

ВВЕДЕНИЕ

МикроЭВМ СМ-1800 представляет собой восьмиразрядную встраиваемую агрегативную машину, выполненную на микросхемах серий К580, К589, К565 и других. Центральный процессор ЭВМ КР580ИК80А является полным функциональным аналогом однокристального микропроцессора Intel 8080.

Все внешние устройства подключаются через соответствующие адаптеры к системной магистрали СМ-1800, выполненной по стандарту И41. Адресация устройств процессором согласно этому стандарту проходит по методу «изолированных линий», что означает использование специальных команд ввода/вывода (IN/OUT).

Лабораторный практикум по дисциплине «Организация ЭВМ и систем» включает в себя ряд работ на базе Эмулятора ЭВМ СМ-1800 на современных персональных компьютерах и содержит элементы программирования в машинных кодах и на ассемблере. Таким образом, настоящие методические указания являются дополнением к материалам, выдаваемым студентам для выполнения каждой из работ.

Методические указания могут быть использованы при проведении курсового проектирования по указанной дисциплине в качестве пособия для изучения приемов программирования на машинно-ориентированных языках.

1 КОМАНДЫ МИКРОЭВМ СМ-1800

1.1 ОБОЗНАЧЕНИЯ, ИСПОЛЬЗУЕМЫЕ ПРИ ОПИСАНИИ КОМАНД.

При описании команд микроЭВМ СМ-1800 используются следующие обозначения регистров:

- R – обозначение одного из однобайтных регистров макропроцессора из следующего набора: A, B, C, D, E, H, L;
- F – регистр флагов (однобайтный), пять из восьми битов которого используется для фиксации следующих признаков результата операции:

Таблица 1 – Назначение флагов

Разряд (бит) в регистре F	Обозначение бита	Содержание признака	Состояние бита	
			«1»	«0»
0	CY	перенос из 7-го разряда	есть перенос	нет переноса
4	AC	перенос из 3-го разряда	есть перенос	нет переноса
6	Z	нулевой результат	результат равен нулю	результат не равен нулю
7	S	знак результата	отрицательный (минус)	положительный (плюс)
2	P	оценка количества единиц в коде результата	четное	нечетное

SP – указатель стека (двухбайтовый регистр);

PC – программный счетчик (двухбайтовый регистр);

M – символ, используемый для обозначения ячейки оперативной памяти, причем адрес этой ячейки указывается косвенно – он содержится в паре регистров H, L.

Система команд содержит команды выполнения логических операций, для обозначения которых используются индексы:

Λ - логическое умножение – «И»;

∨ - логическое сложение – «ИЛИ»;

∇ - сложение по модулю 2 – исключающее «ИЛИ».

Конкретный код команд ADD, ADC, ANA, CMP, DCR, INR, MOV, MVI, ORA, SBB, SUB, XRA определяется подстановкой кода регистра-источника данных(SSS) или кода регистра-приемника данных (DDD).

Приняты следующие коды для обозначения регистров:

Имя регистра	Код регистра
B	000
C	001
D	010
E	011
H	100
L	101
M (память)	110
A (аккумулятор)	111

Примеры.

1. Команда ADD R(код 1000 0SSS) осуществляет сложение содержимого аккумулятора и регистра R. Если в конкретном случае регистром R является регистр C, код которого SSS=001, то код команды записывается так:
1000 0001 или 81H в шестнадцатиричной системе.
2. Команда MOV R1,R2 (код 01DDSSS) осуществляет пересылку содержимого из регистра-источника R2 в регистр-приемник R1. Если в конкретном случае регистром-источником является регистр E (SSS=011), а регистром-приемником регистр B (DDD=010), то код команды: 0101 0011=53H.

В описании команд циклического сдвига содержимого аккумулятора (RAL, RAR, RLC, RRC) использовано обозначение AM - бит M регистра A (аккумулятора), причем A0 и A7-соответственно младший и старший (знаковый) биты аккумулятора.

При чтении описания команд следует учитывать следующие обозначения при записи:

B2 - второй байт команды (в командах ADI, ACI, ANI, CPI, MVI, ORI, SBI, SUI, XRI – это непосредственные данные, в командах IN и OUT – адрес порта),

B3 – третий байт команды, обычно самостоятельно не выступающий,
→ направление передачи данных (...из...в...),

() - содержимое байта (команда, регистра, памяти). Следует читать:

(H) - содержимое регистра H,

(B) - содержимое регистра B,

(B2)- содержимое второго байта команды,

(M) - содержимое ячейки оперативной памяти, адрес которой помещен в паре регистров H, L, т.е. адресом является (H, L).

При описании команд вызова (безусловного CALL и восьми по условию), требующих запоминания адреса возврата из подпрограммы в стековой памяти, а также команд извлечения содержимого регистров из стека (POP) и засылки содержимого регистров в стек (PUSH) использованы квадратные скобки для обозначения адреса ячейки стека.

Например, запись [SP-1] [SP-2] ← (PC) следует читать так: поместить содержимое программного счетчика PC (два байта) в две ячейки стековой памяти, адреса которых определяются как уменьшенное соответственно на единицу и на два содержимое указателя стека SP.

Для команд условного вызова подпрограмм и возврата по условию в их описании даны два значения количества машинных циклов в команде: первое, если условие не выполнено и далее выполняется очередная команда программы, а второе – если условие выполнено и

происходит вызов подпрограммы или возврат. Время выполнения одного цикла команды от 1 до 5 машинных циклов.

ЗАМЕЧАНИЕ:

все команды преобразования данных (арифметические, логические) модифицируют пять описанных в таблице 1 флагов. В результате выполнения любой логической операции флаг CY сбрасывается в 0. Операция инкремента и декремента коротких (байтовых) регистров (INR R, DCR R) модифицируют все флаги кроме CY. Команды INX R и DCX R значения флагов не изменяют.

1.2 СПИСОК КОМАНД

Таблица 2 – Список команд

Мнемоника ассемблера	Байты	Циклы	Коды		Описание
			двоичные	Н	
1	2	3	4	5	6
ADD R	1	1	10000SSS		$(A) \leftarrow (A) + (R)$
ADD M	1	2	10000110	86	$(A) \leftarrow (A) + (M)$
ADI (B2)	2	2	11000110	C6	$(A) \leftarrow (A) + (B2)$
ADC R	1	1	10001SSS		$(A) \leftarrow (A) + (R) + (CY)$
ADC M	1	2	10001110	8E	$(A) \leftarrow (A) + (M) + (CY)$
ACI (B2)	2	2	11001110	CE	$(A) \leftarrow (A) + (B2) + (CY)$
ANA R	1	1	10100SSS		$(A) \leftarrow (A) \wedge (R)$
ANA M	1	2	10100110	A6	$(A) \leftarrow (A) \wedge (M)$
ANI B2	2	2	11100110	E6	$(A) \leftarrow (A) \wedge (B2)$
CALL (B2)(B3)	3	5	11001101	C D	$[SP-1][SP-2] \leftarrow (PC),$ $(SP) = (SP) - 2,$ $(PC) \leftarrow (B3)(B2)$
CC (B2)(B3)	3	3/5	11011100	D C	Если $CY=1$, то $[SP-1][SP-2] \leftarrow (PC)$ $(SP) = (SP) - 2,$ $(PC) \leftarrow (B3)(B2)$ иначе $(PC) = (PC) + 3$

Таблица 2 – Список команд (продолжение)

1	2	3	4	5	6
CNC (B2)(B3)	3	3/5	11010100	D4	Если CY=0, то [SP-1][SP-2]←(PC) (SP)=(SP)-2, (PC)← (B3)(B2) иначе (PC)=(PC) + 3
CZ (B2)(B3)	3	3/5	11001100	CC	Если Z=1, то [SP-1][SP-2]←(PC) (SP)=(SP)-2, (PC)← (B3)(B2) иначе (PC)=(PC) + 3
CNZ (B2)(B3)	3	3/5	11000100	C4	Если Z=0, то [SP-1][SP-2]←(PC) (SP)=(SP)-2, (PC)← (B3)(B2) иначе (PC)=(PC) + 3
CM (B2)(B3)	3	3/5	11111100	FC	Если S=1, то [SP-1][SP-2]←(PC) (SP)=(SP)-2, (PC)← (B3)(B2) иначе (PC)=(PC) + 3
CP (B2)(B3)	3	3/5	11110100	F4	Если S=0, то [SP-1][SP-2]←(PC) (SP)=(SP)-2, (PC)← (B3)(B2) иначе (PC)=(PC) + 3
CPE (B2)(B3)	3	3/5	11101100	EC	Если P=1, то [SP-1][SP-2]←(PC) (SP)=(SP)-2, (PC)← (B3)(B2) иначе (PC)=(PC) + 3
CPO (B2)(B3)	3	3/5	11100100	E4	Если P=0, то [SP-1][SP-2]←(PC) (SP)=(SP)-2, (PC)← (B3)(B2) иначе (PC)=(PC) + 3
CMA	1	1	00101111	2F	(A)← \bar{A}
CMC	1	1	00111111	3F	CY← \overline{CY}
CMP R	1	1	10111SSS		(A) - (R)

Таблица 2 – Список команд (продолжение)

1	2	3	4	5	6
CMP M	1	2	10111110	BE	(A) – (M)
CPI (B2)	2	2	11111110	FE	(A) – (B2)
DAA	1	1	00100111	27	Преобразование сумматора
DAD B	1	1	00001001	09	(H)(L)←(H)(L) + (B)(C)
DAD D	1	1	00011001	19	(H)(L)←(H)(L) + (D)(E)
DAD H	1	1	00101001	29	(H)(L)←(H)(L) + (H)(L)
DAD SP	1	1	00111001	39	(H)(L)←(H)(L) + (SP)
DCR R	1	1	00DDD101		(R)←(R) - 1
DCR M	1	2	00110101	35	(M)←(M) - 1
DCX B	1	1	00001011	0B	(B)(C)←(B)(C) - 1
DCX D	1	1	00011011	1B	(D)(E)←(D)(E) - 1
DCX H	1	1	00101011	2B	(H)(L)←(H)(L) - 1
DCX SP	1	1	00111011	3B	(SP)←(SP) - 1
DI	1	1	11110011	F3	Запрещение системного прерывания
EI	1	1	11111011	FB	Разрешение системного прерывания
HLT	1	1	01110110	76	Останов
IN (B2)	3	2	11011011	DB	(A)←(порт ввода)
INR R	1	1	00DDD100		(R)←(R) + 1
INR M	1	2	00110100	34	(M)←(M) + 1
INX B	1	1	00000011	03	(B)(C)←(B)(C) + 1
INX D	1	1	00010011	13	(D)(E)←(D)(E) + 1
INX H	1	1	00100011	23	(H)(L)←(H)(L) + 1
INX SP	1	1	00110011	33	(SP)←(SP) + 1
JMP (B2)(B3)	3	3	11000011	C3	(PC)← (B3)(B2)
JC (B2)(B3)	3	3	11011010	DA	Если CY=1, то (PC)← (B3)(B2) иначе (PC)=(PC) + 3

Таблица 2 – Список команд (продолжение)

1	2	3	4	5	6
JNC (B2)(B3)	3	3	11010010	D2	Если CY=0, то (PC)← (B3)(B2) иначе (PC)=(PC) + 3
JZ (B2)(B3)	3	3	11001010	CA	Если Z=1, то (PC)← (B3)(B2) иначе (PC)=(PC) + 3
JNZ (B2)(B3)	3	3	11000010	C2	Если Z=0, то (PC)← (B3)(B2) иначе (PC)=(PC) + 3
JM (B2)(B3)	3	3	11111010	FA	Если S=1, то (PC)← (B3)(B2) иначе (PC)=(PC) + 3
JP (B2)(B3)	3	3	11110010	F2	Если S=0, то (PC)← (B3)(B2) иначе (PC)=(PC) + 3
JPE (B2)(B3)	3	3	11101010	EA	Если P=1, то (PC)← (B3)(B2) иначе (PC)=(PC) + 3
JPO (B2)(B3)	3	3	11100010	E2	Если P=0, то (PC)← (B3)(B2) иначе (PC)=(PC) + 3
LDA (B2)(B3)	3	4	00111010	3A	(A)←[(B3)(B2)]
LDAX B	1	2	00001010	0A	(A)←[(B)(C)]
LDAX D	1	2	00011010	1A	(A)←[(D)(C)]
LHL D (B2)(B3)	3	5	00101010	2A	(L)← [(B3)(B2)] (H)← [(B3)(B2) + 1]
LXI B, (B2)(B3)	3	3	00000001	01	(C)← (B2), (B)←(B3)
LXI D, (B2)(B3)	3	3	00010001	11	(E)← (B2), (D)←(B3)
1	2	3	4	5	6
LXI H, (B2)(B3)	3	3	00100001	21	(L)← (B2), (H)←(B3)
LXI SP, (B2)(B3)	3	3	00110001	31	(SP)L← (B2), (SP)H←(B3)
MOV R1,R2	1	1	01DDDSSS		(R1)←(R2)
MOV R,M	1	2	01DDD110		(R)←(M)
MOV M,R	1	2	01110SSS		(M)←(R)

Таблица 2 – Список команд (продолжение)

1	2	3	4	5	6
MVI R, (B2)	2	2	00DDD110		(R) \leftarrow (B2)
MVI M, (B2)	2	3	00110110	36	(M) \leftarrow (B2)
ORA R	1	1	10110SSS		(A) \leftarrow (A) \vee (R)
ORA M	1	2	10110110	B6	(A) \leftarrow (A) \vee (M)
ORI (B2)	2	2	11110110	F6	(A) \leftarrow (A) \vee (B2)
OUT (B2)	2	3	11010011	D3	(Порт вывода) \leftarrow (A)
PCHL	1	1	11101001	E9	(PC) \leftarrow (H)(L)
POP B	1	3	11000001	C1	(C) \leftarrow [SP], (B) \leftarrow [SP+1], (SP)=(SP)+2
POP D	1	3	11010001	D1	(E) \leftarrow [SP], (D) \leftarrow [SP+1], (SP)=(SP)+2
POP H	1	3	11100001	E1	(L) \leftarrow [SP], (H) \leftarrow [SP+1], (SP)=(SP)+2
POP PSW	1	3	11110001	F1	(F) \leftarrow [SP], (A) \leftarrow [SP+1], (SP)=(SP)+2
PUSH B	1	3	11000101	05	[SP-1] \leftarrow (B), [SP-2] \leftarrow (C), (SP)=(SP-2)
PUSH D	1	3	11010101	D5	[SP-1] \leftarrow (D), [SP-2] \leftarrow (E), (SP)=(SP-2)
PUSH H	1	3	11100101	E5	[SP-1] \leftarrow (H), [SP-2] \leftarrow (L), (SP)=(SP-2)
PUSH PSW	1	3	11110101	F5	[SP-1] \leftarrow (A), [SP-2] \leftarrow (F), (SP)=(SP-2)
RAL	1	1	00010111	17	AM+1 \leftarrow AM, A0 \leftarrow CY, CY \leftarrow A7
RAR	1	1	00011111	1F	AM \leftarrow AM+1, A7 \leftarrow CY, CY \leftarrow A0
RLC	1	1	00000111	07	AM+1 \leftarrow AM, A0 \leftarrow A7, CY \leftarrow A7
RRC	1	1	00001111	0F	AM+1 \leftarrow AM, A7 \leftarrow A0, CY \leftarrow A0

Таблица 2 – Список команд (продолжение)

1	2	3	4	5	6
RET	1	3	11001001	C9	$(PC) \leftarrow [SP][SP+1]$, $(SP) = (SP) + 2$
RC	1	1/3	11011000	D8	Если CY=1, то $(PC) \leftarrow [SP][SP+1]$, $(SP) = (SP) + 2$ иначе $(PC) = (PC) + 1$
RNC	1	1/3	11010000	D0	Если CY=0, то $(PC) \leftarrow [SP][SP+1]$, $(SP) = (SP) + 2$ иначе $(PC) = (PC) + 1$
RZ	1	1/3	11001000	C8	Если Z=1, то $(PC) \leftarrow [SP][SP+1]$, $(SP) = (SP) + 2$ иначе $(PC) = (PC) + 1$
RNZ	1	1/3	11000000	C0	Если Z=0, то $(PC) \leftarrow [SP][SP+1]$, $(SP) = (SP) + 2$ иначе $(PC) = (PC) + 1$
RM	1	1/3	111110000	F8	Если S=1, то $(PC) \leftarrow [SP][SP+1]$, $(SP) = (SP) + 2$ иначе $(PC) = (PC) + 1$
RP	1	1/3	11110000	F0	Если S=0, то $(PC) \leftarrow [SP][SP+1]$, $(SP) = (SP) + 2$ иначе $(PC) = (PC) + 1$
RPE	1	1/3	11101000	E8	Если P=1, то $(PC) \leftarrow [SP][SP+1]$, $(SP) = (SP) + 2$ иначе $(PC) = (PC) + 1$
RPO	1	1/3	11100000	E0	Если P=0, то $(PC) \leftarrow [SP][SP+1]$, $(SP) = (SP) + 2$ иначе $(PC) = (PC) + 1$
RST	1	3	11AAA111		$[SP-1][SP-2] \leftarrow (PC)$, $(SP) = (SP) - 2$, $(PC) \leftarrow (00000000$ $00AAA000)$
SBB R	1	1	10011SSS		$(A) \leftarrow (A) - (R)$ – заем
SBB M	1	2	10011110	9E	$(A) \leftarrow (A) - (M)$ – заем

Таблица 2 – Список команд (продолжение)

1	2	3	4	5	6
SBI (B2)	2	2	11011110	DE	$(A) \leftarrow (A) - (B2) - \text{заем}$
SUB R	1	1	10010SSS		$(A) \leftarrow (A) - (R)$
SUB M	1	2	10010110	96	$(A) \leftarrow (A) - (M)$
SUI (B2)	2	2	11010110	D6	$(A) \leftarrow (A) - (B2)$
SHLD (B2)(B3)	3	5	00100010	22	$[(B3)(B2)] \leftarrow (L),$ $[(B3)(B2)+1] \leftarrow (H)$
SPHL	1	1	11111001	F9	$(SP) \leftarrow (H)(L)$
STA (B2)(B3)	3	4	00110010	32	$[(B3)(B2)] \leftarrow (A)$
STAX B	1	2	00000010	02	$[(B)(C)] \leftarrow (A)$
STAX D	1	2	00010010	12	$[(D)(E)] \leftarrow (A)$
STC	1	1	00110111	37	$CY \leftarrow 1$
XRA R	1	1	10101SSS		$A \leftarrow (A) \vee (R)$
XRA M	1	1	10101110	AE	$A \leftarrow (A) \vee (M)$
XRI (B2)	2	2	11101110	EE	$A \leftarrow (A) \vee (B2)$
XCHG	1	1	11011011	EB	$(H) \leftrightarrow (D), (E) \leftrightarrow (L)$
XTHL	1	1	11100101	E5	$(L) \leftrightarrow [SP], (H) \leftrightarrow [SP+1]$

1.3 КРАТКОЕ ОПИСАНИЕ КОМАНД

Таблица 3 – Описание команд

Обозначение	Описание
ADD R	Сложение содержимого регистра R с содержимым аккумулятора
ADD M	Сложение содержимого ячейки памяти с содержимым аккумулятора
ADI (B2)	Сложение непосредственных данных (B2) с содержимым аккумулятора
ADC R	Сложение содержимого регистра R и аккумулятора
ADC M	Сложение содержимого ячейки памяти и аккумулятора
ACI (B2)	Сложение непосредственных данных (B2) с содержимым аккумулятора
ANA R	Логическое умножение («И») содержимого регистра R и аккумулятора
ANA M	Логическое умножение («И») содержимого ячейки памяти и аккумулятора

Таблица 3 – Описание команд (продолжение)

ANI B2	Логическое умножение («И») содержимого непосредственных данных (B2) и аккумулятора
CALL (B2)(B3)	Вызов безусловный, т.е. переход к команде, адрес которой содержится во втором и третьем байтах команды
CC (B2)(B3)	Вызов по условию: при наличии переноса CY=1 – (по переносу)
CNC (B2)(B3)	Вызов по условию: при отсутствии переноса CY=0 – (нет переноса)
CZ (B2)(B3)	Вызов по условию: результат равен нулю Z=1 – (по нулю)
CNZ (B2)(B3)	Вызов по условию: результат не равен нулю Z=0 – (нет нуля)
CM (B2)(B3)	Вызов по условию: результат отрицателен S=1 – (по минусу)
CP (B2)(B3)	Вызов по условию: результат положителен S=0 (по плюсу)
CPE (B2)(B3)	Вызов по условию: сумма единиц в коде результата четная P=1 – (по четности)
CPO (B2)(B3)	Вызов по условию: сумма единиц в коде результата нечетная P=0 – (по нечетности)
CMA	Инвертирование содержимого аккумулятора
CMC	Инвертирование содержимого флага переноса CY
CMP R	Сравнение содержимого регистра R и аккумулятора
CMP M	Сравнение содержимого ячейки памяти и аккумулятора
CPI (B2)	Сравнение непосредственных данных (B2) и аккумулятора
DAA	Преобразование содержимого аккумулятора в BDD (двоично-десятичном коде)
DAD B	Сложение (B,C) с (H,L)
DAD D	Сложение (D,E) с (H,L)
DAD H	Сложение (H,L) с (H,L)
DAD SP	Сложение указателя стека (SP) с (H,L)
DCR R	Отрицательное приращение R
DCR M	Отрицательное приращение содержимого ячейки памяти
DCX B	Отрицательное приращение (B,C)
DCX D	Отрицательное приращение (D,E)
DCX H	Отрицательное приращение (H,L)
DCX SP	Отрицательное приращение содержимого указателя стека (SP)
DI	Запрещение системного прерывания
EI	Разрешение системного прерывания
HLT	Останов

Таблица 3 – Описание команд (продолжение)

IN (B2)	Ввод данных в аккумулятор из порта, адрес которого определяется содержимым второго байта команды (B2)
INR R	Положительное приращение (R)
INR M	Положительное приращение содержимого ячейки памяти
INX B	Положительное приращение (B,C)
INX D	Положительное приращение (D,E)
INX H	Положительное приращение (H,L)
INX SP	Положительное приращение содержимого указателя стека (SP)
JMP (B2)(B3)	Переход безусловный к команде, адрес которой содержится во втором и третьем байтах команды
JC (B2)(B3)	Переход по условию: CY=1 – (по переносу)
JNC (B2)(B3)	Переход по условию: CY=0 – (нет переноса)
JZ (B2)(B3)	Переход по условию: Z=1 – (по нулю)
JNZ (B2)(B3)	Переход по условию: Z=0 – (нет нуля)
JM (B2)(B3)	Переход по условию: S=1 – (по минусу)
JP (B2)(B3)	Переход по условию: S=0 (по плюсу)
JPE (B2)(B3)	Переход по условию: P=1 – (по четности)
JPO (B2)(B3)	Переход по условию: P=0 – (по нечетности)
LDA (B2)(B3)	Загрузка аккумулятора содержимым ячейки памяти, адрес которой содержится во втором и третьем байтах команды
LDAX B	Загрузка аккумулятора содержимым ячейки памяти, адрес которой содержится в регистрах B,C
LDAX D	Загрузка аккумулятора содержимым ячейки памяти, адрес которой содержится в регистрах D,E
LHLD (B2)(B3)	Загрузка регистров H,L содержимым ячейки памяти, адрес которой содержится во втором и третьем байтах команды
LXI B, (B2)(B3)	Загрузка непосредственных данных B2,B3 в регистры B,C
LXI D, (B2)(B3)	Загрузка непосредственных данных B2,B3 в регистры D,E
LXI H, (B2)(B3)	Загрузка непосредственных данных B2,B3 в регистры H,L
LXI SP, (B2)(B3)	Загрузка непосредственных данных B2,B3 в указатель стека SP
MOV R1,R2	Пересылка содержимого регистра R2 в регистр R1
MOV R,M	Пересылка содержимого ячейки памяти в регистр R
MOV M,R	Пересылка содержимого регистра R в ячейку памяти
MVI R, (B2)	Пересылка непосредственных данных B2 в регистр R
MVI M, (B2)	Пересылка непосредственных данных B2 в ячейку памяти

Таблица 3 – Описание команд (продолжение)

ORA R	Логическое сложение («ИЛИ») содержимого регистра R и аккумулятора
ORA M	Логическое сложение («ИЛИ») содержимого ячейки памяти и аккумулятора
ORI (B2)	Логическое сложение («ИЛИ») содержимого непосредственных данных (B2) и аккумулятора
OUT (B2)	Вывод данных из аккумулятора в порт, адрес которого содержится во втором байте команды B2
PCHL	Засылка (H,L) в программный счетчик PC
POP B	Извлечение содержимого регистров B,C из стека
POP D	Извлечение содержимого регистров D,E из стека
POP H	Извлечение содержимого регистров H,L из стека
POP PSW	Извлечение содержимого аккумулятора A и регистра F из стека
PUSH B	Засылка содержимого регистров B,C в стек
PUSH D	Засылка содержимого регистров D,E в стек
PUSH H	Засылка содержимого регистров H,L в стек
PUSH PSW	Засылка содержимого аккумулятора A и регистра F в стек
RAL	Циклический сдвиг содержимого аккумулятора влево на один разряд CY в A0,A7 в CY
RAR	Циклический сдвиг содержимого аккумулятора вправо на один разряд CY в A7,A0 в CY
RLC	Циклический сдвиг содержимого аккумулятора влево на один разряд A7 в A0,A7 в CY
RRC	Циклический сдвиг содержимого аккумулятора вправо на один разряд A0 в A0,A7 в CY
RET	Возврат безусловный к команде с адресом, помещенным последним в стек
RC	Возврат по условию: CY=1 – (по переносу)
RNC	Возврат по условию: CY=0 – (нет переноса)
RZ	Возврат по условию: Z=1 – (по нулю)
RNZ	Возврат по условию: Z=0 – (нет нуля)
RM	Возврат по условию: S=1 – (по минусу)
RP	Возврат по условию: S=0 (по плюсу)
RPE	Возврат по условию: P=1 – (по четности)
RPO	Возврат по условию: P=0 – (по нечетности)
RST	Пуск по результатам обработки прерывания
SBB R	Вычитание из аккумулятора содержимого регистра R с заемом

Таблица 3 – Описание команд (продолжение)

SBB M	Вычитание из аккумулятора содержимого ячейки памяти с заемом
SBI (B2)	Вычитание из аккумулятора непосредственных данных B2 с заемом
SUB R	Вычитание из аккумулятора содержимого регистра R
SUB M	Вычитание из аккумулятора содержимого ячейки памяти
SUI (B2)	Вычитание из аккумулятора непосредственных данных B2
SHLD (B2)(B3)	Запись (H,L) в память с адресом, который содержится во втором и третьем байтах команды
SPHL	Засылка (H,L) в указатель стека SP
STA (B2)(B3)	Запись содержимого аккумулятора в ячейку памяти, адрес которой содержится во втором и третьем байтах команды
STAX B	Запись содержимого аккумулятора в ячейку памяти, адрес которой содержится в регистрах B,C
STAX D	Запись содержимого аккумулятора в ячейку памяти, адрес которой содержится в регистрах D,E
STC	Установка флага переноса в состояние «1»
XRA R	Сложение по модулю («Исключающее ИЛИ») содержимого регистра R и аккумулятора
XRA M	Сложение по модулю («Исключающее ИЛИ») содержимого ячейки памяти и аккумулятора
XRI (B2)	Сложение по модулю («Исключающее ИЛИ») непосредственных данных B2 и аккумулятора
XCHG	Обмен содержимым между регистрами D,E и H,L
XTHL	Обмен содержимым между верхними ячейками стека и регистрами H,L

1.4 КОДЫ ОПЕРАЦИЙ КОМАНД (в порядке возрастания)

Таблица 4 – Коды операций команд

00 NOP	40 MOV B,B	80 ADD B	C0 RNZ
01 LXI B,XXXX	41 MOV B,C	81 ADD C	C1 POP B
02 STAX B	42 MOV B,D	82 ADD D	C2 JNZ XXXX
03 INX B	43 MOV B,E	83 ADD E	C3 JMP XXXX
04 INR B	44 MOV B,H	84 ADD H	C4 CNZ XXXX
05 DCR B	45 MOV B,L	85 ADD L	C5 PUSH B
06 MVI B,XX	46 MOV B,M	86 ADD M	C6 ADI XX
07 RLC	47 MOV B,A	87 ADD A	C7 RST 0

Таблица 4 – Коды операций команд (продолжение)

08 - - - -	48 MOV C,B	88 ADC B	C8 RZ
09 DAD B	49 MOV C,C	89 ADC C	C9 RET
0A LDAX B	4A MOV C,D	8A ADC D	CA JZ XXXX
0B DCX B	4B MOV C,E	8B ADC E	CB - - - -
0C INR C	4C MOV C,H	8C ADC H	CC CZ XXXX
0D DCR C	4D MOV C,L	8D ADC L	CD CALL XXXX
0E MVI C,XX	4E MOV C,M	8E ADC M	CE ACI XX
0F RRC	4F MOV C,A	8F ADC A	CF RST 1
10 - - - -	50 MOV D,B	90 SUB B	D0 RNC
11 LXI D,XXXX	51 MOV D,C	91 SUB C	D1 POP D
12 STAX D	52 MOV D,D	92 SUB D	D2 JNC XXXX
13 INX D	53 MOV D,E	93 SUB E	D3 OUT XXXX
14 INR D	54 MOV D,H	94 SUB H	D4 CNC XXXX
15 DCR D	55 MOV D,L	95 SUB L	D5 PUSH D
16 MVI D,XX	56 MOV D,M	96 SUB M	D6 SUI XX
17 RAL	57 MOV D,A	97 SUB A	D7 RST 2
18 - - - -	58 MOV E,B	98 SBB B	D8 RC
19 DAD D	59 MOV E,C	99 SBB C	D9 - - - -
1A LDAX D	5A MOV E,D	9A SBB D	DA JC XXXX
1B DCX D	5B MOV E,E	9B SBB E	DB IN XX
1C INR E	5C MOV E,H	9C SBB H	DC CC XXXX
1D DCR E	5D MOV E,L	9D SBB L	DD - - - -
1E MVI E,XX	5E MOV E,M	9E SBB M	DE SBI XX
1F RAR	5F MOV E,A	9F SBB A	DF RST 3
20 - - - -	60 MOV H,B	A0 ANA B	E0 RPO
21 LXI H,XXXX	61 MOV H,C	A1 ANA C	E1 POP H
22 SHLD XXXX	62 MOV H,D	A2 ANA D	E2 JPO XXXX
23 INX H	63 MOV H,E	A3 ANA E	E3 XTHL
24 INR H	64 MOV H,H	A4 ANA H	E4 CPO XXXX
25 DCR H	65 MOV H,L	A5 ANA L	E5 PUSH H
26 MVI H,XX	66 MOV H,M	A6 ANA M	E6 ANI XX
27 DAA	67 MOV H,A	A7 ANA A	E7 RST 4
28 - - - -	68 MOV L,B	A8 XRA B	E8 RPE

Таблица 4 – Коды операций команд (продолжение)

29 DAD H	69 MOV L,C	A9 XRA C	E9 PCHL
2A LHLD XXXX	6A MOV L,D	AA XRA D	EA JPE XXXX
2B DCX H	6B MOV L,E	AB XRA E	EB XCHG
2C INR L	6C MOV L,H	AC XRA H	EC CPE XXXX
2D DCR L	6D MOV L,L	AD XRA L	ED - - - -
2E MVI L,XX	6E MOV L,M	AE XRA M	EE XRI XX
2F CMA	6F MOV L,A	AF XRA A	EF RST 5
30 - - - -	70 MOV M,B	B0 ORA B	F0 RP
31 LXI SP,XXXX	71 MOV M,C	B1 ORA C	F1 POP PSW
32 STA XXXX	72 MOV M,D	B2 ORA D	F2 JP XXXX
33 INX SP	73 MOV M,E	B3 ORA E	F3 DI
34 INR M	74 MOV M,H	B4 ORA H	F4 CP XXXX
35 DCR M	75 MOV M,L	B5 ORA L	F5 PUSH PSW
36 MVI M,XX	76 HLT	B6 ORA M	F6 ORI XX
37 STC	77 MOV M,A	B7 ORA A	F7 RST 6
38 - - - -	78 MOV A,B	B8 CMP B	F8 RM
39 DAD SP	79 MOV A,C	B9 CMP C	F9 SPHL
3A LHLD XXXX	7A MOV A,D	BA CMP D	FA JM XXXX
3B DCX SP	7B MOV A,E	BB CMP E	FB EI
3C INR A	7C MOV A,H	BC CMP H	FC CM XXXX
3D DCR A	7D MOV A,L	BD CMP L	FD - - - -
3E MVI A,XX	7E MOV A,M	BE CMP M	FE CPI XX
3F CMC	7F MOV A,A	BF CMP A	FF RST 7

Условные обозначения:

XXXX – адрес ячейки памяти или данные

XX – адрес порта или данные

- - - - – код операции не используется

2 НЕКОТОРЫЕ ПРИЁМЫ ПРОГРАММИРОВАНИЯ В КОДАХ

Ниже будут рассмотрены фрагменты программ, реализующие наиболее распространенные, часто встречающиеся приемы обработки данных. Хотя речь идет о программах в машинных кодах, мы будем пользоваться записью команд в мнемоническом изображении, принятом на языке ассемблера для данного типа микропроцессора. Кроме мнемоники кодов операций для упрощения записи вместо конкретных числовых адресов команд мы воспользуемся метками (также принятыми в языке ассемблера).

2.1 ПЕРЕМЕЩЕНИЕ ДАННЫХ

В системе машинных команд (раздел 1) отсутствуют команды перемещения байта непосредственно из одной ячейки памяти в другую. Такую пересылку приходится реализовывать несколькими командами. Усложним эту задачу условием – поменять местами содержимое двух различных ячеек памяти. Для определенности адреса обменивающихся ячеек возьмем равными 5000H и 5001H. (Буква H здесь и далее обозначает шестнадцатиричную систему счисления, принятую для нумерации ячеек).

...

LXI H, 5000H;	засылка адреса первого числа в пару HL
LXI B, 5001H;	засылка адреса второго числа в пару BC
MOV D,M;	копирование первого числа в регистр D
LDAX B;	копирование второго числа в аккумулятор
MOV M,A;	перемещение второго числа в ячейку 5000H
MOV A,D;	перезапись первого числа в аккумулятор
STAX B;	запись содержимого аккумулятора в ячейку 5001H

...

2.2 ОРГАНИЗАЦИЯ ЦИКЛА

Наиболее рациональный способ организации цикла (многократного повтора одного участка программы) – это создание в одном из регистров процессов счетчика, работающего на вычитание

Перед «телом» цикла счетчик загружается константой числа шагов, а в конце «тела» производится вычитание единицы (декремент) и проверка флага нулевого результата командой условного перехода. Пусть требуется просуммировать массив из 10 целых однобайтных чисел без знака записанных с адреса 5100H.

MVI B, 0AH ;	начальная загрузка счетчика B числом 10
XRA A ;	«очистка» (обнуление) аккумулятора
LXI H, 5100H ;	засылка в пару HL адреса 1-го числа
M1: ADD M ;	A + ячейка <HL> → A
INX H ;	продвижение адреса HL + 1 → HL
DCR B ;	уменьшение содержимого счетчика
JNZ M1 ;	переход к метке M1 по флагу Z=0
HLT ;	останов

2.3 ИСПОЛЬЗОВАНИЕ ПОДПРОГРАММ

Если в какой либо программе требуется многократно и на разных ее участках выполнить одни и те же действия, целесообразно

оформить их в виде подпрограммы. Пусть необходимо просуммировать соответствующие элементы двух массивов из пяти двухбайтных целых чисел без знака. (В младшем байте содержатся младшие разряда числа). Начальные адреса массивов слагаемых 5000H и 5001H, а массив сумм располагается с ячейки 5300H.

Подпрограмма:

SB: PUSH PSW ; сохранение в стеке содержимого аккумулятора и флагов
LDAX B ; загрузка в A младшего байта 1-го слагаемого (адрес в паре BC)
ADD M ; сложение A с младшим байтом 2-го слагаемого (адрес в паре HL)
STAX B ; запись результата в ОЗУ по адресу из пары DE
INX B ; переход по адресам
INX H ; к старшим частям слагаемых
INX D ; и результата
STAX D ; загрузка в A старшего байта второго слагаемого
ADC M ; сложение старших байтов с учетом флага переноса CY
STAX D ; запись старшего байта результата
POP PSW ; восстановление аккумулятора и флагов
RET ; возврат

Основная программа:

LXI SP,8000H ; создание стека программы
LXI B,5000H ; адрес массива 1-х слагаемых в пару BC
LXI H,5100H ; адрес массива 2-х слагаемых в пару HL
LXI D,5300H ; адрес массива результатов в пару DE
MVI A,05 ; создание счетчика в A на 5 шагов
M1: CALL SB ; вызов подпрограммы сложения
INX B ; продвижение адресов
INX H ; на следующую пару слагаемых
INX D ; и ячейку результата
DCR A ; уменьшение счётчика на 1
JNZ M1 ; возврат на M1 до истощения счетчика
HLT ; станов

Следует обратить внимание в приведенном выше примере на использование стековой памяти для временного сохранения текущего значения счетчика, зарезервированного в аккумуляторе, на период работы подпрограммы. Данный приём используется часто для

сохранения значения и других регистров общего назначения при их "нехватке" по ходу реализации сложного алгоритма.

2.4 ПЕРЕКОДИРОВАНИЕ ИНФОРМАЦИИ

Одной из наиболее распространенных задач в составе общего программного обеспечения любой ЭВМ являются программы преобразования данных из числовых форматов в символьные для вывода на устройства отображения и обратно при вводе числовой информации. При этом активно используются команды логических, операций и сдвига. Рассмотрим задачу преобразования байта (целого без знака числа) в пару байтов кодов символического кодирования КОИ-7, отражающих содержимое двух тетрад исходного числа в виде символов соответствующих шестнадцатеричных цифр. Пусть исходное число записано в регистре D, а коды КОИ-7 для старшей и младшей тетрады нужно получить в регистрах В и С соответственно.

Подпрограмма:

PR: CPI 0AH; сравнение содержимого A с числом: (A)-0A
JNC M1 ; переход в случае отсутствия заёма на M1 (буквы)
ORI 30H ; формирование кода КОИ-7 арабской цифры
RET ; возврат
M1: ADI 37H ; формирование кода КОИ-7 букв от A до F
RET ; возврат

Основная программа:

MOV A,D ; в регистр A исходное число
ANI 0F0H ; обнуление младшей тетрады
RAR ;
RAR ; перемещение старшей тетрады
RAR ; вправо на место младшей (на 4 разряда)
RAR ;
CALL PR ; вызов подпрограммы преобразования тетрады
MOV B,A ; запись кода символа старшей тетрады
MOV A,D ; в регистр A снова исходное число
ANI 0FH ; очистка старшей тетрады
CALL PR ; вызов подпрограммы преобразования тетрады
MOV C,A ; запись кода символа младшей тетрады
HLT ; останов

Идея алгоритма приведенной выше программы состоит в выделении в подпрограмму задачи преобразования байта, где левая тетрада содержит нули, а число в правой тетраде может быть заменено символом арабской цифры от 0 до 9 или символом латинской буквы от А до F. Из основной программы обращение к подпрограмме осуществляется в первый раз для преобразования левой тетрады исходного числа, а второй раз – для преобразования правой тетрады.

ЛИТЕРАТУРА

- 1 Токхайм Р. Микропроцессоры: Курс и упражнения / Пер. с англ. ; под ред. В.Н. Грасевича, М.: Энергоатомиздат. 1988. -336 с.
- 2 Гуртовцев А.Л.,Гудыменко С.В. Программы для микропроцессоров: Справ. пособие. М.:Выш.шк.,1989.- 352 с.
- 3 Кринтер П. Задачи, программы, вычисления, результаты, М.:Мир, 1980.- 424 с.
- 4 Левенталь Л. Сейвил У.Программирование на языке ассемблера для микропроцессоров 8080 и 8085 .-М.:Радио и связь, 1987.- 448 с.

СОДЕРЖАНИЕ

Введение.....	3
I Команды микроЭВМ СМ-1800.....	3
1.1 Обозначения, используемые при написании команд.....	3
1.2 Список команд.....	6
1.3 Краткое описание команд.....	12
1.4 Коды операций команд (в порядке возрастания).....	16
2 Некоторые приемы программирования в кодах.....	18
2.1 Перемещение данных.....	19
2.2 Организация цикла.....	19
2.3 Использование подпрограмм.....	20
2.4 Перекодирование информации.....	21
Литература.....	23