BAPATLA ENGINEERING COLLEGE

OOPS AND PSPICE LAB (EC352)

PREPARED BY

CHESTI ALTAFF HUSSAIN M.Tech

Lecturer

&

M.RAJANI DEVI M.Tech

Lecturer

Department of Electronics and Communications Engineering
Bapatla Engineering College
(Affiliated to Acharya Nagarjuna University)
Bapatla-522101
2009-2010

LIST OF PROGRAMS

PROGRAMS BASED ON OOPS

PROGRAMS BASED ON PSPICE

PROGRAMS BASED ON OOPS

1. A C++ Program on Arrays.

```
 #include<iostream.h>
#include<constream.h>
class ac
{
  private:
  char name[15];
  int acno;
  float bal;
  public:
  void read()
  {
    cout<<"name,acno,bal";
    cin>>name>>acno>>bal;
  }
 friend void showbal(ac);
};
cout<<"bal"<<a.bal<<"name"<<a.name<<"acno"<<a.acno;
}
int main()
{
 ac k;
 k.read();
 showbal(k);
 return(0);

}
```

2. A C++ Program on Pointers.

```
#include<iostream.h>
#include<constream.h>
class bill
{
public:
int qty;
float amount,price;
public:
void getdata(int a,float b,float c)
{
qty=a;
price=b;
amount=c;
}
void show()
{
cout<<"quantity="<<qty<<"\n";
cout<<"price="<<price<<"\n";
cout<<"amount="<<amount<<"\n";
}
};
int main()
{
clrscr();
bill s;
bill *ptr=&s;
ptr->getdata(45,10.25,450);
(*ptr).show();
return 0;
}
```

3. A C++ Program to illustrate the concept of Console I/O Operations.

```
#include<iostream.h>
#include<conio.h>
void main()
 {
  cout.fill('/');
  cout.width(20);
  cout<<"well";
  cout.fill('_');
  cout.width(20);
  cout<<"done" ;
  cout.precision(2);
  cout<<113.5678;
  cout.fill('=');
  cout.setf(ios::right,ios::adjustfield);
  cout.width(20);
  cout<<"figure"<<"\n";
  cout.setf(ios::left,ios::adjustfield);
  cout.width(20);
  cout<<"figure"<<"\n";
  }
```

4. A C++ Program on Objects and Classes.

```
#include<iostream.h>
#include<conio.h>
Class amount
{
Public;
Int number;
Float bal;
Display()
{
Cout<<”account number “<<number<<endl

Cout<<”account balance “<<bal<<endl
}
}
Main()
{
Account raja;
Raja.number=1245;
Raja.balance=2435;
Raja.display();
Account kumar;
Kumar.number=4564;
Kumar.balance=5677;
}
```

5. A C++ Program on Operator Overloading.

```
#include<iostream.h>
#include<constream.h>
#include<conio.h>
class num
 {
   private:
        int a,b,c,d;
   public:
        num(int x,int y,int z,int w)
          {
           a=x;b=y;c=z;d=w;
           }

         void show(void);


           void operator-();
           };
           void num::show()
           {
              cout<<"\na:"<<a<<"\n b:"<<b<<"\n c:"<<c<<"\n d:"<<d;
              }
           void num::operator-()
              {
               a=-a;b=-b;c=-c;d=-d;
              }
              main()
              {
                clrscr();
                num x(2,2,8,4);
                cout<<"before negation:";
                x.show();
                -x;
                cout<<"\n after negation:" ;
                x.show();
                return 0;
               }
```

6. A C++ Program on Function Overloading.

```
#include<iostream.h>
#include<constream.h>
#include<conio.h>
Int sqr(int);
Float sqr(float);
Main()
{
Int a=8;
Float b=3.16
Cout<<”square =”<<sqr(a);
Cout<<”square=”<<sqr(b);
Return(0);
}
Int sqr(int x)
{
Return(x*x);
}
Float sqr(float y)
{
Return(y*y);
}
```

7. A C++ Program on File I/O Operations.

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
int main()
  {
    clrscr();
    char name[20];
    int age;
    ofstream out("text");
    cout<<"name";
    cin>>name;
    cout<<"age:";
    cin>>age;
    out<<name<<"\t";
    out<<age<<"\n";
    out.close();
    ifstream in("text");
    in>>name;
    in>>age;
    cout<<"\n name:"<<name<<"\n";
    cout<<"age"<<age;
    in.close();
    return
    0;
    }
```

8. A C++ Program on Virtual Fuuctions.

```
#include<iostream.h>
#include<constream.h>
class first
{
int b;
public:
first()
{
b=10;
}
virtual void display()
{
cout<<"\nb:"<<b;
}
};
class second:public first
{
int d;
public:
second()
{
d=20;
}
void display()
{
cout<<"\nd:"<<d;
}
};
void main()
{
clrscr();
first f;
second s;
first *pt=&f;
pt->display();
pt=&s;
pt->display();  }
```

9.A C++ Program on Inheritance.

```
#include<iostream.h>
#include<constream.h>
#include<conio.h>
class A1
 {
   protected:
    char name[20];
    int age;
    };
   class A2:public A1

   {    protected:       float height;
                 float weight;
                };
   class A3:public A2
      {
       protected:char sex;
       public:
              void get()
               {
                cout<<"name:" ;
                cin>>name;
                cout<<"age:";
                cin>>age;
                cout<<"sex";
                cin>>sex;
                cout<<"height";
                cin>>height;
                cout<<"weight";
                cin>>weight;
                }
        void show()
        {
          cout<<"\n name:" <<name;
          cout<<"\n age:"<<age;
          cout<<"\n sex:"<<sex;
          cout<<"\n height:"<<height;
          cout<<"\n weight:"<<weight;
```

```
 }
};
void main()
 {
   clrscr();

     A3 x;
   x.get();
   x.show();
  }
```

PROGRAMS BASED ON PSPICE

1. Verification of Half–Wave and Full-Wave Rectifier

AIM: To Verify the characteristics of Half–Wave and Full-Wave Rectifier.

CIRCUIT DIAGRAMS:

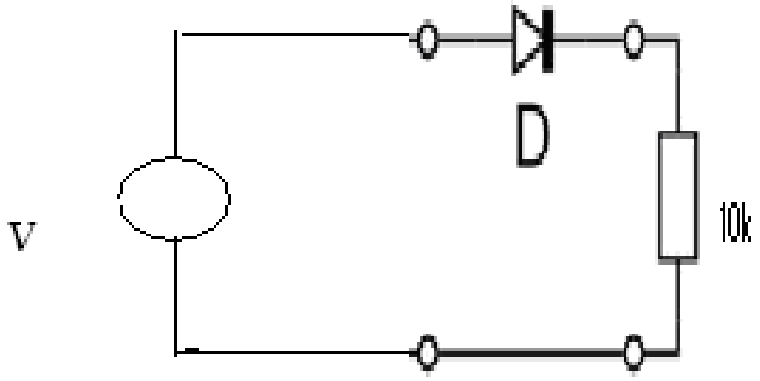


Figure1: Half Wave Rectifier

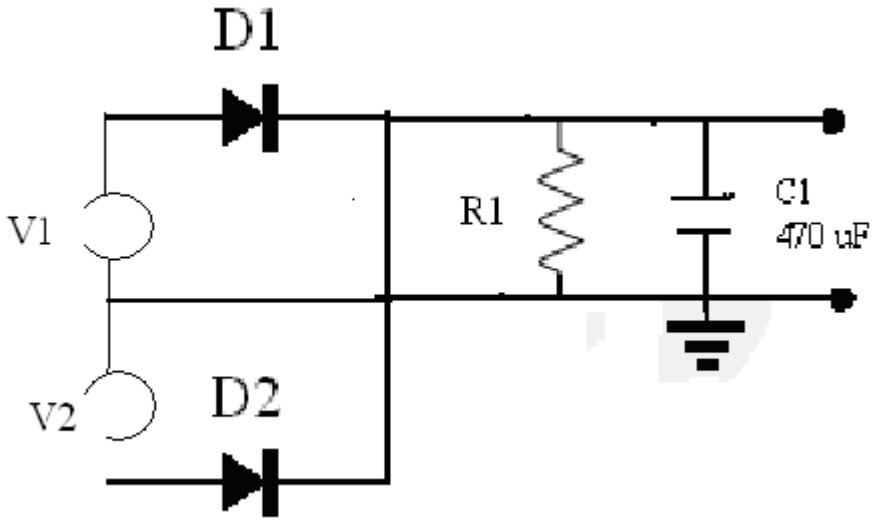


Figure2: Full Wave Rectifier

PROGRAMS:

Half Wave Rectifier

```
v1          1          0          SIN(0  12v  1kHZ)
DA          1          2          D1
R           2          0          1k
.MODEL      D1         D
.TRAN       0.1ms      1ms
.PROBE
```

Full Wave Rectifier

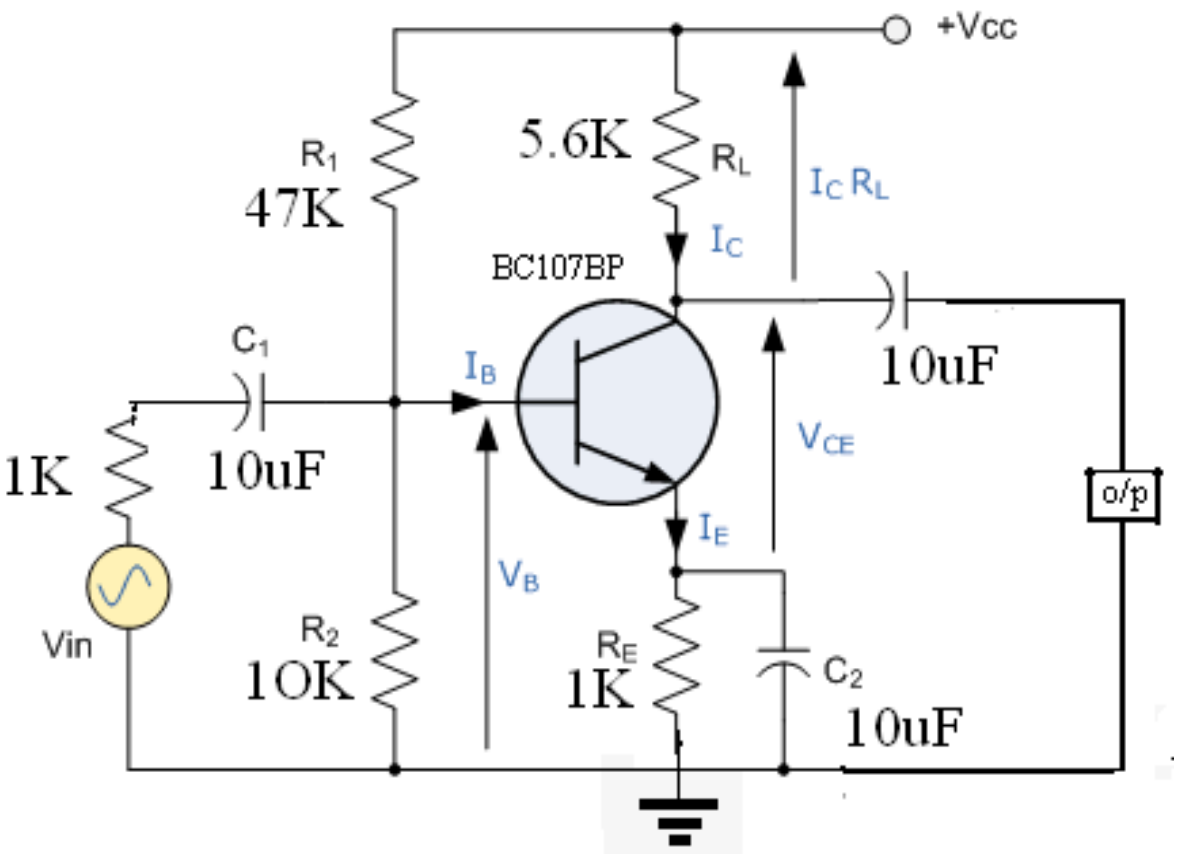
```
v1          1          0          SIN(0   12v   60HZ)
v2          0          3          SIN(0   12v   60HZ)
R           2          0          1K
D1          1          2          DA
D2          3          2          DA
.MODEL      DA         D
.TRAN       0.1ms      25ms
.PROBE
```

Result :

2. Verification CE Amplifier

AIM: To Verify the characteristics of CE Amplifier.

CIRCUIT DIAGRAM:



PROGRAM FOR FREQUENCY RESPONSE:

```
vs          1          0          AC        1mv
vcc         4          0          20v
rs          1          2          1k
r1          4          3          40K
r2          3          0          10K
rc          4          5          4K
re          6          0          2K
rl          7          0          2.2K
cs          2          3          10uf
ce          6          0          20uf
cc          5          7          1uf
q1          5          3          6          Q2N2222
c1          5          3          15pf
c2          3          6          3pf
.AC         DEC        10         10         1000K
.MODEL      Q2N2222    NPN(BF=100)
.PLOT       AC    V(7,0)    V(1,0)
.PROBE
```

PROGRAM FOR TRANSIENT ANALYSIS:

```
vs          1          0          SIN(0    1mv      60HZ)
vcc         4          0          20v
rs          1          2          1k
r1          4          3          40K
r2          3          0          10K
rc          4          5          4K
re          6          0          2K
rl          7          0          2.2K
cs          2          3          10uf
ce          6          0          20uf
cc          5          7          1uf
q1          5          3          6          Q2N2222
c1          5          3          15pf
c2          3          6          3pf
.AC         LIN        1          100HZ      100HZ
.PRINT      AC         I(r1)
.MODEL      Q2N2222    NPN(BF=100)
```

```
.TRAN      0.1ms      50ms
.PROBE
.END
```

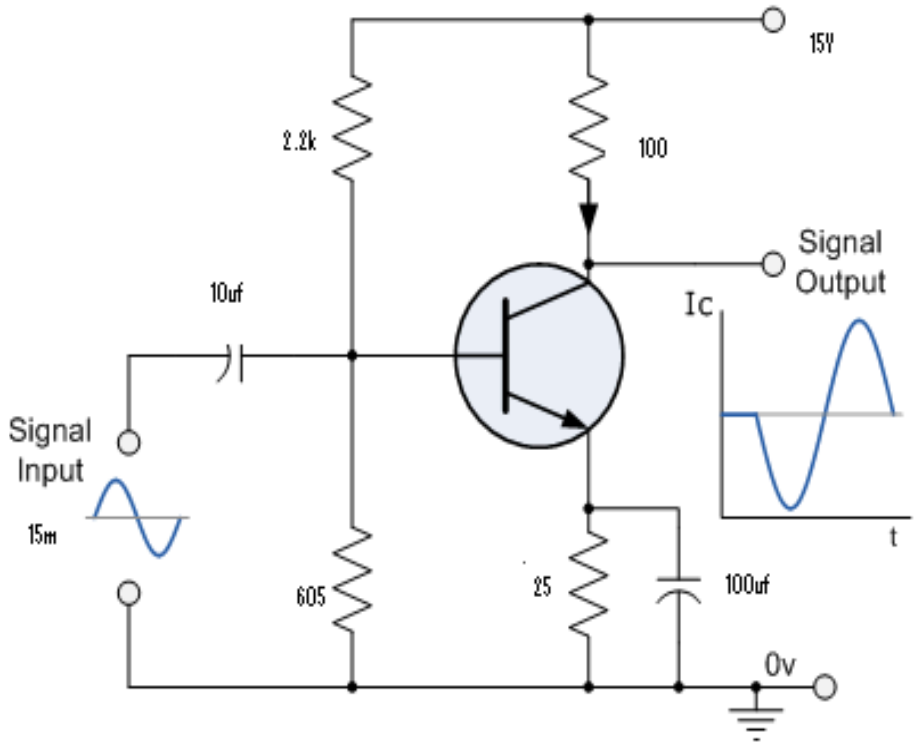
Result :

3. CLASS A POWER AMPLIFIER

AIM: To verify the characteristics of Class A power Amplifier.

CIRCUIT DIAGRAM:

.



PROGRAM FOR CLASS A POWER AMPLIFIER:

VS	1	0	SIN(0 1mv 10KHZ)	
VCC	5	0	15V	
CB	1	2	10UF	
CC	3	6	10UF	
CE	4	0	100UF	
R1	5	2	2.7K	
R2	2	0	605	
RC	5	3	100	
RE	4	0	25	
RL	6	0	47	
Q1	3	2	4	SL100

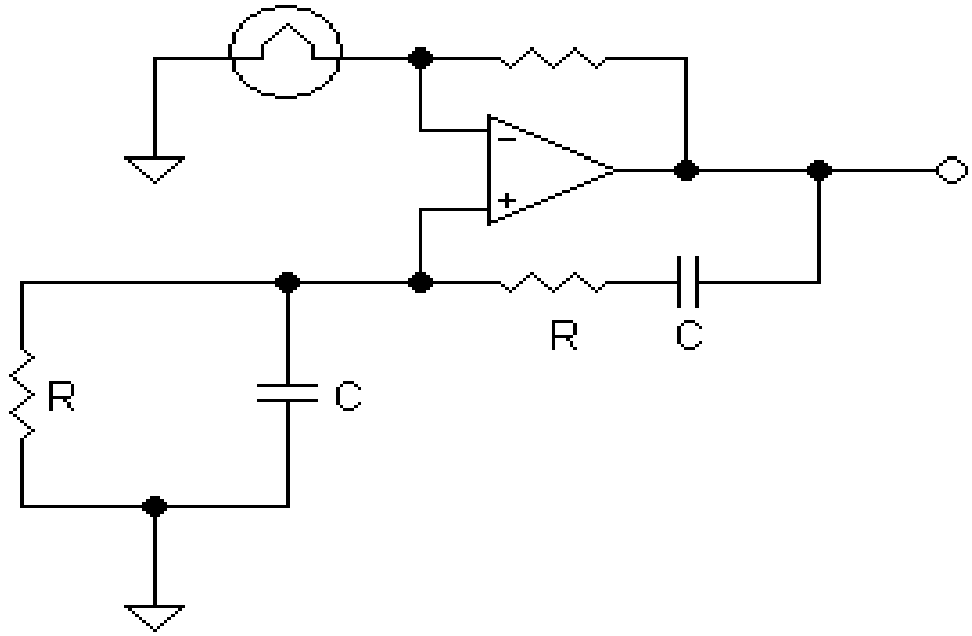
```
.MODEL      SL100      NPN
.TRAN       0.1ms      0.5ms
.PROBE
.END
```

RESULT:

4. WEIN BRIDGE OSCILLATOR

AIM: TO VERIFY THE CHARACTERISTICS OF WEIN BRIDGE OSCILLATOR

CIRCUIT DIAGRAM:



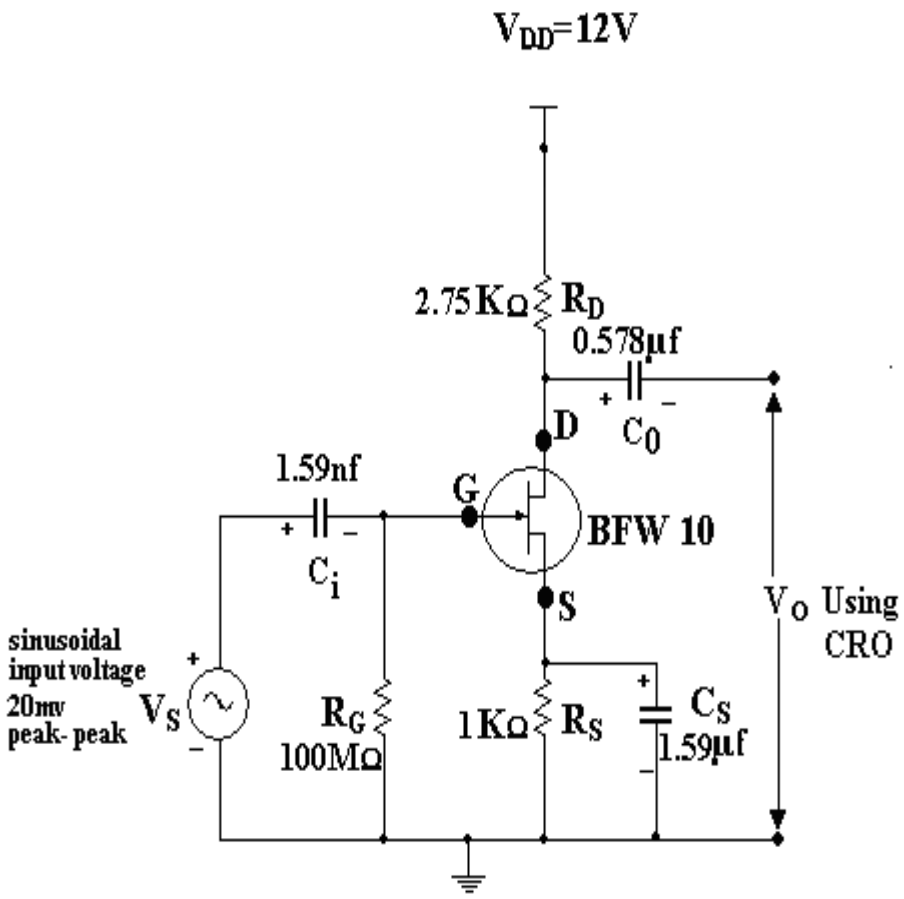
PROGRAM:

```
E       2       0       4       1       2E5
RI      4       1       1E6
RG      1       0       10K
R1      3       4       6366
R2      4       0       6366
RF      2       1       20K
C1      2       3       1NF     IC=2
C2      4       0       1NF
.TRAN   0.05us  50us    UIC
.PROBE
```

5. COMMON SOURCE AMPLIFIER

AIM: To Verify the characteristics of CS Amplifier.

CIRCUIT DIAGRAM:



PROGRAM :

CS AMP

```
vs      1       0       ac      20mv
R       2       0       100M
cg      1       2       1.59uf
R1      2       0       100K
R2      2       5       1MEG
Rs      4       0       1K
Rd      3       5       2.75K
VDD     5       0       15v
Cd      3       6       15uf
Cs      4       0       50uf
RL      6       0       20K
jn      3       2       4       BFW10
.MODEL  BFW10   njf
.AC     DEC     10      10      10MEG
.PLOT   AC      V(6,0)  V(1,0)
.PROBE
.END
```

```
vs      1       0       SIN(0   10mv    5KHZ)
R       1       7       15K
cg      7       2       4.7uf
R1      2       0       100K
R2      2       5       1MEG
Rs      4       0       1.5K
Rd      3       5       4.7K
VDD     5       0       15v
Cd      3       6       15uf
Cs      4       0       50uf
RL      6       0       20K
jn      3       2       4       BFW10
.MODEL  BFW10   njf
.TRAN   0.1ms   0.6ms
.PROBE
.END
```