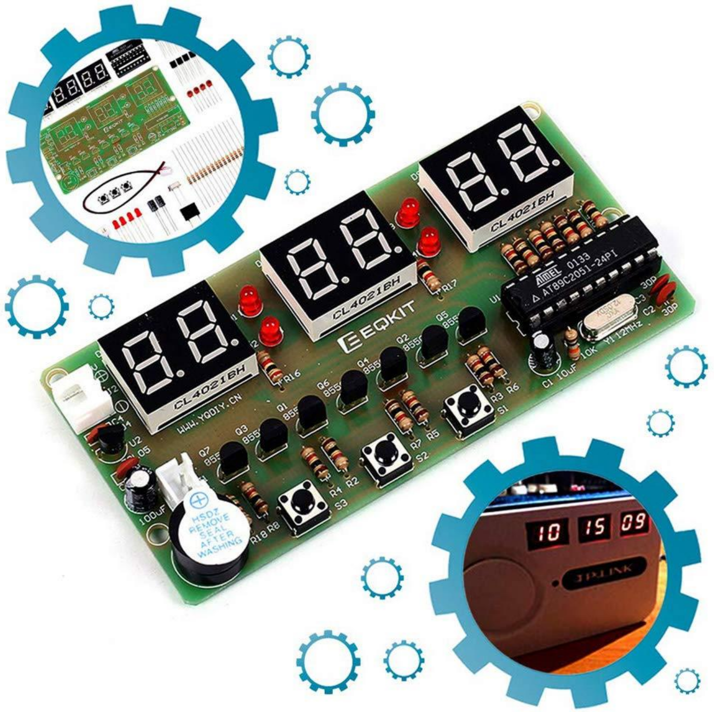


Dr. Hiran Ekanayake

DIGITAL ELECTRONIC FUNDAMENTALS – PART 2

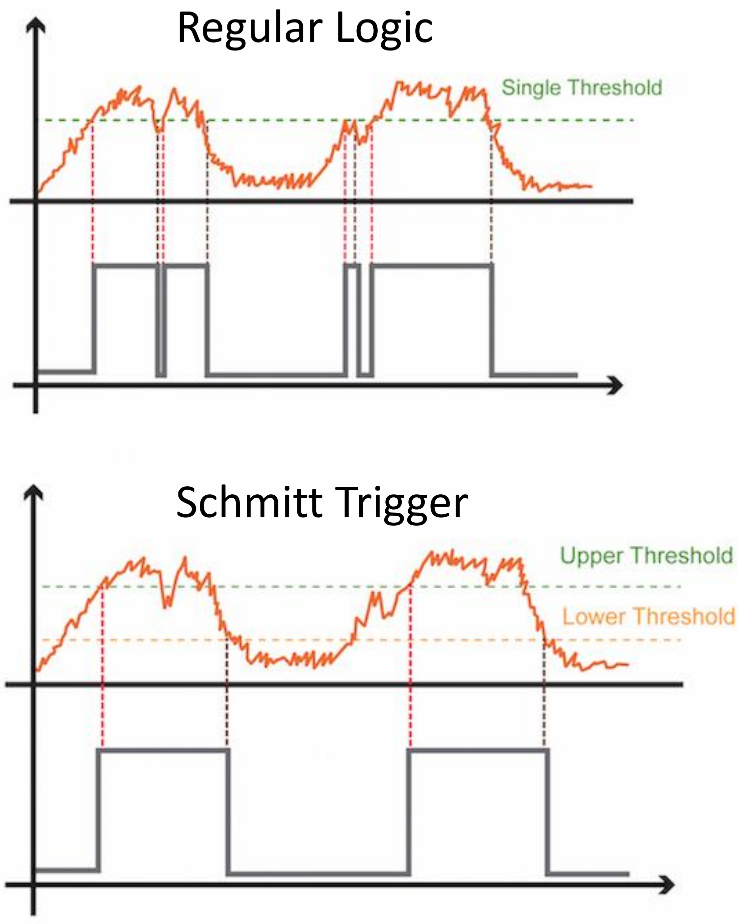
Lesson Outline

- AND, OR, and NOT logic functions
- Digital logic ICs: logic levels, IC families, data sheets, etc.
- Schmitt Trigger
- NAND, NOR, XOR, and XNOR logic functions
- Digital Buffers
- Arduino-based Logic Function Detector

SCHMITT TRIGGER

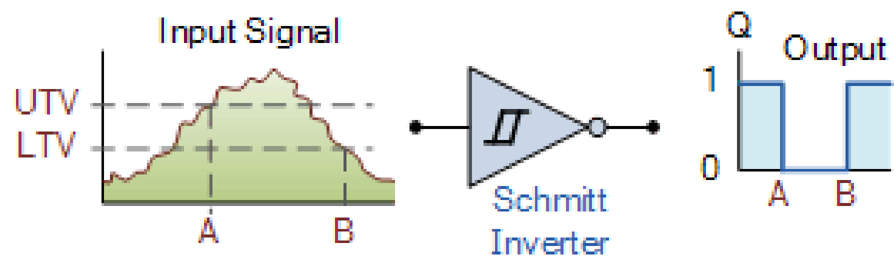
Schmitt Trigger

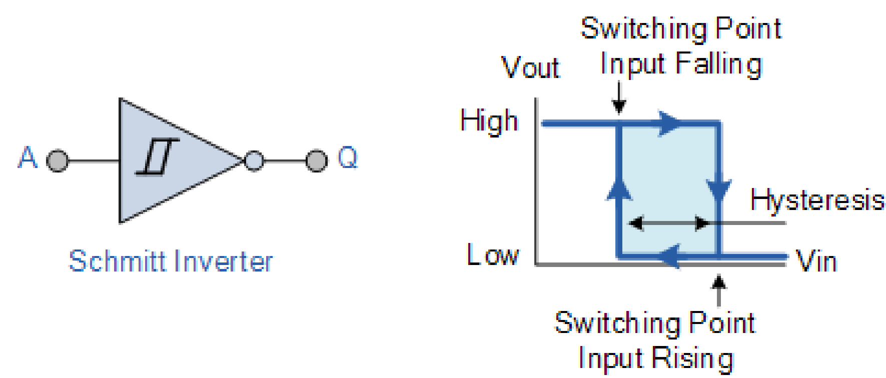
- Briefly explain how a Schmitt inverter minimizes the effects of noise in an input signal.
  - A Schmitt Inverter is designed to operate or switch state when its input signal goes above an “Upper Threshold Voltage” or UTV limit in which case the output changes and goes “LOW” and will remain in that state until the input signal falls below the “Lower Threshold Voltage” or LTV level in which case the output signal goes “HIGH”.



Schmitt Trigger

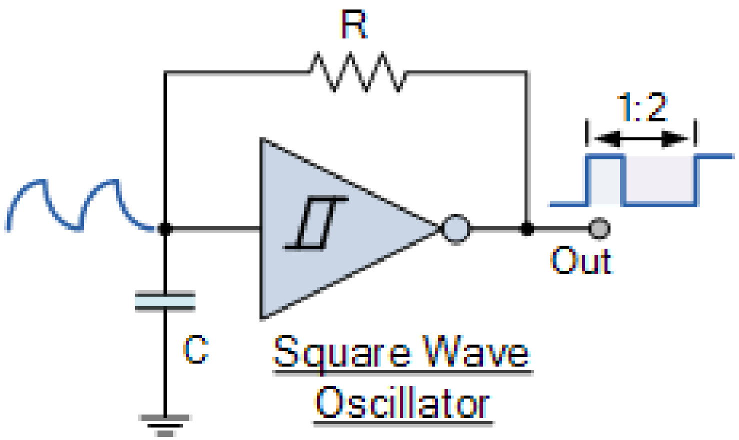
- Briefly explain how a Schmitt inverter minimizes the effects of noise in an input signal.
  - A Schmitt Inverter is designed to operate or switch state when its input signal goes above an "Upper Threshold Voltage" or UTV limit in which case the output changes and goes "LOW" and will remain in that state until the input signal falls below the "Lower Threshold Voltage" or LTV level in which case the output signal goes "HIGH".

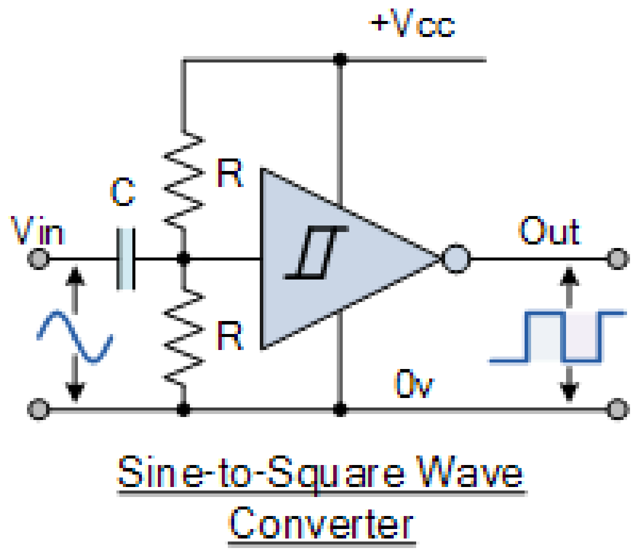




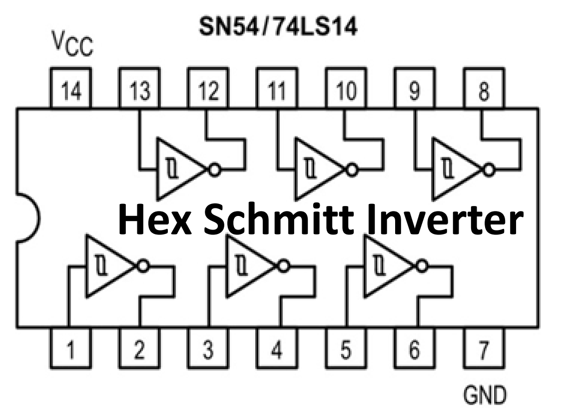
It provides a much cleaner and faster "ON/OFF" switching output signal

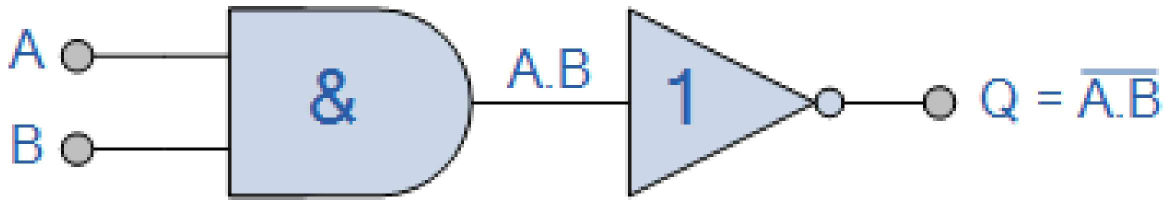
Applications of Schmitt Inverter



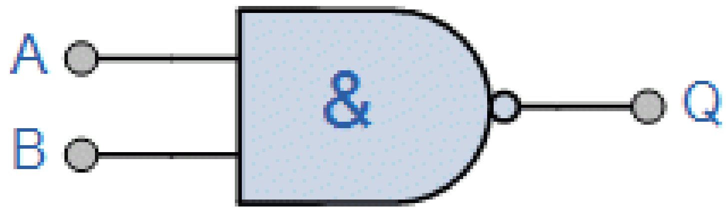


$$f = 680/RC.$$





2-input "AND" gate plus a "NOT" gate

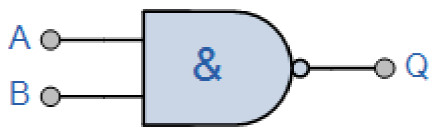


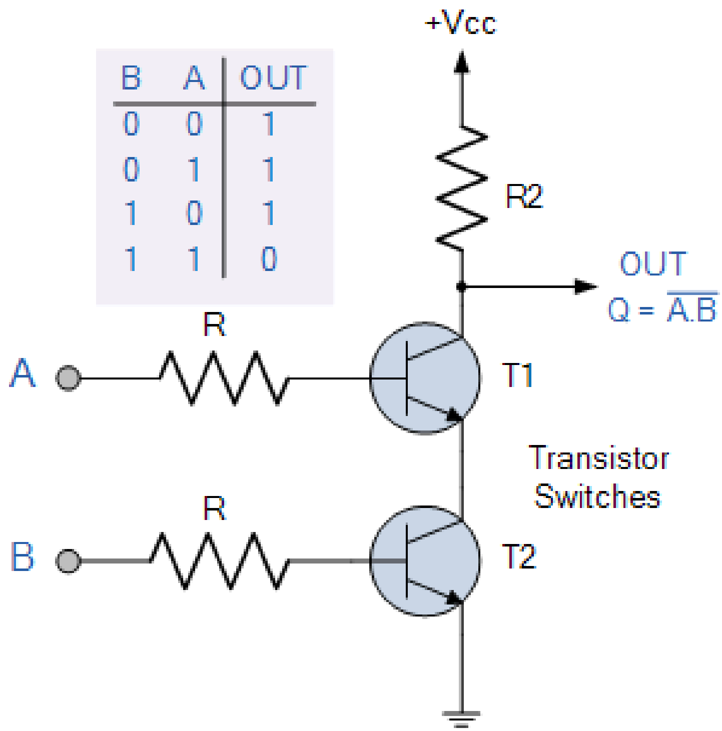
LOGIC NAND

"The Logic NAND Function output is normally true and only goes to false when all of its inputs are true"

"Compliment/Inverse of AND"

LOGIC NAND

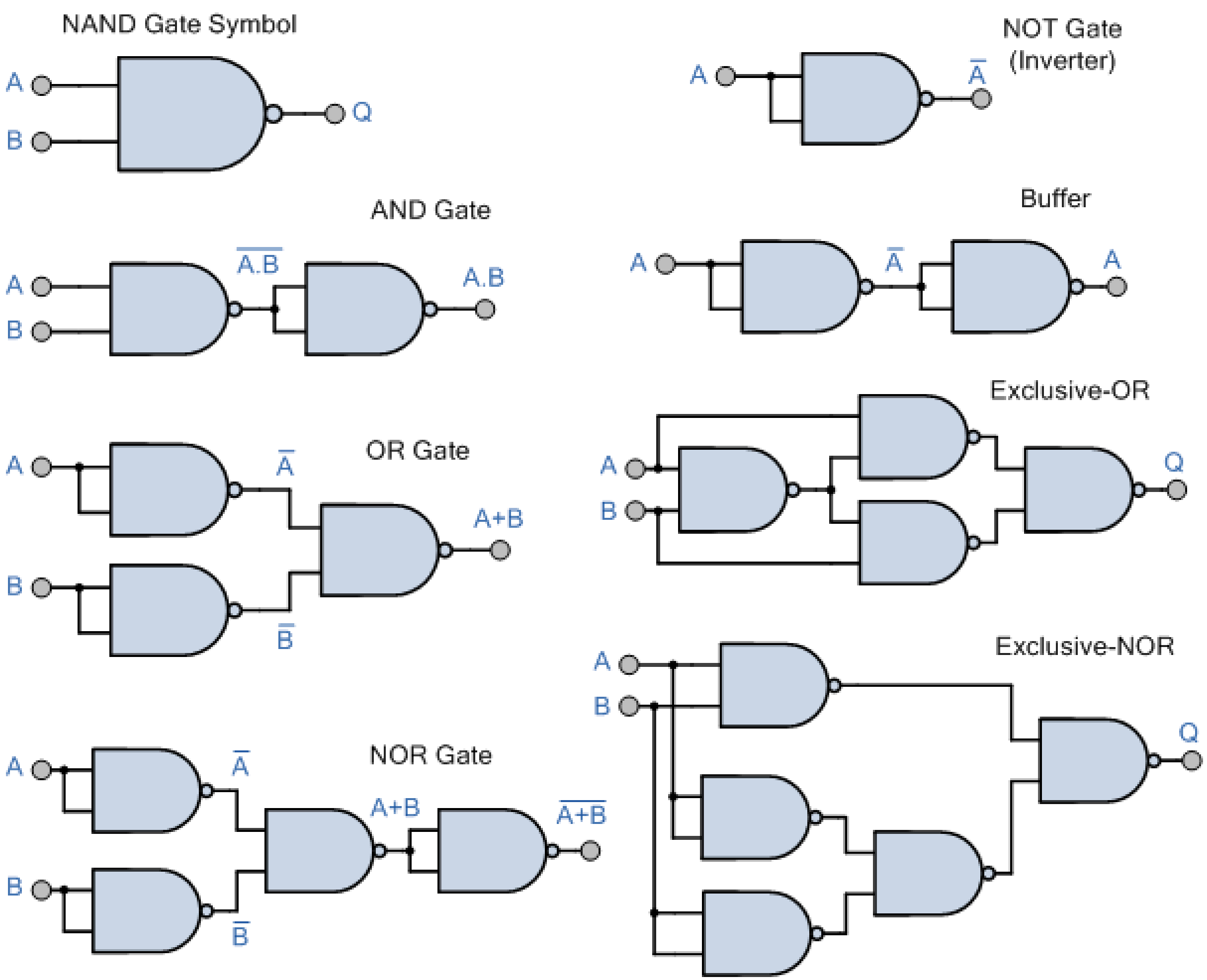
Symbol	Truth Table		
 2-input NAND Gate	B	A	Q
	0	0	1
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = \overline{A.B}$	Read as A AND B gives NOT Q		

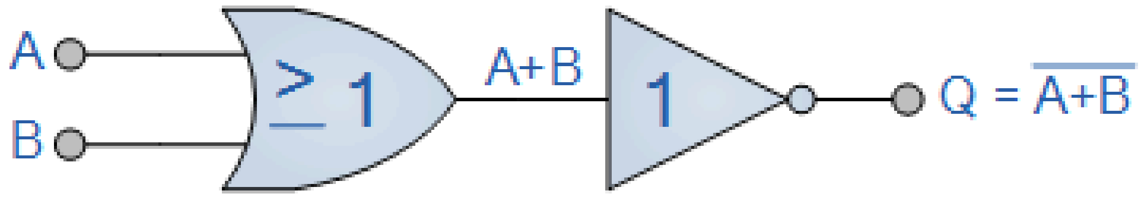


Sheffer Stroke Function

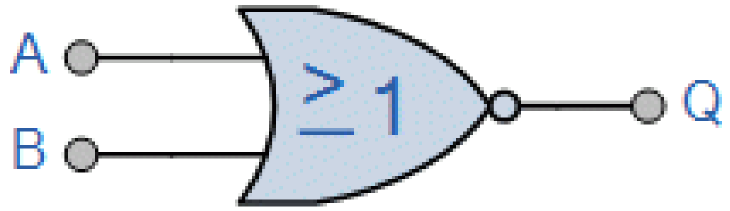
- Logic NAND function is sometimes known as the Sheffer Stroke Function, denoted by A|B or A↑B
- Why? NAND is functionally complete, i.e., it can be used to produce any other type of logic gate function; thus, it is considered as a "universal" gate

Other Logic Functions Using NAND Gate





2-input "OR" gate plus a "NOT" gate

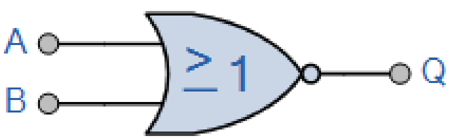


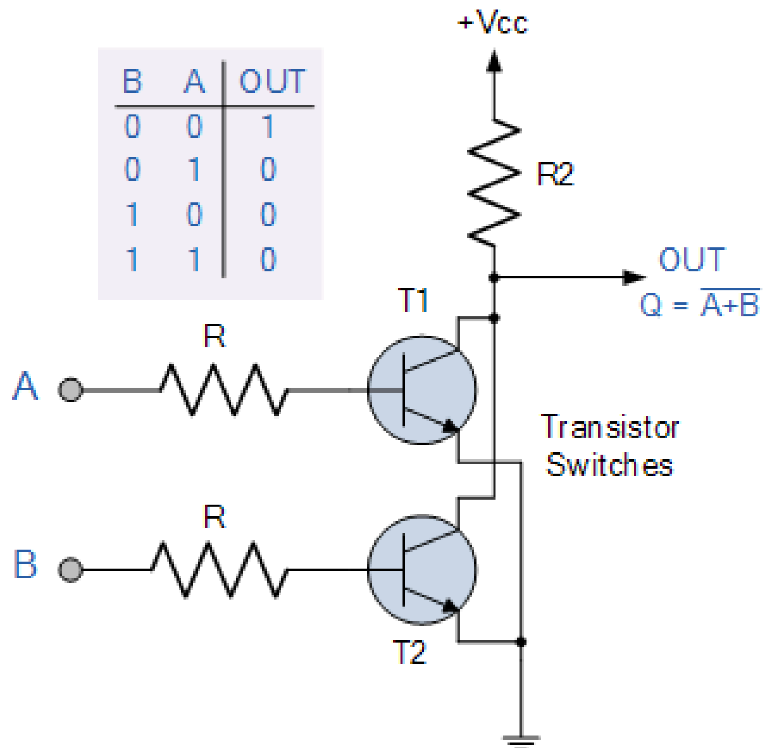
LOGIC NOR

"The Logic NOR Function output is normally true and only goes to false when any of its inputs are true"

"Compliment/Inverse of OR"

LOGIC NOR

Symbol	Truth Table		
	B	A	Q
 2-input NOR Gate	0	0	1
	0	1	0
	1	0	0
	1	1	0
Boolean Expression Q = $\overline{A+B}$	Read as A OR B gives NOT Q		

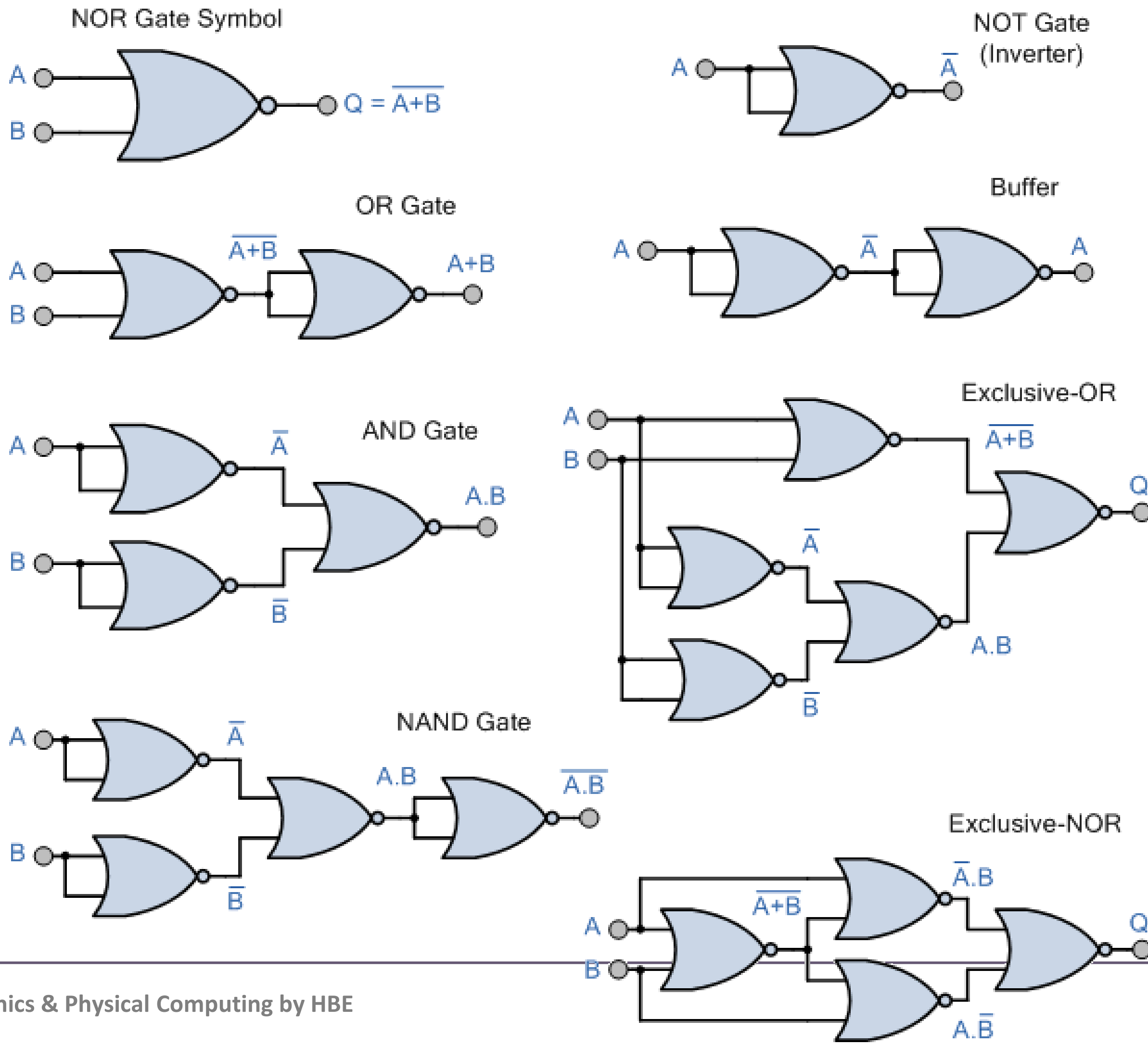


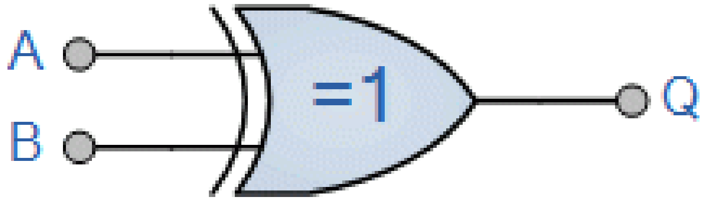
B	A	OUT
0	0	1
0	1	0
1	0	0
1	1	0

Pierce Function

- Logic NOR function is sometimes known as the Pierce Function, denoted by A↓B
- Why? NOR is functionally complete, i.e., it can be used to produce any other type of logic gate function; thus, it is considered a "universal" gate

Other Logic Using NOR Gate

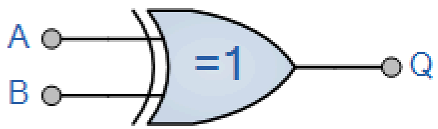




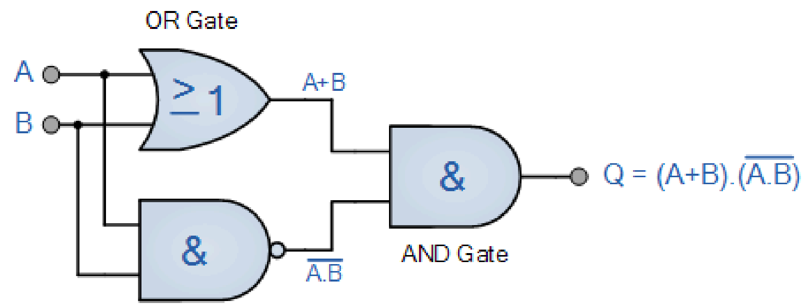
EXCLUSIVE-OR (XOR)

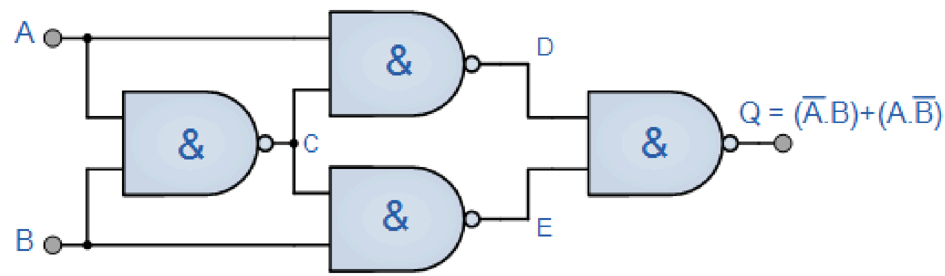
"The Logic XOR Function output is only true when its two input terminals are at different logic levels with respect to each other."

LOGIC XOR

Symbol	Truth Table		
 2-input Ex-OR Gate	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = A \oplus B$	A **OR** B but NOT **BOTH** gives Q		

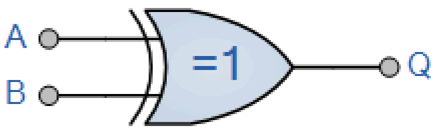
$$Q = (A \oplus B) = \overline{A}.B + A.\overline{B}$$





NAND Gate Realisation

More Than 2-Inputs?

Symbol	Truth Table		
	B	A	Q
 2-input Ex-OR Gate	0	0	0
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = A \oplus B$	A **OR** B but NOT **BOTH** gives Q		

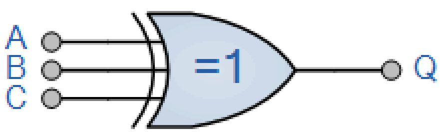
$$Q = (A \oplus B) = \overline{A}.B + A.\overline{B}$$

"A OR B but NOT both"

"Odd Function"

"Mod-2-SUM"

More Than 2-Inputs?

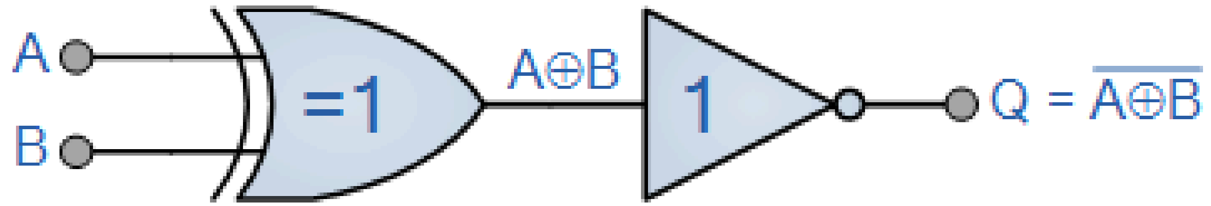
Symbol	Truth Table			
 3-input Ex-OR Gate	C	B	A	Q
	0	0	0	0
	0	0	1	1
	0	1	0	1
	0	1	1	0
	1	0	0	1
	1	0	1	0
	1	1	0	0
	1	1	1	1
Boolean Expression $Q = A \oplus B \oplus C$	"Any **ODD** Number of Inputs" gives Q			

"A OR B but NOT both"

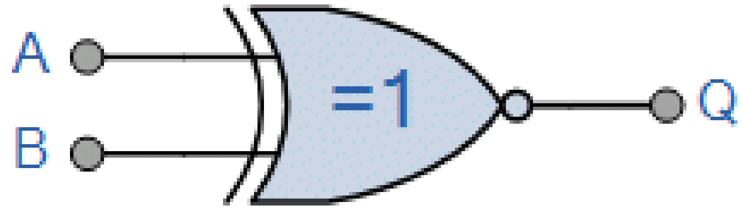
"Odd Function"

"Mod-2-SUM"

$$Q = A\overline{BC} + \overline{A}B\overline{C} + \overline{AB}C + ABC$$



2-input "Ex-OR" gate plus a "NOT" gate

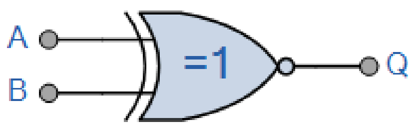


EXCLUSIVE-NOR (XNOR)

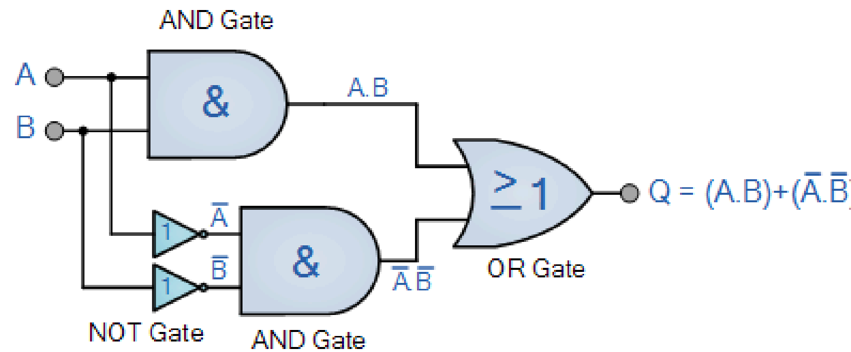
"The Logic XNOR Function output is only true when both of its inputs are at the same/equal logic level"

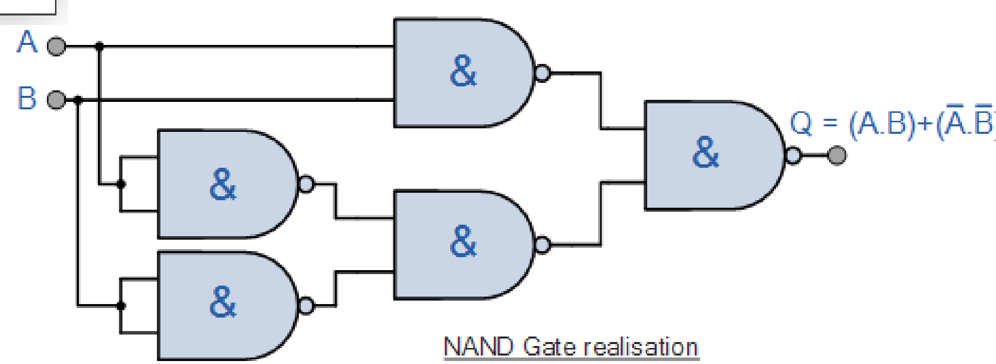
"Compliment/Inverse of XOR"

LOGIC XNOR

Symbol	Truth Table		
 2-input Ex-NOR Gate	B	A	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	1
Boolean Expression $Q = \overline{A \oplus B}$	Read if A **AND** B the **SAME** gives Q		

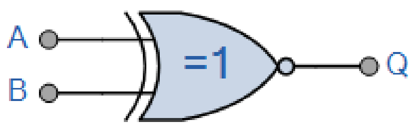
$$Q = \overline{AB} + AB$$





NAND Gate realisation

More Than 2-Inputs?

<table>
<tr><th>Symbol</th><th colspan="3">Truth Table</th></tr>
<tr><td rowspan="5">

2-input Ex-NOR Gate</td><td>B</td><td>A</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>Boolean Expression Q = $\overline{A \oplus B}$</td><td colspan="3">Read if A AND B the SAME gives Q</td></tr>
</table>

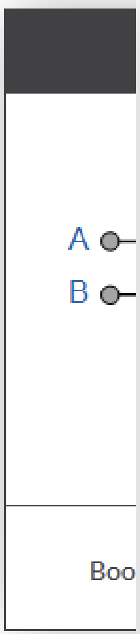
$$Q = \overline{AB} + AB$$

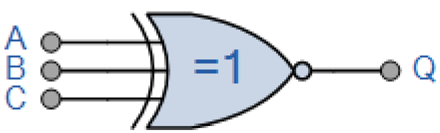
"when both A AND B are the SAME"

"Even Function"

"Mod-2-SUM"

More Than 2-Inputs?





3-input Ex-NOR Gate

Symbol	Truth Table			
	C	B	A	Q
	0	0	0	1
	0	0	1	0
	0	1	0	0
3-input Ex-NOR Gate	0	1	1	1
	1	0	0	0
	1	0	1	1
	1	1	0	1
	1	1	1	0
Boolean Expression $Q = \overline{A \oplus B \oplus C}$	Read as "any **EVEN** number of Inputs" gives Q			

"when both A AND B are the SAME"

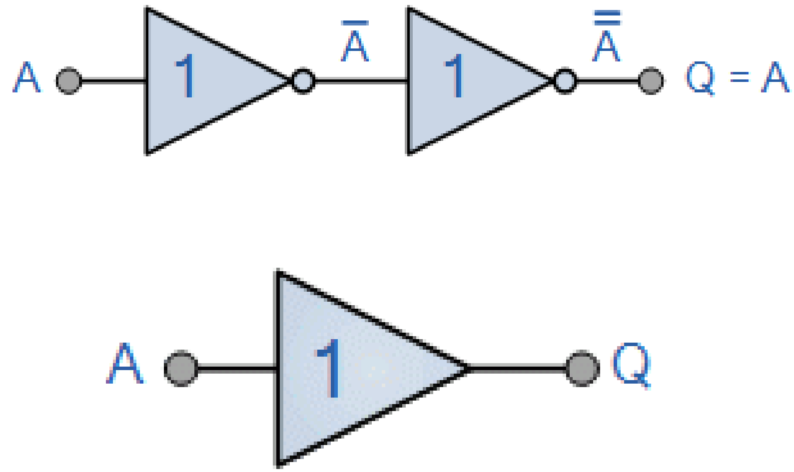
"Even Function"

"Mod-2-SUM"

$$Q = \overline{A}\,\overline{B}\,\overline{C} + AB\overline{C} + A\overline{B}C + \overline{A}BC$$

Digital Logic Gate ICs

	TTL	CMOS
AND	74LS08 Quad 2-input 74LS11 Triple 3-input 74LS21 Dual 4-input	CD4081 Quad 2-input CD4073 Triple 3-input CD4082 Dual 4-input
OR	74LS32 Quad 2-input	CD4071 Quad 2-input CD4075 Triple 3-input CD4072 Dual 4-input
NOT	74LS04 Hex Inverting NOT Gate 74LS14 Hex Schmitt Inverting NOT Gate 74LS1004 Hex Inverting Drivers	CD4009 Hex Inverting NOT Gate CD4069 Hex Inverting NOT Gate
NAND	74LS00 Quad 2-input 74LS10 Triple 3-input 74LS20 Dual 4-input 74LS30 Single 8-input	CD4011 Quad 2-input CD4023 Triple 3-input CD4012 Dual 4-input
NOR	74LS02 Quad 2-input 74LS27 Triple 3-input 74LS260 Dual 4-input	CD4001 Quad 2-input CD4025 Triple 3-input CD4002 Dual 4-input
XOR	74LS86 Quad 2-input	CD4030 Quad 2-input
XNOR	74LS266 Quad 2-input	CD4077 Quad 2-input

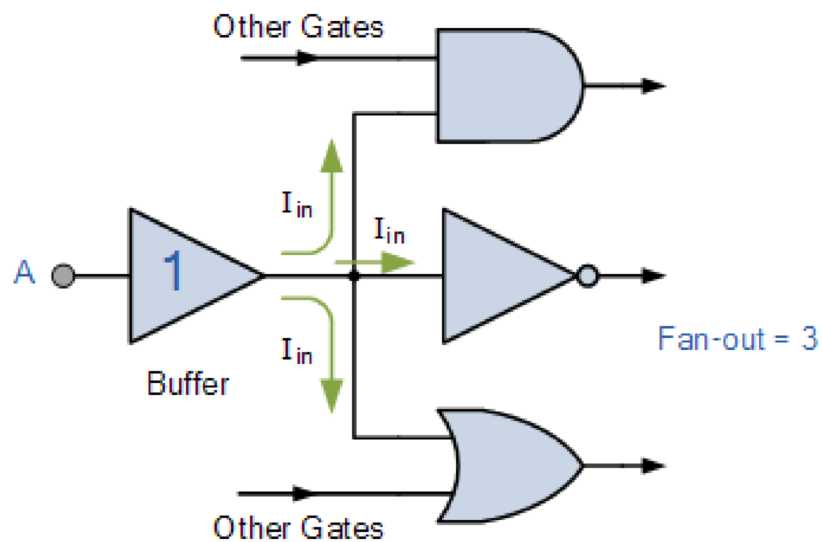


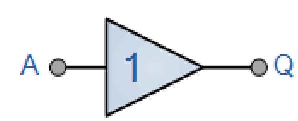
DIGITAL BUFFER

“Q is true, only when A is true”

Digital Buffer

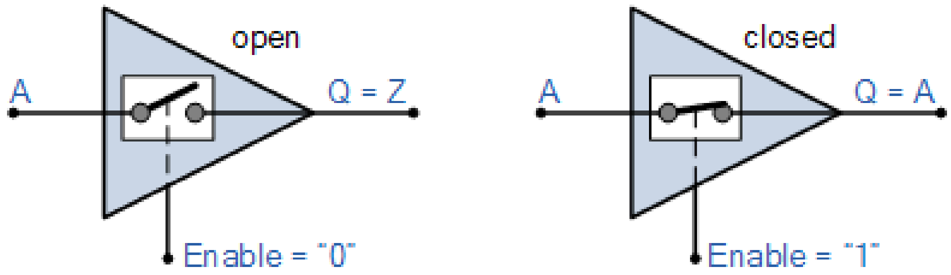
- What is a digital buffer? What is the purpose of it?
  - It can provide current amplification in a digital circuit to drive output loads, such as relays, solenoids and lamps (called “fan-out”)
  - It can be used to isolate other gates or circuit stages from each other preventing the impedance of one circuit from affecting the impedance of another



Symbol	Truth Table	
	A	Q
 The Digital Buffer	0	0
	1	1
Boolean Expression Q = A	Read as: A gives Q	

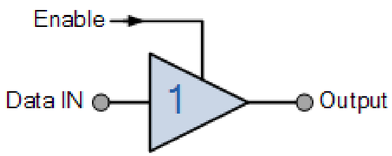
Tri-State Buffer

- A digital buffer with a control (enable) switch
  - Input: 0 or 1
  - Output: when EN=1 → 0 or 1 depending on the input, and when EN=0 → "Hi-Z" (output off, no logic, high-impedance disconnect)



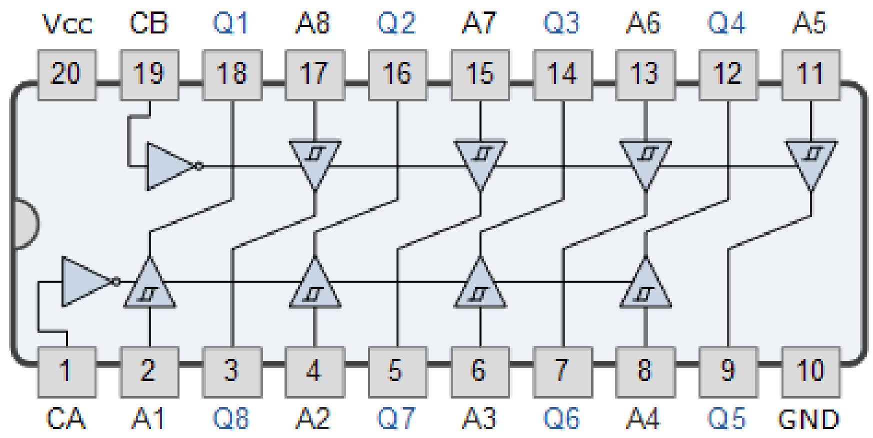
- Why "Hi-Z" is important?
  - There is a limit to the number of inputs and outputs that can be connected together in digital elements, e.g., a buffer has a fan-out limit of about 20

Types of Tri-State Buffers

Symbol	Truth Table		
	Enable	IN	OUT
 Tri-state Buffer	0	0	Hi-Z
	0	1	Hi-Z
	1	0	0
	1	1	1
Read as Output = Input if Enable is equal to “1”			

Active-high Tri-state Buffer

74LS244 Octal Tri-state Buffer

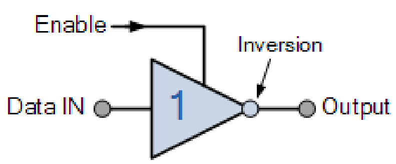


Types of Tri-State Buffers

Active-high Tri-state Buffer

Symbol	Truth Table		
D			

Active-high inverting tri-state buffer

Symbol	Truth Table		
 Inverting Tri-state Buffer	Enable	IN	OUT
	0	0	Hi-Z
	0	1	Hi-Z
	1	0	1
	1	1	0
Read as Output = Inverted Input if Enable equals "1"			

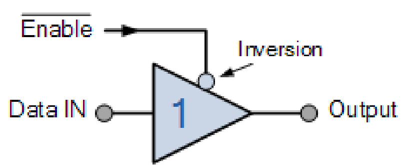
Types of Tri-State Buffers

Active-high Tri-state Buffer

Active-high inverting tri-state buffer

Active-low Tri-state Buffer

Symbol	Truth Table		
	Enable	IN	OUT
	0	0	0
	0	1	1
	1	0	Hi-Z
	1	1	Hi-Z



Tri-state Buffer

Read as Output = Input if Enable is NOT equal to "1"

Types of Tri-State Buffers

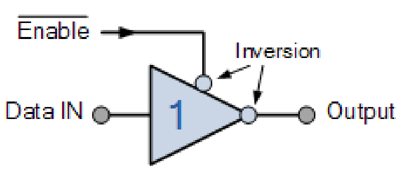
Active-high Tri-state Buffer

Active-high inverting tri-state buffer

Active-low Tri-state Buffer

Active-low Inverting Tri-state Buffer

Symbol	Truth Table

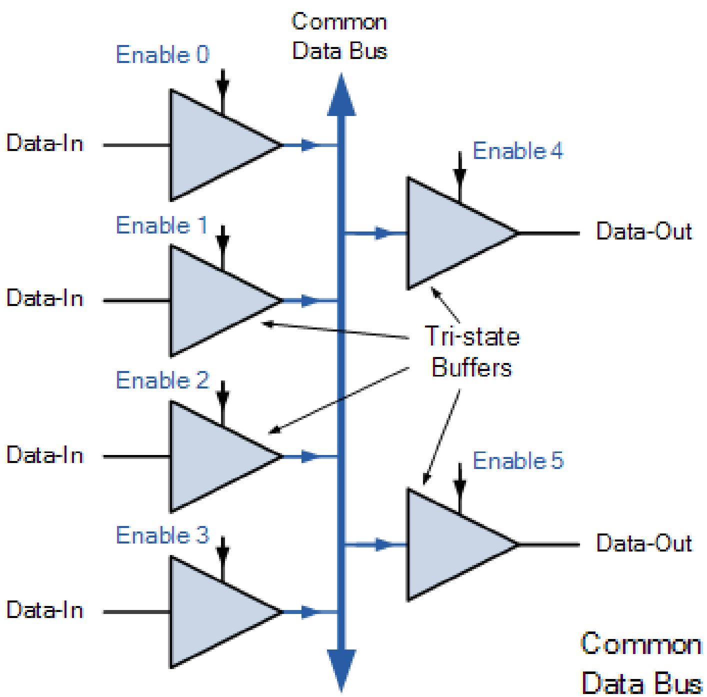


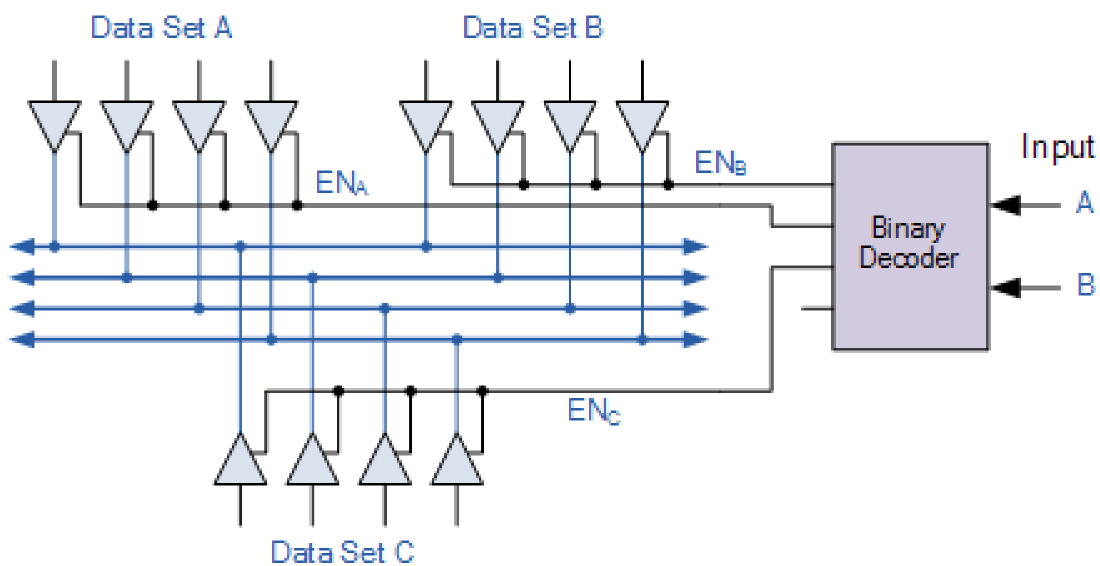
Inverting Tri-state Buffer

Enable	IN	OUT
0	0	1
0	1	0
1	0	Hi-Z
1	1	Hi-Z

Read as Output = Inverted Input if Enable is **NOT** equal to "1"

Application: Data Bus Control





Activity

- Develop an Arduino based system to discover the truth table (logic function) of an unknown logic element. State any assumptions you make.

This was discussed during the lecture