LOVELY PROFESSIONAL UNIVERSITY

REPORT ON

GUI TO PERFORM BITWISE AND, OR, XOR, NOT OPERATION

By
Kunal Kumar,
Subham Patel

Submitted to **Mr. Ishan Kumar**

Department of Computer Science and Engineering

Lovely Professional University

In partial fulfillment of requirement for degree of Bachelor of Technology-CSE
November / 2020

REPORT SUMMARY

Project objectives--

- The objective of this project is to develop Calculator to perform bitwise AND, OR, NOT, XOR operation on two decimal numbers.
- To make the use of GUI in Python.
- Explanation window to show user explanation for each operation selected by user.

ACKNOWLEDGEMENT

A project work is a combination of views, ideas, suggestions and contribution of many people. Thus one of the pleasant part of writing the report is to thank those who have contributed towards its fulfilment. Primarily we would like to thank God for being able to complete the project with success. Then we would like to thank our computer programming teacher Mr. Ishan Kumar, whose valuable guidance has been the once that helped us patch this project and make it full proof success his suggestions and his instruction has served as the major completion towards this project.

Then we would like to thank our parents and friends who have helped us with their valuable suggestions and guidance has been helpful in various phases of the completion of the completion of the project.

Last but not the least we would like to thank my classmates who help me a lot.

Kunal Kumar
Subham Patel

CONTENTS

INTRODUCTION

This project report will introduce how to build bitwise calculator using the GUI. This course management system built using GUI has four major components each of which has different functionality but similar architecture. In the project report we will demonstrate details of using GUI to build this system. Also the technique and process which is showed here can be applied to build the other such projects for various types of calculator such as BMI calculator, simple mathematical calculator, scientific calculator, etc.

SIGNIFICANCE

The bitwise calculator will work on the logical operator AND, OR, XOR and NOT base on the selection made by user and give the output accordingly. This is ultimately going to save time of the user

METHODS USED/TOOLS USED/CONCEPT USED

Graphical User Interface:

A Graphical user interface is an interface through which an user can interact with electronic devices such as computer and other applications ,with the help of mouse there are so many graphical user interfaces tkinter is mostly used as it is fast and easy to create GUI applications This interface uses icons, menus and other visual indicator representations to display information and related user controls, unlike text-based interfaces, where data and commands are in text.

Bitwise Operation:

The purpose of this project was to develop a GUI interface to perform bitwise AND, OR, NOT, XOR operation on two decimal numbers with explanation using python.

NOT operation:

The Bitwise **NOT**, or compliment , is a unary operation that performs logical negation on each bit, forming the ones' complement of the given binary value. Bits that are 0 become 1, and those that are 1 become 0.

Bitwise AND operation:

A **bitwise AND** takes two equal length binary representations and performs the logical AND operation on each pair of the corresponding bits, which is equivalent to multiplying them. Thus, if both bits in the compared position are 1, the bit in the resulting binary representation is 1 (1 × 1 = 1); otherwise, the result is 0 (1 × 0 = 0 and 0 × 0 = 0).

Bitwise OR operation:

A Bitwise **OR** takes two bit patterns of equal length and performs the logical inclusive or operation on each pair of corresponding bits. The result in each position is 0 if both bits are 0, while otherwise the result is 1.

Bitwise XOR operation:

A Bitwise **XOR** takes two bit patterns of equal length and performs the logical exclusive OR operation on each pair of corresponding bits. The result in each position is 1 if only the first bit is 1 *or* only the second bit is 1, but will be 0 if both are 0 or both are 1. In this we perform the comparison of two bits, being 1 if the two bits are different, and 0 if they are the same.

Tools used:

1.Python 3.7.0:

Python is a general purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization.

2.Tkinter:

Tkinter is Python's standard GUI (Graphical User Interface) package. Tkinter is not the only Gui Programming toolkit for Python. It is however the most commonly used one.

PROJECT CONTRIBUTION

We have divided the project into three parts –

Writing the code using python

This section is done by Kunal Kumar. Bitwise operator is use to complete this project.

Collecting Data from Various source

This section is done by Subham Patel. Various online sources are used to complete this project successfully.

Making of final Report

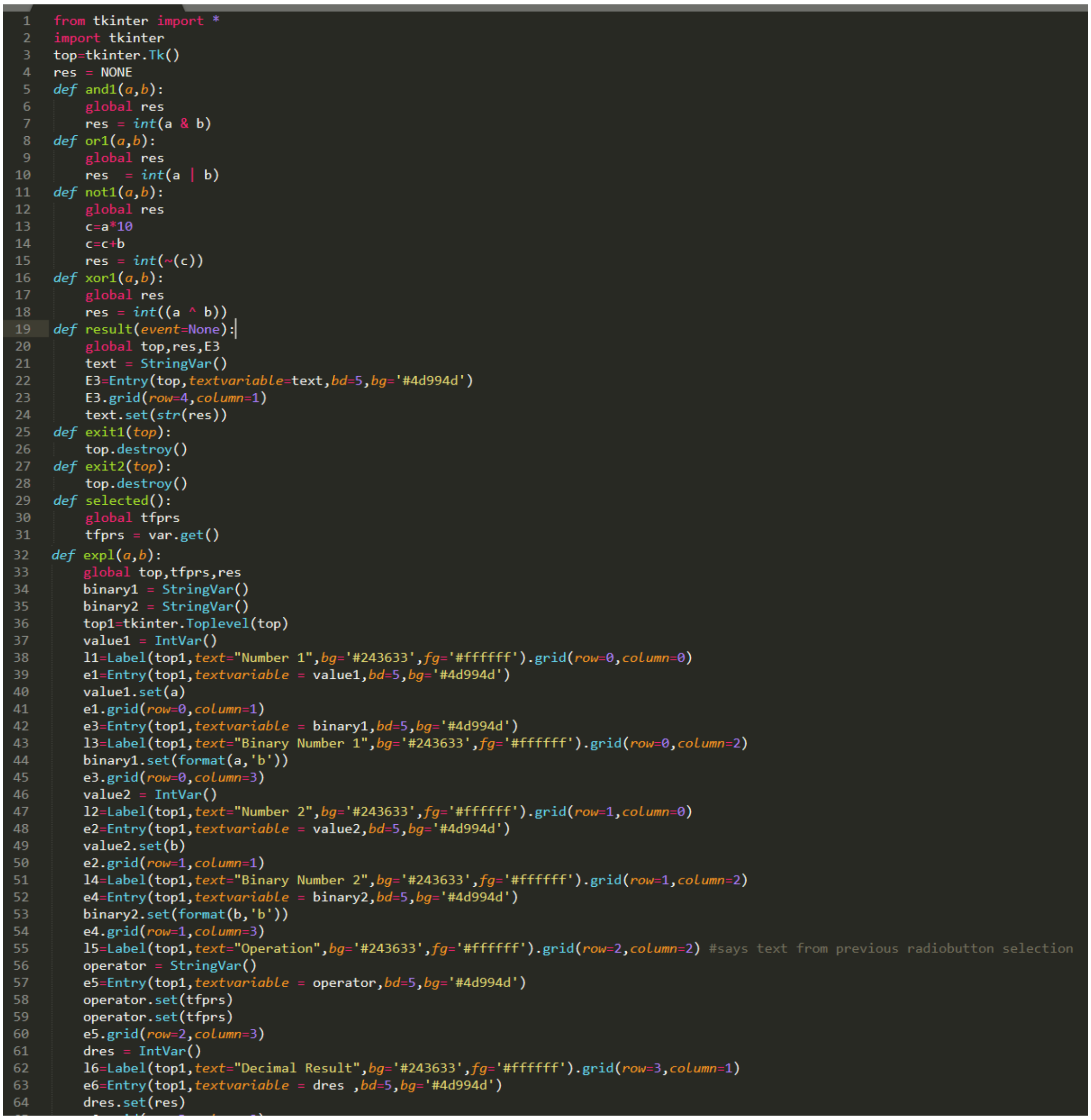
This section is done by Kunal Kumar and Subham Patel. Too many efforts and specifications are done in this step for final touch of Project.

This whole work is divided into the three parts (collection of the required data, coding and final report) in order to complete the project within the limited time.

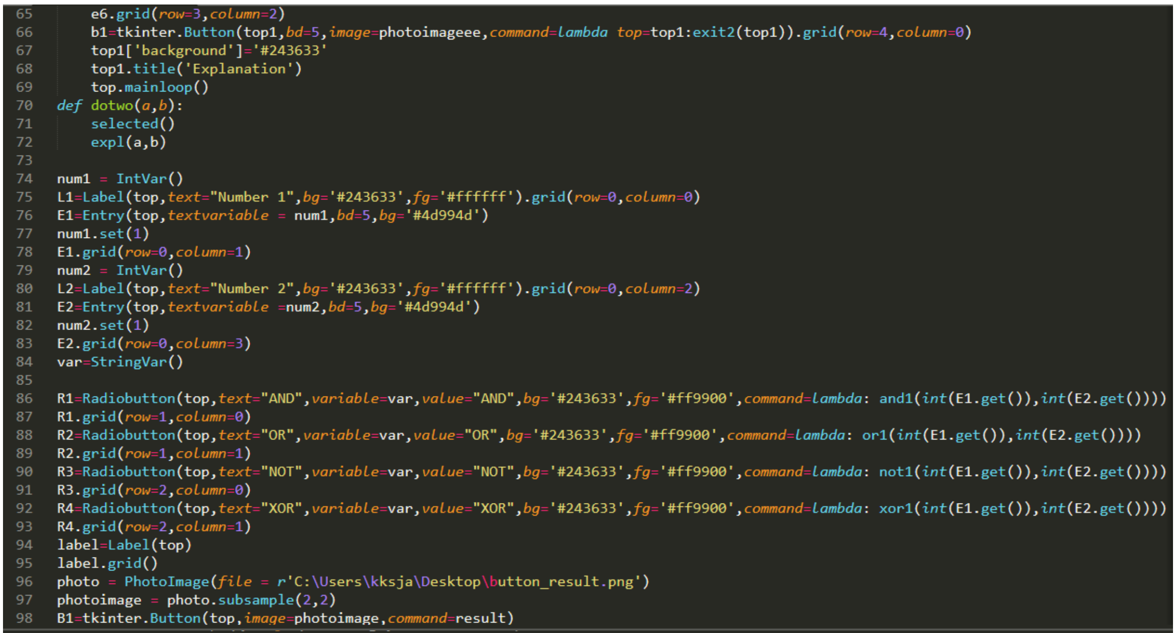
This project is useful in order to calculate basic bitwise result (bitwise AND, bitwise OR, bitwise NOT, bitwise XOR) of the two-given value.

CREATING VIEW

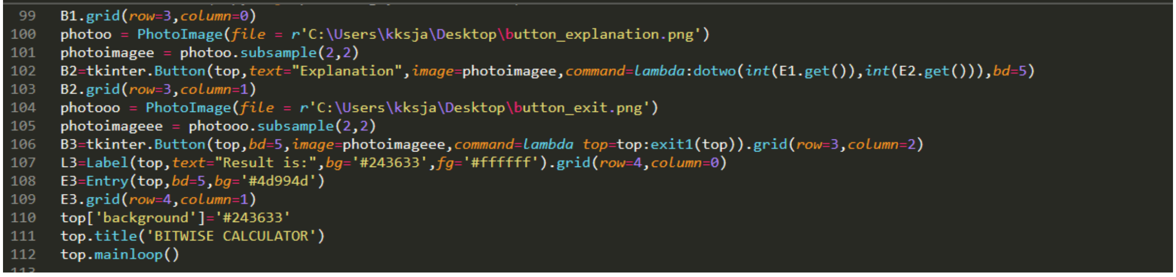
In this section we are basically talking about the code used to build project the code for the following purpose are –
Code :-



```python
from tkinter import *
import tkinter
top=tkinter.Tk()
res = NONE
def and1(a,b):
    global res
    res = int(a & b)
def or1(a,b):
    global res
    res  = int(a | b)
def not1(a,b):
    global res
    c=a*10
    c=c+b
    res = int(~(c))
def xor1(a,b):
    global res
    res = int((a ^ b))
def result(event=None):
    global top,res,E3
    text = StringVar()
    E3=Entry(top,textvariable=text,bd=5,bg='#4d994d')
    E3.grid(row=4,column=1)
    text.set(str(res))
def exit1(top):
    top.destroy()
def exit2(top):
    top.destroy()
def selected():
    global tfprs
    tfprs = var.get()
def expl(a,b):
    global top,tfprs,res
    binary1 = StringVar()
    binary2 = StringVar()
    top1=tkinter.Toplevel(top)
    value1 = IntVar()
    l1=Label(top1,text="Number 1",bg='#243633',fg='#ffffff').grid(row=0,column=0)
    e1=Entry(top1,textvariable = value1,bd=5,bg='#4d994d')
    value1.set(a)
    e1.grid(row=0,column=1)
    e3=Entry(top1,textvariable = binary1,bd=5,bg='#4d994d')
    l3=Label(top1,text="Binary Number 1",bg='#243633',fg='#ffffff').grid(row=0,column=2)
    binary1.set(format(a,'b'))
    e3.grid(row=0,column=3)
    value2 = IntVar()
    l2=Label(top1,text="Number 2",bg='#243633',fg='#ffffff').grid(row=1,column=0)
    e2=Entry(top1,textvariable = value2,bd=5,bg='#4d994d')
    value2.set(b)
    e2.grid(row=1,column=1)
    l4=Label(top1,text="Binary Number 2",bg='#243633',fg='#ffffff').grid(row=1,column=2)
    e4=Entry(top1,textvariable = binary2,bd=5,bg='#4d994d')
    binary2.set(format(b,'b'))
    e4.grid(row=1,column=3)
    l5=Label(top1,text="Operation",bg='#243633',fg='#ffffff').grid(row=2,column=2) #says text from previous radiobutton selection
    operator = StringVar()
    e5=Entry(top1,textvariable = operator,bd=5,bg='#4d994d')
    operator.set(tfprs)
    operator.set(tfprs)
    e5.grid(row=2,column=3)
    dres = IntVar()
    l6=Label(top1,text="Decimal Result",bg='#243633',fg='#ffffff').grid(row=3,column=1)
    e6=Entry(top1,textvariable = dres ,bd=5,bg='#4d994d')
    dres.set(res)
```



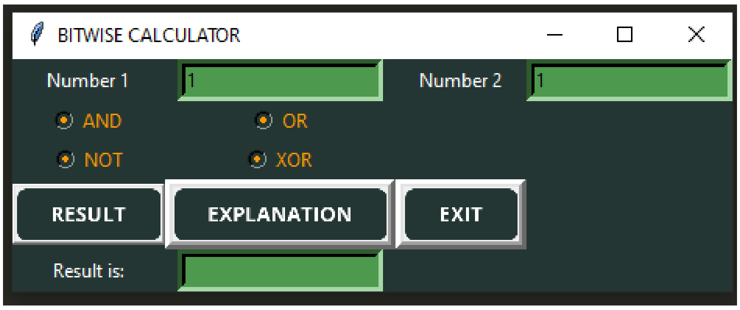
```
    e6.grid(row=3,column=2)
    b1=tkinter.Button(top1,bd=5,image=photoimageee,command=lambda top=top1:exit2(top1)).grid(row=4,column=0)
    top1['background']='#243633'
    top1.title('Explanation')
    top.mainloop()
def dotwo(a,b):
    selected()
    expl(a,b)

num1 = IntVar()
L1=Label(top,text="Number 1",bg='#243633',fg='#ffffff').grid(row=0,column=0)
E1=Entry(top,textvariable = num1,bd=5,bg='#4d994d')
num1.set(1)
E1.grid(row=0,column=1)
num2 = IntVar()
L2=Label(top,text="Number 2",bg='#243633',fg='#ffffff').grid(row=0,column=2)
E2=Entry(top,textvariable =num2,bd=5,bg='#4d994d')
num2.set(1)
E2.grid(row=0,column=3)
var=StringVar()

R1=Radiobutton(top,text="AND",variable=var,value="AND",bg='#243633',fg='#ff9900',command=lambda: and1(int(E1.get()),int(E2.get())))
R1.grid(row=1,column=0)
R2=Radiobutton(top,text="OR",variable=var,value="OR",bg='#243633',fg='#ff9900',command=lambda: or1(int(E1.get()),int(E2.get())))
R2.grid(row=1,column=1)
R3=Radiobutton(top,text="NOT",variable=var,value="NOT",bg='#243633',fg='#ff9900',command=lambda: not1(int(E1.get()),int(E2.get())))
R3.grid(row=2,column=0)
R4=Radiobutton(top,text="XOR",variable=var,value="XOR",bg='#243633',fg='#ff9900',command=lambda: xor1(int(E1.get()),int(E2.get())))
R4.grid(row=2,column=1)
label=Label(top)
label.grid()
photo = PhotoImage(file = r'C:\Users\kksja\Desktop\button_result.png')
photoimage = photo.subsample(2,2)
B1=tkinter.Button(top,image=photoimage,command=result)
```



```
B1.grid(row=3,column=0)
photoo = PhotoImage(file = r'C:\Users\kksja\Desktop\button_explanation.png')
photoimagee = photoo.subsample(2,2)
B2=tkinter.Button(top,text="Explanation",image=photoimagee,command=lambda:dotwo(int(E1.get()),int(E2.get())),bd=5)
B2.grid(row=3,column=1)
photooo = PhotoImage(file = r'C:\Users\kksja\Desktop\button_exit.png')
photoimageee = photooo.subsample(2,2)
B3=tkinter.Button(top,bd=5,image=photoimageee,command=lambda top=top:exit1(top)).grid(row=3,column=2)
L3=Label(top,text="Result is:",bg='#243633',fg='#ffffff').grid(row=4,column=0)
E3=Entry(top,bd=5,bg='#4d994d')
E3.grid(row=4,column=1)
top['background']='#243633'
top.title('BITWISE CALCULATOR')
top.mainloop()
```

RESULT

The result for the above code are as follows –



BITWISE CALCULATOR

Number 1: 1

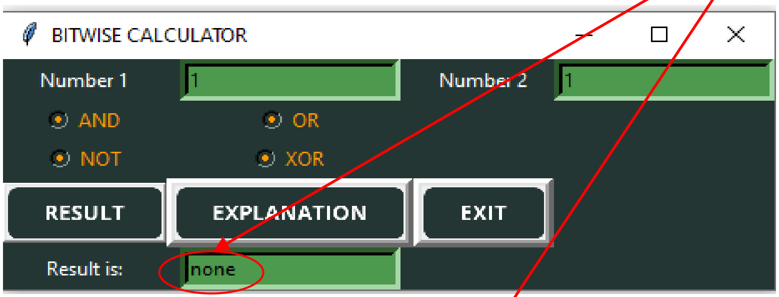
Number 2: 1

● AND ● OR

● NOT ● XOR

Result is:

Here initially ‘1’ is passed in number1 and number 2 to avoid the error and also all the radio button is checked, so if initially user press result button instead of getting the error it will give output as ‘none’ same thing will also appear in explanation tab.



BITWISE CALCULATOR

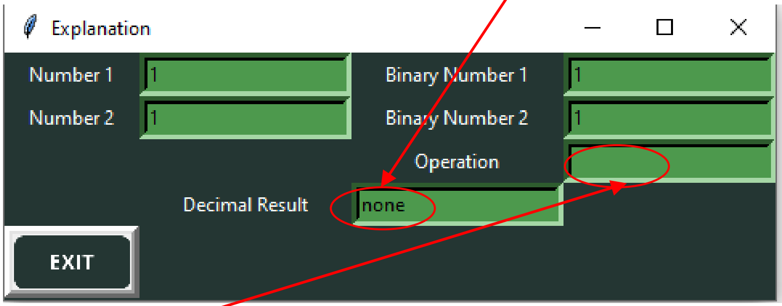
Number 1: 1

Number 2: 1

● AND ● OR

● NOT ● XOR

Result is: none



Explanation

Number 1: 1

Number 2: 1

Binary Number 1: 1

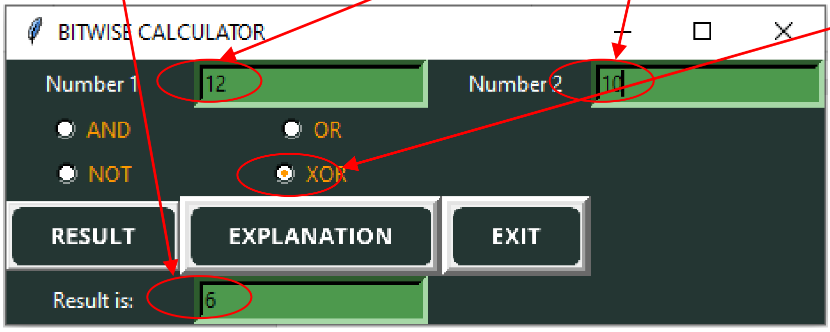
Binary Number 2: 1

Operation:

Decimal Result: none

It is clear from here that all is selected so to avoid the error but in reality not any particular radio button got selected

If user enter any two decimal value, let it be ‘12’ and ‘10’ and he/she selected to perform bitwise XOR then result would be ‘6’



BITWISE CALCULATOR

Number 1: 12

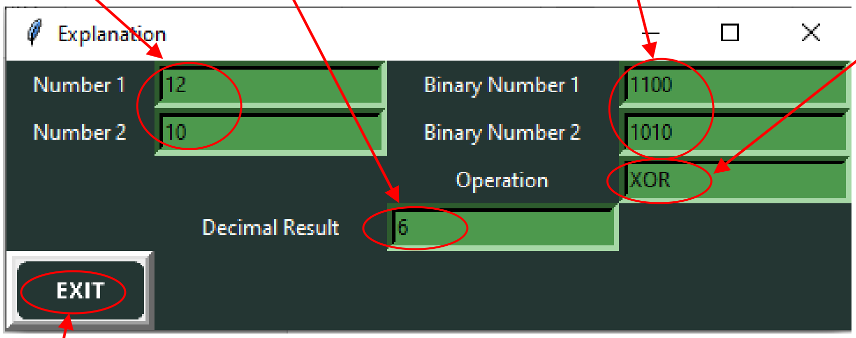
Number 2: 10

○ AND ○ OR

○ NOT ● XOR

Result is: 6

And if he/she click on the explanation button then new window will appear in front of the screen which contains ‘both number’ that is entered by user, it will contain the ‘binary values’ of those number, ‘operation’ that is selected by user and‘decimal result’.


Explanation

Number 1: 12

Number 2: 10

Binary Number 1: 1100

Binary Number 2: 1010

Operation: XOR

Decimal Result: 6

The work of the exit button is to close the current button.

There is exception for bitwise not operation as not operation is perform on single number so we have comine two numbers enter by the user to make it a single number so that we can perform the operation however result is right for the newly formed number but the expalination window won’t work properly for this bitwise operation

CONCLUSION

- Bitwise operation is the common operation that is perform by the computer to perform task ordered by the user and it all happens in the very less unit of the time.
- Human being takes too much to calculate the result of the bitwise operation and even after completion it might be wrong.
- So here we come up with the solution so that human being can calculate bitwise result in lower unit of time and if calculated then can verify their result also by looking to our explanation window they can check the exact step where they did mistakes while performing the same.

LINK OF THE PROJECT

Link of the site where the project is uploaded :-

https://github.com/kunal2020-stack/bitwise-calculator

REFERENCES

1. Geeksforgeeks.com
2. Javatpoint.com
3. Youtube.com/in