Digital Logic Design

Lecture 4

Timing

The functional specification of a combinational circuit expresses the output values in terms of the current input values.

The timing specification of a combinational circuit consists of lower and upper bounds on the delay from input to output.

Whenever abstract logic gates have physical realizations, the waveforms show delays between transitions on the inputs and corresponding responses at the outputs.

The actual values of these delays depend on the technology used to implement the gates.

Timing hazards

Timing hazards are problems in a circuit as a result of timing issues.

These problems can be observed only from a timing analysis of the circuit or from an actual implementation of the circuit.

A functional analysis of the circuit will not reveal timing hazard problems.

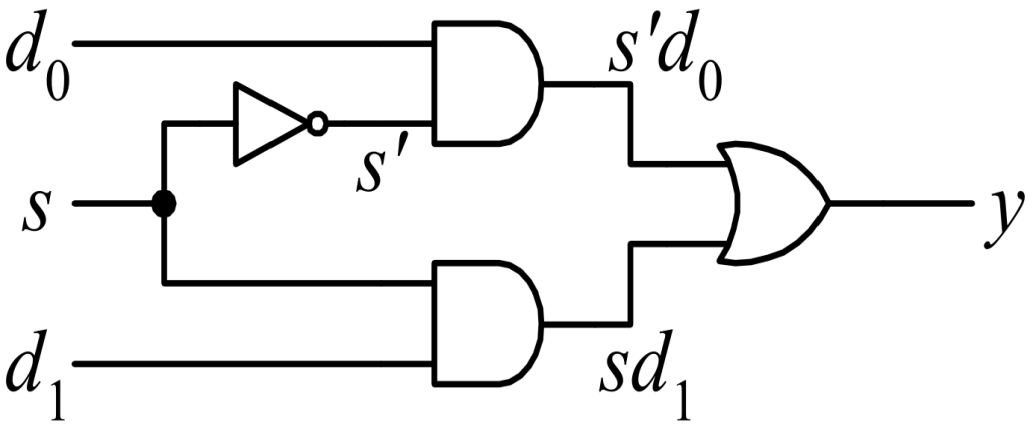
Glitches

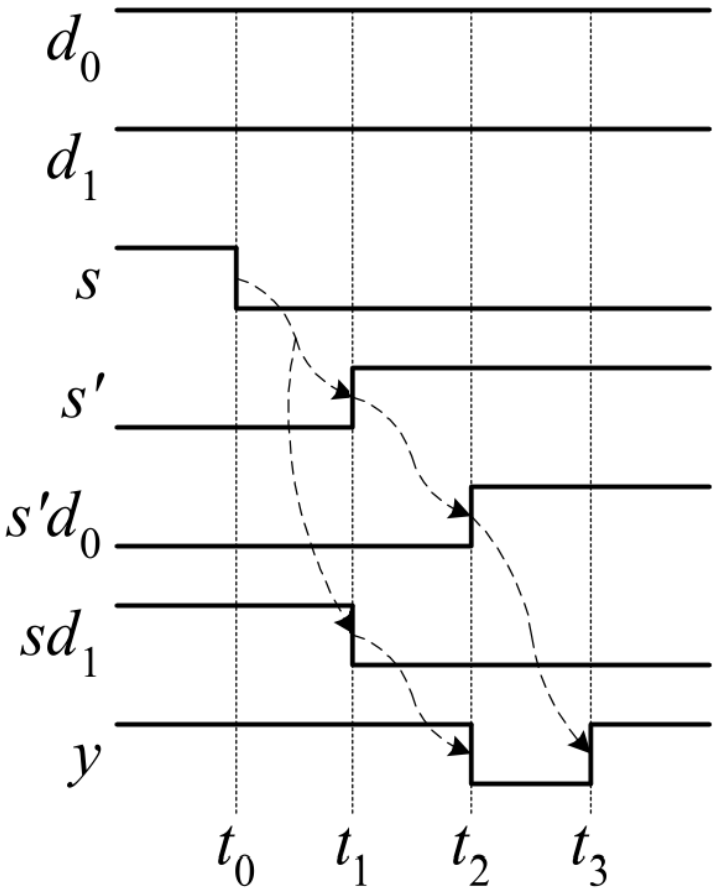
A **glitch** is when a signal is expected to be stable (from a functional analysis), but it changes value for a brief moment and then goes back to what it is expected to be.

For example, if a signal is expected to be at a stable 0, but instead, it goes up to a 1 and then drops back to a 0 very quickly.

This sudden, unexpected transition of the signal is a glitch, and the circuit having this behavior contains a hazard.

Example: 2-to-1 multiplexer



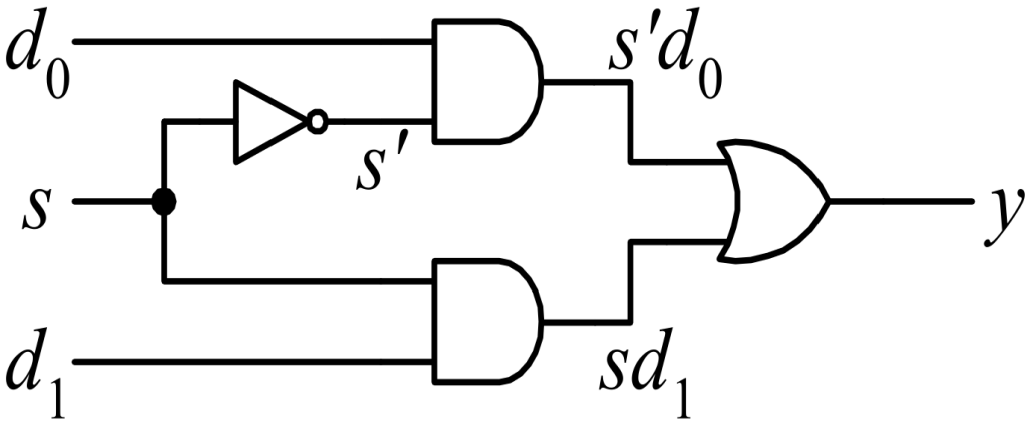


Assume that $d_0 = 1$ and $d_1 = 1$, and that s goes from a 1 to a 0.

For a functional analysis of the circuit, y should remain 1.

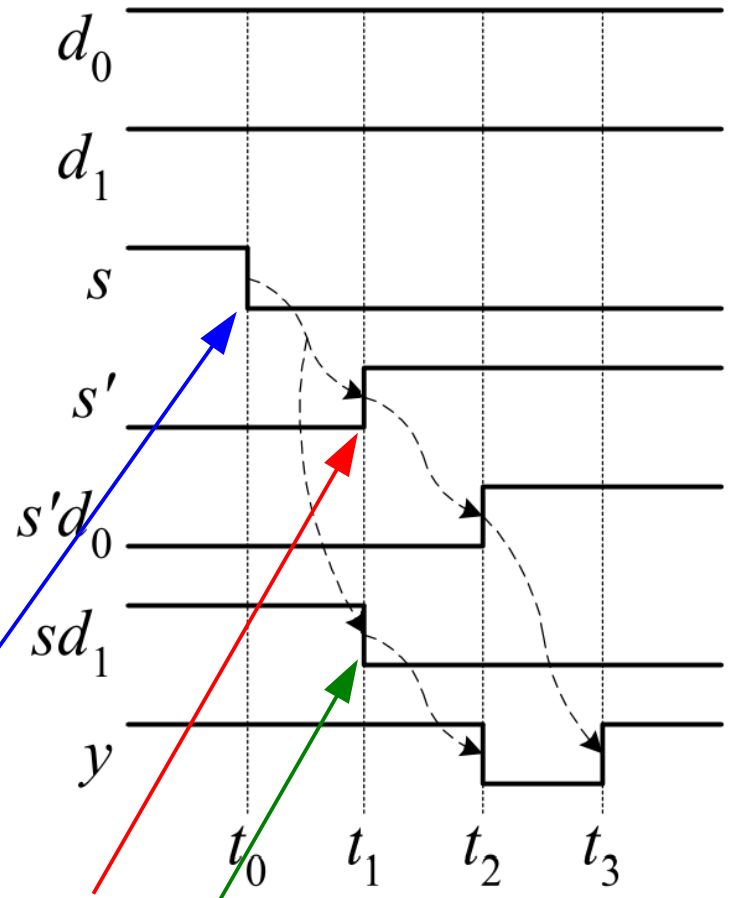
However, a timing analysis shows something different.

Example: 2-to-1 multiplexer



Assume that all of the logic gates in the circuit have a delay of one time unit.

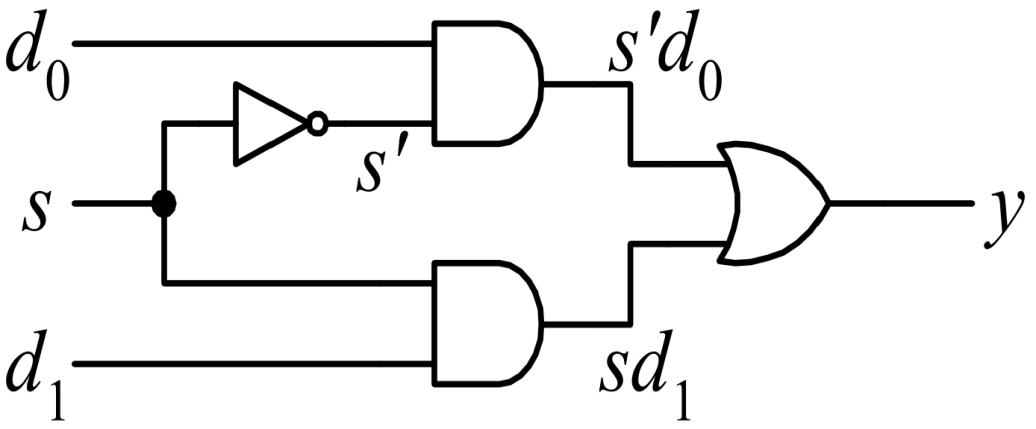
At time t_0, s drops to 0

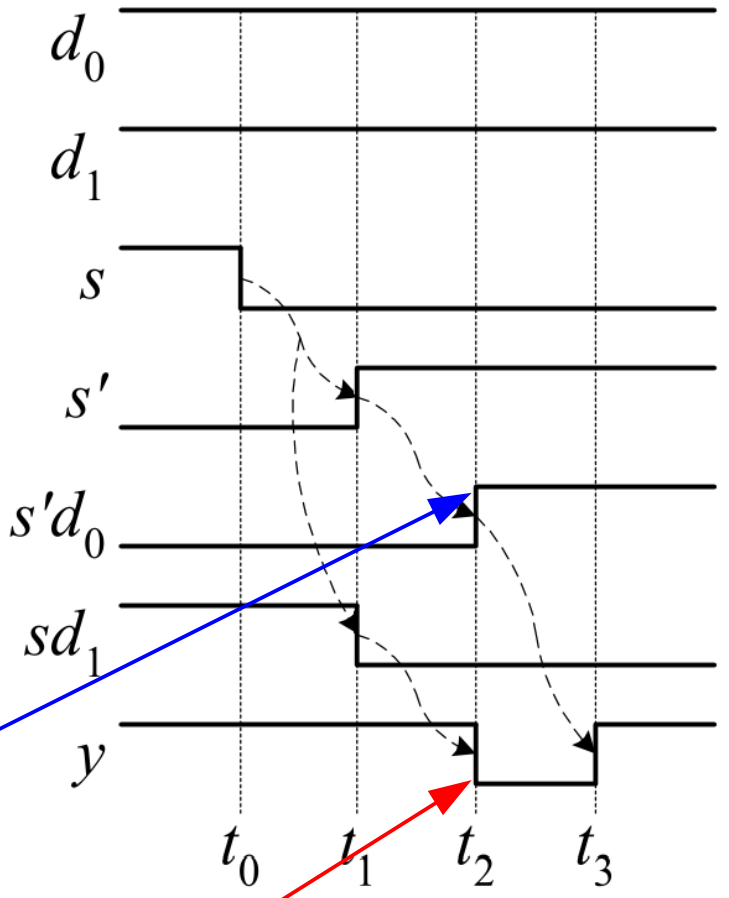


Since it takes one time unit for s to be inverted through the inverter, s' changes to a 1 after one time unit at time t_1

At the same time, it takes the bottom AND gate one time unit for the output sd_1 to change to a 0 at time t_1

Example: 2-to-1 multiplexer

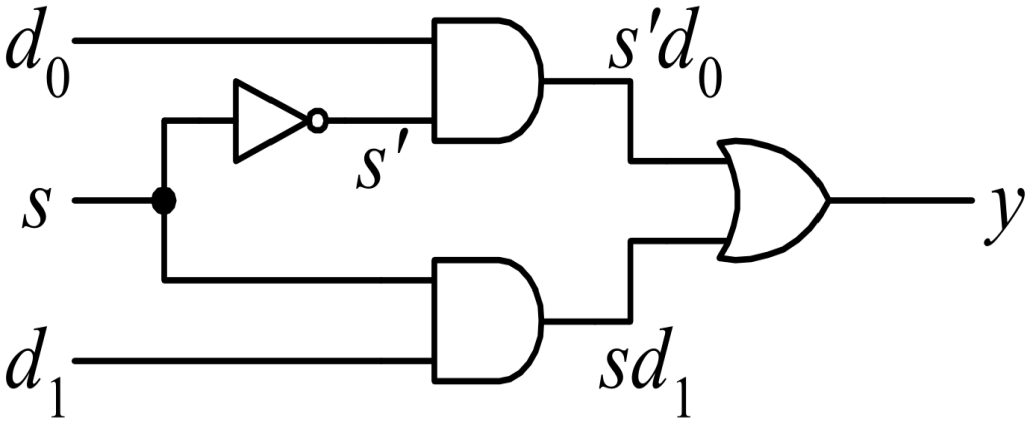


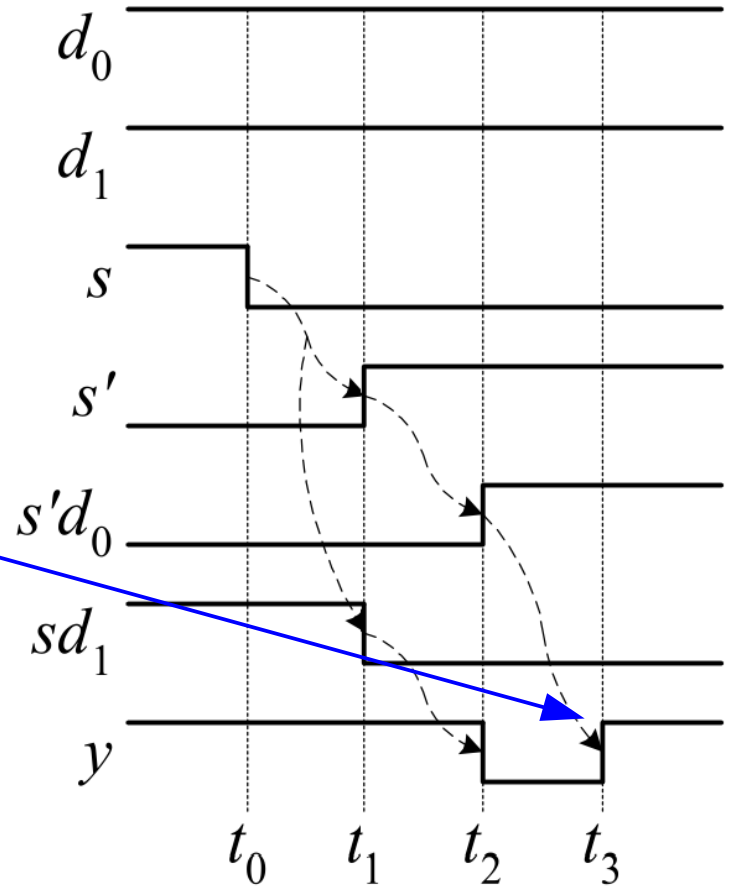


The top AND gate will not see any input change until time t_1, and when it does, it takes another one time unit for its output $s'd_0$ to rise to a 1 at time t_2

Starting at time t_1, both inputs of the OR gate are 0, so after one time unit, the OR gate outputs a 0 at time t_2

Example: 2-to-1 multiplexer





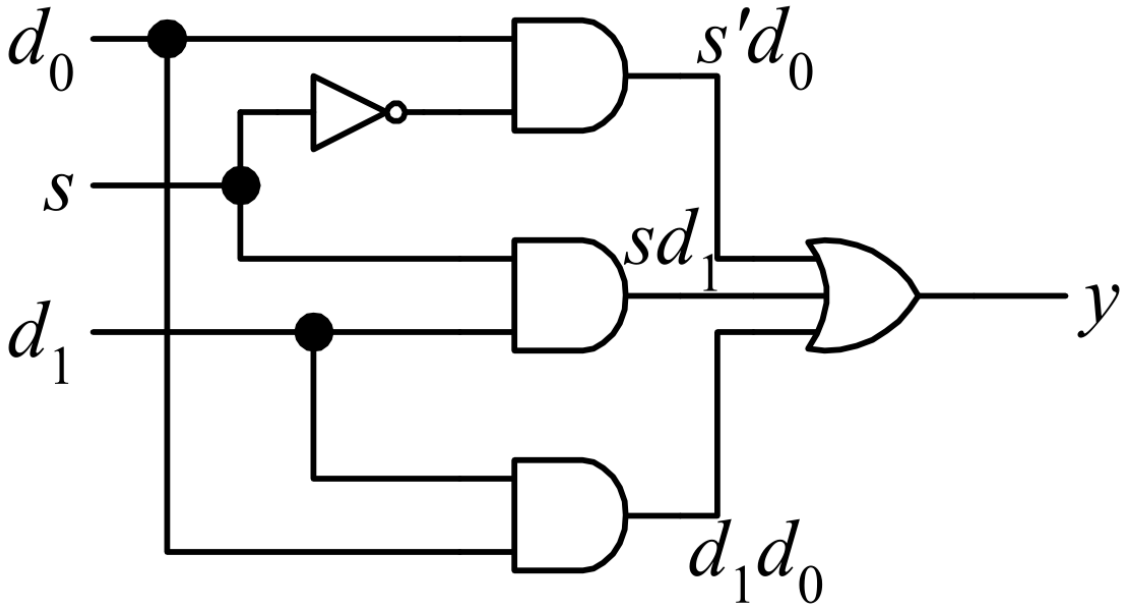
At time t_2, when the top AND gate outputs a 1, the OR gate will take this 1 input, and outputs a 1 after one time unit at t_3.

So between times t_2 and t_3, output y unexpectedly drops to a 0 for one time unit, and then rises back to a 1.

Hence, the output signal y has a glitch, and the circuit has a hazard.

Example: 2-to-1 multiplexer

Glitches in a signal are caused by multiple sources having paths of different delays driving that signal.



The 2-to-1 multiplexer circuit with the extra d_1d_0 added will prevent the glitch from happening.

Example of using glitches: A one-shot circuit

Sometimes, we can use glitches to our advantage.

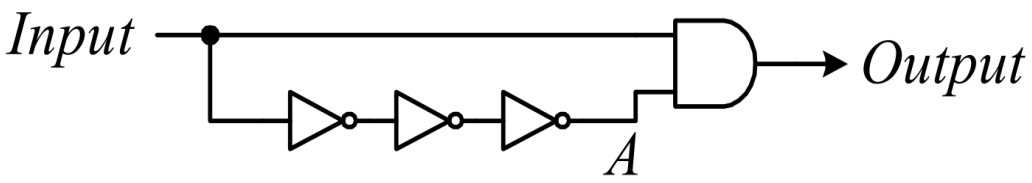
A circuit that outputs a single short pulse when given an input of arbitrary time length is known as a one-shot.

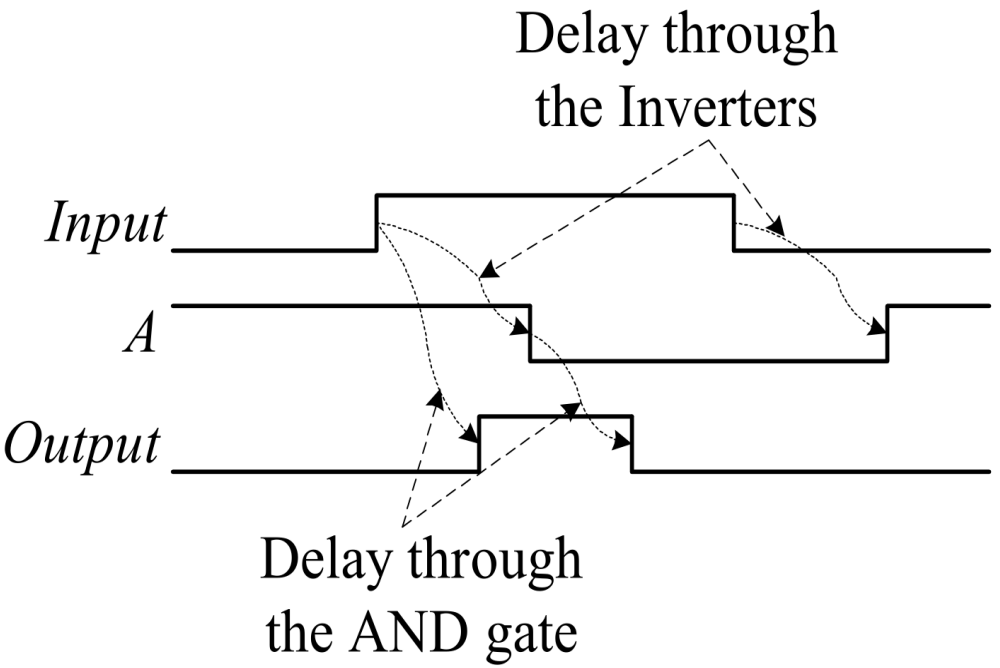
It is used, for example, for generating a single short 1 pulse when a key is pressed.

Sometimes, when a key is pressed, we do not want to generate a continuous 1 signal for as long as the key is pressed.

Instead, we want the output signal to be just a single short 1 pulse, even if the key is still being pressed.

Example of using glitches: A one-shot circuit



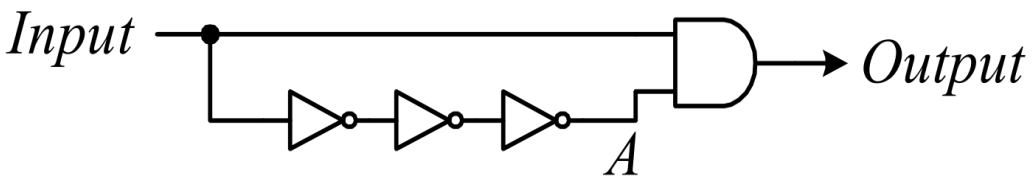


Initial value for Input is a 0, and point A is a 1 => the output of the AND gate is a 0.

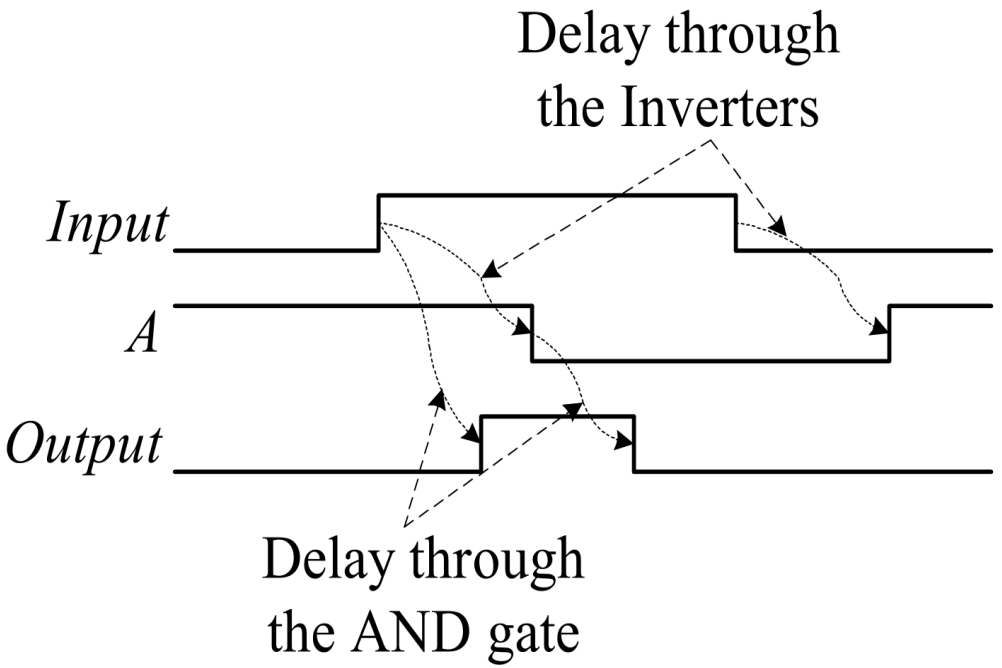
When we set Input = 1, both inputs to the AND gate will be 1's, and so after a delay through the AND gate, Output will be 1

After a delay through the three inverters, with Input still at a 1, point A will go to a 0, and Output will change back to a 0.

Example of using glitches: A one-shot circuit



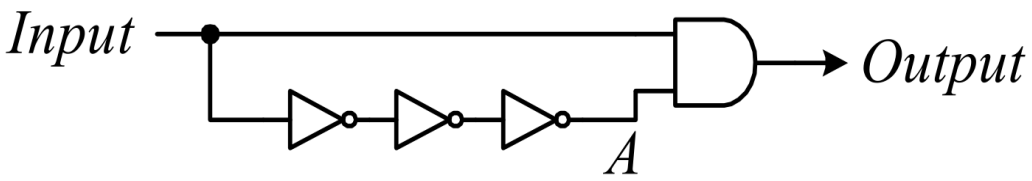
When we set Input back to a 0, Output will continue to be a 0.

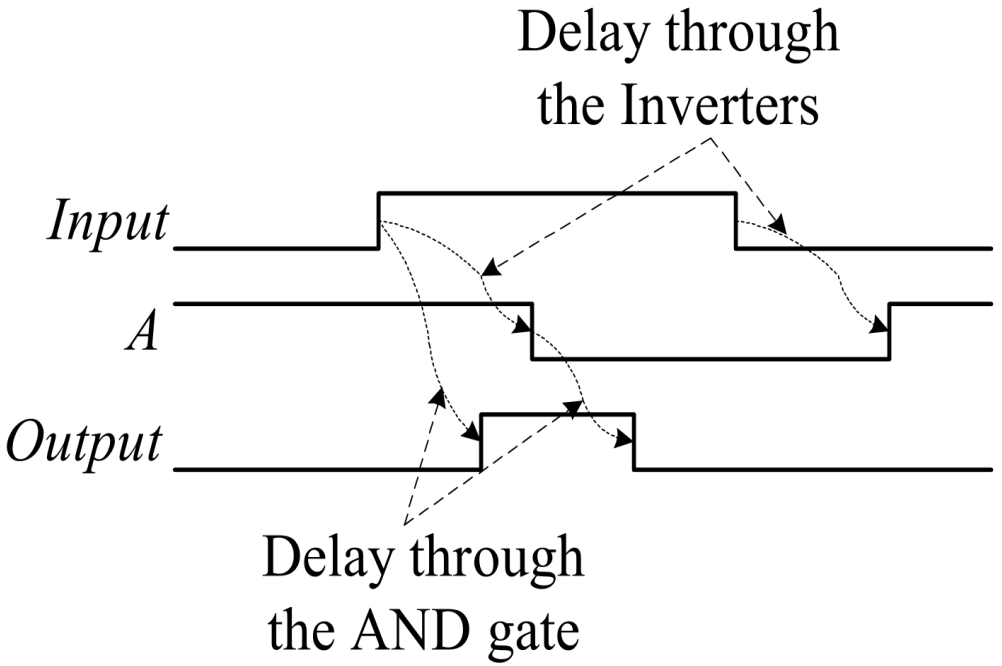


After the delay through the inverters when point A goes back to a 1, Output remains at a 0.

As a result, a glitch is created by the signal delay through the three inverters.

Example of using glitches: A one-shot circuit

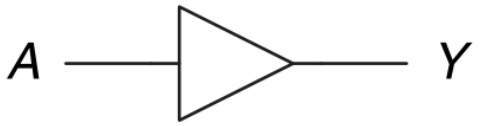




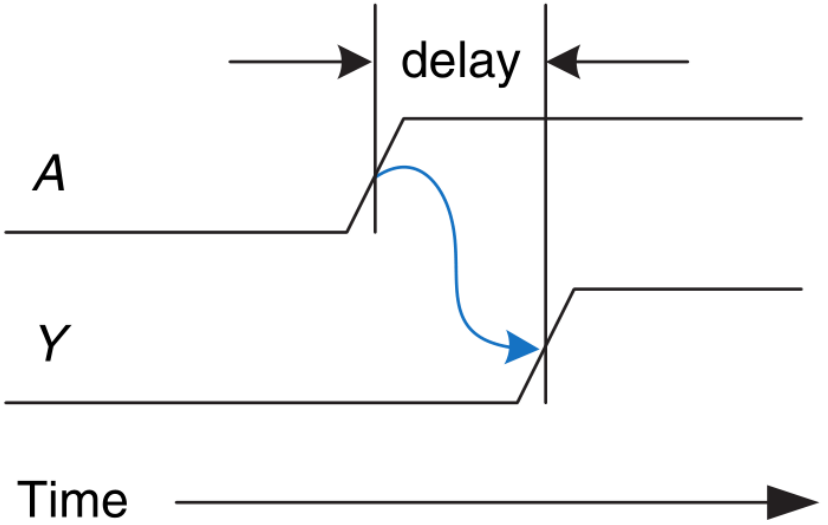
This glitch, however, is the short 1 pulse that we want, and the length of this pulse is determined by the delay through the inverters.

With this one-shot circuit, it does not matter how long the input key is being pressed, the output signal will always be the same 1 pulse each time that the key is pressed.

Circuit delay



This **timing diagram** portrays the **transient response** of the buffer circuit when an input changes.



The transition from LOW to HIGH is called the **rising edge**

The transition from HIGH to LOW is called the **falling edge**

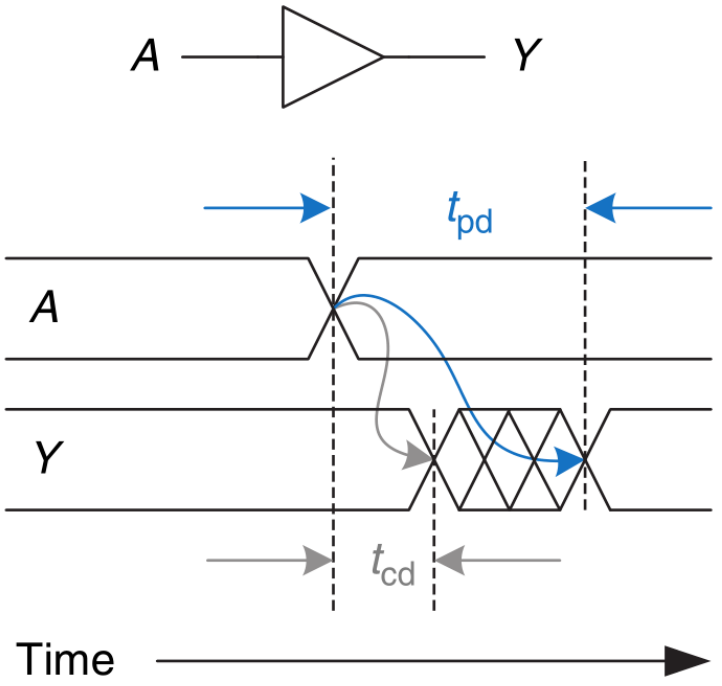
We measure delay from the 50% point of the input signal, *A*, to the 50% point of the output signal, *Y*.

t_{pd} – **propagation delay**

– the maximum time from when an input changes until the output reaches its final value

t_{cd} – **contamination delay**

– the minimum time from when an input changes until any output starts to change its value

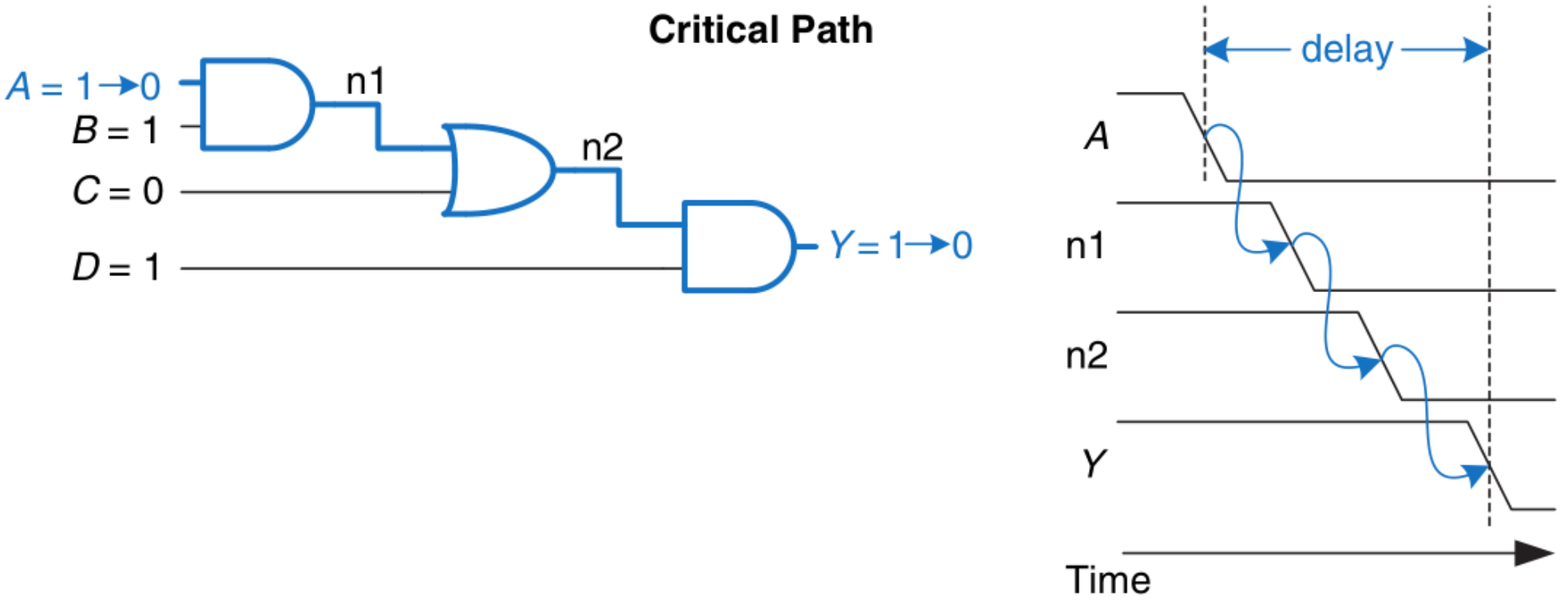


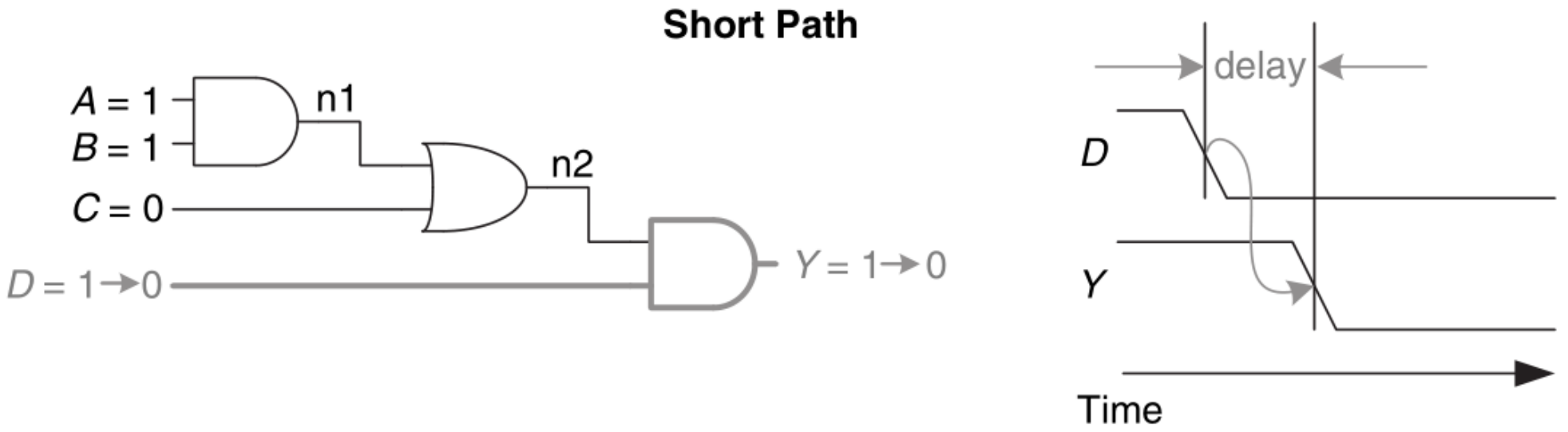
t_{pd} and t_{cd} may be different because of

– different rising and falling delays

– multiple inputs and outputs, some of which are faster than others

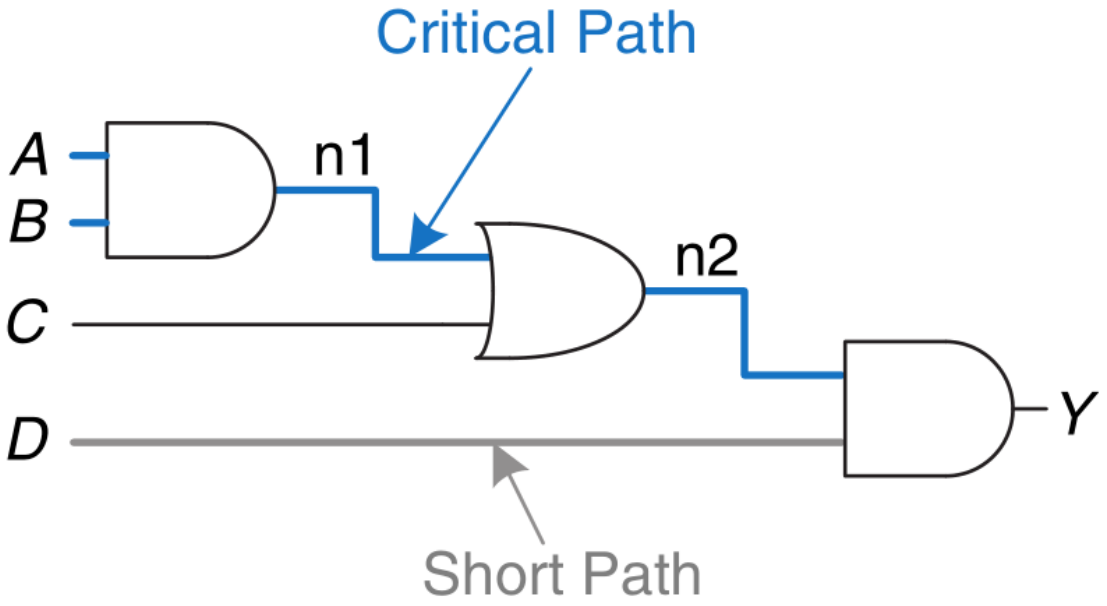
– circuits slowing down when hot and speeding up when cold


Critical Path
A = 1→0
B = 1
C = 0
D = 1
n1
n2
Y = 1→0
delay
A
n1
n2
Y
Time


Short Path
A = 1
B = 1
C = 0
D = 1→0
n1
n2
Y = 1→0
delay
D
Y
Time

The **propagation delay** of a combinational circuit is the sum of the propagation delays through each element on the critical path

The **contamination delay** is the sum of the contamination delays through each element on the short path



$$t_{pd} = 2\,t_{pdAND} + t_{pdOR}$$

$$t_{cd} = t_{cdAND}$$

Circuit delays are on the order of picoseconds ($1\ ps = 10^{-12}\ s$) to nanoseconds ($1\ ns = 10^{-9}\ s$)

Delays in Verilog

HDL statements may be associated with delays

They are helpful

– during simulation to predict how fast a circuit will work
(if you specify meaningful delays)

– for debugging purposes to understand cause and effect

These delays are ignored during synthesis

The delay of a gate produced by the synthesizer depends on its t_{pd} and t_{cd} specifications, not on numbers in HDL code.

$$y = \bar{a}\,\bar{b}\,\bar{c} + a\,\bar{b}\,\bar{c} + a\,\bar{b}\,c$$

$$t_{pdNOT} = 1\,ns$$

$$t_{pdAND3} = 2\,ns$$

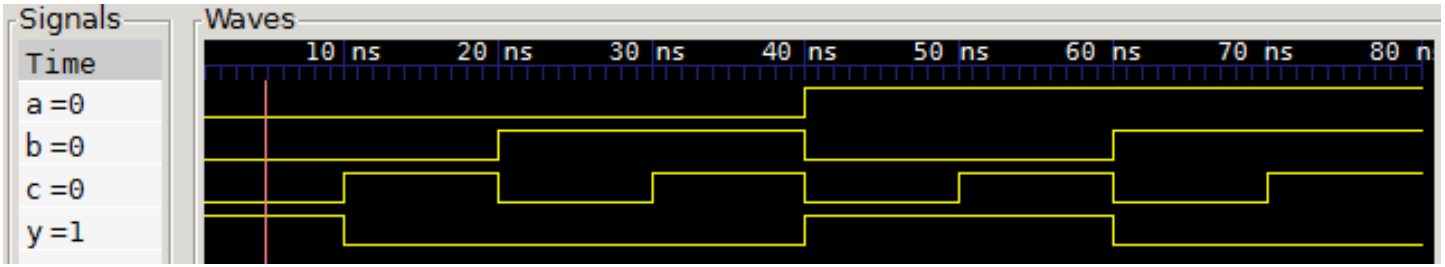
$$t_{pdOR3} = 4\,ns$$

A **timescale directive** indicates the value of each time unit.

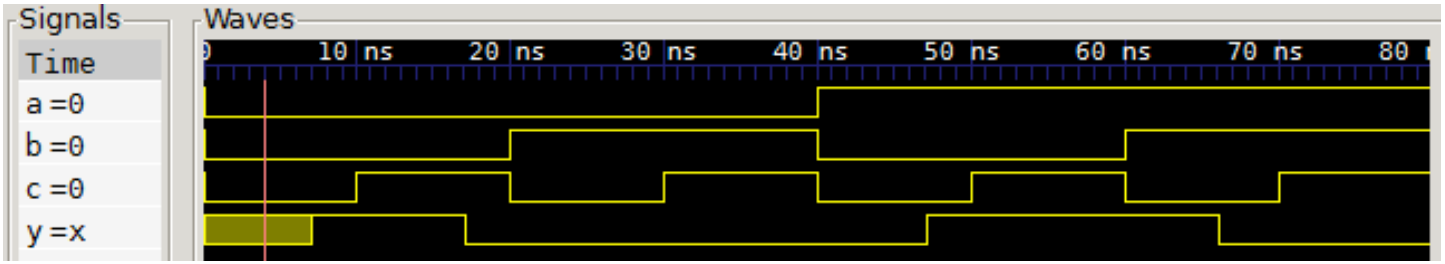
In this file, each unit is 1ns, and the simulation has 1ps precision

```
`timescale 1ns/1ps

module example (input a, b, c,
                output y);
    wire ab, bb, cb, n1, n2, n3;
    assign #1 {ab, bb, cb} = ~ {a, b, c};
    assign #2 n1 = ab & bb & cb;
    assign #2 n2 = a & bb & cb;
    assign #2 n3 = a & bb & c;
    assign #4 y = n1 | n2 | n3;
endmodule
```

without delays



with delays



The output is lagging 7 ns after the inputs

y is initially unknown at the beginning of the simulation

Gate level modeling in Verilog

The gates have one scalar output and multiple scalar inputs.

`and`	*N*-input AND gate
`nand`	*N*-input NAND gate
`or`	*N*-input OR gate
`nor`	*N*-input NOR gate
`xor`	*N*-input XOR gate
`xnor`	*N*-input XNOR gate

The 1st terminal in the list of gate terminals is an output and the other terminals are inputs.

Transmission gate primitives in Verilog

Transmission gates are bi-directional and can be resistive or non-resistive.

Syntax:

```
keyword unique_name (inout1, inout2, control);
```

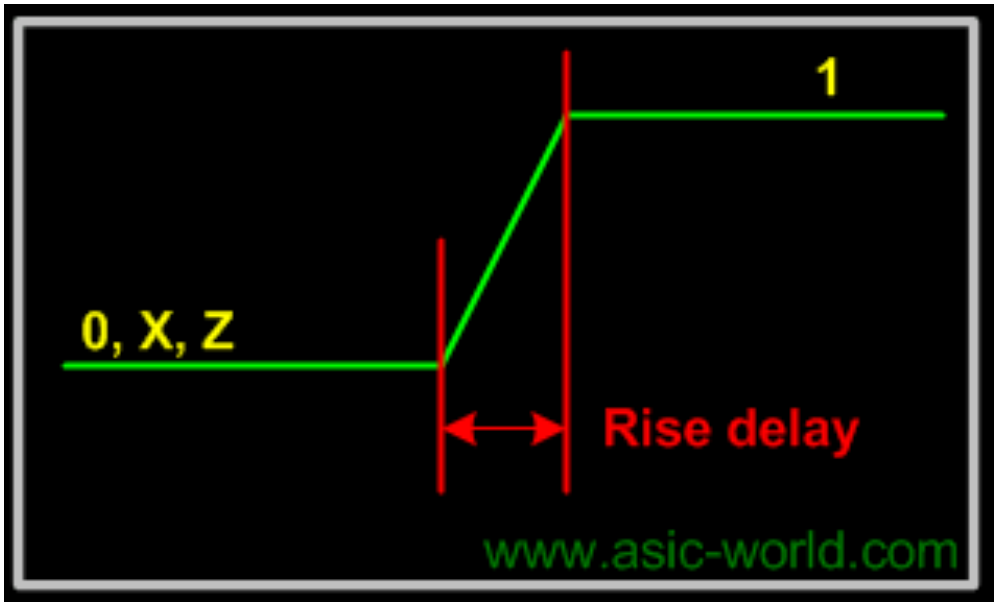
`not`	*N*-output inverter
`buf`	*N*-output buffer
`bufif0`	Tri-state buffer, Active low en.
`bufif1`	Tri-state buffer, Active high en.
`notif0`	Tristate inverter, Low en.
`notif1`	Tristate inverter, High en.

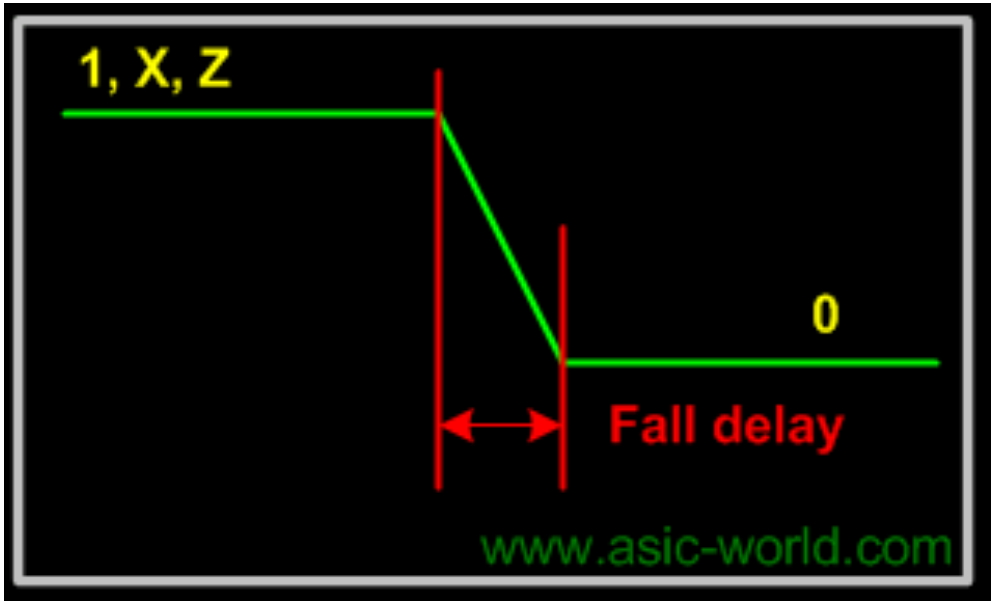
```
module gates();

wire outNOT;
wire outAND4;
wire outXOR3;
reg  in1,in2,in3,in4;

not U1(outNOT,in1);
and U2(outAND4,in1,in2,in3,in4);
xor U3(outXOR3,in1,in2,in3);

  initial begin
    $monitor(
    "in1=%b in2=%b in3=%b in4=%b outNOT=%b outAND4=%b outXOR3=%b",
    in1,in2,in3,in4,outNOT,outAND4,outXOR3);
    in1 = 0;
    in2 = 0;
    in3 = 0;
    in4 = 0;
    #1 in1 = 1;
    #1 in2 = 1;
    #1 in3 = 1;
    #1 in4 = 1;
    #1 $finish;
  end

endmodule
```

```
C:\iverilog\samples>iverilog -o gates gates.v

C:\iverilog\samples>vvp gates
in1=0 in2=0 in3=0 in4=0 outNOT=1 outAND4=0 outXOR3=0
in1=1 in2=0 in3=0 in4=0 outNOT=0 outAND4=0 outXOR3=1
in1=1 in2=1 in3=0 in4=0 outNOT=0 outAND4=0 outXOR3=0
in1=1 in2=1 in3=1 in4=0 outNOT=0 outAND4=0 outXOR3=1
in1=1 in2=1 in3=1 in4=1 outNOT=0 outAND4=1 outXOR3=1
```

Gate delays in Verilog

The **rise delay** is associated with a gate output transition to 1 from another value (0, x, z).





The **fall delay** is associated with a gate output transition to 0 from another value (1, x, z).

Example: single delay

```
`timescale 1ns/1ps
module  buf_gate ();
reg in;
wire out;

buf #(5) (out,in);

  initial begin
    $dumpfile("buf_gate.vcd");
    $dumpvars(0, in, out);
  end

  initial begin
    $monitor ("Time = %g in = %b out=%b", $time, in, out);
    in = 0;
    #10 in = 1;
    #10 in = 0;
    #10 $finish;
  end

endmodule
```

```
C:\iverilog\samples>iverilog -o buf_gate buf_gate.v

C:\iverilog\samples>vvp buf_gate
VCD info: dumpfile buf_gate.vcd opened for output.
Time = 0 in = 0 out=x
Time = 5 in = 0 out=0
Time = 10 in = 1 out=0
Time = 15 in = 1 out=1
Time = 20 in = 0 out=1
Time = 25 in = 0 out=0
```

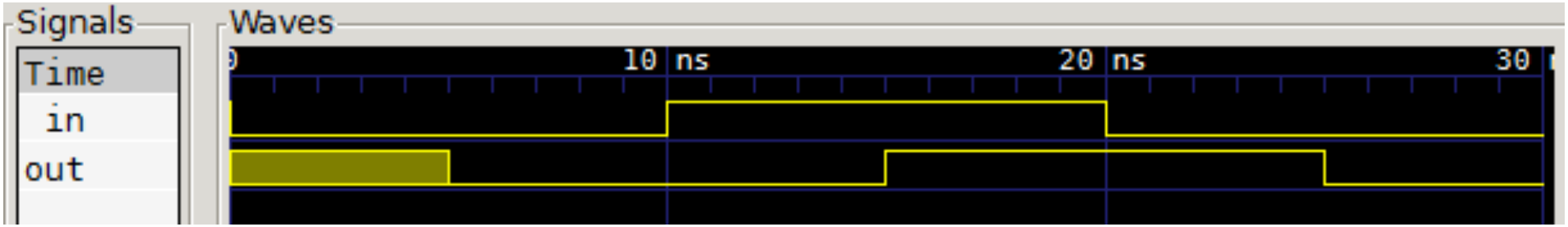
Example: two delays

```
`timescale 1ns/1ps
module  buf_gate1 ();
reg in;
wire out;

buf #(2,3) (out,in);

  initial begin
    $dumpfile("buf_gate1.vcd");
    $dumpvars(0, in, out);
  end

  initial begin
    $monitor ("Time = %g in = %b out=%b", $time, in, out);
    in = 0;
    #10 in = 1;
    #10 in = 0;
    #10 $finish;
  end

endmodule
```

```
C:\iverilog\samples>iverilog -o buf_gate1 buf_gate1.v

C:\iverilog\samples>vvp buf_gate1
VCD info: dumpfile buf_gate1.vcd opened for output.
Time = 0 in = 0 out=x
Time = 3 in = 0 out=0
Time = 10 in = 1 out=0
Time = 12 in = 1 out=1
Time = 20 in = 0 out=1
Time = 23 in = 0 out=0
```

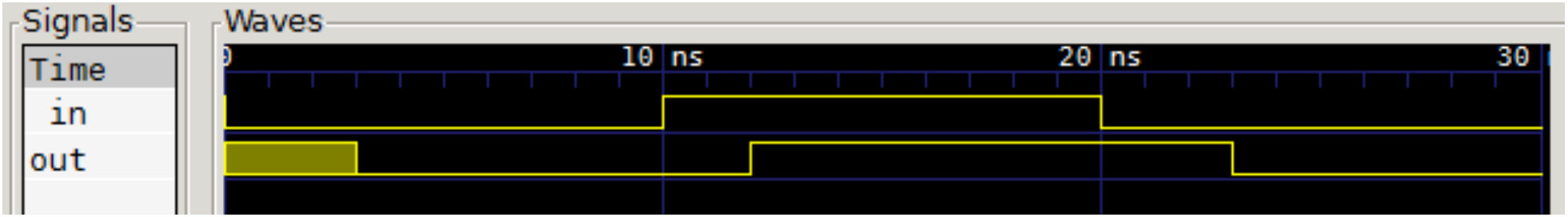
Example: single delay

```
buf #(5) (out,in);
```



Example: two delays

```
buf #(2,3) (out,in);
```



We have been ignoring wire delay so far.

We have assumed that wires are equipotential connections that have a single voltage along their entire length.

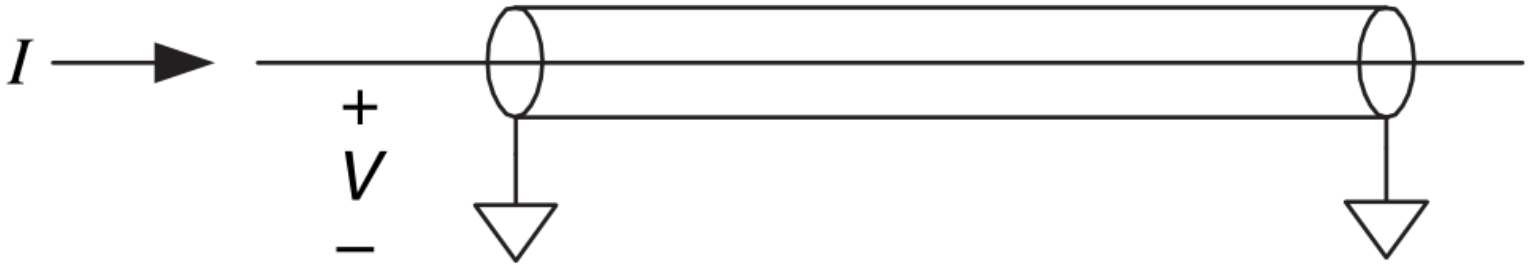
Digital circuits are now so fast that the delay of long wires can be as important as the delay of the gates.

Signals propagate along wires at the speed of light in the form of electromagnetic waves.

If the wires are short enough or the signals change slowly, the equipotential assumption is good enough.

When the wire is long or the signal is very fast, the transmission time along the wire becomes important to accurately determine the circuit delay.

We must model such wires as **transmission lines**, in which a wave of voltage and current propagates at the speed of light.



When the wave reaches the end of the line, it may reflect back along the line.

The reflection may cause noise and odd behaviors unless steps are taken to limit it.

The digital designer must consider transmission line behavior to accurately account for the delay and noise effects in long wires.

The speed of light, v, depends on the permittivity, ε, and permeability, μ, of the medium

$$v = \frac{1}{\sqrt{\mu \epsilon}} = \frac{1}{\sqrt{LC}}$$

The capacitance, *C*, and inductance, *L*, of a wire are related to the permittivity and permeability of the physical medium in which the wire is located.

The speed of light in free space is $v = c \approx 3 \times 10^8 \, m/s$

Signals in a PCB travel at about half this speed:

$$v \approx 1.5 \times 10^8 \, m/s \approx 15 \, cm/ns$$

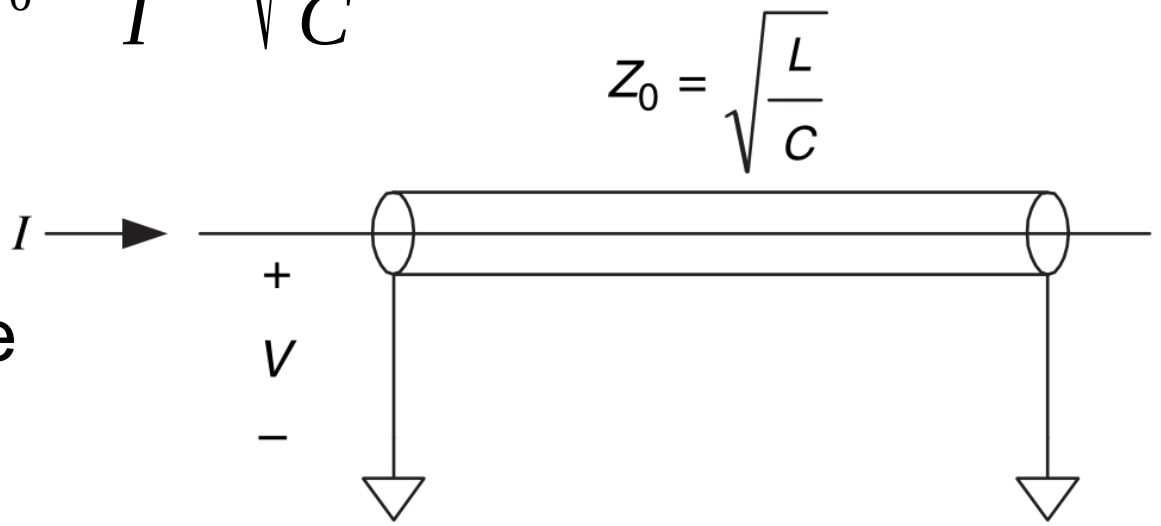
The time delay for a signal to travel along a transmission line of length ℓ is

$$t_d = \frac{l}{v}$$

The **characteristic impedance** of a transmission line is the ratio of voltage to current in a wave traveling along the line:

$$Z_0 = \frac{V}{I} = \sqrt{\frac{L}{C}}$$

Z is not the resistance of the wire

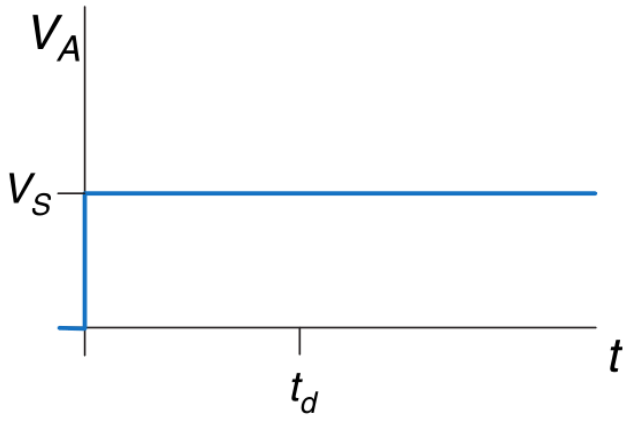


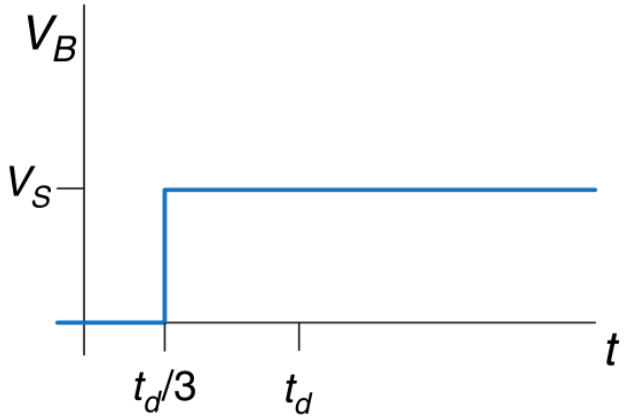
A good transmission line in a digital system has negligible resistance.

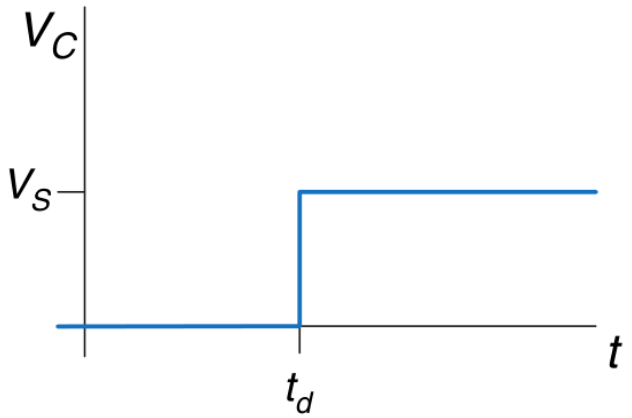
When the wave reaches the end of the line, it may be absorbed or reflected, depending on the termination or load at the end.

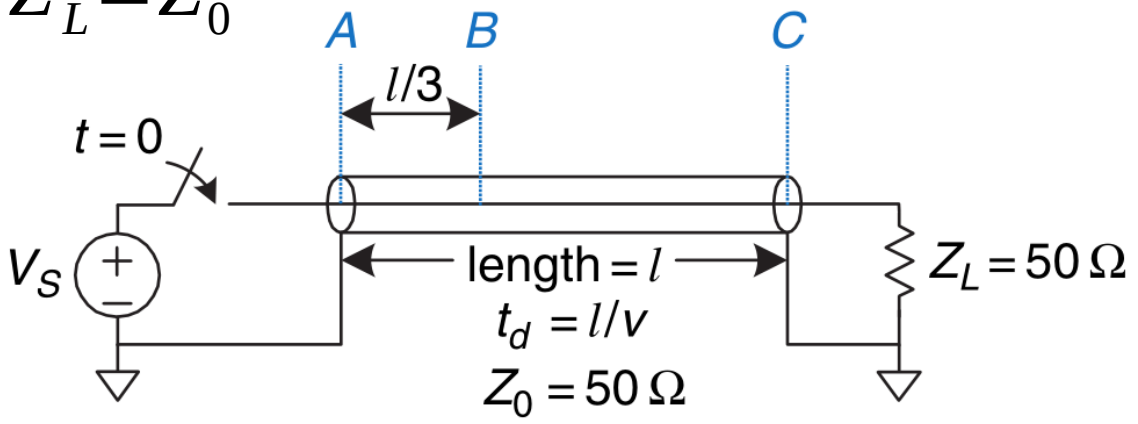
Reflections travel back along the line, adding to the voltage already on the line.

Matched termination: $Z_L = Z_0$









Z_L – the load impedance

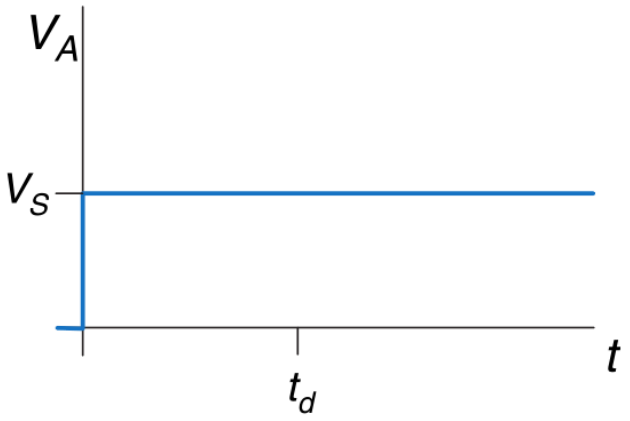
At $t = 0$ the switch closes, and

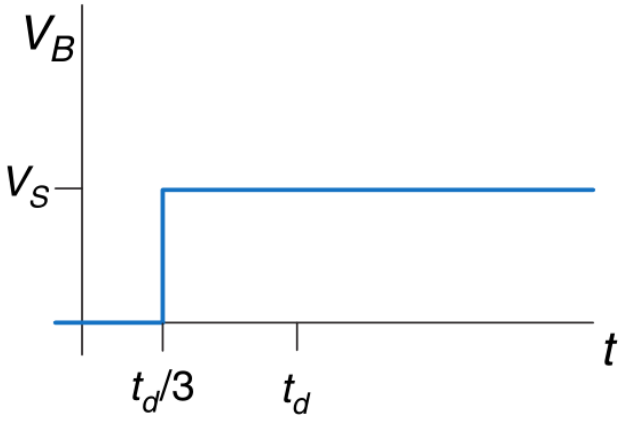
the voltage source launches an **incident wave** with voltage $V = V_S$ along the line.

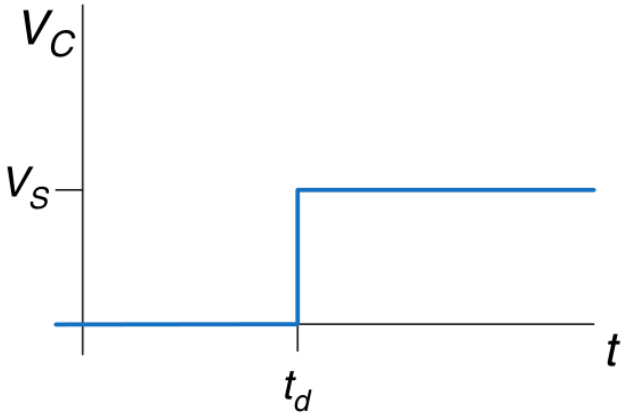
The wave has current $I = V_S / Z_0$

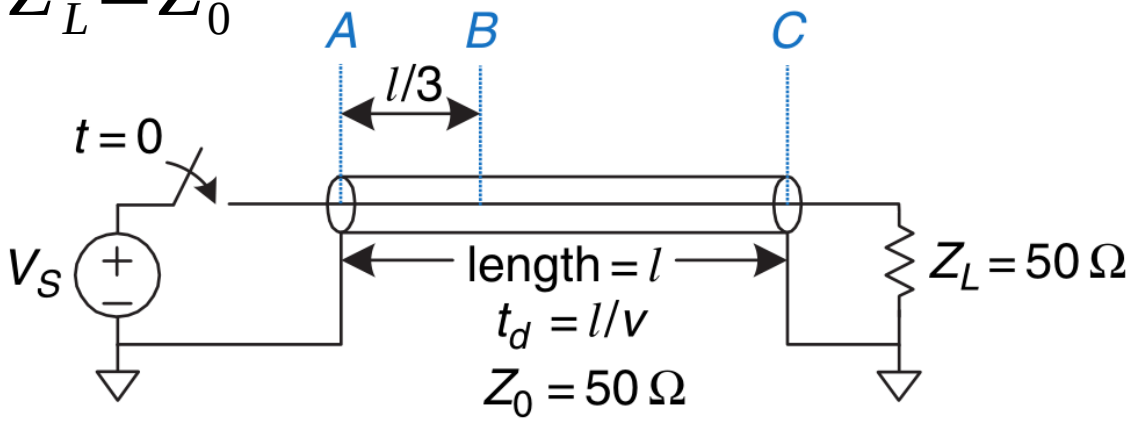
The wave propagates along the line at the speed of light

Matched termination: $Z_L = Z_0$









At $t = t_d/3$ the wave reaches *B*

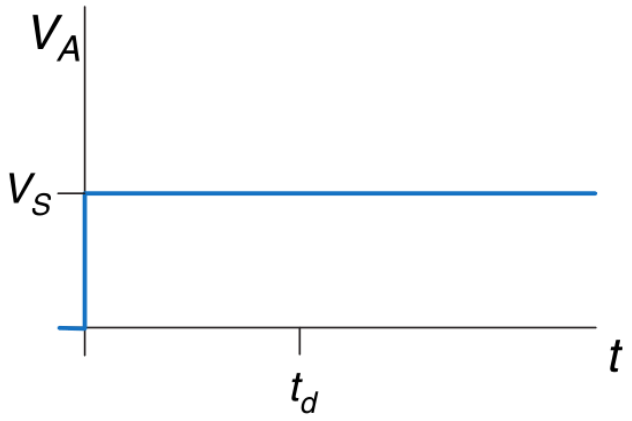
The voltage at *B* abruptly rises $0 \rightarrow V_S$

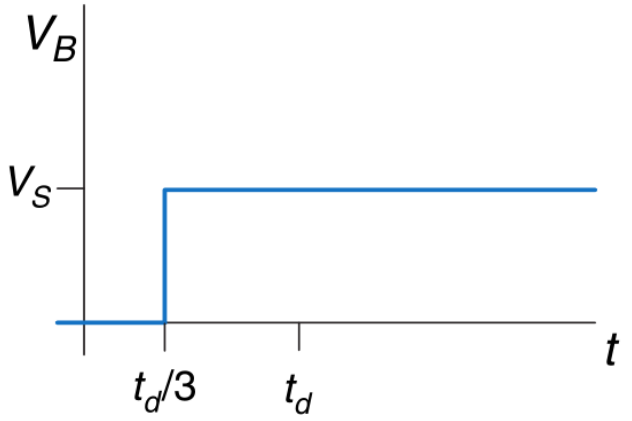
At $t = t_d$ the wave reaches *C*

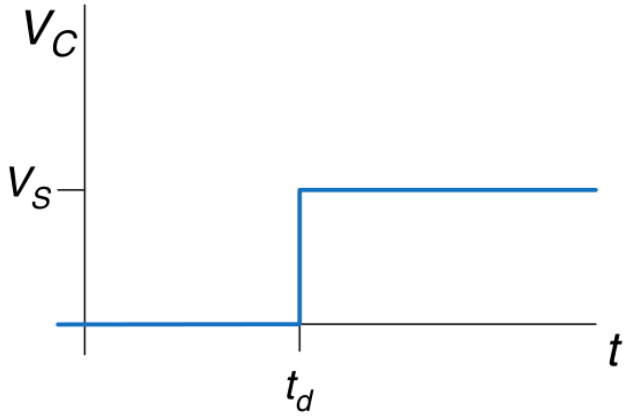
All of the current, *I*, flows into the load, Z_L, producing a voltage across the load

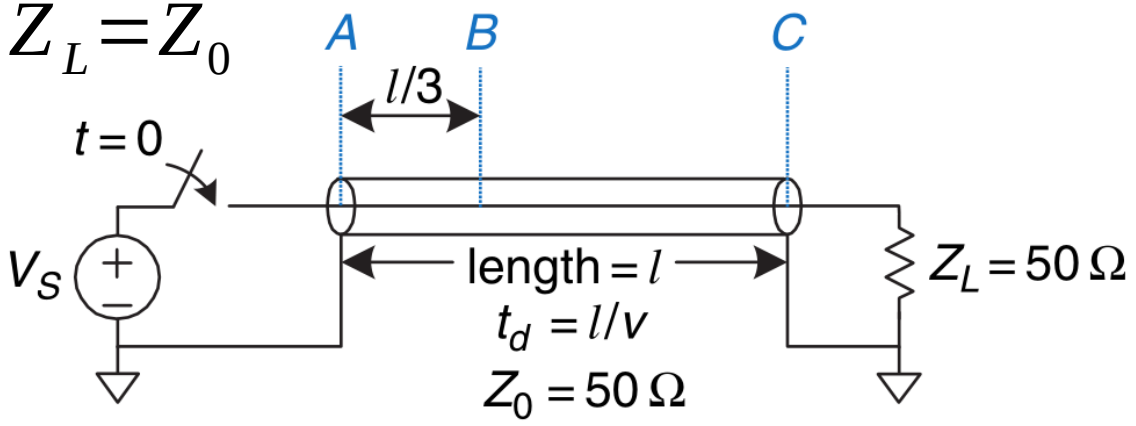
$$Z_L I = Z_L (V_S / Z_0) = V_S$$

Matched termination: $Z_L = Z_0$









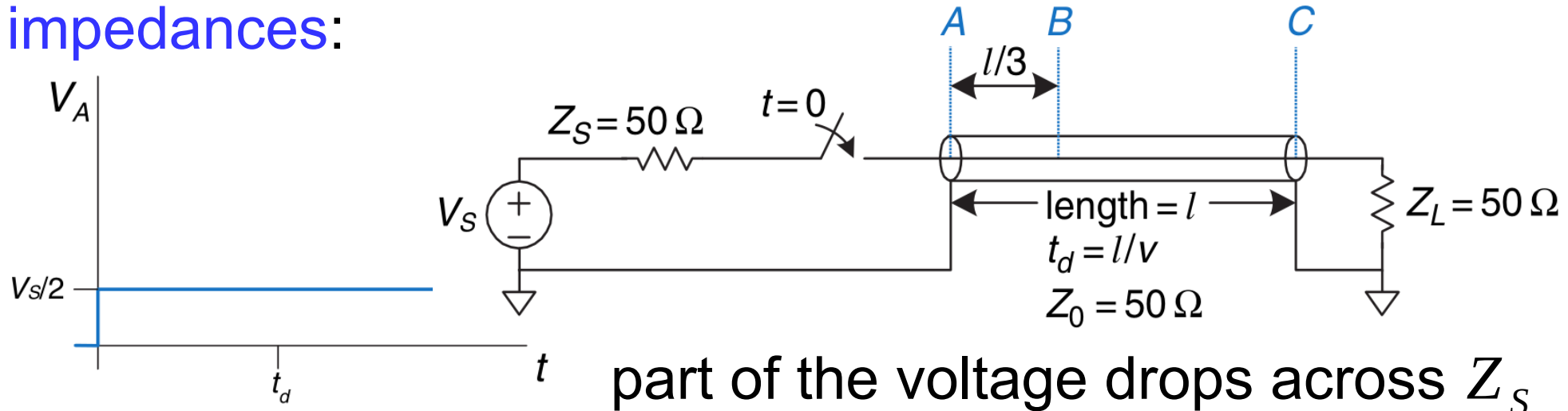
At $t = t_d$ the voltage at C is V_S

This voltage is consistent with the wave flowing along the transmission line.

The wave is absorbed by the load impedance, and the transmission line reaches its steady state.

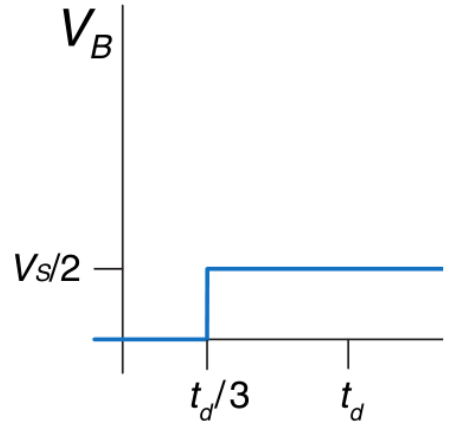
The transmission line starts behaving like an equipotential wire

Transmission line with matched source and load impedances:

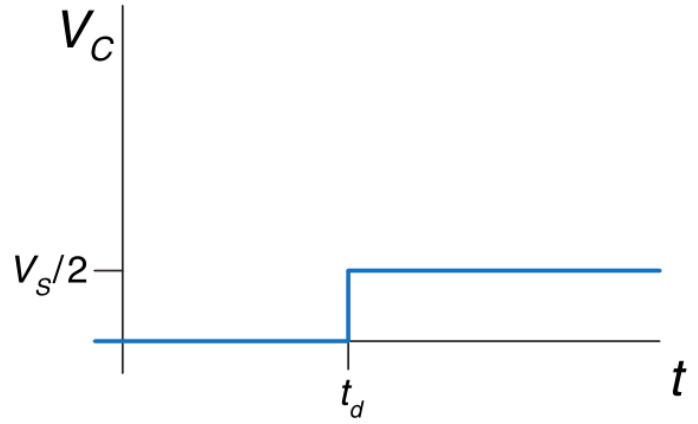


part of the voltage drops across Z_S

Z_L cannot influence the behavior of the line at $t<t_d$



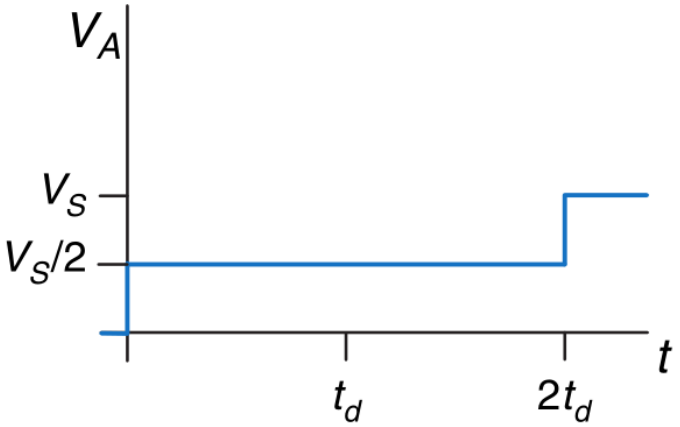
The incident voltage flowing down the line is $V = V_S\left(\frac{Z_0}{Z_0+Z_S}\right) = \frac{V_S}{2}$

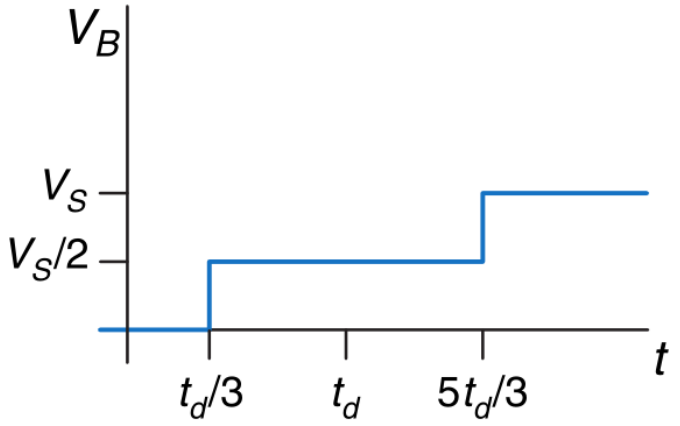


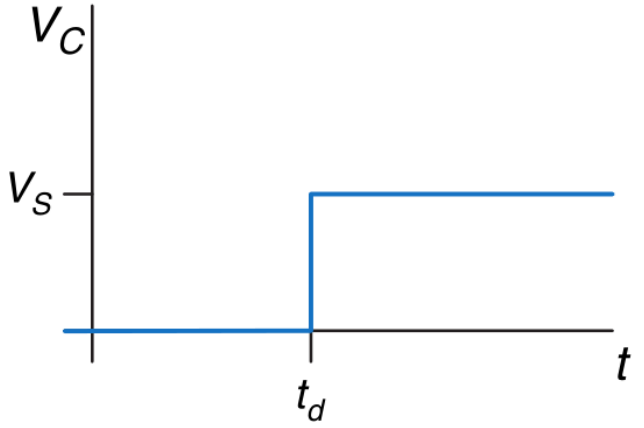
All of the current is absorbed by the load impedance Z_L

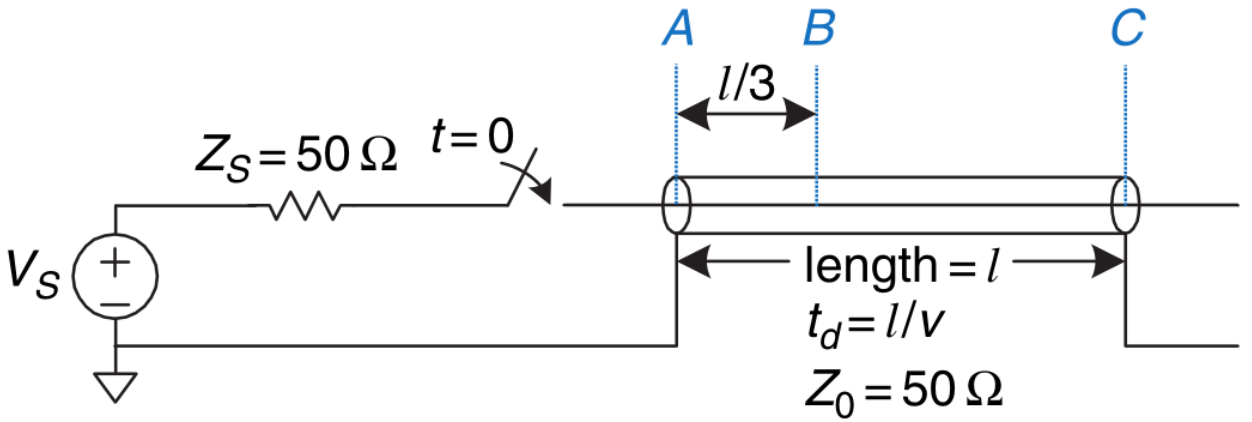
The circuit enters steady-state at $t = t_d$ with $V = V_S/2$

Open termination









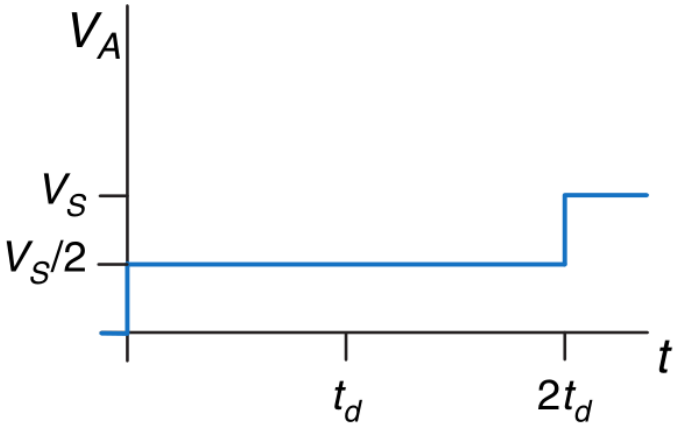
When $Z_L \neq Z_0$

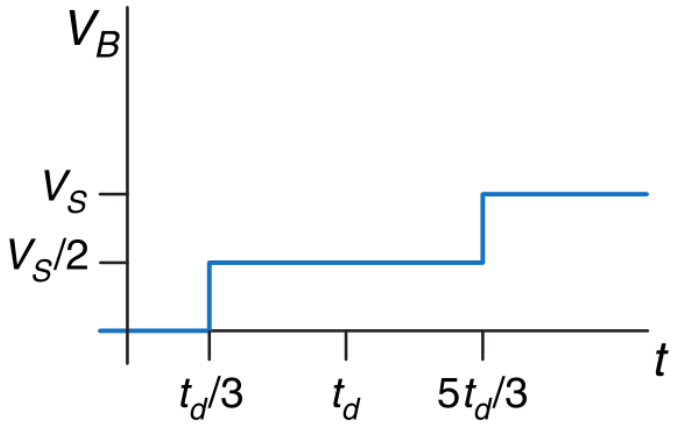
the termination cannot absorb all of the current

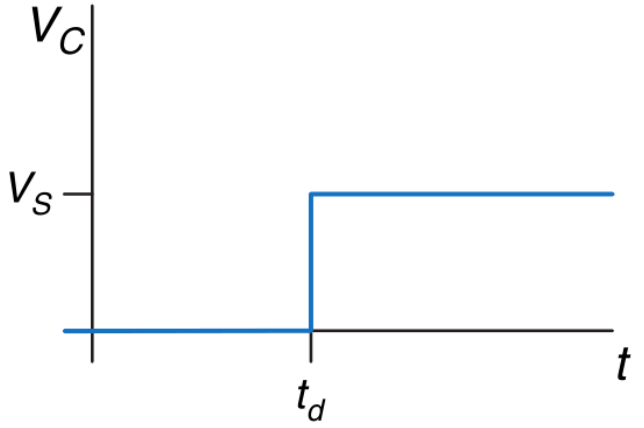
some of the wave must be reflected

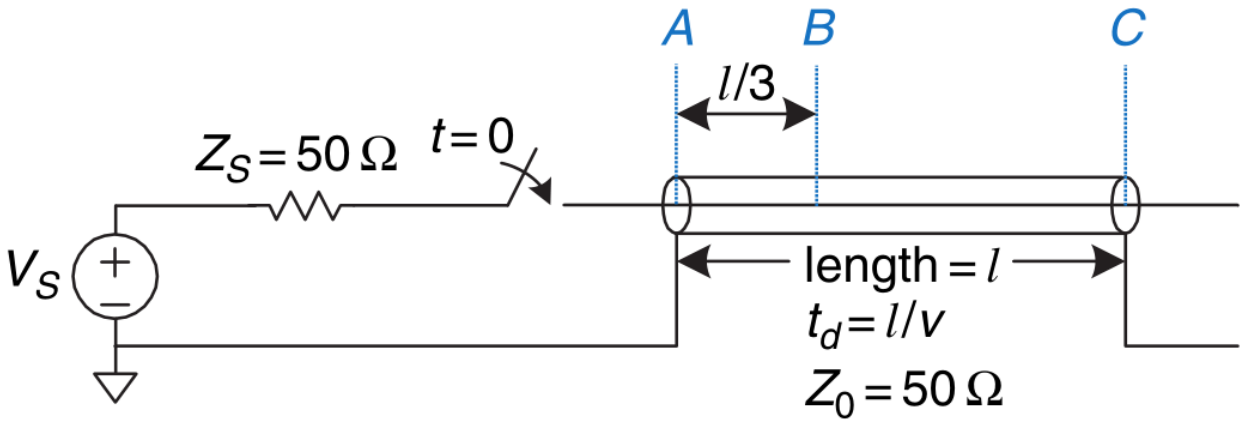
No current can flow through an open termination, so the current at point C must always be 0.

Open termination









At $t = 0$ the switch closes, and a wave of voltage $V = V_S/2$ begins propagating

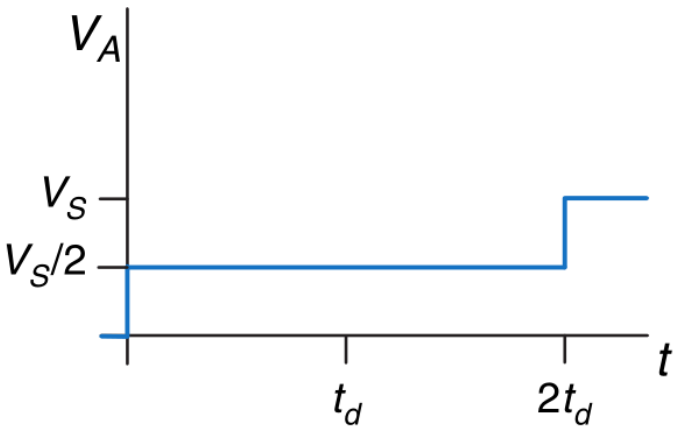
When the wave reaches *C*, it reflects back toward the source.

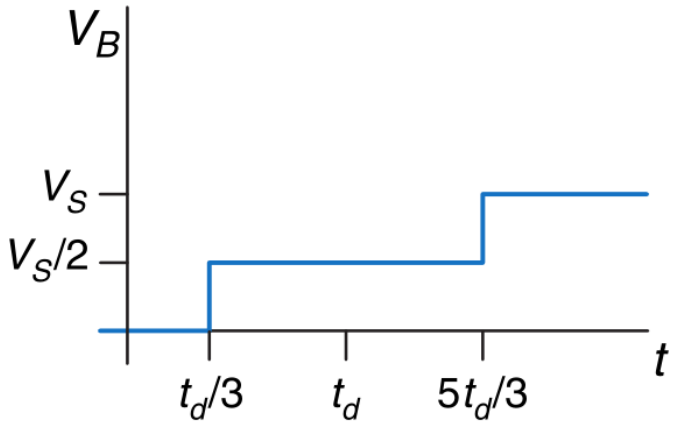
The reflected wave has voltage

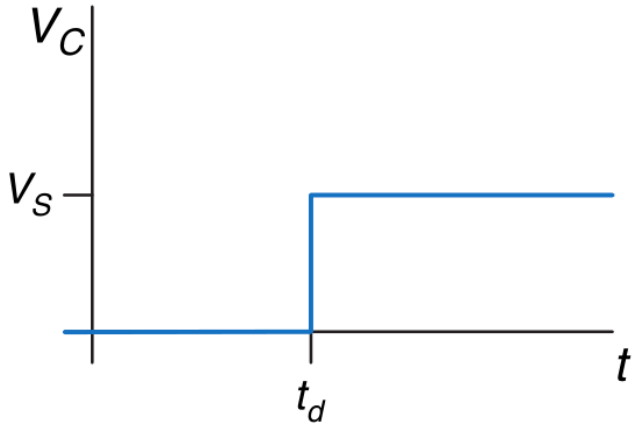
$$V = V_S/2$$

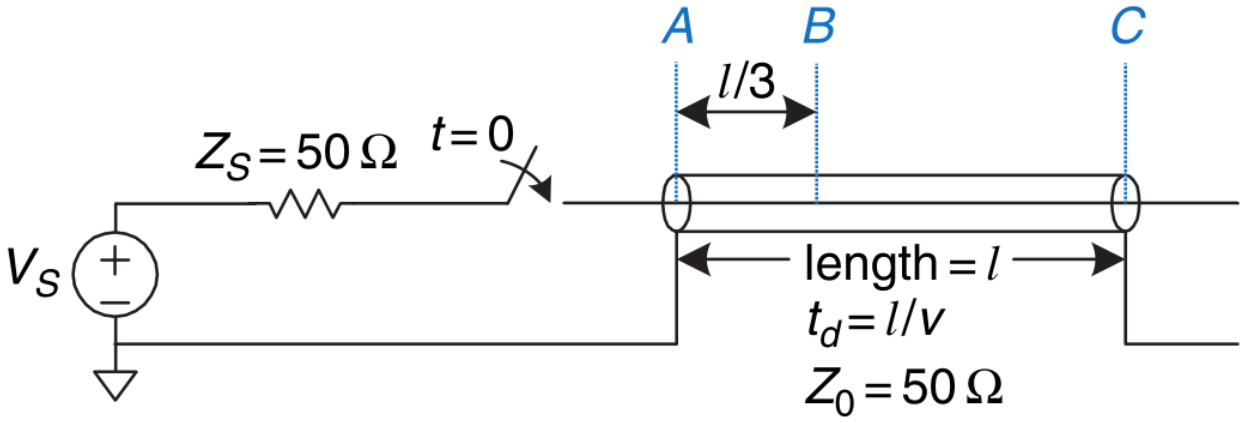
because the open termination reflects the entire wave

Open termination









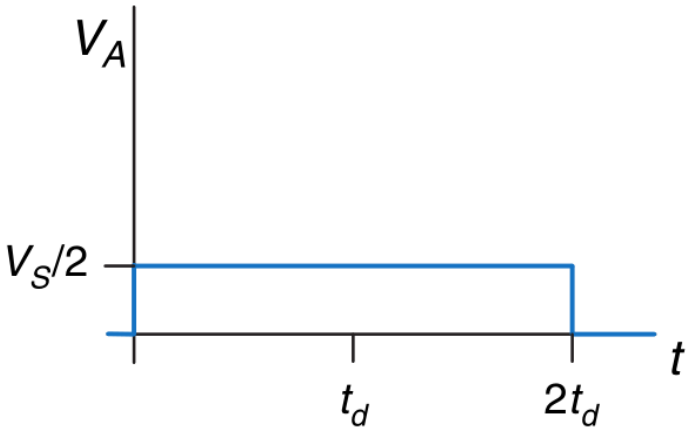
The voltage at any point is the sum of the incident and reflected waves

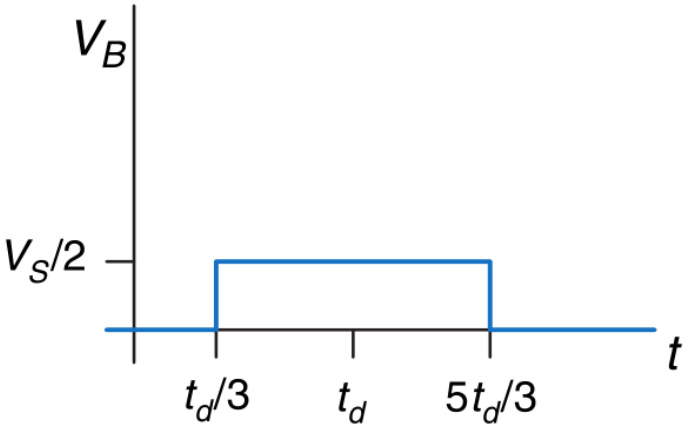
At *C* $t \geq t_d \Rightarrow V = \frac{V_S}{2} + \frac{V_S}{2} = V_S$

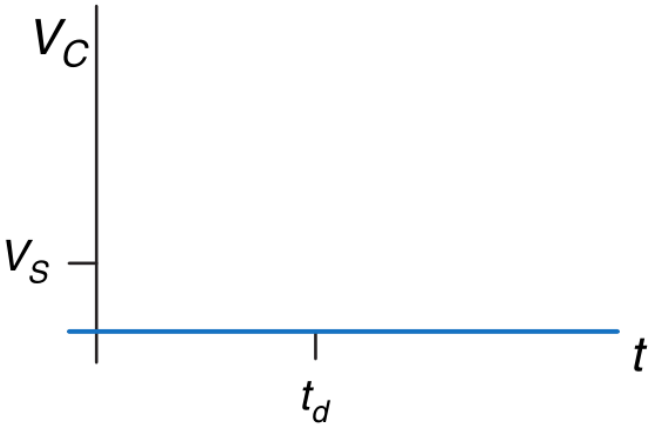
The reflected wave reaches *B* at $t \geq 5t_d/3$ and point *A* at $2t_d$

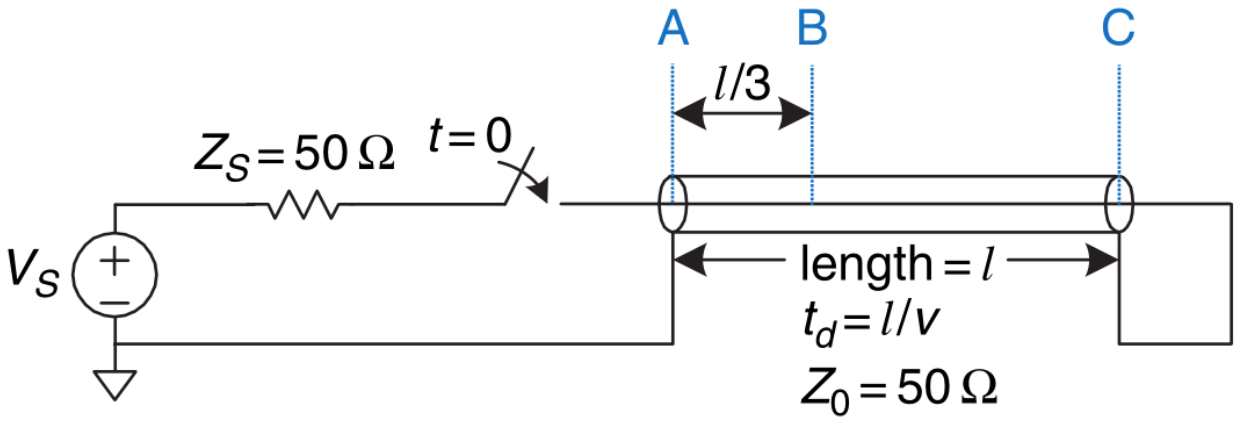
When it reaches *A*, the wave is absorbed by the source termination impedance that matches Z_0 => steady state at $2t_d$

Short termination









The voltage at *C* must always be 0

When the wave reaches *C*, it must reflect with opposite polarity.

The reflected wave, with voltage

$$V = -V_S/2$$

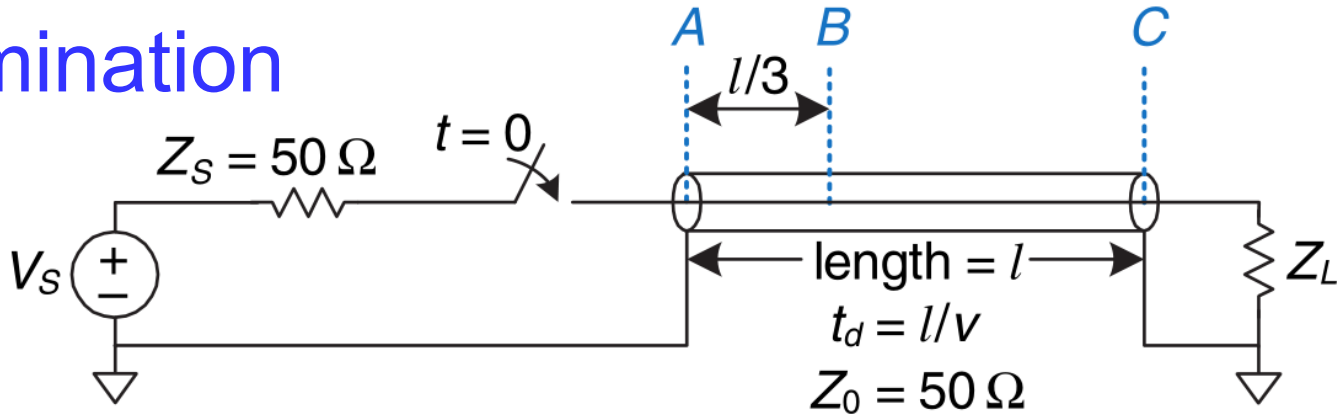
adds to the incident wave

$$t \geq t_d \Rightarrow V = \frac{V_S}{2} - \frac{V_S}{2} = 0$$

=> steady state at $2\,t_d$

Mismatched termination

$Z_L \neq Z_0$



part of the wave is absorbed and part is reflected

$$V_i = k_r V_r$$

$$k_r = \frac{Z_L - Z_0}{Z_L + Z_0}$$

V_r – reflected wave

V_i – incident wave

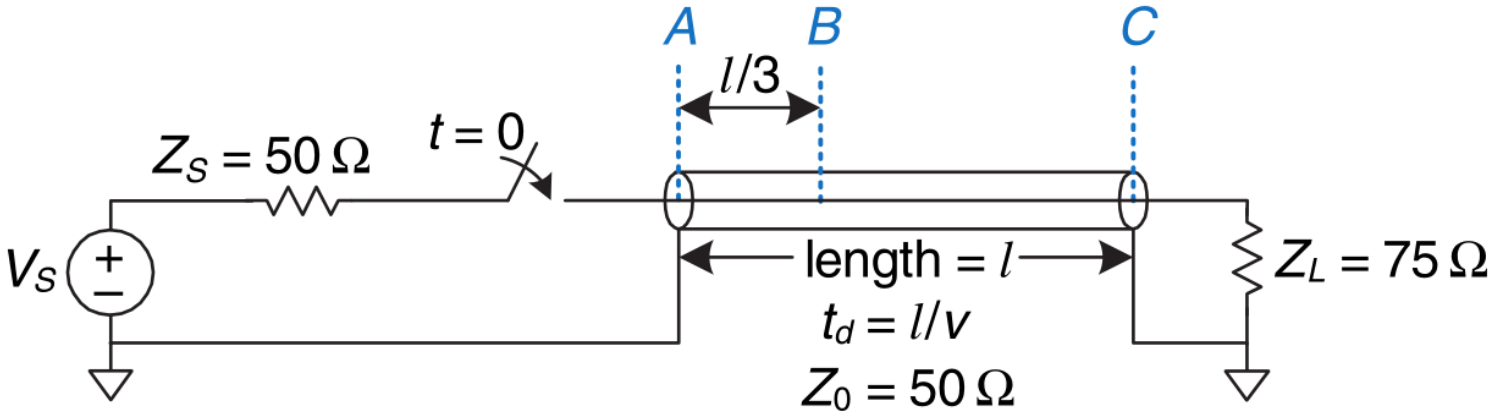
k_r – reflection coefficient

$Z_L = \infty \Rightarrow k_r = 1$ open circuit

$Z_L = 0 \Rightarrow k_r = -1$ short circuit

Example

$$k_r = \frac{1}{5}$$



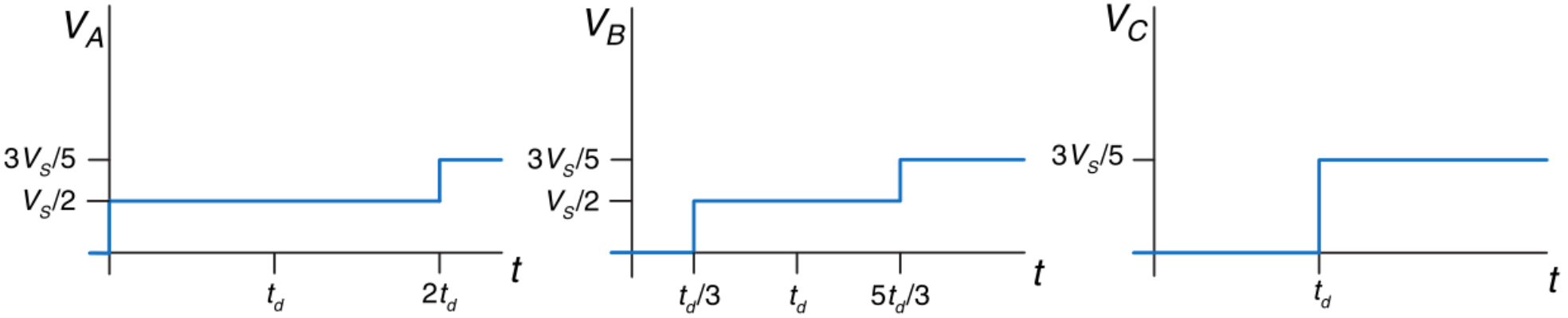
When the incident wave reaches the termination at the endof the line, 1/5 of the wave is reflected, and the remaining 4/5 flows into the load impedance.

The reflected wave has a voltage $V = \frac{1}{5}\frac{V_S}{2} = \frac{V_S}{10}$

The total voltage at point C $V = V_i + V_r = \frac{3}{5}V_S$

At $t = 2t_d$ the reflected wave reaches *A*

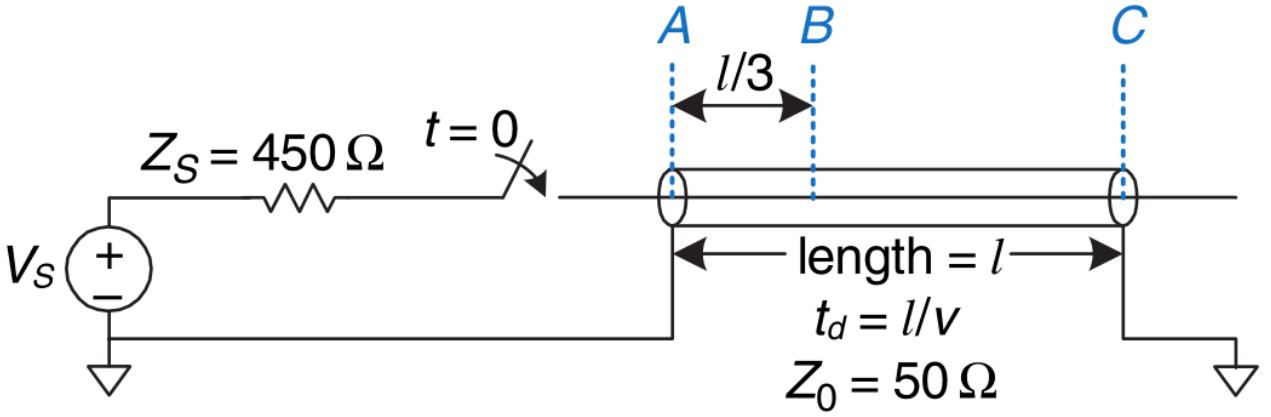
where it is absorbed by the matched termination Z_S



At $t = 2t_d$ steady state

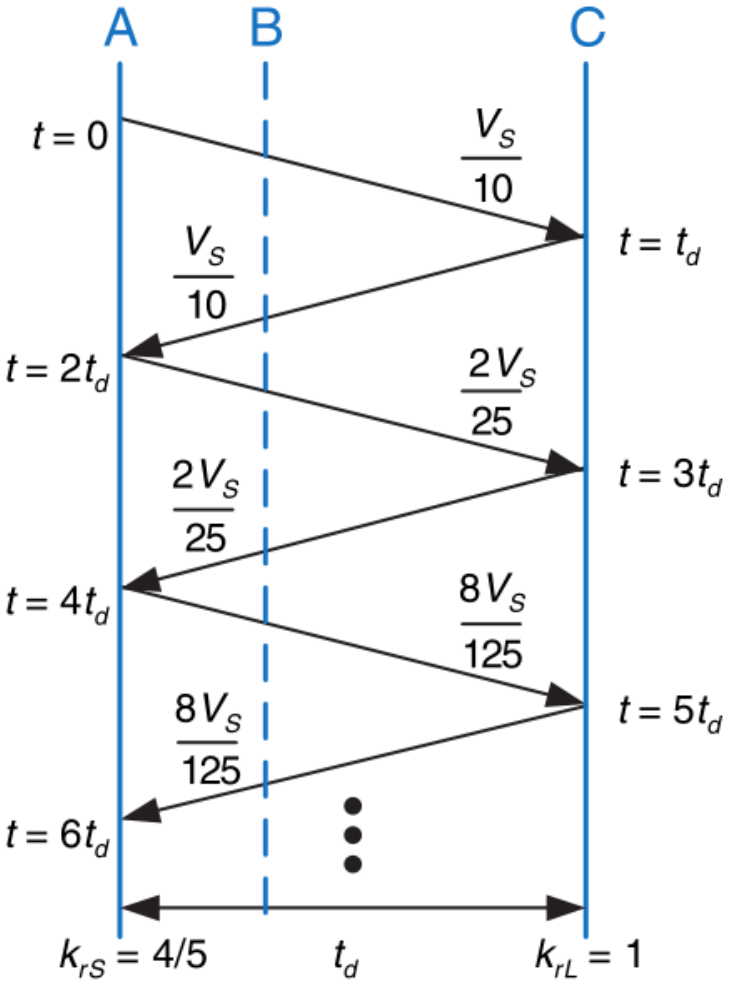
the transmission line is equivalent to an equipotential wire

Transmission line with mismatched source and load terminations



$$k_{rS} = \frac{4}{5} \qquad k_{rL} = 1$$

Waves reflect off both ends of the transmission line until a steady state is reached.



The bounce diagram:

As $t \to \infty$, the voltages approach steady state with

$$V_A = V_B = V_C = V_S$$

Transmission line models for wires are needed whenever the wire delay, t_d, is longer than 1/5 (20%) of the edge rates (rise or fall times) of a signal.

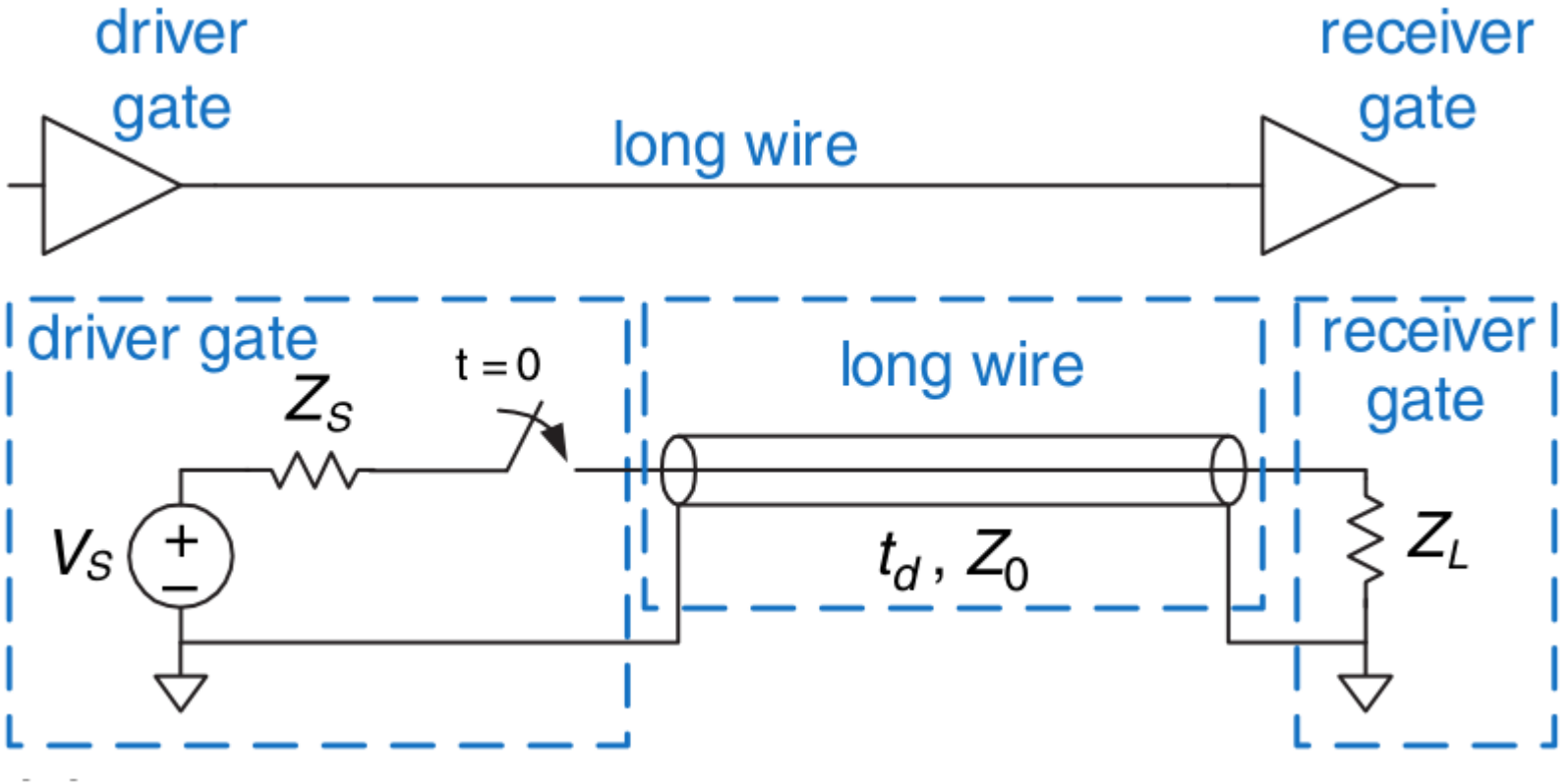
In other words, the transmission line has significant delay and noise effects on signals whose rise/fall times are less than about $5t_d$

This means that, for systems with 2ns rise/fall times, PCB traces longer than about 6 cm must be analyzed as transmission lines to accurately understand their behavior.

The driver gate cannot supply infinite current

This is modeled by Z_S

Z_S is usually small for a logic gate, but a designer may choose to add a resistor in series with the gate to raise Z_S and match the impedance of the line.



The input to the second gate is modeled as Z_L

CMOS circuits usually have little input current, so Z_L may be close to infinity.

The designer may also choose to add a resistor in parallel with the second gate, between the gate input and ground, so that Z_L matches the impedance of the line.