

составе пульта оператора ЭВМ и предназначен для ввода адреса программы, кодов программы и данных. Пульт оператора одержит набор тумблеров и клавиш, позволяющих оператору осуществлять ввод данных в БЭВМ, запуск программы на выполнение и управление режимами работы БЭВМ.

Регистр состояния (PS - Program State) - 16-разрядный регистр, хранит биты управляющие работой БЭВМ (работа, прерывание и пр.) и признаки результата. В актуальной программной реализации используются только 9 младших разрядов.

### 1.3 Система команд базовой ЭВМ

Классификация команд. БЭВМ способна исполнять точно определенный набор команд. При составлении программы пользователь ограничен этими командами. Полный перечень команд базовой ЭВМ приведен в таблице В.3. В зависимости от особенностей выполнения различных операций в БЭВМ команды можно разделить на четыре группы:

- безадресные команды;
- команды ввода-вывода;
- адресные команды;
- команды ветвления.

Выбор одного из типов команды осуществляется МПУ при помощи анализа старших четырех бит кода команды (биты с 12 по 15), которые называются кодом операции (КОП, Opcode - Operation code). Разработчики БЭВМ выбрали шесть форматов 16-битовых (однословных) команд с 4-битовым кодом операции (рис. В.2).

а) Безадресная команда

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП=0000				Расширение КОП											

б) Команда ввода-вывода

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП=0001				Приказ				Устройство							

в) Адресная команда с абсолютной адресацией

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

г) Адресная команда с относительной адресацией;

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

д) Команда с прямой (непосредственной) загрузкой операнда

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП				1	111		Число								

е) Команда ветвления

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП=1111				Расш. КОП				Смещение							

Рисунок В.2. Форматы команд

Безадресные команды выполняют различные действия без ссылок на ячейку памяти. Например, команда `CLA` предписывает ЭВМ очистить аккумулятор (записать в АС код нуля). Это команда обработки операнда, расположенного в конкретном месте, "известном" машине. Другой пример безадресной команды - команда `HLT`. Формат команды состоит из значения 0 в КОП и расширения кода операции (биты 0-11), которое задает необходимую операцию без использования явного указания ячейки памяти. Следует отметить, что безадресные операции могут использовать ячейки памяти неявно, например, команды `POP` или `RET`.

Команды ввода-вывода управляют обменом данными между процессором и внешними устройствами ЭВМ. Эти команды будут подробно рассмотрены в части 2.

## Система команд базовой ЭВМ

Код	Команда	Признаки <sup>1</sup>				Описание
		N	Z	V	C	
0XXX	Безадресные команды					
0000	NOP	-	-	-	-	Нет операции
0100	HLT	-	-	-	-	Останов
0200	CLA	*	*	0	-	0 → AC
0280	NOT	*	*	0	-	(^AC) → AC
0300	CLC	-	-	-	0	0 → C
0380	CMC	-	-	-	*	(^C) → C
0400	ROL	*	*	*	*	AC и C сдвигается влево. AC15 → C, C → AC0
0480	ROR	*	*	*	*	AC и C сдвигается вправо. AC0 → C, C → AC15
0500	ASL	*	*	*	*	AC сдвигается влево. AC15 → C, 0 → AC0
0580	ASR	*	*	*	*	AC сдвигается вправо. AC0 → C, AC15 → AC14
0600	SXTB	*	*	0	-	Расширение знака мл. байта AC7 → AC15...AC8
0680	SWAB	*	*	0	-	Обмен ст. и мл. байта AC7...AC0 ↔ AC15...AC8
0700	INC	*	*	*	*	AC + 1 → AC
0740	DEC	*	*	*	*	AC - 1 → AC
0780	NEG	*	*	*	*	^AC + 1 → AC
0800	POP	*	*	0	-	(SP)+ → AC
0900	POPF	*	*	*	*	(SP)+ → PS
0A00	RET	-	-	-	-	(SP)+ → IP
0B00	IRET	*	*	*	*	(SP)+ → PS, (SP)+ → IP
0C00	PUSH	-	-	-	-	AC → -(SP)
0D00	PUSHF	-	-	-	-	PS → -(SP)
0E00	SWAP	*	*	0	-	Обмен A и вершины стека
1XXX	Команды ввода-вывода					
10XX	DI	-	-	-	-	Запрет прерывания
11XX	EI	-	-	-	-	Разрешение прерываний
12XX	IN REG	-	-	-	-	Чтение из регистров ВУ
13XX	OUT REG	-	-	-	-	Запись в регистры ВУ
18XX	INT NUM	*	*	*	*	Программное прерывание с заданным вектором
XXXX	Адресные команды					
2XXX	AND M	*	*	0	-	M & AC → AC
3XXX	OR M	*	*	0	-	M   AC → AC
4XXX	ADD M	*	*	*	*	M + AC → AC
5XXX	ADC M	*	*	*	*	M + AC + C → AC
6XXX	SUB M	*	*	*	*	AC - M → AC
7XXX	CMP M	*	*	*	*	Установить флаги по результату AC - M
8XXX	LOOP M	-	-	-	-	M - 1 → M; Если M ≤ 0, то IP + 1 → IP
9XXX						Резерв
AXXX	LD M	*	*	0	-	M → AC
BXXX	SWAM M	*	*	0	-	M ↔ AC
CXXX	JUMP M	-	-	-	-	M → IP
DXXX	CALL M	-	-	-	-	SP - 1 → SP, IP → (SP), M → IP
EXXX	ST M	-	-	-	-	AC → M
FXXX	Команды ветвления					
F0XX	BEQ (BZS)	-	-	-	-	Переход если равенство (Z==1)
F1XX	BNE (BZC)	-	-	-	-	Переход если неравенство (Z==0)
F2XX	BMI (BNS)	-	-	-	-	Переход если минус (N==1)
F3XX	BPL (BNC)	-	-	-	-	Переход если плюс (N==0)
F4XX	BHIS (BCS)	-	-	-	-	Переход если выше или равно/перенос (C==1)
F5XX	BLO (BCC)	-	-	-	-	Переход если ниже/нет переноса (C==0)
F6XX	BVS	-	-	-	-	Переход если переполнение (V==1)
F7XX	BVC	-	-	-	-	Переход если нет переполнения (V==0)
F8XX	BLT	-	-	-	-	Переход если меньше (N⊕V==1 / N!=V)
F9XX	BGE	-	-	-	-	Переход если больше или равно (N⊕V==0 / N==V)
FAXX						Резерв

**Примечания:**

1. Значения в столбцах признаков результатов показывают, как изменится соответствующий признак в результате выполнения операции. «-» - команда не влияет на признак, «0» - признак сбросится, «\*» - значение признака установится по результату операции.

содержимое 25 ячейки выбирается из памяти в регистр данных (DR). Т.к. при обращении к памяти левая часть схемы остается незадействованной, то возможны одновременные с обращением к памяти другие операции с АЛУ и регистрами. Соответственно содержимое BR подается на левый вход АЛУ, правый вход АЛУ при этом закрыт, и содержимое BR (равное IP) увеличивается на 1 в АЛУ, результат этой операции попадает в IP. Счетчик команд теперь содержит адрес следующей исполняемой команды.

4) Для завершения цикла выборки команды (рис. В.8г) необходимо переслать код команды, на предыдущих шагах выбранный из памяти в регистр команд. Для этого содержимое регистра данных через правый вход АЛУ по фронту передается в коммутатор, а по спаду выход коммутатора записывается в регистр команд (CR). Теперь CR содержит код исполняемой команды для его дальнейшего декодирования и определения типа исполняемой команды и режимов адресации.

Цикл выборки адреса в адресной команде с абсолютной адресацией отсутствует, т.к. полностью сформированный адрес непосредственно записан в коде команды. БЭВМ переходит к следующему циклу.

Цикл выборки операнда. Для команды ADD 21 БЭВМ должна выбрать содержимое ячейки памяти с адресом 21 в DR, чтобы на следующем машинном цикле исполнения сложить содержимое регистра данных с аккумулятором. Напомним, что перед циклом выборки операнда код команды находится в регистре данных. По тактам происходит следующее:

1) Содержимое DR подается на правый вход АЛУ (рис. В.9а). Младшие 11 разрядов выхода АЛУ и коммутатора передаются в регистр адреса. На левый вход АЛУ в это время подается 0.

2) По адресу 021 в AR из памяти данные загружаются в DR (рис. В.9б).

Циклы выборки адреса и выборки операнда подробно описаны для каждого вида адресации на рис. В.10.

Код				Мнемоника	Описание	Реализация машинных циклов Address Fetch, Operand Fetch
11	10	9	8			
0	M	M	M	ADD 0ADDR ADD \$L	Прямая абсолютная	$DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	0	0	0	ADD (L)	Косвенная относительная	$SXT\_CR(0..7) \rightarrow BR,$ $BR + IP \rightarrow AR, MEM(AR) \rightarrow DR,$ $DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	0	0	1		Резерв	
1	0	1	0	ADD (L)+	Косвенная автоинкрементная (постинкремент)	$SXT\_CR(0..7) \rightarrow BR,$ $BR + IP \rightarrow AR, MEM(AR) \rightarrow DR, DR + 1 \rightarrow DR,$ $DR \rightarrow MEM(AR), DR - 1 \rightarrow DR,$ $DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	0	1	1	ADD -(L)	Косвенная автодекрементная (предедекремент)	$SXT\_CR(0..7) \rightarrow BR,$ $BR + IP \rightarrow AR, MEM(AR) \rightarrow DR, DR - 1 \rightarrow DR,$ $DR \rightarrow MEM(AR),$ $DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	1	0	0	ADD &N ADD (SP+N)	Косвенная относительная, со смещением (SP)	$SXT\_CR(0..7) \rightarrow BR,$ $BR + SP \rightarrow DR,$ $DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	1	0	1		Резерв	
1	1	1	0	ADD L ADD (IP+N)	Прямая относительная	$SXT\_CR(0..7) \rightarrow BR,$ $BR + IP \rightarrow DR,$ $DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	1	1	1	ADD #N	Прямая загрузка	$SXT\_CR(0..7) \rightarrow BR, BR \rightarrow DR$

Рисунок В.10 Циклы выборки адреса и операнда для различных режимов адресации