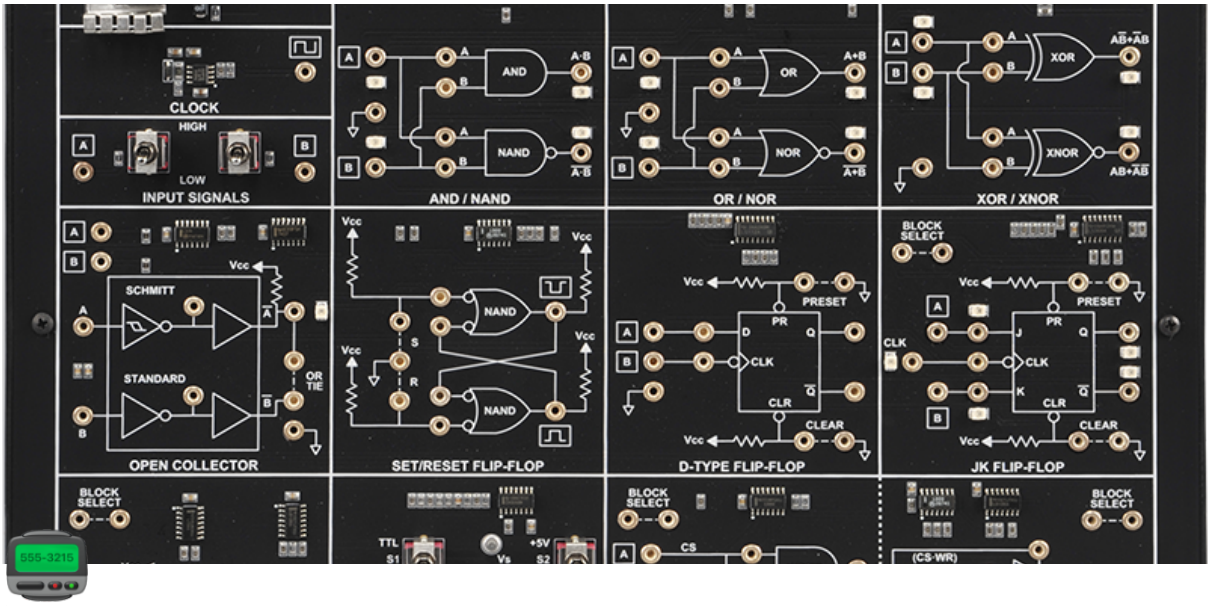


Digital Logic

Table of Contents

1. What is digital logic?

Digital logic refers to the foundational principles and techniques used in the design and implementation of electronic circuits that process binary data. At its core, digital logic involves the manipulation of binary signals using various logical operations, such as AND, OR, NOT, and XOR, performed by electronic components known as logic gates. These logic gates, composed of transistors and other electronic elements, execute predefined functions based on the input signals they receive. Digital logic is essential in modern electronics, enabling the creation of complex digital systems capable of performing tasks ranging from simple arithmetic operations to complex data processing and control functions. Understanding digital logic is crucial for engineers and

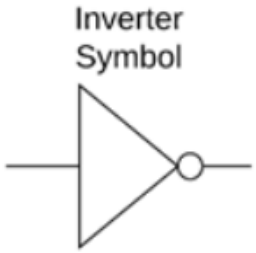
designers working in fields such as computer engineering, telecommunications, and embedded systems development, as it forms the basis for the design and operation of digital devices and systems.

2. Basic digital components

There are in total 4 basic logic components: NOT(Inverter), AND Gate, OR Gate and XOR.

2.1. NOT (Inverter):

The inverter is a logic element with one input and one output, which inverts the input signal. For instance, if the input signal is 1 the output will be 0 and vice versa.

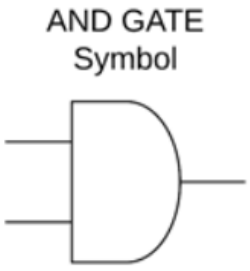


NOT(Inverter) Symbol - IEEE standard

Input A	Output Y
0	1
1	0

2.2. AND Gate:

The AND Gate is a logic element with multiple inputs and one output. This Gate as shown previously have the property to output a logical high signal (1) when all the inputs are logical high (1).IEE

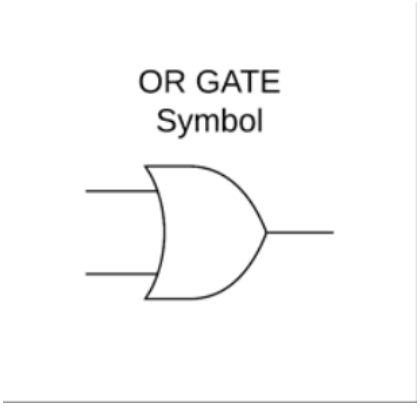


AND Gate Symbol - IEEE standard

Input A	Input B	Output Y
0	0	0
0	1	0
1	0	0
1	1	1

2.3. OR Gate:

The OR Gate is a logic element with multiple inputs and one output. This Gate has the property to output a logical high signal (1) when at least one of the inputs is a logical high (1).

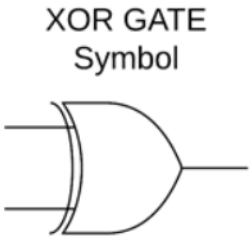


OR Gate Symbol - IEEE standard

Input A	Input B	Output Y
0	0	0
0	1	1
1	0	1
1	1	1

2.4. XOR Gate:

The XOR Gate is a logic element which is composed of an AND Gate with inverter on its output (NAND Gate) and a OR Gate, connected to the inputs of an And Gate. This composition gives the XOR the special property to output a logical high (1) when only one of the inputs is logical high (1).



XOR Gate Symbol - IEEE standard

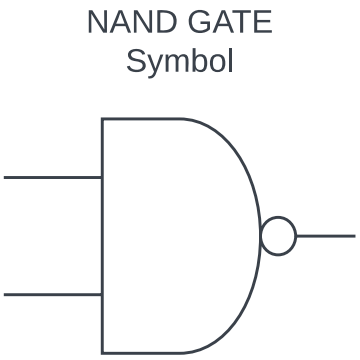
Input A	Input B	Output Y
0	0	0
0	1	1
1	0	1
1	1	0

3. Universal logic gates

Universal logic gates are fundamental building blocks in digital circuit design that can be used to implement any logical function. In other words, with just one type of universal gate, you can create circuits that perform any logical operation, including AND, OR, NOT, NAND, NOR, and XOR.

3.1. NAND Gate:

The NAND Gate is a logic element with multiple inputs and one output. It produces a low (0) output only when both of its inputs are high (1). In other words, it behaves like an AND gate followed by a NOT gate.

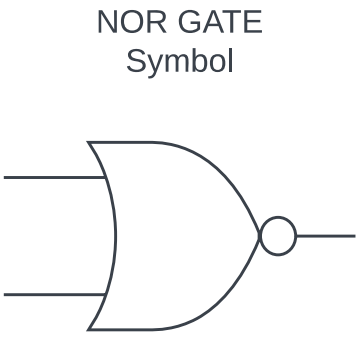


NAND Gate Symbol - IEEE standard

Input A	Input B	Output Y
0	0	1
0	1	1
1	0	1
1	1	0

3.2. NOR Gate:

The NAND Gate is a logic element with multiple inputs and one output. It produces a high (1) output only when both of its inputs are low (0). It behaves like an OR gate followed by a NOT gate.



NOR Gate Symbol - IEEE standard

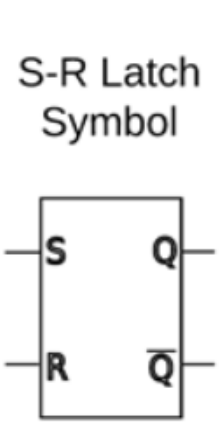
Input A	Input B	Output Y
0	0	1
0	1	0
1	0	0
1	1	0

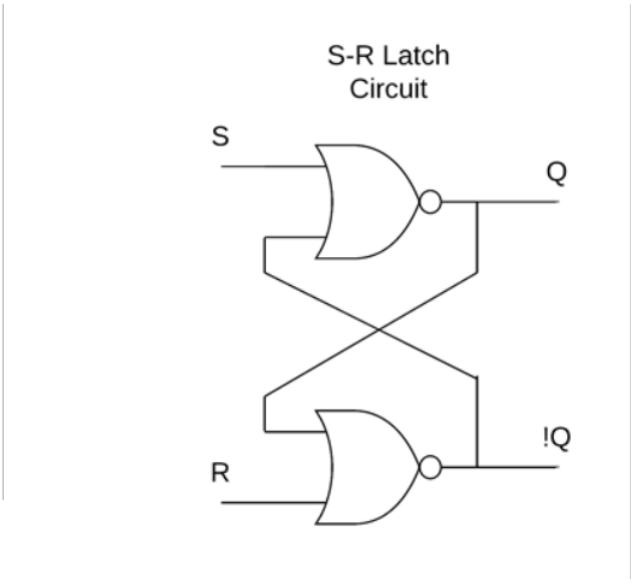
4. Latches and Flip-Flops

Latches and flip-flops are key elements in digital circuits for storing binary data. Latches operate continuously based on control signals, while flip-flops synchronize data storage with clock signals. Latches are common in asynchronous systems, whereas flip-flops are essential for synchronous applications. Both are vital for digital system design.

4.1. SR Latch

The S-R Latch is a logic circuit most notably used in the creation of memory. It consists of two NAND or NOR Gates, where one of the inputs of the gates are connected to the opposite Gate’s output. This way by giving an input signal to the free inputs of the Logic Gates (S-set, R-reset), it is possible to get an output signal (and its compliment), which will be remembered by the circuit.

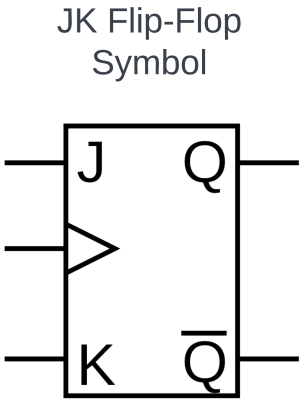


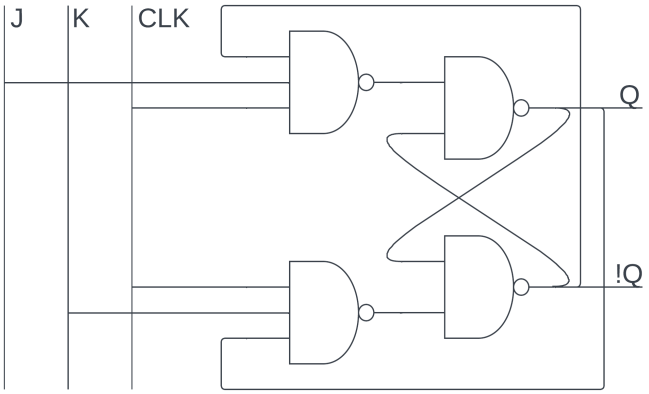


Input A	Input B	Output Q	Output !Q
0	0	Q	!Q
0	1	0	1
1	0	1	0
1	1	Invalid	Invalid

4.2 JK Flip-Flop

The JK flip-flop is another fundamental component in digital electronics, used for storing a single bit of data or as a memory cell. It's an extension of the SR flip-flop with additional features for overcoming the problem of indeterminate states.

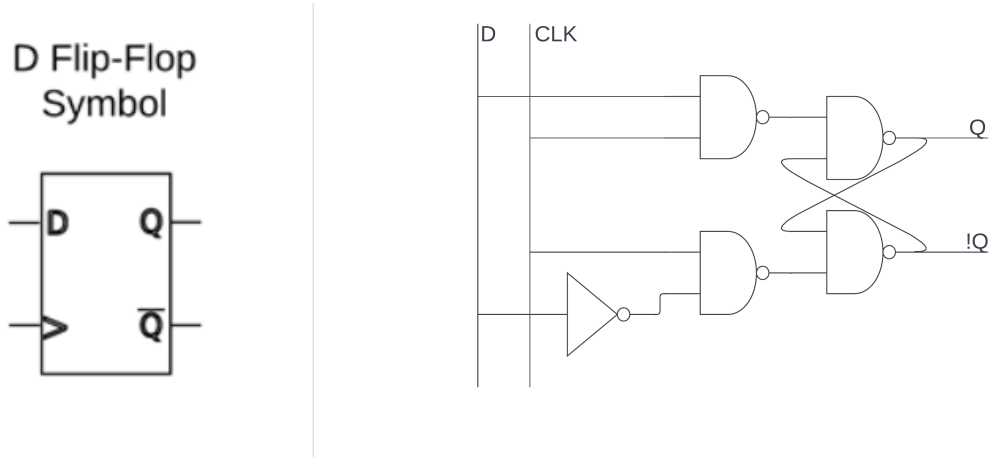




Input CLK	Input J	Input K	Output Q	Output !Q
1	X	X	Q	!Q
1	1	0	1	0
1	0	0	1	0
1	0	1	0	1
1	0	1	0	1

4.3. D Flip-Flop

The D Flip-Flop is a logic circuit which is based on the S-R Latch and could be thought of a basic memory cell. It has two inputs (D – data, CLK – clock pulse) and two output (Q, !Q). Internally the D input is split and connected to two separate NAND Gates, it is connected directly to the first and inverted then connected to the second. The second inputs of the NAND Gates are connected to the CLK, which acts as a control mechanism. When the CLK rises to logic high the D input signal can pass through the Flip-Flop.



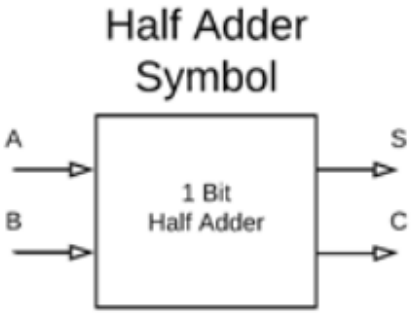
Input D	Input CLK	Output Q	Output !Q
0	0	Q	!Q
0	1	Q	!Q
1	0	0	1
1	1	1	0

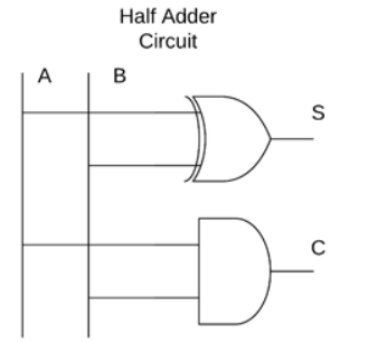
5. Digital logic circuits

A digital logic circuit, foundational in digital electronics, processes binary data using logic gates like AND, OR, NOT, and XOR. Examples include the Half Adder and Full Adder for arithmetic operations, as well as multiplexers and demultiplexers for signal routing and data selection. These circuits enable efficient communication and data distribution within digital systems, facilitating a wide range of functionalities with reliability and efficiency.

5.1. Half Adder

The Half Adder along with the Full Adder are a type of adder circuits. The adder circuit is a combinational digital circuit that is used for adding two numbers. A typical adder circuit produces a sum bit (denoted by S) and a carry bit (denoted by C) as the output. Typically, adders are realized for adding binary numbers, but they can be also realized for adding other formats like BCD. The Half adder is no exception to this rule. If A and B are the input bits, then sum bit (S) is the X-OR of A and B and the carry bit (C) will be the AND of A and B.

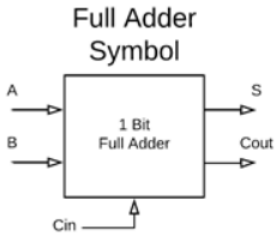


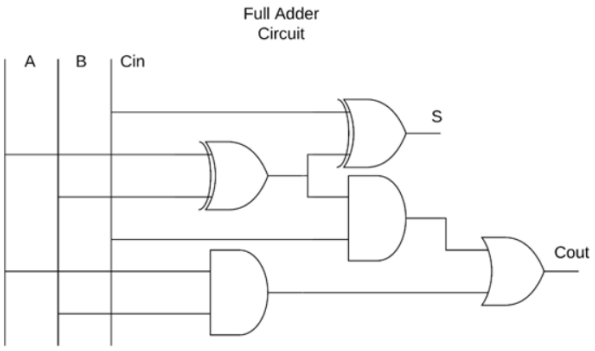


Input A	Input B	Output S	Output C
0	0	0	0
0	1	1	0
1	1	0	1
1	0	1	0

5.2. Full Adder

The Full adder is an extended version of the Half Adder. Full adders are a logic circuit that adds two input operand bits plus a Carry in bit and outputs a Carry out bit and a sum bit. The Sum out (Sout) of a full adder is the XOR of input operand bits A, B, and the Carry in (Cin) bit.

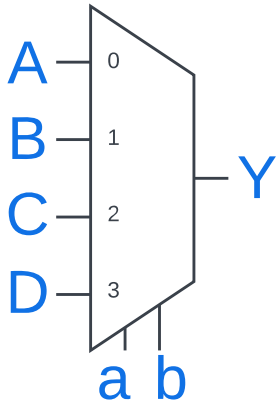


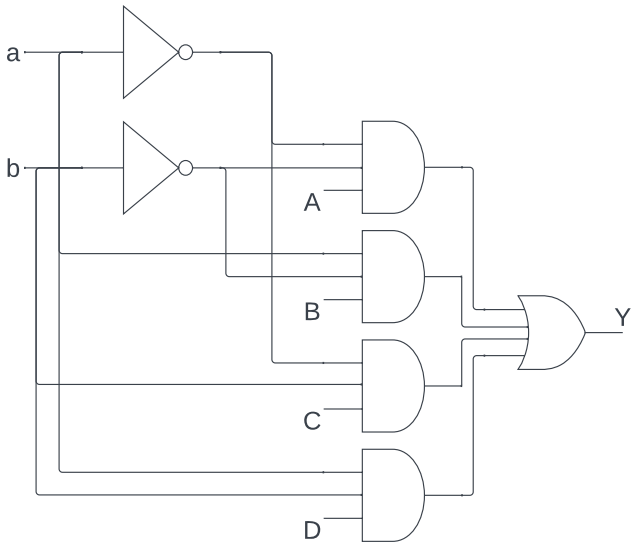


Input A	Input B	Input Cin	Output S	Output Cout
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	1	1	1

5.3. Multiplexer

A multiplexer (often abbreviated as "mux") is a digital circuit used to select one of multiple input signals and route it to a single output. It's like a data selector that chooses the desired input based on the control inputs. The select lines determine which input gets routed to the output. The number of select lines determines the number of inputs the multiplexer can handle. For example, a 2-to-1 multiplexer has one select line, while an 8-to-1 multiplexer has 3 select lines.

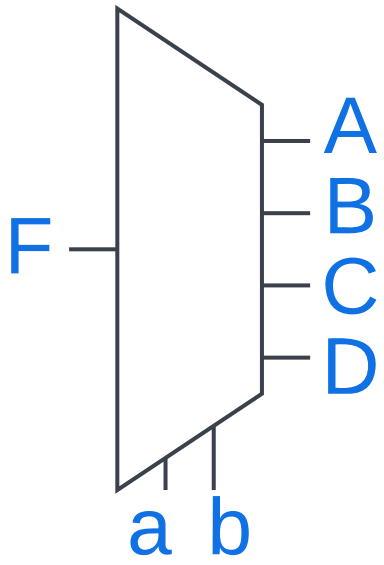


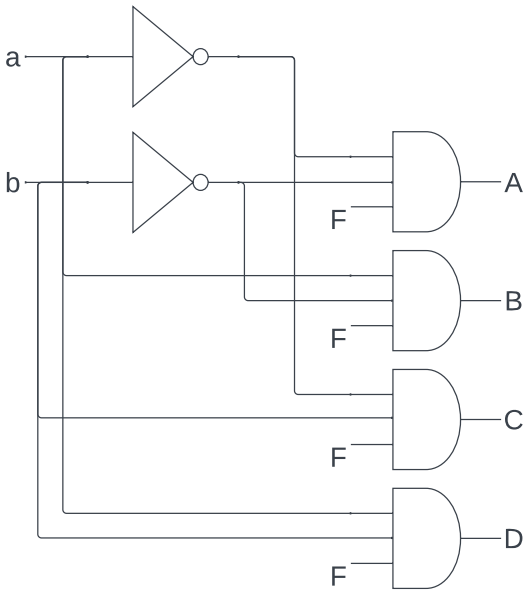


Input a	input b	Input A	Input B	Input C	Input D	Output Y
0	0	1	X	X	X	1
0	1	X	1	X	X	1
1	0	X	X	1	X	1
1	1	X	X	X	1	1

5.4. Demultiplexer

A demultiplexer (often abbreviated as "demux") is the opposite of a multiplexer. It takes a single input and routes it to one of several possible output lines based on control signals. Similar to a multiplexer, a demultiplexer has control inputs that determine which output line the input signal gets routed to.

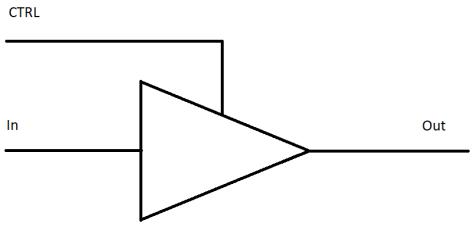




Input a	Input b	Input F	Output A	Output B	Output C	Output D
0	0	1	1	X	X	X
0	1	1	X	1	X	X
1	0	1	X	X	1	X
1	1	1	X	X	X	1

6. Tri-state logic

A three-state logic buffer, also known as a tri-state buffer, is a digital circuit that can assume one of three possible output states: high (logic 1), low (logic 0), or high impedance (Z). The high impedance state effectively disconnects the output from the rest of the circuit, allowing multiple buffers to share a common bus without interference.



CTRL	In	Out
0	X	Hi-Z
1	0	0
1	1	1