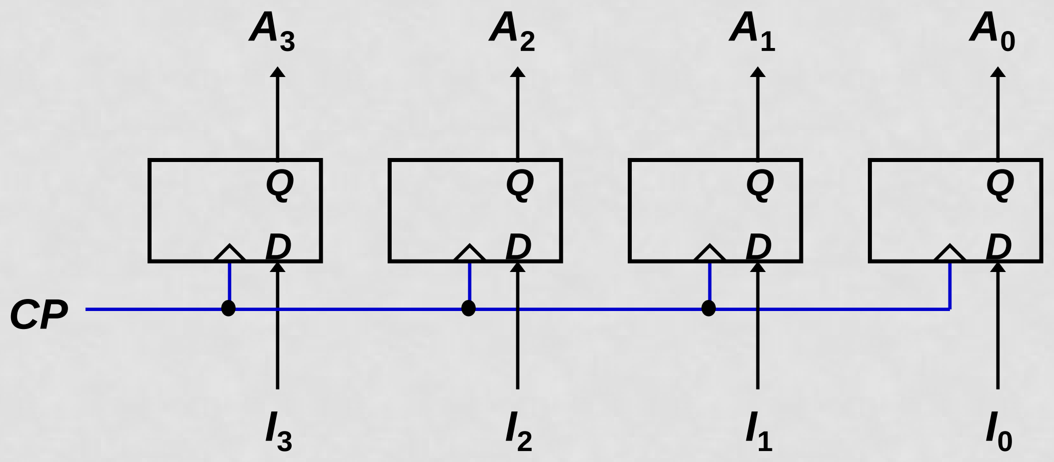
REGISTER & COUNTER

REGISTER

- An *n*-bit register has a group of *n* flip-flops and some logic gates and is capable of storing *n* bits of information.
- The flip-flops store the information while the gates control when and how new information is transferred into the register.
- Some functions of register:
  - retrieve data from register
  - store/load new data into register (serial or parallel)
  - shift the data within register (left or right)

SIMPLE REGISTERS

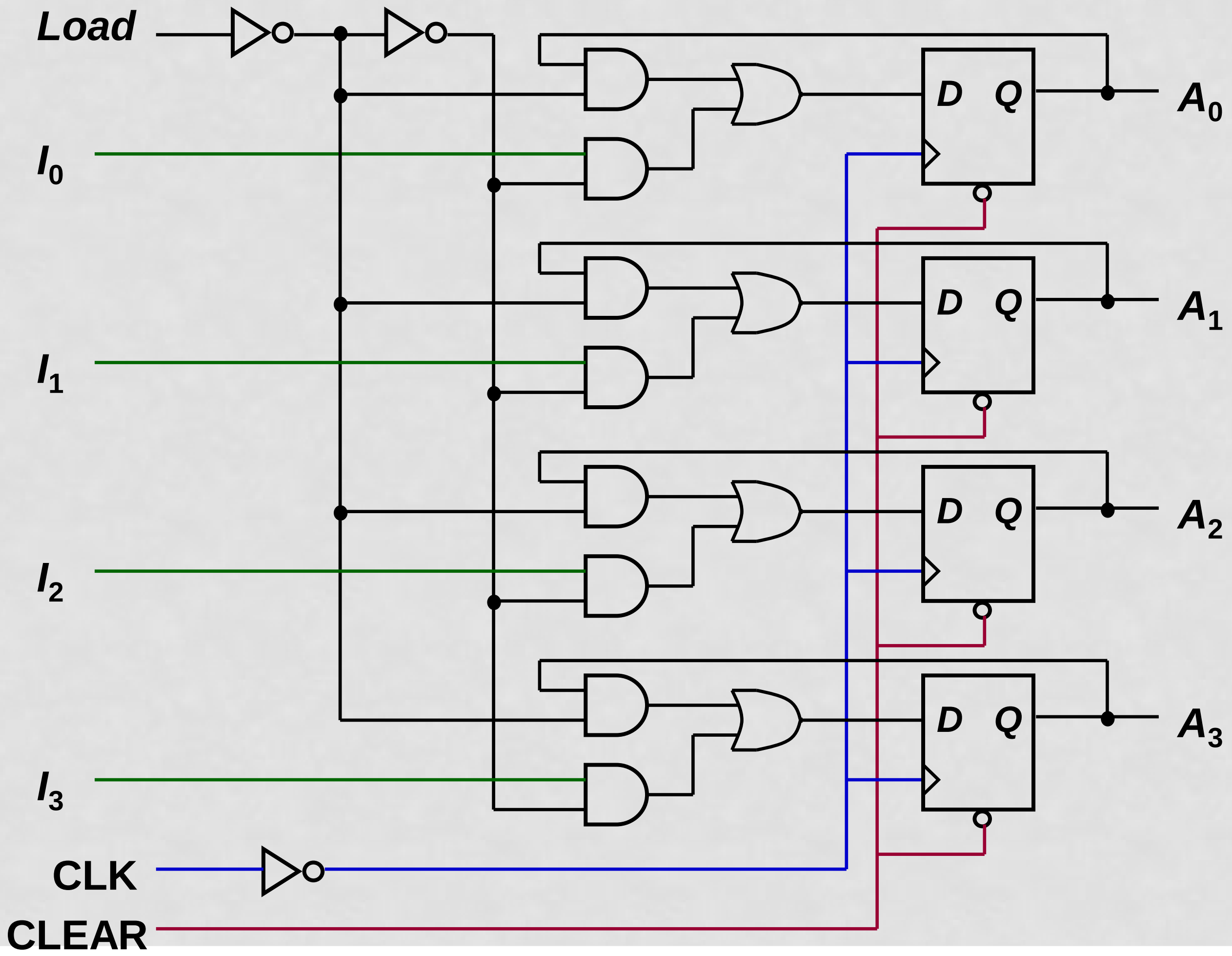
- No external gates.
- Example: A 4-bit register. A new 4-bit data is loaded every clock cycle.



REGISTERS WITH PARALLEL LOAD

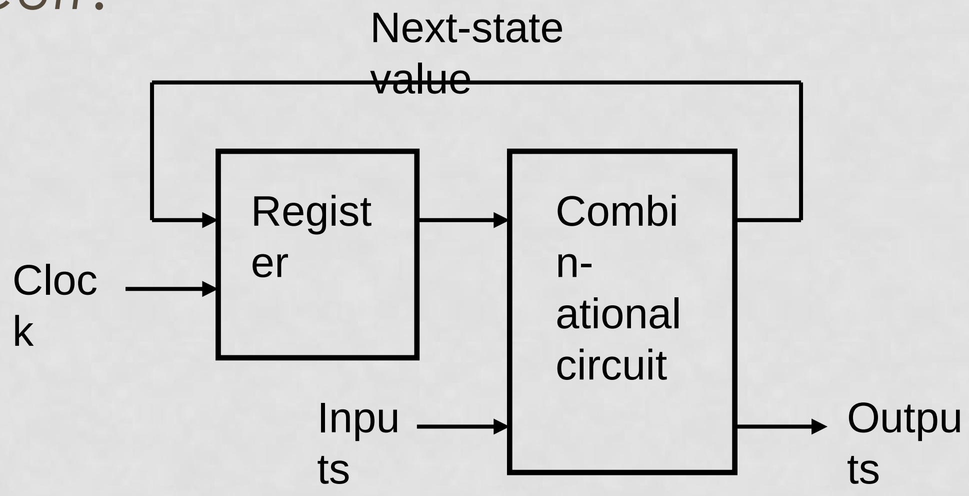
- Instead of loading the register at every clock pulse, we may want to control when to load.
- *Loading* a register: transfer new information into the register. Requires a *load* control input.
- *Parallel loading*: all bits are loaded simultaneously.

REGISTERS WITH PARALLEL LOAD



USING REGISTERS TO IMPLEMENT SEQUENTIAL CIRCUITS

- The external inputs and present states of the register determine the next states of the register and the external outputs, through the combinational circuit.
- The combinational circuit may be implemented by any of the methods covered in *MSI components* and *Programmable Logic Devices*.
- A sequential circuit may consist of a *register* (memory) and a *combinational circuit*.



USING REGISTERS TO IMPLEMENT SEQUENTIAL CIRCUITS

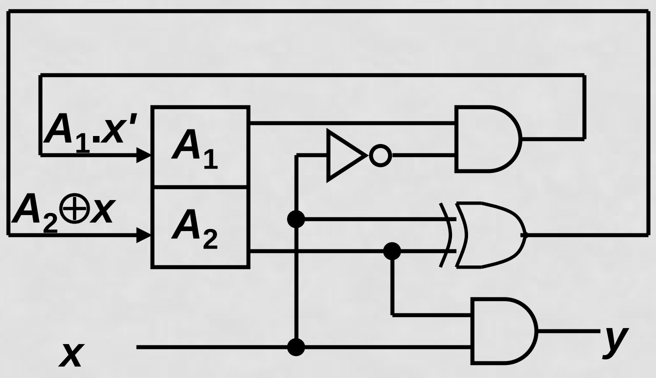
Example 1:

$A_1^+ = \Sigma\, m(4,6) = A_1.x'$

$A_2^+ = \Sigma\, m(1,2,5,6) = A_2.x' + A_2'.x = A_2 \oplus x$

$y = \Sigma\, m(3,7) = A_2.x$

Present state		Input	Next State		Output
A_1	A_2	x	A_1^+	A_2^+	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1

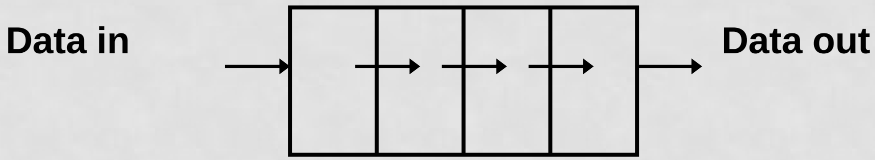


SHIFT REGISTERS

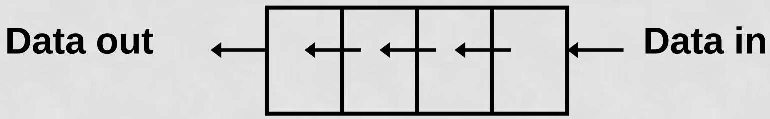
- Another function of a register, besides storage, is to provide for *data movements*.
- Each *stage* (flip-flop) in a shift register represents one bit of storage, and the shifting capability of a register permits the movement of data from stage to stage within the register, or into or out of the register upon application of clock pulses

SHIFT REGISTERS

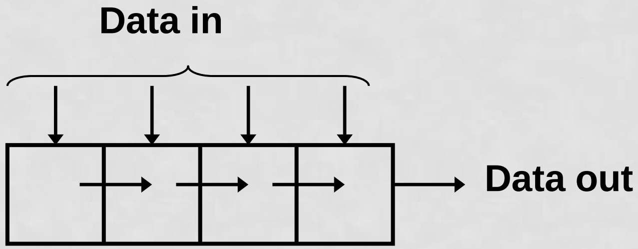
- Basic data movement in shift registers (four bits are used for illustration).



(a) Serial in/shift right/serial out



(b) Serial in/shift left/serial out



(c) Parallel in/serial out

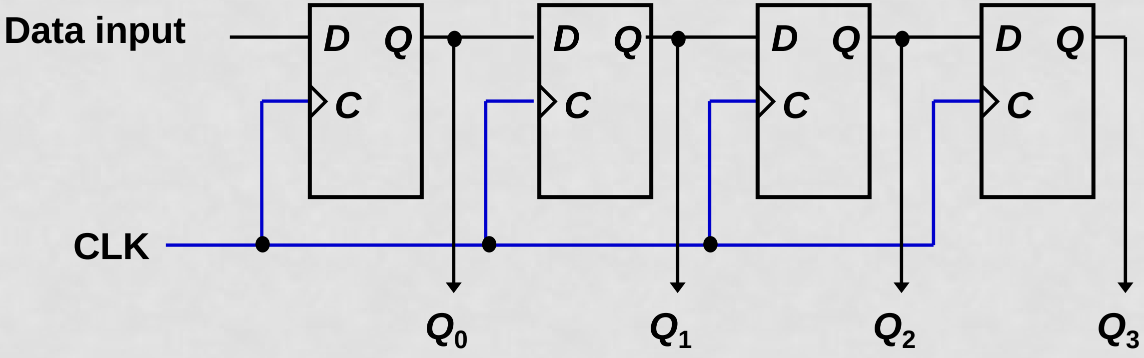
SERIAL IN/SERIAL OUT SHIFT REGISTERS

- Serial-transfer example.

Timing Pulse	Shift register *A*				Shift register *B*				Serial output of *B*
Initial value	1	0	1	1	0	0	1	0	0
After T_1	1	1	0	1	1	0	0	1	1
After T_2	1	1	1	0	1	1	0	0	0
After T_3	0	1	1	1	0	1	1	0	0
After T_4	1	0	1	1	1	0	1	1	1

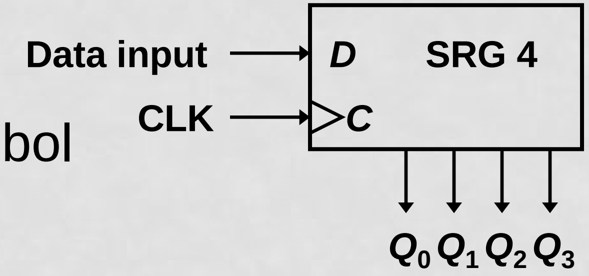
SERIAL IN/PARALLEL OUT SHIFT REGISTERS

- Accepts data serially.
- Outputs of all stages are available simultaneously.



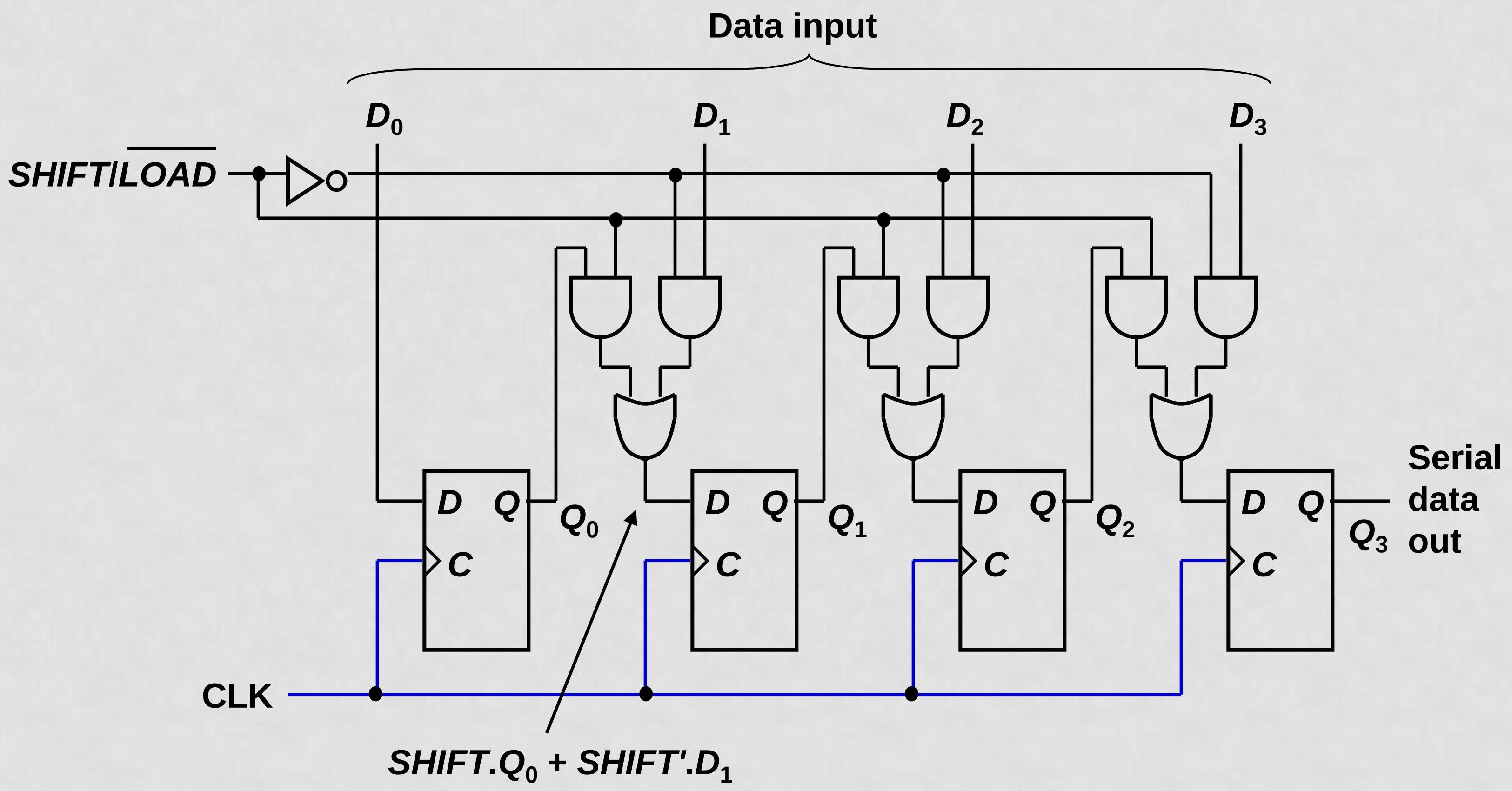
Logic symbol

Logic symbol



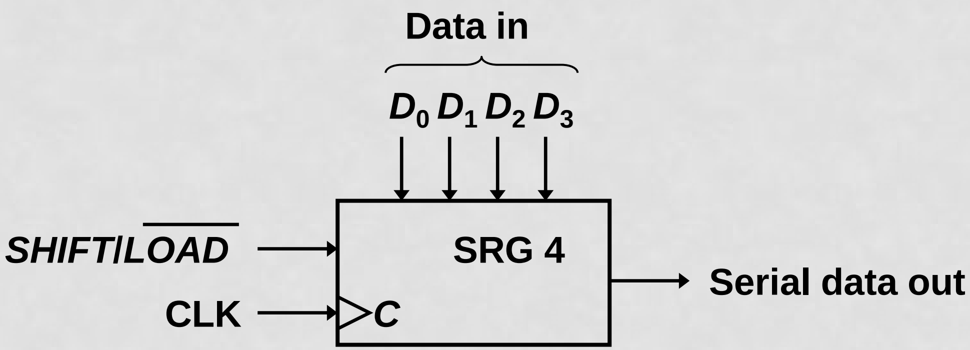
PARALLEL IN/SERIAL OUT SHIFT REGISTERS

- Bits are entered simultaneously, but output is serial.



PARALLEL IN/SERIAL OUT SHIFT REGISTERS

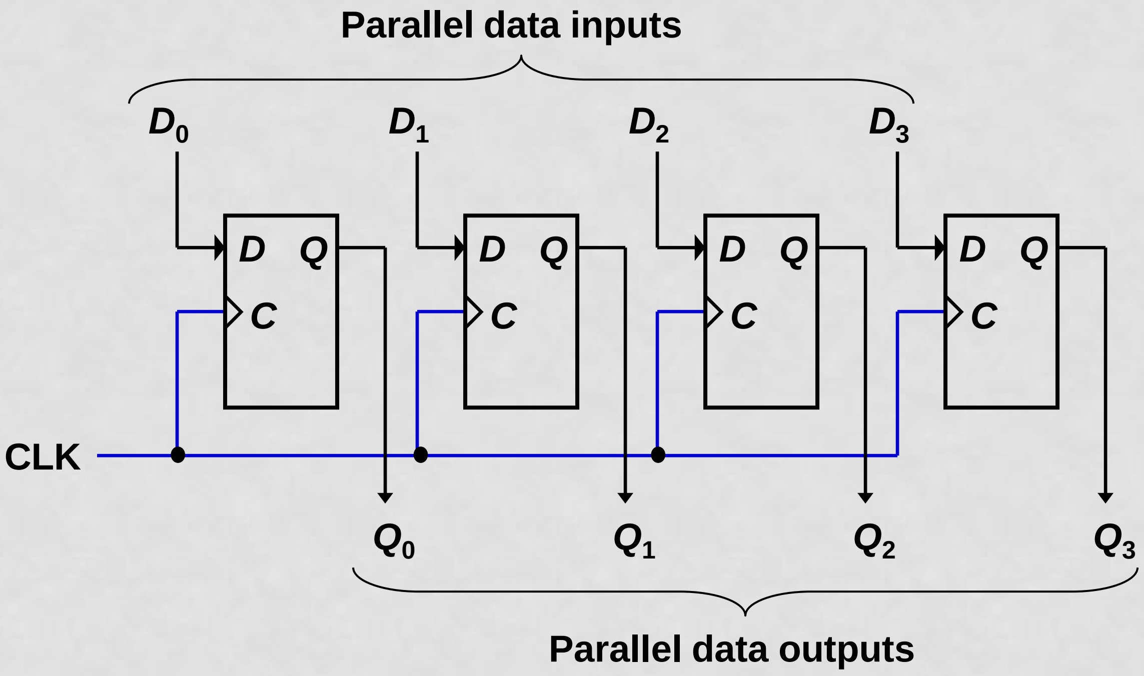
- Bits are entered simultaneously, but output is serial.



Logic symbol

PARALLEL IN/PARALLEL OUT SHIFT REGISTERS

- Simultaneous input and output of all data bits.



INTRODUCTION: COUNTERS

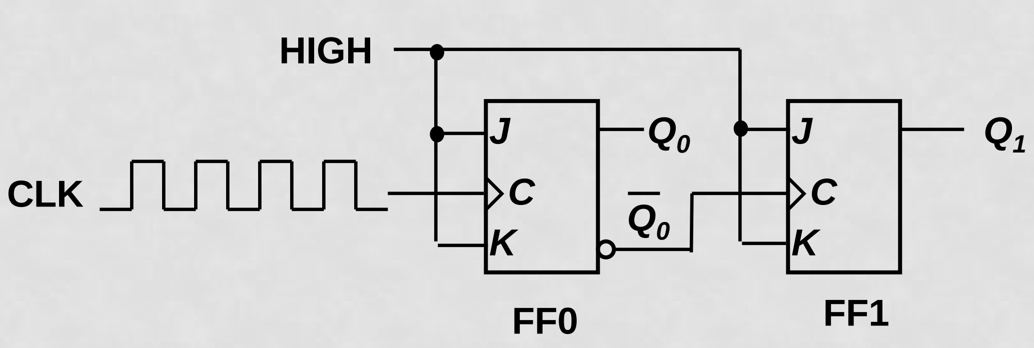
- Counters are circuits that cycle through a specified number of states.
- Two types of counters:
  - synchronous (parallel) counters
  - asynchronous (ripple) counters
- Ripple counters allow some flip-flop outputs to be used as a source of clock for other flip-flops.
- Synchronous counters apply the same clock to all flip-flops.

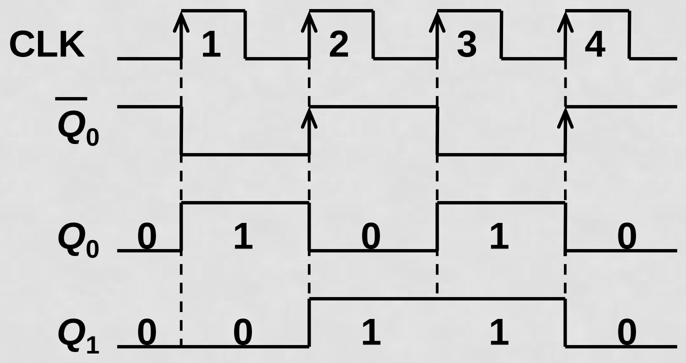
ASYNCHRONOUS (RIPPLE) COUNTERS

- Asynchronous counters: the flip-flops do not change states at exactly the same time as they do not have a common clock pulse.
- Also known as ripple counters, as the input clock pulse "ripples" through the counter – cumulative delay is a drawback.
- *n* flip-flops $\rightarrow$ a MOD (modulus) 2^n counter. (Note: A MOD-x counter cycles through x states.)
- Output of the last flip-flop (MSB) divides the input clock frequency by the MOD number of the counter, hence a counter is also a *frequency divider*.

ASYNCHRONOUS (RIPPLE) COUNTERS

- Example: 2-bit ripple binary counter.
- Output of one flip-flop is connected to the clock input of the next more-significant flip-flop.



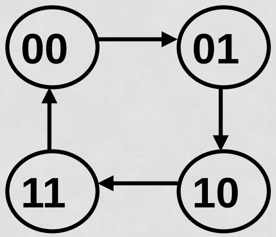


Timing diagram

00 → 01 → 10 → 11 → 00 ...

SYNCHRONOUS (PARALLEL) COUNTERS

- Synchronous (parallel) counters: the flip-flops are clocked at the same time by a common clock pulse.
- We can design these counters using the sequential logic design process.
- Example: 2-bit synchronous binary counter (using T flip-flops, or JK flip-flops with identical J,K inputs).



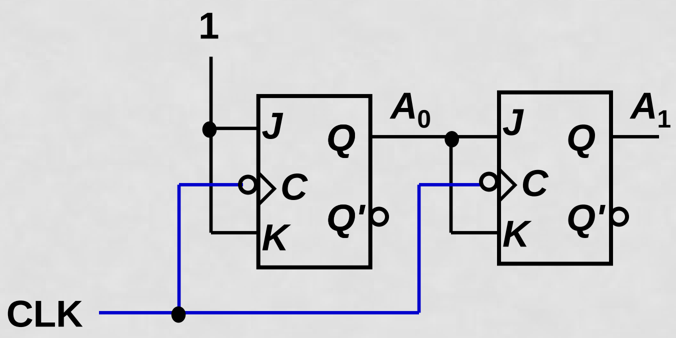
Present state		Next state		Flip-flop inputs	
A_1	A_0	A_1^+	A_0^+	TA_1	TA_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1

SYNCHRONOUS (PARALLEL) COUNTERS

- Example: 2-bit synchronous binary counter (using T flip-flops, or JK flip-flops with identical J,K inputs).

Present state		Next state		Flip-flop inputs	
A_1	A_0	A_1^+	A_0^+	TA_1	TA_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1

$TA_1 = A_0$
$TA_0 = 1$



THANK YOU.