**Assignment no: 01 (On Ubuntu)**

**//Aim: Writing a C/C++ Program to emulate CPU Architecture (Central Bus) Develop register, ALU level GUI to display results.**

#include<iostream>

#include<graphics.h>

using namespace std;

int main()

{

int x,y;

int poly[10];

int gd=DETECT,gm=VGAMAX;

initgraph(&gd,&gm,NULL);

setcolor(WHITE);

poly[0]= 50;

poly[1]= 50;

poly[2]= 600;

poly[3]= 50;

poly[4]= 600;

poly[5]= 600;

poly[6]= 50;

poly[7]= 600;

poly[8]=poly[0];

poly[9]=poly[1];

drawpoly(5,poly);

rectangle(65,100,150,140);

outtextxy(95,115,"ALU");

line(150,120,180,120);

rectangle(180,100,380,140);

outtextxy(200,120,"SEQUENCE CONTROLLER");

rectangle(410,100,550,160);

outtextxy(422,107,"GENERAL");

outtextxy(422,122,"PURPOSE");

outtextxy(422,136,"REGISTER");

rectangle(410,220,550,250);

outtextxy(422,228,"CACHE MEMORY");

rectangle(210,180,350,220);

outtextxy(240,190,"PROGRAM");

outtextxy(240,205,"COUNTER");

line(280,140,280,180);

rectangle(65,280,200,250);

outtextxy(85,255,"INSTRUCTION");

outtextxy(85,268,"REGISTER");

rectangle(65,340,200,310);

outtextxy(85,315,"INSTRUCTION");

outtextxy(85,328," DECODER");

line(50,420,600,420);

line(50,450,600,450);

outtextxy(220,430,"ADDRESS& DATA BUS");

outtextxy(190,460,"BASIC ARCHITECTURE OF CPU");

line(105,140,105,250);

line(260,380,260,420);

line(320,380,320,420);

delay(10000);

closegraph();

return 0;

}

.........................................................................................................................................

**Assignment no: 01 (OnWindows)**

**//Aim: Writing a C/C++ Program to emulate CPU Architecture (Central Bus) Develop register, ALU level GUI to display results.**

/\* CLASS :S E COMP

COMPUTER ARCHIETECTURE

\*/

#include<string.h>

#include<iostream.h>

#include<conio.h>

#include<dos.h>

#include<graphics.h>

#include<stdlib.h>

#include<math.h>

#include<ctype.h>

#include<stdio.h>

class computer

{

int gm,gd;

int c1;

int j;

char n1[5],n2[5];

char \*d1;

int a1,b1;

char a[5];

      int poly[20];

public:

computer()

{  j=0;

}

void init();

void monitor();

void keyboard();

void mouse();

void biu();

void alu();

void flags();

void registers();

void segments();

void buses();

void sound();

void exp1();

void display();

};

void computer::init()

{

 gd=DETECT;

 initgraph(&gd,&gm,"..\\bgi");

 }

 void computer::monitor()

 {

  rectangle(60,60,240,240);   //MONITOR

rectangle(65,65,235,235);

outtextxy(70,70,"Enter Expn:");

outtextxy(70,110,"Output is= ");

for(int i=1;i<5;i++)

{

circle(200+5\*i,237.5,2.5);

}

setfillstyle(1,BLUE);

fillellipse(237.5,237.5,5,5);

moveto(130,200);

line(150,240,170,260);  //MONITOR STAND

line(150,240,130,260);

line(130,260,170,260);

outtextxy(70,90,a);

}

void computer::keyboard()

{

  rectangle(65,300,190,340);  //KEYBOARD

rectangle(67,302,150,338);

rectangle(152,302,188,338);

}

void computer::mouse()

{

setfillstyle(1,BLUE);

 fillellipse(228,320,10,20);    //MOUSE

 arc(175,301,0,90,60);

}

void computer::biu()

  {

   rectangle(300,20,600,470);   //CPU

poly[0]=340;

poly[1]= 35;//1

poly[2]= 335;

poly[3]=100;  //2

poly[4]= 355;

poly[5]= 100  ;  //3

poly[6]= 360;

poly[7]= 70;   //4

poly[8]=  390;

poly[9]=  70;     //5

poly[10]=   395;

poly[11]= 100;  //6

poly[12]=415;

poly[13]=100;//7

poly[14]=410;

poly[15]=   35;//8

poly[16]=340;

poly[17]=   35;   //9

setfillstyle(1,CYAN);

fillpoly(9,poly);

char ope[2];

ope[0]=a[2];

ope[1]='\0';

outtextxy(340,45,"OPERATION");

outtextxy(370,60,ope);

}

void computer::segments()

{

 rectangle(320,130,420,230);  //segments

for(int k=1;k<5;k++)

{   int a=320;

      int b=130+20\*k;

      int c= 420;

      int d=130+20\*k;

line(a,b,c,d);

}

outtextxy(330,133,"ES=0000h");

outtextxy(330,153,"DS=0000h");

outtextxy(330,173,"CS:0000h");

outtextxy(330,193,"IP:0001h");

outtextxy(330,213,"DI:ALU");

rectangle(480,28,560,50);   //memory interface

outtextxy(494,38,"Memory ");

rectangle(480,80,540,180);  // instn queue

for(k=1;k<5;k++)

{   int a=480;

      int b=80+20\*k;

      int c= 540;

      int d=80+20\*k;

line(a,b,c,d);

}

outtextxy(510,167,"1");

outtextxy(546,100,"INSTN");

outtextxy(546,115,"QUEQE");

rectangle(470,220,550,240); //control sys

outtextxy(480,222,"PC:01");

}

void computer::registers()

{

 rectangle(320,310,420,450); //registers

for(int k=1;k<8;k++)

{   int a=320;

      int b=310+20\*k;

      int c= 420;

      int d=310+20\*k;

line(a,b,c,d);

outtextxy(304,312,"AX");

outtextxy(304,332,"BX");

outtextxy(304,352,"CX");

outtextxy(304,372,"DX");

outtextxy(330,394,"SP=0001h");

outtextxy(330,414,"BP=0000h");

outtextxy(367,434,"FLAG");

}

}

void computer::alu()

{

outtextxy(490,320,"ALU");

 int poly2[20];     //alu

poly2[0]=450;

poly2[1]= 330;//1

poly2[2]= 470;

poly2[3]=390;  //2

poly2[4]= 530;

poly2[5]= 390  ;  //3

poly2[6]= 550;

poly2[7]= 330;   //4

poly2[8]=  530;

poly2[9]=  330;     //5

poly2[10]=   520;

poly2[11]= 350;  //6

poly2[12]=480;

poly2[13]=350;//7

poly2[14]=470;

poly2[15]= 330  ;//8

poly2[16]=450;

poly2[17]=330;   //9

drawpoly(9,poly2);

}

void computer::flags()

{

 rectangle(440,410,540,450);  //flags

line(440,430,540,430);

for(int k=1;k<8;k++)

{   int a=440+12.5\*k;

      int b=410;

      int c= 440+12.5\*k;

      int d=450;

line(a,b,c,d);

}

outtextxy(443,420,"-");

outtextxy(456,420,"-");

outtextxy(469,420,"-");

outtextxy(482,420,"-");

outtextxy(494,420,"O");

outtextxy(507,420,"D");

outtextxy(520,420,"I");

outtextxy(533,420,"T");

outtextxy(443,437,"S");

outtextxy(456,437,"Z");

outtextxy(469,437,"-");

outtextxy(482,437,"A");

outtextxy(494,437,"-");

outtextxy(507,437,"P");

outtextxy(520,437,"-");

outtextxy(533,437,"C");

char l=a[2];

switch(l)

{

 case '+':      setfillstyle(1,RED);

            floodfill(538,448,15); //carry

            setfillstyle(1,RED);

            floodfill(512,442,15);  //parity

            setfillstyle(1,RED);

            floodfill(489,443,15);  //auxill

            setfillstyle(1,RED);

            floodfill(492,424,15); //over

            break;

  case '-':     setfillstyle(1,BLUE);

            floodfill(441,441,15);    //sign

            setfillstyle(1,BLUE);

            floodfill(454,441,15) ;   //zero

            setfillstyle(1,BLUE);

            floodfill(538,448,15);

            setfillstyle(1,BLUE);

            floodfill(512,442,15);

            break;

  case '\*':     setfillstyle(1,GREEN);

            floodfill(454,441,15) ;

            setfillstyle(1,GREEN);

            floodfill(489,443,15);

            setfillstyle(1,GREEN);

            floodfill(538,448,15);

            setfillstyle(1,GREEN);

            floodfill(512,442,15);

            break;

  case '/':      setfillstyle(1,BROWN);

            floodfill(454,441,15) ;

            setfillstyle(1,BROWN);

            floodfill(489,443,15);

            setfillstyle(1,BROWN);

            floodfill(538,448,15);

            setfillstyle(1,BROWN);

            floodfill(512,442,15);

            break;

            }

}

void computer::buses()

{

setcolor(10);

line(420,40,470,40);

line(470,40,460,37);

line(470,40,460,43);

delay(500);

line(510,55,510,75);

line(510,75,505,72);

line(510,75,515,72);

delay(500);

line(510,185,510,210);

line(510,210,505,206);

line(510,210,515,206);

delay(500);

setcolor(15);

 int poly1[43];

poly1[0]=340;

poly1[1]=260;//1

poly1[2]=340;

poly1[3]=280;  //2

poly1[4]=455;

poly1[5]=280;   //3

poly1[6]=455;

poly1[7]=290; //4

poly1[8]=450;

poly1[9]= 290;//5

poly1[10]= 460;

poly1[11]=310; //6

poly1[12]= 470;

poly1[13]=290;//7

poly1[14]= 465;

poly1[15]= 290;//8

poly1[16]=465;

poly1[17]=280;//9

poly1[18]=  535;

poly1[19]=  280;//10

poly1[20]=535;

poly1[21]= 290;   //11

poly1[22]= 530;

poly1[23]=  290;    //12

poly1[24]=  540;

poly1[25]=   310;       //13

poly1[26]= 550;

poly1[27]= 290;         //14

poly1[28]=   545;

poly1[29]=   290;//15

poly1[30]=  545;

poly1[31]= 280;//16

poly1[32]= 570;

poly1[33]= 280;//17

poly1[34]=  570;

poly1[35]= 440;//18

poly1[36]= 590 ;

poly1[37]= 440;//19

poly1[38]=  590;

poly1[39]= 260;//20

poly1[40]=  340;

poly1[41]= 260;//21

 setfillstyle(8,RED);

 fillpoly(21,poly1);

delay(1000);

setcolor(10);

line(540,360,570,360);

line(570,360,560,350);

line(570,360,560,370);

delay(1000);

line(490,390,490,410);

line(490,410,487,405);

line(490,410,493,405);

delay(1000);

line(540,430,570,430);

line(540,430,543,425);

line(540,430,543,435);

line(570,430,568,425);

line(570,430,568,435);

setcolor(15);

}

void computer::exp1()

{

cout<<"\n Enter the expression of your choice:";

for(int i=0;i<3;i++)

{

cin>>a[i];

}

cleardevice();

a[3]='\0';

 i=0;

while(isdigit(a[i]))

{ n1[i]=a[i];

i++;

}

n1[i]='\0';

i=2;

while(isdigit(a[i]))

{ n2[j]=a[i];

i++;

j++;

}

n2[j]='\0';

a1= atoi(n1);

b1=atoi(n2);

if(a[1]=='+')

c1=a1+b1;

else if(a[1]=='-')

c1=a1-b1;

else if(a[1]=='\*')

c1=a1\*b1;

else if(a[1]=='/')

c1=a1/b1;

else if(a[1]=='%')

c1=a1%b1;

else

cout<<"\n Enter correctly:";

}

void computer::sound()

{

rectangle(260,100,280,150);//sound

setfillstyle(1,BLUE);

fillellipse(270,125,10,10);

setfillstyle(1,BROWN);

fillellipse(270,125,5,5);

rectangle(20,100,40,150); //sound

setfillstyle(1,BLUE);

fillellipse(30,125,10,10);

setfillstyle(1,BROWN);

fillellipse(30,125,5,5);

}

void computer:: display()

{

outtextxy(324,313,n1);

outtextxy(324,333,n2);

char n3[2];

n3[0]='=';

n3[1]='\0';

char n4[50];

itoa(a1,n1,2);

itoa(b1,n2,2);

outtextxy(340,313,strcat("=",n1));

outtextxy(340,333,n3);

outtextxy(346,333,n2);

 itoa(c1,n4,2);

 outtextxy(322,373,n4);

 outtextxy(480,365,n4);

 flushall();

 char v[20];

 itoa(c1,v,10);

 outtextxy(72,130,v);

}

void main()

{

computer c;

clrscr();

c.init();

delay(500);

c.exp1();

delay(500);

c.monitor();

delay(500);

c.keyboard();

delay(500);

c.mouse();

delay(500);

c.sound();

delay(500);

c.biu();

delay(500);

c.segments();

delay(500);

c.registers();

delay(500);

c.display();

delay(500);

c.alu();

delay(500);

c.flags();

delay(500);

c.buses();

getch();

  closegraph();

restorecrtmode();

}