	0	X
0	1	1	1	0	1	X
1	0	0	0	1	X	1
1	0	1	1	0	X	0
1	1	0	1	0	X	0
1	1	1	1	1	X	0

$$J_Q = x\,y$$

$$K_Q = x'\,y' = (x + y)'$$

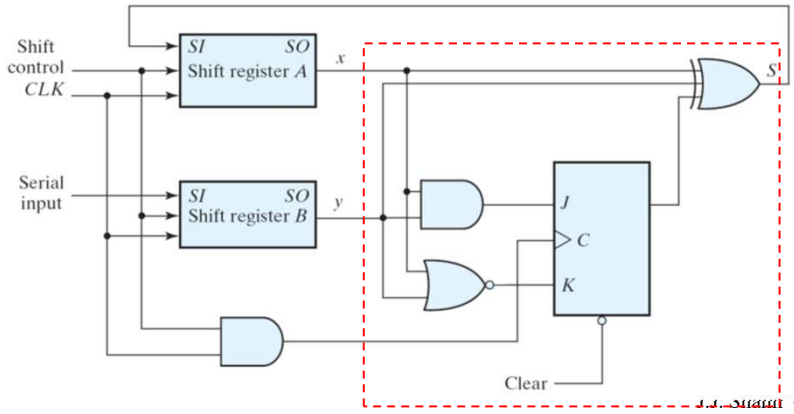
$$S = x \oplus y \oplus Q$$

J.J. Shann 6-20

20

— Circuit diagram:

$$J_Q = x\,y$$
$$K_Q = x'\,y' = (x+y)'$$
$$S = x \oplus y \oplus Q$$



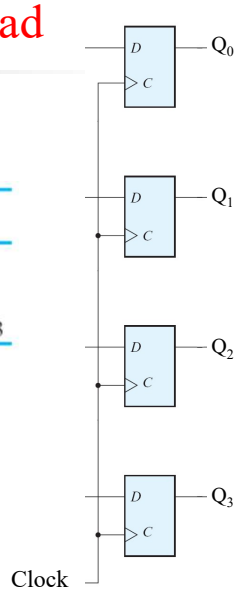
J.J. Shann 6-21

C. Shift Register w/ Parallel Load

- Shift register w/ parallel load:
  - Functional table:

Shift	Load	Operation
0	0	No change
0	1	Load parallel data
1	X	Shift down from Q_0 to Q_3

$$\boxed{\begin{aligned} D_i = {} & \overline{Shift} \cdot \overline{Load} \cdot Q_i \\ & + \overline{Shift} \cdot Load \cdot I_i \\ & + Shift \cdot Q_{i-1} \end{aligned}}$$

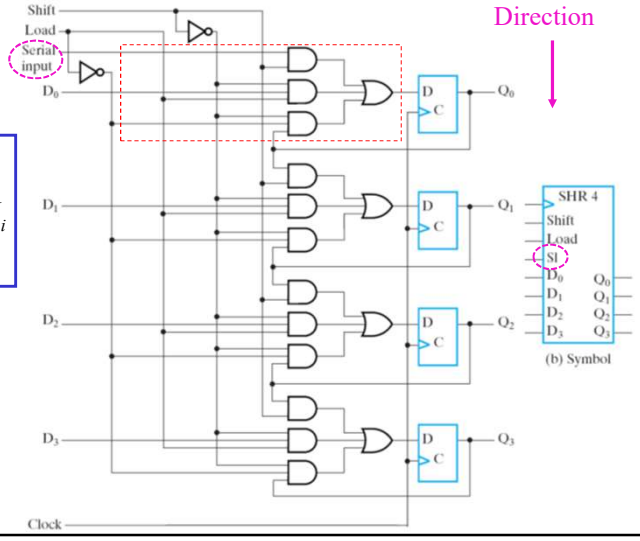


J.J. Shann 6-22

Shift	Load	Operation
0	0	No change
0	1	Load parallel data
1	X	Shift down from Q_0 to Q_3

– E.g.: 4-bit shift register w/ parallel load

$$D_i = \overline{Shift} \cdot \overline{Load} \cdot Q_i + \overline{Shift} \cdot Load \cdot I_i + Shift \cdot Q_{i-1}$$



D. Universal Shift Register

- Unidirectional shift register
- Bidirectional shift register

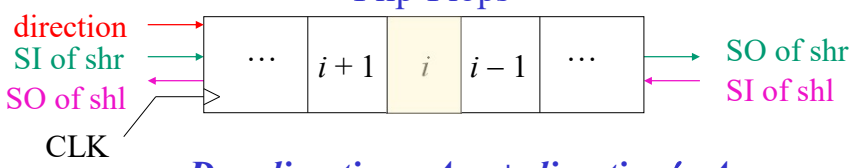
  direction = 0 … shl

  = 1 … shr

$D_0 = \ldots\ldots$

$D_{n-1} = \ldots\ldots$

Flip-Flops



$$D_i = direction \bullet A_{i+1} + direction' \bullet A_{i-1}$$

- Universal shift register:
  – has both direction shifts & parallel load capabilities

J.J. Shann 6-24

- Capability of a universal shift register:
  1. A *clock* input to synchronize the operations.
  2. A *clear* control to clear the register to 0.
  3. A *shift-right* control to enable the shift right operation and the *serial input* and *output* lines associated w/ the shift right.
  4. A *shift-left* control to enable the shift left operation and the *serial input* and *output* lines associated w/ the shift left.
  5. A *parallel-load* control to enable a parallel transfer and the *n parallel input* lines associated w/ the parallel transfer.
  6. *n parallel output* lines.
  7. A control state that leaves the information in the register unchanged in the presence of the clock.

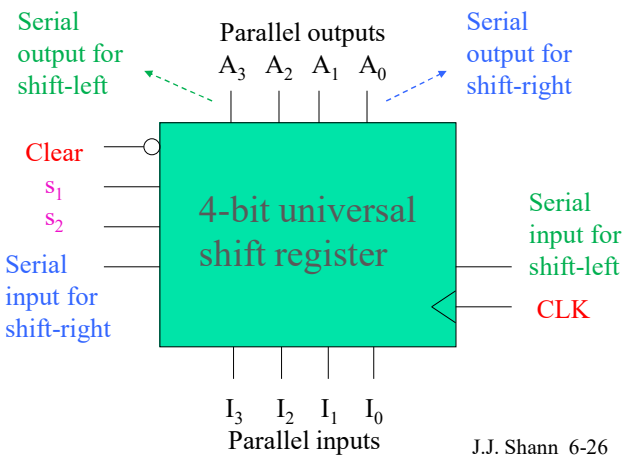
J.J. Shann 6-25

25

- Example: 4-bit universal shift reg (w/ direct clear)

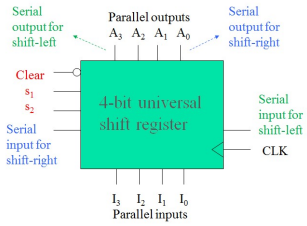
Block diagram:

1. clock
2. clear
3. *shift-right*
4. *shift-left*
5. *parallel-load* (parallel input)
6. parallel output
7. *unchanged*



J.J. Shann 6-26

26



Function table:

direct input
Active LOW

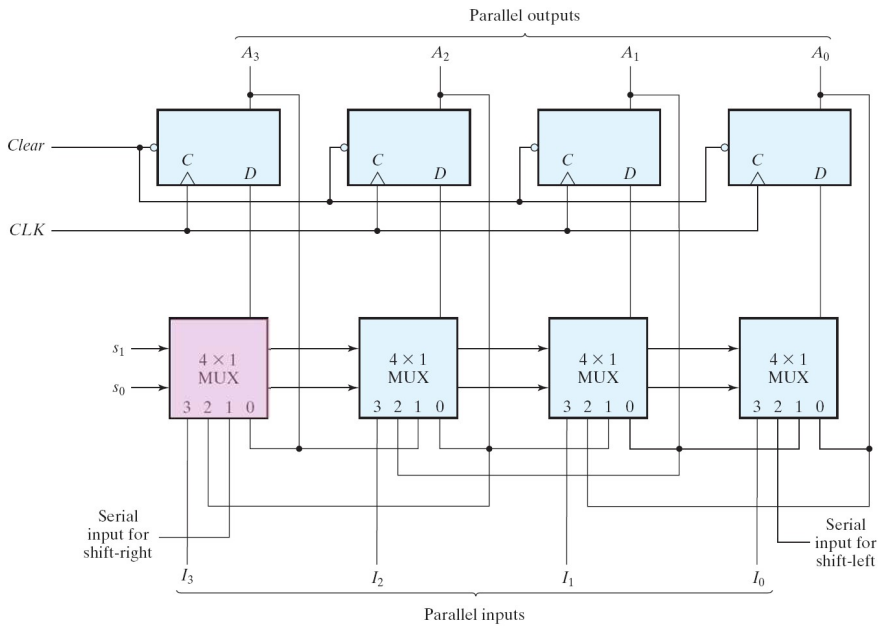
Clear	CLK	s_1	s_0	A_3^+	A_2^+	A_1^+	A_0^+	(Register Op)
0	×	×	×	0	0	0	0	Clear
1	↑	0	0	A_3	A_2	A_1	A_0	No change
1	↑	0	1	SI_{shr}	A_3	A_2	A_1	Shift right
1	↑	1	0	A_2	A_1	A_0	SI_{shl}	Shift left
1	↑	1	1	I_3	I_2	I_1	I_0	Parallel load

$$A_3^+ = s_1's_0'A_3 + s_1's_0\,SI_{shr} + s_1s_0'A_2 + s_1s_0\,I_3 \Rightarrow D_3$$

$\vdots$

6-27

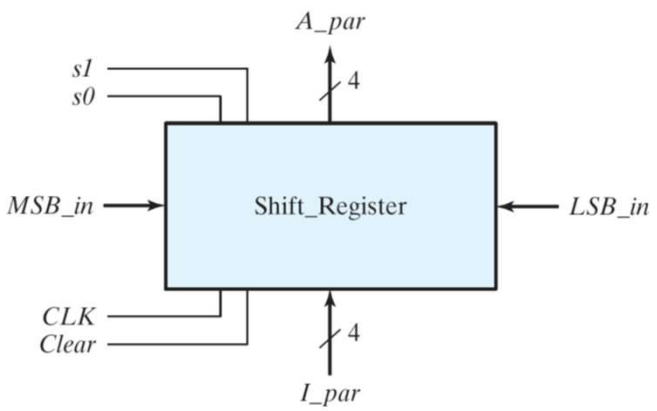
■27



$$A_3^+ = s_1's_0'A_3 + s_1's_0\,SI_{shr} + s_1s_0'A_2 + s_1s_0\,I_3$$

6-28

■28



J.J. Shann 6-29

29

6-3

Ripple Counters

J.J. Shann

30

Ripple Counters

- Counter:
  - a register that goes through a prescribed sequence of states upon the application of input pulses:
    - Input pulses:
      may be clock pulses or
      originate from some external source
    - Timing:
      may occur at regular or
      irregular intervals of time
    - The sequence of states:
      may follow the binary number sequence (⇒ Binary counter) or
      any other sequence of states

J.J. Shann 6-31

31

- Categories of counters:
  1. ***Ripple counters***: (§6-3)
     The flip-flop output transition serves as a source for triggering other flip-flops.
     ⇒ The C input of some or all flip-flops are triggered not by the common clock pulses. (not synchronous)
  2. ***Synchronous counters***: (§6-4, §6-5)
     The C inputs of all flip-flops receive the common clock.

* T or JK flip-flops

J.J. Shann 6-32

32

A. Binary Ripple Counter

- Binary count-up counter:
  - E.g.: 4-bit binary count-up ripple counter

Upward Counting Sequence

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0

J.J. Shann 6-33

33

(by D flip-flops)

Upward Counting Sequence

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0

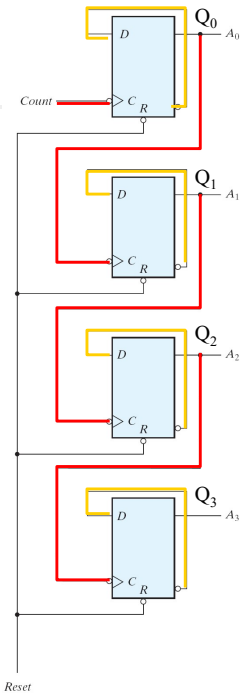
Observations:

FF_0: C_0 = **Clock pulse**
$D_0 = Q_0'$

FF_1: $C_1 = Q_0 (\downarrow)$
$D_1 = Q_1'$

FF_2: $C_2 = Q_1 (\downarrow)$
$D_2 = Q_2'$

FF_3: $C_3 = Q_2 (\downarrow)$
$D_3 = Q_3'$

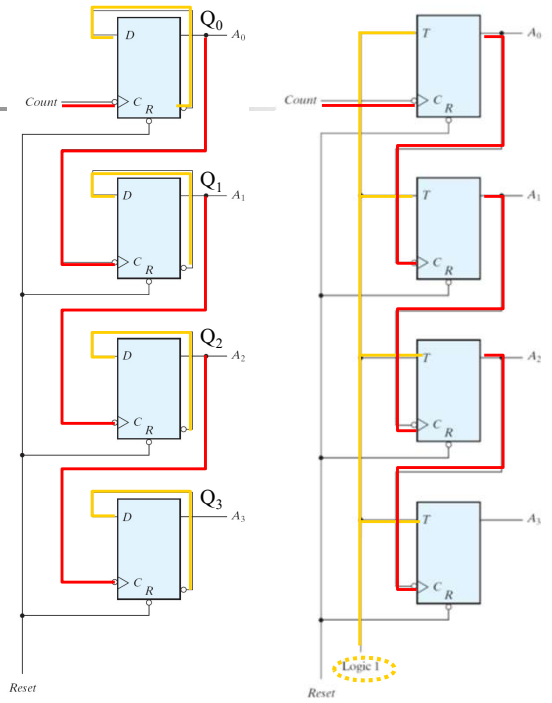


34

(by T flip-flops)

Upward Counting Sequence

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0



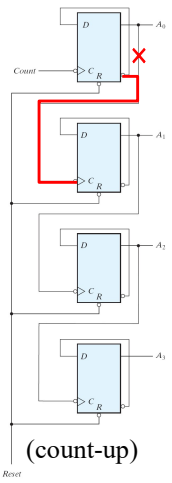
■35

■ Binary count-down counter:

– E.g.: 4-bit binary count-down ripple counter

Downward Counting Sequence

Q_3	Q_2	Q_1	Q_0
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0
1	1	1	1



(count-up)

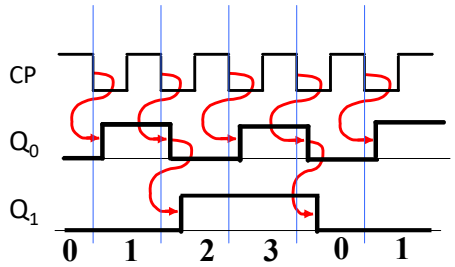
* Modify the figure in p.6-30:
Approach 1
Connect the complement output of each f-f to the C input of the next f-f.
Approach 2
Use positive-edge triggered D f-fs.

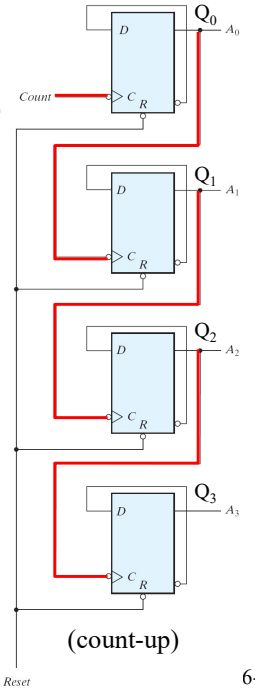
J.J. Shann 6-36

■36

Problem of Ripple Counter

* Problem of ripple counter:
  Accumulation of propagation delay
  - E.g.: 2-bit binary count-up ripple counter





(count-up)

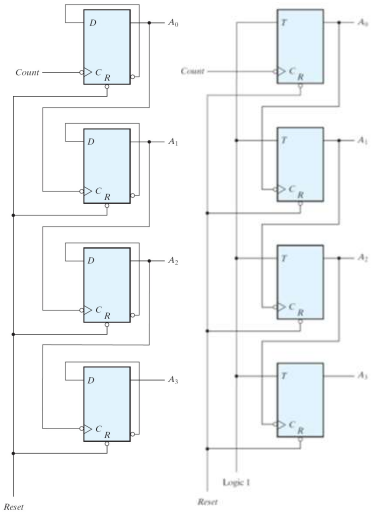
6-37

■37

Summary

■ Summary:
- are asynchronous ckts
- Adv.:
  - simple hardware
- Disadv.:
  - become ckt w/ delay dependence and unreliable op
  - large ripple counters can be slow ckts

    ⇐ the length of time required for the ripple to finish

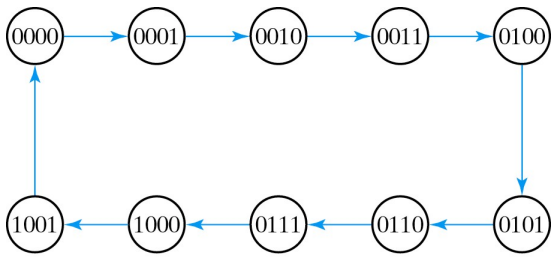


J.J. Shann 6-38

■38

B. BCD Ripple Counter

- State diagram & Count sequence:



State Diagram of a Decimal BCD-Counter

Q_8	Q_4	Q_2	Q_1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0

J.J. Shann 6-39

39

- BCD ripple counter (by J-K flip-flops)

Q_8	Q_4	Q_2	Q_1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0

Observations:

Q_1: C_1 = Count
$J_1 = K_1 = 1$

Q_2: $C_2 = Q_1 (\downarrow)$
$Q_8 = 0, J_2 = 1, K_2 = 1$ (Toggle)
$Q_8 = 1, J_2 = 0, K_2 = 1$ (Reset)
$\Rightarrow J_2 = Q_8', K_2 = 1$

Q_4: $C_4 = Q_2 (\downarrow)$
$J_4 = K_4 = 1$

Q_8: $C_8 = Q_1 (\downarrow)$
$Q_4Q_2 = 1, J_8 = 1, K_8 = 1$ (Toggle)
$Q_4Q_2 = 0, J_8 = 0, K_8 = 1$ (Reset)
$\Rightarrow J_8 = Q_4Q_2, K_8 = 1$

J.J. Shann 6-40

40

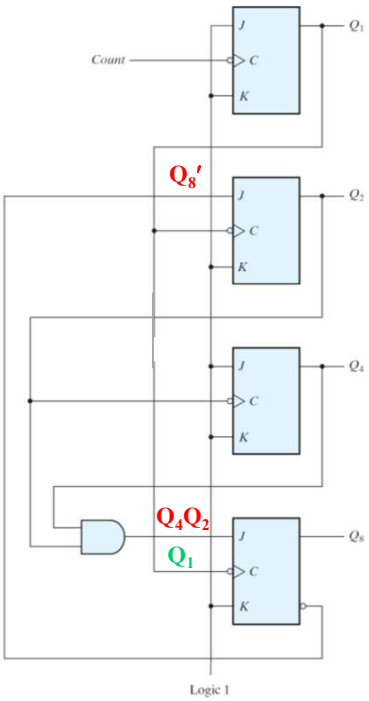
Observations:

Q_1: $C_1 = \text{Count}$
$J_1 = K_1 = 1$

Q_2: $C_2 = Q_1\ (\downarrow)$
$Q_8 = 0,\ J_2 = 1,\ K_2 = 1$ (Toggle)
$Q_8 = 1,\ J_2 = 0,\ K_2 = 1$ (Reset)
$\Rightarrow J_2 = Q_8',\ K_2 = 1$

Q_4: $C_4 = Q_2\ (\downarrow)$
$J_4 = K_4 = 1$

Q_8: $C_8 = Q_1\ (\downarrow)$
$Q_4Q_2 = 1,\ J_8 = 1,\ K_8 = 1$ (Toggle)
$Q_4Q_2 = 0,\ J_8 = 0,\ K_8 = 1$ (Reset)
$\Rightarrow J_8 = Q_4Q_2,\ K_8 = 1$

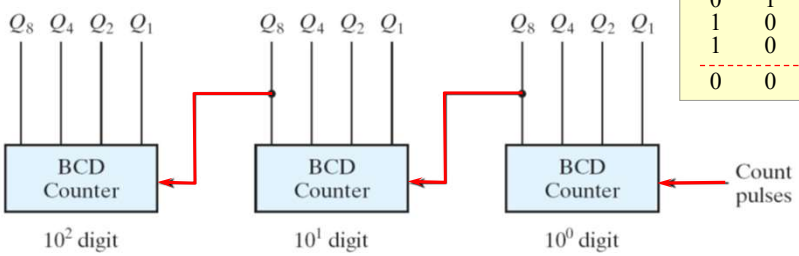


41

■41

■ Three-decade BCD counter: 0~999

Q_8	Q_4	Q_2	Q_1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0



J.J. Shann 6-42

■42

6-4

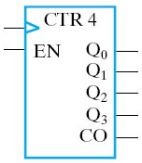
Synchronous Counters

J.J. Shann

43

Synchronous Counters

- Sync counter:
  - A common clock triggers all flip-flops simultaneously.
  - Block diagram: e.g.



* CO: is used to extend the counter to more stages

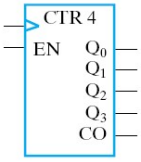
- Design procedure:
  - We can apply the same procedure of sync seq ckts. (Ch5)
  - Sync counter is simpler than general sync seq ckts.

    ⇒ No need to go through a sync seq logic design process.

J.J. Shann 6-44

44

A. Count-Up Binary Counter



- E.g.: 4-bit sync count-up binary counter w/ count enable line using D flip-flops
  - Approach 1: Design procedure of sync seq ckt (Ch5)

Upward Counting Sequence			
Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0

$Q_3Q_2Q_1Q_0$ Present state	$Q_3^+Q_2^+Q_1^+Q_0^+$ Next state EN = 0	EN = 1	Output CO EN = 0	EN = 1
0 0 0 0	0 0 0 0	0 0 0 1	0	0
0 0 0 1	0 0 1 1	0 0 1 0	0	0
0 0 1 0	0 0 1 0	0 0 1 1	0	0
0 0 1 1	0 0 1 1	0 1 0 0	0	0
0 1 0 0	0 1 0 0	0 1 0 1	0	0
0 1 0 1	0 1 0 1	0 1 1 0	0	0
0 1 1 0	0 1 1 0	0 1 1 1	0	0
0 1 1 1	0 1 1 1	1 0 0 0	0	0
1 0 0 0	1 0 0 0	1 0 0 1	0	0
1 0 0 1	1 0 0 1	1 0 1 0	0	0
1 0 1 0	1 0 1 0	1 0 1 1	0	0
1 0 1 1	1 0 1 1	1 1 0 0	0	0
1 1 0 0	1 1 0 0	1 1 0 1	0	0
1 1 0 1	1 1 0 1	1 1 1 0	0	0
1 1 1 0	1 1 1 0	1 1 1 1	0	0
1 1 1 1	1 1 1 1	0 0 0 0	0	1

$$D_0 = Q_0\,EN' + Q_0'\,EN$$
$$= Q_0 \oplus EN$$

$$D_1 = Q_1Q_0' + Q_1EN' + Q_1'Q_0EN$$
$$= Q_1(Q_0' + EN') + Q_1'(Q_0EN)$$
$$= Q_1 \oplus (Q_0 \cdot EN)$$

$$D_2 = Q_2 \oplus (Q_1 \cdot Q_0 \cdot EN)$$

$$D_3 = Q_3 \oplus (Q_2 \cdot Q_1 \cdot Q_0 \cdot EN)$$

$$CO = Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0 \cdot EN$$

J.J. Shann 6-45

- Approach 2: Observation

Upward Counting Sequence			
Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0

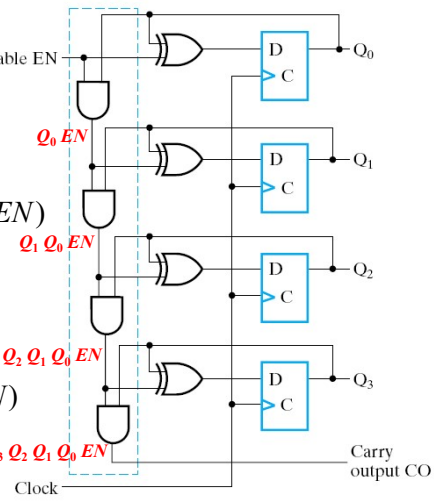
$$D_0 = Q_0\,EN' + Q_0'\,EN$$
$$= Q_0 \oplus EN$$

$$D_1 = Q_1(Q_0EN)' + Q_1'(Q_0EN)$$
$$= Q_1 \oplus (Q_0 \cdot EN)$$

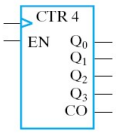
$$D_2 = Q_2 \oplus (Q_1 \cdot Q_0 \cdot EN)$$

$$D_3 = Q_3 \oplus (Q_2 \cdot Q_1 \cdot Q_0 \cdot EN)$$

$$CO = Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0 \cdot EN$$



J.J. Shann 6-46



- E.g.: 4-bit sync count-up binary counter w/ count enable line using JK flip-flops

Upward Counting Sequence			
A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0

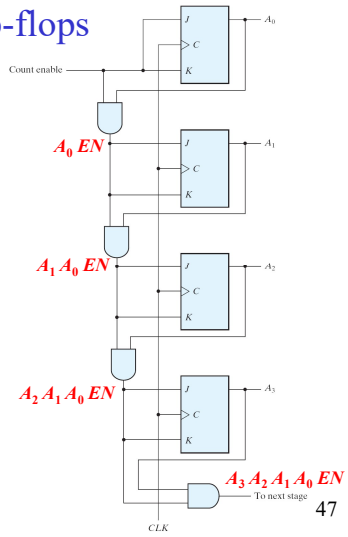
$$J_0 = K_0 = EN$$

$$J_1 = K_1 = A_0 \cdot EN$$

$$J_2 = K_2 = A_1 \cdot A_0 \cdot EN$$

$$J_3 = K_3 = A_2 \cdot A_1 \cdot A_0 \cdot EN$$

$$CO = A_3 \cdot A_2 \cdot A_1 \cdot A_0 \cdot EN$$



47

■47

B. Up-Down Binary Counter

- E.g.: 4-bit up-down binary counter w/ count enable line by using D flip-flops

Upward Counting Sequence				Downward Counting Sequence			
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	0
0	0	1	0	1	1	0	1
0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	0	1	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	0	0
1	0	0	0	0	1	1	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	1	1
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0

J.J. Shann 6-48

■48

Up-ward counting: (p.6-41)

Upward Counting Sequence			
Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0

$$D_0 = Q_0 \oplus EN$$

$$D_1 = Q_1 \oplus (Q_0 \cdot EN)$$

$$D_2 = Q_2 \oplus (Q_0 \cdot Q_1 \cdot EN)$$

$$D_3 = Q_3 \oplus (Q_0 \cdot Q_1 \cdot Q_2 \cdot EN)$$

Down-ward counting:

Do wnward Counting Seqence			
Q_3	Q_2	Q_1	Q_0
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0
1	1	1	1

$$D_0 = Q_0 \oplus EN$$

$$D_1 = Q_1 \oplus (\overline{Q}_0 \cdot EN)$$

$$D_2 = Q_2 \oplus (\overline{Q}_0 \cdot \overline{Q}_1 \cdot EN)$$

$$D_3 = Q_3 \oplus (\overline{Q}_0 \cdot \overline{Q}_1 \cdot \overline{Q}_2 \cdot EN)$$

J.J. Shann 6-49

49

Up-ward counting:

$$D_0 = Q_0 \oplus EN$$

$$D_1 = Q_1 \oplus (Q_0 \cdot EN)$$

$$D_2 = Q_2 \oplus (Q_0 \cdot Q_1 \cdot EN)$$

$$D_3 = Q_3 \oplus (Q_0 \cdot Q_1 \cdot Q_2 \cdot EN)$$

Down-ward counting:

$$D_0 = Q_0 \oplus EN$$

$$D_1 = Q_1 \oplus (\overline{Q}_0 \cdot EN)$$

$$D_2 = Q_2 \oplus (\overline{Q}_0 \cdot \overline{Q}_1 \cdot EN)$$

$$D_3 = Q_3 \oplus (\overline{Q}_0 \cdot \overline{Q}_1 \cdot \overline{Q}_2 \cdot EN)$$

Up-down counter:

S = 0 … up-counting
1 … down-counting

S = 0

S = 1

$$D_0 = Q_0 \oplus EN$$

$$D_1 = Q_1 \oplus ((Q_0 \cdot \overline{S} + \overline{Q}_0 \cdot S) \cdot EN)$$

$$D_2 = Q_2 \oplus ((Q_0 \cdot Q_1 \cdot \overline{S} + \overline{Q}_0 \cdot \overline{Q}_1 \cdot S) \cdot EN)$$

$$D_3 = Q_3 \oplus ((Q_0 \cdot Q_1 \cdot Q_2 \cdot \overline{S} + \overline{Q}_0 \cdot \overline{Q}_1 \cdot \overline{Q}_2 \cdot S) \cdot EN)$$

J.J. Shann 6-50

50

- E.g.: 4-bit up-down binary counter by using T flip-flops (no enable input)

$$T_0 = 1$$
$$T_1 = A_0$$
$$T_2 = A_0 \cdot A_1$$
$$T_3 = A_0 \cdot A_1 \cdot A_2$$

Upward Counting Sequence				Downward Counting Sequence			
A_3	A_2	A_1	A_0	A_3	A_2	A_1	A_0
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	0
0	0	1	0	1	1	0	1
0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	0	1	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	0	0
1	0	0	0	0	1	1	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	1	1
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0

$$T_0 = 1$$
$$T_1 = A_0'$$
$$T_2 = A_0' \cdot A_1'$$
$$T_3 = A_0' \cdot A_1' \cdot A_2'$$

J.J. Shann 6-51

Up-ward counting:

$$T_0 = 1$$
$$T_1 = A_0$$
$$T_2 = A_0 \cdot A_1$$
$$T_3 = A_0 \cdot A_1 \cdot A_2$$

Down-ward counting:

$$T_0 = 1$$
$$T_1 = A_0'$$
$$T_2 = A_0' \cdot A_1'$$
$$T_3 = A_0' \cdot A_1' \cdot A_2'$$

$Up = 1$

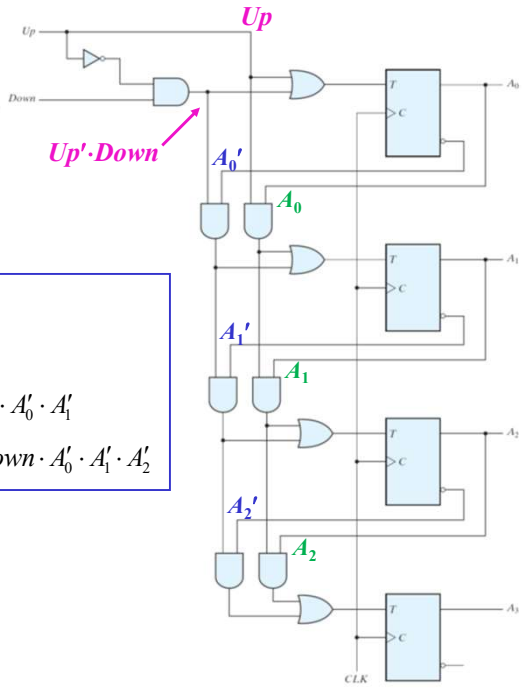
$Up = 0$ & $Down = 1$

Up-down counter:

$Up = 1$ … up-counting
$Up = 0$ and $Down = 1$ … down-counting
$Up = Down = 0$ … no-counting

$$T_0 = Up + Up' \cdot Down = Up + Down$$
$$T_1 = Up \cdot A_0 + Up' \cdot Down \cdot A_0'$$
$$T_2 = Up \cdot A_0 \cdot A_1 + Up' \cdot Down \cdot A_0' \cdot A_1'$$
$$T_3 = Up \cdot A_0 \cdot A_1 \cdot A_2 + Up' \cdot Down \cdot A_0' \cdot A_1' \cdot A_2'$$

J.J. Shann 6-52



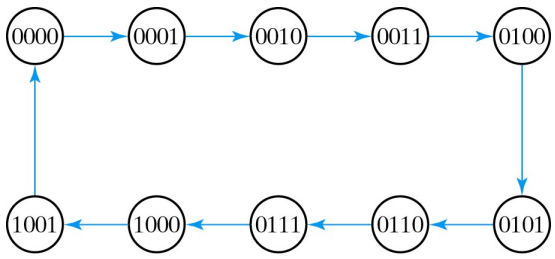
$$T_0 = Up + Up' \cdot Down$$
$$T_1 = Up \cdot A_0 + Up' \cdot Down \cdot A_0'$$
$$T_2 = Up \cdot A_0 \cdot A_1 + Up' \cdot Down \cdot A_0' \cdot A_1'$$
$$T_3 = Up \cdot A_0 \cdot A_1 \cdot A_2 + Up' \cdot Down \cdot A_0' \cdot A_1' \cdot A_2'$$

6-53

■53

C. BCD Counter

- State diagram & Count sequence:



State Diagram of a Decimal BCD-Counter

Q_8	Q_4	Q_2	Q_1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0

J.J. Shann 6-54

■54

Synchronous BCD Counter

- E.g.: 4-bit sync BCD counter w/ D-type f-fs (no enable input)

State Table and Flip-Flop Inputs for BCD Counter

Present State				Next State				Output
Q_8	Q_4	Q_2	Q_1	$D_8 = Q_8(t+1)$	$D_4 = Q_4(t+1)$	$D_2 = Q_2(t+1)$	$D_1 = Q_1(t+1)$	Y
0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	1	1	0
0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	0	1	0
0	1	0	1	0	1	1	0	0
0	1	1	0	0	1	1	1	0
0	1	1	1	1	0	0	0	0
1	0	0	0	1	0	0	1	0
1	0	0	1	0	0	0	0	1
0	0	0	0					

$$D_1 = \overline{Q}_1$$

$$D_2 = Q_2 \oplus Q_1\overline{Q}_8$$

$$D_4 = Q_4 \oplus Q_1Q_2$$

$$D_8 = Q_8 \oplus (Q_1Q_8 + Q_1Q_2Q_4)$$

$$Y = Q_1Q_8$$

J.J. Shann 6-55

55

- E.g.: 4-bit sync BCD counter w/ T-type f-fs

State Table for BCD Counter

Present State				Next State				Output	Flip-Flop Inputs			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	y	TQ_8	TQ_4	TQ_2	TQ_1
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1

$TQ_1 = 1$, $TQ_2 = Q_8'Q_1$, $TQ_4 = Q_2Q_1$, $TQ_8 = Q_8Q_1 + Q_4Q_2Q_1$,
$y = Q_8Q_1$

J.J. Shann 6-56

56

D. Binary Counter w/ Parallel Load

- E.g.: 4-bit count-up binary counter w/ parallel load and async clear
  - Function table:

Function Table for the Counter of Fig. 6.14

direct input

Clear	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change

Clear	CLK	Load	Count	A_3^+	A_2^+	A_1^+	A_0^+
0	×	×	×	0	0	0	0
1	↑	1	×	I_3	I_2	I_1	I_0
1	↑	0	1	A_3	A_2	A_1	A_0 + 1
1	↑	0	0	A_3	A_2	A_1	A_0

J.J. Shann 6-57

Clear	CLK	Load	Count	A_3^+	A_2^+	A_1^+	A_0^+
0	×	×	×	0	0	0	0
1	↑	1	×	I_3	I_2	I_1	I_0
1	↑	0	1	A_3	A_2	A_1	A_0 + 1
1	↑	0	0	A_3	A_2	A_1	A_0

Using D flip-flops with asynchronous clear:

$$D_i = Load \cdot I_i + Load' \, Count \, (A_i \oplus A_{i-1} \ldots A_0) + Load' \cdot Count' \, A_i$$

$D_0 = ?$

Using JK flip-flops with asynchronous clear:

$$J_i = Load \cdot I_i + Load' \, Count \, A_{i-1} \ldots A_0$$
$$K_i = Load \cdot I_i' + Load' \, Count \, A_{i-1} \ldots A_0$$

$J_0 = ?$

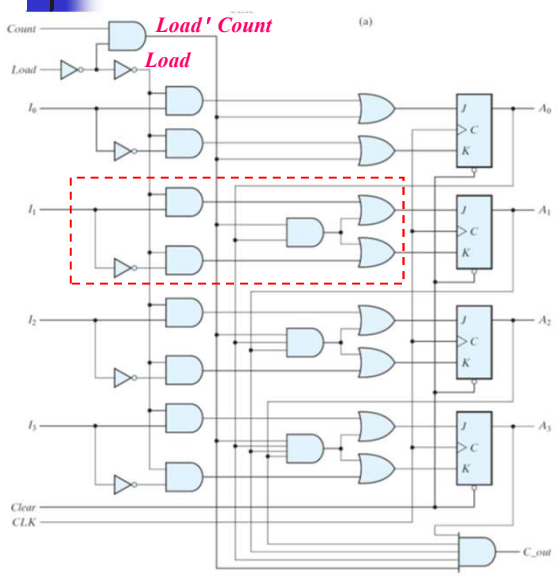
$K_0 = ?$

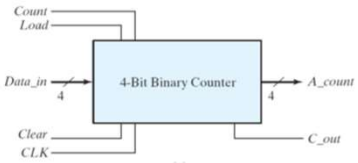
J.J. Shann 6-58

Using JK flip-flops with asynchronous clear:

$J_i = Load \cdot I_i + Load' \; Count \; A_{i-1} \ldots A_0 \quad (J_0 = Load \cdot I_0 + Load' \; Count)$

$K_i = Load \cdot I_i' + Load' \; Count \; A_{i-1} \ldots A_0 \quad (K_0 = Load \cdot I_0' + Load' \; Count)$





Function Table for the Counter of Fig. 6.14

Clear	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change

J.J. Shann 6-59

■59

Generating Other Count Sequences

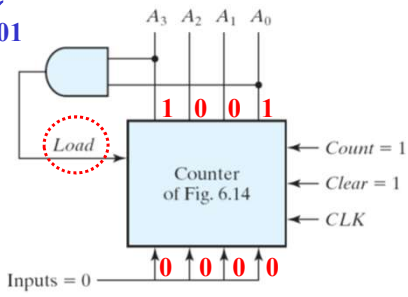
Function Table for the Counter of Fig. 6.14

Clear	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change

- Generate any count sequence:
  - E.g.: design a BCD counter by using a counter w/ parallel load & ***async*** clear

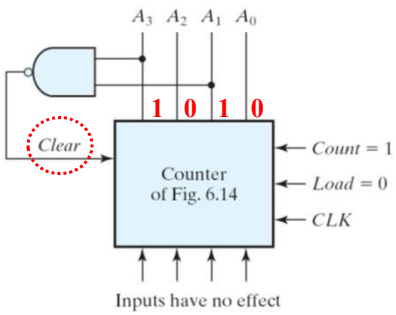
0000 → ⋮ → 1001

Using the load input:



(a) Using the load input

Using the clear input:



(b) Using the clear input

J.J. Shann 6-60

■60

6-5

Other Counters (Sync Seq Ckt)

J.J. Shann

61

Other Counters (Sync Seq Ckt)

- Counters:
  - can be designed to generate any desired sequence of states
- Binary counter
- BCD counter
- Divide-by-*N* counter: modulo-N counter
  - a counter that goes through a repeated sequence of *N* states
  - The sequence may follow the binary count or may be any other arbitrary sequence.

J.J. Shann 6-62

62

A. Counter w/ Unused States

- n flip-flops $\Rightarrow 2^n$ binary states
- Unused states:
  - states that are not used in specifying the sequential ckt
  - may be treated as don't-care conditions or may be assigned specific next states
- Self-correcting counter:
  - Ensure that when a ckt enter one of its unused states, it eventually goes into one of the valid states after one or more clock pulses so it can resume normal operation.

    $\Rightarrow$ Analyze the ckt to determine the next state from an unused state after it is designed.

J.J. Shann 6-63

63

- Example: Two unused states: 011 & 111

State Table for Counter

Present State			Next State			Flip-Flop Inputs					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

The simplified f-f input eqs: Unused states $\Rightarrow$ don't-care conditions

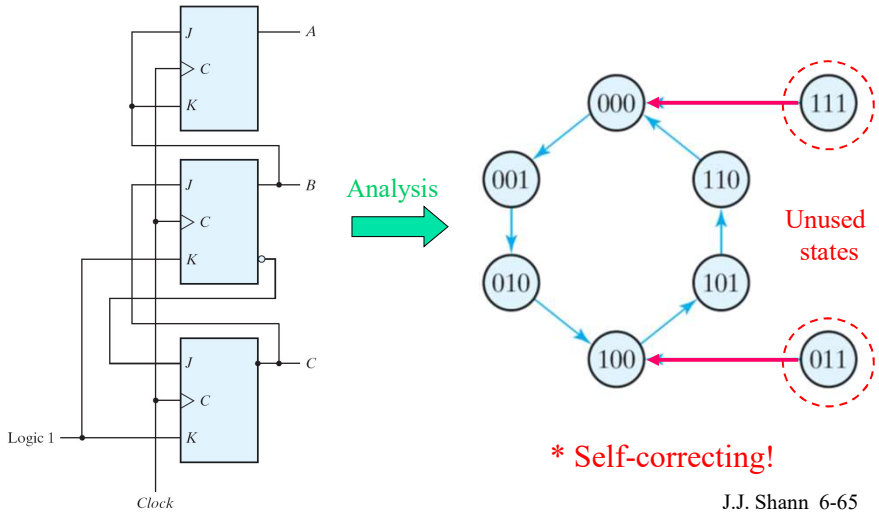
$J_A = B$, $K_A = B$; $J_B = C$, $K_B = 1$; $J_C = B'$, $K_C = 1$

J.J. Shann 6-64

64

$J_A = B,\ K_A = B;\ J_B = C,\ K_B = 1;\ J_C = B',\ K_C = 1$

The logic diagram & state diagram of the ckt:



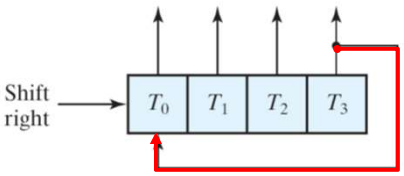
* Self-correcting!

J.J. Shann 6-65

B. Ring Counter

- Ring counter:
  - a circular shift register w/ only one flip-flop being set at any particular time, all others are cleared (initial value = 1 0 0 … 0)
  - The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals.
  - E.g.: 4-bit ring counter

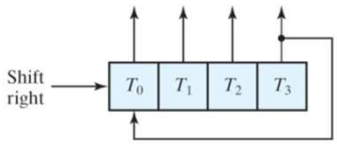
T_0	T_1	T_2	T_3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0



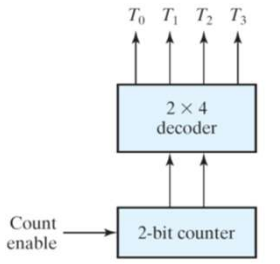
(a) Ring-counter (initial value = 1000)

J.J. Shann 6-66

- Application of counters:
  - Counters may be used to generate timing signals to control the sequence of operations in a digital system.
- Approaches for generation of 2^n timing signals:
  1. a shift register (ring-counter) w/ 2^n flip-flops
  2. an n-bit binary counter + an n-to-2^n-line decoder

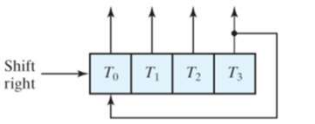


(a) Ring-counter (initial value = 1000)

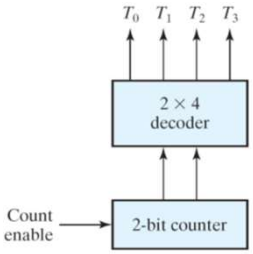


(c) Counter and decoder

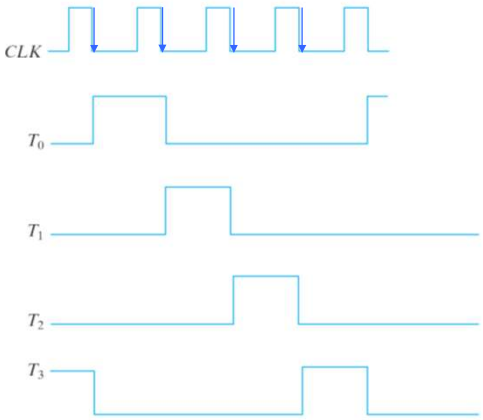
Shann 6-67



(a) Ring-counter (initial value = 1000)



(c) Counter and decoder

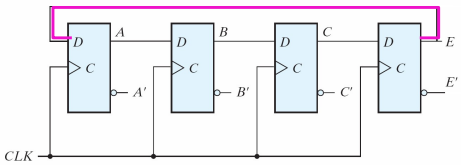


(b) Sequence of four timing signals

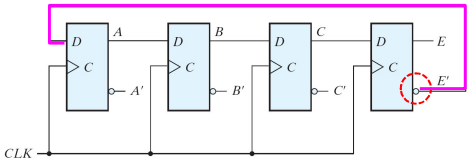
J.J. Shann 6-68

C. Johnson Counter (Switch-tail Ring Counter)

- Ring counter vs. Switch-tail ring counter:
  - Ring counter:
    - a *k*-bit ring counter circulates a single bit among the flip-flops to provide *k* distinguishable states. (initial value = 1 0 … 0)



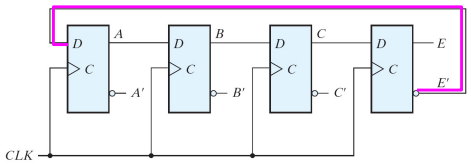
  - Switch-tail ring counter:
    - is a circular shift register w/ the complement output of the last flip-flop connected to the input of the first flip-flop
    - a *k*-bit switch-tail ring counter will go through a sequence of 2*k* distinguishable states. (initial value = 0 0 … 0)



J.J. Shann 6-69

■69

- E.g.: 4-bit switch-tail ring counter
  - initial value = 0 0 … 0



	Sequence number	Flip-flop outputs A	B	C	E	AND gate required for output
T_0	1	0	0	0	0	$A'E'$
	2	1	0	0	0	AB'
	3	1	1	0	0	BC'
.	4	1	1	1	0	CE'
.	5	1	1	1	1	AE
.	6	0	1	1	1	$A'B$
	7	0	0	1	1	$B'C$
T_7	8	0	0	0	1	$C'E$

(b) Count sequence and required decoding

$A'E'$

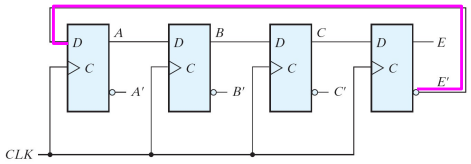
T_0

AB \ CE	00	01	11	10
00	1	0	0	×
01	×	×	0	×
11	0	×	0	0
10	0	×	×	×

J.J. Shann 6-70

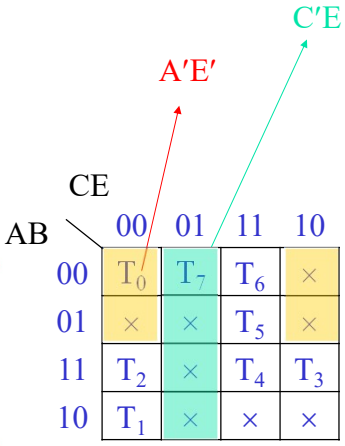
■70

E.g.: 4-bit switch-tail ring counter



	Sequence number	Flip-flop outputs A	B	C	E	AND gate required for output
T_0	1	0	0	0	0	$A'E'$
	2	1	0	0	0	AB'
	3	1	1	0	0	BC'
.	4	1	1	1	0	CE'
.	5	1	1	1	1	AE
.	6	0	1	1	1	$A'B$
	7	0	0	1	1	$B'C$
T_7	8	0	0	0	1	$C'E$

(b) Count sequence and required decoding



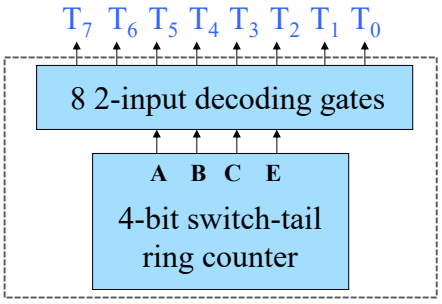
J.J. Shann 6-71

■71

Johnson counter:

- a k-bit switch-tail ring counter + $2k$ 2-input decoding gates
- provide outputs for $2k$ timing signals
  - E.g.: 4-bit Johnson counter

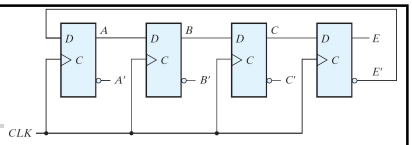
	Sequence number	Flip-flop outputs A	B	C	E	AND gate required for output
T_0	1	0	0	0	0	$A'E'$
	2	1	0	0	0	AB'
	3	1	1	0	0	BC'
.	4	1	1	1	0	CE'
.	5	1	1	1	1	AE
.	6	0	1	1	1	$A'B$
	7	0	0	1	1	$B'C$
T_7	8	0	0	0	1	$C'E$



- The decoding follows a regular pattern:
  - 2 inputs per decoding gate

J.J. Shann 6-72

■72



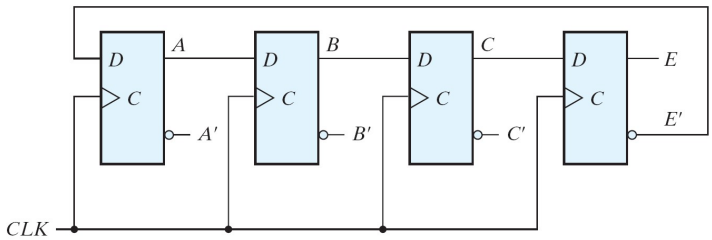
- Disadv. of the switch-tail ring counter:
  - If it finds itself in an unused state, it will persist to circulate in the invalid states and never find its way to a valid state. ⇒ Not self-correcting!

Sequence number	Flip-flop outputs A	B	C	E
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1

Unused states: 8
0010
0100
0101
0110
1001
1010
1011
1101

J.J. Shann 6-73

Analysis of the ckt:



$$\left.\begin{aligned} A^+ &= D_A = E' \\ B^+ &= D_B = A \\ C^+ &= D_C = B \\ E^+ &= D_E = C \end{aligned}\right\}$$

⇒Binary-coded state table
⇒State diagram

J.J. Shann 6-74

State diagram

(Unused states)

AB \ CE	00	01	11	10
00	1	0	0	×
01	×	×	0	×
11	1	×	0	1
10	1	×	×	×

$A^+ = E'$

AB \ CE	00	01	11	10
00	0	0	0	×
01	×	×	0	×
11	1	×	1	1
10	1	×	×	×

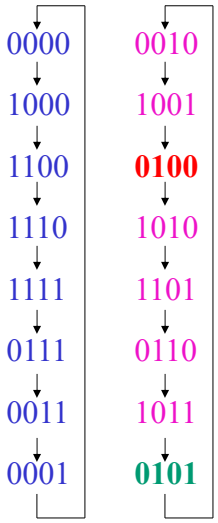
$B^+ = A$

AB \ CE	00	01	11	10
00	0	0	0	×
01	×	×	1	×
11	1	×	1	1
10	0	×	×	×

$C^+ = B$

AB \ CE	00	01	11	10
00	0	0	1	×
01	×	×	1	×
11	0	×	1	1
10	0	×	×	×

$E^+ = C$



J.J. Shann 6-75

■75

State diagram

* Self-correcting!

AB \ CE	00	01	11	10
00	1	0	0	×
01	×	×	0	×
11	1	×	0	1
10	1	×	×	×

$A^+ = E'$

AB \ CE	00	01	11	10
00	0	0	0	×
01	×	×	0	×
11	1	×	1	1
10	1	×	×	×

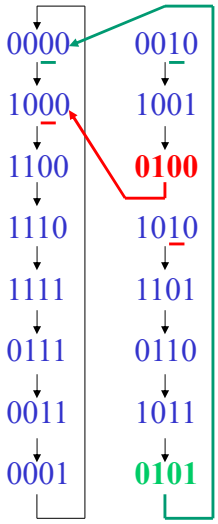
$B^+ = A$

AB \ CE	00	01	11	10
00	0	0	0	×
01	×	×	1	×
11	1	×	1	1
10	0	×	×	×

$C^+ = (A + C)B$

AB \ CE	00	01	11	10
00	0	0	1	×
01	×	×	1	×
11	0	×	1	1
10	0	×	×	×

$E^+ = C$



J.J. Shann 6-76

■76

- Summary:
  - Johnson counters can be constructed for any # of timing sequences:

    # of flip-flops = 1/2 (the # of timing signals)

    # of decoding gates = # of timing signals

    2-input per gate

J.J. Shann 6-77

77

6-6

HDL for Registers and Counters

J.J. Shann

78

HDL for Registers and Counters

- HDL for Shift Register
- HDL for Synchronous Counter
- HDL for Ripple Counter

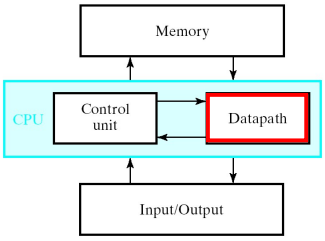
Chapter Summary

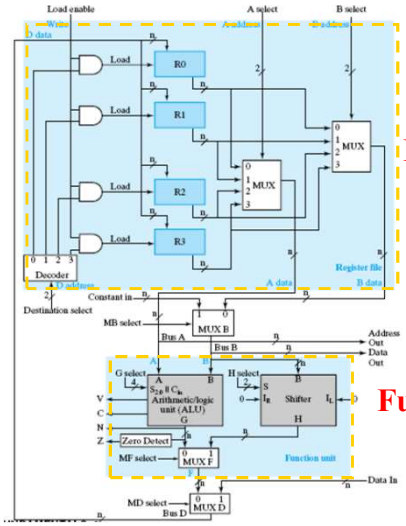
- Registers
  - Simplest register
  - Register with parallel load
  - Shift registers
- Counters
  - Ripple counter
  - Synchronous binary counters
  - Other counters

* Course Summary

- An example of digital systems:

  Digital computer





Register file

Function unit