University of Nottingham
UK | CHINA | MALAYSIA

LECTURE 4

Digital Electronics 1

Electromechanical Devices
MMME2051

Module Convenor – Surojit Sen

Learning Outcomes

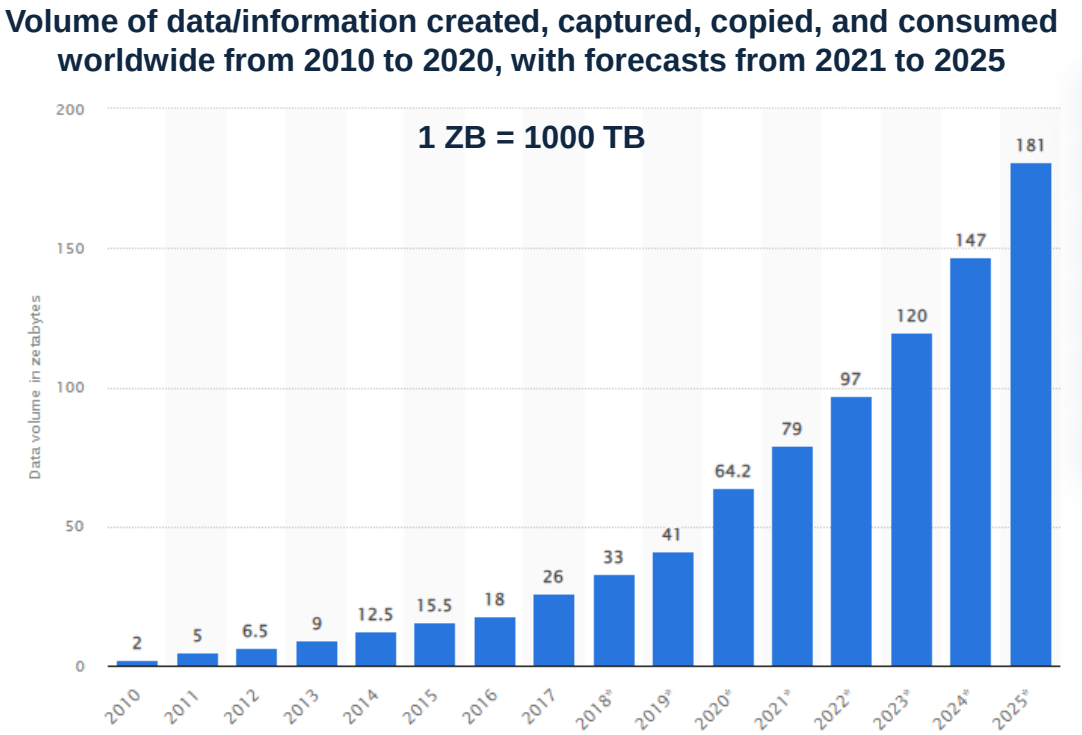
- Introduction to **Information**
  - **Exchange of information** – essence of any machine!
  - **Digital** v **Analog** signal
  - **Accuracy v Precision**
- **Electronics**
  - Revisit Engineering
  - Information in Electronics and Software
  - Common terminologies (**PCB**, **IC**)
- Logic **Gates**

What makes a Machine alive?

Information

Generation, sensing, communication, transformation, storage

- When you press the light switch in your room, a mechanical signal (pressing/flicking action) gets converted to an electrical signal, that lights up an LED by converting the electrical energy into light and heat.
- Everything is **conversion of energy** from one form to another
- Energy can be used to:
  - Do **work** – motor
  - Communicate **information**
    - **Digital**
    - **Analog**



Credit: https://www.statista.com/statistics/871513/worldwide-data-created/

Digital v Analog

Digital

You **ask** the computer/digital device:

"Is the room temperature:"

Q1 – **"15-16°C?"** A1 – **"No"**

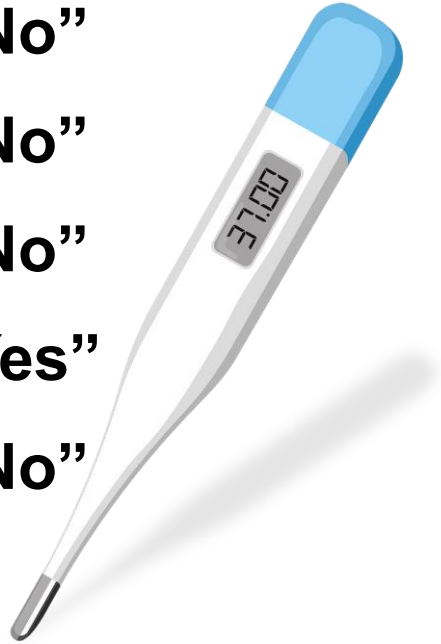
Q2 – **"16-17°C?"** A1 – **"No"**

Q3 – **"17-18°C?"** A1 – **"No"**

Q4 – **"18-19°C?"** A1 – **"No"**

Q5 – **"19-20°C?"** A1 – **"Yes"**

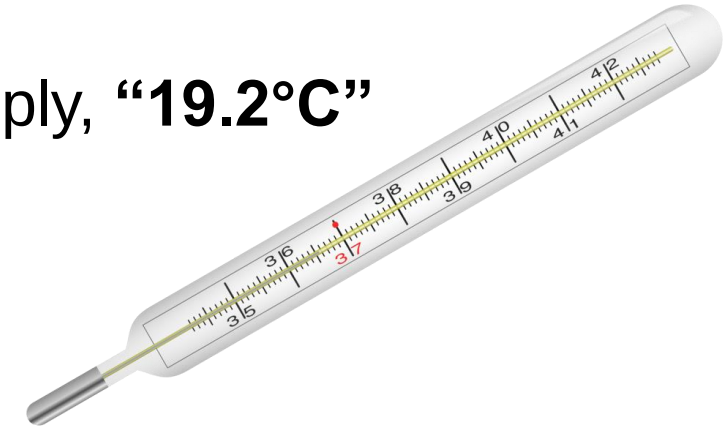
Q6 – **"20-21°C?"** A1 – **"No"**



Analog

You ask a **question** to someone/something, e.g., **"what is the temperature of the room?"**

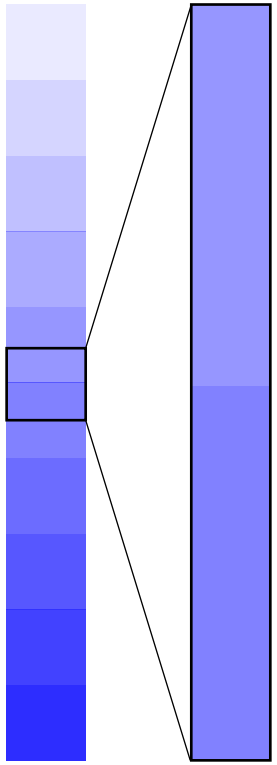
They reply, **"19.2°C"**



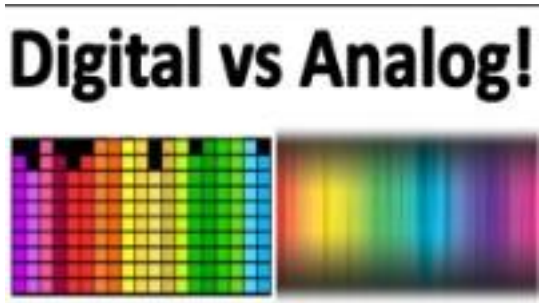
Digital v Analog

Digital

10 unique colours values, or levels



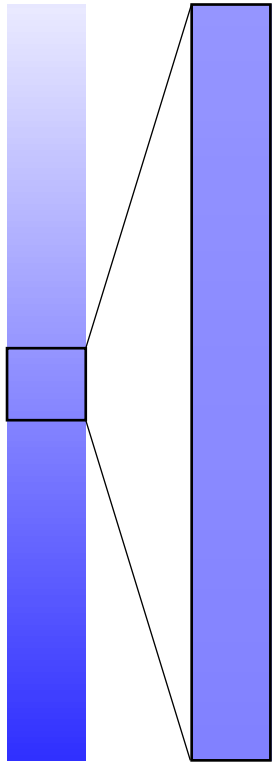
2 unique colours when zoomed in



https://www.youtube.com/watch?v=ZWdT-6Ld71Q

Analog

∞ unique colours values, or levels



∞ unique colours when zoomed in

Digital v Analog

Digital

Information in form of **discrete** symbols, or **levels**

Variable can be only 1 out of a **finite number of options**

Humans interpret physical values in discrete levels

- **Alphabets**
- **Binary number**
- **Logic state**
- **Answer to the question** – "*Are you* enjoying this module?"

Analog

Information in form of **continuous** and **real-valued levels**

Variable can be only 1 out of an **infinite number of options**

The physical values exist naturally in continuous spectrum levels

- **Air pressure in this room**
- **Volume of my voice**
- **Battery voltage in your laptop**
- **Answer to the question** – "*How much* are you enjoying this module?"

Digital Methods of Information Exchange

Language – using letters

There are 26 alphabets in the English language – digital!

Numbers

Every number that we use, uses a distinct number of symbols (including the decimal point)

Binary

e.g.,
11100100

Octal

e.g.,
344

Decimal

e.g.,
228

Hexadecimal

e.g.,
E4

Let us look at an example of a Binary and Octal question

What is the temperature of the room?

- **Option 0 –** Not 18-19°C
- **Option 1 –** 18-19°C

In **Binary**, you can respond with **2 choices** only.

In **Octal**, you can respond with **8 choices**.

Similarly, in **Decimal**, you respond with **10 choices**, and in **Hexadecimal**, you respond with **16 choices**.

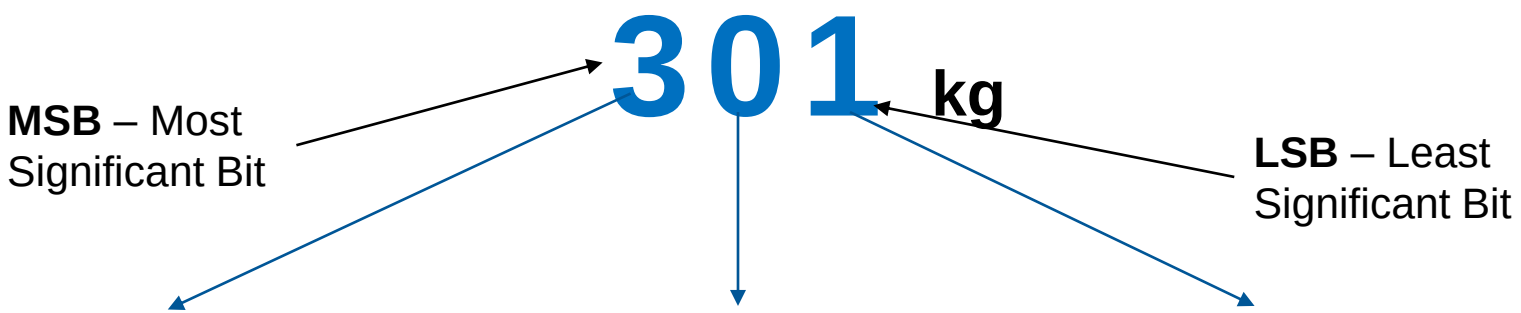
What is the temperature of the room?

- **Option 0 –** 15-16°C
- **Option 1 –** 16-17°C
- **Option 2 –** 17-18°C
- **Option 3 –** 18-19°C
- **Option 4 –** 19-20°C
- **Option 5 –** 20-21°C
- **Option 6 –** 21-22°C
- **Option 7 –** 22-23°C

How does this actually relate to “numbers”?

Let us look at a number in the “Decimal” number-format, the one that we have grown up with.

Weight of the Formula Student 2021 car is



$3 \times 10^2 = 300$

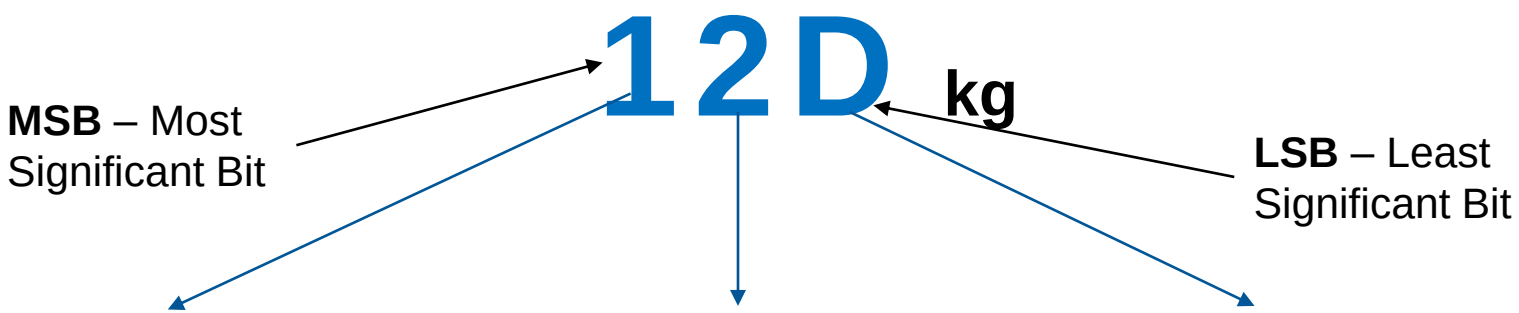
$0 \times 10^1 = 0$

$1 \times 10^0 = 1$

How does this actually relate to “numbers”?

The same number in the Hexadecimal format will be

Weight of the Formula Student 2021 car is



$1 \times 16^2 = 256$

$2 \times 16^1 = 32$

$D \times 16^0 = 13$

How does this actually relate to “numbers”?

How about in Binary?

Weight of the Formula Student 2021 car is

LSB – Least Significant Bit

MSB – Most Significant Bit

0001 0010 1101 kg

- $0 \times 2^{11} = 0$
- $0 \times 2^{10} = 0$
- $0 \times 2^{9} = 0$
- $1 \times 2^{8} = 256$

256

- $0 \times 2^{7} = 0$
- $0 \times 2^{6} = 0$
- $1 \times 2^{5} = 32$
- $0 \times 2^{4} = 0$

32

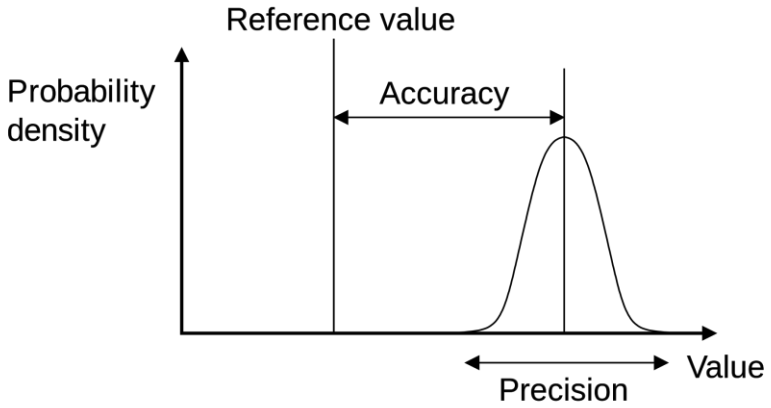
- $1 \times 2^{3} = 8$
- $1 \times 2^{2} = 4$
- $0 \times 2^{1} = 0$
- $1 \times 2^{0} = 1$

13

Some Important Terminologies

Accuracy is a measure of how close a “number” is to the **true value** of what it **represents**

Precision is a measure of how dispersed a “number” is to the central value (central value may be widely inaccurate!)



Accurate
~~Precise~~

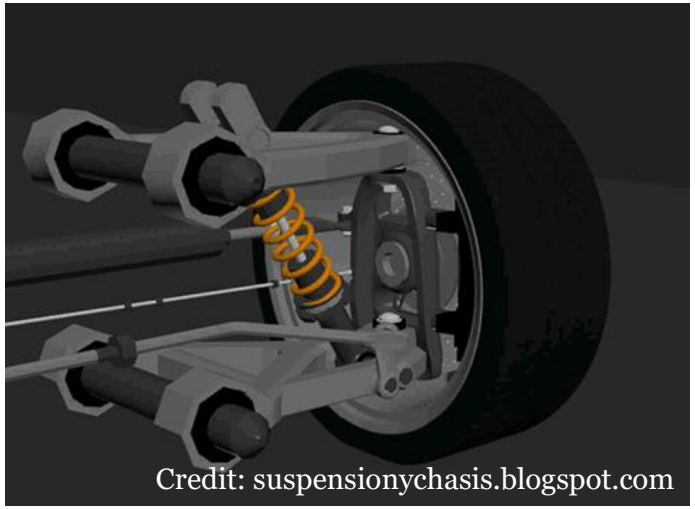
~~Accurate~~
Precise

Accurate
Precise

Learning Outcomes

- Introduction to **Information**
  - **Exchange of information** – essence of any machine!
  - **Digital** v **Analog** signal
  - **Accuracy v Precision**
- **Electronics**
  - Revisit Engineering
  - Information in Electronics and Software
  - Common terminologies (**PCB**, **IC**)
- Logic **Gates**

Electronics Engineering


Credit: suspensionychasis.blogspot.com

Suspension & **chasses**
Mechanical Engineering


Credit: pinterest.com

Vehicle Control Unit (VCU) that sends signals/commands to drive/stop the car
Electronic Engineering

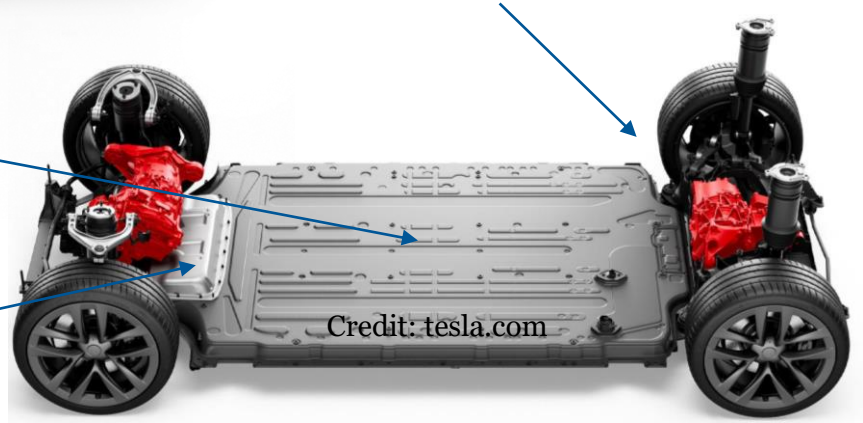
Code written to program the VCU
Computer/Software Engineering


Credit: tesla.com

Motor that converts electrical power from battery to mechanical motion
Electromechanical Engineering

Battery that supplies power to drive the motor
Electrical Engineering

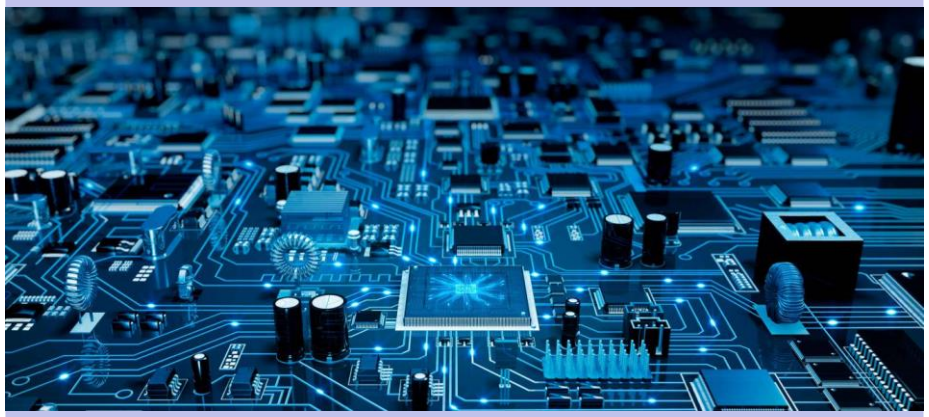
Power Converter that controls the flow of electrical power between battery and motor
Power Electronics

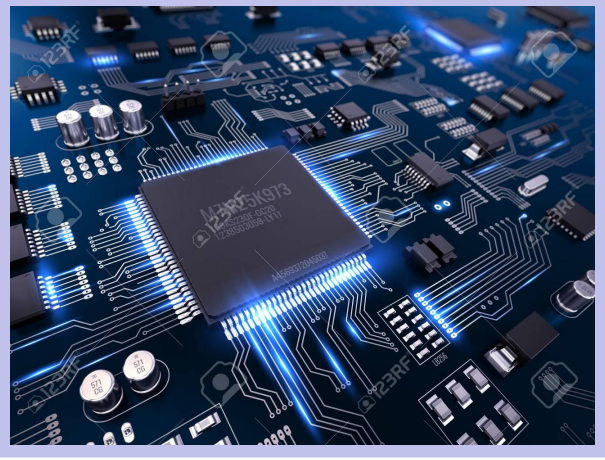

Credit: tesla.com

Electronics Engineering

Electronic

LESS



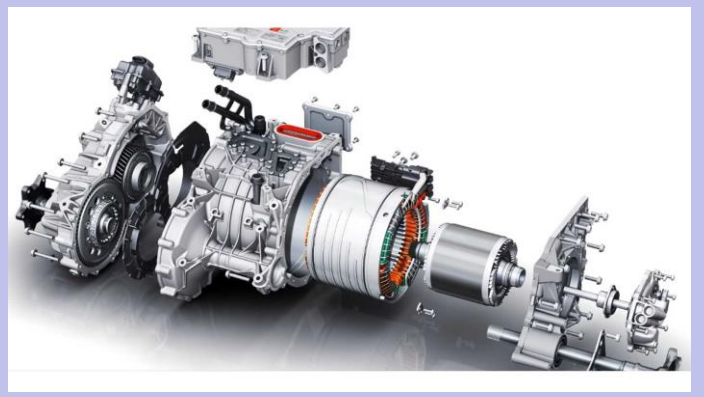


5V

50mA

Electrical

MORE



Information in Electronics

We use **binary number system** in logical circuits in electronics

This aligns with computer/software engineering – binary system used

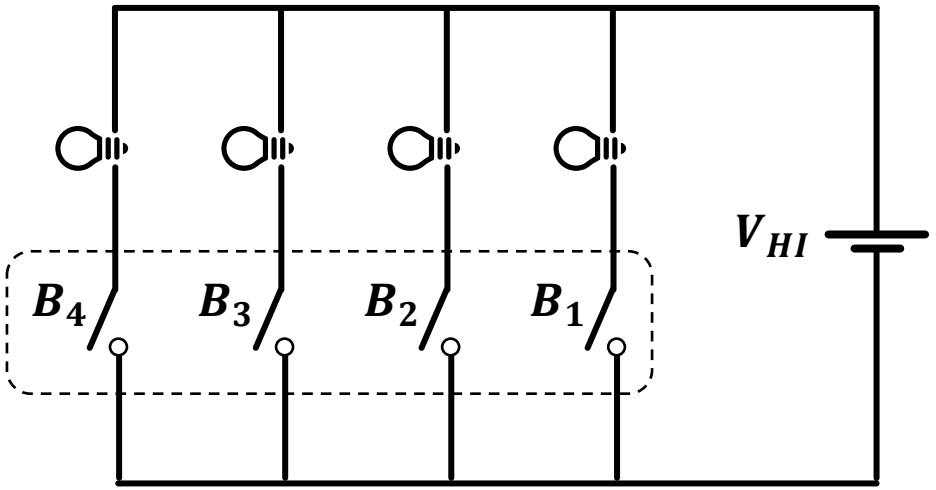
Logic – TRUE/FALSE

We said that **301** (weight of the FS21 in kg) is represented in binary as

0 0 0 1 0 0 1 0 1 1 0 1

How is this actually done in reality?

Two voltage levels – **High & Low**



0 0 0 0 = 0000

MSB – Most Significant Bit

LSB – Least Significant Bit

Information in Electronics

We use **binary number system** in logical circuits in electronics

This aligns with computer/software engineering – binary system used

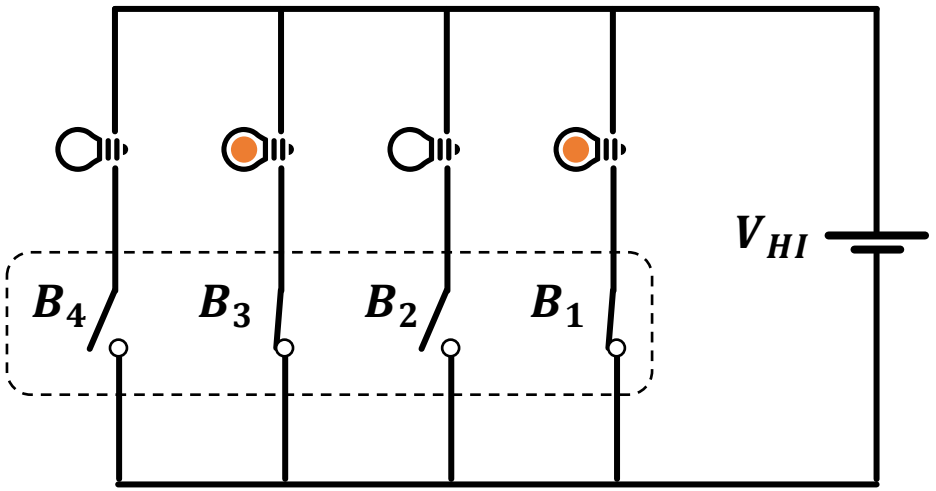
Logic – TRUE/FALSE

We said that **301** (weight of the FS21 in kg) is represented in binary as

0 0 0 1 0 0 1 0 1 1 0 1

How is this actually done in reality?

Two voltage levels – **High & Low**



0 1 0 1 = 0101

MSB – Most Significant Bit

LSB – Least Significant Bit

Information in Electronics

We use **binary number system** in logical circuits in electronics

This aligns with computer/software engineering – binary system used

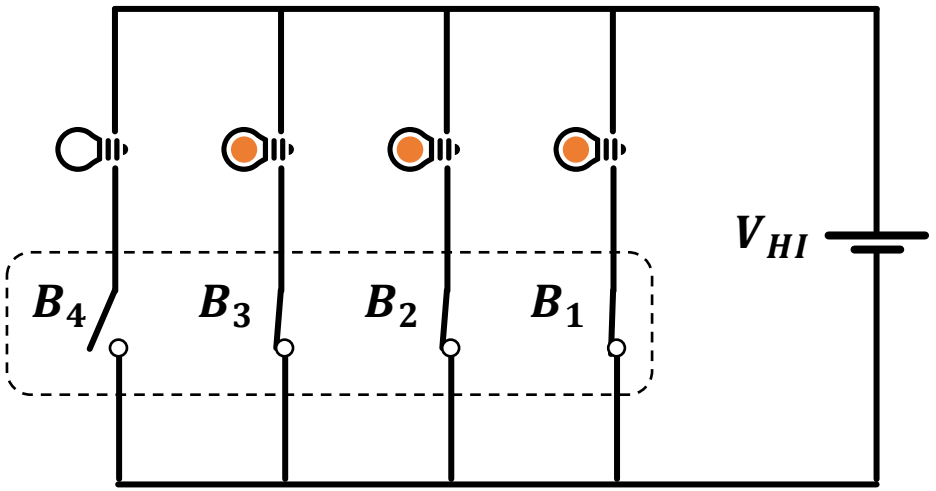
Logic – TRUE/FALSE

We said that **301** (weight of the FS21 in kg) is represented in binary as

0 0 0 1 0 0 1 0 1 1 0 1

How is this actually done in reality?

Two voltage levels – **High & Low**



0 1 1 1 = 0111

MSB – Most Significant Bit

LSB – Least Significant Bit

Information in Electronics

We use **binary number system** in logical circuits in electronics

This aligns with computer/software engineering – binary system used

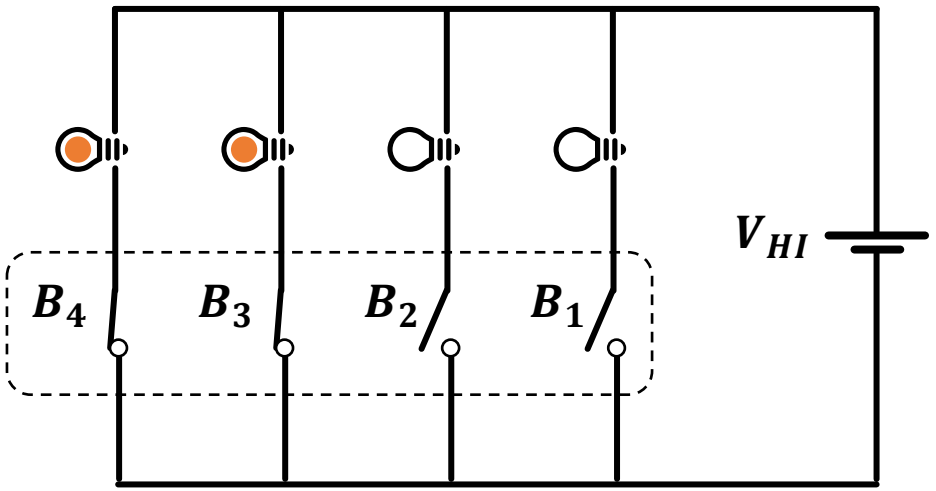
Logic – TRUE/FALSE

We said that **301** (weight of the FS21 in kg) is represented in binary as

0 0 0 1 0 0 1 0 1 1 0 1

How is this actually done in reality?

Two voltage levels – **High & Low**



1 1 0 0 = 1100

MSB – Most Significant Bit

LSB – Least Significant Bit

How to Add/Subtract Binary Numbers?

Just the same way you do for decimal numbers!

Decimal	Binary
1	1 1 1 1 1
124	0111 1100
+229	+1110 0101
353	**1 0110 0001**
124	0111 1100
- 47	- 0010 1111
77	**0100 1101**

We won't do **multiplication** and **division** operations on binary numbers

If you did, you would use a XOR gate – we will study that soon

We shall study **Binary Algebra** later

How to Add/Subtract Binary Numbers?

Decimal	B_4	B_2	B_2	B_1	Binary
0	0	0	0	0	0000
1	0	0	0	1	0001
2	0	0	1	0	0010
3	0	0	1	1	0011
4	0	1	0	0	0100
5	0	1	0	1	0101
6	0	1	1	0	0110
7	0	1	1	1	0111
8	1	0	0	0	1000
9	1	0	0	1	1001
10	1	0	1	0	1010
11	1	0	1	1	1011
12	1	1	0	0	1100
13	1	1	0	1	1101
14	1	1	1	0	1110
15	1	1	1	1	1111

We would call this a 4-bit binary number – it is made of 4 bits

Maximum number we can count up to for a binary number is given by $\mathbf{2^n - 1}$

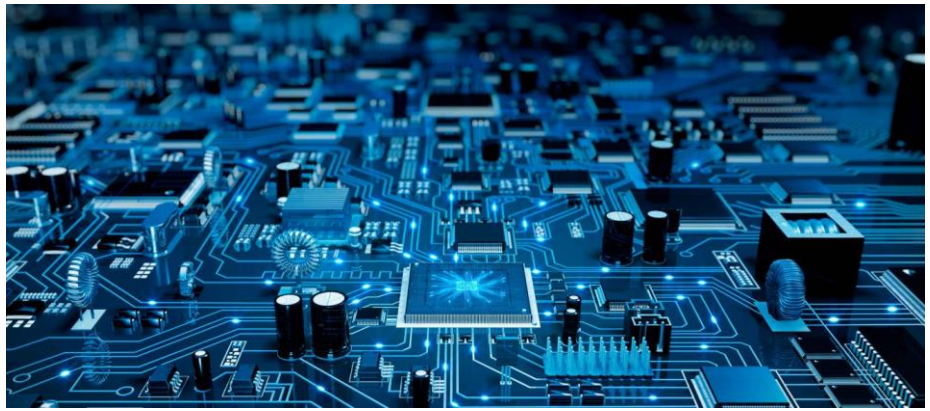
$$1\ byte\ =\ 8\ bits$$

Modern computers use **32-bit** or **64-bit** numbers in its operating system

Remember the numeric data types you learnt in MATLAB last year?

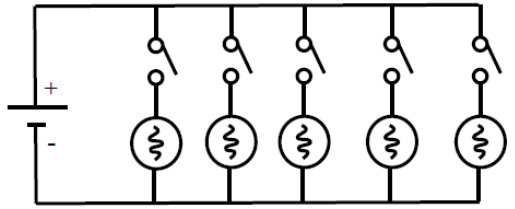
- **Single** – 4 bytes
- **Double** – 8 bytes
- **Int8** – 1 byte

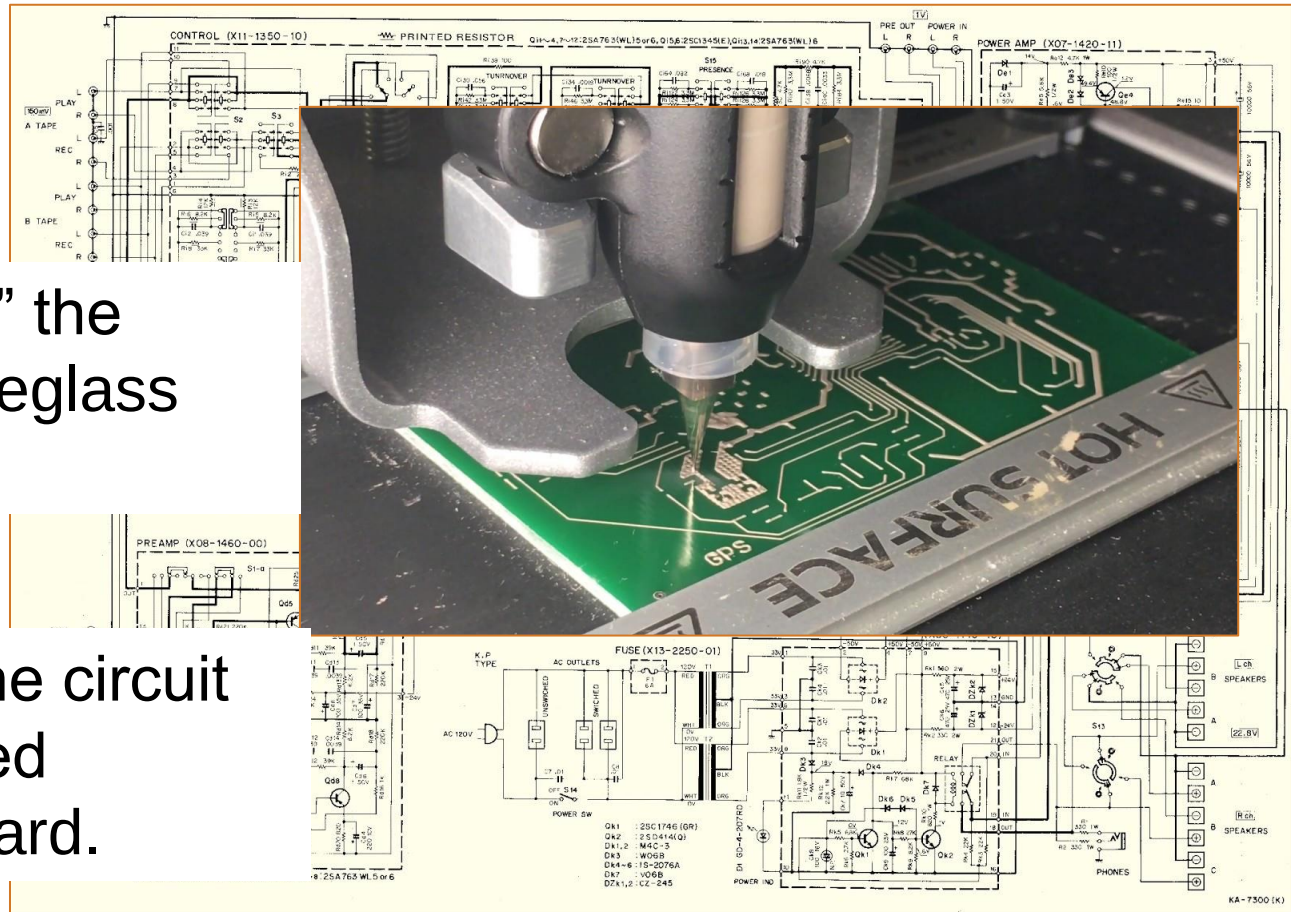
Some Important Terminologies



Printed Circuit Board

Step 1 – Design the electrical/electronic circuit you want





Step 2 – "Print" the circuit on a fibreglass board

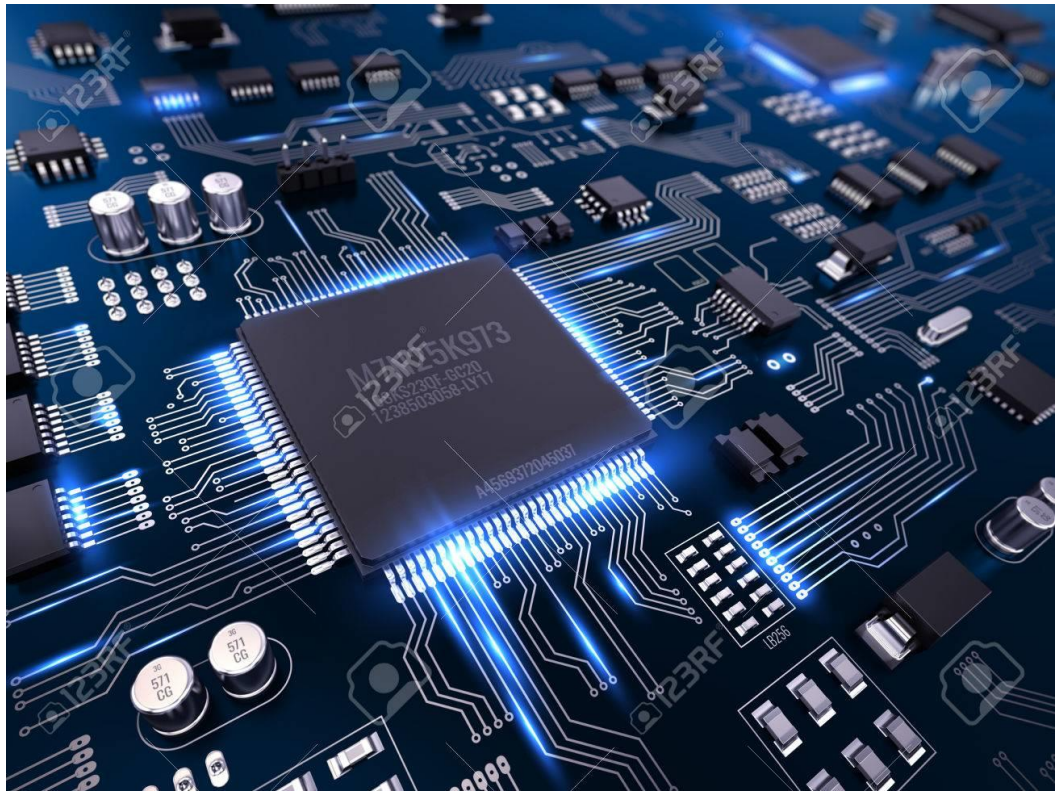
Step 3 – "Complete" the circuit by "placing" the required components on the board.

Some Important Terminologies

Yamaha YSM12 PCB Pick and Place Machine

https://www.youtube.com/watch?v=li7L6jsVdaE

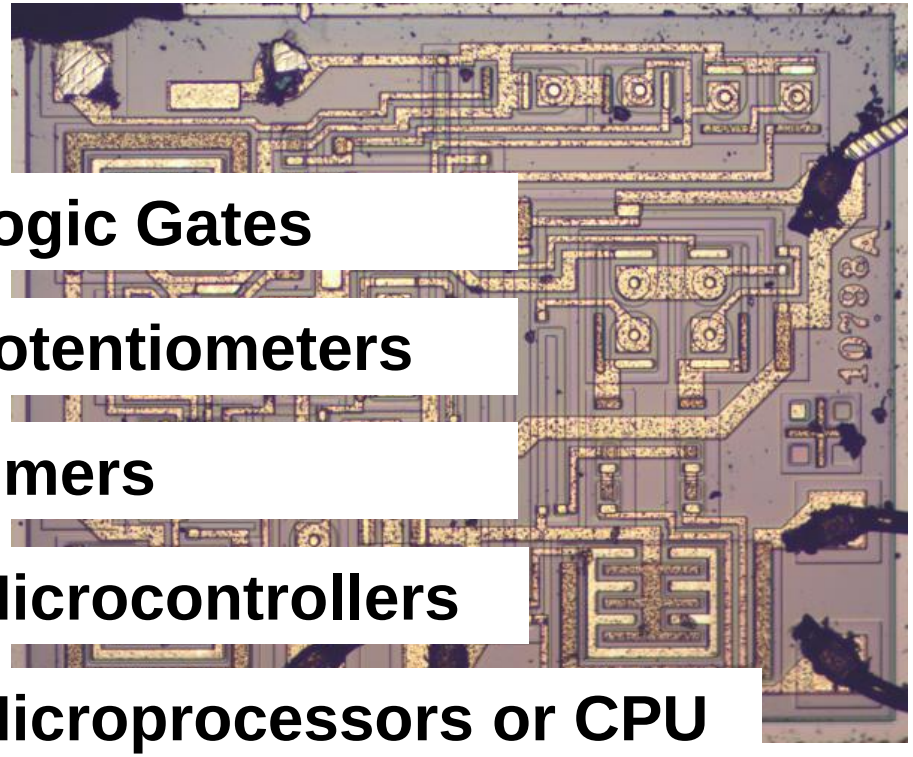
Some Important Terminologies



Integrated Circuit

It is essentially a PCB within a small monolithic plastic package!

555 timer teardown (link)



- **Logic Gates**
- **Potentiometers**
- **Timers**
- **Microcontrollers**
- **Microprocessors or CPU**

https://www.youtube.com/watch?v=cIlwGFcDLhI
30-min intro course on IC design and manufacture

Learning Outcomes

- Introduction to **Information**
  - **Exchange of information** – essence of any machine!
  - **Digital** v **Analog** signal
  - **Accuracy v Precision**
- **Electronics**
  - Revisit Engineering
  - Information in Electronics and Software
  - Common terminologies (**PCB**, **IC**)
- Logic **Gates**

Logic Gates

What is a Computer?

It is essentially a really big and complex electronic circuit that **processes binary information** using **logical circuits**

Logical circuit (as the name suggests) uses logic (*if "A is happening" then "make B happen"*) to arrive at decisions

The basic building block of logical circuits is a **logic gate**

There are mainly 3 kinds of gates:

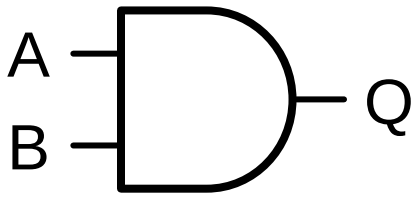
AND – outputs HI if all inputs are HI

OR – outputs HI if any input is HI

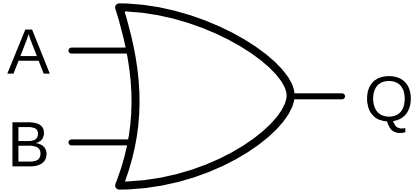
NOT – inverts the binary input (HI becomes LO and LO becomes HI)

Logic Gates

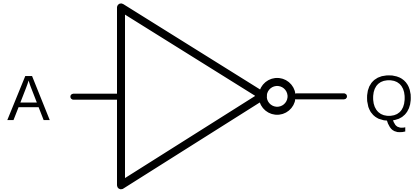
AND



OR



NOT



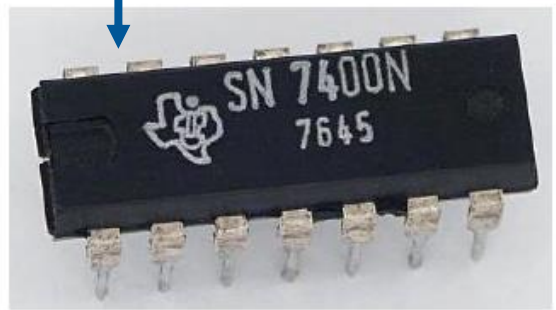
Truth Table

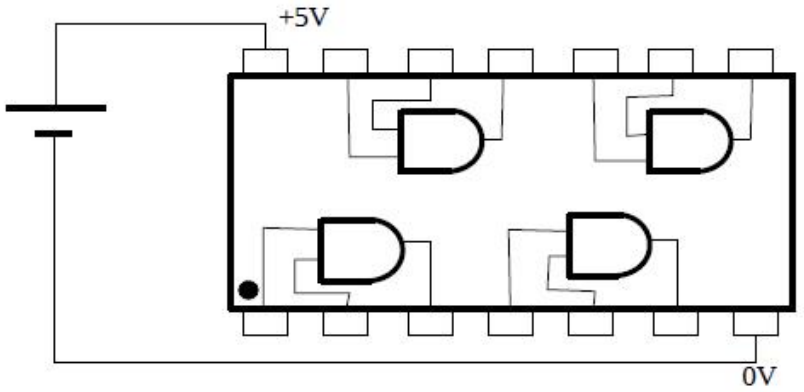
A	B	Q	Remark
0	0	**0**	HI if all inputs are HI
0	1	**0**	
1	0	**0**	
1	1	**1**	

A	B	Q	Remark
0	0	**0**	HI if any input is HI
0	1	**1**	
1	0	**1**	
1	1	**1**	

A	Q	Remark
0	**1**	Bit inversion
1	**0**	

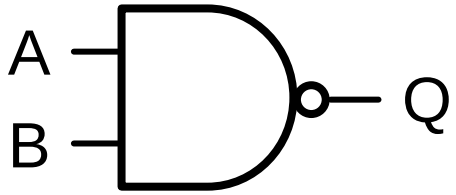
This is an Integrated Circuit, or IC!



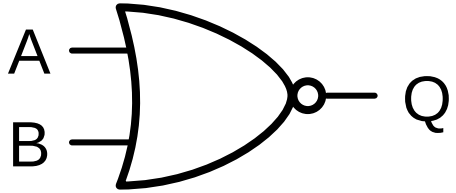


Logic Gates

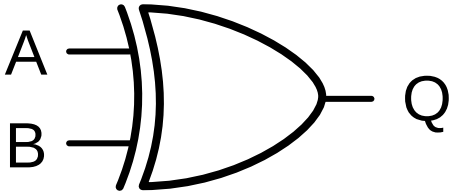
NAND



NOR



XOR



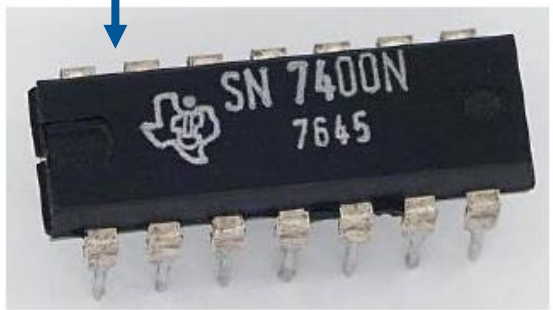
Truth Table

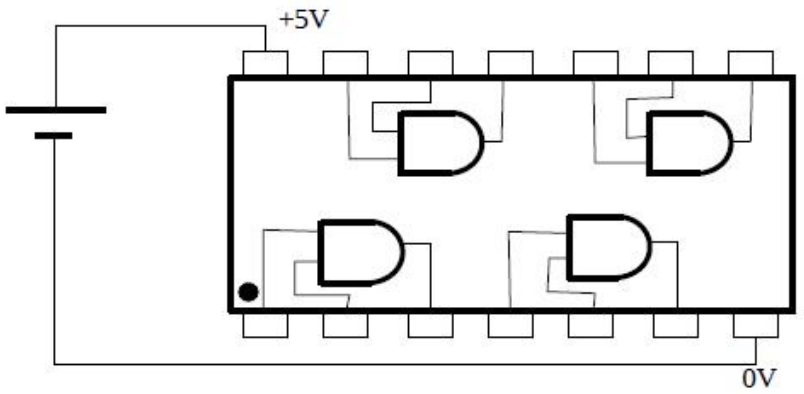
A	B	Q	Remark
0	0	**1**	LO if all inputs are HI
0	1	**1**	
1	0	**1**	
1	1	**0**	

A	B	Q	Remark
0	0	**1**	LO if any input is HI
0	1	**0**	
1	0	**0**	
1	1	**0**	

A	B	Q	Remark
0	0	**0**	HI if at least one input is HI and one is LO
0	1	**1**	
1	0	**1**	
1	1	**0**	

This is an Integrated Circuit, or IC!





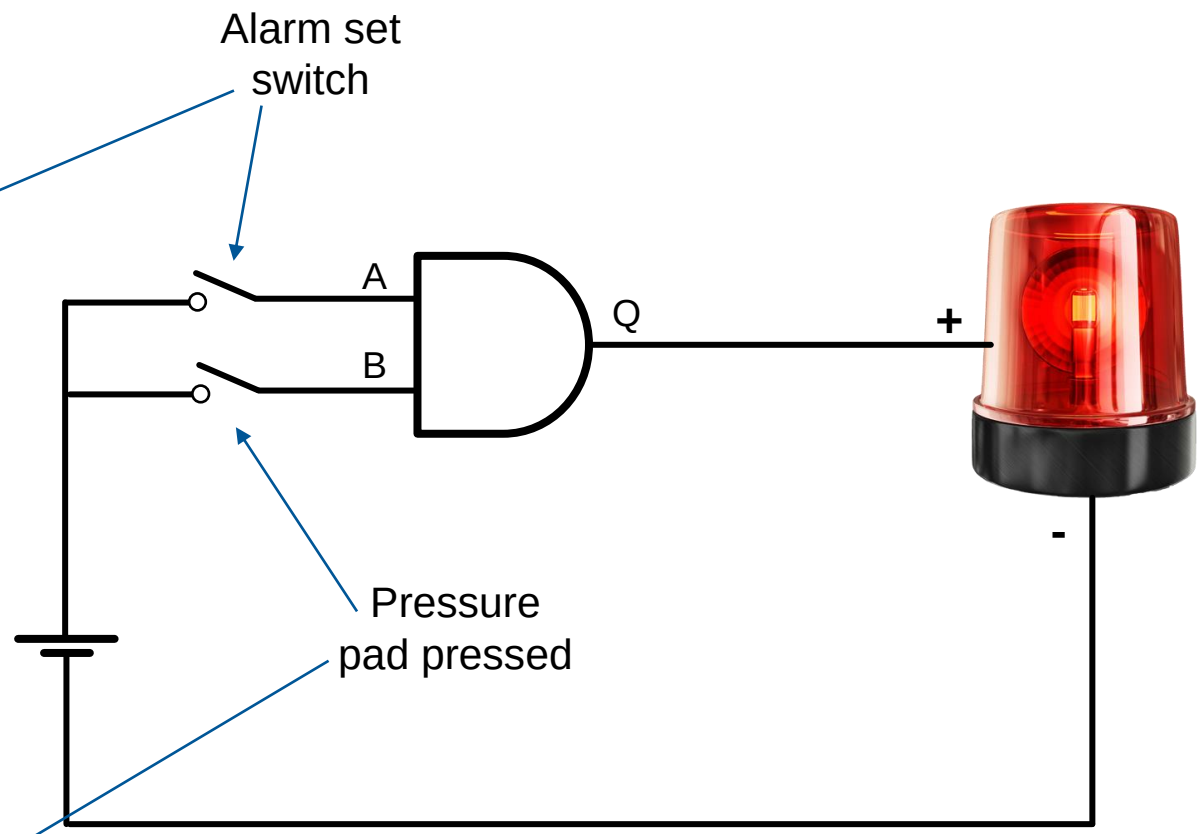
Example 1

We want to sound a siren when:

The alarm has been **set**

AND

Someone **steps on** the pressure sensor



Example 2

We want to sound a siren when:

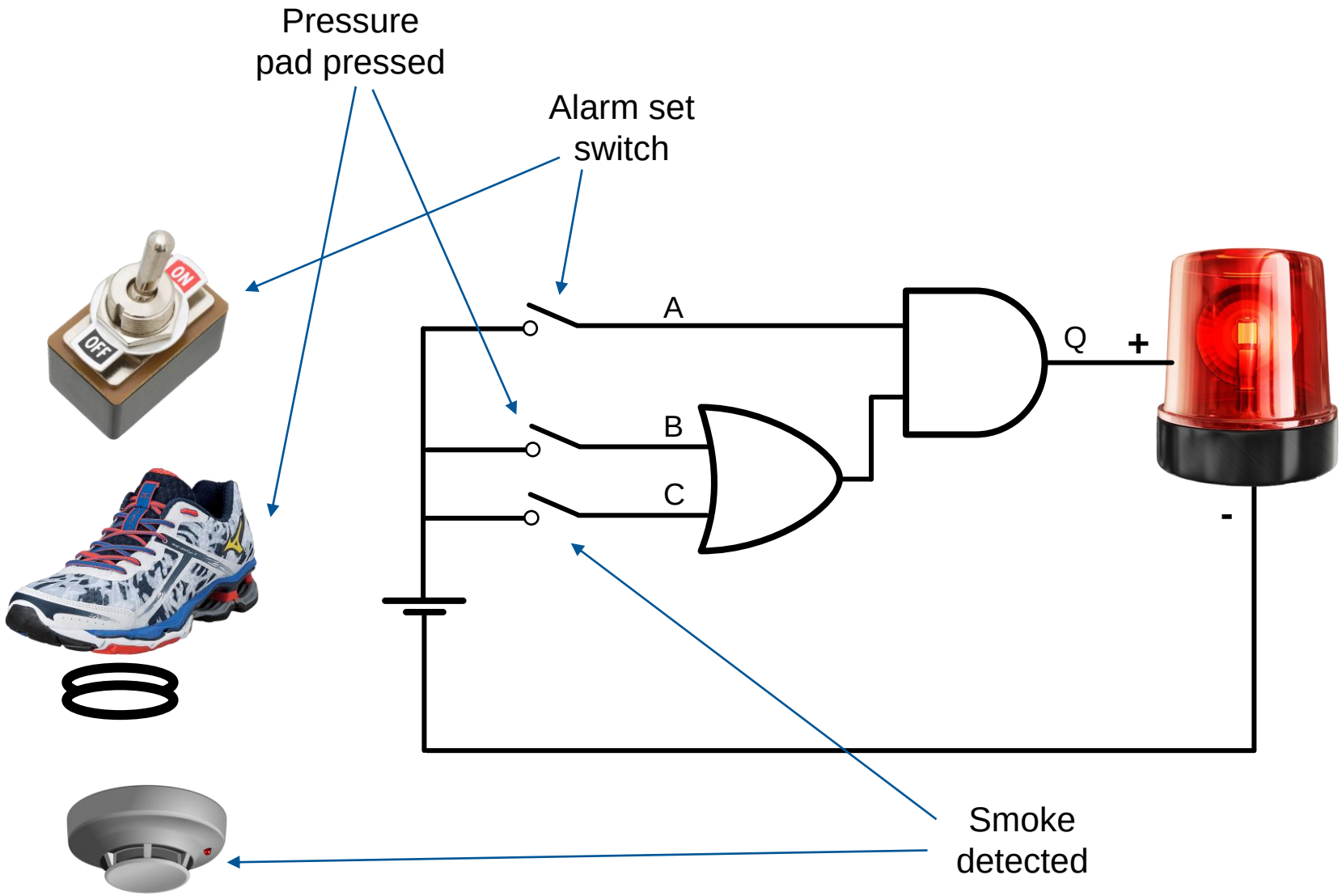
The alarm has been **set**

AND

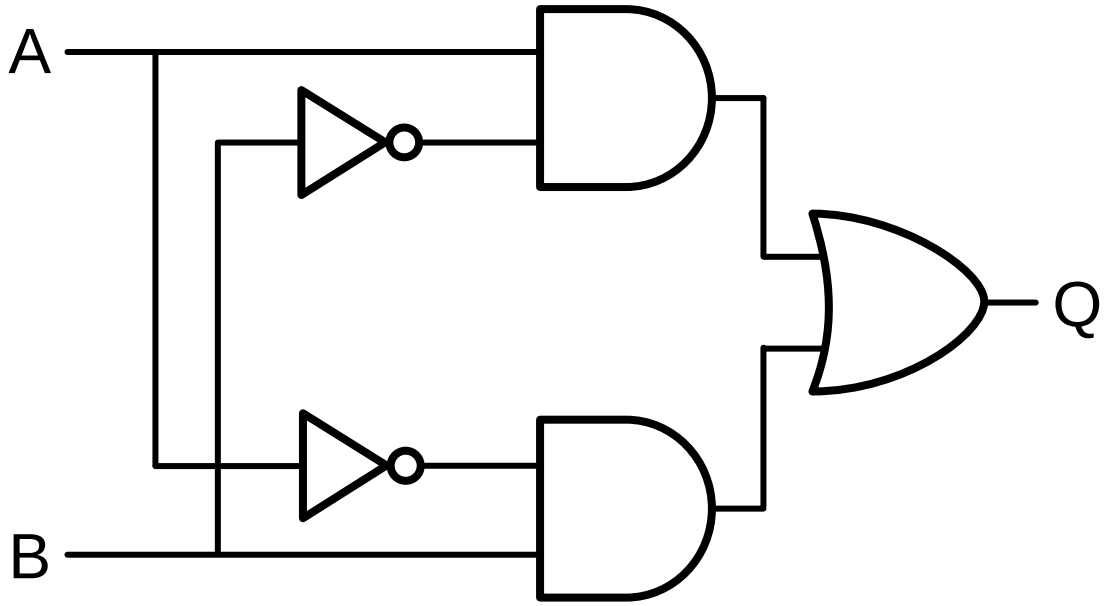
[Someone **steps on** the pressure sensor

OR

Smoke detected]



Example 3



Step 1 – Identify how many inputs there are

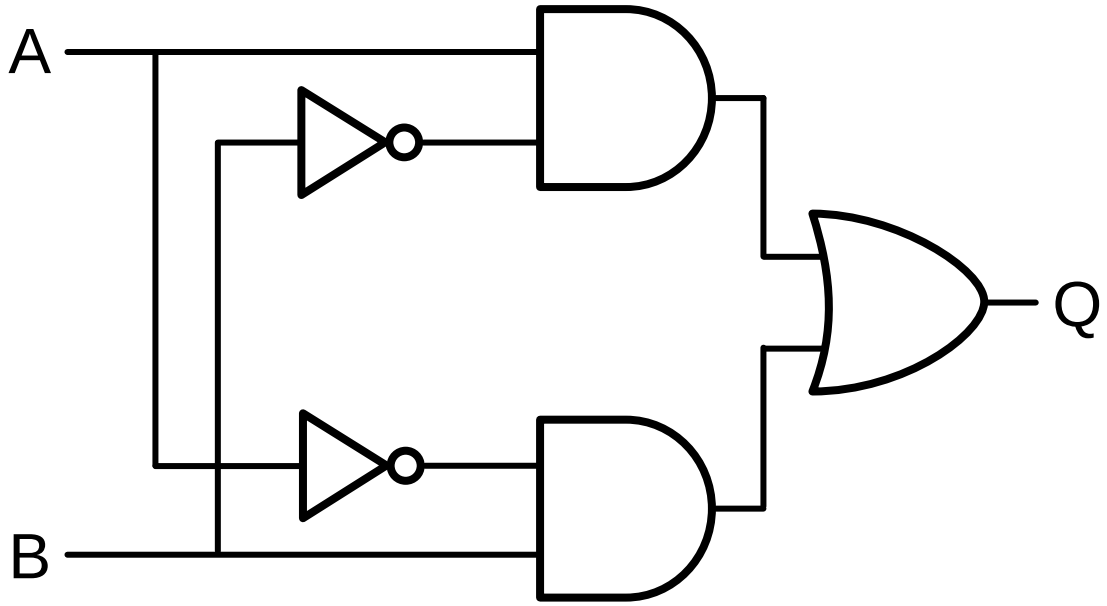
Step 2 – Draw a truth table with as many number of rows as possible combinations of input bits

Step 3 – Try each input combination in the logic gate

Step 4 – Propagate the “logic” all the way to output

Step 5 – Fill the truth table row by row

Example 3



Total inputs = 2

Total combinations possible = $2^n = 4$

4 rows in truth table

~~**Step 1 – Identify how many inputs there are**~~

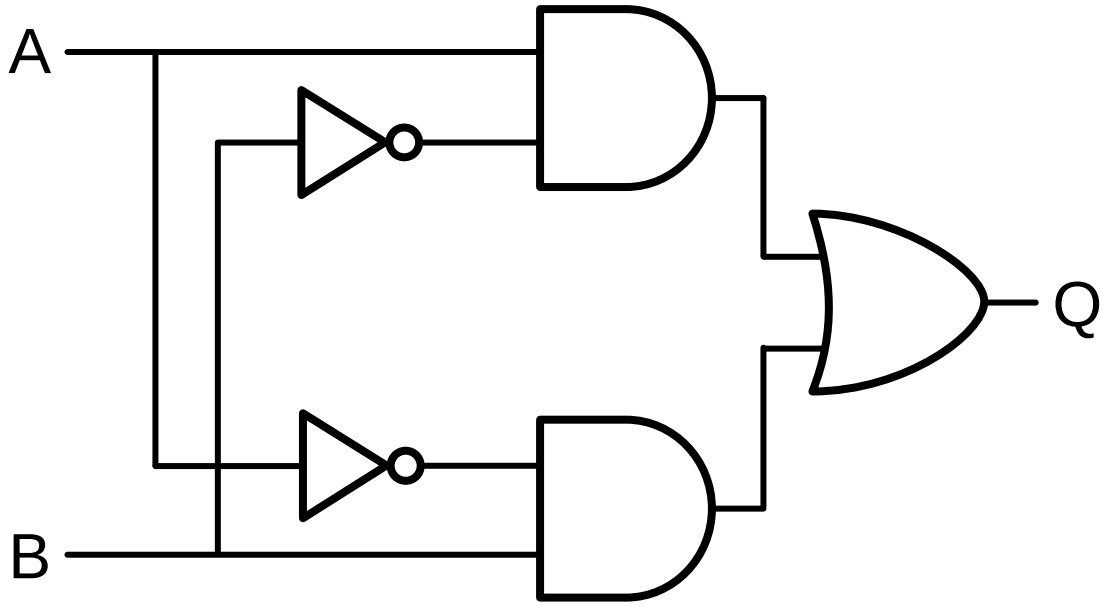
Step 2 – Draw a truth table with as many number of rows as possible combinations of input bits

Step 3 – Try each input combination in the logic gate

Step 4 – Propagate the “logic” all the way to output

Step 5 – Fill the truth table row by row

Example 3



Total inputs = 2

Total combinations possible = $2^n = 4$

4 rows in truth table

A	B	Q	Remark
0	0		
0	1		
1	0		
1	1		

~~**Step 1 – Identify how many inputs there are**~~

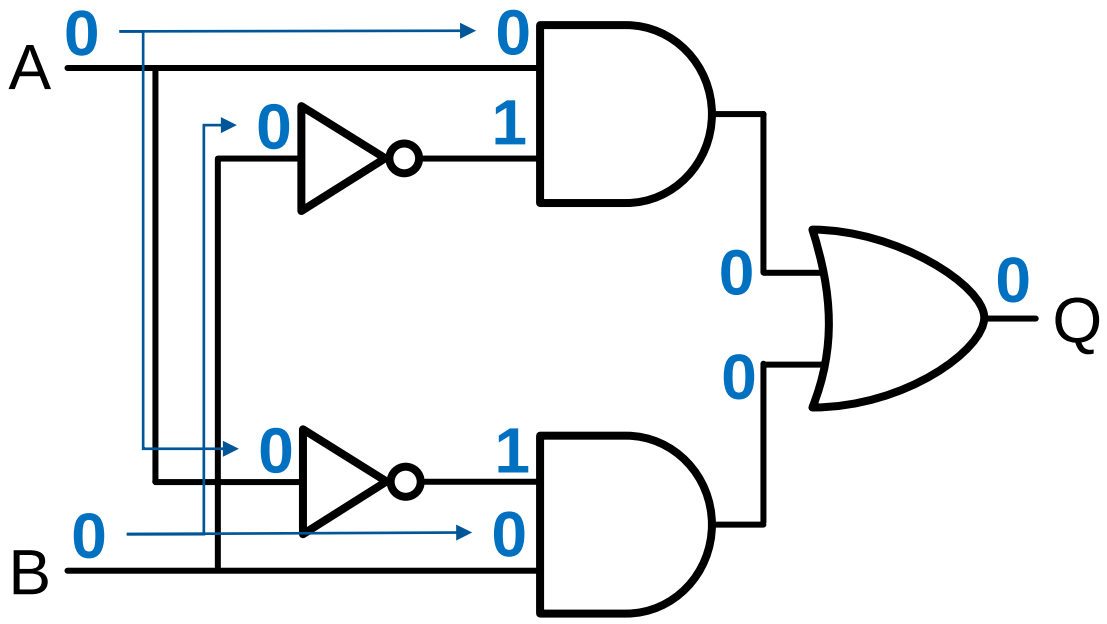
~~**Step 2 – Draw a truth table with as many number of rows as possible combinations of input bits**~~

Step 3 – Try each input combination in the logic gate

Step 4 – Propagate the "logic" all the way to output

Step 5 – Fill the truth table row by row

Example 3



Total inputs = 2

Total combinations possible = $2^n = 4$

4 rows in truth table

A	B	Q	Remark
0	0		
0	1		
1	0		
1	1		

~~**Step 1 – Identify how many inputs there are**~~

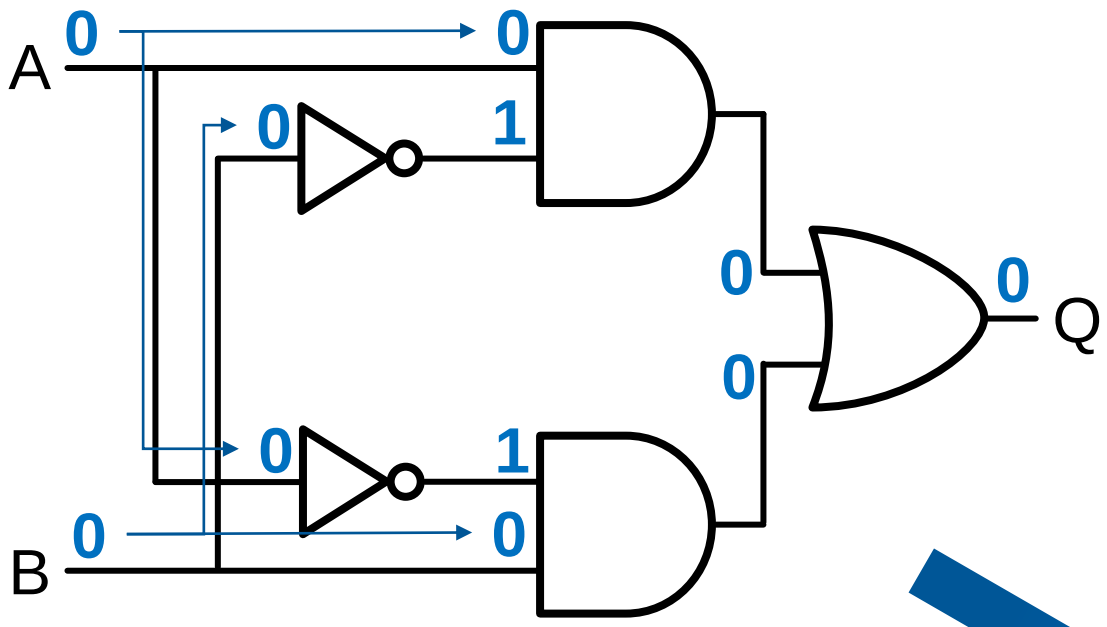
~~**Step 2 – Draw a truth table with as many number of rows as possible combinations of input bits**~~

Step 3 – Try each input combination in the logic gate

Step 4 – Propagate the "logic" all the way to output

Step 5 – Fill the truth table row by row

Example 3



Total inputs = 2

Total combinations possible = $2^n = 4$

4 rows in truth table

A	B	Q	Remark
0	0	**0**	This is XOR gate
0	1	**1**	
1	0	**1**	
1	1	**0**	

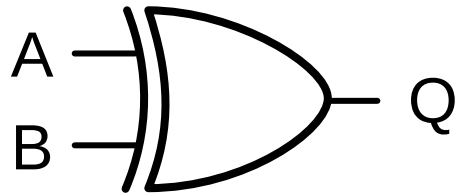
~~**Step 1 – Identify how many inputs there are**~~

~~**Step 2 – Draw a truth table with as many number of rows as possible combinations of input bits**~~

~~**Step 3 – Try each input combination in the logic gate**~~

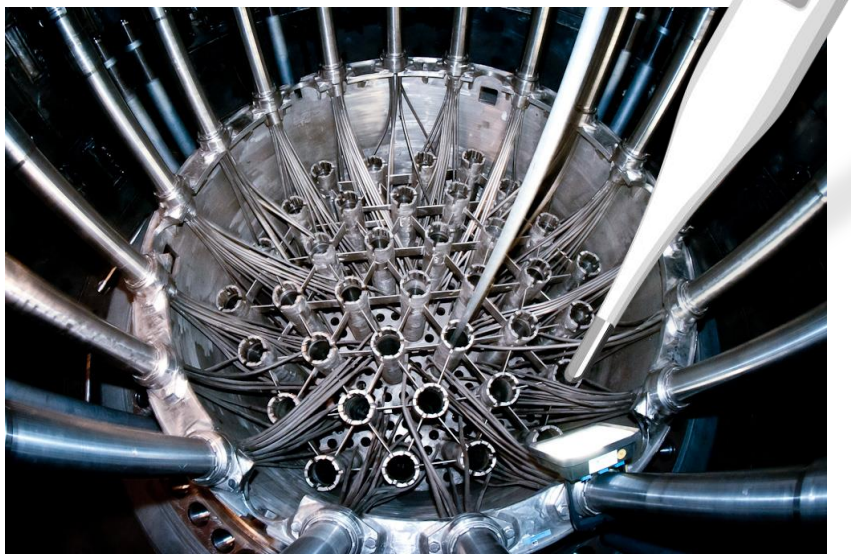
~~**Step 4 – Propagate the "logic" all the way to output**~~

~~**Step 5 – Fill the truth table row by row**~~



Example 4

- Imagine you are designing a circuit to monitor a digital thermometer embedded in a nuclear reactor
- You want to automatically shut off the reactor when the cooling fluid rises above 50°C
- It would also be bad if the coolant froze – shut down the reactor!
- Thermometer gives a 3-bit binary output in 10°C steps –
  - $2^3 =$ **8 levels**
  - Count from 0 to $2^3 - 1 =$ **7**
  - **0°C** to **80°C** range of output



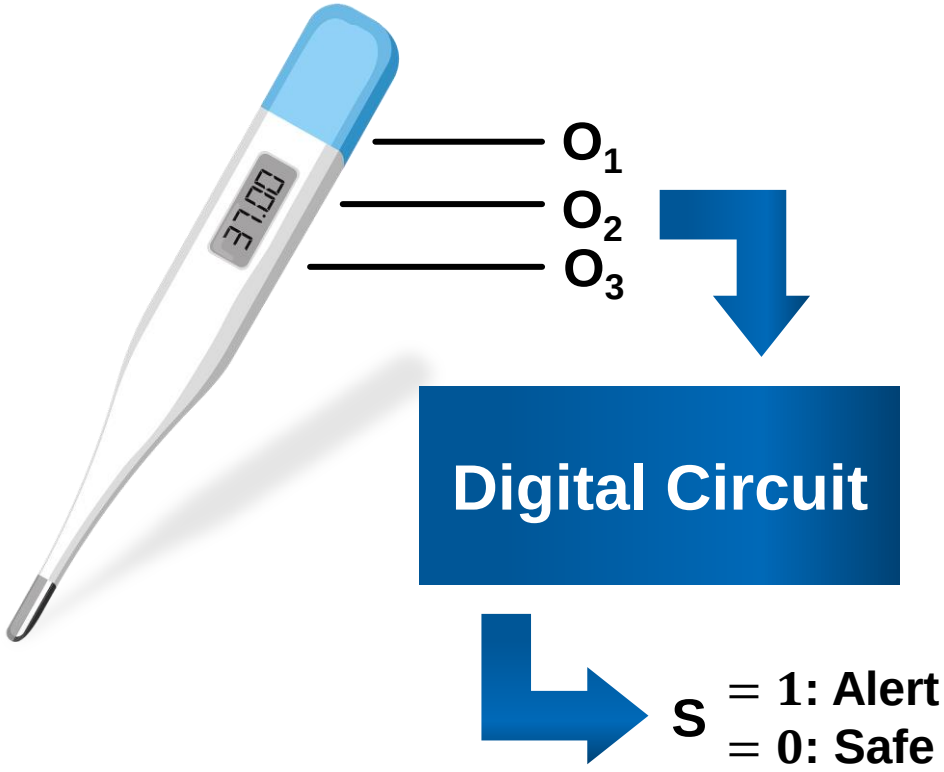
O_1
O_2
O_3

Digital Circuit

S = **1: Alert**
= **0: Safe**

Example 4

O_1	O_2	O_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



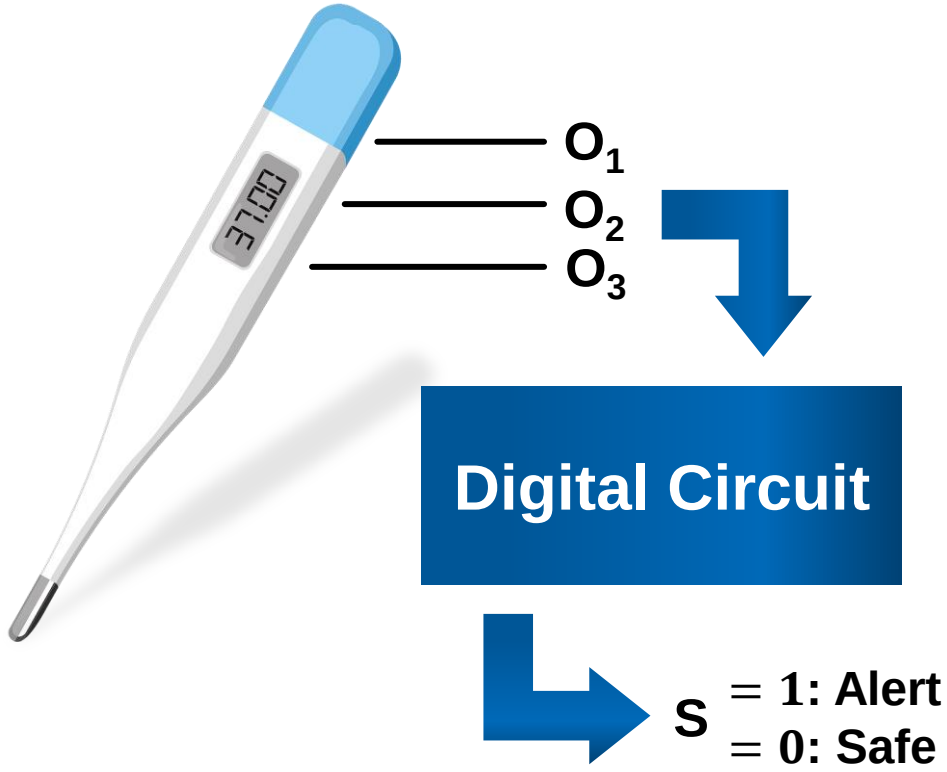
Remember what Step-1 was?

Identify all possible **combinations of input bits** and make the **truth table** – one row per combination

Example 4

O_1	O_2	O_3	Dec
0	0	0	**0**
0	0	1	**1**
0	1	0	**2**
0	1	1	**3**
1	0	0	**4**
1	0	1	**5**
1	1	0	**6**
1	1	1	**7**

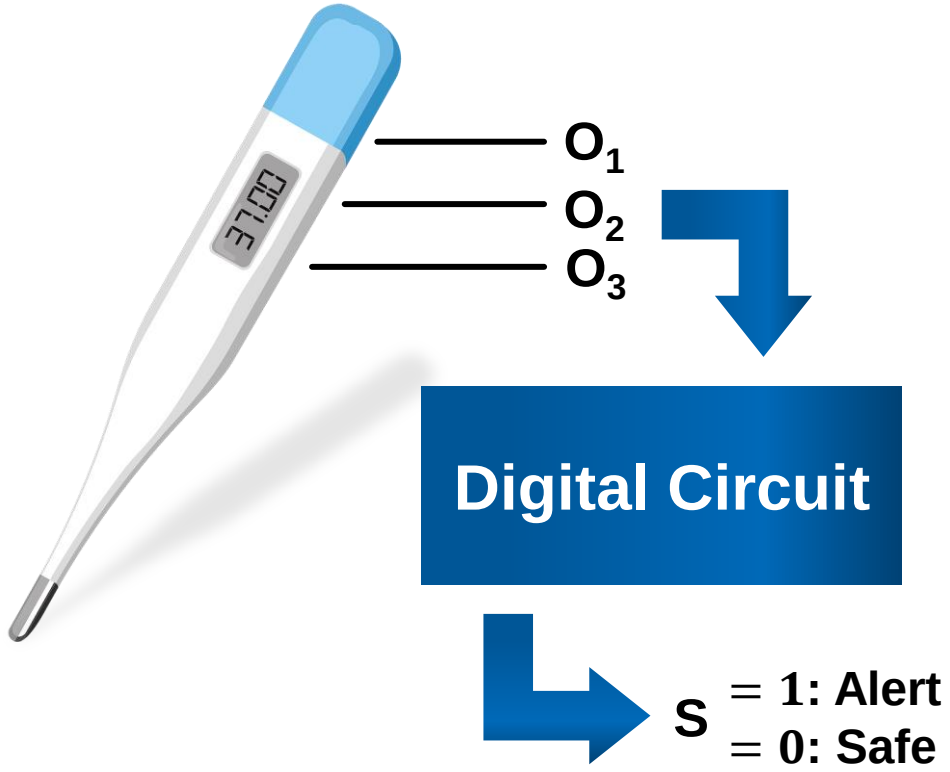
For this case, what would help is to have the **decimal conversion of the binary** number



Example 4

O_1	O_2	O_3	Dec	Temp
0	0	0	**0**	**0°C**
0	0	1	**1**	**10°C**
0	1	0	**2**	**20°C**
0	1	1	**3**	**30°C**
1	0	0	**4**	**40°C**
1	0	1	**5**	**50°C**
1	1	0	**6**	**60°C**
1	1	1	**7**	**70°C**

We know that the binary number represents the physical temperature in **steps of 10°C**, i.e., scaling factor of 10x

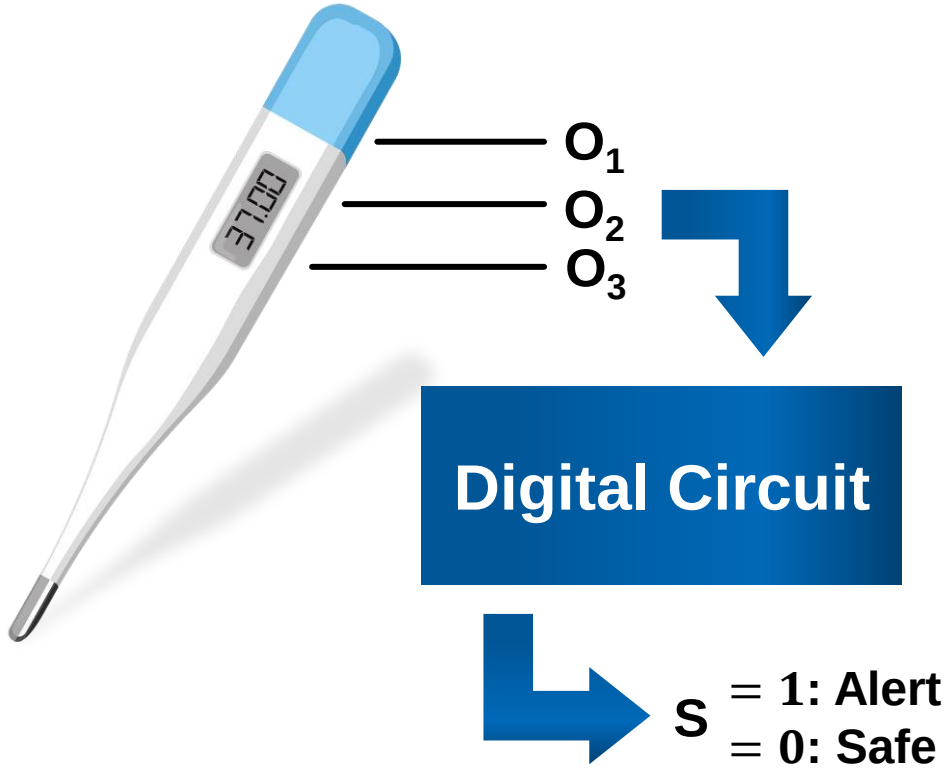


Example 4

O_1	O_2	O_3	Dec	Temp	S
0	0	0	0	0°C	1
0	0	1	1	10°C	0
0	1	0	2	20°C	0
0	1	1	3	30°C	0
1	0	0	4	40°C	0
1	0	1	5	50°C	0
1	1	0	6	60°C	1
1	1	1	7	70°C	1

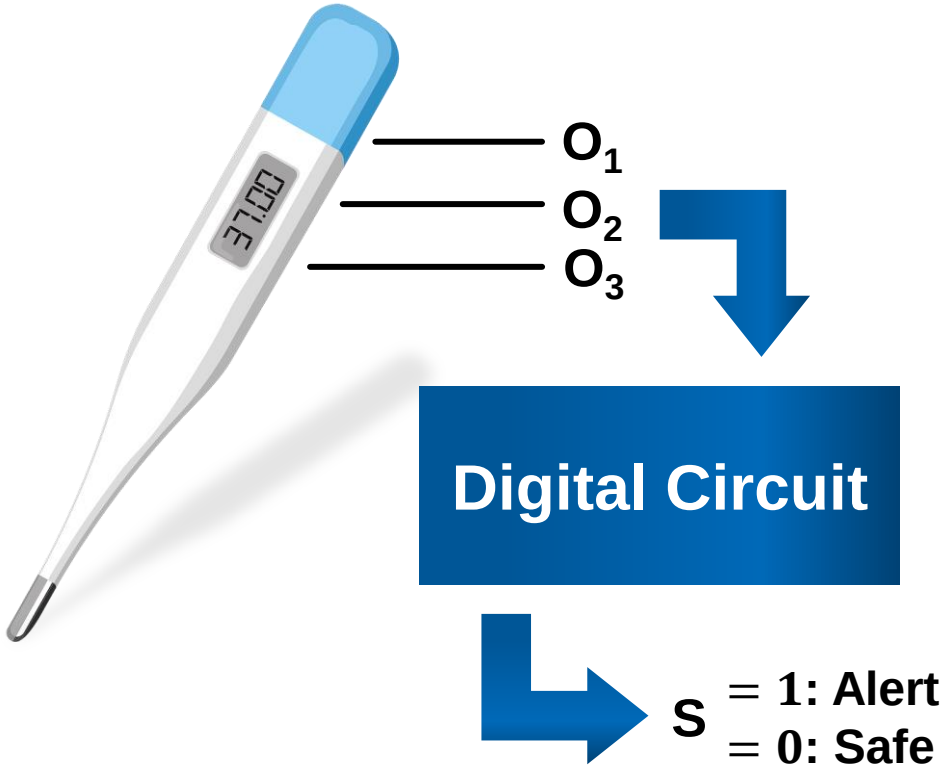
S = 1 indicating Alert when temperature is either <0°C OR >50°C

Otherwise, **S = 0 indicating Safe**



Example 4

O_1	O_2	O_3	Dec	Temp	S
0	0	0	**0**	**0°C**	**1**
0	0	1	**1**	**10°C**	**0**
0	1	0	**2**	**20°C**	**0**
0	1	1	**3**	**30°C**	**0**
1	0	0	**4**	**40°C**	**0**
1	0	1	**5**	**50°C**	**0**
1	1	0	**6**	**60°C**	**1**
1	1	1	**7**	**70°C**	**1**



Once you have the truth table, you have a choice:

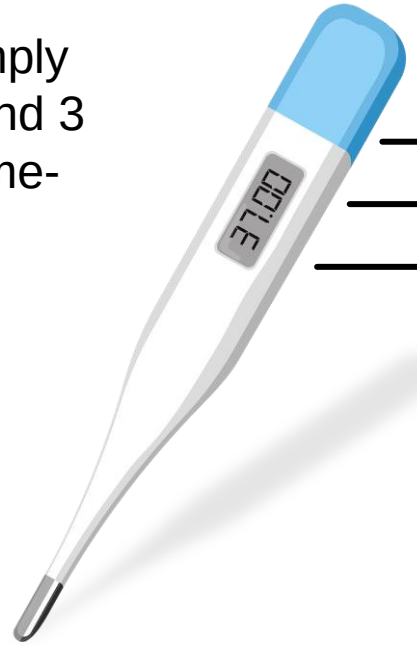
- Design for the **HI**, or **1s** (use AND/OR)
- Design for the **LO**, or **0s** (use NAND/NOR)

Example 4

O_1	O_2	O_3	Dec	Temp	S
0	0	0	0	0°C	1
0	0	1	1	10°C	0
0	1	0	2	20°C	0
0	1	1	3	30°C	0
1	0	0	4	40°C	0
1	0	1	5	50°C	0
1	1	0	6	60°C	1
1	1	1	7	70°C	1

The choice usually requires thinking about practical issues, like what would happen if a fault in the circuit occurs (concept of Active HI and Active LO logic, we don't need to study this), is the default state of output safe etc.

In our example, we will simply see that there are 5 LOs and 3 HIs. This means its less time-consuming to build for HIs.



O_1
O_2
O_3

Digital Circuit

S = **1: Alert**
= **0: Safe**

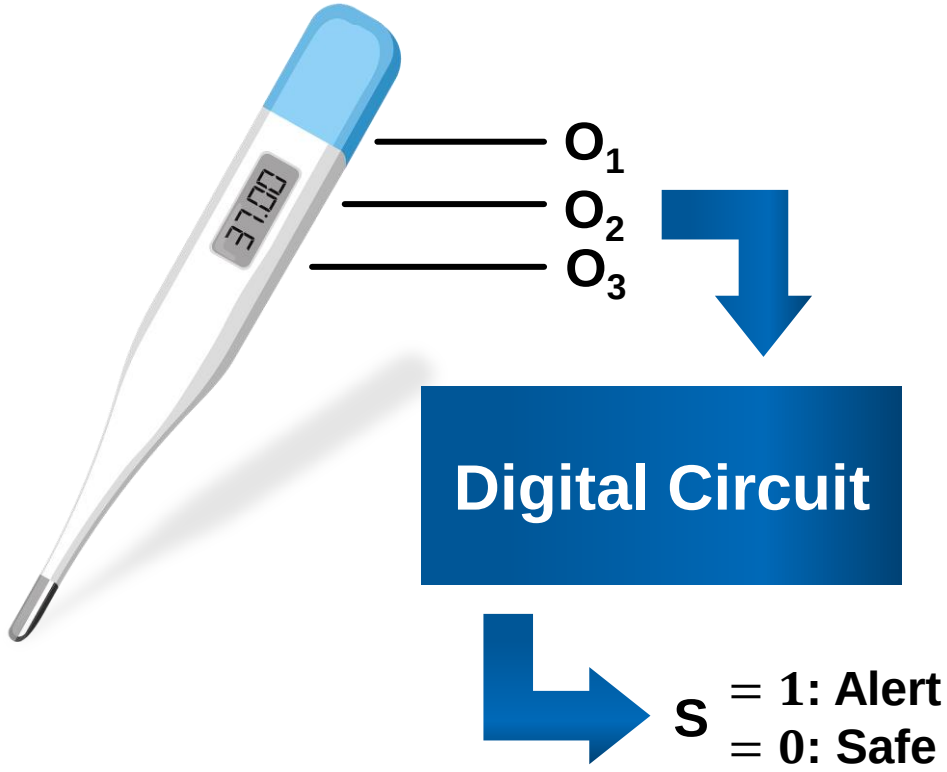
Once you have the truth table, you have a choice:

- Design for the **HI**, or **1s** (use AND/OR)
- Design for the **LO**, or **0s** (use NAND/NOR)

Example 4

O_1	O_2	O_3	Dec	Temp	S
0	0	0	0	0°C	1
0	0	1	1	10°C	0
0	1	0	2	20°C	0
0	1	1	3	30°C	0
1	0	0	4	40°C	0
1	0	1	5	50°C	0
1	1	0	6	60°C	1
1	1	1	7	70°C	1

Now let us try to think of this truth table in "if-then-else" decision-making process

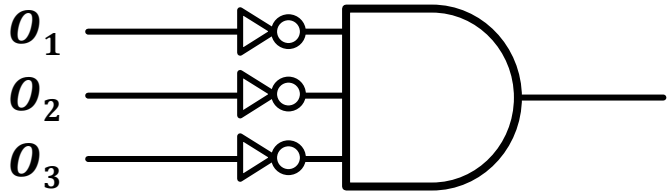


S=1 (as we said solving for HI) if:

- $O_1 = 0$ AND $O_2 = 0$ AND $O_3 = 0$ OR
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 0$ OR
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 1$

Example 4

O_1	O_2	O_3	Dec	Temp	S
0	0	0	0	0°C	1
0	0	1	1	10°C	0
0	1	0	2	20°C	0
0	1	1	3	30°C	0
1	0	0	4	40°C	0
1	0	1	5	50°C	0
1	1	0	6	60°C	1
1	1	1	7	70°C	1

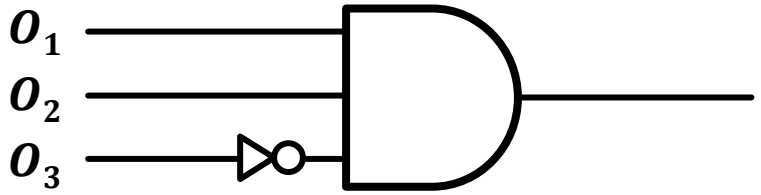


S=1 (as we said solving for HI) if:

- $O_1 = 0$ AND $O_2 = 0$ AND $O_3 = 0$ OR
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 0$ OR
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 1$

Example 4

O_1	O_2	O_3	Dec	Temp	S
0	0	0	0	0°C	1
0	0	1	1	10°C	0
0	1	0	2	20°C	0
0	1	1	3	30°C	0
1	0	0	4	40°C	0
1	0	1	5	50°C	0
1	1	0	6	60°C	1
1	1	1	7	70°C	1

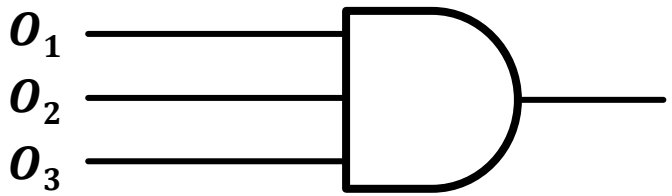


S=1 (as we said solving for HI) if:

- $O_1 = 0$ AND $O_2 = 0$ AND $O_3 = 0$ OR
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 0$ OR
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 1$

Example 4

O_1	O_2	O_3	Dec	Temp	S
0	0	0	**0**	**0°C**	**1**
0	0	1	**1**	**10°C**	**0**
0	1	0	**2**	**20°C**	**0**
0	1	1	**3**	**30°C**	**0**
1	0	0	**4**	**40°C**	**0**
1	0	1	**5**	**50°C**	**0**
1	1	0	**6**	**60°C**	**1**
1	1	1	**7**	**70°C**	**1**

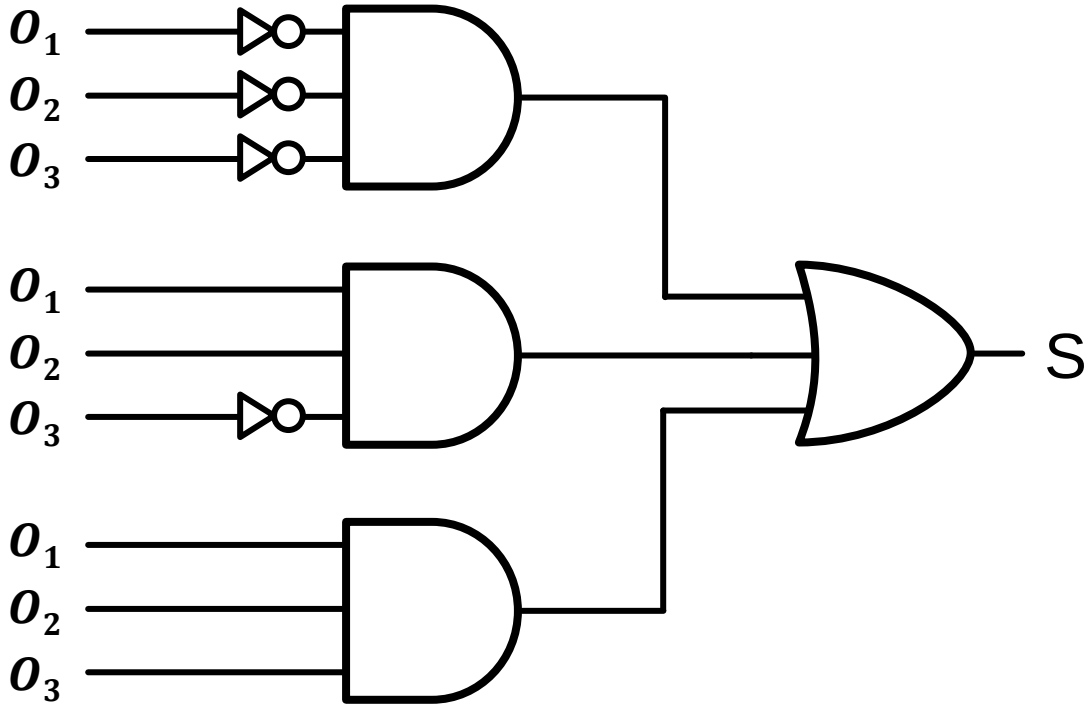


S=1 (as we said solving for HI) if:

- $O_1 = 0$ AND $O_2 = 0$ AND $O_3 = 0$ OR
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 0$ OR
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 1$

Example 4

O_1	O_2	O_3	Dec	Temp	S
0	0	0	0	0°C	1
0	0	1	1	10°C	0
0	1	0	2	20°C	0
0	1	1	3	30°C	0
1	0	0	4	40°C	0
1	0	1	5	50°C	0
1	1	0	6	60°C	1
1	1	1	7	70°C	1

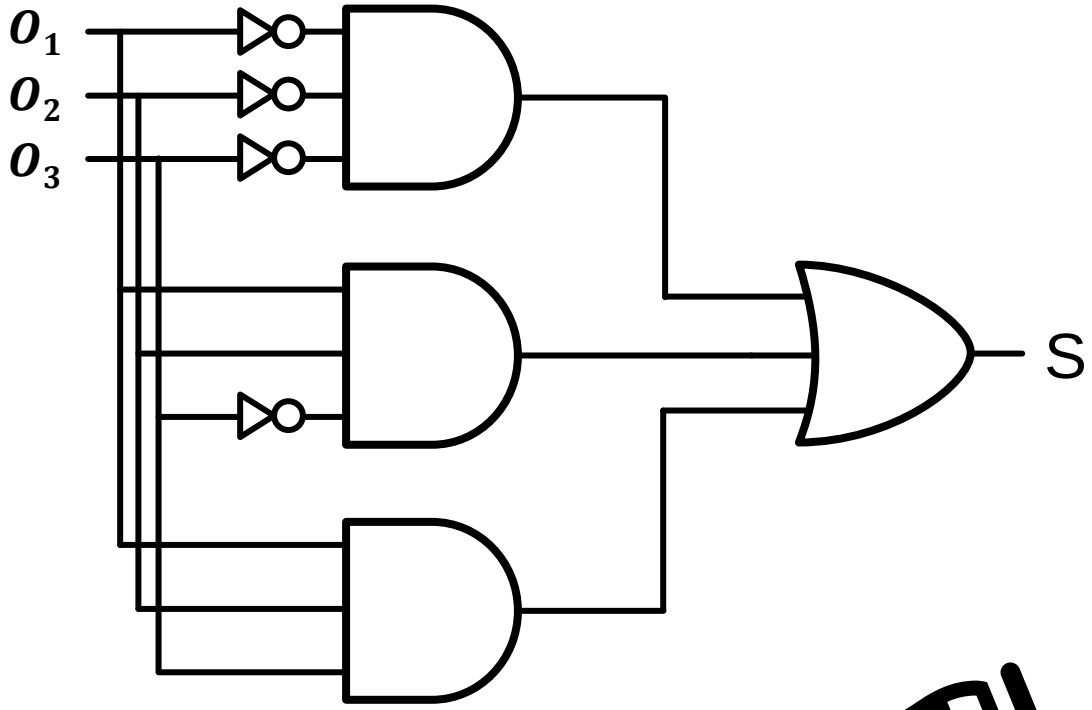


S=1 (as we said solving for HI) if:

- $O_1 = 0$ AND $O_2 = 0$ AND $O_3 = 0$ **OR**
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 0$ **OR**
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 1$

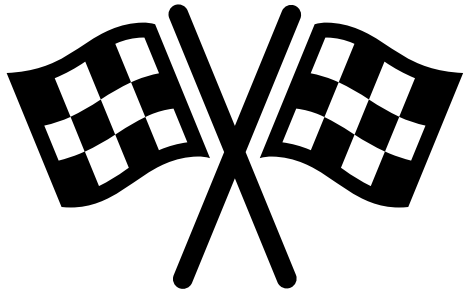
Example 4

O_1	O_2	O_3	Dec	Temp	S
0	0	0	0	0°C	1
0	0	1	1	10°C	0
0	1	0	2	20°C	0
0	1	1	3	30°C	0
1	0	0	4	40°C	0
1	0	1	5	50°C	0
1	1	0	6	60°C	1
1	1	1	7	70°C	1



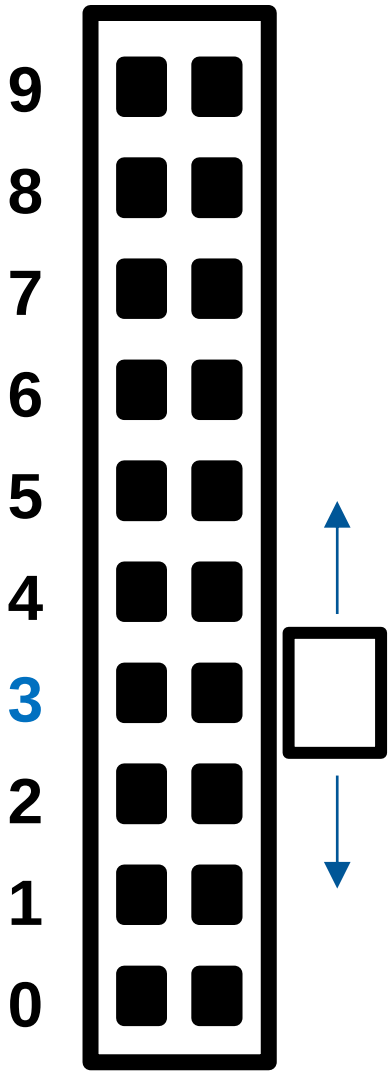
S=1 (as we said solving for HI) if:

- $O_1 = 0$ AND $O_2 = 0$ AND $O_3 = 0$ **OR**
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 0$ **OR**
- $O_1 = 1$ AND $O_2 = 1$ AND $O_3 = 1$

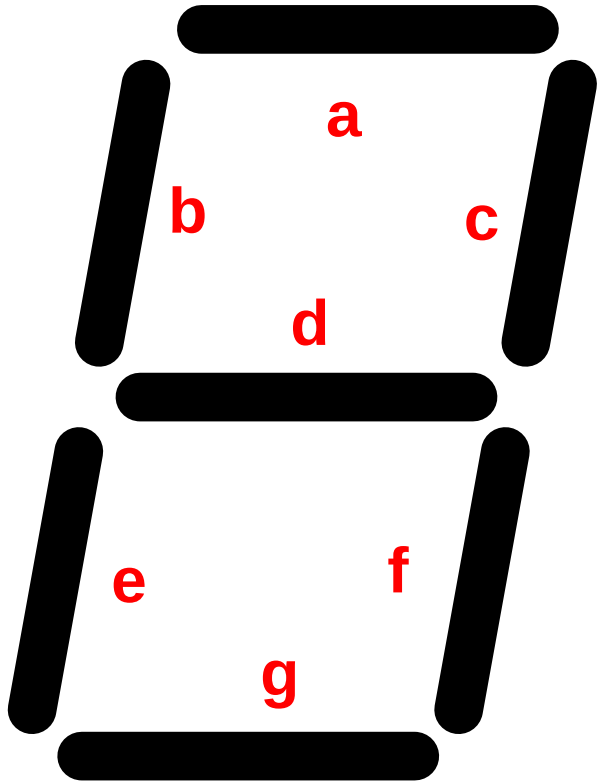


Example 5

What Floor are you on?
Digital Display

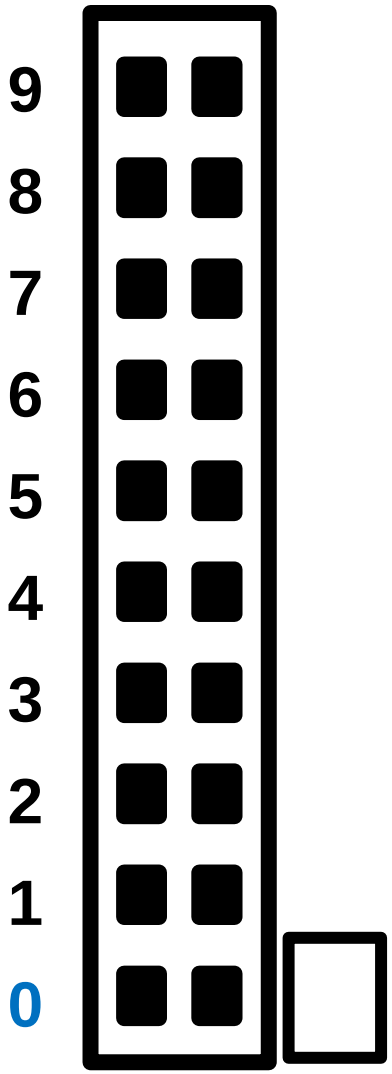


Lift

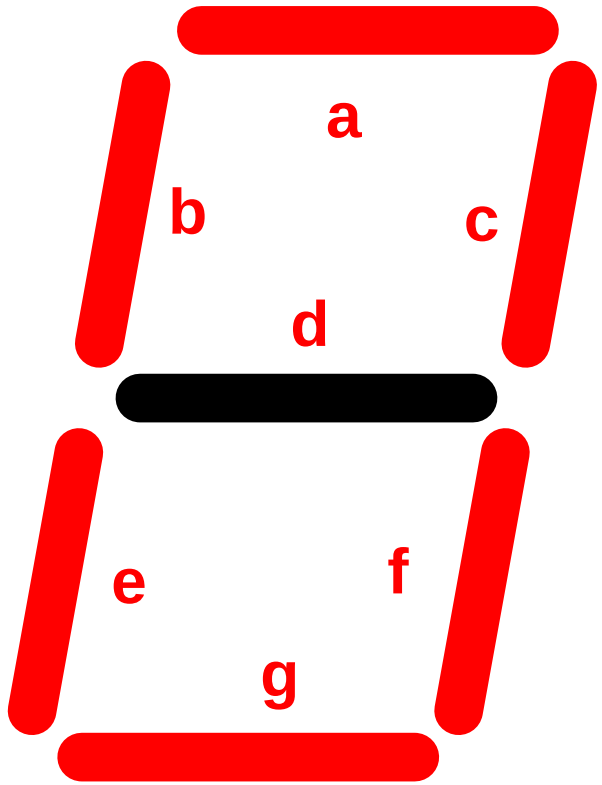


	a	b	c	d	e	f	g
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							

Example 5

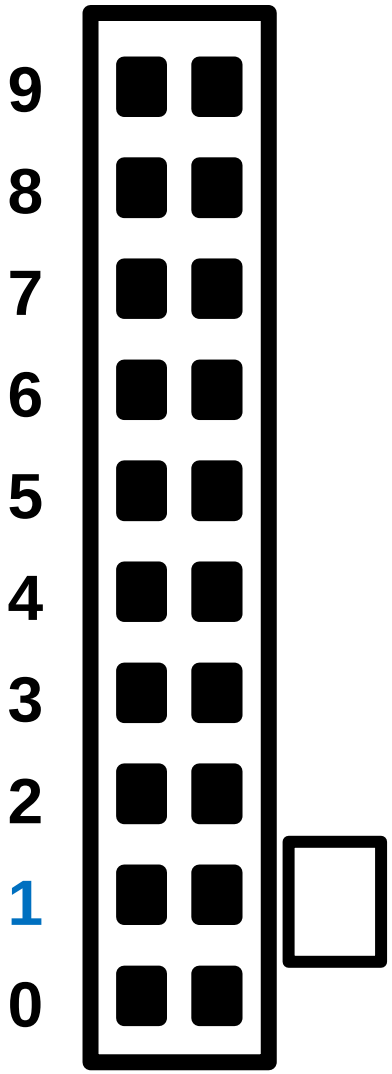


What Floor are you on?
Digital Display

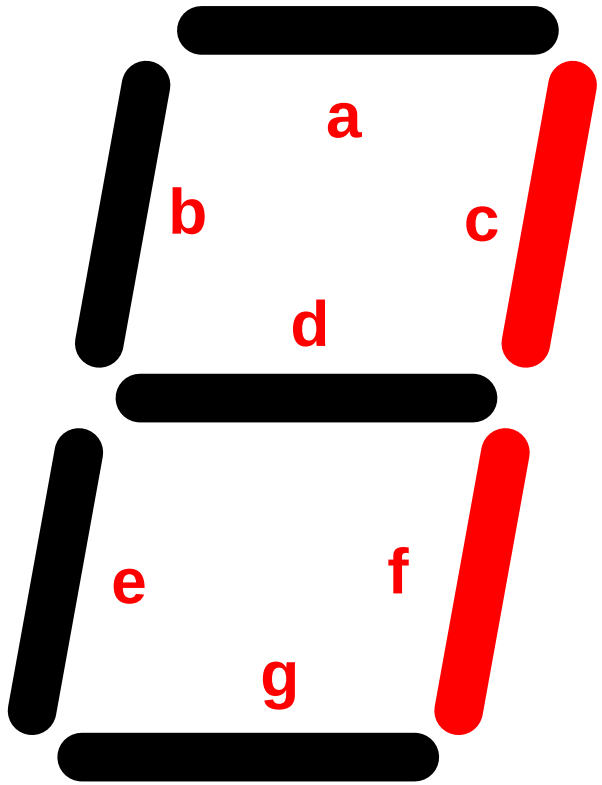


	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1							
2							
3							
4							
5							
6							
7							
8							
9							

Example 5

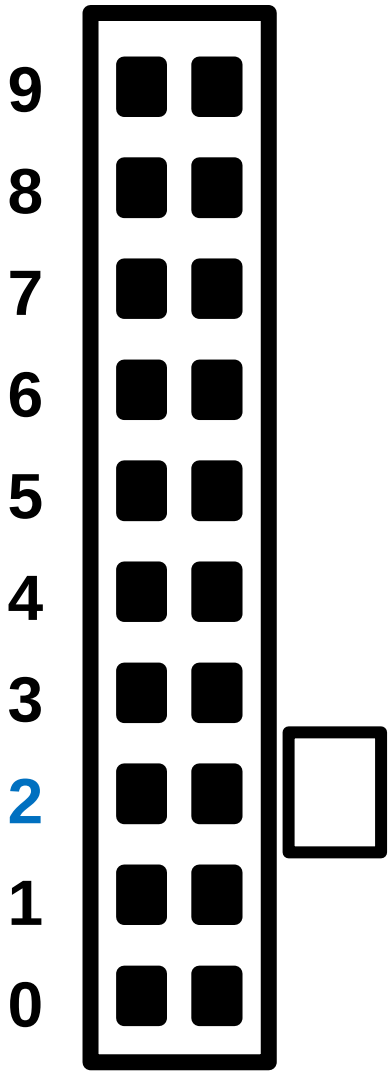


What Floor are you on?
Digital Display

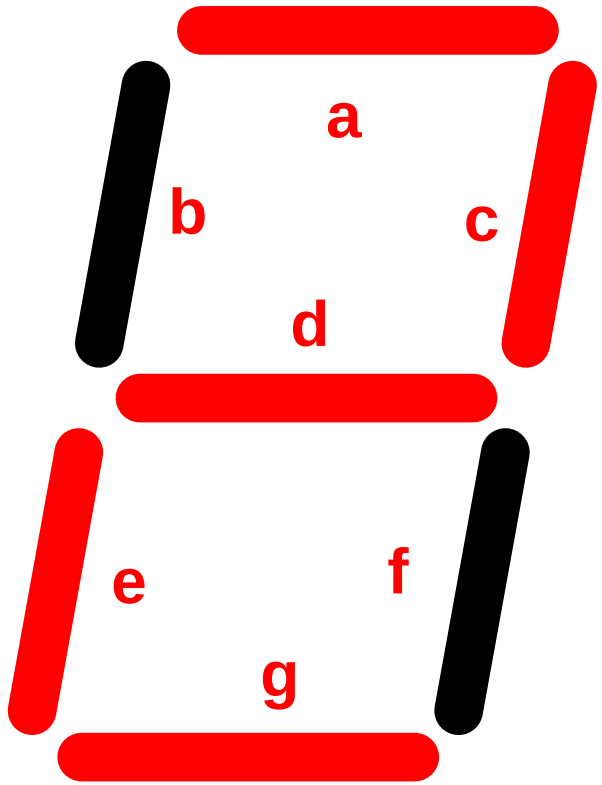


	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2							
3							
4							
5							
6							
7							
8							
9							

Example 5

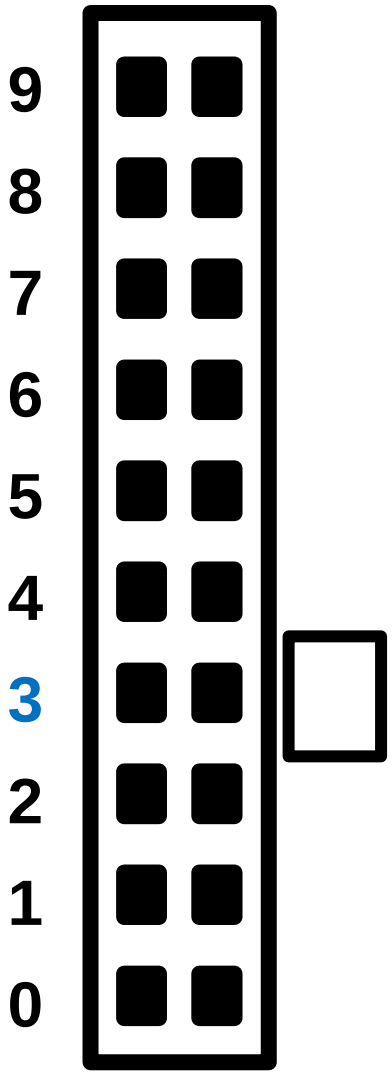


What Floor are you on?
Digital Display

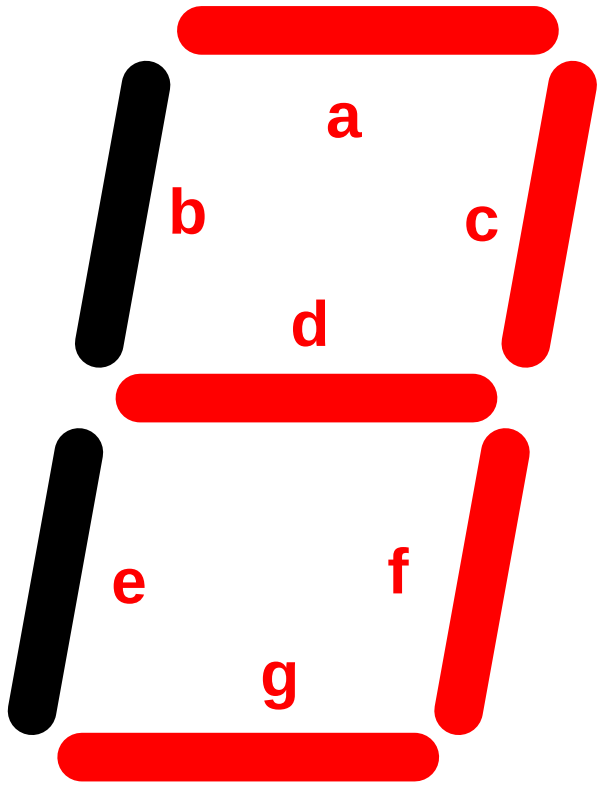


	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3							
4							
5							
6							
7							
8							
9							

Example 5

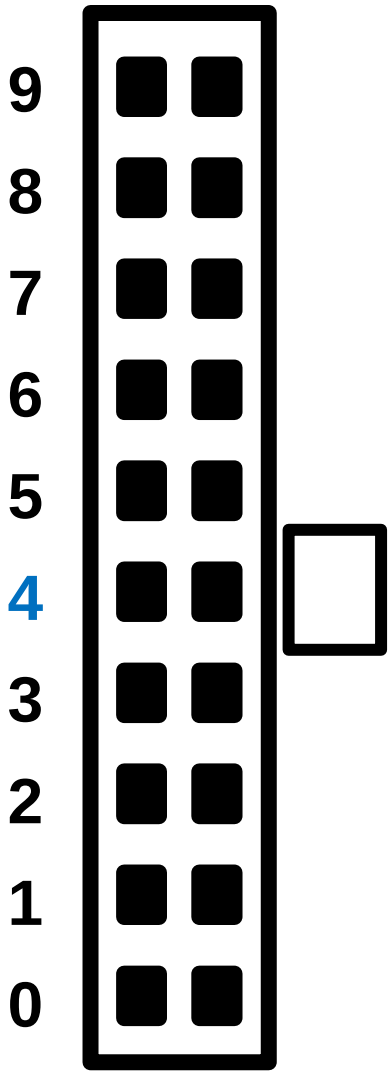


What Floor are you on?
Digital Display

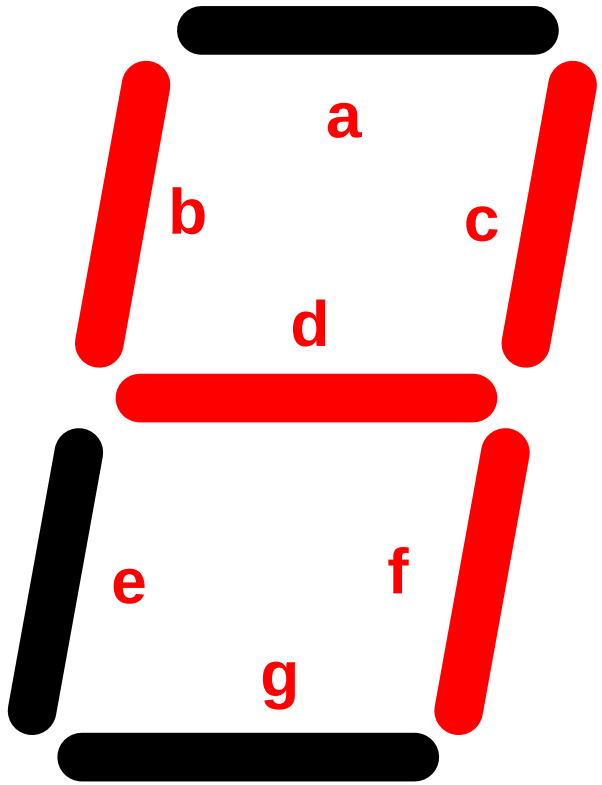


	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4							
5							
6							
7							
8							
9							

Example 5

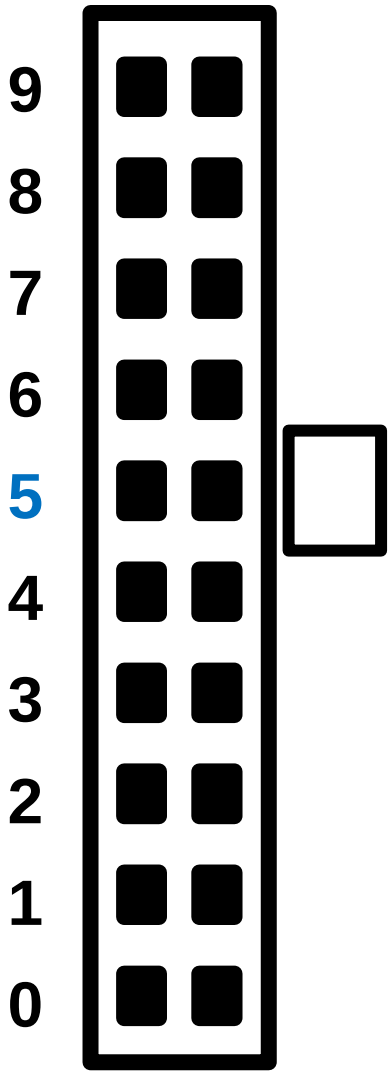


What Floor are you on?
Digital Display

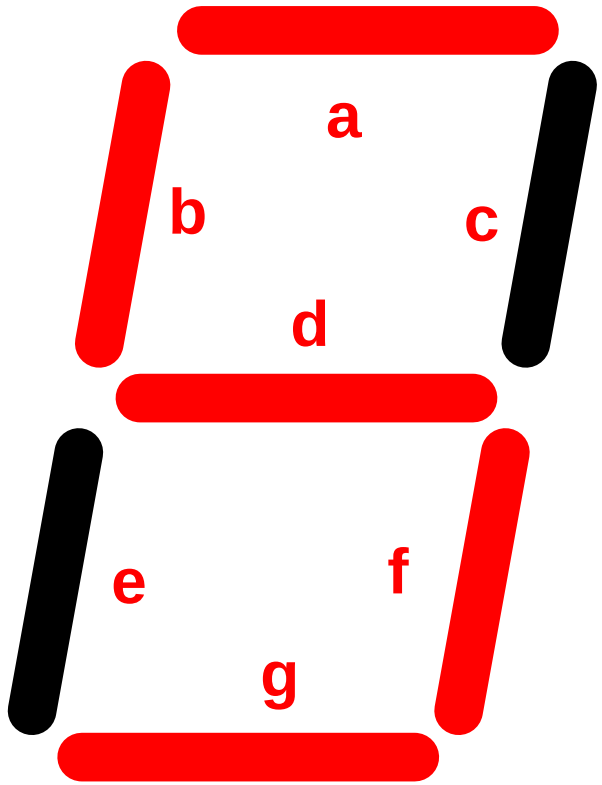


	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5							
6							
7							
8							
9							

Example 5

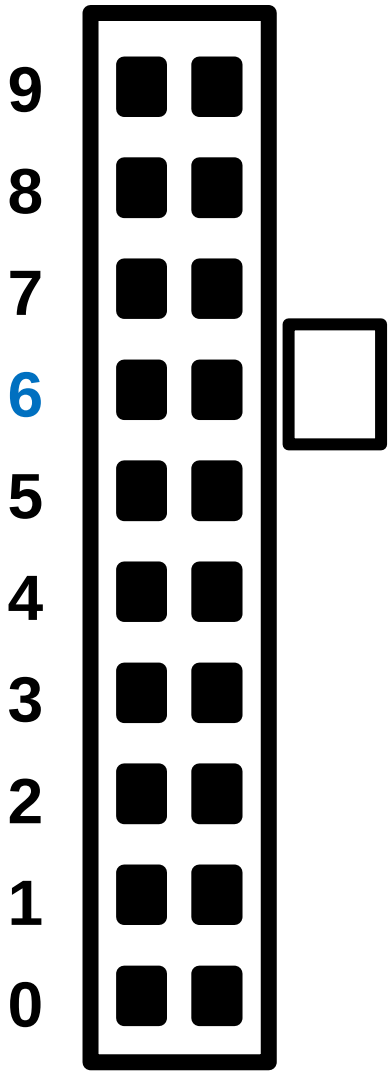


What Floor are you on?
Digital Display

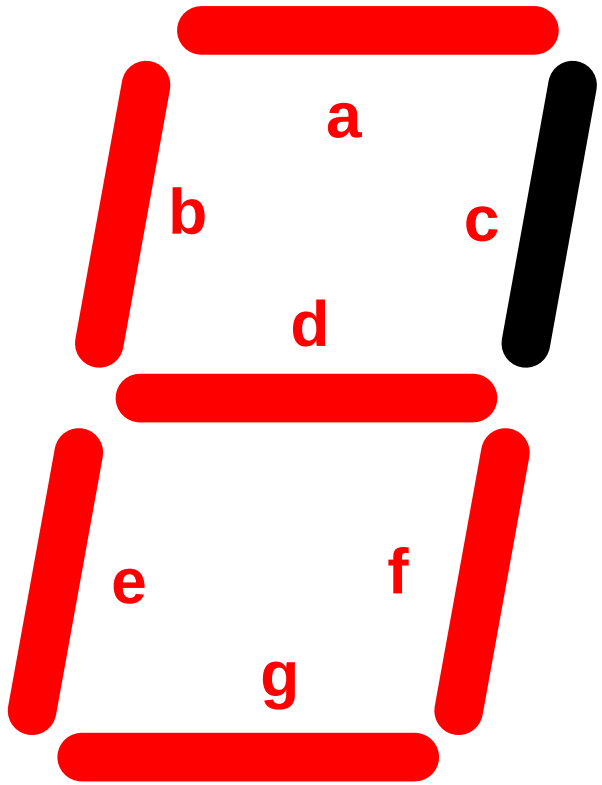


	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5	**1**	**1**	**0**	**1**	**0**	**1**	**1**
6							
7							
8							
9							

Example 5

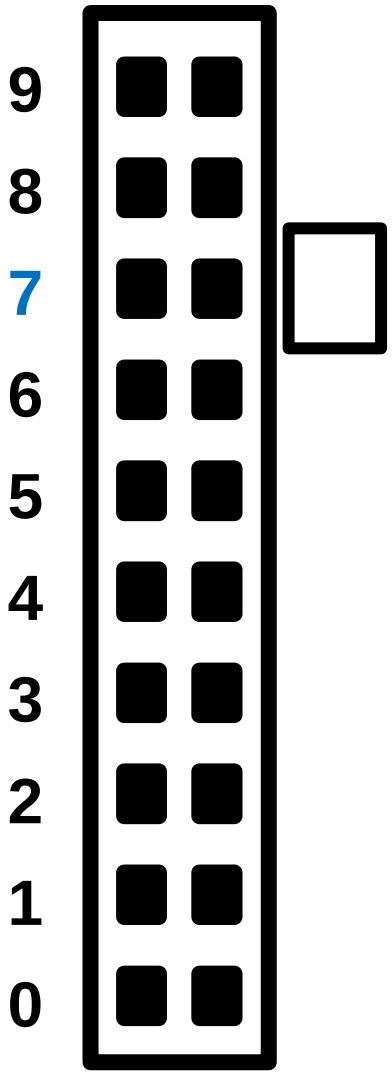


What Floor are you on?
Digital Display

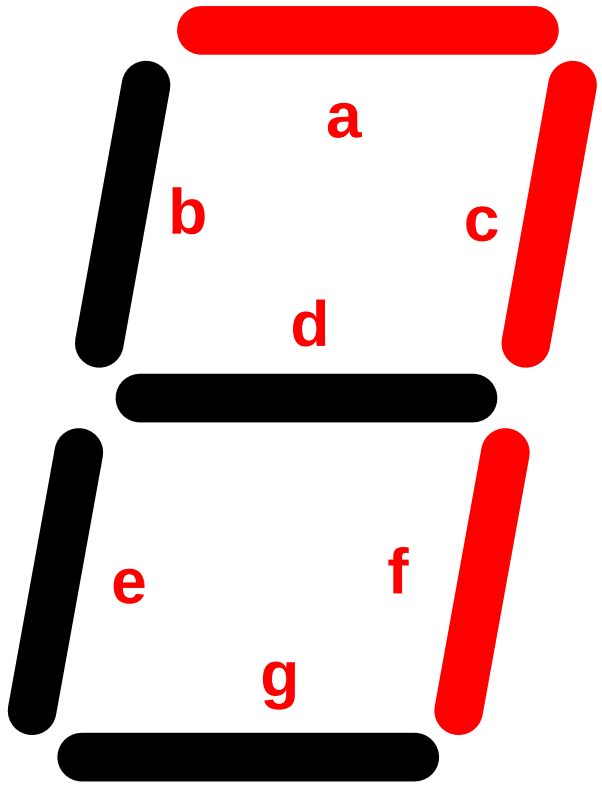


	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5	1	1	0	1	0	1	1
6	1	1	0	1	1	1	1
7							
8							
9							

Example 5

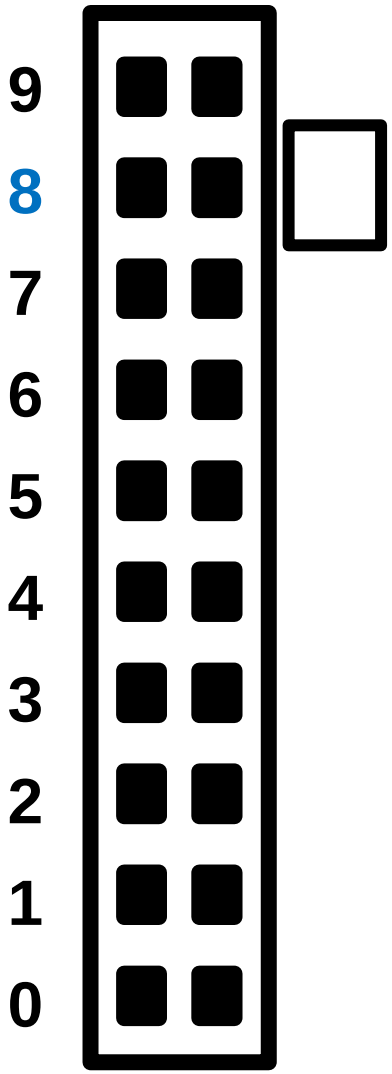


What Floor are you on?
Digital Display

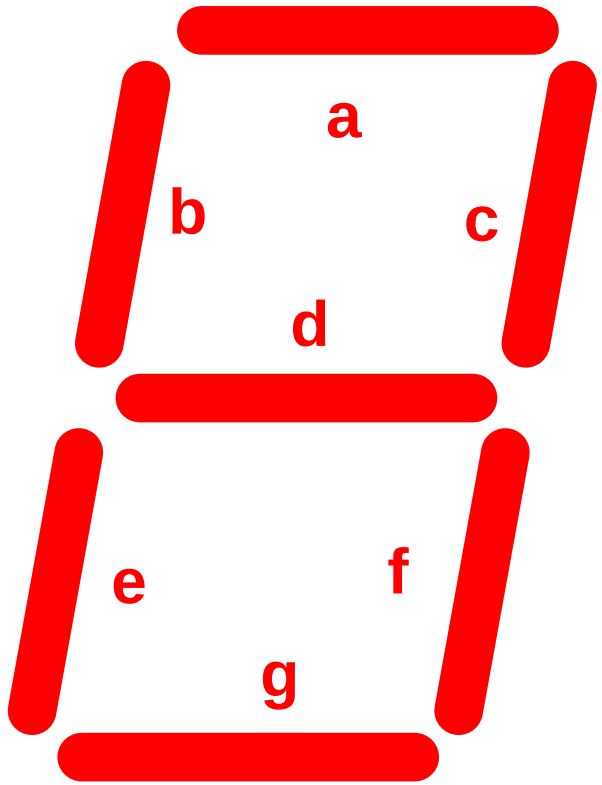


	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5	1	1	0	1	0	1	1
6	1	1	0	1	1	1	1
7	**1**	**0**	**1**	**0**	**0**	**1**	**0**
8							
9							

Example 5

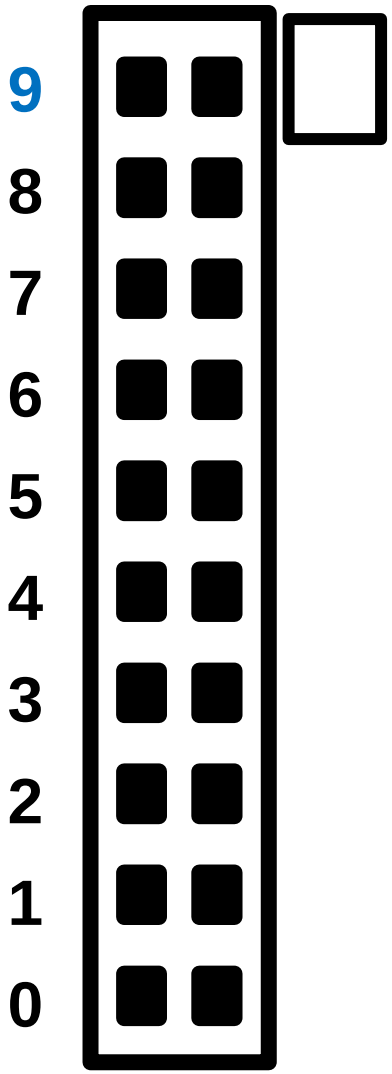


What Floor are you on?
Digital Display

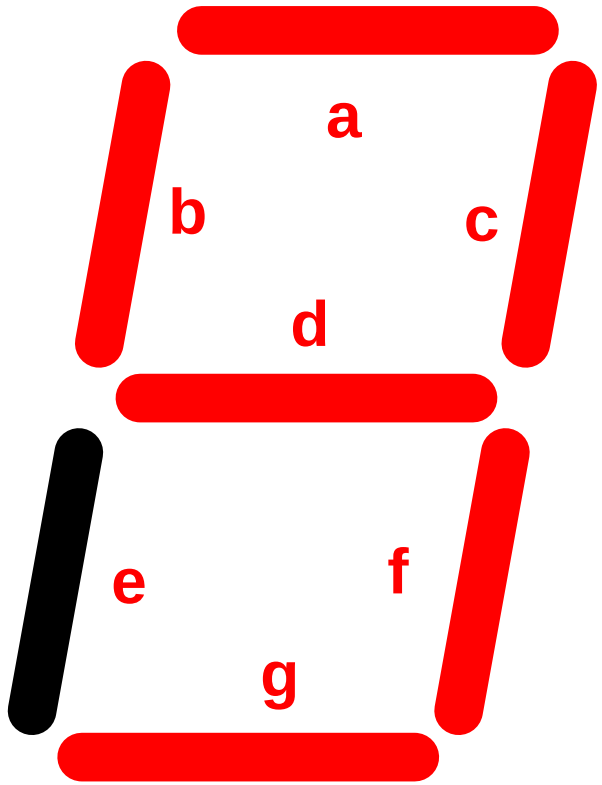


	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5	1	1	0	1	0	1	1
6	1	1	0	1	1	1	1
7	1	0	1	0	0	1	0
8	1	1	1	1	1	1	1
9							

Example 5

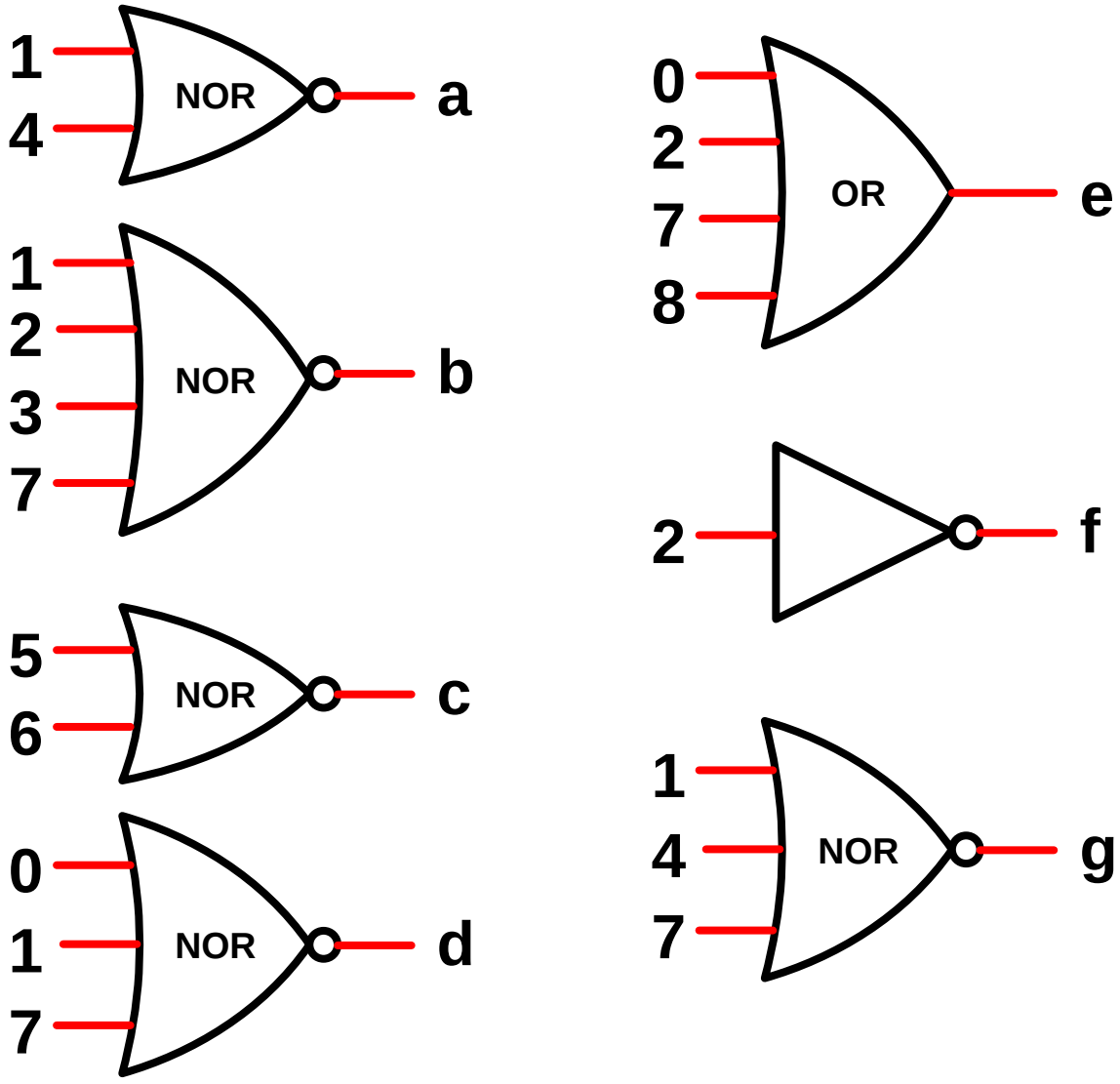


What Floor are you on?
Digital Display



	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5	1	1	0	1	0	1	1
6	1	1	0	1	1	1	1
7	1	0	1	0	0	1	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Digital Design Using Logic Gates



	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5	1	1	0	1	0	1	1
6	1	1	0	1	1	1	1
7	1	0	1	0	0	1	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Summary

- Introduction to **Information**
  - **Exchange of information** – essence of any machine!
  - **Digital** v **Analog** signal
  - **Accuracy v Precision**
- **Electronics**
  - Revisit Engineering
  - Information in Electronics and Software
  - Common terminologies (**PCB**, **IC**)
- Logic **Gates**

University of Nottingham
UK | CHINA | MALAYSIA

Attendance

UP|PHY|B1-MMME2051EMD