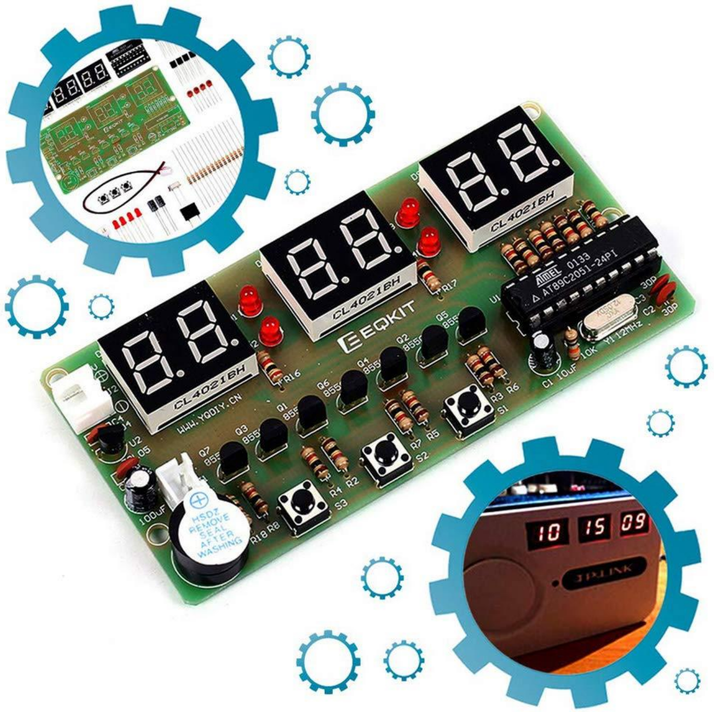


Dr. Hiran Ekanayake

DIGITAL ELECTRONIC FUNDAMENTALS – PART 2

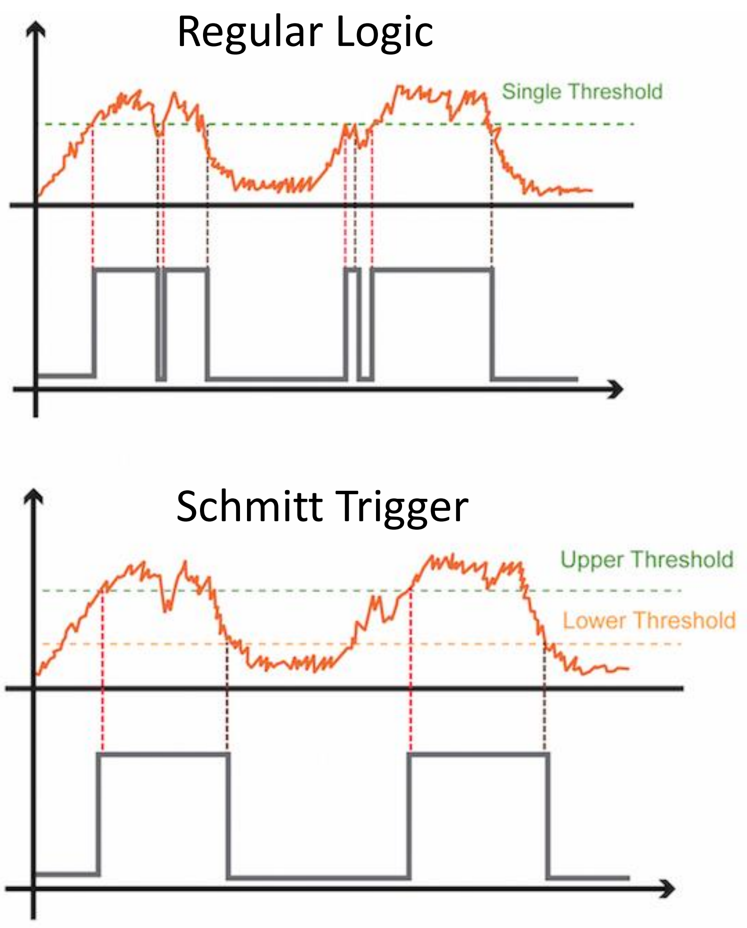
Lesson Outline

- 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

SCHMITT TRIGGER

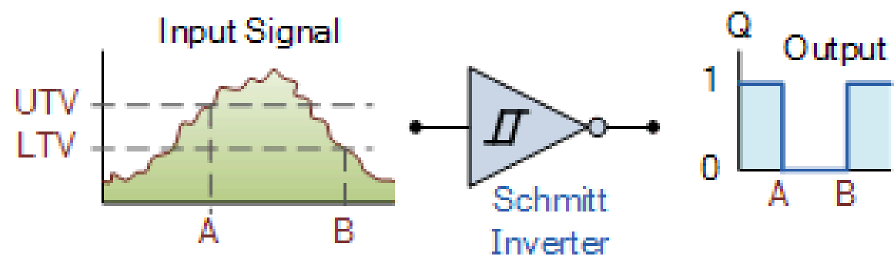
Schmitt Trigger

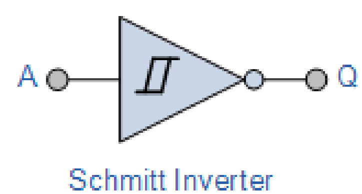
- Briefly explain how a **Schmitt inverter** minimizes the effects of noise in an input signal.
  - A Schmitt Inverter is designed to operate or switch state when its input signal goes above an "Upper Threshold Voltage" or UTV limit in which case the output changes and goes "LOW" and will remain in that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

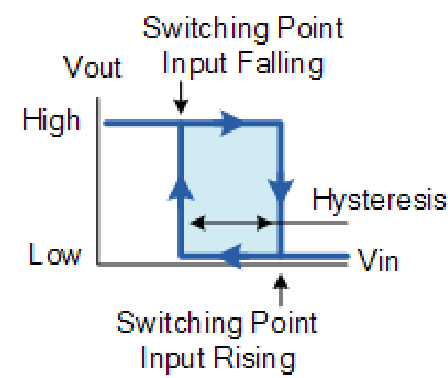


Schmitt Trigger

- 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”.

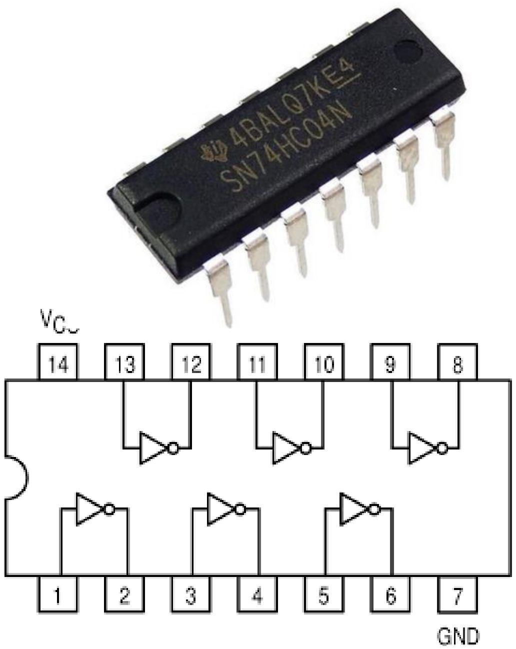




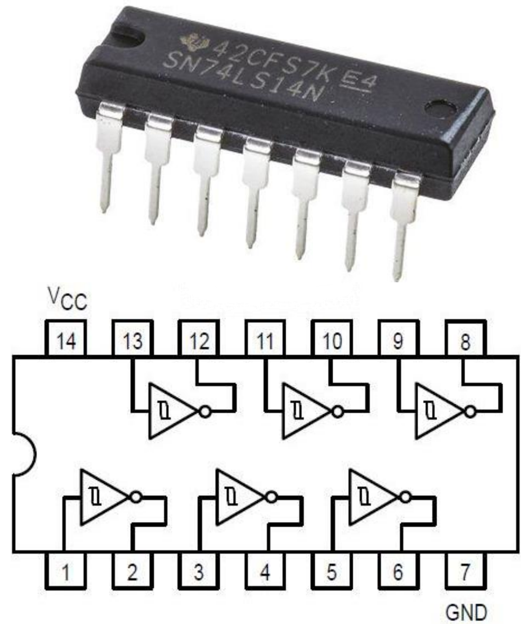


It provides a much cleaner and faster “ON/OFF” switching output signal

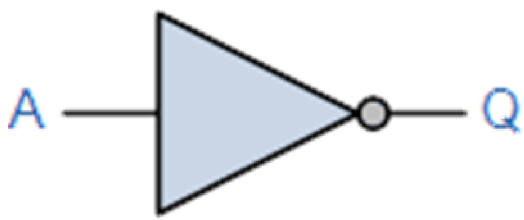
Regular vs. Schmitt Inverters

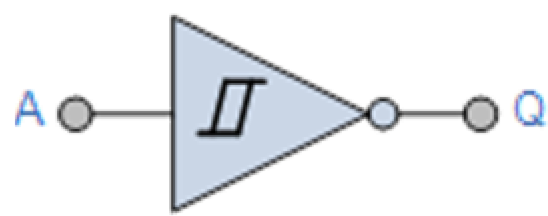


Hex Inverter

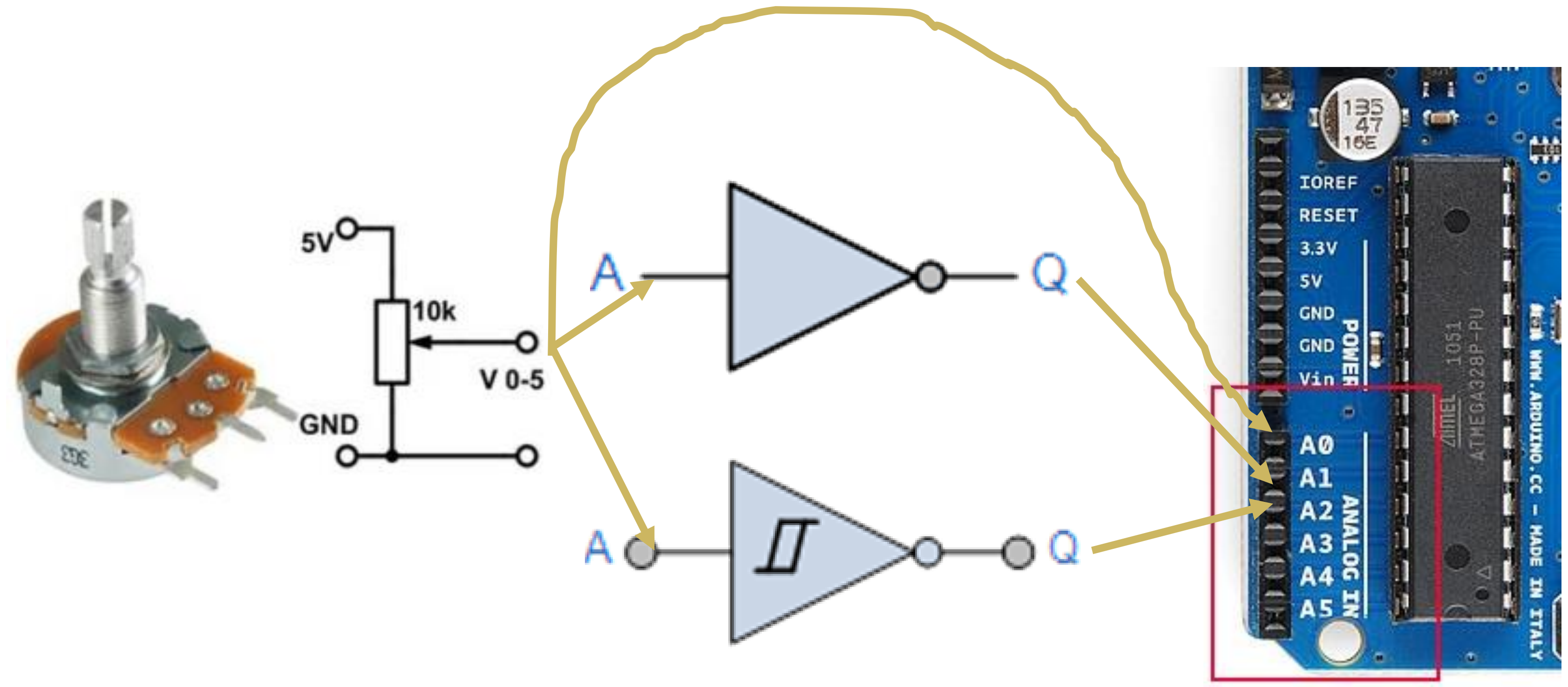


Hex Schmitt Inverter

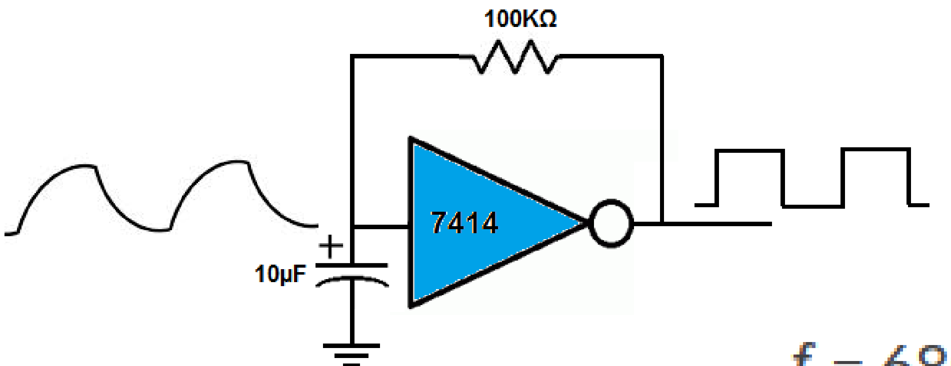


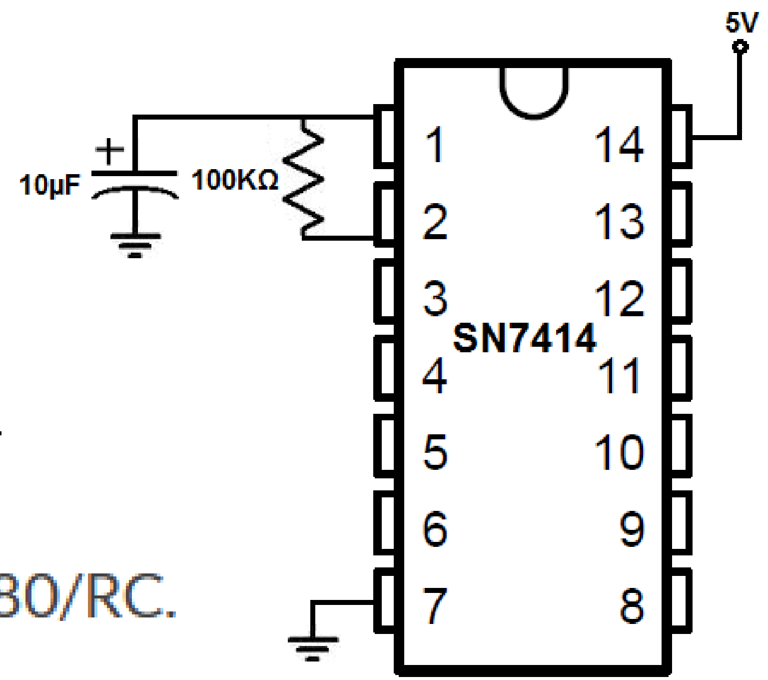


Testing a Schmitt Inverter 1



Testing a Schmitt Inverter 2

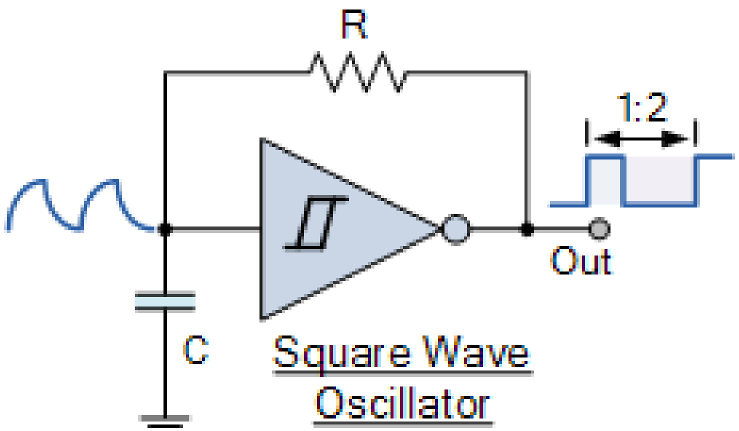


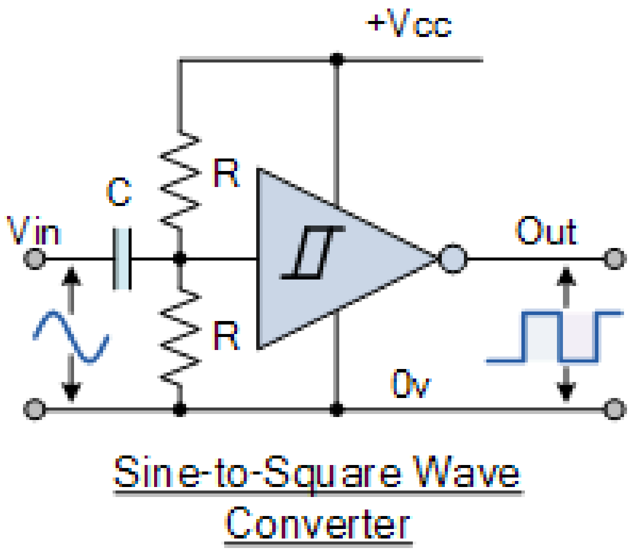


$f = 680/RC.$

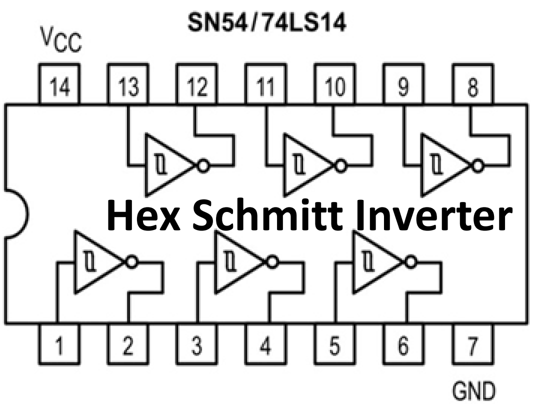
Can you build a capacitor meter using a Schmitt inverter and Arduino?

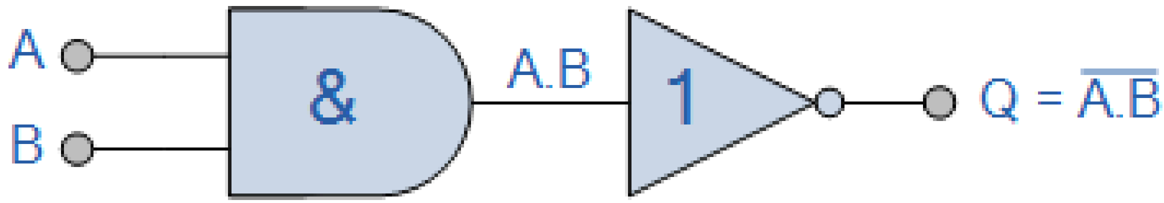
Applications of Schmitt Inverter



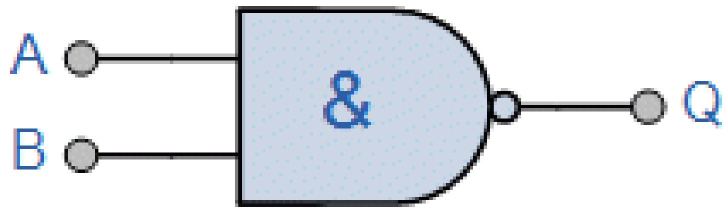


$f = 680/RC.$





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

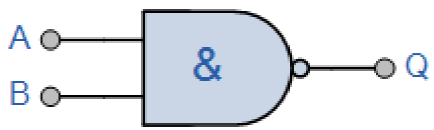


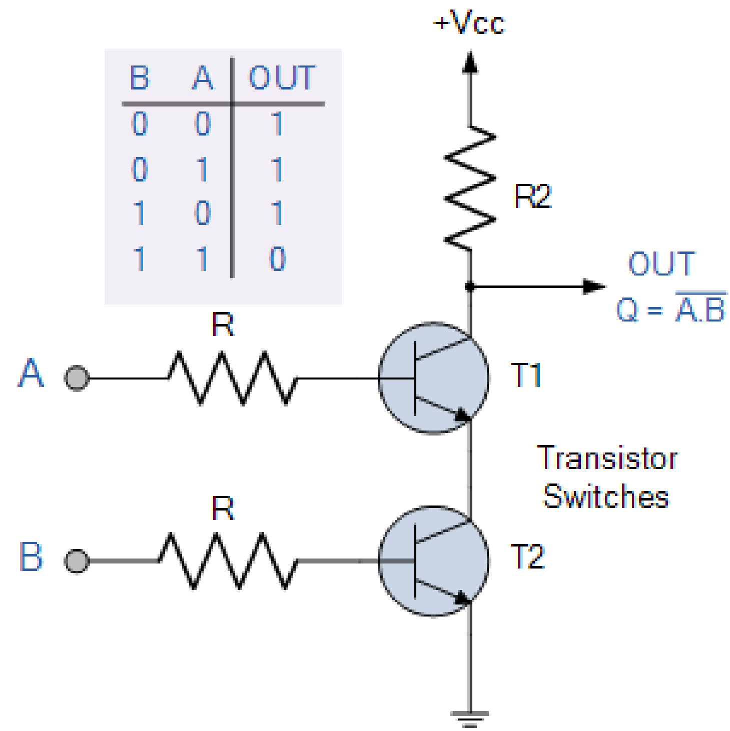
LOGIC NAND

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

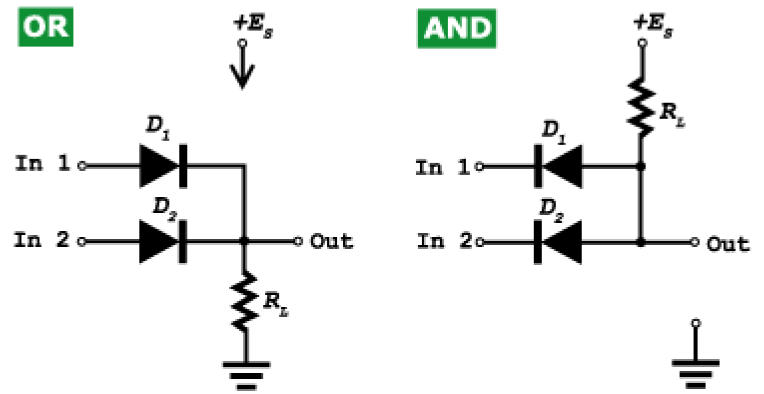
"Compliment/Inverse of AND"

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		



DDL, DTL, TTL, etc.



In 1	In 2	Out
0	0	0
0	1	1
1	0	1
1	1	1

0 - 0V
1 - Es

In 1	In 2	Out
0	0	0
0	1	0
1	0	0
1	1	1

Fig. 2.1 p-n-junction diode's OR and AND gates.

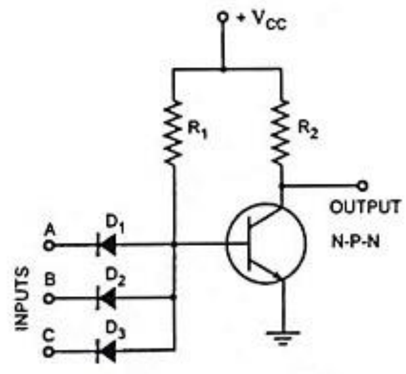
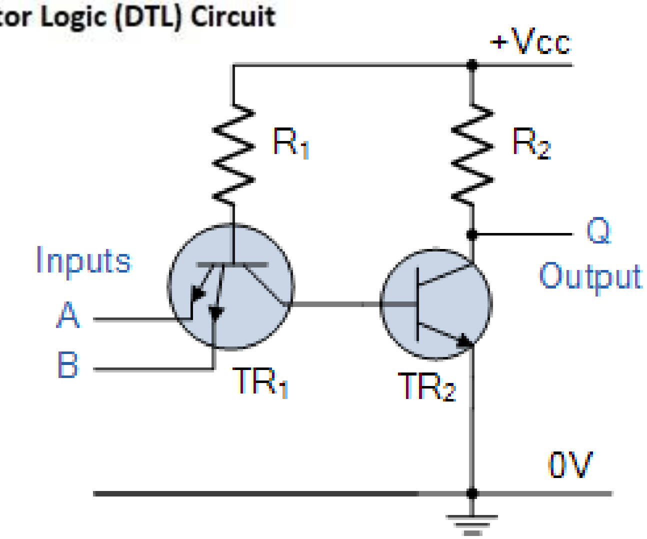


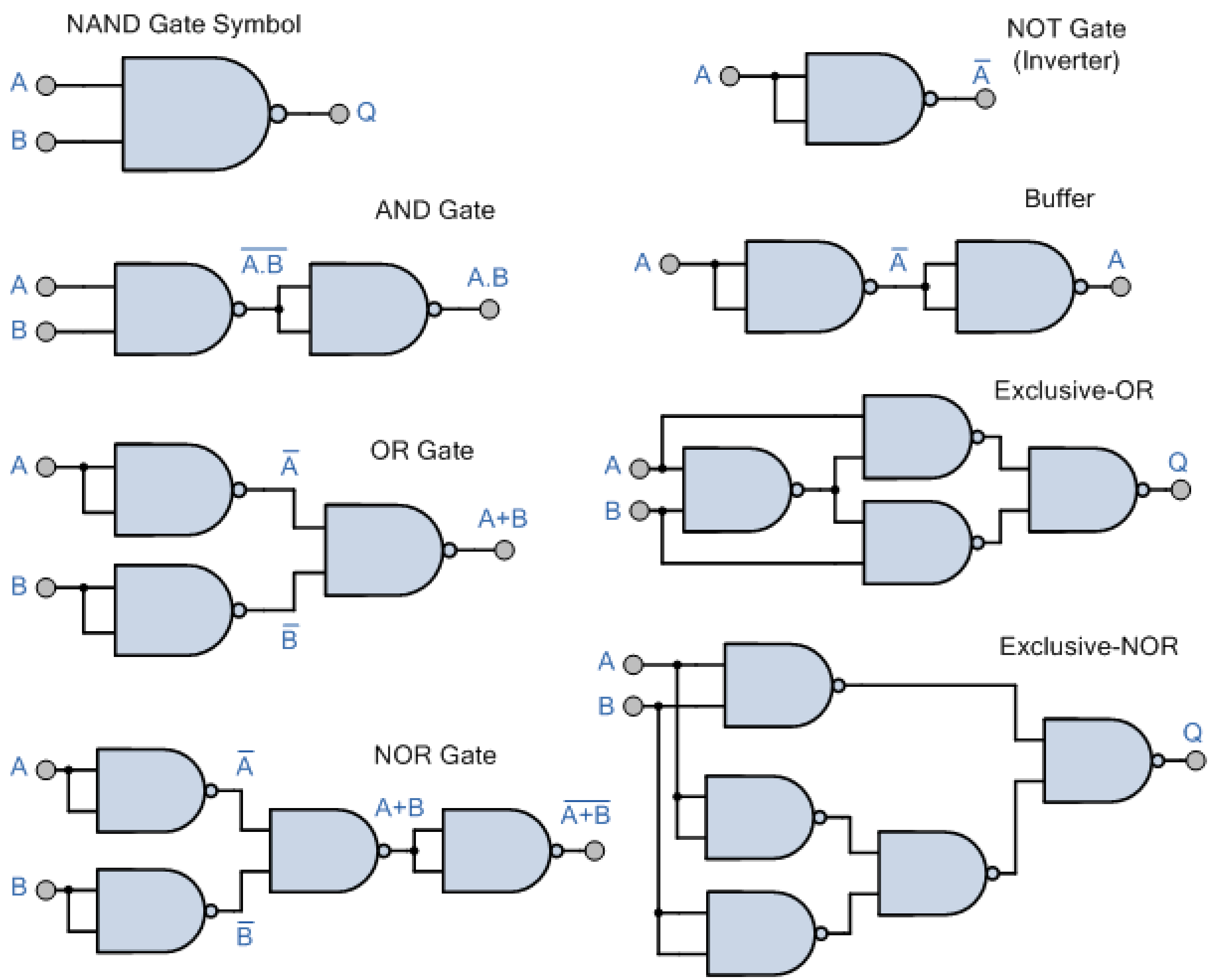
Fig. 47.2 Diode Transistor Logic (DTL) Circuit

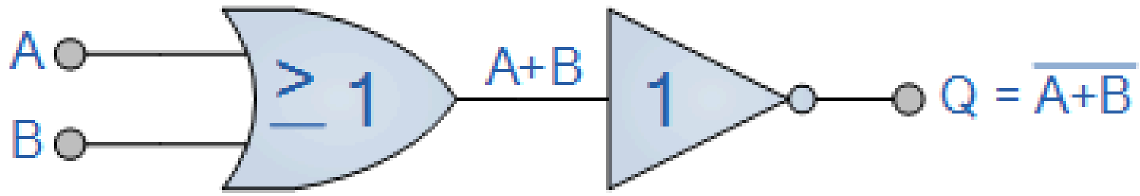


Sheffer Stroke Function

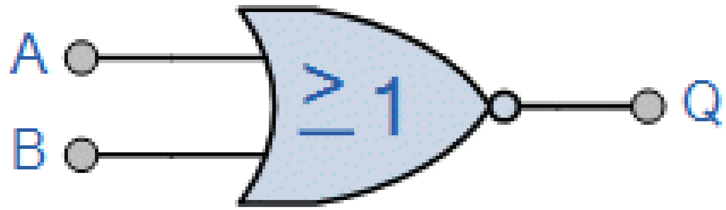
- Logic NAND function is sometimes known as the Sheffer Stroke Function, denoted by $A|B$ or $A\uparrow B$
- Why? NAND is functionally complete, i.e., it can be used to produce any other type of logic gate function; thus, it is considered as a "universal" gate

Other Logic Functions Using NAND Gate





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

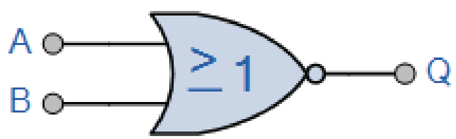


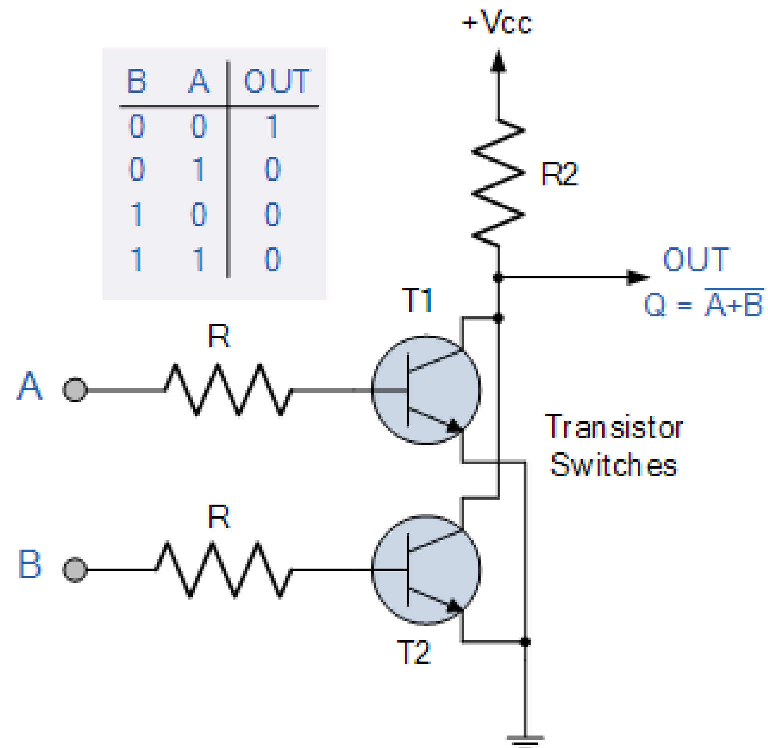
LOGIC NOR

"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

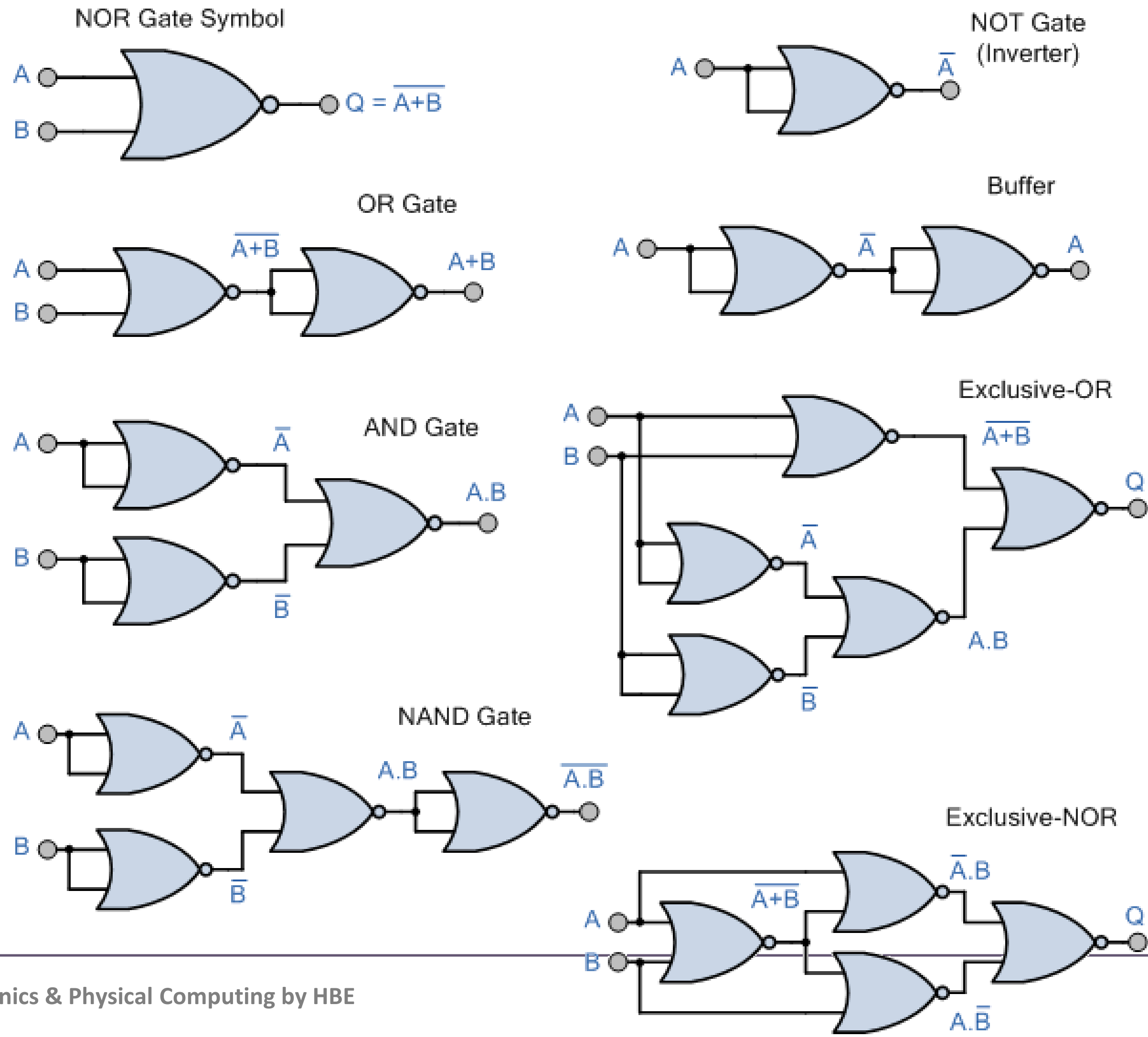
Symbol	Truth Table		
 2-input NOR Gate	B	A	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	0
Boolean Expression $Q = \overline{A+B}$	Read as A OR B gives NOT Q		

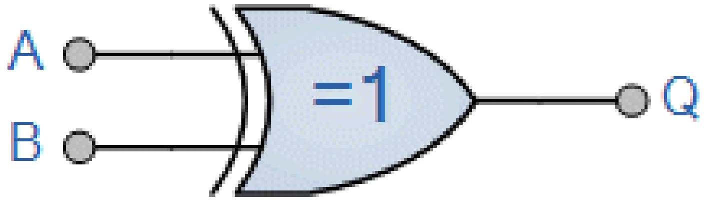


Pierce Function

- 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

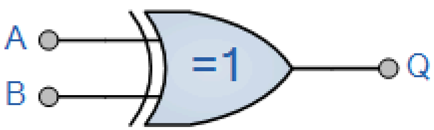


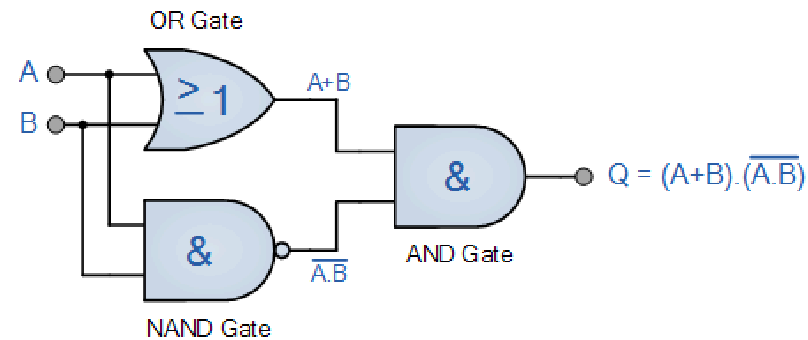


EXCLUSIVE-OR (XOR)

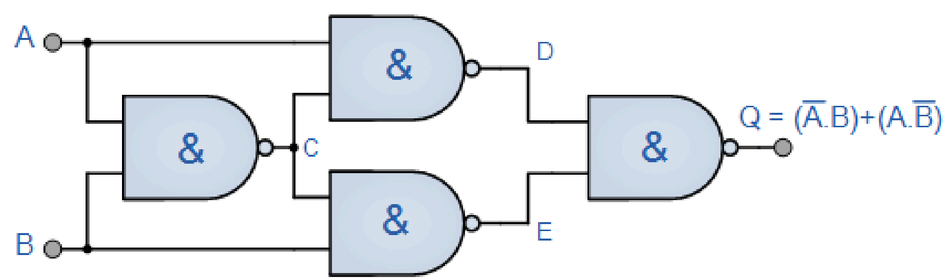
"The Logic XOR Function output is only true when its two input terminals are at different logic levels with respect to each other."

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		

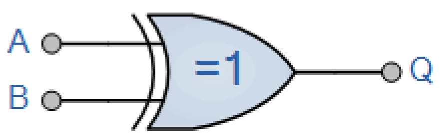


$$Q = (A \oplus B) = \bar{A}.B + A.\bar{B}$$



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		

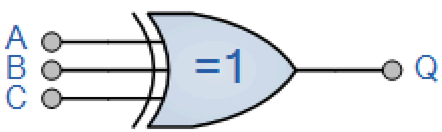
$$Q = (A \oplus B) = \overline{A}.B + A.\overline{B}$$

"A OR B but NOT both"

"Odd Function"

"Mod-2-SUM"

More Than 2-Inputs?

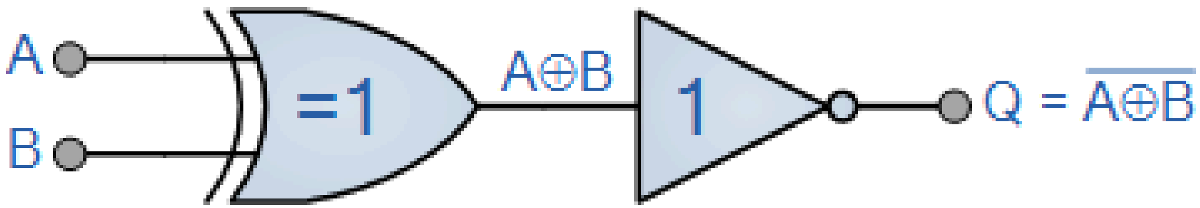
Symbol	Truth Table			
 3-input Ex-OR Gate	C	B	A	Q
	0	0	0	0
	0	0	1	1
	0	1	0	1
	0	1	1	0
	1	0	0	1
	1	0	1	0
	1	1	0	0
	1	1	1	1
Boolean Expression $Q = A \oplus B \oplus C$	"Any **ODD** Number of Inputs" gives Q			

"A OR B but NOT both"

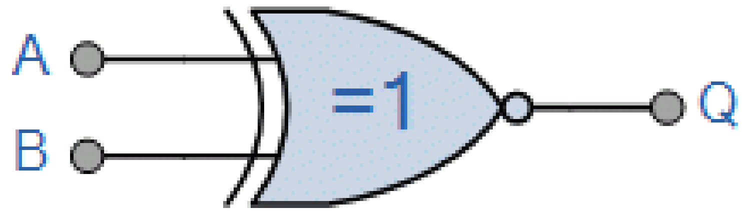
"Odd Function"

"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

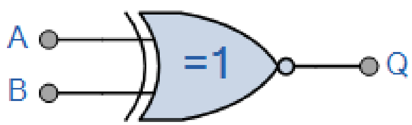


EXCLUSIVE-NOR (XNOR)

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

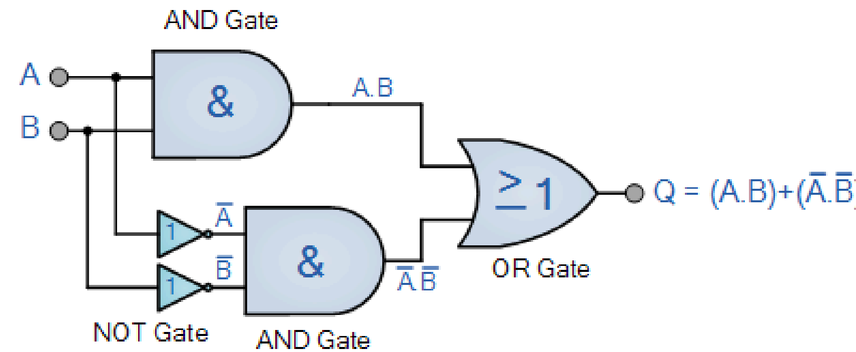
"Compliment/Inverse of XOR"

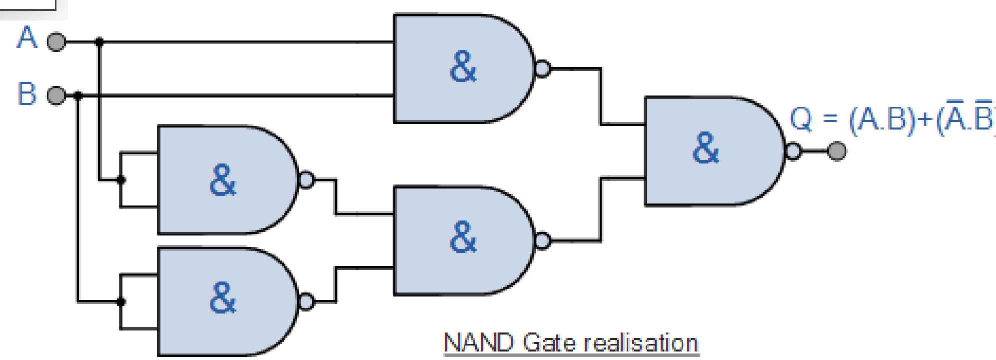
LOGIC XNOR

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

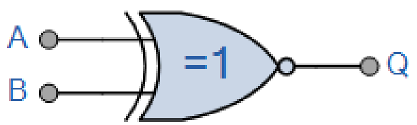
2-input Ex-NOR Gate</td><td>B</td><td>A</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>Boolean Expression Q = $\overline{A \oplus B}$</td><td colspan="3">Read if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

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





More Than 2-Inputs?



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

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

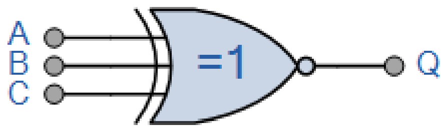
"when both A AND B are the SAME"

"Even Function"

"Mod-2-SUM"

More Than 2-Inputs?

Symbol	Truth Table			
	C	B	A	Q
	0	0	0	1
	0	0	1	0
	0	1	0	0
	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			



3-input Ex-NOR Gate

"when both A AND B are the SAME"

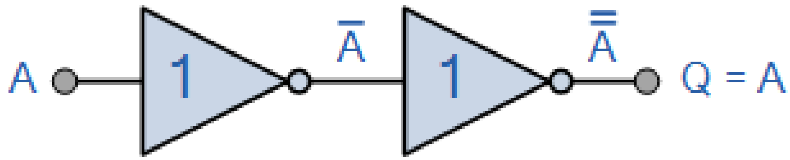
"Even Function"

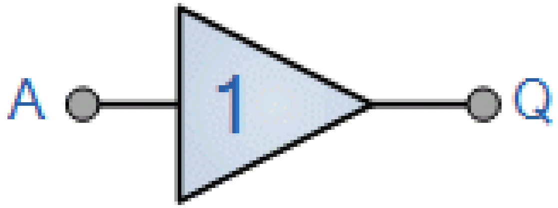
"Mod-2-SUM"

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

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



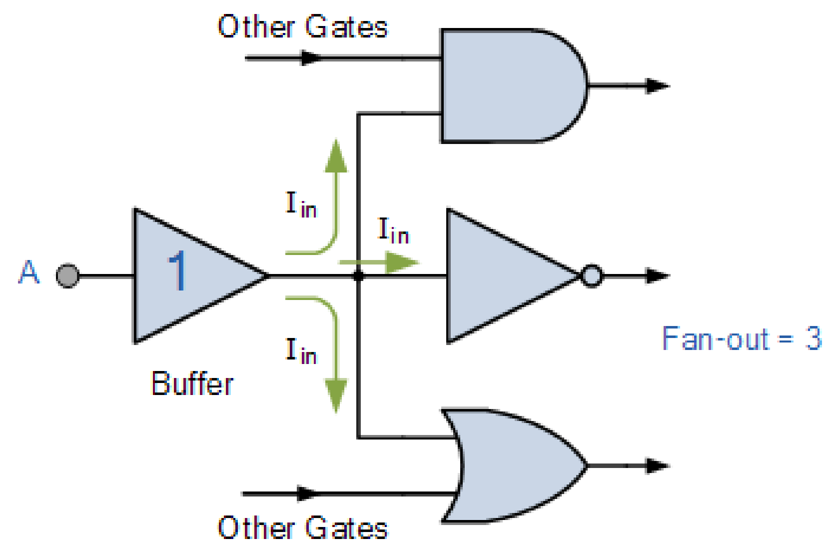


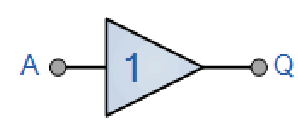
DIGITAL BUFFER

"Q is true, only when A is true"

Digital Buffer

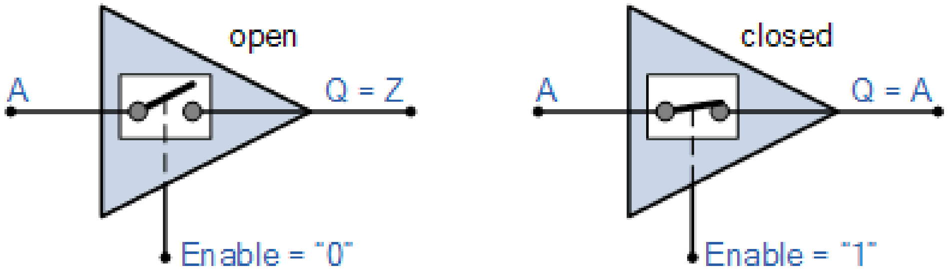
- What is a digital buffer? What is the purpose of it?
  - It can provide current amplification in a digital circuit to drive output loads, such as relays, solenoids and lamps (called “fan-out”)
  - It can be used to isolate other gates or circuit stages from each other preventing the impedance of one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



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

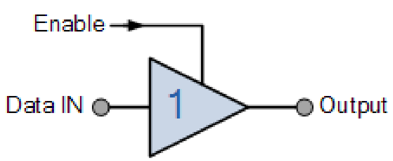
Tri-State Buffer

- 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)



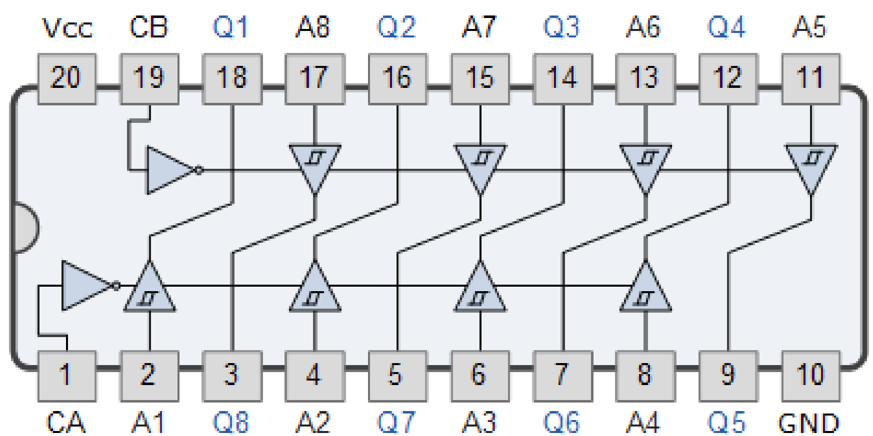
- Why “Hi-Z” is important?
  - There is a limit to the number of inputs and outputs that can be connected together in digital elements, e.g., a buffer has a fan-out limit of about 20

Types of Tri-State Buffers

Symbol	Truth Table		
 Tri-state Buffer	Enable	IN	OUT
	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



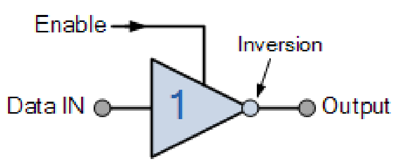
Types of Tri-State Buffers

Active-high Tri-state Buffer

Symbol	Truth Table
D	

Active-high inverting tri-state buffer

Symbol



Inverting Tri-state Buffer

Truth Table

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"

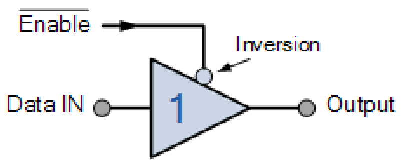
Types of Tri-State Buffers

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"

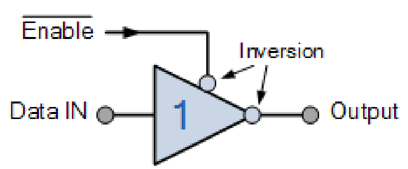
Types of Tri-State Buffers

Active-high Tri-state Buffer

Active-high inverting tri-state buffer

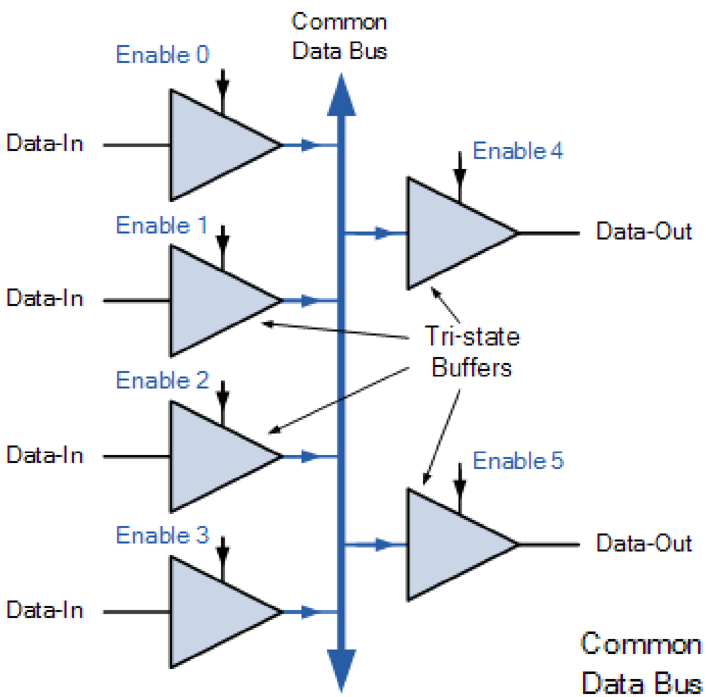
Active-low Tri-state Buffer

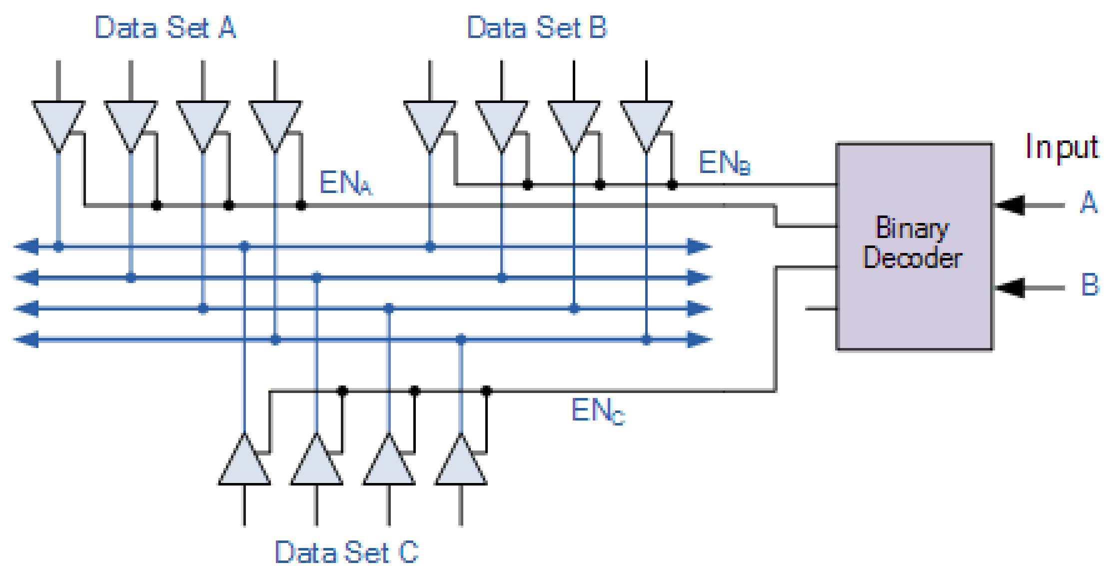
Active-low Inverting Tri-state Buffer

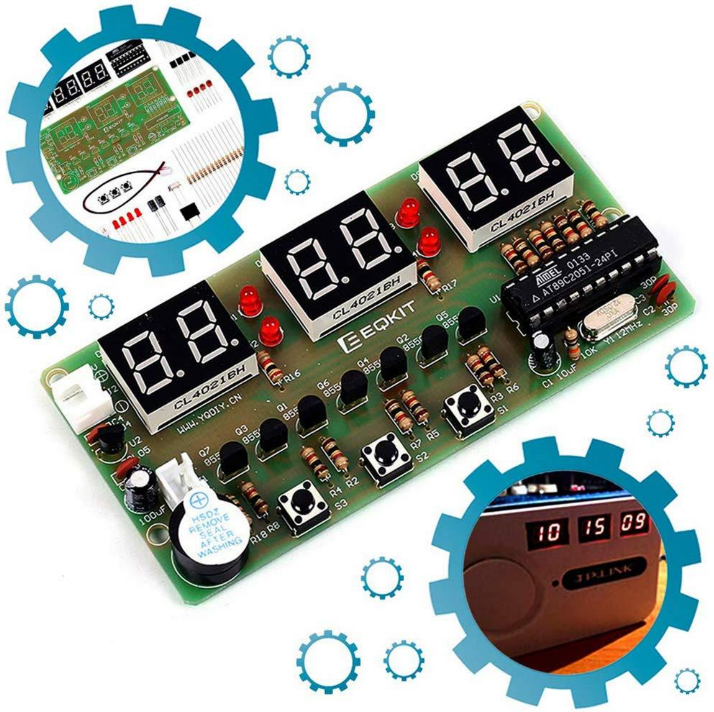
Symbol	Truth Table		
	Enable	IN	OUT
 Inverting Tri-state Buffer	0	0	1
	0	1	0
	1	0	Hi-Z
	1	1	Hi-Z

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

Application: Data Bus Control





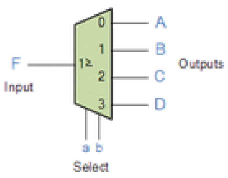


Dr. Hiran Ekanayake

COMBINATIONAL LOGIC CIRCUITS

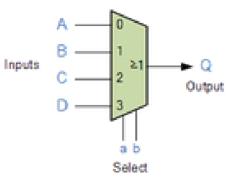
References

- Combinational Logic Circuits
  - https://www.electronics-tutorials.ws/category/combination



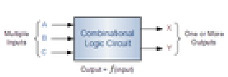
The Demultiplexer

The data distributor, known more commonly as a Demultiplexer or "Demux" for short, is the exact opposite of the Multiplexer we saw in the previous tutorial. The demultiplexer takes one single input data line and then switches it to any one of a number of individual output lines ...



The Multiplexer

Multiplexing is the generic term used to describe the operation of sending one or more analogue or digital signals over a common transmission line at different times or speeds and as such, the device we use to do just that is called a Multiplexer. The multiplexer, shortened to "...



Combinational Logic Circuits

Unlike Sequential Logic Circuits whose outputs are dependant on both their present inputs and their previous output state giving them some form of Memory. The outputs of Combinational Logic Circuits are only determined by the logical function of their current input state, logic "...

Lesson Outline

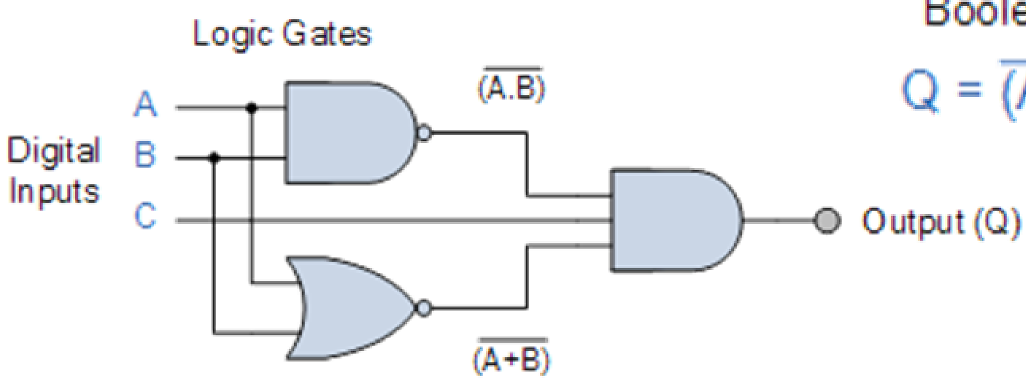
- Introduction
- Multiplexer
- Demultiplexer
- Encoder
- Decoder
- Binary Adder
- Digital Comparator
- Binary Subtractor
- ALU

Combinational Logic Circuits

- What is a combinational logic circuit?
  - A combinational logic circuit is made up from basic logic gates to produce more complicated switching circuits
  - The output of a combinational logic circuit at any instant in time depends only on the combination of its inputs

Combinational Logic Circuits

- How do you specify the function of a combinational logic circuit?
    - Using Boolean algebra
    - Using a truth table
    - Using a logic diagram



Logic Diagram

Boolean Expression

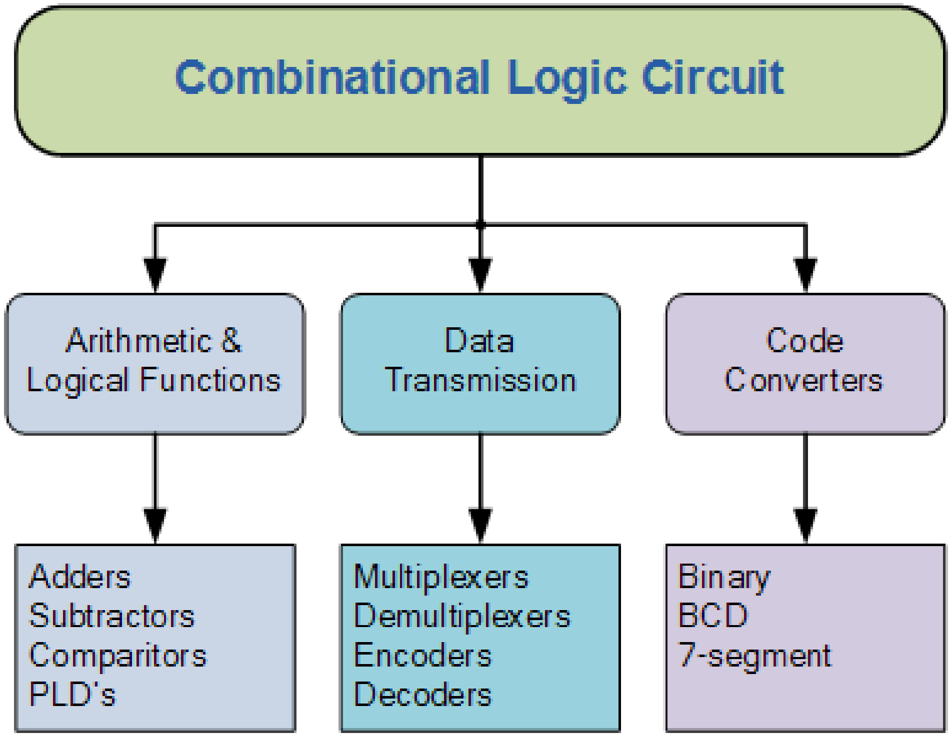
$$Q = \overline{(A.B)}.\overline{(A+B)}.C$$

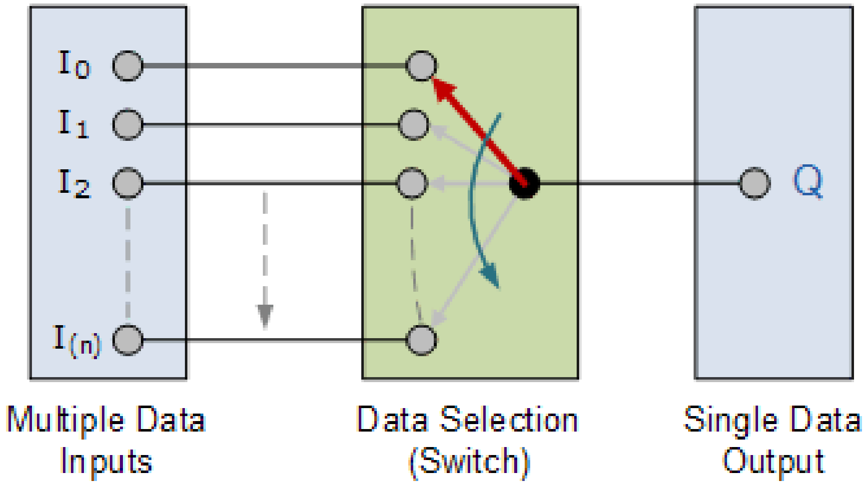
Typical Truth Table

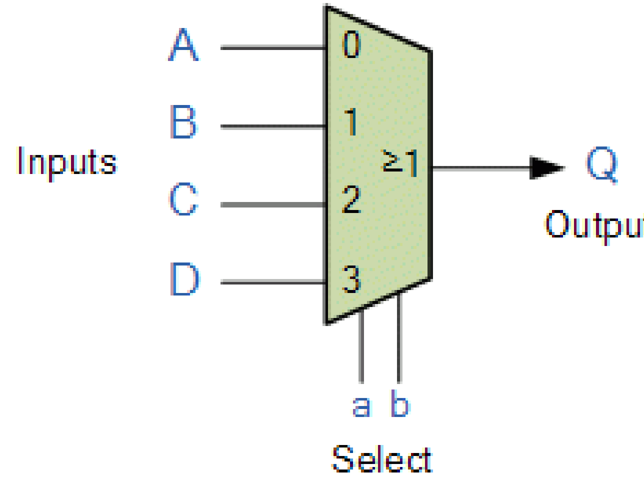
C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Combinational Logic Circuits

- What are different types of combinational logic circuits?







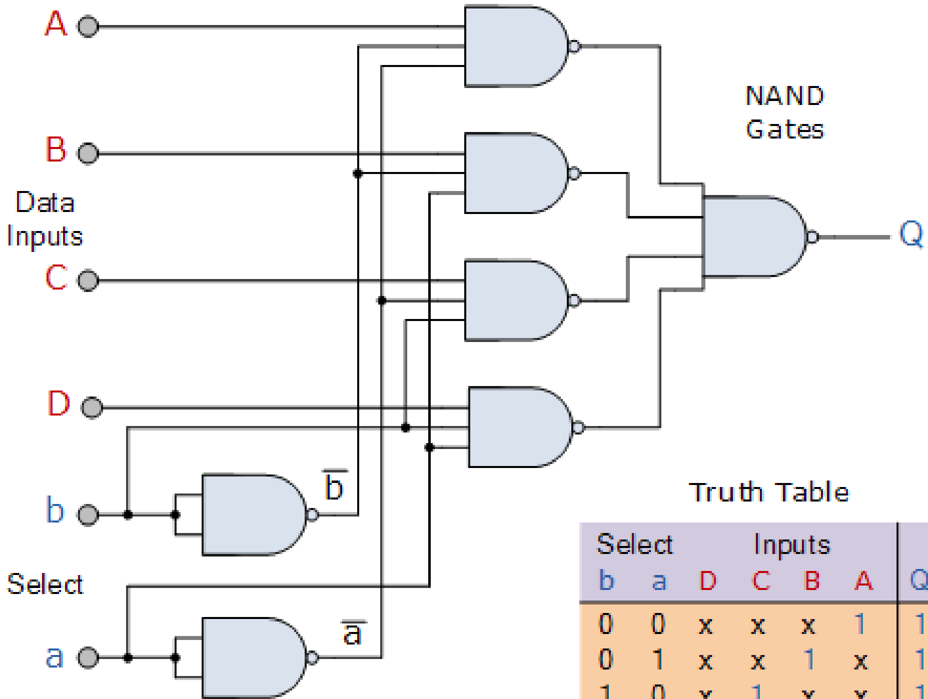
MULTIPLEXER (MUX, MPX, DATA SELECTOR)

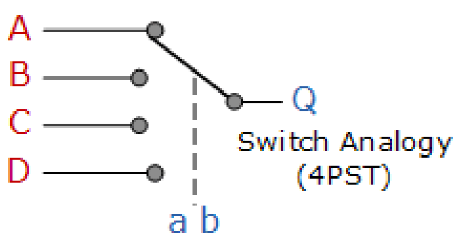
“is a combinational logic circuit designed to switch one of several input lines through to a single common output line by the application of a control signal”

Multiplexer

- What are the applications of a multiplexer?
- Briefly describe the operation of a multiplexer.
- How does an encoder differ from a multiplexer?
- Give the logic diagram of a 4-to-2 multiplexer.

Multiplexer 4-to-1

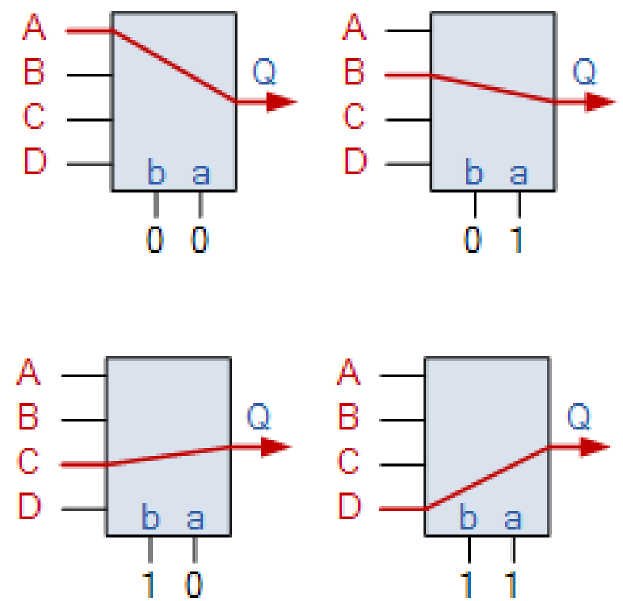




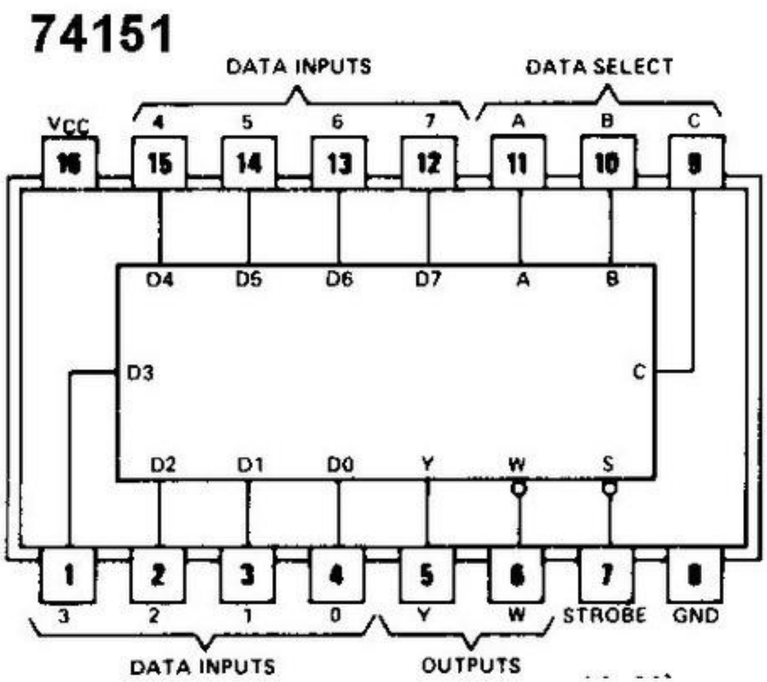
Truth Table

Select		Inputs				
b	a	D	C	B	A	Q
0	0	x	x	x	1	1
0	1	x	x	1	x	1
1	0	x	1	x	x	1
1	1	1	x	x	x	1

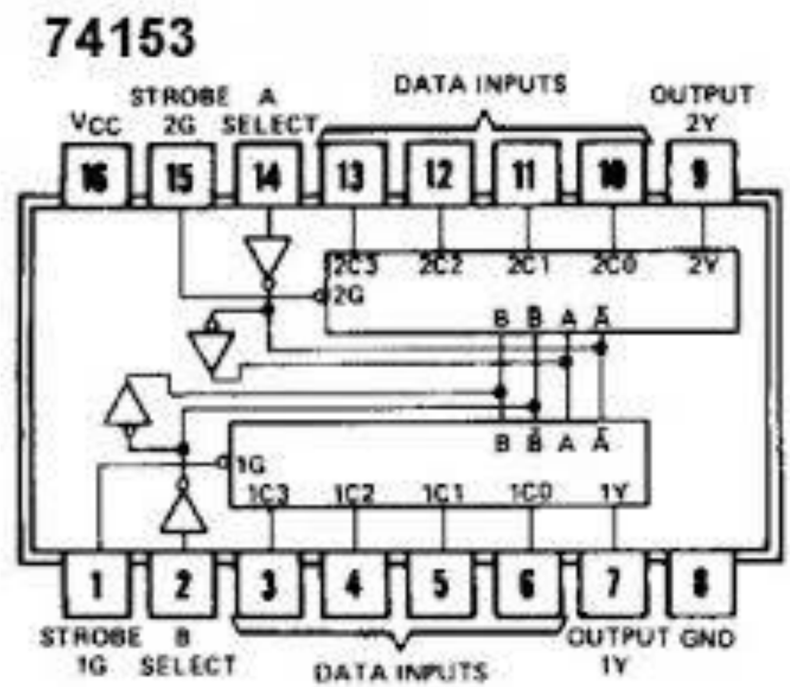
$$Q = \bar{a}\bar{b}A + a\bar{b}B + \bar{a}bC + abD$$



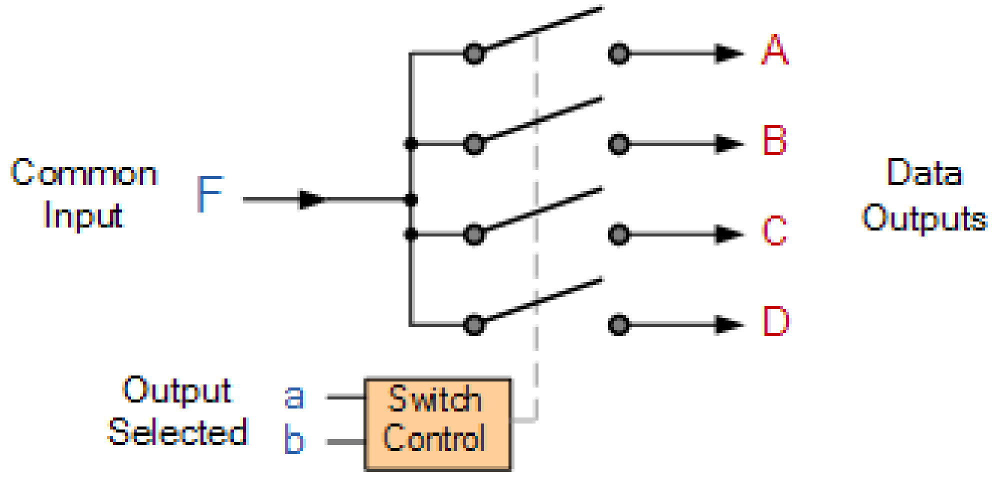
Multiplexer ICs

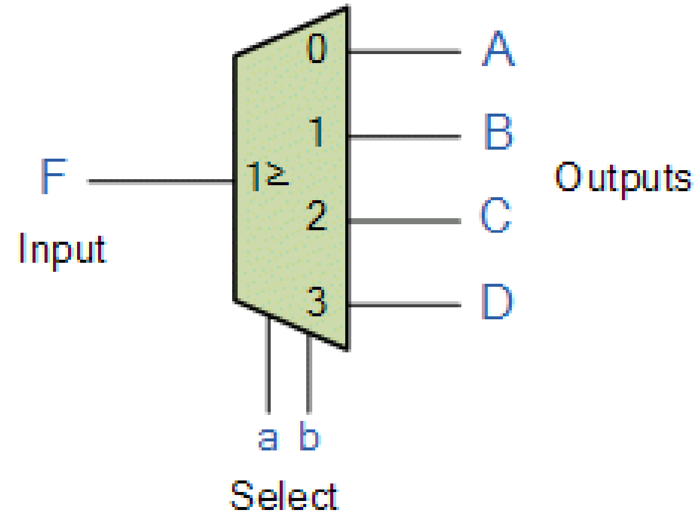


8-to-1 MUX



Dual 4-to-1 MUX





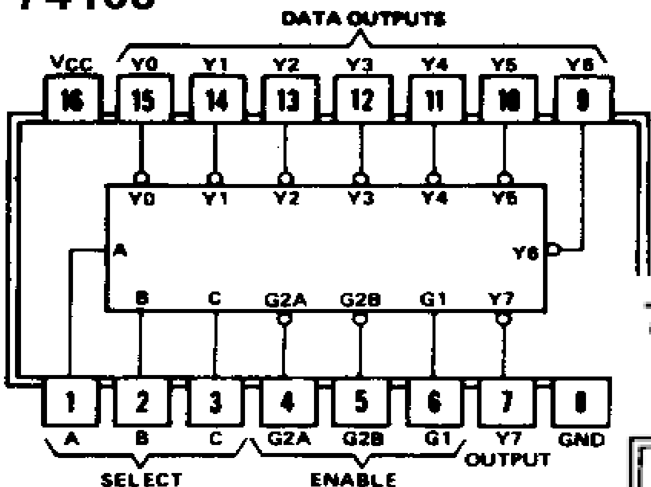
DEMULTIPLEXER (DEMUX, DATA DISTRIBUTOR)

Exact opposite of the multiplexer

"takes one single input data line and then switches it to any one of a number of individual output lines one at a time"

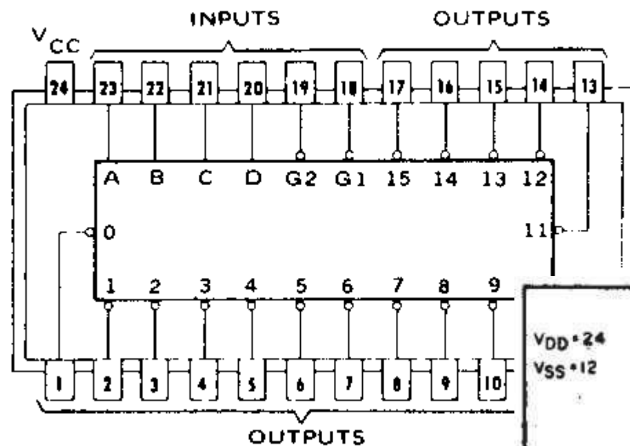
Demultiplexer ICs

74138

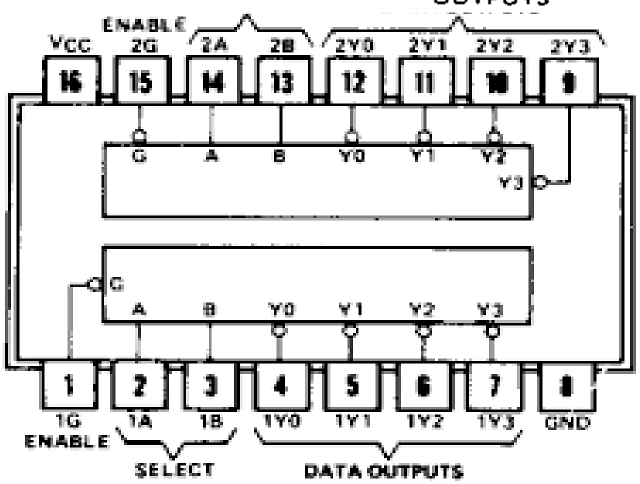


1-to-8 DeMux

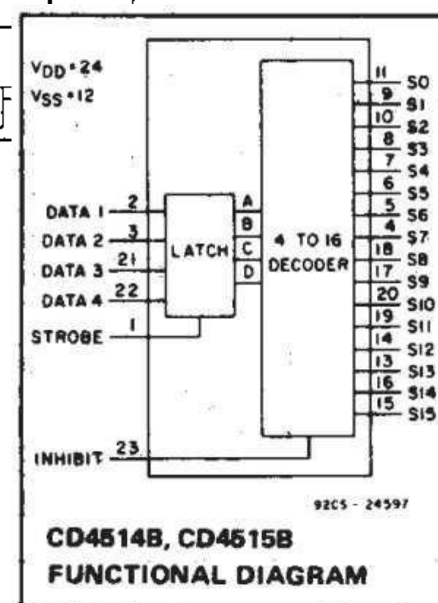
74154



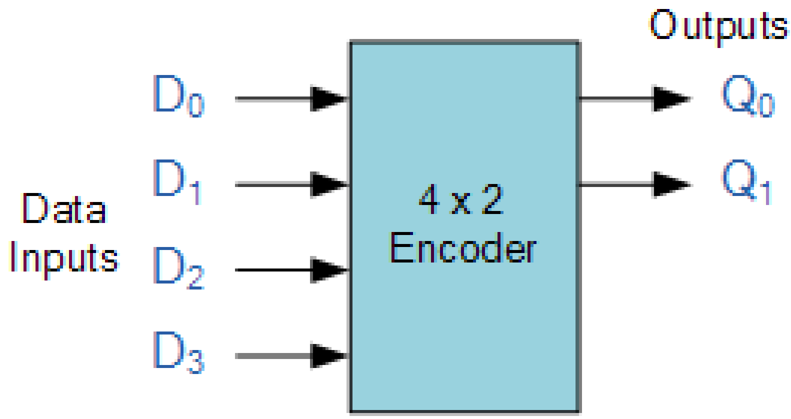
74139



Dual 1-to-4 DeMux



Demultiplexers or decoders?



Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

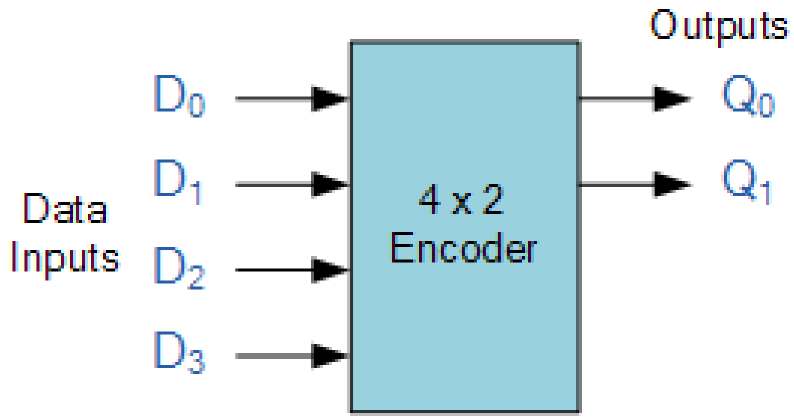
BINARY ENCODER

"takes all its data inputs one at a time and then converts them into a single encoded output"

"usually an "n-bit" binary encoder has 2^n input lines and n-bit output lines"

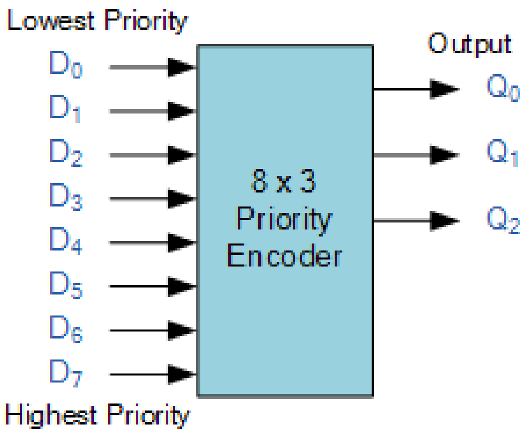
Priority Encoder (P-encoder)

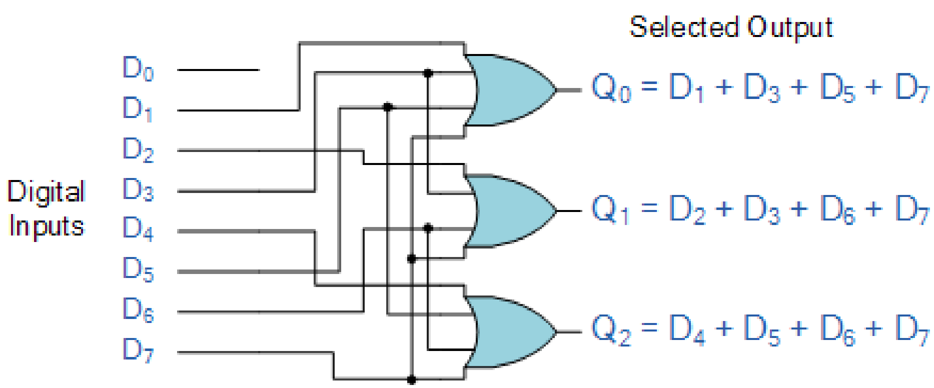
- If there is more than one input at logic level "1" at the same time, the actual output code would only correspond to the input with the highest designated priority



Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

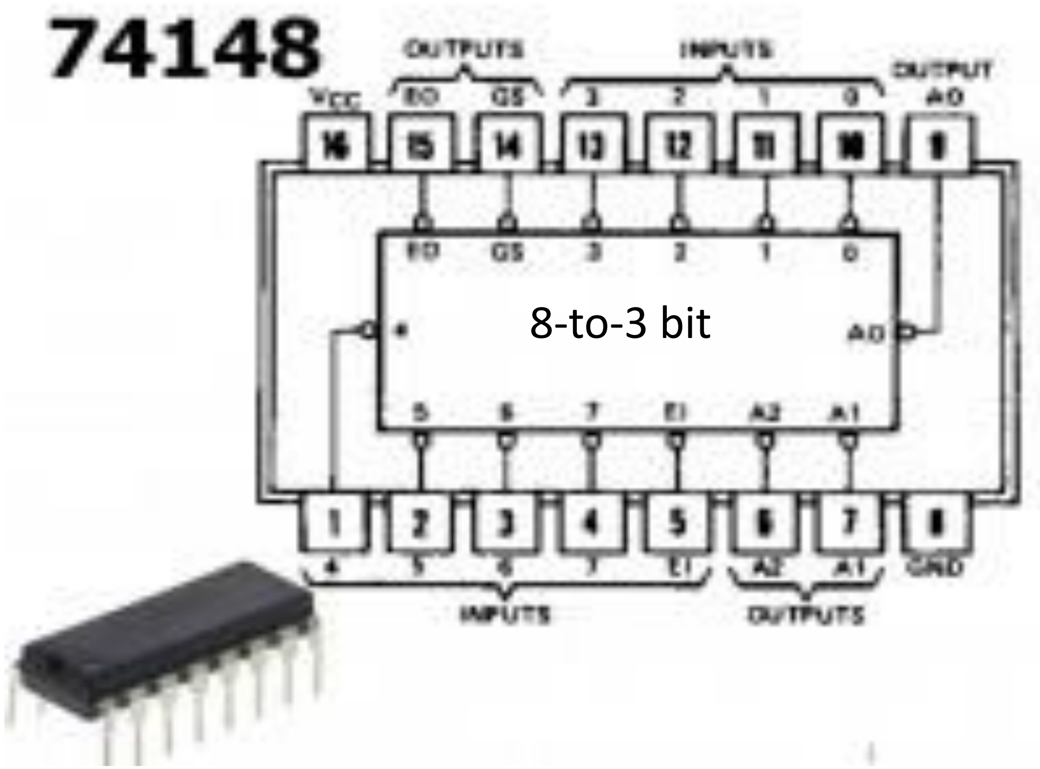
Priority Encoder



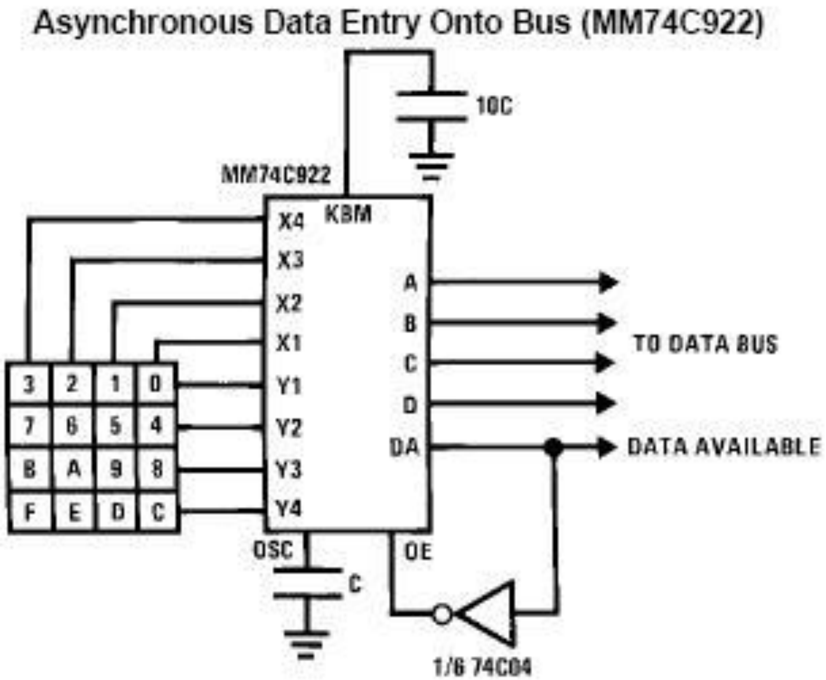


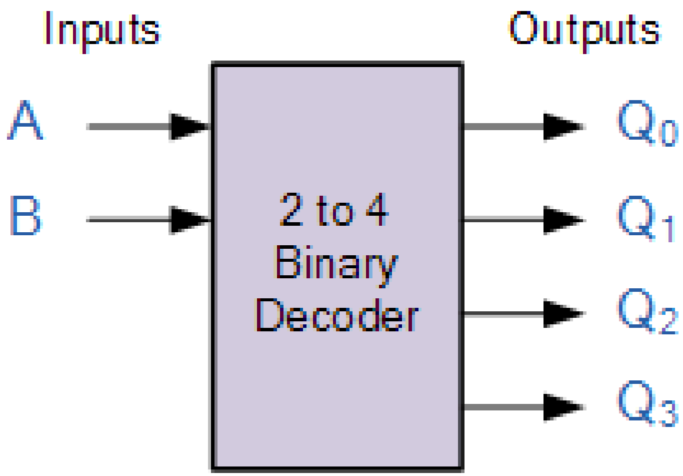
Digital Inputs								Binary Output		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	1	X	X	X	X	1	0	0
0	0	1	Lower priority bits are ignored					1	0	1
0	1	X						1	1	0
1	X	X	X	X	X	X	X	1	1	1

Priority Encoder ICs



Application: Keyboard Encoder





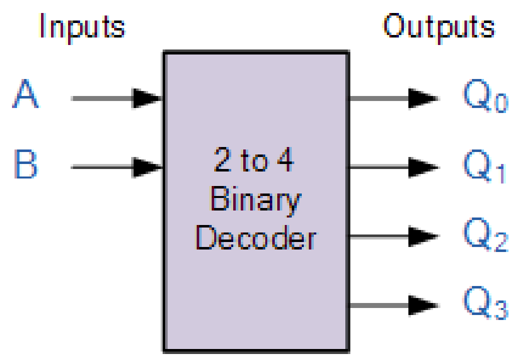
Truth Table

A	B	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

BINARY DECODER

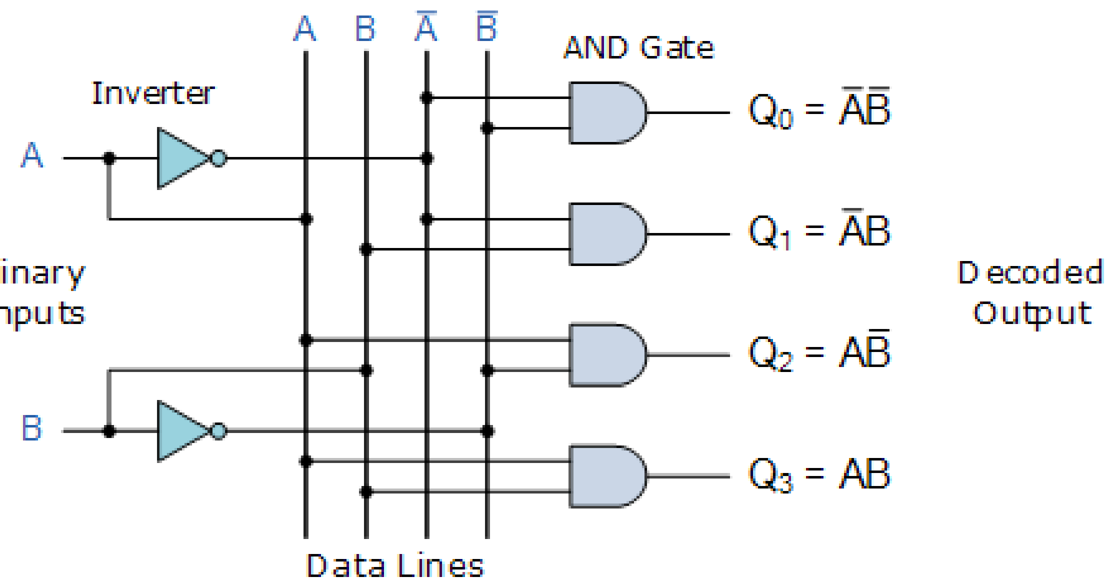
“translate or decode coded information from one format into another, so a binary decoder transforms “n” binary input signals into an equivalent code using 2^n outputs”

Binary Decoder

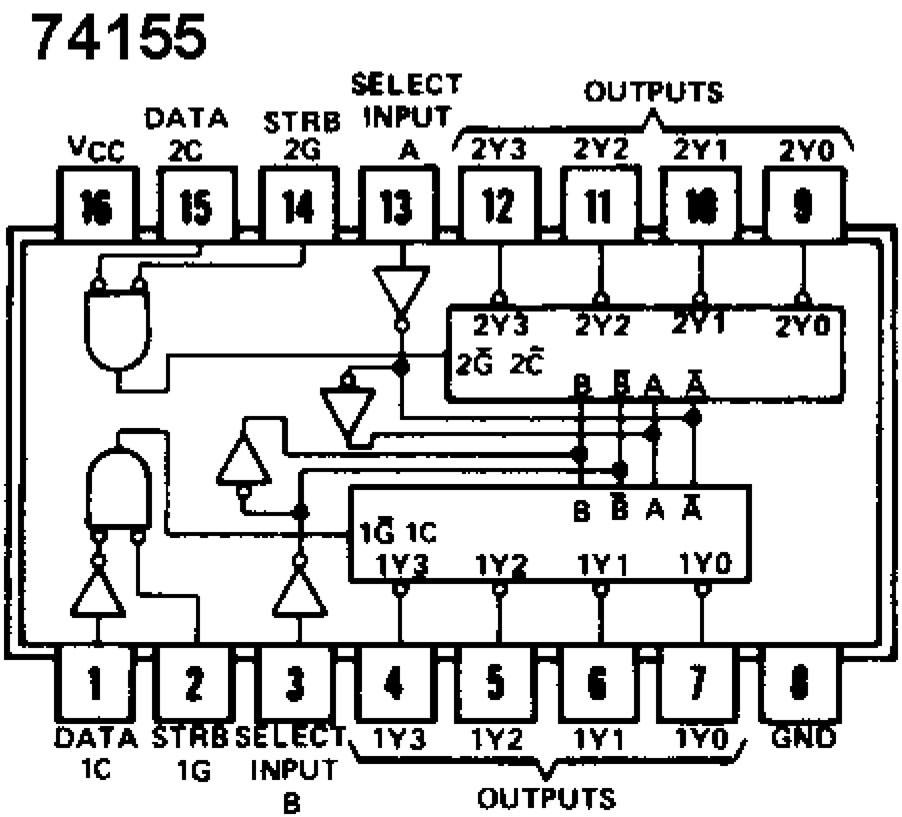


Truth Table

A	B	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



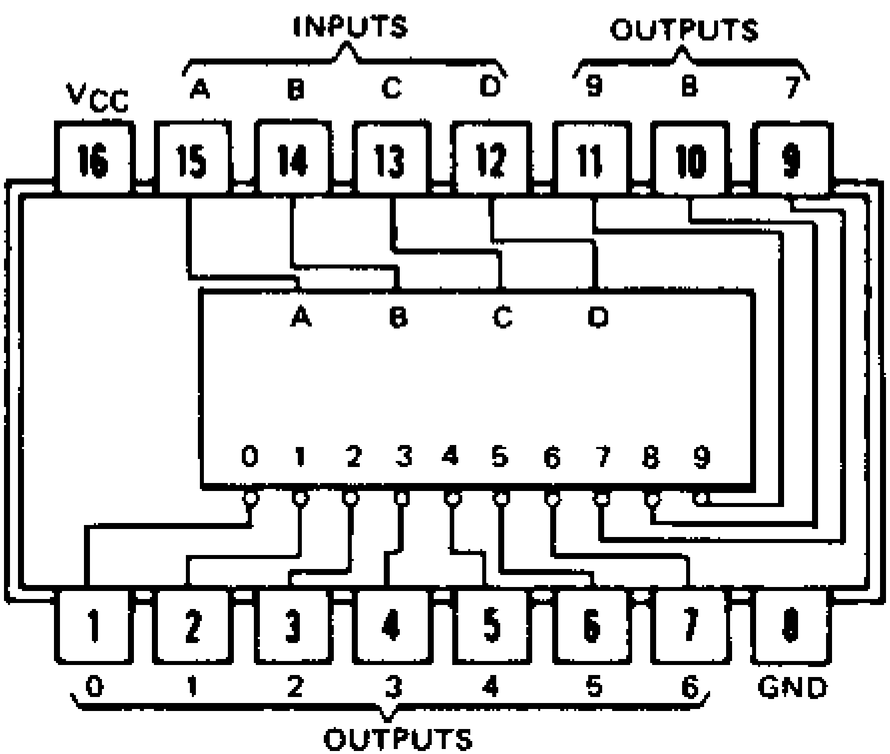
Binary Decoder ICs



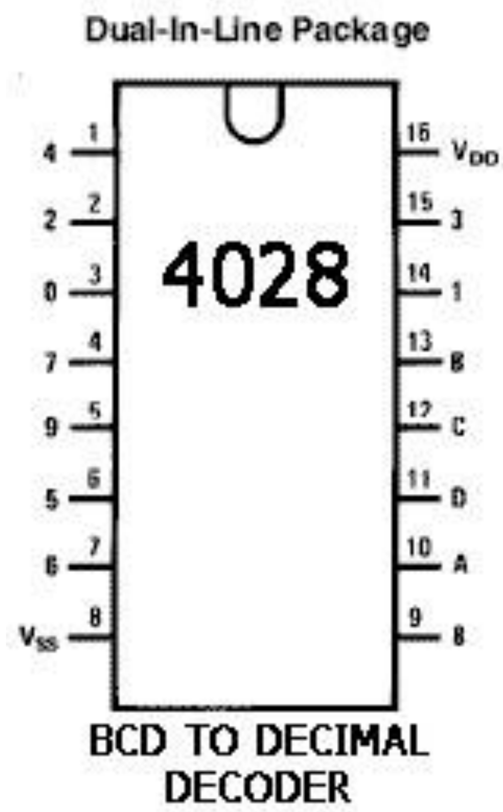
Dual 2-to-4 Decoder/Mux

Binary Decoder ICs

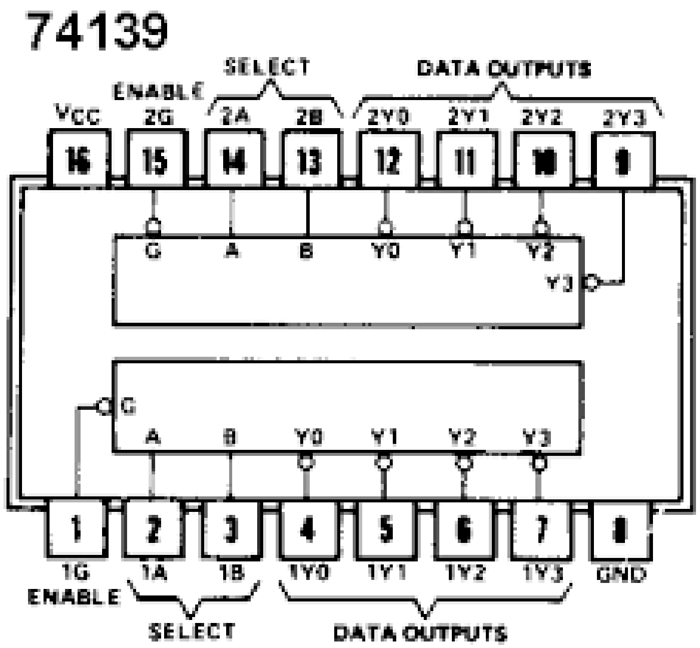
7442/43/44

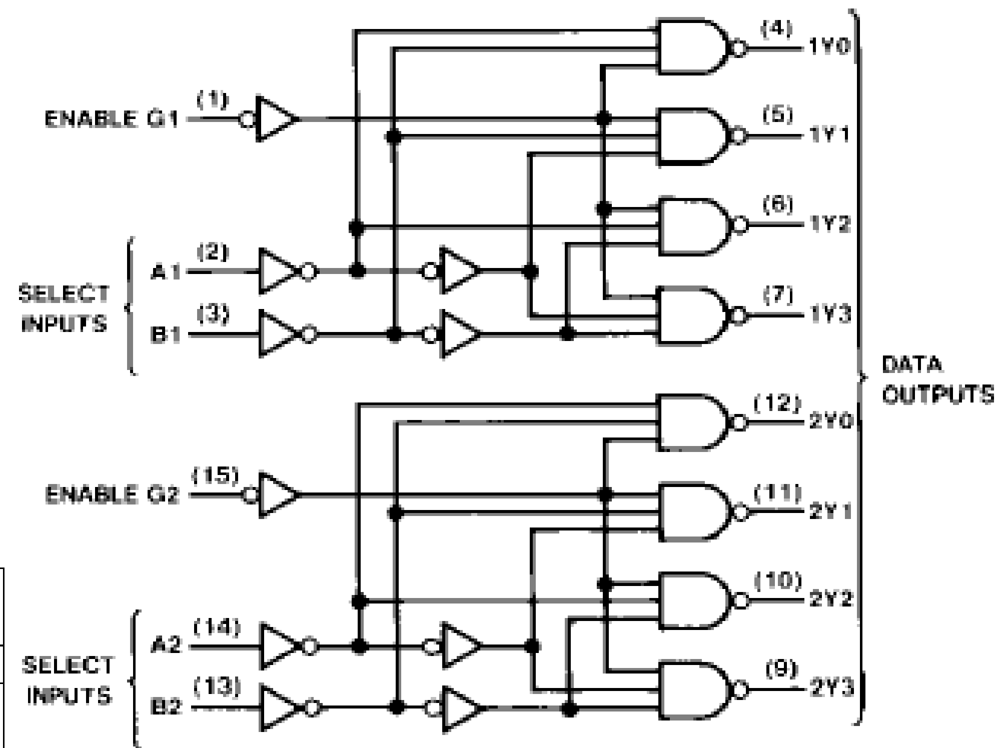


BCD-to-Decimal



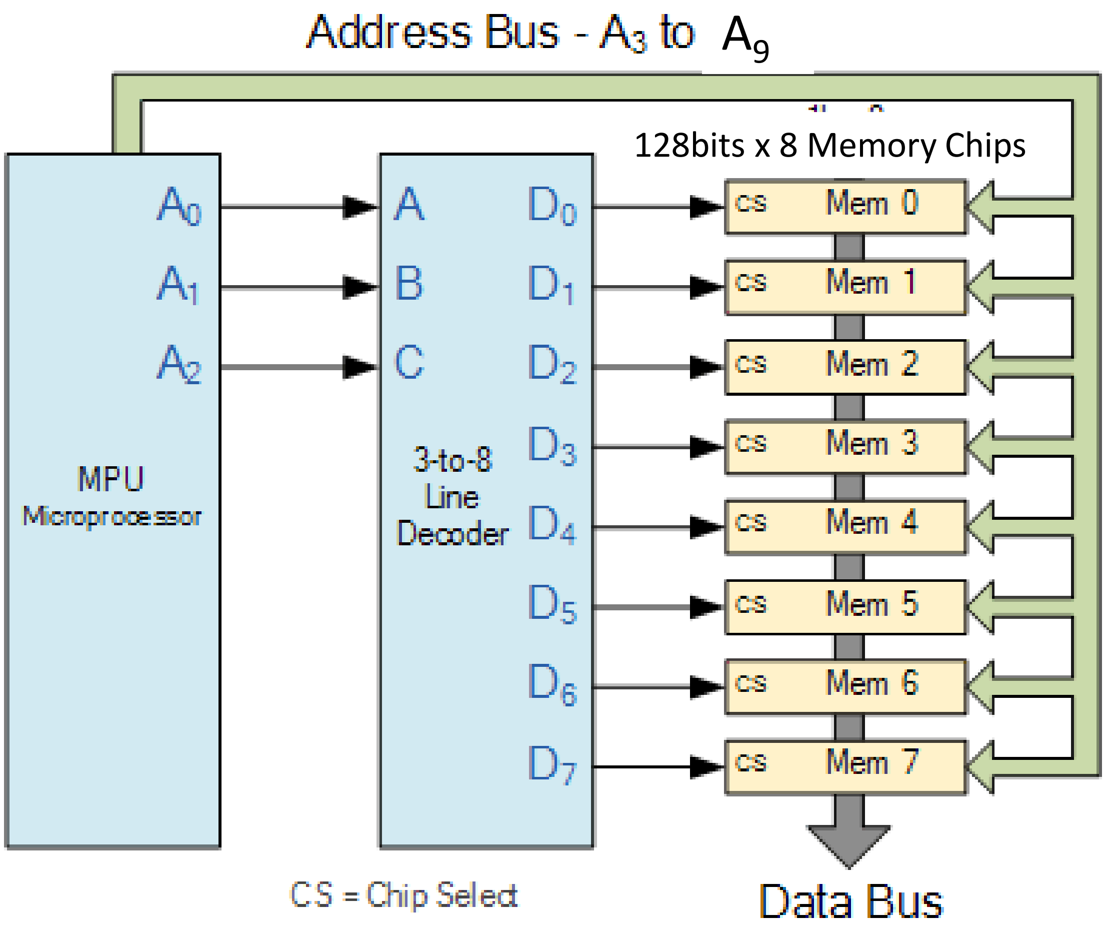
Testing 74LS139 Dual 2-to-4 Decoder/Demux





Inputs			Outputs			
Enable	Select					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

Application: Memory Address Decoder

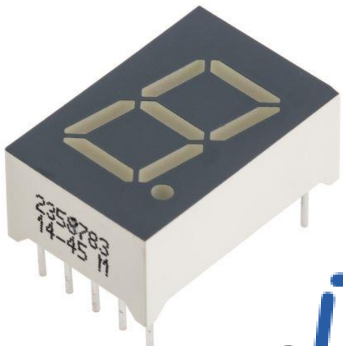


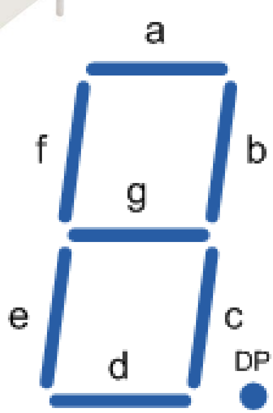
- Which memory chip will be accessed by the following addresses?
  - 1000101001
  - 1100101110
  - 0100101001

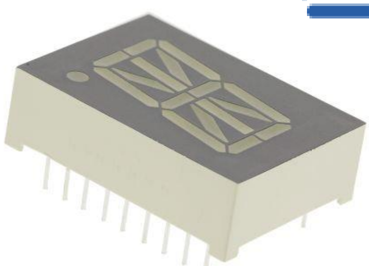
Application: Display Decoder

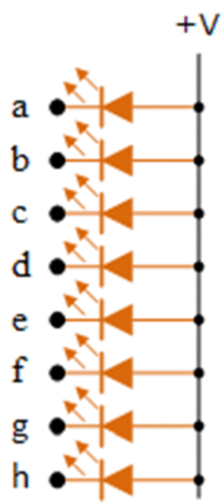
- What is a display decoder?
- What is an SSD?
- How a display decoder is used to drive an SSD?
- What are the advantages of display drivers?
- How do you construct a display decoder using basic logic gates?

Seven-Segment Display (SSD) Unit



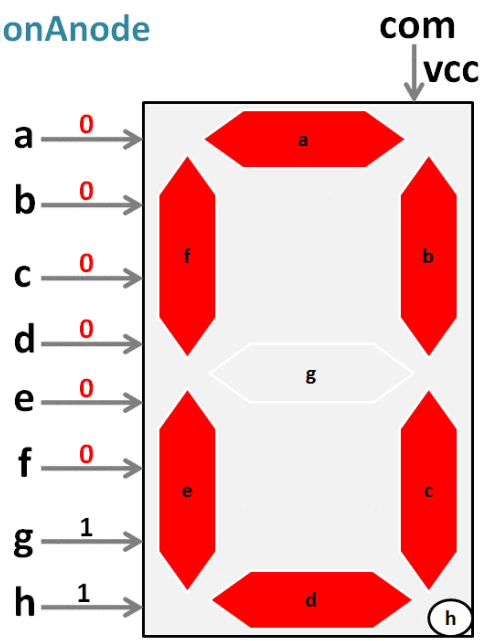


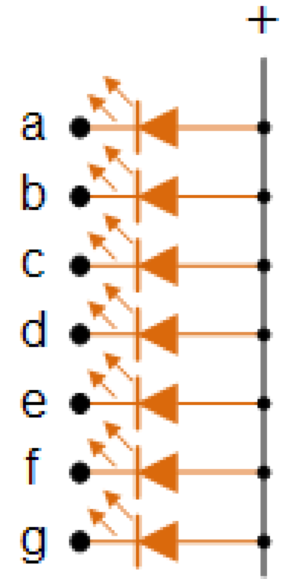




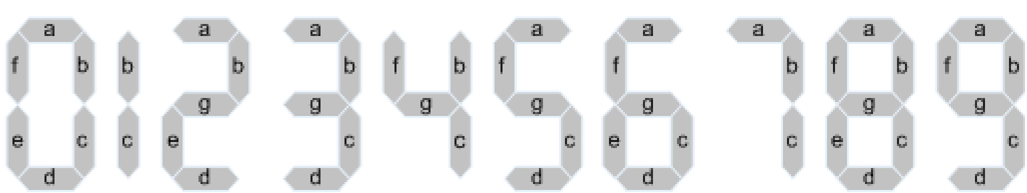
ExploreEmbedded

CommonAnode

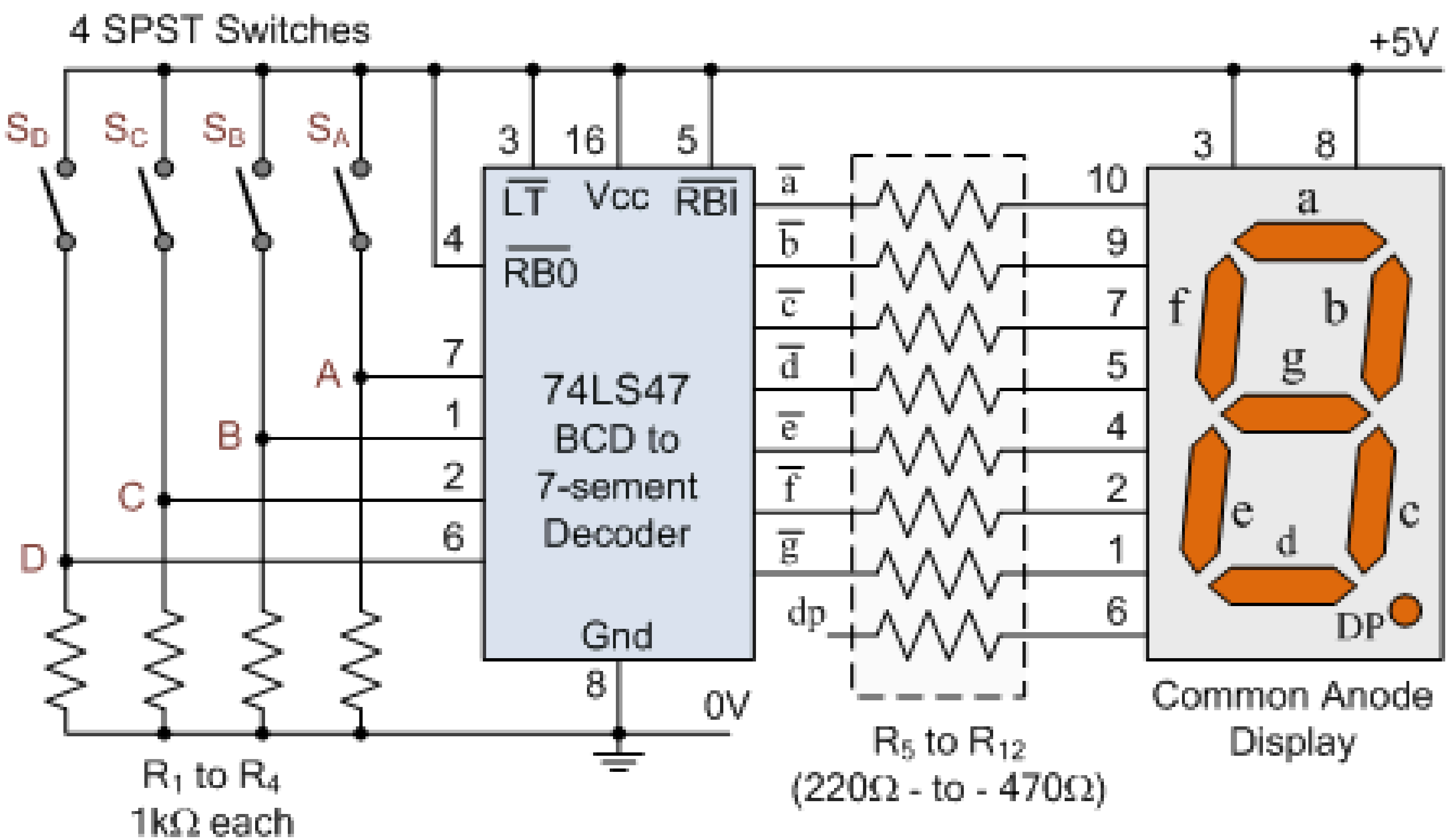




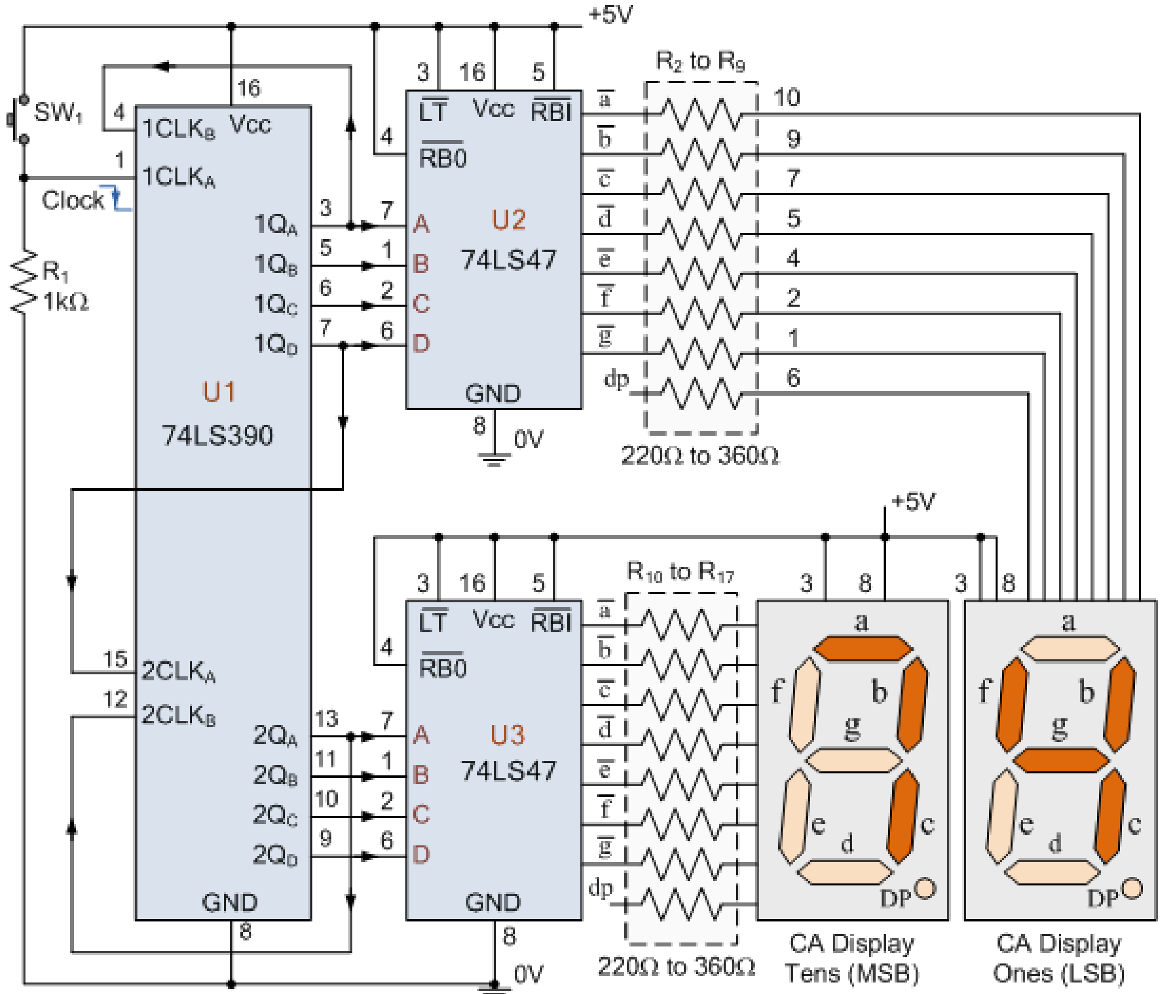
Common Anode

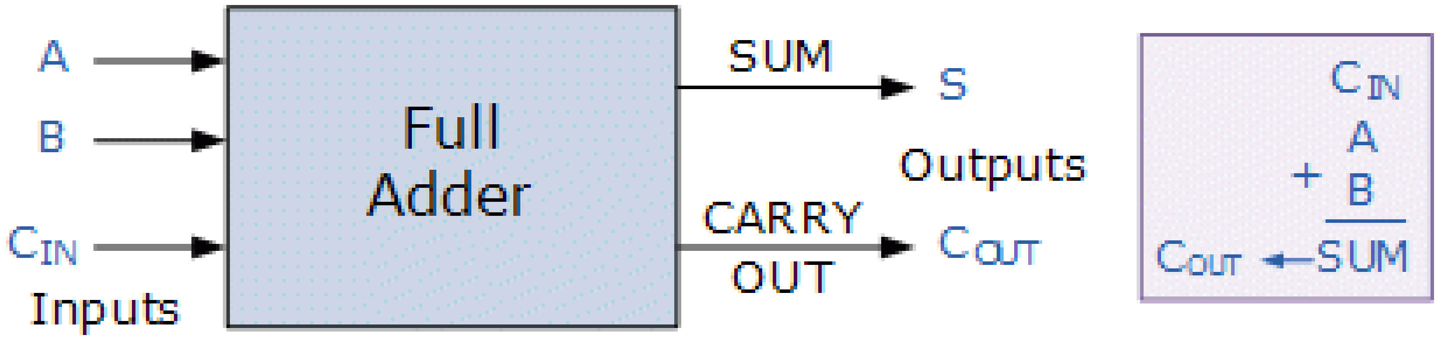


BCD-to-SSD Decoder



Counting Circuit





BINARY ADDER

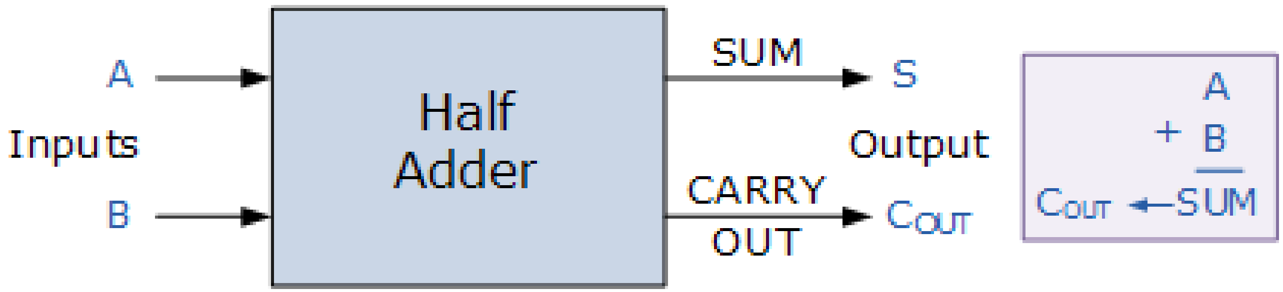
Binary Adder

- How the binary addition differs from denary (base 10) addition?
- How do you implement a binary adder using logic gates?
- What is the difference between a half-adder and a full-adder?
- What are the disadvantages of ripple adders?

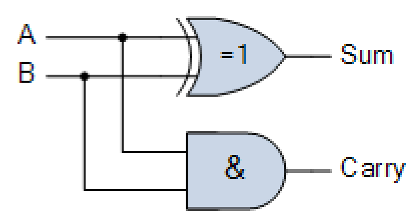
Binary Addition: Half-Adder

123	A	
+ 789	B	(Addend)
912	SUM	

0	0	1	1
+0	+1	+0	+1
0	1	1	(carry) 1←0



Symbol



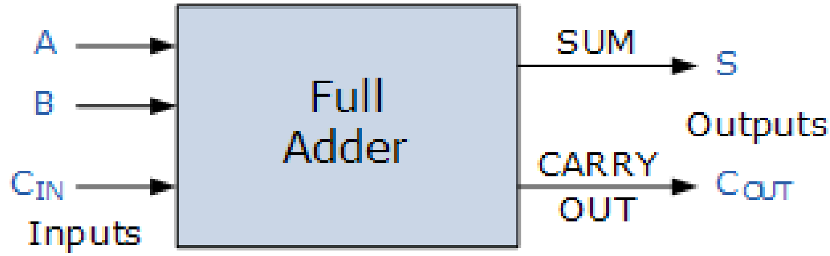
Truth Table

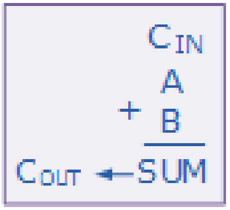
B	A	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

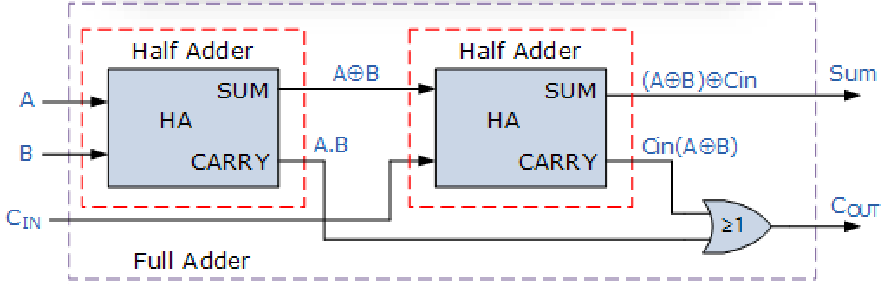
Binary Addition: Full-Adder

123	A	
+ 789	B	(Addend)
912	SUM	

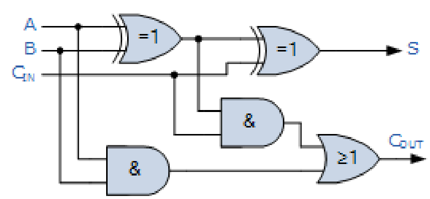
0	0	1	1
+0	+1	+0	+1
0	1	1	(carry) 1←0







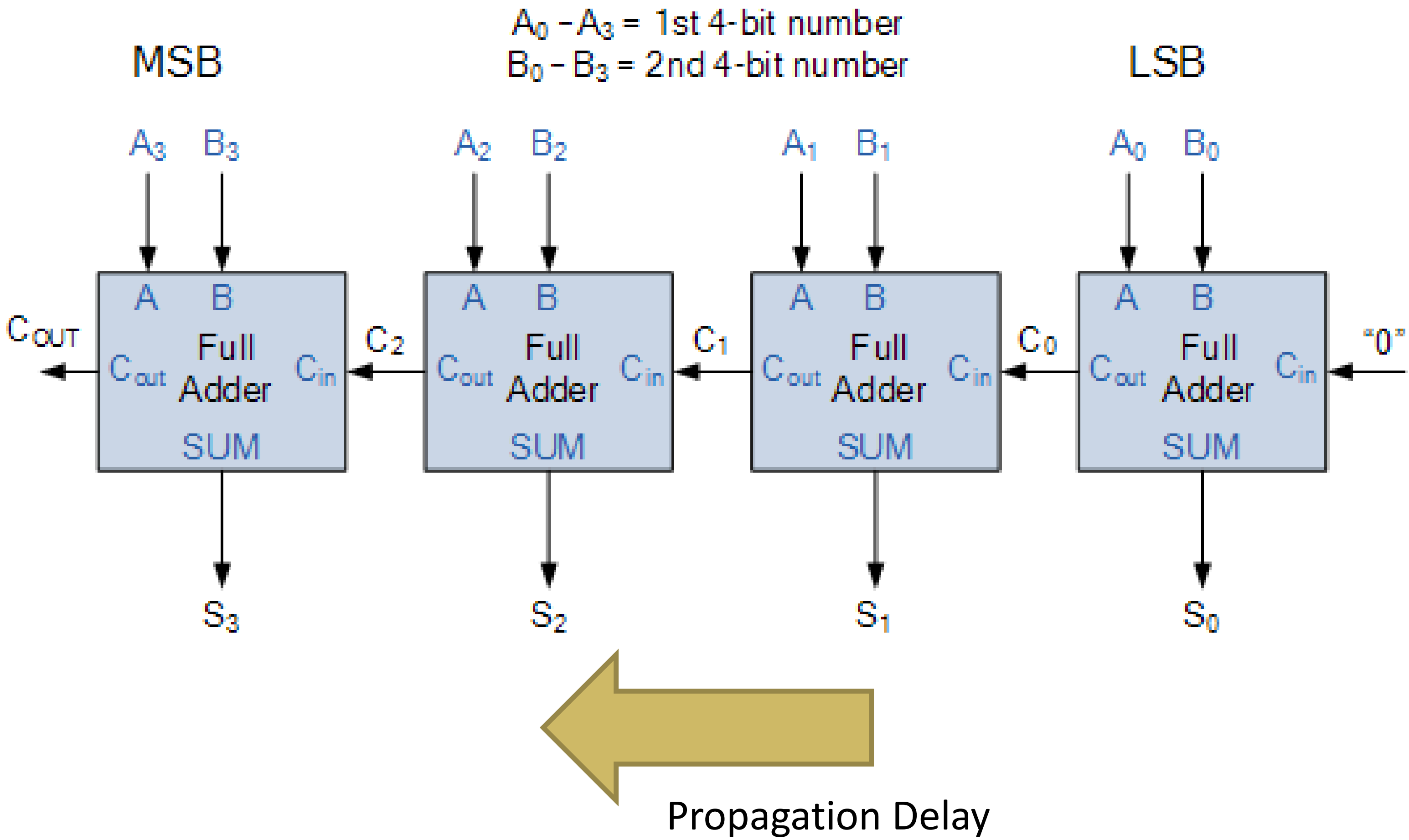
Symbol



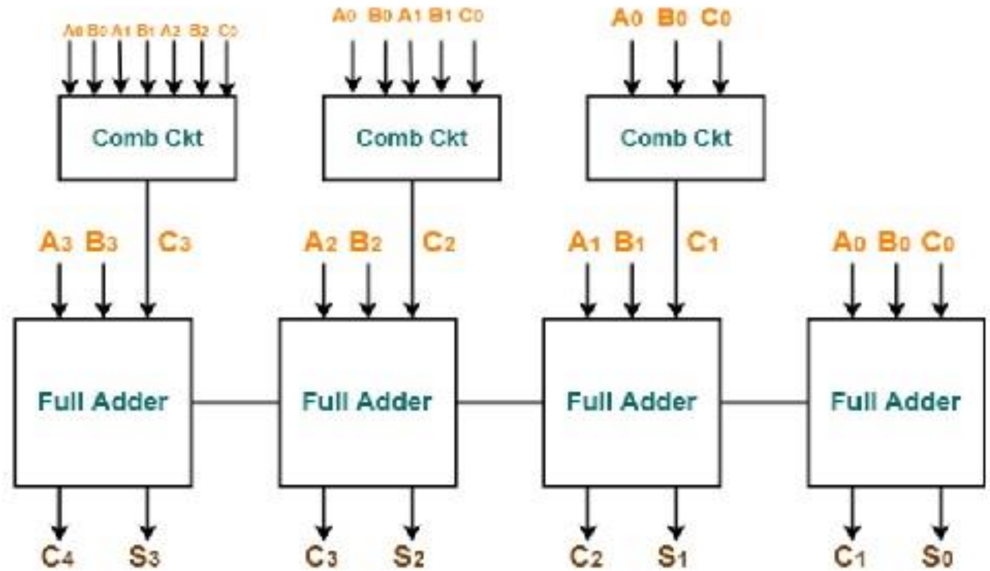
Truth Table

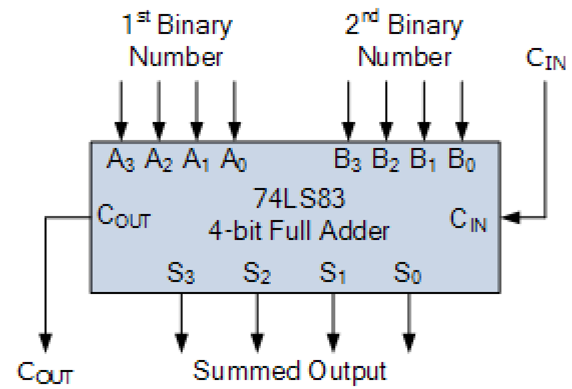
C-in	B	A	Sum	C-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

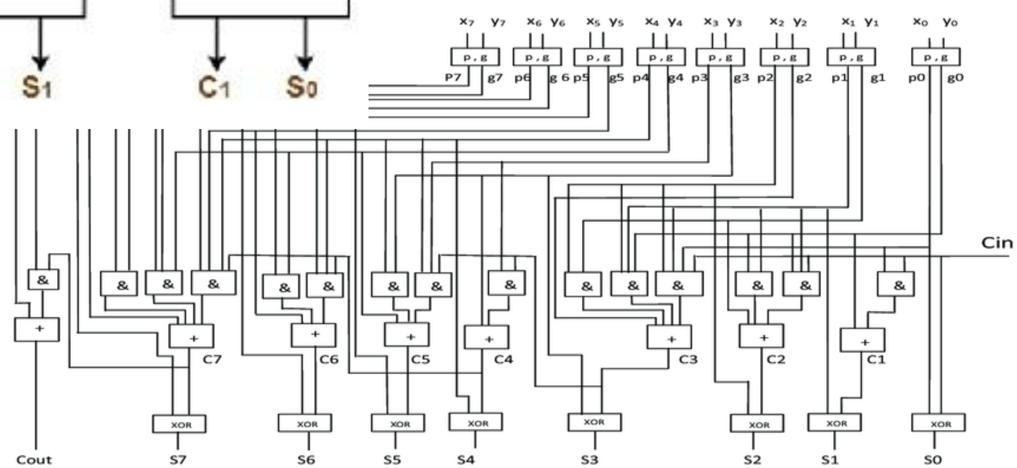
4-Bit Ripple Carry Adder

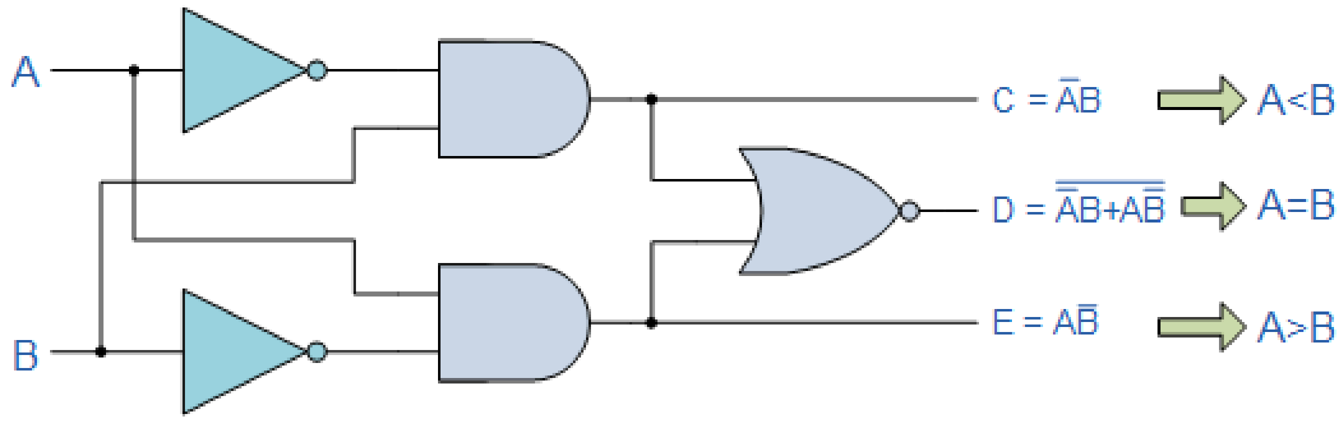


Hierarchical Carry Look Ahead Binary Adders

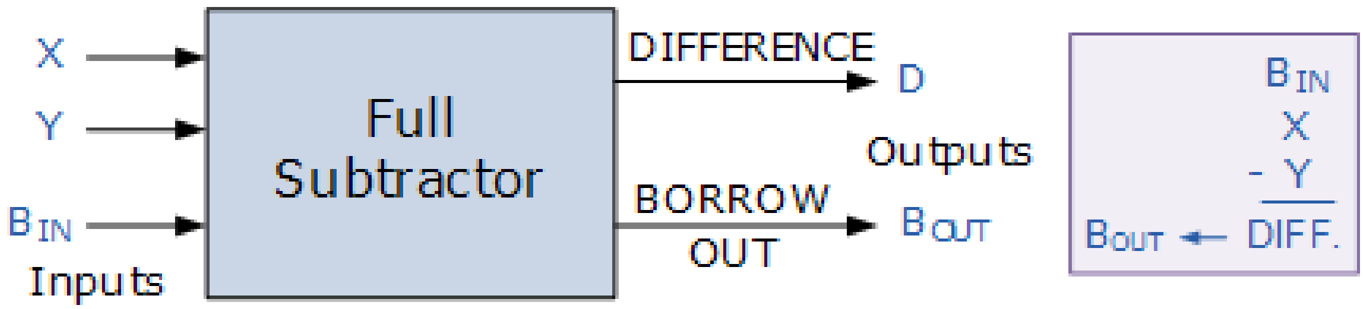








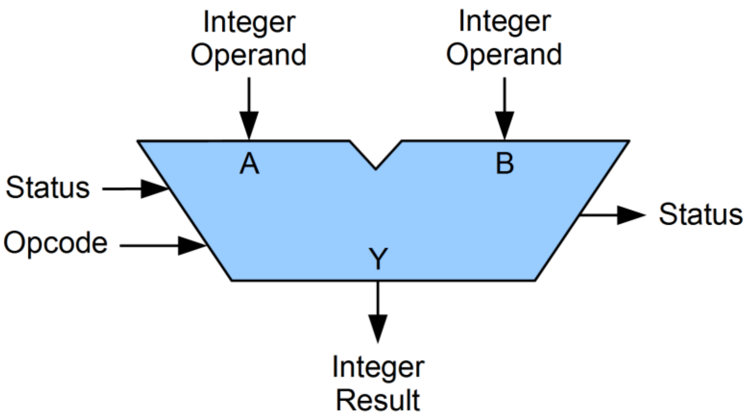
DIGITAL COMPARATOR

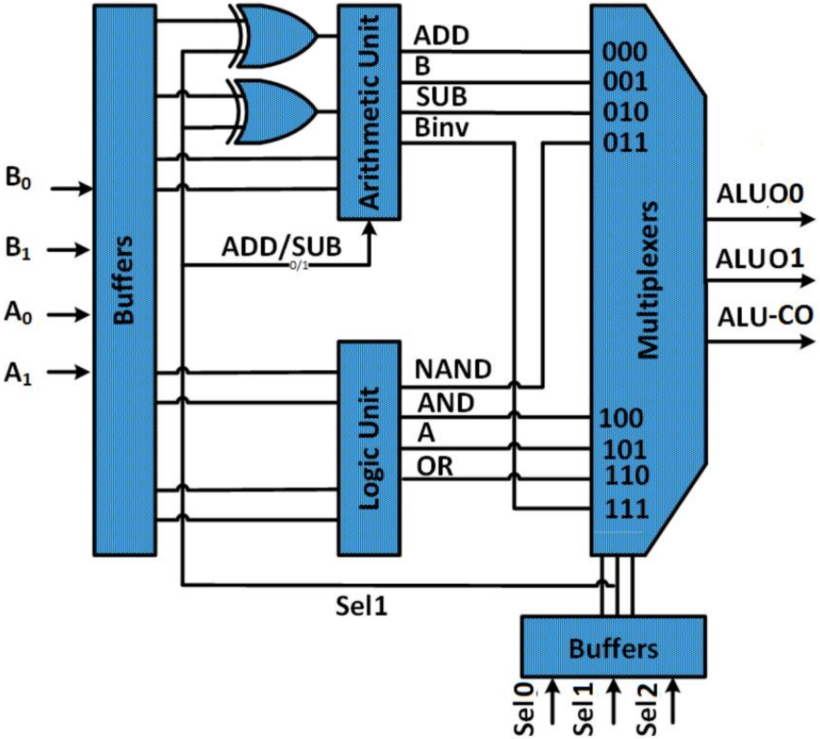


BINARY SUBTRACTOR

APPLICATION: ALU

Arithmetic & Logic Unit





How Computers Calculate

https://www.youtube.com/watch?v=1I5ZMmrOfnA&feature=emb_logo