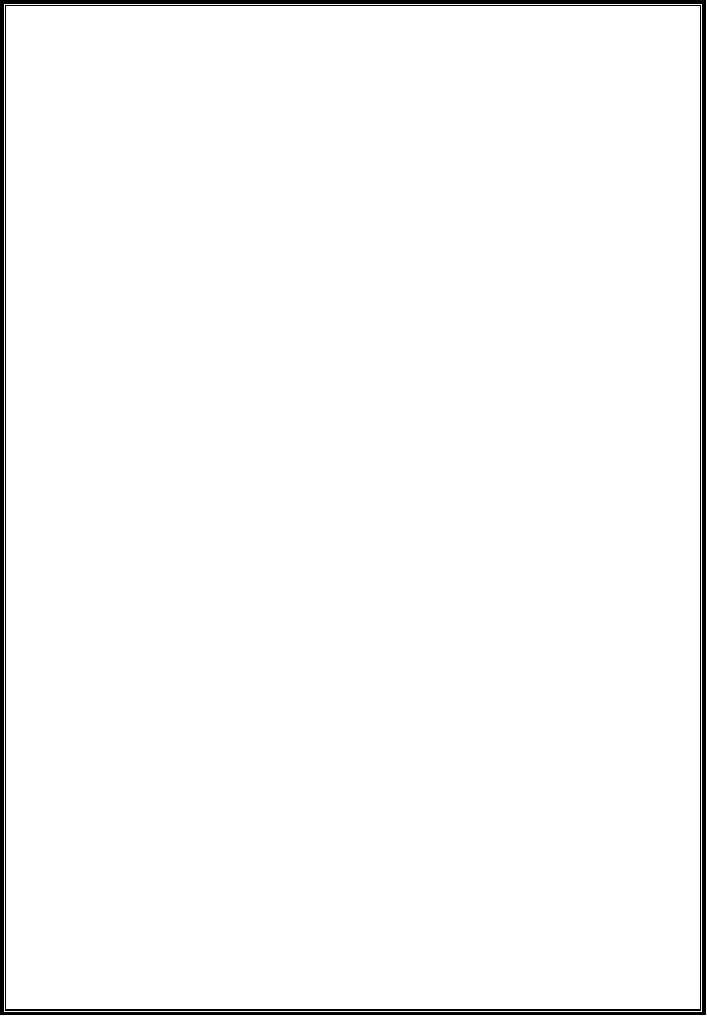
# Математическое моделирование биполярных транзисторов типа p-n-p



Министерство  образования  РФ



Южно-Российский Государственный Технический Университет (НПИ)

   ФАКУЛЬТЕТ                                          ***ИТУ***                                                         \_

   КАФЕДРА                                              ***А и Т***\_

   СПЕЦИАЛЬНОСТЬ                                ***Электроника***                                 \_

## Пояснительная записка

## к курсовой работе

#### *по дисциплине* Методы анализа и расчёта электронных схем

#### *на тему* Математическое моделирование биполярных транзисторов типа p-n-p

*Выполнил студент*        III  курса, группы 1б            Евченко С.  Е.           \_\_

                                                                                                                                                (Ф.И.О.)

*Принял*             \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_                        \_\_\_\_\_      Савёлов Н. С.    \_\_\_      \_

                                                                                                                                            (Ф.И.О.)

НОВОЧЕРКАССК  2001

Содер жание

Введение

1. Проблема математического моделирования биполярных транзисторов

2. Описание изучаемого алгоритма

3. Описание программы

4. Решение контрольной задачи

5. Моделирование электронных устройств

Заключение

Введение

Развитие вычислительной техники и повышение требований к развиваемой электронной аппаратуре выдвинули на первый план создание систем автоматического проектирования.

До начала шестидесятых годов вычислительные методы использовались при анализе и проектировании цепей крайне незначительно. Квалифицированный инженер мог  синтезировать простые цепи, пользуясь минимумом вычислений. Он создавал макет схемы, производил измерения и различные модификации и в результате получал конечный вариант цепи.

            В следующем времени ситуация сильно изменилась. Появились интегральные схемы и стали доступными ЭВМ. Оба эти обстоятельства повлияли друг на друга. Интегральные схемы сделали возможным производство более совершенных ЭВМ, а те в свою очередь облегчили проектирование новых интегральных схем. Несомненно, что в этой связи вычислительные методы  стали иметь огромное значение.

            Технический прогресс  сделал возможным проектирование больших функциональных блоков, содержащих в одной схеме тысячи взаимосвязанных транзисторов. Разработка таких схем стала невозможна при экспериментальной  отладке на макете.

            Достоверное моделирование различных полупроводниковых приборов достаточно критично. Проблема моделирования полупроводниковых приборов требует знания физики полупроводников. Но если специалист в области физики полупроводников может предоставить необходимые уравнения специалисту в области САПР, то последнему остаётся ввести их в свои программы. При этом в значительной степени устраняются трудности, связанные с физикой работы прибора.

            Если в уравнения, описывающие схему, введены нелинейные модели компонентов, то решить эти уравнения становится достаточно сложно. При этом в программах САПРа затраты машинного времени на определение нелинейных функций, описывающих различные полупроводниковые приборы составляют значительную часть общих затрат времени.

1.    Проблема математического моделирования биполярных транзисторов

Под моделированием понимается описание электрических свойств полупроводникового устройства или группы таких устройств, связанных между собой, с помощью математических уравнений, эквивалентных схем  или таблиц.

Термин “биполярный транзистор” связан с тем, что в нём используются носители заряда двух видов: электроны и дырки. Слово “транзистор” (от англ. transfer resistor) означает, что этот прибор согласует низкоомную цепь эмиттера с высокоомной цепью коллектора.

  При описании сложных устройств и многомерных систем используются их выходные характеристики, на основе которых создаются макромодели. Модели, определяющие электрические свойства, используются для точного анализа.

При математическом моделировании главными проблемами являются:

- отсутствие математических зависимостей, точно описывающих реальные биполярные транзисторы.

-из-за увеличения параметров, описывающих математическую модель, усложняются и формулы, по которым производится моделирование.

 Это, в свою очередь, приводит к потребности от компьютера всё больше его ресурсов и всё больше времени на решение конкретных задач. Полученные требования реально ограничивают возможность использования сложных формул.

На практике, в зависимости оп поставленной задачи,  применяются математические модели разной сложности. Так в программах обучающего типа используются модели, описывающие только основные параметры. Такие как:

-      зависимости токов от напряжений;

-      зависимость токов от температуры;

-      зависимость барьерных ёмкостей от напряжений, приложенных к переходам;

-      зависимость диффузионных ёмкостей от токов, проходящих через переходы.

В программах, предназначенных для моделирования сложных реальных схем, используются более сложные математические зависимости.

Реальные транзисторы имеют определённый разброс своих характеристик. Это приходится учитывать при моделировании для получения требуемого результата.

Каждый вид транзистора, выпускаемый на заводах, имеет набор параметров и характеристик, описанных в паспортных данных. На данный момент времени количество таких параметров достигает сотни. При математическом моделировании не всегда возможно все их учесть, а принимаются по умолчанию некоторые средние значения, которые относительно точно отражают физические характеристики. Эти допущения не всегда точно моделируют реальные биполярные транзисторы.

2.    Описание изучаемого алгоритма

Биполярный транзистор состоит из двух взаимодействующих p-n перехода, созданных в объёме  монокристалла кремния или германия. В зависимости от характера примесей в этих областях принято различать транзисторы типа p-n-p  и n-p-n.

Модель Эберса - Молла  применима при моделировании по постоянному току.



                                                                                  Э



Uэк



Б



                                                                                     К

Рис. 1. Биполярный транзистор



                                                                                                                           эмиттер



*DBE*.



*RБ*.                                      *CCS*



            база



*DBC*.                                              *RC* П - подложка



                                                                                                                               коллектор



Рис. 2. Модель Эберса – Молла.

**1)** расчёт температурного потенциала.

;



  - постоянная Больцмана;



*Кл* - элементарный заряд.



*T*  - текущая температура в Кельвинах.

**2)** расчёт тока насыщения, зависящего от температуры и

    напряжения между эмиттером и коллектором.



*iso  -* ток насыщения при *TNOM;*

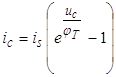
*VA* – напряжение Эрли;

*EG* – ширина запрещённой зоны;

*TNOM –* номинальная температура в Кельвинах*.*

**3)** определения величины *IС* и *UС*  - ток и напряжение на диоде при переходе на линейный участок.

;                            ;



*RJ* – минимальное дифференциальное сопротивление перехода;

**4)** перевод температурного коэффициента в систему С.

TCB (в системе С) = TCB\*10-6 (в PPM) .

**5)** поправка коэффициентов передачи по току в схеме с ОЭ на температуру.

BF (исправленное)  = BF \* (1 + (T-TNOM) \* TCB;

BR (исправленное)  = BR \* (1 + (T-TNOM) \* TCB.

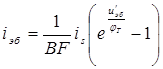
*BF* – коэффициент передачи по току в схеме с ОЭ для нормального включения.

*BR* – коэффициент передачи по току в схеме с ОЭ для инверсного включения.

**6)** Математическое описание зависимости тока диода DBE от напряжения на

    нем.

 при ;



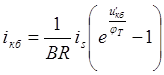
 при ;



**д)** Математическое описание зависимости тока диода DBC от напряжения на

    нем.

 при ;



при ;



**e)** математическое описание управляемого источника тока.

.

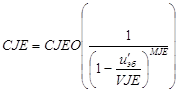


**ж)** расчёт токов коллектора и эмиттера;

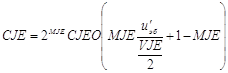
ik = iу - iкб ;             iэ = iу + iэб ;

**и)** расчёт барьерной емкости перехода ЭБ;.

 при ;



 при ;



*CJEO* – барьерная емкость перехода ЭБ при нулевом смещении;

*VJE* – контактная разность потенциалов для переходов ЭБ;

*MJE* – показатель степени в выражении для барьерной емкости.

**к)** расчёт диффузионной емкости перехода ЭБ.

 при >0;



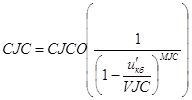
*TAUF* – среднее время пролета носителей через базу в нормальном режиме.

**л)** расчёт емкости *CBE*.

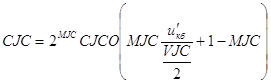
*CBE = CJE+CDE*;

**м)** расчёт барьерной емкости перехода КБ.

 при ;



при ;



*CJCO* – барьерная емкость перехода КБ при нулевом смещении;

*VJC* – контактная разность для переходов КБ;

*MJC* – показатель степени в выражении для барьерной емкости.

**н)** расчёт диффузионной емкости перехода КБ.

 при >0;



CDC = 0 при 0;



*TAUR* – среднее время пролета носителей через базу в инверсном режиме.

**о)** расчёт емкости CBC.

*CBC= CJC+CDC;*

3.    Описание программы

Program bipolar;

uses {wincrt,windos}crt,dos,graph;

Label 1;

var

     t1,t2,t3,t4,

     Uc, Uccs, Ucb, Uec, Ueb, VA, VJC, VJE,

     Is0, Is, Ieb, Icb ,Ib ,Ic, Iy,Ik,Ie,

     CJC0, CJE0, CBE, CJE, CDE, CBC, CJC, CDC,

     RB,RC,RJ,

     T, TNOM, TCB,

     Kyf,a,BF, BR, EG, TAUF, TAUR, MJC, MJE, CCS, K, FIt, q,expon:  extended;

     ii,col,row:integer;                  {    extended  3.4e-4932..1.1e4932}

     an,key:char;

function st(a:extended; b:extended):extended;

begin

    if b<0 then

    begin

         if (-1\*b \* ln(a)) < 1.1356523e4 then

         begin

         st:=1/ (exp((-1\*b) \* ln(a)));

         end

         else st:= 1;

    end

    else

    begin

         if (b \* ln(a)) > 1.1356523e4 then

         begin

              st:=1e4000;

         end

         else

         begin

              st:=exp((b) \* ln(a));

         end;

    end;

end;

Procedure Model(Ueb,Ucb:extended;var Ieb,Icb,Ib,Iy:extended);

Begin

     If Ueb > Uc then Ieb := 1 / BF \* (Ic+ (Ueb - Uc) / RJ )

     else             Ieb :=1 / BF \* Is \* (st(expon,Ueb / FIt) -1);

     If Ucb > Uc then Icb:= 1/BR \* (Ic + (Ucb-Uc)/RJ)

     else     Icb:= 1/BR \* Is\* (st(expon, Ucb/FIt)-1);

     Ib:= Ieb + Icb;

     Iy:= Ieb\*BF - Icb\*BR;

end;

procedure InputTrans;

Label 1,2,3;

begin

     clrscr;

          BF:=75;

          BR:= 0.3;

          TCB:=2500;

          Is0:=3.5e-15;

          EG:=1.11;

          CJC0:=4e-12;

          CJE0:=1.2e-12;

          RB:=67;

          RC:=7.3;

          VA:=102;

          TAUF:=9.4e-11;

          TAUR:=6.692e-8;

          MJC:=0.33;

          VJC:=0.65;

          MJE:=0.69;

          VJE:=0.69;

          CCS:=1e-12;

          RJ:=0.01;

     writeln('       It is default parameters of transistor KT316B (Y/N) ');

     gotoxy(col+3,row+3);

     write('[Forward beta] ');

     gotoxy(col+39,row+3);

     write(BF);

     gotoxy(col+3,row+4);

     write('[Revers beta] ');

     gotoxy(col+39,row+4);

     write(BR);

     gotoxy(col+3,row+5);

     write('[Temp. coef. of BETTA (PPM)] ');

     gotoxy(col+39,row+5);

     write(TCB);

     gotoxy(col+3,row+6);

     write('[Saturation Current] ');

     gotoxy(col+39,row+6);

     write(Is0);

     gotoxy(col+3,row+7);

     write('[Energy gap (0.6 to 1.3)] ');

     gotoxy(col+39,row+7);

     write(EG);

     EG:=EG\*1.6e-19;

     gotoxy(col+3,row+8);

     write('[CJC0] ');

     gotoxy(col+39,row+8);

     write(CJC0);

     gotoxy(col+3,row+9);

     write('[CJE0] ');

     gotoxy(col+39,row+9);

     write(CJE0);

     gotoxy(col+3,row+10);

     write('[Base resistance] ');

     gotoxy(col+39,row+10);

     write(RB);

     gotoxy(col+3,row+11);

     write('[Collector resistance] ');

     gotoxy(col+39,row+11);

     write(RC);

     gotoxy(col+3,row+12);

     write('[Early Valtage] ');

     gotoxy(col+39,row+12);

     write(VA);

     gotoxy(col+3,row+13);

     write('[TAU forward] ');

     gotoxy(col+39,row+13);

     write(TAUF);

     gotoxy(col+3,row+14);

     write('[TAU reverse] ');

     gotoxy(col+39,row+14);

     write(TAUR);

     gotoxy(col+3,row+15);

     write('[MJC] ');

     gotoxy(col+39,row+15);

     write(MJC);

     gotoxy(col+3,row+16);

     write('[VJC] ');

     gotoxy(col+39,row+16);

     write(VJC);

     gotoxy(col+3,row+17);

     write('[MJE] ');

     gotoxy(col+39,row+17);

     write(MJE);

     gotoxy(col+3,row+18);

     write('[VJE] ');

     gotoxy(col+39,row+18);

     write(VJE);

     gotoxy(col+3,row+19);

     write('[CSUB] ');

     gotoxy(col+39,row+19);

     write(CCS);

     gotoxy(col+3,row+20);

     write('[Minimum junction resistance] ');

     gotoxy(col+39,row+20);

     write(RJ);

     gotoxy(col+6,row+25);

     write('Accept parameters of transistor (Y/N) ');

     an:=readkey;

         case an of 'y':    goto 3;

                    'Y':    goto 3;

                    'n':    goto 2;

                    'N':    goto 2;

                 else

                 begin

                     sound(500);

                     delay(1000);

                     nosound;

                     goto 1;

                 end;

          end;

2:   clrscr;

     gotoxy(col+25,row+1);

     write('Input next parameters of transistor');

     gotoxy(col+3,row+3);

     write('[Forward beta] ');

     gotoxy(col+40,row+3);

     read(BF);

     gotoxy(col+39,row+3);

     write(BF);

     gotoxy(col+3,row+4);

     write('[Revers beta] ');

     gotoxy(col+40,row+4);

     read(BR);

     gotoxy(col+39,row+4);

     write(BR);

     gotoxy(col+3,row+5);

     write('[Temp. coef. of BETTA (PPM)] ');

     gotoxy(col+40,row+5);

     read(TCB);

     gotoxy(col+39,row+5);

     write(TCB);

     gotoxy(col+3,row+6);

     write('[Saturation Current] ');

     gotoxy(col+40,row+6);

     read(Is0);

     gotoxy(col+39,row+6);

     write(Is0);

     gotoxy(col+3,row+7);

     write('[Energy gap (0.6 to 1.3)] ');

     gotoxy(col+40,row+7);

     read(EG);

     gotoxy(col+39,row+7);

     write(EG);

     gotoxy(col+3,row+8);

     write('[CJC0] ');

     gotoxy(col+40,row+8);

     read(CJC0);

     gotoxy(col+39,row+8);

     write(CJC0);

     gotoxy(col+3,row+9);

     write('[CJE0] ');

     gotoxy(col+40,row+9);

     read(CJE0);

     gotoxy(col+39,row+9);

     write(CJE0);

     gotoxy(col+3,row+10);

     write('[Base resistance] ');

     gotoxy(col+40,row+10);

     read(RB);

     gotoxy(col+39,row+10);

     write(RB);

     gotoxy(col+3,row+11);

     write('[Collector resistance] ');

     gotoxy(col+40,row+11);

     read(RC);

     gotoxy(col+39,row+11);

     write(RC);

     gotoxy(col+3,row+12);

     write('[Early Valtage] ');

     gotoxy(col+40,row+12);

     read(VA);

     gotoxy(col+39,row+12);

     write(VA);

     gotoxy(col+3,row+13);

     write('[TAU forward] ');

     gotoxy(col+40,row+13);

     read(TAUF);

     gotoxy(col+39,row+13);

     write(TAUF);

     gotoxy(col+3,row+14);

     write('[TAU reverse] ');

     gotoxy(col+40,row+14);

     read(TAUR);

     gotoxy(col+39,row+14);

     write(TAUR);

     gotoxy(col+3,row+15);

     write('[MJC] ');

     gotoxy(col+40,row+15);

     read(MJC);

     gotoxy(col+39,row+15);

     write(MJC);

     gotoxy(col+3,row+16);

     write('[VJC] ');

     gotoxy(col+40,row+16);

     read(VJC);

     gotoxy(col+39,row+16);

     write(VJC);

     gotoxy(col+3,row+17);

     write('[MJE] ');

     gotoxy(col+40,row+17);

     read(MJE);

     gotoxy(col+39,row+17);

     write(MJE);

     gotoxy(col+3,row+18);

     write('[VJE] ');

     gotoxy(col+40,row+18);

     read(VJE);

     gotoxy(col+39,row+18);

     write(VJE);

     gotoxy(col+3,row+19);

     write('[CSUB] ');

     gotoxy(col+40,row+19);

     read(CCS);

     gotoxy(col+39,row+19);

     write(CCS);

     gotoxy(col+3,row+20);

     write('[Minimum junction resistance] ');

     gotoxy(col+40,row+20);

     read(RJ);

     gotoxy(col+39,row+20);

     write(RJ);

     writeln;

1:   gotoxy(col+6,row+25);

     write('Accept parameters of transistor (Y/N) ');

     an:=readkey;

         case an of  'y':    goto 3;

                     'Y':    goto 3;

                     'n':    goto 2;

                     'N':    goto 2;

                 else

                 begin

                     sound(500);

                     delay(1000);

                     nosound;

                     goto 1;

                 end;

          end;

3:;

end;

Procedure InputCurrent;

Label 1,2,3;

begin

     clrscr;

     TNOM:=27;

     T:=21;

     Ueb:=0.8;

     Uec:=2;

     Ucb:=-1.2;

     gotoxy(col+25,row+1);

     write(' Default parameters');

     gotoxy(col+3,row+3);

     write('[Nominal temperature (C)] ');

     gotoxy(col+39,row+3);

     write(TNOM);

     gotoxy(col+3,row+4);

     write('[Current temperature (C)] ');

     gotoxy(col+39,row+4);

     write(T);

     gotoxy(col+3,row+5);

     write('[Emitter-Base voltage] ');

     gotoxy(col+39,row+5);

     write(Ueb);

     gotoxy(col+3,row+6);

     write('[Emitter-Collector voltage] ');

     gotoxy(col+39,row+6);

     write(Uec);

     gotoxy(col+3,row+7);

     write('[Collector-Base voltage] ');

     gotoxy(col+39,row+7);

     write(Ucb);

1:   gotoxy(col+6,row+25);

     write('Accept current parameters (Y/N) ');

     an:=readkey;

         case an of 'y':    goto 3;

                    'Y':    goto 3;

                    'n':    goto 2;

                    'N':    goto 2;

                 else

                 begin

                     sound(500);

                     delay(1000);

                     nosound;

                     goto 1;

                 end;

          end;

2:   clrscr;

     gotoxy(col+20,row+2);

     write('Input next current parameters');

     gotoxy(col+3,row+3);

     write('[Nominal temperature (C)] ');

     gotoxy(col+40,row+3);

     read(TNOM);

     gotoxy(col+39,row+3);

     write(TNOM);

     gotoxy(col+3,row+4);

     write('[Current temperature (C)] ');

     gotoxy(col+40,row+4);

     read(T);

     gotoxy(col+39,row+4);

     write(T);

     gotoxy(col+3,row+5);

     write('[Emitter-Base voltage] ');

     gotoxy(col+40,row+5);

     read(Ueb);

     gotoxy(col+39,row+5);

     write(Ueb);

     gotoxy(col+3,row+6);

     write('[Emitter-Collector voltage] ');

     gotoxy(col+40,row+6);

     read(Uec);

     gotoxy(col+39,row+6);

     write(Uec);

     if (Ueb = 0) or (Uec = 0) then

     begin

          gotoxy(col+3,row+7);

          write('[Collector-Base voltage] ');

          gotoxy(col+40,row+7);

          read(Ucb);

          gotoxy(col+39,row+7);

          write(Ucb);

          if Uec <> 0 then

          begin

               Ueb := Uec + Ucb;

               gotoxy(col+39,row+5);

               write(Ueb);

          end

          else

          begin

               Uec := Ueb - Ucb;

               gotoxy(col+39,row+6);

               write(Uec);

          end;

          goto 1;

     end;

     Ucb := -(Uec - Ueb);

     gotoxy(col+3,row+7);

     write('[Collector-Base voltage] ');

     gotoxy(col+39,row+7);

     write(Ucb);

3:   TNOM:=TNOM+273.15;

     T:=T+273.15;

     BF:=BF\*(1+(T-TNOM)\*TCB\*10e-6);

     BR:=BR\*(1+(T-TNOM)\*TCB\*10e-6);

     gotoxy(col+39,row+8);

end;

procedure OutputResult;

begin

     clrscr;

     writeln('                             It is result ');

     K:=1.38e-23;

     q:=1.6e-19;

     expon:=exp(1.0);

     FIt:=K\*T/q;

     gotoxy(col+3,row+3);

     write('[FIt] ');

     gotoxy(col+39,row+3);

     write(FIt);

     Is:=Is0 \* (1+ Uec/VA) \* st(T/TNOM,3) \* st(expon,-EG/K\*(1/T-1/TNOM));

     gotoxy(col+3,row+4);

     write('[Is]');

     gotoxy(col+39,row+4);

     write(Is);

     Uc:=Fit\*ln(FIt/(Is\*RJ));

     gotoxy(col+3,row+5);

     write('[Uc] ');

     gotoxy(col+39,row+5);

     write(Uc);

     if (st(expon,(Uec/FIt))-1) < 1e4800/Is then

          Ic:= Is \* (st(expon,Uc/FIt)-1)

     else Ic:= 1e4800;

     gotoxy(col+3,row+6);

     write('[Ic]');

     gotoxy(col+39,row+6);

     write(Ic);

     Model(Ueb,Ucb,Ieb,Icb,Ib,Iy);

     gotoxy(col+3,row+7);

     write('[Ieb] ');

     gotoxy(col+39,row+7);

     write(Ieb);

     gotoxy(col+3,row+8);

     write('[Icb] ');

     gotoxy(col+39,row+8);

     write(Icb);

     gotoxy(col+3,row+9);

     write('[Ib] ');

     gotoxy(col+39,row+9);

     write(Ib);

     gotoxy(col+3,row+10);

     write('[Iy] ');

     gotoxy(col+39,row+10);

     write(Iy);

     Ik:=Iy-Icb;

     gotoxy(col+3,row+11);

     write('[Ik] ');

     gotoxy(col+39,row+11);

     write(Ik);

     Ie:=Iy+Ieb;

     gotoxy(col+3,row+12);

     write('[Ie] ');

     gotoxy(col+39,row+12);

     write(Ie);

     If Ueb < VJE/2 then  CJE := CJE0\* 1 / st(1- Ueb/VJE,MJE)

     else           CJE := st(2,MJE) \* CJE0 \* (MJE\* Ueb / VJE/2 + 1 - MJE);

     gotoxy(col+3,row+13);

     write('[CJE] ');

     gotoxy(col+39,row+13);

     write(CJE);

     If Ieb > 0 then      CDE := Ieb \* BF \* TAUF / FIt

     else                 CDE := 0;

     gotoxy(col+3,row+14);

     write('[CDE] ');

     gotoxy(col+39,row+14);

     write(CDE);

     CBE:=CJE+CDE;

     gotoxy(col+3,row+15);

     write('[CBE] ');

     gotoxy(col+39,row+15);

     write(CBE);

     If Ucb < (VJC / 2) then CJC := CJC0 \* 1 / st(1- Ucb/VJC,MJC)

     else     CJC := 2 \* st(2,MJC) \* CJC0 \* (MJC \* Ucb / VJC / 2 + 1 - MJC);

     gotoxy(col+3,row+16);

     write('[CJC] ');

     gotoxy(col+39,row+16);

     write(CJC);

     If Icb > 0 then  CDC := Icb \* BR \* TAUR / FIt

     else             CDC := 0;

     gotoxy(col+3,row+17);

     write('[CDC] ');

     gotoxy(col+39,row+17);

     write(CDC);

     CBC := CJC + CDC;

     gotoxy(col+3,row+18);

     write('[CBC] ');

     gotoxy(col+39,row+18);

     write(CBC);

     gotoxy(col+6,row+25);

     write('Pres any key to Main menu ');

     readkey;

end;

procedure IGraph;

var

  grDriver: Integer;

  grMode: Integer;

  ErrCode: Integer;

  i,x0,y0: Integer;

  stro1,stro2,stro3,stro4:string;

begin

    grDriver := Detect;

    InitGraph(grDriver, grMode,'');

    ErrCode := GraphResult;

    if ErrCode = grOk then

    begin  { Do graphics }

        x0:=320;

        y0:=350;

        Kyf:=0;

        OutTextXY(250,10,'Graphic Ib=f(-Ueb)');

        Line(x0, y0-150, x0, Y0+150);

        Line(x0-200, y0, X0+200, Y0);

        setcolor(0);

        LineTo(x0+200, y0-300);

        setcolor(2);

        for i:=200 downto -200 do

        begin

             Model(Ueb\*(i)/200,-(Uec-Ueb\*(i)/200),Ieb,Icb,Ib,Iy);

             if Kyf=0 then

             begin

                  Kyf:=300/Ib;

                  Str(Ieb,stro1);

                  Str(Ueb,stro2);

             end;

             if abs(Kyf\*Ib)< 10e5 then  Lineto(x0+i, y0-round(Kyf\*Ib));

        end;

        Str(-Ieb,stro3);

        Str(-Ueb,stro4);

        OutTextXY(x0-40,Y0-300,'+Ib = ');

        OutTextXY(x0+3,Y0-300,Stro1);

        OutTextXY(x0+60,Y0+10,'+Ueb = ');

        OutTextXY(x0+100,Y0+10,Stro2);

        OutTextXY(x0-250,Y0+10,'-Ib = ');

        OutTextXY(x0-210,Y0+10,Stro3);

        OutTextXY(x0-300,Y0-10,'-Ueb = ');

        OutTextXY(x0-250,Y0-10,Stro4);

        OutTextXY(40,470,'Pres any key to Main menu');

        Readkey;

        CloseGraph;

    end

    else Writeln('Graphics error:', GraphErrorMsg(ErrCode));

end;

procedure OGraph;

var

  grDriver: Integer;

  grMode: Integer;

  ErrCode: Integer;

  i,x0,y0: Integer;

  stro1,stro2,stro3,stro4:string;

begin

    grDriver := Detect;

    InitGraph(grDriver, grMode,'');

    ErrCode := GraphResult;

    if ErrCode = grOk then

    begin  { Do graphics }

        x0:=320;

        y0:=240;

        Kyf:=0;

        OutTextXY(250,10,'Graphic Ik=f(Uec)');

        Line(x0, y0-150, x0, Y0+150);

        Line(x0-200, y0, X0+200, Y0);

        setcolor(0);

        LineTo(x0+200, round(y0-Kyf\*Ik));

        setcolor(2);

        Kyf:=0;

        for i:=200 downto -200 do

        begin

             if (i>0) or (i=0) then

             begin

                  Is:=Is0 \* (1+ Uec\*(i)/200/VA) \* st(T/TNOM,3) \* st(expon,-EG/K\*(1/T-1/TNOM));

                  Uc:=Fit\*ln(FIt/(Is\*RJ));

                  Ic:= Is \* (st(expon,Uc/FIt)-1);

                  Model(Ueb,Ueb-Uec\*(i)/200,Ieb,Icb,Ib,Iy);

                  Ik:=Iy-Icb;

             end;

             if i<0 then

             begin

                  Is:=Is0 \* (1- Uec\*(i)/200/VA) \* st(T/TNOM,3) \* st(expon,-EG/K\*(1/T-1/TNOM));

                  Uc:=Fit\*ln(FIt/(Is\*RJ));

                  Ic:= Is \* (st(expon,Uc/FIt)-1);

                  Model(Ueb+Uec\*(i)/200,Ueb,Ieb,Icb,Ib,Iy);

                  Ik:=(Iy-Icb);

             end;

             if Kyf=0 then

             begin

                  Kyf:=abs(20/Ik);

                  Str(Ik,stro1);

                  Str(Uec,stro2);

                  setcolor(0);

                  LineTo(x0+200, round(y0-Kyf\*Ik));

                  setcolor(2);

             end;

             if abs(Kyf\*Ik)< 10e5 then

             Lineto(x0+i, y0-round(Kyf\*Ik));

        end;

        OutTextXY(x0-40,Y0-210,'+Ik = ');

        OutTextXY(x0+3,Y0-210,Stro1);

        OutTextXY(x0+60,Y0+10,'+Uec = ');

        OutTextXY(x0+100,Y0+10,Stro2);

        OutTextXY(40,470,'Pres any key to Main menu');

        Readkey;

        CloseGraph;

    end

    else Writeln('Graphics error:', GraphErrorMsg(ErrCode));

end;

begin

     textbackground(1);

     textcolor(14);

     col:=-2;

     row:=0;

repeat

     clrscr;

    writeln('               Mogel p-n-p transistor Eabers-Mol whith 1 SIDI');

     writeln;

     writeln('                                 Main menu');

     writeln;

     writeln('1: Input parameters of transistor');

     writeln('2: Input current parameters');

     writeln('3: Output text result');

     writeln('4: Input-graphic result');

     writeln('5: Output-graphic result');

     writeln('ECS:exit');

     key:=Readkey;

     case key of '1':     InputTrans;

                 '2':     InputCurrent;

                 '3':     OutputResult;

                 '4':     IGraph;

                 '5':     OGraph;

                 #27:     halt ;

                 else

                 begin

                      sound(500);

                      delay(1000);

                      nosound;

                 end;

     end;

     until key=#27;

     {donewincrt;}

end.

4.    Решение контрольной задачи 5.    Моделирование электронных устройств

.model KT315v  NPN(Is=21.11f Xti=3 Eg=1.11 Vaf=157 Bf=81.09 Ise=321.2f

+  Ne=1.458 Ikf=.2017 Nk=.4901 Xtb=1.5 Br=1 Isc=84.36f Nc=1.317

+  Ikr=1.671 Rb=12 Rc=1.426 Cjc=9.716p Mjc=.33 Vjc=.75 Fc=.5 Cje=18.5p

+  Mje=.33 Vje=.75 Tr=275.6n Tf=321.4p Itf=1 Xtf=2 Vtf=60)

           График вида зависимости Ic = f(Ube)            График вида  зависимости Ube = f(Ibe)

                       программы ORCAD 9.1;                                         моей программы

Заключение

Программа, написанная по указанному алгоритму, подтвердила справедливость этих алгоритмов и  относительную точность математического описания биполярных транзисторов. Результаты полученной программы сравнивались с профессиональными CAD программами для одних и тех же транзисторов и одинаковых соответствующих параметрах. Немного изменяя входные параметры возможно достаточно близко приблизится к их результатам.

  Полученная программа позволяет решать поставленные перед нею проблемы, но не обладает удобством в использовании и требует доработки.

В дальнейшем возможно добавление более сложных моделей, позволяющих проводить анализ по постоянному току, но и по переменному на высоких частотах, в режиме малого сигнала.

Для уменьшения погрешности при вычислениях можно написать программу под Windows, где разрядность данных увеличена и достигает 64 бита. Этот метод позволит улучшить удобство в работе с программой.

Библиографический список

1.    Влах Н., Сингхал К.  Машинные методы анализа и проектирования электронных схем.

М. : Радио и связь, 1988 – 560 стр.

2.    В. И. Лачин, Н.С. Савёлов Электроника.

3.    Носов Ю. Р., Петросянц К. О., Шилин В. А. Математические модели Элементов интегральной электроники.

М. : Сов. радио, 1976 – 304 стр.

4.    Сигорский В. П., Петренко А.И. Алгоритмы анализа электронных схем.

М. : Сов. радио, 1976 – 608 стр.

5.    Мигулин И. Н., Чаповский М. З. Усилительные устройства на транзисторах.

6.    Лекции по дисциплине “Методы анализа и расчёта электронных схем”.