

Flip-Flops and Latches

Table of contents

The D flip-flop	1
Latches vs flip-flops	3
The SR latch (cross-coupled NOR gates)	3
Homework: latches from NAND gates	4
A real D flip-flop	5
Symbols used	5
D flip-flop chips	6

Flip-flops and latches are the **storage elements** — the circuits that let a design remember a value even after the input that produced it has changed. They’re what turns a combinational circuit into a [\[\[sequential-circuits|sequential circuit\]\]](#).

The D flip-flop

The D flip-flop is the **only** flip-flop type covered in this course. Its symbol is a rectangle with:

- **D** — the data input, on the left.
- **Q** — the output, on the right.
- **CLK** — the clock, a *control* input on the left, drawn with a small triangle where it meets the box.

How it works: **Q copies the value of D (and remembers it) whenever CLK goes from 0 to 1** — the *rising edge*. In between clock edges the flip-flop is not responsive, so it holds the stored value. Q changes **only** on that 0→1 transition.

Summary: a D flip-flop remembers a single bit — either a “1” or a “0”. It keeps that value until the next clock edge, at which point the D input is transferred to output Q.

- To remember n bits you need n D flip-flops.
- An **n -bit register** is (by definition) n D flip-flops. See [shift-registers¹](#).

¹[courses/csse2010/shift-registers.pdf](#)

- Flip-flops can themselves be made out of logic gates.

Characteristic table

A **characteristic table** defines the operation of a flip-flop in tabular form. The left column is the input; the right column, $Q(t+1)$, is what the output will be **on the next clock edge** (i.e. when CLK goes 0→1).

D	$Q(t+1)$
0	0 Reset
1	1 Set

That is the whole behaviour: the flip-flop is *reset* (to 0) or *set* (to 1) according to D, at the clock edge.

Worked waveform

The lecture's example: D is 0, 0, then 1, 1 across a clock with two rising edges; CLK starts low, pulses high, goes low, then pulses high again after D has risen.

Clock edge	D at that edge	Q after
(before the first edge)	—	unknown/undefined
1st rising edge	0	0
2nd rising edge	1	1

Before the first rising edge Q is marked with an “X” on the slide — its value is undefined, because nothing has been clocked in yet. Between the two rising edges Q stays 0 even though D has already risen to 1: the flip-flop is only sensitive at the edge.

Other flip-flop types

Other types exist — **JK flip-flops** and **T flip-flops** — but they are **not covered in this course**. Only the D flip-flop is used.

Latches vs flip-flops

- **Latches** are **level triggered** devices — they latch the output and respond to changes of logic *levels* on the inputs.
- A latch circuit can be modified so that it becomes sensitive to an **edge** (a momentary transition) of a control input, i.e. a clock signal. Such circuits are called **flip-flops**.
- A flip-flop can store 1 bit of information while being sensitive to a clock edge — it changes its output only *at* the clock edges, based on the inputs.
- So: **latches are level triggered; flip-flops are edge triggered.**

A clock signal has two edges:

Edge	Transition
Positive (rising) edge	$0 \rightarrow 1$
Negative (falling) edge	$1 \rightarrow 0$

A D flip-flop can therefore be **positive edge triggered** or **negative edge triggered**. In a positive edge triggered D flip-flop, D is copied to Q at the positive edge of the clock; in between clock edges the flip-flop is not responsive, thus stores the value.

The **state** of a flip-flop is the value it is storing. Flip-flops are more useful than latches in practice.

The SR latch (cross-coupled NOR gates)

Structure: two 2-input NOR gates cross-coupled.

- The **top** NOR gate takes S as one input and the *bottom* gate's output as its other input; its output is $\bar{Q}$.
- The **bottom** NOR gate takes R as one input and the *top* gate's output ($\bar{Q}$) as its other input; its output is Q .

So each gate's output feeds back into the other gate's input — that feedback loop is what stores the bit. Recall (from logic-gates²) that a NOR gate outputs 1 only when **both** inputs are 0; any 1 on an input forces the output to 0.

The truth table on the slide is **left blank** (“to be completed in class”). Derived from the NOR behaviour and the cross-coupling, assuming a stable starting state Q :

²courses/csse2010/logic-gates.pdf

S	R	Q	$\bar{Q}$	Meaning
0	0	Q (unchanged)	$\bar{Q}$ (unchanged)	Hold — remembers the previous value
0	1	0	1	Reset — Q forced to 0
1	0	1	0	Set — Q forced to 1
1	1	0	0	Invalid — see below

Reasoning for each row:

- $S = 0, R = 0$: neither gate is forced. Say $Q = 1$; then the top gate sees $S = 0$ and $Q = 1$, so $\bar{Q} = 0$; the bottom gate sees $R = 0$ and $\bar{Q} = 0$, so $Q = 1$ — consistent. The same argument works with $Q = 0$. Both states are self-consistent, so the latch simply holds whatever it already had.
- $S = 1, R = 0$: the top gate has a 1 on an input, so $\bar{Q} = 0$. The bottom gate then sees $R = 0$ and $\bar{Q} = 0$, so $Q = 1$. S **sets** the latch.
- $S = 0, R = 1$: mirror image — the bottom gate has a 1 on an input, so $Q = 0$; the top gate then sees $S = 0$ and $Q = 0$, so $\bar{Q} = 1$. R **resets** the latch.
- $S = 1, R = 1$: both gates have a 1 on an input, so both outputs are 0 — $Q = \bar{Q} = 0$. This breaks the whole point of the two outputs being complements, which is why the combination is called forbidden/invalid. What happens *after* both inputs return to 0 simultaneously is not uniquely determined (the two gates race), so the resulting state is **indeterminate**. The slide is blank here and the worked answer was only done live — check the lecture recording for the exact form the lecturer wrote in this row.

Homework: latches from NAND gates

Slide 11 is otherwise blank and sets a homework: *analyse the S-R latch circuit from the previous slide when the NOR gates are replaced with NAND gates, and complete the truth table.*

The slide gives no answer. Derived, keeping the same wiring and the same input labels (top gate: S and the bottom output; bottom gate: R and the top output), and recalling that a NAND outputs 0 only when **both** inputs are 1 — so any **0** on an input forces the output to **1**:

S	R	top output	bottom output	Meaning
1	1	unchanged	unchanged	Hold
0	1	1	0	bottom output forced to 0

S	R	top output	bottom output	Meaning
1	0	0	1	bottom output forced to 1
0	0	1	1	Invalid — outputs no longer complementary

The headline result: with NAND gates the inputs become **active low** (the latch holds at $S = R = 1$ rather than $S = R = 0$, and the invalid combination moves to $S = R = 0$), and with the labels left unchanged the set/reset roles swap over relative to the NOR version. This is why NAND latches are conventionally drawn with the inputs labelled $\bar{S}$ and $\bar{R}$. Worth confirming against the lecturer's own labelling.

A real D flip-flop

A real D flip-flop is built from cross-coupled NAND gates — the lecture's schematic uses six NAND gates. As well as D, CLK and the outputs Q and $\bar{Q}$, it has two extra inputs:

- $\overline{\text{PRE}}$ (**Preset**) — forces Q to 1.
- $\overline{\text{CLR}}$ (**Clear**) — forces Q to 0.

Both are drawn with overbars, i.e. they are **active low**, and both are **asynchronous**: they act on the latch directly, independent of the clock, rather than waiting for a clock edge. In the schematic they feed into the NAND gates of the input latches directly, bypassing the D/CLK path.

Symbols used

Four related symbols, all drawn as a rectangle with **D** in at the top left, **Q** out at the top right, and a clock input **CK** at the bottom left. What distinguishes them is the marking on that CK input:

Symbol	CK input marking	Device	Triggered on
(a)	plain input	Latch	High level of CK
(b)	bubble (inversion circle)	Latch	Low level of CK
(c)	triangle	Flip-flop	Rising (positive) edge of the clock
(d)	bubble + triangle	Flip-flop	Falling (negative) edge of the clock

The two markings read independently and are the whole key to the notation:

- The **triangle** indicates **edge-triggered**, and therefore that the device is a flip-flop rather than a latch. No triangle means level-triggered, i.e. a latch.
- The **bubble** indicates inversion, i.e. the *falling* edge (or the low level) rather than the rising edge (or the high level).

The slide only annotates (c) and (d) explicitly; the readings given above for (a) and (b) are derived from those two rules.

D flip-flop chips

- **74HCT74** — dual D flip-flop (two independent D flip-flops in a 14-pin package), each with CLR and PR (preset) inputs and $Q/\bar{Q}$ outputs. Pinout in device-pinouts³.
- **74HCT273** — eight D flip-flops in a 20-pin package, so it can hold **one byte** (8 bits) of information.

The lecture points to the “device symbols PDF on Blackboard” for the full symbol set.

³courses/csse2010/device-pinouts.pdf