

Latch (electronics)

From Wikipedia, the free encyclopedia

In electronics, a **latch** is a kind of **bistable multivibrator**, an electronic circuit which has two stable states and thereby can store one bit of information. Today the word is mainly used for simple *transparent* storage elements, while slightly more advanced *non-transparent* (or *clocked*) devices are described as **flip-flops**. Informally, as this distinction is quite new, the two words are sometimes used interchangeably.

A circuit incorporating latches has state; its output may depend not only on its current input, but also on its previous inputs. Such a circuit is described as sequential logic.

Contents

- 1 Simple set-reset latches
 - 1.1 SR NOR latch
 - 1.2 SR NAND latch
 - 1.3 JK latch
- 2 Gated latches and conditional transparency
 - 2.1 Gated SR latch
 - 2.2 Gated D-latch
 - 2.3 Gated T-Latch
- 3 Notes
- 4 References
- 5 See also
- 6 Further reading

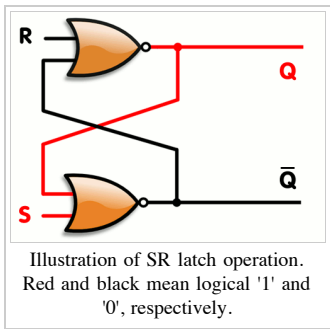
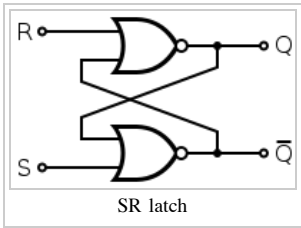
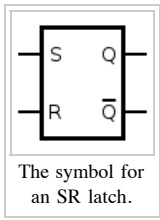
Simple set-reset latches

SR NOR latch

When using static gates as building blocks, the most fundamental latch is the simple *SR latch*, where S and R stand for *set* and *reset*. It can be constructed from a pair of cross-coupled NOR (Not OR) logic gates. The stored bit is present on the output marked Q.

Normally, in storage mode, the S and R inputs are both low, and feedback maintains the Q and $\bar{Q}$ outputs in a constant state, with $\bar{Q}$ the complement of Q. If S (*Set*) is pulsed high while R (*Reset*) is held low, then the Q output is forced high, and stays high when S returns to low; similarly, if R is pulsed high while S is held low, then the Q output is forced low, and stays low when R returns to low.

SR latch operation		
S	R	Action
0	0	Keep state
0	1	Q = 0
1	0	Q = 1
1	1	Restricted combination



The R = S = 1 combination is called a **restricted combination** or a **forbidden state** because, as both NOR gates then output zeros, it breaks the logical equation $Q = \text{not } \bar{Q}$. The combination is also inappropriate in circuits where *both* inputs may go low *simultaneously* (i.e. a transition from *restricted* to *keep*). The output would lock at either 1 or 0 depending on the propagation time relations between the gates (a race condition). In certain implementations, it could also lead to longer ringings (damped oscillations) before the output settles, and thereby result in undetermined values (errors) in high-frequency digital circuits. This condition is therefore usually avoided.

To overcome the restricted combination, one can add gates to the inputs that would convert (S , R) = (1 , 1) to one of the non-restricted combinations. That can be:

- Q = 1 (1,0) – referred to as an *S-latch*

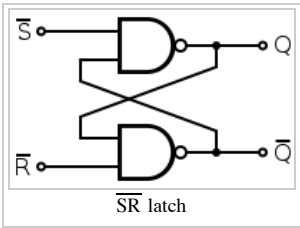
- $Q = 0$ (0,1) – referred to as an *R-latch*
- Keep state (0,0) – referred to as an *E-latch*

Alternatively, the restricted combination can be made to *toggle* the output. The result is the JK latch.

Characteristic: $Q+ = R'Q + R'S$ or $Q+ = R'Q + S^{[1]}$

SR NAND latch

This is an alternate model of the simple SR latch built with NAND (not AND) logic gates. *Set* and *reset* now become active low signals, denoted $\overline{S}$ and $\overline{R}$ respectively. Otherwise, operation is identical to that of the SR latch. Historically, SR-latches have been predominant despite the notational inconvenience of active-low inputs. This is because NAND gates are cheaper to produce than NOR gates in the diode-transistor logic (DTL) and transistor-transistor logic (TTL) families, which were the basis of early integrated circuits before the complementary metal–oxide semiconductor (CMOS) family attained widespread use. Since the 1970s and still as of 2010, most integrated circuits are built using CMOS technology, where NAND gates are also preferred.



SR latch operation		
$\overline{S}$	$\overline{R}$	Action
0	0	Restricted combination
0	1	$Q = 1$
1	0	$Q = 0$
1	1	Keep state

Symbol for an SR NAND latch

JK latch

The JK latch is much less used than the JK Flip-flop. The JK latch follows the following state table:

JK Latch truth table			
J	K	Q_{next}	Comment
0	0	Q_{prev}	No change
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q}_{prev}$	Toggle

Hence, the JK latch is identical to an SR latch that is made to *toggle* its output when passed the restricted combination.

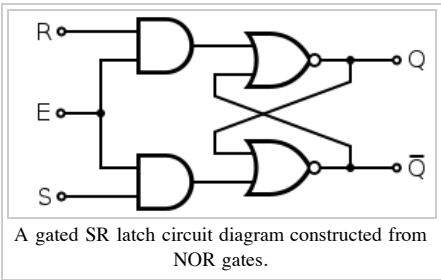
Gated latches and conditional transparency

Latches are designed to be *transparent*. That is, input signal changes cause immediate changes in output.^[nb 1] Alternatively, additional logic can be added to a simple transparent latch to make it *non-transparent* or *opaque* when another input (e.g., an "enable" input) is not asserted. By following a *transparent-high* latch with a *transparent-low* (or *opaque-high*) latch, a simple *edge-triggered* flip-flop can be implemented.^[citation needed]

Gated SR latch

A *synchronous SR latch* (sometimes *clocked SR flip-flop*) can be made by adding a second level of NAND gates to the inverted SR latch (or a second level of NOR gates to the direct SR latch). The extra gates further invert the inputs so the simple $\overline{S}\overline{R}$ latch becomes a gated SR latch (and a simple SR latch would transform into a gated $\overline{S}\overline{R}$ latch with inverted enable).

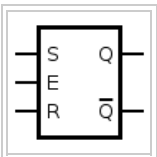
With E high (*enable* true), the signals can pass through the input gates to the encapsulated latch; all signal combinations except for (0,0) = *hold* then immediately reproduce on the (Q, $\overline{Q}$) output, i.e. the latch is *transparent*.



With E low (*enable* false) the latch is *closed (opaque)* and remains in the state it was left the last time E was high.

The *enable* input is sometimes a clock signal, but more often a read or write strobe.

E/C	Action
0	No action (keep state)
1	The same as non-clocked SR latch



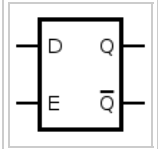
symbol for a gated SR latch

Gated D-latch

This latch is closely related to the gated SR latch and can be similarly constructed. It is also known as *transparent latch*, *data latch*, or simply *gated latch*. It has a *data* input and an *enable* signal (sometimes named *clock*, or *control*). The word *transparent* comes from the fact that, when the enable input is on, the signal propagates directly through the circuit, from the input D to the output Q.

Transparent latches are typically used as I/O ports or in asynchronous systems.^[nb 2] They are available as integrated circuits, usually with multiple latches per circuit. For example, 74HC75 is a quadruple transparent latch in the ubiquitous 7400 series.

E/C	D	Q	$\overline{Q}$	Comment
0	X	Q_{prev}	$\overline{Q}_{prev}$	No change
1	0	0	1	Reset
1	1	1	0	Set



Symbol for a gated D latch

The truth table shows that when the *enable/clock* input is 0, the D input has no effect on the output. When E/C is high, the output equals D.

Gated T-Latch

This is another synchronous SR latch that toggles the previous output. If the toggle (T) input is high, the T latch (well known as T flip-flop) changes state ("toggles") whenever the clock input is strobed. If the T input is low, it holds the previous value. Characteristic equation is; $Q_{next} = T \oplus Q_{pre}$, where Q_{next} is the next state and Q_{pre} is the previous state.

T	Q_{prev}	Q_{next}	Comment
0	0	0	Hold state
0	1	1	
1	0	1	Toggle state
1	1	0	

A T flip-flop can also be built using a JK flip-flop (i.e., with T implemented by connecting J & K pins together) or D flip-flop (T input and $Q_{previous}$ is connected to the D input through an XOR gate).

Notes

1. ^ Note that when several *transparent* latches follow each other, signals propagate through all of them.
2. ^ Transparent latches are also sometimes used in synchronous two-phase systems (for reduced transistor count); however, in single-phase synchronous systems with direct feedback, master-slave devices (often edge-triggered) must be used to avoid analog oscillations.

References

- [^] Foundations of digital logic design (http://books.google.com/books?id=4sX9FTGRo7QC&pg=PA344&lpg=PA344&dq=sr+characteristic+latch+equation&source=web&ots=9WdHN5uzTF&sig=ewGWuNX8g_4KozJuL5VyAS2yErc#PPA343,M1)

See also

- Flip-flop
- Logic gate

Further reading

- Hwang, Enoch (2006). *Digital Logic and Microprocessor Design with VHDL*. (<http://faculty.lasierra.edu/~ehwang/digitaldesign>) . Thomson. ISBN 0-534-46593-5. <http://faculty.lasierra.edu/~ehwang/digitaldesign>.
- Fundamentals of Digital Logic by Brown and Vranesic
- S.P.Vingron: "Switching Theory. Insight through Predicate Logic". Springer Verlag, 2003. ISBN 3-540-40343-4 – extensively covers the theory of latches
- Parallel Port Output expanding with Latches (<http://www.globu.net/pp/english/pp/>)
- CircuitDesign.info: You want latches? We got latches (<http://www.circuitdesign.info/blog/2008/12/you-want-latches-we-got-latches-flip-flop-design/>) CMOS D flip-flop design
- V. O. Vasyukevich. "Analytics of trigger functions". *Automatic Control and Computer Sciences* (<http://www.springerlink.com/content/3h040562h73855w5/>) . 2009, Vol.43, No.4, 184–189.

Retrieved from "http://en.wikipedia.org/wiki/Latch_(electronics)"

Categories: Digital circuits | Digital registers

- This page was last modified on 27 March 2010 at 23:07.
 - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.
- Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.