

**МИНИСТРЕСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН**

**ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛ- ХОРАЗМИЙ**

ФАКУЛЬТЕТ ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

**Кафедра «Программное и аппаратное обеспечение
систем управления в телекоммуникациях»**

Бегматов Ш.А., Ахмедова Х.Х.

МЕТОДИЧЕСКОЕ УКАЗАНИЕ

к выполнению лабораторных работ по предмету

МИКРОПРОЦЕССОРЫ

Ташкент 2019

Авторы: Бегматов Ш.А., Ахмедова Х.Х.

Методическое указание к выполнению лабораторных работ по предмету «Микропроцессоры».- Ташкент: ТУИТ. 2019.-35стр.

В методическом указании рассматривается весь цикл процедуры создания программ для микропроцессоров, даны необходимые справочные данные, приведены примеры диалоговых процедур, даны задания и вопросы для выполнения компьютерных лабораторных работ и приведены пояснения, необходимые для выполнения лабораторных работ.

Методическое указание предназначено для бакалавров направления 5350100-Телекоммуникационные технологии («Телекоммуникации», «Телерадиовещание»), «Мобильные системы».

Решением научно-методического совета Ташкентского университета информационных технологий имени Мухаммада ал-Хоразмий методическое указание рекомендуется для публикации.(№__ протоколом «__» _____)

Ташкентский университет информационных технологий имени Мухаммада

Ал-Хоразмий,

2019

ВВЕДЕНИЕ

На сегодняшний день микропроцессоры прочно вошли в нашу жизнь и буквально с каждым днем открываются все новые и новые возможности их применения. Сейчас уже трудно назвать такие области техники или научных исследований, в которых бы не использовались микропроцессоры. По широте и эффективности применения микропроцессоров одно из первых мест занимает контрольно-измерительная техника. Все нарастающее распространение они получают в системах управления, технике связи, радиотехнике, электронике, медицинской диагностической и лечебной аппаратуре, сфере обслуживания и даже в детских игрушках. Микропроцессоры служат основой создания новых поколений электронных вычислительных машин. Для общего обозначения электронных средств и систем, предназначенных для обработки информации, вычислений и управления.

Широкое внедрение микропроцессорной техники в науку и производство – объективный и необходимый процесс. Для современной цивилизации характерна тенденция роста количества информации. Человек не способен уже воспринимать и обрабатывать эту информацию в реальном масштабе времени. В измерительной аппаратуре прием и обработку информации берут на себя электронные схемы и микропроцессоры. Поэтому знание микропроцессорной техники является необходимым.

ЛАБОРАТОРНАЯ РАБОТА №1

Тема: Создание и выполнение простейших программ

Цель работы:

Исследование команд микропроцессора КР580, составление программ на языке ассемблера.

Содержание работы:

В данной работе подробно рассматривается архитектура микропроцессора КР580ВМ8, на основе которого учащиеся знакомятся эмулятором МП КР580ВМ80А. Данный эмулятор позволяет: написать программ на языке ассемблера, используя систему команд МП КР580ВМ80А, их отладку и выполнение в тактовом, командном и сквозном режимах; изучить принципы и порядок выполнения команд; приобрести навыки работы с внешними устройствами МП-системы; получить представления об организации внешней и внутренней (регистровой) памяти и стековой области. Программа обладает дружелюбным и приятным интерфейсом, удобна в использовании и имеет функции сохранения, экспорта и печати данных. Всё это позволяет легко и удобно освоиться в ней, а также, получить все необходимые вышерассмотренные навыки.

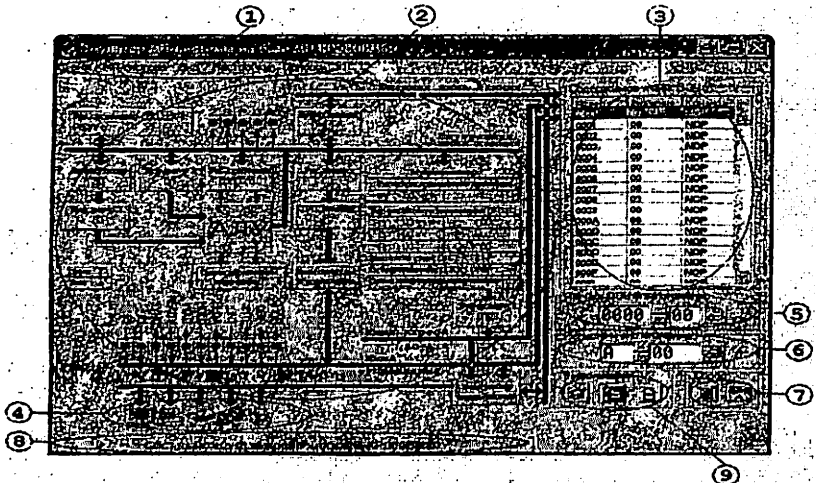


Рисунок 1. Программа-эмулятор микропроцессорной системы КР580

1. Главное меню программы;
2. Структурная схема МП-системы;
3. Таблица содержимого ОЗУ МП-системы;
4. Внешние периферийные устройства, подключенные к портам МП-системы;
5. Панель редактирования значения выбранной (текущей) ячейки ОЗУ МП-системы;
6. Панель редактирования значения содержимого выбранного регистра общего назначения МП-системы;
7. Группа кнопок «Сброс» для обнуления всех ячеек ОЗУ и регистров общего назначения МП-системы;
8. Панель системы команд МП КР580ВМ80А (скрытый вид);
9. Группа кнопок «Выполнение» для выполнения программы МП-системой в сквозном, командном и тактовом режимах.

Таблица 1.1

Команды пересылок

Мнемоника	Код	Число циклов ВМ80А	Число тактов		Содержание
			ВМ80А	ВМ85А	
MOV dst, srs	0b01DDDSSS	1	5	4	dst<- srs
MOV dst, M	0b01DDD110	2	7	7	dst<- (HL)
MOV M, srs	0b01110SSS	2	7	7	(HL) <- srs
MVI dst, data	0b00DDD110	2	7	7	dst<- data
MVI M, data	0x36	3	10	10	(HL) <- data

В таблице 1.1 приведены команды пересылок. Наряду с мнемоникой и кодом операции (первый байт команды) таблица содержит такие важнейшие для команды характеристики, как число обращений к системной магистрали и число периодов тактовой частоты, составляющих ее полный командный

цикл. В условных командах, рассматриваемых ниже, число обращений к шине и длительность МЦ в тактах зависят от выполнения условия. В соответствующих таблицах будут указаны два значения - невыполнения и выполнения условия.

Поля src и dst обозначают один из 8-разрядных регистров A, B, C, D, H, L, закодированных в соответствии со следующей таблицей:

Таблица 1.2

Коды регистров

Регистр	B	C	D	E	H	L	A
Код	000	001	010	011	100	101	111

В таблице 1.2 рассмотрены коды 8-разрядных регистров A, B, C, D, H, L. Рассмотренная группа команд содержит команды обмена между памятью и регистрами. Это наиболее часто встречающиеся в программах команды, занимающие около 45% процентов их общего числа.

Основу группы составляют следующие команды:

MOV, MVI- Перемещение

Таблица 1.3

Команды логической обработки

Мнемоника	Код	Число циклов BM80A	Число тактов		Флаги: CY, Z, M, P, C, AC	Содержание
			BM80A	BM85A		
ANA src	0b10100SSS	1	4	4	0+++0	A <- A &src
XRA src	0b10101SSS	1	4	4	0+++0	A <- A ^ src
ORA src	0b10110SSS	1	4	4	0+++0	A <- A src
CMP src	0b10111SSS	1	4	4	++++	A == src
ANA M	0xA6	2	7	7	0+++0	A <- A &(HL)
XRA M	0xAE	2	7	7	0+++0	A <- A ^ (HL)
ORA M	0xB6	2	7	7	0+++0	A <- A (HL)
CMP M	0xBE	2	7	7	++++	A == (HL)
ANI data	0xE6	2	7	7	0+++0	A <- A &data
XRI data	0xEE	2	7	7	0+++0	A <- A ^ data
ORI data	0xF6	2	7	7	0+++0	A <- A data
CPI data	0xFE	2	7	7	++++	A == data

В таблице 1.3 приведены команды логической обработки.

ANA, ANI Логическое И

XRA, XRI Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ

ORA, ORI Логическое ИЛИ

CMP, CPI Сравнение

Пример:

1. Дана функция $E = d + b \oplus c$.

$$d = 4h;$$

$$b = 3h;$$

$$c = 5h;$$

Адрес(hex)	Мнемоника	Код	Содержание
0000	MVI D, 4h	0x16	Перемещение 4 в D
		0x04	
0002	MVI B, 3h	0x06	Перемещение 3 в B
		0x03	
0004	MVI C, 5h	0x0E	Перемещение 5 в D
		0x05	
0006	MOV A, D	0x7A	Перемещение D в A
0007	ADD B	0x80	Сложение B в A
0008	XRA C	0xA9	Логическое исключающее или в A
0009	MOV E, A	0x5F	Перемещение A в E
000A	HLT	0x76	Конец программы

2) Дана функция $E = d - c | b \& 6$.

$$d = 10h;$$

$$c = 3h;$$

$$b = 2h;$$

Адрес (hex)	Мнемоника	Код	Содержание
0000	MVI D, 10h	0x16	Перемещение 10 в D
		0x10	
0004	MVI C, 3h	0x0E	Перемещение 3 в C
		0x03	
0006	MVI B, 2h	0x06	Перемещение 2 в B
		0x02	
0008	MOVA, D	0x7A	Перемещение D в A
0009	SUB C	0x91	Вычитание C из A
000A	ORA B	0xB0	Логическое Или в A
000B	ANI 6h	0x06	Логическое И в A
000D	MOV E, A	0x5F	Перемещение A в E
000E	HLT	0x76	Конец программы

Таблица 1.4

Варианты заданий

№	Задание	№	Задание
1.	$F = d - !b \oplus c + 05h$	16.	$F = b - !d c \oplus 06h$
2.	$F = b + d \oplus c \& 01h$	17.	$F = c - !b \oplus d 08h$
3.	$F = 45h - c - b d$	18.	$F = b \oplus c - d + 03h$
4.	$F = 50h + d - b \oplus c$	19.	$F = 40h - d + b \oplus c$
5.	$F = d - b \oplus c + 08h$	20.	$F = b - d c + 65h$
6.	$F = !b + d \oplus c - 1Fh$	21.	$F = c - b \oplus d + 06h$
7.	$F = 45h - !c - b d$	22.	$F = b \oplus c - d + 03h$
8.	$F = 50h + d - b \oplus c$	23.	$F = 40h - d + b \& c$
9.	$F = d - b \& c + 05h$	24.	$F = b - !d c + 06h$
10.	$F = b + d \oplus c - 01h$	25.	$F = c - b \oplus d \& 06h$
11.	$F = 45h - c - b d$	26.	$F = b \oplus c - !d + 03h$
12.	$F = 50h + !d - b \oplus c$	27.	$F = 40h - d + b \& c$
13.	$F = !b + d \& c 01h$	28.	$F = 65h \& c - b \oplus d$
14.	$F = 45h - c + b d$	29.	$F = b \& c - !d + 03h$
15.	$F = 55h + d - !b \& c$	30.	$F = 40h - d + b \oplus c$

Комментарии:

$$d=4+N$$

$$b=2+N$$

$$c=5$$

Здесь N, номер варианта.

Контрольные вопросы:

1. Формы представления чисел. Системы счисления и способы преобразования из одной системы в другую.
2. Формы представления чисел. Виды кодов двоичных чисел.
3. Операции сложения и вычитания в двоичной системе счисления.
4. Комбинационные логические схемы НЕ-, ИЛИ-, И-, исключающая ИЛИ. Их определения и таблицы истинности.
5. Комбинационные логические схемы И-НЕ, ИЛИ-НЕ, НЕ-, исключающая ИЛИ. Их определения и таблицы истинности.

ЛАБОРАТОРНАЯ РАБОТА №2

Тема: Выполнение простейших арифметико-логических операций над маленькими цифрами

Цель работы:

Ознакомление с регистрами МП КР580. Изучение обработки команд и перемещения данных.

Содержание работы:

Блок программно - доступных регистров МП КР580 отличается большой функциональной неоднородностью. Практически каждый регистр МП выполняет присущую только ему функцию, что сделано с целью более короткого кодирования системы команд микропроцессора. С другой стороны, это привело к усложнению программирования этого МП.

Восьмиразрядный аккумулятор А используется в подавляющем большинстве команд логической и арифметической обработки. Обычно он

адресуется неявно и служит как источником операнда, так и приемником результата. Благодаря этому в командах ВМ80А явно указывается только один операнд.

16-разрядный регистр HL, как правило, служит адресным регистром. При косвенной регистровой адресации он хранит 16-разрядный адрес основной памяти. В этом случае к нему ссылаются с помощью мнемоники М (Memory), например: MOV A, M; содержимое ячейки (HL) заносится в аккумулятор. В некоторых командах старший и младший байты 16-разрядного регистра HL могут адресоваться независимо и использоваться как отдельные 8-разрядные регистры данных H (HighByte) и L (LowByte) соответственно.

Регистры PC и SP выполняют свою обычную функцию счетчика команд и указателя стека. 8-разрядные регистры F, A вместе с 16-разрядными регистрами HL, SP и PC образуют стандартный регистровый набор микропроцессора с аккумулятором. Этот набор расширен четырьмя 8-разрядными регистрами общего назначения (РОН): B, C, D, E, которые в некоторых командах объединяются в 16-разрядные парные регистры BC и DE. Младшими регистрами пары являются соответственно регистры C и E. Введение РОН позволило создать достаточно эффективный микропроцессор с широкими функциональными возможностями.

Таблица 2.1

Команды пересылок

Мнемоника	Код (hex)	Число циклов ВМ80А	Число тактов		Содержание
			ВМ80А	ВМ85А	
LDA addr	3A	4	13	13	A ← (addr)
STA addr	32	4	13	13	(addr) ← A

В таблице 2.1 приведены команды пересылок

LDA Загрузка

STA Сохранение

Пример:

1) Дана функция $F = d + b \oplus c$

$d = 0x4000;$ $d = 4h;$

$b = 0x3000;$ $b = 3h;$

$c = 0x5000;$ $c = 5h;$

$F = 0x7000;$

Адрес(hex)	Мнемоника	Код	Содержание
0000	LDA 0x4000	0x3A	Загрузка 4 в A
		0x00	
		0x40	
0003	MOV D, A	0x57	Перемещение A в D
0004	LDA 0x3000	0x3A	Загрузка 3 в A
		0x00	
		0x30	
0007	MOV B, A	0x47	Перемещение A в B
0008	LDA 0x5000	0x3A	Загрузка 5 в A
		0x00	
		0x50	
000B	MOV C, A	0x4F	Перемещение A в C
000C	MOV A, D	0x7A	Перемещение D в A
000D	ADD B	0x80	Сложение B в A
000E	XRA C	0xA9	Логическое исключающее или в A
000F	STA 0x7000	0x32	Сохранение результата в ячейку 0x7000
		0x00	
		0x70	
0010	HLT	0x76	Конец программы

Варианты заданий

N	Задание	N	Задание
1	$F = \frac{b - d c \oplus 06h}{2}$	16	$F = \frac{d - b \oplus c + 05h}{4}$
2	$F = (c - b \oplus d 08h) * 2$	17	$F = (b + d \oplus c \& 01h) * 4$
3	$F = \frac{(b \oplus c)}{2} - d + 03h$	18	$F = 45h - (c - b d) * 4$
4	$F = 40h - \frac{(d + b \oplus c)}{2}$	19	$F = \frac{(50h + d)}{4} - b \oplus c$
5	$F = b - 2d c + 15h$	20	$F = d - (b \oplus c + 08h) * 2$
6	$F = 4c - b \oplus d + 06h$	21	$F = \frac{(b + d \oplus c)}{4} - 1Fh$
7	$F = b \oplus c - 2d + 03h$	22	$F = 45h - c - b d$
8	$F = 40h - d + b \& c$	23	$F = 50h + d - b \oplus c$
9	$F = b - d c * 2 + 26h$	24	$F = d - b \& c + 15h$
10	$F = 4c - b \oplus d \& 06h$	25	$F = b + (d \oplus c - 41h) * 4$
11	$F = \frac{(b \oplus c - d)}{2} + 03h$	26	$F = (45h - c) * 2 - b d$
12	$F = 40h - \frac{d + b}{2} \& c$	27	$F = \frac{53h + d}{2} - b \oplus c$
13	$F = 65h \& c - 4b \oplus d$	28	$F = 4b + d \& c 01h$
14	$F = (b \& c - d) * 2 + 03h$	29	$F = 45h - 4c + b d$
15	$F = \frac{40h - d}{4} + b \oplus c$	30	$F = 55h + 2d - b \& c$

Контрольные вопросы:

1. Регистры. Выполняемые функции и их классификации.
2. Регистры с параллельным приемом информации.
3. Сдвигающие регистры.
4. Регистры общего назначения.
5. Команды пересылки.

Флаг дополнительного переноса AC устанавливается при наличии переноса из третьего разряда аккумулятора в четвертый, иначе сбрасывается;

Флаг четности P устанавливается, если результат операции в аккумуляторе содержит четное число единиц, иначе сбрасывается;

Флаг переноса C устанавливается при наличии переноса (при сложении) или заема (при вычитании) из старшего разряда аккумулятора, иначе сбрасывается.

Важнейшими флагами являются:

- флаг C, позволяющий реализовать на 8-разрядном МП обработку данных длиной в произвольное число байт;
- флаг Z, используемый для организации циклов и ветвлений;
- флаг M, используемый для организации ветвлений по знаку результата.

Для организации ветвлений в составе команд микропроцессора предусмотрен обширный набор команд условной передачи управления, осуществляющих выбор одного из двух направлений перехода в зависимости от состояния проверяемого флага.

Таблица 3.1

Команды передачи управления

Мнемоника	Код(hex)	Число циклов VM80A	Флаги: CY, Z, M, P, C, AC	Содержание
PCHL	E9	1	----	PC <- HL
JMP addr	C3	3	----	PC <- addr
JC addr	DA	3	----	if (CY) PC <- addr
JNC addr	D2	3	----	if (!CY) PC <- addr
JZ/JNZ addr	CA/C2	3	----	if (Z / !Z) PC <- addr
JM/JP addr	FA/F2	3	----	if (M / !M) PC <- addr
JPE/JPO addr	EA/E2	3	----	if (P / !P) PC <- addr

В таблице 3.1 приведены команды передачи управления. Каждая операция обеспечивает проверку 8 условий, в соответствии с результатами которой меняются значения поля CC:

C	Carry	CY=1	M	Minus	M=1
NC	Not Carry	CY=0	P	Positive	M=0
Z	Zero	Z=1	PE	Parity Even	P=1
NZ	Not Zero	Z=0	PO	Parity Odd	P=0

Пример:

1). Дана функция $a=d+b \oplus c$ $F = \begin{pmatrix} a=0;255 \\ a>0 \cup a<0; a/2 \end{pmatrix}$ $d=4h$

$b=3h$

$F=0x0200$

Адрес (hex)	Метка	Мнемоника	Код	Содержание
0000		MVI D, 4h	0x16	Перемещение 4 в D
			0x04	
0002		MVI B, 3h	0x06	Перемещение 3 в B
			0x03	
0004		MVI C, 5h	0x0E	Перемещение 5 в C
			0x05	
0006		MOV A, D	0x7A	Перемещение D в A
0007		ADD B	0x80	Сложение B в A
0008		XRA C	0xA9	Логическое исключающее или в A
0009		JZ label	0xCA	Проверка Z флага, Z=1, то переход на метку "label" Z=0, то след.ком
			0x11	
			0x00	
000C		RRC	0x0F	Сдвиг в право
000D		STA 0x0200	0x32	Сохранение результата в ячейку 0x200
			0x00	
			0x02	
0010		HLT	0x76	Конец программы
0011	label:	MVI A, 0xFF	0x3E	Перемещение 255 в регистр A
			0xFF	
0013		STA 0x0200	0x32	Сохранение результата в ячейку 0x200
			0x00	
			0x02	
0016		HLT	0x76	Конец программы

1) Дана функция: $a = \overline{d - c | b}$

$F = 0x0200$

$$addr = \begin{cases} a > 0; 0x0300 \\ a < 0; 0x0400 \\ a = 0; 0x0500 \end{cases}$$

$d = 10h$

$c = 3h$

$b = 2h$

Адрес (hex)	Метка	Мнемоника	Код	Содержание
0000		MVI D, 10h	0x16	Перемещение 10 в D
			0x10	
0002		MVI C, 3h	0x0E	Перемещение 3 в C
			0x03	
0004		MVI B, 2h	0x06	Перемещение 2 в B
			0x02	
0006		MOVA, D	0x7A	Перемещение D в A
0007		SUB C	0x91	Сложение C в A
0008		ORA B	0xB0	Логическое исключающее или в A
0009		CMA	0x2F	Инверсия на 1
000A		JZ zero	0xCA	Проверка Z флага, Z=1, то переход на метку "zero" Z=0, то след.ком
			0x14	
			0x00	
000D		JP positive	0xF2	Проверка M флага, M=0, то переход на метку "positive" M=1, то след.ком
			0x18	
			0x00	
0010		STA 0x0400	0x32	Сохранение результата в ячейку 0x0400
			0x00	
			0x04	
0013		HLT	0x76	Конец программы
0014	zero:	STA 0x0500	0x32	Сохранение результата в ячейку 0x0500
			0x00	
			0x05	
0017		HLT	0x76	Конец программы
0018	positive:	STA 0x0300	0x32	Сохранение результата в ячейку 0x0300
			0x00	
			0x03	
001B		HLT	0x76	Конец программы

Варианты заданий

№	Задание	№	Задание
1.	$F = d - !b \oplus c + 05h$	16.	$F = b - !d c \oplus 06h$
2.	$F = b + d \oplus c \& 01h$	17.	$F = c - !b \oplus d 08h$
3.	$F = 45h - c - b d$	18.	$F = b \oplus c - d + 03h$
4.	$F = 50h + d - b \oplus c$	19.	$F = 40h - d + b \oplus c$
5.	$F = d - b \oplus c + 08h$	20.	$F = b - d c + 65h$
6.	$F = !b + d \oplus c - 1Fh$	21.	$F = c - b \oplus d + 06h$
7.	$F = 45h - !c - b d$	22.	$F = b \oplus c - d + 03h$
8.	$F = 50h + d - b \oplus c$	23.	$F = 40h - d + b \& c$
9.	$F = d - b \& c + 05h$	24.	$F = b - !d c + 06h$
10.	$F = b + d \oplus c - 01h$	25.	$F = c - b \oplus d \& 06h$
11.	$F = 45h - c - b d$	26.	$F = b \oplus c - !d + 03h$
12.	$F = 50h + !d - b \oplus c$	27.	$F = 40h - d + b \& c$
13.	$F = !b + d \& c 01h$	28.	$F = 65h \& c - b \oplus d$
14.	$F = 45h - c + b d$	29.	$F = b \& c - !d + 03h$
15.	$F = 55h + d - !b \& c$	30.	$F = 40h - d + b \oplus c$

Контрольные вопросы:

1. Изобразите структуру микропроцессора КР580 и назначение входных и выходных сигналов.
2. Регистр признаков в микропроцессоре КР580ВМ80.
3. Назовите назначение и виды памяти.
4. Команды передачи управления
5. Дешифраторы. Основные понятия и определения.

ЛАБОРАТОРНАЯ РАБОТА №4

Тема: Обработка данных массива

Цель работы:

Изучение данных массива с помощью циклов и создание их алгоритма.

Содержание работы:

Цикл – особая алгоритмическая структура, без которой не обойдется ни один язык программирования. Организовать циклическое выполнение части кода программы можно при помощи команд передачи управления.

Цикл со счетчиком можно организовать, используя те же условные команды, взяв в качестве счетчика любой свободный регистр. Но в микропроцессоре предусмотрены специальные команды для организации таких циклов: `loop` / `loopr` / `loopne`. Эти команды используют в качестве счетчика регистр `sx`, они сами уменьшают его значение после каждой итерации цикла и сравнивают после уменьшения с нулём (2.1).

Команда `loop` <метка перехода> выполняет следующие действия:

- уменьшает регистр `sx` ;
- сравнивает `sx` с нулем, если `sx > 0`, то управление передается на метку перехода (продолжаем цикл).

Команды `loopr`/`loopne` позволяют выйти из цикла по дополнительному условию.

Алгоритм — это последовательность шагов, ведущая к решению задачи

Блок-схема — это схематическое изображение, иллюстрирующее последовательность операторов, которые необходимо выполнить для достижения результата.

Блок-схемы широко используются при написании программ, так как они:

- Гораздо проще для понимания, чем запись в виде команд.
- Упрощают процесс отладки.
- Позволяют составить эффективную программную документацию.
- Облегчают процесс демонстрации и обсуждения программы.

Блоки, из которых составляются блок-схемы, представлены на Рисунке 2



Рисунок. 2 Составляющие блок-схемы

Рассмотрим работу с блок-схемами на примере.

Таблица 4.1

Команды пересылки

Мнемоника	Код (hex)	Число циклов BM80A	Число тактов		Содержание
			BM80A	BM85A	
LDAX B	0A	2	7	7	A <- (BC)
LDAX D	1A	2	7	7	A <- (DE)
STAX B	02	2	7	7	(BC) <- A
STAX D	12	2	7	7	(DE) <- A
LXI B, data16	01	3	10	10	BC <- data16

LXI D, data16	11	3	10	10	DE <- data16
LXI H, data16	21	3	10	10	HL <- data16
LXI SP, data16	31	3	10	10	SP <- data16

В таблице 4.1 приведены команды пересылок загрузки и сохранения значений парных регистров.

LDAX - Загрузка аккумулятора косвенной адресацией через регистровые пары BC и DE

LXI - непосредственная адресация 16- разрядного регистра константой

STAX Сохранение значений парных регистров в аккумулятор

Логические команды включают также и подгруппу сдвигов вправо (RRC, RAR) и влево (RLC, RAL). Определены операции циклического (RRC, RLC) и расширенного (RAR, RAL) сдвигов.

Пример:

1) Дано: вычислить сумму 10 элементов начиная с 0x1000 адреса. Полученный результат сохранить в 0x2000 адресе.

$$F = \sum_{i=0}^9 a_i - ?$$

$$F = 0x2000$$

Адрес (hex)	Метка	Мнемоника	Код	Содержание
		MVI C, 10h		Перемещение 10h в C
		LXI D, 0x1000		Загрузка значения 0x1000 в DE
	loop:	LDAX D		Загрузка значения DE в A
		ADD B		Сложение B в A
		MOV B, A		Перемещение A в B
		INX D		Инкрементировать DE
		DCR C		Декрементировать C
		JNZ loop		Проверка Z флага,

			Z=0, то переход на метку loop Z=1, то след.ком
		STA 0x2000	Сохранение результата в ячейку 0x2000
		HLT	Конец программы

Таблица 4.2

Варианты заданий

N	Задание	N	Задание
1	$F = \frac{b - d c \oplus 06h}{2}$	16	$F = \frac{d - b \oplus c + 05h}{4}$
2	$F = (c - b \oplus d 08h) * 2$	17	$F = (b + d \oplus c \& 01h) * 4$
3	$F = \frac{(b \oplus c)}{2} - d + 03h$	18	$F = 45h - (c - b d) * 4$
4	$F = 40h - \frac{(d + b \oplus c)}{2}$	19	$F = \frac{(50h + d)}{4} - b \oplus c$
5	$F = b - 2d c + 15h$	20	$F = d - (b \oplus c + 08h) * 2$
6	$F = 4c - b \oplus d + 06h$	21	$F = \frac{(b + d \oplus c)}{4} - 1Fh$
7	$F = b \oplus c - 2d + 03h$	22	$F = 45h - c - b d$
8	$F = 40h - d + b \& c$	23	$F = 50h + d - b \oplus c$
9	$F = b - d c * 2 + 26h$	24	$F = d - b \& c + 15h$
10	$F = 4c - b \oplus d \& 06h$	25	$F = b + (d \oplus c - 41h) * 4$
11	$F = \frac{(b \oplus c - d)}{2} + 03h$	26	$F = (45h - c) * 2 - b d$
12	$F = 40h - \frac{d + b}{2} \& c$	27	$F = \frac{53h + d}{2} - b \oplus c$
13	$F = 65h \& c - 4b \oplus d$	28	$F = 4b + d \& c 01h$
14	$F = (b \& c - d) * 2 + 03h$	29	$F = 45h - 4c + b d$
15	$F = \frac{40h - d}{4} + b \oplus c$	30	$F = 55h + 2d - b \& c$

Контрольные вопросы:

1. Счетчики. Основные определения, классификация.
2. Назовите назначение сигналов устройства управления МП 580.
3. Объясните взаимодействие памяти с микропроцессором.
4. Команды условного и не условного перехода
5. Составляющие блок схемы

ЛАБОРАТОРНАЯ РАБОТА №5

Тема: Стеки и подпрограммы

Цель работы:

Изучение управления потоками. Стеки и подпрограммы.

Содержание работы:

Стек-это своего рода механизм, который позволяет временное хранение данных по принципу LIFO (Last Input First Output).

Заполнение стека выполняется командой PUSH R-введение содержимого РОН в стек (R -> STACK). Извлечение данных командой POP R (R <- STACK).

Также стек является механизмом, без которого не возможны были бы подпрограммы, т.к. механизм стека позволяет временное сохранение счѣтчика программ (PC), во время выполнения подпрограммы. Соответственно, команды типа CALL и RET также вставляют и извлекают данные из стека.

Механизм стека реализован на базе памяти RAM и указательного регистра стека SP (Stack Pointer).

SP-показывает текущий адрес для записи в RAM.

SP является двойным регистром (16-бит), построенный на [SPH:SPL].

Выполняя команду PUSH R содержимое регистра запишется в текущий адрес SP, после которого SP декрементируется:

- PUSH R
- 1. RAM[SP] = R

2. $SP = SP - 1$

Для команды POP R выполняется инкрементирование SP, после чего в POH записывается значение текущего адреса:

- POP R
- 1. $SP = SP + 1$
- 2. $R = RAM[SP]$

Добавление элементов в стек будет включать в себя увеличение стека. Исходя из изложенного, мы убеждаемся, что стек растет в направлении уменьшения адресов RAM.

Подпрограмма

Подпрограмма-последовательность команд сгруппированных по метке и команд -возврата из подпрограммы RET. Метка, представляет собой адрес в памяти программ, с присвоенным именем доступа.

MULL:... ; Имя подпрограммы представлено меткой начала подпрограммы

...
... ; Тело подпрограммы
...
ret ; возврат из подпрограммы

Механизм вызова подпрограммы состоит в сохранения текущего адреса исполнения в стеке для того, чтобы иметь возможность восстановить выполнение следующей команды после возврата из подпрограммы.

Команда вызова подпрограммы CALL.

При вызове подпрограммы происходит сохранение в стеке счётчика программы и восстановление его после возврата из подпрограммы.

1. $PC \rightarrow STACK$ - сохранение текущего адреса в стеке
2. `jmp MULL` - переход по адресу по метке MULL

Возврат из подпрограммы происходит по команде RET , как правило находящейся в конце подпрограммы . Например:

1. $PC \leftarrow STACK$ - возврат адреса из стека

2. $PC \leftarrow PC + 1$ - переход к следующей команде после возврата

Не забывая, что счётчик программ PC - это 16-битовый регистр, он будет занимать 2 места в стеке, выполняя 2 операции для сохранения в стеке.

Таблица 5.1

Команды пересылок

Мнемоника	Код (hex)	Число циклов BM80A	Число тактов		Флаги: CY, Z, M, P, C, AC	Содержание
			BM80A	BM85A		
PUSH B/D/H	C5/D5/E5	3	11	12	----	(SP) <- BC/DE/HL
PUSH PSW	F5	3	11	12	----	(SP) <- PSW
POP B/D/H	C1/D1/E1	3	10	10	----	BC/DE/HL <- (SP)+
POP PSW	F1	3	10	10	++++	PSW <- (SP)+
CALL addr	CD	3	10	7/10	----	-(SP) <- PC <- addr
RET	C9	3	11	10	----	PC <- (SP)+

В таблице 5.1 приведены команды пересылок для передачи параметров через стек.

Пример:

1) Дано:

$$F = \overset{1}{((a \times b) \& c)} \overset{2}{\times} \overset{3}{a} \overset{4}{+} \overset{5}{b} \times c$$

$a = 0x0200(D)$

$b = 0x0210(B)$

$c = 0x021F(C)$

$F = 0x030A$

Адрес (hex)	Метка	Мнемоника	Код	Содержание
0000		LDA 0x0200		Загрузка D в A
0001		MOV D, A		Перемещение A в D
0002		LDA 0x0210		Загрузка B в A
0003		MOV B, A		Перемещение A в B
0004		LDA 0x21F		Перемещение C в A
0005		MOV C, A		Перемещение A в C

Адрес (hex)	Метка	Мнемоника	Код	Содержание
0006		PUSH B		Загрузка BC в стек
0007		MOV C,B		Перемещение B в C
0008		MOV B,D		Перемещение D в B
0009		CALL MULL		Вызов подпрограммы
0010		POP B		Выгрузка BC из стека
0011		ANA C		
0012		PUSH B		Загрузка BC в стек
0013		MOV B,A		Перемещение A в B
0014		MOV C,D		Перемещение D в C
0015		CALL MULL		Вызов подпрограммы
0016		POP B		Выгрузка BC из стека
0017		STA 0x30A		Сохранение результата в ячейку 0x30A
0018		HLT		Конец программы
0019	MULL:	MVI A, 0x00		Подпрограмма: Перемещение D в A
0020	M1:	ADD B		Сложение B в A
0021		DCR C		Декрементирование на 1 C
0022		JNZ M1		Проверка Z флага, Z=0, то переход на метку M1 Z=1, то след.ком
0023		RET		Возврат из подпрограммы

Таблица 5.2

Варианты заданий

№	Задание	№	Задание
1.	$F = d - !b \oplus c + 05h$	16.	$F = b - !d c \oplus 06h$
2.	$F = b + d \oplus c + 01h$	17.	$F = c - !b \oplus d 08h$
3.	$F = 45h - c - b d$	18.	$F = b \oplus c - d + 03h$
4.	$F = 50h + d - b \oplus c$	19.	$F = 40h - d + b \oplus c$
5.	$F = d - b \oplus c + 08h$	20.	$F = b - d c + 65h$
6.	$F = !b + d \oplus c - 1Fh$	21.	$F = c - b \oplus d + 06h$
7.	$F = 45h - !c - b d$	22.	$F = b \oplus c - d + 03h$

№	Задание	№	Задание
8.	$F = 50h + d - b \oplus c$	23.	$F = 40h - d + b \& c$
9.	$F = d - b \& c + 05h$	24.	$F = b - !d c + 06h$
10.	$F = b + d \oplus c - 01h$	25.	$F = c - b \oplus d \& 06h$
11.	$F = 45h - c - b d$	26.	$F = b \oplus c - !d + 03h$
12.	$F = 50h + !d - b \oplus c$	27.	$F = 40h - d + b \& c$
13.	$F = !b + d \& c 01h$	28.	$F = 65h \& c - b \oplus d$
14.	$F = 45h - c + b d$	29.	$F = b \& c - !d + 03h$
15.	$F = 55h + d - !b \& c$	30.	$F = 40h - d + b \oplus c$

Контрольные вопросы:

1. Поясните способы адресации в системе команд микропроцессора КР580
2. Приведите структурные схемы генератора тактовых сигналов КР580
3. Механизм стека
4. Команда вызова подпрограммы и ее функция
5. Приведите команды пересылок для передачи параметров через стек.

ЛАБОРАТОРНАЯ РАБОТА №6

Тема: Образование команд ввода и вывода

Цель работы:

Работа с внешними устройствами эмулятора. Изучение команд ввода и вывода

Содержание работы:

Работа со всеми внешними устройствами эмулятора заключается в отправке или приёме на (с) соответствующий(его) устройству порт(а) МП-системы значения из (в) регистра-аккумулятора. Это осуществляется путём выполнения на эмуляторе команд ввода-вывода, таких как IN (принять из порта) и OUT (вывести в порт).

Работа с монитором

"Монитор КР580" поддерживает отдельную систему команд, обеспечивающую вывод графической или текстовой информации. Графический режим соответствует разрешению 256x256 пикселей и глубине цвета - 128 бит на пиксель, а текстовый - 39x20 символов и глубине цвета 128 бит на символ. Одновременно монитор поддерживает два этих режима, т.е. может содержать и текст и графику.

Команды посылаются в порт 00h побайтно. Различаются 3-х байтные и 2-х байтные команды:

2-х байтная.

1-ый байт: 1-ый бит - 0-текст, 1-графика; остальные 7 бит на цвет, согласно формуле: $FFFFFFh(RGB) / 127 * \text{эти_7_бит}$.

2-ой байт: номер символа в кодовой таблице OEM/DOS.

3-х байтная.

1-ый байт: 1-ый бит - 0-текст, 1-графика; остальные 7 бит на цвет, согласно формуле: $FFFFFFh(RGB) / 127 * \text{эти_7_бит}$.

2 байт: координата по X.

3 байт: координата по Y.

Работа с дисководом

В порт дисковода КР580 (01h) посылаются или читаются значения, что приводит к их одновременному сохранению (чтению) в (из) файл(а) реальной машины, располагающийся на дискете.

Работа с жёстким диском (порт 02h)

Аналогично работе с дисководом, только связано с жёстким диском реальной машины.

Работа с сетевым адаптером

Отправленные значения в порт 03h пересылаются по сети реальным машин по протоколу TCP/IP на IP-адрес, указанный в настройках.

Работа с принтером

Отправленные значения в порт 04h временно хранятся в буфере до тех пор, пока пользователь не отправит их на печать самостоятельно. Печатаемые символы на принтере реальной машины соответствуют кодировке OEM/DOS.

Для удобства работы с рассматриваемым эмулятором предусмотрены следующие возможности работы с данными:

- Загрузка и сохранение образов содержимого ОЗУ и РОИ;
- Частичная загрузка и сохранение программ эмулятора (подпрограмм);
- Экспорт содержимого ОЗУ и РОИ эмулятора в MS Excel;
- Экспорт содержимого ОЗУ и РОИ эмулятора в MS Word;
- Экспорт содержимого ОЗУ и РОИ эмулятора в текстовый файл;
- Печать содержимого ОЗУ и РОИ эмулятора.

OUT - записать данные из регистра в порт I/O

Таблица 6.1

Команды управления процессором

Мнемоника	Код	Число циклов ВМ80А	Число тактов		Флаги: CY, Z, M, P, C, AC	Содержание
			ВМ80А	ВМ85А		
IN port	DB	3	10	10	----	A <- IOSEG (port)
OUT port	D3	3	10	10	----	IOSEG (port) <- A

В таблице 6.1 приведены команды управления процессором.

Пример:

1) Разработать программу вывода слова HELLO на дисплей фиолетовым цветом

Адрес (hex)	Метка	Мнемоника	Код	Содержание
0000	label:	LXI B, 0x3000	0x01	Инкрементирование
			0x00	
			0x30	
0003		MVI L, 0x05	0x2E	Перемещение значения в 0x05 в L
			0x05	

Адрес (hex)	Метка	Мнемоника	Код	Содержание
0005		MVI A, 0x4A	0x3E	Перемещение значения в 0x4A в A
			0x4A	
0007		OUT 0x00	0xD3	Вывод цвета из аккумулятора в монитор
			0x00	
0009		LDAX B	0x0A	Загрузка BC в A
000A		OUT 0x00	0xD3	Вывод кода символа в монитор
			0x00	
000C		DCR L	0x2D	Декрементирование L
000D		INX B	0x03	Инкрементирование BC
000E		JNZ	0xC2	Проверка Z флага, Z=0, то переход на метку M1 Z=1, то след.ком
			0x05	
			0x00	
0011		HLT	0x76	Конец программы.
....				
3000		-	0x48	H
3001			0x45	E
3002		-	0x4C	L
3003			0x4C	L
3004		-	0x4F	O

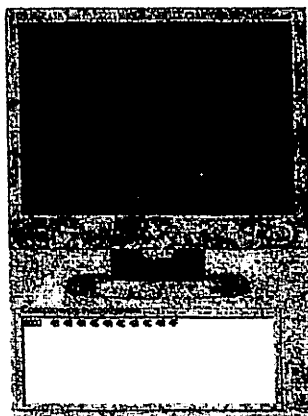


Рисунок 3. Результат программы

Пример:

2) Разработать программу вывода на экран графического интерфейса пользователя

Адрес (hex)	Метка	Мнемоника	Код	Содержание
0000	label:	INR C	0x0C	Инкрементирование
0001		MOV A,C	0x79	Перемещение C в A
0002		ORI 80h	0xF6	Логическое ИЛИ в A
			0x80	
0004		OUT 0x00	0xD3	Вывод цвета из аккумулятора в монитор
			0x00	
0006		MOV A, H	0x7C	Перемещение C в A
0007		OUT 0x00	0xD3	Вывод кода символа в монитор
			0x00	
0009		MOV A, L	0x7D	Перемещение C в A
000A		OUT 0x00	0xD3	
			0x00	
000C		INR H	0x24	Инкрементирование H
000D		JNC label	0xD2	Проверка C флага, C=0, то переход на метку label C=1, то след.ком
			0x00	
			0x00	
0010		INR L	0x2C	Инкрементирование H
0011		JNC label	0xD2	Проверка C флага, C=0, то переход на метку label C=1, то след.ком
			0x00	
			0x00	
0014		HLT	0x76	Конец программы

00:	0	20:	spa	40:	0	60:	0	80:	А	А0:	а	С0:	0	Е0:	р
01:	1	21:	!	41:	А	61:	а	81:	Б	А1:	б	С1:	1	Е1:	с
02:	2	22:	#	42:	В	62:	б	82:	В	А2:	в	С2:	2	Е2:	у
03:	3	23:	\$	43:	С	63:	с	83:	Г	А3:	г	С3:	3	Е3:	ф
04:	4	24:	%	44:	Д	64:	д	84:	Д	А4:	д	С4:	4	Е4:	х
05:	5	25:	&	45:	Е	65:	е	85:	Е	А5:	е	С5:	5	Е5:	ц
06:	6	26:	'	46:	Ф	66:	ф	86:	Ж	А6:	ж	С6:	6	Е6:	ч
07:	7	27:	?	47:	Г	67:	г	87:	З	А7:	з	С7:	7	Е7:	ш
08:	8	28:	[48:	Н	68:	н	88:	И	А8:	и	С8:	8	Е8:	щ
09:	9	29:]	49:	И	69:	и	89:	Я	А9:	я	С9:	9	Е9:	ъ
0A:	A	2A:	*	4A:	Ј	6A:	ј	8A:	К	АA:	к	СA:	A	ЕA:	ы
0B:	B	2B:	+	4B:	К	6B:	к	8B:	П	АB:	п	СB:	B	ЕB:	ь
0C:	C	2C:	,	4C:	Л	6C:	л	8C:	М	АC:	м	СC:	C	ЕC:	э
0D:	D	2D:	-	4D:	М	6D:	м	8D:	О	АD:	о	СD:	D	ЕD:	я
0E:	E	2E:	/	4E:	О	6E:	о	8E:	П	АE:	п	СE:	E	ЕE:	а
0F:	F	2F:	0	4F:	О	6F:	о	8F:	Р	АF:	р	СF:	F	ЕF:	е
10:	10	30:	1	50:	Р	70:	р	90:	С	Б0:	с	Д0:	10	Г0:	и
11:	11	31:	2	51:	Q	71:	q	91:	С	Б1:	с	Д1:	11	Г1:	й
12:	12	32:	3	52:	R	72:	r	92:	Т	Б2:	т	Д2:	12	Г2:	о
13:	13	33:	4	53:	S	73:	s	93:	У	Б3:	у	Д3:	13	Г3:	п
14:	14	34:	5	54:	T	74:	t	94:	Ф	Б4:	ф	Д4:	14	Г4:	қ
15:	15	35:	6	55:	U	75:	u	95:	Х	Б5:	х	Д5:	15	Г5:	ғ
16:	16	36:	7	56:	V	76:	v	96:	Ц	Б6:	ц	Д6:	16	Г6:	х
17:	17	37:	8	57:	W	77:	w	97:	Ч	Б7:	ч	Д7:	17	Г7:	д
18:	18	38:	9	58:	X	78:	x	98:	Ш	Б8:	ш	Д8:	18	Г8:	т
19:	19	39:	:	59:	Y	79:	y	99:	Ъ	Б9:	ъ	Д9:	19	Г9:	н
1A:	1A	3A:	;	5A:	Z	7A:	z	9A:	Ы	БA:	ы	ДА:	1A	ГA:	в
1B:	1B	3B:	<	5B:	[7B:	{	9B:	Ь	БB:	ь	DB:	1B	ГB:	а
1C:	1C	3C:	>	5C:	\	7C:		9C:	Ъ	БC:	ъ	DC:	1C	ГC:	е
1D:	1D	3D:	>	5D:]	7D:	}	9D:	Э	БD:	э	DD:	1D	ГD:	я
1E:	1E	3E:	>	5E:	^	7E:	~	9E:	Ю	БE:	ю	DE:	1E	ГE:	т
1F:	1F	3F:	?	5F:	_	7F:	а	9F:	Я	БF:	я	DF:	1F	ГF:	г

Рисунок 4. ASCII код

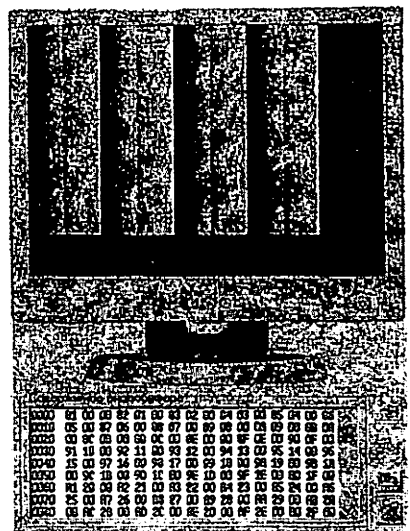


Рисунок 5. Результат программы

Контрольные вопросы:

1. Перечислите основные устройства ввода вывода
2. Сформулируйте методы адресации портов ввода-вывода
3. Перечислите и поясните основные способы ввода-вывода
4. Дайте определение понятиям внешних и внутренних функций порта
5. Поясните способы аппаратной и программной реализации портов безусловного ввода-вывода
6. Поясните способы аппаратной и программной реализации портов условного ввода-вывода
7. Перечислите и кратко поясните особенности ввода информации с кнопок и клавиатуры

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. Лабораторная работа №1. Создание и выполнение простейших программ.....	4
2. Лабораторная работа №2. Выполнение простейших арифметико-логических операций над маленькими цифрами.....	9
3. Лабораторная работа №3. Изучение условных и не условных переходов.....	13
4. Лабораторная работа №4. Обработка данных массива.....	18
5. Лабораторная работа №5. Стеки и подпрограммы.....	22
6. Лабораторная работа №6. Образование команд ввода и вывода.....	26
СПИСОК ЛИТЕРАТУРЫ.....	34

Список литературы

1. O‘zbekiston Respublikasi Prezidentining “O‘zbekiston Respublikasini yanada rivojlantirish bo‘yicha harakatlar strategiyasi to‘g‘risida”gi farmoni. (2017-yil 7-fevral, PF-4947-son).
2. O‘zbekiston Respublikasi Prezidentining “Oliy ta‘lim tizimini yanada rivojlantirish chora-tadbirlari to‘g‘risida”gi qarori (2017 yil 20 aprel, PQ-2909-son).
3. O‘zbekiston Respublikasi Prezidentining “Oliy ma‘lumotli mutaxassislar tayyorlash sifatini oshirishda iqtisodiyot sohalari va tarmoqlarining ishtirokini yanada kengaytirish chora-tadbirlari to‘g‘risida»gi qarori (2017 yil 27 iyul, PQ-3151-son).
4. Новиков, Ю.В. Основы микропроцессорной техники / Ю.В. Новиков, П.К. Скоробогатов. - М.: Бином. Лаборатория знаний, Костров, Б.
5. Архитектура микропроцессорных систем / Б. Костров. - М.: Диалог-МИФИ, 2009. -
6. Кузин, А.В. Микропроцессорная техника: Учебник для студ. сред. проф. образования / А.В. Кузин, М.А. Жаворонков. - М.: ИЦ Академия, 2013.
7. O‘quv adabiyotlarni ishlab chiqish va nashr etishga tayyorlash bo‘yicha uslubiy ko‘rsatmalar. Agzamov F.S., Ergashev A.Q., Tulyaganov A.A., Gulturaev N.X. - Toshkent: Muhammad al-Xorazmiy nomidagi TATU. 2018. - 52 b.

Методическое указание разработанное для лабораторных работ по предмету «Микропроцессоры» предназначено для бакалавров направления 5350100-Телекоммуникационные технологии («Телекоммуникации», «Телерадиовещание», «Мобильные системы»).

Методическое указание обсуждено на заседании кафедры «АПОСУТ» № ___ от _____ марта 2019 года и рекомендовано для изучения в научно-методическом совете факультета.

Методическое указание обсуждено на совещании факультета «Телекоммуникационные технологии» №___ от «___»_____ и рекомендовано для изучения в научно-методическом совете университета.

Методическое указание обсуждено на совещании научно-методического совета ТУИТ №_ от «___»_____ и рекомендовано для печати.

Составили: Бегматов Ш.А.
Ахмедова Х.Х.

Рецензент: Балтаев Ж.Б.

Ответственный редактор: Парсиев С.С.

Корректор: Рахимова Н.Х.

Формат 60x84 1/16. Печ.лист 2,25.
Заказ № 252. Тираж 20.
Отпечатано в «Редакционно издательском»
отделе при ТУИТ.
Ташкент ул. Амир Темур, 108.