Boolean Algebra – Set 3

Appendix C (on the text DVD)

C.1,2,3,5,6,7,8,9

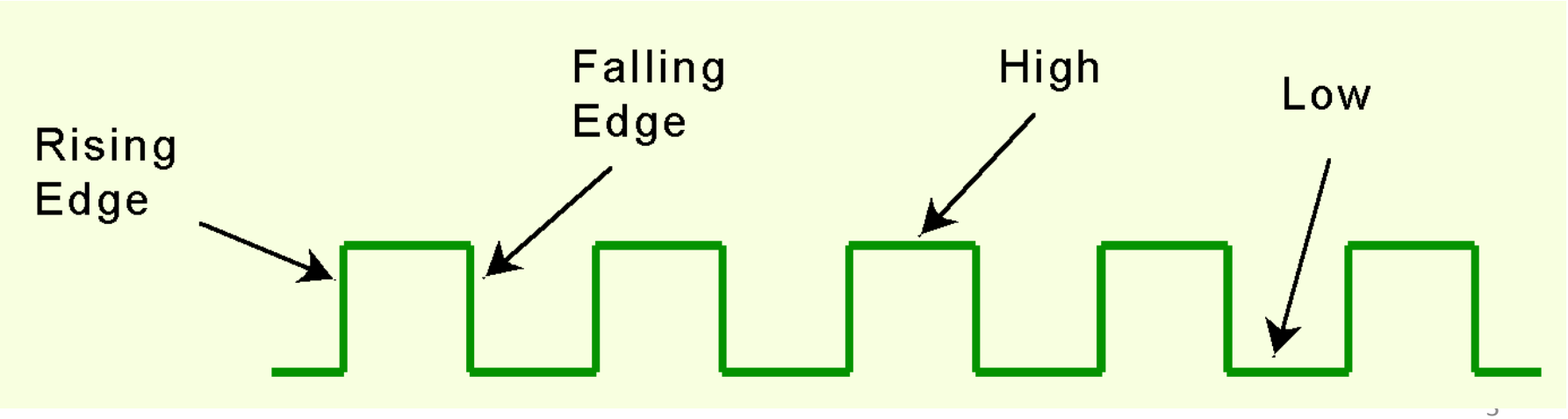
Also – Computer Organization & Architecture – Null & Lobor notes

Sequential Circuits

- Unlike CLC's, sequential circuits (SC) generate their output based on the current input AND some function of the past inputs.
- They have memory!
- In addition to their memory, sequential circuits are also generally controlled by a clock.

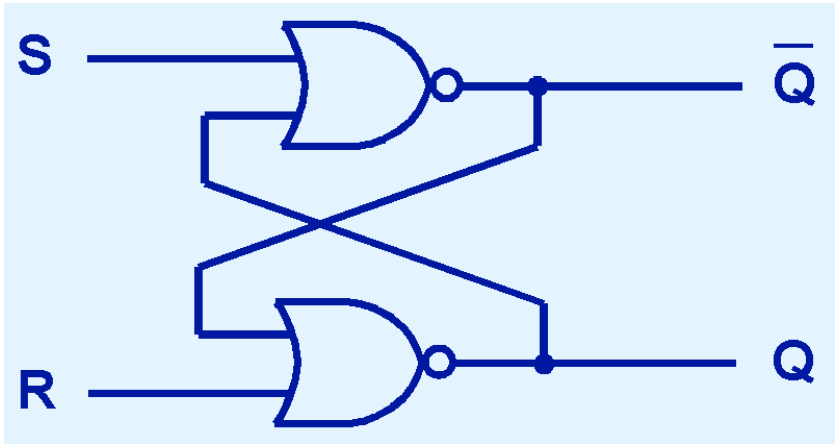
Sequential Circuit - Clock

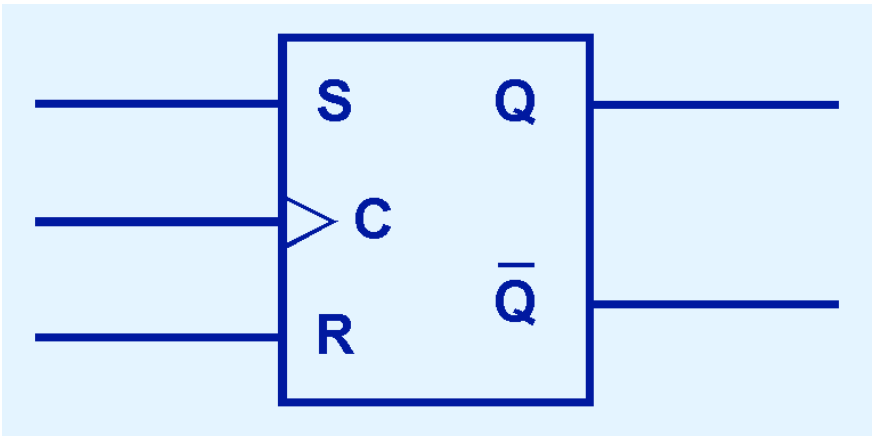
- Unlike CLC's by adding a clock, the state (memory) of a SLC does not change as the input(s) change. The SLC changes state on the leading edge, falling edge, or highest level of the clock signal. These are called edge-triggered and level triggered circuits



SLC

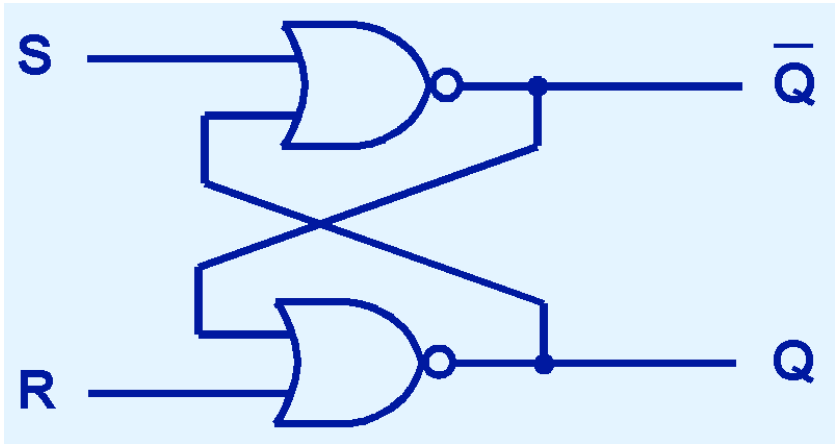
- Generally, the memory of a SLC is implemented using flip-flops
- Internally, flip-clops use feedback to hold their value
- The following is an example of a SR (set/reset) flip-flop





SLC

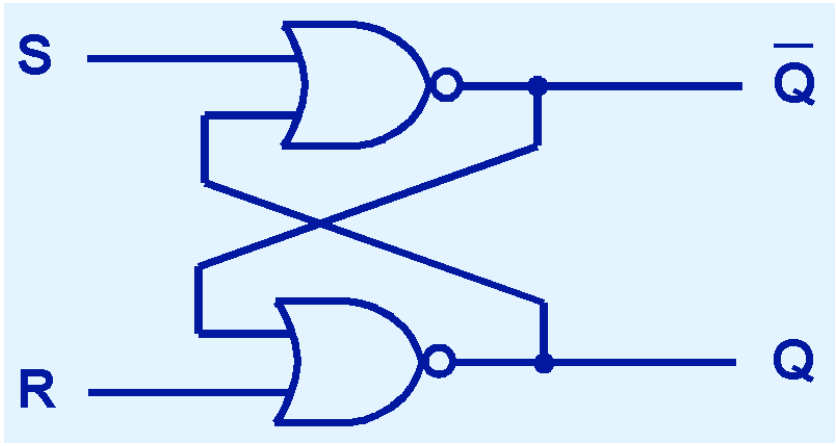
- A flip-flop's behavior is defined by its characteristic table, a truth table, or by a Boolean expression



Characteristic Table

S	R	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined

SLC – Truth Table



Present State			Next State
S	R	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	undefined
1	1	1	undefined

Q(t) vs Q(t+1)

- As noted, most flip-flops are clocked. What this means is that the flip-flop does not change state until the clock pulse has occurred.
- The Q(t) is for the value or state of the flip-flop at time t, and q(t+1) means time t plus one clock pulse later.
- Sometimes one thinks of the clock pulse C as ANDed with the inputs. However, as you will see, this does not work in the case of the D flip-flop.

SLC – Next State Equations

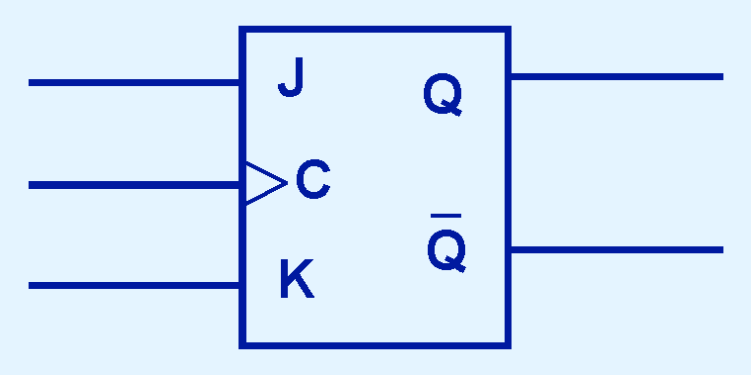
$$Q(t+1) = \begin{cases} S + \overline{R}\,Q & SR = 0 \\ \text{undefined} & SR = 1 \end{cases}$$

Other flip-flops and the clock

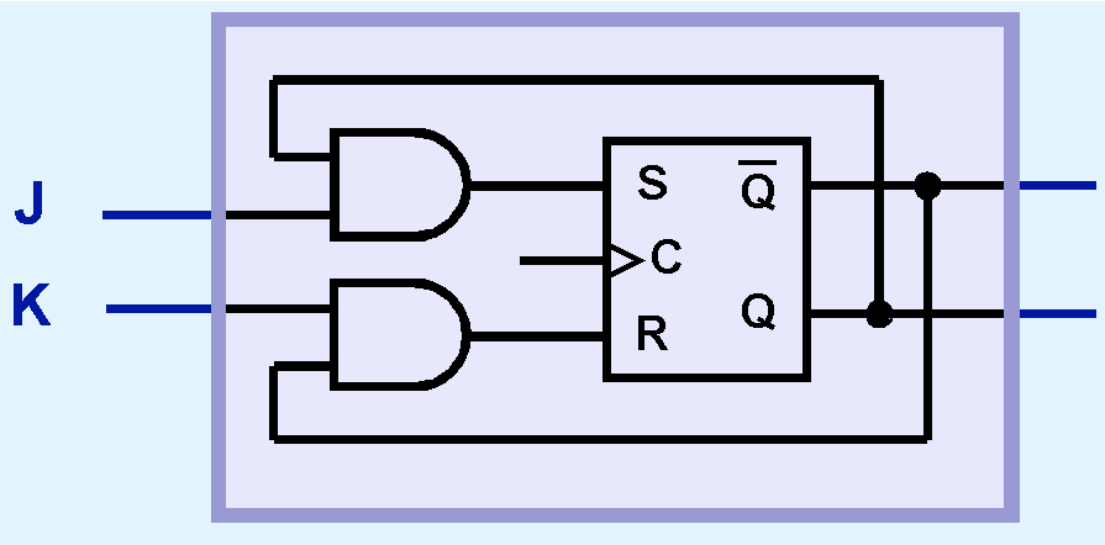
- There are 3 other common types of flip-flops
  - J-K
    - Like the RS with J=set, K=reset, and when J=K=1 the flip-flop toggles, i.e. changes state.
  - D
    - Delay
  - T
    - Toggle

Other flip-flops, clock - JK

- The JK is like the RS, but when both inputs are a 1, instead of the next state being undefined, the flip-flop toggles, i.e. Q(t+1) = Q'(t)



JK



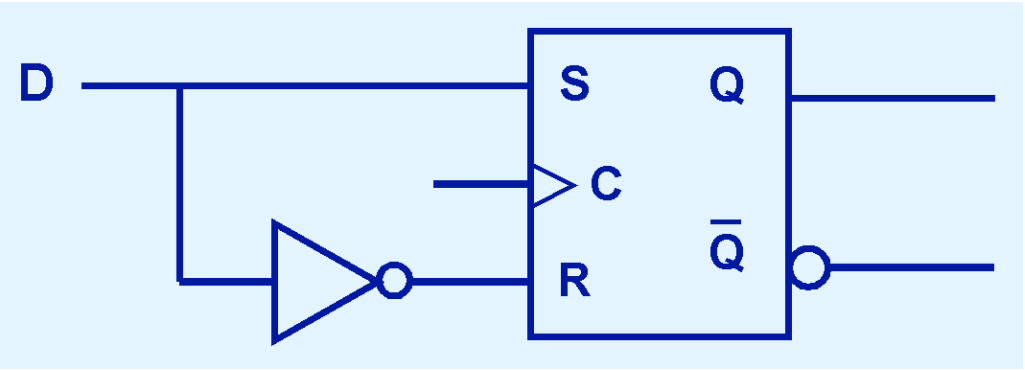
J	K	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\overline{Q}$(t)

JK

$$Q(t+1) = J\overline{Q}(t) + \overline{K}Q(t)$$

D (delay) Filp-Flop

- The D flip-flop accepts as input a single value, and on the next clock pulse the state changes to be the same as the current input.



D	Q(t+1)
0	0
1	1

Sequential Circuits (machines)

- A sequential machine is a machine defined as follows:

$$M(Q, I, \delta, O)$$

$Q = \text{set of states}$

$S = \text{start state}$

$I = \text{set of inputs}$

$\delta = \text{next state transition as a function of S and I}$

$O = \text{output as a function of current state or current state and input}$

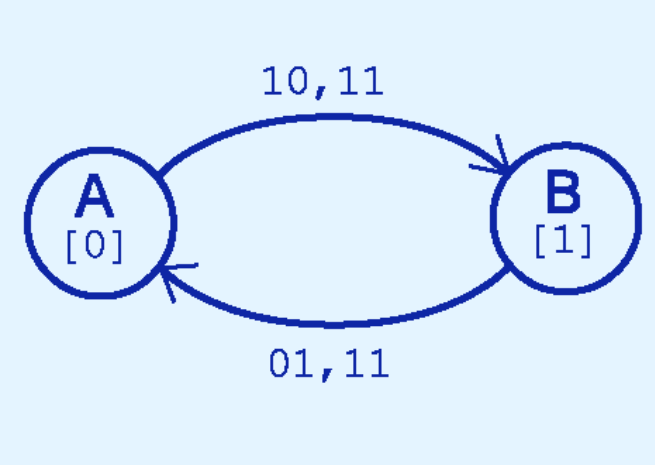
Sequential Circuits (machines)

- Sequential machines, also often called finite state machines (FSM), are generally represented as a state diagram or a characteristic table.
- A characteristic table is simply a “beefed up” truth table
- A state diagram is a pictorial representation of the machine.
  - There are two types of FSM’s Moore and Mealy

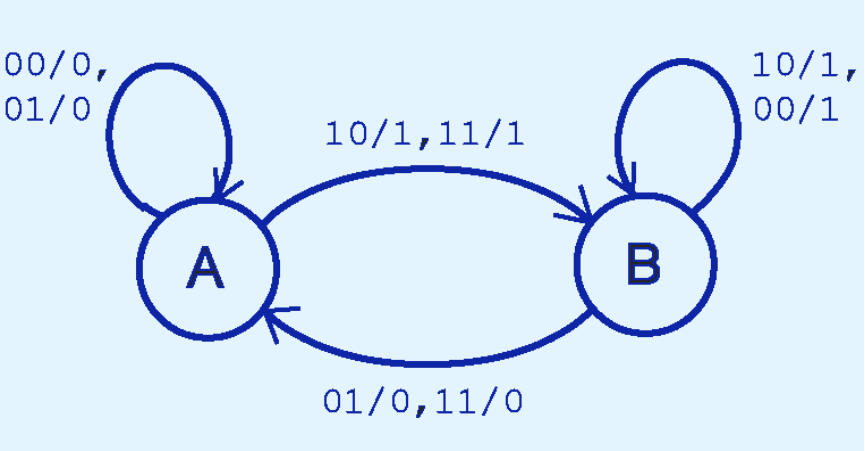
FSM's

- A JK flip-flop is an example of a relatively simple FSM

MOORE Machine



MEALY Machine



FSM

- A computer program is another example of a FSM
- It accepts a set of defined inputs, and based on some sequence of inputs, produces a sequence of outputs.

FSM - Example

- One of the first actions that a compiler must perform is to “tokenize” the input string. This is generally called lexical analysis.
- Let’s say that our programming language consists of the following atomic symbols
  - Arithmetic operators +, -, *, /
  - Integers <digit>{<digit>}$_5$
  - Variable names <letter>{<letter>|<digit>}$_4$
  - The machine should output a 1 for an arithmetic operator, a 2 for an integer, and a 3 for a variable name. Also, output a 0 for no token seen or token not complete.
  - White space is ignored, and no other characters are legal or possible.
  - Implement this FSM with a mealy machine

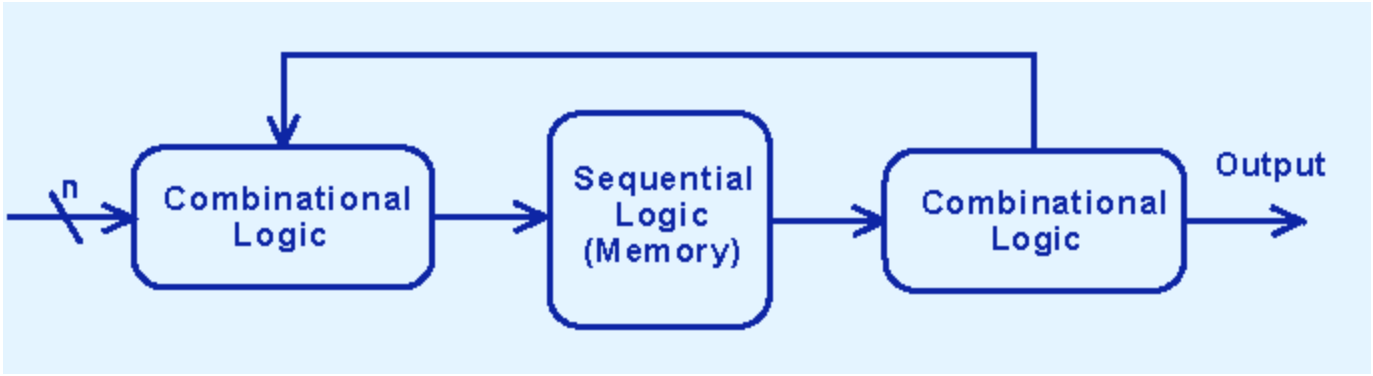
FSM - Example

- $\{\langle digit \rangle\}_n$ means 0 to n digits
- $\{\langle letter \rangle | \langle digit \rangle\}_n$ means a total of 0 to n letters and/or digits
- White space is a space, a tab, a newline, …

Moore vs Mealy

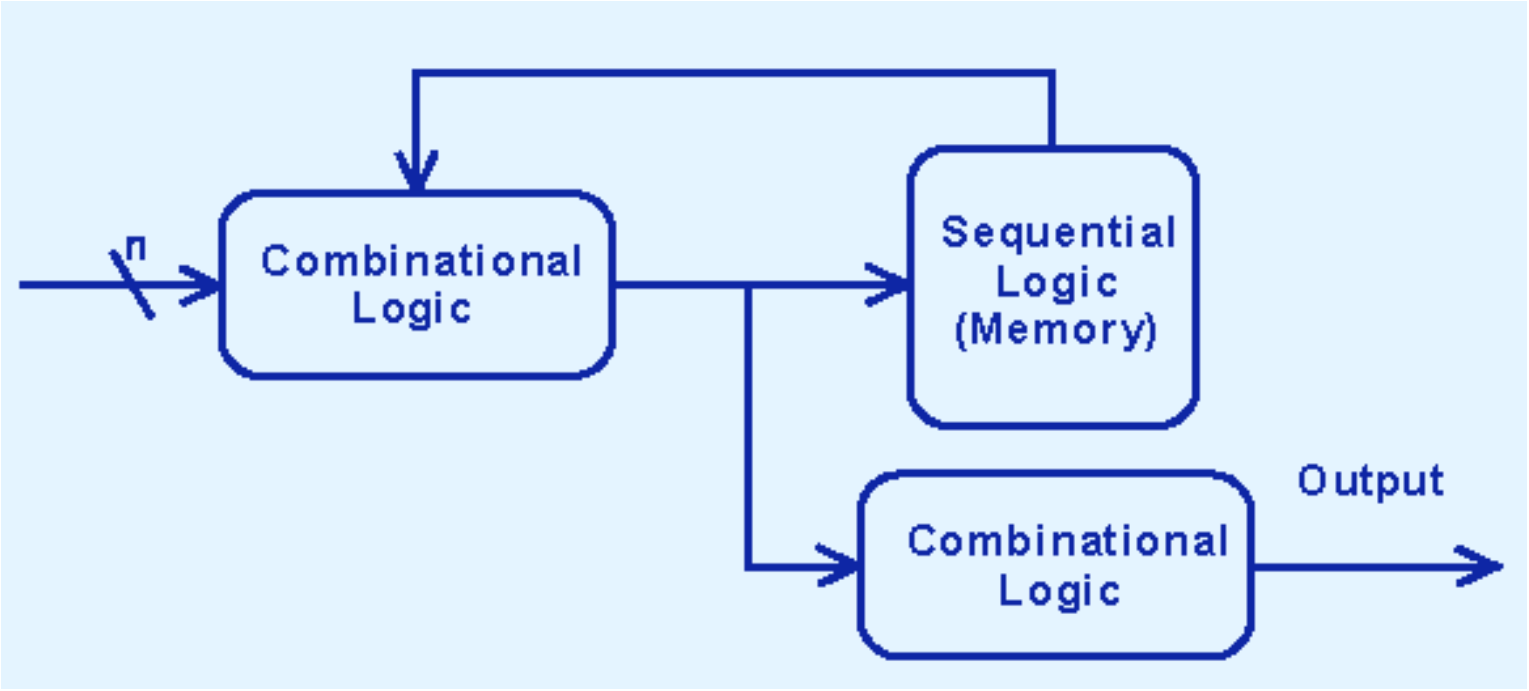
- Functionally, Moore and Mealy machines are the same. Their implementations differ.

Moore Machine



Moore vs Mealy

Mealy Machine

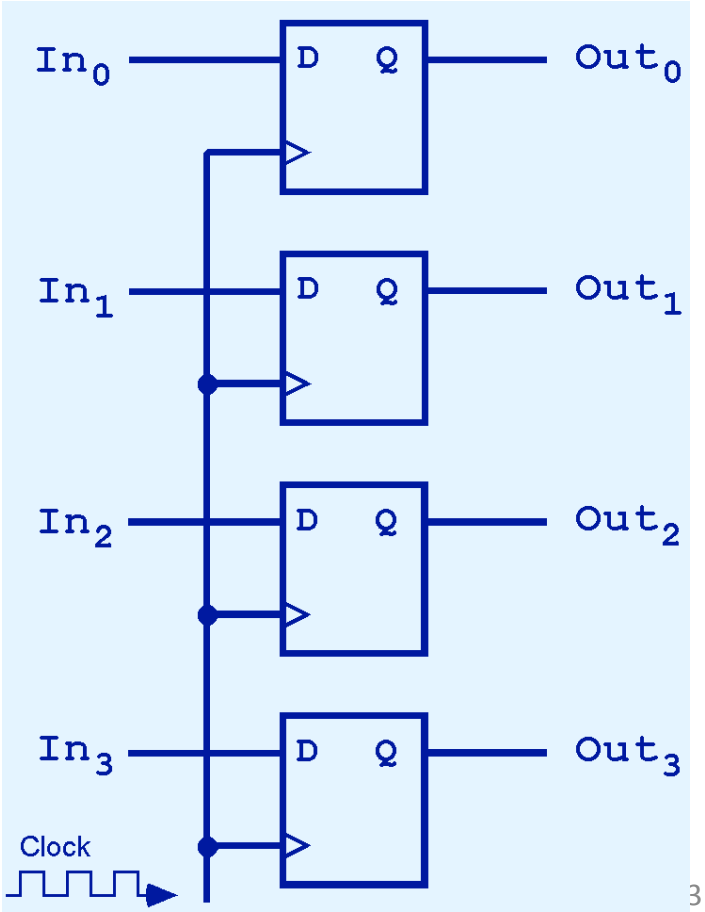


Other FSM's

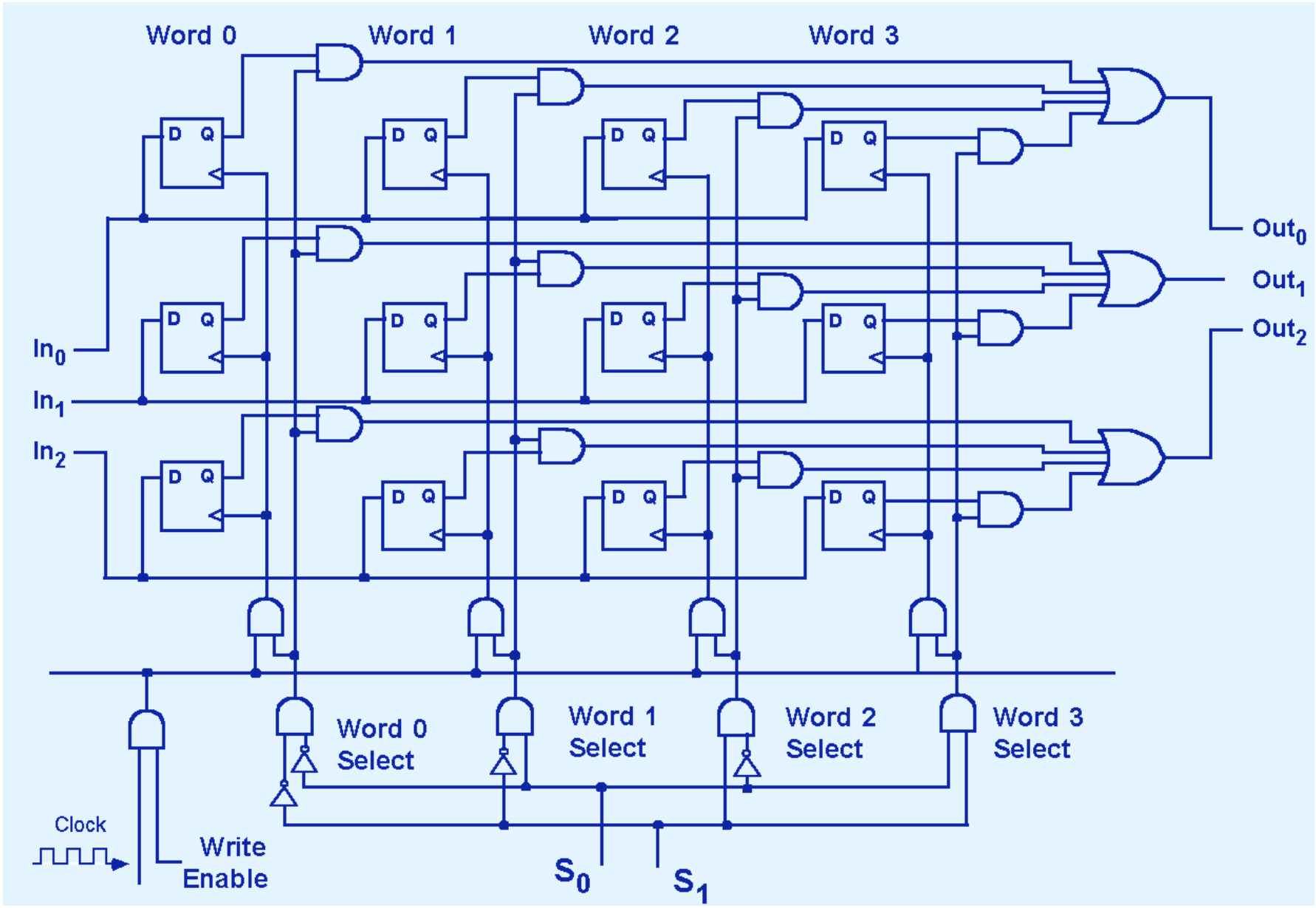
- Soda Pop Machine
- Elevator
- ????

Sequential Circuit Examples

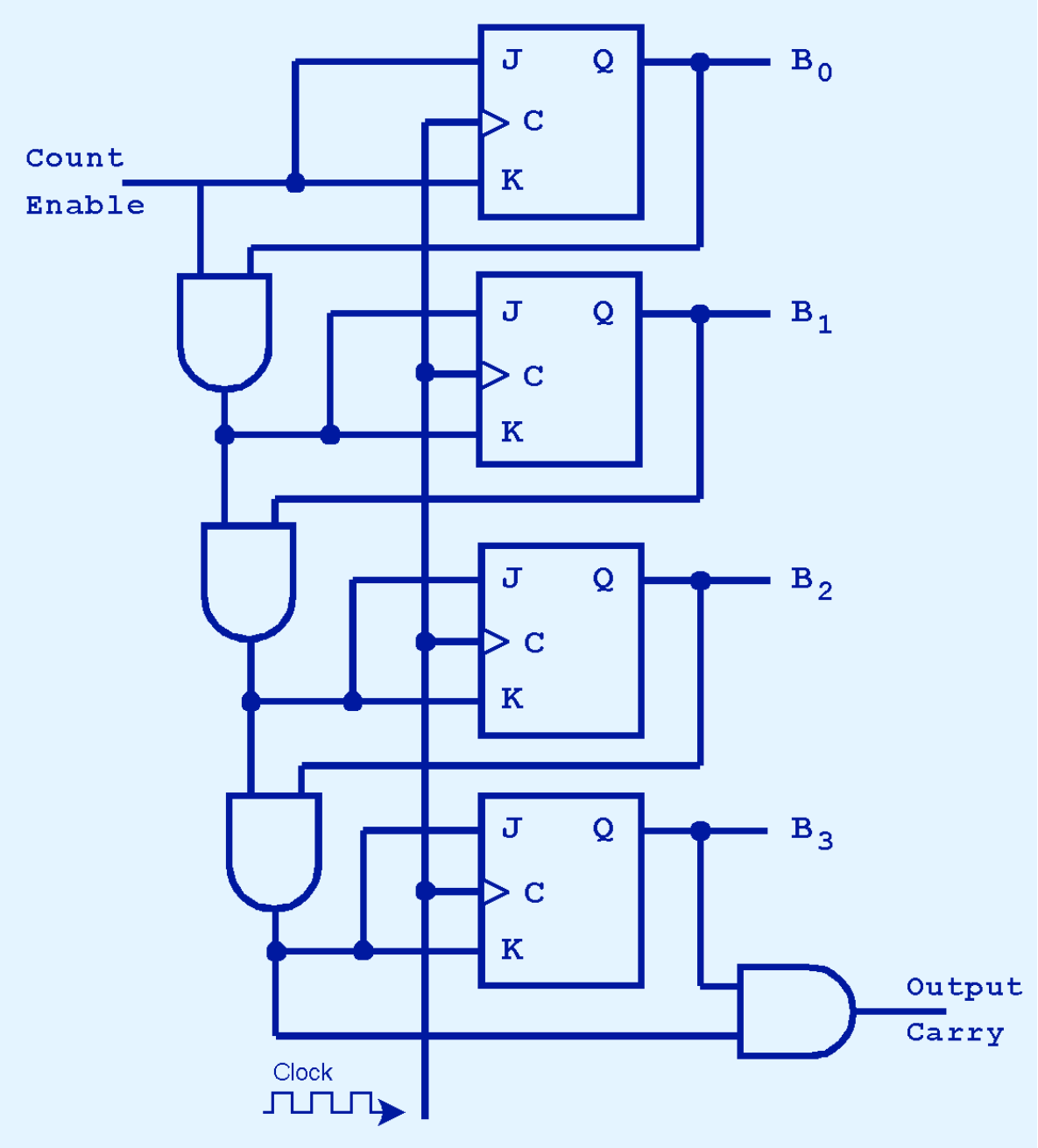
- 4-bit register (made up of D flip flops)



Memory


Word 0
Word 1
Word 2
Word 3
D Q
In_0
In_1
In_2
Out_0
Out_1
Out_2
Word 0 Select
Word 1 Select
Word 2 Select
Word 3 Select
Clock
Write Enable
S_0
S_1

Binary Counter



Sequence Detector

- In a digital communication system, one of the challenges is to determine when a transmission has begun. This is often done by appending to the front of a message a special string of 1’s and 0’s. In this way, random noise cannot be seen as the start of a message

Sequence Detector

- Develop a FSM which accepts as input a string of 1’s and 0’s and output a 1 whenever the sequence 101100 is seen.