Timing Diagrams

Table of contents

Timing diagram as a logic representation 1
Convention for drawing the input waveforms 1
Real gates aren't perfect: the reality of timing 3

A **timing diagram** is the fourth of the four equivalent ways to describe a logic function, alongside the **logic diagram**, the **truth table**, and the **Boolean expression** (see logic-gates[1]).

Timing diagram as a logic representation

It is like a truth table, but in **graphical format**: time runs left to right along the horizontal axis, and each signal gets its own horizontal track that steps between the logic 0 and logic 1 levels (drawn as two dashed reference lines).

Worked example — an inverter. The input waveform starts at logic 0, rises to logic 1 for a period, then falls back to logic 0. The output waveform is the mirror image: it starts at logic 1, falls to logic 0 for exactly that period, then returns to logic 1. Each **change in the input causes the output to change**.

At this idealised level the output changes at exactly the same instant as the input; the transitions are drawn as vertical lines. That is a convenient fiction, corrected at the bottom of this note.

Convention for drawing the input waveforms

The input waveforms must be drawn so that **all possible input combinations are covered** — that is what makes the diagram equivalent to the full truth table.

The standard construction for a 2-input gate divides the time axis into four equal intervals and drives the inputs like a binary count:

[1] courses/csse2010/logic-gates.pdf

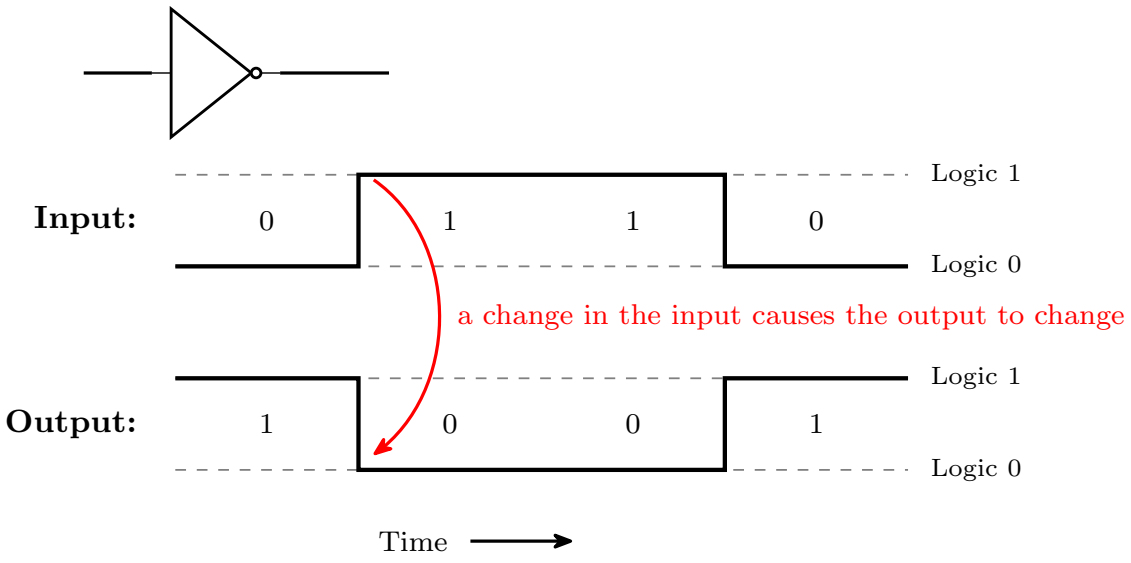


Figure 1: An inverter drawn as a timing diagram. Reading the four intervals off the input gives 0, 1, 1, 0 and off the output gives 1, 0, 0, 1 — which is the inverter's truth table, just laid out along a time axis instead of down a page. The two dashed lines per track are the logic 0 and logic 1 reference levels.

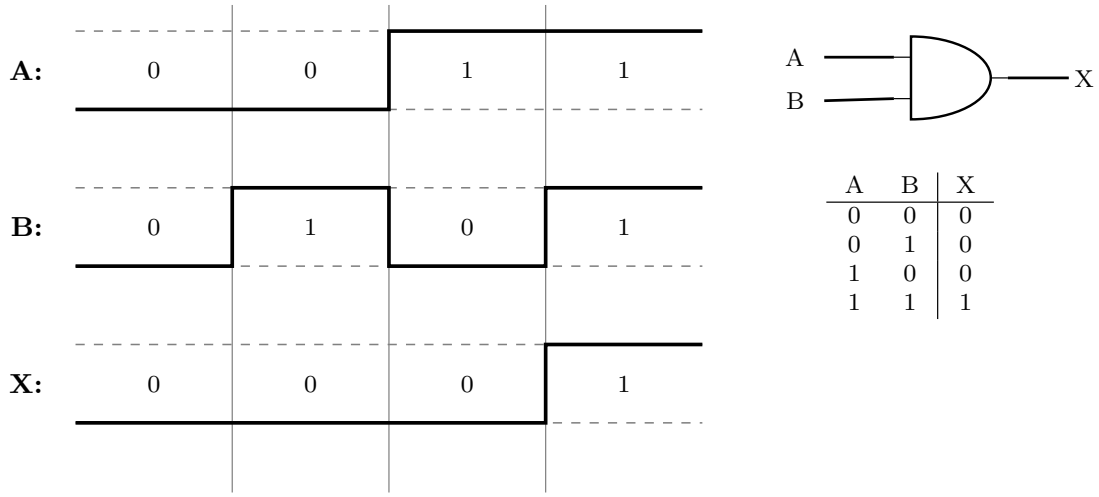
- A is low for the first half of the diagram, then high for the second half.
- B toggles at twice that rate: low, high, low, high.

Reading the four intervals left to right therefore gives $AB = 00, 01, 10, 11$ — every combination, once each, in truth-table order. For n inputs the same scheme generalises: input i toggles at half the rate of input $i + 1$, and the diagram is 2^n intervals wide.

To read a gate's output off the diagram, work interval by interval: identify the AB combination in that interval, look up the gate's truth table row, and draw the output at that level for the width of the interval.

Going the other way — being *given* the output track and asked which gate it is — is the standard exam form of this. Read the output value in each interval, write it beside the corresponding AB row, and compare the resulting column with the gates you know. A worked instance is in 2026-08-06-combinational-logic[2].

[2] courses/csse2010/2026-08-06-combinational-logic.pdf



the four intervals are $AB = 00$, 01, 10, 11 — every combination, once each

Figure 2: The convention on a 2-input example, with the AND output filled in. The vertical rules divide the four intervals; the digits under each track are the truth-table row for that interval. Because the inputs count 00, 01, 10, 11, the output track is literally the output column of the truth table, drawn left to right.

Real gates aren't perfect: the reality of timing

The idealised diagrams above show instantaneous, perfectly square transitions. Real gates do not behave that way — real waveforms have sloped edges, and the output changes some time *after* the input does. Three quantities describe this:

Term	Definition
Propagation delay	Time for a change in the input to affect the output.
Fall time	Time taken for the output to fall from 1 to 0.
Rise time	Time for the output to rise from 0 to 1.

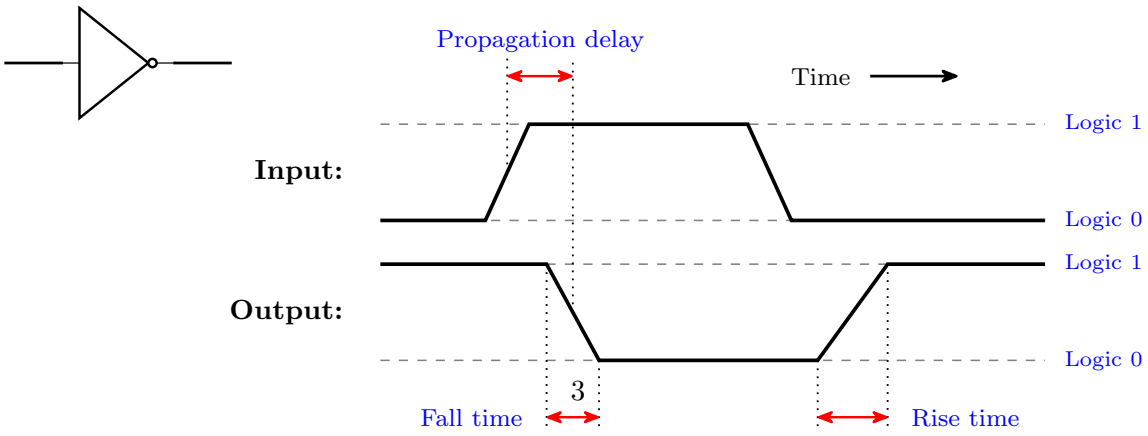


Figure 3: The same inverter drawn honestly. The input's edges slope instead of jumping; the output does not begin to move until the propagation delay has elapsed; and once moving, its downward edge takes the fall time and its later upward edge takes the rise time. Rise and fall times are properties of one edge of one signal — the propagation delay is the gap between two signals.