

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Воронежский государственный технический университет»**

**МОДЕЛИРОВАНИЕ МИКРОЭЛЕКТРОННЫХ ПРОЦЕССОВ  
И ПРИБОРОВ НА ПК**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**к выполнению лабораторных и практических работ  
по дисциплине «Информационные технологии в электронике»  
для студентов направления  
11.03.04 «Электроника и наноэлектроника»  
всех форм обучения**

**Воронеж 2024**

УДК 621.38(07)  
ББК 32.85я7

**Составитель**

*канд. техн. наук Плотникова Е. Ю.*

**Моделирование микроэлектронных процессов и приборов на ПК:** методические указания к выполнению лабораторных и практических работ по дисциплине «Информационные технологии в электронике» для студентов направления 11.03.04 «Электроника и наноэлектроника» всех форм обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост. Е. Ю. Плотникова. – Воронеж: Изд-во ВГТУ, 2024. – 57 с.

В методических указаниях рассматриваются методы моделирования основных технологических процессов на полупроводниковом производстве. Изучаются диффузия, ионная имплантация, кольцевой испаритель. Рассматривается расчёт биполярного транзистора с использованием двухстадийной диффузии. Для каждой работы приведен небольшой блок теории и пошаговый расчёт параметров технологического процесса. В качестве вспомогательной информации рассматриваются методы программирования на C# элементов кода, используемых при построении модели в виде оконного приложения.

Предназначены для студентов направления 11.03.04 «Электроника и наноэлектроника» всех профилей и форм обучения.

Методические указания подготовлены в электронном виде и содержатся в файле МУ\_ИТвЭ.pdf.

Табл. 4. Ил. 61. Библиогр.: 7 назв.

**УДК 621.38(07)**  
**ББК 32.85я7**

**Рецензент** – Т. В. Свистова канд. техн. наук, доц. кафедры  
твердотельной электроники ВГТУ

*Издается по решению редакционно-издательского совета  
Воронежского государственного технического университета*

## ВВЕДЕНИЕ

В завершающем блоке курса «Информационные технологии в электронной технике» изучаются методики моделирования физических процессов, входящих в технологическую карту разработки стандартного микро- и нанозлектронного прибора.

На каждой лабораторной и соответствующей ей практической работе изучается порядок построения характеристик различных технологических процессов в стандартном математическом пакете (например, MathCad) и в свободно распространяемой среде MS Visual Studio Community.

Рассматриваются технологии:

- термического окисления кремния,
- кольцевого испарителя,
- диффузии (из ограниченного и бесконечного источника),
- построения биполярного транзистора методом диффузии,
- ионной имплантации,
- управления двумерным распределением ионов под краем маски.

В качестве дополнительной информации приводятся подходы к расчёту функции ошибок на языке программирования высокого уровня С#, методики построения графиков на С# и применяемый для расчёта интегралов различных сложных функций метод Симпсона.

### **Задание на практические работы**

В каждой практической работе студент согласно информации из соответствующего раздела строит графики и проводит расчёты в математическом пакете, получая результат, аналогичный приведенному в методических указаниях.

Меняя параметры из условия работы в физически возможных пределах (информация берется из анализа справочной литературы по технологическим процессам), студент анализирует изменение технологического процесса.

По итогам работы оформляется отчёт по правилам оформления студенческих работ, принятым в ВГТУ.

### **Задание на лабораторные работы**

В каждой лабораторной работе студент разрабатывает код приложения WindowsForms на языке программирования высокого уровня С#, аналогичный задаче, решенной на соответствующей практической работе.

Вводимые числовые значения переменных задаются с использованием инструмента «*textBox*» таким образом, чтобы изменения в программе, созданной на практической работе, и изменения, внесенные в разработанное приложение С#, давали при расчёте аналогичные результаты и графики.

По итогам работы информация о проекте, разработанном на С#, вносится в соответствующий отчёт практической работы. Добавляются коды приложений и иллюстрации работы программы. В комментариях описывается процесс программирования, внесенные изменения, работоспособность и функционал кода.

## МЕТОДЫ ПОСТРОЕНИЯ СТАНДАРТНЫХ ГРАФИКОВ ФУНКЦИЙ

Код любого приложения типа WindowsForms, разрабатываемого на языке программирования высокого уровня C#, содержит библиотеки, наборы переменных, методы для расчёта функций (формулы), блоки инициализации и загрузки компонента Формы, блок активации расчёта.

```
using System;
using System.Windows.Forms;

namespace Sin_x_графики_и_таблицы
{
    public partial class Form1 : Form
    {
        // задаются первая точка графика и последняя точка графика будущей функции
        double x0;
        double x1;
        // переменная для расчёта
        double x;
        // шаг функции
        double n;

        // метод, возвращающий расчёт функции sin(x), задан для удобства простой заглавной буквой
        F(функция)
        double F(double x)
        {
            return Math.Sin(x);
        }

        // метод, возвращающий расчёт функции tg(x), задан для удобства простой заглавной буквой
        G(функция)
        double G(double x)
        {
            return Math.Tan(x);
        }

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // общие настройки областей ввода начального и конечного значения
            textBox1.Text = (0.0).ToString("0.###E+0");
            textBox2.Text = (3.0 * Math.PI).ToString("0.###E+0");
            textBox3.Text = (0.1).ToString("0.#");
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // очистка графика от предыдущего расчёта
            chart1.Series[0].Points.Clear();
            chart1.Series[1].Points.Clear();

            // настройка формата осей с заданным количеством нулей после запятой
            chart1.ChartAreas[0].AxisX.LabelStyle.Format = "0.E+0";
            chart1.ChartAreas[0].AxisY.LabelStyle.Format = "0.E+0";

            // получение начального и конечного значений, вводимых в textBox-ы в окне формы
            x0 = Double.Parse(textBox1.Text);
            x1 = Double.Parse(textBox2.Text);
        }
    }
}
```

```

n = Double.Parse(textBox3.Text);
// в цикле рассчитываются функции, заданные выше в методах, от x1 до x2 с шагом n
for (x = x0; x <= x1; x += n)
{
    double Y = F(x);
    double Z = G(x);
    chart1.Series[0].Points.AddXY(x, Y);
    chart1.Series[1].Points.AddXY(x, Z);
}
}
}
}

```

На рис. 1 приведен результат выполнения кода по оси  $x$  в диапазоне от 0 до  $3 \cdot \pi$  с шагом 0,1. Строятся графики функций  $\sin(x)$  и  $\operatorname{tg}(x)$ .

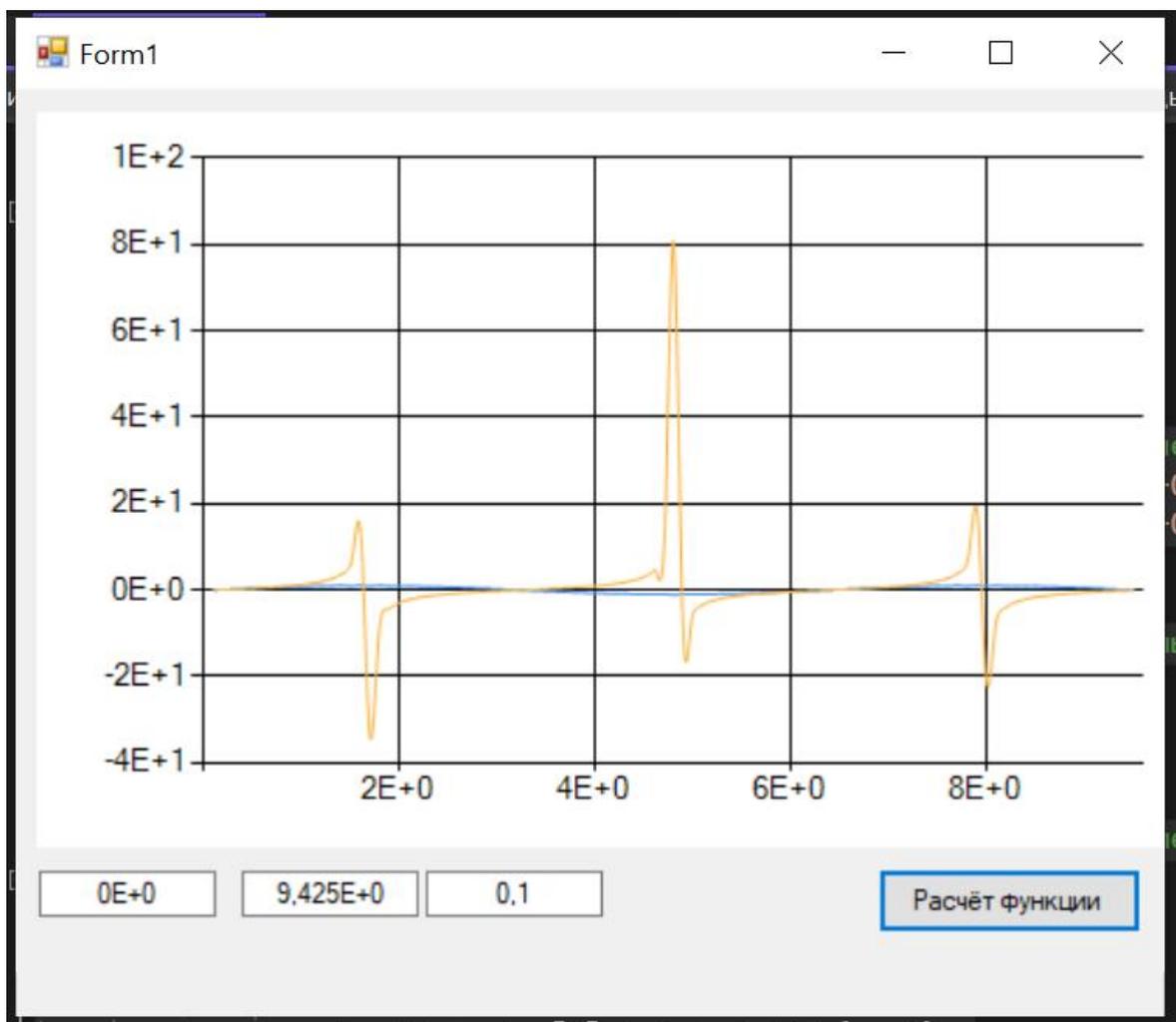


Рис. 1. Пример работы программы для расчёта и построения графика функций  $\sin(x)$  и  $\operatorname{tg}(x)$

## МОДЕЛИРОВАНИЕ ФУНКЦИИ ОШИБОК Erf(x)

Функция ошибок (интеграл вероятности) – неэлементарная функция, используемая в теории вероятностей, математической статистике и дифференциальных уравнениях в частных производных (1)

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \quad (1)$$

Дополнительная функция ошибок определяется через функцию ошибок (2)

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt. \quad (2)$$

Функцию ошибок можно записать в ряд Тейлора, интегрируя её почленно (3)

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n!(2n+1)} = \frac{2}{\sqrt{\pi}} \left( x - \frac{x^3}{3} + \frac{x^5}{10} - \frac{x^7}{42} + \frac{x^9}{216} - \dots \right). \quad (3)$$

Эту формулу можно переписать как (4)

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \left( x \prod_{i=1}^n \frac{-(2i-1)x^2}{i(2i+1)} \right) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{x}{2n+1} \prod_{i=1}^n \frac{-x^2}{i}. \quad (4)$$

При использовании функции ошибок на разных языках программирования существуют несколько возможных путей решения задачи:

1) использование встроенной функции  $\operatorname{erf}(x)$  – для C# не подходит, так как в данный язык функция ошибок не интегрирована;

2) расчёт функции ошибок методами интегрирования – метод наиболее удобен для набора положительных значений переменной  $x$  (код приводится в виде реализации метода `double Erf(double x) { }`; в тексте ниже);

3) расчёт функции ошибок с использованием набора констант – метод позволяет рассчитывать  $\operatorname{erf}(x)$  для любых отрицательных и положительных значений  $x$  (код реализован ниже в блоке метода `double otr_Erf(double x) { }`);

```
using System;
using System.Windows.Forms;

namespace Erf_x
{
    public partial class Form1 : Form
    {
        //переменные, используемые для расчёта, задаются глобально в классе
        double x0, x1, x2;

        //метод, использующий функцию для расчёта экспоненты от числа или выражения; экспонента
        //используется для расчёта функции ошибок – формула (1)
        double Expra(double x)
        {
            return Math.Exp(-x * x);
        }

        // расчёт функции ошибок одним из методов, который стабильно работает для области
        //положительных значений x
        double Erf(double x)
        {
            // начало интервала отсчёта
            double a = 0;
```

```

// количество делений интервала - отрезки
double n = 1000;
// шаг в интервале
double h = (x - a) / n;
double s = 0;
// следующий шаг - новое приближение
double x_sled = a + h;
// шаг интегрирования и сам интеграл определяются как сумма отрезков
while (x_sled < x)
{
    s = s + 4 * Expa(x_sled);
    x_sled = x_sled + h;
    s = s + 2 * Expa(x_sled);
    x_sled = x_sled + h;
}
s = h / 3 * (s + Expa(a) - Expa(x));
// определение интеграла функции ошибок - формула (1)
return s *(2 / Math.Sqrt(Math.PI));
}

// для области отрицательных значений предлагается использовать задание функции ошибок через
набор констант
double otr_Erf(double x)
{
    // константы
    double a1 = 0.254829592;
    double a2 = -0.284496736;
    double a3 = 1.421413741;
    double a4 = -1.453152027;
    double a5 = 1.061405429;
    double p = 0.3275911;

    // степень - знак «минус» для отрицательной области x
    int sign = 1;
    if (x < 0)
        sign = -1;
    x = Math.Abs(x);

    // подстановка констант в расчёт для обеспечения сходимости функции
    double t = 1.0 / (1.0 + p * x);
    double y = 1.0 - (((a5 * t + a4) * t) + a3) * t + a2) * t + a1) * t * Math.Exp(-x * x);

    return sign * y;
}

public Form1()
{
    InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{
    // в указанные текстовые области вносятся переменные, для которых будут рассчитываться функции
    ошибок,
    x0 = double.Parse(textBox1.Text);
    x1 = double.Parse(textBox2.Text);

    // очистка графика и настройка формата осей
    chart1.Series[0].Points.Clear();
    chart1.ChartAreas[0].AxisX.LabelStyle.Format = "0.0";
    chart1.ChartAreas[0].AxisY.LabelStyle.Format = "0.0";

    // предварительная очистка таблицы
    dataGridView1.Rows.Clear();
}

```

```

// заполнение графика и таблицы расчётными значениями для отрицательной и положительной
// областей координат
for (double x = x0; x <= x1; x += 0.01)
{
    if (x0 < 0)
    {
        double Y = otr_Erf(x);
        chart1.Series[0].Points.AddXY(x, Y);
        dataGridView1.Rows.Add(x.ToString("0.00"), Y.ToString("0.000"));
    }
    else {
        double Y = Erf(x);
        chart1.Series[0].Points.AddXY(x, Y);
        dataGridView1.Rows.Add(x.ToString("0.00"), Y.ToString("0.000"));
    }
}
}
// расчёт функции ошибок для указанного числа
private void button2_Click(object sender, EventArgs e)
{
    x2 = double.Parse(textBox3.Text);
    label1.Text = otr_Erf(x2).ToString("0.#####");
}
private void Form1_Load(object sender, EventArgs e)
{
// запись начальных значений в текстовые поля для формирования шаблона расчёта
textBox1.Text = (-5.0).ToString("0.00");
textBox2.Text = (5.0).ToString("0.00");
textBox3.Text = (-0.5).ToString("0.00");
}
}
}

```

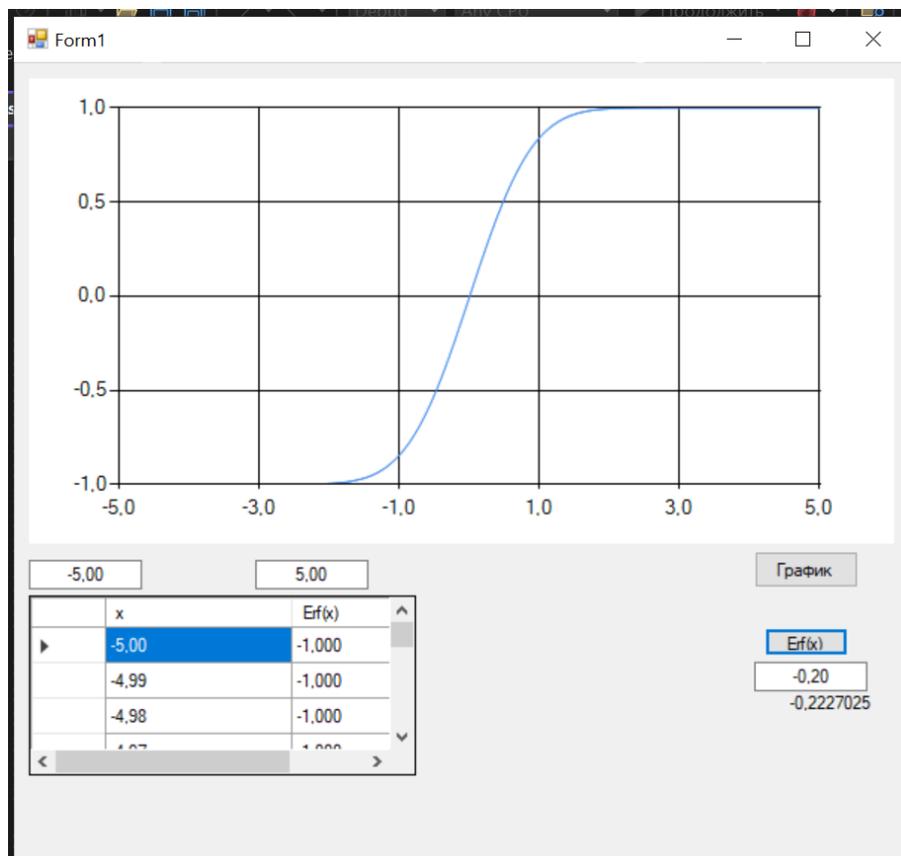


Рис. 2. График и таблица  $erf(x)$

## ИСПОЛЬЗОВАНИЕ МЕТОДА СИМПСОНА ДЛЯ РЕШЕНИЯ ИНТЕГРАЛОВ

Для определения численного значения определенного интеграла используется аппроксимация функции  $f(x)$  квадратичным интерполяционным полиномом, который совпадает с  $f(x)$  в крайних точках  $a$  и  $b$ , а также в средней точке  $\frac{a+b}{2}$ . Интеграл от этого квадратичного полинома выражается формулой (5)

$$I(f) \approx S(f) = \frac{b-a}{6} \left( f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right), \quad (5)$$

которая называется формулой Симпсона.

В методе Симпсона (рис. 3) площадь криволинейной трапеции рассчитывается как сумма площадей ряда криволинейных трапеций, у которых криволинейная сторона — это участок параболы.

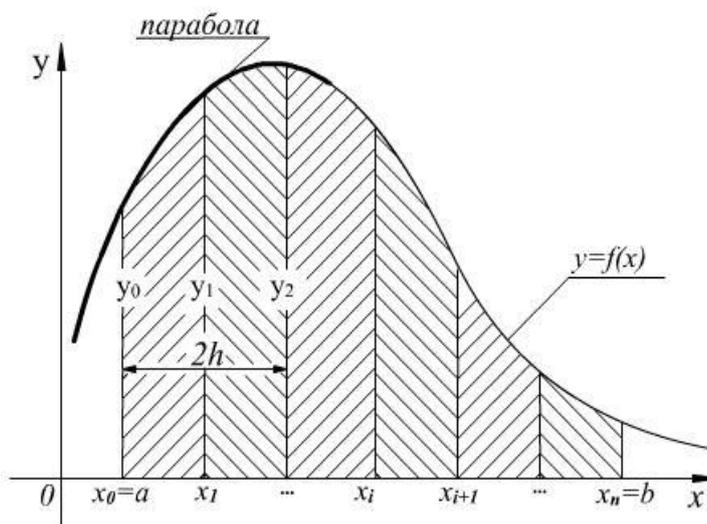


Рис. 3. Метод Симпсона

Каждая парабола может быть проведена только через три граничные точки, принадлежащие двум соседним отрезкам. Тогда число участков разбиения отрезка  $[a, b]$  обязательно будет четным. Таким образом, вместо каждых двух элементарных прямолинейных трапеций рассматривается одна элементарная трапеция, ограниченная параболической дугой. Исходя из этого, определенный интеграл на случай разбиения интервала на  $n$  участков с шагом  $h$  приближенно вычисляется по формуле (6) – полной формуле Симпсона

$$I(f) = \frac{h}{3} \left( f(a) + f(b) + 4 \cdot \sum_{i=1}^n f(x_{2i-1}) + 2 \cdot \sum_{i=1}^n f(x_{2i}) \right),$$

$$h = \frac{b-a}{2n}, \quad x_0 = a, \quad x_i = x_{i-1} + h, \quad i = 1, 2, \dots, 2n. \quad (6)$$

Таким образом, для реализации метода Симпсона для вычисления определенного интеграла необходимо задать в явном виде определенный интеграл, площадь которого необходимо определить. После этого задаются

пределы интегрирования, и шаг интегрирования. Затем проводится расчет по формуле (6).

Для метода Симпсона число разбиений  $n$  должно быть **четным**, что подлежит проверке при составлении программы.

```
using System;
using System.Windows.Forms;

namespace Решение_интегралов_методом_Симпсона
{
    public partial class Form1 : Form
    {
        // x0, x1 - начальный и конечный пределы интегрирования
        // x - переменная для расчёта, Y - переменная для вывода результата
        static double x0 = 0, x1 = 2*Math.PI, x, Y;
        // количество отрезков интегрирования - четное
        static double n = 200;
        // коэффициент для расчёта методом Симпсона - шаг функции
        static double h;

        // рассчитываемая функция - синус от числа
        static double F(double x)
        {
            return Math.Sin(x);
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            label1.Text = "Нижний предел интегрирования =";
            label2.Text = "Верхний предел интегрирования =";
            label3.Text = "Количество отрезков интегрирования =";
            textBox1.Text = x0.ToString("0.###");
            textBox2.Text = x1.ToString("0.###");
            textBox3.Text = n.ToString("0.###");
            label4.Text = "Результат интегрирования:";
            button1.Text = "Рассчитать интеграл!";
        }

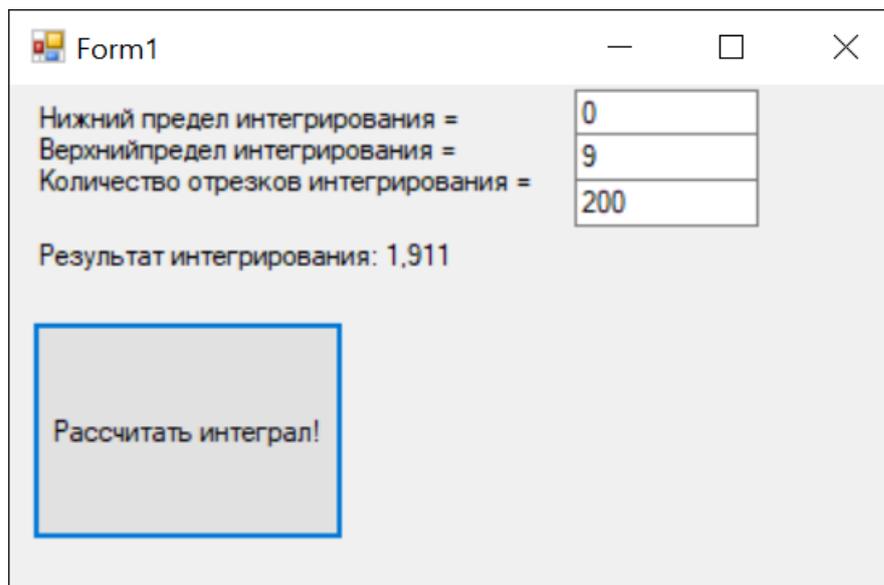
        private void button1_Click(object sender, EventArgs e)
        {
            x0 = Double.Parse(textBox1.Text);
            x1 = Double.Parse(textBox2.Text);
            n = Double.Parse(textBox3.Text);

            h = (x1 - x0) / n;
            Y = 0;
            x = x0 + h;
            while (x < x1)
            {
                Y = Y + 4 * F(x);
                x = x + h;
                Y = Y + 2 * F(x);
                x = x + h;
            }

            Y = h / 3 * (Y + F(x0) - F(x1));
            label4.Text = "Результат интегрирования: " + Y.ToString("0.###");
        }

        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

На рис. 4 приведены расположение модулей разрабатываемой программы и пример расчёта интеграла из кода на языке программирования С# и в математическом пакете.



$$f(x) := \sin(x)$$

$$\int_0^9 f(x) dx = 1.911$$

Рис. 4. Расчёт интеграла функции  $\sin(x)$  в диапазоне от 0 до 9 в программе на С# и в математическом пакете

## ТЕХНОЛОГИЯ ПОСТРОЕНИЯ ГРАФИКОВ ЧЕРЕЗ КОД

Большинство языков программирования, в том числе и С#, имеет возможность реализации проектов с построением графических структур различными инструментами и методами. Разработчики используют встроенные инструменты (*chart*), реализацию напрямую в форме без дополнительных элементов с использованием блока *drawing*, построение с помощью линий и точек на консоли или интегрируют в программу сторонние модули.

Инструмент *chart* зарекомендовал себя как наиболее удобный и оптимизированный блок для построения графиков на С#. С его помощью можно строить различные виды диаграмм, в том числе и для моделирования технологических процессов, применяемых при разработке дискретных полупроводниковых приборов и интегральных схем.

Инструмент *chart* достаточно гибок и позволяет настраивать параметры графика как через окно свойств (*properties*), так и через написание команд в коде проекта.

Рассмотрим пример построения графика функции  $\sin(x)$  без использования кода свойств, через код.

```
// используемые в проекте библиотеки
using System;
using System.Windows.Forms;
using System.Drawing;

namespace WindowsForms_Charting
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // элемент «график» задается не из панели элементов, а с помощью блока кода инициализации
            // здесь же определяются расположение графика, размер, рассчитываемая функция
            var chart = new Chart { Parent = this, Dock = DockStyle.Fill };
            chart.InterpolationStep = 0.2f;
            chart.MinX = -10f;
            chart.MaxX = 10f;
            chart.Function = (x) => Math.Sin(x);
        }

        // создаются переменные, в которые далее будут задаваться значения осей графика и
        // рассчитываемая функция
        public class Chart : Control
        {
            public int Indent { get; set; }

            public float MaxX { get; set; }
            public float MinX { get; set; }
            public float MaxY { get; set; }
            public float MinY { get; set; }

            public float GridStep { get; set; }
            public float InterpolationStep { get; set; }

            public Func<double, double> Function;

            public Chart()
            {
                SetStyle(ControlStyles.AllPaintingInWmPaint | ControlStyles.OptimizedDoubleBuffer |
                ControlStyles.ResizeRedraw | ControlStyles.UserPaint, true);
                Indent = 20;
                MinX = -5;
                MaxX = 5;
                MinY = -5;
                MaxY = 5;
                GridStep = 1;
                InterpolationStep = 0.01f;
            }

            // прямоугольная область графика
            private Rectangle ChartArea
            {
                get
                {
                    var rect = ClientRectangle;
                    rect.Inflate(-Indent, -Indent);
                    return rect;
                }
            }
        }
    }
}
```

```

float YToPixels(float y)
{
    return ChartArea.Height * (y - MinY) / (MaxY - MinY);
}
float XToPixels(float x)
{
    return ChartArea.Width * (x - MinX) / (MaxX - MinX);
}

protected override void OnPaint(PaintEventArgs e)
{
    var rect = ChartArea;
    var gr = e.Graphics;

    gr.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.HighQuality;

    var center = new PointF(rect.Left + XToPixels(0), rect.Bottom - YToPixels(0));

    using (var font = new Font(Font.FontFamily, 8f))
    using (var pen = new Pen(Color.FromArgb(50, Color.Navy), 1))
    {
        for (var x = MinX; x <= MaxX; x += GridStep)
        {
            var absX = rect.Left + XToPixels(x);
            gr.DrawLine(pen, absX, rect.Bottom, absX, rect.Top);
            gr.DrawString(x.ToString("0.0"), font, Brushes.Navy, absX - 7, center.Y + 5);
        }

        for (var y = MinY; y <= MaxY; y += GridStep)
        {
            var absY = rect.Bottom - YToPixels(y);
            gr.DrawLine(pen, rect.Left, absY, rect.Right + 5, absY);
            gr.DrawString(y.ToString("0.0"), font, Brushes.Navy, center.X - 25, absY - 5);
        }
    }
    using (var pen = new Pen(Color.Navy, 1))
    {
        pen.EndCap = System.Drawing.Drawing2D.LineCap.ArrowAnchor;
        gr.DrawLine(pen, center.X, rect.Bottom, center.X, rect.Top - 10);
        gr.DrawLine(pen, rect.Left, center.Y, rect.Right + 7, center.Y);
    }

    if (Function != null)
        using (var pen = new Pen(Color.Red, 2))
            for (var x = MinX; x < MaxX; x += InterpolationStep)
                {
                    var y1 = (float)Function(x);
                    var y2 = (float)Function(x + InterpolationStep);
                    e.Graphics.DrawLine(pen, rect.Left + XToPixels(x), rect.Bottom - YToPixels(y1), rect.Left +
XToPixels(x + InterpolationStep), rect.Bottom - YToPixels(y2));
                }
    }
}
}
}
}

```

На рис. 5 приведен результат реализации кода для функции  $\sin(x)$ .

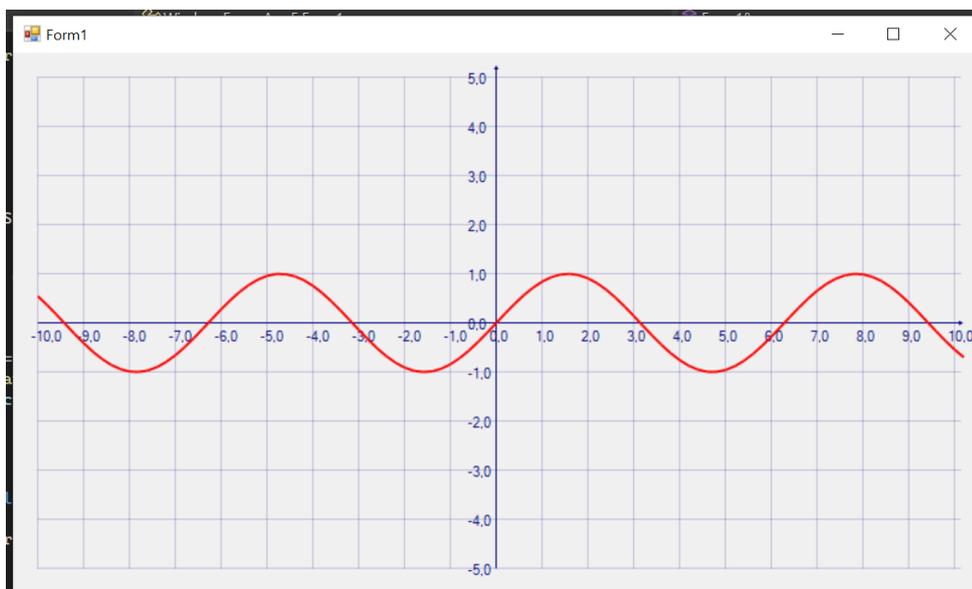


Рис. 5. График функции  $\sin(x)$  с настройками шрифтов и осей, с добавлением стрелок на осях, реализованных через команды в коде

## ЗАДАНИЕ 1. МОДЕЛИРОВАНИЕ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА ТЕРМИЧЕСКОГО ОКИСЛЕНИЯ КРЕМНИЯ

Оксидные пленки на поверхности  $Si$  используют для:

- пассивации поверхности,
- создания маскирующих покрытий в литографических процессах,
- выращивания слоя подзатворного диэлектрика в МОП-технологии.

Один из основных способов выращивания  $SiO_2$  пленок – термическое окисление (в сухом кислороде, во влажном кислороде или в парах воды).

Химическая реакция  $Si + O_2 \rightarrow SiO_2$  или  $Si + 2 H_2O \rightarrow SiO_2 + 2 H_2$  протекает на границе раздела  $Si - SiO_2$ , то есть молекулы кислорода сначала диффундируют через растущий слой окисла к поверхности кремния, и только там вступают в реакцию ( $t_{оксил} = 800 - 1300 \text{ } ^\circ C$ , определяет скорость роста пленки окисла).

Кинетика процесса окисления рассчитывается с помощью модели Дила-Гроува.

На рис. 6 представлен расчет стандартного технологического процесса термического окисления кремния с указанием основных физических величин, номиналов и единиц измерения.

В расчёте используются следующие обозначения:

$k$  – постоянная Больцмана (эВ/К),

$T$  – температура в реакторе (К),

$z_i$  – существующий на поверхности самородный слой окисла (мкм), при сухом окислении может быть принята равной 0, при влажном 20 нм,

$E1, E2$  – эмпирические полученные величины энергии активации для окисления в сухом кислороде для линейного и параболического участков, соответственно (эВ),

$C1, C2$  – эмпирически полученные константы скорости линейного и параболического участков ( $\text{мкм}^2/\text{ч}$ ).

Параметры  $A, B$  и  $\tau$  используются для упрощения расчётной формулы.

На начальном этапе роста (малое время окисления, кислород легко диффундирует сквозь слой окисла) толщина растущей оксидной пленки увеличивается линейно (7)

$$t + \tau \ll \frac{A^2}{4B}, \quad z_0 \approx \frac{B}{A}(t + \tau). \quad (7)$$

При больших временах окисления возникает зависимость в виде квадратного корня (8)

$$t + \tau \gg \frac{A^2}{4B}, \quad z_0 \approx \sqrt{B(t + \tau)}. \quad (8)$$

$$k := 8.617 \cdot 10^{-5} \frac{\text{эВ}}{\text{К}} \quad T := 1473 \quad z_i := 20 \cdot 10^{-3} \text{ мкм}$$

$$E1 := 0.71 \text{ эВ} \quad E2 := 1.97 \text{ эВ}$$

$$C1 := 2.24 \cdot 10^2 \frac{\text{мкм}^2}{\text{ч}} \quad C2 := 8.95 \cdot 10^7 \frac{\text{мкм}^2}{\text{ч}}$$

$$B := C1 \cdot e^{\frac{-E1}{k \cdot T}} = 0.834 \quad A := \frac{B}{\frac{E2}{C2 \cdot e^{\frac{-E2}{k \cdot T}}}} = 0.051 \quad \tau := \frac{z_i^2 + A \cdot z_i}{B}$$

$$z_0(t) := -\frac{A}{2} + \sqrt{\frac{A^2}{4} + B \cdot (t + \tau)} \quad z_0(5) = 2.016$$

Рис. 6. Исходные параметры и расчётная формула для моделирования процесса окисления

На рис. 7 приводится график зависимости толщины оксидной пленки  $z_0$  от прошедшего времени  $t$ .

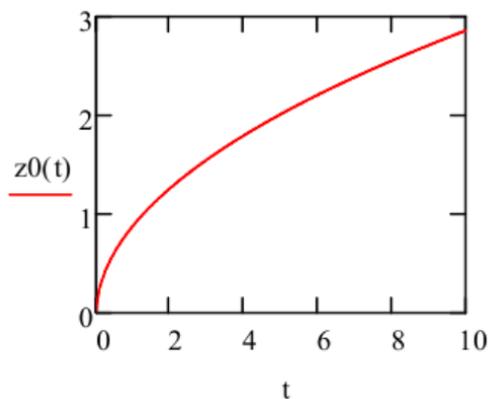


Рис. 7. График зависимости толщины оксидной пленки  $z_0$  от прошедшего времени  $t$

## ЗАДАНИЕ 2. РАСЧЁТ ПАРАМЕТРОВ РАСПЫЛЕНИЯ С ИСПОЛЬЗОВАНИЕМ КОЛЬЦЕВОГО ИСПАРИТЕЛЯ (МАГНЕТРОННОЕ НАПЫЛЕНИЕ)

Магнетронное распыление – это один из основных методов планарной технологии, используемых для нанесения тонких пленок; является разновидностью метода ионной имплантации. Распыление происходит при бомбардировке поверхности мишени ионами рабочего газа.

На рис. 8 приведены постоянные и изменяемые величины, используемые при моделировании параметров кольцевого испарителя.

|                                |                    |                   |
|--------------------------------|--------------------|-------------------|
| Атомный номер материала мишени | $Z_a := 14$        |                   |
| Атомный вес материала мишени   | $M_a := 28$        | г/моль            |
| Плотность материала мишени     | $\rho := 2.42$     | г/см <sup>3</sup> |
| Энергия сублимации             | $E_s := 3.91$      | эВ                |
| Безразмерный параметр          | $\alpha_s := 0.21$ |                   |
| Внутренний радиус эрозии       | $R_1 := 4.5$       | см                |
| Внешний радиус эрозии          | $R_2 := 5$         | см                |
| Тока разряда                   | $I := 5$           | А                 |
| Время распыления               | $t := 1800$        | с                 |
| Радиус пластины                | $r_0 := 5$         | см                |

Рис. 8. Исходные данные для расчёта кольцевого испарителя

Зависимость угла отклонения потока ионов от расстояния «мишень-подложка» (рис. 9) строится из соотношения двух векторов ( $z$  и  $\alpha$ ) в математическом пакете; в коде программы на С#  $z$  и  $\alpha$  задаются как два одномерных массива. К данному графику сглаживание не применяется.

$$z := \begin{pmatrix} 0.01 \\ 0.5 \\ 1.0 \\ 5.0 \\ 10.0 \end{pmatrix} \quad \alpha := \begin{pmatrix} 0.17 \\ 0.20 \\ 0.23 \\ 0.98 \\ 5.0 \end{pmatrix}$$

Рис. 9. Векторы  $z$  и  $\alpha$ , задаваемые в математическом пакете

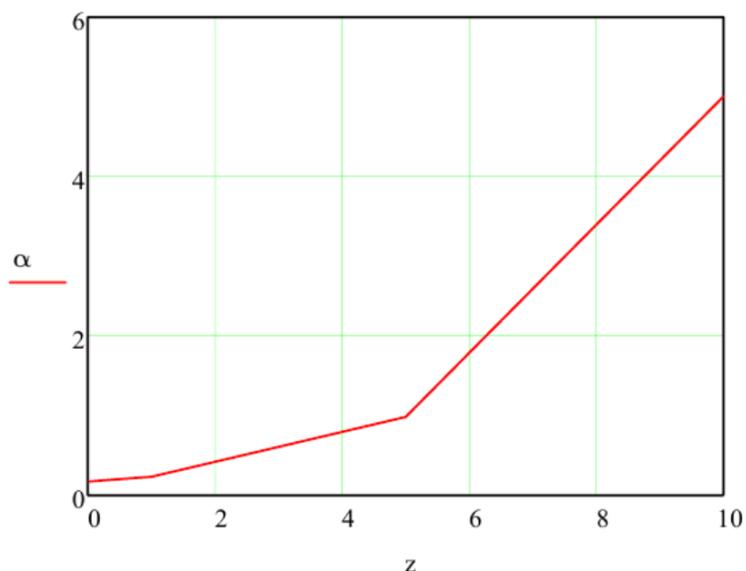


Рис. 10. График зависимости угла распыления  $\alpha$  от расстояния «мишень-подложка»  $z$

В табл. 1 приведены некоторые параметры материалов, используемых в методе магнетронного распыления.

Таблица 1

Справочные данные по используемым материалам

| Материал | Атомный номер $Z$ | Атомный вес $M$ , г/моль | Энергия сублимации, $E_s$ эВ | Плотность, г/см <sup>3</sup> |
|----------|-------------------|--------------------------|------------------------------|------------------------------|
| Be       | 4                 | 9                        | 3,48                         | 1,85                         |
| Mg       | 12                | 24,3                     | 1,59                         | 1,76                         |
| Al       | 13                | 27                       | 3,26                         | 2,7                          |
| Si       | 14                | 28                       | 3,91                         | 2,42                         |
| Ti       | 22                | 47,9                     | 4,34                         | 4,52                         |
| V        | 23                | 51                       | 3,7                          | 6,11                         |
| Cr       | 24                | 52                       | 3,68                         | 7,19                         |
| Mn       | 25                | 54,9                     | 3,15                         | 7,44                         |
| Fe       | 26                | 55,9                     | 4,15                         | 7,87                         |
| Ni       | 28                | 58,7                     | 4,41                         | 8,9                          |
| Cu       | 29                | 63,5                     | 3,56                         | 8,96                         |
| Ge       | 32                | 72,6                     | 3,77                         | 5,32                         |
| Nb       | 41                | 92,9                     | 7,5                          | 8,57                         |
| Mo       | 42                | 96                       | 6,9                          | 10,2                         |
| Ag       | 47                | 107,8                    | 2,7                          | 10,5                         |
| Ta       | 73                | 181                      | 8,7                          | 16,6                         |
| W        | 74                | 183                      | 8,76                         | 19,3                         |
| Pt       | 78                | 195                      | 5,56                         | 21,5                         |
| Au       | 79                | 197                      | 3,92                         | 19,32                        |

На рис. 11 представлены данные для распыления с использованием в качестве рабочего газа аргона *Ar*. Здесь же вводятся некоторые константы, применяемые при моделировании (число Авогадро *Na* и заряд электрона *e*).

| Рабочий газ - аргон: |                             |         |
|----------------------|-----------------------------|---------|
| Атомный номер иона   | $Zu := 18$                  |         |
| Атомный вес иона     | $Mu := 40$                  | г/моль  |
| Энергия иона         | $Eu := 350$                 | эВ      |
| Число Авогадро       | $Na := 6.023 \cdot 10^{23}$ | ат/моль |
| Заряд электрона      | $e := 1.6 \cdot 10^{-19}$   | Кл      |

Рис. 11. Константы для моделирования параметров среды магнетронного напыления

Далее (рис. 11) рассчитываются:

- коэффициент распределения *S*, который характеризует эффективность распыления – среднее число атомов, удаляемых с поверхности твердого тела одной падающей частицей; рассчитывается по теории Зигмунда для аморфных и поликристаллических материалов на низких энергиях ионов (*Eu* ~ 1 кэВ);

- плотность ионного тока *J* в сечении (*s*), перпендикулярном направлению падения ионов;

- скорость распыления *Vp* при нормально падении ионов (расчет сходится с экспериментальными данными с высокой точностью).

Коэффициент распыления

$$S := \frac{3}{\pi^2} \cdot \alpha_s \cdot \frac{Mu \cdot Ma}{(Mu + Ma)^2} \cdot \frac{Eu}{2Ec} \quad S = 0.692$$

Плотность ионного тока

$$s := \pi \cdot (R2^2 - R1^2) \quad s = 14.923 \quad J := \frac{I}{s} \quad J = 0.335 \quad A/cm^2$$

Скорость распыления

$$Vp := \frac{J \cdot S \cdot Ma}{e \cdot Na \cdot \rho} \quad Vp = 2.784 \times 10^{-5} \frac{cm}{c}$$

Рис. 11. Расчёт скорости распыления

Задавая интервалы  $j$  и  $i$  для определения координат по осям  $H_j(r_i)$ , рассчитываем в виде матрицы точек  $M_{i,j}$  и строим как поверхность (рис. 12) зависимость толщины напыляемой пленки (см) от радиуса пластины и расстояния до мишени.

$$j := 1..16$$

$$i := 0..10$$

$$H_j := 1 \cdot j$$

$$r_i := -5 + 1 \cdot i$$

$$F(r, H) := 2 \cdot V_p \cdot t \cdot \int_{R1}^{R2} \frac{H^2 R}{\left[ (H^2 - r^2 + R^2)^2 + 4 \cdot r^2 \cdot H^2 \right]^{\frac{3}{2}}} dR \quad \text{CM}$$

$$M_{i,j} := F(r_i, H_j)$$

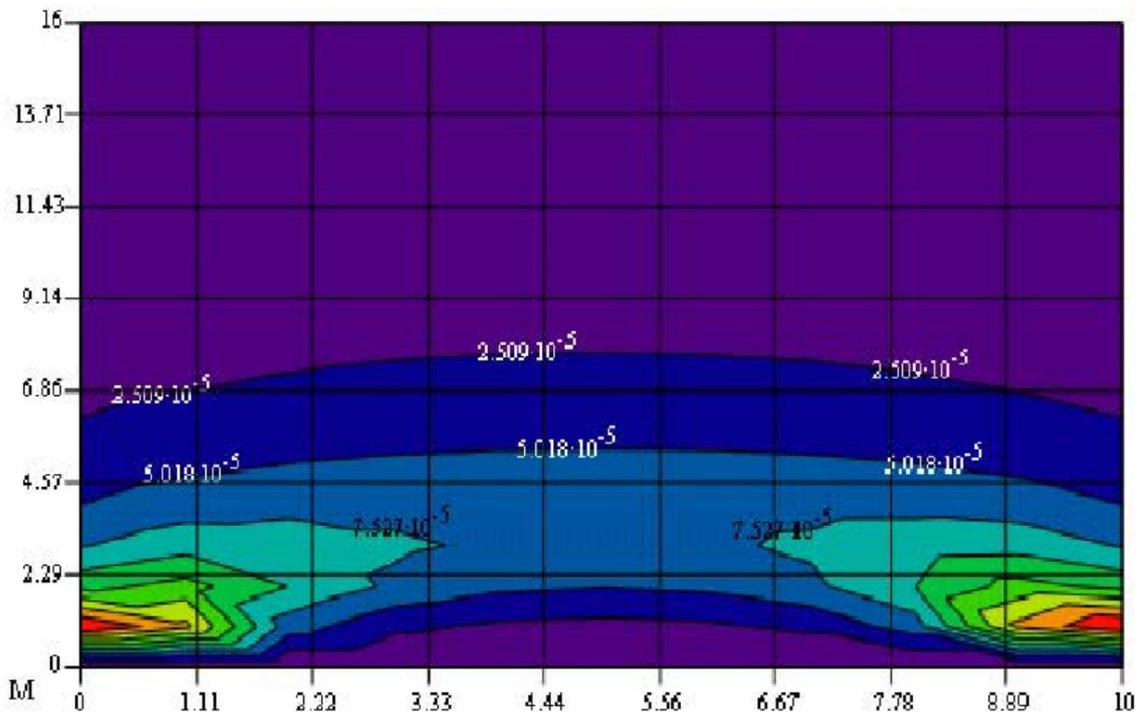


Рис. 12. Расчет зависимости толщины напыляемой пленки (см) от радиуса пластины и расстояния до мишени

Следующим этапом идёт анализ распределения толщины напыленной пленки (мкм) на заданном расстоянии ( $H = 3,15$  мкм) от мишени (рис. 13). Диапазон напыления задается как  $(-6 \dots +6)$  мкм. Для проверки толщина

рассчитывается в середине ( $h(0)$ ) и на половине расстояния ( $h(3)$ ) области напыления.

$$r := -6..6$$

$$H := 3.15 \quad h(r) := 10^4 \cdot 2 \cdot V_p \cdot t \cdot \int_{R1}^{R2} \frac{H^2 R}{\left[ (H^2 - r^2 + R^2)^2 + 4 \cdot r^2 \cdot H^2 \right]^{\frac{3}{2}}} dR \quad \text{МКМ}$$

$$h(3) = 0.863$$

$$h(0) = 0.692$$

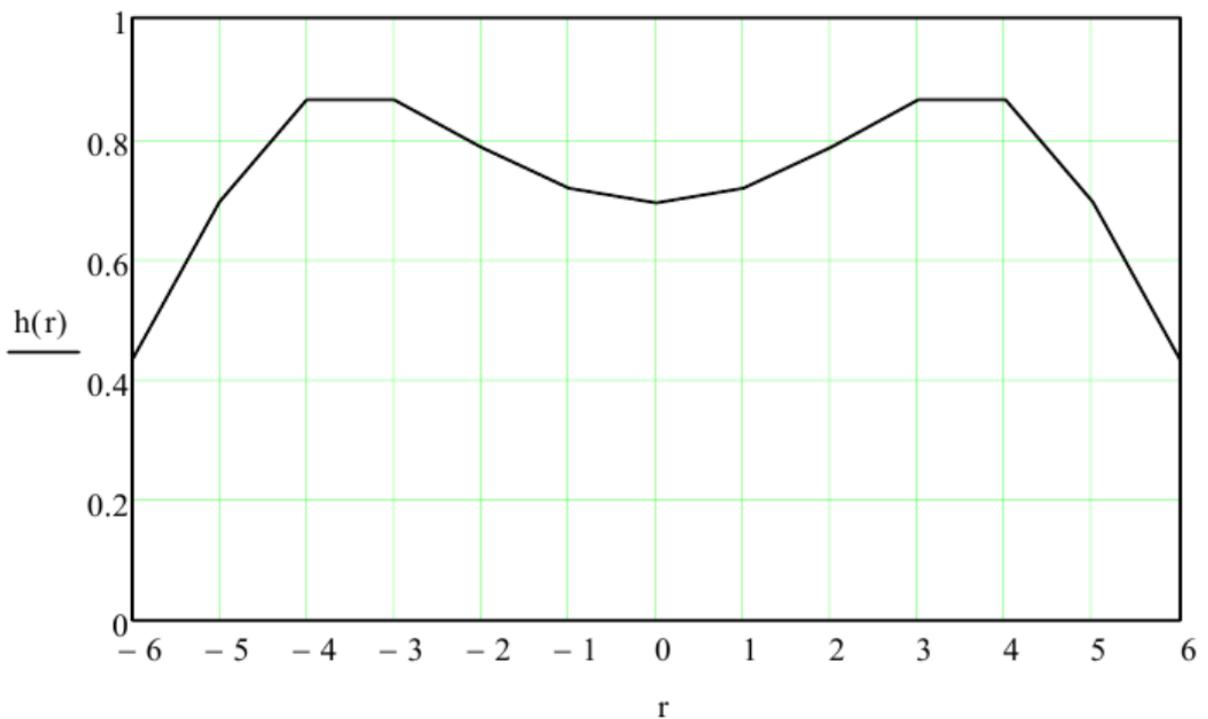


Рис. 13. Анализ распределения толщины напыленной пленки (мкм) на заданном расстоянии ( $H = 3,15$  мкм) от мишени

После расчёта графика распределения напыляемого материала по пластине можно проанализировать это распределение. Если задать максимальное и минимальное значения для толщины пленки ( $h_{max}$ ,  $h_{min}$ ), можно рассчитать среднюю толщину пленки ( $h_{ср}$ , мкм), неравномерность толщины пленки ( $\Delta h$ , мкм;  $\Delta$ , %) (рис. 14).

$$h_{\max} := 0.87$$

$$h_{\min} := 0.7$$

|                                     |  |
|-------------------------------------|--|
| Средняя толщина пленки, мкм         | $h_{\text{ср}} := \frac{h_{\max} + h_{\min}}{2}$     |
| Неравномерность толщины пленки, мкм | $\Delta h := \frac{h_{\max} - h_{\min}}{2}$          |
| Неравномерность толщины пленки, %   | $\Delta := \frac{\Delta h}{h_{\text{ср}}} \cdot 100$ |

Результаты расчета:

$$h_{\text{ср}} = 0.785$$

$$\Delta h = 0.085$$

$$\Delta = 10.828$$

$$\text{Толщина пленки} = h_{\text{ср}} \pm h$$

Рис. 14. Анализ распределения толщины напыленной на подложку пленки

### ЗАДАНИЕ 3. МЕТОДЫ МОДЕЛИРОВАНИЯ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА ДИФФУЗИИ

Диффузия – это перенос атомов примеси (вещества), возникающий из-за хаотического теплового движения атомов. Процесс возникает при наличии градиента концентрации примеси, который направлен в сторону уменьшения данной концентрации.

С помощью диффузии можно задать:

- тип проводимости (электронная или дырочная);
- концентрацию примеси в локальной области полупроводника.

Перемещение атомов примеси происходит по одному из механизмов:

- по вакансиям (дефектам кристаллической решетки) – после охлаждения вакансия исчезает, а примесный атом «фиксируется»;
- по междоузлиям – после охлаждения междоузельный атом может вернуться в узел, заместив основной атом и став электрически активным.

Активация процесса диффузии в зависимости от полупроводника и примеси происходит при температуре  $t \sim 1000$  °С. Атомы по кристаллической решетке перемещаются скачками, поэтому для расчёта рассматривается суммарный поток – статистически усредненный за определенный период времени.

Моделирование процесса диффузии рассматривается в два этапа – на постоянной температуре и с изменением температуры процесса.

#### Задача 1. Загонка и разгонка примеси

В начале моделирования в проект вводятся переменные и константы (рис. 15).

|   |                            |                    |
|---|----------------------------|--------------------|
| Энергия активации диффузии примеси                      | $E_a := 3.63$              | эВ                 |
| Температура диффузии                                    | $T1 := 273 + 1200$         | К                  |
| Время диффузии  | $t := 600$                 | с                  |
| Коэффициент диффузии при бесконечно большой температуре | $D_0 := 0.785$             | см <sup>2</sup> /с |
| Поверхностная концентрация диффузанта                   | $C_0 := 1 \cdot 10^{19}$   | 1/см <sup>3</sup>  |
| Концентрация примеси в исходном кремнии                 | $C_c := 10^{15}$           | 1/см <sup>3</sup>  |
| Постоянная Больцмана                                    | $k := 8.625 \cdot 10^{-5}$ | эВ/К               |

Рис. 15. Заданные переменные и постоянные величины

В табл. 2 приведены основные параметры наиболее распространенных примесей, используемых в кремниевой технологии. В зависимости от вводимой примеси меняются параметры моделирования.

Таблица 2

Основные акцепторные и донорные примеси для Si

| Акцепторы |                             |                  | Доноры  |                             |                  |
|-----------|-----------------------------|------------------|---------|-----------------------------|------------------|
| Элемент   | $D_0, \text{см}^2/\text{с}$ | $E_a, \text{эВ}$ | Элемент | $D_0, \text{см}^2/\text{с}$ | $E_a, \text{эВ}$ |
| B         | 5,1                         | 3,7              | P       | 3,85                        | 3,66             |
| Al        | 1,385                       | 3,39             | As      | 24                          | 4,08             |
| In        | 0,785                       | 3,63             | Sb      | 12,9                        | 3,98             |
| Ga        | 0,374                       | 3,41             | Bi      | 1,08                        | 3,85             |

Зная величину энергии активации примеси ( $E_a$ ) и выбрав температурный диапазон ( $T = 1000 \dots 1600$  К), можно рассчитать температурную зависимость коэффициента диффузии ( $D(T)$ ) (рис. 16). Здесь же рассчитывается коэффициент диффузии при заданной ранее температуре диффузии  $T1 = 1473$  К.

$$T := 1000 \dots 1600 \quad D(T) := D_0 \cdot \exp\left(-\frac{E_a}{k \cdot T}\right) \quad D(T1) = 3.063 \times 10^{-13}$$

Рис. 16. Расчёт температурной зависимости коэффициента диффузии

Сама зависимость  $D(T)$  приведена на рис. 17 и имеет логарифмический масштаб по оси коэффициента диффузии.

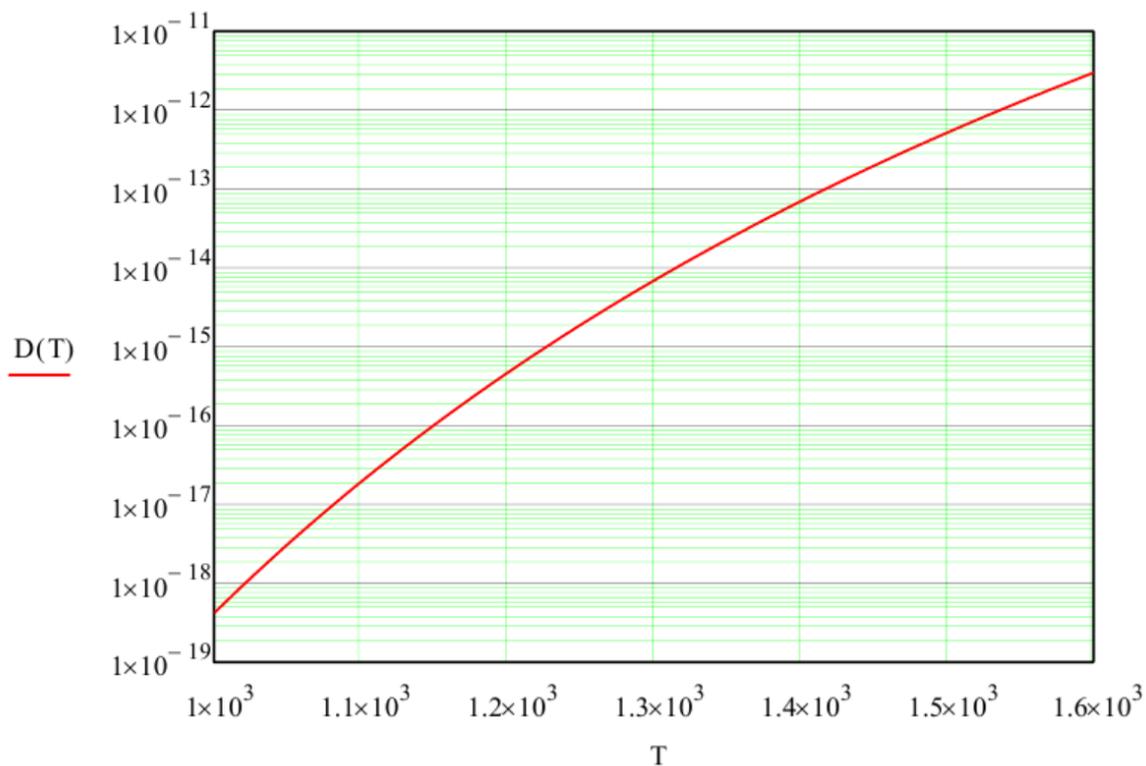


Рис. 17. Температурная зависимость коэффициента диффузии  $D(T)$

Для некоторых видов анализа зависимость коэффициента диффузии рассчитывается от обратной температуры (рис. 18).

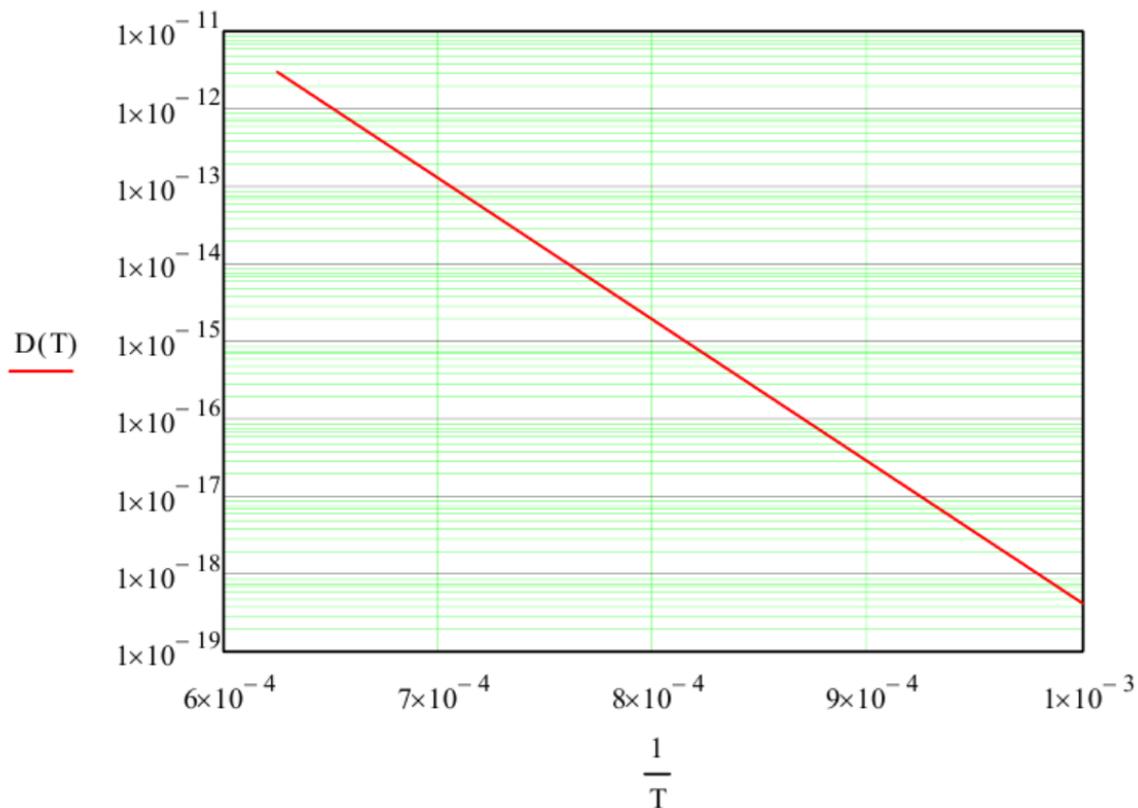


Рис. 18. Температурная зависимость коэффициента диффузии  $D(1/T)$

В стандартном технологическом процессе диффузия проводится в два этапа – из источника ограниченной мощности и из бесконечного источника.

В случае диффузии из источника **бесконечной** мощности на поверхности пластины сохраняется постоянная поверхностная концентрация вводимой примеси (рис. 19).

Для диффузии из бесконечного источника можно рассчитать глубину залегания *p-n* перехода (рис. 20) и графически или расчётно проверить концентрацию примеси в выбранной точке ( $C1(Z2)$ ).

Если диффузия происходит из источника **ограниченной** мощности (рис. 21), то в начальный момент процесса легирующая примесь, находящаяся на поверхности пластины, растворяется в тонком приповерхностном слое.

Для диффузии из источника ограниченной мощности можно рассчитать глубину залегания *p-n* перехода (рис. 22) и графически или расчётно проверить концентрацию примеси в выбранной точке ( $C1(Z2)$ ).

$$C1(z) := C_0 \cdot \left( 1 - \operatorname{erf} \left( \frac{z}{2 \cdot \sqrt{D(T1) \cdot t}} \right) \right)$$

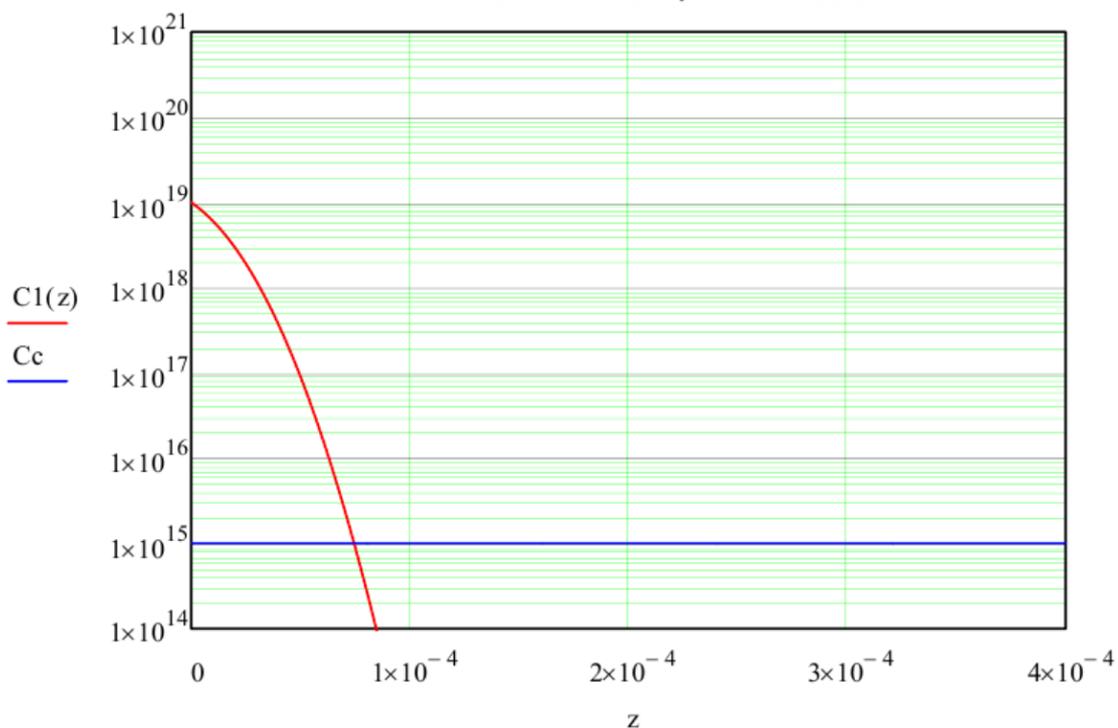


Рис. 19. Расчёт диффузии примеси из источника бесконечной мощности

$$f(z) := \frac{C_0}{C_c} \cdot \left( 1 - \operatorname{erf} \left( \frac{z}{2\sqrt{D(T1) \cdot t}} \right) \right) - 1$$

$$z := 0.2 \cdot 10^{-4} \quad Z_2 := \operatorname{root}(f(z), z)$$

$$Z_2 = 7.458 \times 10^{-5} \quad \text{CM}$$

Проверка

$$C1(Z_2) = 1 \times 10^{15} \quad 1/\text{CM}^3$$

Рис. 20. Расчёт глубины залегания  $p$ - $n$  перехода при диффузии из источника бесконечной мощности

$$C2(z) := C_0 \cdot \exp \left( -\frac{z^2}{4 \cdot D(T1) \cdot t} \right)$$

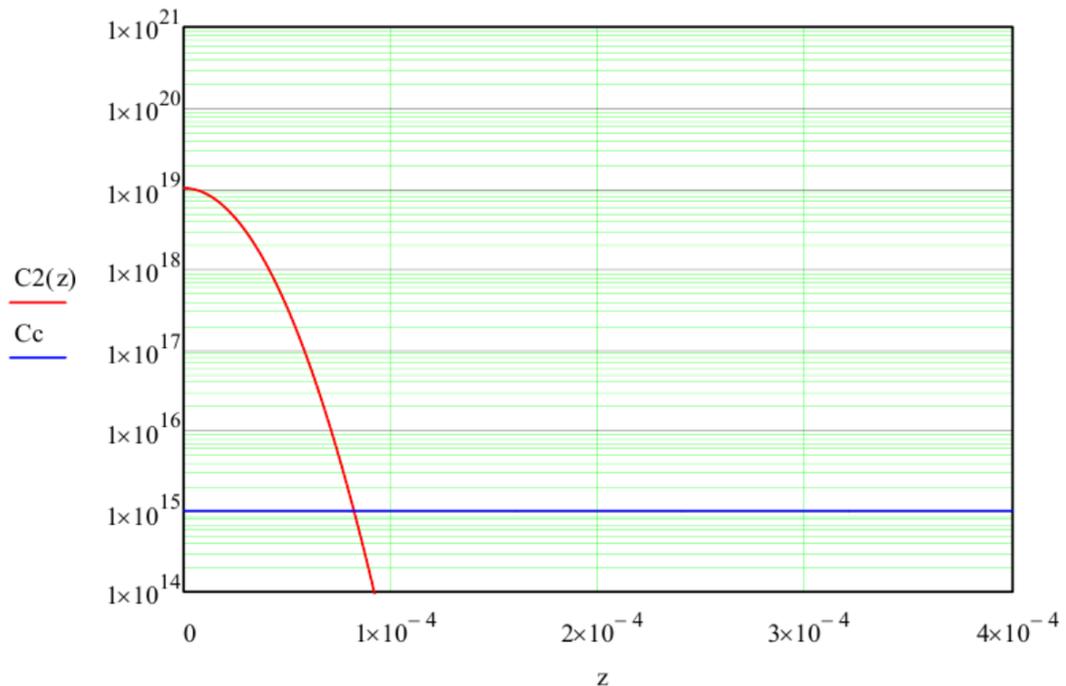


Рис. 21. Расчёт диффузии примеси из источника ограниченной мощности

$$Z1 := 2\sqrt{D(T1) \cdot t} \cdot \sqrt{\ln \left( \frac{C_0}{C_c} \right)} \quad Z1 = 8.228 \times 10^{-5}$$

Рис. 22. Расчёт глубины залегания  $p$ - $n$  перехода при диффузии из источника ограниченной мощности

В стандартном технологическом процессе используется технология двухстадийной диффузии, при которой сначала проводится загонка примеси из неограниченного источника, а затем – разгонка из узкой приповерхностной области (которая служит ограниченным источником примеси).

Рассчитаем распределение коэффициента диффузии  $D(T)$  для двухстадийного процесса. Условия и формула для расчёта приведены на рис. 23.

$$T := 1000 .. 1600$$

$$D(T) := D_0 \cdot \exp\left(-\frac{E_a}{k \cdot T}\right) \quad D(T1) = 3.063 \times 10^{-13}$$

Рис. 23. Расчёт коэффициента диффузии

Двухстадийная диффузия состоит из загонки примеси и разгонки примеси. Рассчитаем распределения концентрации примеси по глубине пластины и количество примесных атомов, вводимых в процессе загонки (рис. 24).

Загонка  
примеси

$$t1 := 100$$

$$C1(x) := C_0 \cdot \left(1 - \operatorname{erf}\left(\frac{x}{2 \cdot \sqrt{D(T1) \cdot t1}}\right)\right)$$

Количество примесных атомов

$$Q := C_0 \cdot \sqrt{\pi \cdot D(T1) \cdot t1}$$

$$Q = 9.809 \times 10^{13}$$

Разгонка  
примеси

$$t2 := 9000$$

$$C3(x) := \frac{Q}{\sqrt{\pi \cdot D(T1) \cdot t2}} \cdot \exp\left(-\frac{x^2}{4 \cdot D(T1) \cdot t2}\right) \quad C3(0) = 1.054 \times 10^{18}$$

Рис. 24. Расчёт загонки и разгонки примеси в двухстадийном процессе

Распределение примеси после загонки локализовано в узкой приповерхностной области пластины (рис. 25, черная пунктирная линия  $C1(x)$ ).

После прекращения подачи газовой смеси, содержащей примесные атомы, в реакторе проводится разгонка примеси, во время которой атомы

примеси распределяются на заданную глубину пластины (рис. 25, красная линия  $C3(x)$ ).

На рис. 25  $Cc$  - это исходная концентрация примеси в подложке.

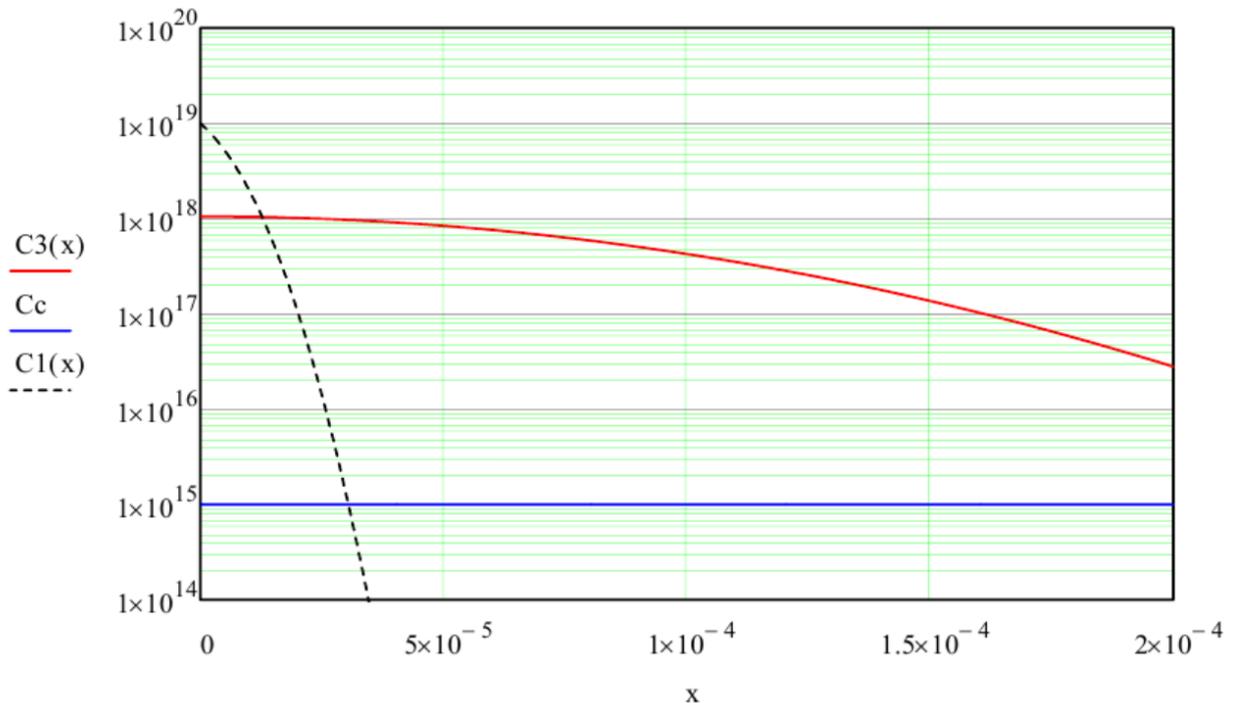


Рис. 25. Распределение примеси при двухстадийной диффузии

Если при диффузии в подложку одного типа проводимости вводилась примесь противоположного типа проводимости, можно рассчитать глубину залегания  $p-n$  перехода (рис. 26).

$$Z3 := 2\sqrt{D(T1)\cdot t2} \cdot \sqrt{\ln\left(\frac{Q}{Cc\cdot\sqrt{\pi\cdot D(T1)\cdot t2}}\right)} \quad Z3 = 2.77 \times 10^{-4} \quad \text{CM}$$

Рис. 26. Глубина залегания  $p-n$  перехода при двухстадийной диффузии

### Задача 2. Анализ температурной зависимости диффузии

На рис. 27 заданы переменные и постоянные для анализа распределения примеси по пластине в зависимости от температуры диффузии. Характеристики стандартных диффундирующих в кремний примесей приведены ранее в табл. 2.

Далее, аналогично рассмотренной в задаче 1 модели диффузии, рассчитаем:

- 1) температурную зависимость коэффициента диффузии (рис. 28);

|   |                            |                    |
|---|----------------------------|--------------------|
| Энергия активации диффузии примеси                      | $E_a := 3.63$              | эВ                 |
| Температура диффузии                                    | $T_1 := 273 + 1100$        | К                  |
| Время диффузии  | $t := 1000$                | с                  |
| Коэффициент диффузии при бесконечно большой температуре | $D_0 := 0.785$             | см <sup>2</sup> /с |
| Поверхностная концентрация диффузанта                   | $C_0 := 1 \cdot 10^{18}$   | 1/см <sup>3</sup>  |
| Концентрация примеси в исходном кремнии                 | $C_s := 10^{15}$           | 1/см <sup>3</sup>  |
| Постоянная Больцмана                                    | $k := 8.625 \cdot 10^{-5}$ | эВ/К               |

Рис. 27. Параметры процесса диффузии

$$T := 1000 \dots 1600 \quad D(T) := D_0 \cdot \exp\left(\frac{-E_a}{k \cdot T}\right) \quad D(T_1) = 3.822 \times 10^{-14}$$

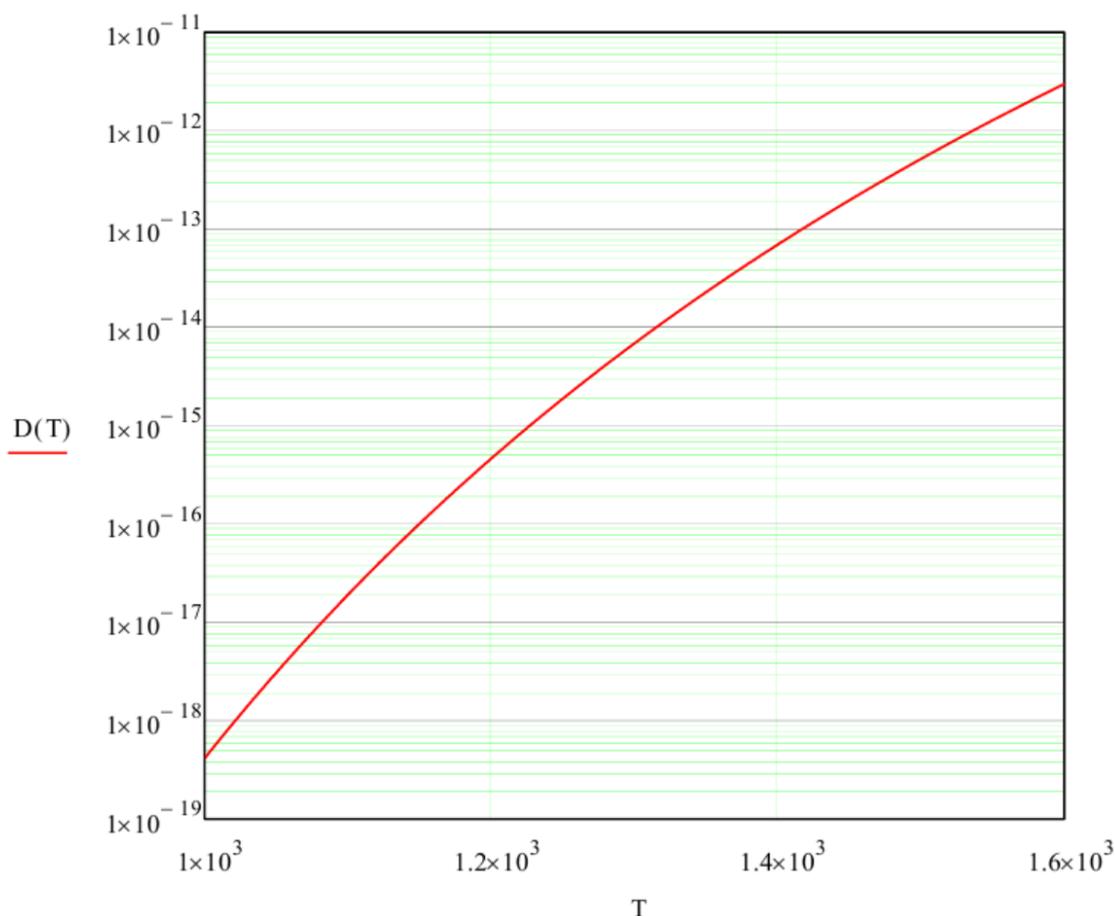
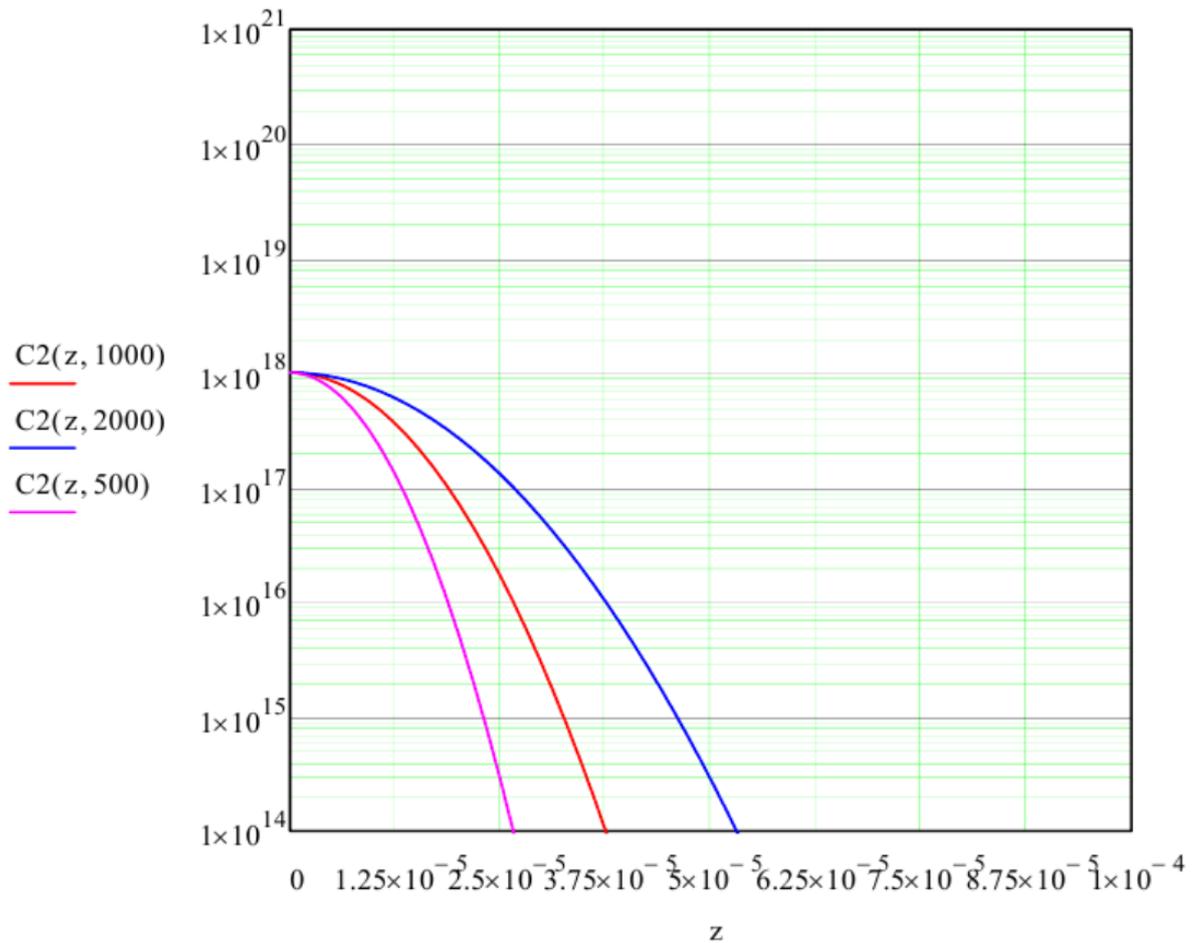


Рис. 28. Температурная зависимость коэффициента диффузии

2) параметры процесса диффузии из источника ограниченной мощности (рис. 29) для трех вариантов времени (500, 1000 и 2000 с);

$$C2(z, tv) := C_0 \cdot \exp\left(\frac{-z^2}{4 \cdot D(T1) \cdot tv}\right)$$



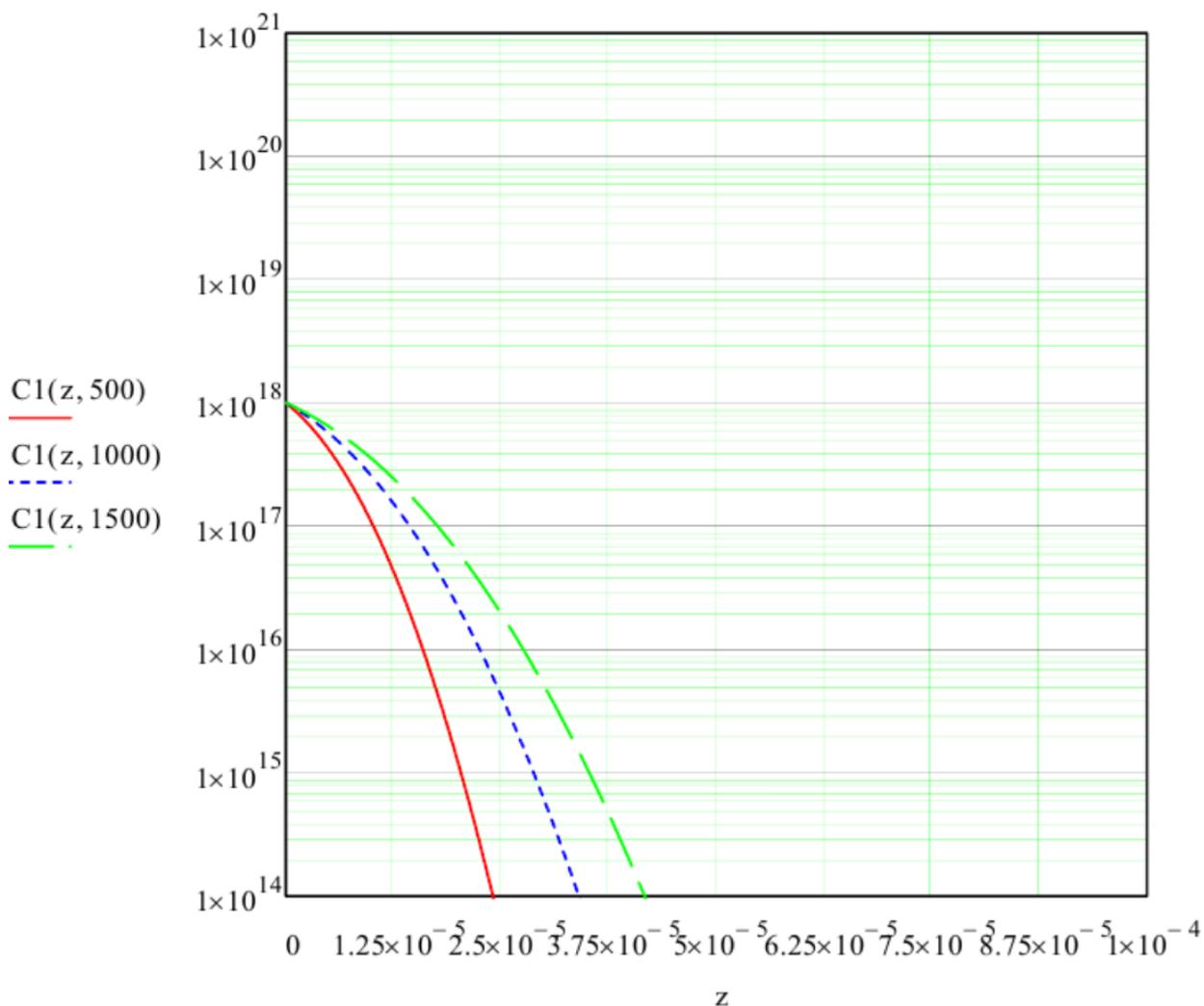
Глубина залегания p-n перехода

$$Z1 := 2 \cdot \sqrt{D(T1) \cdot t} \cdot \sqrt{\ln\left(\frac{C_0}{C_c}\right)} \quad Z1 = 3.25 \times 10^{-5}$$

Рис. 29. Диффузия из источника ограниченной мощности

3) параметры процесса диффузии из источника бесконечной мощности (рис. 30) для трех вариантов времени (500, 1000 и 2000 с).

$$C1(z, vt) := Co \cdot \left( 1 - \operatorname{erf} \left( \frac{z}{2\sqrt{D(T1) \cdot vt}} \right) \right)$$



Глубина залегания p-n перехода

$$f(z) := \left( \frac{Co}{Cc} \right) \cdot \left( 1 - \operatorname{erf} \left( \frac{z}{2\sqrt{D(T1) \cdot t}} \right) \right) - 1$$

$$z := 0.3 \cdot 10^{-4} \quad Z_2 := \operatorname{root}(f(z), z)$$

$$Z_2 = 2.877 \times 10^{-5} \quad \text{CM}$$

Рис. 30. Диффузия из источника бесконечной мощности

Для двухстадийной диффузии (загонка – разгонка примеси) определим количество примесных атомов и глубину залегания  $p$ - $n$  перехода (рис. 31).

$$T := 1000..1600$$

$$D(T) := D_0 \cdot \exp\left(-\frac{E_a}{k \cdot T}\right) \quad D(T1) = 3.822 \times 10^{-14}$$

Загонка примеси  $t1 := 100$

$$Cv(x) := C_0 \cdot \left(1 - \operatorname{erf}\left(\frac{x}{2 \cdot \sqrt{D(T1) \cdot t1}}\right)\right)$$

Количество примесных атомов  $Q := C_0 \cdot \sqrt{\pi \cdot D(T1) \cdot t1} \quad Q = 3.465 \times 10^{12}$

Разгонка примеси  $t2 := 9000$

$$C3(x, tc) := \frac{Q}{\sqrt{\pi \cdot D(T1) \cdot tc}} \cdot \exp\left(-\frac{x^2}{4 \cdot D(T1) \cdot tc}\right) \quad C3(0, t2) = 1.054 \times 10^{17}$$

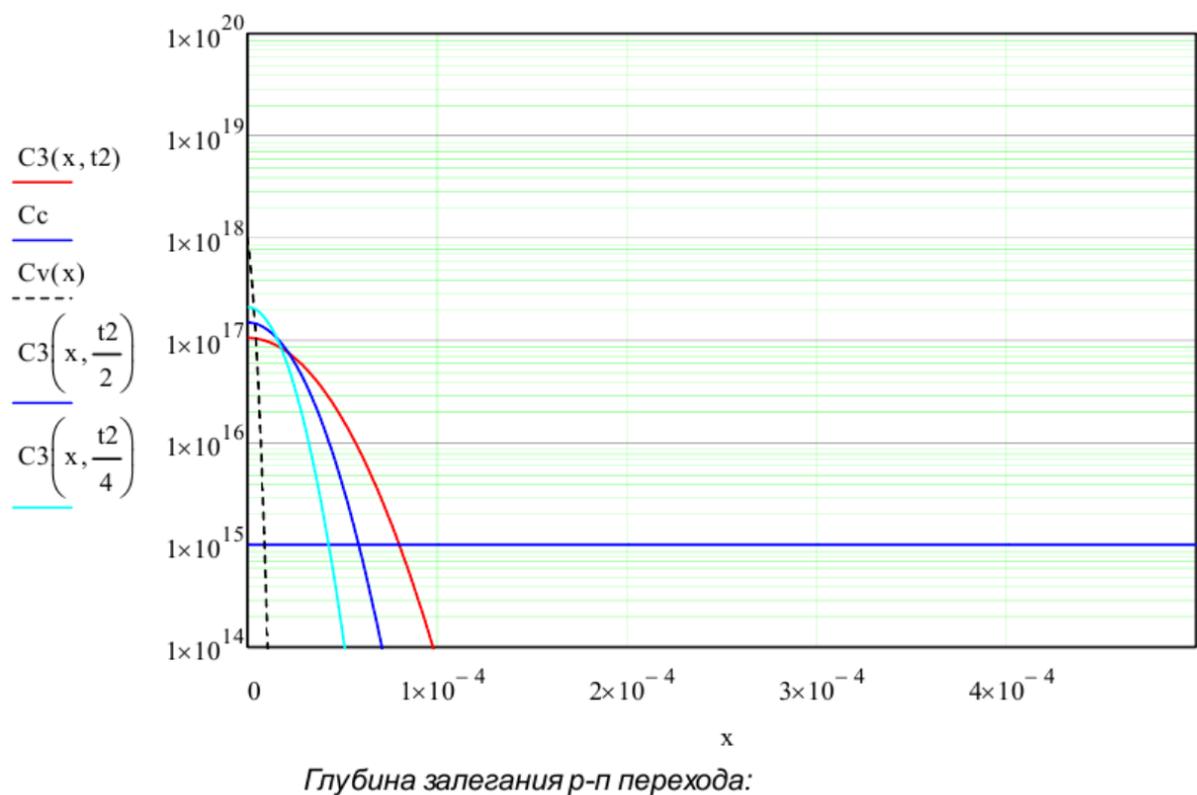
Рис. 31. Расчёт параметров двухстадийной диффузии

Результат моделирования двухстадийной диффузии и расчёт глубины залегания  $p$ - $n$  перехода приведены на рис. 31'.

Индексом  $Cv$  обозначено распределение примеси при загонке из источника неограниченной мощности.

$C3(x, t)$  – рассматриваются варианты разгонки примеси при различной длительности процесса ( $t = 9000, 4500, 2250$  с).

$C_0$  – исходная концентрация примеси в подложке.



$$Z_3 := 2\sqrt{D(T_1) \cdot t_2} \cdot \ln\left(\frac{Q}{C_c \cdot \sqrt{\pi \cdot D(T_1) \cdot t_2}}\right) \quad Z_3 = 8.006 \times 10^{-5} \quad \text{см}$$

Рис. 31'. Глубина залегания  $p$ - $n$  перехода при двухстадийной диффузии и различной длительности процесса

#### ЗАДАНИЕ 4. ОСНОВНЫЕ ЭТАПЫ ФОРМИРОВАНИЯ БИПОЛЯРНОГО ТРАНЗИСТОРА

При разработке стандартного биполярного транзистора в кремниевой пластине формируют два  $p$ - $n$  перехода, последовательно внедряя в подложку  $p$ -типа донорную, акцепторную и донорную примеси (для подложки  $n$ -типа последовательность примесей  $p$ - $n$ - $p$ ) (рис. 32).

Примеси  $n$ - и  $p$ -типа диффундируют в подложку с различной скоростью, то есть в технологическом процессе можно сформировать определенную поверхностную концентрацию примеси заданного типа проводимости («коллектор»), а далее получить  $n$ - $p$  – структуру (переход «эмиттер-база») одновременной диффузией обоих элементов из паровой фазы.

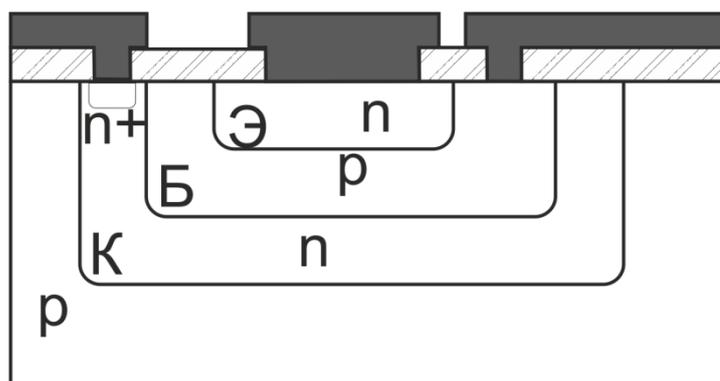


Рис. 32. Структура биполярного транзистора, реализованного по планарной технологии

Для кремниевой технологии применяются два подхода:

- 1) одновременная диффузия примесей элементов III-V групп;
- 2) последовательная диффузия  $n$ - и  $p$ -примесей из сплавов, соединений или оксидов.

Процесс одновременной диффузии проще в реализации на оборудовании, но не позволяет отдельно управлять концентрациями обеих примесей и температурой их разгонки.

При последовательной диффузии (которая используется как стандартная технология) сначала в полупроводник вводится медленно-диффундирующая примесь (донорная), а далее – быстро-диффундирующие акцепторы.

Для многокомпонентных систем предполагается (и согласуется с опытами), что каждая примесь будет диффундировать независимо от других.

При расчёте биполярного транзистора требуется определить соотношение между концентрациями примесей при разных коэффициентах диффузии доноров и акцепторов и при разных их поверхностных концентрациях (общая концентрация примеси на поверхности пластины постоянна ( $C_s$ )). Для расчёта предполагаем, что донорная примесь более растворима и менее подвижна, чем акцепторная.

На рис. 33 приведены константы и переменные для расчёта биполярного транзистора (табл. 3).

|                                    |                            |          |
|------------------------------------|----------------------------|----------|
| Коэффициент диффузии В при 1200 С  | $DoB := 5.1$               | $cm^2/c$ |
| Коэффициент диффузии As при 1200 С | $DoAs := 3.85$             | $cm^2/c$ |
| Энергия активации диффузии для В   | $EaB := 3.7$               | эВ       |
| Энергия активации диффузии для As  | $EaAs := 3.85$             | эВ       |
| Поверхностная концентрация В       | $No_1 := 10^{21}$          | $1/cm^3$ |
| Поверхностная концентрация As      | $No_3 := 10^{21}$          | $1/cm^3$ |
| Концентрация примеси в исходном Si | $Nc := 10^{15}$            | $1/cm^3$ |
| Максимальная растворимость В в Si  | $MB := 4 \cdot 10^{20}$    | $1/cm^3$ |
| Максимальная растворимость As в Si | $MAs := 1.3 \cdot 10^{21}$ | $1/cm^3$ |
| Постоянная Больцмана               | $k := 8.625 \cdot 10^{-5}$ | эВ/К     |

Рис. 33. Основные переменные и постоянные для расчёта биполярного транзистора

Для примеси бора  $B$  заданы температура и время процессов загонки и разгонки; для мышьяка  $As$  – время диффузии (рис. 34).

Диффузию бора стандартно проводят в две стадии.

Загонка длится относительно короткий промежуток времени, количество примеси строго регламентировано. Для загонки (рис. 35) рассчитывают распределение концентрации  $C(x)$  по функции ошибок и величину концентрации атомов примеси  $Q$ .

Разгонка примеси бора (рис. 36) длится большее время и характеризуется распределением по экспоненте.

После разгонки определяется глубина залегания первого  $p$ - $n$ -перехода  $x_i$ .

|                  |                      |   |
|------------------|----------------------|---|
| Загонка бора     | $T_1 := 273 + 1100$  | К |
|                  | $t_1 := 60 \cdot 15$ | с |
| Разгонка бора    | $T_2 := 273 + 1200$  | К |
|                  | $t_2 := 60 \cdot 60$ | с |
| Диффузия мышьяка | $T_3 := 273 + 1200$  | К |
|                  | $t_3 := 60 \cdot 15$ | с |

Рис. 34. Температура и время диффузии примесей  $B$  и  $As$

Параметры растворимости примесей в кремнии

| Элемент | Максимальная растворимость, ат/см <sup>3</sup> | Температура максимальной растворимости, °С |
|---------|--|--|
| Al      | $2 \cdot 10^{19}$                              | 1150                                       |
| B       | $4 \cdot 10^{20}$                              | 1200                                       |
| Bi      | $8 \cdot 10^{17}$                              | 1300                                       |
| Ga      | $4 \cdot 10^{19}$                              | 1250                                       |
| In      | $1 \cdot 10^{19}$                              | 1300                                       |
| As      | $2 \cdot 10^{21}$                              | 1150                                       |
| Sb      | $8 \cdot 10^{19}$                              | 1300                                       |
| P       | $1,3 \cdot 10^{21}$                            | 1150                                       |

$$D(T) := D_{0B} \cdot \exp\left(-\frac{E_{aB}}{k \cdot T}\right) \quad D(T_1) = 1.375 \times 10^{-13}$$

$$N1(x) := \begin{cases} N_{01} \cdot \left(1 - \operatorname{erf}\left(\frac{x}{2 \cdot \sqrt{D(T_1) \cdot t_1}}\right)\right) & \text{if } N_{01} < MB \\ MB \cdot \left(1 - \operatorname{erf}\left(\frac{x}{2 \cdot \sqrt{D(T_1) \cdot t_1}}\right)\right) & \text{otherwise} \end{cases}$$

Количество примесных атомов

$$Q := \begin{cases} N_{01} \cdot \sqrt{\pi \cdot D(T_1) \cdot t_1} & \text{if } N_{01} < MB \\ (MB \cdot \sqrt{\pi \cdot D(T_1) \cdot t_1}) & \text{otherwise} \end{cases}$$

Рис. 35. Первая стадия. Загонка *B* в *Si*

**2-я стадия - разгонка *B***

$$N2(x) := \frac{Q}{\sqrt{\pi \cdot D(T_2) \cdot (t_2 + t_3)}} \cdot \exp\left[-\frac{x^2}{4 \cdot D(T_2) \cdot (t_2 + t_3)}\right] \quad D(T_2) = 1.147 \times 10^{-12}$$

Рис. 36. Вторая стадия. Разгонка *B* в *Si*

Диффузия эмиттера (примесь мышьяка) обычно проводится из источника с постоянной концентрацией мышьяка на поверхности и рассчитывается через функцию ошибок (рис. 37).

$$D_{As}(T) := D_{0As} \cdot \exp\left(-\frac{E_{aAs}}{k \cdot T}\right)$$

$$D_3 := D_{As}(T_3) \qquad D_3 = 2.658 \times 10^{-13}$$

$$N_3(x) := \begin{cases} N_{03} \cdot \left(1 - \operatorname{erf}\left(\frac{x}{2 \cdot \sqrt{D_{As}(T_3) \cdot t_3}}\right)\right) & \text{if } N_{03} < M_{As} \\ M_{As} \cdot \left(1 - \operatorname{erf}\left(\frac{x}{2 \cdot \sqrt{D_{As}(T_3) \cdot t_3}}\right)\right) & \text{otherwise} \end{cases}$$

Рис. 37. Третья стадия. Диффузия As в Si

Полное распределение примесей в биполярном транзисторе после загонки B (кривая  $N1(x)$ ), после его разгонки (кривая  $N2(x)$ ) и после диффузии фосфора (кривая  $N3(x)$ ) приведено на рис. 38.  $N_c$  – исходная концентрация примеси в подложке. Профиль распределения концентрации примеси по глубине пластины отображён на рис. 39.

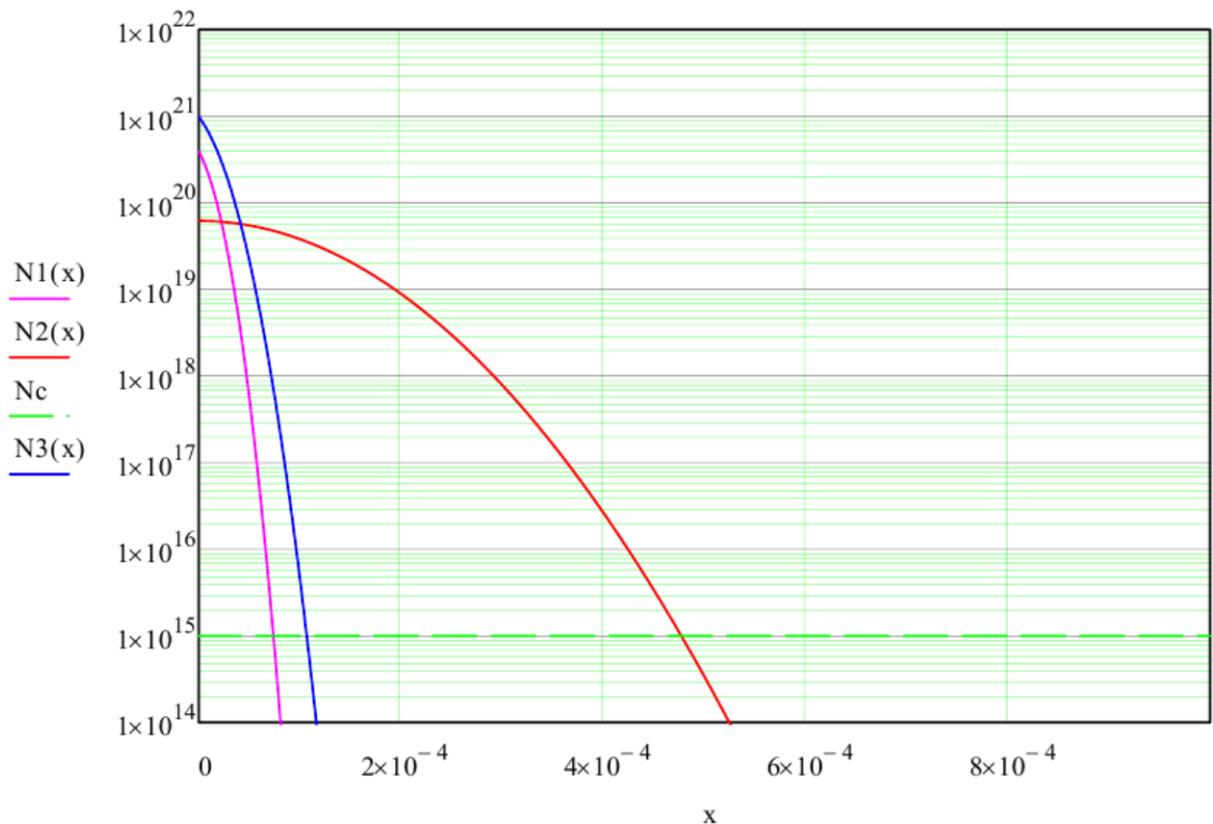


Рис. 38. Распределение примесей B и As в Si

$$x := 0, 10^{-6} \dots 10^{-3} \quad F1(x) := |N3(x) - N2(x) + Nc|$$

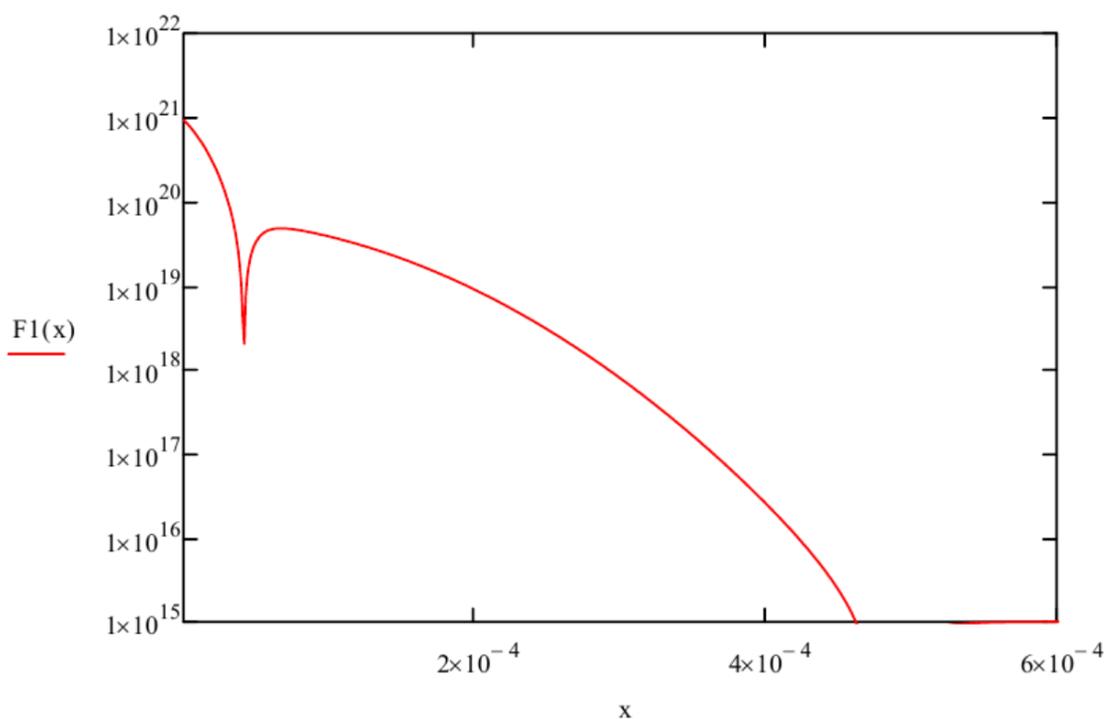


Рис. 39. Профиль распределения концентрации

На рис. 40 приводится соотношение концентраций примесей  $K(x)$  без учета их расположения на пластине.

$$x := 0, 10^{-5} \dots 10^{-3} \quad K(x) := \frac{N2(x)}{N3(x) + Nc}$$

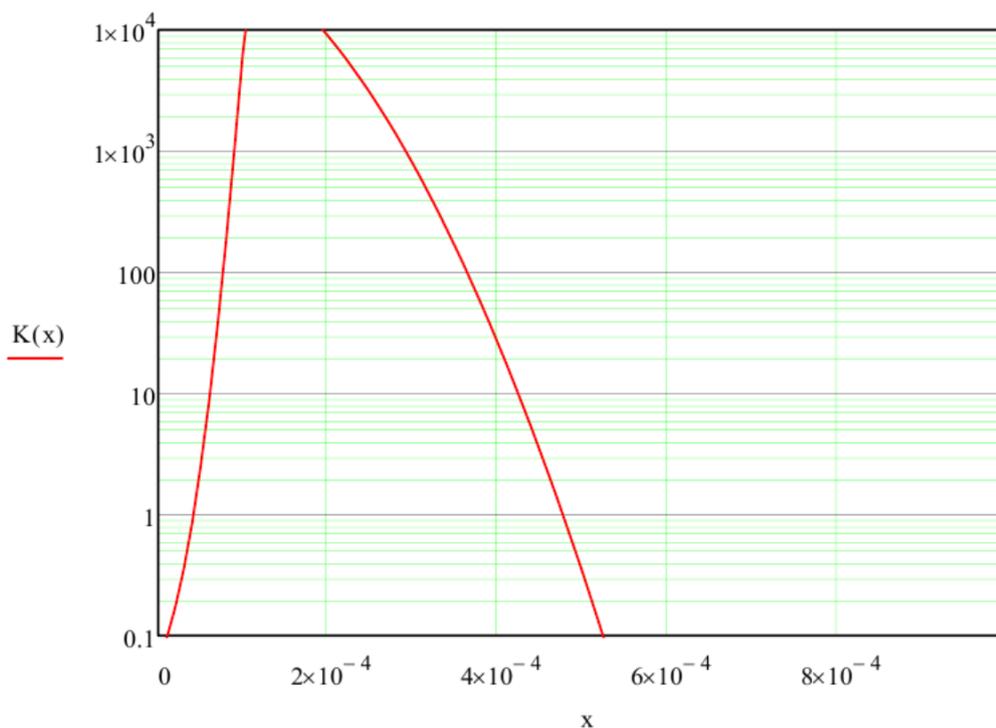


Рис. 40. Отношение концентрации доноров и акцепторов

Расчёт основных параметров областей биполярного транзистора определяет расположение и величину максимальной концентрации примеси в области базы транзистора, глубину залегания эмиттерного и коллекторного *p-n*-переходов, а также ширину базы (рис. 41).

Максимальная концентрация примеси в базе и глубина ее залегания

$$DK(x) := \frac{d}{dx}K(x)$$

$$x := 8 \cdot 10^{-5}$$

$$x3 := \text{root}(DK(x), x)$$

$$x3 = -1.059 \times 10^{-3}$$

$$F1(x3) = 2 \times 10^{21}$$

Глубина залегания эмиттерного *p-n* перехода.

$$x := 1 \times 10^{-4}$$

$$x1 := \text{root}(K(x), x)$$

$$x1 = -3.824 \times 10^{-4}$$

Глубина залегания коллекторного *p-n* перехода.

$$x2 := 2 \sqrt{D(T_2) \cdot (t_2 + t_3)} \cdot \sqrt{\ln \left[ \frac{Q}{\sqrt{\pi \cdot D(T_2) \cdot (t_2 + t_3)} \cdot N_c} \right]}$$

$$x2 = 4.773 \times 10^{-4}$$

Ширина базы.

$$x1 - x2 = -8.596 \times 10^{-4}$$

Рис. 41. Параметры разработанного биполярного транзистора

## ЗАДАНИЕ 5. МОДЕЛИРОВАНИЕ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА ИОННОЙ ИМПЛАНТАЦИИ

При ионной имплантации заряженные частицы ионного пучка внедряются в твердое тело, изменяя его свойства. Внедряющиеся ионы снижают свою энергию за счёт упругих и неупругих соударений.

При неупругих столкновениях часть энергии ионного пучка теряется на возбуждение (ионизацию) атомов мишени (пластины), то есть взаимодействует с электронами кристаллической решетки.

При упругих столкновениях энергия ионов передается ядрам атомов мишени.

Также энергетические потери идут на обмен зарядами между ионами и атомами мишени.

Если переданная атому при упругом соударении энергия оказывается больше, чем энергия связи атомов в теле мишени, атом покидает узел – образуется пара Френкеля (вакансия + атом в междоузлии).

Атомы поверхностного слоя, получив энергию иона, могут отрываться от кристалла – происходит процесс распыления – образуется дефект Шоттки (вакансия).

Первично-смещенный атом – атом отдачи – обладает большой энергией; при его движении возникает каскад смещений, создающий дискретные разупорядоченные зоны размером 3 – 10 нм. То есть имплантация обеспечивает накопление дефектов и при превышении критической величины (дозы аморфизации) в приповерхностной области пластины формируется сплошной аморфный слой.

Внедренные ионы могут сразу попадать в места вакансий в кристаллической решетке (что происходит редко) или занимать места в междоузлиях (обычно). В междоузлиях они не являются электрически активными. Для их перемещения в узлы и восстановления структуры кристаллической решетки проводится отжиг, при котором:

- распадаются и аннигилируют радиационные дефекты кристалла;
- внедренная примесь занимает места вакансий в узлах решетки, то есть образуется слой *p*-/*n*-типа проводимости.

При моделировании процесса ионной имплантации можно использовать упрощенную классическую модель Линхарда-Шарфа-Шиотта, которая описывает процесс для:

- изотропной или аморфной полубесконечной мишени;
- рассеяния ионов в мишени случайным образом;
- распределения пробега ионов по Гауссу.

На рис. 42 приведены константы используемой при моделировании ионной имплантации мишени.

В табл. 4 приводится справочная информация по основным материалам полупроводниковой технологии (полужирным шрифтом выделены основные полупроводники кремний *Si* и германий *Ge*). Дана информация по основным легирующим примесям и стандартным металлам, применяемым в технологии ионной имплантации.

|                                     |                  |                         |
|-------------------------------------|------------------|-------------------------|
| атомный номер                       | $Z_2 := 14$      |                         |
| атомный вес                         | $m_2 := 28$      | г·моль <sup>-1</sup>    |
| число атомов в единице объема       | $N := 50.4$      | ат·нм <sup>-3</sup>     |
| концентрация примеси в исходном п/п | $C_0 := 10^{21}$ | $\frac{1}{\text{см}^3}$ |

Рис. 42. Константы мишени

Таблица 4

Параметры материалов полупроводниковой технологии, используемые для расчёта имплантации

| Материал  | Атомный номер,<br>$Z$ | Атомный вес $M$ ,<br>г/моль | Число ат. в ед.<br>объема, $N \times 10^{23}$<br>ат/см <sup>2</sup> |
|-----------|-----------------------|-----------------------------|---|
| Be        | 4                     | 9                           | 123   |
| B         | 5                     | 10,8                        | 137   |
| C         | 6                     | 12                          | 113   |
| Mg        | 12                    | 24,3                        | 43,1  |
| Al        | 13                    | 27                          | 60,4  |
| <b>Si</b> | <b>14</b>             | <b>28</b>                   | <b>50,4</b>   |
| Ti        | 22                    | 47,9                        | 56,6  |
| V         | 23                    | 51                          | 66,3  |
| Cr        | 24                    | 52                          | 76,4  |
| Mn        | 25                    | 54,9                        | 80,1  |
| Fe        | 26                    | 55,9                        | 84,8  |
| Ni        | 28                    | 58,7                        | 90,3  |
| Cu        | 29                    | 63,5                        | 85,2  |
| <b>Ge</b> | <b>32</b>             | <b>72,6</b>                 | <b>45,3</b>   |
| Nb        | 41                    | 92,9                        | 54,5  |
| Mo        | 42                    | 96                          | 63,9  |
| Ag        | 47                    | 107,8                       | 58,5  |
| Ta        | 73                    | 181                         | 55,2  |
| W         | 74                    | 183                         | 60,9  |
| Pt        | 78                    | 195                         | 66,1  |
| Au        | 79                    | 197                         | 59  |

Параметры имплантируемой примеси бора  $B$  представлены на рис. 43.

|                  |  |
|------------------|--|
| атомный номер    | $Z1 := 5$  |
| атомный вес      | $m1 := 10.8 \quad \text{г} \cdot \text{моль}^{-1}$ |
| энергия ионов    | $E := 10000 \quad \text{эВ}$                       |
| доза имплантации | $D := 10^{16} \quad \frac{1}{\text{см}^2}$         |

Рис. 43. Константы имплантируемой примеси

На рис. 44 приведены формулы для расчёта основных параметров процесса ионной имплантации: коэффициент замедляющей способности ( $K$ ,  $K1$ ), полный пробег ионов  $R$  и его дисперсия  $\Delta R$ , проекция пробега ионов  $R_p$  и дисперсия проекции пробега ионов  $\Delta R_p$ .

|  |   |
|--|---|
| Коэффициент ядерной замедляющей способности: | $K := 1.8 \cdot \frac{\sqrt{Z1^{\frac{2}{3}} + Z2^{\frac{2}{3}} \cdot (m1 + m2)}}{N \cdot Z1 \cdot Z2 \cdot m1} \quad \text{нм} \cdot \text{эВ}^{-1}$ |
| Коэффициент ядерной замедляющей способности: | $K1 := 3.28 \cdot 10^{-3} \cdot (Z1 + Z2) \cdot \frac{N}{\sqrt{m1}} \quad \text{эВ}^2 \cdot \text{нм}^{-1}$   |
| Полный пробег иона:                          | $R := 2 \frac{\sqrt{E}}{K1} - \frac{\ln(1 + 2 \cdot K \cdot K1 \cdot \sqrt{E})}{K \cdot K1^2} \quad \text{нм}$  |
| Дисперсия полного пробега иона:              | $\Delta R := R \frac{2\sqrt{m1 \cdot m2}}{3 \cdot (m1 + m2)} \quad \text{нм}$   |
| Проекция пробега иона: $b := \frac{1}{3}$    | $R_p := \frac{R}{1 + b \cdot \frac{m2}{m1}} \quad \text{нм}$  |
| Дисперсия проекции пробега иона:             | $\Delta R_p := \frac{2}{3} \cdot \frac{\sqrt{m1 \cdot m2}}{m1 + m2} R_p \quad \text{нм}$  |

Рис. 44. Формулы для расчёта основных параметров процесса ионной имплантации

Расчёт по формулам с рис. 44 приведен на рис. 45.

$$\underline{K} := 1.8 \cdot \frac{\sqrt{\frac{2}{Z1^3} + \frac{2}{Z2^3}} \cdot (m1 + m2)}{N \cdot Z1 \cdot Z2 \cdot m1} = 5.417 \times 10^{-3}$$

$$\underline{K1} := 3.28 \cdot 10^{-3} \cdot (Z1 + Z2) \cdot \frac{N}{\sqrt{m1}} = 0.956$$

$$\underline{R} := 2 \frac{\sqrt{E}}{K1} - \frac{\ln(1 + 2 \cdot K \cdot K1 \cdot \sqrt{E})}{K \cdot K1^2} = 65.625$$

$$\underline{\Delta R} := R \frac{2\sqrt{m1 \cdot m2}}{3 \cdot (m1 + m2)} = 19.608$$

$$\underline{Rp} := \frac{R}{1 + b \cdot \frac{m2}{m1}} = 35.203$$

$$\underline{\Delta Rp} := \frac{2}{3} \cdot \frac{\sqrt{m1 \cdot m2}}{m1 + m2} Rp = 10.518$$

Рис. 45. Расчёт по формулам (рис. 44)

Если вводимая в структуру примесь и легирование подложки выполнены элементами одного типа проводимости (рис. 46, линии синего цвета  $C0$  и  $C1(x)$ ), то в структуре не будет формироваться  $p-n$  переход.

Если вводимая в структуру примесь и легирование подложки выполнены элементами разного типа проводимости (рис. 46, линии  $C0$  синего и  $C2(x)$  красного цвета), то в структуре будут созданы один или два  $p-n$  перехода в зависимости от формируемой структуры.

Легирующая примесь и исходная  
одного типа проводимости:

$$C1(x) := \frac{10^7 \cdot D}{\sqrt{2\pi} \cdot \Delta Rp} \cdot \exp\left[\frac{-1}{2} \cdot \left(\frac{x - Rp}{\Delta Rp}\right)^2\right] + Co \quad \frac{1}{\text{см}^3}$$

Легирующая примесь и исходная  
противоположных типов проводимости:

$$C2(x) := \frac{10^7 \cdot D}{\sqrt{2\pi} \cdot \Delta Rp} \cdot \exp\left[\frac{-1}{2} \cdot \left(\frac{x - Rp}{\Delta Rp}\right)^2\right] - Co \quad \frac{1}{\text{см}^3}$$

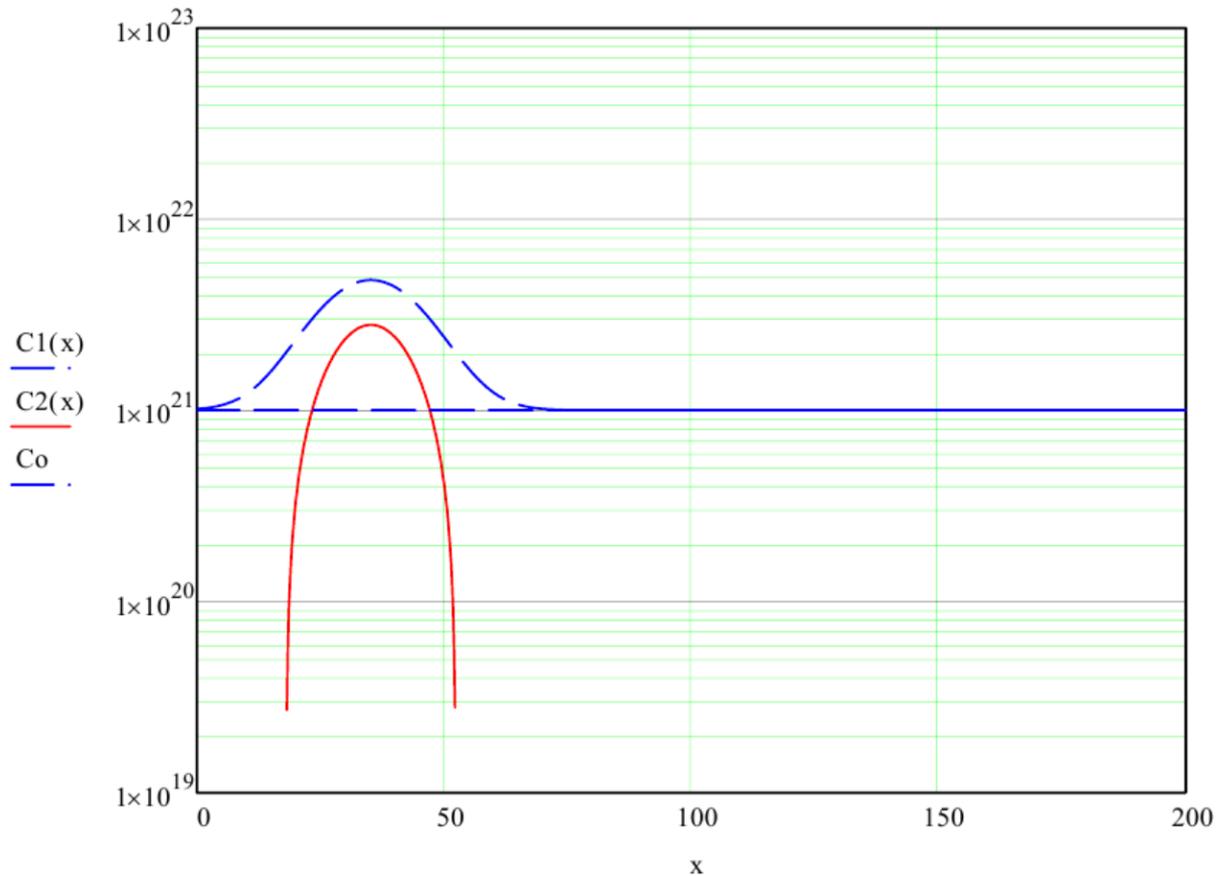


Рис. 46. Расчёт профиля распределения концентрации легирующей примеси

Глубины залегания  $p$ - $n$  переходов рассчитываются по формулам, приведенным на рис. 47.

$$x1 := Rp + \Delta Rp \cdot \sqrt{2 \cdot \ln\left(\frac{D \cdot 10^7}{\sqrt{2\pi} \cdot \Delta Rp}\right) - 2 \cdot \ln(Co)} \quad x1 = 52.377 \quad \text{HM}$$

$$x2 := Rp - \Delta Rp \cdot \sqrt{2 \cdot \ln\left(\frac{D \cdot 10^7}{\sqrt{2\pi} \cdot \Delta Rp}\right) - 2 \cdot \ln(Co)} \quad x2 = 18.028 \quad \text{HM}$$

Рис. 47. Расчёт глубины залегания  $p$ - $n$  переходов

На завершающем этапе рассчитываются максимальные концентрации легирующей примеси  $C1$  и  $C2$  в зависимости от проекции пробега ионов  $R_p$ .

$$C1(R_p) = 4.793 \times 10^{21} \qquad C2(R_p) = 2.793 \times 10^{21} \qquad \frac{1}{\text{см}^3}$$

Рис. 48. Расчёт максимальной концентрации легирующей примеси

Наиболее подходящим для описания распределения ионов примеси в кремнии, окисле  $SiO_2$  и нитриде кремния  $Si_3N_4$  является модель PEARSON IV, которая позволяет с достаточной точностью смоделировать эффект каналирования.

### ЗАДАНИЕ 6. РАСЧЁТ ПАРАМЕТРОВ ДВУМЕРНОГО РАСПРЕДЕЛЕНИЯ ИОНОВ С ЗАХОДОМ ПОД КРАЙ МАСКИ

Если рассчитываются и изготавливаются электронные приборы с размерами более 10-ков мкм, то в них боковым отклонением имплантируемых ионов можно пренебречь.

В СБИС с элементами микронных и субмикронных размеров боковое отклонение является важным фактором, определяющим двумерные эффекты возле края маски, которые в значительной мере изменяют выходные характеристики разрабатываемого прибора.

Боковое отклонение при имплантации возникает из-за рассеяния ионов и сопоставимо с разбросом пробегов  $R_p$ . Простейшей методикой его моделирования может служить двумерное распределение примесей по Гауссу.

На рис. 49 приведены основные параметры имплантации ионов в кремний через маску с вертикальным краем.

|                           |                                   |                      |
|---------------------------|-----------------------------------|----------------------|
| энергия иона              | $E := 100$                        | кэВ                  |
| проективный пробег        | $R_p := 397 \cdot 10^{-7}$        | см                   |
| дисперсия пробега         | $\Delta R_p := 94 \cdot 10^{-7}$  | см                   |
| боковое рассеяние         | $\Delta R_x := 123 \cdot 10^{-7}$ | см                   |
| полуширина окна в маске   | $a := 500 \cdot 10^{-7}$          | см                   |
| доза облучения            | $N_0 := 10^{14}$                  | мккл/см <sup>2</sup> |
| концентрация примеси в Si |                                   | 1/см <sup>3</sup>    |

Рис. 49. Параметры имплантации примеси в кремний  $Si$  через непрозрачную маску с вертикальным краем

Геометрия окна, направление имплантации ионов и полубесконечная структура пластины кремния приведены на рис. 50. Маска имеет вертикальный край и непрозрачна для ионов. Рассмотрим окно в маске шириной  $2a$ . Из-за симметрии модель можно строить для половины окна (на рис. – в положительном направлении вдоль оси  $X$ ). Предполагаем длину маски равной бесконечности, то есть распределение примесей будет двумерным в координатах  $x$ - $y$  и будет отвечать Гауссовому профилю.

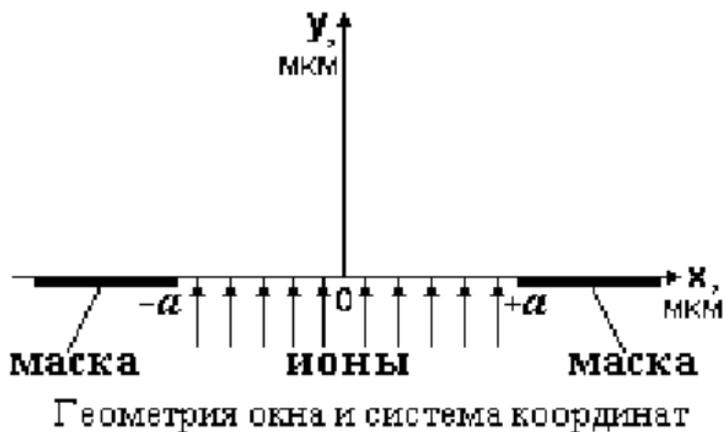


Рис. 50. Геометрия окна в маске и направление внедрения ионов примеси

На рис. 51 приведены данные по проективному пробегу ионов  $R_p$  и его дисперсии  $\Delta R_p$  для различных примесей в кремнии в зависимости от энергии облучения  $E$ , эВ.

|    | $E$ , кэВ    | 20   | 40   | 60    | 80    | 100   | 120   | 140   | 160   | 180   | 200   |
|----|--------------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| B  | $R_p$        | 78,4 | 161  | 243,8 | 323,8 | 397,7 | 468,8 | 537,4 | 603   | 665,4 | 725,3 |
|    | $\Delta R_p$ | 32,1 | 53,8 | 70,8  | 83,9  | 93,9  | 102,5 | 109,8 | 116   | 121,3 | 125,9 |
| P  | $R_p$        | 26   | 49   | 73,2  | 98    | 123,3 | 149,1 | 175,2 | 201,4 | 227,7 | 253,8 |
|    | $\Delta R_p$ | 9,4  | 16,4 | 23,1  | 29,5  | 35,4  | 41,1  | 46,6  | 51,8  | 56,6  | 61    |
| As | $R_p$        | 15,9 | 27,1 | 37,7  | 48,1  | 58,4  | 68,6  | 78,9  | 89,1  | 99,5  | 110,1 |
|    | $\Delta R_p$ | 3,7  | 6,2  | 8,4   | 10,5  | 12,5  | 14,5  | 16,3  | 18,2  | 20    | 21,9  |
| Sb | $R_p$        | 14   | 22,9 | 30,8  | 38,4  | 45,7  | 52,8  | 59,9  | 66,9  | 73,9  | 80,9  |
|    | $\Delta R_p$ | 2,4  | 3,8  | 5,1   | 6,3   | 7,4   | 8,4   | 9,5   | 10,5  | 11,5  | 12,5  |

Рис. 51. Проективный пробег  $R_p$  и его дисперсия  $\Delta R_p$  для различных примесей в кремнии в зависимости от энергии облучения  $E$ , эВ

Доза имплантации примеси  $N$  (рис. 52) рассчитывается через заданную в условии дозу облучения  $D$  (количество ионов, падающих на единицу поверхности мишени за время имплантации), деленную на заряд электрона на квадратный сантиметр (равен  $6,25 \times 10^{12}$ ).

$$N := 6.25 \cdot 10^{12} \cdot D \quad \text{ион/см}^2 \quad N = 6.25 \times 10^{14}$$

Рис. 52. Расчёт дозы имплантации  $N$  через известную величину дозы облучения  $D$

На рис. 53 приведен Гауссовый профиль для двумерного распределения примеси. Если край маски имеет произвольную форму, то расчёт ведётся через интеграл. При расчёте структуру можно разбивать на тонкие вертикальные зоны таким образом, чтобы поверхность между соседними границами считалась плоской.

$$C(x, y) := \frac{N}{\Delta R_p \cdot \sqrt{2\pi}} \cdot \exp\left[-\frac{1}{2} \cdot \left(\frac{y - R_p}{\Delta R_p}\right)^2\right] \cdot \left(1 - \operatorname{erf}\left(\frac{x - a}{\Delta R_x}\right)\right)$$

$$x_{\text{low}} := 0 \quad x_{\text{high}} := 10^{-4}$$

$$y_{\text{low}} := 0 \quad y_{\text{high}} := 10^{-4}$$

$$x_n := 60 \quad y_n := 60$$

$$i := 0 .. x_n - 1 \quad x_{\text{ind}_i} := x_{\text{low}} + i \cdot \frac{x_{\text{high}} - x_{\text{low}}}{x_n - 1}$$

$$j := 0 .. y_n - 1 \quad y_{\text{ind}_j} := y_{\text{low}} + j \cdot \frac{y_{\text{high}} - y_{\text{low}}}{y_n - 1}$$

$$M_{i,j} := C(x_{\text{ind}_i}, y_{\text{ind}_j})$$

Рис. 53. Методика построения распределения примеси по Гауссу

Для упрощения расчёта уравнение для  $C(x, y)$  из рис. 53 можно преобразовать для расчёта контуров двумерного профиля распределения концентрации примеси (рис. 54, 55).

Расчет  $R_x=f(E)$

|    |   |                  |    |
|----|---|------------------|----|
| Sb | $f1(E) := 1.07 + 0.124 \cdot E$                               | $f1(E) = 13.47$  | HM |
| As | $f2(E) := -0.25 + 0.186 \cdot E$                              | $f2(E) = 18.35$  | HM |
| P  | $f3(E) := 0.48 \cdot E$                                       | $f3(E) = 48$     | HM |
| B  | $f4(E) := 2.09 + 1.48 \cdot E - 2.72 \cdot 10^{-3} \cdot E^2$ | $f4(E) = 122.89$ | HM |

$$E := 0, 1 \dots 250$$

Рис. 54. Расчёт бокового пробега ионов для различных примесей (*Sb*, *As*, *P*, *B*) как функция от энергии ионов *E*

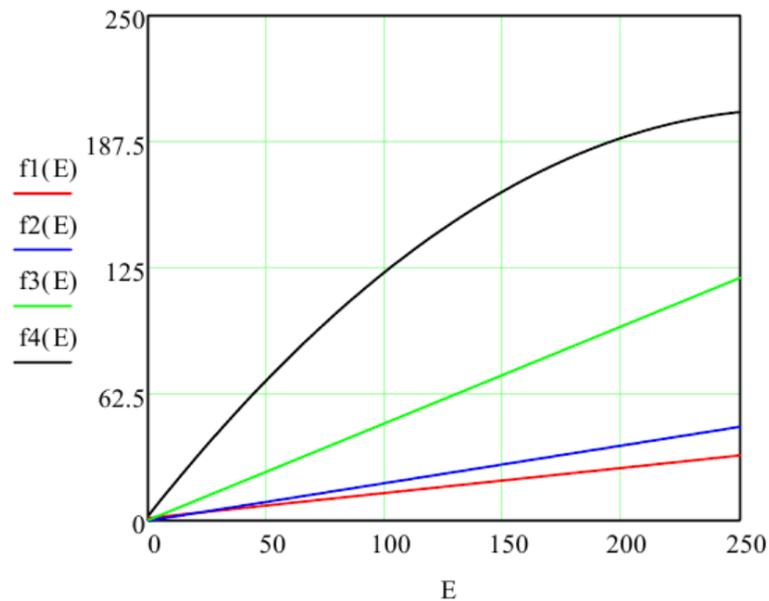


Рис. 55. Боковой пробег ионов для различных примесей (*Sb*, *As*, *P*, *B*) как функция от энергии ионов *E*

Концентрационная зависимость может быть построена с любым шагом, начиная с начального значения концентрации ( $N = N_0$ ) – рис. 56 - 58.

$$f1(x, No) := Rp + \sqrt{2\Delta Rp^2 \cdot \ln \left[ \frac{N}{No \cdot \Delta Rp \cdot \sqrt{2\pi}} \cdot \left( 1 - \operatorname{erf} \left( \frac{x-a}{\Delta Rx} \right) \right) \right]}$$

$$f2(x, No) := Rp - \sqrt{2\Delta Rp^2 \cdot \ln \left[ \frac{N}{No \cdot \Delta Rp \cdot \sqrt{2\pi}} \cdot \left( 1 - \operatorname{erf} \left( \frac{x-a}{\Delta Rx} \right) \right) \right]}$$

$$L(x) := \begin{cases} 0 & \text{if } x > a \\ -1 & \text{otherwise} \end{cases}$$

Рис. 56. Преобразование уравнения бокового пробега ионов для расчёта контуров 2D-концентрации

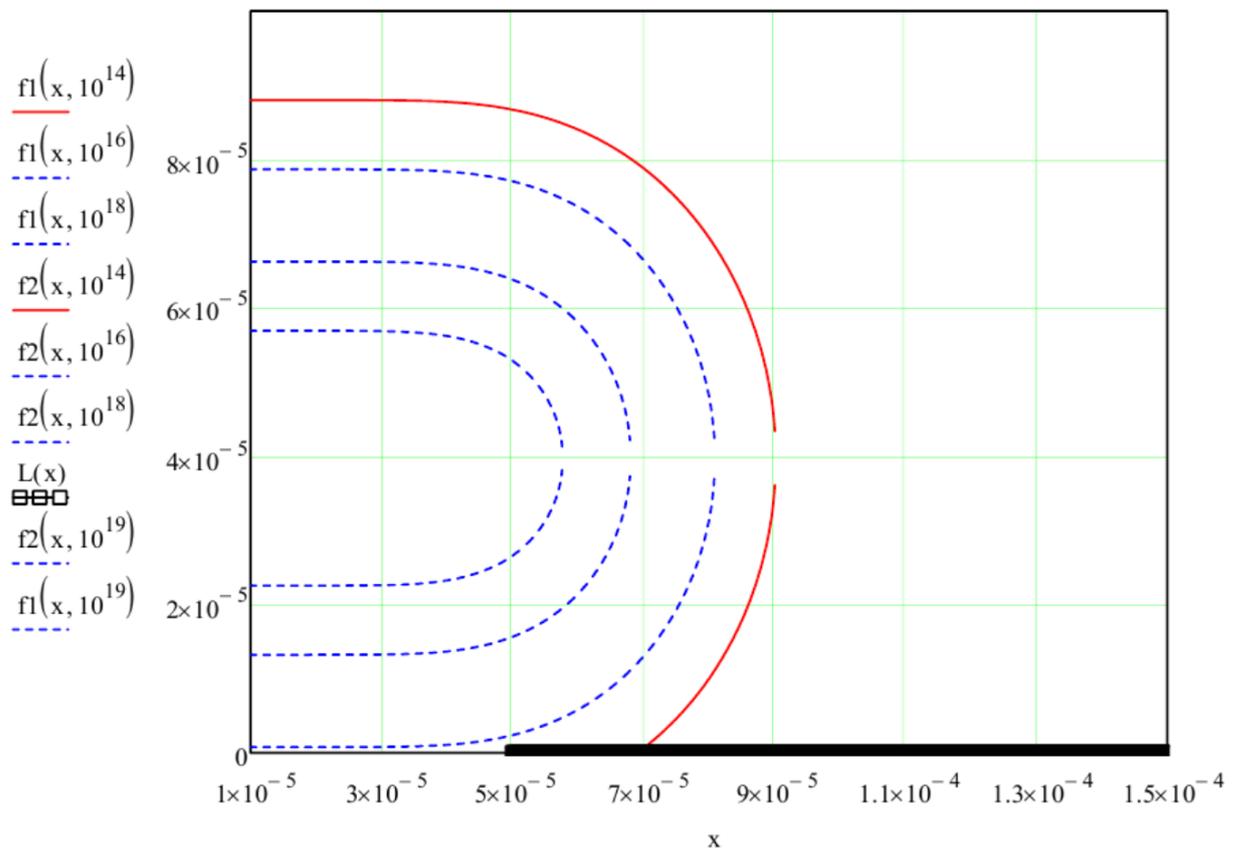


Рис. 57. Контуров 2D-концентрации бокового пробега ионов

$x1$  - максимальный уход  $p$ - $n$  перехода за край маски ( $y=Rp$ ).

$$C1(x) := \frac{N}{N_0 \cdot \Delta Rp \cdot \sqrt{2\pi}} \cdot \left( 1 - \operatorname{erf}\left(\frac{x-a}{\Delta Rx}\right) \right) - 1$$

$x := a$

$$x1 := \operatorname{root}(C1(x), x) \quad x1 = 9.021 \times 10^{-5}$$

$$\Delta x1 := x1 - a \quad \Delta x1 = 4.021 \times 10^{-5} \quad C(x1, Rp) = 1 \times 10^{14}$$

$x2$  - положение  $p$ - $n$  перехода под маской ( $y=0$ )

$$C2(x) := \frac{1}{N_0} \cdot \left[ \frac{N}{\Delta Rp \cdot \sqrt{2\pi}} \cdot \exp\left[-\frac{1}{2} \cdot \left(\frac{Rp}{\Delta Rp}\right)^2\right] \cdot \left( 1 - \operatorname{erf}\left(\frac{x-a}{\Delta Rx}\right) \right) \right] - 1$$

$x := a$

$$x2 := \operatorname{root}(C2(x), x) \quad x2 = 6.909 \times 10^{-5}$$

$$\Delta x2 := x2 - a \quad \Delta x2 = 1.909 \times 10^{-5} \quad C(x2, 0) = 1 \times 10^{14}$$

Рис. 58. Расчёт расположения параметров  $p$ - $n$  перехода

Глубину залегания примеси можно определить в любой точке окна, например в его центре ( $x = 0$ ,  $N = N_0$ ), рис. 59.

Максимальная концентрация примеси в центре окна и глубина её залегания определяются в точке  $x = 0$  при значении проективного пробега  $Rp$ .

Положение  $p$ - $n$  перехода под краем маски ( $y = 0$ ,  $N = N_0$ ) и на глубине с максимальной концентрацией примеси ( $y = Rp$ ,  $N = N_0$ ) может быть рассчитано численно с использованием функции «*root*» в математическом пакете или с использованием метода Ньютона на языках программирования.

Глубина р-п перехода в центре окна ( $x=0$ ).

$$f_1(0, N_0) = 8.796 \times 10^{-5} \quad \text{см}$$

Максимальная концентрация примеси в легированной области ( $x=0, y=R_p$ ).

$$C(0, R_p) = 5.305 \times 10^{19} \quad 1/\text{см}^3$$

Максимальный уход р-п перехода за край маски ( $y=R_p$ ).

$$\Delta x_1 := x_1 - a \quad \Delta x_1 = 4.021 \times 10^{-5} \quad \text{см}$$

Уход р-п перехода за край маски ( $y=0$ ).

$$\Delta x_2 := x_2 - a \quad \Delta x_2 = 1.909 \times 10^{-5} \quad \text{см}$$

Проверка.

$$C(x_1, R_p) = 1 \times 10^{14}$$

$$C(x_2, 0) = 1 \times 10^{14}$$

Рис. 59. Расчёт параметров легированных областей структуры

## ИСПОЛЬЗОВАНИЕ МЕТОДА НЬЮТОНА НАХОЖДЕНИЯ КОРНЕЙ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Для определения корней уравнений, например при расчёте процесса диффузии, в математическом пакете принято использовать встроенную функцию «*root*», для которой задается первое приближение, а программа сама находит корень.

При разработке программы, ищущей корни уравнений, можно использовать различные методы (метод Ньютона, метод бисекций, метод хорд, метод простых итераций, ...). При моделировании технологических процессов наиболее удобным показал себя метод Ньютона, для которого задаётся функция и рассчитывается её первая производная.

Метод Ньютона (метод касательных) – это итерационный численный метод нахождения корня (пересечения функции с осью абсцисс  $OX$ ) заданной функции. Метод был впервые предложен английским физиком, математиком и астрономом Исааком Ньютоном.

В этом методе задача поиска корня функции  $f(x)$  сводится к определению точки пересечения с осью абсцисс касательной, построенной к графику функции  $f(x)$  (рис. 60).

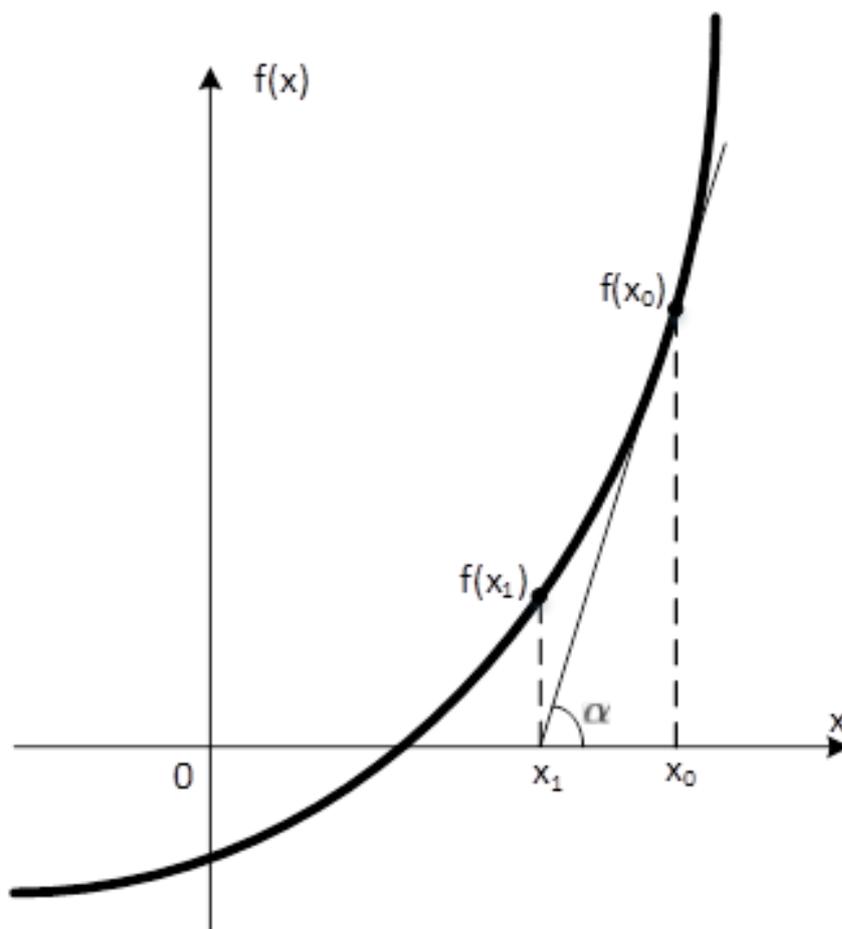


Рис. 60. График изменения функции  $f(x)$

Проведенная в любой точке касательная линия к графику функции  $f(x)$  определяется производной данной функции в рассматриваемой точке, которую можно найти как тангенс угла  $\alpha$  ( $\operatorname{tg}(\alpha) = f'(x_0)$ ).

Точка пересечения касательной с осью абсцисс определяется по правилу соотношения сторон в прямоугольном треугольнике: тангенс угла  $\alpha$  в прямоугольном треугольнике – это отношение противолежащего катета к прилежащему катету прямоугольного треугольника. В результате на каждом шаге смещения строится касательная к графику функции в точке очередного приближения  $x_i$ . Точка пересечения касательной с осью абсцисс будет являться следующей точкой приближения  $x_{i+1}$ .

Расчет приближенного значения корня на  $i$ -итерации сводится к формуле (9)

$$x_{i+1} = x_i - \frac{1}{f'(x_i)} \cdot f(x_i). \quad (9)$$

Наклон прямой подбирается на каждом шаге, однако алгоритм не учитывает кривизну графика и, следовательно, в процессе расчета остается не известно, в какую сторону график будет отклоняться.

Условием окончания итерационного процесса является выполнение условия (10)

$$|x_{i+1} - x_i| \leq \varepsilon, \quad (10)$$

где  $\varepsilon$  – допустимая погрешность определения корня.

Метод обладает квадратичной сходимостью, то есть число верных знаков в приближённом значении удваивается с каждой итерацией.

Ниже приведён алгоритм расчёта уравнений методом Ньютона. Программа состоит из отдельных блоков. В основном коде выбирается метод расчёта, вводится требуемая точность *eps* и задаётся начальное приближение *x1*. В комментариях в начале кода в качестве подсказки приведены корни уравнений, используемых в примере.

```
using System;
using System.Diagnostics;

namespace osnovnoy
{
    internal class osnovnoy
    {
        static void Main(string[] args)
        {
            char log;
            do
            {
                Console.WriteLine("Выберите интересующий раздел: \n\t1 - Метод Ньютона\n\t2 - Метод бисекций\n\t3
- Метод хорд\n\t4 - Метод простых итераций");
                char op = Char.Parse(Console.ReadLine());
                switch (op)
                {
                    // (1) при x = 0.73    f(x) = x - Math.Cos(x) = 0
                    // (2) при x = 5.7    f(x) = Math.Pow(x, 3) - 18 * x - 83 = 0
                    // (3) при x = 0      f(x) = Math.Sin(x) - 2 * x = 0
                    case '1':
                        // Newton
                        Console.Write("введи точность eps = ");
                        double eps = Double.Parse(Console.ReadLine());
                        Console.Write("Введи x1 = ");
                        double x = Double.Parse(Console.ReadLine());

                        int count = 0;
                        double x1;
                        x1 = x - (Newton.f(x) / Newton.df(x));
                        Console.WriteLine(x1);
                        while (Math.Abs(Newton.f(x)) >= eps)
                        {
                            count++;
                            x1 = x - (Newton.f(x) / Newton.df(x));
                            x = x1;

                            Console.WriteLine("Y = " + Newton.f(x));
                            Console.WriteLine("Y' = " + Newton.df(x));
                            Console.WriteLine("x = " + x);
                            Console.WriteLine("x1 = " + x1 + "\n");
                        }
                        Console.WriteLine();
                        Console.WriteLine("Корень: " + x);
                        Console.WriteLine("Количество итераций: " + count);
                    }
                }
                break;
            }
        }
    }
}

// остальные методы в данном коде не рассматриваются
```

// в примере рассматривается решение только методом Ньютона, который является наиболее удобным для нахождения корней уравнений моделируемых технологических процессов

```
        default:
            Console.WriteLine("Данного блока не существует");
            break;
        }
        Console.WriteLine("Хотите остаться в программе? Y/N");
        log = Char.Parse(Console.ReadLine());
    }
    while (log == 'Y' | log == 'y' | log == 'N' | log == 'n');
}
}
```

Присоединяемый в проект класс с двумя методами. Первый «*public static double f(double x) { }*» включает в себя рассчитываемую функцию. Второй «*public static double df(double x) { }*» - её производную.

```
using System;

namespace osnovnoy
{
    public class Newton
    {
        public static double f(double x)
        {
            return x - Math.Cos(x);
            // return Math.Pow(x, 3) - 18 * x - 83;
            // return Math.Sin(x) - 2 * x;
            //return Math.Sin(x) * Math.Sin(x) + Math.Sin(x);
        }
        public static double df(double x)
        {
            return Math.Sin(x) + 1;
            // return 3 * x * x - 18;
            // return Math.Cos(x) - 2;
            //return Math.Cos(x) + 2 * Math.Cos(x) * Math.Sin(x);
        }
    }
}
```

На рис. 61 показаны работа программы и рассчитанный корень уравнения  $x = \text{Math.Cos}(x)$ .

```
D:\Языки программирования\C#\repos\2022 Расчёт нелинейных
Выберите интересующий раздел:
    1 - Метод Ньютона
    2 - Метод бисекций
    3 - Метод хорд
    4 - Метод простых итераций
1
введи точность eps = 0,0001
Введи x1 = 0,5
0,7552224171056364
Y = 0,027103311857467283
Y' = 1,6854506317544726
x = 0,7552224171056364
x1 = 0,7552224171056364
Y = 9,461538061772412E-05
Y' = 1,6736538107583572
x = 0,7391416661498792
x1 = 0,7391416661498792
Корень: 0,7391416661498792
Количество итераций: 2
Хотите остаться в программе? Y/N
-
```

Рис. 61. Расчёт уравнения  $x = \text{Math.Cos}(x)$  методом Ньютона

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Chart Класс. - Электрон. дан. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.forms.datavisualization.charting.chart?view=netframework-4.8.1>. – Текст: электронный.
2. Графики С#. – Электрон. дан. – Режим доступа: <https://itarticle.ru/graphics-csharp/>. – Текст: электронный.
3. WorksheetFunction.Erf(Object, Object) Метод. – Электрон. дан. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/api/microsoft.office.interop.excel.worksheetfunction.erf?view=excel-pia>. – Текст: электронный.
4. Функция ошибок. – Электрон. дан. – Режим доступа: [https://ru.wikipedia.org/wiki/Функция\\_ошибок](https://ru.wikipedia.org/wiki/Функция_ошибок). – Текст: электронный.
5. Вычисление определённых интегралов: базовые алгоритмы. – Электрон. дан. – Режим доступа: <https://habr.com/ru/articles/420867/>. – Текст: электронный.
6. Метод Ньютона. – Электрон. дан. – Режим доступа: [https://ru.wikipedia.org/wiki/Метод\\_Ньютона](https://ru.wikipedia.org/wiki/Метод_Ньютона). – Текст: электронный.
7. Никоненко В. А. Математическое моделирование технологических процессов. Моделирование в среде MathCAD. Практикум / В. А. Никоненко. - М: МИСиС, 2001. – 48 с.: ил. – Текст: непосредственный.

## ОГЛАВЛЕНИЕ

|  |    |
|--|----|
| Введение.....  | 3  |
| Задание на практические работы.....  | 3  |
| Задание на лабораторные работы.....  | 3  |
| Методы построения стандартных графиков функций.....  | 4  |
| Моделирование функции ошибок $\text{erf}(x)$ .....   | 6  |
| Использование метода Симпсона для решения интегралов.....  | 9  |
| Технология построения графиков через код.....  | 11 |
| Задание 1. Моделирование технологического процесса термического окисления кремния.....                       | 14 |
| Задание 2. Расчёт параметров распыления с использованием кольцевого испарителя (магнетронное напыление)..... | 16 |
| Задание 3. Методы моделирования технологического процесса диффузии.....                                      | 21 |
| Задание 4. Основные этапы формирования биполярного транзистора....   | 32 |
| Задание 5. Моделирование технологического процесса ионной имплантации.....                                   | 38 |
| Задание 6. Расчёт параметров двумерного распределения ионов с заходом под край маски.....                    | 44 |
| Использование метода Ньютона нахождения корней нелинейных уравнений.....                                     | 50 |
| Библиографический список.....  | 55 |

# **МОДЕЛИРОВАНИЕ МИКРОЭЛЕКТРОННЫХ ПРОЦЕССОВ И ПРИБОРОВ НА ПК**

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

к выполнению лабораторных и практических работ  
по дисциплине «Информационные технологии в электронике»  
для студентов направления  
11.03.04 «Электроника и наноэлектроника»  
всех форм обучения

**Составитель**  
**Плотникова Екатерина Юрьевна**

В авторской редакции

Подписано к изданию 17.01.2024.  
Уч.-изд. л. 3,1.

ФГБОУ ВО «Воронежский государственный технический университет»  
394006 Воронеж, ул. 20-летия Октября, 84