In the Name of God

Digital Circuit Design

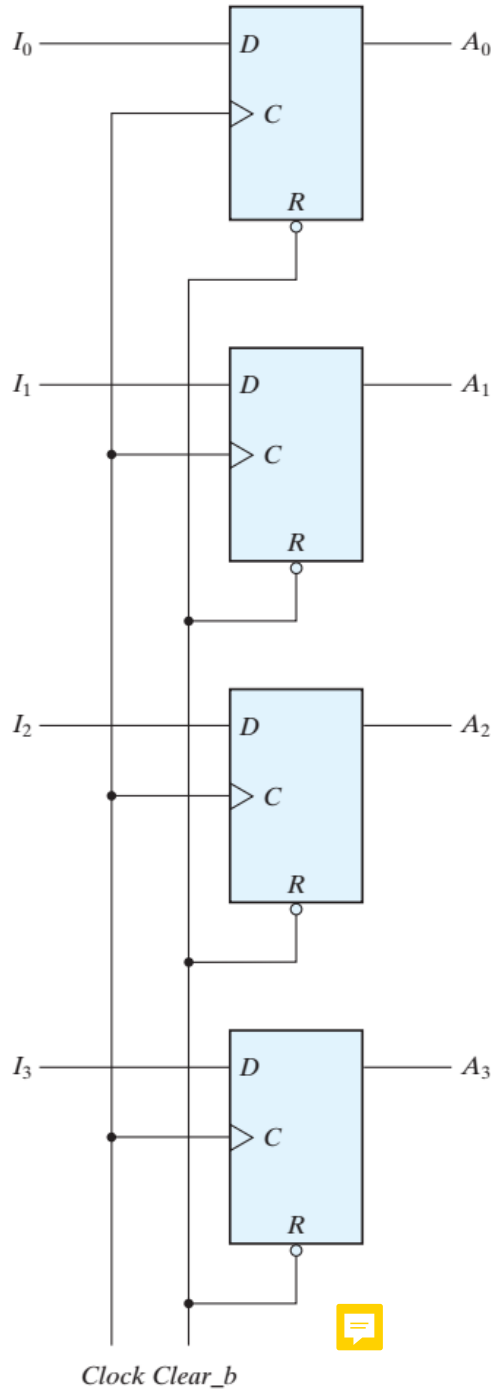
Chapter 6:

Registers & Counters

Dr. M. Beigizadeh

✓Registers

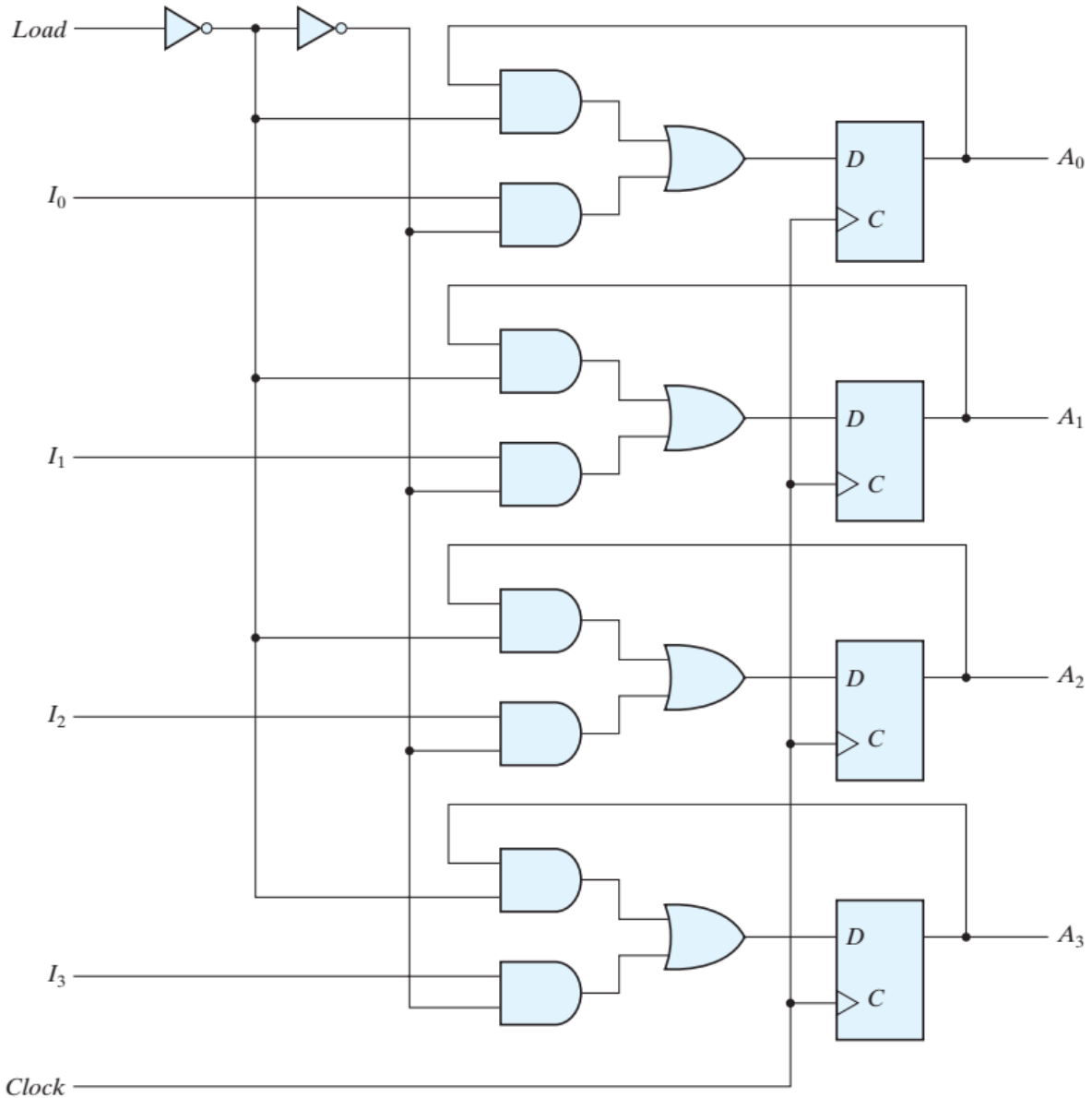
- A group of flip-flops, each one of which is capable of storing one bit, together with gates that affect their operation
- An n-bit register consists of n flip-flops capable of storing n bits
- The simplest register consists of only flip-flops, without any gates
- The common clock input triggers all flip-flops and the binary data available at the four inputs are transferred into the register
- The four outputs can be sampled at any time to obtain the binary information stored in the register



✓Registers

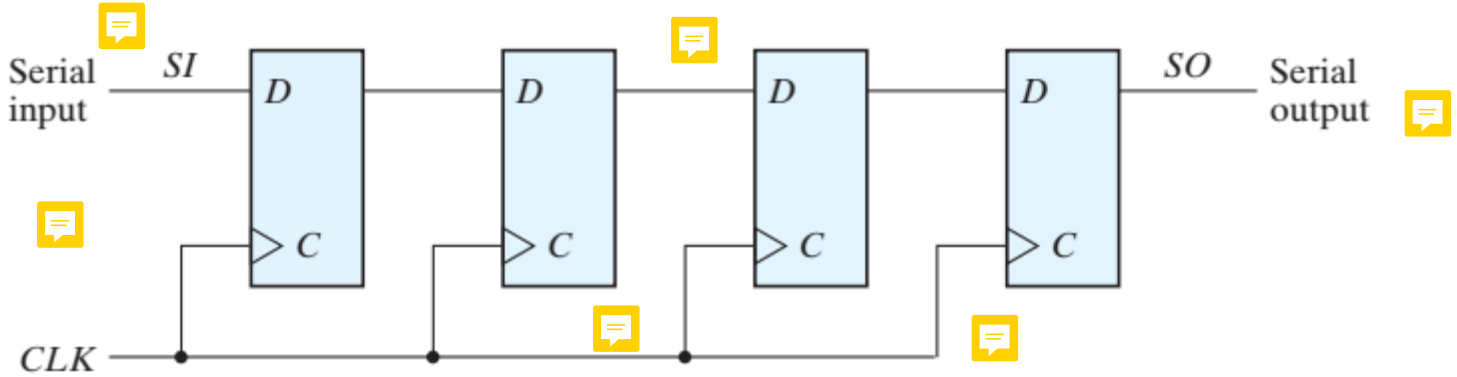
➢ *Register with parallel load*

- The transfer of new data into a register is referred to as *loading* or *updating* the register
- If all bits are loaded simultaneously with a common clock pulse, we say that the loading is done in *parallel*
- The load input determines whether the next pulse will accept new data or leave the data



✓Shift Registers

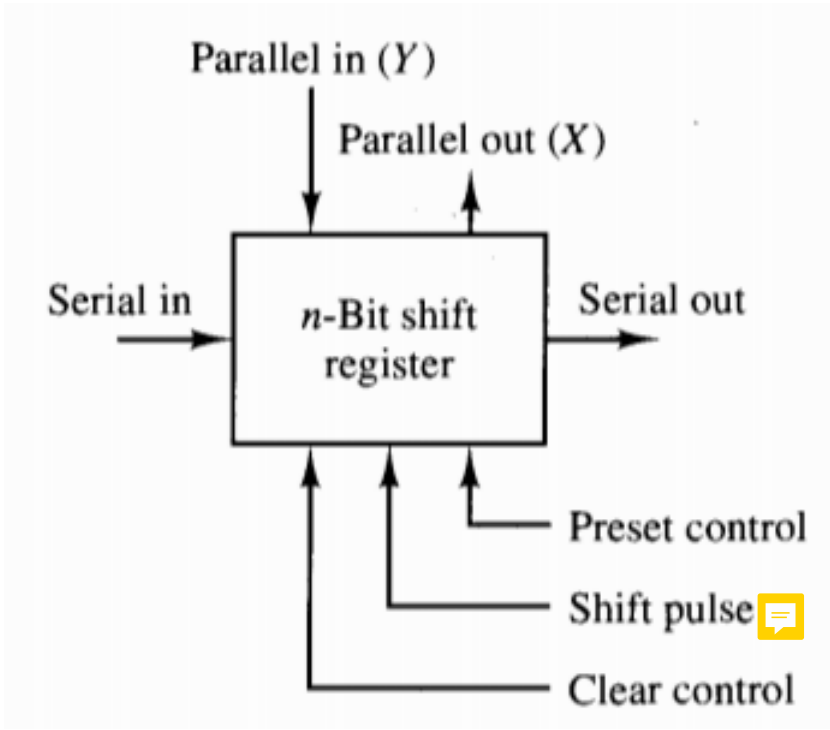
- A Shift register is constructed from flip-flops that shift the data bits to the right or left
- Flip-flops are placed in cascade, with the output of one flip-flop connected to the input of the next flip-flop

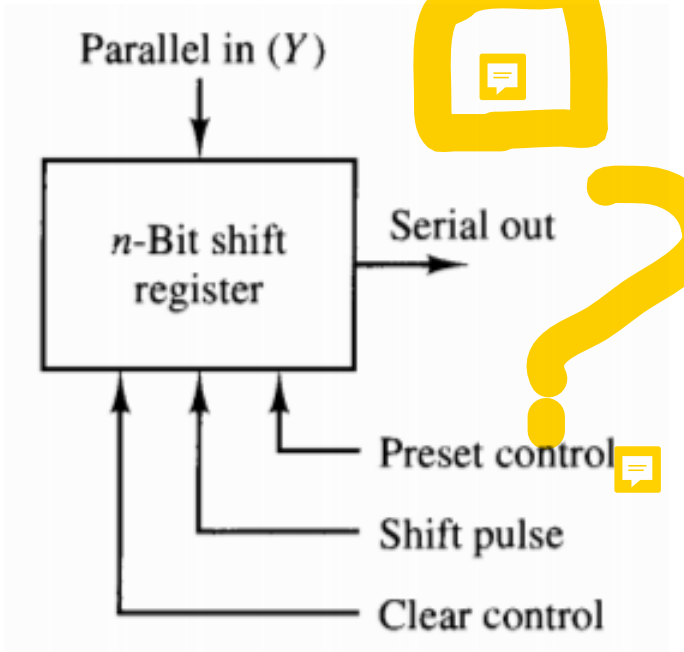


- Each clock pulse shifts the contents of the register one bit position to the right
- This configuration performs *serial-in*, *serial-out* function

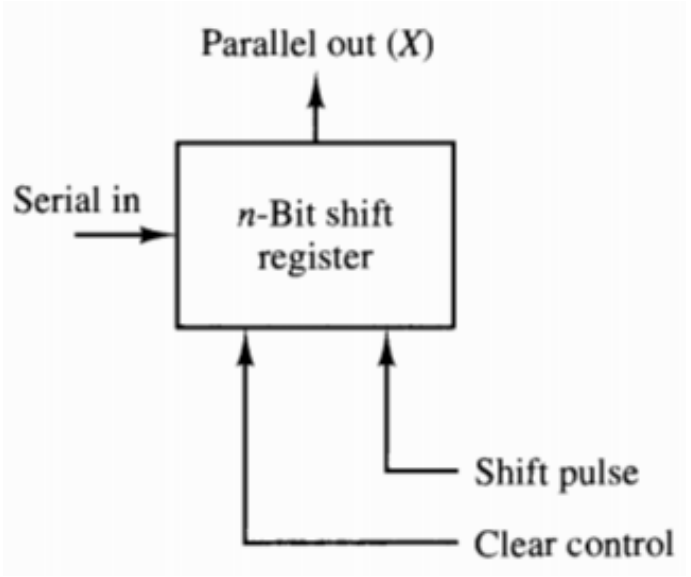
✓Shift Registers

❖ Generic shift register





1) Clear
2) Parallel in
3) Preset
4) Serial out



1) Clear
2) Serial in
3) Parallel out

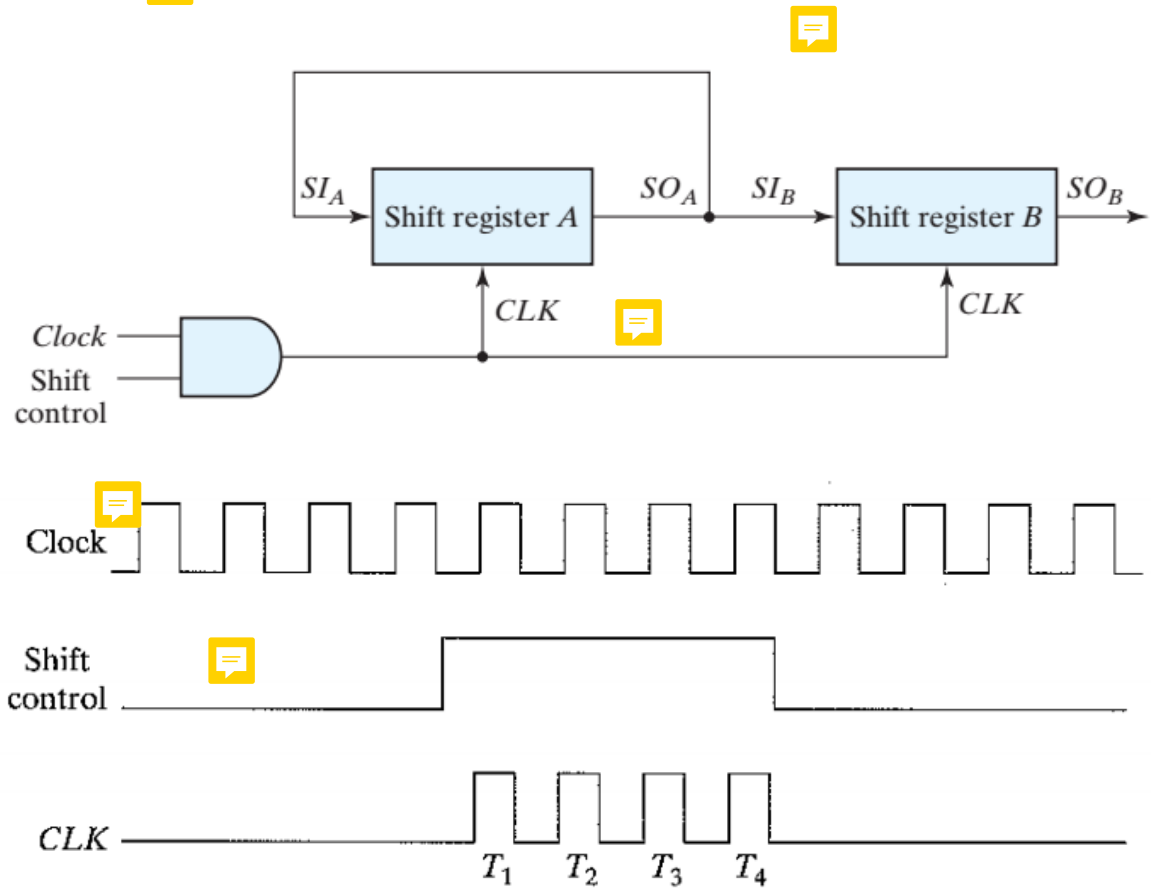
✓Shift Registers

➢ *Serial Transfer:* data is transferred one bit at a time

❖ A serial transfer of data from register A to register B is done with shift registers

❖ The serial output of register A is connected to the serial input of register B

❖ To prevent the loss of data stored in the source register, we connect the serial output of register A to its serial input



❖ The shift control input determines when and how many times the registers are shifted

✓Shift Registers

➢ Serial Transfer

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

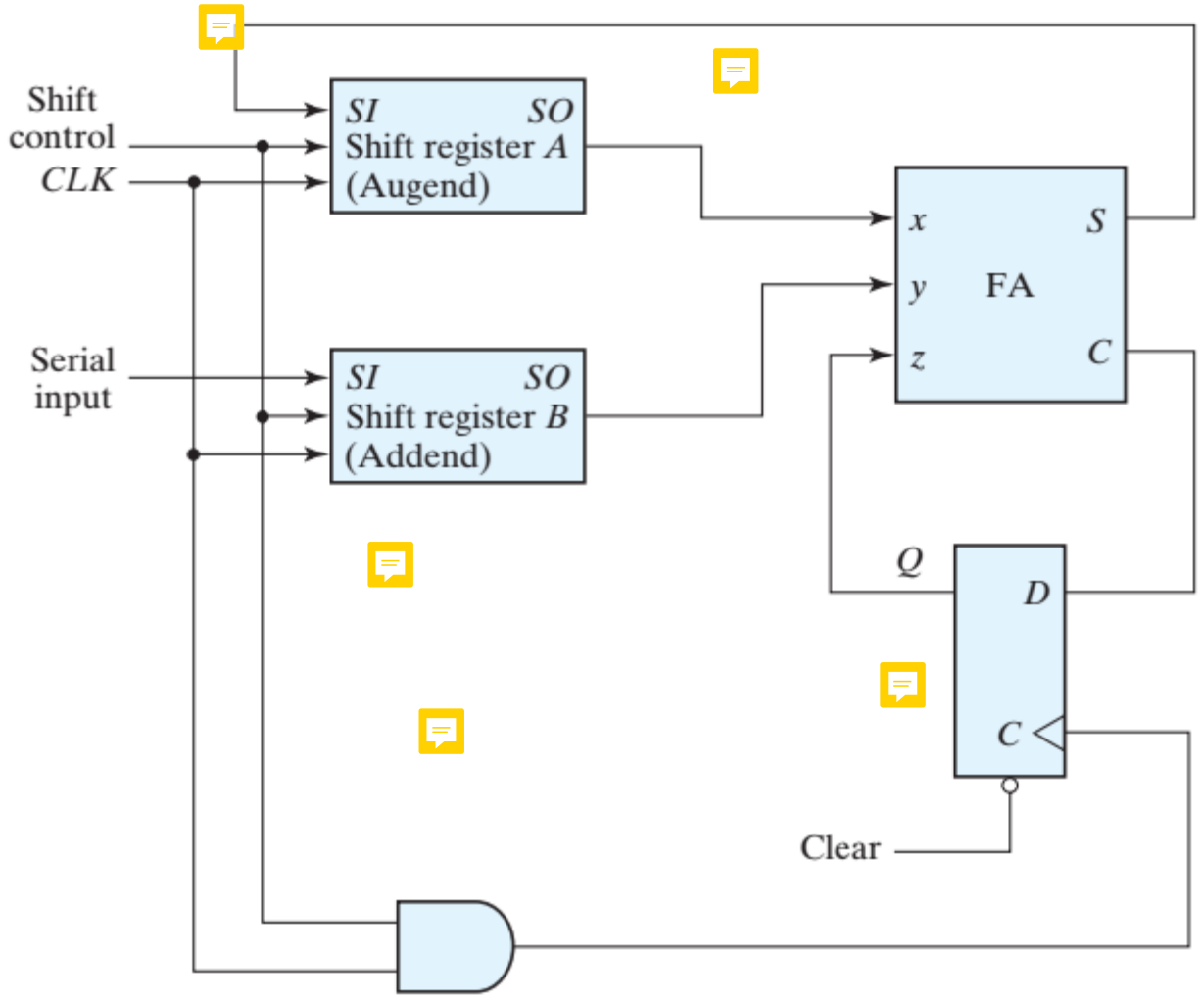
- ❖ In the serial mode of operation:
  - ✓The registers have a single serial input and a single serial output
  - ✓The information is transferred one bit at a time while the registers are shifted in the same direction
- ❖ In the parallel mode of operation, information is available from all bits of a register and all bits can be transferred simultaneously during one clock pulse

Serial mode operations are slower, but require fewer hardware components

✓Shift Registers

➢ Serial Addition

- ❖ Register A is used to store both the augend and the sum bits
- ❖ A serial input of register B can be used to transfer a new binary number
- ❖ *For each succeeding clock pulse:*
  - ✓A new sum bit is transferred to A
  - ✓A new carry is transferred to Q
  - ✓Both registers are shifted once to the right

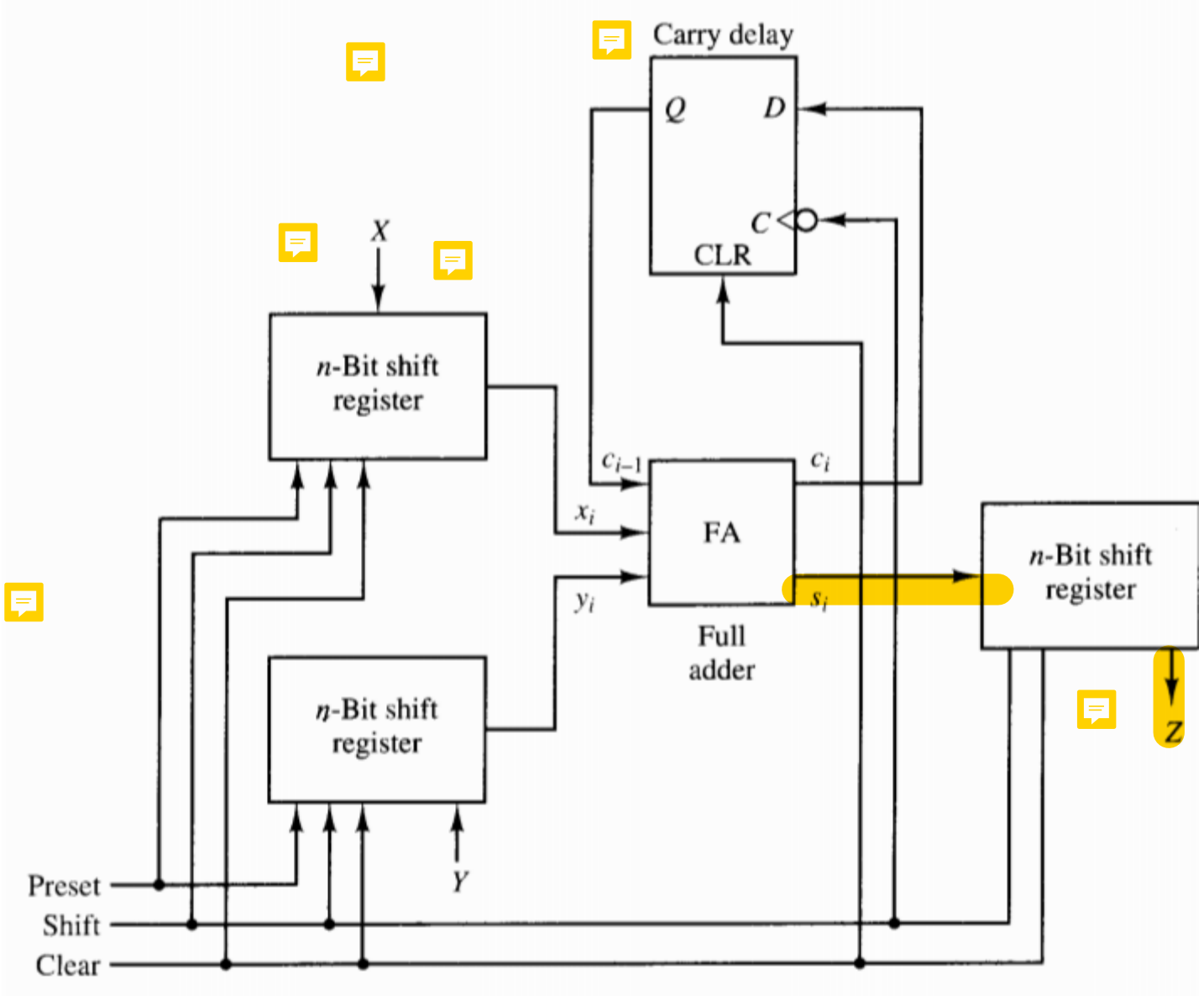


✓Shift Registers

➢ Serial Addition

❖ Two parallel-in, serial-out shift registers are used to accept the parallel data X and Y

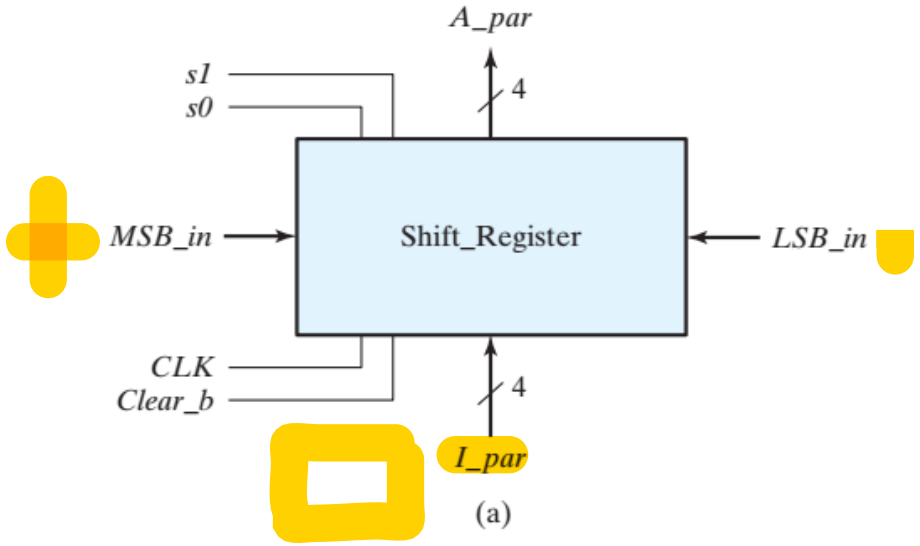
❖ A serial-in, parallel-out shift register is used to store the results

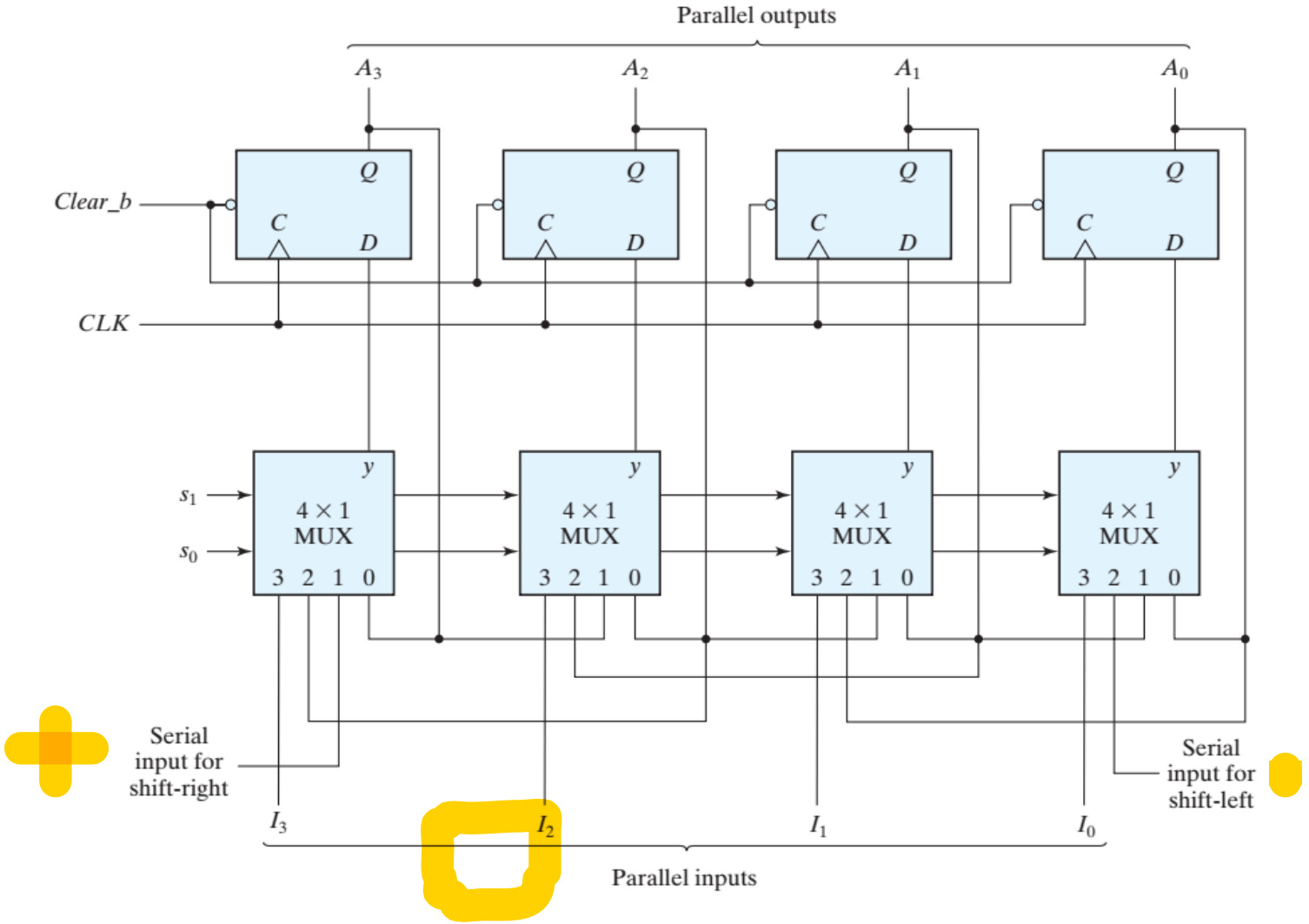


✓Shift Registers

- *Universal Shift Register:* A register which has both shifts and parallel-load capabilities

Mode Control		
s_1	s_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load





✓Shift Registers

- Shift registers are often used to interface digital systems suited remotely between two points
- If distance is far, it will be expensive to use n lines to transmit the n bits in parallel
- The transmitter accepts the n-bit data in parallel and then transmits the data serially along the common line
- The receiver accepts the data serially into a shift register
- When all n bits are received, they can be taken from the output of the register in parallel

✓Counters

- Counters can be designed to generate any desired sequence of states
- *Modulo-N counters :* Counters that can count from state 0 through N-1 and then cycle back to state 0

➢ Synchronous Counter

✓ A common clock is applied to the all flip-flops

- **Binary Counter**

✓A counter that follows the binary number sequence

✓An n-bit binary counter consist of *n* flip-flops and can count in binary from 0 through 2^n-1

✓N is equal to 2^n, where *n* is the number of counter stages

✓Synchronous Counters

000
001
010
011
100

❖ Binary Counter

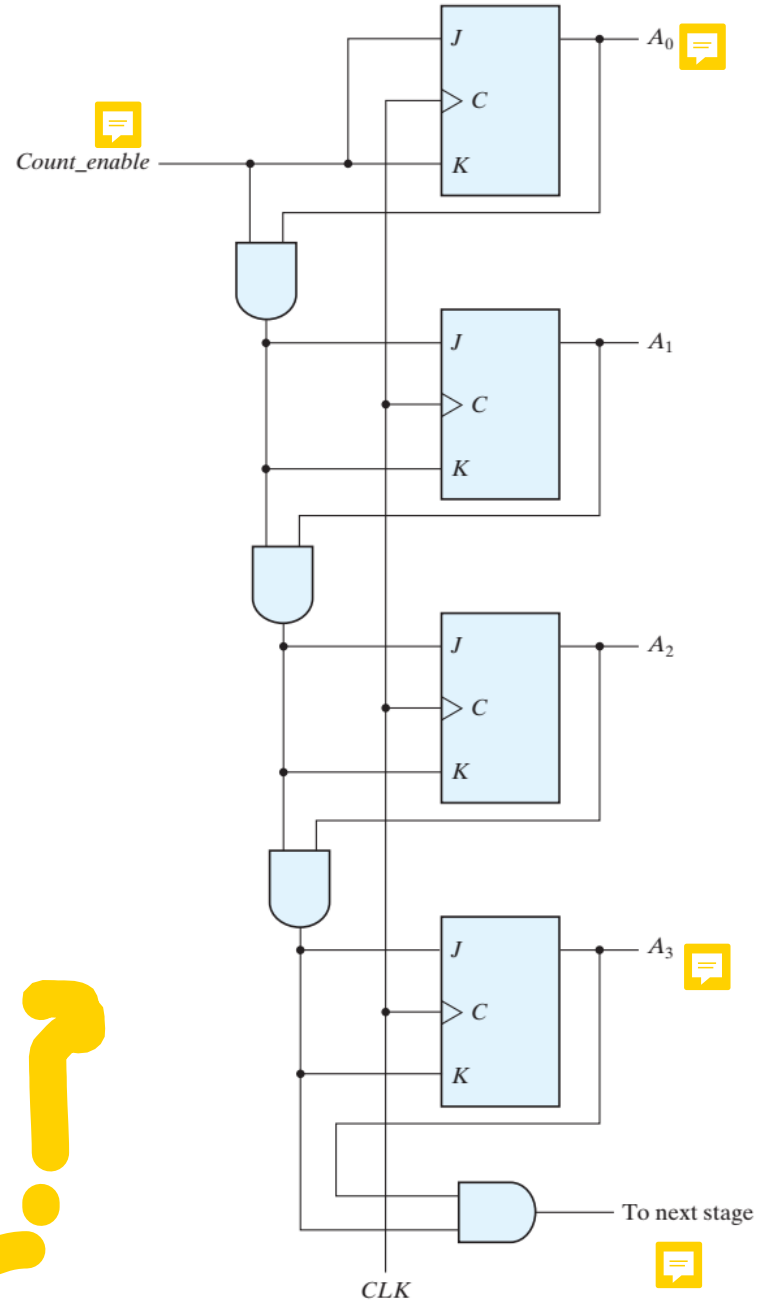
✓The flip-flop in the least significant position is complemented with every pulse

✓A flip-flop in any other position is complemented when all the bits in the lower significant positions are equal to 1

✓The counter is enabled with the count enable input

✓The counter can be extended to any number of stages

✓The complementing flip-flops in a binary counter can be of either the JK type, the T type or the D type with XOR gates



✓Synchronous Counters

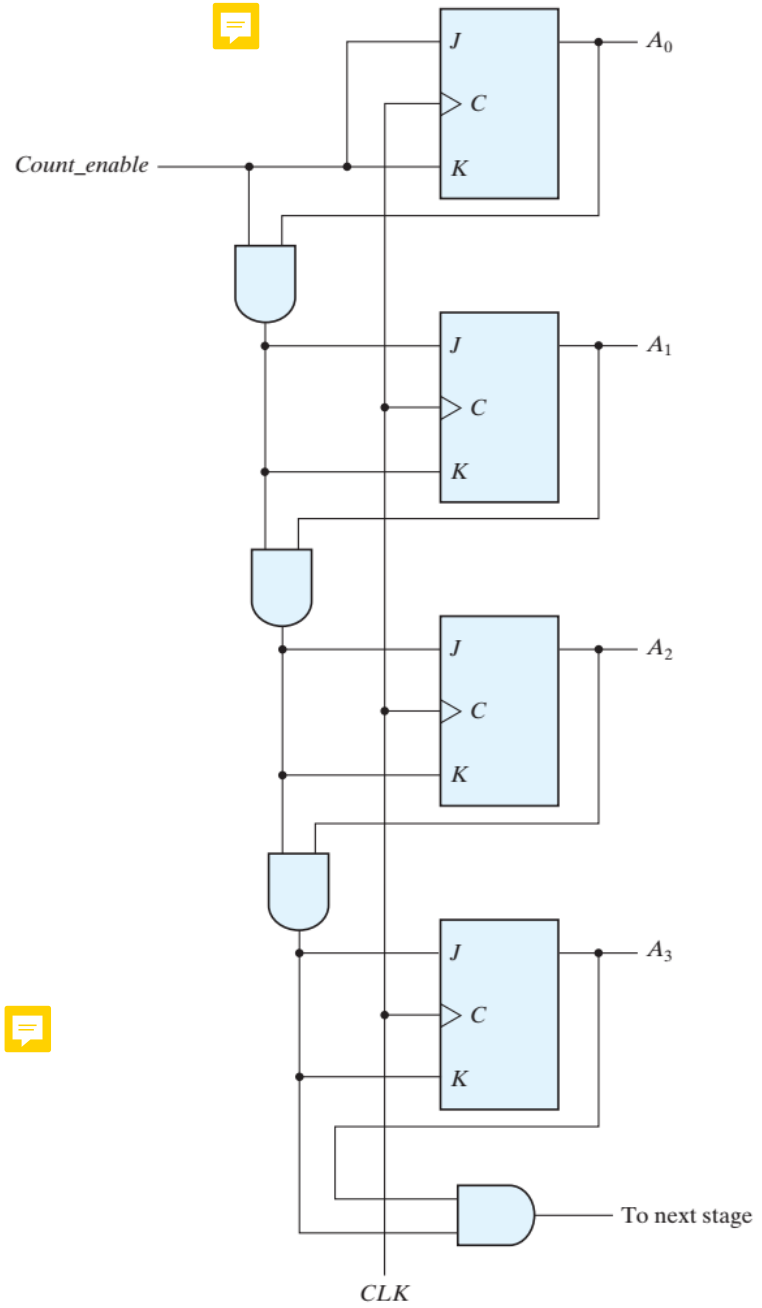
➢ Up-Down Binary Counter

❖*Countdown Binary Counter*

✓The bit in the least significant position is complemented with each pulse

✓A bit in any other position is complemented when all the bits in the lower significant positions are equal to 0

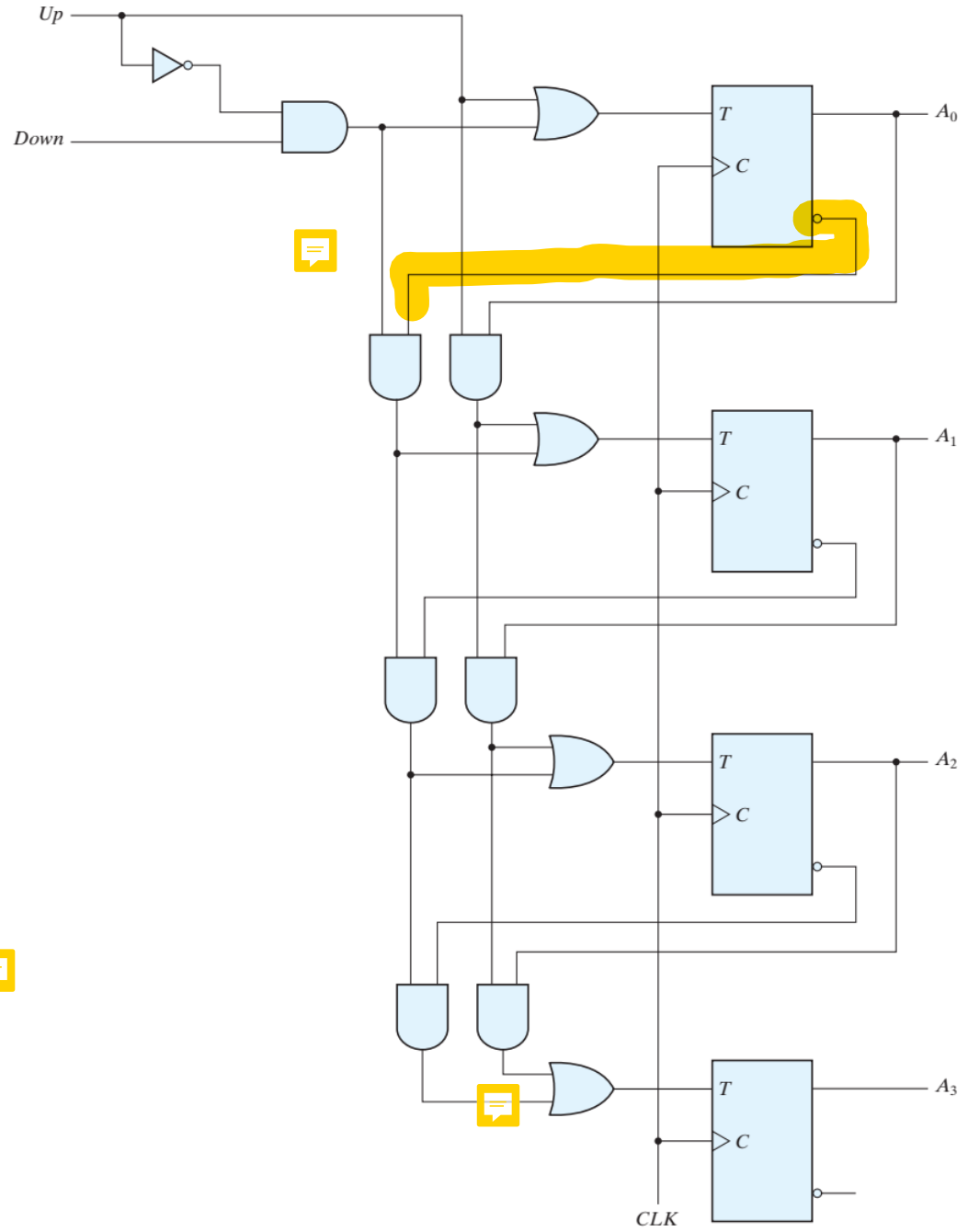
✓The circuit is similar to binary counter, except that the inputs to the AND gates must be come from the complemented outputs, instead of the normal outputs, of the previous flip-flops



✓Synchronous Counters

➢ Up-Down Binary Counter

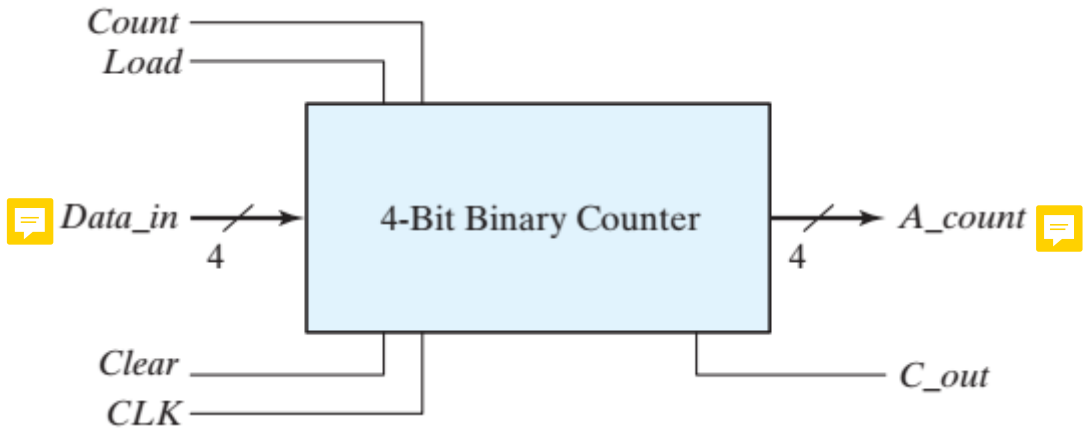
- ❖ When the up input is 1, the circuit counts up
- ❖ When the down input is 1 and the up input is 0 the circuit counts down
- ❖ When the up and down input are both 0, the circuit does not change state
- ❖ When the up and down input are both 1, the circuit counts up
- ❖ Only one operation is performed at any given time
- ❖ The up input has priority over the down input



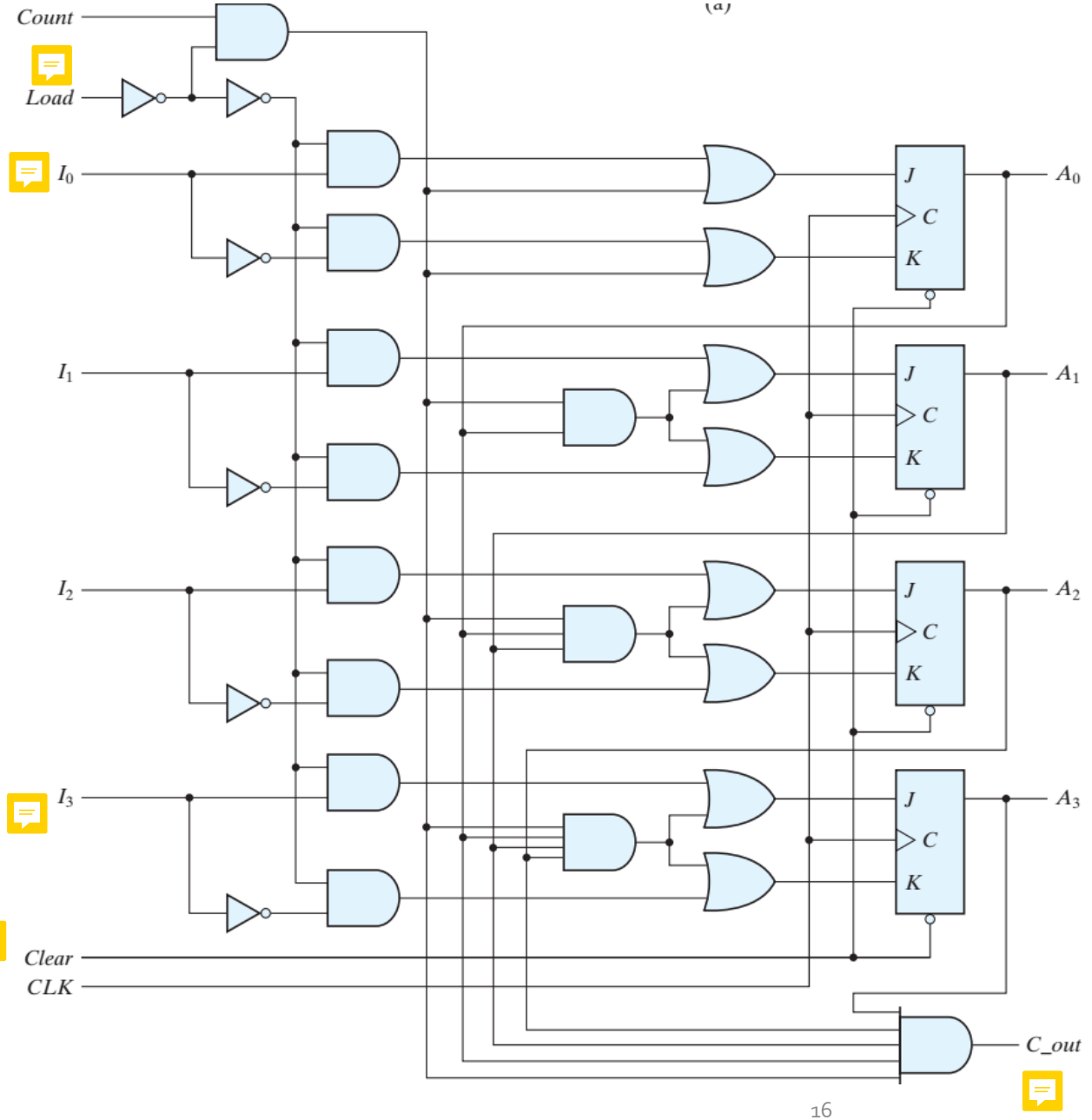
✓Synchronous Counters

➢ Binary Counter with Parallel Load

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

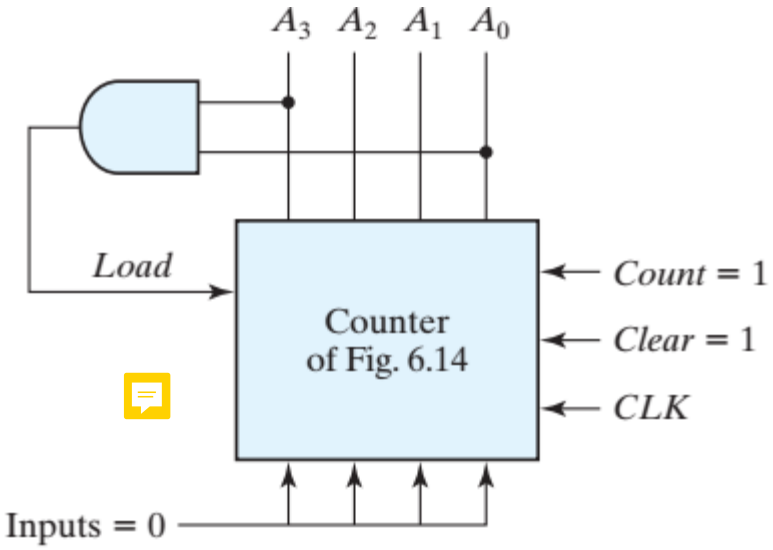


❖The carry output is useful for expanding the counter to more than four bits

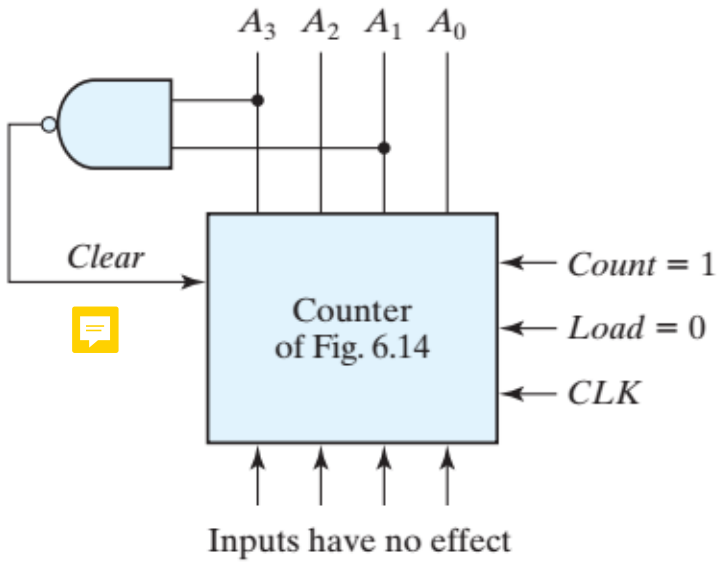


✓Synchronous Counters

- **BCD Counter (Modulo-10):** *counts from 0000 to 1001 and back to 0000*
  - Sequential circuit design procedure
  - Using counter with parallel load



The AND gate detects
the occurrence of 1001

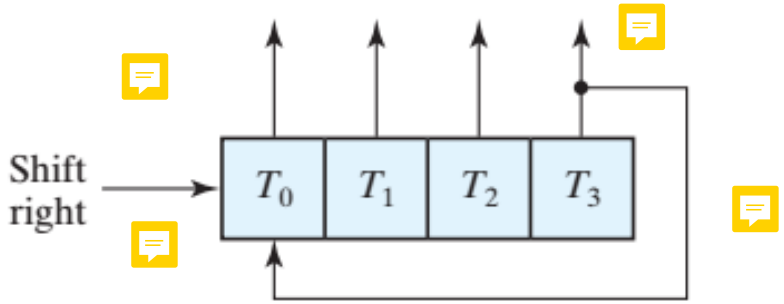


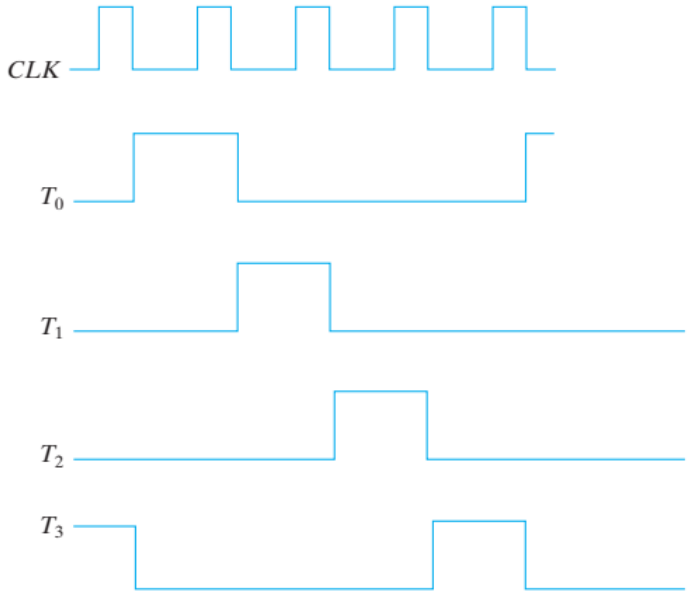
The NAND gate detects
the count of 1010

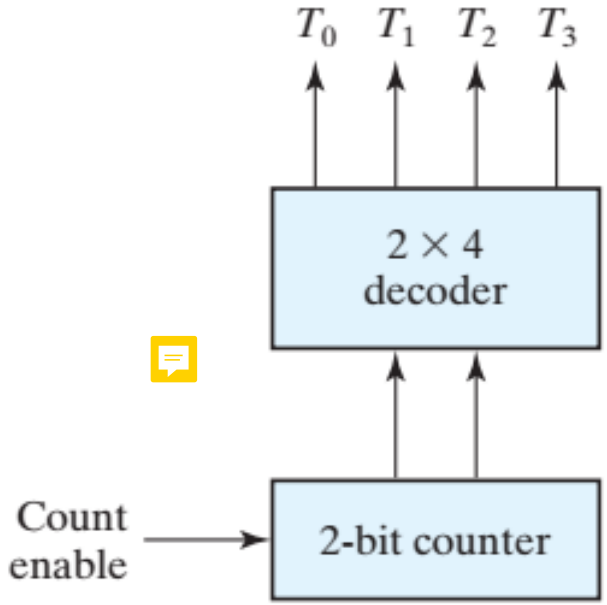
✓Synchronous Counters

- Timing signals that control the sequence of operations in a digital system can be generated by a shift register or by a counter with a decoder

- **Ring Counter**
  - A circular shift register with only one flip-flop being set at any particular time, all others are cleared







- The initial value of the register is 1000 and requires preset/clear flip-flops

✓Synchronous Counters

➢ Ring Counter

❖ To generate 2^n timing signals, we need either a shift register with 2^n flip-flops or an n-bit binary counter together with an n-to-2^n line decoder

❖ *16 timing signal can be generated with:*

✓16-bit shift register connected as a ring counter (16 flip-flops)

✓4-bit binary counter and a 4-to-16 line decoder (4 flip-flops and 16 four-input AND gates for the decoder)

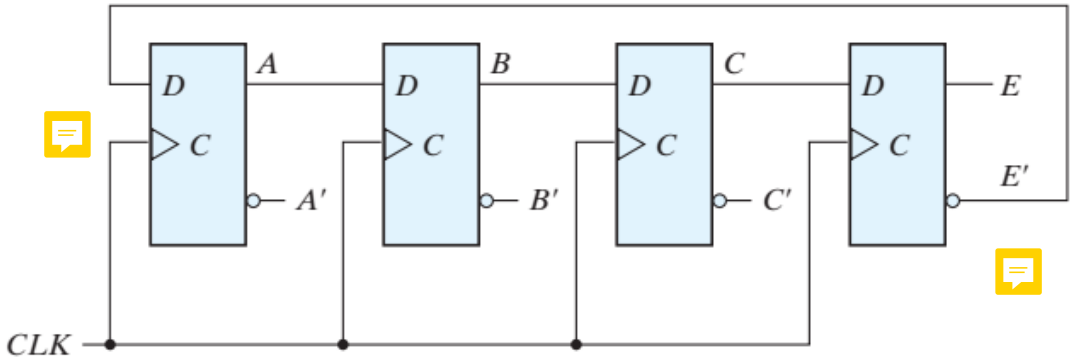
❖ The application is in controlling some operations or in analog adder

✓Synchronous Counters

➢ Johnson Counter

❖ The number of states in a ring counter can be doubled if the shift register is connected as a *switch-tail* ring counter

❖ A switch-tail ring counter is a circular shift register with the complement output of the last flip-flop connected to the input of the first flip-flop



Sequence number	Flip-flop outputs A	B	C	E	AND gate required for output
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	$B'C$
8	0	0	0	1	$C'E$

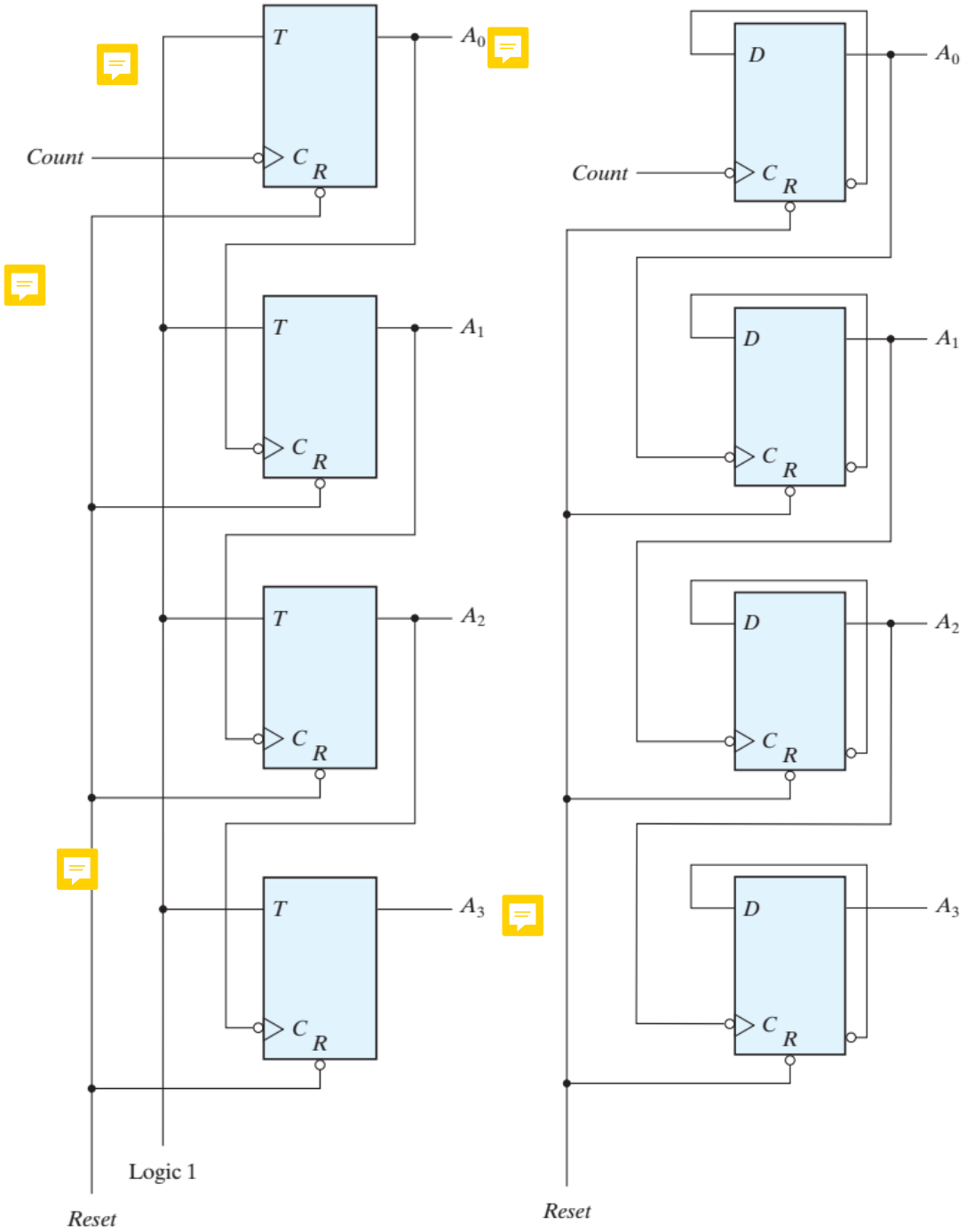
❖ The number of flip-flops needed is one-half the number of timing signals

❖ The number of decoding gates is equal to the number of timing signals

✓Ripple Counters

➢ Binary Ripple Counter

- ❖ The output of each flip-flop connected to the C input of the next higher order flip-flop
- ❖ The diagram of a binary *countdown counter* looks the same as the binary ripple counter, provided that all flip-flops trigger on the positive edge of the clock



✓Ripple Counters

➢ BCD Ripple Counter

- ❖ A BCD ripple counter is similar to a binary counter, except that the state after 1001 is 0000
- ❖ To count in decimal from 0 to 99, we need a two-BCD counter
- ❖ Multiple decade counters can be constructed by connecting BCD counters is cascade, one for each decade