**Program 4: Python script for simulating the working of serial ripple carry adder**

#!/usr/bin/python

**'''Program for Graphical simulation of 4 bit serial ripple carry adder**

**Authors: Musaddiq Faraz, Sameer Shaik, Shashi Kiran Reddy, Sujay'''**

from graphics import \*

import time

class Element:

**'''Provides the outline for single full adder block'''**

def \_\_init\_\_(self,point,size):

**#constructor for the Element class**

pt1 = point.clone()

pt2 = point.clone()

pt1.move(size/2,size/2)

pt2.move(-size/2,-size/2)

self.rect = Rectangle(pt2,pt1)

self.rect.setOutline('blue')

self.rect.setFill('red')

pt4 = point.clone()

self.label = Text(pt4,'Full Adder')

pt1 = point.clone()

pt1.move(size/4,-3\*size/4)

pt4 = pt1.clone()

pt4.move(0,-size/8)

self.label1 = Text(pt4,'x')

pt2 = pt1.clone()

pt2.move(0,size/4)

self.line1 = Line(pt1,pt2)

self.line1.setArrow('last')

pt1.move(-size/2,0)

pt2 = pt1.clone()

pt2.move(0,size/4)

pt4 = pt1.clone()

pt4.move(0,-size/8)

self.label2 = Text(pt4,'y')

self.line2 = Line(pt1,pt2)

self.line2.setArrow('last')

pt1 = point.clone()

pt1.move(-size/2,0)

pt2 = pt1.clone()

pt2.move(-size/4,0)

pt4 = pt1.clone()

pt4.move(-size/8,-size/8)

self.label3 = Text(pt4,'co')

self.line3 = Line(pt1,pt2)

self.line3.setArrow('last')

pt1.move(5\*size/4,0)

pt2 = pt1.clone()

pt2.move(-size/4,0)

pt4 = pt1.clone()

pt4.move(-size/8,-size/8)

self.label4 = Text(pt4,'ci')

self.line4 = Line(pt1,pt2)

self.line4.setArrow('last')

pt1 = point.clone()

pt4 = pt1.clone()

pt4.move(0,5\*size/6)

self.label5 = Text(pt4,'sum = ?')

pt1.move(0,size/2)

pt2 = pt1.clone()

pt2.move(0,size/4)

self.line5 = Line(pt1,pt2)

self.line5.setArrow('last')

def set\_text(self,ci,co,x,y,s):

**#changes the texts that has to be changed during animation**

self.label1.setText(x)

self.label2.setText(y)

self.label3.setText(co)

self.label4.setText(ci)

self.label5.setText(s)

def display\_ele(self,win):

**#Displays the single block of adder**

self.rect.draw(win)

self.line1.draw(win)

self.line2.draw(win)

self.line3.draw(win)

self.line4.draw(win)

self.line5.draw(win)

self.label.draw(win)

self.label1.draw(win)

self.label2.draw(win)

self.label3.draw(win)

self.label4.draw(win)

self.label5.draw(win)

def set\_color(self):

**#changes the color of the block, called during animation**

self.rect.setFill('green')

def main(x,y,c,s):

**#the interface for getting data and managing it**

win1 = GraphWin("4-bit DECIMAL SERIAL ADDER",1000,400)

pt = Point(500,50)

label = Text(pt,'Simple 4-bit Decimal Serial Adder')

label.setStyle('bold italic')

label.setSize(20)

label.setTextColor('blue')

label.draw(win1)

elements = []

create(elements)

display(win1,elements)

win1.getMouse()

animate(x,y,c,s,elements,win1)

answer = Text(Point(500,350),x[::-1]+" + "+y[::-1]+" = "+s[::-1])

answer.setSize(20)

answer.setTextColor('blue')

answer.draw(win1)

win1.getMouse()

win1.close()

def create(elements):

**#creates the 4 bit adder**

size = 120

point = Point(1000 - 3\*size/2,200)

i=0

for i in range(0,4):

e = Element(point,size)

elements.append(e)

point.move(- 7\*size/4,0)

def animate(x,y,c,s,elements,win):

**#animates**

line = elements[0].line4.clone()

i=0

line.move(30,0)

line.draw(win)

line.setOutline('green')

time.sleep(.01)

for i in range(0,4):

x1 = line.getP2().getX()

x2 = elements[i].line4.getP2().getX()

while x1!=x2:

x1-=1

line.move(-1,0)

time.sleep(.01)

line.setOutline('black')

elements[i].set\_text(c[i],c[i+1],x[i],y[i],s[i])

elements[i].set\_color()

time.sleep(.15)

line = elements[i].line3.clone()

line.setOutline('green')

line.draw(win)

def display(win,elements):

**#displays the adders**

i = 0

for i in range(0,4):

elements[i].display\_ele(win)