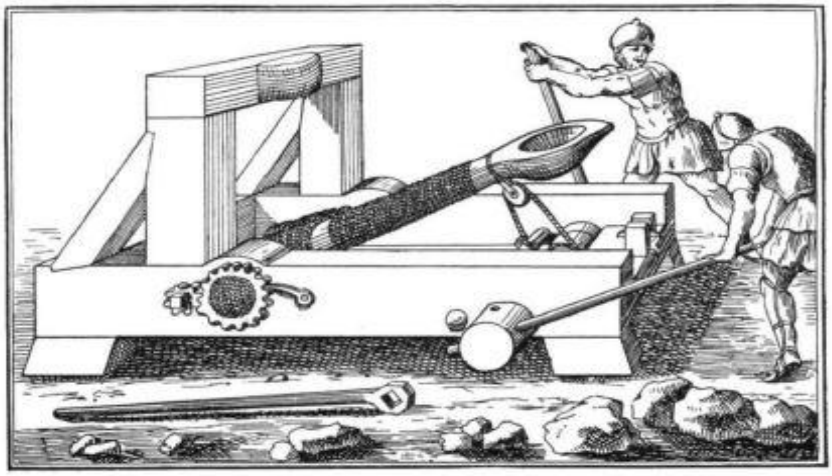
Gaurav Pahwa 102003087 2c04

DEPARTMENT
OF
ELECTRONICS AND COMMUNICATION ENGINEERING

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Handout/Assignment-1
for
Engineering Design Project-I (UTA013)



INSTRUCTOR INCHARGE

ASSIGNMENT - 1
INPUT / OUTPUT INTERFACE DESIGN

Exercise 1 (A) – Blink

Use Tinkercad to blink the LED with an Arduino for the amount of time equal to the sum of last two digits of your Roll number, leaving the amount of time the LED is 'OFF' to 2 second. *(For example, if your Roll No. is 1019***27, the LED must be kept on for 9 second).*

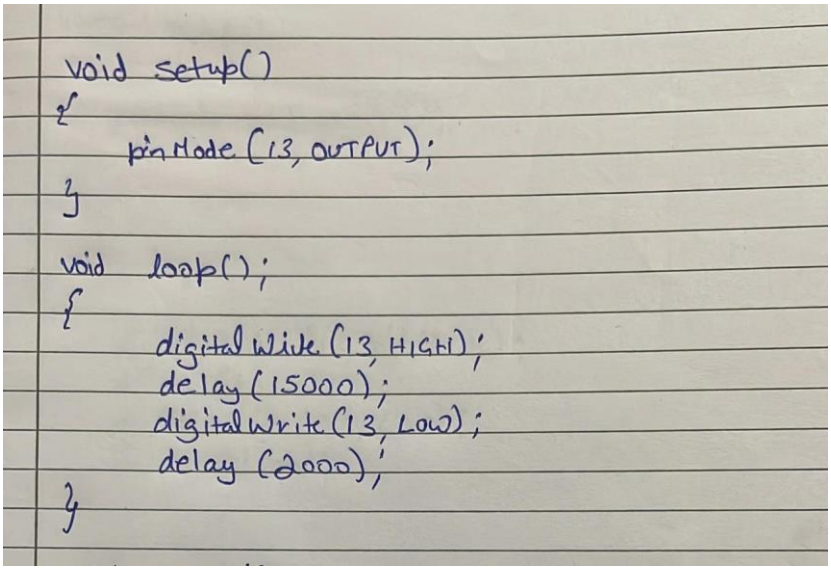
Hardware/Software Required

- Arduino Board
- Breadboard
- Tinkercad Software tool (https://www.tinkercad.com/)
- LED and Resistors

Circuit description

To build the circuit, attach a 100-ohm resistor to either leg of the diode. Attach the leg of the LED connected to the flat edge of the body (the negative leg, called the cathode) to ground. Connect the remaining leg (the positive leg, called the anode) to pin 13. Then plug your Arduino board into your computer, start the Arduino program, and upload the code.

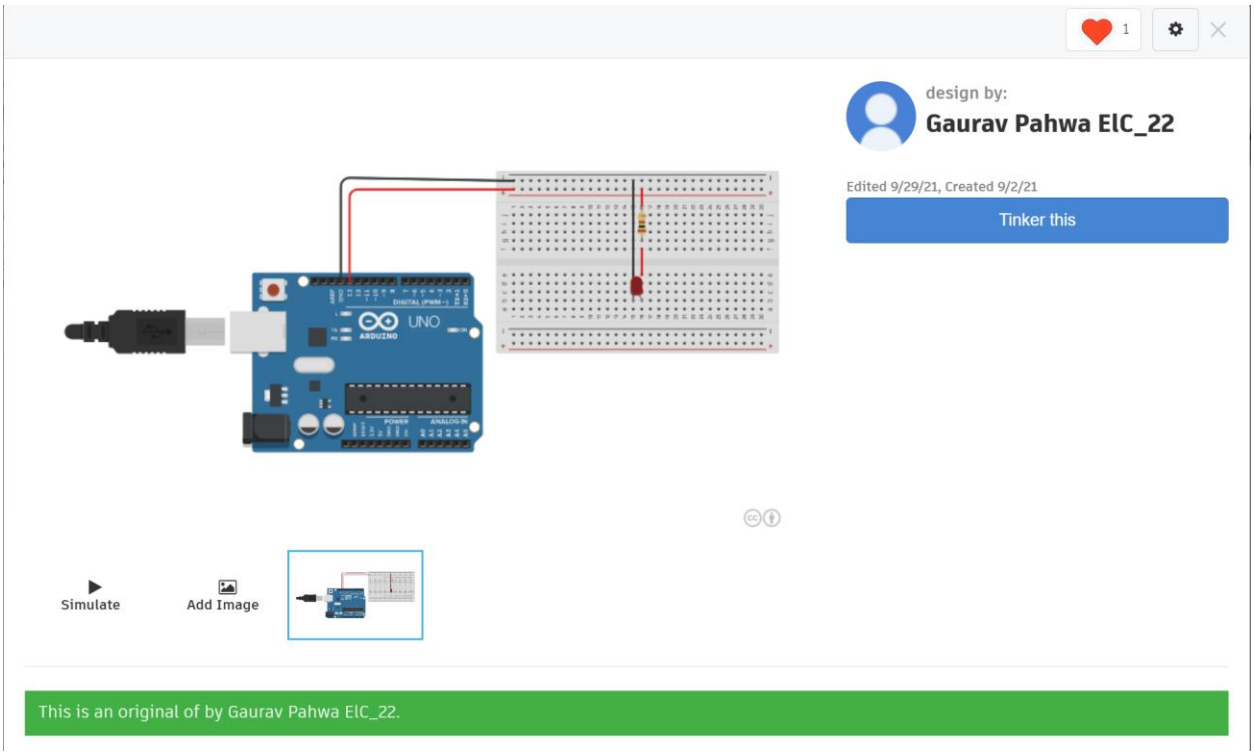
Code

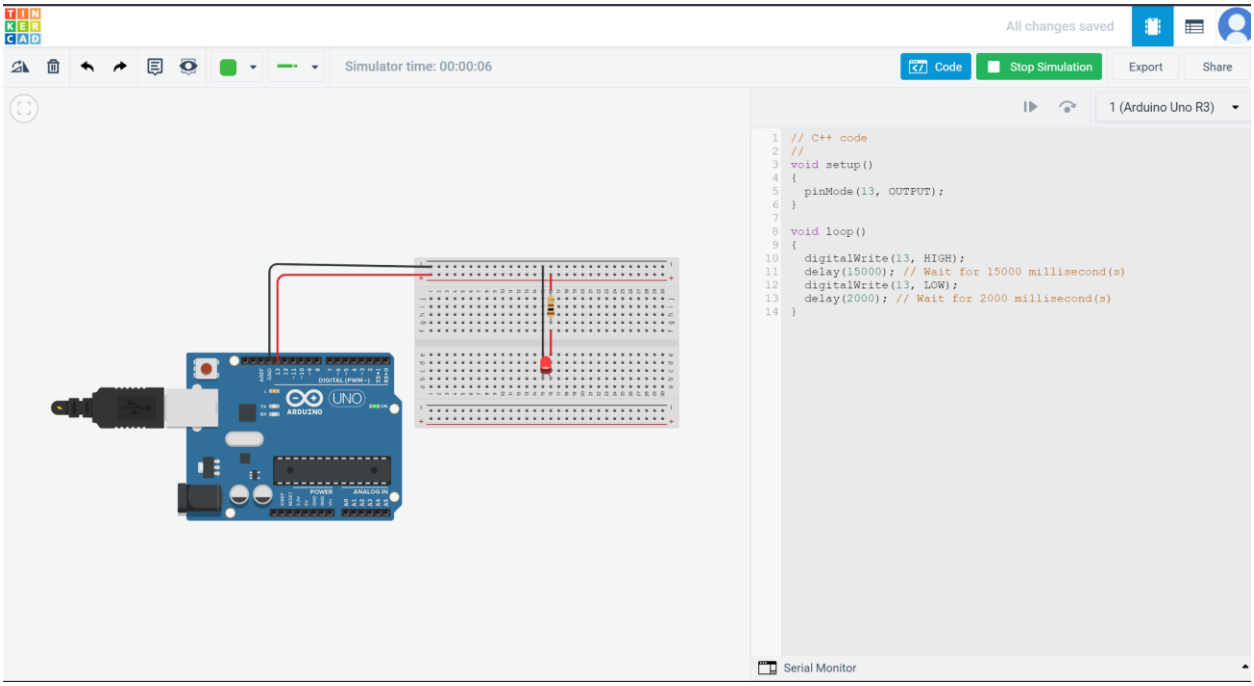


```
void setup()
{
    pinMode(13, OUTPUT);
}

void loop();
{
    digitalWrite(13, HIGH);
    delay(15000);
    digitalWrite(13, LOW);
    delay(2000);
}
```

Tinkercad Snapshots





Reflections (Conclusions):

Here, we saw that we can control the blinking of LED light with the help of Arduino and we can keep them on or off for the desired time.

Exercise 1 (B) – Push Button

Using Tinkercad, connect Pushbutton to turn on the built-in LED on pin 13 i.e., when you press the button LED should glow otherwise it should be in off state.

Hardware Required

- Arduino Board
- momentary button or switch
- 10K ohm resistor
- breadboard
- hook-up wire
- Tinkercad Software tool (https://www.tinkercad.com/)

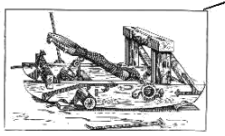
Circuit description

Connect three wires to the Arduino board. The first two, red and black, connect to the two long vertical rows on the side of the breadboard to provide access to the 5 volt supply and ground. The third wire goes from digital pin 10 to one leg of the pushbutton. The same leg of the button connects through a pull-down resistor (here 10 KOhms) to ground. The other leg of the button connects to the 5 volt supply.

When the pushbutton is open (unpressed) there is no connection between the two legs of the pushbutton, so the pin is connected to ground (through the pull-down resistor) and we read a LOW. When the button is closed (pressed), it makes a connection between its two legs, connecting the pin to 5 volts, so that we read a HIGH.

You can also wire this circuit the opposite way, with a pullup resistor keeping the input HIGH, and going LOW when the button is pressed. If so, the behaviour of the sketch will be reversed, with the LED normally on and turning off when you press the button.

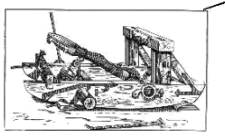
If you disconnect the digital I/O pin from everything, the LED may blink erratically. This is because the input is "floating" - that is, it will randomly return either HIGH or LOW. That's why you need a pull-up or pull-down resistor in the circuit.



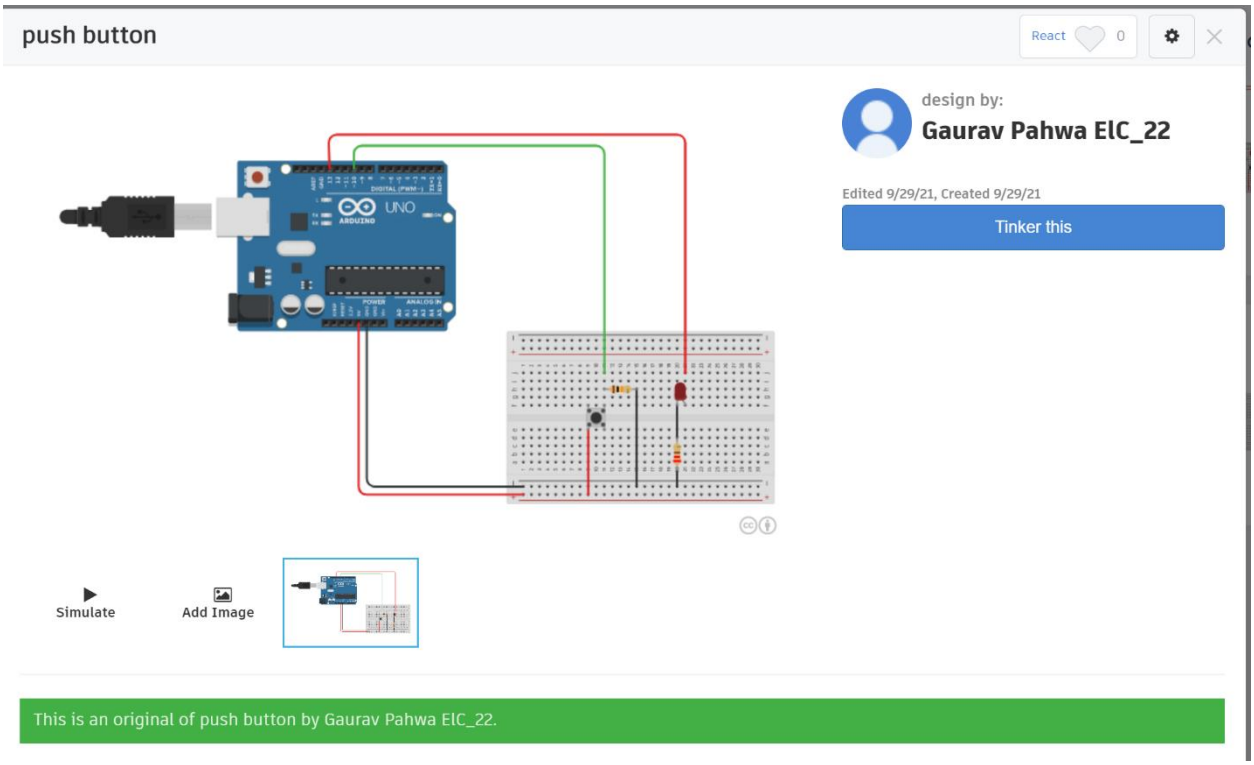
Code

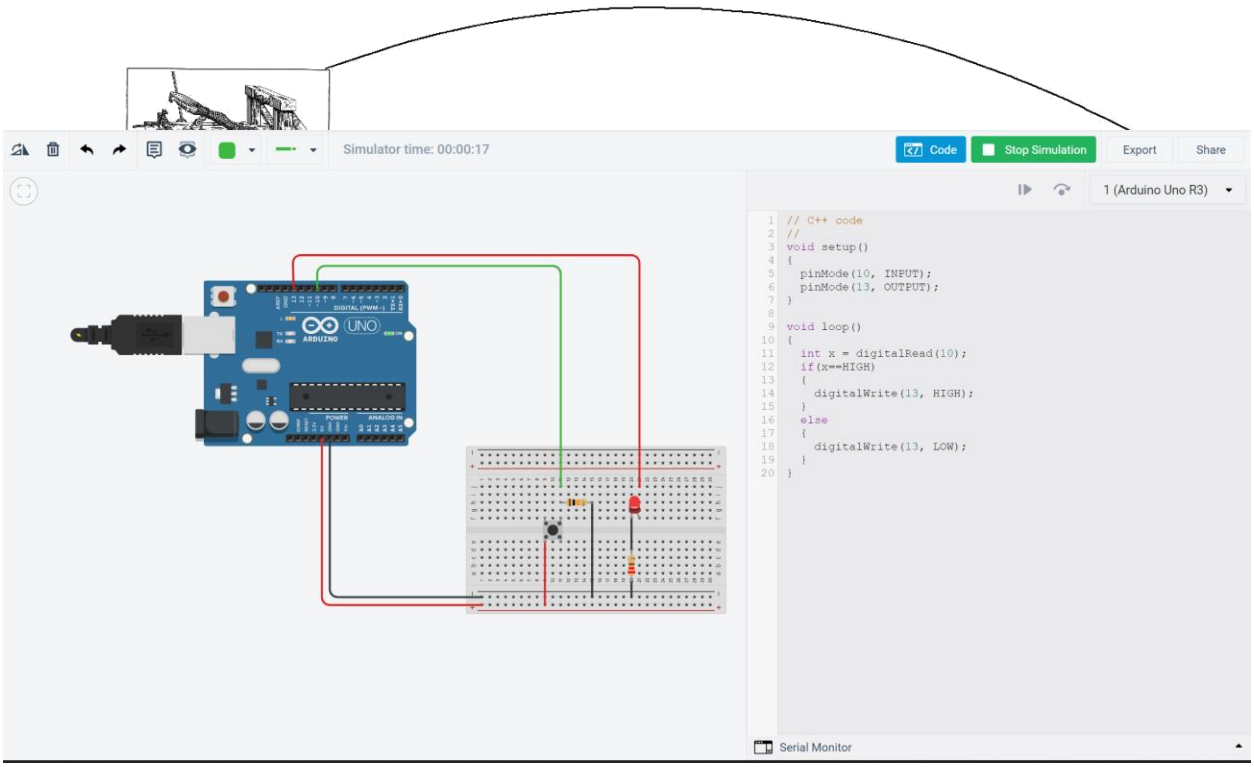
```
void setup()
{
    pinMode (10, INPUT);
    pinMode (13, OUTPUT);
}
void loop()
{
  int x= digitalRead(10);
  if (x == HIGH)
   digitalWrite (13, HIGH);
  else
   digitalWrite (13, LOW);

}
```



Tinkercad Snapshots





Reflections (Conclusions):

Here we observed that we can also use a push button to control a LED light The push button will keep it on till the time it is pressed otherwise it will turn off.

Exercise 1 (C) – Tinkercad

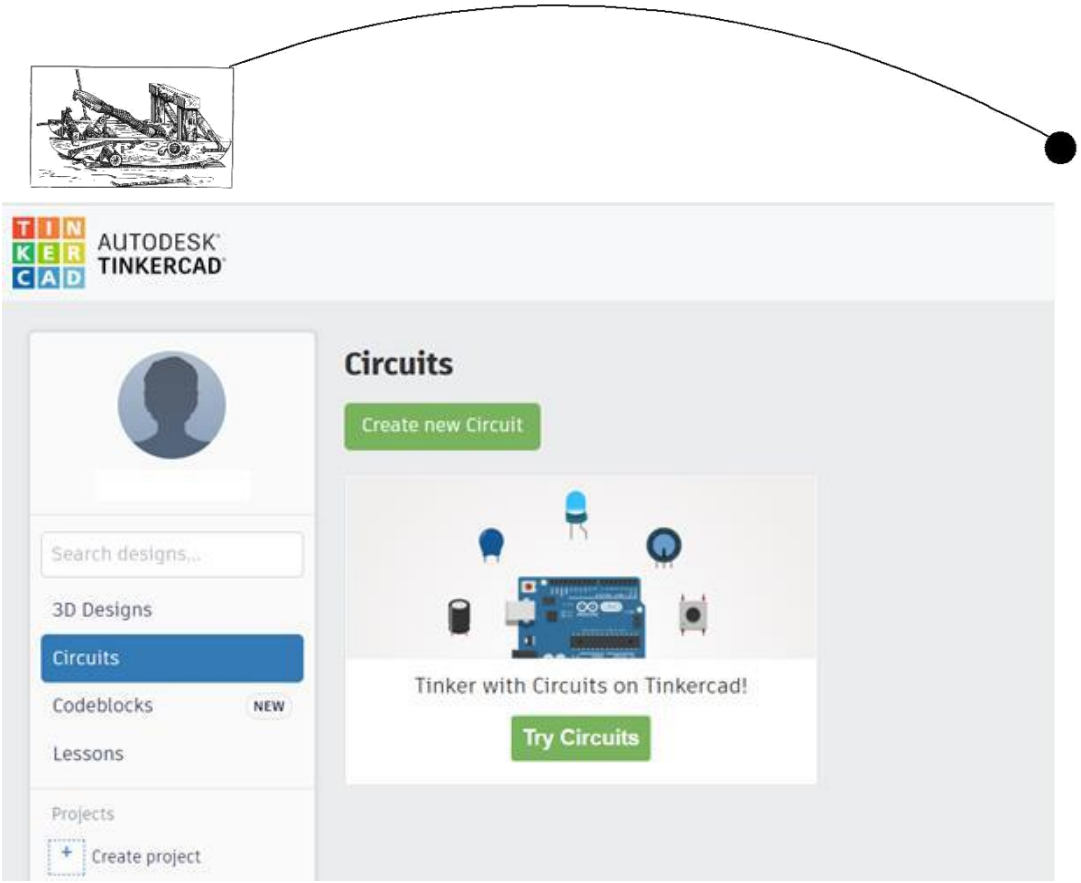
Tinkercad is a free, online collection of software tools for 3D design, electronics, and coding

Software Required

- Tinkercad Software tool (https://www.tinkercad.com/)

Getting Started:

1. Visit https://www.tinkercad.com/
2. Sign in through your google account (Thapar Email ID only)
3. On the Dashboard, select **Circuits** from the drop box and click on **Create new Circuit**



Exercise 2– Study of ICs

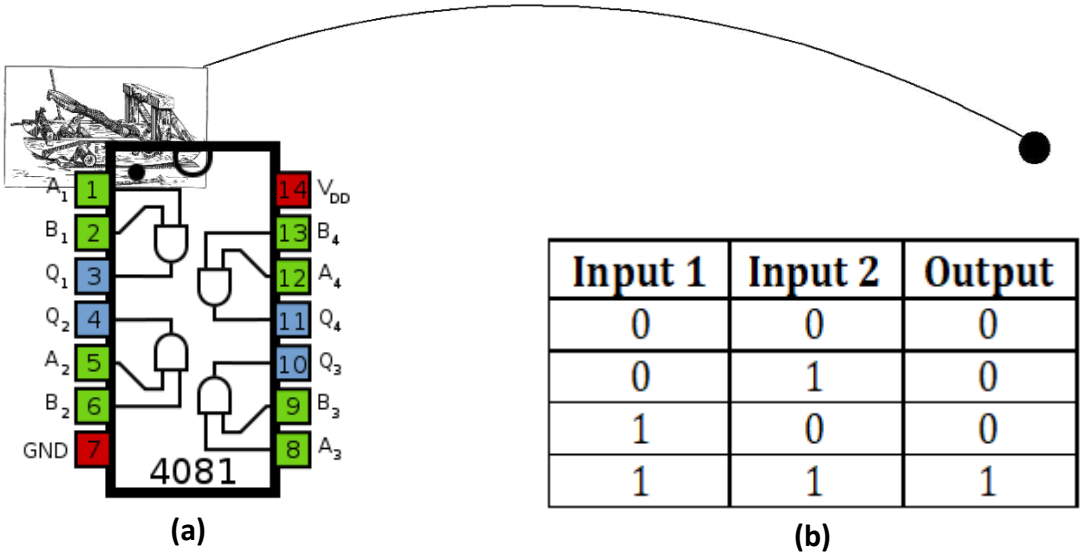
To verify the function tables of CD4027 and CD4081 ICs.

Hardware Required

- Breadboard
- CD 4027 and CD 4081
- Single core connecting wires
- Tinkercad Software tool (https://www.tinkercad.com/)

Theory

The data sheet of CD4027 and CD4081 is given below.



Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Figure 1: Datasheet of CD4081 IC (a) pin diagram (b) functional truth table

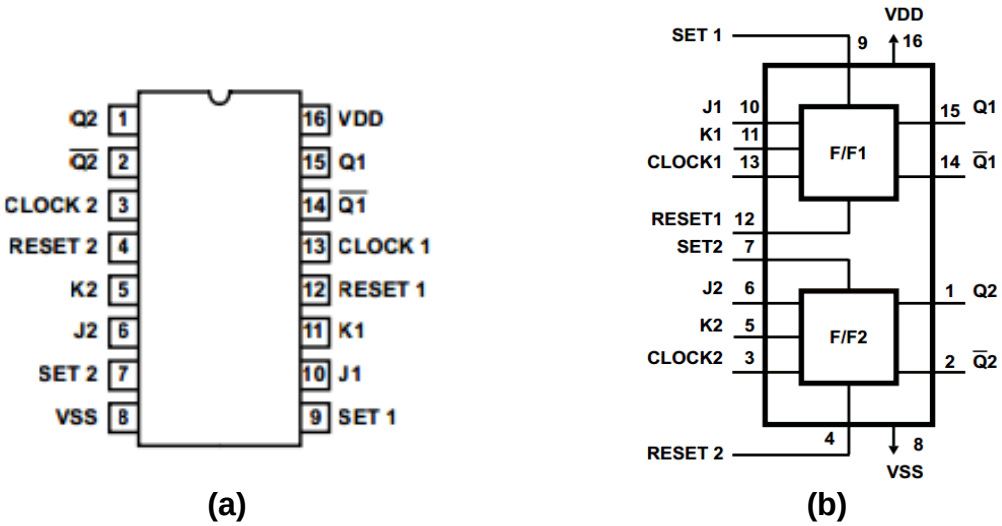
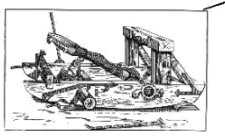


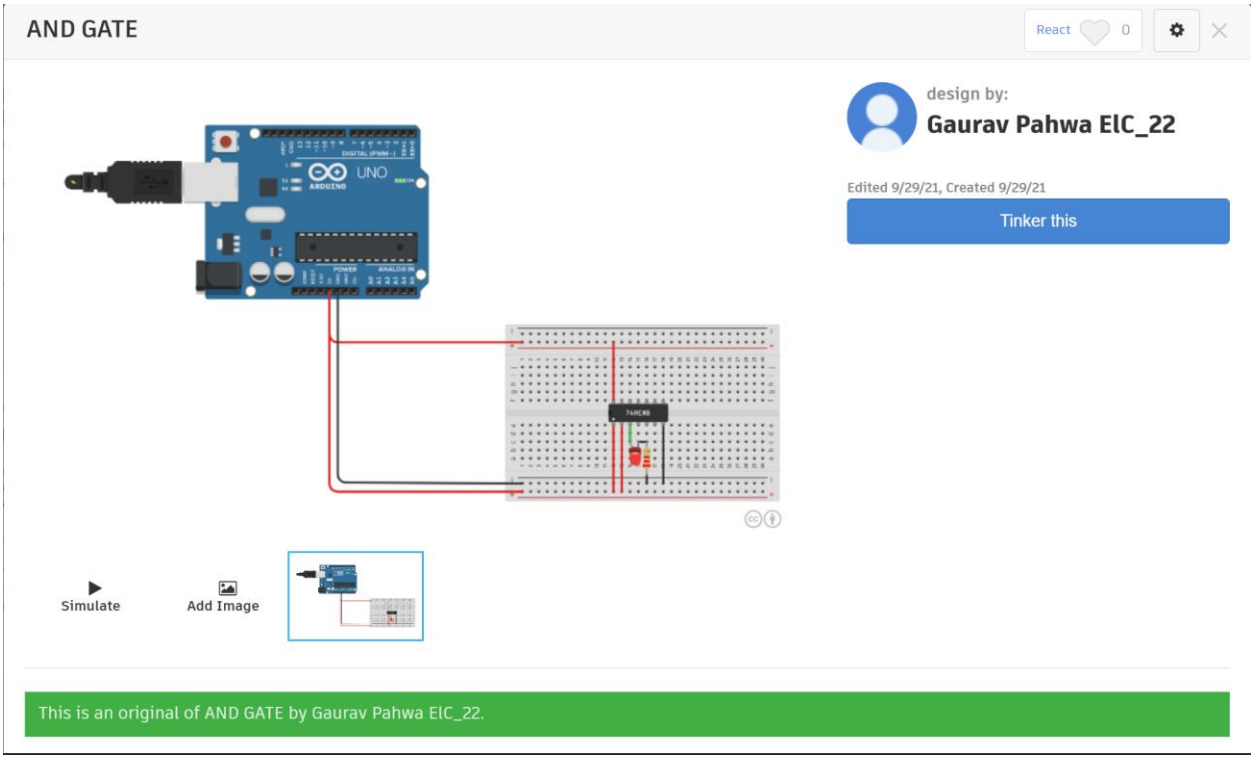
Figure 2: CD4027 IC (a) pin diagram (b) Internal architecture

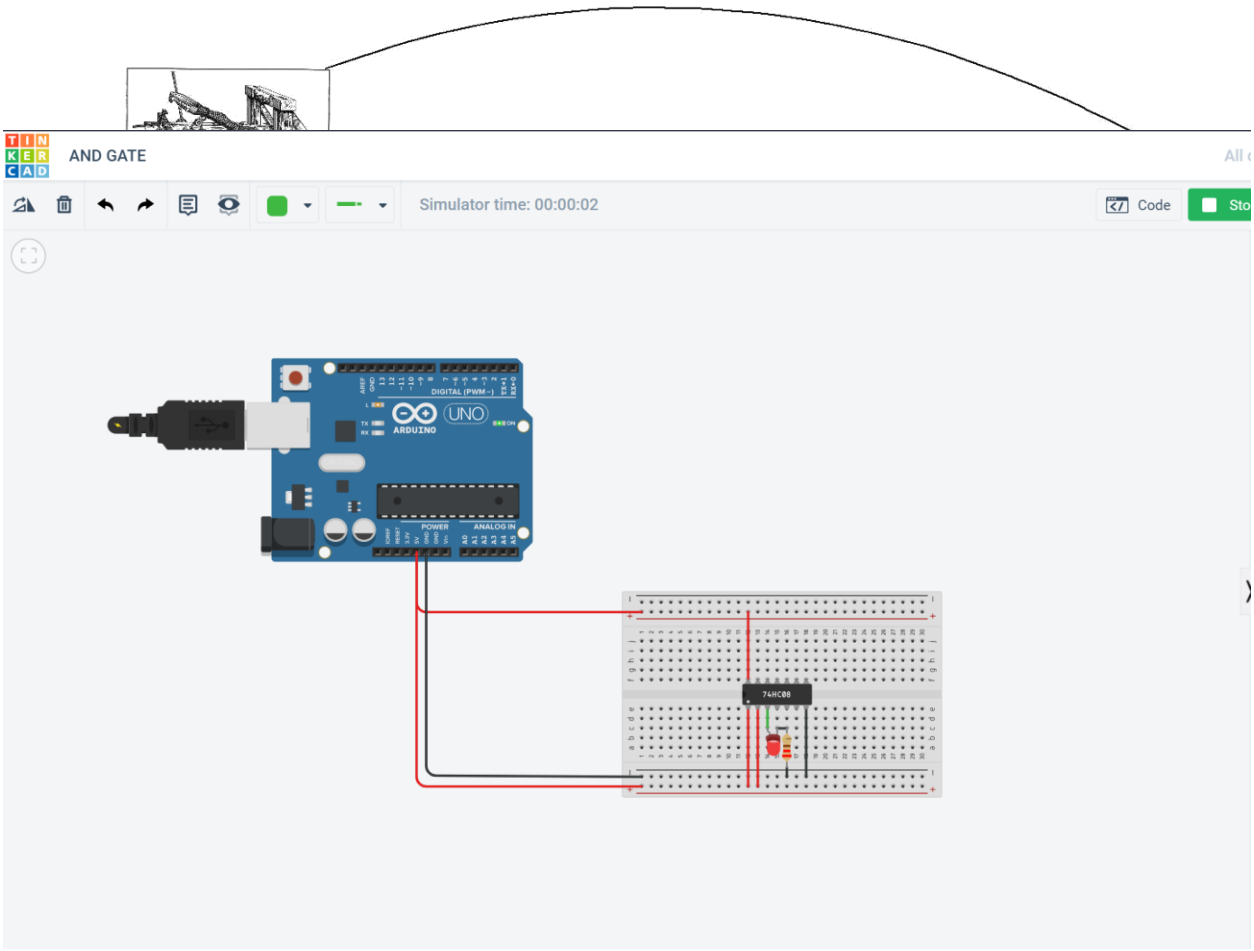
Trigger	Inputs		Output				Inference
			Present State		Next State		
CLK	J	K	Q	Q'	Q	Q'	
⊠	x	x	-		-		Latched
⊓	0	0	0	1	0	1	No Change
⊓			1	0	1	0	
⊓	0	1	0	1	0	1	Reset
⊓			1	0	0	1	
⊓	1	0	0	1	1	0	Set
⊓			1	0	1	0	
⊓	1	1	0	1	1	0	Toggles
⊓			1	0	0	1	

Schematic:

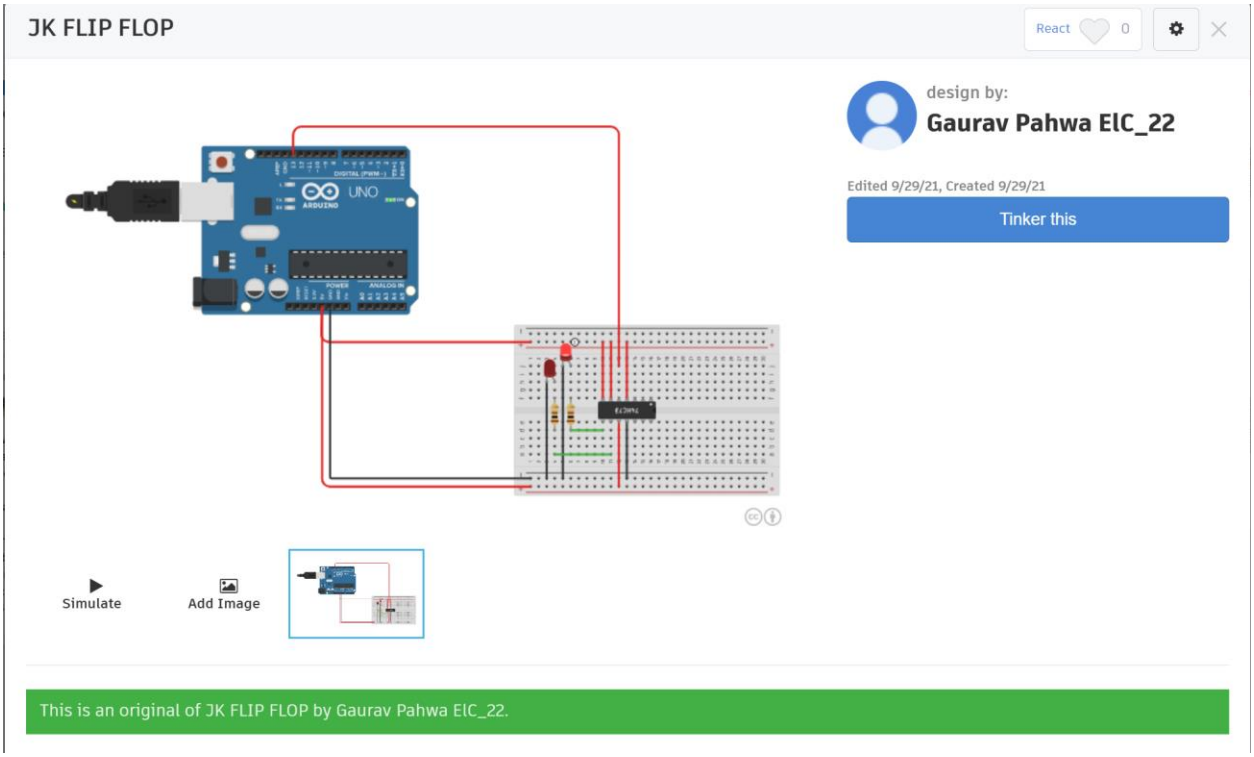


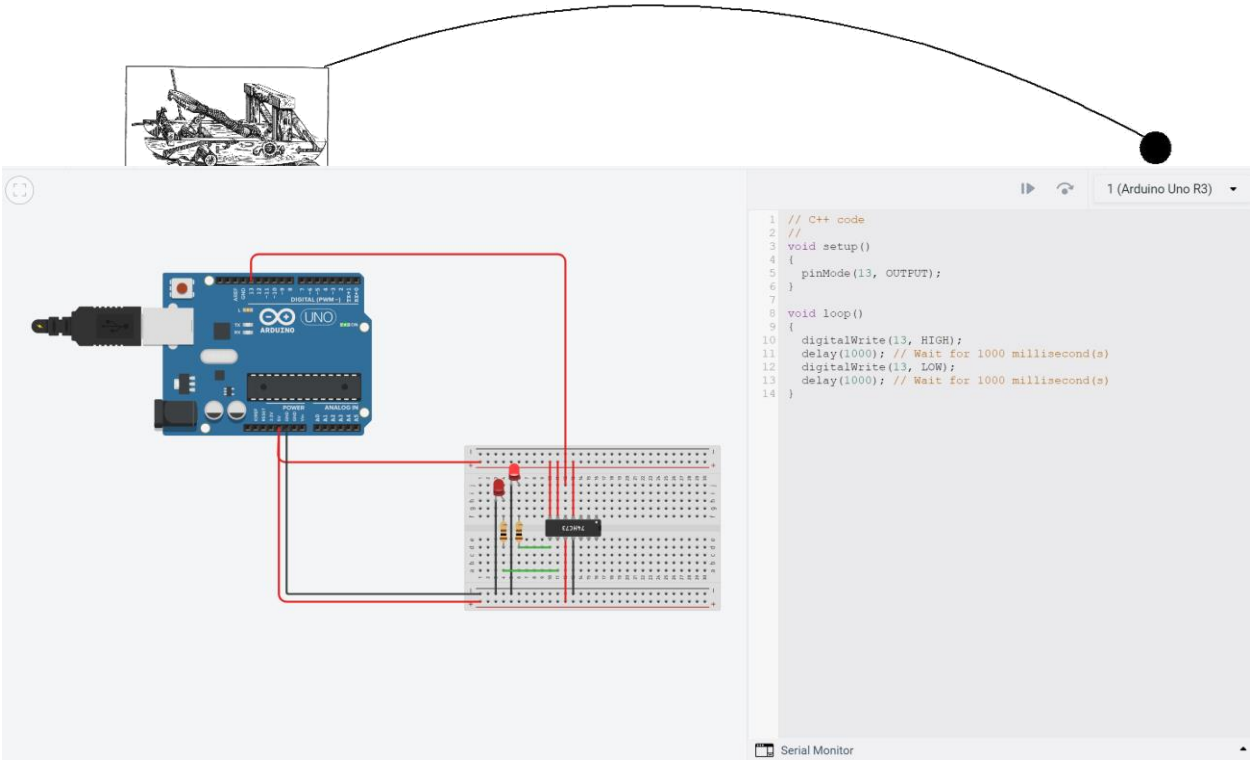
IC 74HC08


AND GATE
React
0
design by:
Gaurav Pahwa ElC_22
Edited 9/29/21, Created 9/29/21
Tinker this
UNO
ARDUINO
74HC08
Simulate
Add Image
This is an original of AND GATE by Gaurav Pahwa ElC_22.


AND GATE
Simulator time: 00:00:02
Code
Sto
ARDUINO
UNO
74HC08

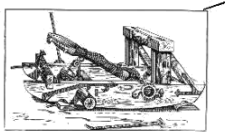
IC 74H73


JK FLIP FLOP
React
0
design by:
Gaurav Pahwa EIC_22
Edited 9/29/21, Created 9/29/21
Tinker this
Simulate
Add Image
This is an original of JK FLIP FLOP by Gaurav Pahwa EIC_22.



Reflections (Conclusions):

Here we verified the truth table of AND Gate and JK flip flop in tinkercad.



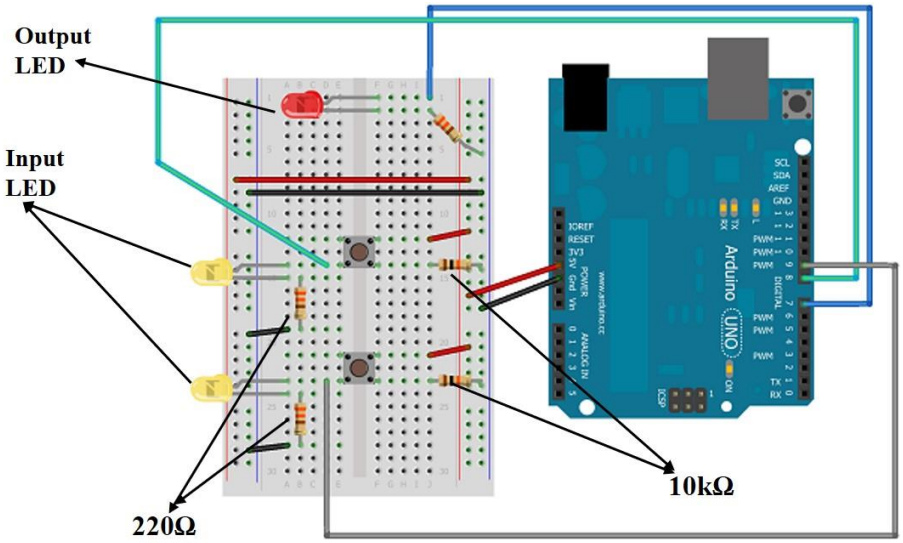
Assignment Tasks:

Note: Each student must attach separate sheets for submitting the below mentioned Assignment Task A-B.

A. Using Tinkercad, Illustrate the pin configuration and verify the truth table of IC 74HC08 and IC 74H7.
Already done above.

B. With the help of Tinkercad, use push buttons to simulate the behaviour of listed logic gates (Without using ICs of logic gates)
   a) Logic gates: AND, NAND, XOR for students with odd numbered Roll Number.
   b) Logic gates: OR, NOR, XNOR for students with even numbered Roll Number.

Hint:

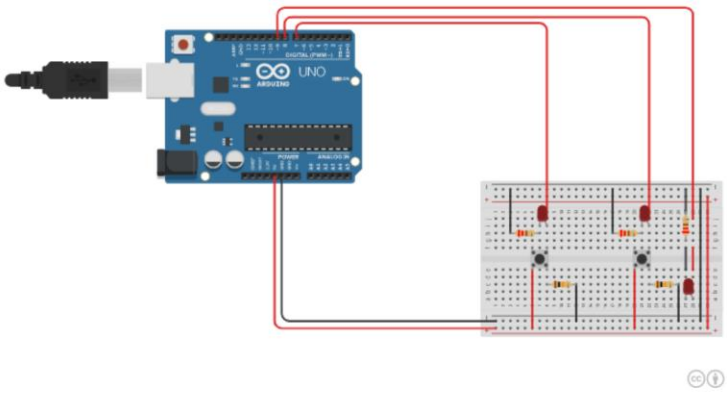


AND

```
Void setup()
{
    pinMode (9, OUTPUT);
    pinMode (8, INPUT);
    pinMode (7, INPUT);
}
void loop()
{
    int x = digitalRead(7);
    int y = digitalRead(8);
    if(x==1 && y==1)
    digitalWrite(9, HIGH);
    else
    digitalWrite(9, LOW);
}
```

Teacher's Signat

And gate with push button



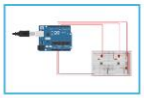
design by:
Gaurav Pahwa EIC_22

Edited 9/29/21, Created 9/29/21

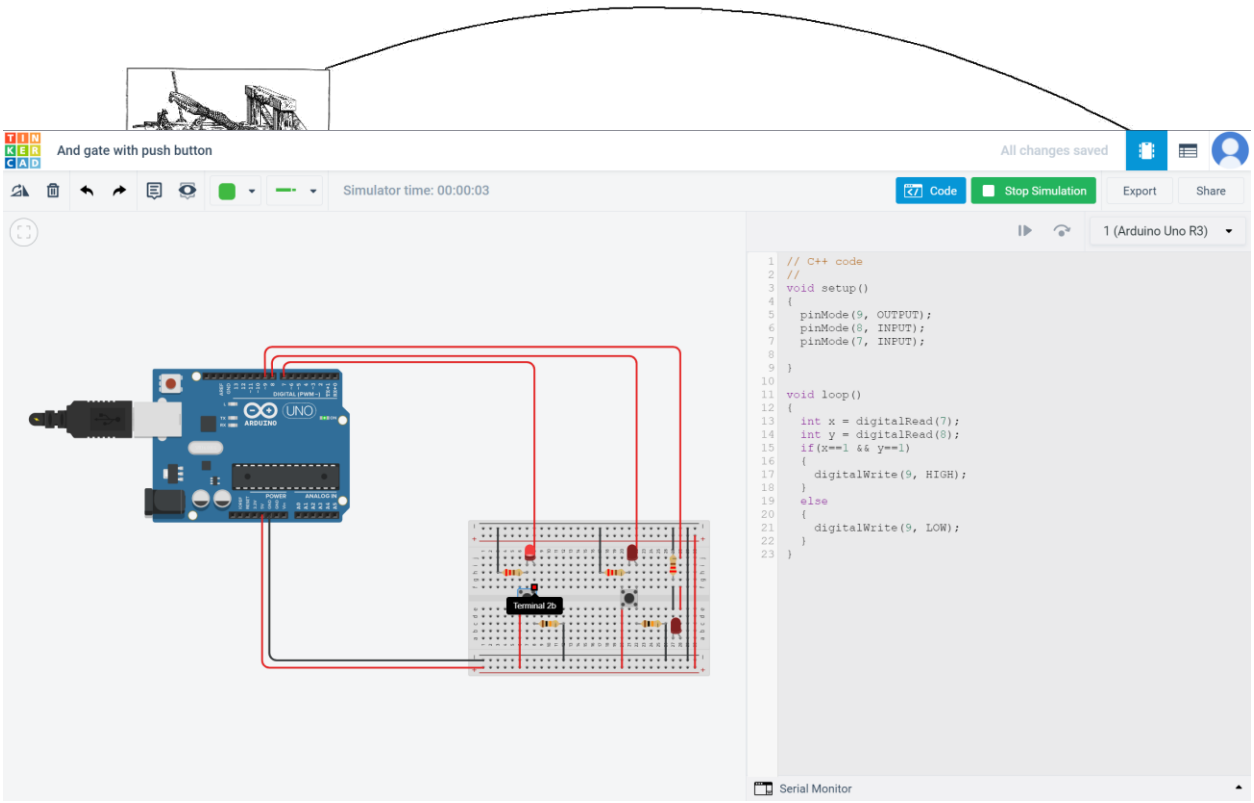
Tinker this

Simulate

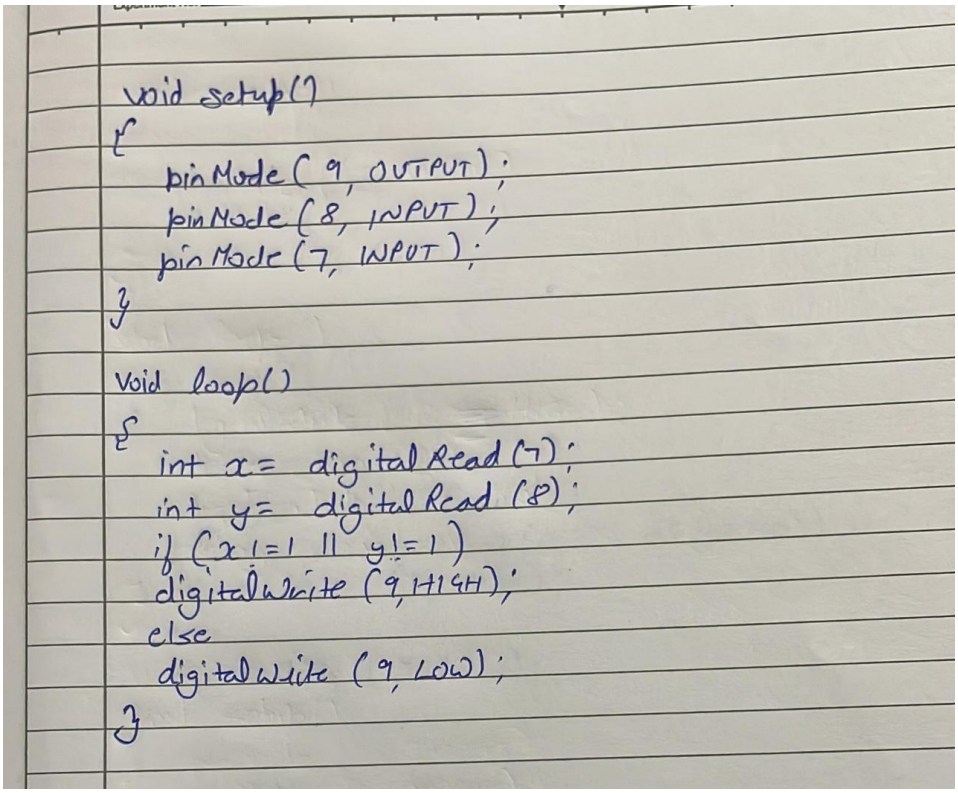
Add Image



This is an original of And gate with push button by Gaurav Pahwa EIC_22.

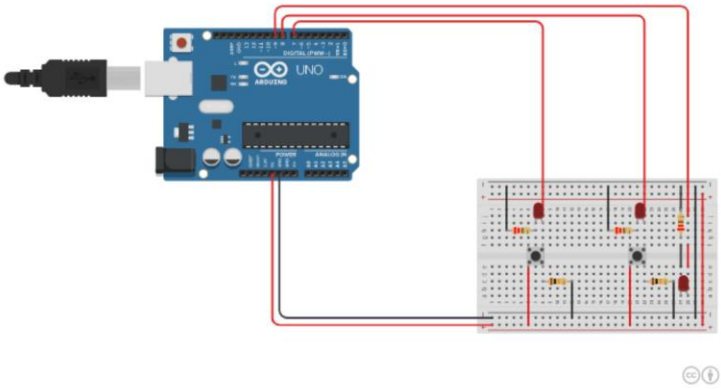

And gate with push button
All changes saved
Simulator time: 00:00:03
Code
Stop Simulation
Export
Share
1 (Arduino Uno R3)
// C++ code
//
void setup()
{
pinMode(9, OUTPUT);
pinMode(8, INPUT);
pinMode(7, INPUT);
}
void loop()
{
int x = digitalRead(7);
int y = digitalRead(8);
if(x==1 && y==1)
{
digitalWrite(9, HIGH);
}
else
{
digitalWrite(9, LOW);
}
}
Terminal 2b
Serial Monitor

NAND


void setup()
{
pinMode(9, OUTPUT);
pinMode(8, INPUT);
pinMode(7, INPUT);
}
Void loop()
{
int x= digitalRead(7);
int y= digitalRead(8);
if(x!=1 || y!=1)
digitalWrite(9,HIGH);
else
digitalWrite(9,LOW);
}

Nand gate with push button

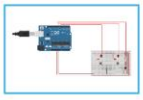
React 0



design by:
Gaurav Pahwa EIC_22

Edited 9/29/21, Created 9/29/21

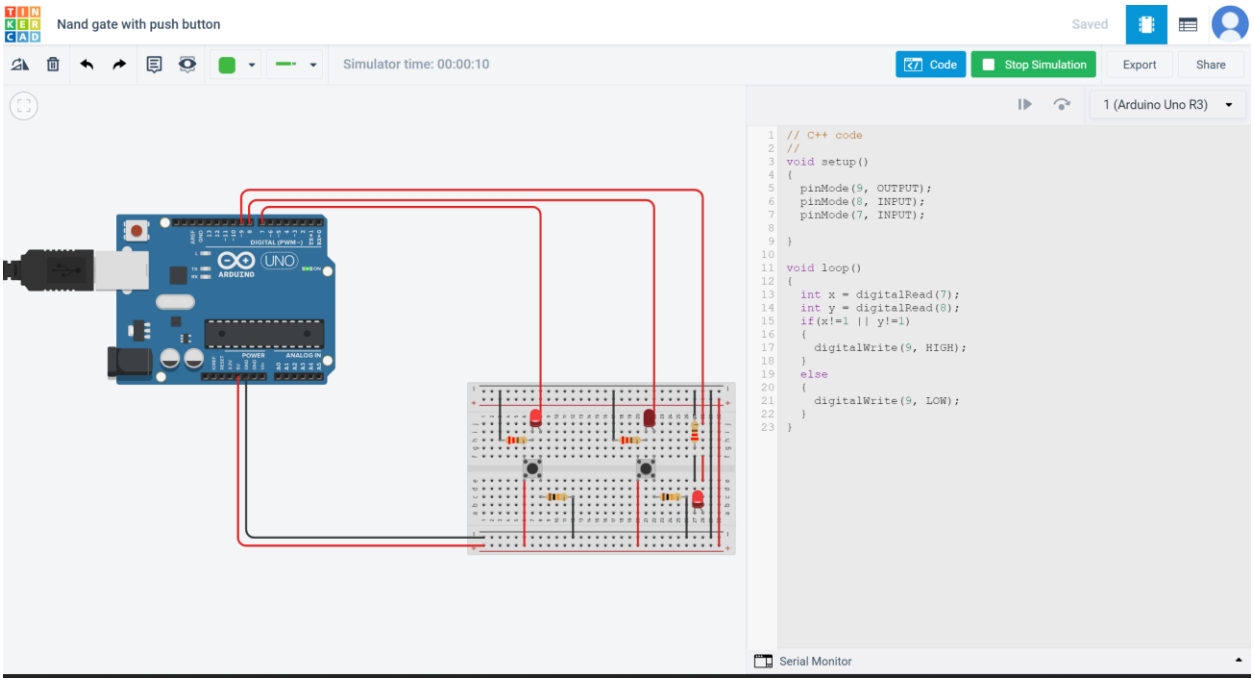
Tinker this

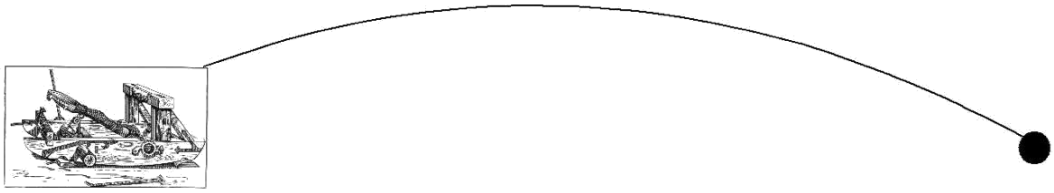


Simulate

Add Image

This is an original of Nand gate with push button by Gaurav Pahwa EIC_22.



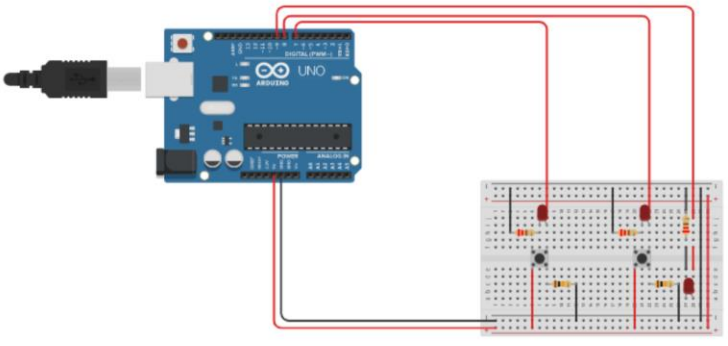


XOR

```
void setup()
{
  pinMode (9, OUTPUT);
  pinMode (8, INPUT);
  pinMode (7, INPUT);
}

void loop()
{
  int x = digitalRead (7);
  int y = digitalRead (8);
  if((x==0 && y==1) || (x==1 && y==0))
  digitalWrite (9, HIGH);
  else
  digitalWrite (9, LOW);
}
```

XOR gate with push button



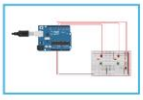
design by:
Gaurav Pahwa EIC_22

Edited 9/29/21, Created 9/29/21

Tinker this

Simulate

Add Image



This is an original of XOR gate with push button by Gaurav Pahwa EIC_22.

XOR gate with push button

Saved

Simulator time: 00:00:13

Code | Stop Simulation | Export | Share

1 (Arduino Uno R3)

```
// C++ code
//
void setup()
{
  pinMode(9, OUTPUT);
  pinMode(8, INPUT);
  pinMode(7, INPUT);

}

void loop()
{
  int x = digitalRead(7);
  int y = digitalRead(8);
  if((x==0 && y==1) || (x==1 && y==0))
  {
    digitalWrite(9, HIGH);
  }
  else
  {
    digitalWrite(9, LOW);
  }
}
```

Serial Monitor