Fundamentals of Electrical and Electronic Engineering

400462

# Digital Electronics

Dr Ian M. Bell

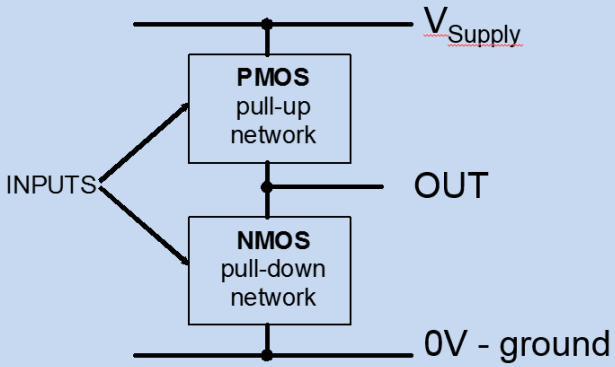
# CMOS Gate Circuits

Circuits for other gates are an extension of the form of the inverter

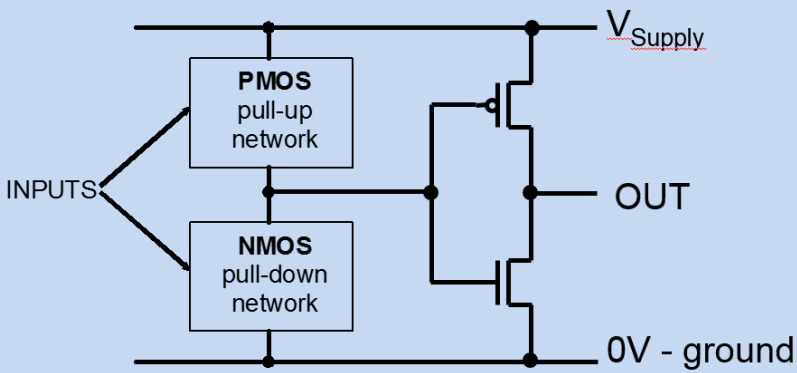
Basic gates are inverting (e.g. NAND and NOR)

An additional inverter is required for a non-inverting gates (e.g. AND)

**Inverting Gate**




**Non-Inverting Gate**

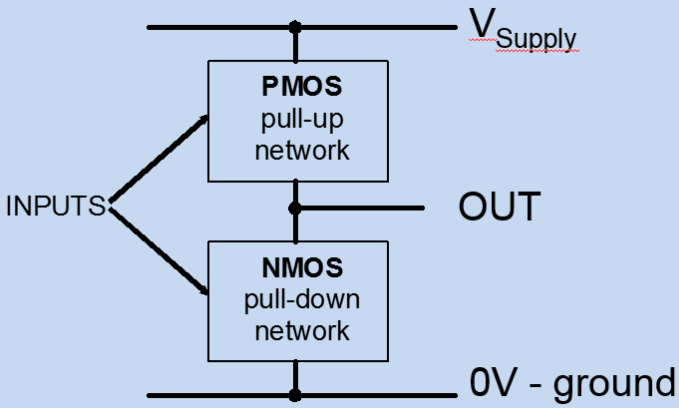


# CMOS Gate Circuits

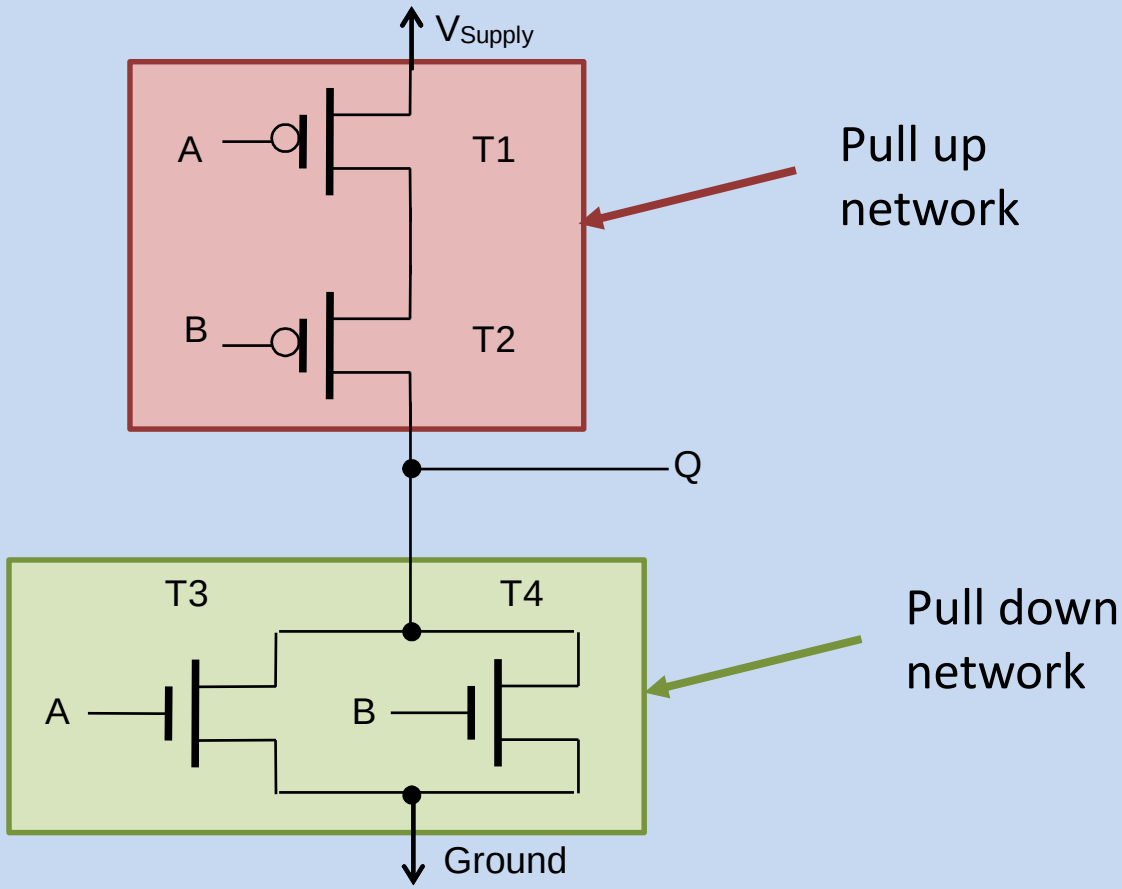
Understanding the operation of logic gates and evaluation block

- PMOS transistor: ON when its gate terminal is at logic 0
- NMOS transistor: ON when its gate terminal is at logic 1
- The P and N networks are configured to connect the output to the supply when a 1 is required and to ground when a 0 is required
- When one evaluation block is ON the other must be OFF

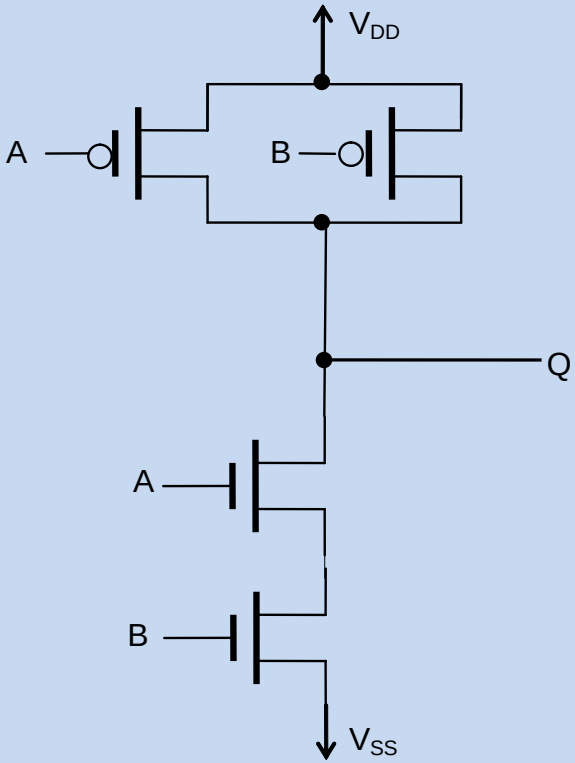
**What happens if this is not true?**



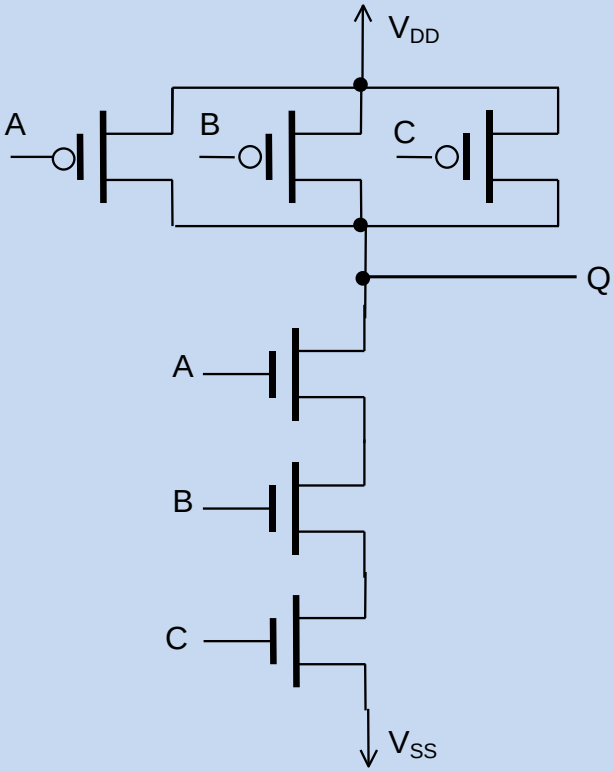
# Example Two-Input gate



# Example CMOS Gate Circuits



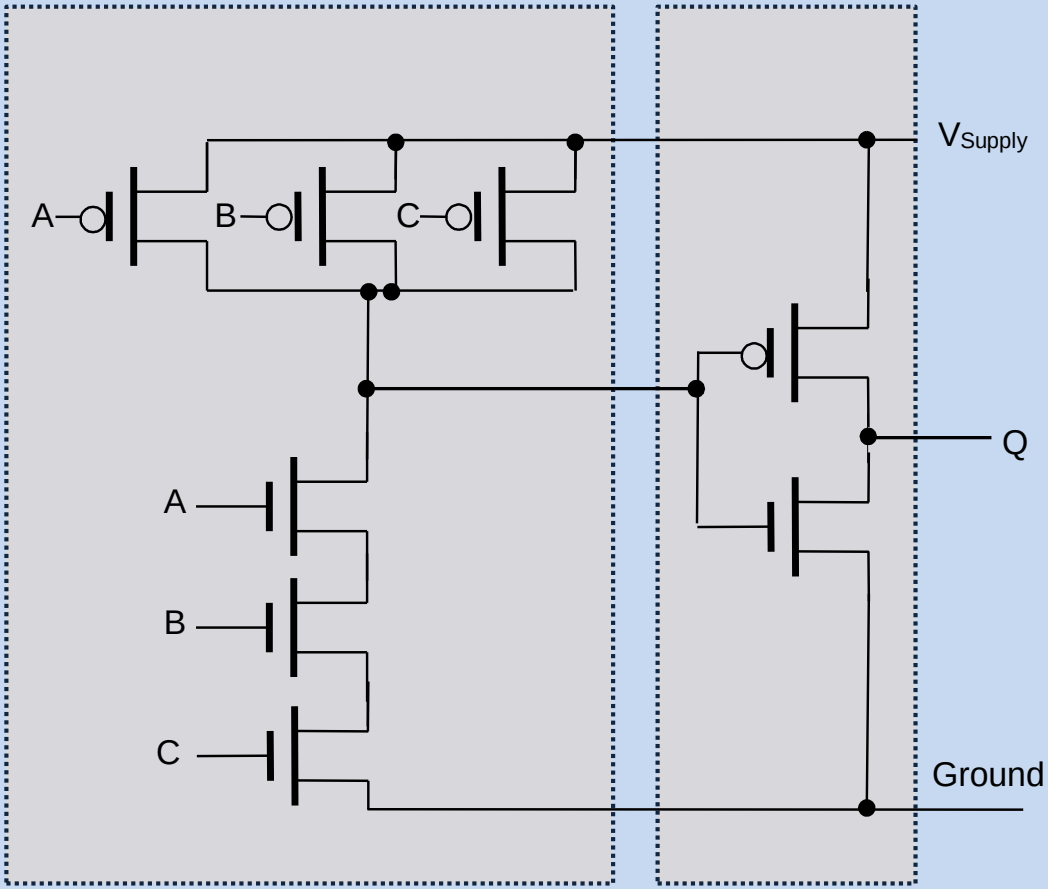

*2-Input NAND*

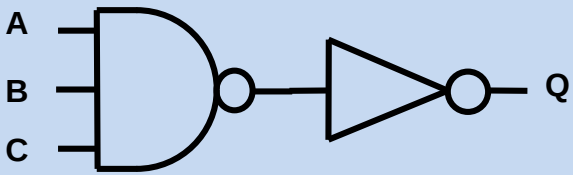



*3-Input NAND*

# Non-inverting CMOS Logic Gates

Non inverting gates (AND and OR) are formed from the corresponding inverting gate (NAND and NOR) plus an inverter (NOT gate)




*3-Input AND*

# Complex Gates

CMOS logic gates circuits may be devised to evaluate combinational logic functions directly (as a transistor circuit), rather than using a network of basic gates (NAND, NOR, and NOT).

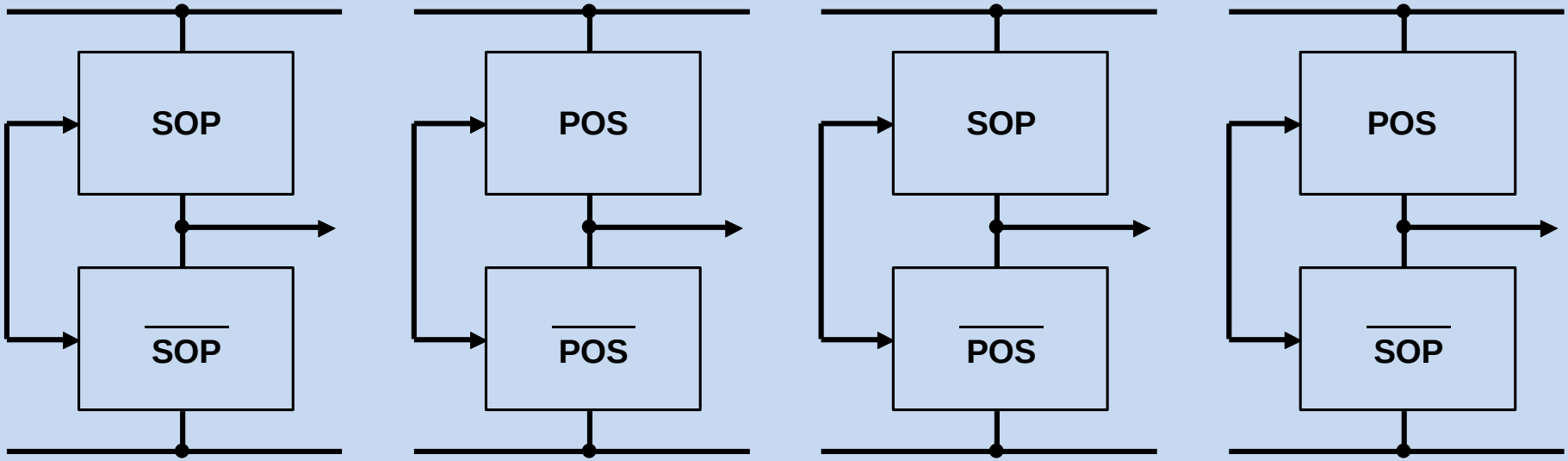
For example, function such as the following can be implemented directly with transistor switches

$$Q = \overline{AB + C} = \left(\overline{A} + \overline{B}\right) \cdot \overline{C}$$

Gates like this are sometimes referred to as *complex gates*.

# Complex Gates

- In complex gates the pull-up and pull-down networks may be realised using both Sum Of Products (**SOP**) and Product Of Sums (**POS**) networks and their complements.
- This leads to a number of possible versions of some complex gates.
- Note that the NMOS network always implements the complement of the logic function because its job is to produce a 0 at the output and logic function defines the condition for an output being 1



# Complex Gate Example

$$f = \overline{AB + C}$$

| C AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |

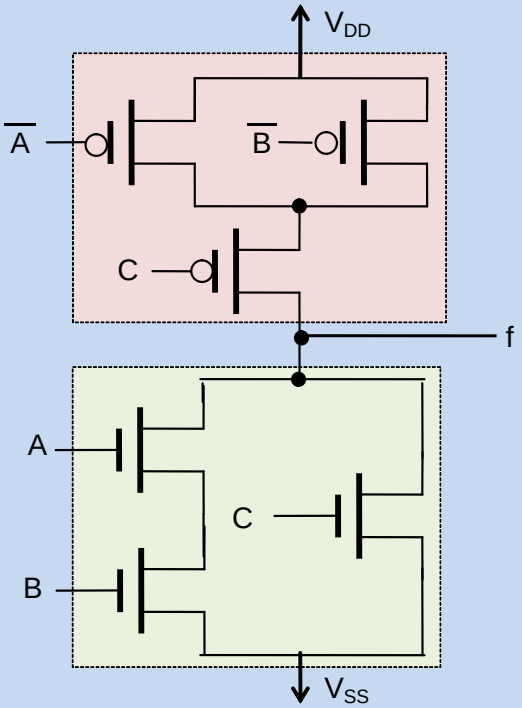
SOP $\quad f = \overline{A}\,\overline{C} + \overline{B}\,\overline{C}$

$\overline{\text{SOP}}$ $\quad \overline{f} = (A + C)(B + C)$

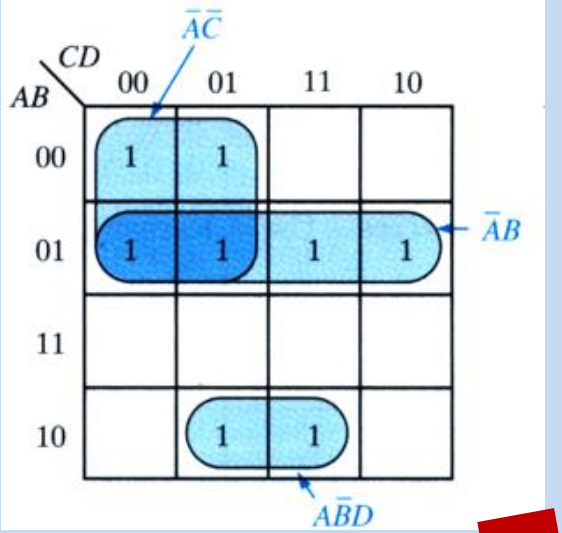
POS $\quad f = \left(\overline{A} + \overline{B}\right)\overline{C}$

$\overline{\text{POS}}$ $\quad \overline{f} = AB + C$

EXAMPLE $\text{POS} / \overline{\text{POS}}$

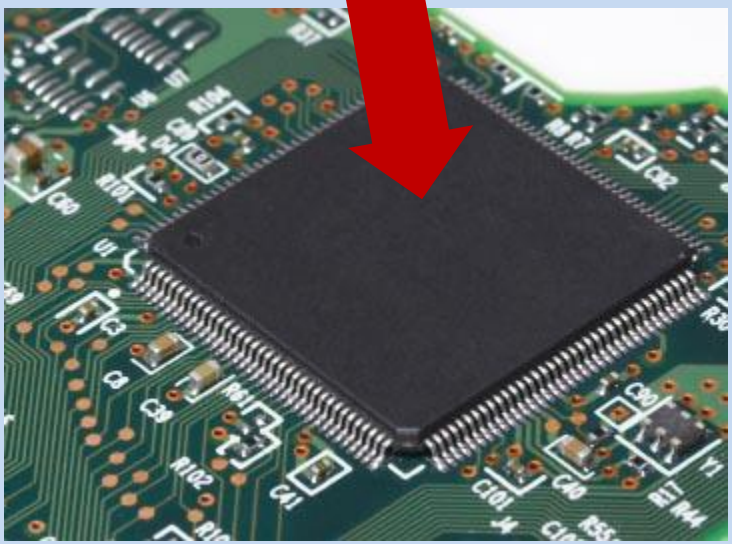


# In reality there is more to worry about than just 1s and 0s




**From abstract logic design to physical circuit**

**?**



clemson.edu

***Implementation Technology***

- Choice of technologies for implementing a digital system
- **Internal circuitry of logic gates**

***Voltage & Current***

- Logic Voltages
- Noise and errors
- Supply voltages
- Power Consumption

***Time***

- Speed of operation
- Timing errors

# Logic in the Real World - Logic Levels

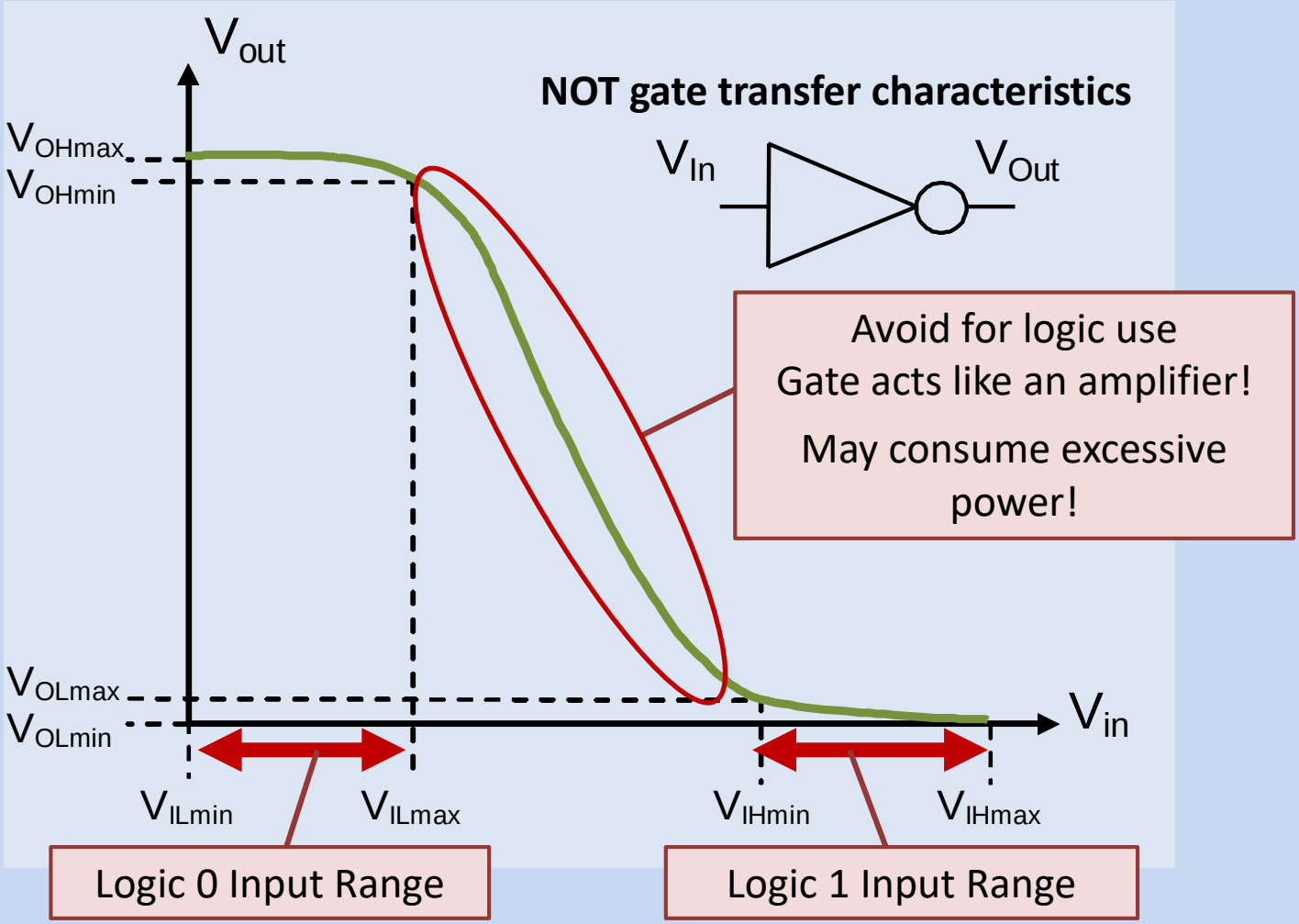
How do we relate Boolean 1's and 0's to voltages in real circuits?

**0.000 V** = logic 0 **+5.000 V** = logic 1

**Will this work?**

# The Logic Gate as an Analogue Circuit

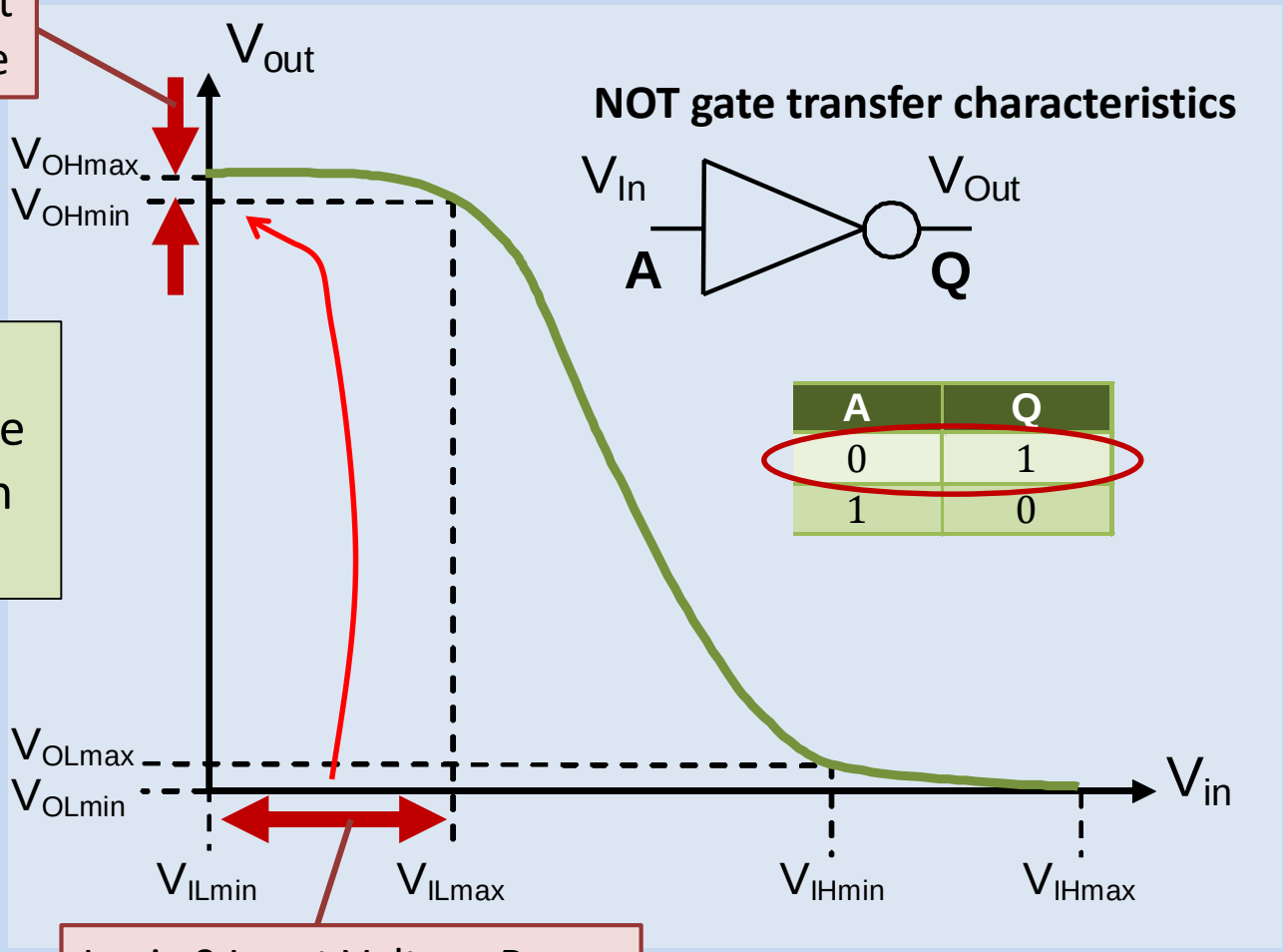
We can plot the input-output voltage characteristics for a gate just as we can for an analogue circuit



# “Analogue” NOT gate: 0 in 1 out

Logic 1 Output Voltage Range

✓ Note: output range smaller than input range

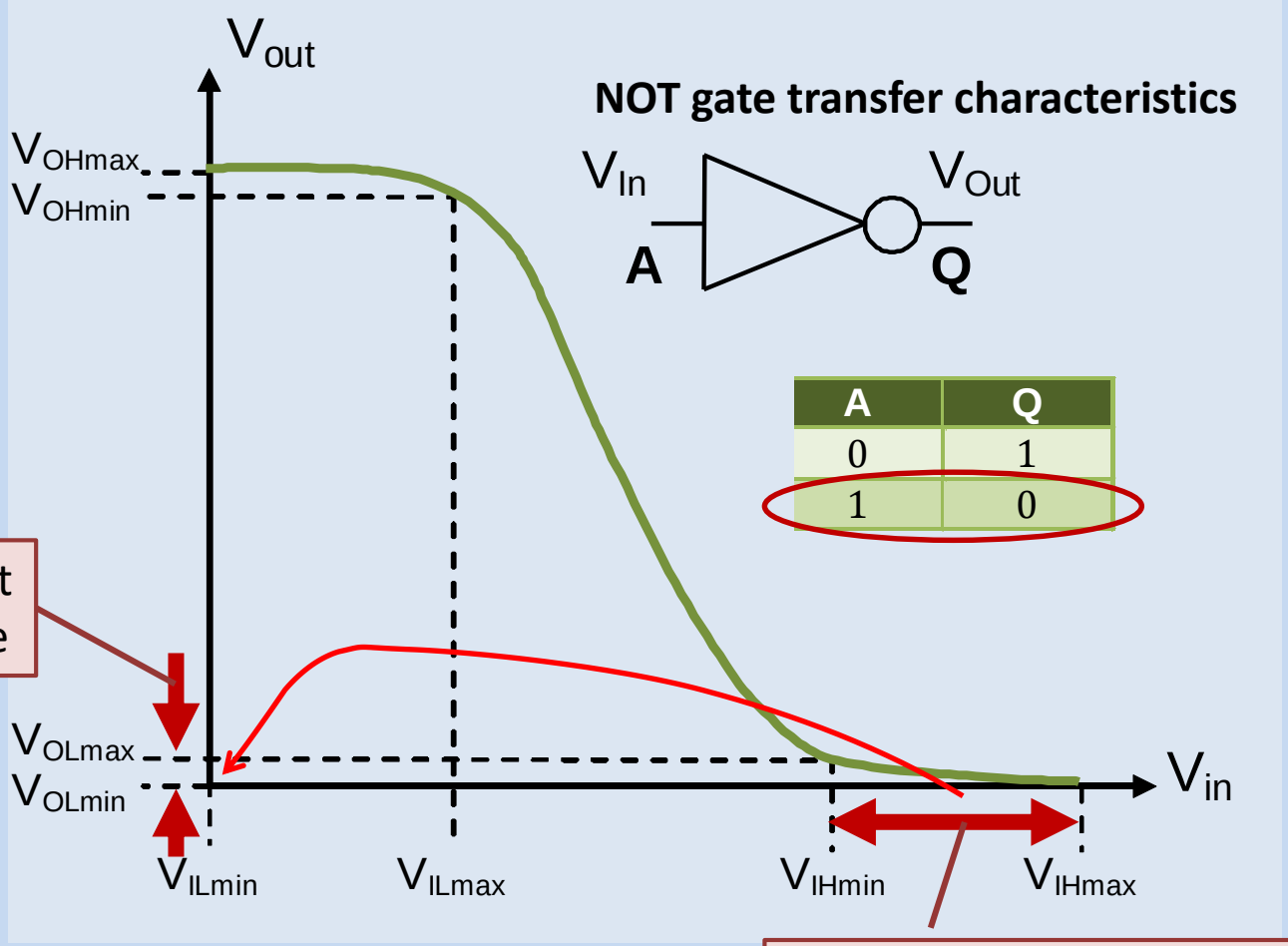



| A | Q |
|---|---|
| 0 | 1 |
| 1 | 0 |

Logic 0 Input Voltage Range

# “Analogue” NOT gate: 1 in 0 out

✓ Note: output range smaller than input range



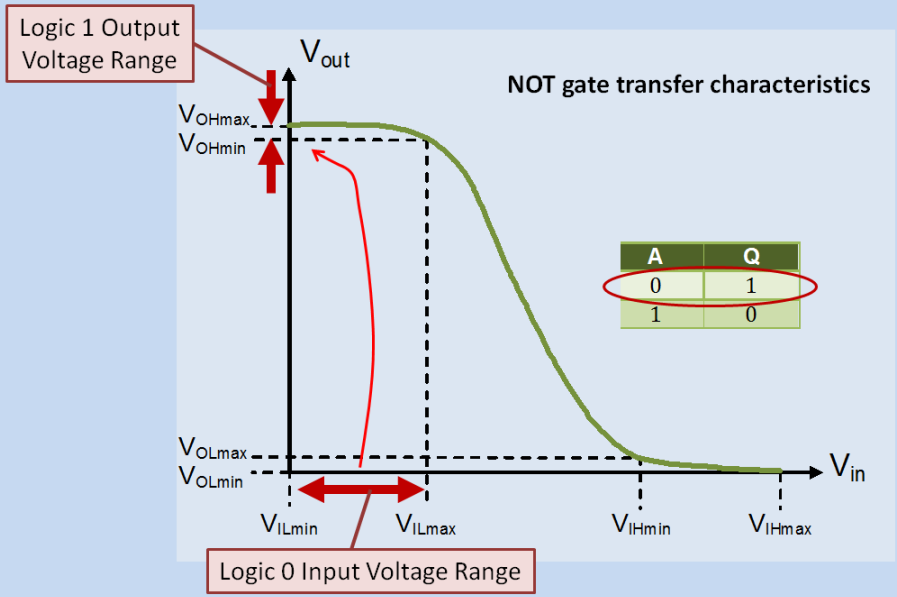

| A | Q |
|---|---|
| 0 | 1 |
| 1 | 0 |

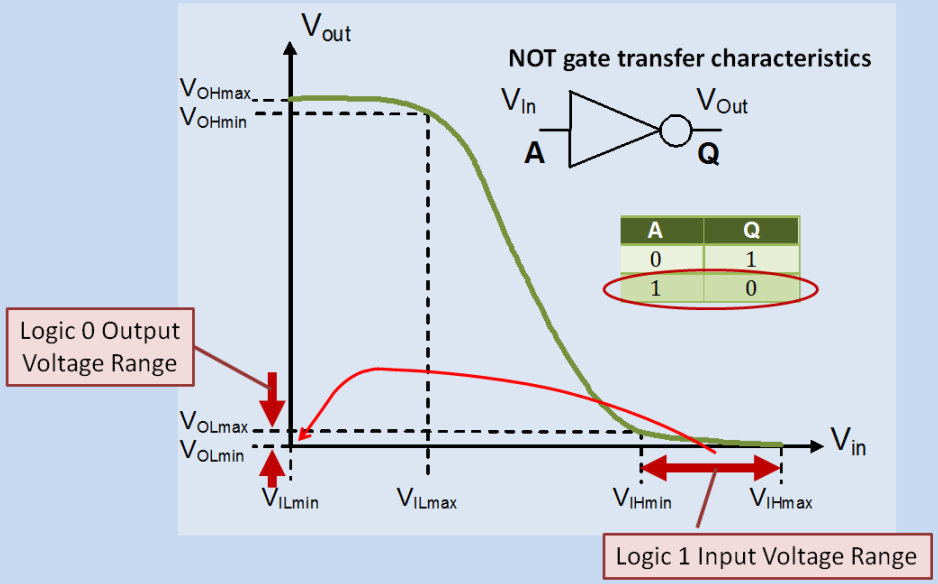
Logic 0 Output Voltage Range

Logic 1 Input Voltage Range

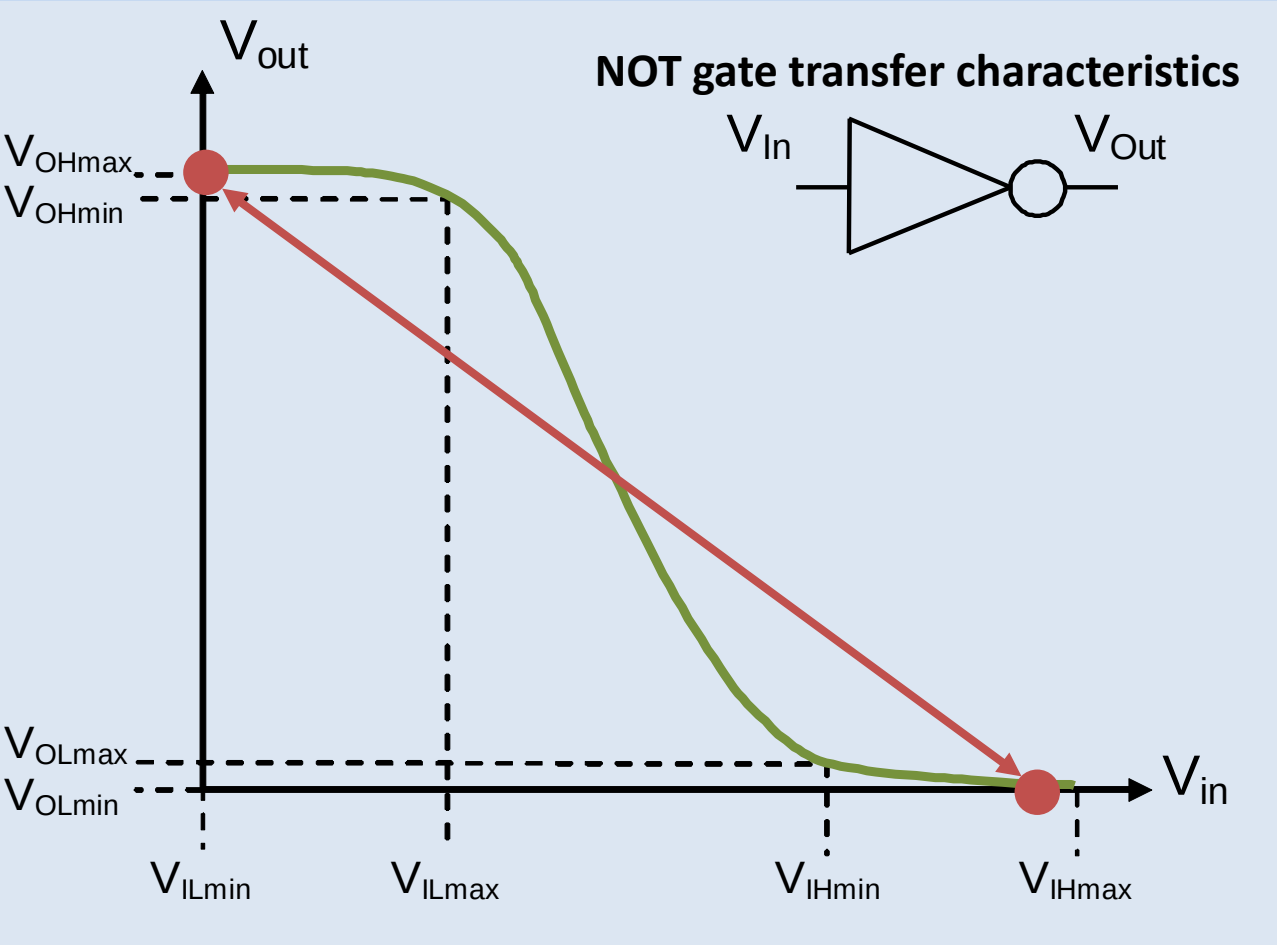
# Logic Voltage Levels: Summary

- Logic gates will accept a given range of inputs as 1 or 0
- Guaranteed to produce smaller ranges of possible output voltages
- Gate tends to restore the voltage towards the ideal



# The Logic Gate: from Analogue to Digital Representation



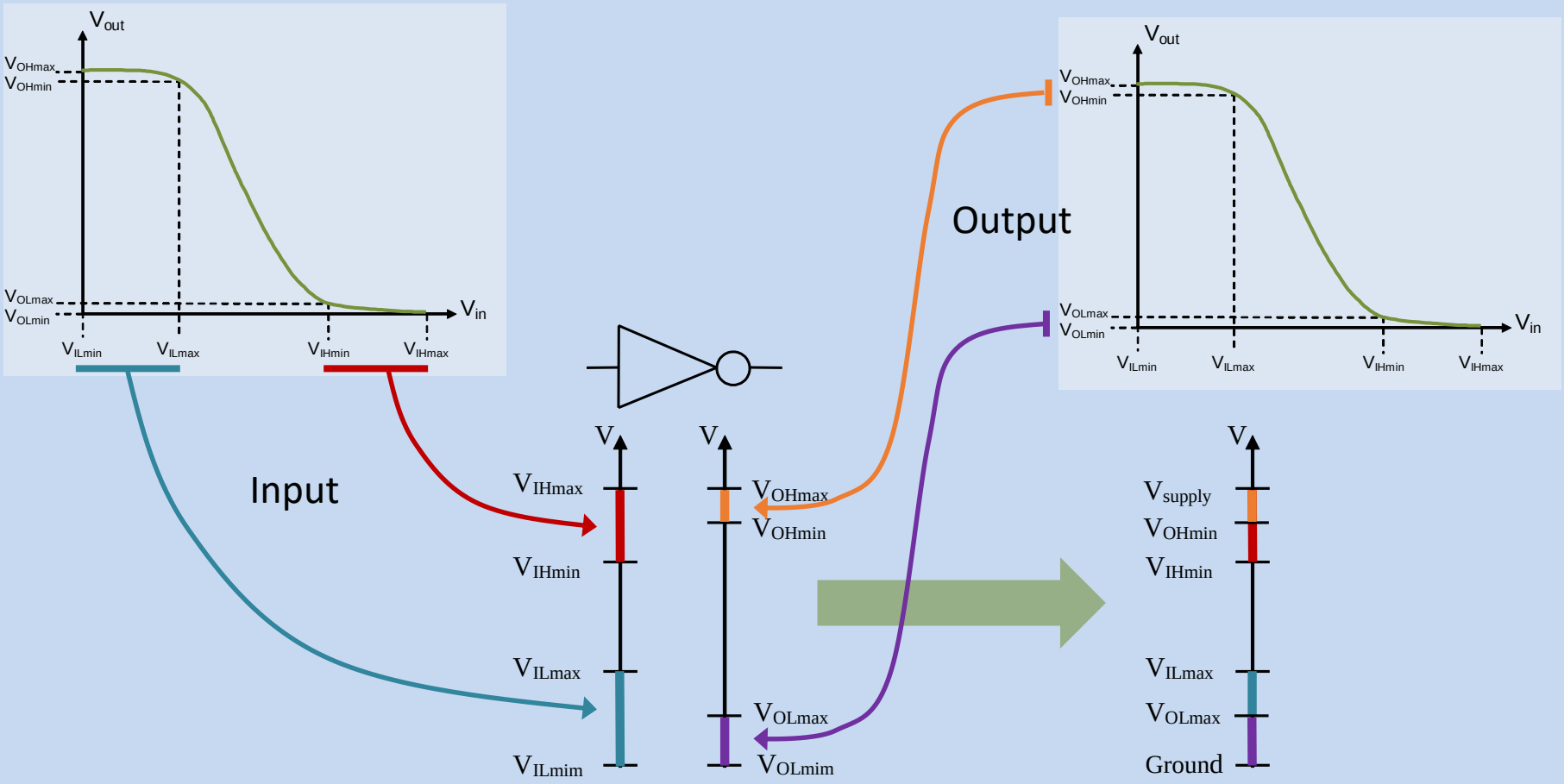

In logic circuits the gate switches between acceptable logic input and output levels and is not used in other parts of the curve

We can use the abstract Boolean description

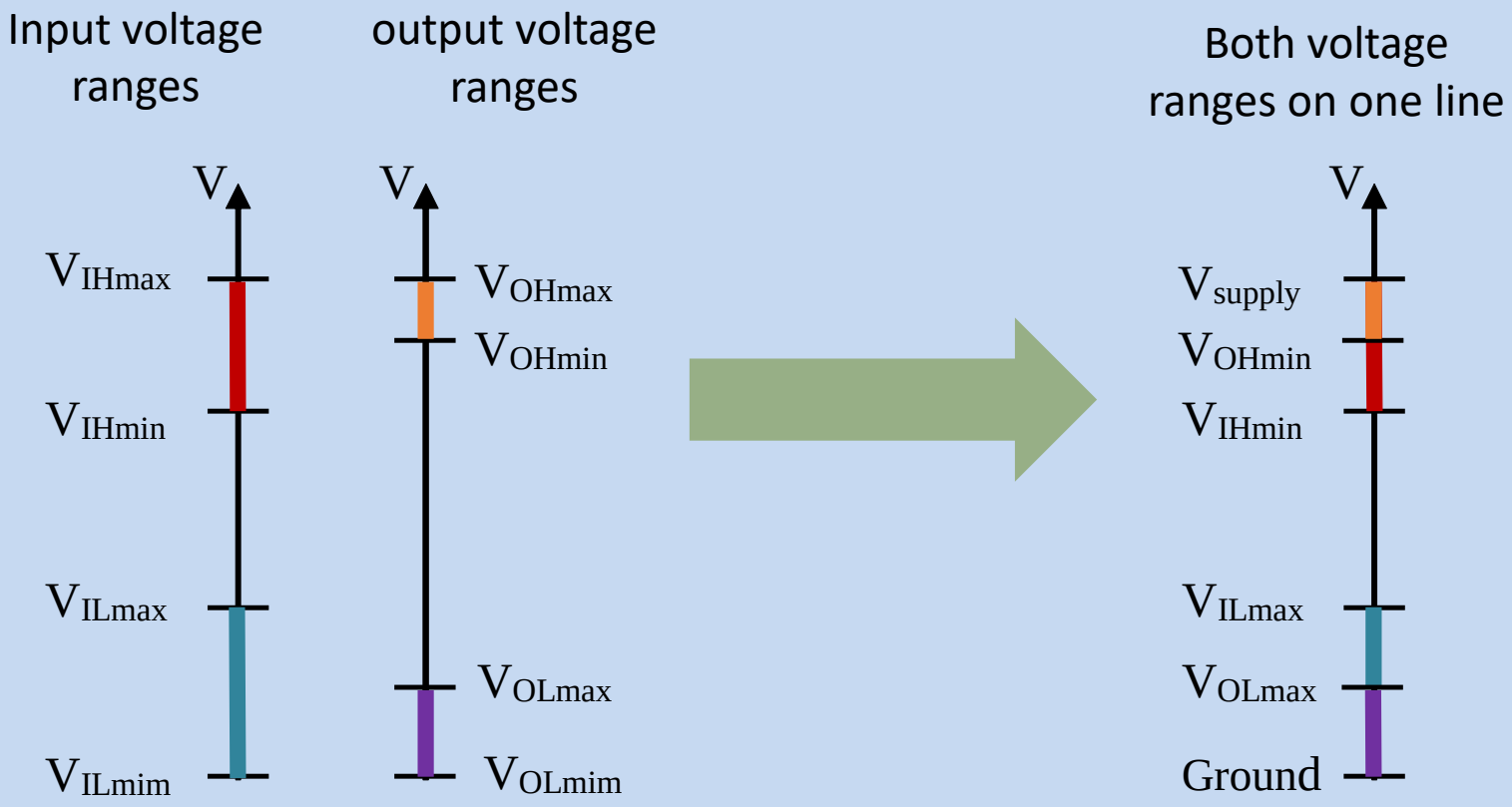
| IN | OUT |
|---|---|
| 0 | 1 |
| 1 | 0 |

# Logic Switching Standards

- We only need the voltage range information – which we can visualise this on a line rather than using the full transfer curve
- These voltages are the same for devices of a given technology/logic family
- Referred to as the logic switching standard or characteristic



# Diagram Representing Logic Switching Standards



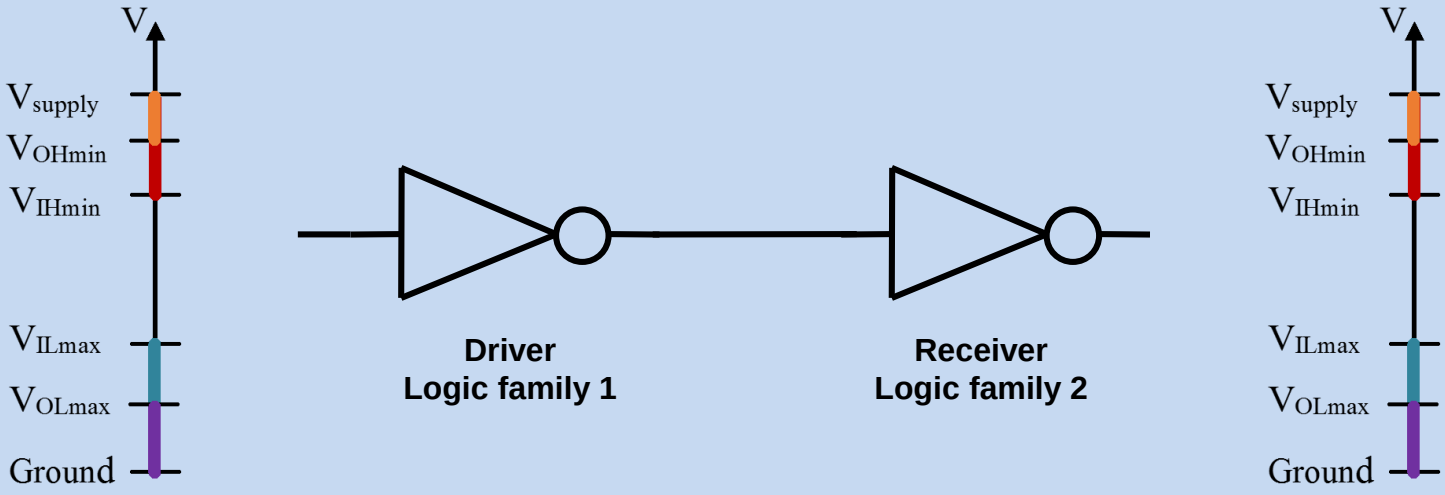

Larger view of diagrams from previous slide

# We can design with Boolean 0s and 1s …
# But we cannot always forget about Logic Voltage Levels…

- … Remember there are many different logic families – these operate at different supply voltage and logic voltages
- **What happens if we need to connect two ICs from different families together?**
- Switching standard indicates compatibility (or otherwise) of different logic technologies/families
- If devices are compatible they can be directly connected together
- If they are not compatible a logic level translator circuit will be required

| Code | Logic Family | Technology |
|---|---|---|
| **ABT** | Advanced BiCMOS Technology | BiCMOS |
| **ABTE** | Advanced BiCMOS Technology / Enhanced Transceiver Logic | BiCMOS |
| **AC** | Advanced CMOS Logic | CMOS |
| **ACT** | Advanced CMOS Logic | CMOS |
| **AHC** | Advanced High-Speed CMOS | CMOS |
| **AHCT** | Advanced High-Speed CMOS | CMOS |
| **ALB** | Advanced Low-Voltage BiCMOS | BiCMOS |
| **ALS** | Advanced Low-Power Schottky Logic | BiPolar |
| **ALVC** | Advanced Low-Voltage CMOS Technology | CMOS |
| **ALVT** | Advanced Low-Voltage CMOS Technology | BiCMOS |
| **AS** | Advanced Schottky Logic | BiPolar |
| **AUC** | Advanced Ultra-Low-Voltage CMOS Logic | CMOS |
| **AUP** | Advanced Ultra-Low-Power CMOS Logic | CMOS |
| **AVC** | Advanced Very-Low-Voltage CMOS Logic | CMOS |
| **BCT** | BiCMOS Technology | BiCMOS |
| **CB3Q** | Low-Voltage, High-Bandwidth Bus Switch Technology | CMOS |
| **CB3T** | Low-Voltage, Translator Bus Switch Technology | CMOS |
| **CBT** | Crossbar Technology | CMOS |
| **CBT-C** | CBT with Undershoot Protection | CMOS |
| **CBTLV** | Low-Voltage Crossbar Technology | CMOS |
| **CD4000** | CMOS Logic | CMOS |
| **F** | Fast Logic | BiPolar |
| **FB** | Backplane Transceiver Logic | BiCMOS |
| **FCT** | Fast CMOS Technology | CMOS |
| **GTL** | Gunning Transceiver Logic | BiCMOS |
| **GTLP** | Gunning Transceiver Logic Plus | CMOS |
| **HC** | High-Speed CMOS Logic | CMOS |
| **HCT** | High-Speed CMOS Logic | CMOS |
| **HSTL** | High-Speed Transceiver Logic | BiCMOS |
| **JTAG** | JTAG Boundary Scan Support | BiCMOS |
| **LS** | Low-Power Schottky Logic | BiPolar |
| **LV-A** | Low-Voltage CMOS Technology | CMOS |
| **LV-AT** | Low-Voltage CMOS Technology | CMOS |
| **LVC** | Low-Voltage CMOS Technology | CMOS |
| **LVT** | Low-Voltage BiCMOS Technology | BiCMOS |
| **PCA** | Inter Integrated Circuit | CMOS |
| **PCF** | Inter Integrated Circuit | CMOS |
| **S** | Schottky Logic | BiPolar |
| **SSTL** | Stub Series Terminated Logic | BiCMOS |
| **SSTV** | Stub Series Terminated Low-Voltage Logic | CMOS |
| **TCA** | Inter Integrated Circuit | CMOS |
| **TS** | TI Switch | CMOS |
| **TTL** | Transistor-Transistor Logic | BiPolar |
| **TVC** | Translation Voltage Clamp | CMOS |
| **VME** | VME Bus Products | CMOS |

# Connecting Different Logic Families



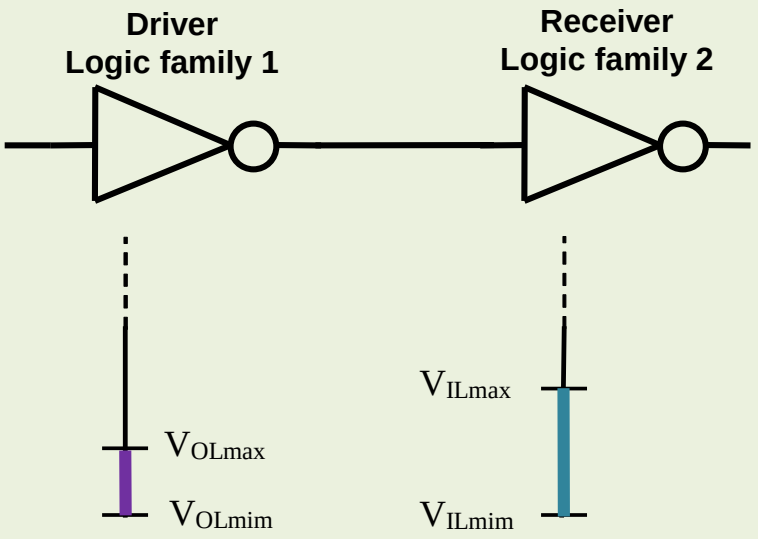

## Compatibility Checks

- Is $V_{OHmin}$ of the driver higher than $V_{IHmin}$ of the receiver?
- Is $V_{OLmax}$ of the driver lower than $V_{ILmax}$ of the receiver?
- If the driver's supply voltage is greater than the supply voltage of the receiver then receiver's input must be **tolerant** of that voltage
  - For example 3.3V supply with "5V tolerant" inputs

# Connecting Different Logic Families – Example logic 0 Compatibility

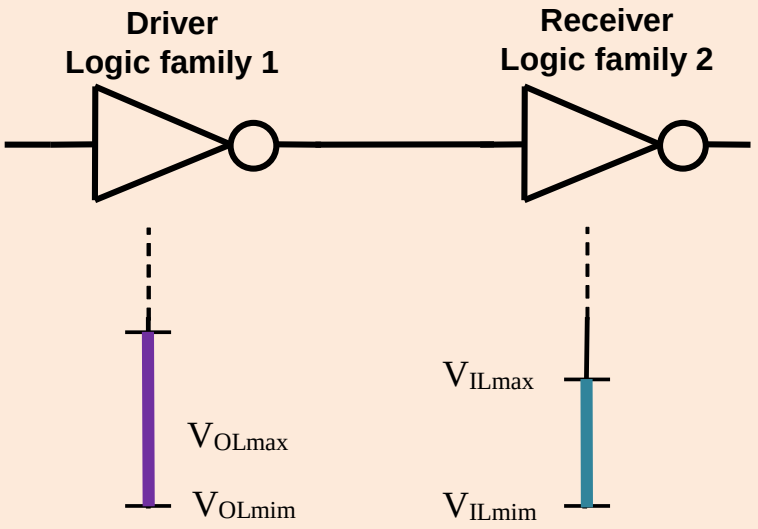
**Compatible logic 0 switching**

Logic 0 output range is within logic 0 input range



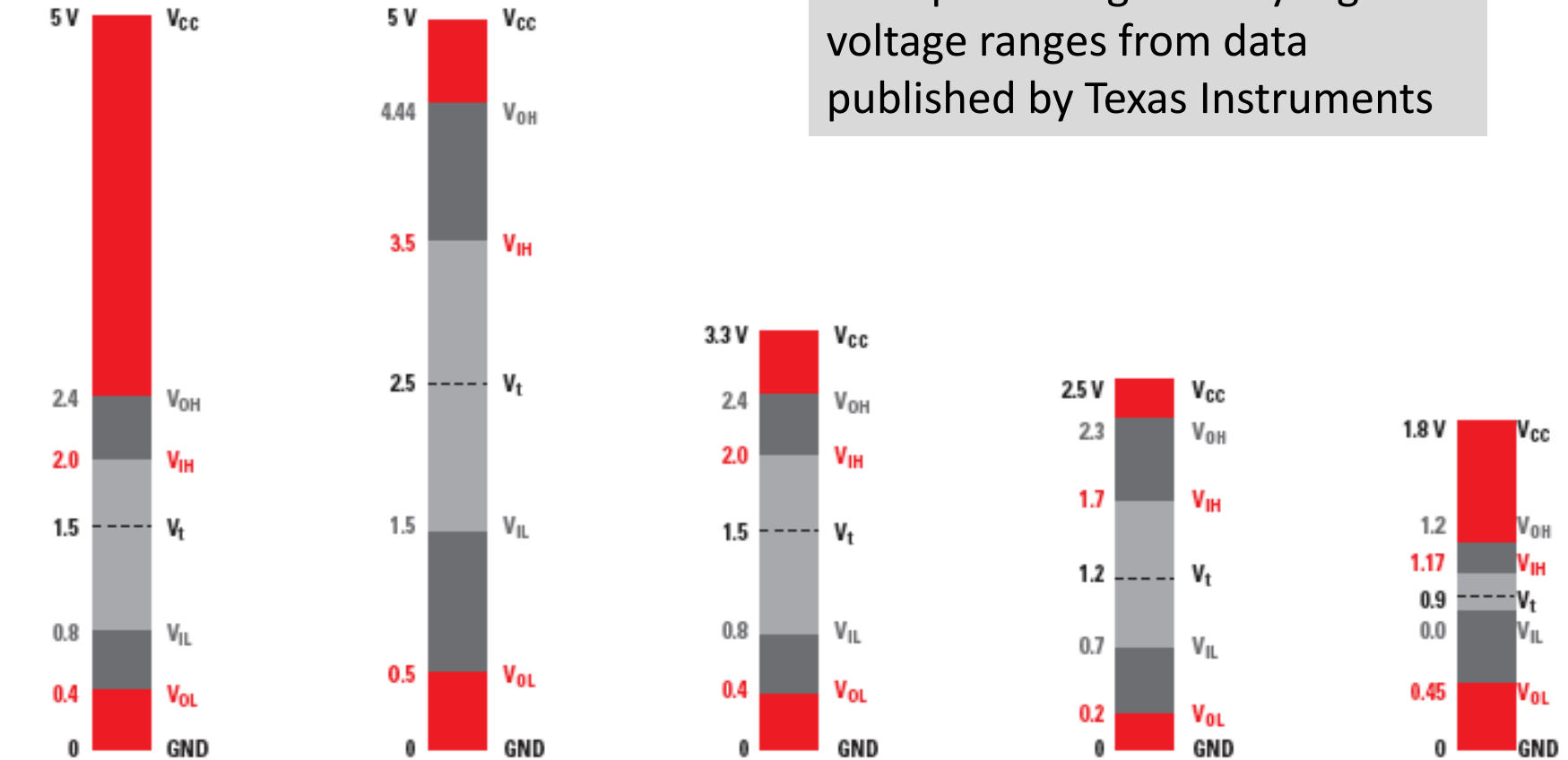

**Incompatible logic 0 switching**

Logic 0 output range is outside logic 0 input range



## IC Basics: Comparison of Switching Standards

Examples of logic family logic voltage ranges from data published by Texas Instruments




**5-V TTL**
Standard TTL: ABT, AHCT, HCT, ACT, bipolar

**5-V CMOS**
Rail-to-Rail 5 V HC, AHC, AC, LV-A

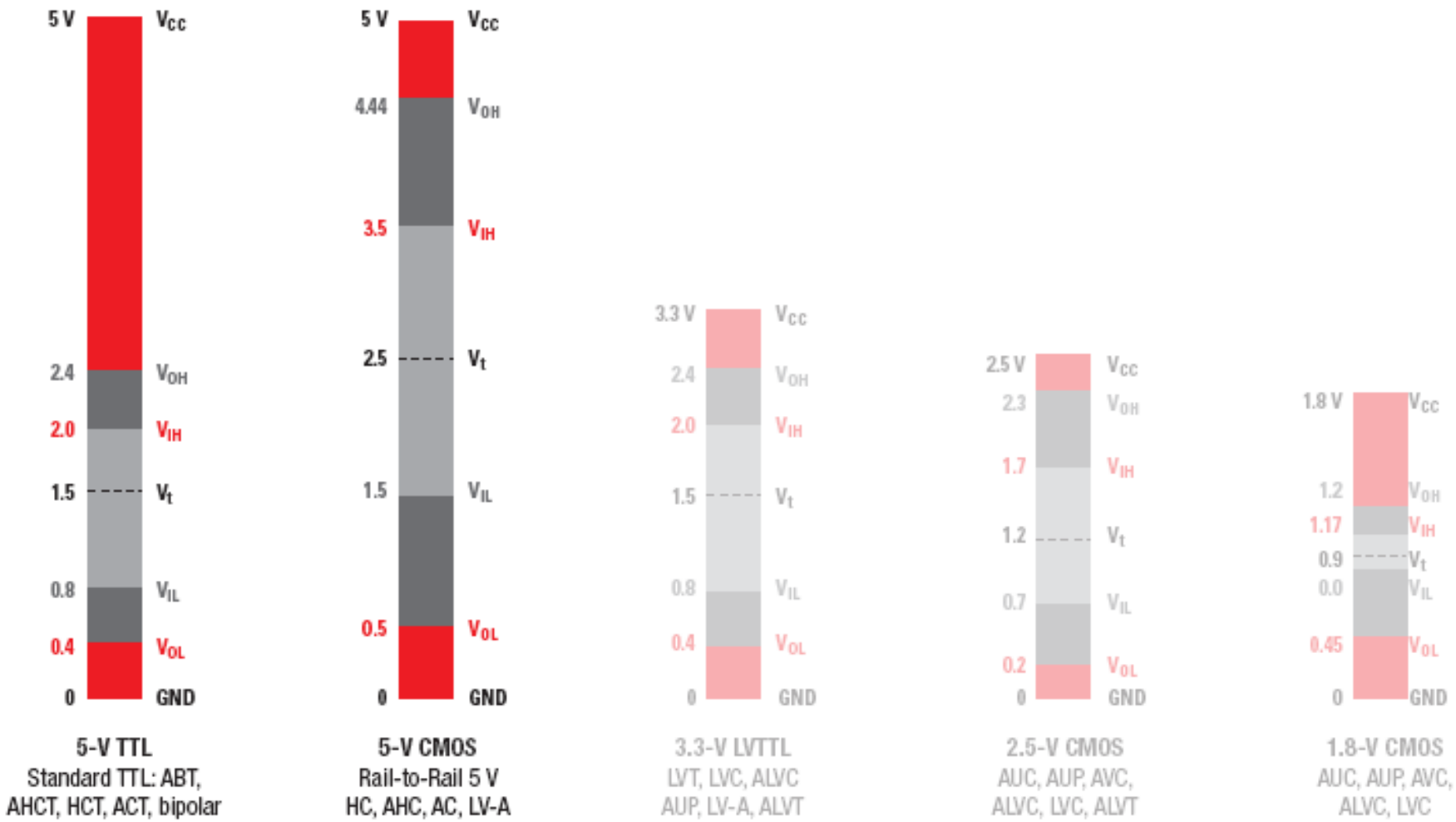
**3.3-V LVTTL**
LVT, LVC, ALVC AUP, LV-A, ALVT

**2.5-V CMOS**
AUC, AUP, AVC, ALVC, LVC, ALVT

**1.8-V CMOS**
AUC, AUP, AVC, ALVC, LVC

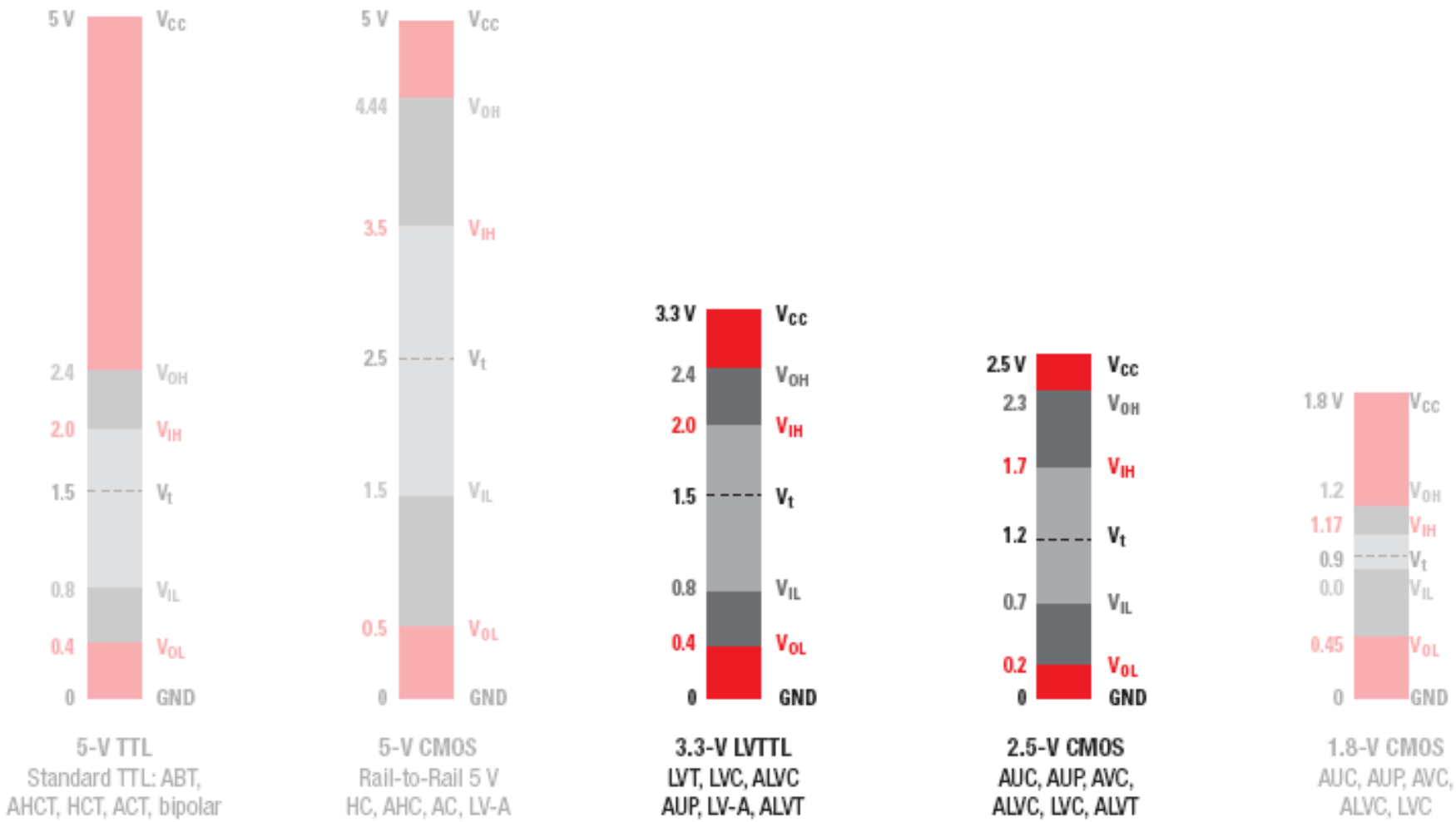
Texas Instruments. Logic Guide 2014. www.ti.com/lit/sg/sdyu001aa/sdyu001aa.pdf

If you connect a 5V TTL output to a 5V CMOS input – will the circuit work correctly?



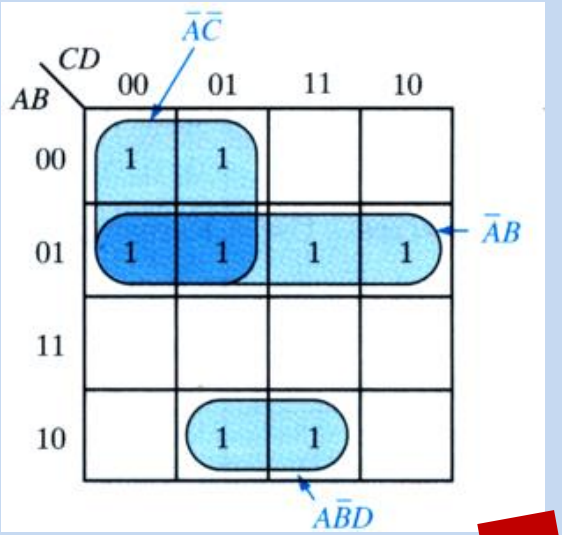

Texas Instruments. Logic Guide 2014. www.ti.com/lit/sg/sdyu001aa/sdyu001aa.pdf

If you connect a 3.3V TTL output to a 2.5V CMOS input – will the circuit work correctly?



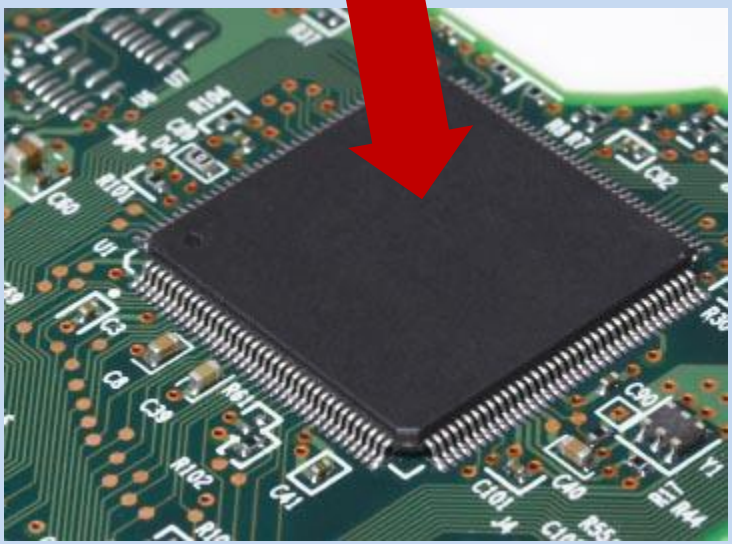

Texas Instruments. Logic Guide 2014. www.ti.com/lit/sg/sdyu001aa/sdyu001aa.pdf

# In reality there is more to worry about than just 1s and 0s




**From abstract logic design to physical circuit**

**?**



clemson.edu

***Implementation Technology***

- Choice of technologies for implementing a digital system
- **Internal circuitry of logic gates**

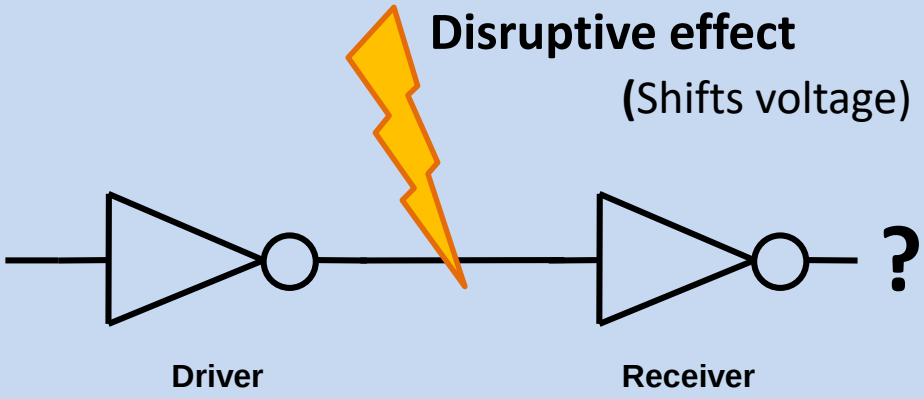
***Voltage & Current***

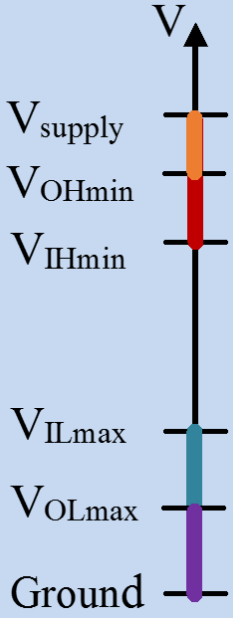
- Logic Voltages
- Noise and errors
- Supply voltages
- Power Consumption

***Time***

- Speed of operation
- Timing errors

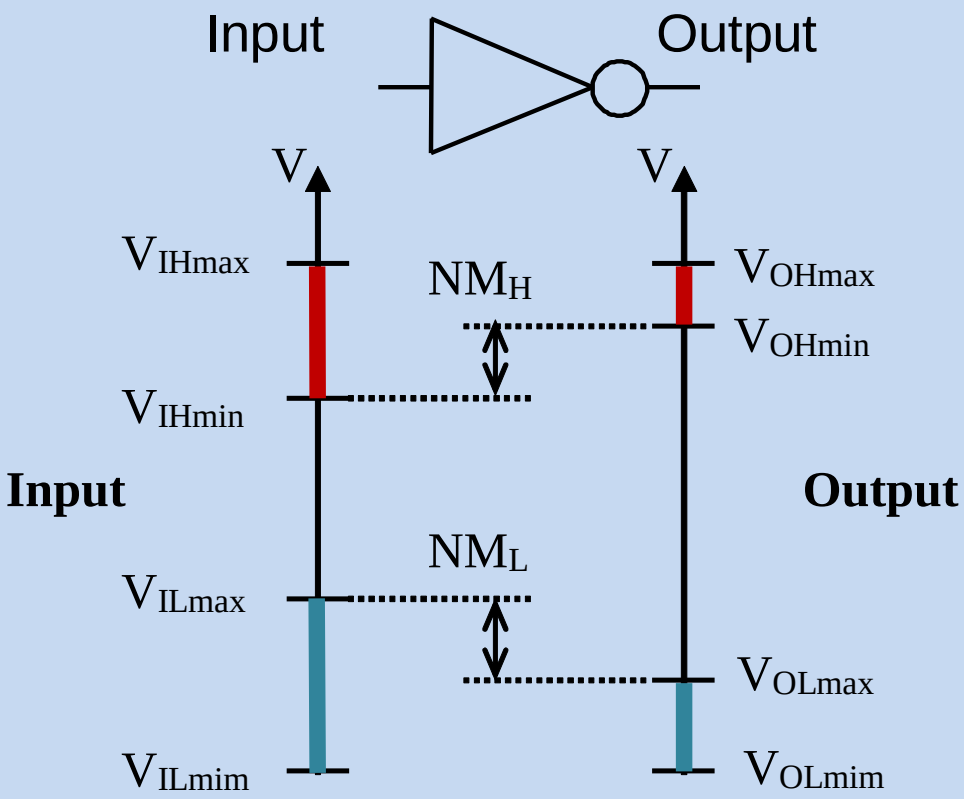
# Disruption to logic voltages




- External influences may disrupt the voltage on the connection between two logic gates
- If the disruption is large enough the receiving gate may interpret its input as the wrong logic level

# Noise Margin




Voltage range of valid logic levels

- Difference between worst case output level and worst acceptable input level for a given logic value is called *noise margin* (MN).
- Noise margin indicates the amount of disruption to logic voltages which can be tolerated by a circuit

**Noise margins**

Logic 1 $NM_H = V_{OHmin} - V_{IHmin}$

Logic 0 $NM_L = V_{ILmax} - V_{OLmax}$

Example

What are the noise margins for the 2.5V CMOS families?

Logic 1 $NM_H$ =

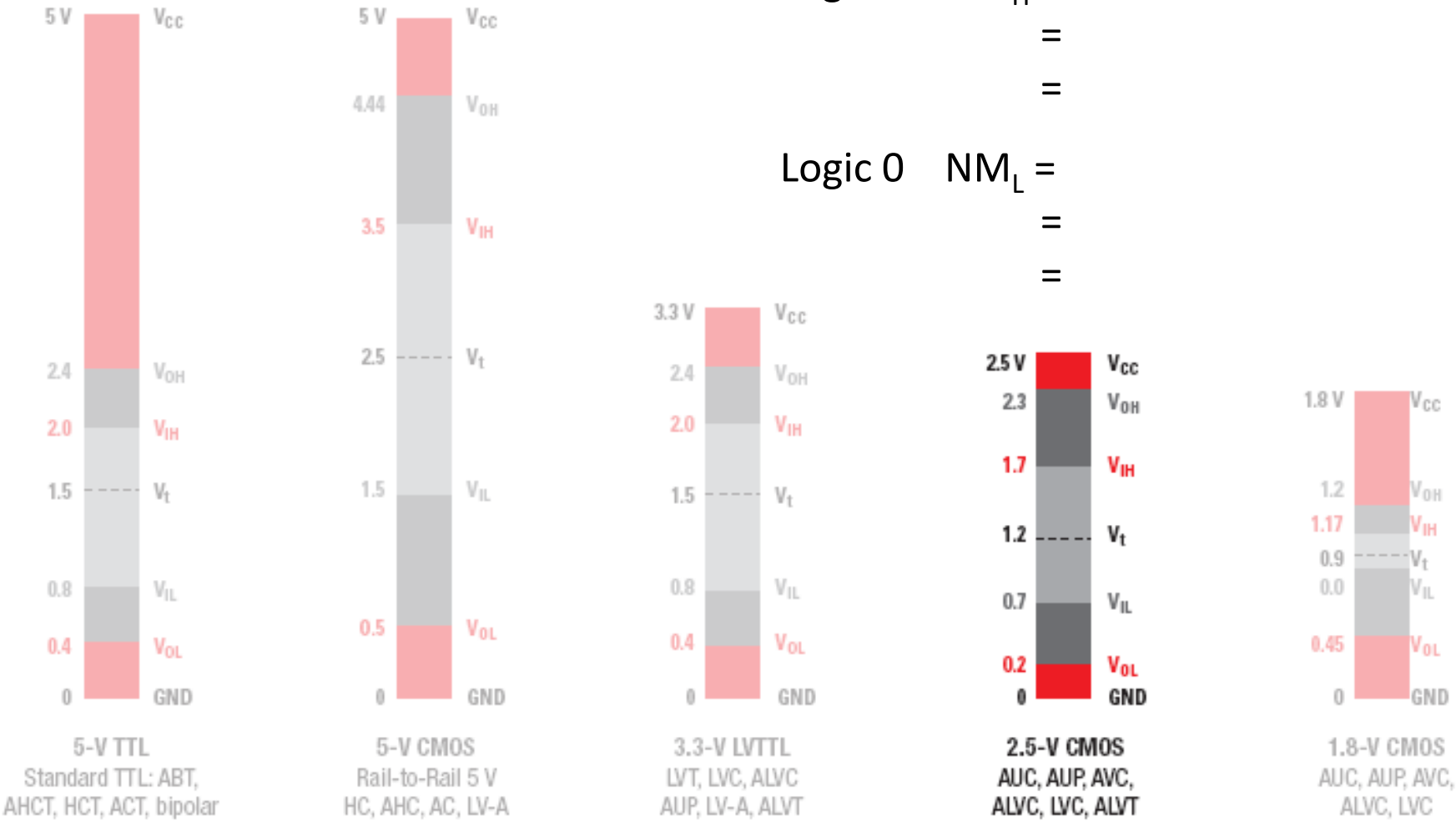
=

=

Logic 0 $NM_L$ =

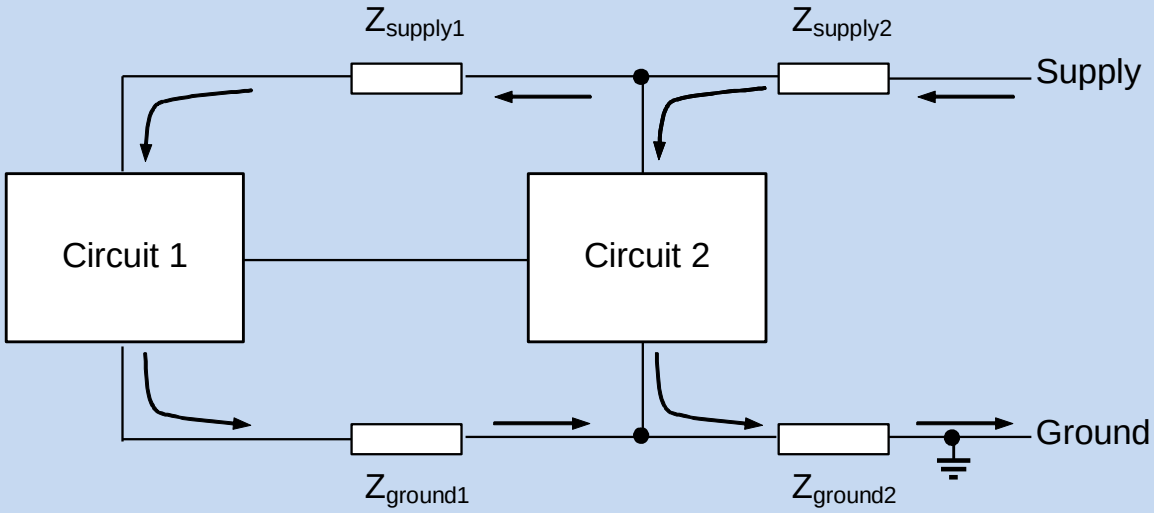
=

=



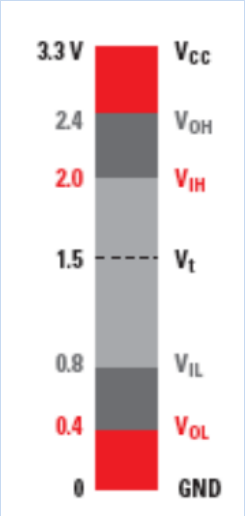

Texas Instruments. Logic Guide 2014. www.ti.com/lit/sg/sdyu001aa/sdyu001aa.pdf

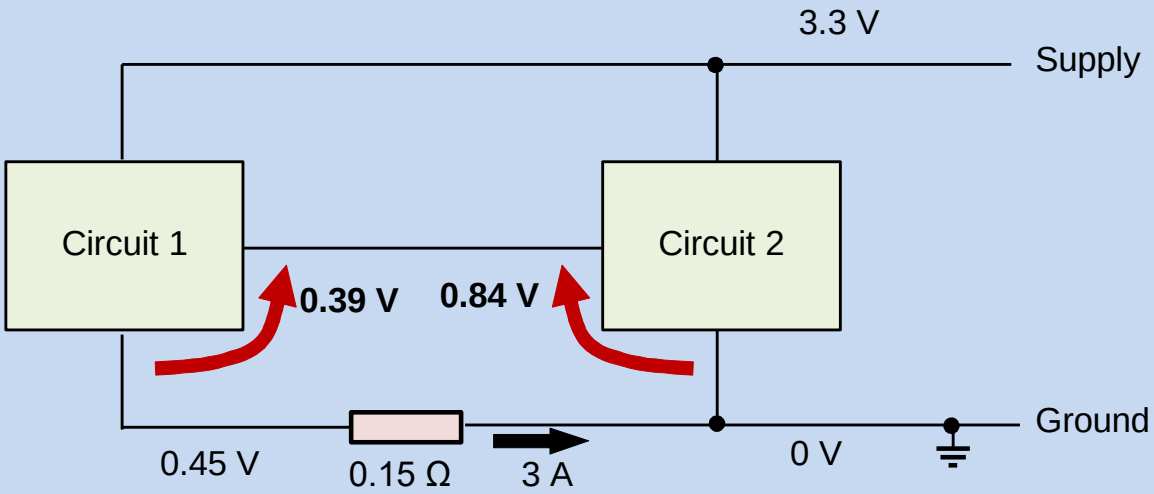
# Key cause of disruption - Supply and Ground Voltage Drops




- Supply wiring resistance and inductance may lead to logic level errors
- Supply current causes a voltage drop across the supply wiring
- The “ground” voltage at each subcircuit will not actually be OV.
- May lead to misinterpretation of logic levels (shift > logic margins)
- Error in the ground voltage is not necessarily constant : **ground bounce and supply sag**
- Current surge in one part causes another to switch when it shouldn’t.

# Supply and Ground Voltage Drops - Example




This example uses resistance for simplicity but in digital circuits the main problem is usually due to supply wiring inductance combined with rapid current changes as logic switches

Circuit 1 outputs **a valid logic 0** at 0.39 V

- $V_{OL}$ = 0.4 V is the maximum

Circuit 1 is taking 3 A supply current

- Typically a spike of high current due to switching

This supply current drops 0.45 V across the 0.15 Ω ground resistance

Circuit 2 sees circuit 1's 0.39V on top of the 0.45 ground drop (0.39+0.45 = 0.84)

Circuit 2 sees an input voltage of 0.84 – **not a valid logic 0**

Circuit 2's input is above the acceptable maximum input voltage for logic 0

- $V_{IL}$ = 0.8 V is the maximum

# Ground Bounce Example shown on Oscilloscope

Switching logic signals causing bounce

This is a unchanging logic zero output. It should a constant zero volts!

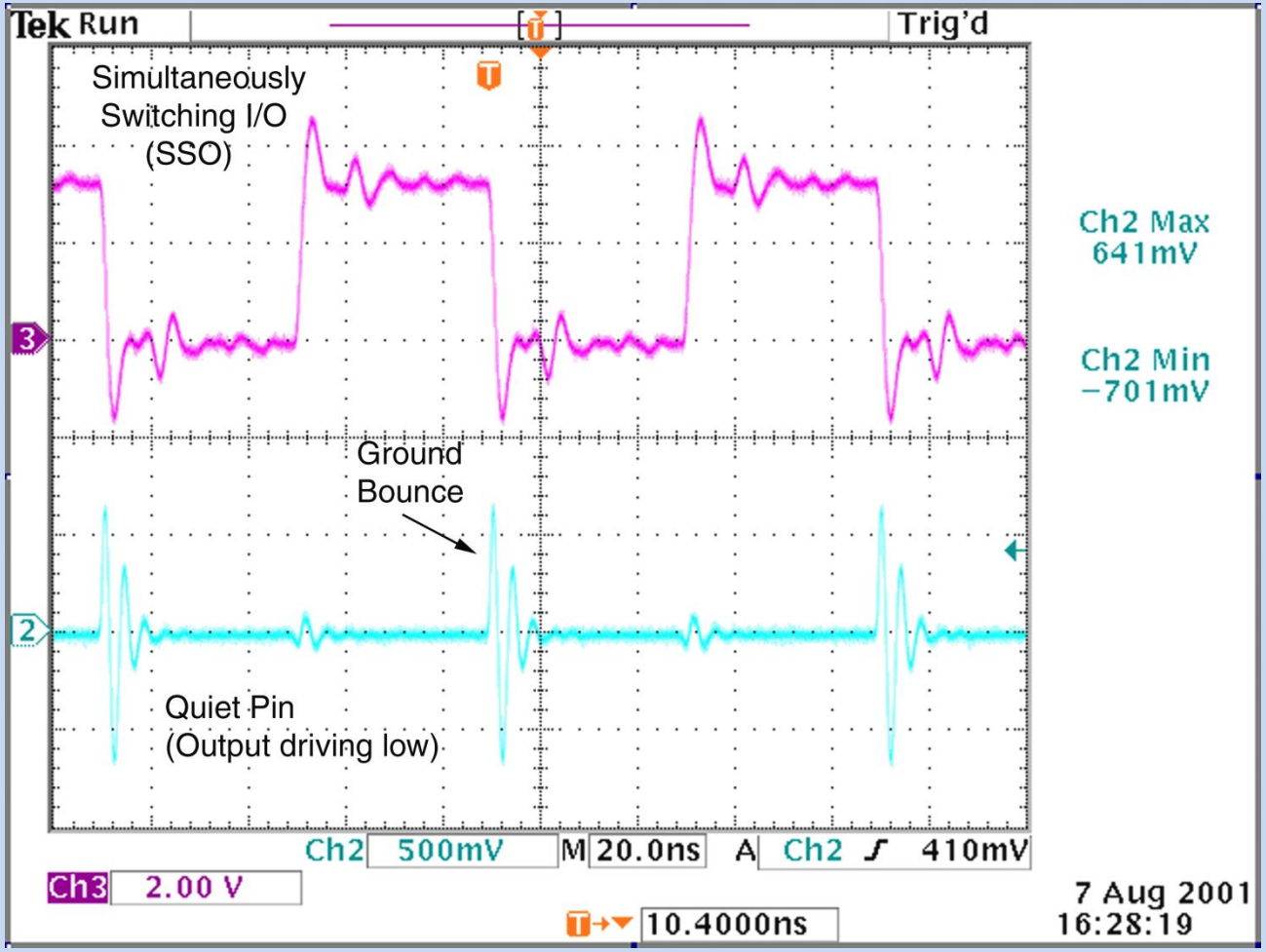
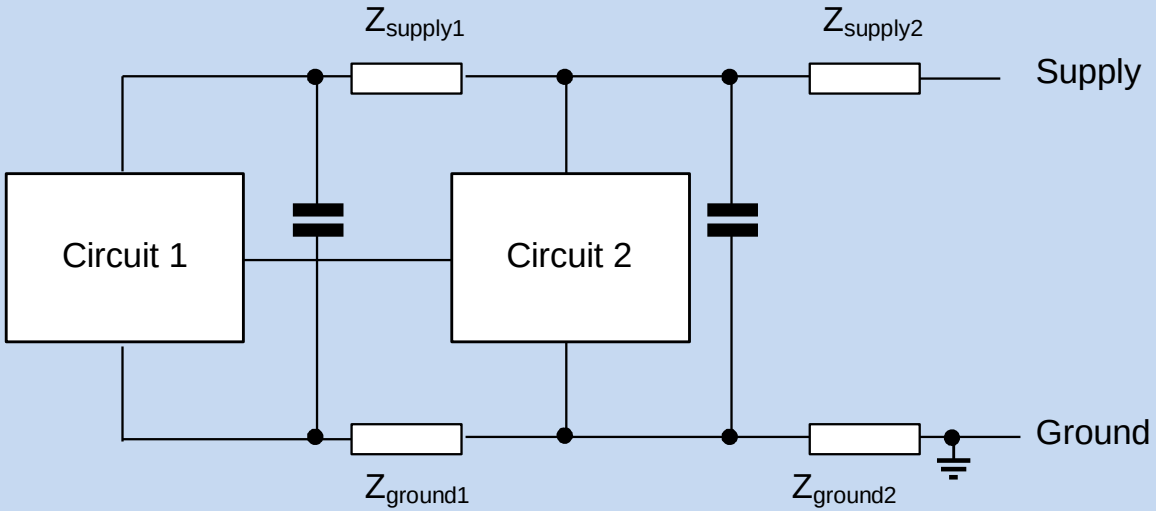
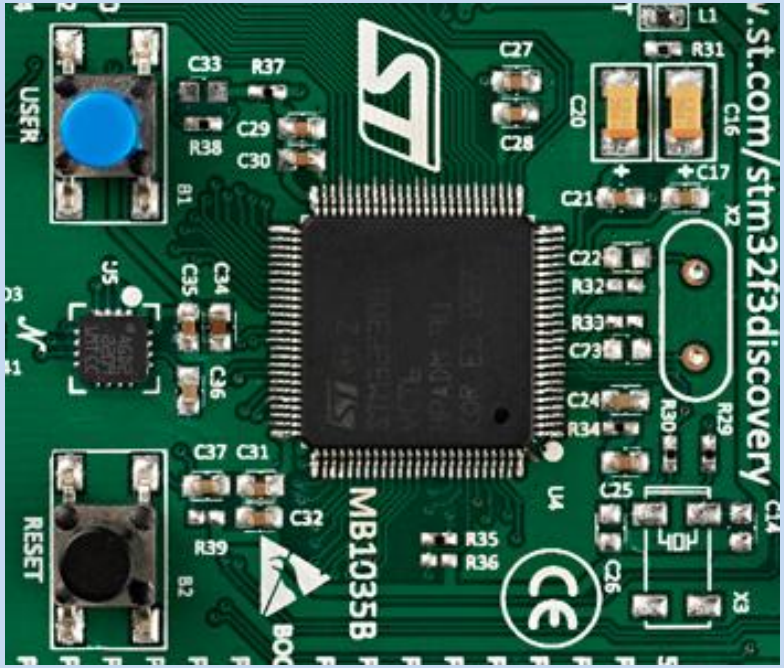



Image Source: Altera Corporation "Minimizing Ground Bounce & VCC Sag"

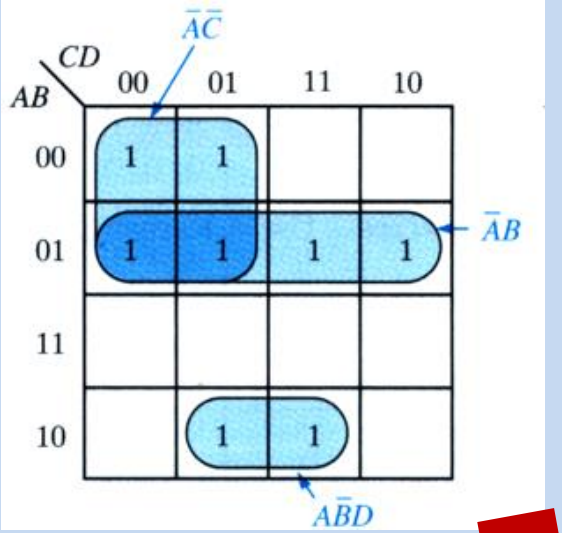
# Decoupling Capacitors




- Capacitors across the supply close to each IC power pin can help provide the required transient currents
- These are called **decoupling capacitors**
- You will seem many small capacitors on a typical circuit board – many will be decoupling capacitors

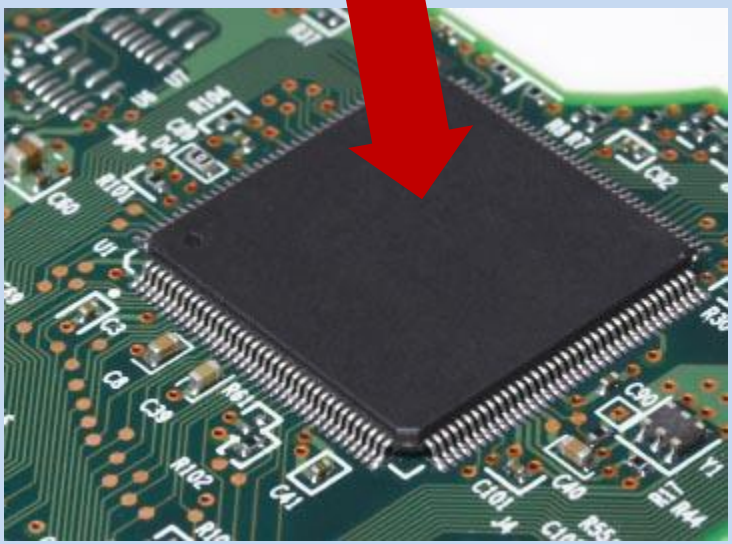


# In reality there is more to worry about than just 1s and 0s




**From abstract logic design to physical circuit**

**?**



clemson.edu

***Implementation Technology***

- Choice of technologies for implementing a digital system
- **Internal circuitry of logic gates**

***Voltage & Current***

- Logic Voltages
- Noise and errors
- Supply voltages
- Power Consumption

***Time***

- Speed of operation
- Timing errors

# Is 1 and 0 Enough?

Are 1 and 0 sufficient to describe what happens in real logic circuits?

Boolean algebra is based on variables which can take the values 1 and 0

Is this enough to represent what happens in real logic circuits?

On basic level **YES** – we can design useful circuits using the principles of Boolean algebra

But, in practice the answer is often **NO** – we need to be able to describe other situations which occur in real circuits.

# The X State

If the voltage is *in between* the defined logic levels (except during switching)

- Undefined logic value
- Unpredictably circuit behaviour

The undefined condition is often represented by the symbol X

- Meaning "*don't know*"
- Indicates that there is something wrong with the circuit or the design

The symbol X is also used to mean "*don't care*"

- Value may legitimately be either 1 or 0
- Used to simplify logic design

Whether X means "**don't know**" or "**don't care**" should be obvious from the context in which it is used.

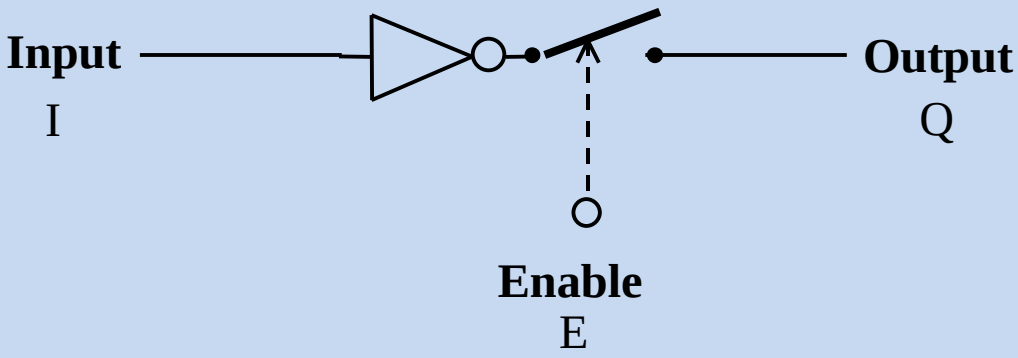
# The Z State

Some digital circuits that are able to disconnect and isolate their outputs

- We will look at these circuits in detail later
- The binary numbers 1 and 0 do not cover this situation
- X not appropriate - disconnection is a deliberate setting of a known condition

We use the symbol Z to represent this state

- Z for “high impedance”, as Z is the symbol for impedance




**Truth table**

| E | I | Q |
|---|---|---|
| 0 | 0 | Z |
| 0 | 1 | Z |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# Logic Levels

Basic set of logic levels

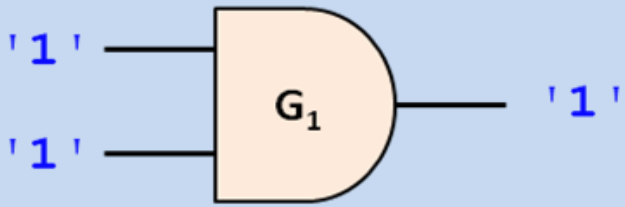
**0, 1, X, Z**

Other logic conditions are also encountered, in particular

weak-0, weak-1, uninitialized

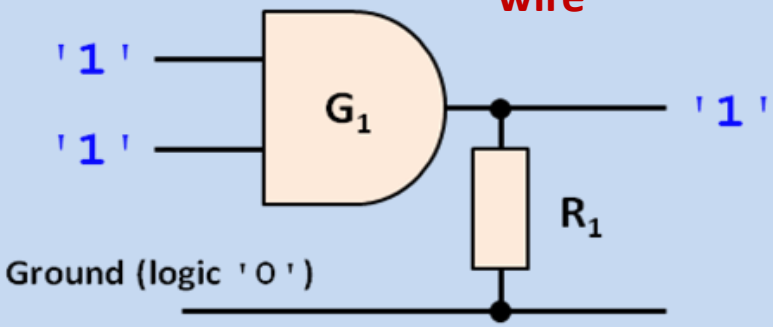
- **Weak** logic levels occur when a signal is not actively driven by a gate output or other strong voltage source
  - For example, a logic 0 obtained by connecting to ground via a (large) resistor
- **Uninitialized** values occur in digital circuit simulation
  - For example, when the user fails to specify what is happening to the circuit's inputs at the start of a simulation.

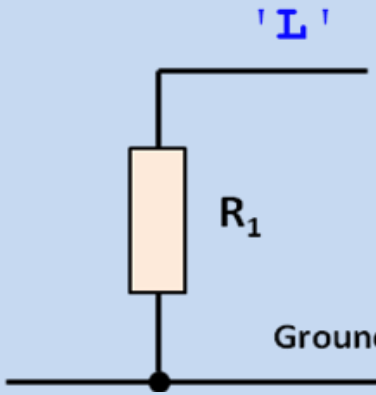
# Weak and Strong Logic Levels Example




Strong 1 output from a gate

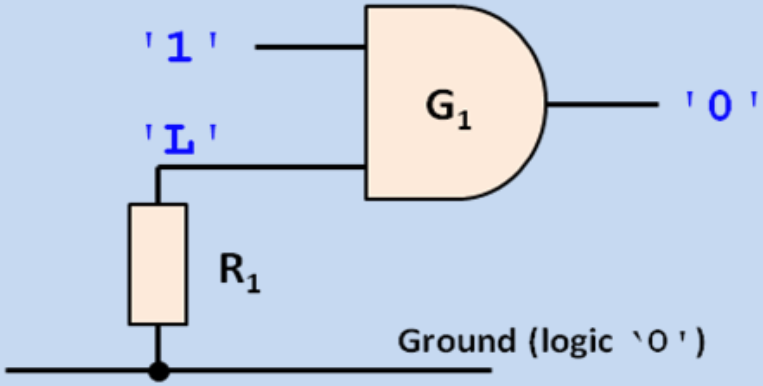
Strong 1 overrides weak zero on same wire




Weak 0 from grounded resistor

'L' rather than '0'




Weak 0 providing valid logic input

# Example - IEEE Standard 1164 Logic Levels

Defined by the Design Automation Standards Committee of the Institute of Electrical and Electronics Engineers (IEEE) in 1993

Defines nine logic levels

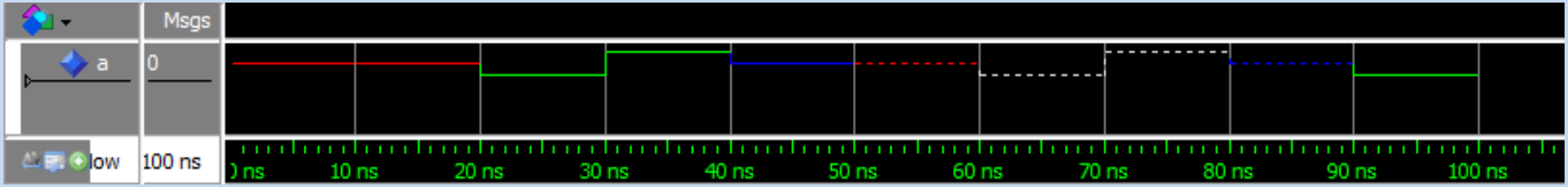
This standard is related to the VHDL “Hardware Description” computer language (HDL) used for describing, simulating and creating digital circuits

| | |
|---|---|
| `'U'` | **Uninitialized** |
| `'X'` | **Unknown**, *strongly driven* |
| `'0'` | **Logic 0**, *strongly driven* |
| `'1'` | **Logic 1** *strongly driven* |
| `'Z'` | **High Impedance** |
| `'W'` | **Unknown**, *weakly driven* |
| `'L'` | **Logic 0**, *weakly driven* |
| `'H'` | **Logic 1**, *weakly driven* |
| `'–'` | **Don't Care** |

# Simulators and Logic Levels

- Different logic simulators may use a different set of possible logic levels
- Most would include 0, 1, Z and X
- Some have various logic strengths (e.g. weak-1 and weak-0)
- It is important to read the simulator's documentation carefully
- Logic levels commonly relate to HDL standards

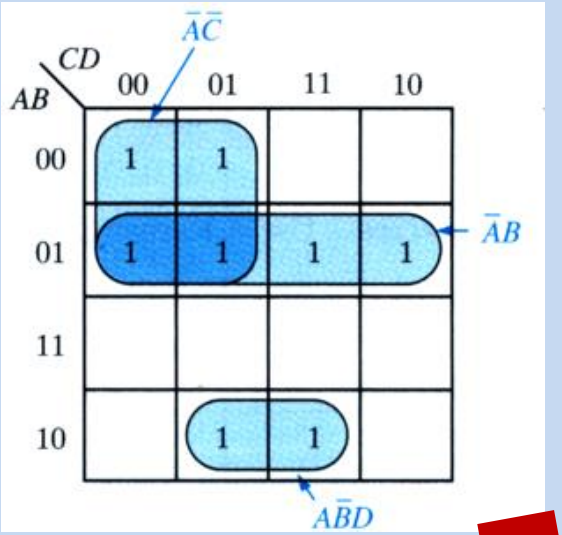
U X 0 1 Z W L H – 0




The graph is the output from a simulator as the signal “a” takes on a sequence of values from the IEEE Standard 1164 Logic Levels in VHDL

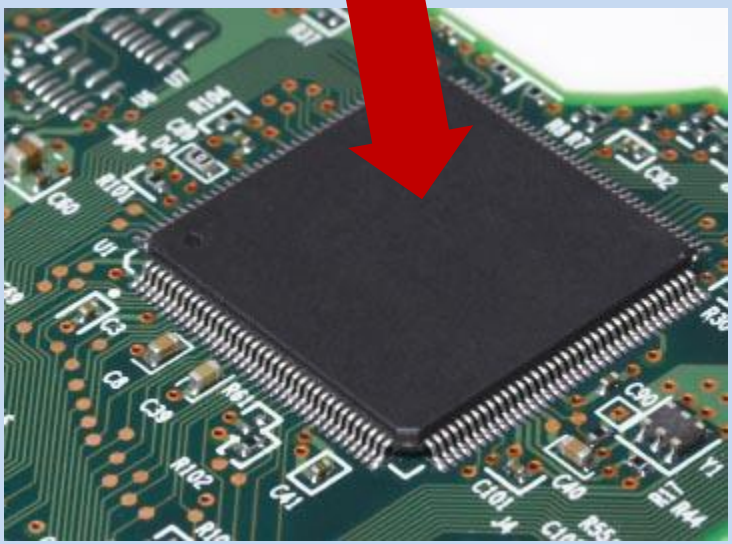
Position, colour and line style are used to indicate the logic level

# In reality there is more to worry about than just 1s and 0s




**From abstract logic design to physical circuit**

**?**



clemson.edu

***Implementation Technology***

- Choice of technologies for implementing a digital system
- **Internal circuitry of logic gates**

***Voltage & Current***

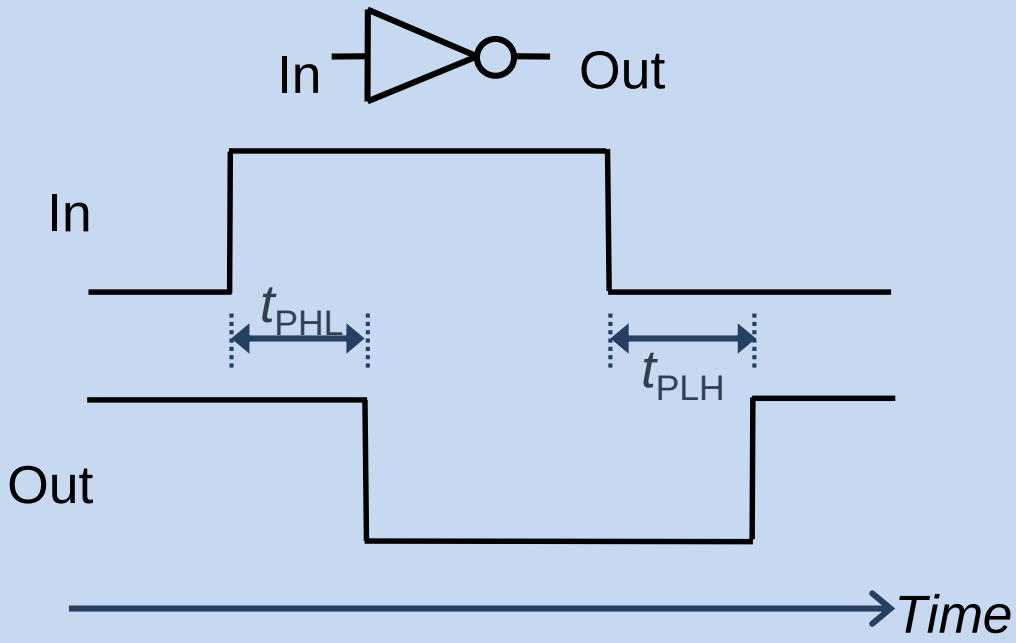
- Logic Voltages
- Noise and errors
- Supply voltages
- Power Consumption

***Time***

- Speed of operation
- Timing errors

# Logic in the Real World - Speed

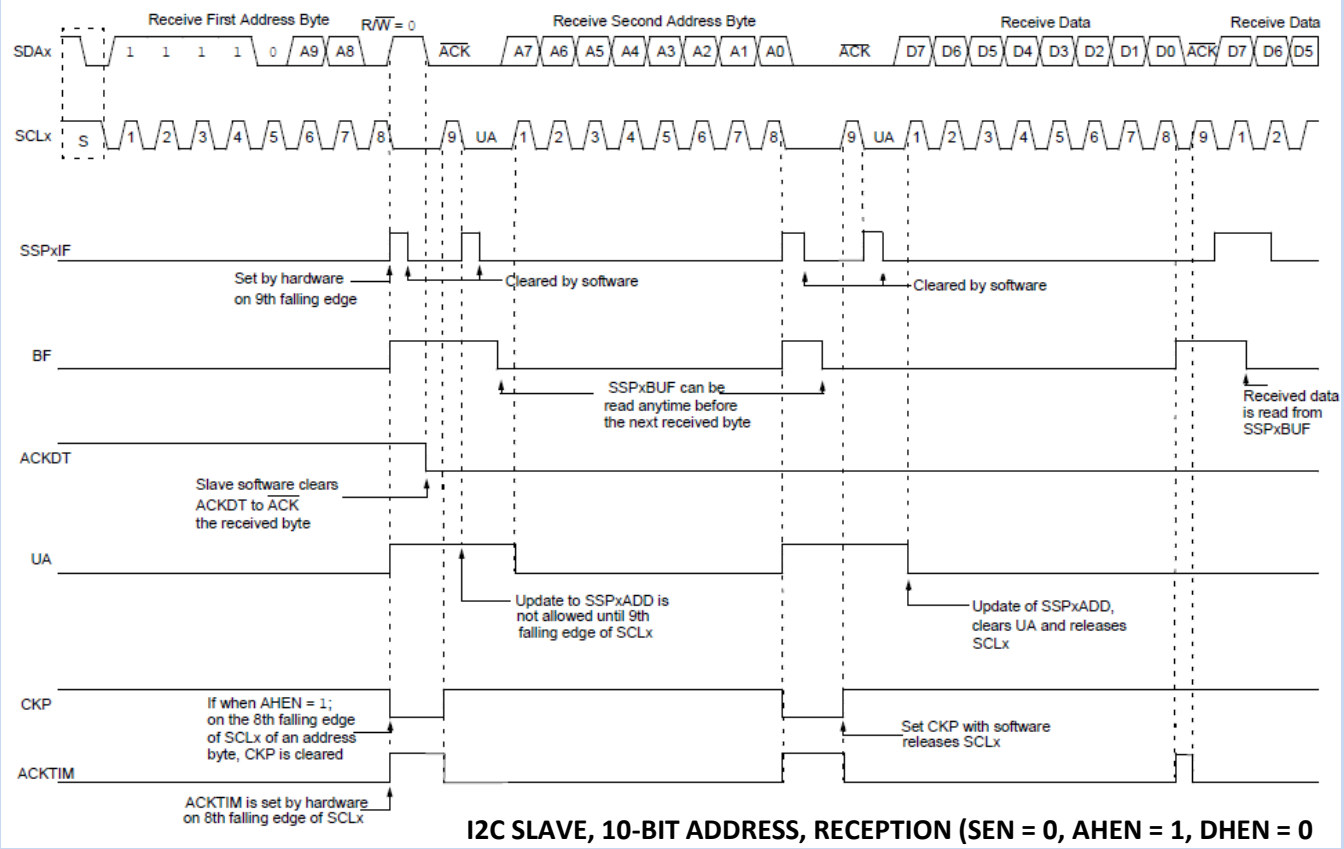
- Logic gates take time to react to changes on their inputs
- Time from input changing to output reacting is the **propagation delay $t_P$**
- Propagation delays may differ depending direction of change of the output
  - ❑ $t_{PHL}$ **Propagation delay high-to-low** (output change 1 to 0)
  - ❑ $t_{PLH}$ **Propagation delay low-to-high** (output change 0 to 1)




**Timing diagrams** are used to depict the timing of digital circuits

# Datasheet Timing Diagram Example

Timing diagrams are a very effective way to communicate information about circuit operation (e.g. in datasheets and reports)

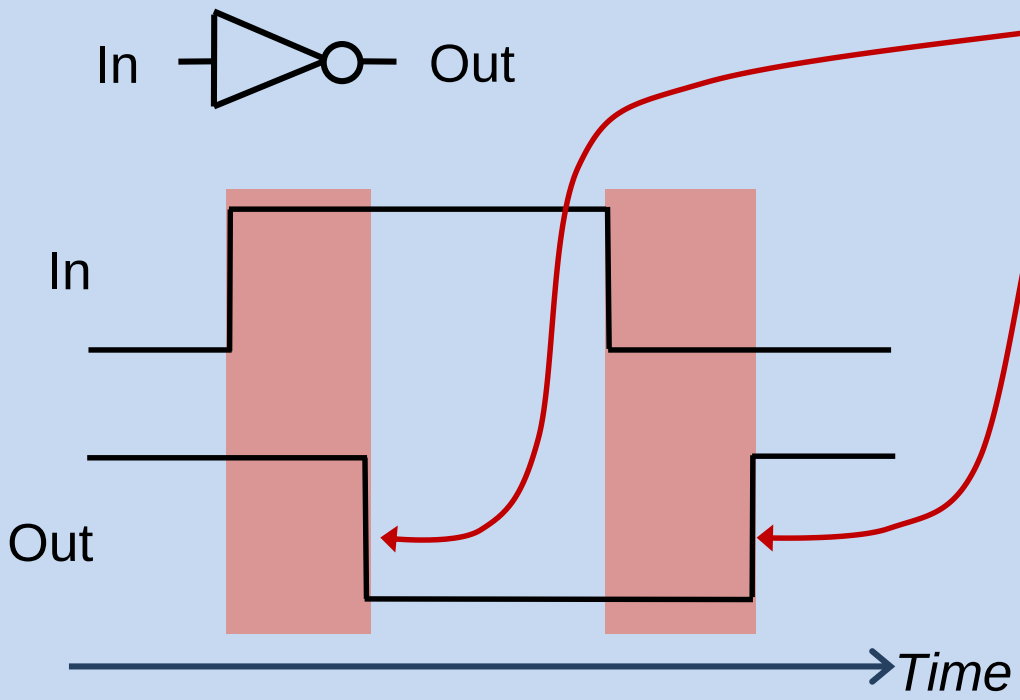



PIC18(L)F2X/4XK22 Datasheet DS41412D, 2010, Microchip Technology Inc.

**As a digital designer you need to be able to read and draw timing diagrams**

# Logic in the Real World - Speed

After a gate input changes the output may be “wrong” according to the Boolean function until the output has reacted



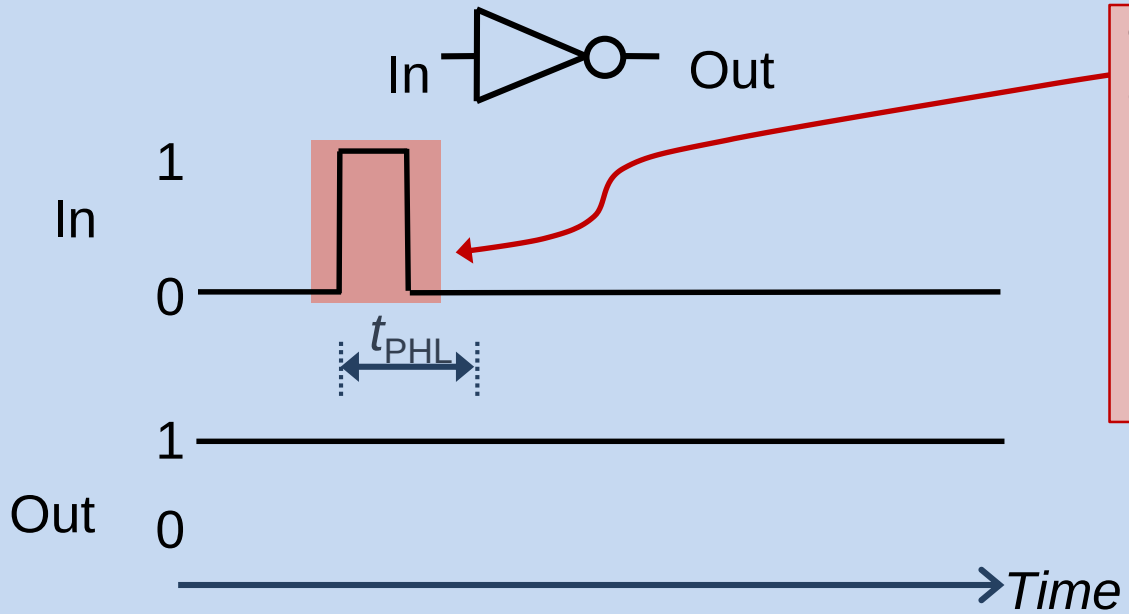

**During these times the output has not had time to respond to input changes**

**The output is not equal to the logical NOT of the input**

We have to be careful that intermediate states which occur as circuits switch do not cause incorrect behaviour in other parts of the circuit

# Logic in the Real World - Speed

Very short pulses on gate inputs may not produce any output response

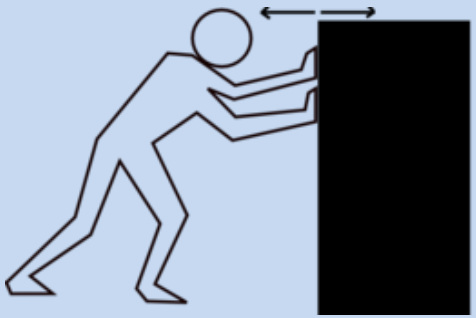



**This input pulse is so short that the output never changes**

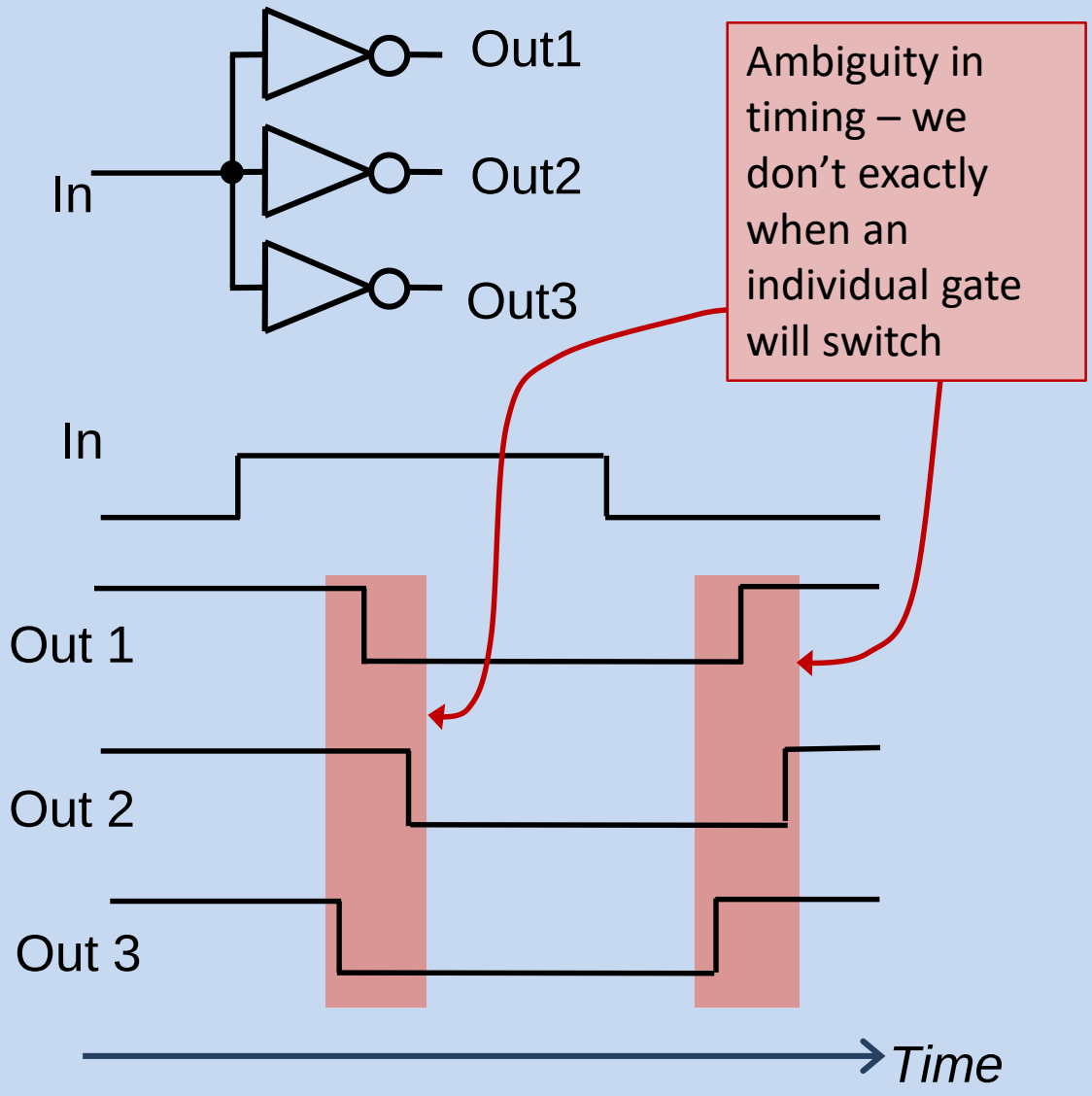
**In general if the input pulse is shorter than the propagation delay the output will not respond**

The short pulse effect is similar to **inertia** – a heavy object will not move if you only push it lightly

The idea that the gate does not respond to pulses shorter than the propagation delay is called the **inertial delay model** and is used by simulators.



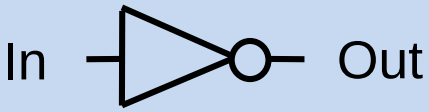
# Variability of delay

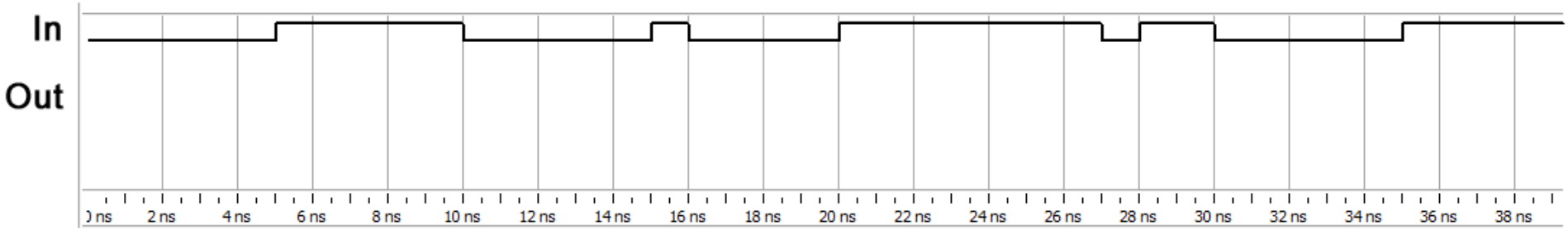



- Individual logic gates from same family will have different delays
- Delay varies with
  - ❑ Supply voltage
  - ❑ Temperature
  - ❑ Load (number of other gates driven)
- Even if all these are equal individual gates will have different delays
- A gate cannot generally be assumed to have a specific delay

# Logic in the Real World - Speed

Draw the output waveform




Propagation delay 2 ns