Министерство науки и высшего образования Российской Федерации

Чайковский филиал

федерального государственного автономного образовательного учреждения

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра Автоматизации, информационных и инженерных технологий

Направление: 09.03.01 Информатика и вычислительная техника

Направленность: Автоматизированные системы обработки информации и управления

**КУРСОВАЯ РАБОТА**

на тему «Теоретический обзор аппаратных средств ЭВМ и периферийных устройств. Работа с прерываниями в Ассемблере. Вариант №11»

по дисциплине ЭВМ и периферийные устройства

|  |  |  |
| --- | --- | --- |
|  |  | Выполнил: |
|  |  | студент группы АСУ-22-1б ЧФ |
|  |  | Кузнецов Никита Иванович |
|  |  | *(Фамилия Имя Отчество)* |
|  |  | Руководитель: |
|  |  | старший преподаватель |
|  |  | *(должность, ученая степень, ученое звание)* |
|  |  | Сухих Илья Игоревич |
|  |  | *(Фамилия Имя Отчество)* |
|  |  | *Оценка*  *«\_\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_\_г* |
|  |  |  |
|  |  | *(подпись)* |

Чайковский 2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc185085509)

[1 Теоретическая часть 5](#_Toc185085510)

[1.1 История изобретения Flash-памяти 5](#_Toc185085511)

[1.2 Принцип действия Flash-памяти 5](#_Toc185085512)

[1.3 Основные характеристики 7](#_Toc185085513)

[1.4 Области применения Flash-памяти 10](#_Toc185085514)

[1.5 Преимущества и недостатки Flash-памяти 12](#_Toc185085515)

[2 Практическая часть 13](#_Toc185085516)

[3 Реализация полученных знаний в ассемблере, фигура – усеченная пирамида 7 граней 16](#_Toc185085517)

[ЗАКЛЮЧЕНИЕ 19](#_Toc185085518)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 20](#_Toc185085519)

[Приложение А «Код программы построения фигуры усеченная пирамида 7 граней» 22](#_Toc185085520)

# **ВВЕДЕНИЕ**

Современное общество характеризуется стремительным развитием цифровых технологий, которые во многом определяют качество жизни, уровень взаимодействия и доступ к информации. Центральное место в этом процессе занимает технология хранения данных, позволяющая эффективно сохранять, обрабатывать и передавать информацию. Одной из наиболее востребованных и широко применяемых технологий хранения данных в последние десятилетия является Flash-память.

С момента своего появления Flash-память прочно заняла позиции в самых различных устройствах – от простых USB-накопителей до сложных систем хранения данных в дата-центрах. Её энергонезависимость, компактность, высокая производительность и устойчивость к физическим воздействиям сделали её ключевым элементом в развитии мобильных технологий, облачных сервисов и встраиваемых систем.

Актуальность данной темы обусловлена несколькими факторами:

* **Широким распространением Flash-памяти.** Она используется в таких повседневных устройствах, как смартфоны, ноутбуки, камеры, а также в специализированных областях, включая автомобильные и медицинские системы.
* **Быстрым технологическим развитием.** Flash-память прошла путь от плоской (2D) архитектуры до 3D NAND-технологий, что позволило значительно увеличить плотность хранения данных.
* **Растущими объёмами данных.** В эпоху больших данных и интернета вещей (IoT) необходимость в энергоэффективных и ёмких решениях для хранения информации становится всё более очевидной.

Цель данной работы заключается в изучении принципов действия Flash-памяти, её технических характеристик и основных областей применения. Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить физические и технологические основы работы Flash-памяти.
2. Рассмотреть основные характеристики Flash-памяти, включая её производительность, надёжность и долговечность.
3. Проанализировать области применения Flash-памяти в потребительской и промышленной электронике.
4. Подвести итоги анализа и оценить перспективы развития технологии.

Практическая часть курсовой работы включает написание программы на языке ассемблера. Она будет демонстрировать принципы низкоуровневого программирования через реализацию функции рисования трехмерной усеченной пирамиды с семью гранями.

# **1 Теоретическая часть**

## ****1.1 История изобретения Flash-памяти****

Первый патент, связанный с флеш-памятью, был зарегистрирован в США инженером Toshiba по имени Фуцзио Масуока, который придумал эту идею вместе со своим коллегой Хисаказу Иизукой еще в 1981. Название «flash» («вспышка») было придумано также в Toshiba — Сёдзи Ариидзуми, который сравнил быстроту процессов записи и чтения, с быстротой работы вспышки камеры.

В то время передовые чипы EPROM стирали с помощью луча ультрафиолетового света, а весь процесс занимал около 20 минут и требовал использования прозрачного корпуса. Придумали и тип EEPROM, информацию в котором можно было стирать электрическим путем, но в нем за каждый бит отвечали два транзистора.

Первые чипы флеш-памяти, которые напоминают современные, выпустила та же Toshiba в 1987 году. Популярными устройства на ее основе, однако, начали становиться лишь в конце девяностых, когда во всевозможных портативных девайсах начали использовать карты памяти от Toshiba, Matsushita и SanDisk. В 1988 году Intel выпустила первый коммерческий флеш-чип NOR-типа. NAND-тип флеш-памяти был анонсирован Toshiba в 1989 году на International Solid-State Circuits Conference.

**1.2** **Принцип действия Flash-памяти**

Основным компонентом в флеш-памяти является транзистор с плавающим затвором, который является разновидностью МОП-транзисторов. Его отличие в том, что у него есть дополнительный затвор (плавающий), расположенный между управляющим затвором и p-слоем. Плавающий затвор изолирован, и хранимый в нём отрицательный заряд будет оставаться надолго. Пример транзистора с помавающим затвором представлен ан рисунке 1.



Рисунок 1 - Транзистор с плавающим затвором

Запись данных в ячейку памяти осуществляется путём изменения заряда на плавающем затворе. Этот процесс включает 2 этапа.

* Эффект горячих электронов. При приложении высокого напряжения электроны из канала транзистора «перескакивают» через изолирующий слой и накапливаются на плавающем затворе. Этот заряд создаёт электрическое поле, которое изменяет пороговое напряжение транзистора.
* Программирование напряжением. Для каждой ячейки создаются определённые уровни заряда, соответствующие различным значениям данных (например, 0 или 1 в случае SLC).

Чтение данных основывается на измерении порогового напряжения транзистора. Если заряд присутствует на плавающем затворе, пороговое напряжение будет выше. Если же заряда нет, то пороговое напряжение ниже. Для многослойных ячеек (MLC, TLC, QLC) различают несколько пороговых уровней, что позволяет считывать более одного бита из одной ячейки.

Флеш-память различается методом соединения ячеек в массив.

Конструкция NOR использует классическую двумерную матрицу проводников, в которой на пересечении строк и столбцов установлено по одной ячейке. При этом проводник строк подключался к стоку транзистора, а столбцов — ко второму затвору. Исток подключался к общей для всех подложке.

Конструкция NAND — трёхмерный массив. В основе та же самая матрица, что и в NOR, но вместо одного транзистора в каждом пересечении устанавливается столбец из последовательно включенных ячеек. В такой конструкции получается много затворных цепей в одном пересечении. Плотность компоновки можно резко увеличить (ведь к одной ячейке в столбце подходит только один проводник затвора), однако алгоритм доступа к ячейкам для чтения и записи заметно усложняется. Также в каждой линии установлено два МОП-транзистора: управляющий транзистор разрядной линии (англ. bit line select transistor), расположенный между столбцом ячеек и разрядной линией и управляющий транзистор заземления, расположенный перед землёй (англ. ground select transistor).

Технология NOR позволяет получить быстрый доступ индивидуально к каждой ячейке, однако площадь ячейки велика. Наоборот, NAND имеют малую площадь ячейки, но относительно длительный доступ сразу к большой группе ячеек. Соответственно, различается область применения: NOR используется как для непосредственной памяти программ микропроцессоров, так и для хранения небольших вспомогательных данных.

Названия NOR и NAND произошли по ассоциации схемы включения ячеек в массив со схемотехникой микросхем КМОП-логики — NOR- и NAND-элементов.

NAND чаще всего применяется для USB-флеш-накопителей, карт памяти, SSD; а NOR — во встраиваемых системах.

Существовали и другие варианты объединения ячеек в массив, но они не прижились.

## ****1.3 Основные характеристики****

Flash-память обладает несколькими основными характеристиками, такими как: плотность хранения, скорость работы, износостойкость и стоимость. Все эти характеристик в основном зависят от технологий хранения памяти. Технологий для хранения памяти 4 типа: SLC, MLC, TLC и QLC.

Плотность хранения определяет количество информации, которое можно сохранить в одной ячейке памяти. На рисунке 2 представлены возможные состояния ячейки памяти для разных видов памяти.



Рисунок 2 - Возможные состояния ячейки памяти для разных видов памяти.

Скорость работы включает в себя скорость чтения и скорость записи. Время стирания варьируется от единиц до сотен миллисекунд в зависимости от размера стираемого блока. Время записи — десятки-сотни микросекунд. Обычно время чтения для NOR-микросхем нормируется в десятки наносекунд. Для NAND-микросхем время чтения составляет десятки микросекунд.

Износостойкость Flash-памяти измеряется в количестве циклов записи и стирания. Изменение заряда сопряжено с накоплением необратимых изменений в структуре и потому количество записей для ячейки флеш-памяти ограничено. Типичные количества циклов стирания-записи составляют от тысячи или менее до десятков и сотен тысяч, в зависимости от типа памяти и технологического процесса. Гарантированный ресурс значительно более низок при хранении нескольких бит в ячейке (MLC и TLC). Для продления срока службы используются технологии равномерного распределения операций записи по всем ячейкам памяти и исправление ошибок чтения и записи, возникающих из-за деградации ячеек.

Стоимость так же зависит от технологии хранения памяти, но благодаря различным технологиям стоимость снижается. Для более детального анализа представлена таблица с основными характеристиками Flash-памяти (см. Таблица 1).

Таблица 1 - Основные характеристики Flash-памяти

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Технология хранения памяти | SLC | MLC | TLC | QLC |
| Плотность хранения (бит на ячейку) | 1 | 2 | 3 | 4 |
| Скорость чтения/записи | Высокая | Оптимальная | Низкая | Очень низкая |
| Износостойкость (циклы) | До 100 000 | До 10 000 | От 3000 до 5000 | Около 1000 |
| Стоимость | Высокая | Оптимальная | Низкая | Очень низкая |

Благодаря энергонезависимости, компактности, высокой скорости работы и устойчивости к физическим воздействиям Flash память имеет широкую область применения. Основные области применения:

* Хранение данных: USB-накопители, карты памяти, SSD-диски
* Мобильные устройства: смартфоны, умные часы, планшеты
* Промышленные и встраиваемые системы: микроконтроллеры, IoT-устройства
* Автомобили: регистраторы, бортовые компьютеры\
* Фото и видео: камеры, дроны
* Системы безопасности: электрозамки

## 1.4 Области применения Flash-памяти

Существует два основных способа применения флеш-памяти: как носитель информации для компьютеров и электронных гаджетов и как хранилище программного обеспечения («прошивки») цифровых устройств. Зачастую эти два применения совмещаются в одном устройстве. При хранении во флеш-памяти возможно простое обновление прошивок устройств в процессе эксплуатации.

NOR-флеш наиболее применима в устройствах с энергонезависимой памятью относительно небольшого объёма, требующих быстрого доступа по случайным адресам и с гарантией отсутствия сбойных элементов:

* Встраиваемая память программ однокристальных микроконтроллеров. Типовые объёмы — от 1 кбайта до 1 Мбайта.
* Стандартные микросхемы ПЗУ произвольного доступа для работы вместе с микропроцессором.
* Специализированные микросхемы начальной загрузки компьютеров (POST и BIOS), процессоров ЦОС и программируемой логики. Типовые объёмы — единицы и десятки мегабайт.
* Микросхемы хранения среднего размера данных, например, DataFlash. Обычно снабжаются интерфейсом SPI и упаковываются в миниатюрные корпуса. Типовые объёмы — от сотен кбайт до технологического максимума.

NAND флеш-память в первую очередь применяется во всевозможных мобильных носителях данных и устройствах, требующих для работы больших объёмов хранения. В основном, это USB-накопители и карты памяти всех типов, а также мобильные устройства, такие как телефоны, фотоаппараты, медиаплееры. На рисунке 3 представлена Flash-память внутри Flash накопителя.

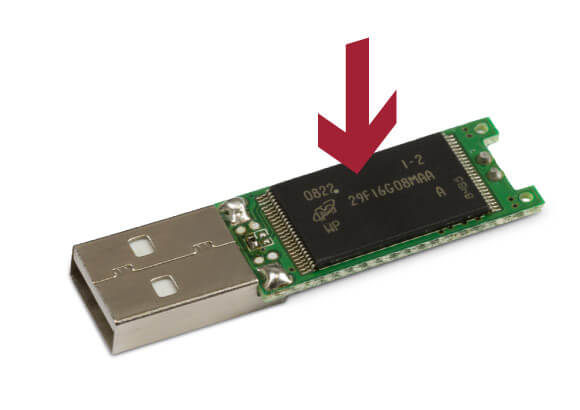


Рисунок 3 - Flash-память внутри Flash накопителя.

Флеш-память типа NAND позволила миниатюризировать и удешевить вычислительные платформы на базе стандартных операционных систем с развитым программным обеспечением. Их стали встраивать во множество бытовых приборов: телевизоры, сетевые маршрутизаторы и точки доступа, медиаплееры и игровые приставки, фоторамки и навигаторы.

Высокая скорость чтения делает NAND-память привлекательной для кэширования винчестеров. При этом часто используемые данные операционная система хранит на относительно небольшом твердотельном устройстве, а данные общего назначения записывает на дисковый накопитель большого объёма. Также возможно объединение флеш-буфера на 4—8 ГБ и магнитного диска в едином устройстве, гибридном жёстком диске (SSHD, Solid-state hybrid drive).

Благодаря большой скорости, объёму и компактным размерам, NAND-память активно вытесняет из обращения носители других типов. Сначала исчезли дискеты и дисководы гибких магнитных дисков, снизилась популярность накопителей на магнитной ленте. Магнитные носители практически полностью вытеснены из мобильных и медиаприменений.

## ****1.5 Преимущества и недостатки Flash-памяти****

**Преимущества.**

**Flash-память обладает рядом значительных преимуществ, которые делают её востребованной в различных областях. Flash-память компактна, обладает высокой скоростью чтения и записи, обеспечивает быструю обработку данных. Отсутствие движущихся частей делает её надёжной, бесшумной и устойчивой к ударам, вибрациям и перепадам температур.**

**Недостатки.**

Основные недостатки Flash-памяти включают ограниченное число циклов записи и стирания, что со временем приводит к износу ячеек памяти. Она имеет относительно высокую стоимость за гигабайт по сравнению с традиционными жесткими дисками. Flash-память также чувствительна к электромагнитным воздействиям, что может привести к потере данных. Кроме того, восстановление информации в большинстве случаев затруднено или вообще невозможно.

# **2 Практическая часть**

**Неисправности Flash-памяти и способы их устранения**

Несмотря на надежность и широкое применение, Flash-память подвержена ряду неисправностей, которые могут повлиять на её работоспособность и срок службы.

Износ ячеек памяти. Каждая ячейка Flash-памяти имеет ограниченный ресурс циклов записи/стирания. С каждым циклом стирания и записи способность ячейки сохранять данные ослабевает, и со временем ячейки начинают выходить из строя. Особенно это выражено в многослойных ячейках (MLC, TLC, QLC), которые имеют меньший ресурс по сравнению с SLC. Замедлить износ памяти может технология Wear leveling. Это технология, которая позволяет равномерно распределять операции записи по всем ячейкам, предотвращая избыточный износ отдельных ячеек. Wear leveling помогает продлить срок службы памяти и избежать преждевременных сбоев.

Ошибки контроллера. Контроллер Flash-памяти управляет процессами записи, стирания, выравнивания износа и исправления ошибок. Ошибка в контроллере может привести к сбоям в работе устройства, включая потерю данных, зависания или неправильную работу системы. Причины неисправности контроллера могут включать программные ошибки, перегрев или физическое повреждение. Устранить эту неисправность могут:

Обновление прошивки. Производители периодически выпускают обновления прошивки для контроллеров, которые исправляют ошибки в их работе, повышают производительность и устраняют уязвимости. Регулярное обновление прошивки помогает избежать многих проблем с контроллером

Диагностика и замена контроллера: В случае аппаратных проблем с контроллером может потребоваться его замена. Многие современные устройства предлагают диагностику ошибок, что позволяет вовремя выявить неисправности и заменить компоненты.

Ошибки чтения и записи (битовые ошибки). Ошибки при чтении или записи данных могут происходить по причине повреждения ячеек памяти или внешних воздействий, таких как электромагнитные помехи. Битовые ошибки могут привести к повреждению данных, что делает их непригодными для дальнейшего использования. Избежать ошибок может технологии ECC (Error Correction Code). Технологии ECC помогают выявлять и исправлять ошибки, возникающие при чтении или записи данных. Это значительно повышает надёжность Flash-памяти, особенно в высоконадежных системах. Так же не помешает регулярное создание резервных копий, особенно для критически важной информации.

Проблемы с интерфейсами (USB или SATA). Ошибки могут возникать на уровне интерфейсов, через которые Flash-память подключается к системе. Например, нестабильное подключение через USB или SATA может привести к потере данных или сбоям в работе устройства. Помочь устранить ошибку может регулярная диагностика интерфейса, замена ненадежный или поврежденных частей, а также использование устройств с качественными разъемами.

Повреждение от электростатических зарядов (ESD). Flash-память и другие компоненты могут быть повреждены из-за воздействия электростатических разрядов (ESD). Это может привести к кратковременным сбоям в работе устройства или его полной неработоспособности. Для недопущения повреждения от ESD важно избегать контакта с материалами, которые могут накапливать заряд, а также использовать заземлённые рабочие места для сборки и обслуживания устройств.

Потеря логической структуры диска. Основная причина таких неисправностей - преждевременное извлечение устройства из разъема USB, незапланированное отключение питания и некорректная работа пользователя (работа без антивирусных программ, запуск программ форматирования и управления дисками). При таких неисправностях разрушается только таблица с информацией о размещении файлов (FAT), а сами файлы, в большинстве случаев, остаются не повреждёнными. При данной неисправности диск в системе опознается как пустой или неформатированный, хранящиеся данные на нем не видны, но при этом он остается физически исправным. Данные с такого поврежденного диска можно извлечь вручную или в автоматическом режиме при помощи специальных утилит восстановления

Рассмотренные неисправности и методы их устранения показывают нам, что несмотря на все преимущества Flash-памяти, она может сталкиваться с определёнными проблемами. Чтобы избежать или минимизировать вероятность этих проблем и продлить срок службы устройств, необходимо следить за их состоянием, обновлять прошивки, делать резервные копии данных и использовать проверенные компоненты. Применение этих простых мер поможет минимизировать риски и сохранить информацию в безопасности.

# **3 Реализация полученных знаний в ассемблере, фигура – усеченная пирамида 7 граней**

Ассемблер — это уникальный язык программирования, который обеспечивает максимально близкий контакт программиста с архитектурой компьютера. Как низкоуровневый язык, он предоставляет возможность прямого управления процессором с помощью команд, практически идентичных машинным инструкциям. В отличие от высокоуровневых языков, таких как Python, Java или C++, ассемблер ориентирован на максимальную эффективность и оптимизацию выполнения задач, что делает его незаменимым для определённых областей применения.

Термин "ассемблер" может обозначать как сам язык программирования, так и программу, которая преобразует написанный код в машинный, понятный процессору. Каждая процессорная архитектура имеет собственный набор инструкций, поэтому программы на ассемблере не являются универсальными. Например, команды для процессора x86 будут отличаться от команд для ARM, что добавляет сложности при программировании.

Программы на ассемблере состоят из мнемоник, сокращённых обозначений команд, и операндов, указывающих данные или адреса памяти, участвующие в операциях. Регистры представляют собой небольшие области памяти внутри процессора, которые активно используются при выполнении команд. В архитектуре x86 выделяют регистры общего назначения (AX, BX, CX, DX) и специализированные, такие как указатель стека (SP) и указатель инструкций (IP).

Код на ассемблере обычно структурируется в сегменты: сегмент данных для переменных и констант, сегмент кода с инструкциями и сегмент стека для временного хранения данных. Такое разбиение облегчает чтение и управление программами, написанными на этом языке.

Ассемблер предоставляет широкий набор базовых команд, включая арифметические операции (ADD, SUB, MUL, DIV), логические операции (AND, OR, XOR, NOT) и управляющие инструкции (JMP, JE, CALL), позволяющие строить сложные алгоритмы.

К преимуществам языка относятся высокая производительность и точный контроль над аппаратными средствами. Программы на ассемблере могут быть максимально компактными и быстрыми, что особенно важно для встраиваемых систем и микроконтроллеров. Однако разработка на ассемблере требует глубокого понимания архитектуры процессоров и значительных усилий из-за его сложности и зависимости от конкретного оборудования.

Для демонстрации работы с прерываниями в библиотеке DOS реализована программа, которая позволяет создавать графические фигуры. В качестве примера построена усечённая пирамида с семью гранями. Для написания кода использовалась среда разработки GUI Turbo Assembler, которая предоставляет удобные инструменты для редактирования и тестирования программ на языке Ассемблер. Интерфейс этой среды программирования показан на рисунке 2.

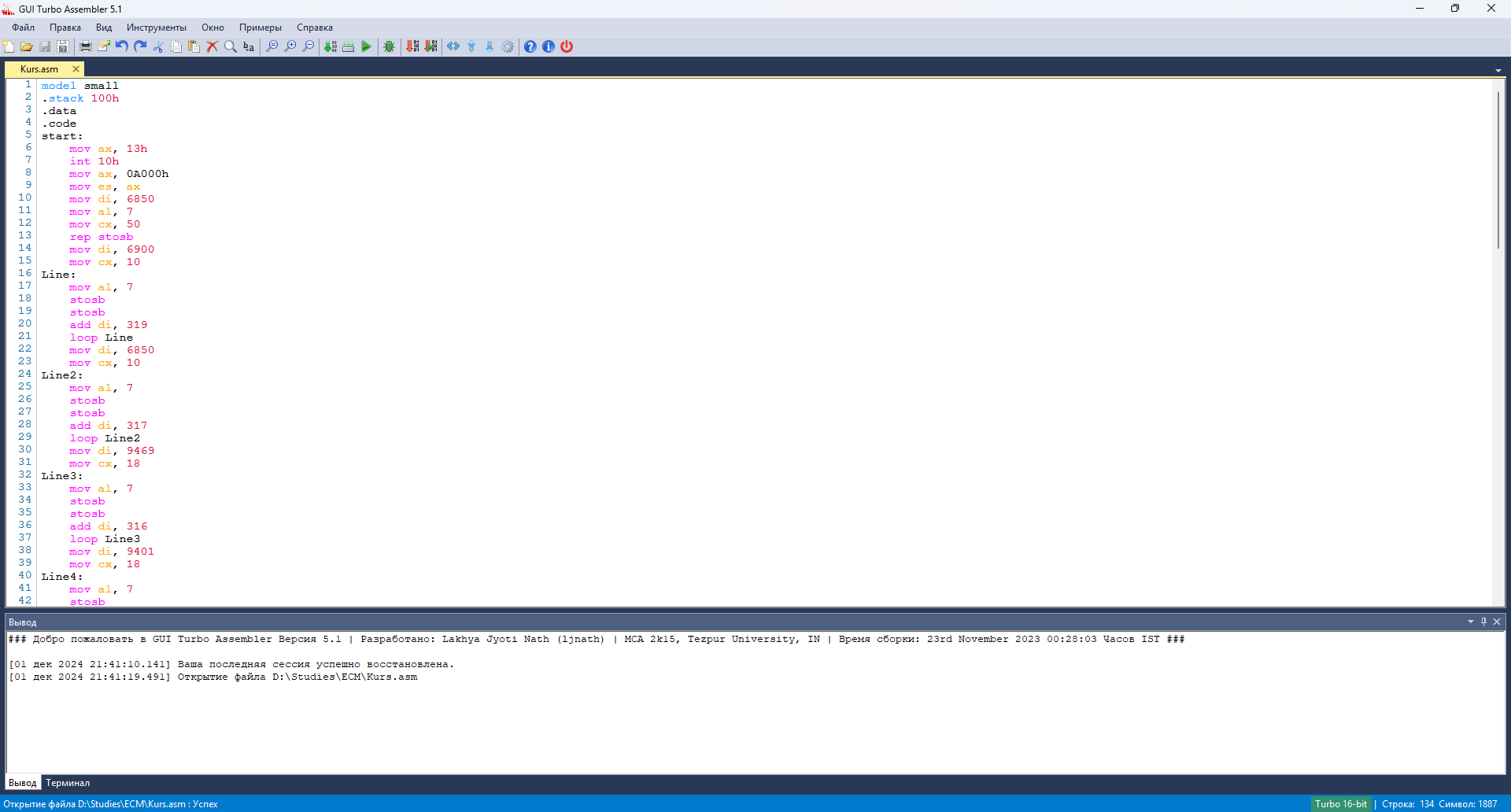


Рисунок 4 - Интерфейс среды программирования GUI Turbo Assembler

Используя полученные ранее знания по программированию на языке Ассемблер, разработана программа, способная нарисовать усечённую пирамиду с семью гранями. Результат выполнения программы представлен на рисунке 3.

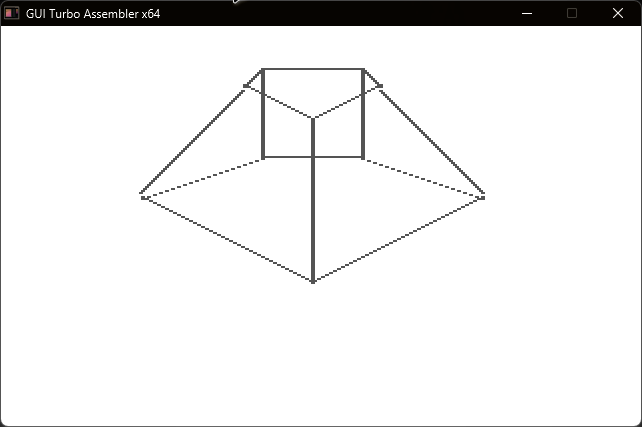


Рисунок 5 - Результат выполнения программы: усечённая пирамида (7 граней)

# **ЗАКЛЮЧЕНИЕ**

В данной курсовой работе проведён анализ Flash-памяти, включающий изучение её принципа действия, основных характеристик и областей применения. Были рассмотрены ключевые преимущества технологии, такие как энергонезависимость, высокая скорость работы, компактность и устойчивость к физическим воздействиям. Также выявлены основные недостатки Flash-памяти, включая ограниченный ресурс циклов записи/стирания, снижение производительности со временем и относительную высокую стоимость.

Исследование принципа работы Flash-памяти позволило глубже понять, как используется механизм плавающего затвора для хранения данных. Также были проанализированы современные типы памяти, такие как SLC, MLC, TLC и QLC, их особенности и области применения.

Практическая часть работы включала использование знаний в программировании для создания программы, которая демонстрирует возможности работы с графическим режимом. Программа разработана на языке Ассемблер и иллюстрирует использование прерываний DOS для управления графическим выводом.

Таким образом, в рамках данной работы удалось объединить теоретическое исследование технологии Flash-памяти с практическим применением знаний в области низкоуровневого программирования. Это позволило наглядно продемонстрировать широкие возможности использования Flash-памяти и укрепить навыки работы с аппаратурой и программным обеспечением.

Задачи курсовой работы выполнены в полном объёме. Цель курсовой работы достигнута.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Работа с TASM. [Электронный ресурс]. URL: <https://citforum.ru/programming/tasm3/tasm3_1.shtml> (дата обращения: 1.12.2024).
2. Начало работы с Turbo Assembler. [Электронный ресурс]. URL: <https://www.i-assembler.ru/3/index-glava-1-nachalo-raboty-na-turbo-assemblere.htm> (дата обращения: 1.12.2024).
3. Ассемблер для чайников. [Электронный ресурс]. URL: <https://av-assembler.ru/asm/afd/assembler-for-dummy.htm> (дата обращения: 30.11.2024).
4. Flash memory [Электронный ресурс]. URL: https://www.techtarget.com/searchstorage/definition/flash-memory (дата обращения: 1.12.2024).
5. Фоксфорд [Электронный ресурс]. URL: https://foxford.ru/wiki/informatika/flesh-pamyat?srsltid=AfmBOooTXEARwqLkwlWxN6K5LCbNPLCXBhInbQ35W55uts6H3PpWqf8x&utm\_referrer=https%3A%2F%2Fwww.google.com%2F (дата обращения: 29.11.2024).
6. Прерывания в DOS. Теория и практика. [Электронный ресурс]. URL: https://www.programminginassembly.com/dos-interrupts (дата обращения: 12.12.2024).
7. Преимущества и недостатки флэш-памяти. [Электронный ресурс]. URL: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-flash-memory/ (дата обращения: 14.12.2024).
8. Как устроена и работает флеш-память? [Электронный ресурс]. URL: https://review.1k.by/pc/Kak\_ystroena\_i\_rabotaet\_flesh\_pamyat-1327.html (дата обращения 14.12.2024).
9. Флеш-память. [Электронный ресурс]. URL: https://ru.ruwiki.ru/wiki/Флеш-память (дата обращения 14.12.2024).
10. QLC, TLC, MLC и SLC [Электронный ресурс]. URL: https://www.nix.ru/computer\_hardware\_news/hardware\_news\_viewer.html?id=177974 (дата обращения 14.12.2024).
11. Технологии флэш-памяти. [Электронный ресурс]. URL: https://www.ixbt.com/storage/flash-tech.shtml (дата обращения 14.12.2024).
12. Неисправности флешек (флэш-дисков) и карт памяти. [Электронный ресурс]. URL: https://comcenter.ru/Articles/Details/31/neispravnosti-fleshek-flesh-diskov-i-kart-pamyati.-chast-1 (дата обращения 14.12.2024).

# **Приложение А «Код программы построения фигуры усеченная пирамида 7 граней»**

model small

.stack 100h

.data

.code

start:

mov ax, 13h

int 10h

mov ax, 0A000h

mov es, ax

mov di, 6850

mov al, 7

mov cx, 50

rep stosb

mov di, 6900

mov cx, 10

Line:

mov al, 7

stosb

stosb

add di, 319

loop Line

mov di, 6850

mov cx, 10

Line2:

mov al, 7

stosb

stosb

add di, 317

loop Line2

mov di, 9469

mov cx, 18

Line3:

mov al, 7

stosb

stosb

add di, 316

loop Line3

mov di, 9401

mov cx, 18

Line4:

mov al, 7

stosb

stosb

add di, 320

loop Line4

mov di, 20930

mov al, 7

mov cx, 50

rep stosb

mov di, 6900

mov cx, 45

Line5:

mov al, 7

stosb

stosb

add di, 318

loop Line5

mov di, 6850

mov cx, 45

Line6:

mov al, 7

stosb

stosb

add di, 316

add di, 2

loop Line6

mov di, 27270

mov cx, 43

Line7:

mov al, 7

stosb

stosb

add di, 322

sub di, 2

loop Line7

mov di, 27440

mov cx, 43

Line8:

mov al, 7

stosb

stosb

add di, 316

loop Line8

mov di, 15195

mov cx, 82

Line9:

mov al, 7

stosb

stosb

add di, 322

sub di, 4

loop Line9

mov di, 10360

mov cx, 52

Line10:

mov al, 7

stosb

stosb

add di, 317

sub di, 0

loop Line10

mov di, 10429

mov cx, 52

Line11:

mov al, 7

stosb

stosb

add di, 319

sub di, 0

loop Line11

mov di, 21250

mov cx, 21

Line12:

mov al, 7

stosb

stosb

add di, 315

loop Line12

mov di, 21300

mov cx, 21

Line13:

mov al, 7

stosb

stosb

add di, 321

loop Line13

xor ax, ax

int 16h

mov ax, 3

int 10h

mov ax, 4C00h

int 21h

end start