EECS1010 Logic Design
Lecture 6 Registers and Counters

Jenny Yi-Chun Liu

jennyliu@gapp.nthu.edu.tw

Outline

- Digital systems and information
- Boolean algebra and logic gates
- Gate-level minimization
- Combinational logic
- Sequential circuits
- Registers and counters
- Memory

Chapter Outline

- Registers
- Counters

Registers

Registers (1/2)

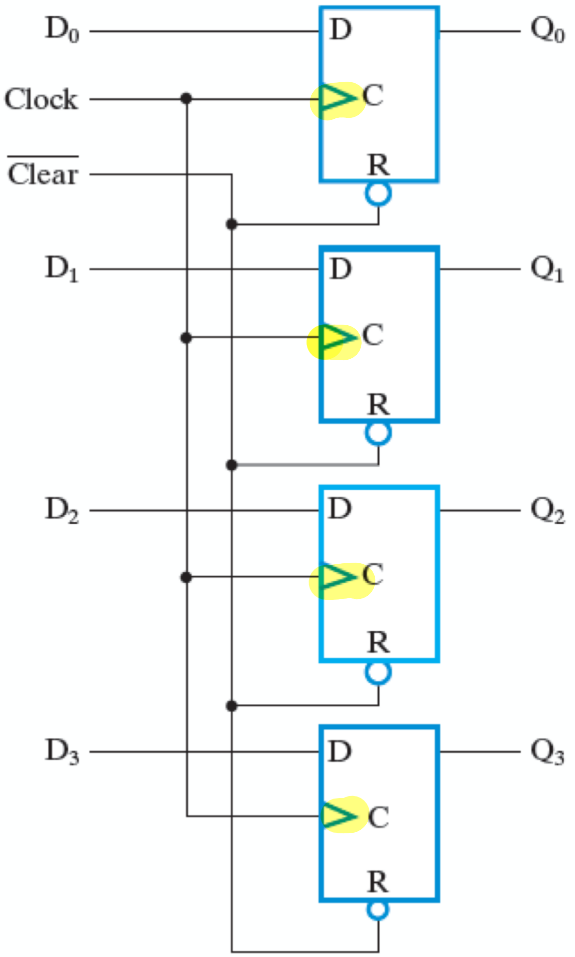
- Register: a collection of binary storage elements

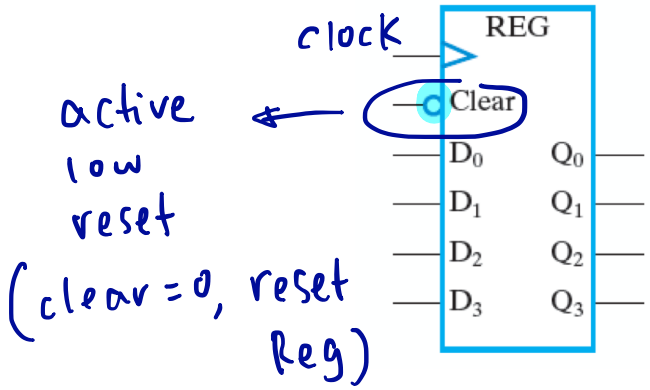
- An n-bit register stores n bit binary information.
- Registers are commonly used to perform data storage and processing.
- A register may have combinational gates to control when and how the new information is transferred into the register.
- Transfer new information into a register is called loading the register.
- Counter: a register that goes through a predetermined sequence of states.

Register (2/2)

- A register can be viewed as a bitwise extension of a flip-flop.
  - The simplest storage component with n inputs, n outputs, and a clock signal.
  - All flip-flops are driven by a common clock signal.

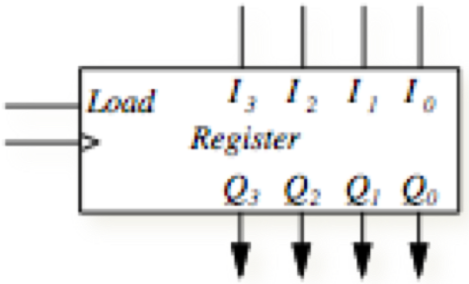
4-bit register



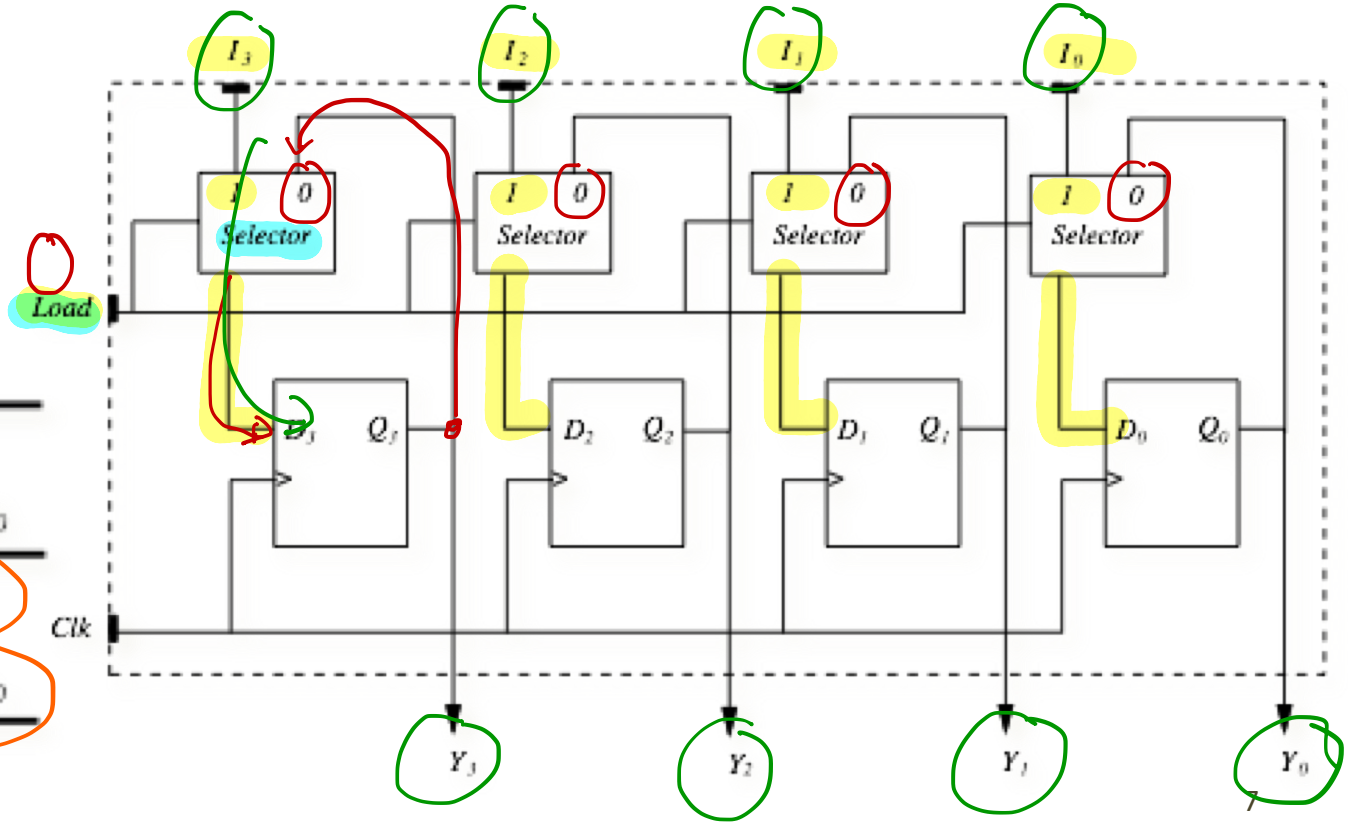


Register with Parallel Load

- Load: control when the data is transferred into a register, and how long it will be stored.

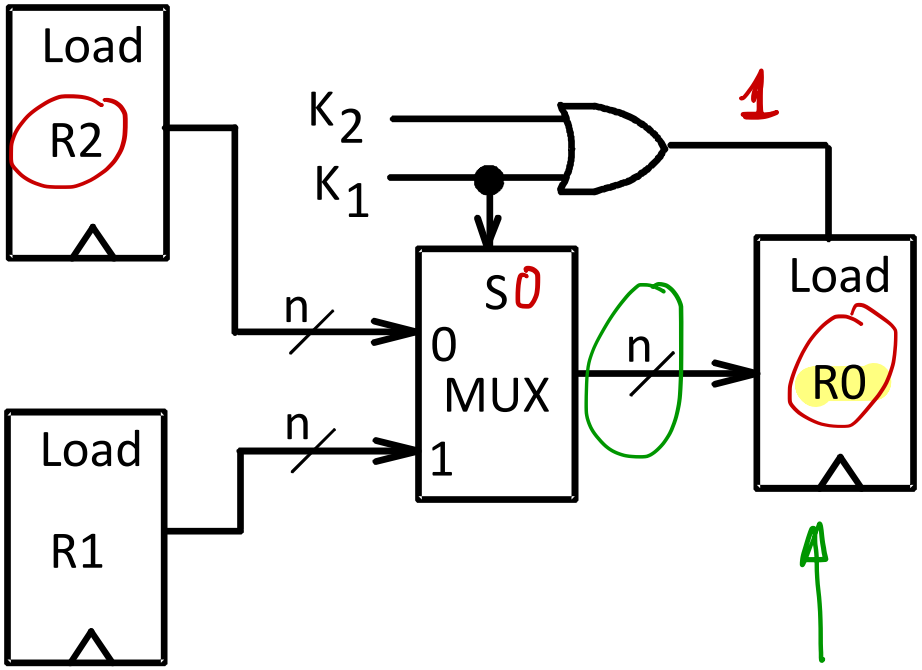


Present state Load	Next state Q_3 Q_2 Q_1 Q_0
0	No change
1	I_3 I_2 I_1 I_0



Multiplexer-Based Transfers

- Register may receive data from many different sources at different times.

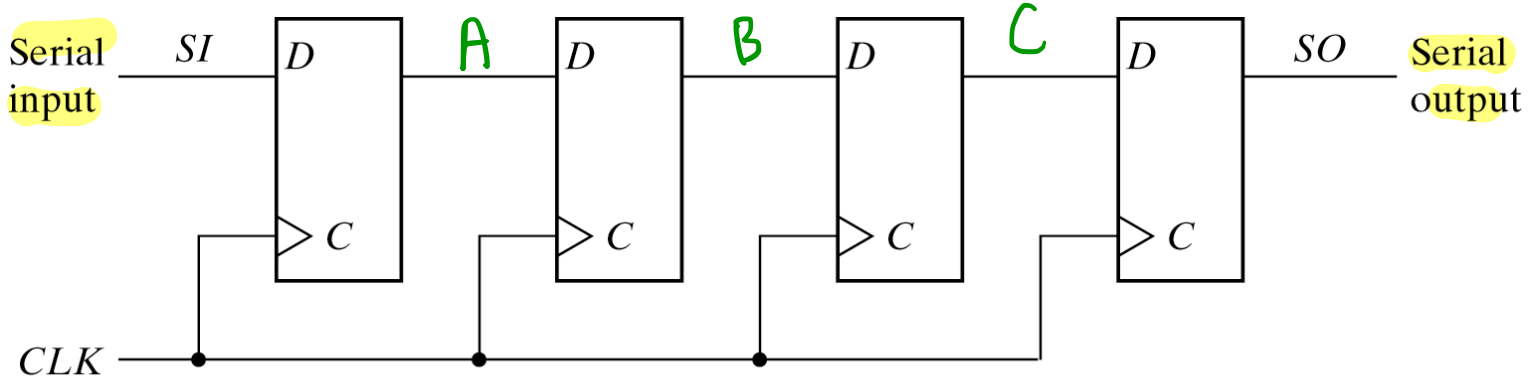


$k_1 = 1, \quad R_0 \leftarrow R_1$

$\bar{k_1} \cdot k_2 = 1, \quad R_0 \leftarrow R_2$

Shift Registers

- Shift registers move data within the register toward its MSB or LSB position. It shifts one bit at a time.
- In the simplest case, the shift register is simply a set of D flip-flops connected in a row like this:

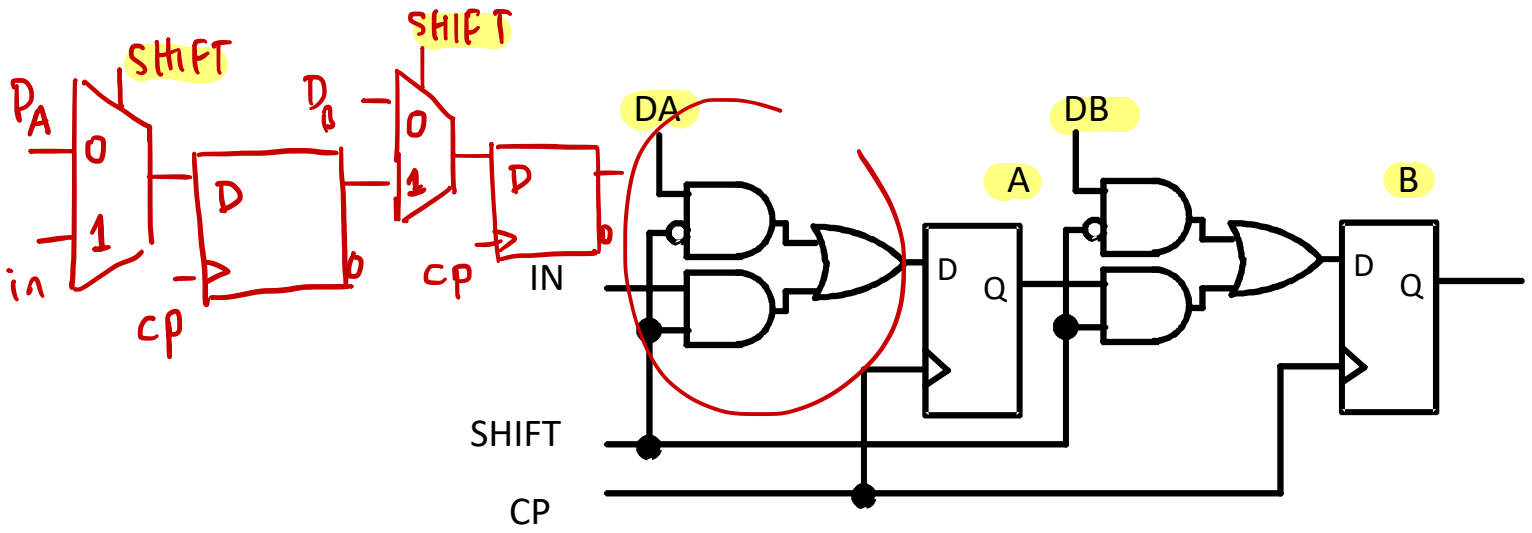


- Data output, SO, is often called Serial output
- The vector (A, B, C, SO) is called parallel output
- Generally, input and output can be serial or parallel.

Shifter Types

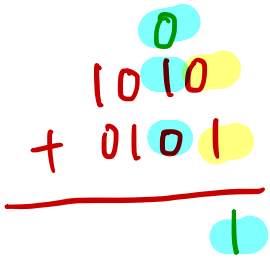
- Logical shifter: shift the number to the left or right and fills empty spots with 0's.
- Arithmetic shifter: same as logical shifter but on right shift fills empty MSBs with the sign bit (sign extension).
- Barrel shifter: rotate numbers in a circle such that empty spots are filled with the bits shifted off the other end.

(p.7+p.9) Parallel Load Shift Registers



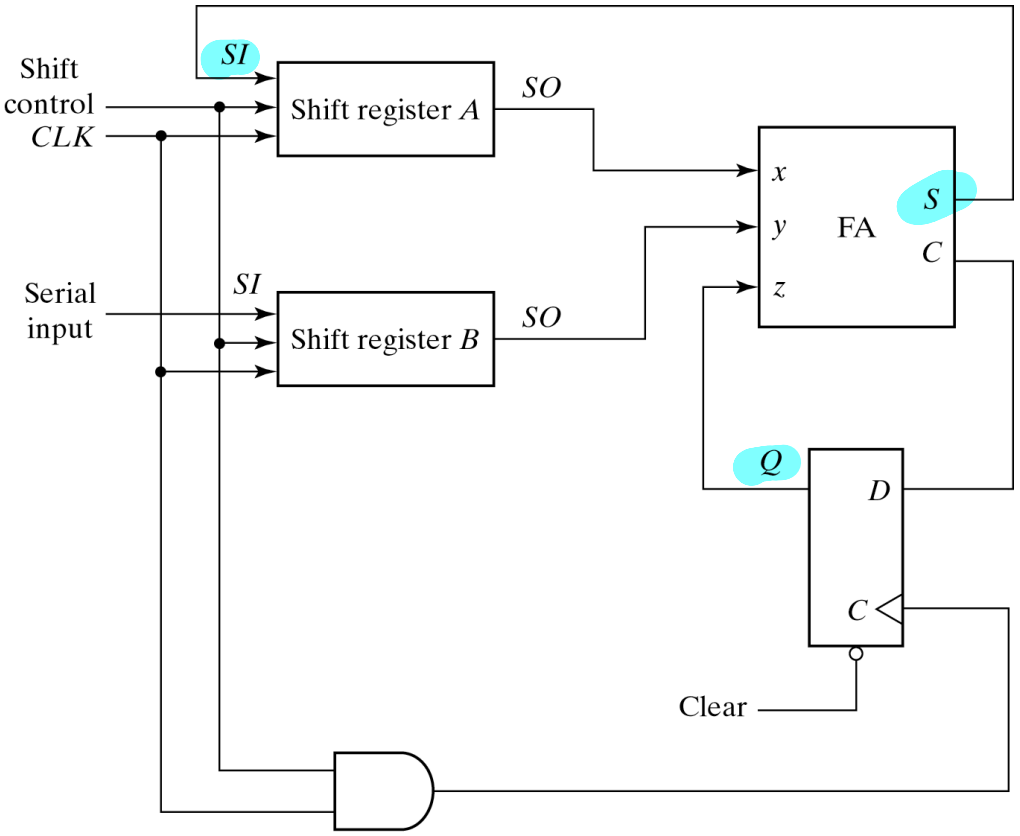
- Add a MUX between each shift register.
- Data can be shifted or loaded.
- When SHIFT = 0, D_A and D_B are loaded into DFF_A and DFF_B.

SHIFT = 1, data shift right.



Serial Adder using DFFs

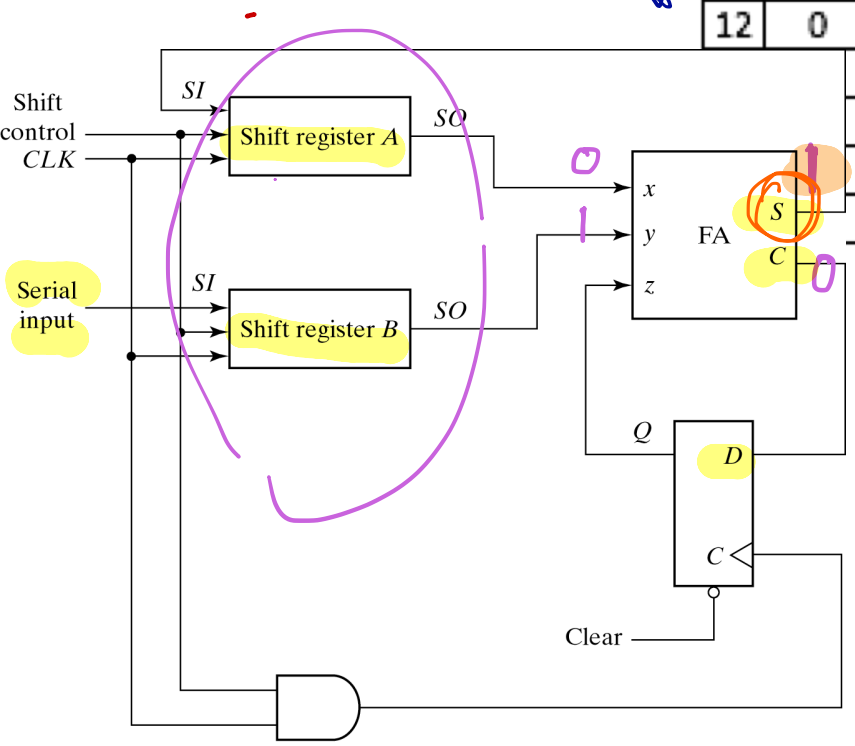
- Initially, augend in register A and addend in register B.
- Shift control enables the triggering of clock and 1-bit addition of two operands from LSB to MSB.
- A new sum bit is transferred to shift register A.
- A carry-out is transferred to Q.
- Finally, shift control is disabled and the sum is stored in shift register A.



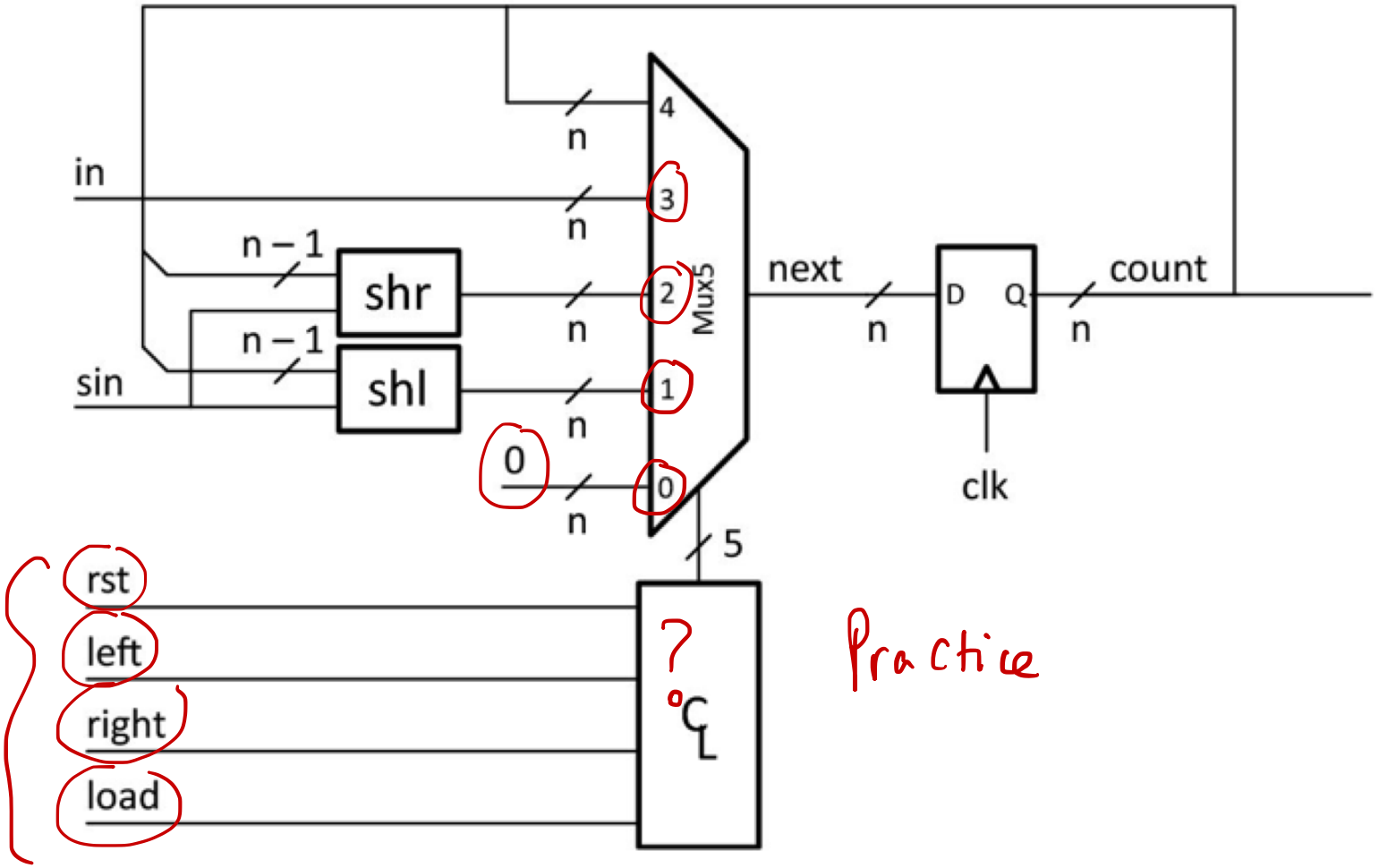
- Example:

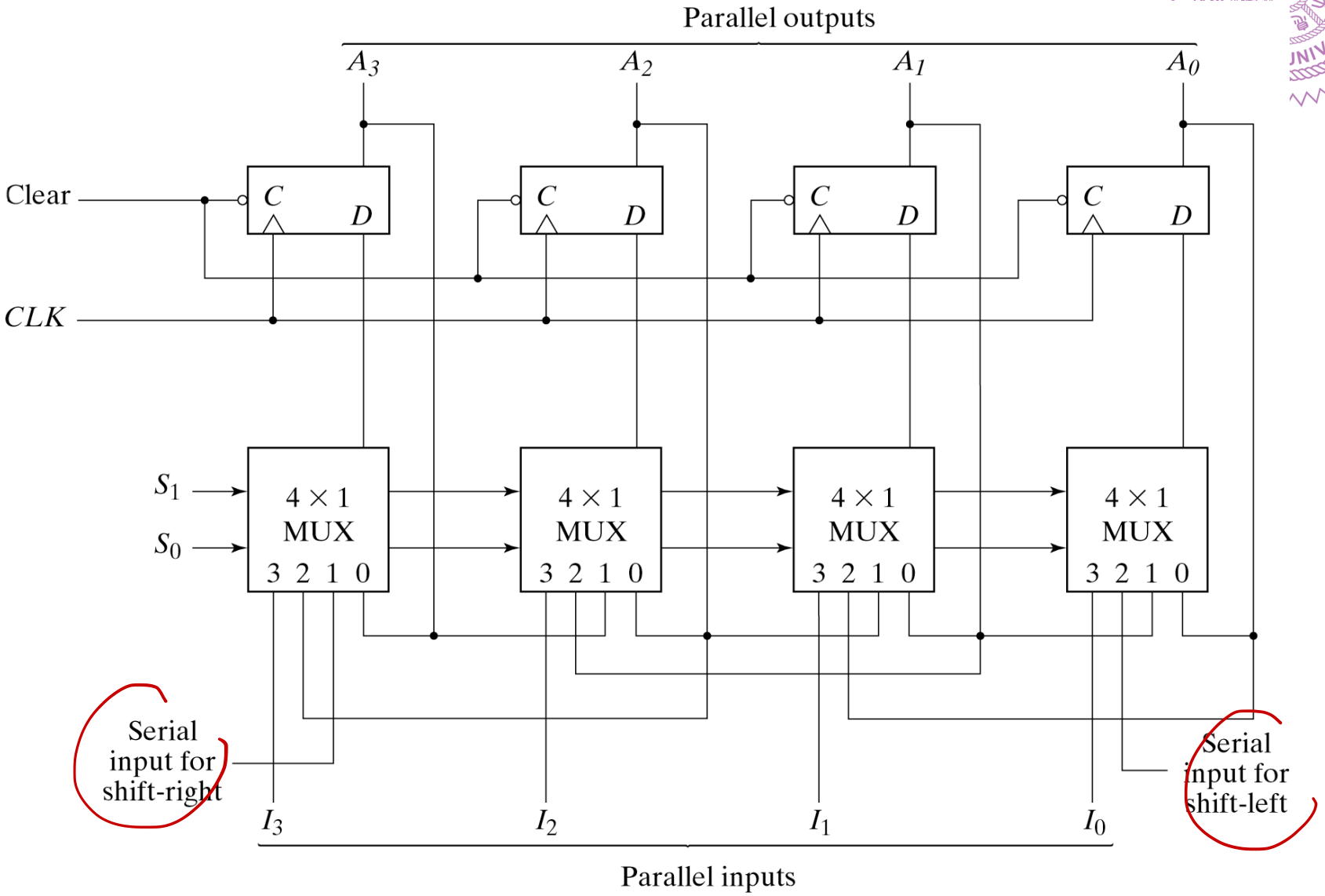
0101
+ 1010

t	SI (B)	B3	B2	B1	B0	SI(A),s	A3	A2	A1	A0	x (A0)	y (B0)	z	c	s
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	0	1	1	0	0	0	0	0	1	0	0	1
6	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
7	0	1	0	0	1	1	0	1	0	0	0	1	0	0	1
8	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0
9	0	1	0	1	0	1	0	1	0	1	1	0	0	0	1
10	0	0	1	0	1	1	1	0	1	0	0	1	0	0	1
11	0	0	0	1	0	1	1	1	0	1	1	0	0	0	1
12	0	0	0	0	1	1	1	1	1	0	0	1	0	0	1
							1	1	1	1	1				
								1	1	1	1				
									1	1	1				
										1	1				



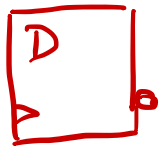
Left/Right/Load Shift Register



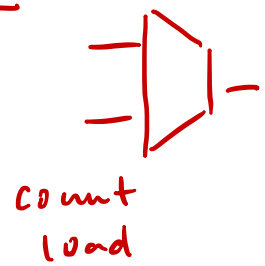

Parallel outputs
A_3
A_2
A_1
A_0
Clear
CLK
C
D
S_1
S_0
4 × 1
MUX
3 2 1 0
Serial
input for
shift-right
Serial
input for
shift-left
I_3
I_2
I_1
I_0
Parallel inputs

1. Ripple
2. Synchronous
3. BCD
4. Ring
5. Johnson

- Edge-triggered
  - DFFs



- Parallel load, reset / enabling



Counters

Counters

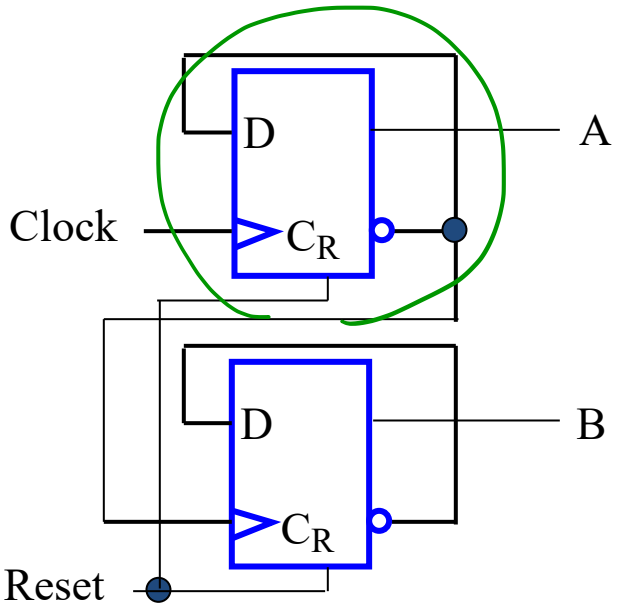
- Counters are sequential circuits which "count" through a specific state sequence.

- Two distinct types are in common usage:
  - Ripple Counters
    - The output transition of flip-flop serves as a source for triggering other flip-flops
  - Synchronous Counters
    - The clock inputs of all flip-flops receive a common clock

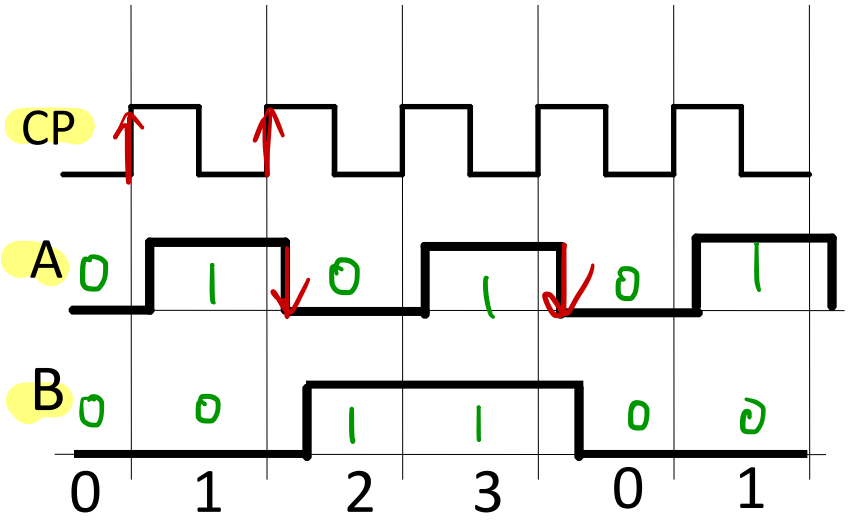
1- Ripple Counter

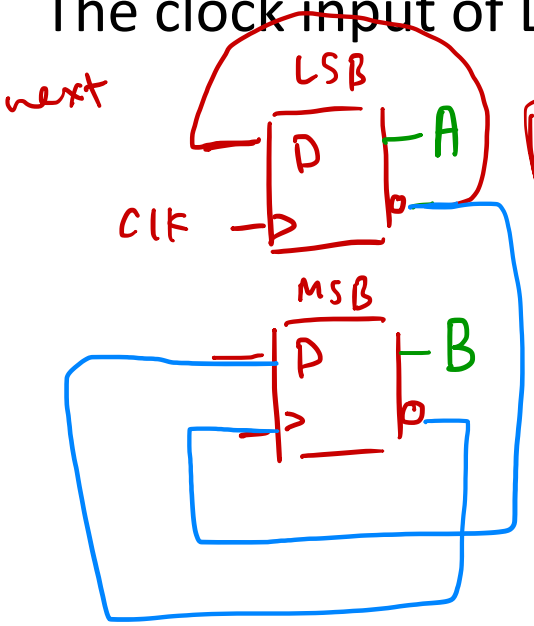
up counting :
= 00 → 01 → 10 → 11 → 00 → ...

down counting
11 → 10 → 01 → 00 → 11 → 10 → ...



- Clock only connects to the LSB FF.
- At positive edge, A complements ($A^+ = \bar{A}$)
- The clock input of DFF_B is $\bar{A}$





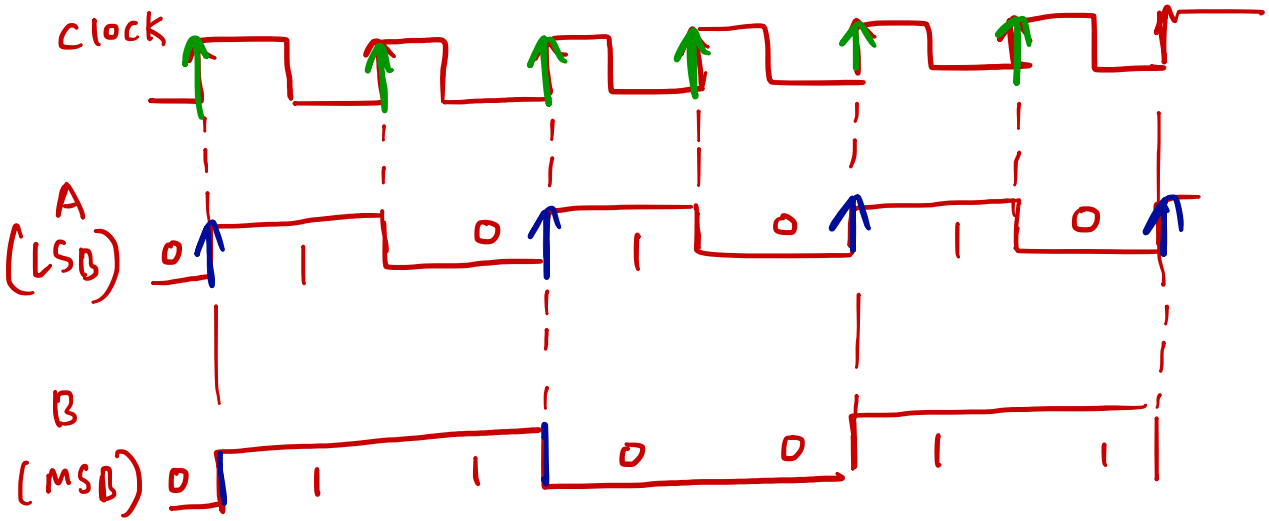
$T_{clock} = \frac{1}{2} T_A$

$= \frac{1}{4} T_B$

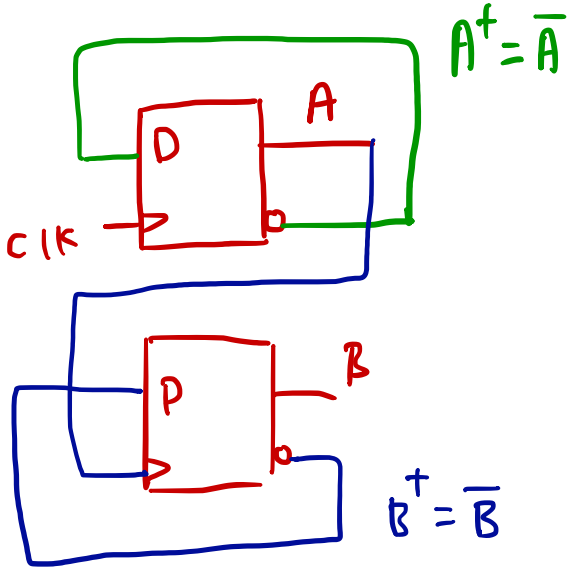
$f_{clock} = 2 f_A$

$= 4 f_B$

2-Bit Downward Ripple Counter Example



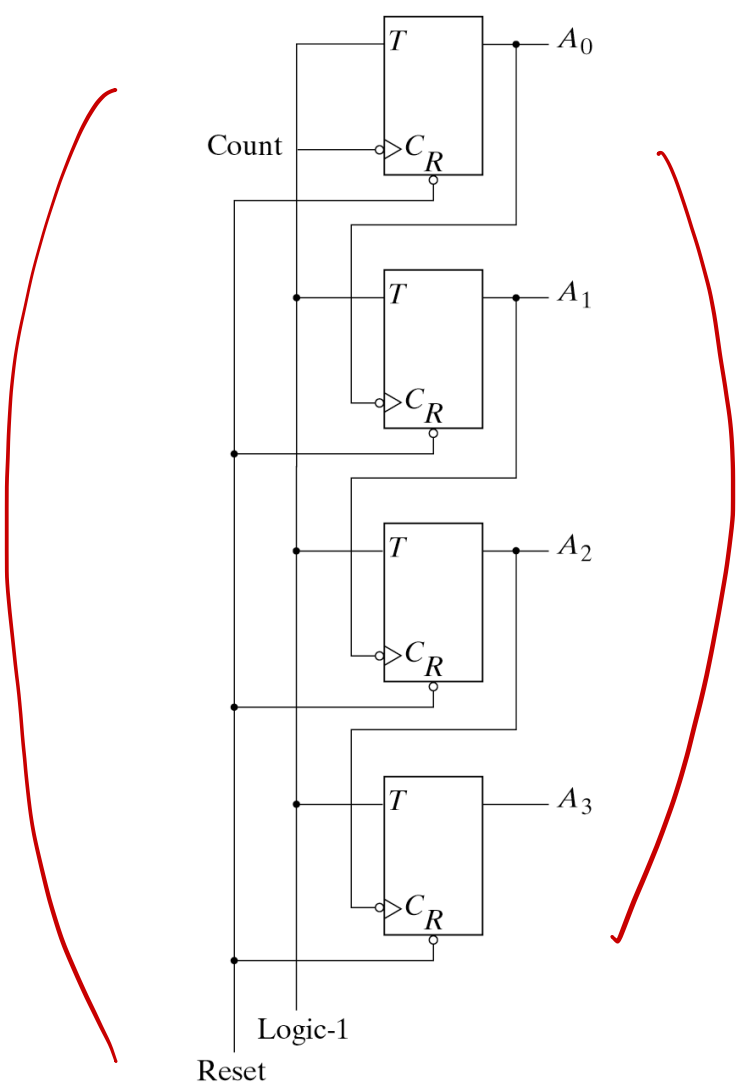
11 → 10 → 01 → 00 → 11 ...



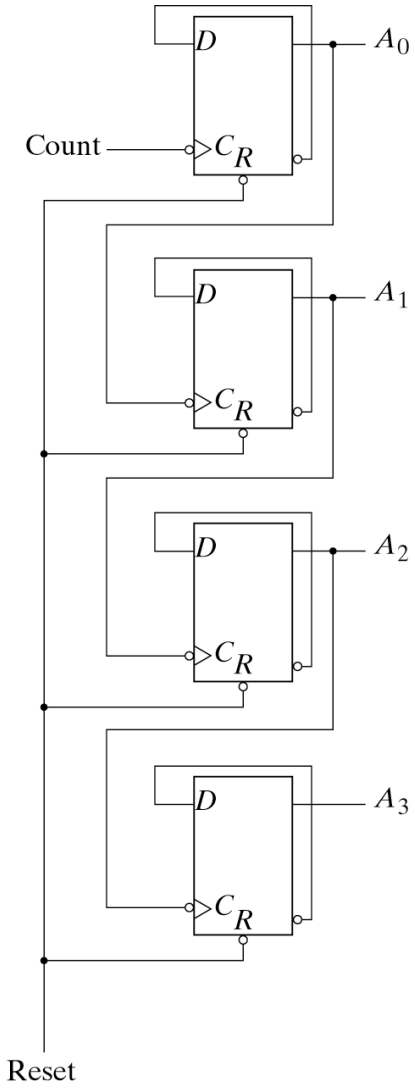
$A^+ = \bar{A}$

$B^+ = \bar{B}$

4-Bit Ripple Up Counter (1/2)



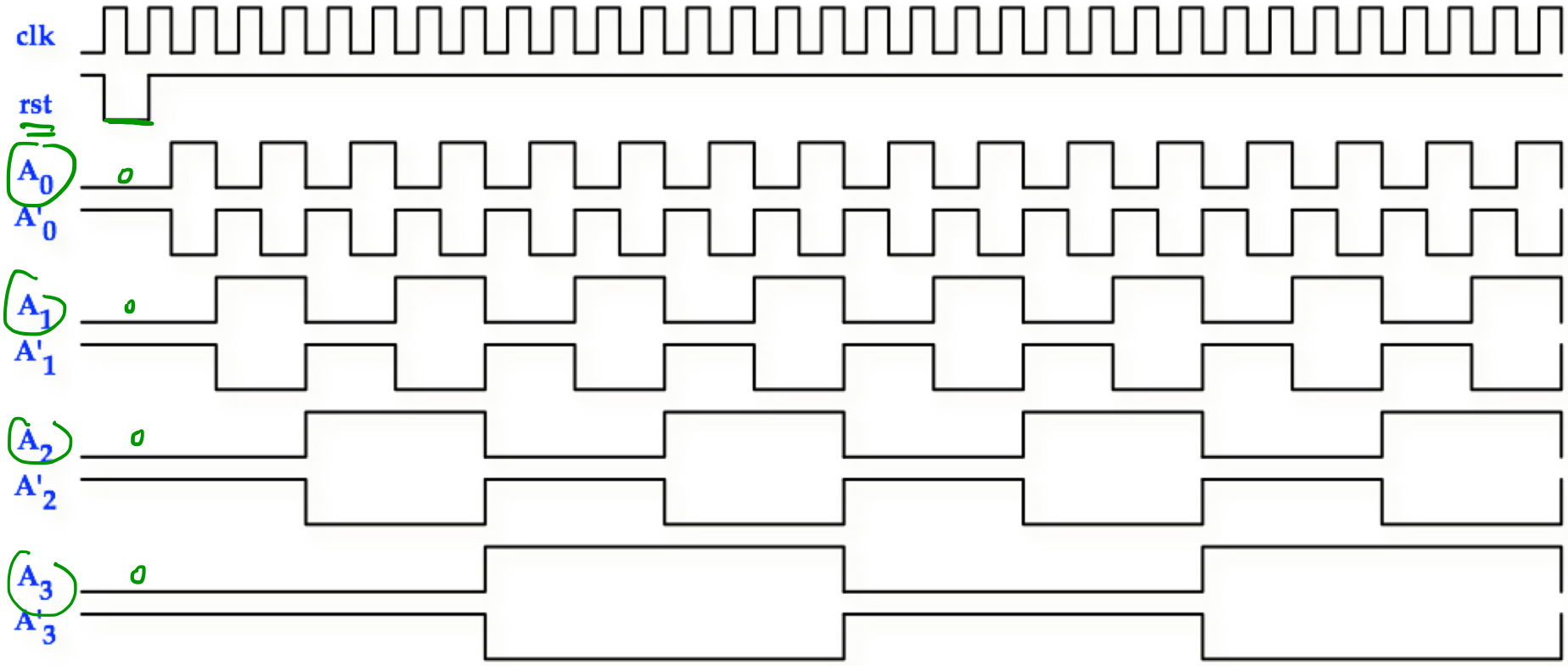
(a) With T flip-flops



(b) With D flip-flops

4-Bit Ripple Up Counter (2/2)

- Use DFFs



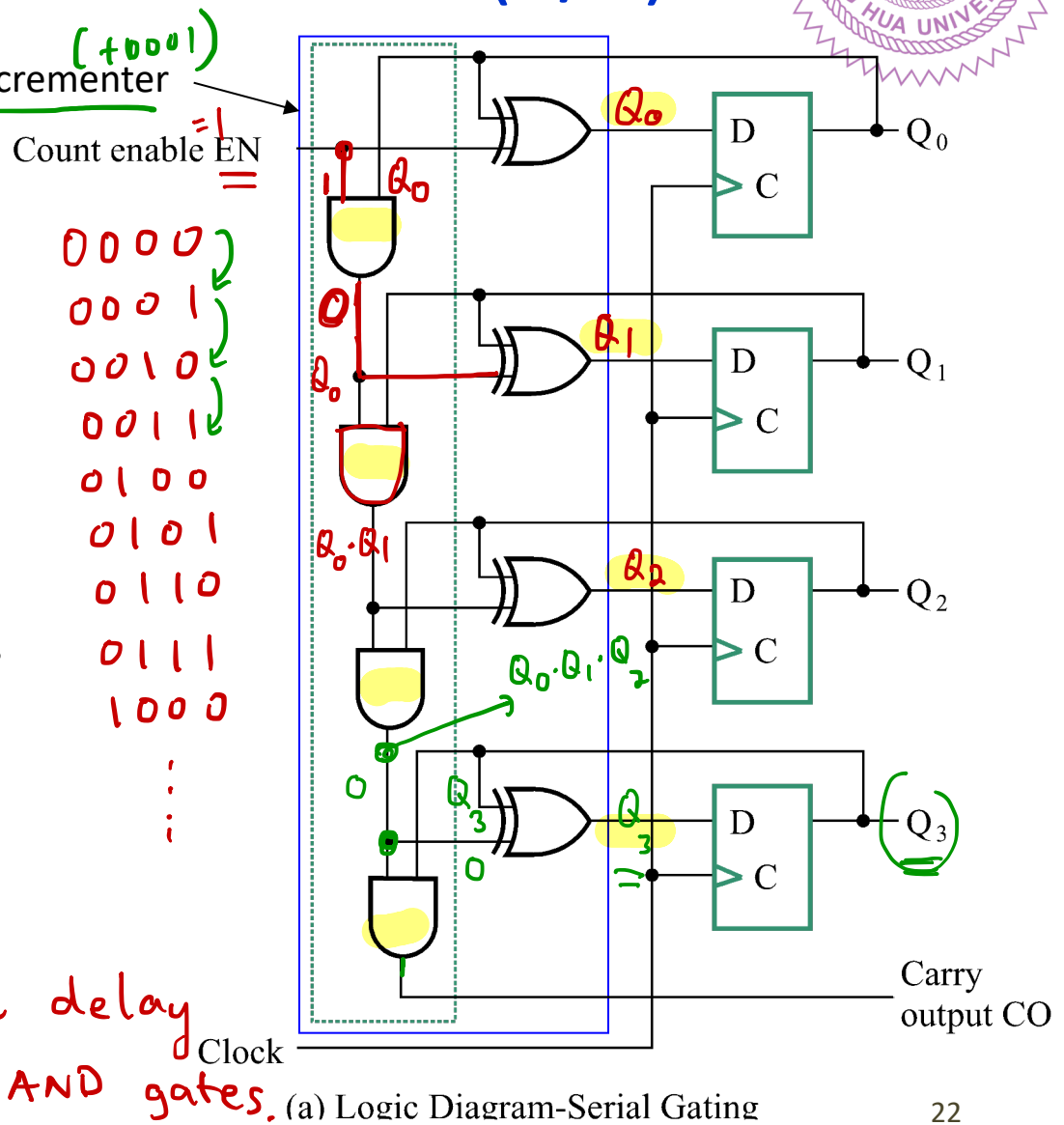
2.

Synchronous Counters (1/2)

- Clock is applied to all FFs.
- Formed by incrementer and DFFs.
- AND chain causes complement of a bit if all bits toward LSB from it equal 1.
- XOR complements each bit.
- Count enable (EN)

EN=1, count

EN=0, hold



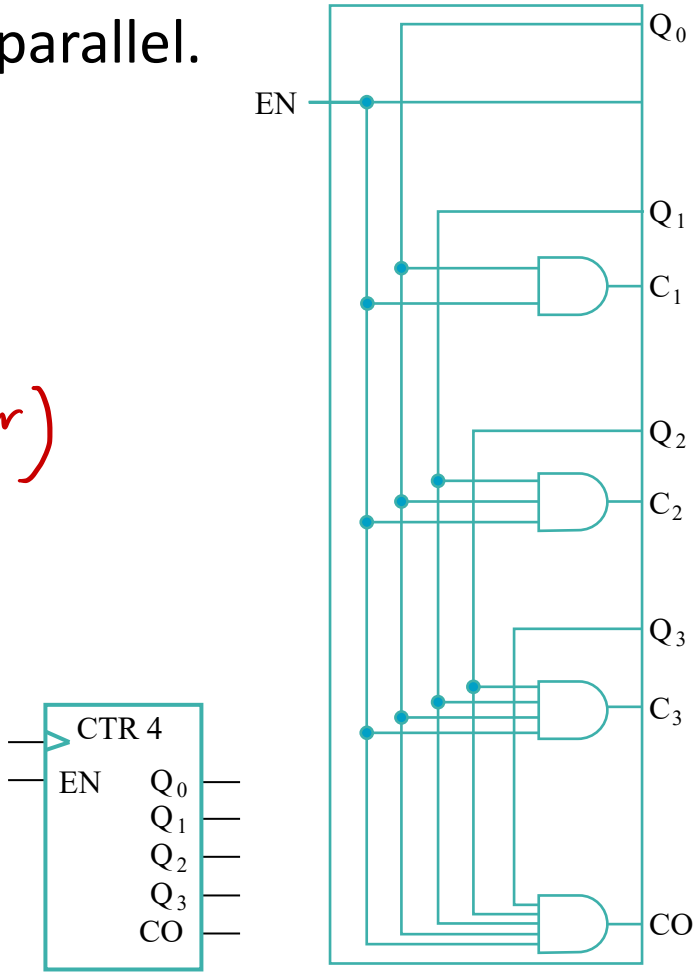
(a) Logic Diagram-Serial Gating

- Serial gating: large path delay toward MSB due to serial AND gates.

Synchronous Counters (2/2)

- Replace AND carry chain with ANDs in parallel.
- EN applied at each bit.
- Parallel gating reduce path delay.

Practice. sync 2-bit up counter (5 chapter)

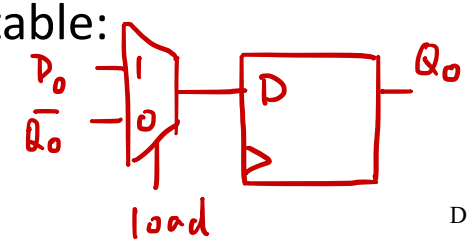


Symbol

Counter with Parallel Load

- Add path for input data.
  - Enabled for Load = 1
- Add logic to
  - Disable count logic for Load = 1
  - Disable feedback from outputs for Load = 1
  - Enable count logic for Load = 0 and Count = 1
- The resulting function table:

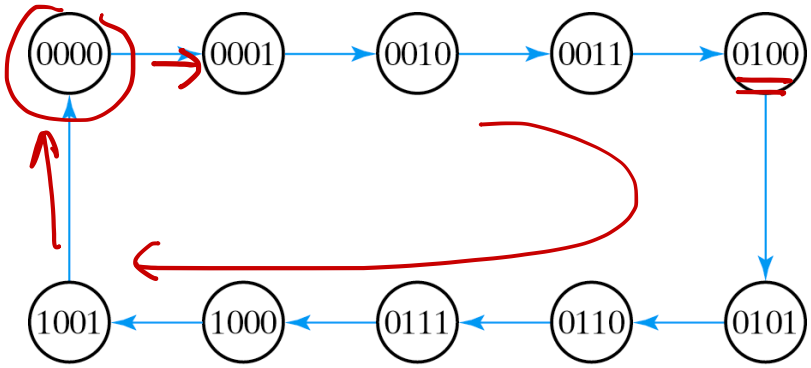
Load	Count	operation
1	x	load
0	1	count
0	0	hold



3.

BCD Counter (1/2)

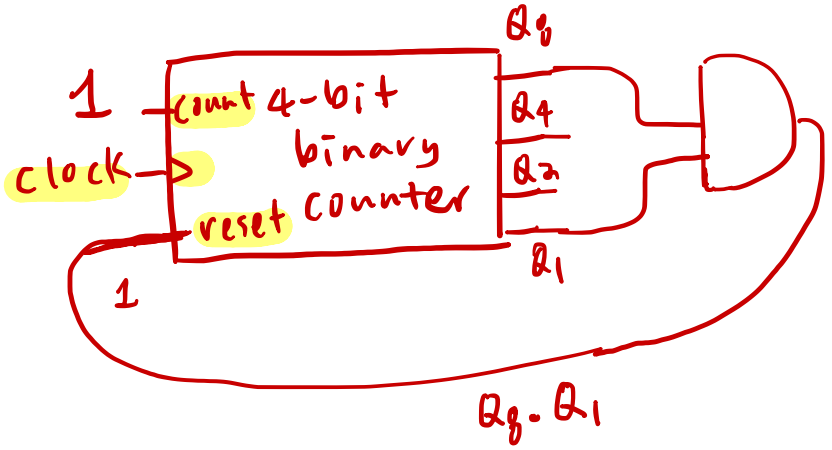
State diagram



State table

Present State				Next State			
Q8	Q4	Q2	Q1	Q8	Q4	Q2	Q1
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

o If a 4-bit up binary counter is available,



BCD Counter (2/2)

Sync. BCD counter

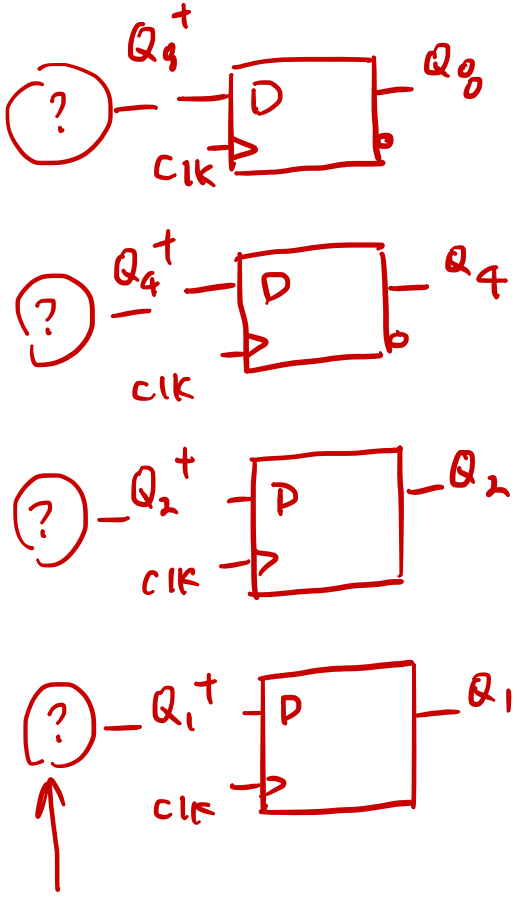
• Practice.

$Q_0^+ =$

$Q_4^+ =$

$Q_2^+ =$

$Q_1^+ =$



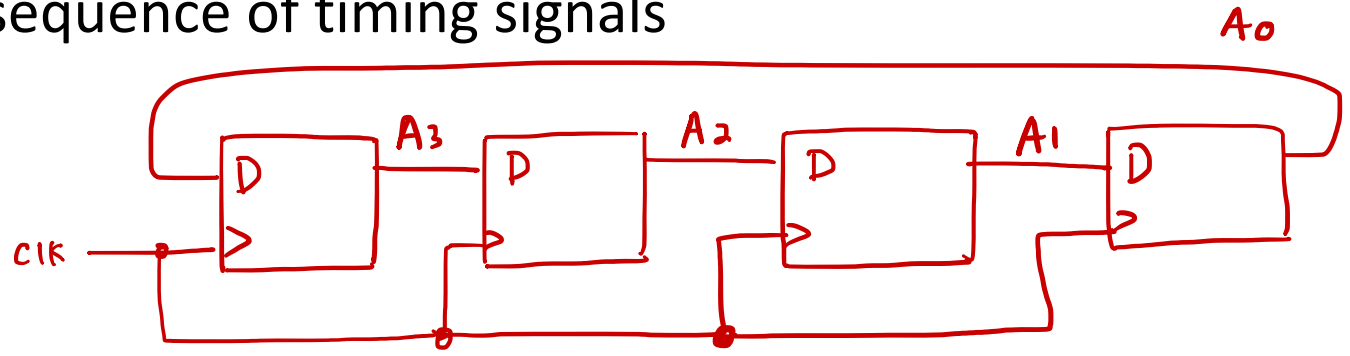
$Q_1^+ = f(Q_8, Q_4, Q_2, Q_1)$

4.

Ring Counter

- A circular shift register with only one flip-flop being set at any particular time, all others are cleared. (initial value 1000...000)
- The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals

	A_3	A_2	A_1	A_0
#1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
#1	1	0	0	0

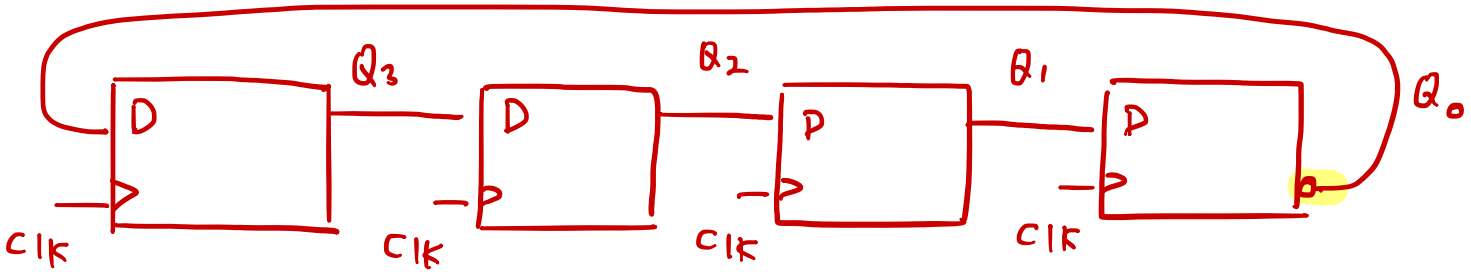


- Advantage: simple implementation
- Disadvantage: only represents n states with n FFs.

5.

Johnson Counter

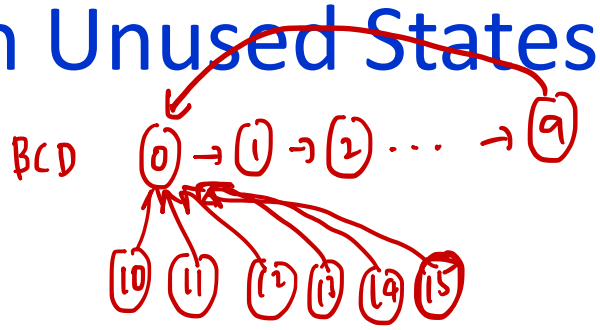
- Johnson counter: modified ring counter that the last output is inverted and fed back as the input of the first FF. Johnson counter circulates a stream of ones followed by zeros.



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

8 states.
(2w states with w FFs)

Counters with Unused States



- n flops => 2^n states
- Unused states
  - States that are not used in specifying the FSM, may be treated as don't-care conditions or may be assigned specific next states.
- Self-correcting counters
  - Ensure that when a circuit enter one of its unused states, it eventually goes into one of the valid states after one or more clock pulses so that it can resume normal operation.
    - Analyze the circuit to determine the next state from an unused state after it is designed.