

СИСТЕМА КОМАНД МИКРОПРОЦЕССОРА КР580ИК80

При написании программ для микро-ЭВМ программисту необходимо хорошо знать ее систему команд. Это означает, что программист должен помнить весь перечень команд, хорошо представлять себе те действия, которые будут выполняться микропроцессором при выполнении каждой из них.

Код операции любой команды (для однобайтовой команды — это просто код команды) в ЗУ микро-ЭВМ представляется двоичным 8-разрядным числом. Например, код команды пересылки из регистра С в регистр А будет иметь вид 01111001, код операции команды непосредственной записи 8-разрядного операнда в память записывается так: 0011 0110. а команда загрузки аккумулятора с непосредственной адресацией будет иметь код операции 0011 1010. Всего двоичным кодом можно представить 2^8 — 256 различных комбинаций. Почти столько же команд имеет и микропроцессор (некоторые комбинации двоичных 8-разрядных чисел не используются и поэтому команд несколько меньше, чем 256).

Естественно, что запомнить более 200 кодов команд, представленных в виде двоичных 8-разрядных чисел, т. е. в виде набора единиц и нулей, почти невозможно. Поэтому каждому коду команды ставится в соответствие мнемоническое название (мнемоника) команды, которая является сокращением от английских слов, описывающих ее действие. Мнемонический код команд позволяет легче запомнить их функции и значительно упрощает написание программ.

После мнемоники для двухбайтовых команд записывается 8-разрядный операнд, обозначаемый при описании системы команд через **D8**, а для трехбайтовых команд — 16-разрядный адрес или операнд, обозначаемые соответственно через **ADR** и **D16**. Через **M** обозначается ячейка памяти, адресуемая в соответствии с описанием команды. Так. первая из упомянутых выше команд в мнемоническом коде будет иметь вид **MOV A,C (MOVE REGISTER)**, вторая — **MVI M,D8 (MOVE TO MEMORY IMMEDIATE)**. а третья как **LDA ADR (LOAD DIRECT)**.

Названия регистровых пар в мнемонике команд даются в сокращенном виде с помощью первых букв их названия: так вместо **BC**, **DE** и **HL** записывается соответственно **B**, **D** или **H**. Например, команда увеличения на 1 содержимого регистровой пары **HL** записывается как **INX H**.

Все команды микропроцессора КР580ИК80А представлены в табл. 1. С помощью этой таблицы можно легко и быстро сопоставить мнемонике команды с ее кодом операции. Код операции каждой команды приведен здесь в верхней и нижней горизонтальных строках (младшие разряды) и в крайних левом и правом столбцах (старшие разряды) в шестнадцатиричном виде.

Далее для наглядности и упрощения записи все двоичные коды будем представлять в шестнадцатиричном виде. Для этого двоичный код числа делится на группы по 4 разряда. Для 8-разрядного кода операции или операнда таких групп будет две. а для 16-разрядного адреса или операнда — четыре. Четырехразрядным двоичным кодом можно представить любое десятичное число от 0 до 15. Обозначив эти величины цифрами от 0 до 9 и далее буквами латинского алфавита от А до F, мы получим шестнадцатиричные цифры. В табл. 2 приведено соответствие между десятичными, двоичными и шестнадцатиричными значениями величин. Например, двоичный код 1100 0011 можно представить в виде шестнадцатиричного числа **C3**, операнду или коду команды **B8**, записанному в шестнадцатиричной форме, будет соответствовать код 1011 1000, а адресу **F204** код 1111 0010 0000 0100.

Таким образом, зная мнемонике команды, например, **ORA C** из табл. 1 можно определить ее шестнадцатиричный код операции **B1**, что будет соответствовать двоичному коду 1011 0001.

При выполнении микропроцессором некоторых команд в регистре признаков **F** вырабатываются признаки состояния, при этом устанавливаются в **1** следующие биты регистра **F**.

КОДЫ КОМАНД МИКРОПРОЦЕССОРА КР580ИКВ0А

!	!	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	!
!	0	NOP	LXI	STAX	INX	INR	DCR	MVI	RLC	-	DAD	LDAX	DCX	INR	DCR	MVI	RRC	!
!			B,&	B	B	B	B	B,#			B	B	B	C	C	C,#		!
!	1	-	LXI	STAX	INX	INR	DCR	MVI	RAL	-	DAD	LDAX	DCX	INR	DCR	MVI	RAR	!
!			D,&	D	D	D	D	D,#			D	D	D	E	E	E,#		!
!	2	-	LXI	SHLD	INX	INR	DCR	MVI	DAA	-	DAD	LHLD	DCX	INR	DCR	MVI	CMA	!
!			H,&	*	H	H	H	H,#			H	*	H	L	L	L,#		!
!	3	-	LXI	STA	INX	INR	DCR	MVI	STC	-	DAD	LDA	DCX	INR	DCR	MVI	CMC	!
!			SP,&	*	SP	M	M	M,#			SP	*	SP	A	A	A,#		!
!	4	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	!
!			B,B	B,C	B,D	B,E	B,H	B,L	B,M	B,A	C,B	C,C	C,D	C,E	C,H	C,L	C,M	C,A
!	5	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	!
!			D,B	D,C	D,D	D,E	D,H	D,L	D,M	D,A	E,B	E,C	E,D	E,E	E,H	E,L	E,M	E,A
!	6	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	!
!			H,B	H,C	H,D	H,E	H,H	H,L	H,M	H,A	L,B	L,C	L,D	L,E	L,H	L,L	L,M	L,A
!	7	MOV	MOV	MOV	MOV	MOV	MOV	HLT	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	MOV	!
!			M,B	M,C	M,D	M,E	M,H	M,L		M,A	A,B	A,C	A,D	A,E	A,H	A,L	A,M	A,A
!	8	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADC	ADC	ADC	ADC	ADC	ADC	ADC	ADC	ADC	!
!			B	C	D	E	H	L	M	A	B	C	D	E	H	L	M	A
!	9	SUB	SUB	SUB	SUB	SUB	SUB	SUB	SBB	SBB	SBB	SBB	SBB	SBB	SBB	SBB	SBB	!
!			B	C	D	E	H	L	M	A	B	C	D	E	H	L	M	A
!	A	ANA	ANA	ANA	ANA	ANA	ANA	ANA	XRA	XRA	XRA	XRA	XRA	XRA	XRA	XRA	XRA	!
!			B	C	D	E	H	L	M	A	B	C	D	E	H	L	M	A
!	B	ORA	ORA	ORA	ORA	ORA	ORA	ORA	CMP	CMP	CMP	CMP	CMP	CMP	CMP	CMP	CMP	!
!			B	C	D	E	H	L	M	A	B	C	D	E	H	L	M	A
!	C	RNZ	POP	JNZ	JMP	CNZ	PUSH	ADI	RST	RZ	RET	JZ	-	CZ	CALL	ACI	RST	!
!			B	*	*	*	B	#	0			*		*	*	#	1	!
!	D	RNC	POP	JNC	OUT	CNC	PUSH	SUI	RST	RC	-	JC	IN	CC	-	SBI	RST	!
!			D	*	N	*	D	#	2			*	N	*		#	3	!
!	E	RPO	POP	JPO	XTHL	CPO	PUSH	ANI	RST	RPE	PCHL	JPE	XCHG	CPE	-	XRI	RST	!
!			H	*		*	H	#	4			*		*		#	5	!
!	F	RP	POP	JP	DI	CP	PUSH	ORI	RST	RM	SPHL	JM	EI	CM	-	CPI	RST	!
!			PSW	*		*	PSW	#	6			*		*		#	7	!

N - НОМЕР ПОРТА ВВОДА/ВЫВОДА ; ПРИМЕР: КОМАНДА STAX D ИМЕЕТ
& - ДВУХБАЙТОВЫЙ ОПЕРАНД - D16 ; КОД ОПЕРАЦИИ 12.
***** - ДВУХБАЙТОВЫЙ ОПЕРАНД - ADR ; КОД ОПЕРАЦИИ CA ПРИНАДЛЕЖИТ
- ОДНОБАЙТОВЫЙ ОПЕРАНД - DB ; КОМАНДЕ JZ ADR

Таблица 2

Десятичное значение	Двоичное значение	Шестнадцатиричное значение	Десятичное значение	Двоичное значение	Шестнадцатиричное значение
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

Бит Z — признак нуля, устанавливается, если результат выполнения команды равен 0

Бит S — признак знака, устанавливается, если результат выполнения команды отрицателен. При выполнении арифметических команд каждый двоичный операнд представляется как 7-разрядное двоичное число со знаком, записанным в старшем разряде. Единица в восьмом разряде соответствует отрицательному числу в дополнительном коде

Бит P - признак четности, устанавливается, если количество единиц в двоичном коде результата четное.

Бит C — признак переноса, устанавливается, если в результате сложения двух 8-разрядных чисел возникает перенос из старшего разряда или в результате вычитания возникает заем.

Бит AC — признак вспомогательного переноса, устанавливается, когда перенос возникает из четвертого разряда двоичного числа (*из разряда D3*). Этот признак используется при различных операциях с 4-разрядными операндами.

Функциональное описание команд приведено в табл. 3. В ней описываются действия, которые совершает микропроцессор при их выполнении. Таблица довольно компактна, так как в системе команд имеется много однотипных выполняющих одинаковые операции над операндами, хранящимися в различных регистрах. В этой же таблице показано расположение битов в регистре признаков **F**.

Перед дальнейшим изучением материала необходимо познакомиться с условными обозначениями, используемыми в табл. 3.

Вместо нескольких однотипных команд в таблице помещена одна обобщенная команда. В такой команде вместо обозначения конкретного регистра или регистровой пары используется обобщенное обозначение нескольких регистровых пар. Рядом, через точку с запятой, помещается условное описание действия команды. Подставляя вместо обобщенного обозначения названия конкретных, допустимых для этой команды регистров или регистровых пар, можно получить мнемонику нужной команды и описание ее работы. Стрелкой - обозначается направление пересылки данных при выполнении команды, а через **M** () обозначается ячейка памяти, адресуемая по содержимому, приведенному в скобках. Это может быть содержимое одной из регистровых пар регистра *SP* (указатели стека), или адрес *ADR*, записанный во втором и третьем байте команды с непосредственной адресацией.

Теперь кратко охарактеризуем группы команд микропроцессора.

Группа команд однобайтовых пересылок данных. Это наиболее многочисленная группа команд. С их помощью производится обмен данными между внутренними регистрами микропроцессора, а также между внутренними регистрами и ячейками памяти. Внутренние межрегистровые пересылки осуществляются с помощью группы команд **MOV R1, R**, где **R1** и **R** — стры микропроцессора или их содержимое. При выполнении этой команды содержимое регистра **R** пересылается в регистр **R1**, причем в регистре **R** сохраняется прежнее значение данных. Например, после выполнения команды **MOV C, A** данные из регистра **A** будут переписаны в регистр **C**. При этом в регистре **A** данные не изменяются. В качестве **R1** или **R** может быть также определена ячейка памяти, адрес которой находится в регистровой паре **HL**. Например, по команде **MOV C, M** произойдет пересылка данных из ячейки памяти с адресом, записанным в регистровой паре **HL** во, внутренний регистр **C** микропроцессора Содержимое ячейки памяти **M** при этом не изменится Другой пример: по команде **MOV M, E** данные из регистра **E** запишутся в ячейку памяти **M**, адресуемую через содержимое регистровой пары **HL**, а данные в регистре **E** останутся неизменными.

Для пересылок данных между аккумулятором и ячейками памяти в качестве адреса ячейки памяти может быть использовано также содержимое регистровых пар **BC** или **DE**. Тогда для записи в память данных из аккумулятора используются однобайтовые команды **STAX B** или **STAX D**, а при обратной пересылке **LDAX B** или **LDAX D**. Адрес ячейки памяти для обмена с аккумулятором можно задать также с помощью трехбай-

товых команд с непосредственной адресацией. В этом случае для записи данных из аккумулятора в память служит команда **STA ADR**, для обратной пересылки — **LDA ADR**.

СИСТЕМА КОМАНД МИКРОПРОЦЕССОРА КРЭВОИКВОА			
ОДНОБАЙТОВЫЕ ПЕРЕСЫЛКИ		ДВУХБАЙТОВЫЕ ПЕРЕСЫЛКИ	
MOV R1,R	; R ---> R1	LXI YZ,D16	; D16 ---> YZ
MVI R,DB	; DB ---> R	MHLD ADR	; HL ---> M(ADR) M(ADR+1)
STAX YZ'	; A ---> M(YZ)	LHLD ADR	; M(ADR) M(ADR+1) ---> HL
LDAX YZ'	; M(YZ) ---> A	PUSH YZ''	; YZ ---> M(SP-1) M(SP-2),
STA ADR	; A ---> M(ADR)		; SP-2 ---> SP
LDA ADR	; M(ADR) ---> A	POP YZ''	; M(SP) M(SP+1) ---> YZ,
SPHL	; HL ---> SP	(POP' PSM)	; SP+2 ---> SP
КОМАНДЫ ВВОДА И ВЫВОДА		ОБМЕН БАЙТАМИ	
IN N	; (N) ---> A	XCHG	; HL <--> DE
OUT N	; A ---> (N)	XTHL	; H <--> M(SP+1), L <--> M(SP)
АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ ОПЕРАЦИИ С ОДНИМ ОПЕРАНДОМ			
CMC''	; C ---> C	INR''' R	; R+1 ---> R
STC''	; 1 ---> C	DEC''' R	; R-1 ---> R
CMA	; A ---> A	INX YZ	; YZ+1 ---> YZ
DAA'	; ДЕСЯТИЧН. КОРРЕКЦИЯ	DCX YZ	; YZ-1 ---> YZ
АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ ОПЕРАЦИИ С ДВУМЯ ОПЕРАНДАМИ			
8-БИТОВЫЕ ОПЕРАЦИИ			
ADD' R	; A+R ---> A	ADI' DB	; A+DB ---> A
ADC' R	; A+R+C ---> A	ACI' DB	; A+DB+C ---> A
SUB' R	; A-R ---> A	SUI' DB	; A-DB ---> A
SBB' R	; A-R-C ---> A	SBI' DB	; A-DB-C ---> A
ANA' R	; A∧R ---> A	ANI' DB	; A∧DB ---> A
ORA' R	; A∨R ---> A	ORI' DB	; A∨DB ---> A
XRA' R	; A⊕R ---> A	XRI' DB	; A⊕DB ---> A
			16-БИТОВЫЕ ОПЕРАЦИИ
			DAD'' YZ ; HL+YZ ---> HL
КОМАНДЫ СДВИГА СОДЕРЖ. АККУМУЛЯТОРА		КОМАНДЫ ПЕРЕДАЧИ УПРАВЛЕНИЯ	
RLC''	; СДВИГ ВЛЕВО	PCHL	; HL ---> PC
RAL''	; СДВИГ ВЛЕВО ЧЕРЕЗ БИТ ПРИЗНАКА C	JMP ADR	; ADR ---> PC
RRC''	; СДВИГ ВПРАВО	J-CON ADR	; ADR ---> PC
RAR''	; СДВИГ ВПРАВО ЧЕРЕЗ БИТ ПРИЗНАКА C		
СПЕЦИАЛЬНЫЕ КОМАНДЫ		КОМАНДЫ ВЫЗОВА И ВОЗВРАТА ИЗ ПОДПРОГРАММЫ	
EI	; РАЗРЕШЕНИЕ ПРЕРЫВАНИЯ	CALL ADR	; PC ---> M(SP-1) M(SP-2),
DI	; ЗАПРЕЩЕНИЕ ПРЕРЫВАНИЯ	C-CON ADR	; ADR ---> PC
HLT	; ОСТАНОВ	RST X	; PC ---> M(SP-1) M(SP-2),
MOP	; ХОЛОСТАЯ ОПЕРАЦИЯ		; ADD ---> PC, ГДЕ X=0,1,...,7,
			; ADD СООТВ. РАВЕН 0H,0H,10H,18H,
			; 20H,28H,30H,38H
		RET	; M(SP) M(SP+1) ---> PC,
		R-CON	; SP+2 ---> SP
ФОРМАТ РЕГИСТРА F			
D7 D6 D5 D4 D3 D2 D1 D0			
B Z 0 AC 0 P 1 C			
УСЛОВНЫЕ ОБОЗНАЧЕНИЯ			
'	- КОМАНДА ОКАЗЫВАЕТ ВОЗДЕЙСТВИЕ НА ВСЕ ПРИЗНАКИ		
''	- КОМАНДА ОКАЗЫВАЕТ ВОЗДЕЙСТВИЕ НА ПРИЗНАК C		
'''	- КОМАНДА ОКАЗЫВАЕТ ВОЗДЕЙСТВИЕ НА ВСЕ ПРИЗНАКИ, КРОМЕ ПРИЗНАКА C		
R,R1	- СОДЕРЖИМОЕ РЕГИСТРОВ A, B, C, D, E, H, L ИЛИ ЯЧЕЙКИ ПАМЯТИ M(HL)		
YZ	- СОДЕРЖИМОЕ РЕГИСТРОВОЙ ПАРЫ BC, DE, HL ИЛИ РЕГИСТРА SP		
YZ+	- СОДЕРЖИМОЕ РЕГИСТРОВОЙ ПАРЫ BC ИЛИ DE		
YZ''	- СОДЕРЖИМОЕ РЕГИСТРОВОЙ ПАРЫ BC, DE, HL ИЛИ PSM		
SP	- СОДЕРЖИМОЕ УКАЗАТЕЛЯ СТЕКА ПЕРЕД ВЫПОЛНЕНИЕМ КОМАНДЫ		
DB	- 8-РАЗРЯДНЫЙ ОПЕРАНД (СОДЕРЖИМОЕ ВТОРОГО БАЙТА ДВУХБАЙТОВОЙ КОМАНДЫ)		
(N)	- СОДЕРЖИМОЕ ПОРТА ВВОДА ИЛИ ВЫВОДА С НОМЕРОМ N (N=0,1,...,255)		
D16	- 16-РАЗРЯДНЫЙ ОПЕРАНД (СОДЕРЖИМОЕ ВТОРОГО И ТРЕТЬЕГО БАЙТА КОМАНДЫ)		
ADR	- 16-РАЗРЯДНЫЙ АДРЕС В ТРЕХБАЙТОВОЙ КОМАНДЕ		
M()	- СОДЕРЖИМОЕ ЯЧЕЙКИ ПАМЯТИ (АДРЕС ЯЧЕЙКИ УКАЗАН В СКОБКАХ)		
-CON	- ЧАСТЬ МНЕМОНИКИ КОМАНДЫ, ОПРЕДЕЛЯЮЩАЯ УСЛОВИЕ ПЕРЕДАЧИ УПРАВЛЕНИЯ, ВЫЗОВА И ВОЗВРАТА ИЗ ПОДПРОГРАММЫ (-CON В МНЕМОНИКЕ ЗАМЕНЯЕТСЯ НА		

С помощью двухбайтовой команды **MVI R.D8** (**R** буквенное наименование регистра или ячейки памяти, адресуемой по содержимому **HL**) можно записать операнд в любой 8-разрядный регистр микропроцессора или ячейку памяти. Операндом здесь будет содержимое второго байта команды.

Команда **SPHL** позволяет занести адрес из регистровой пары **HL** в указатель стека **SP**.

Группа команд двухбайтовых пересылок данных. Трехбайтовые команды **LXI B.D16**, **LXI D.D16**, **LXI H.D16** служат для непосредственной записи в соответствующие регистровые пары 16-разрядного операнда **D16**.

Используя команды **SHLD ADR**, **LHLD ADA**, можно организовать пересылки данных между регистровой парой **HL** и ячейкой памяти, посредственно адресуемой по содержимому второго и третьего байта команды. Остальные команды этой группы осуществляют пересылки с адресацией по указателю стека **SP**. С помощью команд **PUSH B**, **PUSH D** и **PUSH H** содержимое регистровых пар **BC**, **DE** и **HL** запишется в стек, по команде **PUSH PSW** в стек засылаются данные из аккумулятора и регистра признаков

Команды **POP B**, **POP D**, **POP H** служат для пересылки 16-разрядного слова из ячеек памяти, адресуемых указателем стека **SP** в соответствующую пару регистров. Командой **POP PSW** данные из стека пересылаются в аккумулятор и регистр признаков **F**. Таким образом, команда **POP PSW** может изменять все биты регистра признаков **F**.

Группа команд ввода вывода микропроцессор имеет всего две команды для ввода-вывода данных. С помощью команды ввода **IN N** можно переписать байт данных в регистр **A** микропроцессора из одного из 256 портов ввода. Номер порта определяется вторым байтом команды. Аналогично по команде вывода **OUT N** байт данных из регистра **A** микропроцессора будет переписан в любой из 256 портов вывода, адресуемых вторым байтом команды.

Группа команд обмена. В этой группе также всего две команды: **XCHG** — команда для обмена содержимым между регистровыми парами **HL** и **DE**, и **XTHL** — команда для обмена содержимым между регистровой парой **HL** и ячейками памяти, адресуемых по указателю стека **SP**.

Группа команд арифметических и логических операций с одним операндом. С помощью команды **CMC** можно изменить значение бита признака переноса на противоположное, т. е. инвертировать признак переноса. Команда **STC** позволяет установить значение признака переноса в 1. Значения всех битов в регистре **A** можно инвертировать, применив команду **CMA**. Команда **DAA** предназначена для выполнения двоично-десятичного сложения. Очень часто при написании программ используются команды **INR R**, **DEC R**, **INX YZ**, **DCX YZ**, служащие для увеличения или уменьшения значения содержимого регистра, ячейки памяти или регистровой пары на единицу. Многие команды этой группы воздействуют на различные биты регистра признаков **F** (см. табл. 3).

Группа команд арифметических и логических операций с двумя операндами. Перед началом выполнения любой команды из этой группы один из операндов должен быть помещен в регистр **A**, а другой операнд (если команда одиобайтовая) в один из 8-разрядных внутренних регистров микропроцессора или ячейку памяти, адресуемую содержимым регистровой пары **HL**. В двухбайтовой команде значение второго операнда непосредственно задается во втором байте команды. Результат выполнения команды помещается в регистр **A**.

Команды **ADD R** или **ADI D8** позволяют сложить два операнда. Сложение двух операндов со значением бита переноса **C** происходит по команде **ADC R** или **ACI D8**. Вычитание из аккумулятора второго операнда и учет значения бита заема **C** производится соответственно командами **SUB R**, **SUI D8** **SBB R** или **SBI D8**.

Операции поразрядного логического умножения (операция **И**, обозначается знаком \wedge) содержимого аккумулятора с вторым операндом происходит при выполнении команды **ANA R** или **ANI D8**. При этом результатом выполнения команды является 8-разрядное двоичное число, отдельные разряды которого равны 1 только тогда, когда соответствующие разряды у обоих операндов также равны 1.

При выполнении поразрядного логического сложения (операция ИЛИ, обозначается знаком \vee) с помощью команд **ORA R** или **ORA D8** образуется двоичное число, отдельные разряды которого равны 1 в случае, когда соответствующие разряды какого-либо одного или обоих операндов так же равны 1.

Результатом' выполнения операции ИСКЛЮЧАЮЩЕГО ИЛИ (обозначается знаком \oplus) командами **XRA R** или **XRI D8** является байт, отдельные разряды которого равны 1 только тогда, когда соответствующие разряды операндов имеют противоположные значения.

Рассмотрим пример различных логических операций над двумя операндами.

I операнд	1 0 0 1 0 0 1 1
II операнд	1 1 0 0 1 1 1 0
Результат операции:	$\hat{\vee}$ 1 0 0 0 0 0 1 0 \oplus 0 1 0 1 1 1 1 1

После выполнения рассмотренных команд логической обработки двух операндов значения признаков **C** и **AC** регистра признаков **F** всегда равны 0.

Команды **CMP R** или **CPI D8** позволяют сравнивать два операнда. Сравнение происходит вычитанием из первого операнда, хранящегося в аккумуляторе, второго. Если в результате операции вычитания окажется, что операнды равны, то признак нуля **Z** устанавливается в 1. Если же значение операнда, хранимого в аккумуляторе, меньше значения второго операнда, то устанавливается в 1 признак переноса **C**.

Есть в системе команд микропроцессора команды **DAD B**, **DAD D**, **DAD H**, **DAD SP**, позволяющие сложить два 16-разрядных числа. Одно из этих чисел должно быть записано в регистровую пару **HL**, а другое — в регистровую пару **BC**, **DE**, **HL** или **SP**. Результат сложения помещается в пару **HL**.

Группа команд сдвигов содержимого аккумулятора. На рис. 1 схематически показано, как происходит сдвиг содержимого аккумулятора влево или вправо командами сдвига **RAL** и **RAR** и командами циклического сдвига **RLC** и **RRC**. В операциях сдвига участвует бит переноса **C** регистра признаков **F**. Под воздействием каждой из этих команд происходит сдвиг содержимого аккумулятора только на один разряд. Если необходимо сдвинуть содержимое аккумулятора на большее число разрядов, то команду сдвига необходимо повторить требуемое число раз.

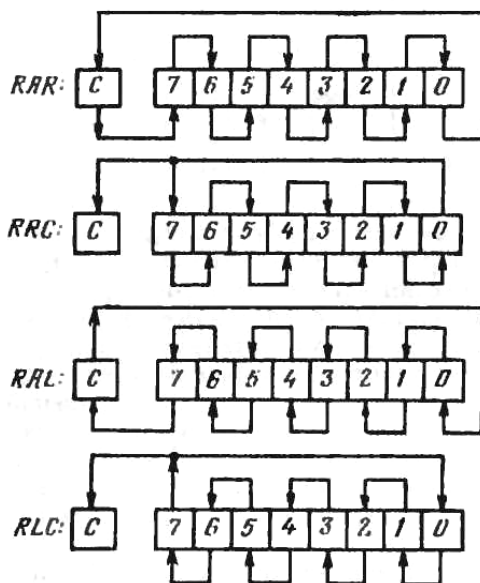


Рис. 1

Группа команд передачи управления и работы с подпрограммами. Эти команды играют особую роль в организации выполнения программ микро-ЭВМ. Пока в программе не встречаются команды этой группы, счетчик команд **PC** постоянно увеличивает свое значение, и микропроцессор выполняет команду за командой в порядке их расположения в памяти.

Порядок выполнения программы может быть изменен, если занести в регистр счетчика команд микропроцессора код адреса, отличающийся от адреса очередной команды. Это вызовет передачу управления работой микропроцессора другой части программы. Такая передача управления (или переход в программе) 'может быть выполнена с помощью трехбайтовой команды безусловного перехода — **JMP ADR**. Как только такая команда встретится в программе, в регистр счетчика команд **PC** микропроцессора запишется величина **ADR**. Таким образом, следующей командой, которую будет выполнять микропроцессор вслед за командой **JMP ADR**, будет команда, код операции которой записан в ячейке с адресом, равным значению **ADR**.

Безусловную передачу управления можно произвести также по команде **PCHL**, в результате выполнения которой произойдет передача управления по адресу, хранящемуся в регистровой паре **HL**.

Кроме команды безусловного перехода микропроцессор имеет восемь трехбайтовых команд условного перехода. При появлении команды условного перехода передача управления по адресу, указанному в команде, происходит только в случае выполнения определенного условия. Если условие не удовлетворяется, то выполняется команда, непосредственно следующая за командой условного перехода. Условия, с которыми оперируют команды условной передачи управления, определяются состоянием битов (разрядов) регистра признаков **F**:

NZ (NOT ZERO)	— ненулевой результат $Z=0$.
Z (ZERO)	— нулевой результат $Z=1$.
NC (NO CARRY)	— отсутствие переноса. $CY = 0$.
C (CARRY)	— перенос. $CY=1$.
PO (PARITY ODD)	— нечетный результат. $P=*0$,
PE (PARITY EVEN)	— четный результат, $P>>1$.
P (PLUS)	— число неотрицательное, $S=0$.
M (MINUS)	— число отрицательное, $S=*1$.

Эти условия проверяются соответствующими командами условного перехода: **JNZ ADR JZ ADR JNC ADR, JC ADR, JPO ADR, JPE ADR JP ADR. JM ADR.**

При написании программ, обычно, можно выделить одинаковые последовательности команд, часто встречающиеся в разных частях программы. Для того чтобы многократно не переписывать такие последовательности команд, их объединяют в так называемые *подпрограммы*. В любой части основной программы программист может поставить трехбайтовую команду безусловного вызова подпрограммы **CALL ADR**, во втором и третьем байте которой указывается адрес вызываемой подпрограммы. Выполнение команды **CALL ADR** начинается с побайтовой засылки в стек адреса следующей после этой команды ячейки памяти. Этот адрес называется адресом возврата из подпрограммы. Он необходим для того, чтобы по окончании выполнения подпрограммы вернуться к продолжению выполнения основной программы.

После записи в стек адреса возврата из подпрограммы в счетчик команд **PC** микропроцессора загружается величина **ADR**, т. е. адрес первой команды вызываемой подпрограммы. Таким образом, управление из основной программы передается на вызываемую подпрограмму.

Выполнение подпрограммы всегда заканчивается командой возврата из подпрограммы, например, однобайтовой командой безусловного возврата из подпрограммы **RET**. При этом содержимое стека, т. е. адрес возврата из подпрограммы пересылается из стека в регистр **PC** микропроцессора и управление вновь передается основной программе.

Кроме трехбайтовой команды безусловного вызова подпрограммы **CALL ADR**, в системе команд микропроцессора имеется восемь однобайтовых команд **RST 0 — RST 7** вызова подпрограмм, расположенных по фиксированному адресу. Появление в основной программе любой из этих команд вызывает запись в стек адреса возврата из подпрограммы и передачу управления на соответствующую ячейку памяти, где расположена первая команда подпрограммы. В табл. 4 дано соответствие между командами **RST 0 — RST 7** и шестнадцатиричными адресами ячеек памяти, куда передается управление при их выполнении.

Таблица 4

Команды	Адрес начала подпрограммы	Команды	Адрес начала подпрограммы
RST 0	0000	RST 4	0020
RST 1	0008	RST 5	0028
RST 2	0010	RST 6	0030
RST 3	0018	RST 7	0038

Кроме команды безусловного вызова и возврата из подпрограмм, в системе команд имеется восемь команд условного вызова подпрограмм и восемь команд условного возврата из подпрограмм, действие которых определяется, так же как и у команд условной передачи управления, состоянием регистра признаков **F**. Если условие для выполнения команды отсутствует, то вызов подпрограммы или возврат из нее не выполняется.

Группа специальных команд. Команда **NOP** этой группы не производит никаких операций, однако так как она выполняется за определенный отрезок времени, ее можно использовать в программах для задания временных интервалов. Появление в программе команды **HLT** ведет к останову выполнения программы. Продолжить выполнение программы можно только подачей сигнала сброс или запрос прерывания на соответствующие входы микропроцессора.