



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
филиал в г. Сызрани

---

Кафедра «Информатика и системы управления»

А.В. ТАРАКАНОВ

# **ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, СИСТЕМЫ И СЕТИ**

*Лабораторный практикум*  
Часть 1

Самара  
Самарский государственный технический университет  
2016

Печатается по решению редакционно-издательского совета СамГТУ

УДК 004

**Тараканов А. В.**

**Вычислительные машины, системы и сети:** практикум / *А. В. Тараканов.*

– Самара: Самар. гос. техн. ун-т, 2016. – 59 с.: ил.

Лабораторный практикум содержит рекомендации по выполнению первой части лабораторных работ, соответствующих курсу «Вычислительные машины, системы и сети». Изложены принципы организации и работы основных блоков ЭВМ, таких как микропроцессор и память. Рассмотрены вопросы программирования микропроцессора и систем на его основе на уровне машинных команд.

Предназначен для бакалавров, обучающихся по направлению 15.03.04 «Автоматизация технологических процессов и производств») при выполнении ими лабораторных работ.

УДК 004

Рецензенты: зав. кафедрой ЭПП, Сф Сам ГТУ канд. техн. наук  
*Вокин И. А.*

зав. кафедрой «Промышленная информатика» ОАНО  
ВО «ВУ и Т» канд. техн. наук, доцент *Куралесов Н. О.*

© А. В. Тараканов, 2016

© Самарский государственный  
технический университет, 2016

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ЛАБОРАТОРНАЯ РАБОТА №1. Изучение устройства и технических характеристик учебного микропроцессорного комплекса .....	5
ЛАБОРАТОРНАЯ РАБОТА №2. Изучение системы команд микропроцессора КР 580 ИК80А.....	12
ЛАБОРАТОРНАЯ РАБОТА №3. Разработка и отладка программ для микропроцессора КР 580 ИК80А.....	17
ЛАБОРАТОРНАЯ РАБОТА №4. Разработка программы контроля управляющих параметров .....	21
ЗАКЛЮЧЕНИЕ.....	23
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	24
ПРИЛОЖЕНИЕ .....	25

## ВВЕДЕНИЕ

В настоящее время при освоении информационных технологий неизбежно возникает необходимость детально разобраться в работе основного узла любой ЭВМ – микропроцессора. Однако современные микропроцессоры настолько сложны, что это в значительной степени затруднительно. Используя учебный микропроцессорный комплекс можно достаточно быстро и наглядно постичь принципы работы микропроцессора. К числу преимуществ использования учебного микропроцессорного комплекса можно отнести его достаточную простоту и наглядность, возможность создания программ различной сложности, пошаговое выполнение программы, с просмотром результатов после выполнения каждой отдельной итерации. Все вышеперечисленное определяет актуальность применения учебного микропроцессорного комплекса при изучении и получении практических навыков работы с микропроцессором в рамках подготовки бакалавров. Первая часть лабораторного практикума включает четыре работы.

В первой лабораторной работе студенты изучают устройство и технические характеристики учебного микропроцессорного комплекса и приобретают навыки работы с ним.

Вторая работа посвящена изучению системы команд микропроцессора КР 580 ИК80А, а также составлению простейших программ.

Третья лабораторная работа предполагает разработку и отладку программ, реализующих арифметические и логические операции обработки данных с циклической и разветвляющейся структурой.

Четвертая работа посвящена разработке программ контроля управляющих параметров.

По результатам каждой лабораторной работы студентами должен быть составлен отчет о выполнении заданий, с анализом полученных результатов.

## Лабораторная работа № 1

### Изучение устройства и технических характеристик учебного микропроцессорного комплекса

*Цель работы:* изучение устройства микро-ЭВМ, построенных на базе микропроцессора КР 580ИК80А, изучение технических характеристик учебного микропроцессорного комплекса.

Оборудование: учебный микропроцессорный комплекс (УМК).

### Устройство и принцип работы УМК

Учебный микропроцессорный комплекс, показанный на рисунке 1, состоит из следующих частей: микро-ЭВМ, пульт оператора, блок питания.

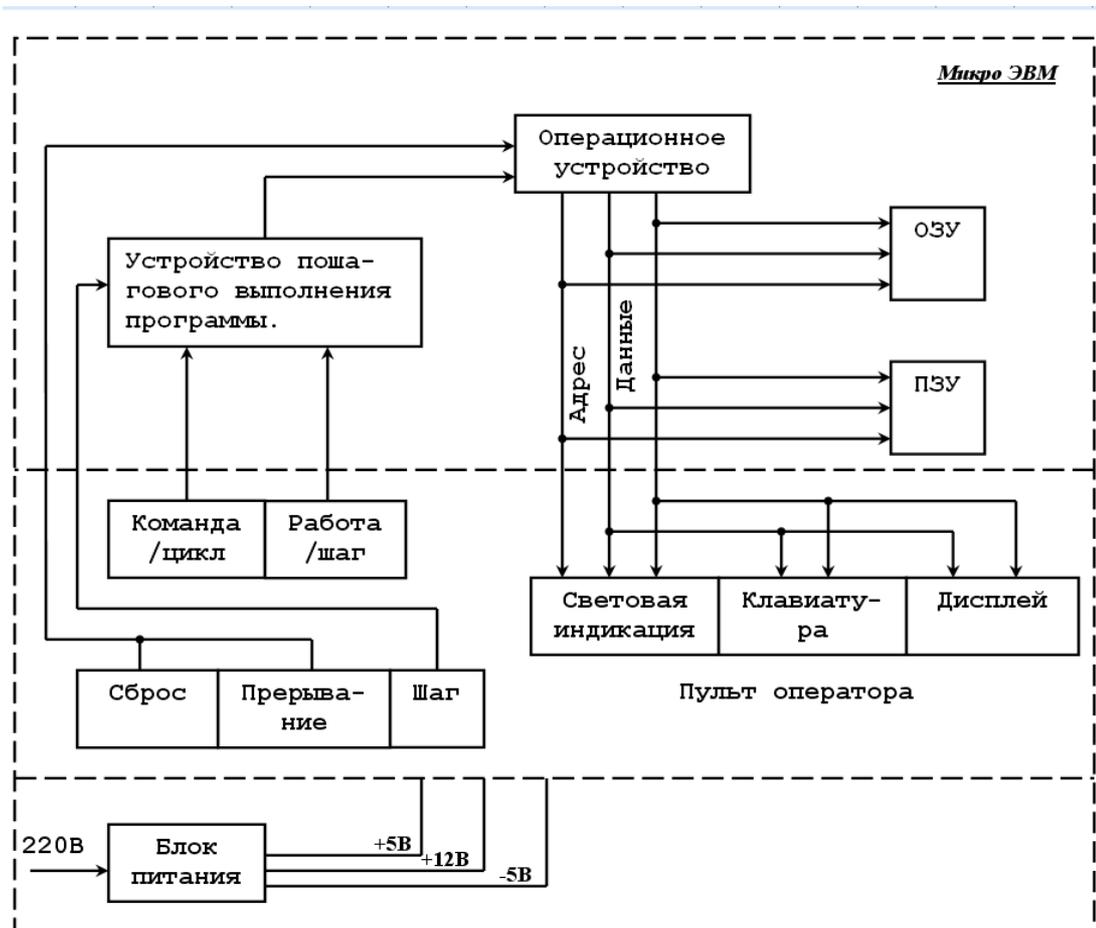


Рис. 1. Внутреннее устройство УМК

Основой всего УМК является микро-ЭВМ, которая и решает основные задачи. Пульт оператора предназначен для взаимодействия оператора с микро-ЭВМ. Блок питания обеспечивает постоянным стабилизированным напряжением все узлы и блоки УМК. Микро-ЭВМ состоит из операционного устройства, оперативного запоминающего устройства (ОЗУ), постоянного запоминающего устройства (ПЗУ) и устройства пошагового выполнения программы. Пульт оператора состоит из клавиатуры, шестиразрядного дисплея, световых индикаторов и кнопок управления: «СБ» - сброс, «ПР» - прерывание, «ШГ» - шаг, а также переключателей «РБ/ШГ» - работа/шаг и «КМ/ЦК» - команда/цикл.

Основой микро-ЭВМ является операционное устройство (ОУ), которое в общем случае, бывает представлено, как правило, арифметико-логическим устройством микропроцессора. После нажатия кнопки «СБ» операционное устройство принимает исходное состояние. Информация о состоянии операционного устройства фиксируется в регистре состояния в начале каждого машинного цикла. В таблице 1 приведены возможные состояния операционного устройства.

*Таблица 1*

**Возможные состояния операционного устройства**

Состояние операционного устройства	Разряды регистра состояния операционного устройства							
	D0 INTA	D1 WO	D2 STACK	D3 HLTA	D4 OUT	D5 MI	D6 INP	D7 MEMR
Выборка команды	0	1	0	0	0	1	0	1
Чтение из памяти	0	1	0	0	0	0	0	1
Запись в память	0	0	0	0	0	0	0	0
Чтение из стека	0	1	1	0	0	0	0	1
Запись в стек	0	0	1	0	0	0	0	0
Ввод	0	1	0	0	0	0	1	0
Вывод	0	0	0	0	1	0	0	0
Прерывание	1	1	0	0	0	1	0	0
Останов	0	1	0	1	0	0	0	1

Определение значения каждого бита регистра состояния операционного устройства представлено в таблице 2.

Таблица 2

**Значения отдельных бит регистра состояния операционного устройства**

Сигнал	Разряд регистра состояния операционного устройства	Значение
INTA	D0	Подтверждение запроса на прерывание. Используется для подачи на шину данных команды.
WO	D1	Указывает, что в текущем цикле выполняется запись в память или операция ввода.
STACK	D2	Означает наличие на шине адреса: указателя стека.
HLTA	D3	Сигнал подтверждения команды.
OUT	D4	Означает, что в текущем цикле выполняется операция вывода.
MI	D5	Указывает, что текущий машинный цикл служит для выборки первого байта команды.
INP	D6	Означает, что в текущем цикле выполняется операция ввода.
MEMR	D7	Указывает, что в текущем машинном цикле будет производиться чтение из памяти.

Краткие технические характеристики учебного микропроцессорного комплекса представлены в таблице 3.

Таблица 3

**Технические характеристики УМК**

Тип применяемого микропроцессора	КР 580 ИК 80 А
Объем ОЗУ	1 к Байт
Общий объем ПЗУ	2 к Байта
Объем ПЗУ пользователя	1 к Байт
Возможность прерывания	1 вектор
Программное обеспечение	Системная программа «Монитор»

В постоянном запоминающем устройстве хранятся неизменяющиеся программы и данные, где 1 к Байт занимает системная программа «Монитор», находящаяся по адресам 0000 ... 03FF. Оперативное запоминающее устройство предназначено для изменяющихся данных и программ. Область ОЗУ, доступная для пользователя располагается по адресам с 0800 по 0BFF и имеет объем в 1 к Байт.

Устройство пошагового выполнения программ переводит операционное устройство в состояние «Ожидание» после выполнения очередного шага. Возможны два пошаговых режима работы: покомандный шаг и поцикловой шаг. Выбор пошагового режима работы осуществляется переключателем «РБ/ШГ», выбор величины шага – переключателем «КМ/ЦК». Для выполнения последующего шага необходимо нажать кнопку «ШГ». Выполнение программы может быть приостановлено нажатием управляющей кнопки «ПР».

Клавиатура состоит из 24 клавиш, из них 8 клавиш – директивные, 16 – информационные.

Директивные клавиши:

«П» – чтение и изменение содержимого оперативной памяти.

«РГ» – чтение и изменение содержимого регистров микропроцессора.

«СТ» – передача управления программе пользователя.

«КС» – определение контрольной суммы массива памяти.

«ЗК» – заполнение массива константой.

«ПМ» – перемещение массива данных в адресном пространстве памяти.

«| \_ |» – клавиша пробела, служит для разделения нескольких переменных при вводе.

«ВП» – выполнение, означает конец директивы.

Информационные клавиши служат для ввода чисел в шестнадцатиричном коде. Клавиши с 4/PH по F служат для вызова идентификаторов регистра микропроцессора.

При неправильной работе с клавиатурой в крайней правой позиции дисплея появляется знак «?».

### **Порядок выполнения работы.**

1. Подготовка УМК к работе.
  - 1.1. Предварительно внимательно ознакомиться с инструкцией по технике безопасности и паспортом на УМК.
  - 1.2. Изучить клавиатуру УМК для управления с клавиатуры с помощью директив.
2. Оформить в виде таблицы (по примеру таблицы 4) запись четырех чисел в оперативную память и четырех чисел в регистры, после чего предоставить преподавателю на проверку.
3. Установить кнопку УМК «~» в отжатое состояние.
4. Подключить УМК к сети переменного тока 220 В.
5. Переключатель «РБ/ШГ» установить в состояние – работа.
6. Включить УМК нажатием кнопки «~».
7. Нажать управляющую кнопку сброса «СБ», при этом на дисплее в крайней левой позиции должен появиться знак «-». При отсутствии сброса повторное включение УМК производить через промежутки не менее 10 секунд после включения.
8. Ввести данные в память и вывести индикацию содержимого для контроля.

Для ввода данных в память необходимо последовательно нажать следующие клавиши: «П» X1, X2, X3, X4 «| \_ |» Д1, «| \_ |» Д2, «| \_ |» ... Д<sub>n</sub>, «ВП», где X1, X2, X3, X4 – адрес начальной ячейки памяти для записываемых данных; Д1, Д2, ..., Д<sub>n</sub> – данные, подлежащие записи в память. Для перехода к следующей ячейке памяти без изменения содержимого иницируемой, не набирая новых данных, нажмите клавишу «| \_ |».

9. Ввести данные в четыре произвольных регистра микропроцессора и вывести индикацию содержимого для контроля.

Для ввода данных в регистры микропроцессора необходимо нажать клавишу «РГ», а затем идентификатор регистра. Идентификатором регистра являются символы, обозначающие регистры микропроцессора.

А – регистр А (8 бит) – «аккумулятор»;

В – регистр В (8 бит);

С – регистр С (8 бит);

D – регистр D (8 бит);

Е – регистр Е (8 бит);

Н – регистр Н (8 бит) – содержит старшую часть адреса при косвенной адресации;

L – регистр L (8 бит) – содержит младшую часть адреса при косвенной адресации;

F – регистр условий (8 бит);

SL – младший байт указателя стека (8 бит);

SH – старший байт указателя стека (8 бит);

PL – младший байт счетчика команд (8 бит);

PH – старший байт счетчика команд (8 бит).

Ответом на ввод идентификатора является индикация содержимого соответствующего регистра на дисплее в виде двух шестнадцатеричных цифр. Для изменения содержимого регистра необходимо задать новое значение с помощью информационных клавиш и нажать кнопку «ВП».

10. Представить результаты выполненной работы преподавателю.

11. Выключить УМК.

12. Оформить отчет по результатам работы.

Отчет должен содержать: название и цель работы; краткое описание устройства и принципа работы УМК; результаты работы, оформленные в виде таблицы; ответы на контрольные вопросы.

**Таблица оформления результатов лабораторной работы**

Адрес ячейки памяти/Регистр	Данные

***Контрольные вопросы.***

1. Что показывает регистр состояния операционного блока?
2. Как осуществить пошаговое выполнение программы?
3. Как записать данные в оперативную память, регистр?
4. Для чего служит программный счетчик?
5. Что такое стек, и каковы основы работы с ним?
6. Для чего служит регистр условий?
7. Какими регистрами обладает микропроцессор, опишите их характеристики и назначение?

## Лабораторная работа № 2

### Изучение системы команд микропроцессора КР 580 ИК80А

*Цель работы:* изучение системы команд микропроцессора КР 580 ИК80А для написания простейших программ пользователя.

Оборудование: учебный микропроцессорный комплекс (УМК).

#### Команды микропроцессора

Команды микропроцессора можно разделить на пять групп:

1. Команды пересылки данных и загрузки – команды, позволяющие передавать данные между оперативной памятью и регистрами процессора, а также получать данные непосредственно из программного кода.
2. Арифметические команды – команды, реализующие арифметические операции над числами с фиксированной и плавающей запятой (сложение, вычитание, умножение, деление и др.).
3. Логические команды – команды, реализующие логические операции, такие как: конъюнкция (И), дизъюнкция (ИЛИ), логическое отрицание (НЕ) и др.
4. Команды передачи управления – команды переходов (условные и безусловные), команды переходов к подпрограмме.
5. Команды ввода/вывода – это команды приема и передачи информации, соответственно, из или во внешнюю среду. Прием и передача данных, как правило, осуществляется через соответствующие порты микро-ЭВМ.

Команды пересылки осуществляют передачу данных в/из регистра в ячейку(и) оперативной памяти, не изменяя при этом состояние флагов.

## Способы адресации в командах

Способом адресации можно назвать различные режимы интерпретации поля адреса команды. Число слов в команде зависит от реализуемых способов адресации, допустимого размера памяти и архитектуры микропроцессора. Для краткости изложения содержимое поля адреса будем называть исходным адресом, а адрес ячейки памяти, содержащей необходимые данные, будем называть эффективным адресом. Различают следующие способы адресации:

*Прямая адресация.* Наиболее простой и естественный способ адресации, при котором в команде задается эффективный адрес, т.е. исходный адрес равен эффективному.

*Непосредственная адресация.* При непосредственной адресации операнд является частью команды. В этом случае операнд располагается непосредственно в команде вслед за операционной частью команды.

*Индексная адресация.* Индексной называется адресация, в которой эффективный адрес получается путем сложения исходного адреса с содержимым регистра (индексного регистра).

*Косвенная адресация.* При косвенной адресации исходный адрес служит указателем на ячейку, в которой находится эффективный адрес. Таким образом, исходный адрес это «адрес адреса». Частным случаем косвенной адресации является косвенная регистровая адресация, когда исходный адрес, указывает на общий регистр, содержащий эффективный адрес. Косвенная регистровая адресация повышает эффективность программы, так как в команде в этом случае адресное поле должно вмещать только ссылку на общий регистр или регистры. Для косвенной регистровой адресации используется, как правило, пара регистров H и L.

*Относительная адресация.* При относительной адресации эффективный адрес образуется путем сложения исходного адреса и содержимого программного счетчика. Этот способ адресации особенно эффективен в командах переходов, когда переход

осуществляется в близкие ячейки, поскольку не требует задания полного адреса, по которому осуществляется переход.

*Страничная адресация.* Зачастую объем памяти микропроцессора превышает адресное пространство, т.е. максимальное число адресов для данного формата команд. В этом случае память разбивается на страницы, размер которых определяется максимальным числом ячеек, адресуемых в команде. При таком подходе программный счетчик можно считать состоящим из двух частей. Старшие разряды содержат номер страницы, а младшие разряды содержат адрес слова внутри страницы.

### **Порядок ввода и запуска программы пользователя на УМК.**

1. Выбрать участок памяти в ОЗУ и занести в него программу в шестнадцатеричном коде.
2. Выбрать участок памяти в ОЗУ и занести в него данные.
3. Передать управление программе пользователя, для чего:

Последовательно нажать следующие клавиши: «СТ» Адрес 1 «| \_ |» Адрес 2 «| \_ |» Адрес 3 «ВП», где Адрес 1 – начальный адрес программы, Адрес 2 и Адрес 3 – адреса выполнения прерывания программы.

При написании программ пользователя используются команды, описание которых приведено в приложении 1.

### **Порядок выполнения работы.**

1. Составить программу, реализующую выполнение следующих операций согласно варианту задания (таблица 5):

- 1.1. Непосредственная пересылка данных в регистр;
- 1.2. Непосредственная пересылка данных в память;
- 1.3. Непосредственная загрузка пары регистров;
- 1.4. Прямая загрузка аккумулятора;

- 1.5. Перемещение данных из аккумулятора в регистр;
- 1.6. Перемещение данных из памяти в регистр;
- 1.7. Прямая запись содержимого аккумулятора в память;
- 1.8. Прямая загрузка пары регистров H и L;
- 1.9. Косвенная загрузка аккумулятора;
- 1.10. Прямая запись содержимого пары регистров H и L в память;
- 1.11. Обмен данными между регистрами H и D, L и E, соответственно;
- 1.12. Помещение двух данных в стек;
- 1.13. Извлечение двух данных из стека.

Таблица 5

**Варианты заданий**

Вариант	Задание									
	1	2	3	4	5	6	7	8	9	10
Вариант № 1	E	0900	HL	0900	C	0900/H	0901	0900	D	0902
Вариант № 2	B	0901	BC	0901	E	0901/D	0902	0901	D	0903
Вариант № 3	C	0902	DE	0902	B	0902/L	0903	0902	B	0904
Вариант № 4	D	0903	DE	0903	C	0903/B	0904	0903	B	0905
Вариант № 5	E	0904	BC	0904	D	0904/C	0905	0904	B	0906

2. Представить программу на проверку преподавателю.
3. Ввести программу в УМК и запустить на выполнение.
4. Посмотреть ход выполнения программы в пошаговом режиме с промежуточной индикацией.
5. Представить результаты выполненной работы преподавателю.
6. Выключить УМК.
7. Оформить отчет по результатам работы.

Отчет должен содержать: название и цель работы; краткие теоретические сведения о системе команд микропроцессора КР 580 ИК80А; результаты работы, оформленные в виде таблицы; ответы на контрольные вопросы.

**Таблица оформления результатов лабораторной работы**

Адрес ячейки памяти	Команда на машинном языке	Комментарий

**Контрольные вопросы.**

1. На что влияют структура и формат команды?
2. Какие виды адресации существуют?
3. Расскажите о достоинствах и недостатках различных способов адресации.
4. Какие способы адресации применяются в современных микропроцессорах?
5. Как реализуется механизм адресации в виртуальной памяти?

### **Лабораторная работа № 3**

#### **Разработка и отладка программ для микропроцессора КР 580 ИК80А**

*Цель работы:* изучение системы команд микропроцессора КР 580 ИК80А для написания простейших программ пользователя.

Оборудование: учебный микропроцессорный комплекс (УМК).

#### **Написание программ для микропроцессора КР 580 ИК80А**

Написание программы для микропроцессора КР 580 ИК80А, входящего в состав учебного микропроцессорного комплекса, заключается в написании последовательности кодов команд, в соответствии с заданным алгоритмом.

Основную часть команд, используемых программистом при написании программ, составляют арифметические и логические команды. В полном объеме эти команды (мнемоника на ассемблере и машинные коды) представлены в приложении 1.

Команды, используемые для написания программ, могут состоять из одного, двух или трех байт. В зависимости от объема памяти отводимого для той или иной команды соответственно требуется от одного до трех ячеек оперативной памяти. В целях сокращения формата команд широко используется такой способ адресации как подразумеваемый операнд или адрес, т.е. когда операнд или его адрес находятся в каком-либо заранее определенном месте. В арифметических командах для этих целей используется регистр А, называемый «аккумулятором». При выполнении арифметических операций один из операндов, как правило, помещается в «аккумулятор», а второй может находиться как в любом регистре, так и в оперативной памяти. Результат арифметической операции также записывается в «аккумулятор».

При обработке цепочки данных или массивов довольно удобно применять косвенную регистровую адресацию, когда значение адреса операнда помещается в пару регистров H и L. Соответственно в регистр H помещается старшая часть адреса, а в регистр L младшая часть адреса. Далее для перемещения по адресному пространству достаточно изменить значение адреса, находящегося в паре регистров H и L. Для этих целей используются отдельные команды, позволяющие увеличить или уменьшить содержимое регистров H и L на единицу.

Наряду с реализацией линейных алгоритмов возникает необходимость написания программ с разветвляющейся и циклической структурой. Для этих целей арифметико-логическое устройство микропроцессора формирует ряд значений – признаков результата выполнения отдельных операций. Все эти значения объединены и хранятся в специальном регистре – регистре флагов. Каждый отдельный бит регистра флагов информирует о наступлении какого-либо события. Используя значения отдельных бит регистра флагов может осуществляться переход на другие ветви алгоритма, используя соответствующие команды условного перехода.

Завершающей командой каждой программы должна быть команда HLT (стоп) или NOT (нет операций), сигнализирующей микропроцессору о завершении программы.

### **Порядок выполнения работы**

1. Изучить арифметические и логические команды микропроцессора КР 580 ИК80А.

2. Разработать программу сложения трех произвольных чисел и записи результата в память. Первое число вводится с клавиатуры в память УМК, второе записывается непосредственной адресацией в один из регистров, третье записывается непосредственной адресацией в одну из ячеек оперативной памяти.

3. Разработать программу, реализующую следующие операции с записью результата в память:

№ варианта	Выражение
1	$(A+B-C)\vee D$
2	$(A+B+C)\vee D$
3	$(A-B+C)\vee D$
4	$(A-B-C)\vee D$
5	$(A+B-C)\wedge D$
6	$A+(B+C)\wedge D$
7	$(A-B)+C\wedge D$
8	$(A-B-C)\wedge D$
9	$A+B-(C\vee D)$
10	$A+B-(C\wedge D)$

4. Разработать программу с циклической структурой сложения двух произвольных чисел (число итераций цикла - 10) , с занесением конечного и промежуточного результата в память.

5. Разработать программу умножения двух целых положительных чисел с записью результата в память.

6. Представить программы на проверку преподавателю.

7. Ввести программы по отдельности в УМК и запустить на выполнение.

8. Посмотреть ход выполнения программы в пошаговом режиме с промежуточной индикацией.

9. Представить результаты выполненной работы преподавателю.

10. Выключить УМК.

11. Оформить отчет по результатам работы.

Отчет должен содержать: название и цель работы; краткие теоретические сведения о программировании микропроцессора КР 580 ИК80А; результаты работы, оформленные в виде таблицы; ответы на контрольные вопросы.

**Таблица оформления результатов лабораторной работы**

Адрес ячейки памяти	Команда на машинном языке	Комментарий

**Контрольные вопросы.**

1. Каким будет содержание пары регистров с данными 0001 0000 0000 1111 после инкрементирования?
2. Каким будет содержимое регистра с данными 0100 1011 после декрементирования?
3. Как влияет выполнение команды CPI на состояние флага нуля?
4. Что будет результатом операции A or A?

## **Лабораторная работа № 4**

### **Разработка программы контроля управляющих параметров**

*Цель работы:* разработка программы аварийного контроля параметров технологического процесса с использованием масштабированной суммы.

Оборудование: учебный микропроцессорный комплекс (УМК).

#### **Задание на выполнение работы**

В данной лабораторной работе моделируется ситуация, когда два параметра технологического процесса  $N_1$  и  $N_2$  загружаются и складываются, полученная сумма при этом умножается на коэффициент масштабирования  $K$ . Далее масштабированная сумма сравнивается с максимально допустимым значением, и в случае превышения выдается аварийный сигнал (выполнение программы прекращается). В противном случае осуществляется ввод новых параметров, и управляющая программа циклически повторяется. Операцию умножения на масштабируемый коэффициент  $K$  необходимо оформить в виде подпрограммы.

#### **Порядок выполнения работы.**

1. Изучить команды обращения к подпрограммам и возврата из подпрограмм, записи и извлечения данных из стека, условных и безусловных переходов.
2. Для заданных значений параметров технологического процесса и масштабного коэффициента составить программу контроля.
3. Просмотреть выполнение программы в пошаговом режиме с индикацией состояния.
4. Представить преподавателю результат выполнения программы.
5. Оформить отчет по результатам работы.

Отчет должен содержать: название и цель работы; краткие сведения о командах обращения к подпрограмме и возврата из подпрограммы, записи и извлечения данных из стека, условных и безусловных переходов; результаты работы, оформленные в виде таблицы; ответы на контрольные вопросы.

*Таблица 8*

**Таблица оформления результатов лабораторной работы**

Адрес ячейки памяти	Команда на машинном языке	Комментарий

**Контрольные вопросы.**

1. Объясните порядок записи в стек и извлечения из стека.
2. Каков алгоритм умножения?
3. Какие команды используются при обращении к подпрограммам и при возврате в основную программу?
4. Какие команды условного перехода вы знаете?
5. Как выполнение операций сравнения действует на шаговый регистр?

## **ЗАКЛЮЧЕНИЕ**

Данный лабораторный практикум позволяет сформировать профессиональные компетенции у будущих бакалавров в области информационных технологий.

В первой части лабораторных работ изучаются основы устройства и работы микропроцессора, принципы составления и отладки программ, раскрываются этапы и особенности выполнения отдельных команд, организация передачи управления в программах при создании программ с разветвляющейся и циклической структурой. Лабораторные работы подразумевают выполнение заданий и самостоятельных работ на учебном микропроцессорном комплексе.

Знания и умения, полученные в результате выполнения лабораторных работ, помогут избежать ошибок при разработке программного обеспечения, способствуют эффективному использованию средств отладки программ.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мелехин В.Ф. Вычислительные машины, системы и сети: учеб. пособие для вузов / В.Ф. Мелехин, Е.Г. Павловский, Академия. – М., 2006. - 560с.;
2. Пятибратов А.П. и др. Вычислительные системы, сети и телекоммуникации: Учебник для ВУЗов. – 4-е изд., перераб. и доп. / А.П. Пятибратов, Л.П. Гудыно, А.А. Кириченко, Финансы и статистика. – М., 2008. - 736 с., ил.;
3. Цилькер Б.Я. Организация ЭВМ и систем: учебник для вузов / Б.Я. Цилькер, С.А. Орлов, Питер. – СПб., 2006. - 668 с., ил.;
4. Таненбаум Э. Архитектура компьютера. Пер. с англ. 4-е изд. / Э. Таненбаум, Питер. – СПб., 2006. - 699 с., ил.;
5. Бройдо В.Л. Ильина О.П. Архитектура ЭВМ и систем 2-е изд. / В.Л. Бройдо, О.П. Ильина, Питер. – СПб., 2009. - 720 с., ил.;
6. Древис Ю.Г. Организация ЭВМ и вычислительных систем: учебник для вузов / Ю.Г. Древис, Высшая школа. – М., 2006. - 501 с., ил.;
7. Хорошевский В.Г. Архитектура вычислительных систем: учеб. пособ. для вузов 2-е изд. перераб. и доп. / В.Г. Хорошевский, Изд-во МГТУ им. Н.Э. Баумана. – М., 2008. - 510 с., ил.;
8. Каган Б.М. Электронные вычислительные машины и системы / Б.М. Каган, Энергоатомиздат. – М., 1991. - 592 с.

## ПРИЛОЖЕНИЕ

### *Набор команд микропроцессора КР580ИК80А*

Все команды микропроцессора КР580ИК80А делятся на пять групп.

1. Команды пересылки данных. Осуществляют пересылку данных между регистрами или между памятью и регистрами.

2. Арифметические команды. Предназначены для выполнения сложения, вычитания, увеличения или уменьшения содержимого регистров или ячеек памяти.

3. Логические команды. Выполняют логические операции "И", "ИЛИ" "исключающее ИЛИ", сравнение, сдвиг и дополнение данных в регистрах и ячейках памяти.

4. Команды условных и безусловных переходов, вызова подпрограмм и возвращения из подпрограмм.

5. Команды ввода/вывода, управления и работы со стеком. Предназначены для выполнения операции ввода/вывода, работы со стеком, управления флагами, разрешения и запрещения прерываний.

### *Краткие пояснения для описания команд микропроцессора КР 580 ИК80А*

- ✓  $r_h$  – первый (старший) регистр указанной пары;
- ✓  $r_l$  – второй (младший) регистр указанной пары;
- ✓ PC – 16-битовый регистр – программный счетчик (обозначения PCН и PCЛ используется для указания старших и младших 8 бит данного регистра соответственно) ;
- ✓ SP – 16-битовый регистр – указатель стека (обозначения SPН и SPL используются для указания старших и младших 8 бит данного регистра соответственно);
- ✓  $r_m$  – бит  $m$  регистра  $r$  (биты нумеруются от 7 до 0 слева направо);
- ✓ Z – флаг нуля;
- ✓ S – флаг знака;

- ✓ P – флаг четности ;
- ✓ CY – флаг переноса;
- ✓ AC – флаг вспомогательного переноса;
- ✓ ( ) – в скобки заключается содержимое ячейки памяти или регистра;
- ✓ ← – "Пересылается в";
- ✓ ∧ – Логическое "И";
- ✓ ∨ – "Исключающее ИЛИ";
- ✓ ∨ – Логическое "ИЛИ";
- ✓ + – Сложение;
- ✓ – – Вычитание с дополнением до 2;
- ✓ × – Умножение;
- ✓ ↔ – "Обменивается с";
- ✓ — – Дополнение до 1, т.е. инверсия;
- ✓ n – Номер рестарта от 0 до 7;
- ✓ NNN – Двоичное представление кода рестарта от 000 до 111 /от 0 до 7/.

### *Команды пересылки*

Команды этой группы осуществляют пересылку данных в/из регистр (a) и ячейки памяти и не изменяют состояния флагов.

**MOV r1, r2 (ПЕРЕСЫЛКА ДАННЫХ ИЗ РЕГИСТРА В РЕГИСТР)**

**(r1)←(r2)**

Содержимое регистра r2 пересылается в регистр r1.

0	1	D	D	D	S	S	S
---	---	---	---	---	---	---	---

MOV	A, A	7F	MOV	B, A	47	MOV	C, A	4F
	A, B	78		B, B	40		C, B	48
	A, C	79		B, C	41		C, C	49
	A, D	7A		B, D	42		C, D	4A
	A, E	7B		B, E	43		C, E	4B
	A, H	7C		B, H	44		C, H	4C
	A, L	7D		B, L	45		C, L	4D

MOV	D, A	57	MOV	E, A	5F	MOV	H, A	67
	D, B	50		E, B	58		H, B	60
	D, C	51		E, C	59		H, C	61
	D, D	52		E, D	5A		H, D	62
	D, E	53		E, E	5B		H, E	63
	D, H	54		E, H	5C		H, H	64
	D, L	55		E, L	5D		H, L	65
MOV	L, A	6F						
	L, B	68						
	L, C	69						
	L, D	6A						
	L, E	6B						
	L, H	6C						
	L, L	6D						

Циклы:1. Состояния: 4. Адресация: регистровая. Флаги: не используются.

MOV r, M (ПЕРЕСЫЛКА ДАННЫХ ИЗ ПАМЯТИ)

$(r) \leftarrow ((H) (L))$

Содержимое ячейки памяти, адрес которой хранится в регистрах H и L, пересылается в регистр r.

0	1	D	D	D	1	1	0
---	---	---	---	---	---	---	---

MOV	A, M	7E	MOV	D, M	56	MOV	L, M	6E
	B, M	46		E, M	5E			
	C, M	4E		H, M	66			

Циклы:2. Состояния: 7. Адресация: косвенная регистровая. Флаги: нет.

MOV M, r (ПЕРЕСЫЛКА ДАННЫХ В ПАМЯТЬ)

$((H) (L)) \leftarrow (r)$

Содержимое регистра r пересылается в ячейку памяти, адрес которой содержится в регистрах H и L.

0	1	1	1	0	S	S	S
---	---	---	---	---	---	---	---

MOV	M, A	77	MOV	M, D	72	MOV	M, L	75
	M, B	70		M, E	73			
	M, C	71		M, H	74			

Циклы: 2. Состояния: 7. Адресация: косвенная регистровая. Флаги: нет.

**MVI r (НЕПОСРЕДСТВЕННАЯ ПЕРЕСЫЛКА)**

$(r) \leftarrow (\text{байт } 2)$

Содержимое второго байта команды пересылается в регистр *r*.

0	0	D	D	D	1	1	0
Данные							

MVI	A, байт	3E	MVI	D, байт	16	MVI	L, байт	2E
	B, байт	06		E, байт	1E			
	C, байт	0E		H, байт	26			

Циклы: 2. Состояния: 7. Адресация: непосредственная. Флаги: нет.

**MVI M, байт (НЕПОСРЕДСТВЕННАЯ ПЕРЕСЫЛКА В ПАМЯТЬ)**

$((H) (L)) \leftarrow (\text{байт } 2)$

Содержимое второго байта команды пересылается в ячейку памяти, адрес которой указан в регистрах H и L.

0	0	1	1	0	1	1	0
Данные							

MVI	M, байт	36
-----	---------	----

Циклы: 3. Состояния: 10. Адресация: непосредственная/косвенная регистровая. Флаги: нет.

**LXI rp, 2 байта (НЕПОСРЕДСТВЕННАЯ ЗАГРУЗКА ПАРЫ РЕГИСТРОВ)**

$(rh) \leftarrow (\text{байт } 3)$

$(rl) \leftarrow (\text{байт } 2)$

Третий байт команды пересылается в старший регистр (rh) пары регистров, второй – в младший регистр (rl) пары регистров.

0	0	R	P	0	0	0	1
Младший байт данных							
Старший байт данных							

LXI	B, 2 байта	01	непосредственная загрузка пары регистров B и C
	D, 2 байта	11	непосредственная загрузка пары регистров D и E
	H, 2 байта	21	непосредственная загрузка пары регистров H и L
LXI	SP, 2 байта	31	непосредственная загрузка указателя стека

Циклы: 3. Состояния: 10. Адресация: непосредственная. Флаги: нет.

LDA [адр.] (ПРЯМАЯ ЗАГРУЗКА АККУМУЛЯТОРА)

(A) ← ((байт 3) (байт 2))

Содержимое ячейки памяти, адрес которой указан во втором и третьем байтах команды, загружается в аккумулятор.

0	0	1	1	1	0	1	0
Младший байт адреса							
Старший байт адреса							

LDA	[адр.]	3A
-----	--------	----

Циклы: 4. Состояния: 13. Адресация: прямая. Флаги: нет.

STA [адр.] (ПРЯМАЯ ЗАПИСЬ СОДЕРЖИМОГО АККУМУЛЯТОРА В ПАМЯТЬ)

((байт 3) (байт 2)) ← (A)

Содержимое аккумулятора пересылается в ячейку памяти, адрес которой указан во втором и третьем байтах команды.

0	0	1	1	0	0	1	0
Младший байт адреса							
Старший байт адреса							

STA	[адр.]	32
-----	--------	----

Циклы: 4. Состояния: 13. Адресация: прямая. Флаги: нет.

LHLD [адр.] (ПРЯМАЯ ЗАГРУЗКА Н и L)

$(L) \leftarrow ((\text{байт } 3) (\text{байт } 2))$

$(H) \leftarrow ((\text{байт } 3) (\text{байт } 2) + 1)$

Содержимое ячейки памяти, адрес которой указан во втором и третьем байтах команды, пересылается в регистр L. Содержимое ячейки памяти со следующим адресом  $((\text{адр.}) + 1)$  пересылается в регистр H.

0	0	1	0	1	0	1	0
Младший байт адреса							
Старший байт адреса							

LHLD	[адр.]	2A
------	--------	----

Циклы: 5. Состояния: 16. Адресация: прямая. Флаги: нет.

SHLD [адр.] (ПРЯМАЯ ЗАПИСЬ СОДЕРЖИМОГО Н, L РЕГИСТРОВ В ПАМЯТЬ)

$((\text{байт } 3) (\text{байт } 2)) \leftarrow (L)$

$((\text{байт } 3) (\text{байт } 2) + 1) \leftarrow (H)$

Содержимое регистра L пересылается в ячейку памяти, адрес которой определен во втором и третьем байтах команды. Содержимое H регистра пересылается в следующую ячейку памяти.

0	0	1	0	0	0	1	0
Младший байт адреса							
Старший байт адреса							

SHLD	[адр.]	22
------	--------	----

Циклы: 5. Состояния: 16. Адресация: прямая. Флаги: нет.

LDAX rp (КОСВЕННАЯ ЗАГРУЗКА АККУМУЛЯТОРА)

$(A) \leftarrow ((rp))$

Содержимое ячейки памяти, адрес которой указан в паре регистров rp, пересылается в аккумулятор.

Примечание. Могут быть указаны только пары регистров  $gr=B$  /регистры В и С/ и  $gr=D$  /регистры D и E/.

0	0	R	P	1	0	1	0
---	---	---	---	---	---	---	---

LDAX	B	0A
	D	1A

Циклы: 2. Состояния: 7. Адресация: косвенная регистровая. Флаги: нет.

STAX  $gr$  (КОСВЕННАЯ ЗАПИСЬ СОДЕРЖИМОГО АККУМУЛЯТОРА В ПАМЯТЬ)

$((gr)) \leftarrow (A)$

Содержимое аккумулятора пересылается в ячейку памяти, адрес которой определен в паре регистров  $gr$ .

Примечание. Могут быть указаны только пары регистров  $gr=B$  /регистры В и С/ или  $gr=D$  /регистры D и E/.

0	0	R	P	0	0	1	0
---	---	---	---	---	---	---	---

STAX	B	02
	D	12

Циклы: 2. Состояния: 7. Адресация: косвенная регистрация. Флаги: нет.

XCHG (ОБМЕН ДАННЫМИ МЕЖДУ РЕГИСТРАМИ H, L и D, E)

$(H) \leftrightarrow (D)$

$(L) \leftrightarrow (E)$

Содержимое регистров H и L обменивается с содержимым регистров D и E.

1	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

XCHG	EB
------	----

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги: нет.

## *Арифметические команды*

Команды этой группы выполняют арифметические операции над данными, находящимися в регистрах и ячейках памяти.

Если нет особых указаний, то команды изменяют состояния флагов нуля, четности, переноса и вспомогательного переноса в соответствии со стандартными правилами.

Все операции вычитания выполняются через арифметическое дополнение до 2, причем флаг переноса устанавливается в логическую 1, если имеется заем, и сбрасывается, если заема не было.

**ADD r (СЛОЖЕНИЕ СОДЕРЖИМОГО АККУМУЛЯТОРА С СОДЕРЖИМЫМ РЕГИСТРА)**

$$(A) \leftarrow (A) + (r)$$

Содержимое регистра r складывается с содержимым аккумулятора. Результат помещается в аккумулятор.

1	0	0	0	0	S	S	S
---	---	---	---	---	---	---	---

ADD	A	87	ADD	D	82	ADD	L	85
	B	80		E	83			
	C	81		H	84			

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги: Z, S, P, CY, AC.

**ADD M (СЛОЖЕНИЕ СОДЕРЖИМОГО АККУМУЛЯТОРА С СОДЕРЖИМЫМ ЯЧЕЙКИ ПАМЯТИ)**

$$(A) \leftarrow (A) + ((H) (L))$$

Содержимое ячейки памяти, адрес которой содержится в регистрах H и L, складывается с содержимым аккумулятора. Результат помещается в аккумулятор.

1	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

ADD	M	86
-----	---	----

Циклы: 2. Состояния: 7. Адресация: косвенная регистровая. Флаги: Z, S, P, CY, AC.

ADI байт (НЕПОСРЕДСТВЕННОЕ СЛОЖЕНИЕ)

$(A) \leftarrow (A) + (\text{байт } 2)$

Содержимое второго байта команды складывается с содержимым аккумулятора. Результат помещается в аккумулятор.

1	1	0	0	0	1	1	0
Данные							

ADI	байт	C6
-----	------	----

Циклы: 2. Состояния: 7. Адресация: непосредственная. Флаги: Z, S, P, CY, AC.

ADC r (СЛОЖЕНИЕ СОДЕРЖИМОГО АКУМУЛЯТОРА С СОДЕРЖИМЫМ РЕГИСТРА И БИТОМ ФЛАГА ПЕРЕНОСА)

$(A) \leftarrow (A) + (r) + (CY)$

Содержимое регистра r и бит флага переноса складываются с содержимым аккумулятора. Результат помещается в аккумулятор.

1	0	0	0	1	S	S	S
---	---	---	---	---	---	---	---

ADC	A	8F	ADC	D	8A	ADC	L	8D
	B	88		E	8B			
	C	89		H	8C			

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги: Z, S, P, CY, AC.

ADC M (СЛОЖЕНИЕ СОДЕРЖИМОГО АКУМУЛЯТОРА С СОДЕРЖИМЫМ ЯЧЕЙКИ ПАМЯТИ И БИТОМ ПЕРЕНОСА)

$(A) \leftarrow (A) + ((H) (L)) + (CY)$

Содержимое ячейки памяти, адрес которой указан в регистрах H и L, и содержимое бита флага переноса складываются с содержимым аккумулятора. Результат помещается в аккумулятор.

1	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---

ADC	M	8E
-----	---	----

Циклы: 2. Состояния: 7. Адресация: косвенная регистровая. Флаги: Z, S, P, CY, AC.

ACI байт (НЕПОСРЕДСТВЕННОЕ СЛОЖЕНИЕ С БИТОМ ПЕРЕНОСА)

$$(A) \leftarrow (A) + (\text{байт 2}) + (CY)$$

Содержимое второго байта команды и бита флага переноса складывается с содержимым аккумулятора. Результат помещается в аккумулятор.

1	1	0	0	1	1	1	0
Данные							

ACI	байт	CE
-----	------	----

Циклы: 2. Состояния: 7. Адресация: непосредственная. Флаги: Z, S, P, CY, AC.

SUB r (ВЫЧИТАНИЕ СОДЕРЖИМОГО РЕГИСТРА)

$$(A) \leftarrow (A) - (r)$$

Содержимое регистра вычитается из содержимого аккумулятора. Результат помещается в аккумулятор.

1	0	0	1	0	S	S	S
---	---	---	---	---	---	---	---

SUB	A	97	SUB	D	92	SUB	L	95
	B	90		E	93			
	C	91		H	94			

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги: Z, S, P, CY, AC.

**SUB M (ВЫЧИТАНИЕ СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ)**

$(A) \leftarrow (A) - ((H) (L))$

Содержимое ячейки памяти, адрес которой содержится в H и L регистрах, вычитается из содержимого аккумулятора. Результат помещается в аккумулятор.

1	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

SUB	M	96
-----	---	----

Циклы: 2. Состояния: 7. Адресация: косвенная регистровая. Флаги: Z, S, P, CY, AC.

**SUI байт (НЕПОСРЕДСТВЕННОЕ ВЫЧИТАНИЕ)**

$(A) \leftarrow (A) - (\text{байт } 2)$

Содержимое второго байта команды вычитается из содержимого аккумулятора. Результат помещается в аккумулятор.

1	1	0	1	0	1	1	0
Данные							

SUI	байт	D6
-----	------	----

Циклы: 2. Состояния: 7. Адресация: непосредственная. Флаги: Z, S, P, CY, AC.

**SBV r (ВЫЧИТАНИЕ СОДЕРЖИМОГО РЕГИСТРА И БИТА ПЕРЕНОСА)**

$(A) \leftarrow (A) - (r) - (CY)$

Содержимое регистра r и бит флага переноса CY вычитаются из содержимого аккумулятора. Результат помещается в аккумулятор.

1	0	0	1	1	S	S	S
---	---	---	---	---	---	---	---

SBB	A	9F	SBB	D	9A	SBB	L	9D
	B	98		E	9B			
	C	99		H	9C			

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги: Z, S, P, CY, AC.

**SBB M (ВЫЧИТАНИЕ СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ И БИТА ПЕРЕНОСА)**

$$(A) \leftarrow (A) - ((H) (L)) - (CY)$$

Содержимое ячейки памяти, адрес которой содержится в регистрах H и L, и содержимое бита флага переноса CY вычитаются из содержимого аккумулятора. Результат помещается в аккумулятор.

1	0	0	1	1	1	1	0
---	---	---	---	---	---	---	---

SBB	M	9E
-----	---	----

Циклы: 2. Состояния: 7. Адресация: косвенная регистровая. Флаги: Z, S, P, CY, AC.

**SBI байт (НЕПОСРЕДСТВЕННОЕ ВЫЧИТАНИЕ С ЗАЕМОМ)**

$$(A) \leftarrow (A) - (\text{байт 2}) - (CY)$$

Содержимое второго байта команды и бита флага переноса CY вычитаются из аккумулятора. Результат помещается в аккумулятор.

1	1	0	1	1	1	1	0
Данные							

SBI	байт	DE
-----	------	----

Циклы: 2. Состояния: 7. Адресация: непосредственная. Флаги: Z, S, P, CY, AC.

### INR r (УВЕЛИЧЕНИЕ СОДЕРЖИМОГО РЕГИСТРА)

$$(r) \leftarrow (r) + 1$$

Содержимое регистра r увеличивается на единицу.

Примечание. Команда влияет на состояние всех флагов, кроме флага переноса CY.

0	0	D	D	D	1	0	0
---	---	---	---	---	---	---	---

INR	A	3C	INR	D	14	INR	L	26
	B	04		E	1C			
	C	0C		H	24			

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги: Z, S, P, AC.

### INR M (УВЕЛИЧЕНИЕ СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ)

$$((H) (L)) \leftarrow ((H) (L)) + 1$$

Содержимое ячейки памяти, адрес которой содержится в регистрах H и L, увеличивается на единицу.

Примечание. Команда воздействует на все флаги, кроме флага переноса CY.

0	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---

INR	M	34
-----	---	----

Циклы: 3. Состояния: 10. Адресация: косвенная регистровая. Флаги: Z, S, P, AC.

### DCR r (УМЕНЬШЕНИЕ СОДЕРЖИМОГО РЕГИСТРА)

$$(r) \leftarrow (r) - 1$$

Содержимое регистра r уменьшается на единицу.

Примечание. Команда воздействует на все флаги, кроме флага переноса CY.

0	0	D	D	D	1	0	1
---	---	---	---	---	---	---	---

DCR	A	3D	DCR	D	15	DCR	L	2D
	B	05		E	1D			
	C	0D		H	25			

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги: Z, S, P, AC.  
**DCR M (УМЕНЬШЕНИЕ СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ)**  
 $((H) (L)) \leftarrow ((H) (L)) - 1$

Содержимое ячейки памяти, адрес которой содержится в регистрах H и L, уменьшается на единицу.

Примечание. Команда воздействует на все флаги, кроме флага переноса CY.

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

DCR	M	35
-----	---	----

Циклы: 3. Состояния: 10. Адресация: косвенная регистровая. Флаги: Z, S, P, AC.

**INX rp (УВЕЛИЧЕНИЕ СОДЕРЖИМОГО ПАРЫ РЕГИСТРОВ)**  
 $(rh) (rl) \leftarrow (rh) (rl) + 1$

Содержимое пары регистров rp увеличивается на единицу.

Примечание. Команда не воздействует на флаги.

0	0	R	P	0	0	1	1
---	---	---	---	---	---	---	---

INX	B	03	(пара регистров B и C)
	D	13	(пара регистров D и E)
	H	23	(пара регистров H и L)
	SP	33	(указатель стека)

Циклы: 1. Состояния: 6. Адресация: регистровая. Флаги: нет.

## DCX rp (УМЕНЬШЕНИЕ СОДЕРЖИМОГО ПАРЫ РЕГИСТРОВ)

$(rh) (rl) \leftarrow (rh) (rl) - 1$

Содержимое пары регистров rp уменьшается на единицу.

Примечание. Команда не воздействует на флаги.

0	0	R	P	1	0	1	1
---	---	---	---	---	---	---	---

DCX	B	0B	(пара регистров B и C)
	D	1B	(пара регистров D и E)
	H	2B	(пара регистров H и L)
	SP	3B	(указатель стека)

Циклы: 1. Состояния: 6. Адресация: регистровая. Флаги: нет.

## DAD rp (СЛОЖЕНИЕ СОДЕРЖИМОГО ПАРЫ РЕГИСТРОВ С СОДЕРЖИМЫМ РЕГИСТРОВ Н И L)

$(H) (L) \leftarrow (H) (L) + (rh) (rl)$

Содержимое пары регистров rp складывается с содержимым регистров H и L. Результат помещается в пару регистров H,L.

Примечание. Используется только флаг CY. Он устанавливается, если имеется перенос при сложении с двойной точностью, в противном случае он сбрасывается.

0	0	R	P	1	0	0	1
---	---	---	---	---	---	---	---

DAD	B	09	(пара регистров B и C)
	D	19	(пара регистров D и E)
	H	29	(пара регистров H и L)
	SP	39	(указатель стека)

Циклы: 3. Состояния: 10. Адресация: регистровая. Флаги: CY.

## ДАА (ДЕСЯТИЧНОЕ ДОПОЛНЕНИЕ АККУМУЛЯТОРА)

8-битное число в аккумуляторе дополняется до представления в виде 4-битных чисел в двоично-десятичном коде с помощью следующих операций.

1. Если число, представленное четырьмя младшими битами аккумулятора, больше 8 или установлен флаг вспомогательного переноса AC, то к содержимому аккумулятора добавляется 6.

2. Если теперь число, представленное четырьмя старшими битами аккумулятора, больше 9 или установлен флаг переноса CY, то число 6 добавляется к числу, образованному четырьмя старшими битами аккумулятора.

Примечание. Используются все флаги.

0	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---

DAA	27
-----	----

Циклы: 1. Состояния: 4. Флаги: Z, S, P, CY, AC.

### *Логические команды*

Команды этой группы выполняют логические (булевы) операции над данными, хранящимися в регистрах и ячейках памяти и флагами.

Если нет особых указаний, то все команды воздействуют на флаги: нуля, четности, знака, переноса и вспомогательного переноса в соответствии со стандартными правилами.

ANA r (ОПЕРАЦИЯ «И» НАД СОДЕРЖИМЫМ РЕГИСТРА И АККУМУЛЯТОРА)

$(A) \leftarrow (A) \wedge (r)$

Над содержимым регистра r и аккумулятора выполняется операция логическое «И». Результат помещается в аккумулятор.

Сбрасывается флаг переноса  $CY$ , устанавливается флаг вспомогательного переноса  $AC$ .

1	0	1	0	0	S	S	S
---	---	---	---	---	---	---	---

ANA	A	A7	ANA	D	A2	ANA	L	A5
	B	A0		E	A3			
	C	A1		H	A4			

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги: Z, S,  $CY$ ,  $AC$ , P.

ANA M (ОПЕРАЦИЯ «И» НАД СОДЕРЖИМЫМ ЯЧЕЙКИ ПАМЯТИ И АККУМУЛЯТОРА)

$(A) \leftarrow (A) \wedge ((H) (L))$

Над содержимым ячейки памяти, адрес которой содержится в регистрах H и L, и содержимым аккумулятора выполняется операция логическая «И». Результат помещается в аккумулятор. Сбрасывается флаг переноса  $CY$  и вспомогательного переноса  $AC$ .

1	0	1	0	0	1	1	0
---	---	---	---	---	---	---	---

ANA	M	A6
-----	---	----

Циклы: 2. Состояния: 7. Адресация: косвенная регистровая. Флаги: Z, S, P,  $AC$ ,  $CY$ .

XRI байт (НЕПОСРЕДСТВЕННАЯ ОПЕРАЦИЯ)

$(A) \leftarrow (A) \wedge (\text{байт } 2)$

Над содержимым второго байта команды и аккумулятора выполняется операция логическое «И». Результат помещается в аккумулятор. Сбрасываются флаги переноса  $CY$  и вспомогательного переноса  $AC$ .

1	1	1	0	0	1	1	0
Данные							

XRI	байт	E6
-----	------	----

Циклы: 2. Состояния: 7. Адресация: непосредственная. Флаги: Z, S, P, CY, AC.

**XRA r** (ОПЕРАЦИЯ «ИСКЛЮЧАЮЩЕЕ ИЛИ» НАД СОДЕРЖИМЫМ АККУМУЛЯТОРА И РЕГИСТРА)

$(A) \leftarrow (A) \vee (r)$

Над содержимым регистра r и аккумулятора выполняется операция «исключающее ИЛИ». Результат помещается в аккумулятор. Сбрасываются флаги переноса CY и вспомогательного переноса AC.

1	0	1	0	1	S	S	S
---	---	---	---	---	---	---	---

XRA	A	AF	XRA	D	AA	XRA	L	AD
	B	A8		E	AB			
	C	A9		H	AC			

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги: Z, S, P, CY, AC.

**XRA M** (ОПЕРАЦИЯ «ИСКЛЮЧАЮЩЕЕ ИЛИ» НАД СОДЕРЖИМЫМ ЯЧЕЙКИ ПАМЯТИ И АККУМУЛЯТОРА)

$(A) \leftarrow (A) \vee ((H) (L))$

Над содержимым ячейки памяти, адрес которой содержится в регистрах H и L, и содержимым аккумулятора выполняется операция «исключающее ИЛИ». Результат помещается в аккумулятор. Сбрасываются флаги переноса CY и вспомогательного переноса AC.

1	0	1	0	1	1	1	0
---	---	---	---	---	---	---	---

XRA	M	AE
-----	---	----

Циклы: 2. Состояния: 7. Адресация: косвенная регистровая. Флаги: Z, S, P, CY, AC.

XRI байт (НЕПОСРЕДСТВЕННАЯ ОПЕРАЦИЯ «ИСКЛЮЧАЮЩЕЕ ИЛИ»)

$(A) \leftarrow (A) \vee (\text{байт } 2)$

Над содержимым аккумулятора и второго байта команды выполняется операция «исключающее ИЛИ». Результат помещается в аккумулятор. Сбрасываются флаги переноса CY и вспомогательного переноса AC.

1	1	1	0	1	1	1	0
Данные							

XRI	байт	EE
-----	------	----

Циклы: 2. Состояния: 7. Адресация: непосредственная. Флаги: Z, S, P, CY, AC.

ORA r (ЛОГИЧЕСКОЕ СЛОЖЕНИЕ («ИЛИ» АККУМУЛЯТОРА С СОДЕРЖИМЫМ РЕГИСТРА)

$(A) \leftarrow (A) \vee (r)$

Содержимое регистра r логически складывается с содержимым аккумулятора. Результат помещается в аккумулятор. Сбрасываются флаги переноса CY и вспомогательного переноса AC.

1	0	1	1	0	S	S	S
---	---	---	---	---	---	---	---

ORA	A	B7	ORA	D	B2	ORA	L	B5
	B	B0		E	B3			
	C	B1		H	B4			

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги: Z, S, P, CY, AC.

### ORA M (ЛОГИЧЕСКОЕ СЛОЖЕНИЕ («ИЛИ») СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ ИЛИ АККУМУЛЯТОРА)

$$(A) \leftarrow (A) \vee ((H) (L))$$

Содержимое ячейки памяти, адрес которой определен в регистрах H и L, логически складывается с содержимым аккумулятора. Результат помещается в аккумулятор. Сбрасываются флаги переноса CY и вспомогательного переноса AC.

1	0	1	1	0	1	1	0
---	---	---	---	---	---	---	---

ORA	M	B6
-----	---	----

Циклы: 2. Состояния: 7. Адресация: косвенная регистровая. Флаги: Z, S, P, CY, AC.

### ORI байт (НЕПОСРЕДСТВЕННОЕ ЛОГИЧЕСКОЕ СЛОЖЕНИЕ)

$$(A) \leftarrow (A) \vee (\text{байт } 2)$$

Содержимое второго байта команды складывается непосредственно с содержимым аккумулятора. Результат помещается в аккумулятор. Сбрасываются флаги переноса CY и вспомогательного переноса AC.

1	1	1	1	0	1	1	0
Данные							

ORI	байт	F6
-----	------	----

Циклы: 2. Состояния: 7. Адресация: непосредственная. Флаги: Z, S, P, CY, AC.

### CMR r (СРАВНЕНИЕ СОДЕРЖИМОГО РЕГИСТРА С СОДЕРЖИМЫМ АККУМУЛЯТОРА)

$$(A) - (r)$$

Содержимое регистра  $r$  вычитается из содержимого аккумулятора. Аккумулятор не изменяется. По результату вычитания флаги устанавливаются следующим образом: флаг нуля  $Z$  – в 1, если  $(A) = (r)$ ; флаг переноса  $CY$  – в 1, если  $(A) < (r)$ .

1	0	1	1	1	S	S	S
---	---	---	---	---	---	---	---

CMP	A	BF	CMP	D	BA	CMP	L	BD
	B	B8		E	BB			
	C	B9		H	BC			

Циклы: 1. Состояния: 4. Адресация: регистровая. Флаги:  $Z$ ,  $S$ ,  $P$ ,  $CY$ ,  $AC$ .

**СМР М (СРАВНЕНИЕ СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ С СОДЕРЖИМЫМ АККУМУЛЯТОРА)**

$(A) - ((H) (L))$

Содержимое ячейки памяти, адрес которой определен в регистрах  $H$  и  $L$ , вычитается из содержимого аккумулятора. Аккумулятор не изменяется. Флаги устанавливаются по результату вычитания следующим образом: флаг нуля  $Z$  – в 1, если  $(A) = ((H) (L))$ ; флаг переноса  $CY$  – в 1, если  $A < ((H) (L))$ .

1	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---

CMP	M	BE
-----	---	----

Циклы: 2. Состояния: 7. Адресация: косвенная регистровая. Флаги:  $Z$ ,  $P$ ,  $S$ ,  $CY$ ,  $AC$ .

**СРІ (байт) (НЕПОСРЕДСТВЕННОЕ СРАВНЕНИЕ)**

$(A) - (\text{байт } 2)$

Содержимое второго байта команды вычитается из содержимого аккумулятора. По результату вычитания флаги устанавливаются

следующим образом: флаг нуля Z – в 1, если (A) = (байт 2); флаг переноса CY – в 1, если (A) < (байт 2).

1	1	1	1	1	1	1	0
Данные							

CPI	байт	FE
-----	------	----

Циклы: 2. Состояния: 7. Адресация: непосредственная. Флаги: Z, S, P, CY, AC.

### RLC (ЦИКЛИЧЕСКИЙ СДВИГ ВЛЕВО)

$(A_{n+1}) \leftarrow (A_n), (A_0) \leftarrow (A_7), (CY) \leftarrow (A_7)$

Содержимое аккумулятора сдвигается влево на одну позицию. Содержимое самого старшего бита заносится в младший бит и бит флага переноса. Команда воздействует только на флаг переноса CY.

0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---

RLC	07
-----	----

Циклы: 1. Состояния: 4. Флаги: CY.

### RRC (ЦИКЛИЧЕСКИЙ СДВИГ ВПРАВО)

$(A_n) \leftarrow (A_{n+1}), (A_7) \leftarrow (A_0), (CY) \leftarrow (A_0)$

Содержимое аккумулятора сдвигается на одну позицию вправо. Содержимое младшего бита заносится в самый старший бит и бит флага переноса. Команда воздействует только на флаг переноса CY.

0	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---

RRC	0F
-----	----

Циклы: 1. Состояния: 4. Флаги: CY.

### RAL (СДВИГ ВЛЕВО ЧЕРЕЗ ПЕРЕНОС)

$(A_{n+1}) \leftarrow (A_n), (CY) \leftarrow (A_7), (A_0) \leftarrow (CY)$

Содержимое аккумулятора сдвигается на одну позицию влево через бит флага переноса CY. Младший бит устанавливается равным флагу переноса, а бит флага переноса CY – равным величине старшего бита аккумулятора. Команда воздействует только на флаг переноса CY.

0	0	0	1	0	1	1	1
---	---	---	---	---	---	---	---

RAL	17
-----	----

Циклы: 1. Состояния: 4. Флаги: CY.

### RAR (СДВИГ ВПРАВО ЧЕРЕЗ ПЕРЕНОС)

$$(A_n) \leftarrow (A_{n+1}), (CY) \leftarrow (A_0), (A_7) \leftarrow (CY)$$

Содержимое аккумулятора сдвигается на одну позицию вправо через бит флага переноса. Старший бит аккумулятора устанавливается равным флагу переноса, флаг переноса – равным младшему биту аккумулятора. Используется только флаг переноса CY.

0	0	0	1	1	1	1	1
---	---	---	---	---	---	---	---

RAR	1F
-----	----

Циклы: 1. Состояния: 4. Флаги: CY.

### CMA (ДОПОЛНЕНИЕ АККУМУЛЯТОРА)

$$(A) \leftarrow (\bar{A})$$

Содержимое аккумулятора инвертируется (бит, равный 1, становится равным 0; бит, равный 0, становится равным 1). Флаги не используются.

0	0	1	0	1	1	1	1
---	---	---	---	---	---	---	---

CMA	2F
-----	----

Циклы: 1. Состояния: 4. Флаги: нет.

СМС (ДОПОЛНЕНИЕ ФЛАГА ПЕРЕНОСА)

$СУ \leftarrow (СУ)$

Инвертируется бит флага переноса СУ. Другие флаги не используются.

0	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---

СМС	3F
-----	----

Циклы: 1. Состояния: 4. Флаги: СУ.

STC (УСТАНОВКА ФЛАГА ПЕРЕНОСА)

$(СУ) \leftarrow 1$

Бит флага переноса устанавливается в 1. Другие флаги не используются.

0	0	1	1	0	1	1	1
---	---	---	---	---	---	---	---

STC	37
-----	----

Циклы: 1. Состояния: 4. Флаги: СУ.

### *Команды перехода*

Команды этой группы изменяют нормальное последовательное исполнение программы. Исполнение команд не влияет на состояние флагов.

Имеется два типа команд перехода: безусловные и условные. Безусловные команды перехода осуществляют передачу управления, изменяя значение программного счетчика РС. Команды условного перехода сначала проверяют состояние одного из четырех флагов процессора, чтобы определить, выполнено ли указанное в команде условие. Имеются следующие условия перехода:

<i>Условие</i>	<i>Код</i>
NZ – не ноль (Z=0)	000
Z – ноль (Z=1)	001
NC – нет переноса (CY=0)	010
C – есть перенос (CY=1)	011
PO – нечетный результат (P=0)	100
PE – четный результат (P=1)	101
P – плюс (S=0)	110
M – минус (S=1)	111

JMP [адр.] (ПЕРЕХОД)

(PC) ← (байт 3) (байт 2)

Управление передается команде, адрес которой указан во втором и третьем байтах команды перехода.

1	1	0	0	0	0	1	1
Младший байт адреса							
Старший байт адреса							

JMP	[адр.]	C3
-----	--------	----

Циклы: 3. Состояния: 10. Адресация: непосредственная. Флаги: нет.

J (УСЛОВИЕ) [адр.] (УСЛОВНЫЙ ПЕРЕХОД)

Если (ССС), то(PC) ← (байт 3) (байт 2)

Если указанное условие истинно, то управление передается команде, адрес которой указан во втором и третьем байтах команды перехода.

Если условие ложное, то последовательный код программы не изменяется.

1	1	C	C	C	0	1	0
Младший байт адреса							
Старший байт адреса							

JNZ	[адр.]	C2	JC	[адр.]	DA	JP	[адр.]	F2
JZ	[адр.]	CA	JPO	[адр.]	E2	JM	[адр.]	FA
JNC	[адр.]	D2	JPE	[адр.]	EA			

Циклы: 3. Состояния: 10. Адресация: непосредственная. Флаги: нет.

CALL [адр.] (ВЫЗОВ)

$((SP) - 1) \leftarrow (PCH)$

$((SP) - 2) \leftarrow (PCL)$

$(SP) \leftarrow (SP) - 2$

$(PC) \leftarrow (\text{байт 3}) (\text{байт 2})$

Старшие 8 бит адреса следующей команды пересылаются в ячейку памяти, адрес которой на 1 меньше содержимого указателя стека SP. Младшие 8 бит адреса следующей команды пересылаются в ячейку памяти, адрес которой на 2 меньше величины указателя стека SP. Содержимое указателя стека уменьшается на 2. Управление передается команде, адрес которой указан во втором и третьем байтах команды вызова.

1	1	0	0	1	1	0	1
Младший байт адреса							
Старший байт адреса							

CALL	[адр.]	CD
------	--------	----

Циклы: 5. Состояния: 18. Адресация: непосредственная/косвенная регистровая. Флаги: нет.

(УСЛОВИЕ) (адр.) (УСЛОВНЫЙ ВЫЗОВ)

если (CCC), то  $((SP) - 1) \leftarrow (PCH)$ ,  $((SP) - 2) \leftarrow (PCL)$ ,  $(SP) \leftarrow (SP) - 2$ ,  $(PC) \leftarrow (\text{байт 3}) (\text{байт 2})$

Если указанное условие истинно, то выполняются действия, описанные в команде CALL, в противном случае последовательность исполнения команды не изменяется.

1	1	С	С	С	1	0	0
Младший байт адреса							
Старший байт адреса							

CNZ	[адр.]	C4	CC	[адр.]	DC	CP	[адр.]	F4
CZ	[адр.]	CC	CPO	[адр.]	E4	CM	[адр.]	FC
CNC	[адр.]	D4	CPE	[адр.]	EC			

Циклы: 5. Состояния: 18. Адресация: непосредственная/косвенная регистровая. Флаги: нет.

RET (ВОЗВРАТ)

$(PCL) \leftarrow ((SP))$

$(PCH) \leftarrow ((SP) + 1)$

$(SP) \leftarrow (SP) + 2$

Содержимое ячейки памяти, адрес которой содержится в указателе стека SP, пересылается в 8 младших бит программного счетчика PC. Содержимое ячейки памяти, адрес которой на 1 больше содержимого указателя стека, пересылается в 8 старших бит программного счетчика. Содержимое указателя стека увеличивается на 2.

1	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---

RET	C9
-----	----

Циклы: 3. Состояния: 10. Адресация: косвенная регистровая. Флаги: нет.

R условие (УСЛОВНЫЙ ВОЗВРАТ)

если (CCC), то  $(PCL) \leftarrow ((SP))$ ,  $(PCH) \leftarrow ((SP) + 1)$ ,  $(SP) \leftarrow ((SP) + 2)$

Если указанное условие истинно, то выполняются действия, описанные в команде RET, в противном случае последовательность исполнения команд не нарушается.

1	1	C	C	C	0	0	0
---	---	---	---	---	---	---	---

RNZ	C0	RC	D8	RP	F0
RZ	C8	RPO	E0	RM	F8
RNC	D0	RPE	E8		

Циклы: 3. Состояния: 12. Адресация: косвенная регистровая. Флаги: нет.

RST n (РЕСТАРТ)

$((SP) - 1) \leftarrow (PCH)$

$((SP) - 2) \leftarrow (PCL)$

$(SP) \leftarrow ((SP) - 2)$

$(PC) \leftarrow 8^x (NNN)$

Старшие 8 бит адреса следующей команды пересылаются в ячейку памяти, адрес которой на 1 меньше содержимого указателя стека. Младшие 8 бит адреса следующей команды пересылаются в ячейку памяти, адрес которой на 2 меньше содержимого указателя стека. Содержимое указателя стека уменьшается на 2. Управление передается команде, адрес которой равен коду NNN, умноженному на 8.

1	1	N	N	N	1	1	1
---	---	---	---	---	---	---	---

RST	0	C7	RST	3	DF	RST	6	F7
	1	CF		4	E7		7	FF
	2	D7		5	EF			

Циклы: 3. Состояния: 12. Адресация: косвенная регистровая. Флаги: нет.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	N	N	N	0	0	0

Программный счетчик после рестарта.

РСНL (ЗАГРУЗКА СОДЕРЖИМОГО РЕГИСТРОВ Н и L В ПРОГРАММНЫЙ СЧЕТЧИК)

$(РСН) \leftarrow (Н)$

$(РСL) \leftarrow (L)$

Содержимое регистра Н пересылается в 8 старших бит программного счетчика РС. Содержимое регистра L пересылается в 8 младших бит программного счетчика.

1	1	1	0	1	0	0	1
---	---	---	---	---	---	---	---

РСНL	E9
------	----

Циклы: 1. Состояния: 6. Адресация: регистровая. Флаги: нет.

### ***Команды ввода/вывода, управления, работы со стеком***

Команды этой группы выполняют операции ввода/вывода, изменяют содержимое стека и состояние флагов.

Регистр флагов

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	AC	X	P	X	CY

X – безразлично

PUSH gr (ЗАСЫЛКА В СТЕК СОДЕРЖИМОГО ПАРЫ РЕГИСТРОВ)

$((SP) - 1 \leftarrow (rh))$

$((SP) - 2 \leftarrow (rl))$

$(SP) \leftarrow (SP) - 2$

Содержимое старшего регистра пары регистров gr пересылается в ячейку памяти, адрес которой на 1 меньше содержимого указателя стека SP. Содержимое младшего регистра пары регистров gr пересылается в ячейку памяти, адрес которой на 2 меньше

содержимого указателя стека. Содержимое указателя стека уменьшается на 2.

Примечание. Пара регистров  $gr=SP$  не может быть указана в команде.

1	1	R	P	0	1	0	1
---	---	---	---	---	---	---	---

PUSH	B	C5	(заслать пару регистров B и C)
PUSH	D	D5	(заслать пару регистров D и E)
PUSH	H	E5	(заслать пару регистров H и L)

Циклы: 3. Состояния: 12. Адресация: косвенная регистровая. Флаги: нет.

PUSH PSW (ЗАСЛАТЬ В СТЕК СЛОВО СОСТОЯНИЯ ПРОГРАММЫ)

$((SP) - 1) \leftarrow (A)$

$((SP) - 2)_0 \leftarrow (CY), ((SP) - 2)_1 \leftarrow X$

$((SP) - 2)_2 \leftarrow (P), ((SP) - 2)_3 \leftarrow X$

$((SP) - 2)_4 \leftarrow (AC), ((SP) - 2)_5 \leftarrow X$

$((SP) - 2)_6 \leftarrow (Z), ((SP) - 2)_7 \leftarrow (S)$

$(SP) \leftarrow (SP) - 2$

X – безразлично

Содержимое аккумулятора пересылается в ячейку памяти, адрес которой на 1 меньше содержимого указателя стека SP. Содержимое флагов объединяется в слово состояния, и это слово пересылается в ячейку памяти, адрес которой на 2 меньше содержимого указателя стека. Содержимое указателя стека уменьшается на 2.

1	1	1	1	0	1	0	1
---	---	---	---	---	---	---	---

PUSH	PSW	F5
------	-----	----

Циклы: 3. Состояния: 12. Адресация: косвенная регистровая. Флаги: нет.

POP rp (СЧИТАТЬ ИЗ СТЕКА СОДЕРЖИМОЕ ПАРЫ РЕГИСТРОВ)

$(rl) \leftarrow ((SP))$

$(rh) \leftarrow ((SP) + 1)$

$(SP) \leftarrow (SP) + 2$

Содержимое ячейки памяти, адрес которой определяется содержимым указателя стека SP, пересылается в младший регистр пары регистров rp. Содержимое ячейки памяти, адрес которой на 1 больше содержимого указателя стека, пересылается в старший регистр пары регистров rp. Содержимое указателя стека увеличивается на 2.

Примечание. В команде не может быть указана пара регистров  $rp=SP$ .

1	1	R	P	0	0	1	
---	---	---	---	---	---	---	--

POP	B	C1	(считать из СТЕКА пару регистров B и C)
POP	D	D1	(считать из СТЕКА пару регистров D и E)
POP	H	E1	(считать из СТЕКА пару регистров H и L)

Циклы: 3. Состояния: 10. Адресация: косвенная регистровая. Флаги: нет.

POP PSW (СЧИТАТЬ ИЗ СТЕКА СЛОВО СОСТОЯНИЯ ПРОГРАММЫ)

$(CY) \leftarrow ((SP))_0$

$(P) \leftarrow ((SP))_2$

$(AC) \leftarrow ((SP))_4$

$(Z) \leftarrow ((SP))_6$

$(S) \leftarrow ((SP))_7$

$(A) \leftarrow ((SP) + 1)$

$(SP) \leftarrow (SP) + 2$

Содержимое ячейки памяти, адрес которой определяется содержимым указателя стека SP, используется для восстановления состояния флагов. Содержимое ячейки памяти, адрес которой на 1

больше содержимого указателя стека, переключается в аккумулятор. Содержимое указателя стека увеличивается на 2.

1	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---

POP	PSW	F1
-----	-----	----

Циклы: 3. Состояния: 10. Адресация: косвенная регистровая. Флаги: Z, S, P, CY, AC.

**XTHL (ОБМЕН СОДЕРЖИМОГО ВЕРШИНЫ СТЕКА И СОДЕРЖИМОГО РЕГИСТРОВ Н И L)**

(L) <-> ((SP))

(H) <-> ((SP) + 1)

Содержимое регистра L обменивается на содержимое ячейки памяти, адрес которой содержится в указателе стека SP. Содержимое регистра H обменивается на содержимое ячейки памяти, адрес которой на 1 больше содержимого указателя стека.

1	1	1	0	0	0	1	1
---	---	---	---	---	---	---	---

XTHL	E3
------	----

Циклы: 5. Состояния: 16. Адресация: косвенная регистровая. Флаги: нет.

**SPHL (ПЕРЕСЫЛКА СОДЕРЖИМОГО РЕГИСТРОВ Н И L В УКАЗАТЕЛЬ СТЕКА)**

(SP) <-> (H)(L)

Содержимое регистров H и L (16 бит) пересылается в указатель стека.

1	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---

SPHL	F9
------	----

Циклы: 1. Состояния: 6. Адресация: регистровая. Флаги: нет.

IN порт (ВВОД)

$/A/ \leftarrow (\text{Данные})$

Данные (1 байт), имеющиеся на восьми линиях двунаправленной шины данных указанного порта, пересылаются в аккумулятор.

1	1	0	1	1	0	1	1
Порт							

IN	порт	DB
----	------	----

Циклы: 3. Состояния: 10. Адресация: прямая. Флаги: нет.

OUT порт (ВЫВОД)

$(\text{Данные}) \leftarrow (A)$

Содержимое аккумулятора помещается на двунаправленную шину данных для передачи в указанный порт.

1	1	0	1	0	0	1	1
Порт							

OUT	порт	D3
-----	------	----

Циклы: 3. Состояния: 10. Адресация: прямая. Флаги: нет.

EI (РАЗРЕШЕНИЕ ПРЕРЫВАНИЙ)

Система прерываний разрешается при исполнении следующей команды.

1	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---

EI	FB
----	----

Циклы: 1. Состояния: 4. Флаги: нет.

Примечание. Прерывания не распознаются во время исполнения команды EI.

### DI (ЗАПРЕШЕНИЕ ПРЕРЫВАНИЙ)

Система прерываний запрещается непосредственно при выполнении следующей за DI командой.

1	1	1	1	0	0	1	1
---	---	---	---	---	---	---	---

DI	FB
----	----

Циклы: 1. Состояния: 4. Флаги: нет.

Примечание. Прерывания не распознаются во время исполнения команды.

### HLT (ОСТАНОВ)

В PC заносится адрес следующей команды. Процессор затем бездействует до прихода прерывания. Регистры и флаги не изменяют состояния.

0	1	1	1	0	1	1	0
---	---	---	---	---	---	---	---

HLT	76
-----	----

Циклы: 1. Состояния: 5. Флаги: нет.

### NOP (НЕТ ОПЕРАЦИИ)

В PC заносится адрес следующей команды, и микропроцессор переходит к ее обработке. Состояние флагов и регистров не изменяется.

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

NOP	00
-----	----

Циклы: 1. Состояния: 4. Флаги: нет.

*Практикум*

*ТАРАКАНОВ Алексей Валерьевич*

**Вычислительные машины, системы и сети**

**Часть 1**

Редакторы:

*Е.С. Захарова*

*И. А. Назарова*

Подписано в печать 25.11.2015г.

Формат 60x84 1/16. Бумага офсетная

Усл. п. л.3,4 Уч.-изд. л. 2,4

Тираж 100 экз. Рег. № 11/15sf

---

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Самарский государственный технический университет»  
443100, г. Самара, ул. Молодогвардейская, 244. Главный корпус

Отпечатано в типографии  
Самарского государственного технического университета  
Филиал в г. Сызрани, 446001, г. Сызрань, ул. Советская 45