جامعة الملك سعود
King Saud University

CSC 220: Computer Organization

Unit 9
Counters & RAM

Prepared by:
Md Saiful Islam, PhD

Updated by:
Isra Al-Turaiki, PhD

Department of Computer Science
College of Computer and Information Sciences

Overview

- Counters:
- Asynchronous (Ripple) Counters
- Synchronous Counters
- Introduction to Memory
  - RAM
  - ROM

Chapter-6, 7

M. Morris Mano, Charles R. Kime and Tom Martin, **Logic and Computer Design Fundamentals**, Global (5th) Edition, Pearson Education Limited, 2016. ISBN: 9781292096124

Binary Counters

- A *counter* is sequential circuits which "counts" through a specific state sequence.
- A counter that follows the binary number sequence is called a *binary counter*.
- A n-bit binary counter
    - consists of n flip-flops
    - counts: 0 – (2^n-1)
- Example: 2 bit binary counter: 00 $\rightarrow$ 01 $\rightarrow$ 10 $\rightarrow$ 11 $\rightarrow$ 00.

Binary Counters

- They can count up, count down, or count through other fixed sequences.
- Categories of Counter
  - Ripple / asynchronous: clock pulses are different for flip-flops
  - Synchronous: all flip-flops receive the same clock pulse

Ripple Counters

- The LSB is complemented by each count pulse.
- If Q_0 goes from 1 to 0, complement Q_1.
- If Q_1 goes from 1 to 0, complement Q_2.
- If Q_2 goes from 1 to 0, complement Q_3.
- and so on..

Upward Counting Sequence

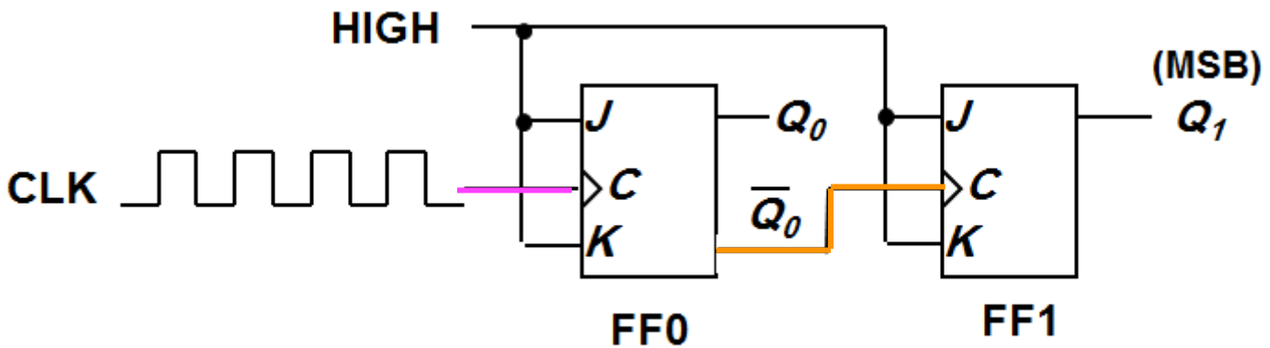
Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Ripple Counters

- Consists of a series of complementing flip-flops (ex. JK).
- The LSB flip-flop is connected to clock.
- Output of one flip-flop is connected to the clock input of the next more-significant flip-flop.
- The state change of one flip-flop triggers the next flip-flop in line.
- An n-bit Asynchronous counter can have 2^n possible counting states
  - **Example:** 3-bit asynchronous counter

    has (0-7) states and is called MOD-8 counter

2-bit Ripple Counter (MOD-4)

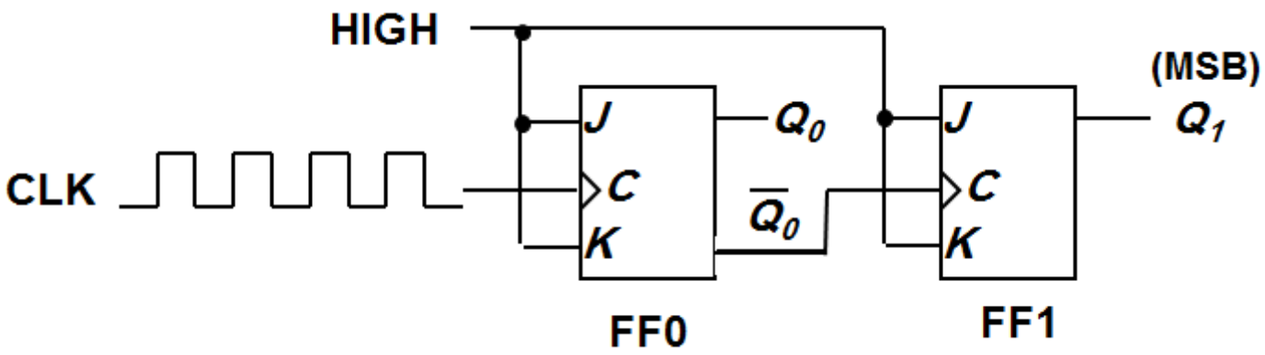


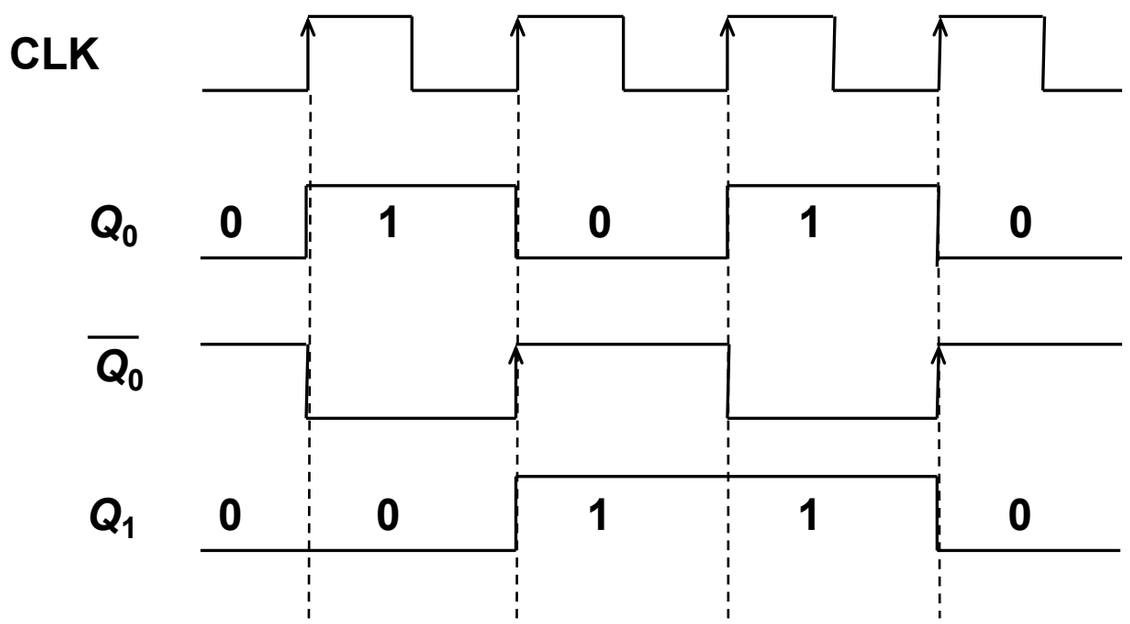
When Q'_0 goes from 0 to 1, Q_1 is complemented.

The LSB Q_0 is complemneted with each clock pulse.

J	K	C_{FF1}	Q1	Q0	CLK
			0	0	
1	1	↓	0	1	↑
1	1	↑	1	0	↑
1	1	↓	1	1	↑

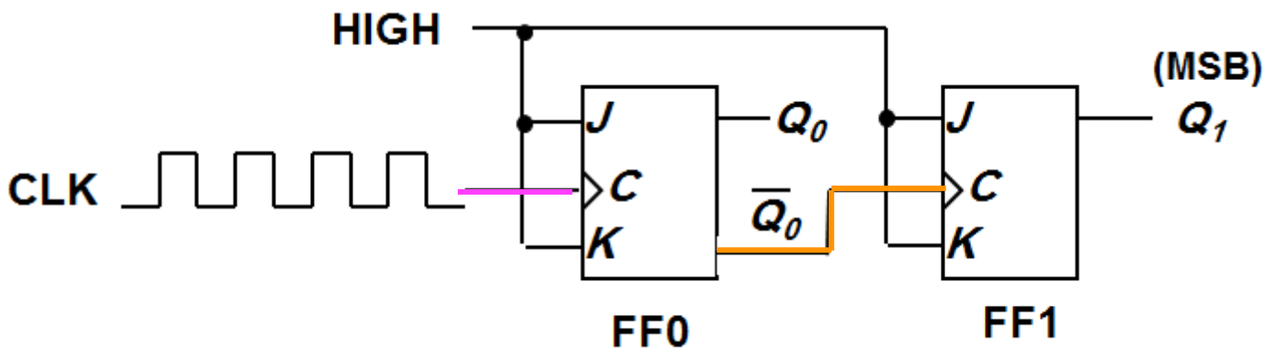
2-bit Ripple Counter (MOD-4)

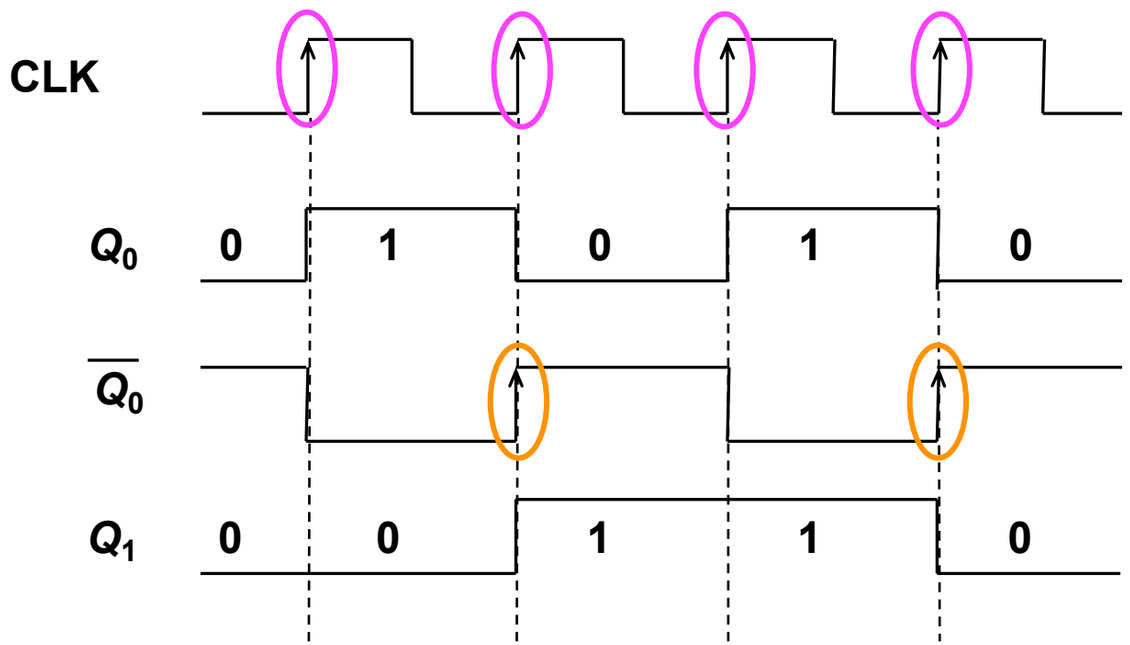




Timing diagram: 00 → 01 → 10 → 11 → 00...

2-bit Ripple Counter (MOD-4)





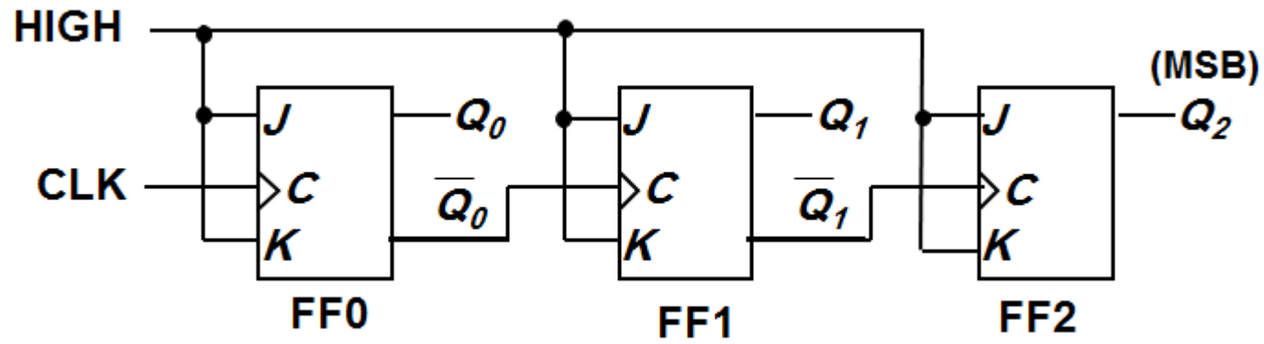
Q_0 will always be complemented on the rising

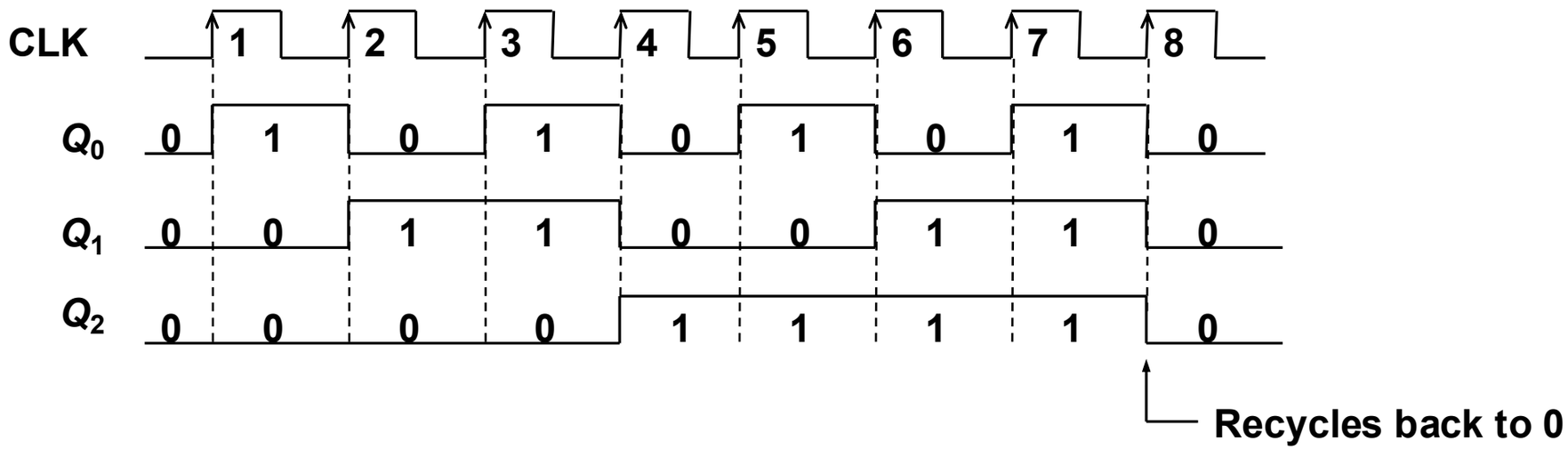
Q_1 will be complemented on the transition of of Q'_0 from 0 to 1

Timing diagram: 00 → 01 → 10 → 11 → 00...

3-bit Ripple Counter (MOD-8)

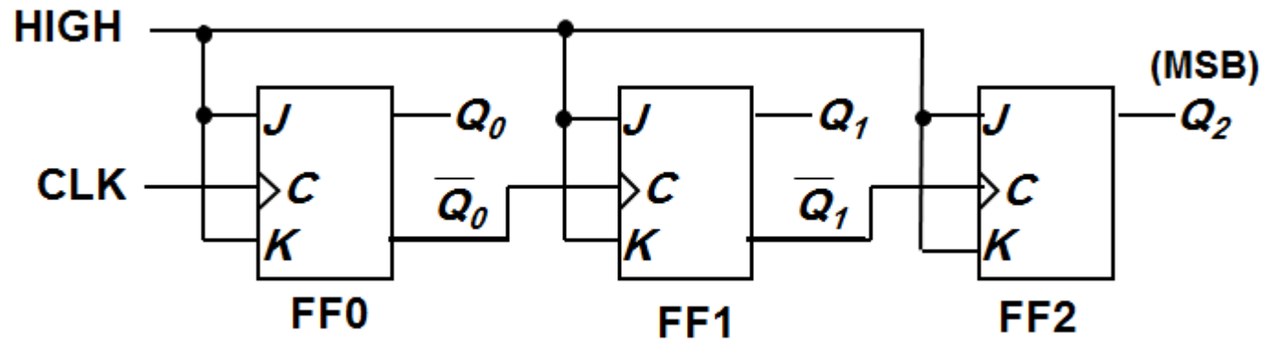
Design of a 3-bit ripple binary counter:

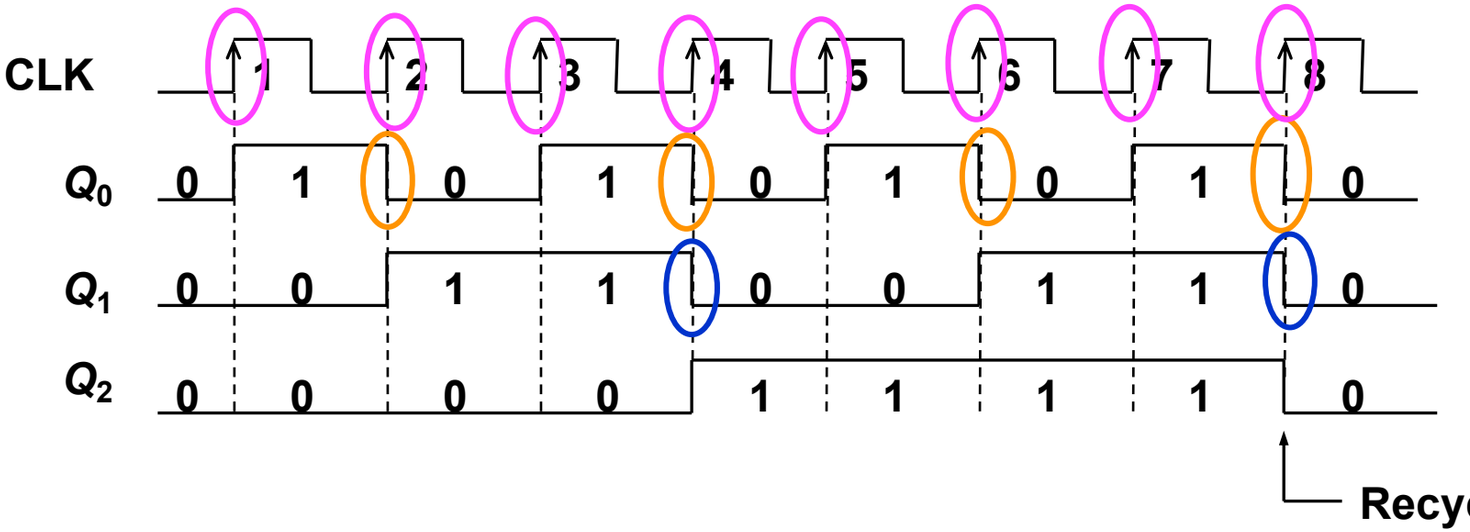




3-bit Ripple Counter (MOD-8)

Design of a 3-bit ripple binary counter:





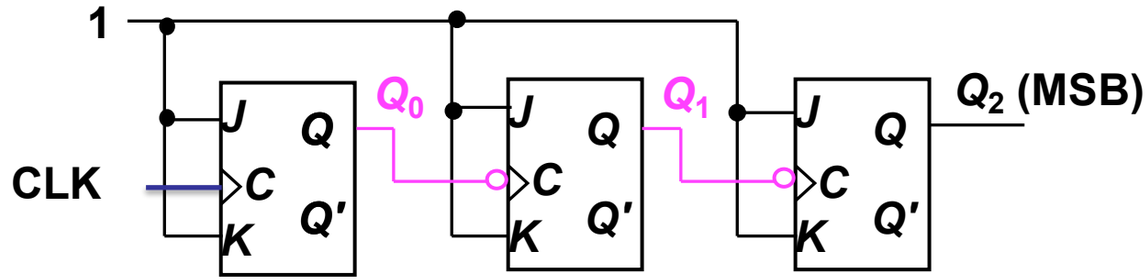
Q_0 will always be complemented on the rising edge of CLK

Q1 will be complemented on the transition of Q'$_0$ from 0 to 1

Q_2 will be complemented on the transition of Q_1' from 0 to 1

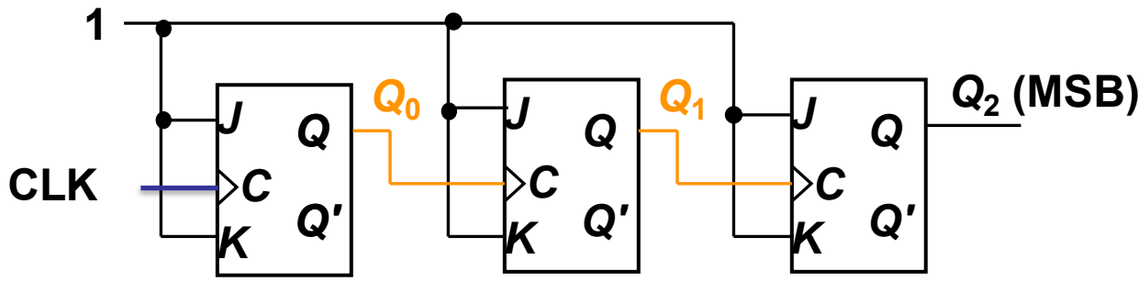
Up and Down Counters

- Up counters: count upward from zero to a maximum value, and repeat



3-bit binary up counter

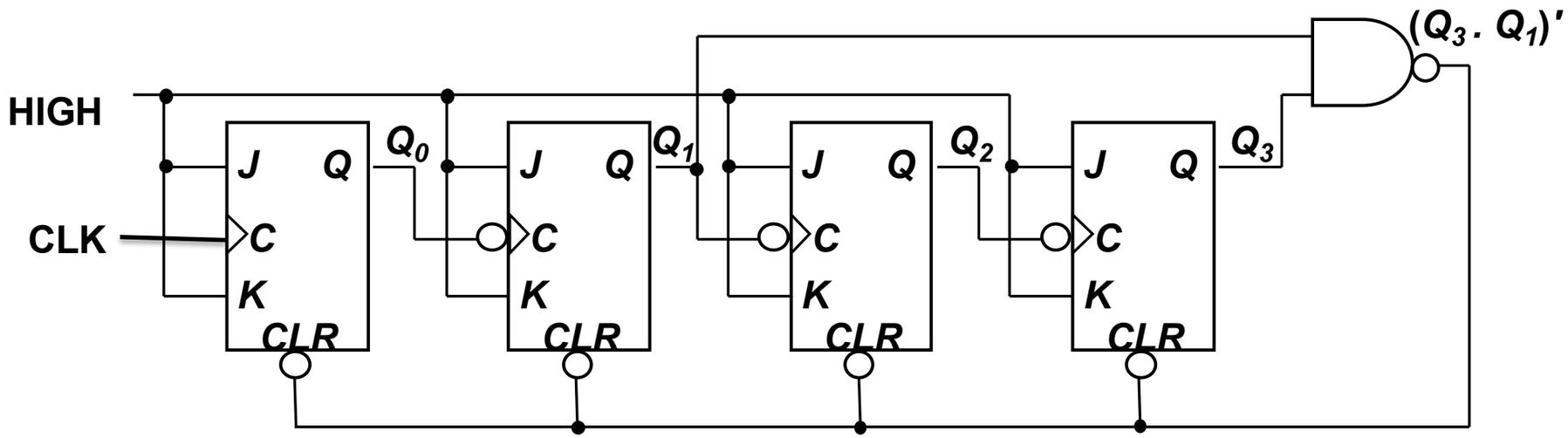
- Down counters: count downward from a maximum value to zero, and repeat (decrement by 1)



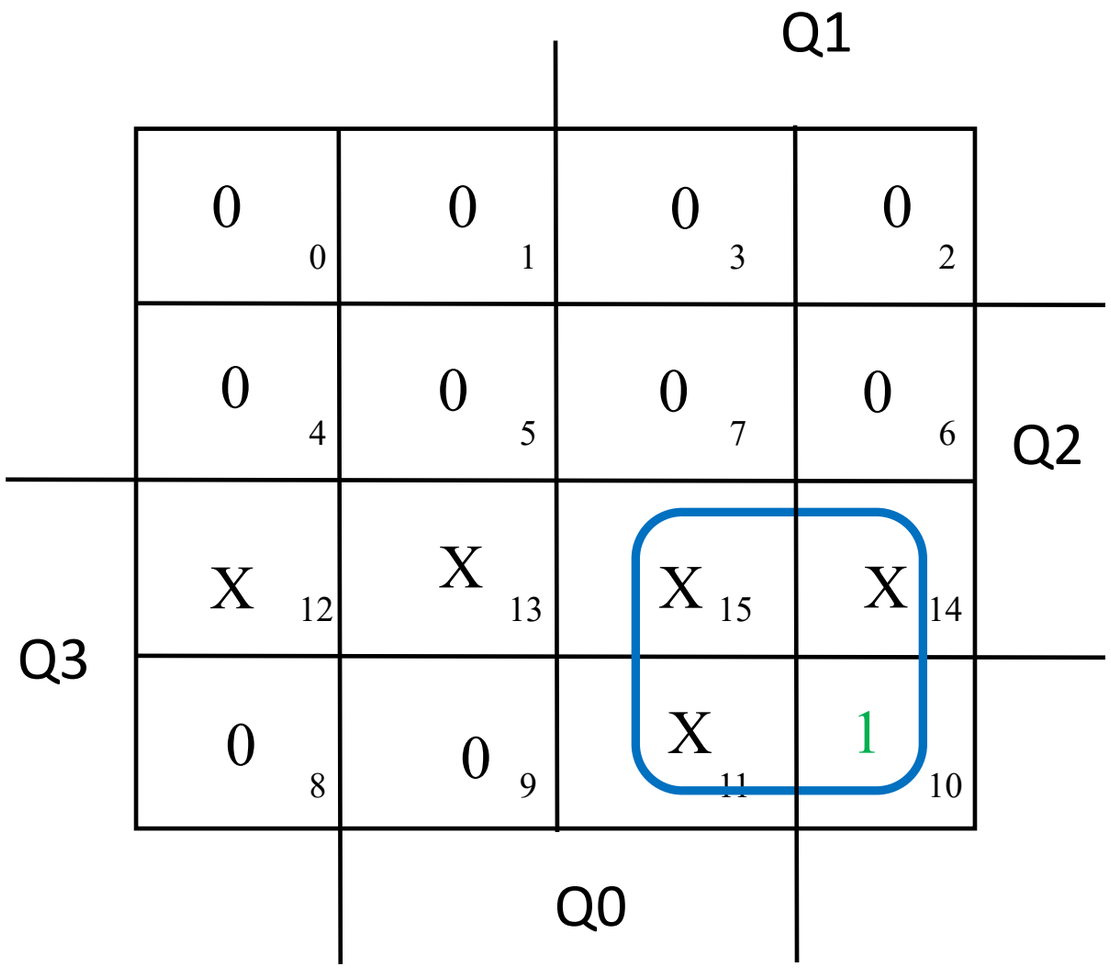
3-bit binary down counter

Asynchronous Counters with MOD no. < 2^n

- Design of an asynchronous MOD-10 counter: counts (0-9)



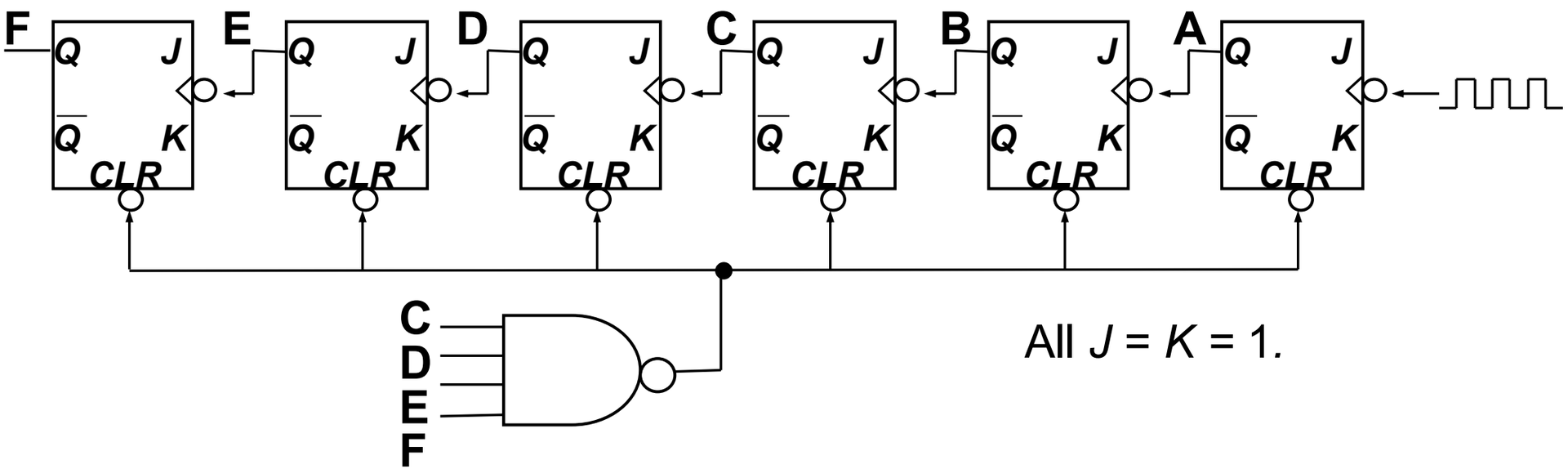
- MOD-10 counter: known as decade counter or BCD counter
- Counter with 10 states (modulus-10) in their sequence
- They are commonly used in daily life (e.g.: utility meters, odometers, etc.).


Q1
Q2
Q3
Q0
0 0
0 1
0 3
0 2
0 4
0 5
0 7
0 6
X 12
X 13
X 15
X 14
0 8
0 9
X 11
1 10

Asynchronous Counters with MOD no. < 2^n

- Exercise: How to construct an asynchronous MOD-5 counter? MOD-7 counter? MOD-12 counter?

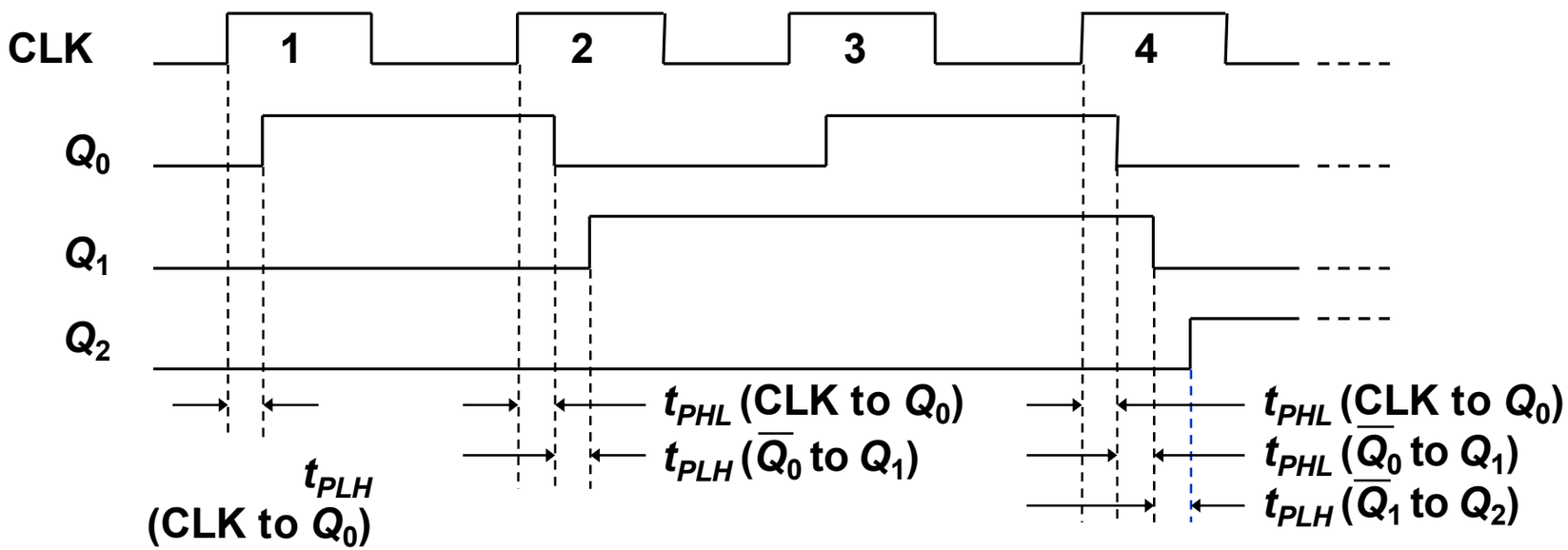
- **Question:** The following is a MOD-? counter?



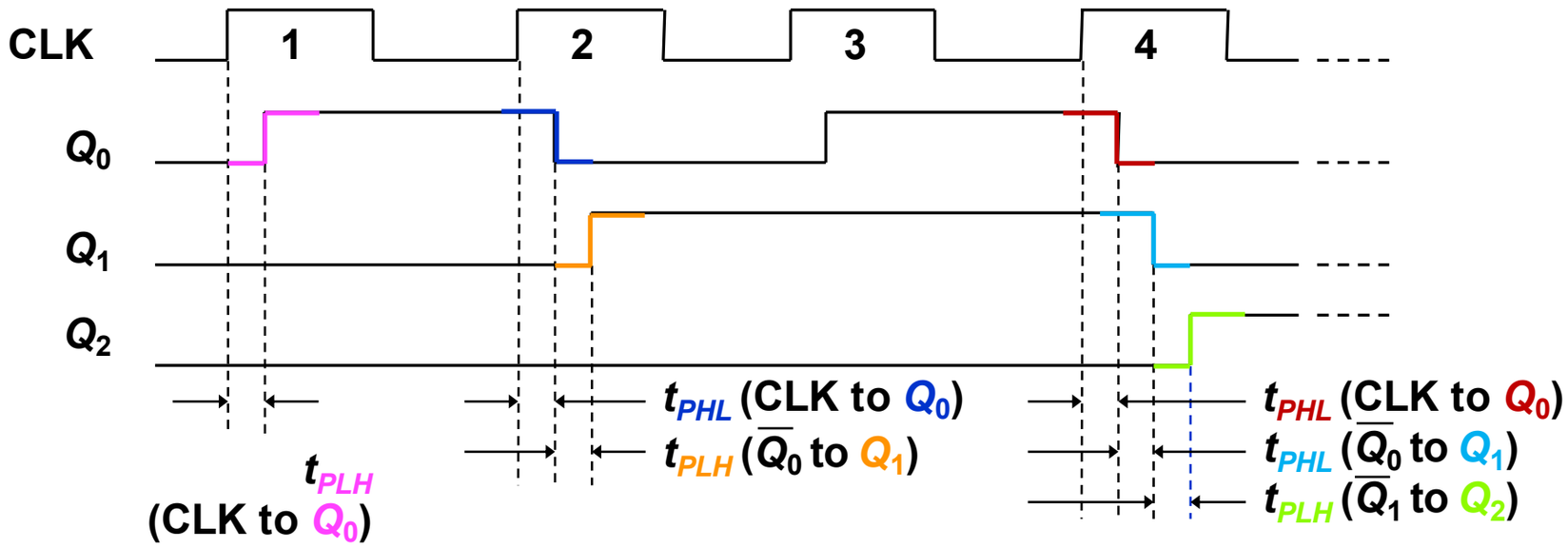
The counter is to be cleared when it reaches the count of $(111100)_2$ so it is MOD-60.

Ripple Counter: Propagation Delays

- *Propagation delay* is the time required for a change in value of a signal to propagate from input to output.
  - *high-to-low propagation time* t_{PHL}
  - *low-to-high propagation time* t_{PLH}
- If the accumulated delay is greater than the clock pulse, some counter states may be misrepresented!
- Large ripple counters can be slow circuits→ synchronous binary counters are favored

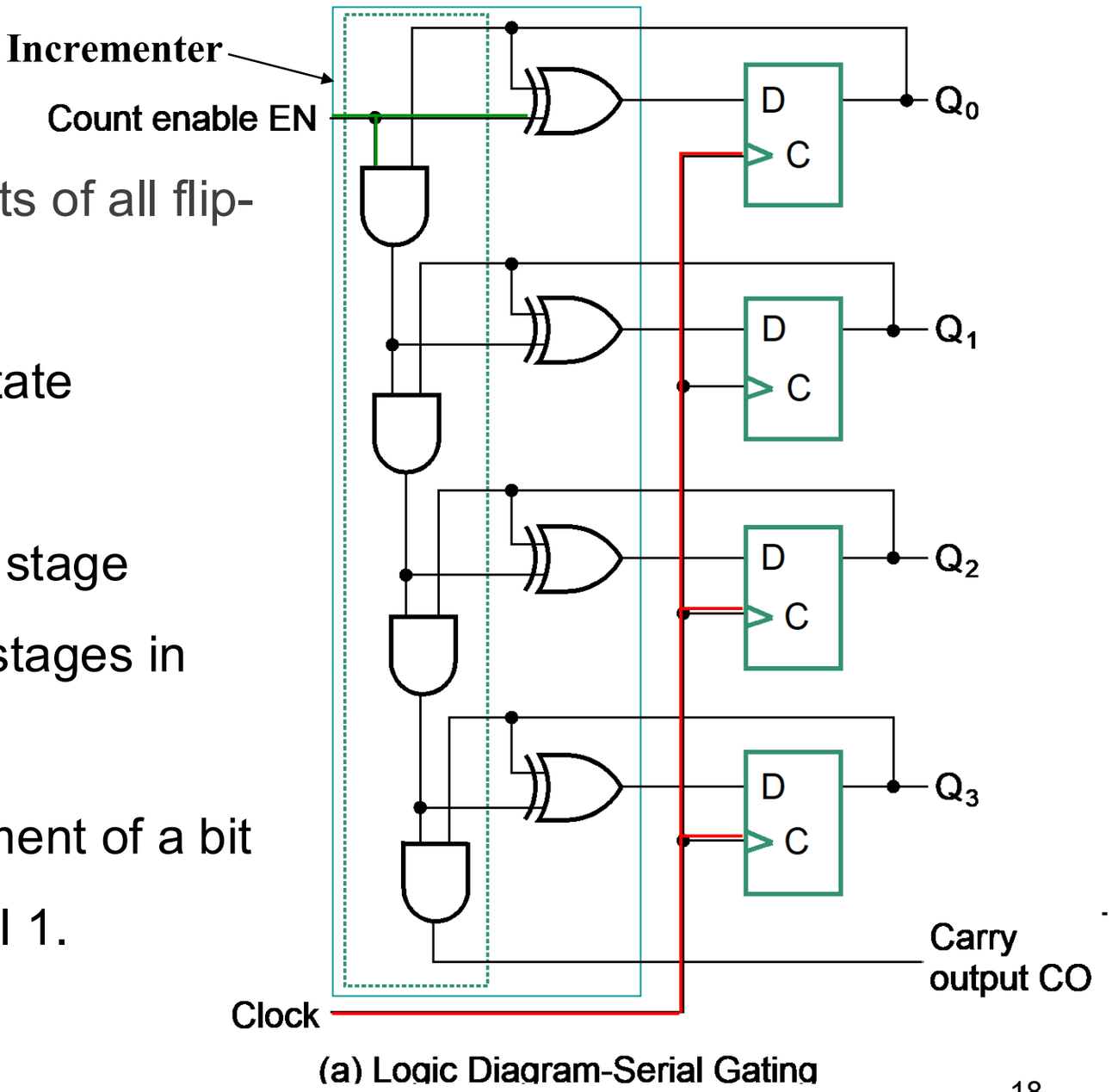


Ripple Counter: Propagation Delays



Synchronous Counters

- The clock applied to the C inputs of all flip-flops.
- Count Enable: If 0, “hold” the state
- 2-input AND gates:
  - provide information to each stage about the state of the prior stages in the counter.
  - AND chain causes complement of a bit if all previous LSB are equal 1.



(a) Logic Diagram-Serial Gating

Synchronous Counters (continued)

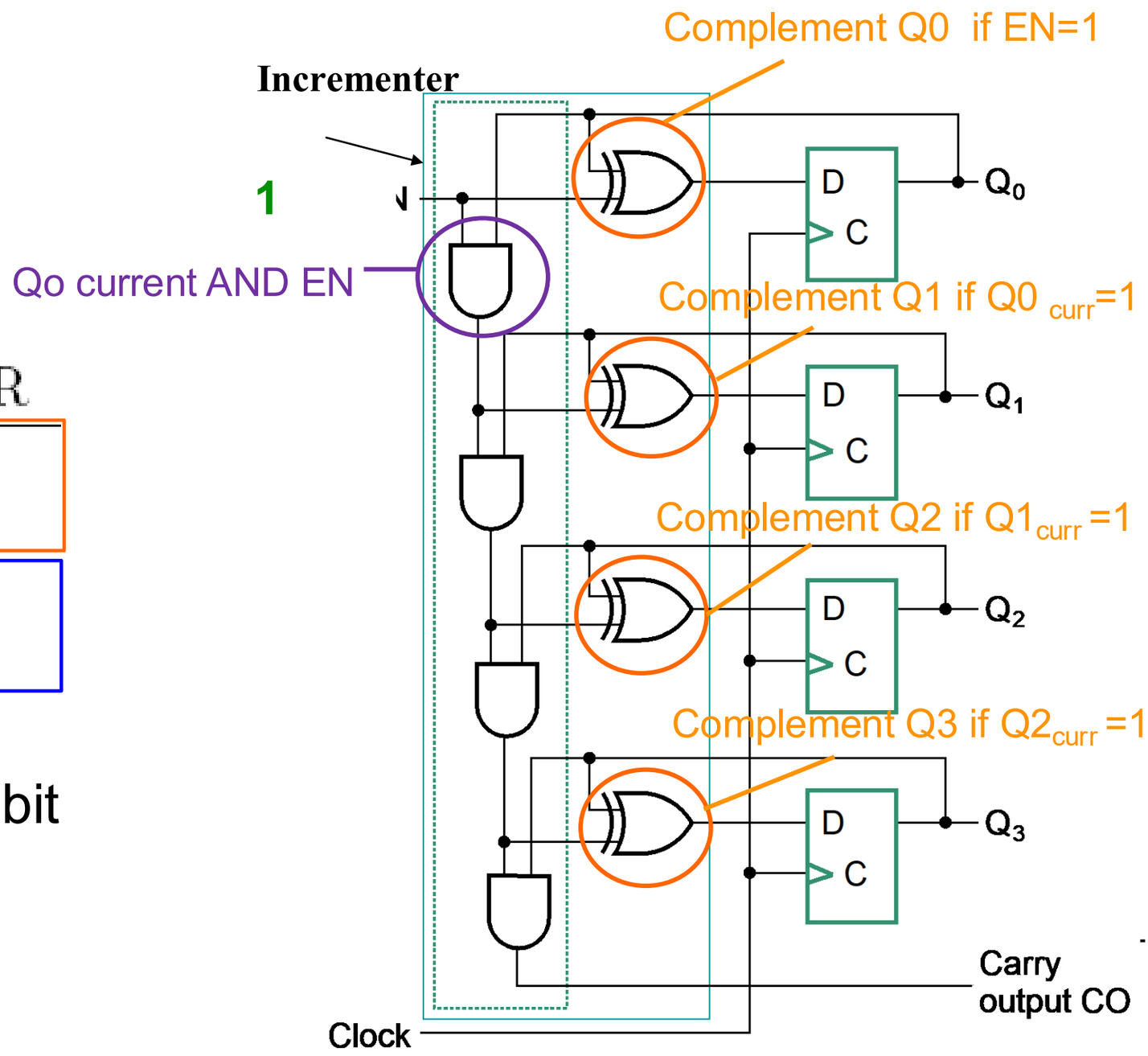
A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Hold

Complemen

t

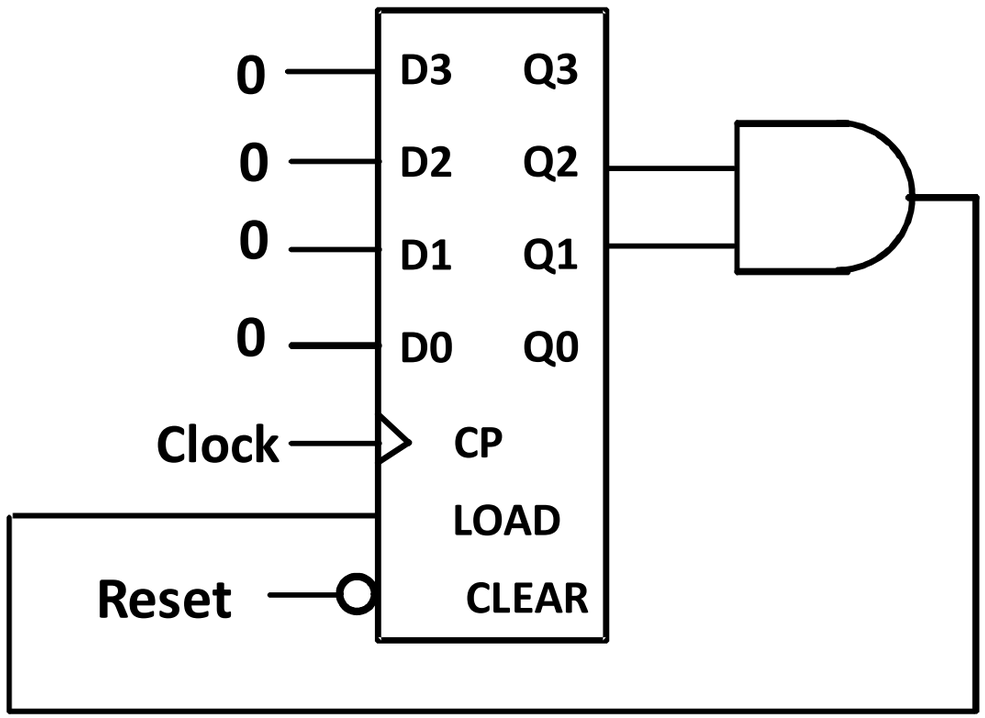
XOR complements each bit



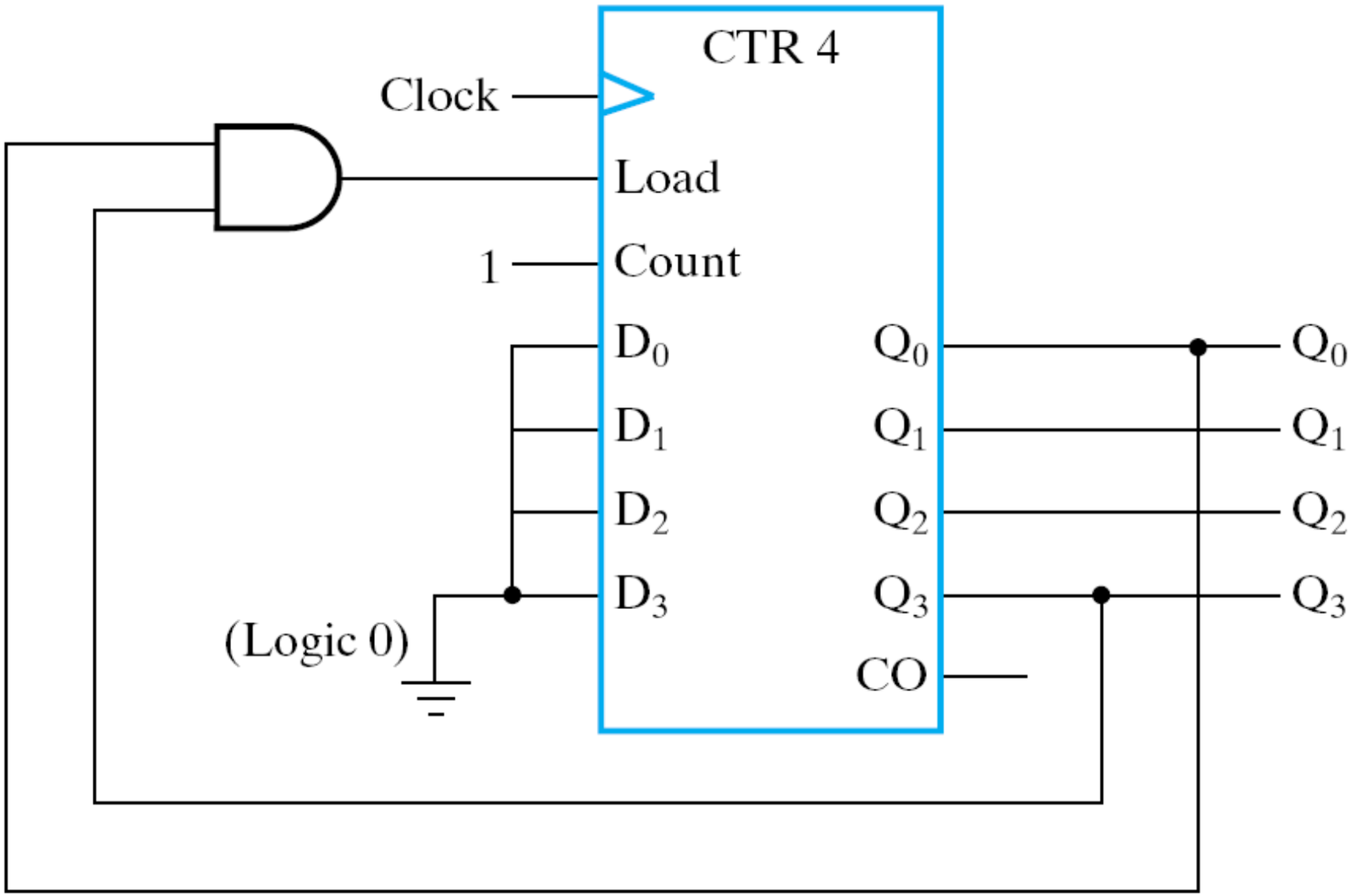
(a) Logic Diagram-Serial Gating

Synchronous Counters with MOD no. < 2^n

- A synchronous 4-bit binary counter with a synchronous load and an asynchronous clear is used to make a Modulo 7 counter
- Use the Load feature to detect the count "6" and load in "zero". This gives a count of 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, 4, 5, 6, 0, ...
- Using don't cares for states above 0110, detection of 6 can be done with Load = $Q_2\ Q_1$

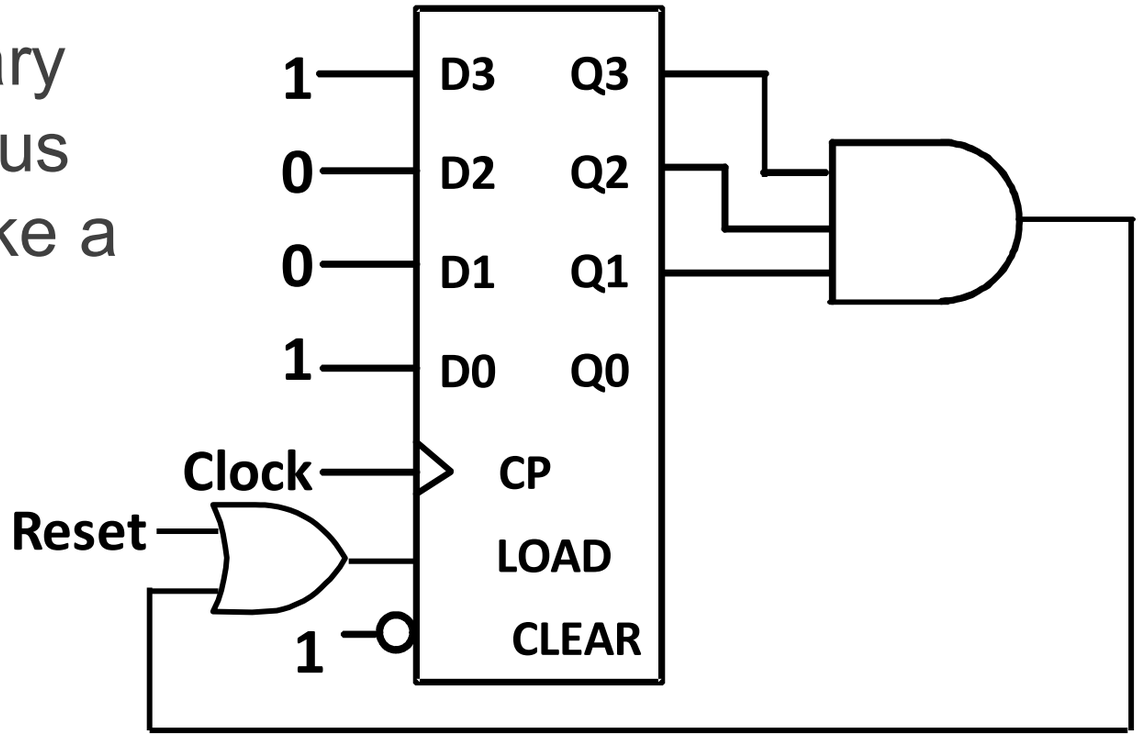


Synchronous BCD Counter



Synchronous Modulo 6 Counter: 9 to 14

- A synchronous, 4-bit binary counter with a synchronous Load is to be used to make a Modulo 6 counter.
- Use the Load feature to preset the count to 9 on Reset and detection of count 14.

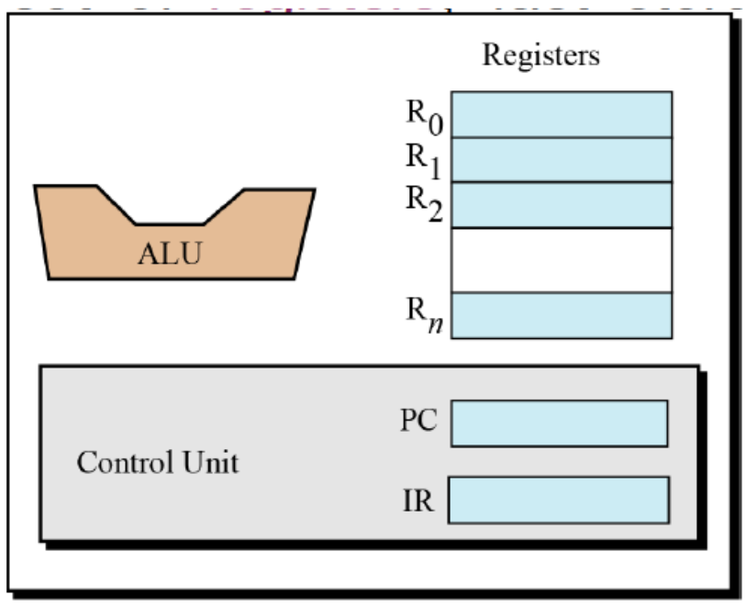


- This gives a count of 9, 10, 11, 12, 13, 14, 9, 10, 11, 12, 13, 14, 9, …

Memory

- Memory:
  A collection of storage cells together with the necessary circuits to transfer information to and from them.

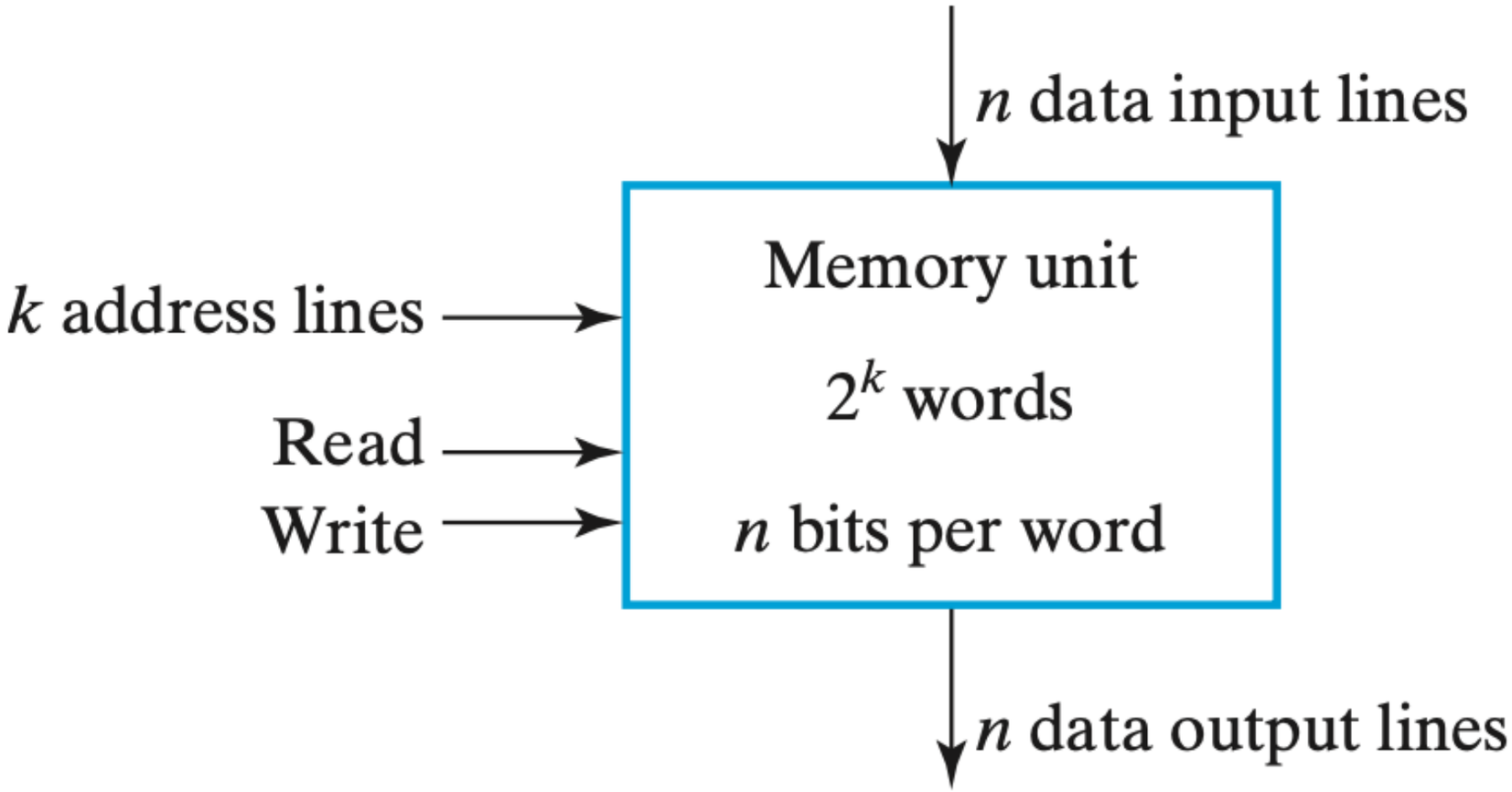
- Random Access Memory (RAM):
  a memory organized such that data can be transferred to or from any desired location, with the access taking the same time regardless of the location.



Memory Definitions

- Typical data elements are:
  - Bit: a single binary digit
  - Byte: a collection of eight bits accessed together
  - Word:
    - is an entity of bits that moves in and out of memory as a unit.
    - It is typically a power of two multiple of bytes (e.g., 1 byte, 2 bytes, 4 bytes, 8 bytes, etc.)

Memory Block Diagram



Memory Block Diagram

Communication between a memory and its environment is achieved through:

- n data input lines: since each word is n bits.
- n output lines.
- k address selection lines: to address 2^k words of memory.
- Read and Write are single control lines defining the simplest of memory operations.

Memory Example

- This is a 1024 X 16 Memory

- Number of address lines, k=10
- Number of input datal lines, n=16
- Number of output datal lines, n=16

Memory Address		
Binary	Decimal	Memory Contents
0000000000	0	10110101 01011100
0000000001	1	10101011 10001001
0000000010	2	00001101 01000110
	.	.
	.	.
	.	.
	.	.
	.	.
1111111101	1021	10011101 00010101
1111111110	1022	00001101 00011110
1111111111	1023	11011110 00100100

FIGURE 7-2
Contents of a 1024 × 16 Memory

Write and Read Operations

- Read Memory: an operation that reads a data value stored in memory:
  - Place a valid address on the address lines.
  - Activate the Read input
  - Wait for the read data to become stable.
- Write Memory: an operation that writes a data value to memory:
  - Place a valid address on the address lines
  - Apply the data to the data lines.
  - Toggle the memory write control line

Control Inputs to a Memory Chip

- Instead of separate Read and Write control lines, most ICs provide a Chip Select that selects the chip to be read from or written to and a Read/Write that determines the particular operation.

TABLE 7-1
Control Inputs to a Memory Chip

Chip Select CS	Read/$\overline{\text{Write}}$ R/$\overline{\text{W}}$	Memory Operation
0	×	None
1	0	Write to selected word
1	1	Read from selected word

Memory Size

- we refer to 2^{10} as K (kilo), 2^{20} as M (mega), 2^{30} as G (giga), and 2^{40} as T (tera)

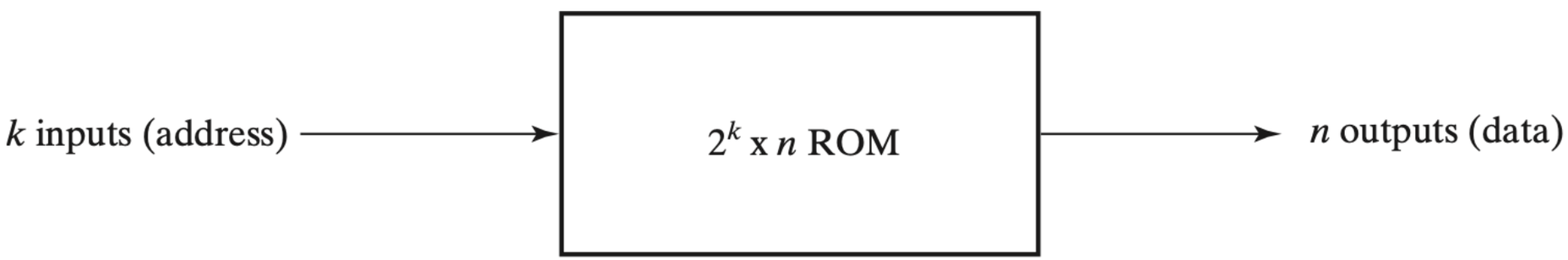
- Example: a memory of 2^{28} bits

$2^{28}/ 8= 2^{25}$ Bytes

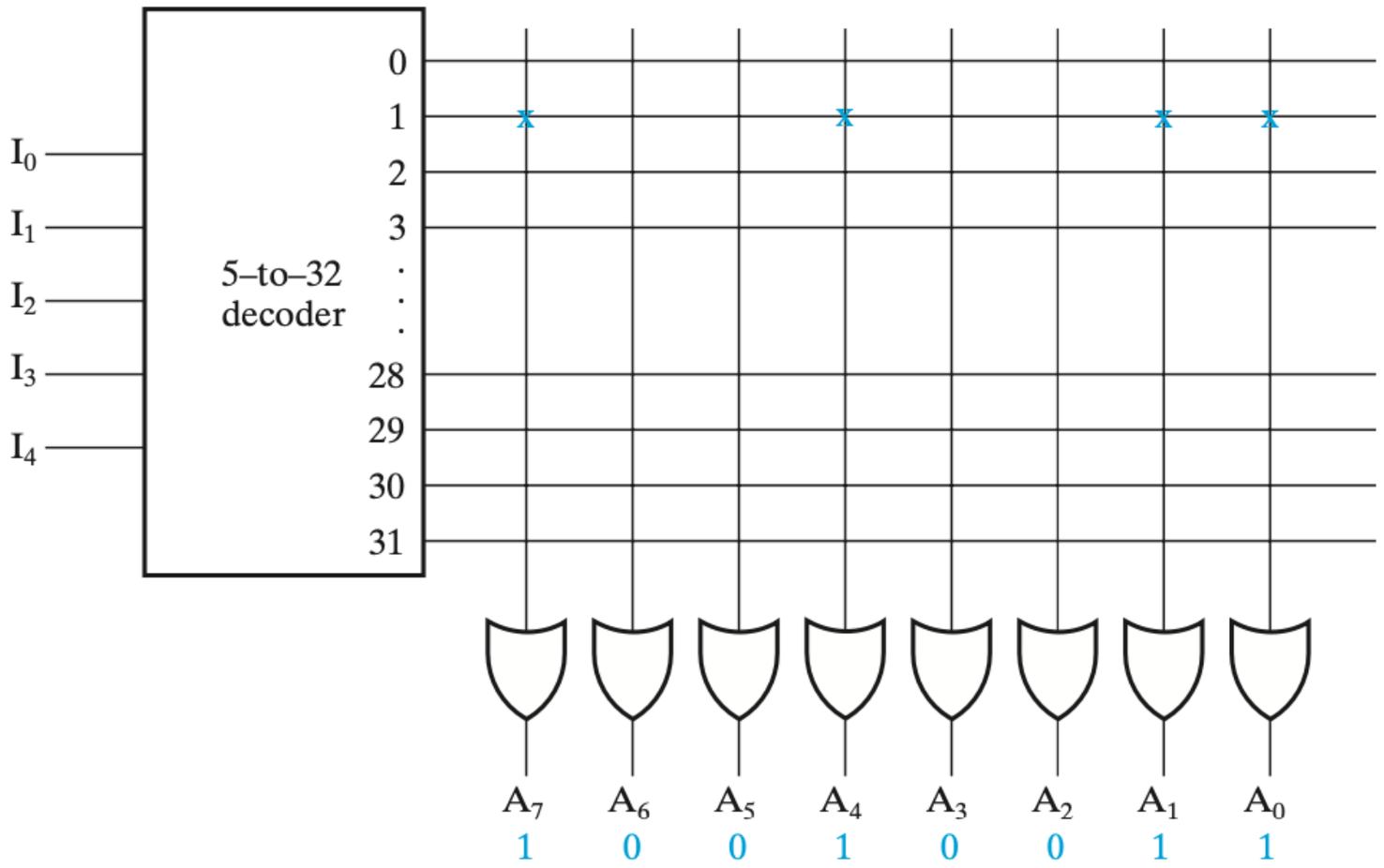
$2^{25} = 2^{5} X 2^{20}= 32$ MB

Read Only Memory (ROM)

- Characteristics
  - –Perform read operation only, write operation is not possible
  - – Information stored in a ROM is made permanent during production and cannot be changed



32 X 8 ROM



- Each output of the decoder represents a memory address.
- The ROM is programmed with the word 10010011 in memory address 1.

RTL and Memory

TABLE 6-1
Basic Symbols for Register Transfers

Symbol	Description	Examples
Letters (and numerals)	Denotes a register	$AR, R2, DR, IR$
Parentheses	Denotes a part of a register	$R2(1), R2(7{:}0), AR(L)$
Arrow	Denotes transfer of data	$R1 \leftarrow R2$
Comma	Separates simultaneous transfers	$R1 \leftarrow R2, R2 \leftarrow R1$
Square brackets	Specifies an address for memory	$DR \leftarrow M[AR]$

- Memory to register transfers are called *read operations,*
- Register to memory transfers are called *write operations.*
- Both require specification of the memory location to be used (which can be done through a special register (AR) or a special bus (Address bus).

RTL and Memory

- RTL expressions for a Read operation, assuming the use of an address registers:

$$AR \leftarrow \text{address}$$
$$DR \leftarrow M[AR]$$

- RTL expressions for a Write operation, assuming use of a data register:

$$AR \leftarrow \text{address}$$
$$DR \leftarrow \text{value}$$
$$M[AR] \leftarrow DR$$