TOSHIBA

e-Learning

Basics of CMOS Logic ICs

Chapter3 Basic CMOS Logic ICs

Toshiba Electronic Devices & Storage Corporation

03

Basic CMOS Logic ICs

Operations of typical ICs

3.1 Basic CMOS Logic ICs

CMOS logic ICs are broadly divided into combinational and sequential logic. This section describes major logic circuits.

CMOS logic ICs

Combinational logic

Type of logic circuits whose output is a pure function of the present input(s) only

Inverters, buffers, bidirectional bus buffers, Schmitt-trigger devices, decoders, multiplexers, analog multiplexer/demultiplexers, analog switches, etc.

Sequential logic

Type of logic circuits whose output depends on the sequence of previous input values and is controlled by the present inputs such as control signals

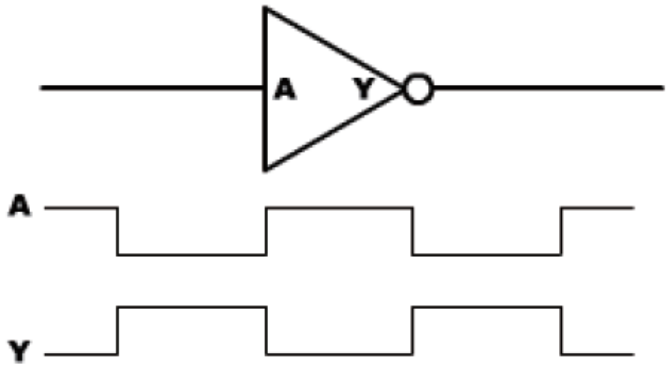
Flip-flops, latches, counters, shift registers, etc.

3.2.1 Combinational Logic: Inverters and buffers

Inverters

Example: 74VHC04

An inverter is a logic gate whose output (X) is the inverse of its input (A).



The output is the opposite of the input.

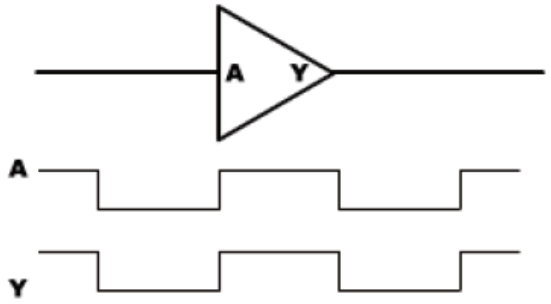
Operation of an inverter (Example: VHC04)

3.2.1 Combinational Logic: Inverters and buffers

Buffers

Example: 74VHC244

A buffers increase the drive capability to increase the number of connectable signal lines, and perform waveform shaping.
A buffer does not perform logical operations.



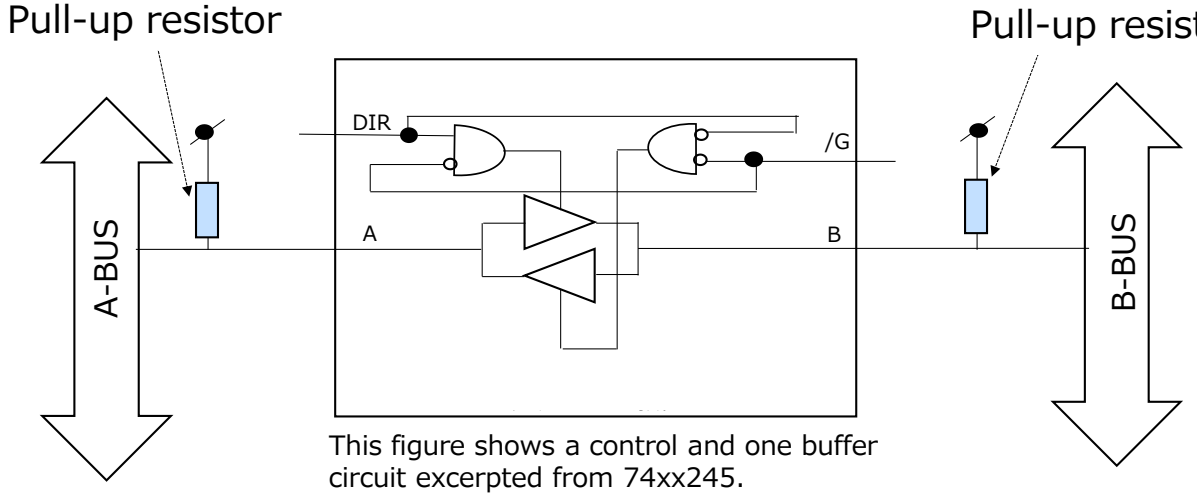
Operation of a buffer (Example: VHC244)

3.2.2 Combinational Logic: Bidirectional bus buffers

Bidirectional bus buffers (transceivers)

Example: 74VHC245

There is a type of buffer called a bidirectional bus buffer (transceiver) that can both transmit and receive data with a single input/output terminal. Transceivers can switch the direction of passing signals with a control signal (DIR), so they are used for bus lines that pass data in both directions. A usage example is shown below.
When using bus signals bi-directionally, connect both the bus input and bus output to VCC or GND via a pull-up resistor to prevent the input signal from becoming open (undefined) when the signal is switched by the control signal (DIR).
Please be careful not to short-circuit the output with the output of the bus line when switching signals.

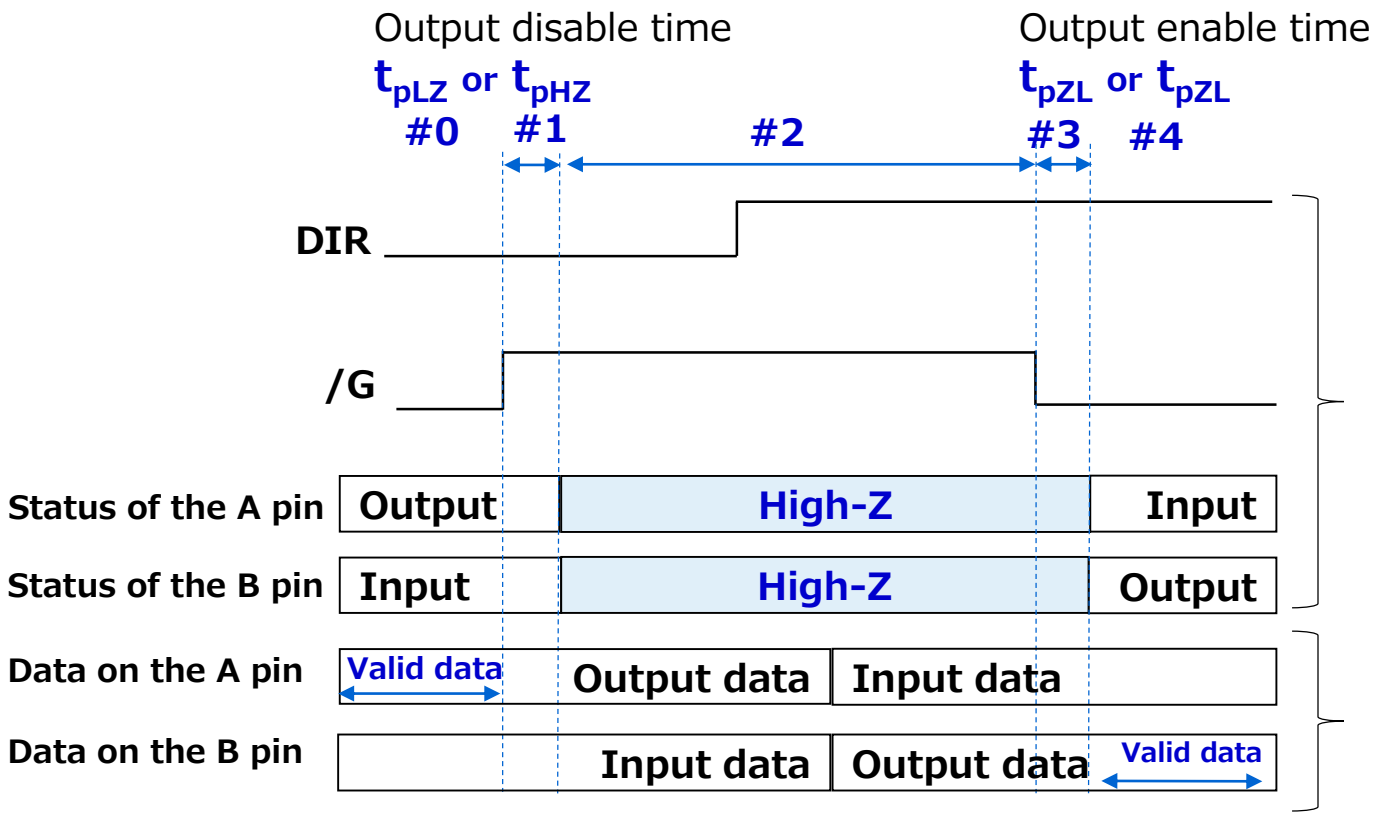


This figure shows a control and one buffer circuit excerpted from 74xx245.

Example of application of a bidirectional bus buffer

3.2.2 Combinational Logic: Bidirectional bus buffers

The direction of the A and B pins can be changed easily by changing the value of DIR while /G is High.



Change the value of DIR and the direction of external data while /G is High.

Although the directions of the A and B pins can be changed easily, care should be exercised as to the periods during which data are valid.

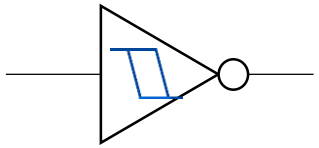
Change the value of DIR and the direction of external data while /G is High.
During period #0, data are transferred from B to A.
During period #1, the A pin assumes the High-Z state. Therefore, output data are invalid.
During period #2, change the value of DIR and the direction of external data.
During period #3, enable the A and B pins. Then, output data stabilize at the beginning of period #4.
During period #4, data are transferred from A to B.

3.2.3 Combinational Logic: Schmitt-trigger devices

Schmitt-trigger devices

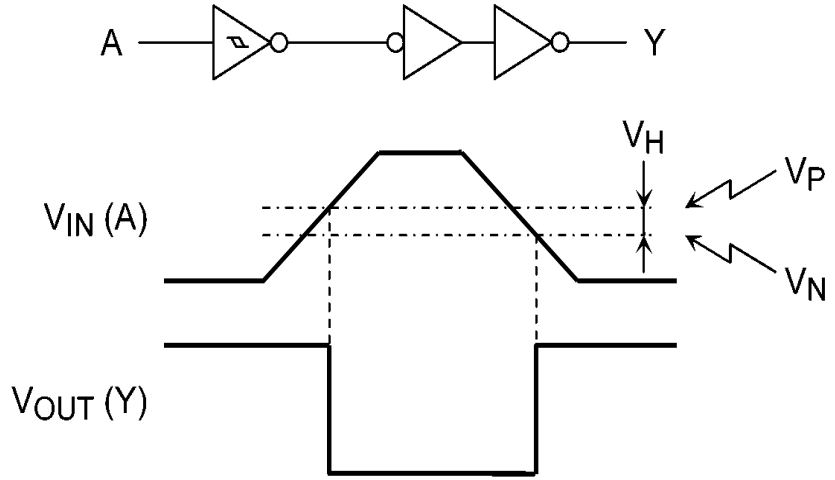
Example: VHC14

A Schmitt-trigger device has a hysteresis band between two input threshold voltages.
The following shows the input and output waveforms of a Schmitt inverter with an input threshold hysteresis.
In the case of an IC with hysteresis, the positive-going threshold voltage (V_P) differs from the negative-going threshold voltage (V_N). For a slowly rising or falling input, an input threshold hysteresis (V_H) helps stabilize the output. Even in the presence of input noise or power supply or ground bounce due to noise, an IC does not produce a false output unless the noise or bounce exceeds the hysteresis width.



Logic symbol

Schmitt inverter



Input and output waveforms of a Schmitt inverter

3.2.4 Combinational Logic: Decoders

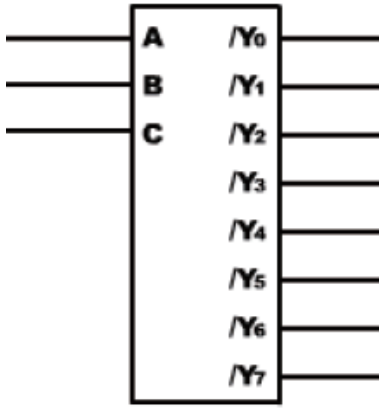
Decoders

Example: VHC138

A decoder converts binary information from the N coded inputs to a maximum of 2^N unique outputs. It is commonly used to increase the number of ports and generate chip select signals.

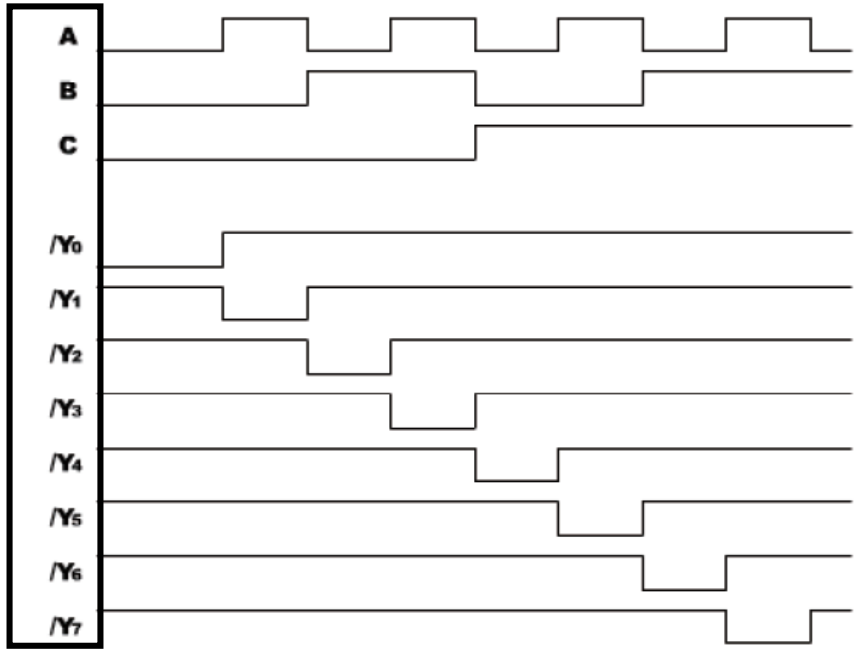
Operation of a decoder

The following shows a logic symbol, truth table, and timing diagram of a 3-to-8 decoder, i.e., a decoder with three inputs and eight outputs.



Input			Output							
A	B	C	/Y0	/Y1	/Y2	/Y3	/Y4	/Y5	/Y6	/Y7
L	L	L	L	H	H	H	H	H	H	H
H	L	L	H	L	H	H	H	H	H	H
L	H	L	H	H	L	H	H	H	H	H
H	H	L	H	H	H	L	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H
H	L	H	H	H	H	H	H	L	H	H
L	H	H	H	H	H	H	H	H	L	H
H	H	H	H	H	H	H	H	H	H	L

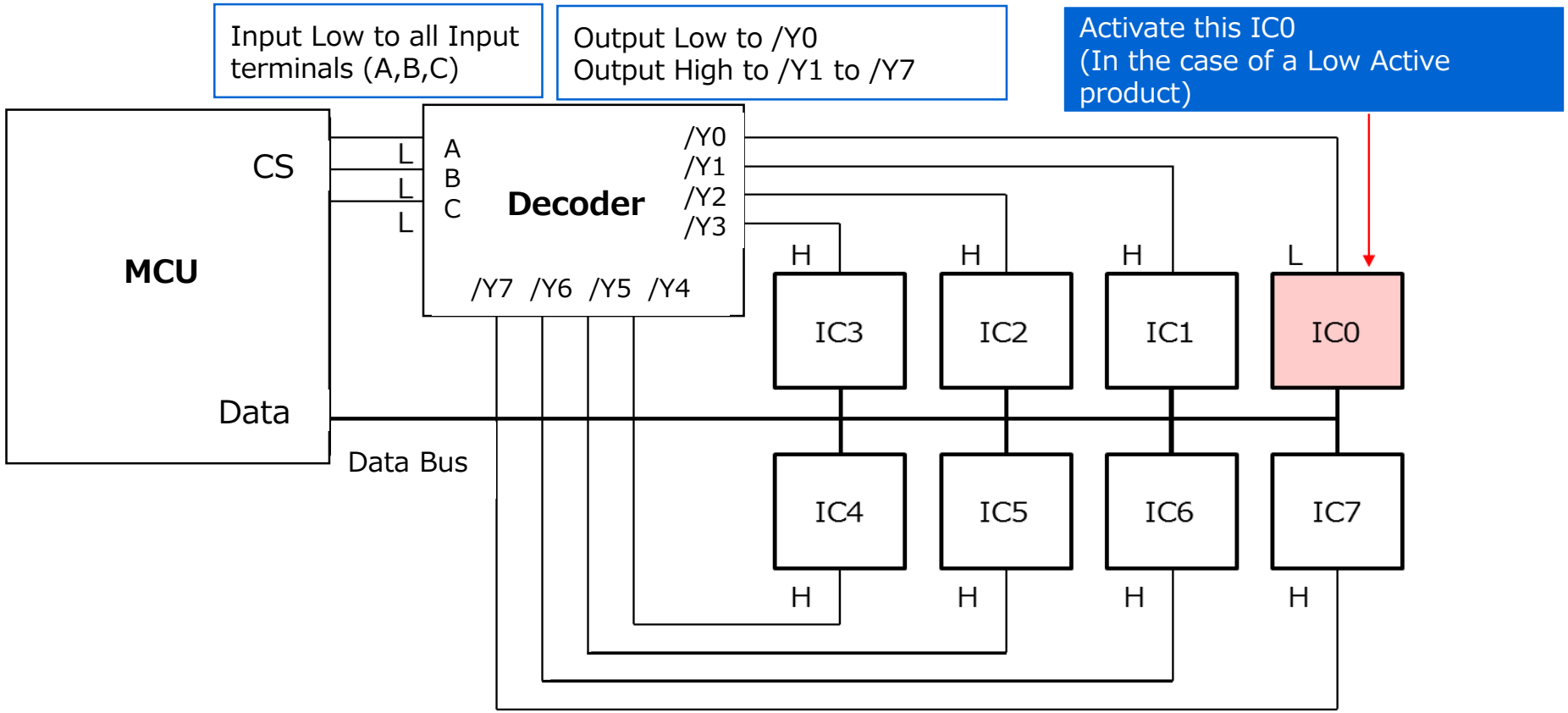
Logic symbol and truth table of a 3-to-8 decoder



Timing diagram of a 3-to-8 decoder

3.2.4 Combinational Logic: Decoders

The following shows how a 3-to-8 decoder is used to generate eight chip select signals from three inputs (A, B, and C). When A, B, and C are all Low, only the /Y0 output provides a logic Low, causing IC0 to be selected.
This figure indicates that, with a combination of three inputs, an arbitrary chip can be selected from up to eight chips.



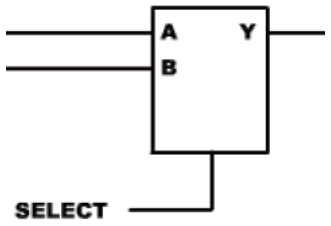
Generating chip select signals

3.2.5 Combinational Logic: Multiplexers

Multiplexers

Examples: 74VHC157, 74VHC153

A multiplexer selects one signal from multiple input signals and forwards it to a single output line.

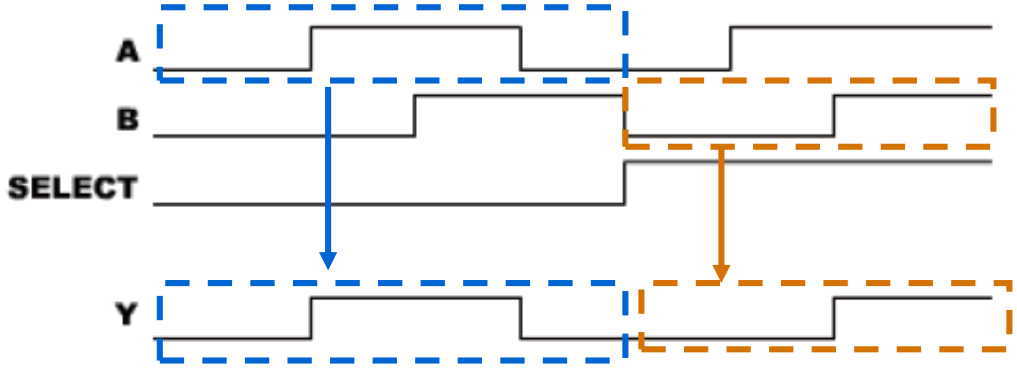


Input			Output
S	A	B	Y
L	L	X	L
L	H	X	H
H	X	L	L
H	X	H	H

Logic symbol and truth table of a 2-to-1 multiplexer

3.2.5 Combinational Logic: Multiplexers

The following timing diagram shows how one signal is selected from two inputs.
When the SELECT pin is Low (0), a signal is forwarded from the A pin to the Y pin.
When the SELECT pin is High (1), a signal is forwarded from the B pin to the Y pin.

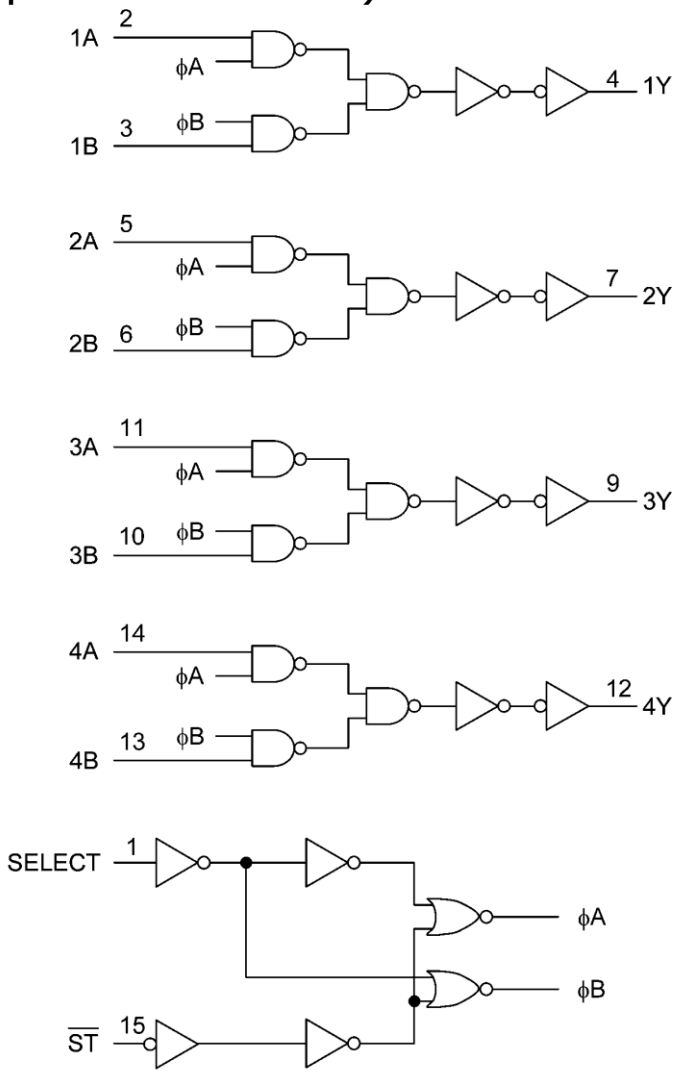


Timing diagram of a 2-to-1 multiplexer

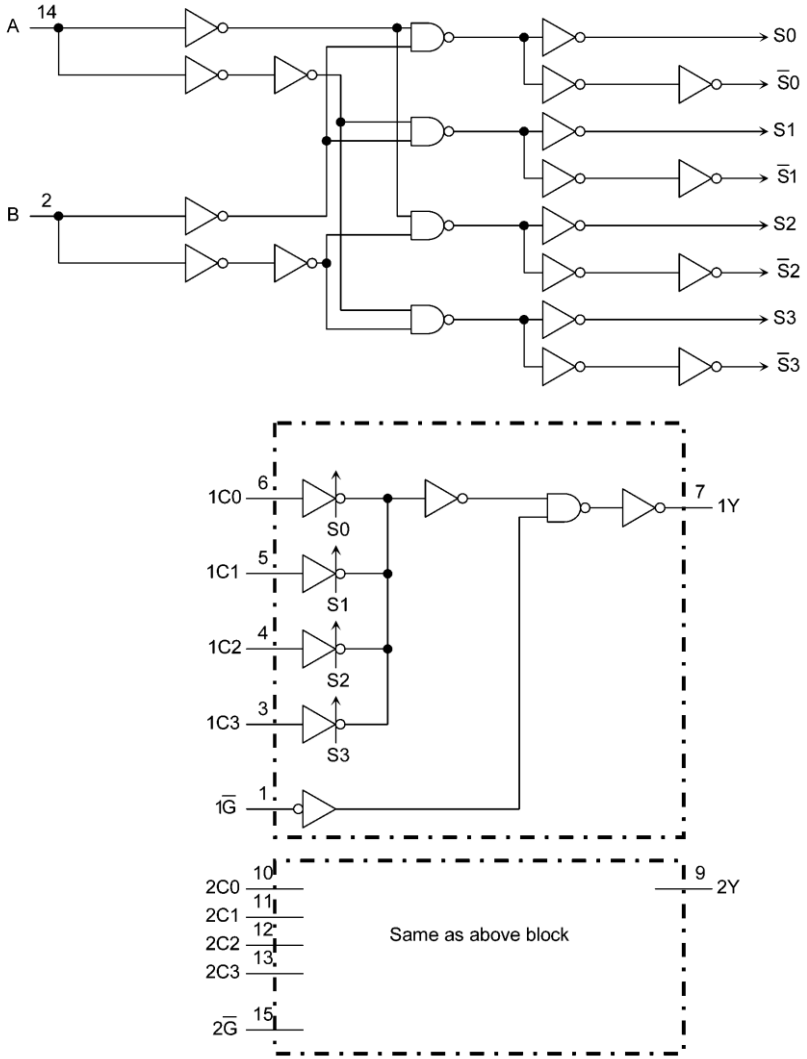
3.2.5 Combinational Logic: Multiplexers

Examples of multiplexers

Logic schematic of a 2-to-1 multiplexer (Example: 74VHC157)



Logic schematic of a 4-to-1 multiplexer (Example: 74VHC153)

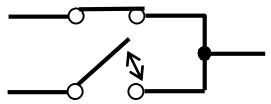


Analog multiplexer/demultiplexers

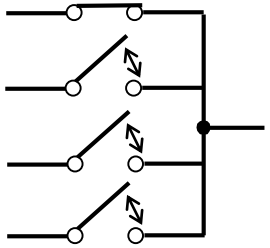
Example: 74VHC4051, 74VHC4052, 74VHC4053

An analog multiplexer incorporates analog switches (see the next page) to select one signal from multiple analog inputs and forward it to a single output line. The analog multiplexer can also be used as a demultiplexer since analog switches can transfer a signal bidirectionally.

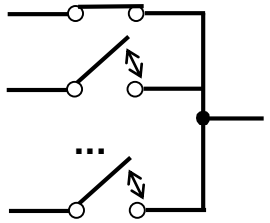
2-channel analog multiplexer/demultiplexer (xxxx4053)



4-channel analog multiplexer/demultiplexer (xxxx4052)



8-channel analog multiplexer/demultiplexer (xxxx4051)



Analog multiplexer/demultiplexers can be used to transfer both analog and digital signals.

3.2.7 Combinational Logic: Analog switches

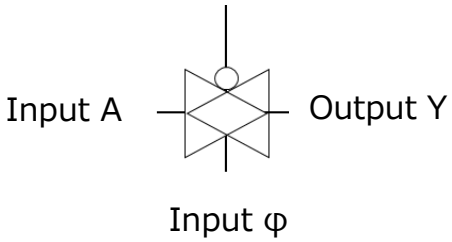
Analog switches

Example: 74VHC4066

An analog switch can conduct an analog signal such as a sine-wave signal in either direction. It passes a signal when on and blocks it when off.

An analog switch is composed of a pair of n-channel and p-channel MOSFETs connected in parallel in order to reduce on-resistance and improve the linearity of I/O characteristics.

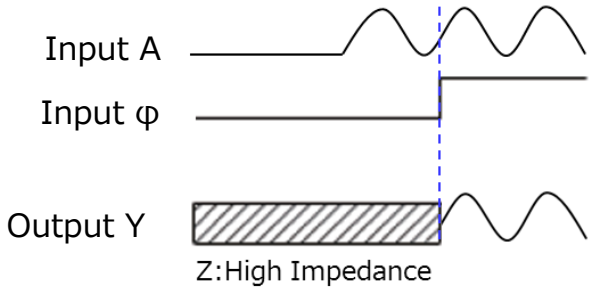
The datasheet for the analog switch shows typical sine-wave distortion, maximum frequency response, feed-through attenuation, crosstalk, and other specifications as analog switch characteristics. The following shows the logic symbol, truth table, timing diagram, and logic schematic of an analog switch.



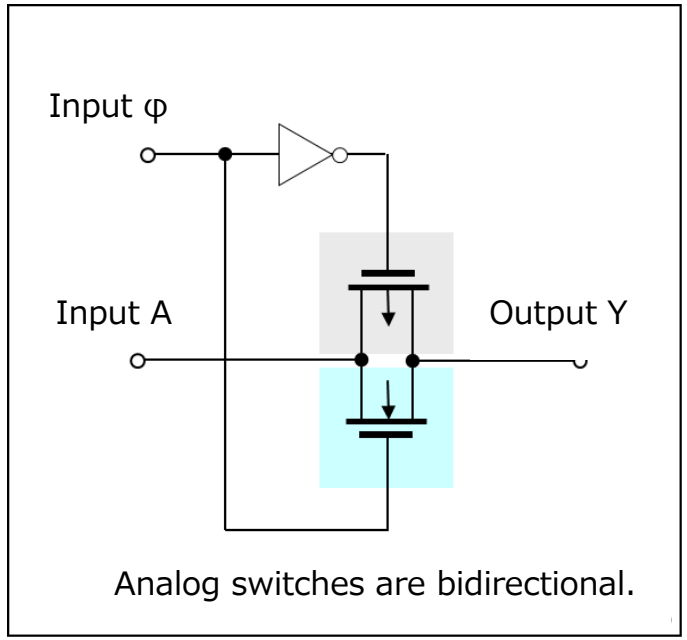
Input		Output
φ	A	Y
H	H	H
H	L	L
L	X	Z

X:Don't care
Z:High Impedance

Logic symbol and truth table of an analog switch



Timing diagram of an analog switch



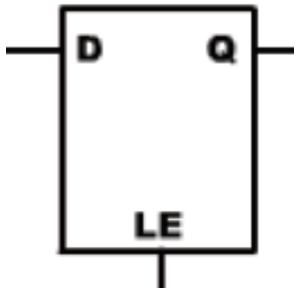
Logic schematic of an analog switch

3.3.1 Sequential Logic: Latches

Latches

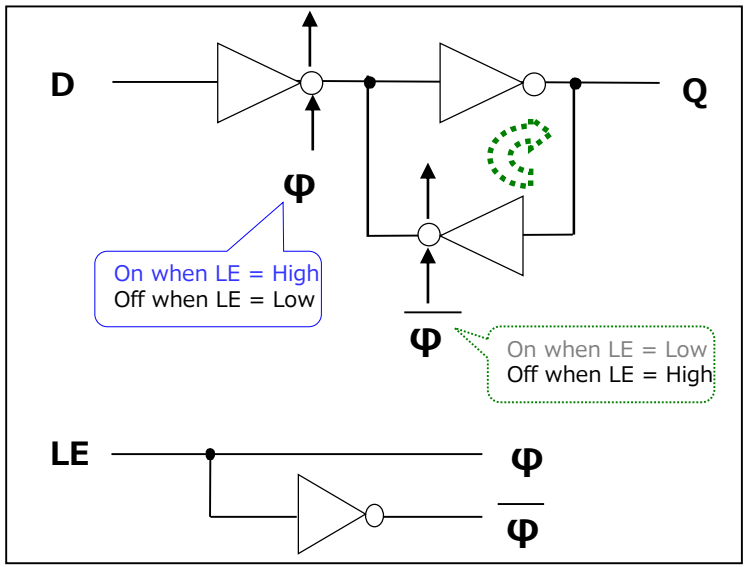
Example: 74VHC373

A latch can retain data under specific conditions.
There are several types of latches such as D-type and RS (Reset and Set) latches. As an example, the following describes the operation of a D-type latch.
For example, a D-type latch has an input data pin (D), a latch enable pin (LE), and an output data pin (Q). In this case, when LE is Low, Q retains the previous value of D. When LE is High, Q follows the changes of D. The following shows the timing diagram of a D-type latch.

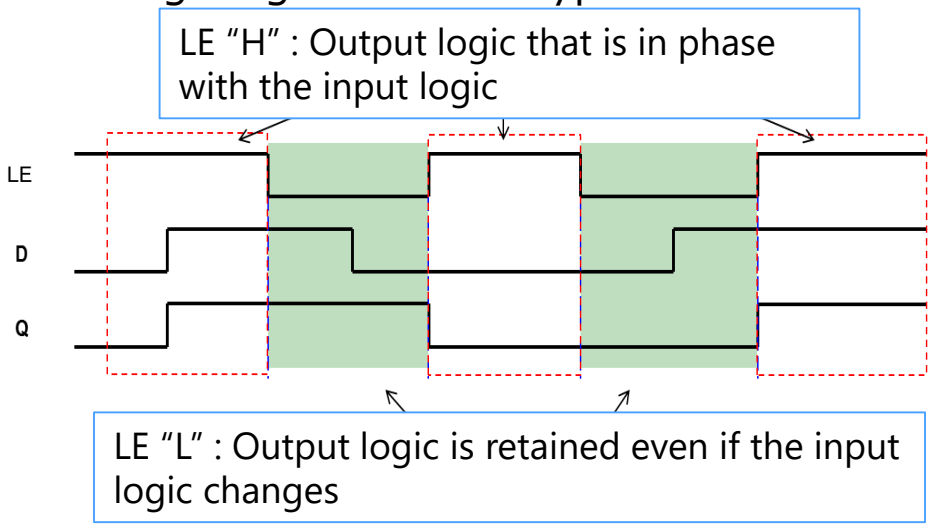


Input		Output
LE	D	Q
H	L	L
H	H	H
L	X	HOLD

Logic symbol and truth table of a D-type latch



Logic schematic of a D-type latch



Timing diagram of a D-type latch

3.3.2 Sequential Logic: Flip-flops

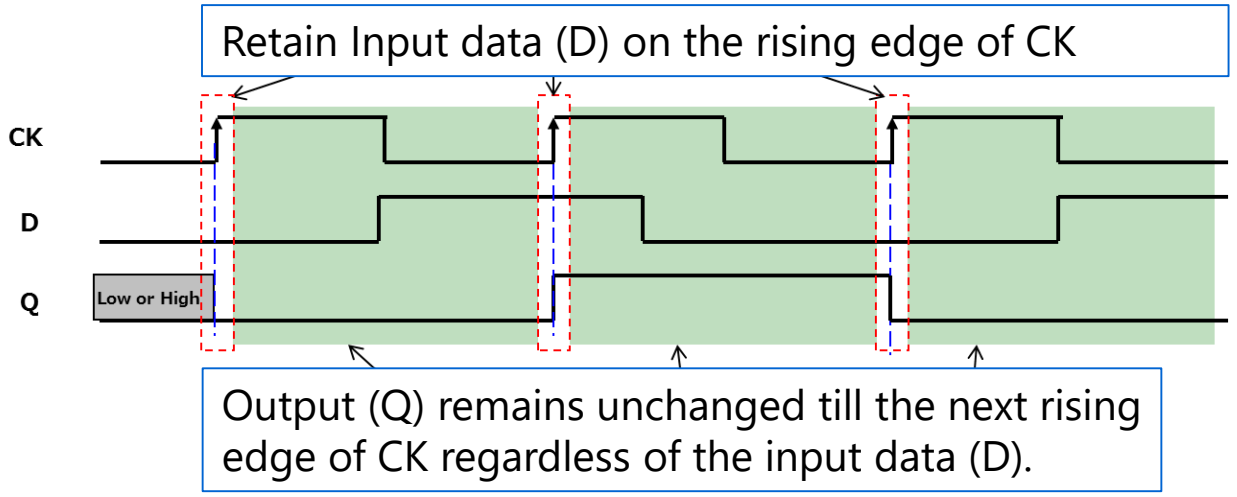
Flip-flops Example: 74VHC74

A flip-flop can retain data under specific conditions. The word “flip-flop” is sometimes abbreviated as FF. There are several types of flip-flops such as D-type and JK flip-flops. As an example, the following describes the operation of a D-type flip-flop.

A D-type flip-flop differs from a D-type latch in that the D-type flip-flop retains output data even after the clock is set inactive (Low in this example). (A D-type latch transfers data from the D input to the Q output while the LE input is High.)

For example, a D-type flip-flop has an input data pin (D), a clock pin (CK), and an output data pin (Q). This flip-flop latches input data (D) on the rising edge of CK and transfers them to Q. Q remains unchanged till the next rising edge of CK regardless of the input data (D). In other words, Q retains the input data (D) latched on the previous rising edge of CK. The following shows the timing diagram of a D-type flip-flop. Some flip-flops have a clear (CLR) or preset (PR) input pin that is used to initialize the internal state to a known value.

Flip-flops are used for synchronizers for asynchronous signals and delay circuits for digital signals as well as counters, frequency dividers, etc.

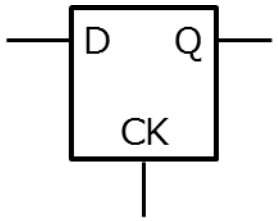


Timing diagram of a D-type flip-flop

3.3.2 Sequential Logic: Flip-flops

Operation of a D-type flip-flop

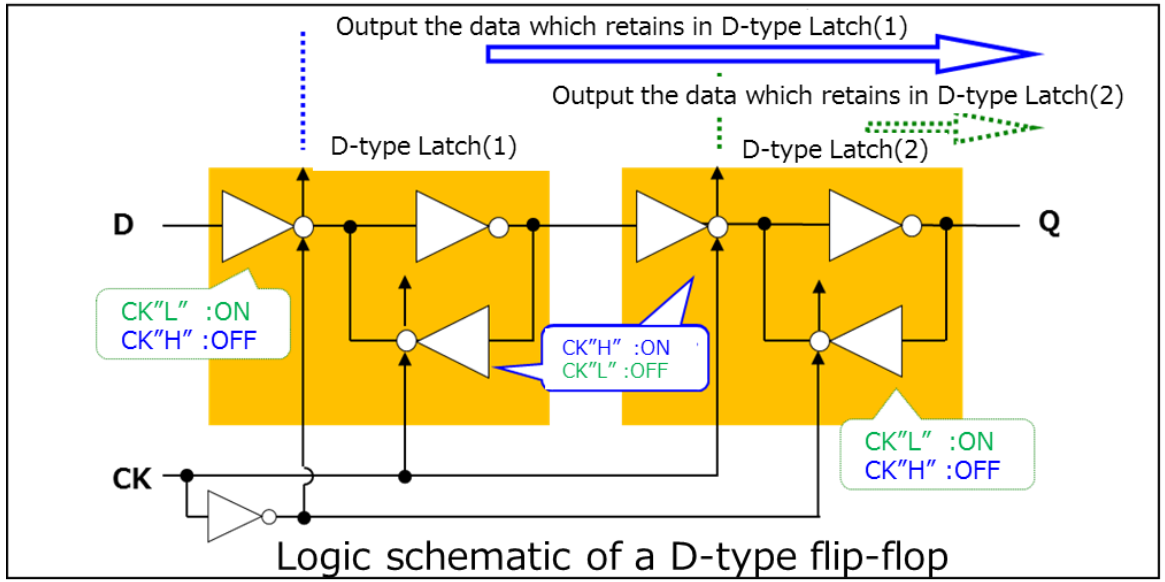
The following describes the operation of a D-type flip-flop using a logic schematic.
A D-type flip-flop consists of two D-type latches.
When a rising clock edge is applied to CK, D-type latch #1 is activated. While the clock (CK) is High, D-type latch #1 remains active, and thus the first clocked inverter in D-type Latch #2 is also active. Therefore, the data held in D-type latch #1 are transferred to the output (Q) as highlighted by the blue arrow. The output remains unchanged even if the input changes state.
When a falling clock edge is applied to CK, D-type latch #2 is activated.
As a result, the data held in D-type latch #2 continue to appear at the output (Q) as highlighted by the green arrow. Again, the output remains unchanged even if the input changes state.
It should be noted that the value of the output (Q) is unknown until a known input is latched on the rising edge of the clock (CK).



Input		Output
CK	D	Q
↑	L	L
↑	H	H
↓	X	Q

X:Don't care
Q:No change

Logic symbol and truth table of a D-type flip-flop



Logic schematic of a D-type flip-flop

3.3.3 Sequential Logic: Counters

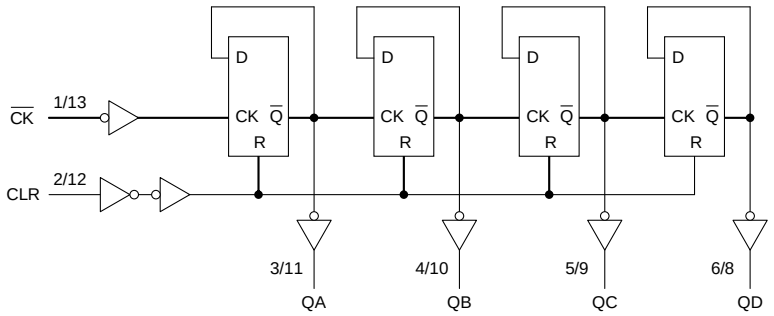
Counters

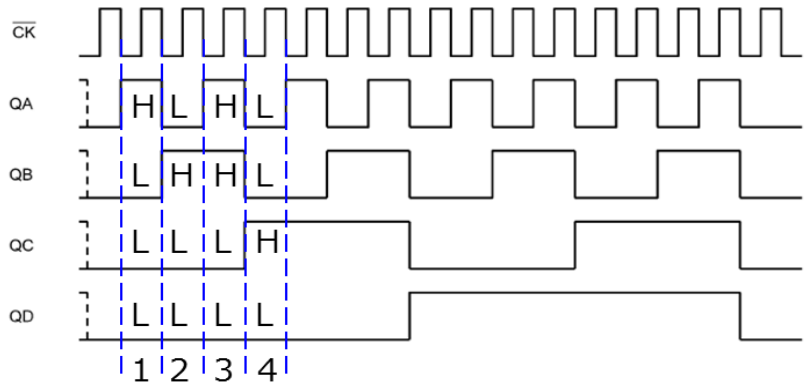
Examples: 74VHC393, 74VHC161

Counters count up or down sequentially on every clock (CK) pulse. A four-bit counter can have a modulus of up to 16; an eight-bit counter up to 256; and a 14-bit counter up to 16384. Some counters have the CLR input that is used to initialize the internal state to a known value.
Counters are incorporated in digital timers, electronic calculators, stopwatches, and many other devices.

Counters are broadly divided into asynchronous (ripple carry) and synchronous (parallel carry) counters. Let the propagation delay time of a single flip-flop be t_{pd}. Then, an n-stage asynchronous counter incurs a large delay equal to $n \times t_{pd}$. It should also be noted that the asynchronous counter could produce hazards when the counter output is fed to a logic gate.

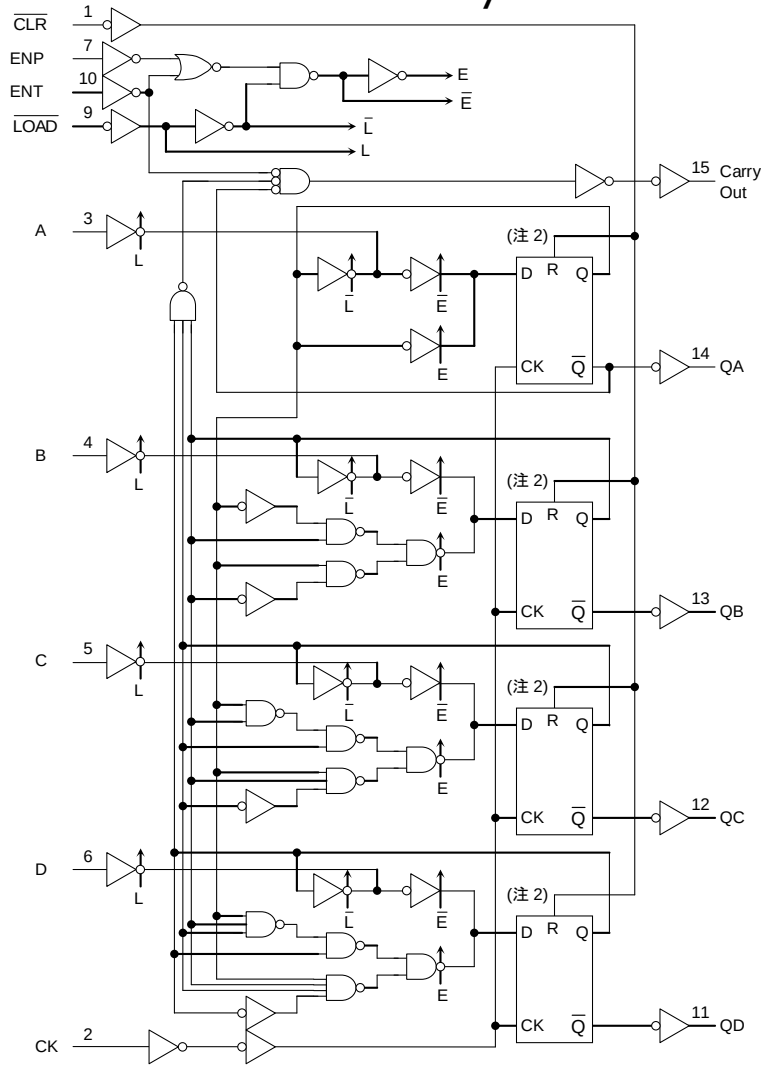
74VHC393 four-bit counter
Asynchronous: Ripple carry counter



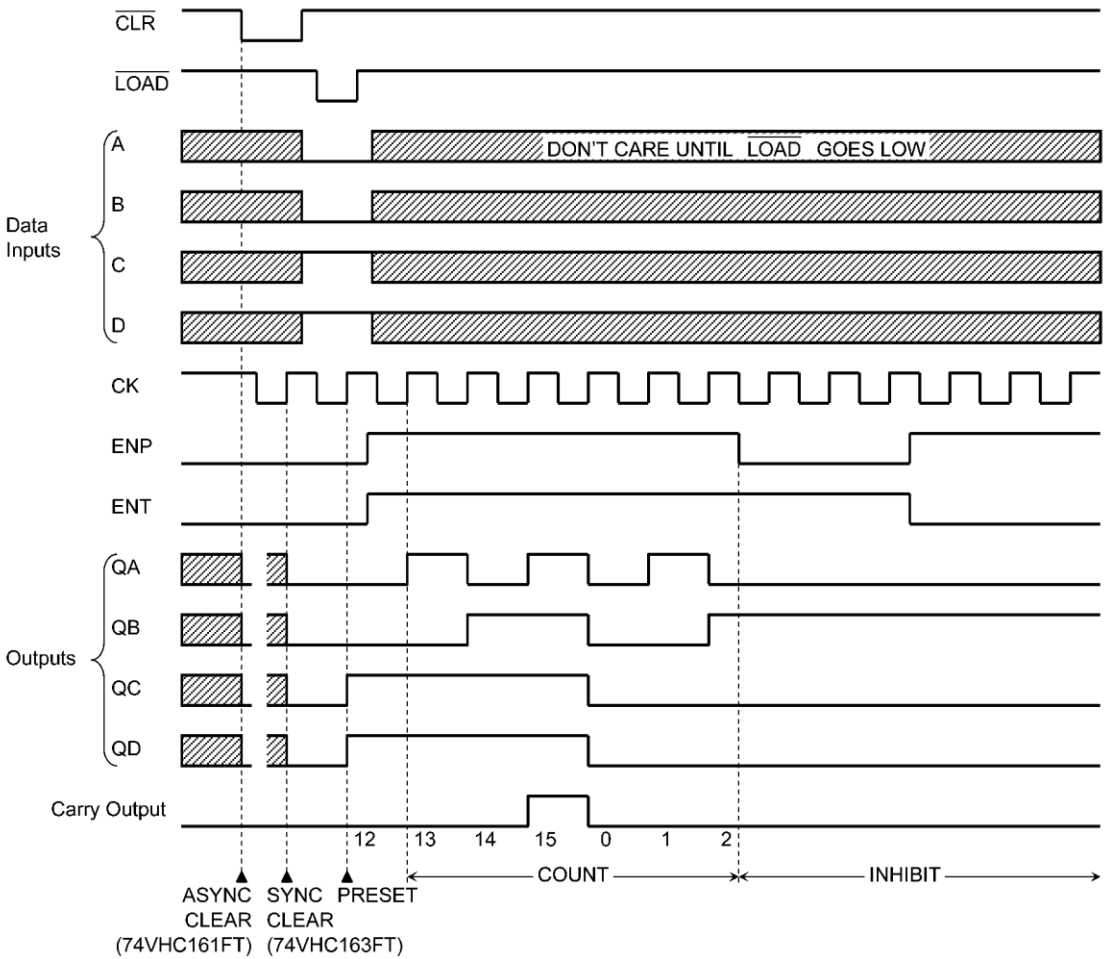


3.3.3 Sequential Logic: Counters

Logic schematic of the 74VHC161 four-bit counter
Synchronous: Parallel carry counter



Timing diagram: The following shows the timing diagram of a typical synchronous (parallel carry) counter, which counts upward sequentially on each edge of the clock (CK).

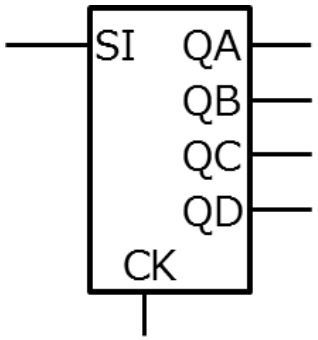


3.3.4 Sequential Logic: Shift registers

Shift registers

Examples: 74VHC164, 74VHC165

Shift registers can be configured for serial-parallel (SI-PO) or parallel-serial (PI-SO) conversion.
Parallel-serial conversion helps reduce the number of transmission lines (i.e., transmission bit width).
Some shift registers have the CLR input that is used to initialize the internal state to a known value.
The following shows the logic symbol and truth table of a shift register.
A shift register is composed of multiple flip-flops.



Input		Output			
CK	SI	QA	QB	QC	QD
↑	L	L	QAn	QBn	QCn
↑	H	H	QAn	QBn	QCn

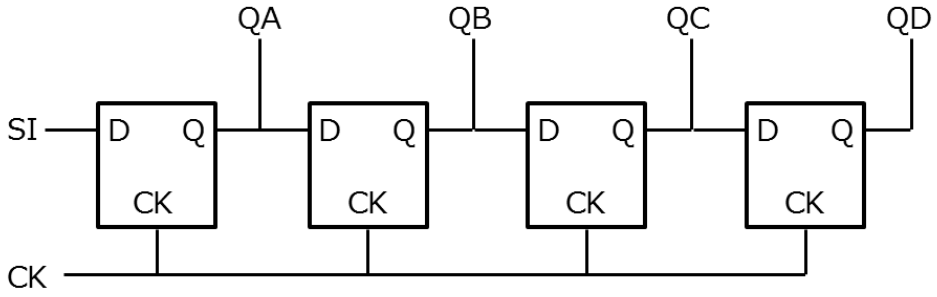
QAn to QCn: indicates the level of QA to QC output just before the rising edge of the clock.

Logic symbol and truth table of a shift register

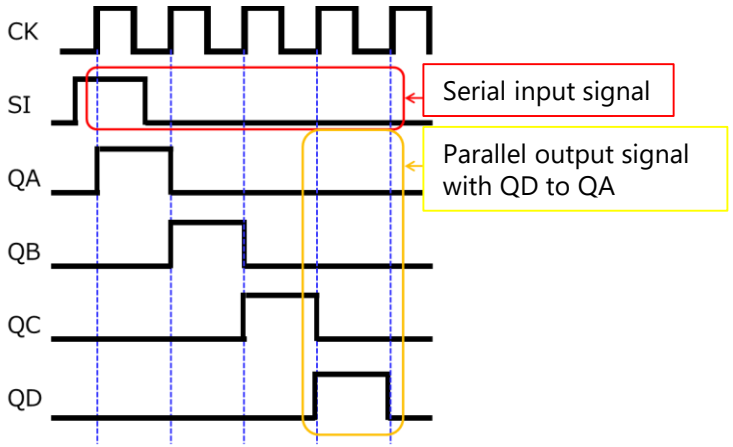
3.3.4 Sequential Logic: Shift registers

Operation of a shift register

The following describes the operation of a shift register using a logic schematic and a timing diagram. A shift register is composed of a cascade of flip-flops in which the output (Q) of each flip-flop is connected to the data (D) input of the next flip-flop in the chain.
A serial input (SI) is applied to the data (D) input of the first flip-flop. The data from SI are latched on the rising edge of the clock (CK) and appear at QA. With four clock pulses, the data from SI are transferred to the fourth flip-flop. As a result, the serial input (SI) is converted to parallel output data appearing at QD, QC, QB, and QA.



Logic schematic of a shift register

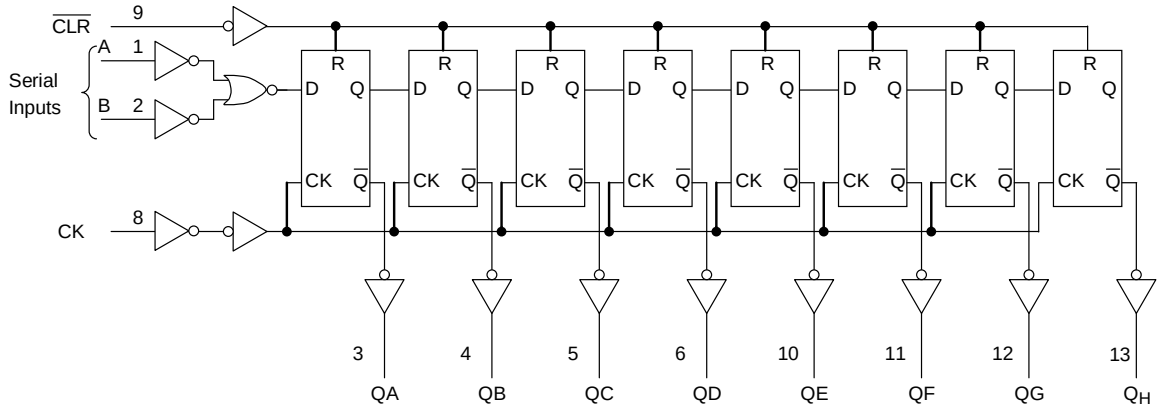


Timing diagram of a shift register
(Serial-in, parallel-out)

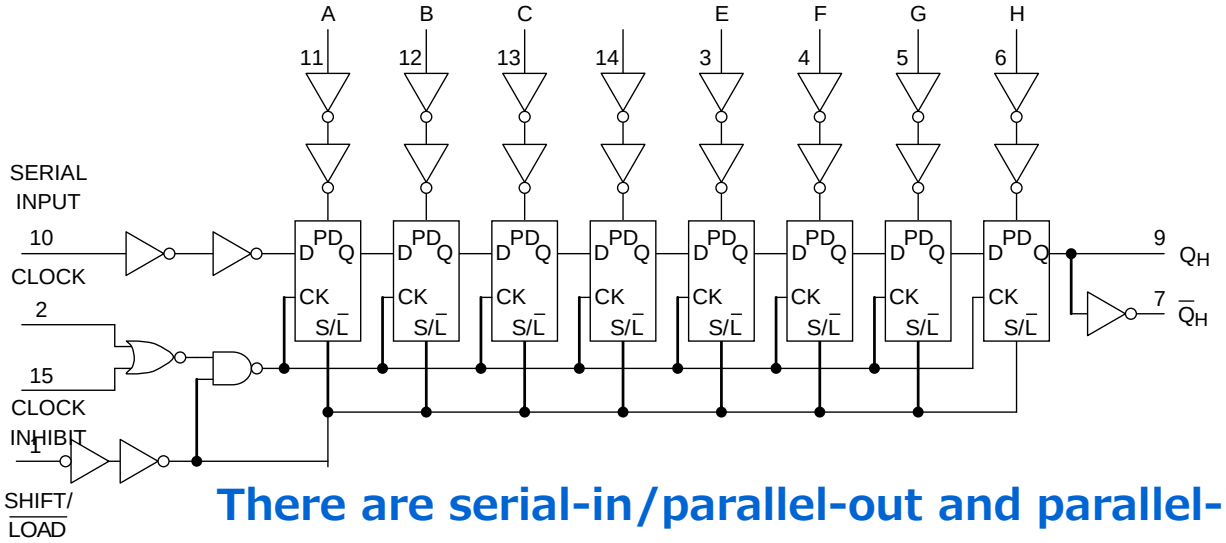
3.3.4 Sequential Logic: Shift registers

Types of shift registers

Logic schematic of the 74VHC164 eight-bit SI-PO shift register



Logic schematic of the 74VHC165 eight-bit PI-SO shift register



There are serial-in/parallel-out and parallel-in/serial-out shift registers.

TOSHIBA