МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«Витебский государственный технологический университет»

СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ОБОРУДОВАНИЕМ

Методические указания по выполнению лабораторных работ
для студентов специальности
1-36 01 01 «Технология машиностроения»

Витебск
2023

УДК 621.9

Составитель:

А. Л. Климентьев

Рекомендовано к изданию редакционно-издательским советом УО «ВГТУ», протокол № 10 от 29.06.2023.

Системы управления технологическим оборудованием : методические указания по выполнению лабораторных работ / сост. : А. Л. Климентьев. — Витебск : УО «ВГТУ», 2023. — 55 с.

Методические указания являются руководством по выполнению лабораторных работ по учебной дисциплине «Системы управления технологическим оборудованием». Изложены содержание, методика и порядок выполнения работ. Также приведен краткий справочник по основным командам управляющих программ и описан пример создания управляющей программы. Предназначены для студентов специальности 1-36 01 01 «Технология машиностроения».

Издание в электронном виде расположено в репозитории библиотеки УО «ВГТУ».

УДК 621.9

© УО «ВГТУ», 2023

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА 1. РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ НА БАЗЕ ЛОГИЧЕСКОГО АВТОМАТА……...4

1.1 Элементы теории..... 4

1.2 Содержание работы 10

1.3 Варианты заданий..... 11

ЛАБОРАТОРНАЯ РАБОТА 2. РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ НА БАЗЕ ЛОГИЧЕСКОГО АВТОМАТА В БАЗИСЕ РЕЛЕЙНО-КОНТАКТНЫХ ЭЛЕМЕНТОВ 13

2.1 Элементы теории..... 13

2.2 Содержание работы 19

2.3 Варианты заданий..... 19

ЛАБОРАТОРНАЯ РАБОТА 3. ПРОГРАММИРОВАНИЕ МИКРОПРОЦЕССОРНОГО КОМПЛЕКТА..... 20

3.1 Элементы теории..... 20

3.2 Содержание работы 22

3.3 Варианты заданий..... 23

3.4 Микропроцессорная лаборатория «Микролаб КР580ИК80»..... 25

ЛАБОРАТОРНАЯ РАБОТА 4. ПРОГРАММИРОВАНИЕ ТЕХНОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ С ПРОГРАМНЫМ УПРАВЛЕНИЕМ 29

4.1 Элементы теории..... 29

4.2 Порядок разработки управляющей программы 33

4.3 Содержание работы 46

СПИСОК РЕКОМЕНДУЕМЫХ ИСТОЧНИКОВ 47

ПРИЛОЖЕНИЕ А. Краткий справочник по основным командам..... 48

Лабораторная работа 1
РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ НА БАЗЕ ЛОГИЧЕСКОГО АВТОМАТА

Цель работы: изучение методов синтеза комбинационных логических схем путем минимизации переключательных функций и получение навыков структурного синтеза логических автоматов без памяти.

1.1 Элементы теории

Комбинационные логические схемы (КЛС) (автоматы без памяти) представляют собой частный случай цифровых автоматов, также являются составной частью структурной схемы цифровых автоматов с памятью. Синтез КЛС производится на основе законов алгебры-логики (и следствий из них) с помощью различных методов минимизации.

Основные положения и законы алгебры-логики (булевой алгебры).

Логические переменные (ЛП) — переменные, принимающие только два значения — 0 или 1. Другие названия логических переменных — двоичные или булевы переменные.

Логические переменные можно разделить на простые и сложные.

Простые ЛП — независимые переменные.

Сложные ЛП — зависимые переменные или функции ЛП, принимающие также значения 0 или 1, называются двоичными или переключательными функциями (ПФ). Для ПФ справедлив принцип суперпозиции.

Обычно ПФ является функцией конечного числа переменных аргументов, обозначаемых x_i, $i = \overline{1,n}$, а сама ПФ обозначается как $y = f(x_i, i = \overline{1,n})$.

Конечность числа переменных (n) и конечность принимаемых ими значений ($k = 2$) определяет конечность значений ПФ (m), которая определяется количеством сочетаний любого из 2 значений каждой из n переменных. Каждое сочетание (комбинация) называется набором, а количество наборов определяется как $m = 2^n$.

Значение, принимаемое ПФ на конкретном наборе, называется *значением истинности*, а таблица, содержащая значения истинности ПФ на всех наборах, — *таблицей истинности* (ТИ).

ТИ является исходной формой для аналитической записи ПФ. Аналитическая запись ПФ предполагает (аналогично обычной математике) набор элементарных операций, называемых логическими связями, которые определяются как ПФ 2 переменных, а затем на основе принципа суперпозиции позволяют построить сколь угодно сложную ПФ.

Для двух переменных возможны $2^2 = 4$ набора, на них можно определить $2^4 = 16$ различных логических связей (ЛС), т. е. ЛС, которые на основе

принципа суперпозиции позволяют построить ПФ любой сложности, составляют функционально-полный набор ЛС (ФПН), определяемый совокупностью свойств, входящих в него ЛС.

Наиболее широкое распространение получил основной ФПН (ОФПН), включающий три ЛС из всего набора:

1) f_1 — *отрицание* (операция НЕ, логическая инверсия);
2) f_2 — *конъюнкция* (операция И, логическое умножение);
3) f_3 — *дизъюнкция* (операция ИЛИ, логическое сложение).

Таблица истинности операций ОФПН и их аналитическая запись.

x_1	x_2	f_1	f_2	f_3	f_4	f_5
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	1	0	1	1	0	0

$f_1 = f(x_1) = \overline{x_1}$;

$f_2 = f(x_1, x_2) = x_1 x_2 = x_1 \,\&\, x_2 = x_1 \wedge x_2$;

$f_3 = f(x_1, x_2) = x_1 + x_2 = x_1 \vee x_2$.

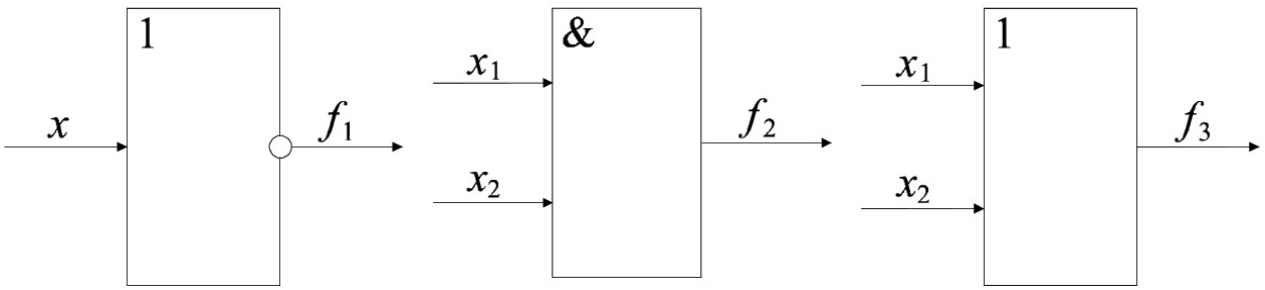
ОФП — набор избыточный, т. к. функционально-полными являются наборы ЛС (f_1, f_2) и (f_1, f_3). ФПН представляют также функции f_4 и f_5:

– f_4 — *операция Шеффера*, операция И-НЕ, $f_4 = \overline{x_1 x_2} = \overline{x_1 \,\&\, x_2} = x_1 \wedge x_2 = f_2$.

– f_5 — *операция Пирса*, операция ИЛИ-НЕ, $f_5 = \overline{x_1 + x_2} = \overline{x_1 \vee x_2} = \overline{f_3}$.

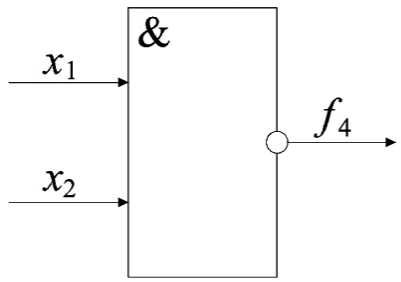
Свойства операций ОФПН и их обозначения на функциональных схемах.

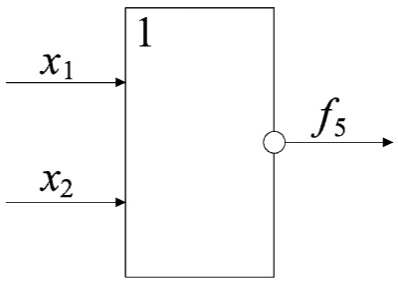
Отрицание	Конъюнкция	Дизъюнкция
$\overline{\overline{x}} = x$ $\overline{\equiv x} = \overline{x}$ Если $x_1 = x_2$, то $\overline{x_1} = \overline{x_2}$	$xx...x = x$ $1 \cdot x = x$ $0 \cdot x = 0$ $x \cdot \overline{x} = 0$	$x + x + ... + x = x$ $1 + x = 1$ $0 + x = x$ $x + \overline{x} = 1$



Обозначение на функциональных схемах операций Шеффера и Пирса.

Операция Шеффера (И-НЕ)	Операция Пирса (ИЛИ-НЕ)





Законы алгебры логики

	Переместительный	Сочетательный
И	$x_1x_2 = x_2x_1$	$x_1x_2x_3 = x_1(x_2x_3)$
ИЛИ	$x_1 + x_2 = x_2 + x_1$	$x_1 + x_2 + x_3 = x_1 + (x_2 + x_3)$
	Распределительный	Инверсия
И	$x_1(x_2 + x_3) = x_1x_2 + x_1x_3$	$\overline{x_1x_2} = \overline{x_1} + \overline{x_2}$
ИЛИ	$x_1(x_2x_3) = (x_1 + x_2)(x_1 + x_3)$	$\overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2}$

Следствия:

а) правило *старшинства* логических операций:

- отрицание,
- конъюнкция,
- дизъюнкция;

б) правило *склеивания* $x_1x_2 + x_1\overline{x_2} = x_1$; $x_1x_2x_3 = x_1x_2\overline{x_3} = x_1x_2$;

в) правило *поглощения* $x_1 + x_1x_2 = x_1$.

Конъюнкция (дизъюнкция) называются *элементарными*, если выполняются над конечным числом одиночных аргументов или их отрицаний. Например: $x_1\overline{x_2x_3}x_4$, $x_1 + \overline{x_2} + \overline{x_3} + x_4$.

Количество аргументов в элементарной конъюнкции (дизъюнкции) — ранг *r*.

Две элементарные конъюнкции (дизъюнкции) с одинаковым рангом называются *соседними*, если отличаются знаком отрицания только одного из аргументов. Например: $x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4$, $x_1 \cdot x_2 \cdot \overline{x_3} \cdot x_4$; $(x_1 + \overline{x_2} + x_3 + \overline{x_4}$, $x_1 + x_2 + x_3 + \overline{x_4})$, к ним применимо правило склеивания (они склеиваются).

Если $r = n$, где n — количество аргументов в ПФ $f = f(x_i, i = \overline{1,n})$, то элементарная конъюнкция (дизъюнкция) называется *конституантой* единицы (нуля).

Очевидно, что конституанта сохраняет единицу (нуль) только на одном наборе значений, входящих в нее аргументов из таблицы истинности. На этом правиле основаны способы (формы) аналитической записи переключательных функций по таблице истинности.

Совершенная дизъюнктивная нормальная форма (СДНФ).

Для наборов, дающих единицу переключательной функции, записываются конституанты единицы, в которых переменные, имеющие на наборе нулевое значение, берутся с отрицанием (например, для набора $x_1 = 0$, $x_2 = 1$, $x_3 = 0$ имеем конституанту $\overline{x_1} \cdot x_2 \cdot \overline{x_3}$) и объединяются знаком дизъюнкции.

Совершенная конъюнктивная нормальная форма (СКНФ).

Симметрично конституанты нуля, составленные для наборов, дающих нулевое значение ПФ, объединяются знаком конъюнкции. С отрицанием в конституантах нуля берутся переменные, имеющие на наборе единичное значение (например, для набора $x_1 = 0$, $x_2 = 1$, $x_3 = 0$ имеем конституанту $x_1 + \overline{x_2} + x_3$).

Пример.

СДНФ для рассматриваемых функций

$f_2 = x_1 x_2$;

$f_3 = \overline{x_1} x_2 + x_1 \overline{x_2} + x_1 x_2$;

$f_4 = \overline{x_1}\,\overline{x_2} + x_1 \overline{x_2} + \overline{x_1} x_2$;

$f_5 = \overline{x_1}\,\overline{x_2}$;

для них же СКНФ

$f_2 = (x_1 + x_2)(x_1 + \overline{x_2})(\overline{x_1} + x_2)$;

$f_3 = x_1 + x_2$;

$f_4 = \overline{x_1} + \overline{x_2}$;

$f_5 = (x_1 + \overline{x_2})(\overline{x_1} + x_2)(\overline{x_1} + \overline{x_2})$.

В результате преобразований СДНФ (СКНФ) на основании законов алгебры логики ранг некоторых конъюнкций (дизъюнкций), входящих в форму может быть понижен. Такие формы называются, соответственно, дизъюнктивной (ДНФ) и конъюнктивной (КНФ) нормальной формой.

Минимизация переключающих функций.

Одним из критериев минимизации комбинационных логических схем является критерий минимума аргументов, записанных в форме, иначе критерий минимума входов (логической переменной соответствует вход логической схемы), а также минимума элементарных конъюнкций (дизъюнкций) в форме.

Минимизация производится путем:

1) многократного попарного склеивания конъюнкций (дизъюнкций), в результате чего получается Д(К)НФ, называемая *сокращенной*, а ее составляющие, попарно несклеиваемые, — простыми *импликантами*;

2) последующего исключения лишних импликант, удаление которых не влияет на истинность ПФ, в результате чего получается «тупиковая форма», дальнейшая минимизация которой в рамках нормальных форм невозможна.

Дальнейшая минимизация возможна путем применения законов инверсии и распределительного, нарушающих ПФ. Этот путь не формализован и основан на интуиции и опыте разработчика (интуитивно-эвристический путь).

Одним из методов, включающим в себя два первых этапа является табличный метод (метод диаграмм Вейча – Карно). Он использует специальные таблицы — диаграммы Вейча – Карно, заполненные в соответствии с таблицей истинности, преобразуемые по специальным правилам.

Таблица прямоугольной формы состоит из клеток, количество которых определяется количеством переменных в ПФ и равно $2n$. Для четных n каждая сторона прямоугольника образуется из $2^{n/2}$ клеток, а для нечетных n — одна сторона из $2^{(n-1)/2}$; а другая из $2^{(n+1)/2}$ клеток.

Каждой малой клетке соответствует свой набор значений переменных из таблицы истинности (количество наборов $m = 2n$) и располагаются наборы в клетках диаграммы так, чтобы в соседних клетках находились соседние наборы. Крайние на верхнем и нижнем, а также на правом и левом полях клетки также считаются соседними.

Диаграммы Вейча – Карно.

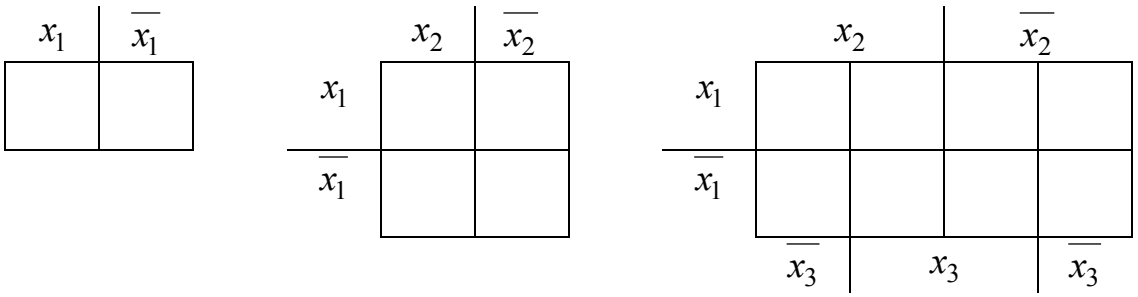
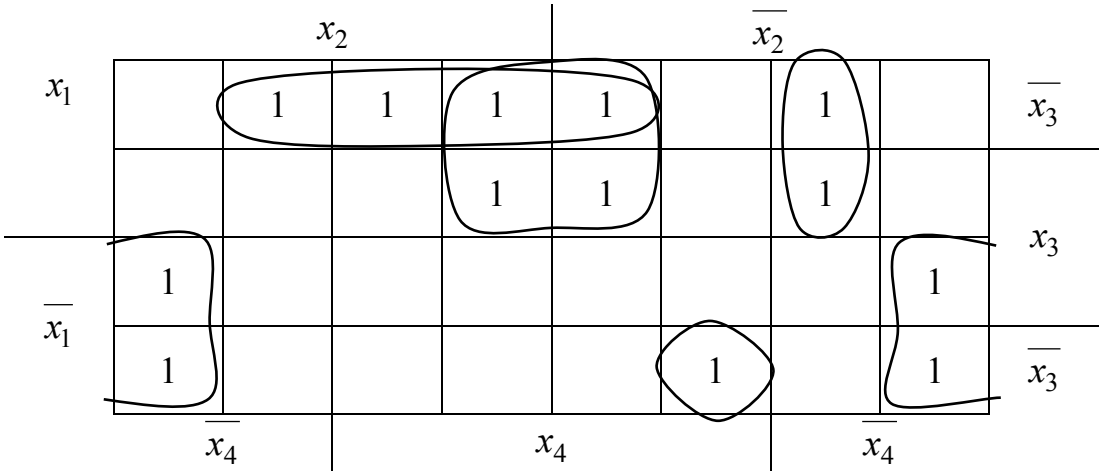


Диаграмма Вейча – Карно заполняется значениями таблицы истинности, соответствующими наборам, стоящими в клетках. Причем, если минимизируется СДНФ, то записываются только 1, а нулевые клетки остаются пустыми. Для СКНФ записываются только нули ПФ.

Некоторые единицы (нули) в таблице стоят в соседних клетках. Любое 2^S число соседних единиц (нулей) может быть обведено контуром.

Например,



Количество контуров определяет количество простых импликант в полученной из диаграммы «тупиковой форме».

2^s соседних клеток, составляющих контур, соответствуют одной импликанте, ранг которой меньше ранга n конституанты на s единиц. Это означает что чем больше клеток в контуре, тем проще член тупиковой формы, полученных из него; чем меньше контуров, тем меньше импликант содержит «тупиковая форма».

Если диаграмма для n переменных содержит q контуров, в каждом из которых 2^{s_j}, $j = \overline{1,q}$ клеток, то тупиковая форма позволит создать схему, состоящую из логических элементов с суммарным количеством входов:

$$C = \sum_{j=1}^{q} (n - s_j) + q .$$

Величина C является *ценой схемы*.

Таким образом, не выписывая «тупиковой формы», можно сравнить по цене диаграммы, составленные соответственно для СДНФ и СКНФ и выбрать минимальную из них.

Выписыванием «тупиковой формы» заканчивается процесс формальной минимизации.

Структурный синтез автоматов без памяти.

Структурный синтез автоматов без памяти состоит из 4-х этапов:

1. Составление таблицы истинности синтезируемого автомата на основе анализа его назначения и (словесного) описания принципа работы. Таблица истинности заполняется путем выяснения отклика ПФ на каждый набор переменных.

2. Получение математической формулы для логической функции, возможно оптимизированной по какому-либо критерию, например, по минимуму цены схемы.

3. Выбор базиса логических элементов и приведение минимальной формы к этому базису. Выполнение этого этапа зависит от наличия логических элементов допустимой, например, указанной в технических условиях, физической природы. В частности, при использовании логических микросхем наиболее легко реализуются базисом И-НЕ и ИЛИ-НЕ.

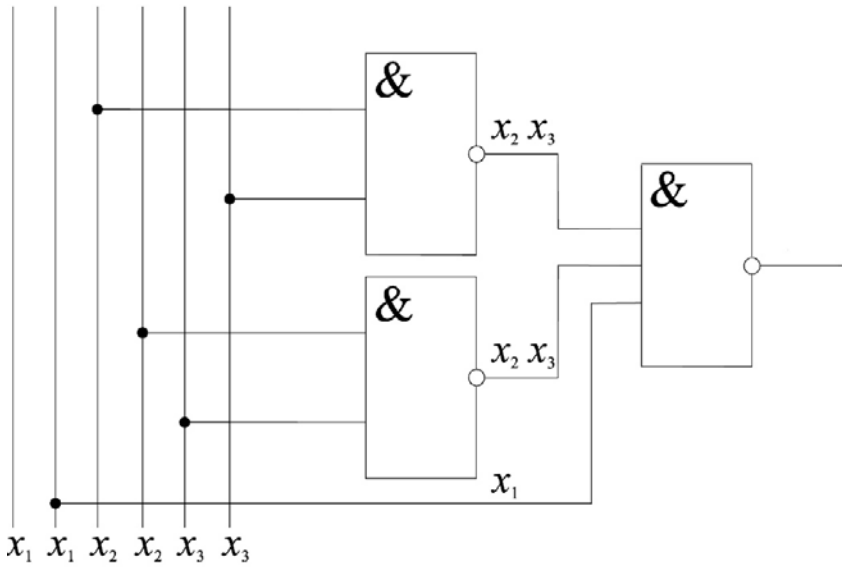
Приведение минимальной формы к этим базисам производится путем двойного отрицания с последующим преобразованием по закону отрицания.

Например,

$$f = x_1 + x_2\overline{x_3} + \overline{x_2}x_3 = \overline{\overline{x_1 + x_2\overline{x_3} + \overline{x_2}x_3}} = \overline{x_1 \cdot \overline{x_2\overline{x_3}} \cdot \overline{\overline{x_2}x_3}}\,.$$

4. Составление требуемой схемы автомата: функциональной или принципиальной.

Например, для рассматриваемой формулы принципиальная схема автомата



В схеме использован базис И-НЕ.

1.2 Содержание работы

В процессе работы необходимо решить поставленную задачу в соответствии с индивидуальным заданием и оформить отчет с выводами.

Порядок выполнения работы

1. Изучить теоретическую часть.
2. По заданному словесному описанию работы логического автомата составить таблицу истинности (также возможно задание в виде таблицы истинности).
3. По таблице истинности:

– записать аналитическое выражение в СДНФ и СКНФ булевой функции;

– минимизировать выражение ПФ в СДНФ (методом диаграмм Вейча – Карно), определить цену схемы;

– то же для СКНФ.

4. Для МДНФ построить релейно-контактную схему, реализующую ПФ в СДНФ и СКНФ.

5. Привести выражение МДНФ, реализующую ПФ в СДНФ и СКНФ, к базисам Шеффера и Пирса (соответственно).

6. Построить логическую схему в базисе Шеффера и логическую схему в базисе Пирса.

1.3 Варианты заданий

Таблица 1.1 — Варианты заданий

Состояния входов			Состояния выхода (по вариантам)									
x_1	x_2	x_3	1	2	3	4	5	6	7	8	9	10
0	0	0	1	1	1	1	1	0	1	0	0	1
0	0	1	1	0	1	0	0	1	0	0	1	0
0	1	0	1	0	0	0	1	1	0	1	1	1
0	1	1	0	0	1	1	0	1	1	1	0	0
1	0	0	1	1	0	1	0	1	0	0	0	0
1	0	1	0	1	0	0	1	0	1	0	0	0
1	1	0	0	0	0	1	0	0	0	1	1	1
1	1	1	0	0	1	0	0	0	1	0	0	1

Состояния входов			Состояния выхода (по вариантам)									
x_1	x_2	x_3	11	12	13	14	15	16	17	18	19	20
0	0	0	0	1	1	0	0	1	0	1	0	0
0	0	1	1	0	0	0	1	0	0	1	0	1
0	1	0	1	0	1	0	1	0	0	1	1	0
0	1	1	1	1	0	0	0	1	1	0	1	1
1	0	0	0	1	0	0	0	1	1	1	0	1
1	0	1	0	0	1	1	0	0	0	0	0	0
1	1	0	0	1	1	1	0	0	1	1	1	0
1	1	1	1	1	0	1	1	0	1	0	1	0

Окончание таблицы 1.1

Состояния входов			Состояния выхода (по вариантам)									
x_1	x_2	x_3	21	22	23	24	25	26	27	28	29	30
0	0	0	0	0	0	1	0	1	0	0	1	0
0	0	1	1	1	0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0	1	0	0	1	0
0	1	1	1	0	0	1	0	0	1	0	1	1
1	0	0	1	1	0	1	1	1	0	1	0	0
1	0	1	0	0	1	1	0	0	1	1	0	1
1	1	0	1	0	0	0	1	1	0	1	0	1
1	1	1	0	1	1	1	1	0	1	1	0	1

Состояния входов				Состояния выхода (по вариантам)									
x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
0	0	0	0	0	0	1	0	1	0	0	1	1	–
0	0	0	1	0	0	0	0	0	–	1	1	–	1
0	0	1	0	0	1	1	1	1	1	0	1	1	0
0	0	1	1	–	0	0	1	–	0	–	1	0	0
0	1	0	0	1	0	–	0	0	1	0	0	0	–
0	1	0	1	0	–	0	–	1	0	0	–	–	1
0	1	1	0	1	0	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	0	–	0	1
1	0	0	0	1	1	–	0	0	0	1	1	0	0
1	0	0	1	–	0	1	–	–	1	–	0	1	1
1	0	1	0	–	1	1	1	–	1	0	1	–	0
1	0	1	1	1	–	–	–	1	–	1	–	0	–
1	1	0	0	0	1	–	–	–	–	–	0	–	–
1	1	0	1	–	–	0	1	0	1	0	0	1	1
1	1	1	0	1	0	1	0	–	0	1	–	–	–
1	1	1	1	–	0	1	1	1	1	–	1	1	0

Лабораторная работа 2
РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ НА БАЗЕ ЛОГИЧЕСКОГО АВТОМАТА В БАЗИСЕ РЕЛЕЙНО-КОНТАКТНЫХ ЭЛЕМЕНТОВ

Цель работы: получение навыков структурного синтеза логических автоматов без памяти и реализации системы управления на базе релейно-контактных элементов.

2.1 Элементы теории

Для разработки и моделирования работы схемы управления можно использовать программного обеспечения для виртуального моделирования и симуляции FluidSIM (FESTO). FluidSIM представляет собой комплексное программное обеспечение для создания, моделирования, обучения и изучения электропневматических, электрогидравлических, цифровых и электронных схем. FluidSIM сочетает в себе интуитивно понятный редактор принципиальных схем с подробным описанием компонентов, их фотографиями и соответствующей анимацией.

Для запуска FluidSIM можно воспользоваться ярлыком — Все программы / Festo Didactic / FluidSIM Pneumatics V 4.2 Student Version.

В результате откроется основное окно редактора FluidSIM (рис. 2.1).

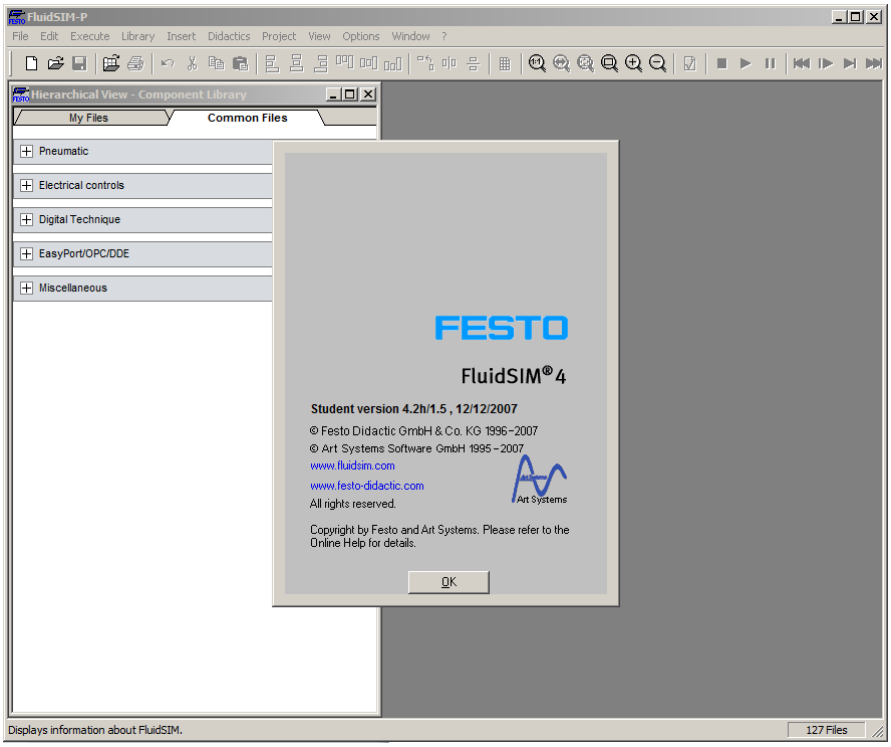


Рисунок 2.1 — Окно редактора FluidSIM (с окном «О программе»)

Основное окно редактора FluidSIM имеет интерфейс, который в том числе, содержит строку меню и панель инструментов (рис. 2.2).

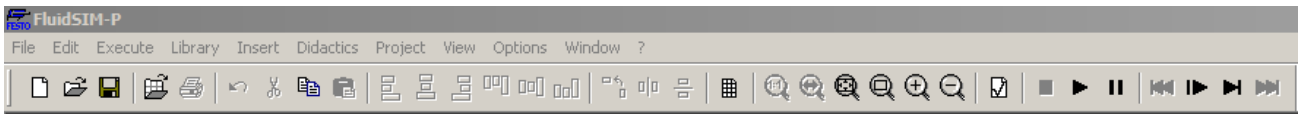


Рисунок 2.2 — Строка меню и панель инструментов редактора FluidSIM

Для создания схемы необходимо создать новый файл (рабочее пространство). При этом для удобства можно воспользоваться привязкой элементов по сетке (рис. 2.3), отображение которой можно включить в ленте инструментов.

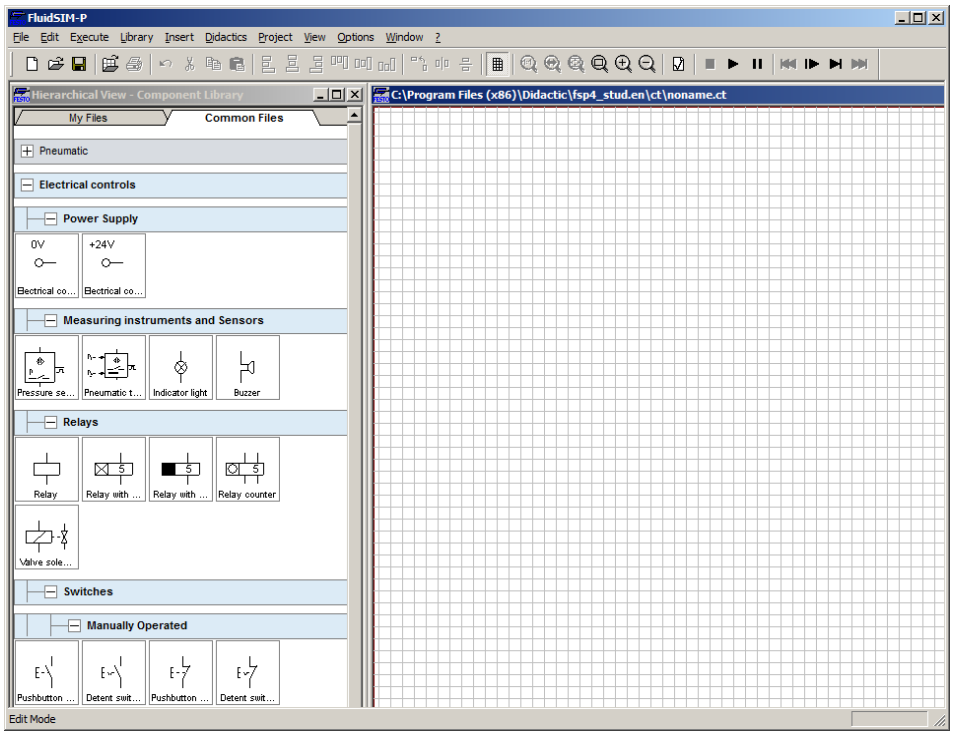


Рисунок 2.3 — Окно редактора FluidSIM (с отображением сетки для привязки элементов в рабочем пространстве)

Для разработки схемы управления необходимо использовать элементы из библиотеки компонентов (рис. 2.4).

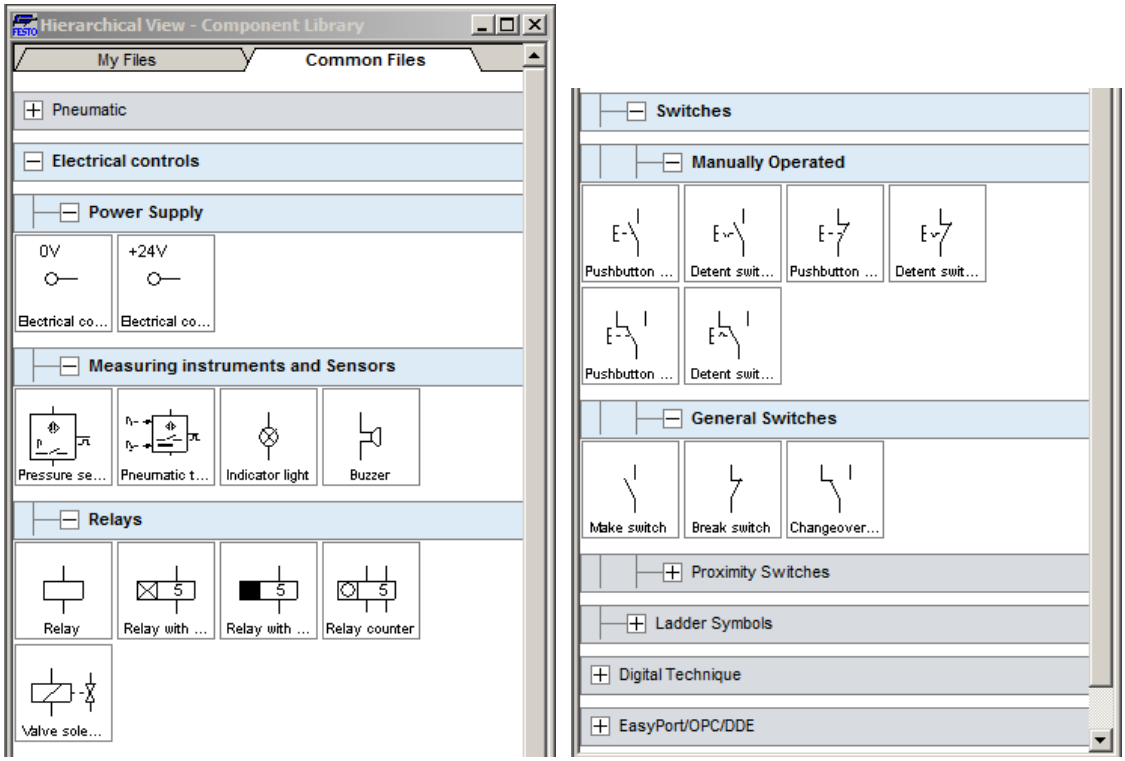
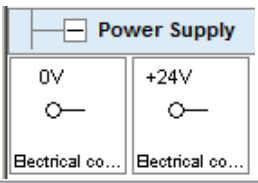


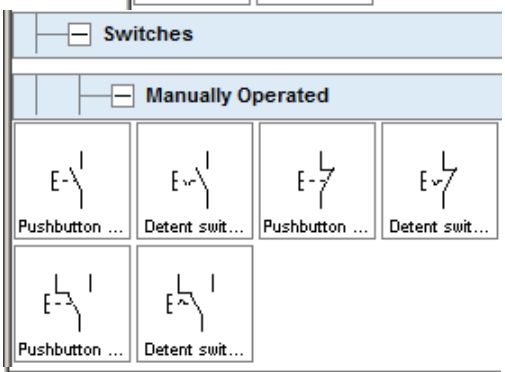
Рисунок 2.4 — Фрагменты окна библиотеки компонентов редактора FluidSIM (иерархическое представление компонентов)

Для разрабатываемой схемы управления необходимо использовать соответствующие компоненты.

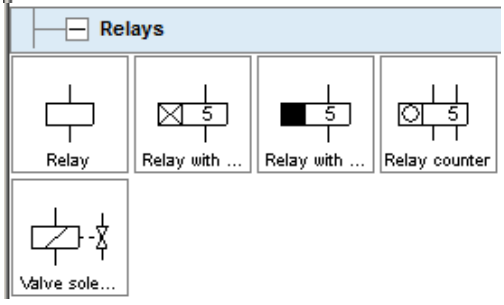
Источник питания /
Power Supply



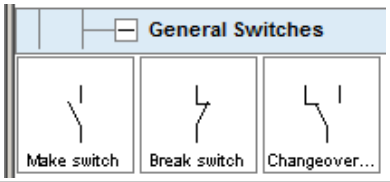
Переключатели с ручным управлением /
Switches Manually Operated



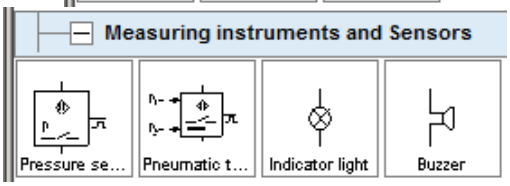
Реле /
Relay



Общие переключатели /
General Switches



Измерительные инструменты и Сенсоры /
Measuring instruments and Sensors



Для рассматриваемого примера по логической функции $f = x_1 + x_2\overline{x_3} + \overline{x_2}x_3$ схема управления на базе релейно-контактных элементов должна содержать (рис. 2.5):

– 1 нормально-замкнутую кнопку с фиксацией положения (Вкл.);
– 1 нормально-разомкнутую кнопку с фиксацией положения (СТОП);
– 3 кнопки без фиксации положения с соответствующими 3 реле (X1–X3);
– 1 реле для исполнительной цепи (Y);
– компоненты исполнительной цепи в составе 1 общего переключателя (Y), 1 светового индикатора (Indicator light) и зуммера (Buzzer);
– некоторое количество общих переключателей для реализации собственно цепи управления в виде соответствующего количества нормально-замкнутых и нормально-разомкнутых общих переключателей.

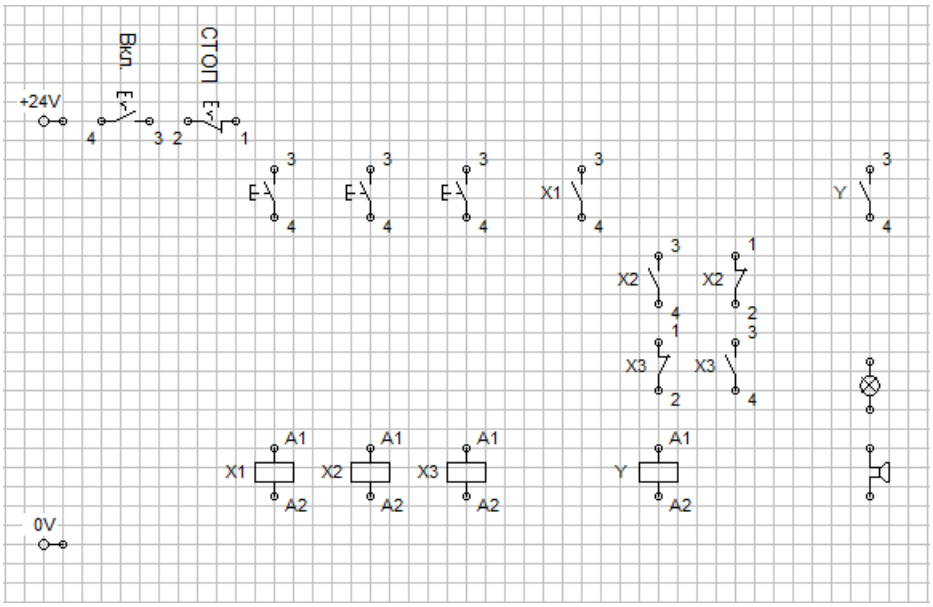


Рисунок 2.5 — Компоненты схемы управления в окне редактора FluidSIM (с выравниванием компонентов по сетке)

Необходимо отметить, что для привязки реле и основных переключателей необходимо для реле и соответствующих переключателей добавить соответствующую метку (Label), для чего, в свою очередь, необходимо

двойным кликом ЛКМ на компоненте вызвать соответствующее окно (рис. 2.6).

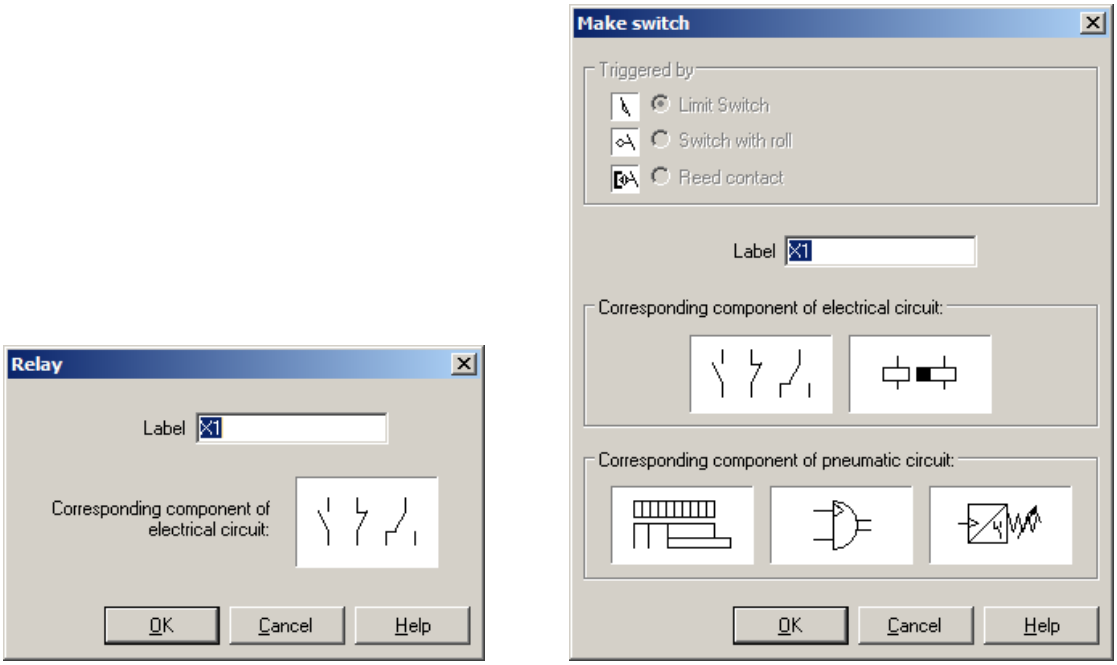


Рисунок 2.5 — Окна для ввода метки (Label) для реле X1 (Relay) и одного из основных переключателей (Switch)

Для соединения размещенных в рабочем пространстве компонентов необходимо нажать и, удерживая ЛКП курсором, соединить соответствующий контакт (на схеме имеют нумерацию 1–4 или обозначение A1, A2) выбранного компонента вначале с источником питания, а затем с полученными линиями соединений (рис. 2.6).

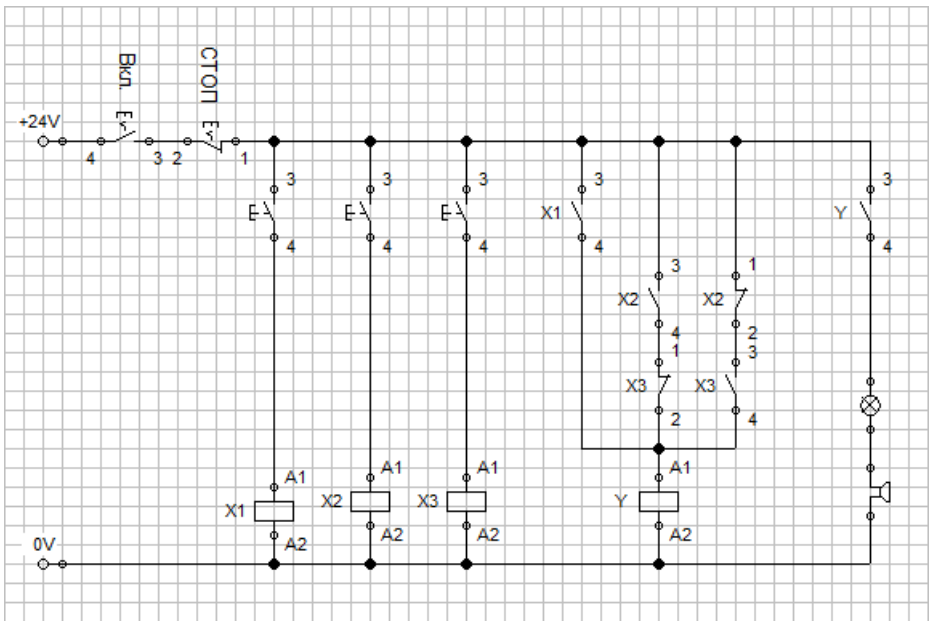


Рисунок 2.6 — Схема управления, полученная в результете соединения компонентов в рабочем пространстве

Окончательно схема управления состоит из источника питания, имитатора вводного выключателя «Вкл.», имитатора кнопки аварийного выключения «СТОП», цепей имитации входов (*а*), цепи управления (*б*) и исполнительной цепи (*в*), представленных на рисунке 2.7.

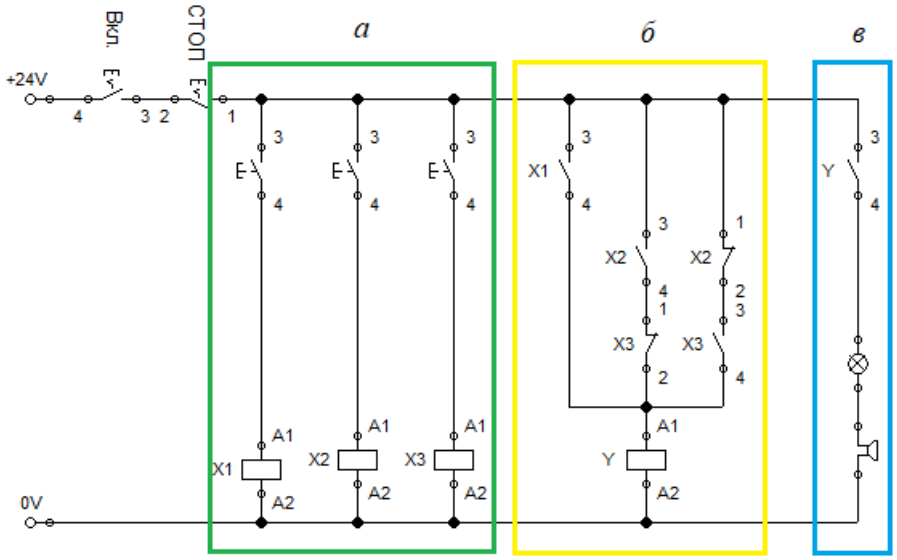
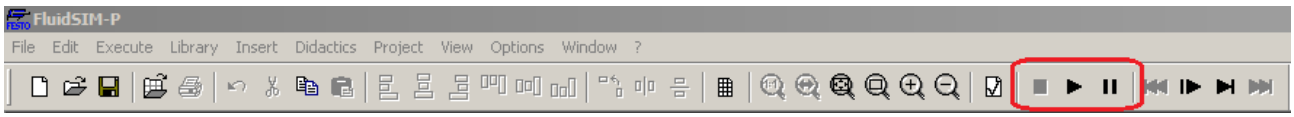


Рисунок 2.7 — Схема управления для логической функции $f = x_1 + x_2\overline{x_3} + \overline{x_2}x_3$

Программное обеспечение FluidSIM позволяет выполнить симуляцию работы схемы для чего необходимо воспользоваться соответствующими инструментами/командами в меню или панели инструментов.



Обязательным условием выполнения работы является выполнение валидации разработанной схемы управления. Для этого необходимо в режиме симуляции выполнить перебор всех условий по таблице состояний в соответствии с вариантом заданий и проверить соответствие состояния исполнительной цепи (включено/выключено). Для имитации работы входов использовать соответствующие кнопки в цепях реле X1, X2, X3. Состояние исполнительной цепи контролируется включением светового индикатора и зуммера, рисунок 2.7.

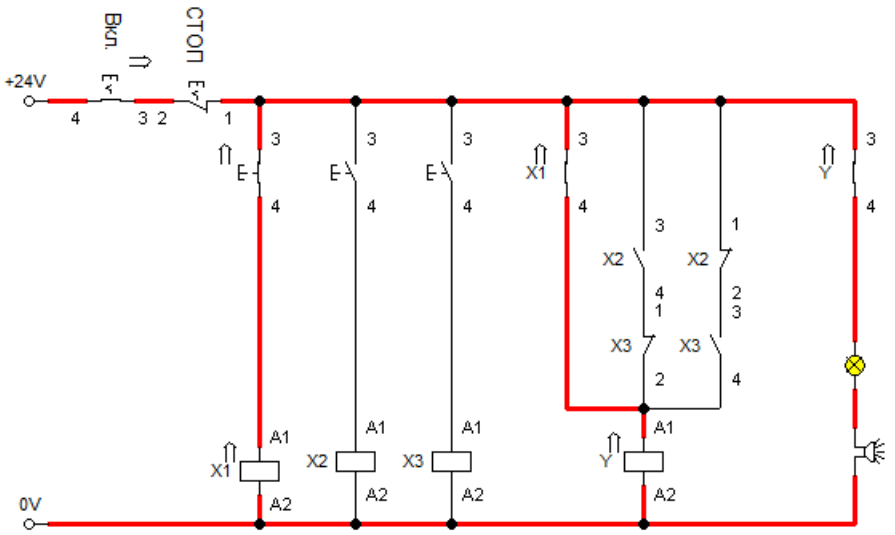


Рисунок 2.7 — Состояние схемы управления для логической функции $f = x_1 + x_2\overline{x_3} + \overline{x_2}x_3$ при проверке условия 1–0–0

2.2 Содержание работы

В процессе работы необходимо решить поставленную задачу в соответствии с индивидуальным заданием и оформить отчет с выводами.

Порядок выполнения работы

1. Изучить теоретическую часть.
2. Для МДНФ с использованием среды FluidSIM построить релейно-контактную схему, реализующую ПФ в СДНФ и СКНФ.
3. Выполнить валидацию полученных схем управления в базисе релейно-контактных элементов на соответствие заданной таблице истинности.
4. Выбрать одну из полученных схем и реализовать её (путем сборки) на стенде FESTO с использованием релейно-контактных элементов, входящих в комплектацию стенда.

2.3 Варианты заданий

Вариантами заданий являются минимизированные переключательные функции, полученные согласно индивидуальным вариантам заданий работы 1.

Лабораторная работа 3
ПРОГРАММИРОВАНИЕ МИКРОПРОЦЕССОРНОГО КОМПЛЕКТА

Цель работы: освоение процесса создания обеспечения микропроцессорных систем на примере программной реализации конечного логического автомата, заданного в виде переключательной функции.

3.1 Элементы теории

В состав рассматриваемой микропроцессорной системы входят: центральный процессор КР580ИК80; оперативная память на БИС ЗУ К565РУ2 объемом 1 Кбайт; занимающая адресное пространство 8000H÷83FFH; порты ввода/вывода, организованные на базе БИС программируемого параллельного интерфейса КР580ИК55, содержащей три порта ввода/вывода с адресами F8 — порт *A*, P9 — порт *B*, FA — порт *C* и порт приема программирующего слова (порт *D*) с адресом FB. Адреса даны в шестнадцатеричной системе счисления. Способ организации микропроцессорной системы (МПС) — «раздельная шина».

К трем разделам (1÷3) порта *C* интерфейса подключены тумблеры, позволяющие вводить с него наборы трех логических переменных x_1, x_2, x_3, соответственно, порт *C* должен программироваться на ввод. К восьми разрядам порта *B* подключены светодиодные индикаторы, позволяющие выводить информацию в двоичной форме. Этот порт должен программироваться как порт вывода. В соответствии с таким распределением функций портов программирующее слово имеет код 81H.

Программная реализация процессов программирования интерфейса и ввода/вывода данных при организации с разделенной шиной производится с помощью команд IN (ввод) и OUT (вывод) и состоит из этапов: подготовки программирующего слова, программирования интерфейса, ввода/вывода данных.

Пример программы, содержащей эти этапы, приведен в таблице 3.1.

Для ускорения процесса ввода данные о состоянии тумблеров набора логических переменных вводятся через порт *C* параллельно путем подачи сигналов на разные разряды порта. Для выполнения логических операций с этими данными их необходимо выделить и записать в отдельные ячейки памяти в разряды с одинаковыми номерами. Эта цель достигается с помощью сдвигов (команда RRC) и маскирования (выделения битов), а также команд пересылки. Маскирование осуществляется выполнением логической операции «И» над данными и маскирующим словом.

Фрагмент программы, выполняющей выделение и размещение данных в памяти, приведен в таблице 3.2. В результате работы программы данные

должны быть помещены в младшие разряды регистров, соответственно, $x_1 \to E$, $x_2 \to D$, $x_3 \to C$.

Таблица 3.1 — Пример программы, реализующей процессы программирования интрефейса и ввода/вывода данных

Адрес	Содержание	Метка	Команда	Комментарий
8000 8001	3E 81		MVI A,81	Запись в аккумулятор программирующего слова
8002 8003	D3 FB		OUT FB	Программирующее слово передается на порт *D* (адрес FB)
8004 8005	DB FA	MET:	IN FA	Чтение данных из порта *C* (адрес FA)
8006	00		NOP	—
8007 8008	D3 F9		OUT F9	Запись данных в порт *B* (адрес F9)
8009 800A 800B	C3 04 80		JMP MET	Переход к метке MET для повторения

Таблица 3.2 — Пример программы, выполняющей выделение и размещение данных в памяти

Адрес	Содержание	Метка	Команда	Комментарий
				После ввода данные помещены в разряды [1–3] аккумулятора

			MVI C,01
			RRC
			MOV B,A
			ANA C
			MOV E,A
				Маскирующее слово $\Rightarrow C$
				Сдвиг вправо. Данные в разрядах [0–2]
				Данные $\Rightarrow B$ на хранение
				Выделение младшего бита $x_1 \Rightarrow A$
				$x_1 \Rightarrow E$

			MOV A,B
			RRC
			MOV B,A
			ANA C
			MOV D,A
				Вызов данных на аккумулятор
				Сдвиг x_2 в разряде 0
				На хранение в регистр *B*
				Выделение $x_2 \Rightarrow A$
				$x_2 \Rightarrow D$

			MOV A,B
			RRC
			ANA C
			MOV C,A
				Вызов данных на аккумулятор
				Сдвиг x_3 в разряде 0
				Выделение $x_3 \Rightarrow A$
				$x_3 \Rightarrow C$

Для выполнения логических операций над выделенными переменными используются операции функционального полного набора: CMA — инверсия, ANA — логическое умножение (конъюнкция), ORA — логическое сложение (дизъюнкция). Первый операнд находится на аккумуляторе, результат помещается туда же.

3.2 Содержание работы

В процессе работы необходимо решить поставленную задачу, в соответствии с индивидуальным заданием, и оформить отчет с выводами.

Задача, решаемая в данной работе, состоит в программной реализации переключательной функции.

Рассмотрим функциональную схему, составленную для переключательной функции

$$f = x_1 \cdot \overline{x_2} + x_1 \cdot x_2 \cdot x_3 + \overline{x_2} \cdot x_3 .$$

Схема представлена на рисунке 3.1. После ввода с тумблеров через порт *C* на аккумулятор байт информации подвергается сдвигу и маскированию. Маскируемый бит (младший бит байта) выделяется и передается на соответствующий регистр. Информация в регистрах *E*, *D* и *C* подвергается обработке через логическую схему, содержащую восемь двухвходовых элементов, расположенных в восьми каскадах, имитирующих последовательность выполнения логических операций. Результат — переключательная функция *f* передается на аккумулятор и выводится через порт *B* на индикацию.

Программа, реализующая эту функцию, представлена в таблице 3.3. Последовательность обработки данных — на рисунке 3.1.

Для сокращения длины программы последовательность обработки должна выбираться по возможности так, чтобы в следующей операции использовать результат предыдущий, хранимый на аккумуляторе.

Порядок выполнения работы

1. Изучить элементы теории по выполняемой работе.
2. Ввести в память МПС программу (таблица 3.1). Запустить программу в работу. Проследить изменение состояния индикаторов при переключении тумблеров.
3. Ввести по адресу 8006 команду RRC кодом 0F. Запустить программу. Сравнить положение выводимых бит с предыдущим опытом (п. 2).
4. Ввести по адресу 8006 команду CMA кодом 2F. Запустить программу. Проследить изменение состояния индикаторов. Сравнить с исходным опытом (п. 2).
5. Разработать программу, реализующую заданную переключательную функцию по примеру в таблице 3.3. Ввести программу в память МПС. Запустить программу. Изменением положения тумблеров ввести последовательно 8 наборов переключательной функции, считывая се значения с индикатора. Результаты свести в таблицу. Сравнить с исходной.

3.3 Варианты заданий

Вариантами заданий являются минимизированные переключательные функции, полученные согласно индивидуальным вариантам заданий работы 1.

Таблица 3.3 — Пример программы, реализующей переключательную функцию $\left(x_1 \wedge \overline{x_2}\right) \vee \left(x_1 \wedge x_2 \wedge x_3\right) \vee \left(\overline{x_2} \wedge x_3\right)$

Адрес	Содержание	Метка	Команда	Комментарий
8000	3E		MVI A,81	Запись в аккумулятор (A) программирующего слова
8001	81			
8002	D3		OUT FB	Программирующее слово передается на порт D
8003	FB			(адрес FB)
8004	DB	ST:	IN FA	Чтение данных из порта C (адрес FA)
8005	FA			
8006	0F		RRC	Сдвиг вправо данных
8007	0E		MVI C,01	Маска 01 $\Rightarrow C$
8008	01			
8009	47		MOV B,A	Данные $\Rightarrow B$ на хранение
800A	A1		ANA C	Маскирование x_1
800B	5F		MOV E,A	$x_1 \Rightarrow E$
800C	78		MOV A,B	Вызов данных на A
800D	0F		RRC	Сдвиг данных вправо
800E	47		MOV B,A	Данные $\Rightarrow B$ на хранение
800F	A1		ANA C	Маскирование x_2
8010	57		MOV D,A	$x_2 \Rightarrow D$
8011	78		MOV A,B	Вызов данных на A
8012	0F		RRC	Сдвиг данных вправо
8013	A1		ANA C	Маскирование x_3
8014	4F		MOV C,A	$x_3 \Rightarrow C$
8015	7A		MOV A,D	$x_2 \Rightarrow A$
8016	2F		CMA	Инверсия x_2 ($\overline{x_2}$) $\Rightarrow A$
8017	A3		ANA E	$x_1 \wedge \overline{x_2} \Rightarrow A$
8018	47		MOV B,A	Результат (из A) $\Rightarrow B$
8019	7B		MOV A,E	$x_1 \Rightarrow A$
801A	A2		ANA D	$x_1 \wedge x_2 \Rightarrow A$
801B	A1		ANA C	$\left(x_1 \wedge x_2\right) \wedge x_3 \Rightarrow A$
801C	B0		ORA B	$\left(x_1 \wedge \overline{x_2}\right) \vee \left(x_1 \wedge x_2 \wedge x_3\right) \Rightarrow A$
801D	47		MOV B,A	Результат (из A) $\Rightarrow B$
801E	7A		MOV A,D	$x_2 \Rightarrow A$
801F	2F		CMA	$\overline{x_2} \Rightarrow A$
8020	A1		ANA C	$\overline{x_2} \wedge x_3 \Rightarrow A$
8021	B0		ORA B	$\left(x_1 \wedge \overline{x_2}\right) \vee \left(x_1 \wedge x_2 \wedge x_3\right) \vee$ $\vee \left(\overline{x_2} \wedge x_3\right) \Rightarrow A$
8022	D3		OUT F9	Вывод результата на порт B (адрес F9)
8023	F9			
8024	C3		JMP ST	Переход к метке ST (адрес 8004) для повторения
8025	04			
8026	80			

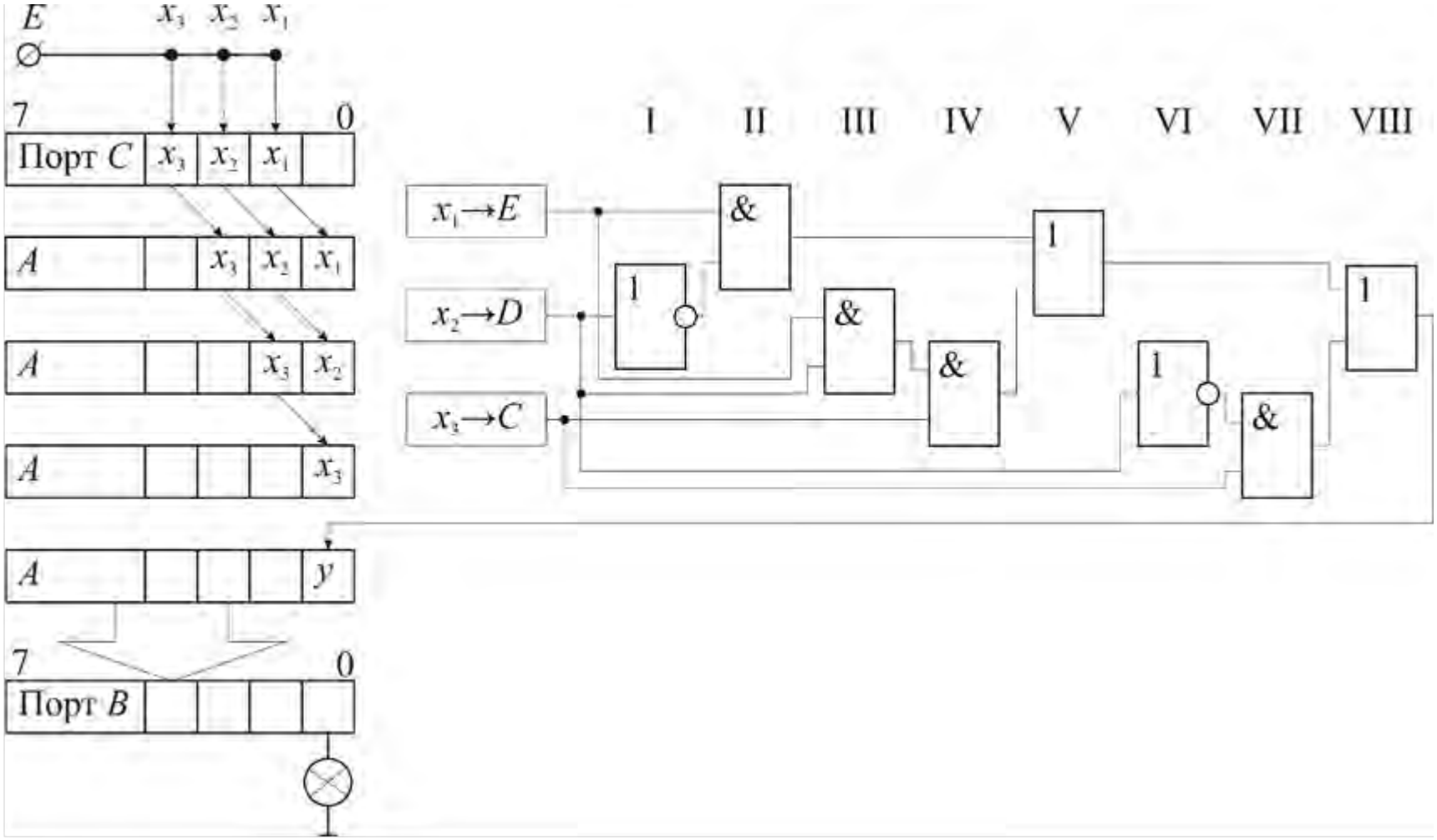


Рисунок 3.1 — Последовательность обработки данных для рассматриваемого примера переключательной функции

3.4 Микропроцессорная лаборатория «Микролаб КР580ИК80»

Микролаб МК580ИК80 — это портативная микроЭВМ, предназначенная для изучения аппаратно-программных средств микропроцессорных устройств, созданных на базе микропроцессорного набора серии 580.

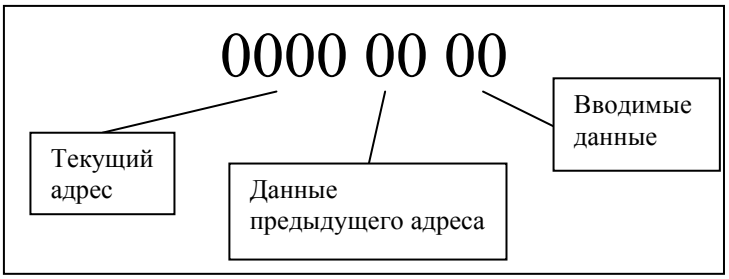


Рисунок 2.2 — Индикатор микроЭВМ

Клавиатура микроЭВМ «Микролаб КР580ИК80»

Группа клавиш управления исполнением программ:

ПУСК — выполнение программы;

ВОЗВР — возвращение к исполняемой программе;

ВВОД — ввод информации с магнитофона;

ВЫВОД — вывод информации на магнитофон;

СБРОС — остановка выполнения заданной программы.

Группа клавиш управления памятью:

ЗП — запись данных в ячейку ЗУ и увеличение номера адреса на единицу;

УСТ.АД — установка адреса;

АД+ — увеличение номера адреса на единицу;

АД- — уменьшение номера адреса на единицу.

Группа клавиш ввода программы:

«C» «D» «E» «F»

«8» «9» «A» «B»

«4» «5» «6» «7»

«0» «1» «2» «3»

Пример исполнения программы:
1. Нажмите клавишу «СБРОС»
2. Нажмите клавишу «3»
3. Нажмите клавишу «0»
4. Нажмите клавишу «0»
5. Нажмите клавишу «УСТ.АД»
6. Нажмите клавишу «ПУСК»
Индикаторы гаснут. Выполнение программы — музыкальный фрагмент.

Команды микропроцессора КР580ИК80

Таблица 3.4 — Команды пересылки данных

Команды пересылки данных								
Пересылка						Непосредственная пересылка		
MOV	A,A	7F	MOV	E,A	5F	MVI	A, байт	3E
	A,B	78		E,B	58		B, байт	06
	A,C	79		E,C	59		C, байт	0E
	A,D	7A		E,D	5A		D, байт	16
	A,E	7B		E,E	5B		E, байт	1E
	A,H	7C		E,H	5C		H, байт	26
	A,L	7D		E,L	5D		L, байт	2E
	A,M	7E		E,M	5E		M, байт	36
MOV	B,A	47	MOV	H,A	67	Непосредственная загрузка		
	B,B	40		H,B	60	LXI	B, 2 байта	01
	B,C	41		H,C	61		D, 2 байта	11
	B,D	42		H,D	62		H, 2 байта	21
	B,E	43		H,E	63		SP, 2 байта	31
	B,H	44		H,H	64	Загрузка/хранение		
	B,L	45		H,L	65	LDAX B		0A
	B,M	46		H,M	66	LDAX D		1A
MOV	C,A	4F	MOV	L,A	6F	LHLD адр		2A
	C,B	48		L,B	68	LDA адр		3A
	C,C	49		L,C	69			
	C,D	4A		L,D	6A	STAX B		02
	C,E	4B		L,E	6B	STAX D		12
	C,H	4C		L,H	6C	SHLD адр		22
	C,L	4D		L,L	6D	STA адр		32
	C,M	4E		L,M	6E			
MOV	D,A	57	MOV	M,A	77			
	D,B	50		M,B	70			
	D,C	51		M,C	71			
	D,D	52		M,D	72			
	D,E	53		M,E	73			
	D,H	54		M,H	74			
	D,L	55		M,L	75			
	D,M	56	XCHG		EB			

Таблица 3.5 — Арифметические и логические команды

Арифметические и логические команды								
Сложение			Увеличение			Логические		
ADD	A	87	INR	A	3C	ANA	A	A7
	B	80		B	04		B	A0
	C	81		C	0C		C	A1
	D	82		D	14		D	A2
	E	83		E	1C		E	A3
	H	84		H	24		H	A4
	L	85		L	2C		L	A5
	M	86		M	34		M	A6
ADC	A	8F	INX	B	03	XRA	A	AF
	B	88		D	13		B	A8
	C	89		H	23		C	A9
	D	8A		SP	33		D	AA
	E	8B	Уменьшение				E	AB
	H	8C	DCR	A	3D		H	AC
	L	8D		B	05		L	AD
	M	8E		C	0D		M	AE
Вычитание				D	15	ORA	A	B7
SUB	A	97		E	1D		B	B0
	B	90		H	25		C	B1
	C	91		L	2D		D	B2
	D	92		M	35		E	B3
	E	93	DCX	B	0B		H	B4
	H	94		D	1B		L	B5
	L	95		H	2B		M	B6
	M	96		SP	3B	CMP	A	BF
SBB	A	9F	Специальные				B	B8
	B	98	DAA*		27		C	B9
	C	99	CMA		2F		D	BA
	D	9A	STC+		37		E	BB
	E	9B	CMC+		3F		H	BC
	H	9C	Сдвиг				L	BD
	L	9D	RLC		07		M	BE
	M	9E	RRC		0F	Непосредственные логические и арифметические		
Двойное сложение			RAL		17			
DAD	B	09	RAR		1F	ADI байт		C6
	D	19				ACI байт		CE
	H	29				SUI байт		D6
	SP	39				SBI байт		DE
						ANI байт		E6
						XRI байт		EE
						ORI байт		F6
						CPI байт		FE

Таблица 3.6 — Команды перехода, прерывания, ввода/вывода и управления

Команды перехода			**Команды ввода/вывода и управления**		
Переходы			Операции со стеком		
JMP адр		C3	PUSH	B	C5
JNZ адр		C2		D	D5
JZ адр		CA		H	E5
JNC адр		D2		PSW	F5
JC адр		DA	POP	B	C1
JPO адр		E2		D	D1
JPE адр		EA		H	E1
JP адр		F2		PSW	F1
JM адр		FA	XTHL		E3
PCHL адр		E9	SPHL		F9
Вызовы			Ввод/вывод		
CALL адр		CD	OUT байт		D3
CNZ адр		C4	IN байт		DB
CZ адр		CC	Управление		
CNC адр		D4	DI		F3
CC адр		DC	EI		FB
CPO адр		E4	NOP		00
CPE адр		EC	HLT		76
CP адр		F4			
CM адр		FC			
Возврат					
RET		C9			
RNZ		C0			
RZ		C8			
RNC		D0			
RC		D8			
RPO		E0			
RPE		E8			
RP		F0			
RM		F8			
Прерывания					
RST	0	C7			
	1	CF			
	2	D7			
	3	DF			
	4	E7			
	5	EF			
	6	F7			
	7	FF			

Лабораторная работа 4
ПРОГРАММИРОВАНИЕ ТЕХНОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ С ПРОГРАМНЫМ УПРАВЛЕНИЕМ

Цель работы: освоение процесса разработки управляющих программ для технологического оборудования с программным управлением.

4.1 Элементы теории

G-код — условное именование языка программирования устройств с числовым программным управлением (ЧПУ). Был создан компанией Electronic Industries Alliance в начале 1960-х. Окончательная доработка была одобрена в феврале 1980 года как стандарт RS274D. Комитет ISO утвердил G-код как стандарт ISO 6983–1:2009, Госкомитет по стандартам СССР — как ГОСТ 20999–83 [1]. В советской технической литературе G-код обозначается как код ИСО 7-бит (ISO 7-bit), это вызвано тем, что G-код кодировали на 8-дорожечную перфоленту в коде ISO 7-bit (разработан для представления информации УЧПУ в виде машинного кода так же, как и коды AEG и PC8C), восьмая дорожка использовалась для контроля чётности. [2]

Производители систем УЧПУ (CNC), как правило, используют ПО управления станком, для которого написана (оператором) программа обработки, в качестве осмысленных команд управления, используется G-код в качестве базового подмножества языка программирования, расширяя его по своему усмотрению [2].

G-Code — это также стандартный язык, используемый многими моделями 3D-принтеров для управления процессом печати. Файлы GCODE могут быть открыты с помощью различных программ 3D-печати, например, Simplify3D, GCode Viewer, а также с помощью текстового редактора, поскольку их содержимое представляет собой обычный текст. [2]

Основные требования к структуре. [2]

Программа, написанная с использованием G-кода, имеет жёсткую структуру. Все команды управления объединяются в кадры — группы, состоящие из одной или более команд. Кадр завершается символом перевода строки (CR/LF) и может необязательно иметь явно указанный номер, начинающийся с буквы N, за исключением первого кадра программы и комментариев. Этот номер является, по сути, меткой кадра и необязательно должен нарастать в программе или представлять собой последовательные целые числа, важно, чтобы номер не повторялся в пределах программы, например, допустимо:

```
...
N200 G0
n100 x0
x5y4
...
```

В большинстве современных интерпретаторов кода допустимо использовать в коде программы строчные и прописные буквы, как в примере.

Пробелы в строке кадра игнорируются, поэтому допустимо слитное написание команд кадра.

Первый (а в некоторых случаях ещё и последний) кадр содержит только один необязательный знак <%>. Завершается программа командами M02 или M30.

Комментарии к программе размещаются в круглых скобках. Комментарий может располагаться как в отдельной строке, так и в любом месте кадра среди команд. Недопустимо оформлять в качестве комментария несколько строк, охваченных парой круглых скобок.

Элементарные команды в каждом кадре выполняются одновременно, поэтому порядок команд в кадре строго не оговаривается, но традиционно предполагается, что первыми указываются подготовительные команды (например, выбор плоскости круговой интерполяции, скоростей перемещений по осям и др.), затем задание координат перемещения, затем выбора режимов обработки и технологические команды.

Максимальное число элементарных команд и заданий координат в одном кадре зависит от конкретного интерпретатора языка управления станками, но для большинства популярных интерпретаторов (стоек управления) не превышает 6.

Координаты задаются указанием оси с последующим числовым значением координаты. Целая и дробная части числа координаты разделяются десятичной точкой. Допустимо опускание незначащих нулей, либо их добавление. Также в подавляющем количестве интерпретаторов допустимо не добавлять десятичную точку к целым числам. Например, Y0.5 и Y.5, Y77, Y77. и Y077.0.

Существуют так называемые модальные и немодальные команды. Модальные команды изменяют некоторый параметр/настройку, и эта настройка действует на все далее исполняемые кадры программы до их смены очередной модальной командой либо её отмены. К модальным командам, например, относятся скорости перемещения инструмента, управления скоростью шпинделя, подачи смазочно-охлаждающей жидкости (СОЖ) и др. Немодальные команды действуют только внутри их содержащего кадра. К немодальным командам относятся, например, команды разгона и торможения шпинделя.

Интерпретатор кода (стойка управления) станком запоминает значение введённых параметров и настроек до их смены очередной модальной командой или отмены ранее введённой модальной команды, поэтому необязательно указание в каждом кадре, например, скорости перемещения инструмента.

Описание и вызов подпрограмм [2]

Язык допускает многократное исполнение однократно записанной последовательности команд и перемещений инструмента, вызываемых из разных частей программы, например, вырезания в листовой заготовке многих отверстий с одинаковым сложным контуром, расположенных в разных местах будущей детали. При этом в теле подпрограммы описывается траектория движения инструмента для вырезания одного отверстия, а в программе производится многократный вызов подпрограммы из разных мест. В теле подпрограммы перемещения инструмента задаются в относительных координатах — координатах, описывающих траекторию инструмента при обработке отверстия, переход к относительной системе координат (иногда такую систему координат называют <инкрементной>) производится командой G91 в начале тела подпрограммы, а возврат к абсолютной системе координат командой G90 — в конце тела подпрограммы. В инкрементной системе команды перемещения инструмента интерпретируются как приращения, например,

```
g90 x5
(назначение абсолютной системы координат, после
исполнения этого кадра машинная координата по оси X
станет равной 5)
g91 x10
(назначение инкрементной системы координат, после
исполнения этого кадра машинная координата по оси X
станет равной 15)
x-15
(после исполнения этого кадра машинная координата по оси
X станет равной 0, так как продолжает действовать
инкрементная система координат, заданная модальной
командой g91).
```

Тело подпрограммы обязательно должно быть описано до команды конца программы — M30, но допустимо расположение подпрограммы после команды M02 — конца программы и иметь имя, начинающееся с буквы O с цифрами номера подпрограммы, например, O112. В конце тела подпрограммы помещается команда возврата в основную программу — M99.

В программе вызов подпрограммы производится командой M98 с указанием обязательного параметра имени подпрограммы P. Недопустимо совпадение имён подпрограмм в пределах одной программы. Пример вызова подпрограммы O112: M98 P112. Допустимо при вызове подпрограммы указание числа вызовов подпрограммы добавлением необязательного параметра L, например, двукратный вызов подпрограммы 112: M98 P112 L2, что, например, может быть полезно при описании выполнения второго прохода чистовой обработки после первого прохода черновой обработки. При опущенном параметре L подпрограмма вызывается однократно.

Управляющее математическое обеспечение некоторых станков или некоторые интерпретаторы G-кода допускают вызов подпрограмм по номеру строки (кадра) в программе, для этого используется команда M97 с параметром P, указывающем на номер (метку) строки, например, M97 P321 L4 — четырёхкратный вызов подпрограммы, начинающейся с меткой N321. Оформленная таким образом подпрограмма как обычно должна заканчиваться командой M99 — возвратом в вызвавшую программу.

Допустимо вложение подпрограмм, то есть из подпрограммы возможен вызов другой подпрограммы. Максимально допустимое число уровней вложения зависит от реализации конкретного интерпретатора G-кода.

Пример программы вырезания 2 прямоугольных отверстий 10 х 20 мм, увеличенных на диаметр торцевой фрезы, с координатами левых нижних углов отверстий x=57, y=62 и x=104, y=76 в листовой заготовке толщиной 5 мм с вызовом подпрограммы описывающей вырезание одного отверстия

```
...
(Фрагмент программы)
G00 X57 Y62 (позиционирование по X, Y на 1-е отверстие)
M98 P112 (вырезание 1-го отверстия)
G00 X104 Y76 (позиционирование по X, Y на 2-е отверстие)
M98 P112 (вырезание 2-го отверстия)
...
M02 (Конец программы)
...
(Тело подпрограммы)
O112 (Метка подпрограммы, номер 112)
G00 Z1 (Подвод инструмента на высоту 1 мм над поверхностью
заготовки со скоростью холостого перемещения)
G01 F40 Z-5.5 (Врезание инструмента на глубину -5,5 мм в заготовку
со скоростью 40 мм/мин)
G91 (Переход в относительную систему координат, в этой системе
вначале X=0, Y=0)
G01 F20 X10 (Вырезание 1-й стороны прямоугольника со скоростью 20
мм/мин)
Y20 (Вырезание 2-й стороны прямоугольника со скоростью 20 мм/мин)
X-10 (Вырезание 3-й стороны прямоугольника со скоростью 20 мм/мин.
Так как включена инкрементальная система координат, то возврат
инструмента в исходную точку до вызова подпрограммы указывается в
виде приращения координаты, здесь -10.)
Y-20 (Вырезание 4-й стороны прямоугольника со скоростью 20 мм/мин)
G90 (Переход в абсолютную систему координат, восстановление
текущих координат до перехода в относительную систему)
G00 Z5 (Подъём инструмента на высоту 5 мм над поверхностью
заготовки со скоростью холостого перемещения)
M99 (Возврат в вызывавшую программу или подпрограмму)
...
M30 (Конец интерпретируемого кода программы. После исполнения этой
команды указатель номера кадра устанавливается на 1-ю строку
программы и исполнение программы останавливается)
```

4.2 Порядок разработки управляющей программы

В качестве программно-управляемого технологического оборудования используем лабораторную установку на базе фрезерно-гравировального станка, в качестве объекта производства — логотип университета.

Для разработки управляющей программы можно использовать, например, свободно распространяемый генератор G-кода (рис. 4.1).

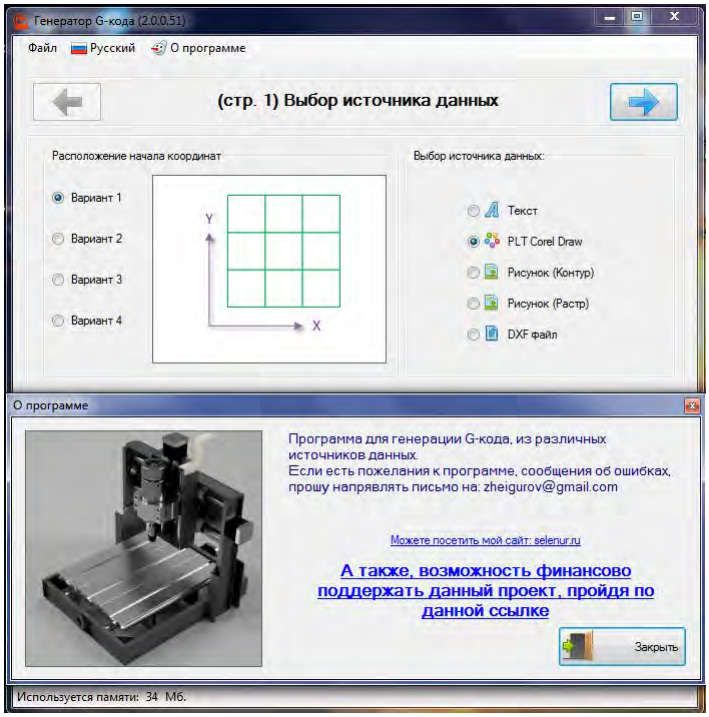


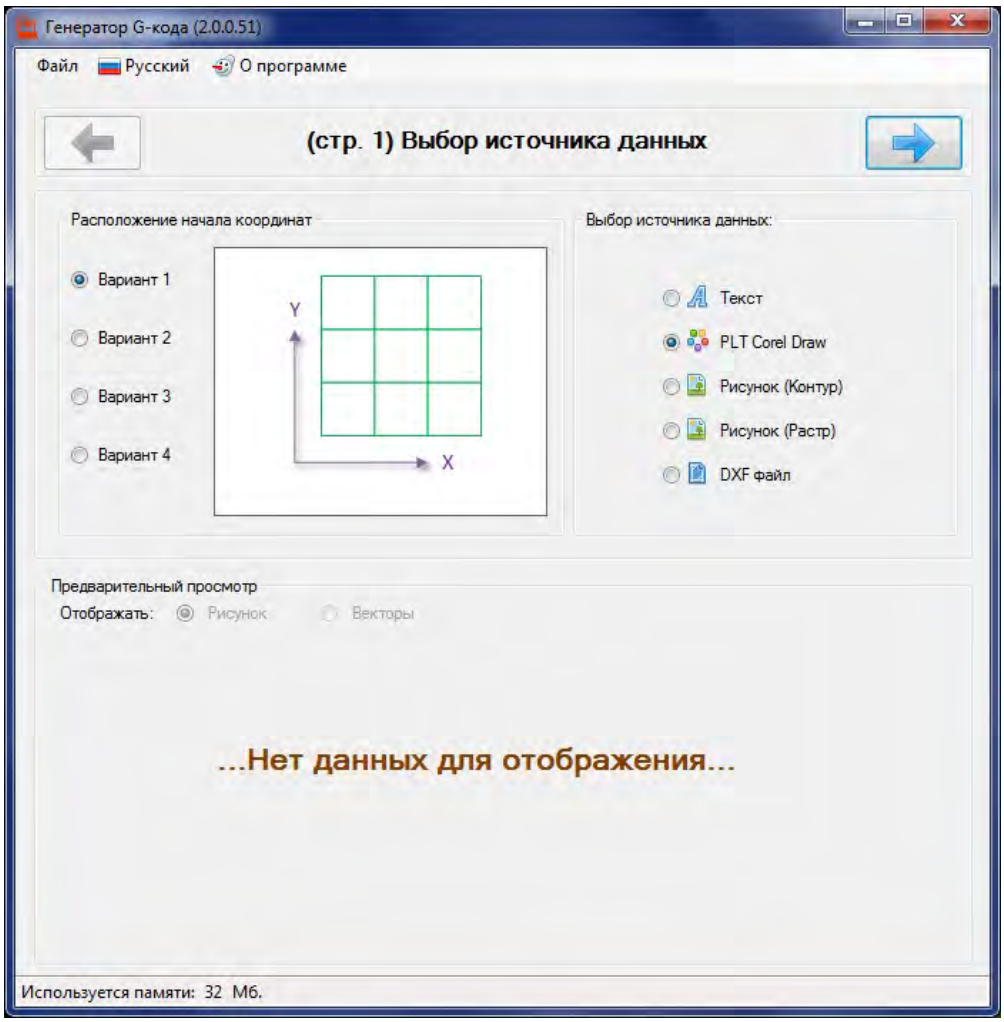
Рисунок 4.1 — Окно генератора G-кода (с окном «О программе»)

На первом этапе выбирается источник данных, в качестве которого можно использовать вводимый в окне программы текст или файлы в одном из поддерживаемых форматов.

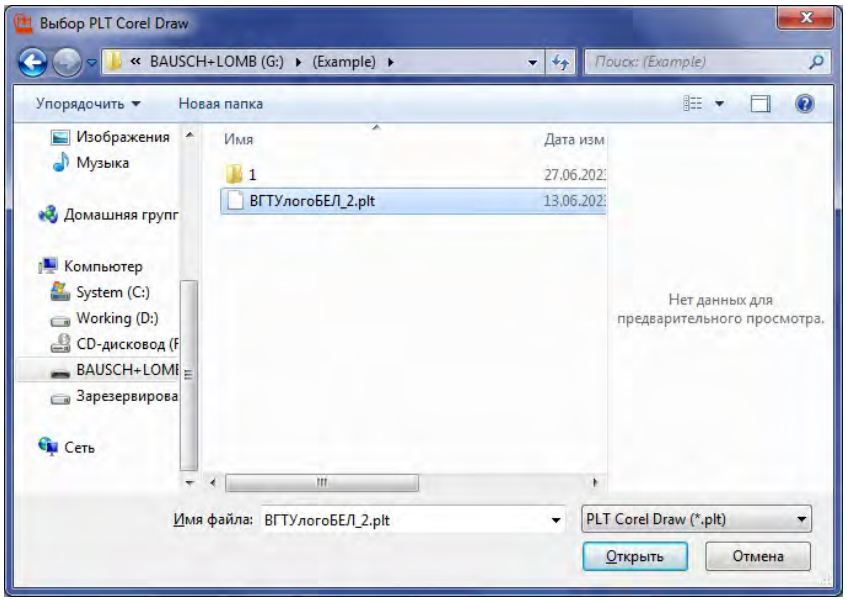
Программное обеспечение поддерживает следующие форматы файлов:

- векторный формат файла плоттера PLT CorelDraw;
- векторный формат рисунков (Рисунок (Контур));
- растровый формат рисунков (Рисунок (Растр));
- формат обмена графической информацией САПР *.dxf (DXF файл).

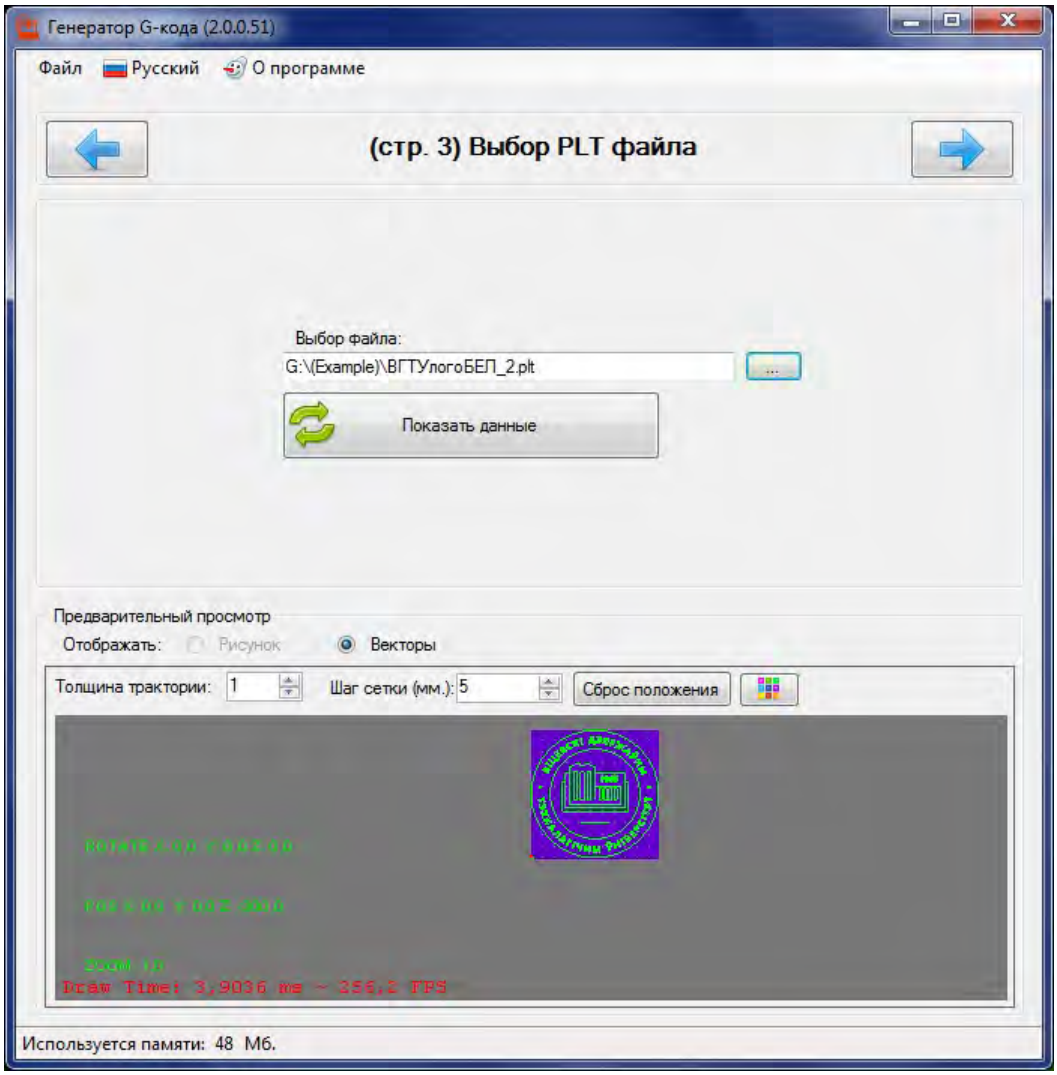
Кроме выбора источника данных можно выбрать расположения локальной системы координат. Доступны 4 варианта расположения с размещением центра системы координат в левом нижнем, левом верхнем, правом верхнем и правом нижнем углу относительно обрабатываемого контура.



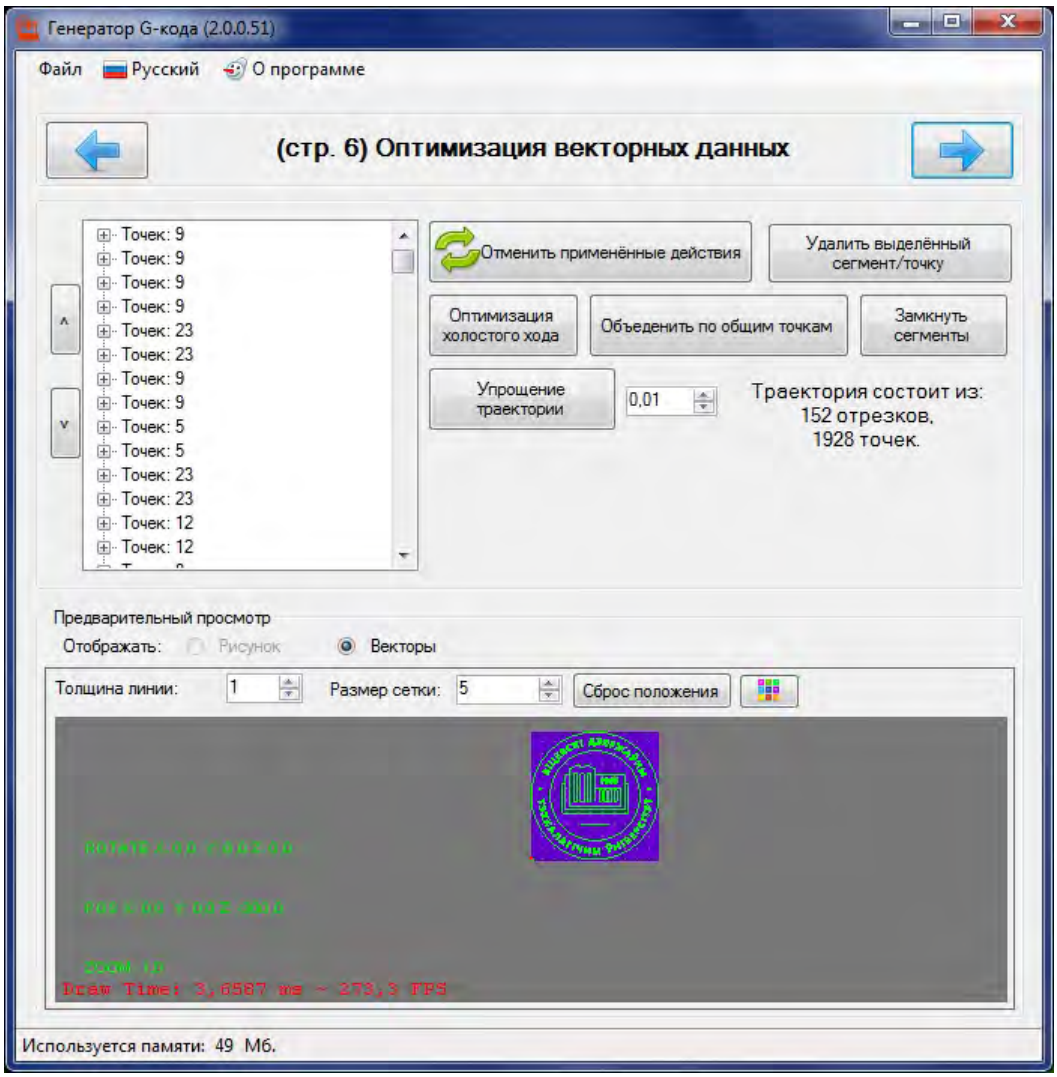
Для загрузки файла (например, в формате PLT Corel Draw) необходимо в окне «стр. 1» установить переключатель в соответствующую позицию и нажать кнопку «Далее» (→). В открывшемся окне выбрать соответствующий файл и нажать кнопку «Открыть».



После загрузки файла в окне предварительного просмотра окна «стр. 3» отобразится обрабатываемый контур, после чего необходимо нажать кнопку «Далее».



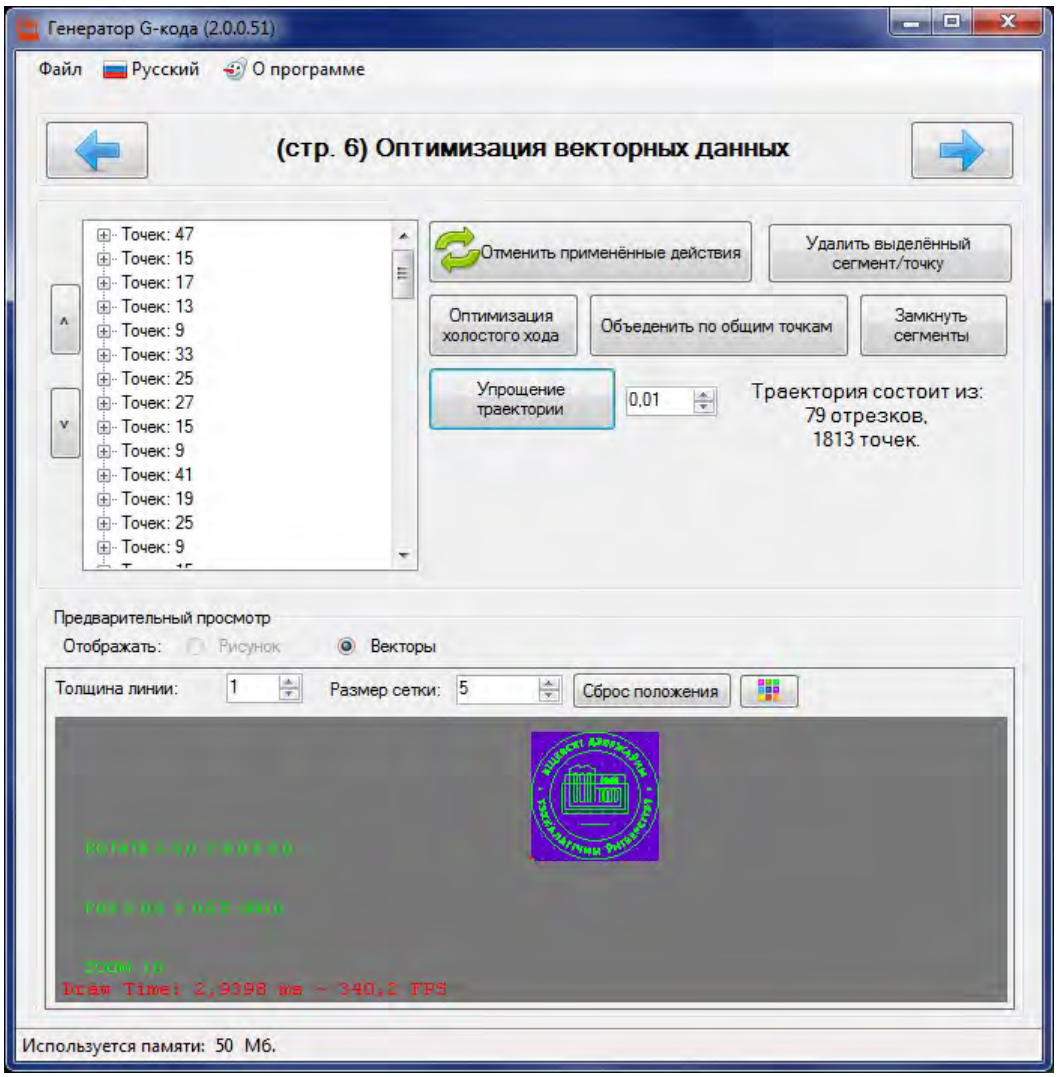
После завершения автоматической генерации траектории перемещения в окне «стр. 6» отобразится структура траектории, которую при необходимости можно редактировать вручную или автоматически.



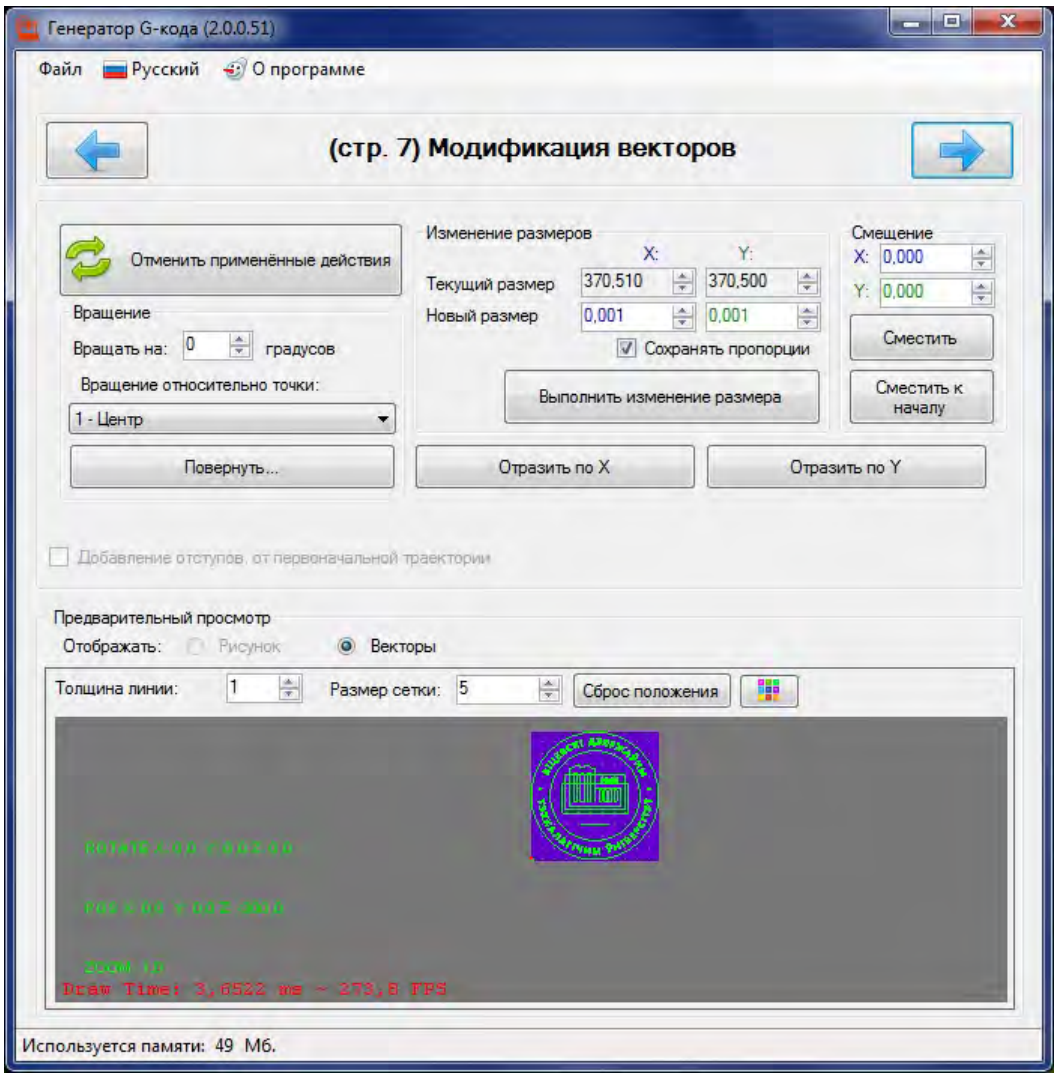
Доступны варианты автоматической оптимизации траектории перемещений: оптимизация холостого хода, объединение по общим точкам, замыкание сегментов и упрощение траектории.

Для рассматриваемого примера использование автоматической оптимизации траектории перемещений позволило сократить траекторию с первоначальных значений 152 отрезка, 1928 точек до значений 79 отрезков, 1813 точек.

После завершения оптимизации траектории перемещений необходимо нажать кнопку «Далее».

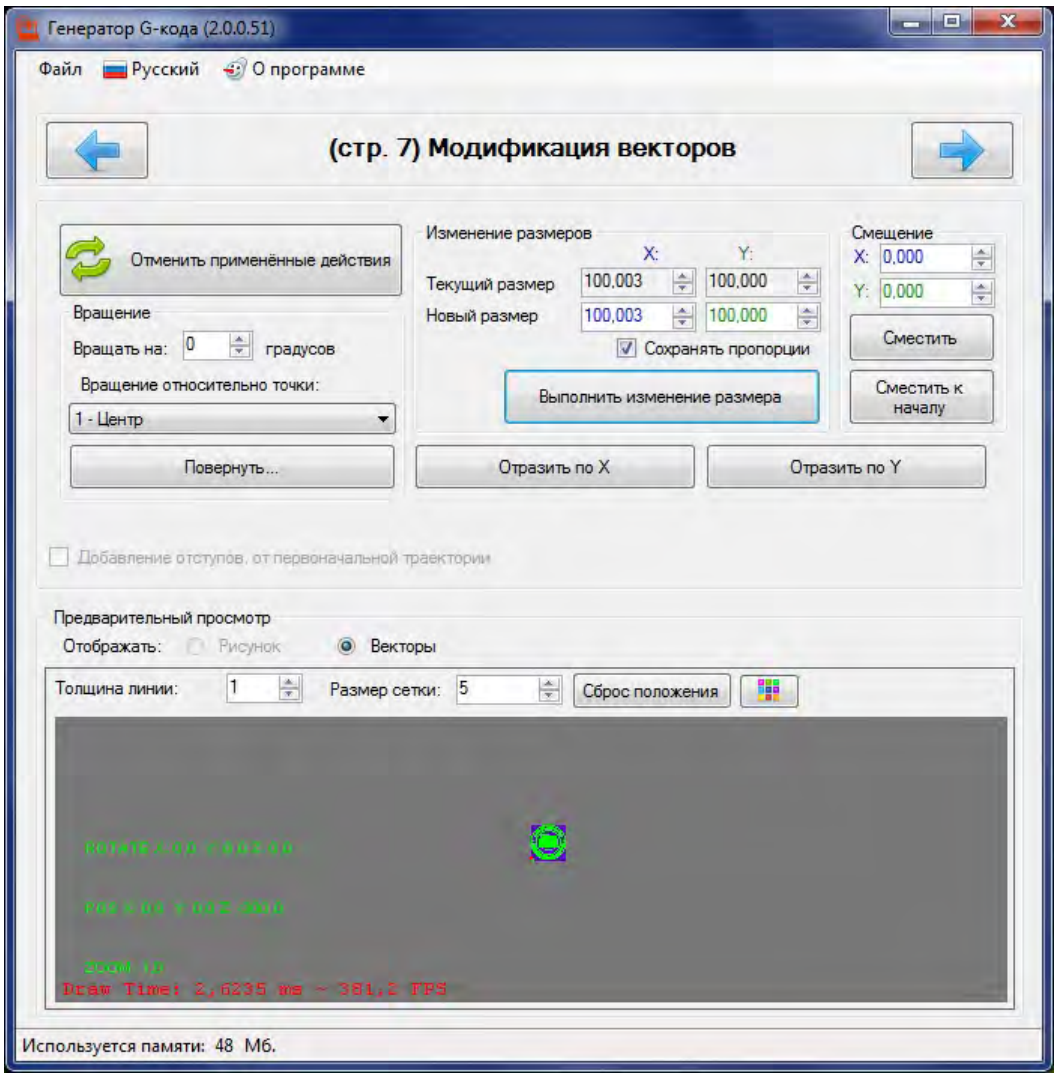


На следующем этапе в окне «стр. 7» можно осуществить модификацию ряда параметров траектории, в частности, можно изменить размеры, установить смещение от первоначального положения, повернуть относительно первоначального положения, а также зеркально отразить относительно оси *X* или оси *Y*.

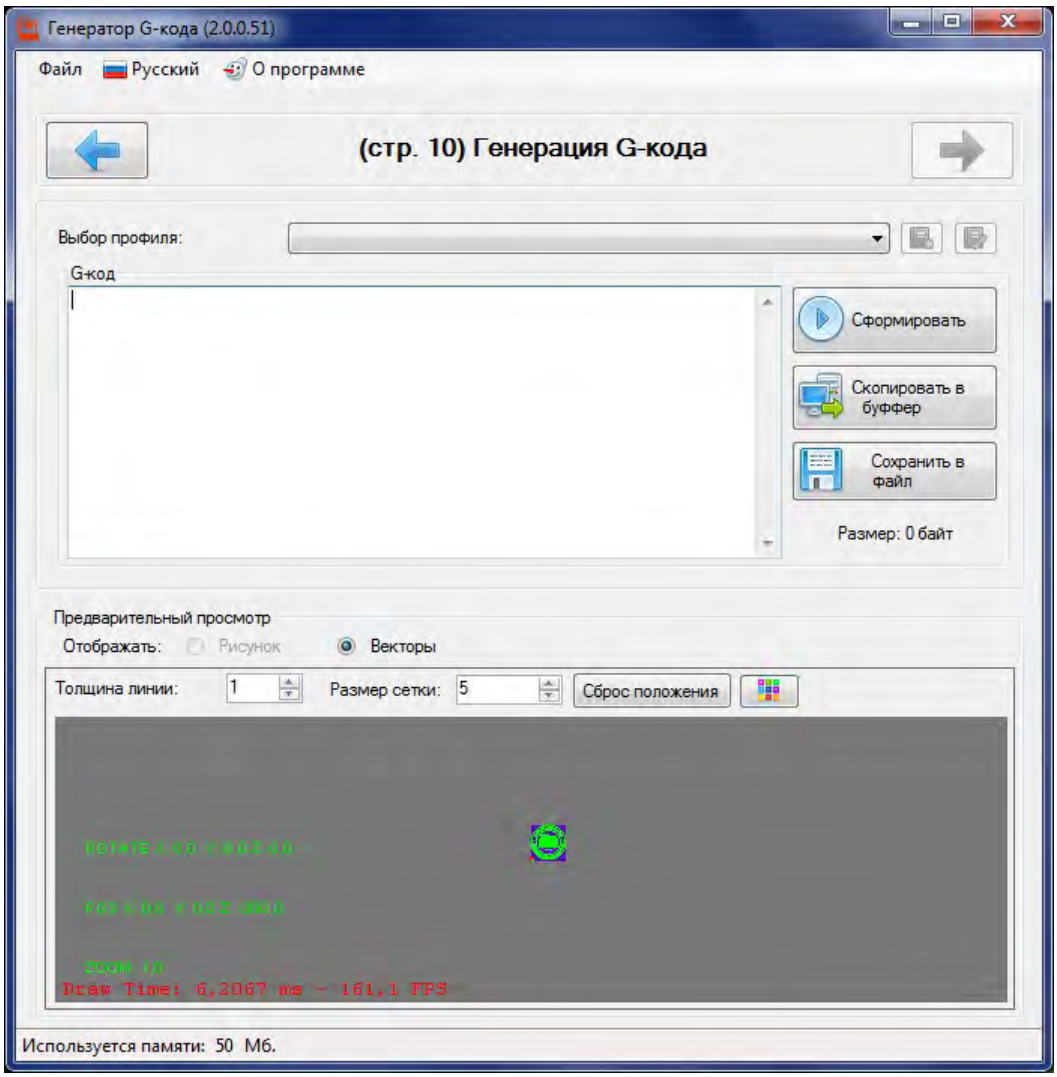


В качестве примера изменим первоначальные размеры обрабатываемого профиля со значений 370 мм по обоим осям, установив новые значения по обоим осям в 100 мм.

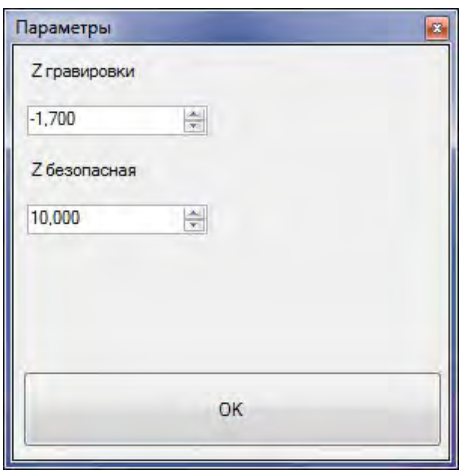
После завершения модификации необходимо нажать кнопку «Далее».



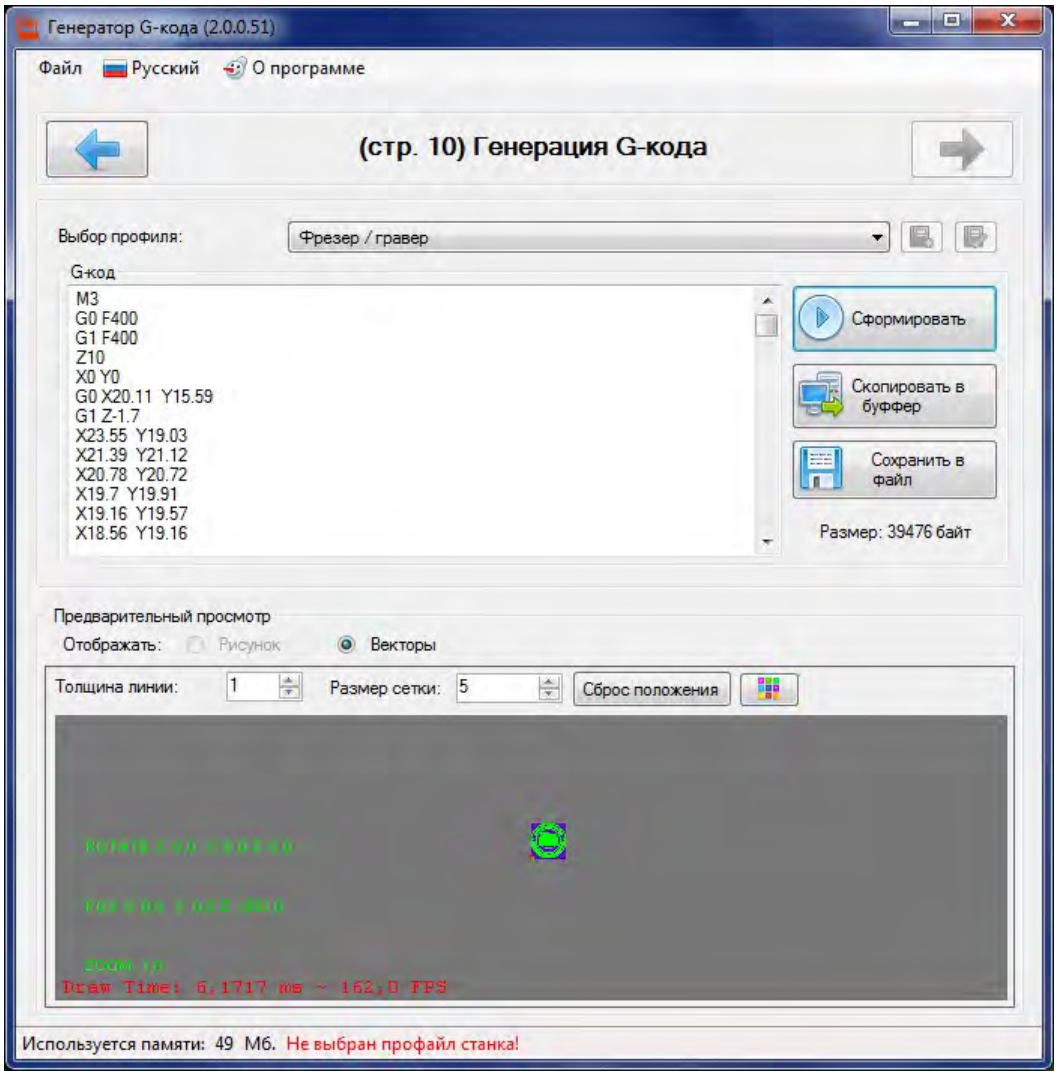
На следующем этапе в окне «стр. 10» можно сгенерировать код управляющей программы.



Для генерации кода управляющей программы необходимо выбрать профиль (для рассматриваемого примера необходимо использовать профиль «фрезер/гравер»). При этом откроется окно параметров профиля, в котором можно установить требуемые значения параметров. В профиле «фрезер/гравер» необходимо установить значения перемещений по оси Z — глубину гравирования (Z гравировки, в примере, –1,7 мм) и высоту отвода инструмента (Z безопасная, в примере, 10 мм).

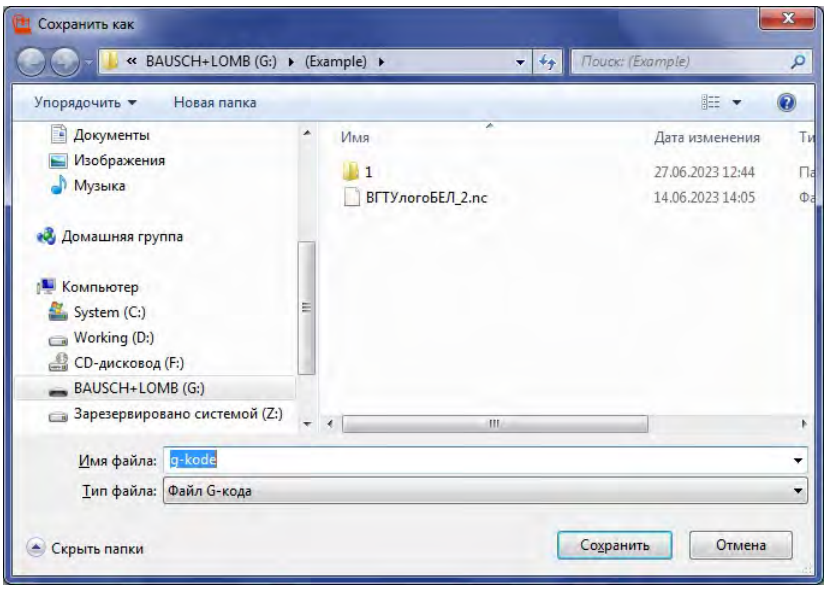


После чего, выбрав команду «Сформировать», будет осуществлена автоматическая генерация кода управляющей программы, который отобразится в соответствующей части окна «стр. 10».



Полученный текст управляющей программы можно скопировать в буфер обмена или сохранить в файл.

При сохранении управляющей программы в файл откроется окно, в котором необходимо указать имя файла и выбрать место его сохранения.



Далее используя полученную управляющую программу можно осуществить обработку изделия (в примере, логотипа университета) на лабораторной установке.

Для управления лабораторной установкой можно использовать свободно распространяемый программный продукт gbrlControl (рис. 4.2).

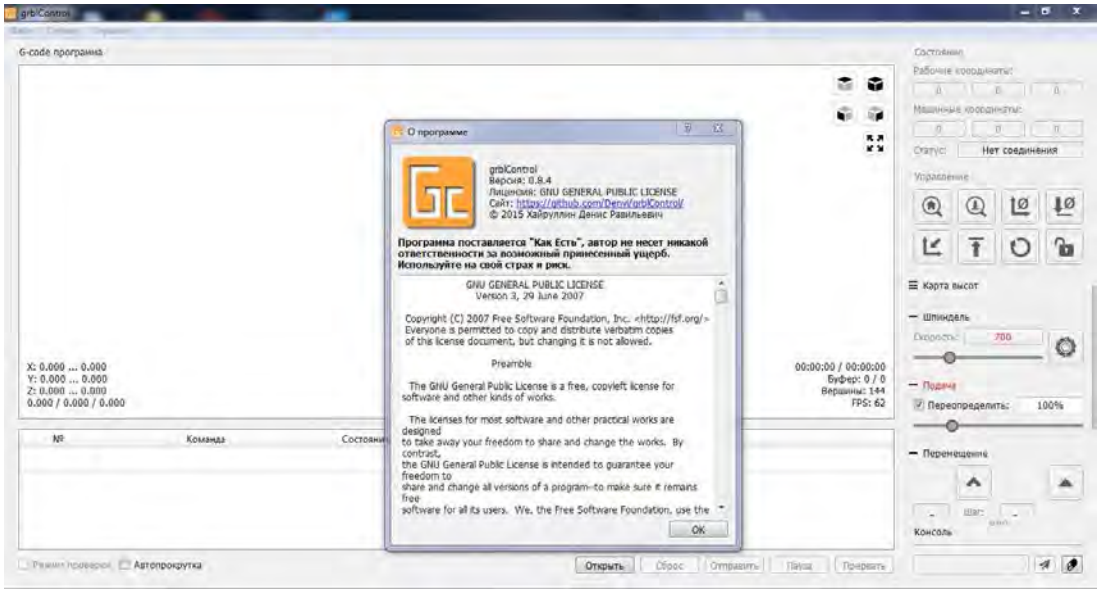
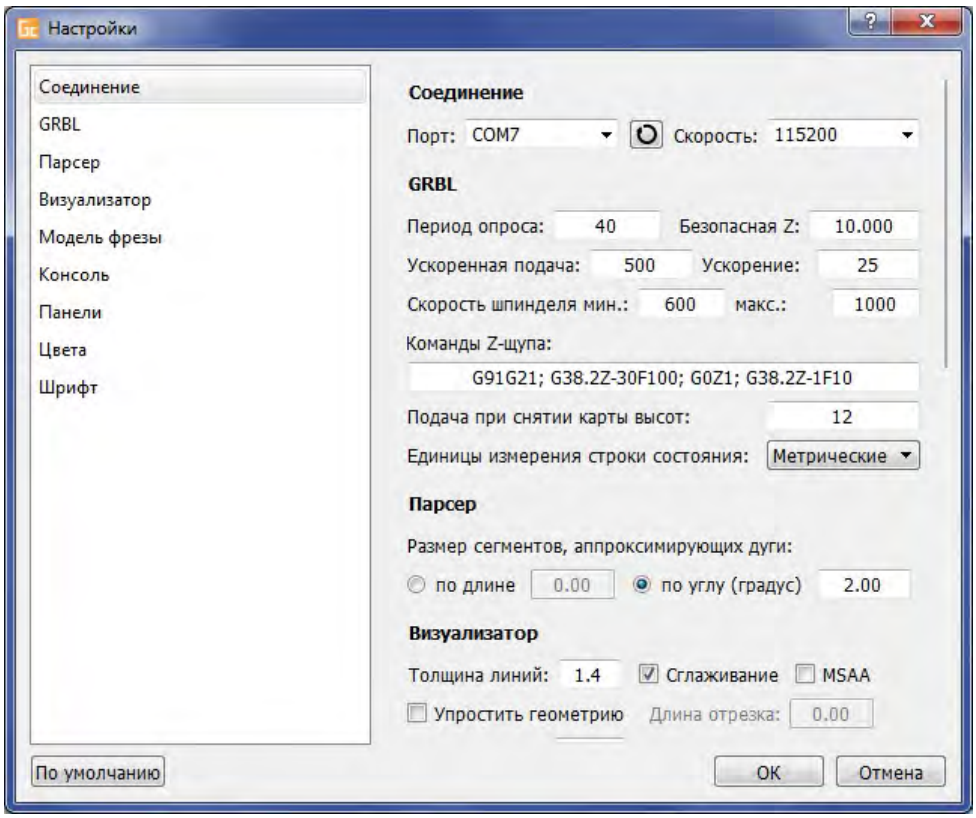


Рисунок 4.2 — Окно программы gbrlControl (с окном «О программе»)

ВНИМАНИЕ! Перед запуском программы целесообразно убедиться в подключении лабораторной установки к сети, используя внешний блок питания, и к компьютеру, используя кабель с USB-разъемом.

Для установки значений параметров подключения лабораторной установки к компьютеру необходимо выбрать Сервис/Настройки и в открывшемся окне установить соответствующие значения параметров.



Для активного мониторинга состояния и управления лабораторной установкой в правой части основного окна программы размещены поля отображения координат положения инструмента (рабочие координаты, машинные координаты) и статуса подключенной установки (в примере, «Готов»), а также доступны команды управления и ручного перемещения.

Команды управления:



Домой; перемещение шпинделя в исходную точку;

Z-щуп;

Обнулить XY;

Обнулить Z;

Восстановить XYZ;

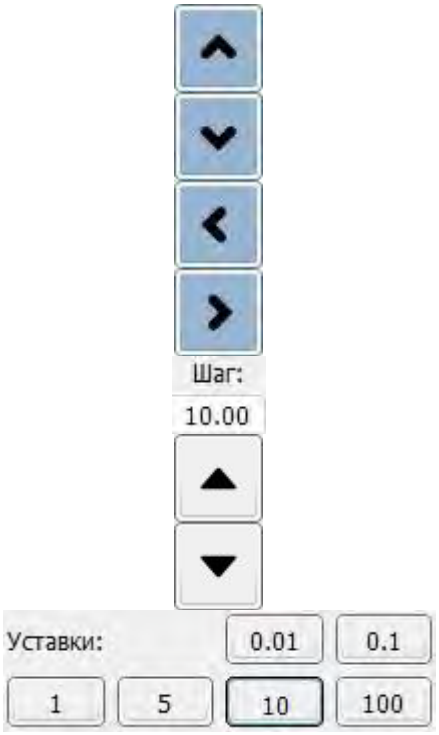
Безопасная Z;

Сброс;

Разблокировать;

Вкл/выкл шпиндель; включение/выключение вращения шпинделя.

Команды ручного перемещения:



Y+; перемещение стола «Вперед»;

Y–; перемещение стола «Назад»;

X–; перемещение стола «Влево»;

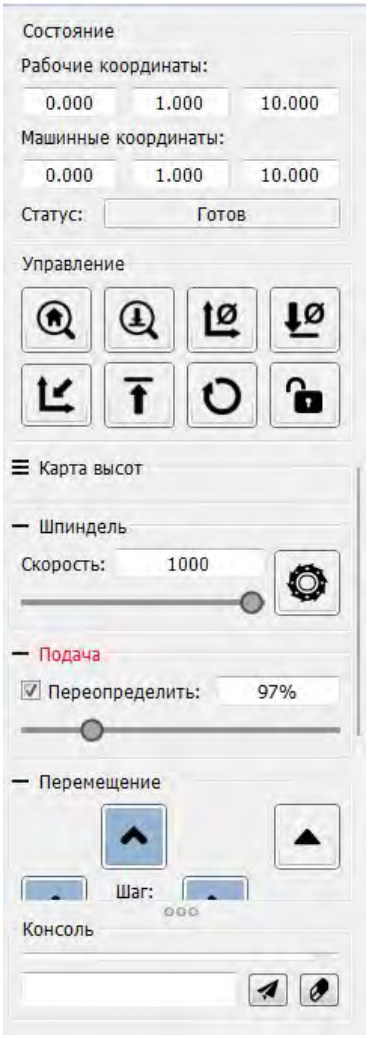
X+; перемещение стола «Вправо»;

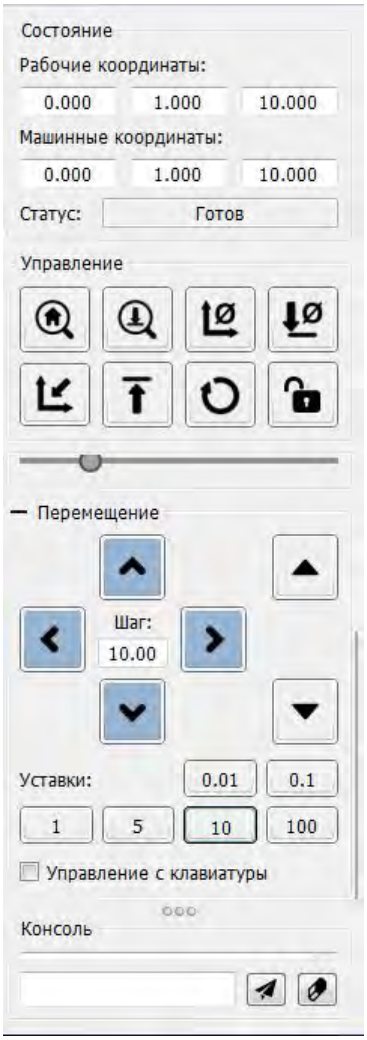
Дискретность перемещения стола, мм;

Z+; перемещение шпинделя «Вверх»;

Z–; перемещение шпинделя «Вниз»;

Дискретность перемещения шпинделя, мм.

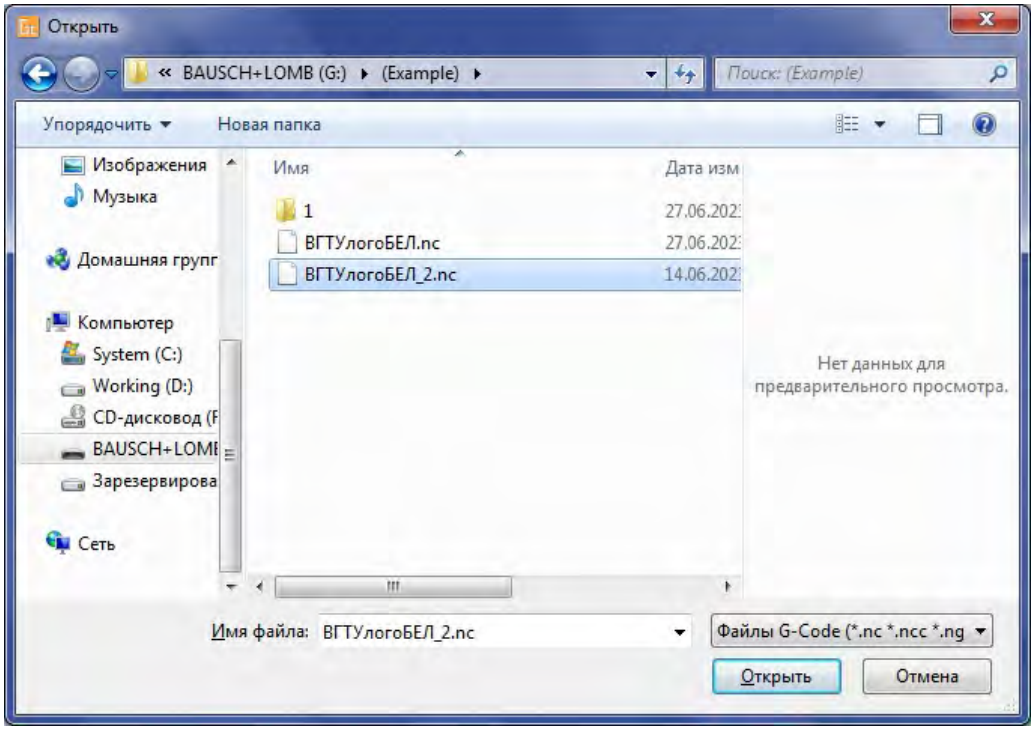




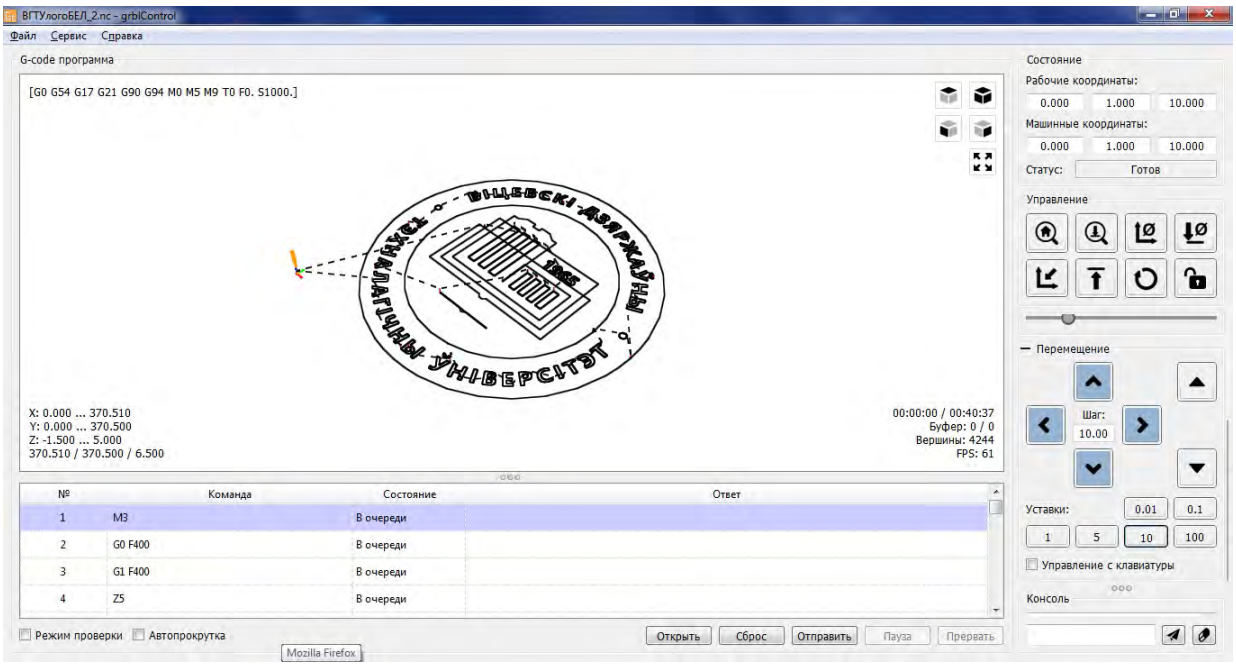
Перед запуском обработки необходимо, используя инструменты управления, выставить первоначальное положение инструмента относительно обрабатываемой заготовки. Для этого используются кнопки перемещения стола «Вперед», «Назад», «Влево», «Вправо» с заданной дискретностью перемещений (в примере, шаг 10 мм) и кнопки перемещения шпиндель «Вверх», «Вниз» с заданной дискретностью перемещения (в примере, уставки 10 мм).

После завершения установки первоначального положения необходимо переопределить положения центра координат кнопками <Обнулить XY>, <Обнулить Z>.

Далее необходимо загрузить ранее полученную управляющую программу, для чего можно воспользоваться кнопкой «Открыть» в нижней части основного окна. В открывшемся окне необходимо выбрать соответствующий файл управляющей программы и подтвердить выбор кнопкой «Открыть».



После чего в окне симуляции обработки появиться обрабатываемый контур и траектории перемещения инструмента. Для начала процесса обработки необходимо нажать кнопку «Отправить», после чего лабораторная установка перейдет в режим автоматической обработки с отображением в окне симуляции текущего этапа обработки и соответствующего кадра управляющей программы.



Для временной приостановки процесса обработки можно использовать кнопку «Пауза», для прекращения обработки — кнопку «Прервать».

4.3 Содержание работы

В процессе работы необходимо решить поставленную задачу в соответствии с индивидуальным заданием и оформить отчет с выводами.

Порядок выполнения работы

1. Изучить теоретическую часть.
2. По заданному растровому изображению разработать изображение в векторном формате.
3. На основе полученного векторного изображения разработать соответствующую управляющую программу, результаты сохранить в формате g-code.
4. Выполнить анализ полученной управляющей программы в формате g-code.
5. Загрузить полученную управляющую программу в лабораторную установку и осуществить обработку изделия.

СПИСОК РЕКОМЕНДУЕМЫХ ИСТОЧНИКОВ

1. Устройства числового программного управления для металлообрабатывающего оборудования. Кодирование информации управляющих программ : ГОСТ 20999–83. — 1983. — Введ. 1984–07–01. — Москва : Государственный комитет СССР по стандартам, 1983. — 28 с.

2. G-code / Wikipedia [Электронный ресурс]. — Режим доступа : https://ru.wikipedia.org/wiki/G-code.

3. Системы управления технологическим оборудованием / Виртуальная образовательная среда ВГТУ [Электронный ресурс]. — Режим доступа : https://sdo.vstu.by/course/view.php?id=2103.

ПРИЛОЖЕНИЕ А
Краткий справочник по основным командам

Основные (подготовительные) команды (G, General)

– перемещение рабочих органов оборудования с заданной скоростью (линейное и круговое);

– выполнение типовых последовательностей (таких, как обработка отверстий и резьба);

– управление параметрами инструмента, системами координат, и рабочих плоскостей.

Код	Описание
G00-G03	Позиционирование инструмента
G17-G19	Переключение рабочих плоскостей (XY, ZX, YZ)
G20-G21	Не стандартизовано
G40-G44	Компенсация размера различных частей инструмента (длина, диаметр)
G53-G59	Переключение систем координат
G80-G85	Циклы сверления, растачивания, нарезания резьбы
G90-G91	Переключение систем координат (абсолютная, относительная)

Код	Описание	Пример
G00	Ускоренное перемещение инструмента (холостой ход). При холостом перемещении *НЕ ОБЯЗАТЕЛЬНО* производится линейная интерполяция перемещения аналогично команде G01. В некоторых интерпретаторах при выполнении команды перемещения по нескольким осям одновременно, перемещение по осям отрабатывается с максимальной скоростью, поэтому линейное перемещение от исходной точки в конечную точку не обеспечивается, поэтому нельзя производить ходы обработки детали при действии этой модальной команды.	G0 X0 Y0 Z100.
G01	Линейная интерполяция, модальная команда. Инструмент (рабочий орган) перемещается по отрезку прямой линии от исходной точки с координатами до исполнения команды в точку с заданными в команде координатами, скорость перемещения задаётся здесь же или ранее модальной командой F.	G01 X0. Y0. Z100. F200.
G02	Круговая интерполяция по часовой стрелке, модальная команда. Инструмент перемещается по дуге окружности по часовой стрелке от исходной точки с координатами до исполнения команды в точку с заданными в команде координатами, скорость перемещения задаётся в этой команде параметром F, радиус дуги задаётся параметром R, либо указанием координат центра дуги	G02 G17 X15. Y15. R5. F200. Или G02 G17 X20. Y15. I-50. J-60.

Код	Описание	Пример
	параметрами I — (смещение центра по оси X относительно начальной координаты X), J — (смещение центра по оси Y относительно начальной координаты Y), K — (смещение центра по оси Z относительно начальной координаты Z) относительно начальных координат инструмента. Для указания плоскости, в которой производится круговая интерполяция, предварительно должна быть указана плоскость круговой интерполяции (в этом же или в другом предварительном кадре) модальной командой G17 (плоскость X-Y), или G18 (плоскость X-Z), или G19 (плоскость Y-Z). Скорость перемещения задана модальной командой F.	
G03	Круговая интерполяция против часовой стрелки. Параметры и действие аналогичны команде G02.	G03 X15. Y15. R5. F200.
G04	Задержка выполнения программы, способ задания величины задержки зависит от реализации системы управления, P обычно задает паузу в миллисекундах, X — в секундах. В некоторых интерпретаторах P задает паузу в секундах и параметр X в этой команде не используется. Также в некоторых интерпретаторах возможно задание задержки параметром U.	G04 P500 или G04 X.5
G10	Переключение абсолютной системы координат. В примере начало координат станет в точке 10, 10, 10 старых координат.	G10 X10. Y10. Z10.
G15	Переход в полярную (цилиндрическую) систему координат. В этой системе параметр X задаёт радиус, а Y угол в градусах. Если включена абсолютная система координат (G90), то начало полярных координат будет в точке текущих координат 0; 0, если включена инкрементная система координат, то начало координат будет в точке, достигнутой при отработке предыдущего кадра.	G15 X15. Y22.5
G16	Отмена полярной системы координат.	G16 X15. Y22.5
G17	Выбор рабочей плоскости X-Y.	G17
G18	Выбор рабочей плоскости Z-X.	G18
G19	Выбор рабочей плоскости Y-Z.	G19
G20	Режим работы в дюймовой системе.	G90 G20
G21	Режим работы в метрической системе.	G90 G21
G22	Активировать установленный предел перемещений (Инструмент не выйдет за их предел).	G22 G01 X15. Y25.
G28	Вернуться на референтную точку.	G28 G91 Z0 Y0
G30	Поднятие по оси Z на точку смены инструмента.	G30 G91 Z0

Код	Описание	Пример
G40	Отмена компенсации радиуса инструмента.	G1 G40 X0. Y0. F200.
G41	Компенсировать радиус инструмента слева от траектории.	G41 X15. Y15. D1 F100.
G42	Компенсировать радиус инструмента справа от траектории.	G42 X15. Y15. D1 F100.
G43	Компенсировать длину инструмента в положительную сторону. В основном применяется при смене инструмента.	G43 X15. Y15. Z100. H1 S1000 M3
G44	Компенсировать длину инструмента в отрицательную сторону. Действие аналогично G43.	G44 X15. Y15. Z4. H1 S1000 M3
G49	Отмена компенсации длины инструмента.	G49 Z100.
G50	Сброс всех масштабирующих коэффициентов в 1,0.	G50
G51	Назначение масштабов. В примере — уменьшение масштаба по оси X в 10 раз. После этой модальной команды все указанные в командах перемещения и координаты по оси X будут умножаться на масштабирующий коэффициент 0,1 и результат интерпретироваться как требуемое перемещение. Если задать масштабирующий коэффициент по некоторой оси (или по любым осям) равным −1, то последующее движение будет зеркальным по этой оси (или осям, где масштабирующий коэффициент −1).	G51 X.1 или G51 X-1
G53	Переход в систему координат станка.	G53 G0 X0. Y0. Z0.
G54-G59	Переключиться на заданную оператором систему координат.	G54 G0 X0. Y0. Z100.
G61-G64	Переключение режимов Точный Стоп/Постоянная скорость.	-
G68	Поворот координат на нужный угол.	G68 X0 Y0 R45.
G70	Цикл продольного чистового точения.	G70 P10 Q15.
G71	Цикл многопроходного продольного чернового точения.	G71 P10 Q15. D.5 U.2 W.5
G80	Отмена циклов сверления, растачивания, нарезания резьбы метчиком и т. д.	G80
G81	Цикл сверления.	G81 X0 Y0. Z-10. R3. F100.
G82	Цикл сверления с задержкой.	G82 X0. Y0. Z-10. R3. P100 F100.
G83	Цикл прерывистого сверления (с периодическим полным выводом сверла). Параметр Z указывает полную глубину сверления от поверхности (Z=0), R — высота вывода инструмента над поверхностью для вывода стружки и также конечное положение после завершения сверления, Q — величина заглубления одного из нескольких заглублений при сверлении, F — скорость подачи (необязательна, при отсутствии этого параметра скорость определяется ранее заданной скоростью в команде G1.	G83 Z-20 R1 Q2 F20

Код	Описание	Пример
G84	Цикл нарезания резьбы.	G95 G84 M29 X0. Y0. Z-10. R3 F1.411
G90	Задание абсолютных координат опорных точек траектории.	G90 G1 X0.5. Y0.5. F10.
G91	Задание координат инкрементально относительно координат последней введённой опорной точки, перемещение инструмента в этой системе координат задаётся в виде приращений.	G91 G1 X4. Y5. F100.
G94	F (подача) — в формате мм/мин.	G94 G80 Z100. F75.
G95	F (подача) — в формате мм/об.	G95 G84 X0. Y0. Z-10. R3 F1.411
G99	После каждого цикла не отходить на <проходную точку>	G99 G91 X10. K4.

Дополнительные (технологические) команды (M, Miscellaneous)

- сменить инструмент;
- включить/выключить шпиндель;
- включить/выключить охлаждение;
- работа с подпрограммами;
- и пр.

Код	Описание	Пример
M00	Приостановить работу станка до нажатия кнопки <старт> на пульте управления, так называемая <безусловная технологическая остановка>	G0 X0 Y0 Z100 M0
M01	Приостановить работу станка до нажатия кнопки <старт>, если включён режим подтверждения остановки. Если этот режим отключён, то команда игнорируется. Используется для начальной проверки (отладки) кода.	G0 X0 Y0 Z100 M1
M02	Конец программы, без сброса модальных функций. Указатель номера кадра не изменяется.	M02
M03	Начать вращение шпинделя по часовой стрелке	M3 S2000
M04	Начать вращение шпинделя против часовой стрелки	M4 S2000
M05	становить вращение шпинделя	M5
M06	Сменить инструмент	T15 M6
M07	Включить дополнительное охлаждение	M3 S2000 M7
M08	Включить основное охлаждение. Иногда использование более одного М-кода в одной строке (как в примере) недопустимо, для этого используются M13 и M14	M3 S2000 M8
M09	Выключить охлаждение	G0 X0 Y0 Z100 M5 M9
M13	включить одновременно охлаждение и вращение шпинделя по часовой стрелке	S2000 M13
M14	Включить одновременно охлаждение и вращение шпинделя против часовой стрелки	S2000 M14
M17	Возврат из подпрограммы или из макроса	M17

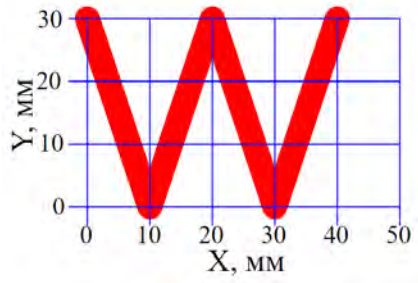
Код	Описание	Пример
	(действие аналогично M99)	
M48	Разрешить переопределять скорость подачи	
M49	Запретить переопределение скорости подачи	
M25	Замена инструмента вручную	M25
M97	Запуск подпрограммы, находящейся в той же программе (где P — номер кадра, в примере переход осуществится к строке с меткой N25), *реализована не во всех интерпретаторах*, предположительно — только на станках HAAS	M97 P25
M98	Запуск подпрограммы, находящейся отдельно от основной программы (где P — номер подпрограммы, в примере переход осуществится к программе O1015)	M98 P1015
M99	Конец подпрограммы и переход в вызвавшую программу	M99
M30	Конец программы, со сбросом модальных функций и изменением указателя номера кадра на начало программы	M30

Параметры команд

Код	Описание	Пример
X	Перемещение инструмента в заданную точку с заданной координатой по оси X при работе в абсолютной системе координат (см. G90) или задание смещений относительно точки, достигнутой в предыдущем кадре при работе в инкрементной системе координат (см. G91)	G0 X100 Y0 Z0
Y	Аналогично предыдущему по оси Y	G0 X0 Y100 Z0
Z	Аналогично предыдущему по оси Z	G0 X0 Y0 Z100
P	При использовании в команде вызова подпрограммы (M98) — указание номера вызываемой подпрограммы с именем, заданным после буквы O, например, P301 вызовет подпрограмму с меткой O301. При использовании в команде задержки (G04) указывает время задержки в миллисекундах.	G04 P500; M98 P301
O	Метка подпрограммы с указанным номером	O301
F	Линейная скорость перемещения инструмента. Для фрезерных станков это дюймы в минуту (IPM) или миллиметры в минуту (мм/мин), Для токарных станков это дюймы за оборот (IPR) или миллиметры за оборот (мм/об). Выбор единиц измерения, дюймы или миллиметры выполняется командами G20 и G21.	G1 G91 X10 F100
S	Частота вращения шпинделя в оборотах в минуту.	S3000 M3
T	Указание номера инструмента в команде смены инструмента. Обычно указывается перед командой M6.	T1 M6

Код	Описание	Пример
R	Расстояние отвода инструмента в повторяющихся циклах обработки, например, прерывистого сверления глубоких отверстий (G81-G89) или радиус дуги при круговых интерполяциях перемещения инструмента (G02, G03).	G81 Z-20 R2 или G2 G91 X12.5 R12.5
D	Параметр коррекции радиуса выбранного инструмента.	G1 G41 D1 X10. F150.
L	Число вызовов подпрограммы, число вызовов макроса, или количество циклов в повторяющихся операциях X_Y_R_ — параметры, передаваемые в макрос.	M98 L82 P10 или G65 L82 P10 X_Y_R_
I	Указание смещения по оси X координаты центра дуги при круговой интерполяции перемещения инструмента (см G02, G03). Координаты центра дуги по осям указываются в виде смещения относительно начальной точки (достигнутой в предыдущем кадре). Плоскость интерполяции (плоскость, которая параллельна заданной координатной плоскости указывается командами G17, G18, G19.	G03 X10 Y10 I10 J0 F10
J	Аналогично предыдущему для оси Y.	G03 X10 Y10 I0 J10 F10
K	Аналогично предыдущему для оси Z.	G03 X10 Y10 I0 K0 F10

Пример гравировки буквы W на глубину 2 мм, вписанной в прямоугольник 40×30 мм, (см. рисунок) на условном вертикально-фрезерном станке с ЧПУ в листовой заготовке концевой фрезой диаметром 2 мм.
[https://ru.wikipedia.org/wiki/G-code]




```
%
O200
G21 G40 G49 G53 G80 G90 G17
```

```
(метка начала программы, необязательна)
(ось Z настроена так, что при Z=0 инструмент
касается поверхности заготовки)
(метка программы, необязательна)
(Строка безопасности.)
(Состояние станка или интерпретатора определяется
предысторией, либо устанавливаются в  некоторое
исходное состояние при включении питания, и эти
настройки могут вызвать нежелательные и
непредвиденные действия, поэтому необходимо
привести станок в <исходное состояние> с помощью
«строки безопасности».)
(G21 — выбор метрической системы единиц -
миллиметры,)
(G40 — отменяет автоматическую коррекцию на радиус
инструмента.)
```

```
                        (G49 — отменяет автоматическую коррекцию на длину
                        инструмента.)
                        (G53 — отменяет возможно введённые ранее
                        дополнительные системы координат, смещённые
                        относительно исходной и переводит станок в основную
                        систему координат.)
                        (G80 — отменяет все постоянные циклы, например,
                        циклы сверления и их параметры.)
                        (G90 — переводит в абсолютную систему координат.)
                        (G17 — выбирается плоскость круговой интерполяции
                        X-Y.)
G0 F300                 (задание скорости холостого перемещения инструмента
                        в мм/мин)
M3 S500                 (включение вращения шпинделя по часовой стрелке и
                        задание его скорости вращения 500 об/мин)
G4 P2000                (выдержка 2 секунды для раскрутки шпинделя)
X0 Y30 Z5               (подвод инструмента в точку с координатами X=0 Y=30
                        Z=5 со скоростью холостого перемещения)
G1 Z-2 F40              (врезание в заготовку на глубину 2 мм со скоростью
                        40 мм/мин)
G1 F20 X10 Y0           (фрезерование 1-го отрезка буквы W со скоростью 20
                        мм/мин)
X20 Y30                 (фрезерование 2-го отрезка буквы W со скоростью 20
                        мм/мин)
X30 Y0                  (фрезерование 3-го отрезка буквы W со скоростью 20
                        мм/мин)
X40 Y30                 (фрезерование 4-го отрезка буквы W со скоростью 20
                        мм/мин)
G0 Z5                   (отвод инструмента на высоту 5 мм над поверхностью
                        заготовки со скоростью 300 мм/мин)
M5                      (выключение вращения шпинделя)
M30                     (конец программы и конец интерпретируемого кода)
```

Учебное издание

СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ОБОРУДОВАНИЕМ

Методические указания по выполнению лабораторных работ

Составитель:

Климентьев Андрей Леонидович

Редактор *А.В. Пухальская*
Корректор *А.В. Пухальская*
Компьютерная верстка *А.Л. Климентьев*

Подписано к печати 01.09.2023. Формат 60x90 $^{1}/_{16}$. Усл. печ. листов 3,4.
Уч.-изд. листов 4,4. Тираж 2 экз. Заказ № 222.

Учреждение образования «Витебский государственный технологический университет»
210038, г. Витебск, Московский пр., 72.
Отпечатано на ризографе учреждения образования
«Витебский государственный технологический университет».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/172 от 12 февраля 2014 г.
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 3/1497 от 30 мая 2017 г.

Учебное издание

СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ОБОРУДОВАНИЕМ

Методические указания по выполнению лабораторных работ

Составитель:

Климентьев Андрей Леонидович

Редактор *А.В. Пухальская*
Корректор *А.В. Пухальская*
Компьютерная верстка *А.Л. Климентьев*

Подписано к печати 01.09.2023. Усл. печ. листов 3,4.
Уч.-изд. листов 4,4. Заказ № 223.

Учреждение образования «Витебский государственный технологический университет»
210038, г. Витебск, Московский пр., 72.
Отпечатано на ризографе учреждения образования
«Витебский государственный технологический университет».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/172 от 12 февраля 2014 г.
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 3/1497 от 30 мая 2017 г.