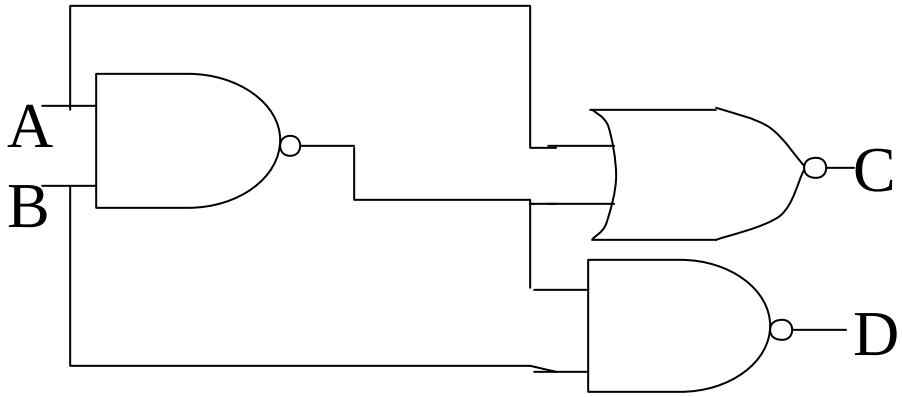
  

# PHYSICAL LIMITATIONS OF LOGIC GATES

## From Lecture 16:

- Computer Datapath: Boolean algebraic functions using binary variables
- Symbolic representation of functions using logic gates
- Example:




## Today:

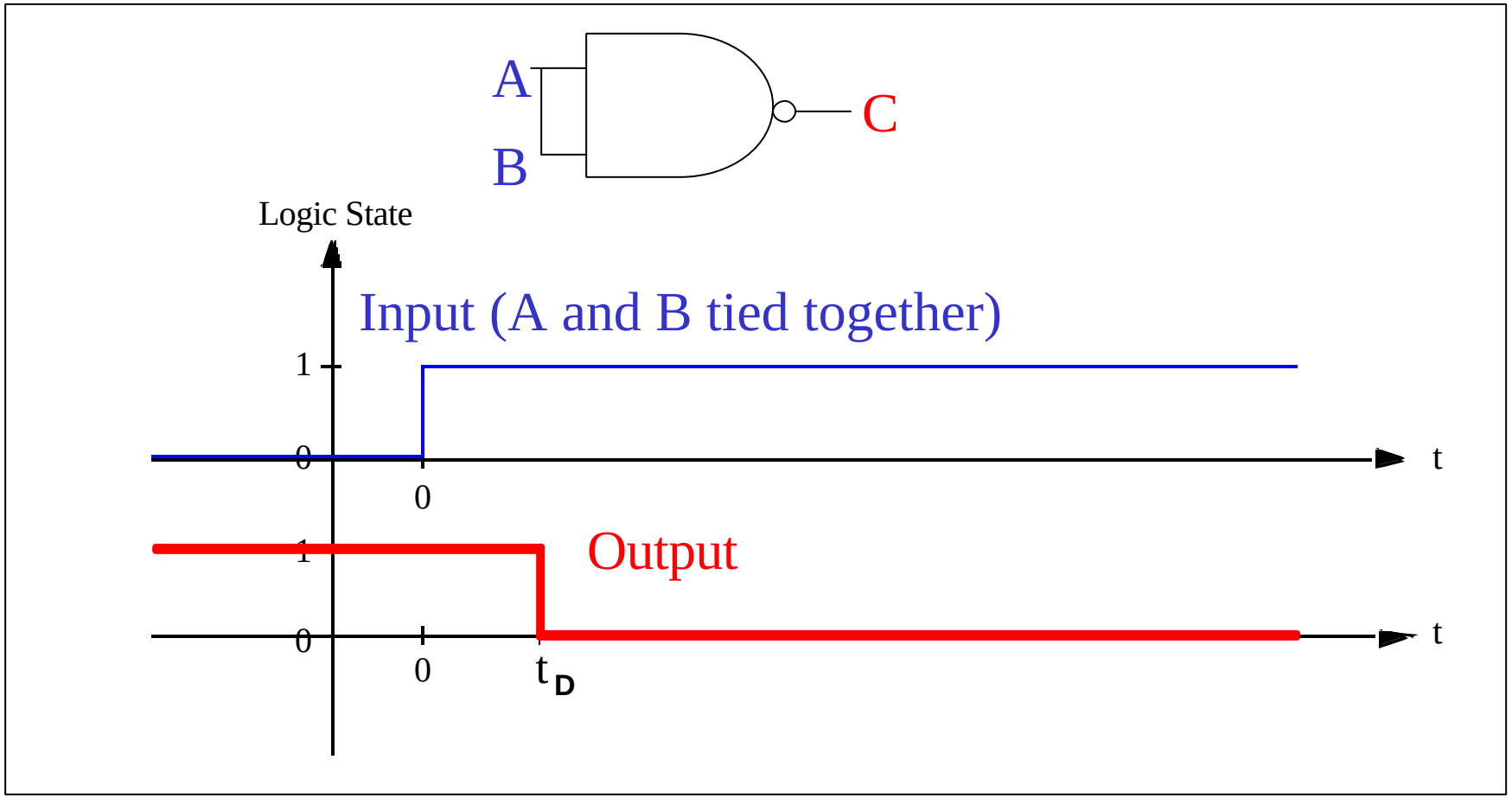
- Every node has capacitance and interconnects have resistance. It takes time to charge these capacitances.
- Thus, output of all circuits, including logic gates is **delayed** from input.
- For example we will define the *unit gate delay*

# UNIT GATE DELAY $t_D$

Time delay $t_D$ occurs between input and output: "computation" is not instantaneous

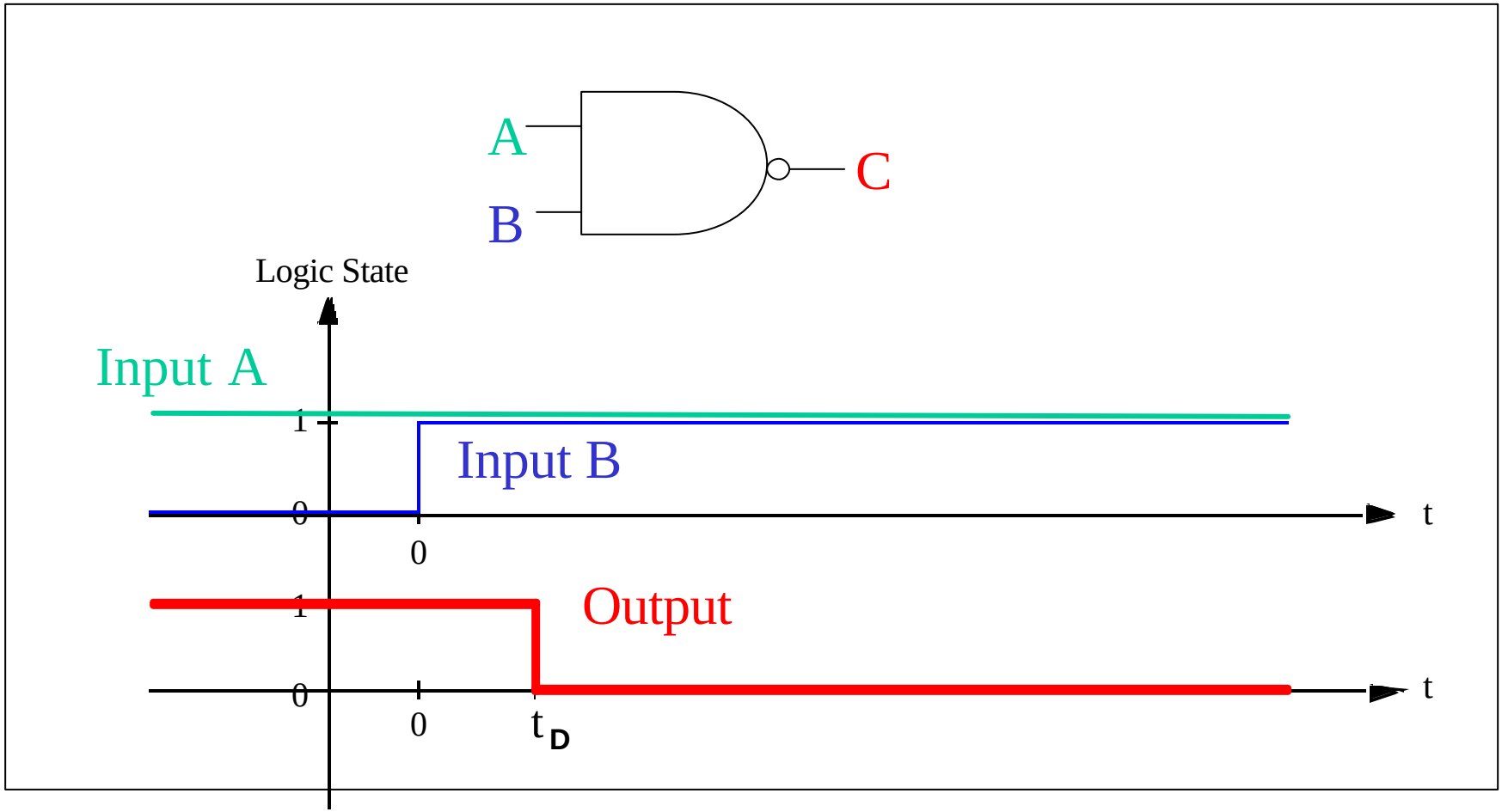
Value of input at $t = 0^+$ determines value of output at later time $t = t_D$



# UNIT GATE DELAY $t_D$ in ASYNCHRONOUS LOGIC

Time delay $t_D$ is measured from the last input change

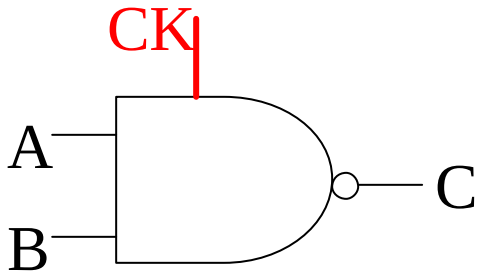


# Synchronous and Asynchronous Logic

Time delay occurs between input and output in real logic circuits.

Therefore the time at which output appears is difficult to predict… it depends for example on how many gates you go through.




We will often not distinguish asynchronous vs synchronous logic.

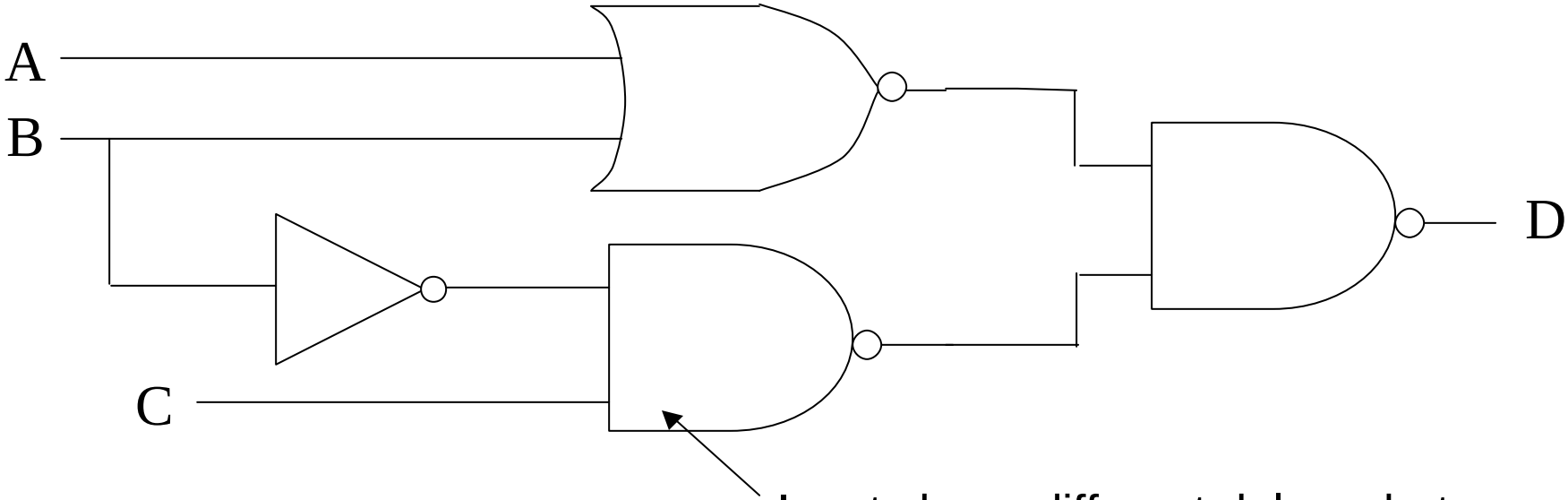
To make logic operations as fast as possible, we need predictability of signal availability. That is we want to know exactly when "C" is correctly computed from A and B. This requirement argues for *synchronous logic*, in which a *clock signal* CK actually initiates the computation of C.

Thus in the modified gate, C will is valid precisely one gate delay ($\tau_D$) after the clock input CK, goes high (A and B are evaluated precisely when K goes high, what they do before or after this is irrelevant; K must go low, then high again before the NAND gate again looks at A and B).

# EFFECT OF GATE DELAY

Cascade of Logic Gates




Inputs have different delays, but we ascribe a single worst-case delay $\tau_D$ to every gate

How many "gate delays for shortest path? ANSWER : 2

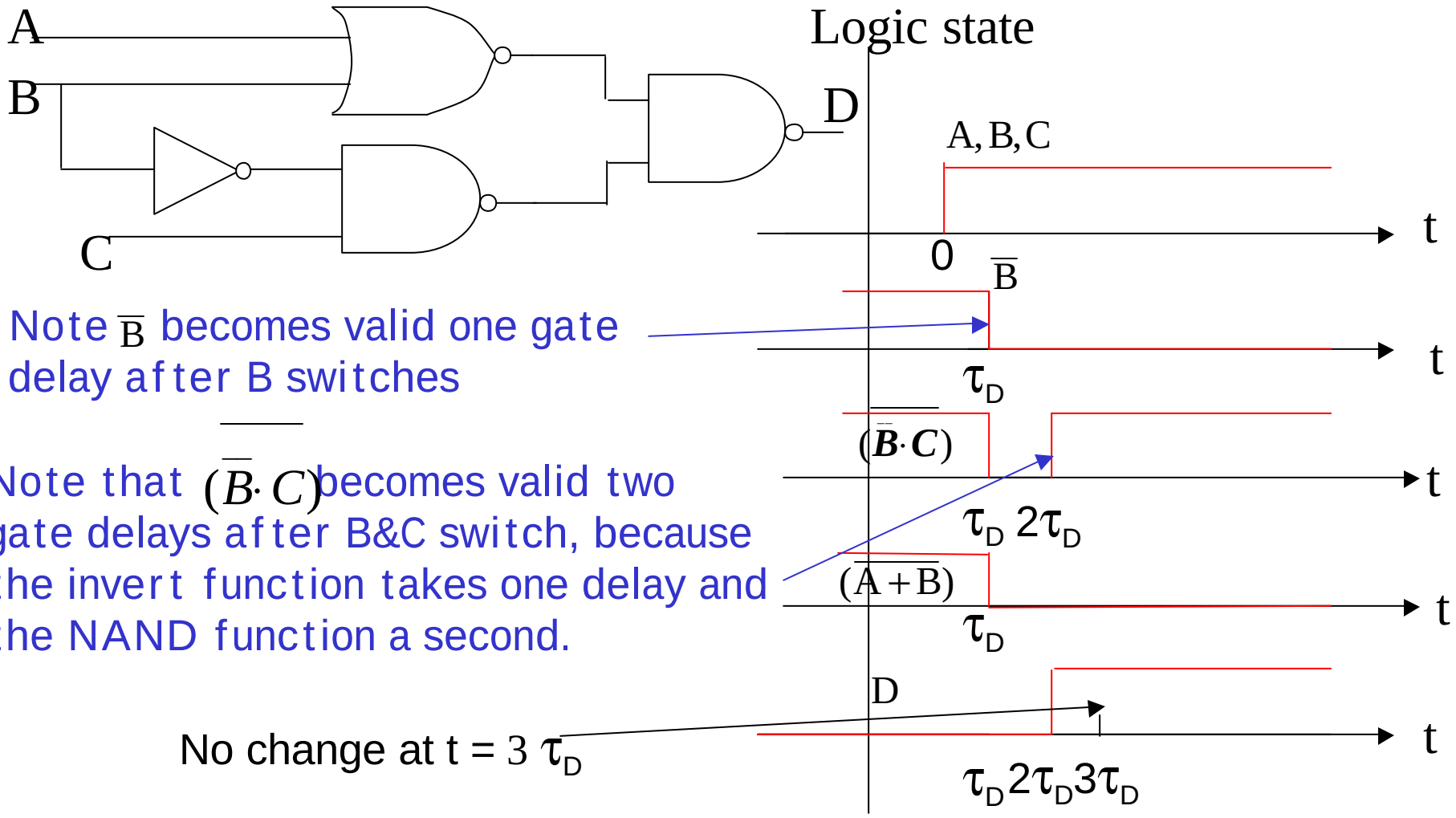
How many gate delays for longest path? ANSWER : 3

Which path is the important one? ANSWER : LONGEST

# TIMING DIAGRAMS

Show transitions of variables vs time




Note $\overline{B}$ becomes valid one gate delay after B switches

Note that $(\overline{B}\cdot C)$ becomes valid two gate delays after B&C switch, because the invert function takes one delay and the NAND function a second.

No change at t = 3 $\tau_D$

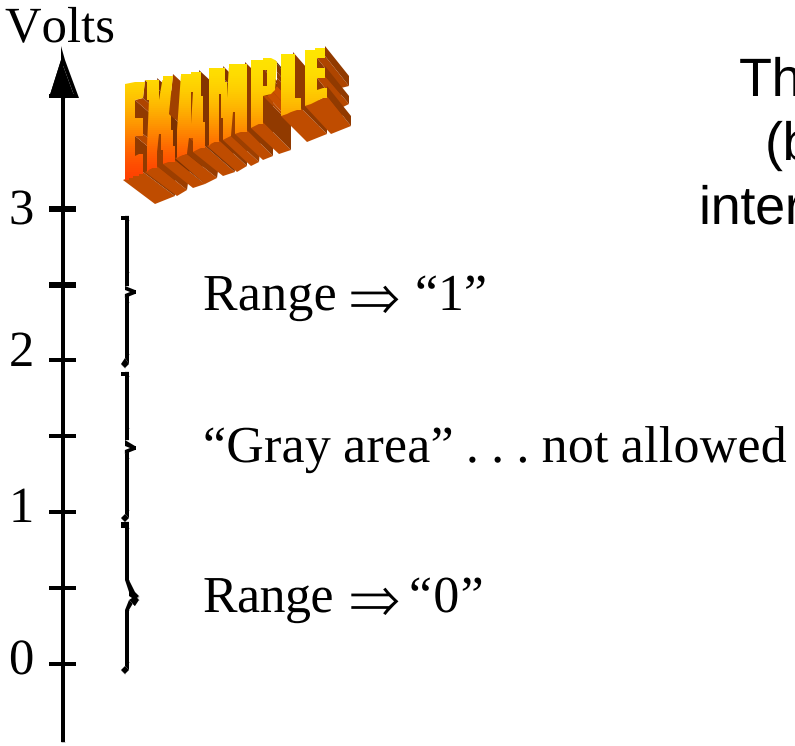
  

# WHAT IS THE ORIGIN OF GATE DELAY?

Logic gates are *electronic circuits* that process *electrical* signals

Most common signal for logic variable: voltage

Specific voltage ranges correspond to “0” or “1”



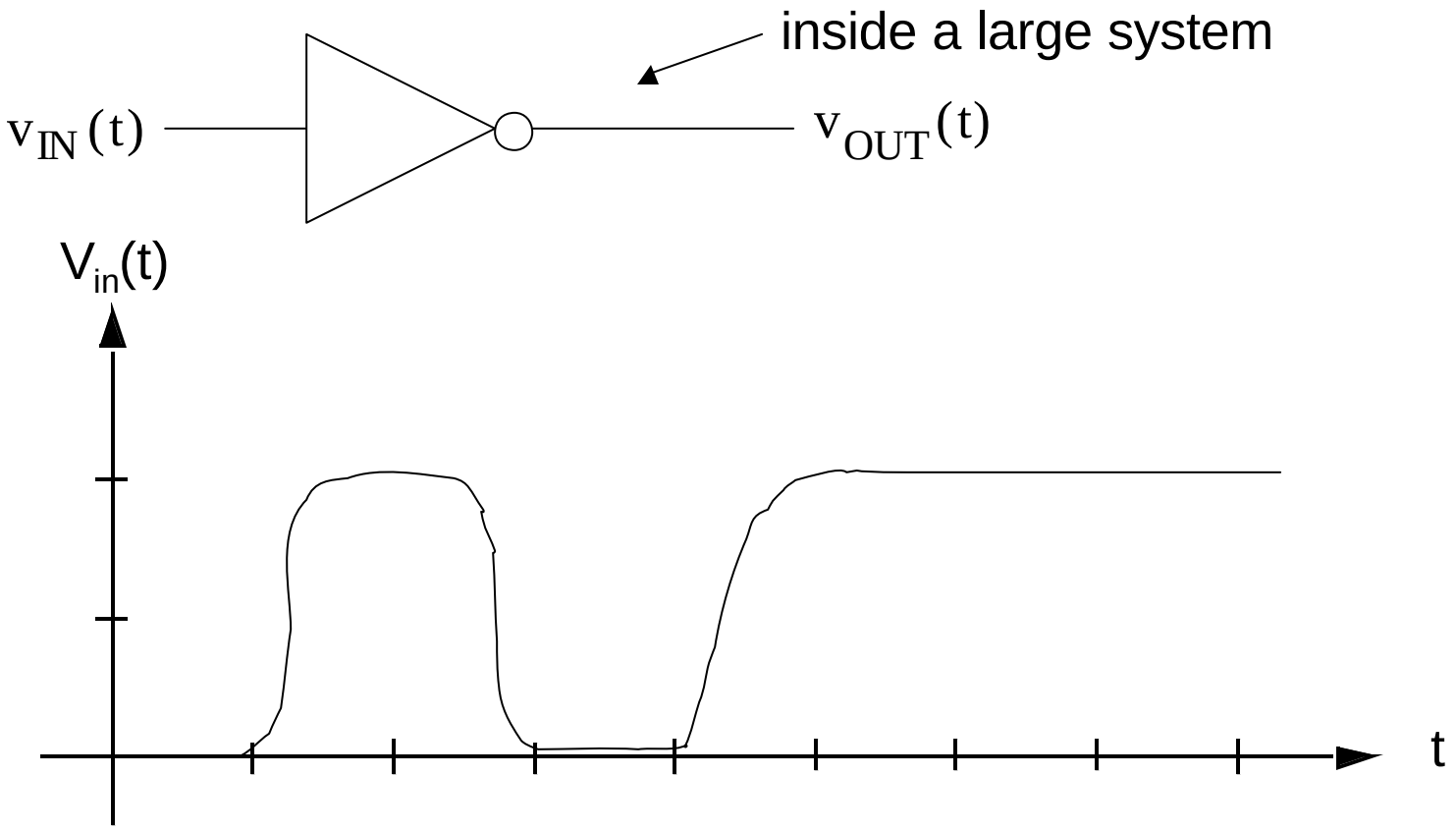

Thus delay in voltage rise or fall (because of delay in charging internal capacitances) will translate to a delay in signal timing

Note that the specific voltage range for 0 or 1 depends on “logic family,” and in general decreases with logic generations

# VOLTAGE WAVEFORMS (TIME FUNCTIONS)

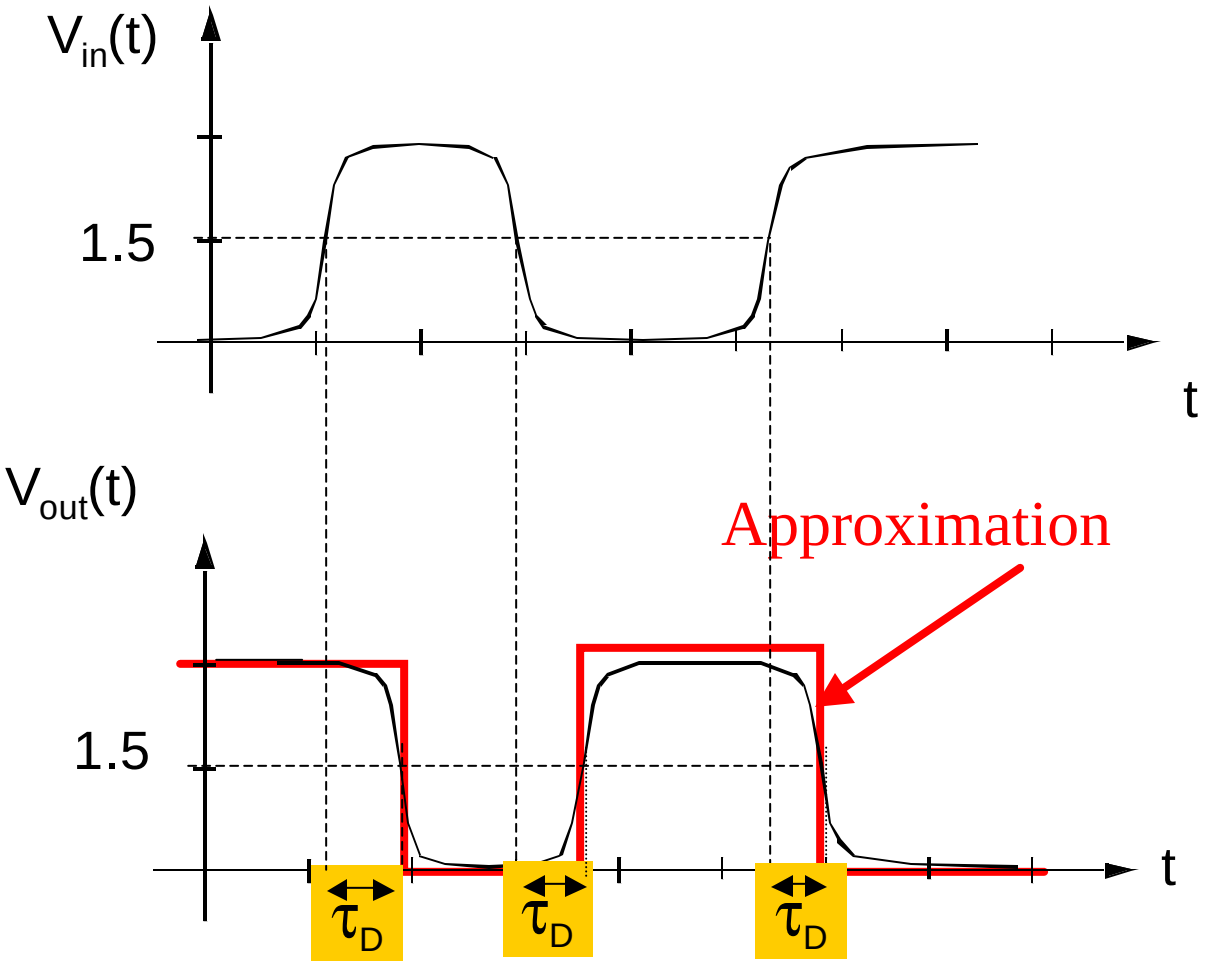
Inverter input is $v_{IN}(t)$, output is $v_{OUT}(t)$



# GATE DELAY (PROPAGATION DELAY)

Define $\tau$ as the delay required for the output voltage to reach 50% of its final value. In this example we will use 3V logic, so halfway point is 1.5V. Inverters are designed so that the gate delay is symmetrical (rise and fall)



# EFFECT OF PROPAGATION DELAY ON PROCESSOR SPEED

Computer architects would like each system clock cycle to have between 20 and 50 gate delays … use 35 for calculations

Implication: if clock frequency = 500 MHz clock period = $(5\times10^{8}\ s^{-1})^{-1}$

Period = $2 \times 10^{-9}$s = 2 ns (nanoseconds)

Gate delay must be $\tau_D = (1/35) \times$ Period = (2 ns)/35 = 57 ps (picoseconds)

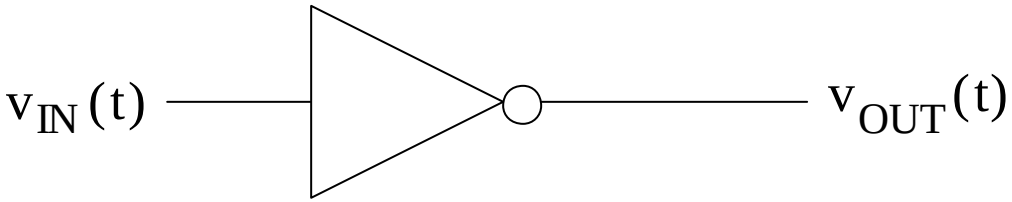
How fast is this? Speed of light: $c = 3 \times 10^{8}$ m/s

Distance traveled in 57 ps is:

$C \times \tau_D = (3x10^{8}m/s)(57x10^{-12}s) = 17 \times 10^{-4}$ m = 1.7cm

# WHAT DETERMINES GATE DELAY?




The delay is mostly simply the charging of the capacitors at internal nodes.

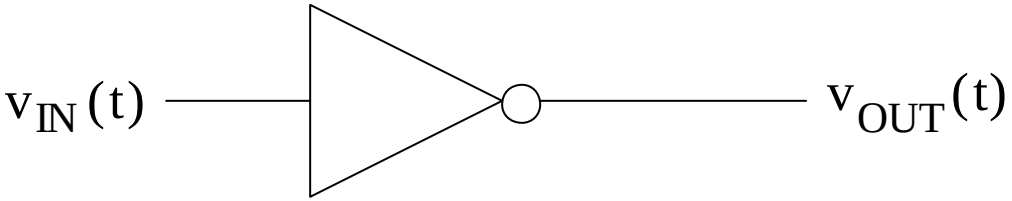
We already know how to analyze this.

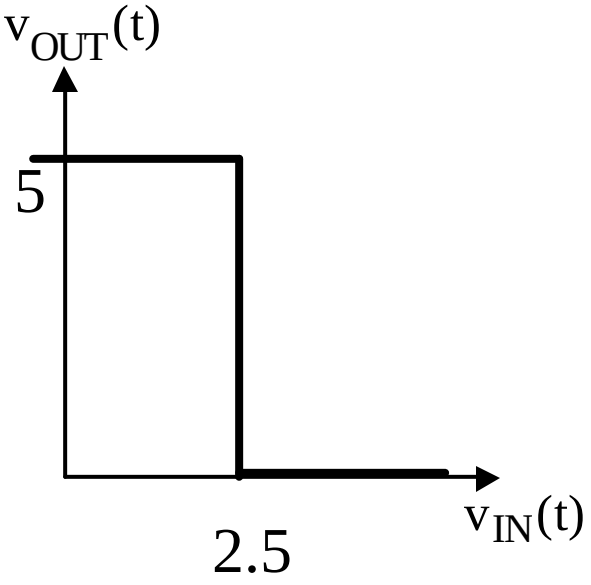
  

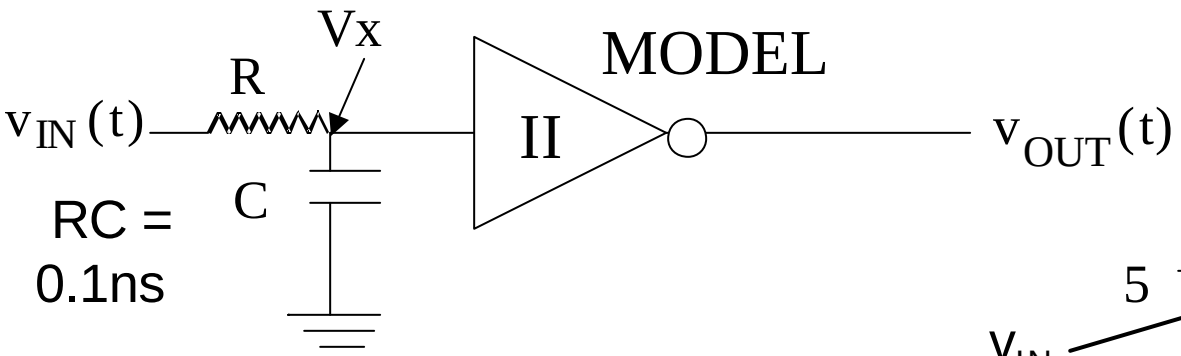
# Example

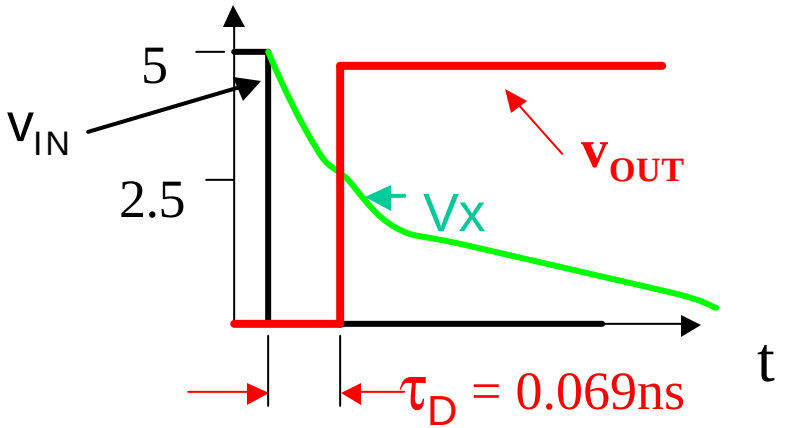
The gate delay is simply the charging of the capacitors at internal nodes.

Oversimplified example using "ideal inverter, II" and 5V logic swing



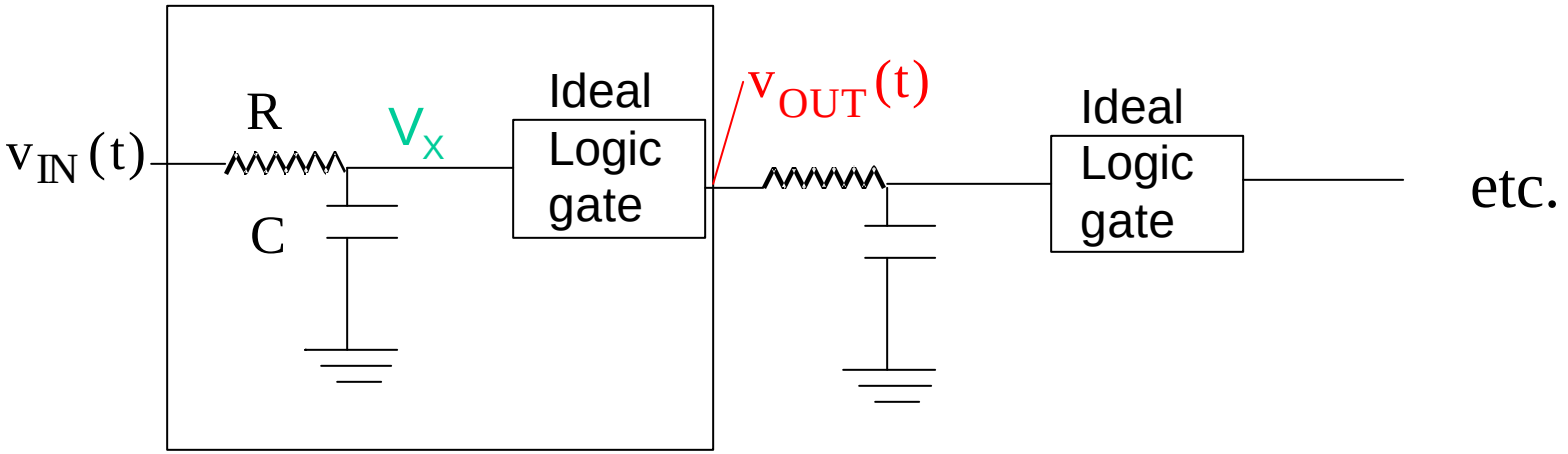




RC = 0.1ns so 0.069ns after $v_{IN}$ switches by 5V, Vx moves 2.5V

# Simple model for logic delays

Model actual logic gate as an ideal logic gate fed by an RC network which represents the dominant R and C in the gate.




Actual Logic Gate