Одеський національний університет імені І.І.Мечникова

Кафедра математичного забезпечення комп’ютерних систем

# КУРСОВА РОБОТА

з дисципліни «Системне програмування»

на тему:«Створення інтерпретатора

обчислювальної системи»

Студента (ки) \_\_2\_\_ курсу \_\_\_\_2\_\_\_\_ групи

напряму підготовки \_\_\_\_«Компьютерна інженерія»\_\_\_\_\_

спеціальності \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_ Оцінка: ECTS \_\_\_

м. Одеса – 2015 рік

ОГЛАВЛЕНИЕ

[КУРСОВА РОБОТА 1](#_Toc420880829)

[ПОСТАНОВКА ЗАДАЧИ 2](#_Toc420880830)

[1 СТРУКТУРА ЭВМ 3](#_Toc420880831)

[1.1 Процессор. Такт работы процессора 4](#_Toc420880832)

[1.2 Оперативная память 5](#_Toc420880833)

[1.3 Внешние устройства 6](#_Toc420880834)

[1.4 Система команд 7](#_Toc420880835)

[1.5 Схема работы процессора 8](#_Toc420880836)

[2 УЧЕБНАЯ ТРЕХАДРЕСНАЯ МАШИНА УМ-3 9](#_Toc420880837)

[2.1 Процессор УМ-3 9](#_Toc420880838)

[2.2 Оперативная память 10](#_Toc420880839)

[2.3 Система команд УМ-3 11](#_Toc420880840)

[3 УЧЕБНАЯ ДВУХАДРЕСНАЯ МАШИНА УМ-2 14](#_Toc420880841)

[3.1 Описание УМ-2 14](#_Toc420880842)

[4 УЧЕБНАЯ МАШИНА С РЕГИСТРАМИ И МОДИФИКАЦИЕЙ АДРЕСОВ УМ-РМ 16](#_Toc420880843)

[4.1 Устройство процессора машины УМ-РМ 16](#_Toc420880844)

[4.2 Описание УМ-РМ 18](#_Toc420880845)

[4.3 Система Команд УМ-РМ 19](#_Toc420880846)

[5 АЛГОРИТМ РАБОТЫ ПРОГРАММЫ 21](#_Toc420880847)

[6 ТЕСТИРОВАНИЕ ПРОГРАММЫ 23](#_Toc420880848)

[ЗАКЛЮЧЕНИЕ 25](#_Toc420880849)

ПРИЛОЖЕНИЕ………………………………………………………………………....26

# ПОСТАНОВКА ЗАДАЧИ

Требуется разработать программную систему для процесса выполнения команд учебной машины УМ-3, УМ-2 и УМ-РМ.

При построении модели функционирования программной системы должны учитываться следующие основные моменты выполнения команд:

- пошаговое выполнение команд с возможностью приостановки на каждом шаге;

- отображение внутреннего состояния УМ;

- выполнение всего списка команда (всей программы) и получение результата.

# 1 СТРУКТУРА ЭВМ

Основными элементами любой вычислительной машины являются центральный процессор, оперативная память и внешние устройства. Схематично компьютер изображён на рисунке 1.



Рис.1. Схема ЭВМ

Центральный процессор (ЦП) выполняет программу пользователя и управляет всеми устройствами ЭВМ. Оперативная память (ОП) во время работы компьютера хранит данные и выполняемую программу. Внешние устройства служат для связи вычислительной машины с внешним миром: для ввода и вывода данных, для долговременного хранения данных и программ.

## 1.1 Процессор. Такт работы процессора

Центральный процессор – основной компонент вычислительной машины. Именно процессор выполняет программу пользователя. Кроме этого, процессор организует взаимодействие между всеми частями вычислительной машины, так что они образуют единое целое – компьютер.

Элементами центрального процессора являются устройства и регистры. Регистры – это запоминающие элементы (ячейки), расположенные внутри процессора; они предназначены для хранения информации. Схема центрального процессора представлена на рис. 2

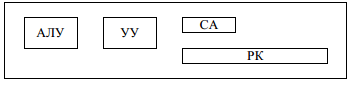


Рис. 2. Схема центрального процессора

Устройство управления управляет работой процессора в целом, координирует работу всех остальных устройств, организует процесс выполнения программы. Арифметико-логическое устройство выполняет арифметические и логическое операции, собственно, производит вычисления. Регистр счётчик адреса содержит адрес очередной команды программы. Регистр команды содержит команду, которая выполняется процессором в текущий момент времени.

## 1.2 Оперативная память

Оперативная память хранит программу, которую выполняет вы-числительная машина, и данные для программы. Оперативная память может хранить данные только во время работы машины. Когда машина выключается, содержимое оперативной памяти теряется.

Оперативная память состоит из элементов – ячеек. Количество ячеек называется объёмом оперативной памяти. Ячейки перенумерованы, номер ячейки называется адресом ячейки. Схематически оперативная память изображена на рисунке 3.

|  |  |
| --- | --- |
|  | О.П. |
| 0 |  | |
| 1 |  | |
| . |  | |
| . |  | |
| . |  | |
| N-1 |  | |

Рис. 3. Оперативная память

Объем ОП и размер регистра СА взаимосвязаны: количество разрядов в СА должно быть достаточно для хранения любого возможного адреса.

Ячейки состоят из разрядов. Количество разрядов во всех ячейках одинаково и называется разрядностью машины. Каждый разряд содержит одну двоичную цифру.

Ячейки могут хранить данные и команды. Команды, основные операции, выполняемые с оперативной памятью – записать в ячейку число или команду (из процессора или с внешнего устройства) и переслать содержимое ячейки из памяти в процессор (или на внешнее устройство). Доступ к ячейке осуществляется по её адресу, номеру. Работать с частью ячейки нельзя, поэтому часто понятие ячейка определяют так: ячейка – это минимальная адресуемая единица памяти.

В любой момент времени можно обратиться к любой ячейке, независимо от того, с какой ячейкой работали раньше. Время доступа к ячейке не зависит от расположения ячейки в памяти. Оба эти обстоятельства имеют в виду, говоря, что оперативная память – устройство прямого (или произвольного) доступа.

Оперативная память намного более медленное устройство, чем центральный процессор.

## 1.3 Внешние устройства

Внешние устройства предназначены для связи вычислительной машины с внешним миром (записи программы в оперативную память, ввода данных и вывода результата счёта) и для долговременного хранения программ и данных. В соответствии со своим назначением внешние устройства делятся на два класса: устройства ввода-вывода и внешние запоминающие устройства.

К устройствам ввода-вывода относятся клавиатура, монитор, мышь, принтер, сканер и т.п. Устройства ввода-вывода работают намного медленнее процессора и оперативной памяти. Примерами внешних запоминающих устройств являются жесткий диск (винчестер), флэш-накопитель, магнитофон (стример). По сравнению с оперативной памятью внешняя память имеет намного больший объём, работает существенно медленнее. Время доступа к данным может зависеть от их расположения на носителе.

Важным фактором является то, что существует большое разнообразие внешних устройств, каждое из них обладает своими физическими характеристиками и особенностями функционирования, которые требуется учитывать при организации взаимодействия вычислительной машины и внешних устройств.

## 1.4 Система команд

Каждый процессор имеет набор встроенных, так называемых машинных, операций. Машинная операция – это элементарное действие, выполняемое аппаратным путём. Машинные операции реализованы в виде электронных схем в процессоре. Примеры машинных операций: пересылка содержимого ячейки из оперативной памяти в процессор, сложение, вычитание. Любое вычисление, которое нужно произвести на вычислительной машине, необходимо разложить на элементарные действия, представить в виде последовательности машинных операций.

Конкретный набор машинных операций процессора зависит от специфики класса задач, для решения которых предполагается использовать вычислительную машину, и от экономических факторов. Действия, часто встречающиеся при решении задач, целесообразно реализовывать аппаратно, редкие операции можно реализовать программно, выразив их через машинные операции.

Чем шире набор машинных операций, тем, вообще говоря, проще программировать для процессора, тем процессор сложнее и дороже. Чем уже набор машинных операций, тем проще процессор.

Приказ выполнить машинную операцию для указанных данных называется машинной командой. Правила записи машинной команды называется форматом команды. Список машинных операций вместе с форматами команд называется системой команд процессора. Машинная программа представляет собой алгоритм решения задачи, реализованный в виде последовательности машинных команд.

Стоит заметить, что часто слово "команда" используют как синоним словосочетания "машинная операция". Однако эта многозначность слова "команда" путаницы не вызывает, так как из контекста всегда понятно, о чём идёт речь – об аппаратно реализованной операции ("в процессоре имеется команда сложения") или о выполнении действия с конкретными данными.

Как правило, в набор машинных операций входят следующие группы команд.

– пересылки: предназначены для обмена данными между процессором и оперативной памятью;

– арифметические команды: сложение, вычитание, умножение, деление и другие операции;

– переходы: команды передачи управления, позволяющие програм-мировать условные операторы и циклы.

## 1.5 Схема работы процессора

Условимся, что в начале работы вычислительной машины в её оперативную память каким-то образом загружена программа. Исходные данные находятся в ячейках памяти с известными адресами. Результаты также нужно записать в известные ячейки оперативной памяти.

Работа вычислительной машины состоит из шагов – тактов. Тактом работы процессора называется выполнение одной машинной команды.

Работа происходит следующим образом.

При включении компьютера прежде всего в регистр счётчик адреса (СА) записывается некоторое число. Это действие выполняется аппаратно, независимо от программиста, причём при каждом запуске процессора в СА записывается одно и то же число. Будем считать, что это число 0.

После загрузки начального значения в регистр СА, процессор начинает последовательно, шаг за шагом, выполнять команды программы, пока не встретит команду останова.

На каждом такте процессор выполняет действия:

1. В регистр команды (РК) записывается содержимое ячейки, адрес которой находится в регистре СА.

2. Значение СА увеличивается на 1, так что теперь СА указывает на следующую команду программы.

3. Устройство управления расшифровывает команду, находящуюся в РК, и организует её выполнение. Как именно выполняется команда, зависит от вида самой команды.

Далее работа повторяется с первого шага.

Процесс заканчивается, когда выполнилась команда остановки процессора.

Заметим, что, поскольку при включении машины в СА записывается число 0 и далее именно содержимое ячейки с адресом 0 помещается в ЦП, расшифровывается и выполняется, в нулевой ячейке должна находиться первая исполняемая команда программы. Итак, либо все программы должны начинаться по нулевому адресу, то есть с ячейки с адресом 0, либо в ячейку с адресом 0 должна быть помещена команда перехода на начало программы..

Итог: регистр счётчик адреса хранит адрес команды, которая будет выполняться на следующем шаге работы процессора; регистр команды содержит команду, выполняемую на текущем шаге.

При записи машинных программ для сокращения текста и повышения наглядности вместо выписывания последовательности двоичных цифр (содержимого ячейки), записывать шестнадцатиричные цифры. Каждая шестнадцатиричная цифра изображает четыре двоичные цифры.

# 2 УЧЕБНАЯ ТРЕХАДРЕСНАЯ МАШИНА УМ-3

## 

## 2.1 Процессор УМ-3

Рассмотрим подробнее часть процессора, отвечающую за выполнение арифметических операций, представленную на рисунке 4.

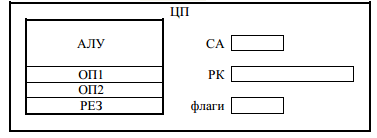


Рис. 4. Схема процессора УМ-3

С арифметико-логическим устройством связаны регистры первого и второго операндов (ОП1 и ОП2) и регистр результата (РЕЗ). Регистр результата также называют сумматором. При выполнении арифметической операции в регистры ОП1 и ОП2 записываются первый и второй операнды соответственно, а в регистр РЕЗ арифметико-логическое устройство поме-щает полученный результат.

## 2.2 Оперативная память

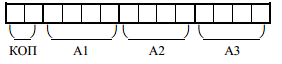
Оперативная память машины УМ-3 состоит из 65536 (216, или 164) ячеек, которые имеют адреса от 000016 до FFFF16. Каждая ячейка состоит из 14 шестнадцатеричных разрядов.

Ячейка может содержать число или команду.

В машине УМ-3 используются числа без знака и числа со знаком.

Числа со знаком представляются в дополнительном коде.

Если ячейка содержит команду, разряды ячейки группируются следующим образом:



Здесь КОП – код операции (указывает, какую машинную операцию надо выполнить); А1, А2, А3 – адреса первого, второго операндов и результата. Разрядность адресов совпадает с разрядностью регистра СА и определяется объемом ОП (а именно, количеством возможных адресов). Количество разрядов в регистрах ОП1, ОП2, РЕЗ и РК должно совпадать с количеством разрядов ячейки ОП, так как в ОП1, ОП2 и РЕЗ должны поместиться числа, а в РК – команда, которые занимают по одной ячейке ОП.

## 2.3 Система команд УМ-3

В таблице 1 приведены все команды учебной машины УМ-3.

Таблица1. Команды учебной машины УМ-3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | КОП | Операция | | Примечание |
| останов | 99 | стоп | | А1,А2,А3 - любые |
| пересылка | 00 | [А1] → [А3] | | А2 - любой |
|  |  | арифметические | |  |
| сложение | 01 | [А1]+[А2] → [А3] | | Устанавливаются |
| вычитание | 02 | [А1]-[А2] → [А3] | | арифметические флаги |
| умножение |  |  | |  |
| со знаком | 03 | [А1]\*[А2] → [А3] | |  |
| без знака | 13 |  | |  |
| деление |  |  | |  |
| со знаком | 04 | [А1] div [А2] → [А3], [А1] mod [А2] → [А3+1] | | |
| без знака | 14 |  | |  |
|  |  | переходы | |  |
| безусловный | 80 | перейти к А3 | | А1,А2 - любые |
| условные |  |  | | После выполнения |
| по = | 81 | при [А1]=[А2]  перейти к А3 | | команды перехода работа продолжается |
| по ≠ | 82 | при [А1]≠[А2]  перейти к А3 | | с команды, записанной в ячейке с адресом A3 |
| по < | с/зн - 83 б/зн - 93 | | при [А1]<[А2] перейти к А3 | |
| по ≥ | с/зн - 84 б/зн - 94 | | при [А1]≥ [А2] перейти к А3 | |
| по > | с/зн - 85 б/зн - 95 | | при [А1]>[А2] перейти к А3 | |
| по ≤ | с/зн - 86 б/зн - 96 | | при [А1]≤ [А2] перейти к А3 | |
|  |  | Ввод-вывод | |  |
| ввод  со знаком  без знака | 06  16 | ввод A2 чисел в память, начиная с адреса A1 | | А3 - любой |
| вывод  со знаком  без знака | 07  17 | вывод A2 чисел в память, начиная с адреса A1 | | А3 - любой |

В группу Арифметических команд входят команды сложения, вычитания, умножения и деления. Арифметические команды выполняются следующим образом: содержимое ячеек с адресами A1 и A2 пересылается в регистры ОП1 и ОП2, выполняется арифметическая операция (результат – в регистре РЕЗ), содержимое регистра РЕЗ пересылается в ячейку с адресом А3.

Сложение чисел без знака и сложение чисел со знаком выполняются по единому алгоритму, так как числа со знаком представляются в дополнительном коде. Поэтому в УМ-3 существует одна машинная команда сложения, работающая как с числами со знаком, так и с числами без знака.

В результате выполнения операции вычисляется сумма и формируются флаги. По значениям флагов можно отследить особые ситуации, например, переполнение при сложении чисел без знака (по флагу CF) и переполнение при работе с числами со знаком (по значению флага OF). Вычитание аналогично сложению: один и тот же алгоритм подходит для обработки чисел без знака и для обработки чисел со знаком, следовательно, требуется одна машинная команда вычитания.

С умножением ситуация сложнее. Числа без знака и числа со знаком обрабатываются по разным алгоритмам, поэтому существует две машинные команды умножения – умножение чисел без знака и умножение чисел со знаком. То же с делением. Есть машинная команда деления чисел без знака и машинная команда деления чисел со знаком.

Деление имеет ещё одну особенность: одна команда деления вырабатывает два результата, частное и остаток. Они записываются в соседние ячейки: частное – по адресу А3, остаток – в ячейку с адресом А3+1.

Действие команд перехода заключается в изменении содержимого регистра счётчик адреса (СА). Таким образом, на следующем такте работы процессора будет выполнена не та команда, которая бы выполнилась при нормальном порядке выполнении команд, а другая, чей адрес теперь записан в СА.

В машине УМ-3 есть безусловный и условные переходы.

Код операции безусловного перехода 80, формат команды 80 А1 А2 А3, здесь А3 – адрес перехода, т.е. адрес команды, на которую нужно передать управление; операнды А1 и А2 не используются. Действие команды: запись в регистр СА адреса А3.

Все команды условного перехода выполняются так: проверяется, удовлетворяют ли числа, находящиеся в ячейках памяти по адресам А1 и А2, условию перехода. Если условие оказывается истинным, в регистр СА записывается число А3, в противном случае содержимое СА не меняется.

Сравнение значений [А1] и [А2] выполняется так: в ОП1 записывается [А1], в ОП2 – [А2]; выполняется вычитание ОП1-ОП2, при этом получают значение все четыре арифметических флажка. По значениям флагов и определяется истинность условия. Некоторым условиям (например, <) соответствуют разные наборы значений флагов для разных данных – чисел со знаком и чисел без знака. Поэтому команды переходов для чисел со знаком и чисел без знака разные.

Рассмотрим на примерах соответствие между условиями и значениями флагов.

1) Переход по равенству. В результате вычитания [А1] - [А2] получится ZF=1 тогда и только тогда, когда [А1] = [А2].

2) Переход по "меньше", числа без знака.

[А1] < [А2] ⇔ [А1] - [А2] < 0 ⇔ CF=1.

3) Переход по "меньше", числа со знаком. Если вычитаются числа со знаком, то при [А1] < [А2], [А1] - [А2] < 0, возможны ситуации:

а) результат вычислили правильно (OF=0) и он оказался отрицательным , значение SF=1,

б) результат вычислили неправильно (OF=1), при этом знак вычисленного значения противоречит знаку числа [А1] - [А2], вычис­лен­ный результат оказался положительным, то есть SF=0.

Таким образом, для чисел со знаком   
[А1] < [А2] ⇔ [А1] - [А2] < 0 ⇔ (OF=1) *and* (SF=0) *or* (OF=0) *and* (SF=1), или более коротко OF ≠ SF. При выполнении команды перехода по "меньше" для чисел со знаком процессор проверяет выполнимость условия OF ≠ SF.

Выбор, какой вариант команды использовать в программе, зависит от того, с какими данными работает программа**.**

Формат команды пересылки: 00 А1 А2 А3. Действие команды заключается в пересылке содержимого ячейки памяти с адресом А1 в ячейку с адресом А3. Адрес А2 несущественен, однако должен быть указан в команде.

Формат команды остановки 99 А1 А2 А3. По этой команде процессор завершает выполнение программы, останавливается.

В таблице использовано обозначение [A] – содержимое ячейки с адресом A.

# 3 УЧЕБНАЯ ДВУХАДРЕСНАЯ МАШИНА УМ-2

## 

## 3.1 Описание УМ-2

Анализ большого числа программ показывает, что только 25% команд содержат три различных адреса. В остальных командах либо два, либо все три адреса совпадают. Этот факт наталкивает на идею отказаться от одного из адресов — сделать команды двухадресными. Тогда можно использовать освободившиеся разряды для задания более длинных адресов — появляется возможность адресовать больший объем памяти. Либо можно сократить размер ячейки, сделав более дешевой оперативную память.

Результат арифметических команд записывается на место первого операнда (по адресу А1). Таким образом, формат арифметических команд принимает вид КОП А1 А2. Действие команды заключается в следующем: поместить в регистр первого операнда содержимое ячейки с адресом А1, поместить в регистр второго операнда содержимое ячейки памяти с адресом А2, выполнить операцию, записать результат в ячейку с адресом А1.

Чтобы использовать команды условного перехода с двумя адресами, разделим действие команды на два этапа – сравнение операндов адресами, разделим действие команды на два этапа – сравнение операндов и сам переход по условию.

Команда сравнения устанавливает значения арифметических флагов CF, OF, SF и ZF. Команда условного перехода по значениям флагов анализирует, истинно ли анализирует, истинно ли условие перехода; если условие истинно, истинно ли условие перехода; если условие истинно, записывает адрес А2 в регистр СА, если условие ложно, содержимое СА не изменяется. Команда перехода не меняет значения флагов. Соответствия условий перехода значениям флагов сведены в таблицу 2.

Таблица 2. Соответствия условий перехода значениям флагов

|  |  |  |  |
| --- | --- | --- | --- |
| Условие перехода | Проверяемое условие | Условие перехода | Проверяемое условие |
| = | ZF=1 ≠ | ZF=0 |  |
| числа без знака | | | |
| < | CF=1 | ≤ (CF=1) | or (ZF=1) |
| ≥ | CF=0 | > | (CF=0) and (ZF=0) |
| числа со знаком | | | |
| < | OF≠SF | ≤ | (OF≠SF) or (ZF=1) |
| ≥ | OF=SF | > | (OF=SF) and (ZF=0) |

Структура процессора та же, что и в УМ-3; разрядность регистров соответствует разрядности ячейки и объему ОП.

Ячейка ОП — 10 шестнадцатиричных разрядов.

Объем ОП — 164 ячеек.

Число занимает одну ячейку. Отрицательные числа записываются в дополнительном коде.

Команда занимает одну ячейку:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |

⎝ ⎠ ⎝ ⎠ ⎝ ⎠

КОП А1 А2

Коды операций те же, что в УМ-3. Отличия приведены в таблице 3.

Таблица 3. Отличия системы команд УМ-2 от УМ-3

|  |  |
| --- | --- |
| Команды | Действие |
| Пересылка 00 А1 А2 | [А1] := [А2] |
| Арифметические команды | [А1] := [А1] ⊗ [А2], где ⊗ ∈ {+, -, \*} |
| Деление | вычисляет частное и остаток от деления [А1] на [А2]; частное записывается в [А1], остаток в [А1+1] |
| Сравнение 05 А1 А2 | вычисляется [А1] - [А2], устанавливаются арифметические флаги; результат вычитания не сохраняется |
| Условные переходы   КОП А1 А2 | переход по А2, если условие выполняется; А1 не используется |
| Ввод-вывод  КОП А1 А2 | Вводили вывод A2 чисел в память, начиная с адреса A1 |

# 4 УЧЕБНАЯ МАШИНА С РЕГИСТРАМИ И МОДИФИКАЦИЕЙ АДРЕСОВ УМ-РМ

## 

## 4.1 Устройство процессора машины УМ-РМ

Схема процессора УМ-РМ приведена на рисунке. На рисунке 4 изображены только элементы, важные нам для понимания работы процессора.

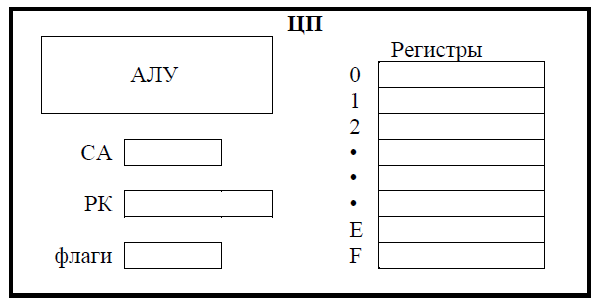


Рис. 4. Основные элементы ЦП УМ-РМ

Основные элементы (устройство управления, АЛУ, регистры СА и РК, регистр флагов) те же, что и в процессорах машин УМ-3 и УМ-2. Помимо этих устройств, в процессоре находятся так называемые регистры. Регистры подобны ячейкам памяти, в них можно хранить числа. Можно сказать, что регистр – это ячейка, расположенная в процессоре, а не в оперативной памяти. Программист может использовать регистры в качестве операндов машинных команд. Если регистр является операндом машинной команды, в команде указывается номер этого регистра. Если оба операнда команды являются регистрами, длина команды составляет четыре разряда, т.к. два разряда занимает код операции и по одному разряду нужно для указания номеров регистров. Следовательно, ячейки ОП должны состоять из четырёх разрядов, чтобы было удобно хранить команды.

В программировании большое количество задач связано с обработкой массивов – последовательностей элементов, расположенных в оперативной памяти. Поэтому желательно сделать работу с массивами удобной. Решается эта проблема с помощью механизма модификации адресов. Суть этого механизма состоит в следующем. В команде кроме адреса операнда указывается регистр-модификатор. При выполнении команды процессор вычисляет сначала исполнительный адрес как сумму адреса, указанного в команде, и содержимого регистра-модификатора. Затем из ячейки с полученным адресом берется операнд. Изменив содержимое модификатора, мы заставляем процессор использовать другой адрес операнда. Действие по вычислению исполнительного адреса выполняется аппаратно, что дает преимущество в скорости выполнения программ.

Быстродействие машины УМ-РМ намного выше, чем УМ-2, за счёт сокращения количества обращений к памяти при выполнении арифметических команд. Программы для УМ-РМ получаются короче, чем для УМ-2 из-за того, что наиболее используемые команды регистр-регистр короче команд УМ-2, несмотря на появление в программах лишних команд загрузки регистров и команд переписывания результатов из регистров в память.

## 4.2 Описание УМ-РМ

Кроме обычных регистров, ЦП содержит 16 регистров общего назначения. Регистры нумеруются от 0 до F16. Размер регистров совпадает с разрядностью чисел.

Ячейка ОП состоит из четырех шестнадцатиричных разрядов.

Объем ОП — 164 ячеек. Размер ячейки – четыре 16-ричных разряда. Ячейки могут хранить числа или команды.

В четырёхразрядную ячейку можно записать относительно маленькое число, не больше, чем 164. Но желательно иметь возможность работать и с большими числами, поэтому будем считать, что число занимает две соседние ячейки, т.е. восемь разрядов. Так как размер регистров совпадает с разрядностью чисел, каждый регистр состоит из восьми 16-ричных разрядов.

Отрицательные числа записываются в дополнительном коде.

Команды бывают двух типов.

а) Регистр-регистр. Занимает одну ячейку. Формат команды

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

⎝ ⎠

КОП R1 R2,

где R1 — первый операнд, R2 — второй операнд.

б) Регистр-память. Занимает две ячейки. Формат команды

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

⎝ ⎠ ⎪ ⎪ ⎝ ⎠

КОП R1 M2 А2

где R1 — первый операнд, A2 — адрес второго операнда, M2 — регистр, используемый для модификации адреса A2 - это может быть любой регистр общего назначения, кроме R0

При выполнении команд регистр-память процессор вычисляет исполнительный адрес по правилу



Второй операнд берется из ячейки с адресом Аисп.

Запись [*М2*] обозначает, что в формуле используется не номер регистра *М2*, а содержимое этого регистра. Формула говорит о том, что для получения реального адреса второго операнда нужно сдвинуться от ячейки с адресом *А2* на [*М2*] ячеек.

Исполнительный адрес вычисляется по модулю 164. Это означает, что ячейки оперативной памяти как бы образуют кольцо: за ячейкой с адресом FFFF16 идёт ячейка 000016. Например, если R6=2016, то для команды 01 3 6 FFFF второй операнд берётся из ячейки с адресом 001F16.

Если в качестве второго регистра указан регистр с номером 0, мо-дификации адреса не происходит. Таким образом, регистр R0 не может быть модификатором.

## 4.3 Система Команд УМ-РМ

Все арифметические команды УМ-РМ имеют две формы – короткую и длинную. Семантика арифметических команд традиционна: два операнда помещаются в регистры R1 и R2, связанные с АЛУ, выполняется операция с установлением значений флагов, результат вычислений записывается в первый операнд. Результаты команды деления записываются в два соседних регистра: частное – в первый операнд, остаток – в регистр, следующий за ним (в списке регистров ЦП). Команды переходов имеют только длинную форму, поскольку в команде необходимо указать адрес перехода. Список всех команд УМ-РМ приведен на рисунке 5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Название** | **КОП** | **Регистр-память** | | **КОП** | **Регистр-регистр** |
| останов |  |  | | 99 | стоп |
| пересылки | 00 | R1:=(A2,A2+1) | | 20 | R1:=R2 |
|  | 10 | R1→(A2,A2+1) | |  |  |
|  |  | **Арифметические** | |  |  |
| сложение | 01 | R1:=R1+(А2,A2+1) | | 21 | R1:=R1+R2 |
| вычитание | 02 | R1:=R1-(А2,A2+1) | | 22 | R1:=R1-R2 |
| умножение |  |  | |  |  |
| - со знаком | 03 | R1:=R1\*(А2,A2+1) | | 23 | R1:=R1\*R2 |
| - без знака | 13 |  | | 33 |  |
| деление |  |  | |  |  |
| - со знаком | 04 | R1:=R1 div (А2,A2+1),  (R1+1):=R1 mod (А2,A2+1) | | 24 | R1:=R1 div R2,  (R1+1):=R1 mod R2 |
| - без знака | 14 |  | | 34 |  |
| сравнение | 05 | R1-(А2,A2+1) | | 25 | R1-R2 |
| **Переходы по А2 (типа регистр-память)(R1- любой)** | | | | | |
| безусловный | 80 | |  |  |  |
| по = | 81 | |  |  |  |
| по ≠ | 82 | |  |  |  |
| по < | с/зн – 83 | | б/зн – 93 | |  |
| по ≥ | с/зн – 84 | | б/зн – 94 | |  |
| по > | с/зн – 85 | | б/зн – 95 | |  |
| по ≤ | с/зн – 86 | | б/зн – 96 | |  |

Рис. 5. Список команда УМ-РМ

# 5 АЛГОРИТМ РАБОТЫ ПРОГРАММЫ

Программная система предназначена для выполнения команд определенной вычислительной системы (ВС).

Программа на входе получает список необходимых переменных, их значения либо имена и список команд для выполняемой программы (ВП). Выполнение команд возможно двумя способами:

- выполнение каждой команды отдельно в интерактивном режиме по запросу пользователя;

- выполнение сразу всех команд в автоматическом режиме.

Каждая ВП характеризуется:

- количеством переменных и команд;

- занятым объемом оперативной памяти.

На рисунке 6 представлена диаграмма классов программы.

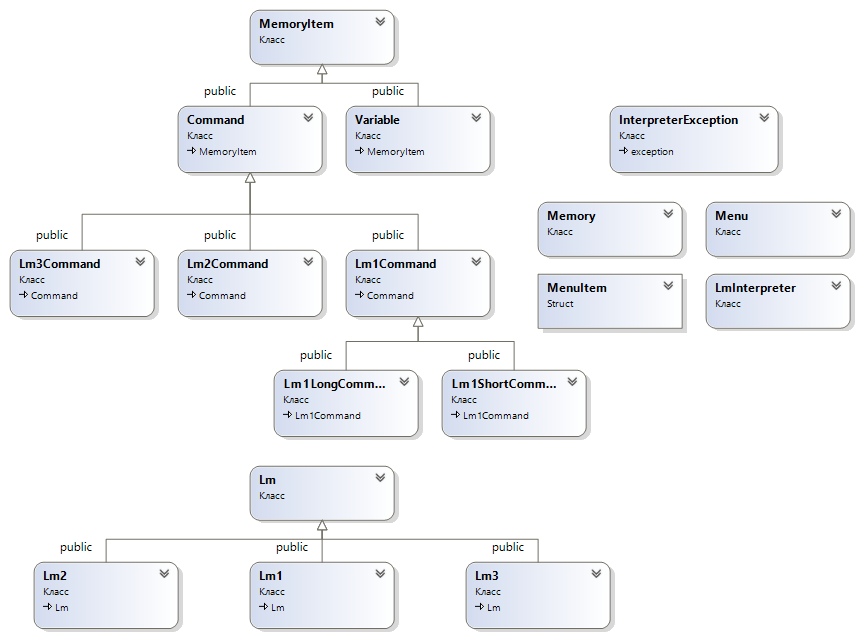


Рис. 6. Диаграмма классов

# 6 ТЕСТИРОВАНИЕ ПРОГРАММЫ

Программная система успешно прошла финальные тесты. При разработке системы были выявлены и устранены многочисленные ошибки наблюдавшиеся у всех учебных машин. Были устранены ошибки:

1. Ошибка очистки оперативной памяти.
2. Ошибка присвоения остатка при делении.
3. Ошибка множественного присвоения.
4. Ошибка сохранения файлов.
5. Ошибка с запуском ненаписанной программы.

Создан класс InterpritorExeption для обработки всех возможных ошибок.

Финальный тест УМ-3:

#init

0001 "A"

0002 "B" 5

0003 10

0005 "Variable" 1

0006 "Var" 2

#end

#code

0100 06 0001 FFFF FFFF ; Инициализация

0101 07 0001 FFFF FFFF

0102 07 0002 FFFF FFFF

0103 07 0003 FFFF FFFF

0104 01 0001 0002 0003 ; Сложение

0105 07 0003 FFFF FFFF

0106 02 0001 0002 0003 ; Вычитание

0107 07 0003 FFFF FFFF

0108 00 0001 FFFF 0003 ; Пересылка

0109 07 0003 FFFF FFFF

010A 03 0001 0002 0003 ; Умножение со знаком

010B 07 0003 FFFF FFFF

010C 13 0001 0002 0003 ; Умножение без знака

010D 07 0003 FFFF FFFF

010E 04 0001 0002 0003 ; деление со знаком

010F 07 0003 FFFF FFFF

0110 14 0001 0002 0003 ; деление без знака

0111 07 0003 FFFF FFFF

0112 80 FFFF FFFF 0113 ; Без условный переход

0113 81 0001 0001 0115 ; Переход по =

0114 07 0006 FFFF FFFF

0115 07 0005 FFFF FFFF

0116 82 0001 0002 0118 ; Переход по !=

0117 07 0006 FFFF FFFF

0118 07 0005 FFFF FFFF

0119 83 0001 0001 011B ; Переход по меньше

011A 07 0006 FFFF FFFF

011B 07 0005 FFFF FFFF

011C 93 0001 0002 011E ; Переход по меньше без знака

011D 07 0006 FFFF FFFF

011E 07 0005 FFFF FFFF

011F 84 0001 0002 0121 ; Переход по больше либо равно

0120 07 0006 FFFF FFFF

0121 07 0005 FFFF FFFF

0122 94 0001 0001 0124 ; Переход по больше либо равно без знака

0123 07 0006 FFFF FFFF

0124 07 0005 FFFF FFFF

0125 85 0001 0002 0127 ; Переход по больше

0126 07 0006 FFFF FFFF

0127 07 0005 FFFF FFFF

0128 95 0002 0001 012A ; Переход по больше без знака

0129 07 0006 FFFF FFFF

012A 07 0005 FFFF FFFF

012B 86 0002 0001 012D ; Переход по меньше либо равно

012C 07 0006 FFFF FFFF

012D 07 0005 FFFF FFFF

012E 96 0001 0001 0130 ; Переход по меньше либо равно без знака

012F 07 0006 FFFF FFFF

0130 07 0005 FFFF FFFF

0131 99 FFFF FFFF FFFF ; Стоп

#end

# ЗАКЛЮЧЕНИЕ

В процессе курсового проектирования была разработана программная система. При этом были решены следующие задачи:

1. спроектирован интерфейс и структура программы;
2. реализована концепция учебных машин.