

Boolean Algebra

- Boolean algebra is a mathematical system for the manipulation of variables that can have one of two values.
- A Boolean function has:
 - At least one Boolean variable, At least one Boolean operator, and At least one input from the set $\{0,1\}$.
- It produces an output that is also a member of the set $\{0,1\}$.

- The truth table for the Boolean function:

$$F(x, y, z) = x\bar{z} + y$$

is shown at the right.

- To make evaluation of the Boolean function easier, the truth table contains extra (shaded) columns to hold evaluations of subparts of the function.

$$F(x, y, z) = x\bar{z} + y$$

| x | y | z | \bar{z} | $x\bar{z}$ | $x\bar{z} + y$ |
|---|---|---|-----------|------------|----------------|
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |

- The simpler that we can make a Boolean function, the smaller the circuit that will result.

- Most Boolean identities have an AND (product) form as well as an OR (sum) form. We give our identities using both forms. Our first group is rather intuitive:

| Identity Name | AND Form | OR Form |
|----------------|----------------|-------------------|
| Identity Law | $1x = x$ | $0 + x = x$ |
| Null Law | $0x = 0$ | $1 + x = 1$ |
| Idempotent Law | $xx = x$ | $x + x = x$ |
| Inverse Law | $x\bar{x} = 0$ | $x + \bar{x} = 1$ |

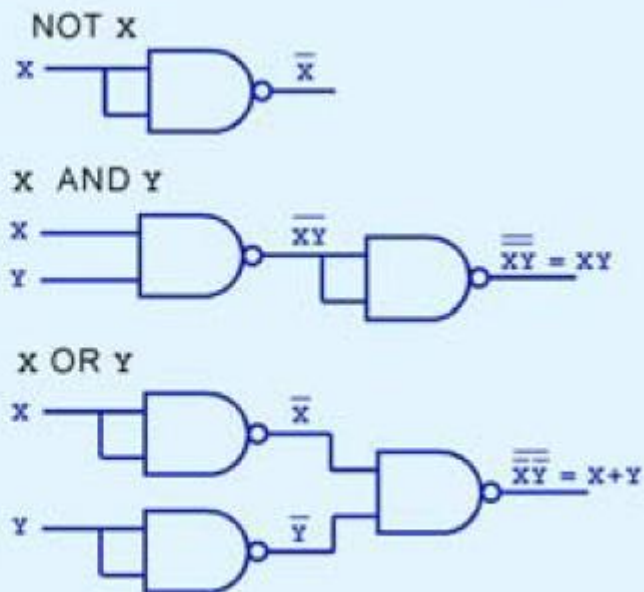
| Identity Name | AND Form | OR Form |
|------------------|---------------------|-----------------------|
| Commutative Law | $xy = yx$ | $x+y = y+x$ |
| Associative Law | $(xy)z = x(yz)$ | $(x+y)+z = x + (y+z)$ |
| Distributive Law | $x+yz = (x+y)(x+z)$ | $x(y+z) = xy+xz$ |

| Identity Name | AND Form | OR Form |
|-----------------------|---------------------------------------|-------------------------------------|
| Absorption Law | $x(x+y) = x$ | $x + xy = x$ |
| DeMorgan's Law | $\overline{(xy)} = \bar{x} + \bar{y}$ | $\overline{(x+y)} = \bar{x}\bar{y}$ |
| Double Complement Law | $\overline{(\bar{x})} = x$ | |

- We can use Boolean identities to simplify the function: $F(X, Y, Z) = (X + Y) (X + \bar{Y}) \overline{(XZ)}$ as follows:

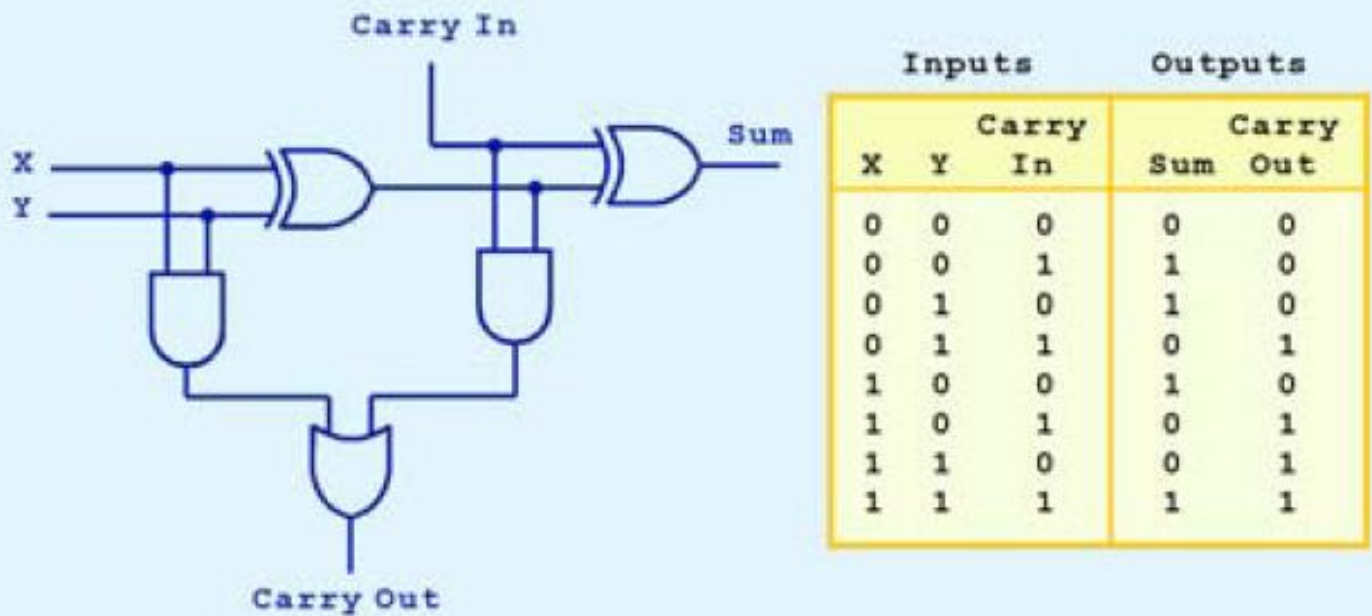
| | |
|---|---------------------------------|
| $(X + Y) (X + \bar{Y}) \overline{(XZ)}$ | Idempotent Law (Rewriting) |
| $(X + Y) (X + \bar{Y}) (\bar{X} + Z)$ | DeMorgan's Law |
| $(XX + X\bar{Y} + XY + Y\bar{Y}) (\bar{X} + Z)$ | Distributive Law |
| $((X + Y\bar{Y}) + X(Y + \bar{Y})) (\bar{X} + Z)$ | Commutative & Distributive Laws |
| $((X + 0) + X(1)) (\bar{X} + Z)$ | Inverse Law |
| $X(\bar{X} + Z)$ | Idempotent Law |
| $X\bar{X} + XZ$ | Distributive Law |
| $0 + XZ$ | Inverse Law |
| XZ | Idempotent Law |

- NAND and NOR are known as *universal gates* because they are inexpensive to manufacture and any Boolean function can be constructed using only NAND or only NOR gates.

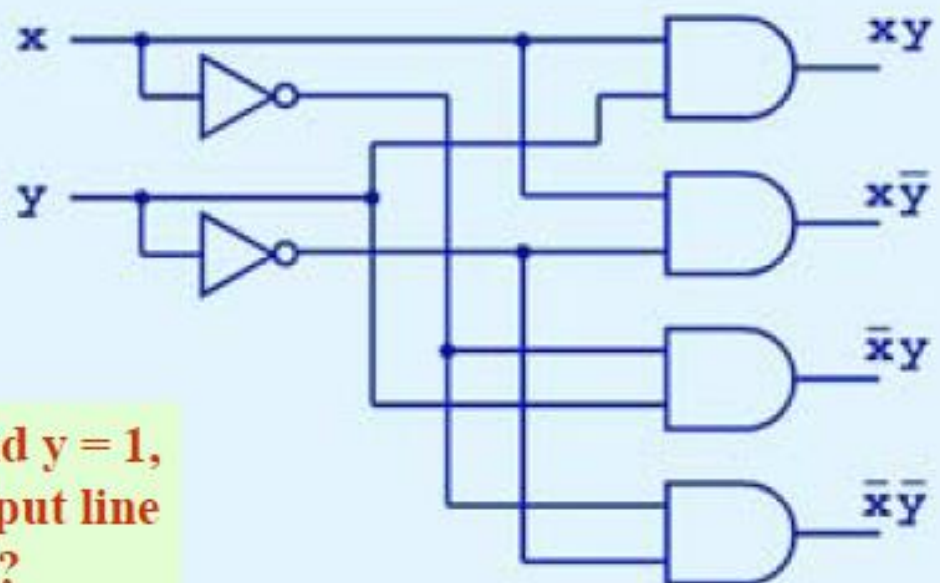


• Combinational Circuits

- Here's our completed full adder.

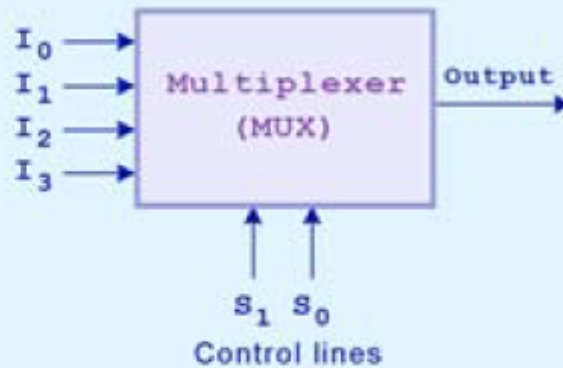


- This is what a 2-to-4 decoder looks like on the inside.



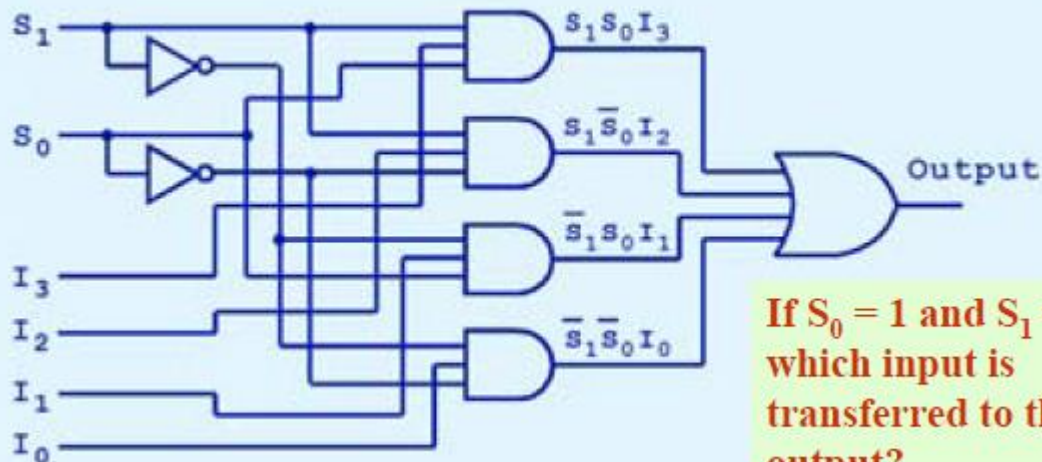
**If $x = 0$ and $y = 1$,
which output line
is enabled?**

- A multiplexer does just the opposite of a decoder.
- It selects a single output from several inputs.
- The particular input chosen for output is determined by the value of the multiplexer's control lines.
- To be able to select among n inputs, $\log_2 n$ control lines are needed.



This is a block diagram for a multiplexer.

- This is what a 4-to-1 multiplexer looks like on the inside.

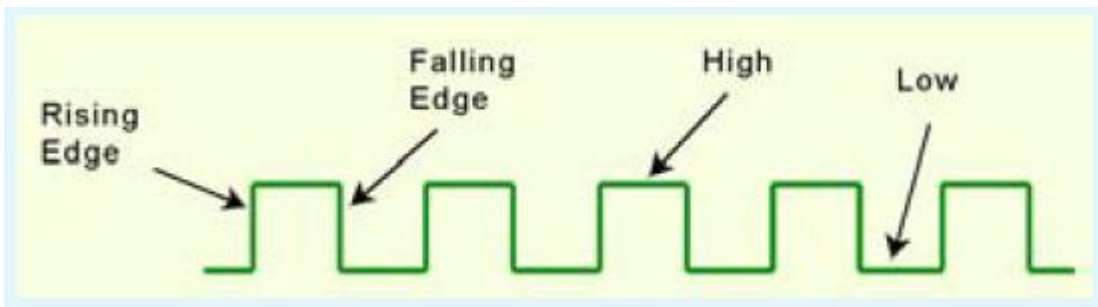


If $S_0 = 1$ and $S_1 = 0$, which input is transferred to the output?

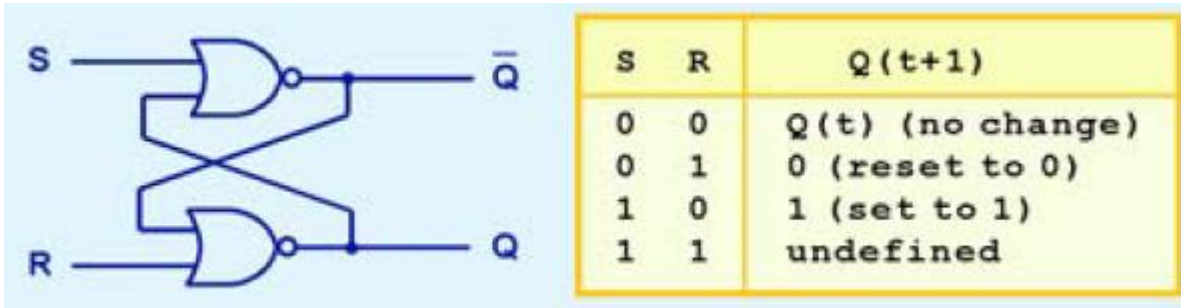
• Sequential Circuits

- There are other times, however, when we need a circuit to change its value with consideration to its current state as well as its inputs.

- These circuits have to “remember” their current state.
 - Sequential logic circuits provide this functionality for us.
- As the name implies, sequential logic circuits require a means by which events can be sequenced.
 - State changes are controlled by clocks.
 - A “clock” is a special circuit that sends electrical pulses through a circuit.
 - Clocks produce electrical waveforms such as the one shown below.
- State changes occur in sequential circuits only when the clock ticks.
- Circuits can change state on the rising edge, falling edge, or when the clock pulse reaches its highest voltage.



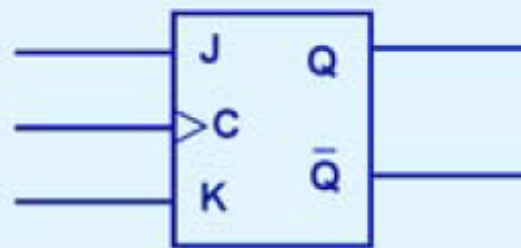
- Circuits that change state on the rising edge, or falling edge of the clock pulse are called edgetriggered.
- Level-triggered circuits change state when the clock voltage reaches its highest or lowest level.
- You can see how feedback works by examining the most basic sequential logic components, the SR flip-flop.
 - The “SR” stands for set/reset.
 - The internals of an SR flip-flop are shown below, along with its block diagram.
- The behavior of an SR flip-flop is described by a characteristic table.
 - $Q(t)$ means the value of the output at time t .
 - $Q(t+1)$ is the value of Q after the next clock pulse.



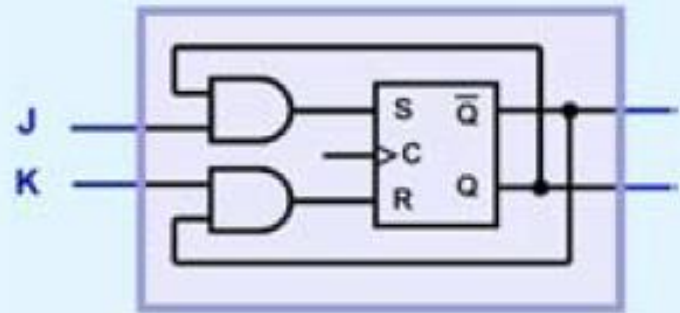
- The SR flip-flop actually has three inputs: S, R, and its current output, Q.
- Thus, we can construct a truth table for this circuit, as shown at the right.
- Notice the two undefined values. When both S and R are 1, the SR flip-flop is unstable.

| 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 |

- If we can be sure that the inputs to an SR flip-flop will never both be 1, we will never have an unstable circuit. This may not always be the case.
- The SR flip-flop can be modified to provide a stable state when both inputs are 1.
- This modified flip-flop is called a JK flip-flop, shown at the right.
 - The “JK” is in honor of Jack Kilby.

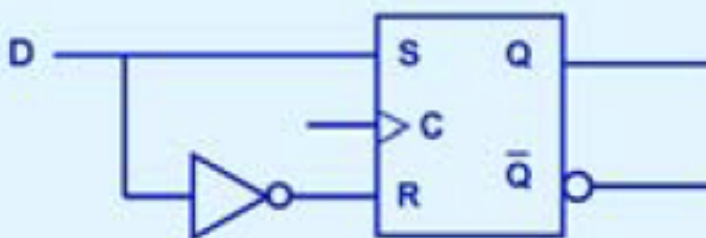


- At the right, we see how an SR flip-flop can be modified to create a JK flip-flop.
- The characteristic table indicates that the flip-flop is stable for all inputs.



| 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 | $\bar{Q}(t)$ |

- Another modification of the SR flip-flop is the D flip-flop, shown below with its characteristic table.
- You will notice that the output of the flip-flop remains the same during subsequent clock pulses. The output changes only when the value of D changes.

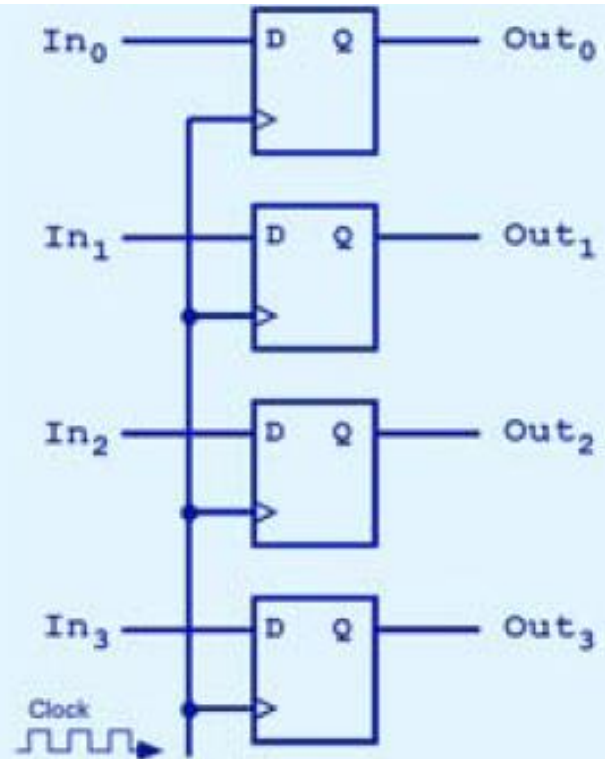


| D | $Q(t+1)$ |
|---|----------|
| 0 | 0 |
| 1 | 1 |

- This illustration shows a 4-bit register consisting of D flip-flops. You will usually see its block diagram (below) instead.



A larger memory configuration is in your text.



- A binary counter is another example of a sequential circuit.
- The low-order bit is complemented at each clock pulse.
- Whenever it changes from 0 to 1, the next bit is complemented, and so on through the other flip-flops.

