



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий
Кафедра вычислительной техники

Отчет по лабораторной работе №1
по дисциплине
«Архитектура процессоров и микропроцессоров»

Выполнил: студент группы ИВБО-02-19

А. М. Сосунов

Принял: старший преподаватель кафедры ВТ

Ю. М. Скрыбин

Работа выполнена «____» _____ 202__

«Зачтено» «____» _____ 202__

Москва 2021

Цель работы

Целью работы является исследование работы процессора при выполнении команд пересылок и арифметических операций.

Описание работы

В ходе данной лабораторной работы нам были предложены две программы, которые необходимо было занести в память эмулируемого процессора КР580ВМ80. В таблице 1 данные программы приведены побайтово, на рисунках А.1 и А.2 приведены состояния регистров и флагов во время выполнения данных программ.

Таблица 1 — Программы к выполнению

1)	3A	48	77	1	D5	B8	7	25	2D	89	86	3F
2)	B2	C3	E9	CD	C9	76	0	DC				

Ход работы

1. Войти в «окно», «структурная схема микропроцессора»;
2. Установить потактный режим работы, указав «мышкой» кнопку «Тк»;
3. Установить режим «ОЗУ» указав «мышкой» клавишу «ОЗУ», набрать на цифровой клавиатуре адрес ОЗУ, затем ввести код требуемой команды согласно варианту задания и «нажать» клавишу «Ввод». Если данная команда требует участия регистров или регистровых пар, необходимо записать в указанные регистры информацию следующим образом: войти в режим работы с регистровой памятью, указав клавишу «Рег.», ввести в него информацию. Любая процедура записи в ОЗУ или в регистры завершается «нажатием» клавиши «Ввод».
4. Установить адрес ячейки ОЗУ, по которому записана команда.

5. Нажимая «мышкой» клавишу «Вып.», выполнить команду в потактном режиме, фиксируя в соответствующей таблице, состояние процессора в каждом такте каждого цикла. Структурная схема позволяет проследить в этом режиме последовательность всех процедур выполнения команды.

Вывод: в ходе данной лабораторной работы мы ознакомились со структурной схемой процессора КР580ВМ80, регистрами общего назначения данного процессора КР580ВМ80, научились заносить программу в память процессора, исследовали работу процессора при выполнении данных программ.

ПРИЛОЖЕНИЕ А

Таблица состояний процессора																							
Команда	N цикла	N такта	PK	СК	УС	Бух. адр. данных	W	Z	A	B	C	D	E	H	L	Состояние флагов							Оценка правильности выполнения командой своих функций
																Z	S	P	C	AC			
	0	0	00	0000	FFFF	0000	00	00	00	00	00	00	00	00	00	00	0	0	0	0	0		
	1	1	00	0000	FFFF	0000	00	00	00	00	00	00	00	00	00	00	0	0	0	0	0		
	2	2	00	0000	FFFF	0000	3A	00	00	00	00	00	00	00	00	00	0	0	0	0	0		
	3	3	00	0000	FFFF	0000	3A	00	00	00	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	1	2	3A	0000	FFFF	0000	3A	00	00	00	00	00	00	00	00	00	0	0	0	0	0	LDAR #B - записать в аккумулятор данные по адресу 774B	
LDA #B	1	4	3A	0001	FFFF	0000	3A	00	00	00	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	2	5	3A	0001	FFFF	0001	3A	00	00	00	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	2	6	3A	0001	FFFF	0001	4B	00	00	00	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	2	7	3A	0002	FFFF	0001	4B	00	00	00	00	00	00	00	00	00	0	0	0	0	0	LDAR #B - записать в аккумулятор данные по адресу 774B	
LDA #B	3	8	3A	0002	FFFF	0002	4B	00	00	00	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	3	9	3A	0002	FFFF	0002	77	77	4B	00	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	4	11	3A	0003	FFFF	774B	77	77	4B	00	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	4	12	3A	0003	FFFF	774B	01	77	4B	01	00	00	00	00	00	00	0	0	0	0	0	LDAR #B - записать в аккумулятор данные по адресу 774B	
LDA #B	4	13	3A	0003	FFFF	774B	01	77	4B	01	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	1	1	3A	0003	FFFF	0003	01	77	4B	01	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	1	4	01	0004	FFFF	0003	01	77	4B	01	00	00	00	00	00	00	0	0	0	0	0	LDAR #B - записать в аккумулятор данные по адресу 774B	
LDA #B	2	5	01	0004	FFFF	0004	01	77	4B	01	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	2	6	01	0004	FFFF	0004	30	77	4B	01	00	00	00	00	00	00	0	0	0	0	0		
LDA #B	2	7	01	0005	FFFF	0005	50	77	4B	01	00	30	00	00	00	00	0	0	0	0	0		
LDA #B	2	8	01	0005	FFFF	0005	50	77	4B	01	00	50	00	00	00	00	0	0	0	0	0	LDAR #B - записать в аккумулятор данные по адресу 774B	
LDA #B	2	9	01	0005	FFFF	0005	50	77	4B	01	00	50	00	00	00	00	0	0	0	0	0		
LDA #B	2	10	01	0006	FFFF	0006	50	77	4B	01	00	50	00	00	00	00	0	0	0	0	0		
LDA #B	2	11	01	0006	FFFF	0006	50	77	4B	01	00	50	00	00	00	00	0	0	0	0	0		
RLC	1	1	01	0006	FFFF	0006	07	4B	01	00	50	00	00	00	00	00	0	0	0	0	0	RLC - циклический сдвиг влево регистров A	
RLC	1	2	01	0006	FFFF	0006	07	4B	01	00	50	00	00	00	00	00	0	0	0	0	0		
RLC	1	3	07	0006	FFFF	0006	07	77	4B	01	00	50	00	00	00	00	0	0	0	0	0		
RLC	1	4	07	0006	FFFF	0006	07	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
RLC	1	5	07	0007	FFFF	0006	07	77	4B	02	00	50	00	00	00	00	0	0	0	0	0	RDR #A - декремент регистра	
RDR H	1	1	07	0007	FFFF	0007	07	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
RDR H	1	2	07	0007	FFFF	0007	25	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
RDR H	1	3	25	0007	FFFF	0007	25	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
RDR H	1	4	25	0007	FFFF	0007	25	77	4B	02	00	50	00	00	00	00	0	0	0	0	0	RDR #A - декремент регистра	
RDR H	1	5	25	0008	FFFF	0007	25	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
RDR L	1	1	25	0008	FFFF	0008	25	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
RDR L	1	2	25	0008	FFFF	0008	20	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
RDR L	1	3	20	0008	FFFF	0008	20	77	4B	02	00	50	00	00	00	00	0	0	0	0	0	RDR #A - декремент регистра	
RDR L	1	4	20	0008	FFFF	0008	20	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
RDR L	1	5	20	0009	FFFF	0008	20	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
RDR L	1	6	20	0009	FFFF	0009	20	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
ADC C	1	1	20	0009	FFFF	0009	20	77	4B	02	00	50	00	00	00	00	0	0	0	0	0	ADC C - Ак-С: Сумма регистра А суммируется с содержимым регистра С, результат сохраняется в регистре А	
ADC C	1	2	20	0009	FFFF	0009	0B	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
ADC C	1	3	0B	0009	FFFF	0009	0B	77	4B	02	00	50	00	00	00	00	0	0	0	0	0		
ADC C	1	4	0B	0009	FFFF	0009	0B	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC C	1	5	0B	000A	FFFF	0009	0B	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0	ADC M - Содержимое регистра А суммируется с содержимым регистра М, результат сохраняется в регистре А	
ADC M	1	1	0B	000A	FFFF	000A	0B	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	1	2	0B	000A	FFFF	000A	0B	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	1	3	0B	000A	FFFF	000A	0B	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	1	4	0B	000A	FFFF	000A	0B	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0	ADC M - Содержимое регистра А суммируется с содержимым регистра М, результат сохраняется в регистре А	
ADC M	2	5	0B	000A	FFFF	FFFF	0B	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	2	6	0B	000A	FFFF	FFFF	30	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	2	7	0B	000B	FFFF	FFFF	30	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	2	8	0B	000B	FFFF	FFFF	30	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0	ADC M - Содержимое регистра А суммируется с содержимым регистра М, результат сохраняется в регистре А	
ADC M	1	1	0B	000B	FFFF	FFFF	30	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	1	2	0B	000B	FFFF	FFFF	3F	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	1	3	3F	000B	FFFF	FFFF	3F	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	1	4	3F	000C	FFFF	FFFF	3F	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0	ADC M - Содержимое регистра А суммируется с содержимым регистра М, результат сохраняется в регистре А	
ADC M	1	5	3F	000C	FFFF	FFFF	3F	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	1	6	3F	000C	FFFF	FFFF	3F	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		
ADC M	1	7	3F	000C	FFFF	FFFF	3F	77	4B	5F	00	50	00	00	00	00	0	0	0	0	0		

Рисунок А.1 — Таблица состояний процессора. Первая программа

Таблица состояний регистров общ. назначения																	Оценка правильности выполнения юмандрой своих функций				
Команда	N цикла	N такта	PK	СК	УС	Биф. адр.	Биф. данных	Состояние флагов													
								W	Z	A	B	C	D	E	H	L	Z	S	P	C	AC
XRA B	0	0			FFFF			00	00	00	F1	00	00	00	00	00	0	0	0	0	0
XRA B	1	1			FFFF			00	00	00	F1	00	00	00	00	00	1	0	0	0	0
XRA B	1	2			FFFF	A8		00	00	00	F1	00	00	00	00	00	1	0	0	0	0
XRA B	1	3	A8	0000	FFFF	A8		00	00	00	F1	00	00	00	00	00	1	0	0	0	0
XRA B	1	4	A8	0000	FFFF	A8		00	00	00	F1	00	00	00	00	00	1	0	1	0	0
XRA B	1	5	A8	0001	FFFF	A8		00	00	00	F1	00	00	00	00	00	1	0	1	0	0
ORA B	1	1	A8	0001	FFFF	A8		00	00	00	F1	00	00	00	00	00	1	0	1	0	0
ORA B	1	2	A8	0001	FFFF	B0		00	00	00	F1	00	00	00	00	00	1	0	1	0	0
ORA B	1	3	B0	0001	FFFF	B0		00	00	00	F1	00	00	00	00	00	1	0	1	0	0
ORA B	1	4	B0	0001	FFFF	B0		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
ORA B	1	5	B0	0002	FFFF	B0		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
JM adr	1	1	B0	0002	FFFF	B0		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
JM adr	1	2	B0	0002	FFFF	FA		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
JM adr	1	3	FA	0002	FFFF	FA		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
JM adr	1	4	FA	0003	FFFF	FA		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
JM adr	2	5	FA	0003	FFFF	FA		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
JM adr	2	6	FA	0003	FFFF	07		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
JM adr	2	7	FA	0004	FFFF	07		07	07	F1	F1	00	00	00	00	00	0	1	0	0	0
JM adr	3	8	FA	0004	FFFF	07		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
JM adr	3	9	FA	0004	FFFF	00		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
JM adr	3	10	FA	0007	FFFF	00		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
NOP	1	1	FA	0007	FFFF	00		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
NOP	1	2	FA	0007	FFFF	00		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
NOP	1	3	00	0007	FFFF	00		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
NOP	1	4	00	0008	FFFF	00		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	1	1	00	0008	FFFF	00		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	1	2	00	0008	FFFF	CC		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	1	3	CC	0008	FFFF	CC		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	1	4	CC	0009	FFFF	CC		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	2	5	CC	0009	FFFF	CC		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	2	6	CC	0009	FFFF	00		07	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	2	7	CC	000A	FFFF	00		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	3	8	CC	000A	FFFF	00		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	3	9	CC	000A	FFFF	00		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	3	10	CC	000B	FFFF	00		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0
CZ adr	1	1	CC	000B	FFFF	00		00	00	F1	F1	00	00	00	00	00	0	1	0	0	0

Рисунок А.2 — Таблица состояний процессора. Вторая программа