

5. ОДНОКРИСТАЛЬНЫЙ 8-БИТНЫЙ МП

Организация МП КР580. Типичным представителем 8-битных однокристальных МП является МП КР580ИК80А (далее для краткости — КР580). Данный МП, выполненный по NМОп-технологии, содержит примерно 5000 элементов и реализован в 40-выводном корпусе. Число базовых команд МП КР580 составляет 78, время выполнения команд для тактовой частоты 2 МГц лежит в диапазоне 2—9 мкс. Команды МП могут быть одно-, двух- и трехбайтными. Двух- и трехбайтные команды хранятся в соседних ячейках памяти. МП КР580 имеет четыре режима адресации:

1. Прямая адресация. В этом режиме второй и третий байты команды содержат исполнительный адрес команды, причем во втором байте — младшие разряды, а в третьем — старшие.

2. Регистровая адресация — для обращения к внутренним регистрам МП.

3. Непосредственная адресация, при которой в команде указывается 8- или 16-битный операнд.

4. Косвенная регистровая адресация.

Структурная схема МП КР580 приведена на рис. 3-10. функциональное назначение выводов МП следующее:

A15—A0 —шина адреса с тремя состояниями, обеспечивает адресацию памяти емкостью до 64К байт, адресацию 256 портов ввода и 256 портов вывода информации;

D7—D0 — двунаправленная шина данных с тремя состояниями, обеспечивает обмен информацией между МП, памятью и периферийными устройствами;

DBIN — выходной сигнал приема, указывает памяти и периферийным устройствам, что шина данных находится в режиме приема информации в МП;

WR — выходной сигнал выдачи, используется для управления выдачей информации из МП в память и периферийные устройства, активным является сигнал $WR=0$;

INT — входной сигнал прерывания, воспринимаемый МП после выполнения текущей команды или в режиме останова; прерывание не воспринимается МП, если он находится в режиме захвата или в режиме запрещенных прерываний, что обеспечивается установкой триггера разрешения прерываний в состояние 0 командой DI;

INTE — выходной сигнал разрешения прерываний, индицирует состояние триггера разрешения прерывания;

данный триггер сбрасывается в 0 после команды DI, а также после приема сигнала прерывания INT или сигнала сброса RESET;

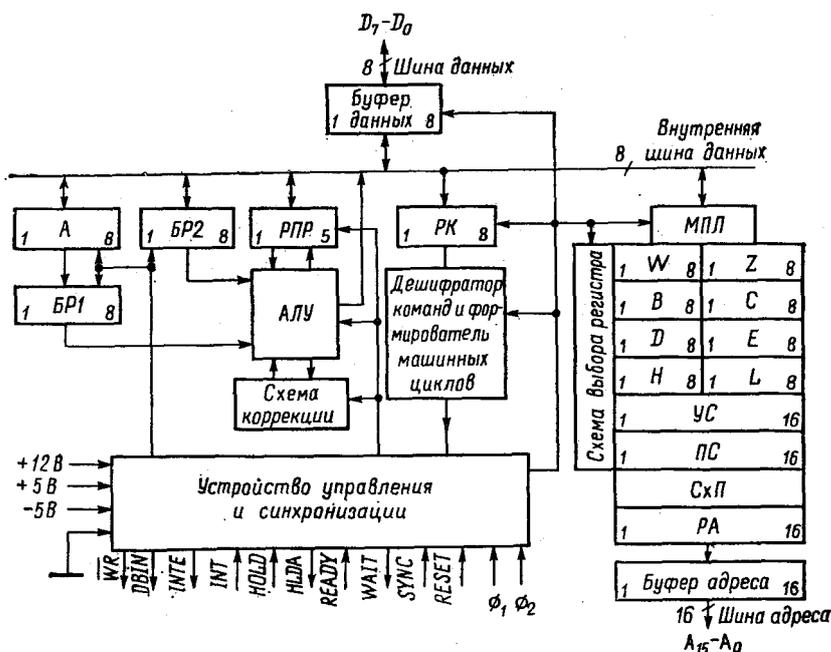


Рис. 3-10. Структурная схема МП КР580

HOLD — входной сигнал захвата, переводит МП в состояние захвата после завершения обмена данными между МП, памятью или периферийными устройствами в текущем машинном цикле; после захвата МП шины данных и адреса переходят в состояние высокого выходного сопротивления;

HLDA — выходной сигнал подтверждения состояния захвата МП;

READY — входной сигнал готовности, информирует МП, что данные из внешнего источника переданы на шину данных, синхронизирует работу МП с более медленнодействующими памятью или периферийными устройствами; при нулевом значении этого сигнала МП переходит в состояние ожидания T_w ;

WAIT — выходной сигнал ожидания, подтверждающий, что МП находится в состоянии ожидания T_w ;

RESET — входной сигнал сброса, обеспечивает установку в состояние 0 регистра команд, программного счетчика, триггеров разрешения прерывания и подтверждения захвата; при этом состояние остальных регистров не изменяется;

SYNC — выходной сигнал синхронизации, определяет начало машинного цикла;

ϕ_1, ϕ_2 — тактовые сигналы.

МП КР580 содержит шесть 8-битных РОН, которые обозначаются В, С, D, E, H, L и могут объединяться в 16-битные пары В и С; D и E; H и L. Регистры W, Z являются программно-недоступными и выполняют вспомогательные функции, например используются для временного хранения адресов. Содержимое программного счетчика (ПС) автоматически увеличивается на 1 при выборке каждого байта команды, при этом первый байт, определяющий код операции, всегда передается в регистр команд (РК). Указатель стека (УС) хранит адрес последней занятой ячейки в стековой памяти. Содержимое УС уменьшается перед записью кода в стек и увеличивается после чтения.

Обмен данными внутри МП осуществляется по внутренней 8-битной шине данных через двунаправленный мультиплексор (МПЛ). На регистр адреса (РА) коды передаются из регистровых пар, УС и ПС при обращении к памяти и периферийным

устройствам. Схема приращения (СхП) обеспечивает добавление 1 к содержимому регистров или вычитание 1 из него. АЛУ используется для выполнения всех арифметических и логических операций, сдвигов и т. п.

Арифметические операции выполняются в дополнительных кодах. Аккумулятор А является источником одного из операндов при выполнении бинарных операций и приемником результата. Буферные регистры БР1 и БР2 выполняют вспомогательные функции.

АЛУ после выполнения команды формирует признаки результатов, которые фиксируются в 5-битном регистре признаков РПР. АЛУ формирует следующие признаки: нулевого результата Z, знака S, переноса C из старшего бита, переноса AC из третьего бита аккумулятора, четности P. Признаки AC и C при выполнении операций над числами в десятичной системе счисления используются для коррекции результата.

Первый байт команды, выбранный из программной памяти, передается по внутренней шине данных на РК. Выход РК связан с дешифратором команд ДШК, который определяет тип выполняемой операции. Реализация любой команды складывается из машинных циклов, число которых может быть от одного (для самой короткой команды) до пяти (для самой длинной). Для передачи одного байта между МП и памятью или периферийным устройством требуется три состояния машинного цикла, а максимальное число состояний машинного цикла - пять. Время одного состояния определяется периодом следования тактирующих сигналов. Внутреннее преобразование данных в МП выполняется в состояниях T4, T5. Начало каждого машинного цикла идентифицируется синхросигналом SYNC. Минимальная длительность одного состояния T равняется 0,5 мкс для $F_{max} = 2$ МГц, но имеется три состояния, продолжительность которых может быть значительно больше: это состояние ожидания, захвата и ожидания при останове.

В МП КР580 имеется десять типов машинных циклов, которые могут быть при выполнении команды: выборка байта команды, чтение из памяти, запись в память, чтение из стековой памяти, запись в стековую память, ввод, вывод, обработка прерывания, останов, обработка прерывания при останове. При этом первым машинным циклом всегда является выборка команды. Для определения типа машинного цикла в первом состоянии каждого машинного цикла на шину данных передается 8-битный код, который запоминается во внешнем регистре и используется для формирования системных управляющих сигналов для обращения к памяти, периферийным устройствам. Рассмотрим назначение каждого разряда в коде слова состояния:

INTA (D0) — признак подтверждения прерывания, используется для разрешения передачи команды вызова подпрограммы обработки запроса прерывания; WO (D1) — признак записи-вывода, указывает, что в текущем машинном цикле будет выполняться запись в память или вывод кода в порт вывода (при WO = 0);

STACK (D2) — признак указывает, что на шину адреса передается адрес из указателя стека, используемый для адресации области стековой памяти;

HLTA (D3) — признак подтверждает выполнение команды останова HLT;

OUT(D4) — признак вывода, указывает, что шина адреса содержит адрес порта вывода, а шина данных будет содержать данные при сигнале WR == 0;

MI (D5) — признак указывает, что МП находится в цикле выборки первого

байта команды;

INP(D6) — признак ввода, указывает, что шина адреса содержит адрес порта ввода и входные данные передаются по шине данных при сигнале $DBIN = 1$;

MEMR(D7) — признак указывает, что шина данных будет использована для приема данных из памяти.

Диаграмма переходов при выполнении машинного цикла в МП КР580 представлена на рис. 3-11, на котором используются следующие обозначения Tw— состояние ожидания МП; Twн— состояние ожидания МП при выполнении команды останова HLT; Ti — состояние МП, $i = 1-5$.

Каждый машинный цикл содержит 3, 4 или 5 состояний в зависимости от типа выполняемой команды. Первое состояние T1 идентифицируется синхросигналом SYNC, который используется для загрузки слова состояния в регистр. Кроме того, в T1 на шину адреса передается адрес устройства, к которому будет обращение.

В состоянии T2 МП анализирует признак подтверждения останова HLTА и сигнал готовности READY и соответственно переходит в состояние Twн или Tw. Если имел место сигнал захвата HOLD, то внутренний триггер захвата устанавливается в состояние 1.

В состоянии T3 действия определяются типом машинного цикла. В цикле выборки МП интерпретирует код на шине данных как байт КОП, а в циклах чтения из памяти, записи в память, ввода и вывода информации осуществляет обмен данными.

Состояния T4 и T5 являются необязательными, так как используются для внутренних преобразований в МП.

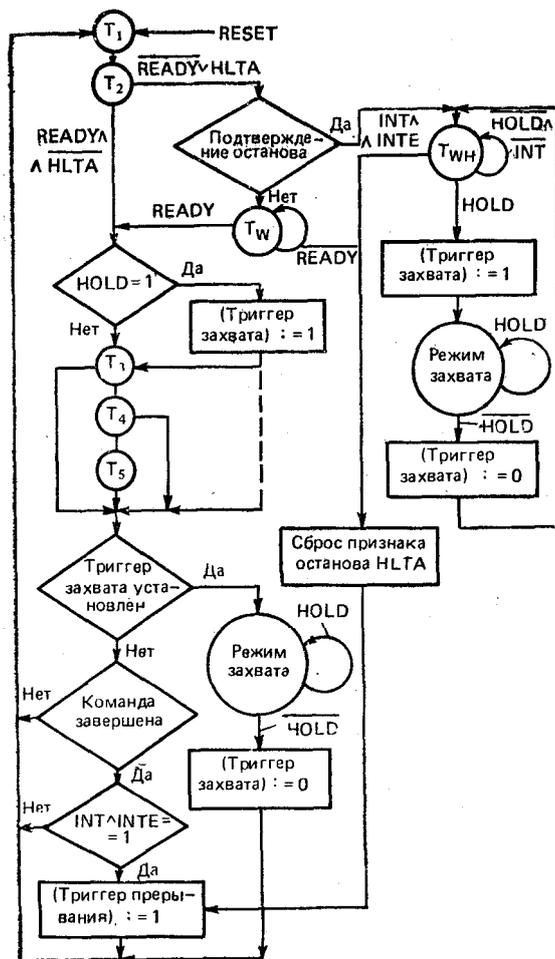


Рис. 3-11. Диаграмма переходов для машинного цикла МП КР580

Таким образом, МП может переходить в следующее состояние T1 после T3, T4 или T5.

формирование системных управляющих сигналов. Набор управляющих сигналов, таких как чтение из памяти (MEMR), запись в память (MEMW), ввод информации (I/OR), вывод информации (I/OW), подтверждение прерывания (INTA) обеспечивает прием и передачу кодов между МП, памятью и периферийными устройствами в определенные интервалы времени в соответствии с диаграммой переходов машинного цикла. Данные сигналы непосредственно не формируются микропроцессором КР580, для их формирования используются сигналы приема (DBIN) и записи (WR) из МП и необходимые признаки из слова состояния:

$$\begin{aligned} \overline{\text{MEMR}} &= \overline{\text{DBIN}} \wedge \overline{\text{MEMR}}; \\ \overline{\text{MEMW}} &= \overline{\text{WR}} \wedge \overline{\text{WO}}; \\ \overline{\text{I/OR}} &= \overline{\text{DBIN}} \wedge \overline{\text{INP}}; \\ \overline{\text{I/OW}} &= \overline{\text{WR}} \wedge \overline{\text{OUP}}; \\ \overline{\text{INTA}} &= \overline{\text{DBIN}} \wedge \overline{\text{INTA}} \end{aligned}$$

Слово состояния загружается в 8-битный регистр, например K589IP12, посредством сигнала синхронизации SYNC и тактирующего сигнала ϕ_2 уровня ТТЛ, формируемого генератором тактовых сигналов.

На рис. 3-12 показано формирование системных управляющих сигналов в соответствии с вышеприведенными выражениями. Другие признаки слова состояния могут использоваться при тестировании МП. Остановимся несколько подробнее на обработке запросов прерывания. Устройство, обслуживаемое МП в режиме прерывания, инициирует прерывание путем формирования на входе INT МП сигнала 1, который может возникнуть в любом машинном цикле команды. Однако, как видно из диаграммы переходов, текущая команда должна завершиться и только после этого МП переходит к машинному циклу подтверждения прерывания, который имеет ряд отличий от цикла выборки команды. Во-первых, в слове состояния дополнительно к признаку машинного цикла выборки команды МП содержится признак INTA, подтверждающий, что сигнал прерывания воспринят МП. Во-вторых, ПС не увеличивается на 1 в данном цикле, так как, увеличенный после выборки предыдущей команды, он должен быть запомнен в стековой памяти без изменения. В-третьих, вместо команды из программной памяти на шину данных передается код, представляющий собой однобайтную команду RST. Команда RST обеспечивает запоминание содержимого программного счетчика в стеке, и в ПС формируется один из восьми начальных адресов, в которых хранятся первые команды подпрограмм обслуживания прерываний.

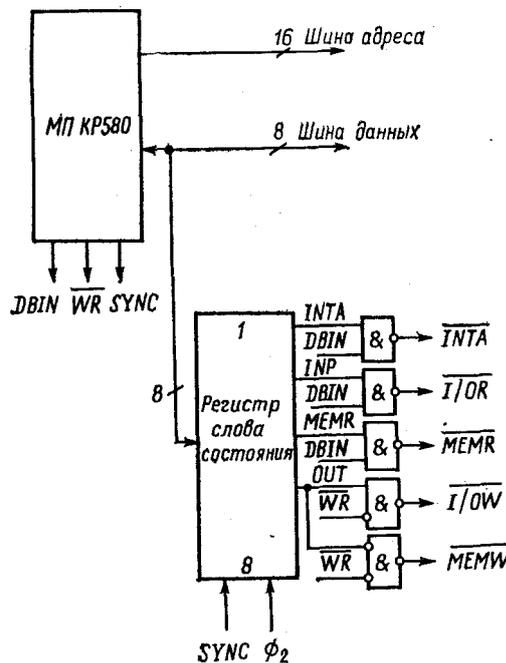


Рис. 3-12. Формирование системных управляющих сигналов

Если область памяти в 8 байт недостаточна для размещения подпрограммы, что, как правило, имеет место, то путем использования команд перехода ее можно расширить. Таким образом, первые 64 ячейки памяти (0000H — 003FH) зарезервированы для подпрограмм обработки прерываний.

Система команд МП КР580. Все множество команд данного МП можно подразделить на следующие 5 групп:

1. Команды пересылки кодов, обеспечивающие пересылку данных между регистрами или памятью и регистрами. Команды данной группы не формируют

признаков результатов операций.

2. Арифметические команды, обеспечивающие выполнение операций сложения и вычитания, изменения кодов на 1. Один операнд для бинарных операций хранится в аккумуляторе А, другой — в регистре или ячейке памяти, а результат помещается в аккумулятор. Такие операции, как умножение и деление,

Таблица 3-2

Система команд МП КР580

Группа команд	Мнемоника команды	Описание команды	Число байт	Признаки результата операции				
				Z	S	P	C	AC
Команды пересылки кодов	MOV D, S	$(D) \leftarrow (S)$	1	—	—	—	—	—
	MVI D, data	$(D) \leftarrow \text{data}$	2	—	—	—	—	—
	LXI rp, data	$(D) \leftarrow \text{data}$	3	—	—	—	—	—
	LDA addr	$(A) \leftarrow (\text{addr})$	3	—	—	—	—	—
	STA addr	$(\text{addr}) \leftarrow (A)$	3	—	—	—	—	—
	LHLD addr	$L \leftarrow (\text{addr}), H \leftarrow (\text{addr} + 1)$	3	—	—	—	—	—
	SHLD addr	$(\text{addr}) \leftarrow L, (\text{addr} + 1) \leftarrow (H)$	3	—	—	—	—	—
	LDAX rp	$A \leftarrow (rp)$	1	—	—	—	—	—
	STAX rp	$((rp)) \leftarrow A$	1	—	—	—	—	—
	XCHG	$(H) \leftrightarrow (D), (L) \leftrightarrow (E)$	1	—	—	—	—	—
Команды арифметической обработки	ADD S	$(A) \leftarrow (A) + (S)$	1	+	+	+	+	+
	ADI data	$(A) \leftarrow (A) + \text{data}$	2	+	+	+	+	+
	ADC S	$(A) \leftarrow (A) + (S) + (C)$	1	+	+	+	+	+
	ACI data	$(A) \leftarrow (A) + \text{data} + (C)$	2	+	+	+	+	+
	DAD rp	$(H, L) \leftarrow (H, L) + (rp)$	1	—	—	—	+	—
	SUB S	$(A) \leftarrow (A) - (S)$	1	+	+	+	+	+
	SUI, data	$(A) \leftarrow (A) - \text{data}$	2	+	+	+	+	+
	SBB S	$(A) \leftarrow (A) - (S) - (C)$	1	+	+	+	+	+
	SBI data	$(A) \leftarrow (A) - \text{data} - (C)$	2	+	+	+	+	+
	INR D	$(D) \leftarrow (D) + 1$	1	+	+	+	—	+
	INX rp	$(rp) \leftarrow (rp) + 1$	1	—	—	—	—	—
	DCR D	$(D) \leftarrow (D) - 1$	1	+	+	+	—	+
	DCR rp	$(rp) \leftarrow (rp) - 1$	1	+	+	+	—	+
DAA	Десятичная коррекция	1	+	+	+	+	+	
Команды логической обработки	ANA S	$(A) \leftarrow (A) \wedge (S)$	1	+	+	+	0	+
	ANI data	$(A) \leftarrow (A) \wedge \text{data}$	2	+	+	+	0	+
	XRA S	$(A) \leftarrow (A) \oplus (S)$	1	+	+	+	0	0
	XRI data	$(A) \leftarrow (A) \oplus \text{data}$	2	+	+	+	0	0
	ORA S	$(A) \leftarrow (A) \vee (S)$	1	+	+	+	0	0
	ORI data	$(A) \leftarrow (A) \vee \text{data}$	2	+	+	+	0	0
	CMP S	$(A) - (S)$	1	+	+	+	+	+
	CPI data	$(A) - \text{data}$	2	+	+	+	+	+
	RLC	$(A_{n+1}) \leftarrow (A_n), (A_0) \leftarrow (A_7), (C) \leftarrow (A_7)$	1	—	—	—	+	—
	RRC	$(A_n) \leftarrow (A_{n+1}), (A_7) \leftarrow (A_0), (C) \leftarrow (A_0)$	1	—	—	—	+	—
	RAL	$(A_{n+1}) \leftarrow (A_n), (C) \leftarrow (A_7), (A_0) \leftarrow (C)$	1	—	—	—	+	—
	RAR	$(A_n) \leftarrow (A_{n+1}), (C) \leftarrow (A_0), (A_7) \leftarrow (C)$	1	—	—	—	+	—
	STC	$(C) \leftarrow 1$	1	—	—	—	1	—
	CMC	$(C) \leftarrow \overline{(C)}$	1	—	—	—	+	—
CMA	$(A) \leftarrow \overline{(A)}$	1	—	—	—	—	—	

Продолжение табл. 3-2

Группа команд	Мнемоника команды	Описание команды	Число байт	Признаки результата операции				
				Z	S	P	C	AC
Команды передачи управления	JMP addr	$(PC) \leftarrow (addr)$	3	—	—	—	—	—
	Jcond addr	$(PC) \leftarrow (addr)$, если условие CCC в поле cond истинно	3	—	—	—	—	—
	CALL addr	$(PC) \leftarrow (addr)$	3	—	—	—	—	—
	Ccond addr	$(PC) \leftarrow (addr)$, если условие CCC в поле cond истинно	3	—	—	—	—	—
	RET	Возврат из подпрограммы	1	—	—	—	—	—
	Rcond	Возврат из подпрограммы, если условие CCC в поле cond истинно	1	—	—	—	—	—
Команды передачи управления	PCHL	$(PC) \leftarrow (H, L)$	1	—	—	—	—	—
	RSTn	Команда повторного старта	1	—	—	—	—	—
Команды ввода и вывода, обращения к стеку и управления МП	IN port	$(A) \leftarrow (port)$	2	—	—	—	—	—
	OUT port	$(port) \leftarrow (A)$	2	—	—	—	—	—
	PUSH rp	$((SP)) \leftarrow (rp)$	1	—	—	—	—	—
	PUSH PSW	$((SP)) \leftarrow (A)$ и (Z, S, P, C, AC)	1	—	—	—	—	—
	POP rp	$(rp) \leftarrow ((SP))$	1	—	—	—	—	—
	POP PSW	(Z, S, P, C, AC) и $(A) \leftarrow ((SP))$	1	+	+	+	+	+
	XTHL	$(H, L) \leftrightarrow ((SP))$	1	—	—	—	—	—
	SPHL	$(SP) \leftarrow (H, L)$	1	—	—	—	—	—
	EI	Разрешение прерывания	1	—	—	—	—	—
	DI	Запрещение прерывания	1	—	—	—	—	—
	HLT	Останов	1	—	—	—	—	—
	NOP	Пустая команда	1	—	—	—	—	—

выполняются программным путем с использованием подпрограмм. Отрицательные числа при выполнении арифметических операций необходимо преобразовывать в дополнительный код.

3. Логические команды реализуют операции логического сложения и умножения, исключаящего ИЛИ, инвертирования, левого и правого сдвигов и некоторые другие. Исходные операнды хранятся в регистрах или ячейках памяти, а результат помещается в аккумулятор.

4. Команды передачи управления, в число которых входят команды безусловной и условной передачи управления, обращения и выхода из подпрограмм. Данные команды не формируют признаков результатов операций.

5. Команды ввода и вывода информации, обращения к стековой памяти. Кроме того, в эту группу входит ряд команд управления работой МП.

Система команд микропроцессора КР580 приведена в табл. 3-2, в которой использованы следующие условные обозначения: D, S — приемник (Destination) или источник (Source) информации, в качестве которых используются регистры МП В, С, D, Е, H, L, аккумулятор А и ячейки памяти М (Memory); data—8- или 16-битное данное;

addr — 16-битный адрес памяти; (..J—содержимое ячейки памяти или регистра МП; (SP)—содержимое указателя стековой памяти; ((SP)) —содержимое ячейки стековой памяти; port — 8-битный адрес периферийного устройства; гр—регистровая пара В, С; D, Е; H, L или указатель стека SP; CCC — 3-битный код признака, используемого в командах передачи управления (см. табл. 3-1); п—номер команды повторного старта, п=0—7.

5.1. ОДНОКРИСТАЛЬНЫЙ 16-БИТНЫЙ МП

Наиболее известен 16-битный микропроцессор 8086¹ фирмы Intel. Он выполнен по НМОП-технологии, имеет 40-контактный корпус и одно напряжение питания +5 В. Номинальная частота однофазных сигналов синхронизации составляет 5 МГц. На кристалле размером 5,5 x 5,5 мм размещено 29 000 транзисторов. Для достижения номинальной производительности необходима память с циклом 500—800 нс и временем обращения при считывании 290—460 нс. Все регистры и двунаправленная мультиплексная шина AD адреса и данных 16-битные. Старшие 4 бита адреса мультиплексируются с сигналами состояния. Следовательно, длина физического адреса памяти составляет 20 бит, что обеспечивает адресное пространство 1М байт. Микропроцессор прямо адресует 256 портов ввода и 256 портов вывода и косвенно — 64К 8-битных портов ввода-вывода.

В архитектуре микропроцессора видна тенденция сохранения программной совместимости с 8-битным микропроцессором 8080. Хотя прямой совместимости вверх не достигнуто, ассемблерные программы 8080 легко трансформируются в программы 8086 (имеется специальная программа преобразования).

Отечественный аналог—микропроцессор К.1810ВМ86.

Функционально микропроцессор состоит из двух функционально автономных устройств: шинного интерфейса и операционного (исполнительного) устройства. Шинный интерфейс содержит секцию управления, 6-байтную очередь команд (эквивалент регистра команд), 4 сегментных регистра и программный счетчик (называемый указателем команд IP). Это устройство обеспечивает взаимодействие с остальными компонентами ЭВМ и нахождение в очереди команд достаточного числа байт. Как только в очереди фиксируются два свободных байта, инициируется обращение к программной памяти. Операционное устройство выполняют все преобразования данных, последовательно считывая и интерпретируя байты из очереди команд.

Микропроцессор допускает два режима системной конфигурации: максимальный и минимальный, которые определяются уровнем сигнала на входе MN/MX. В зависимости от выбранного режима изменяется интерпретация 8 управляющих сигналов. При минимальной конфигурации управляющие сигналы для памяти и периферийных устройств генерирует сам микропроцессор, а в максимальном режиме необходим специальный контроллер шины. Для реализации мультипроцессорных систем предусмотрен выходной сигнал LOCK блокировки шины. При низком уровне этого сигнала блокируется (запрещается) доступ к системной шине другим компонентам системы. Формированием этого сигнала управляет пользователь, применяя в программах префикс LOCK.

Память представляет собой массив 1М байт, причем любые два смежных байта образуют слово, т. е. слово может начинаться с четного или нечетного адреса. В первом случае слово передается за один цикл шины, а во втором требуется два цикла шины, поэтому целесообразно размещать слова данных по четным адресам. Длина команд составляет 1—6 байт, и их размещение в памяти на производительность не влияет. Микропроцессор оперирует 16-битными операндами, поэтому все адресные объекты должны иметь длину 16 бит. Следовательно, для формирования 20-битных адресов необходим дополнительный

механизм. Таким механизмом в микропроцессоре 8086 является сегментация памяти.

Можно считать, что память состоит из произвольного числа сегментов емкостью по 64К байт. Начальный адрес каждого сегмента (20 бит) кратен 16, т. е. содержит нули в 4 младших битах. Таким образом, сегменты могут начинаться на границах блоков по 16 байт и их начальные адреса имеют вид XXXX0.

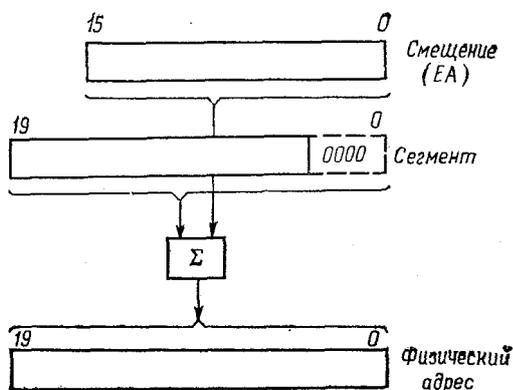


Рис. 3-13. Формирование физического адреса в МП 8086

Микропроцессор управляет 4 сегментами (программы (кода), данных, стека и дополнительного сегмента данных), начальные адреса которых без младших нулей хранятся в 16-битных сегментных регистрах CS, DS SS и ES. Команды с обращением к памяти формируют 16-битный исполнительный (эффективный) адрес EA, представляющий собой смещение в сегменте. Физический 20-битный адрес памяти равен сумме содержащегося EA и сегментного регистра (рис. 3-13). При адресации портов ввода-вывода сегментные регистры не используются.

В программной модели микропроцессора, приведенной на рис. 3-14, заштрихованы регистры, в известной степени эквивалентные регистрам микропроцессора 8080. Для удобства рассмотрения регистры программной модели разделены на три группы (файла).

Регистры общего назначения представлены 16-битными регистрами AX, BX, CX и DX, допускающими отдельную адресацию старших (H) и младших (L) 8-битных регистров. Двойственный характер РОН обеспечивает простую обработку 8- и 16-битных данных. Остальные регистры можно адресовать только как 16-битные. В основном все РОН в арифметических и логических операциях используются одинаково. Но имеется значительное число команд, которые специализируют некоторые РОН, что отражено в их названиях. Регистр AX выполняет функции аккумулятора, он является источником и получателем в операциях ввода-вывода, с ним связаны операции преобразования, десятичной коррекции, умножения и деления.

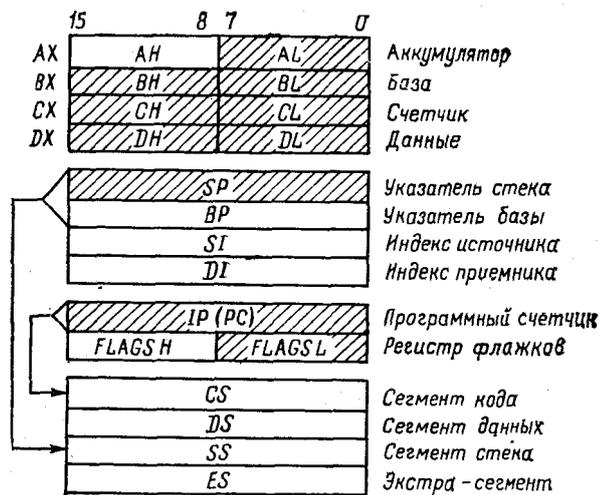


Рис. 3-14. Программная модель МП 8086

Регистр ВХ в некоторых командах участвует как регистр базового адреса и более или менее соответствует регистрам Н, L микропроцессора 8080. Регистр СХ используется как счетчик в операциях сдвигов, командах зацикливания и в операциях с цепочками байт и слов. Наконец, регистр ДХ неявно адресуется в операциях умножения и деления, а также содержит адрес портов ввода или вывода в режиме косвенной адресации.

Группа указательных и индексных регистров включает в себя 16-битные регистры SP, BP, SI и DI. Они обычно содержат внутрисегментные смещения и обеспечивают косвенную адресацию и динамическое вычисление исполнительных адресов. Гибкость таких вычислений достигается тем, что рассматриваемые регистры могут участвовать в арифметических и логических операциях как регистры общего назначения. Регистры SP (указатель стека) и BP (указатель базы) адресуют данные в текущем сегменте стека, а не в сегменте данных. Поэтому, если сегмент специально не определен, смещения в SP и BP относятся к текущему сегменту стека. Индексные регистры SI (источника) и DI (получателя) содержат смещения, которые обычно относятся к текущему сегменту данных.

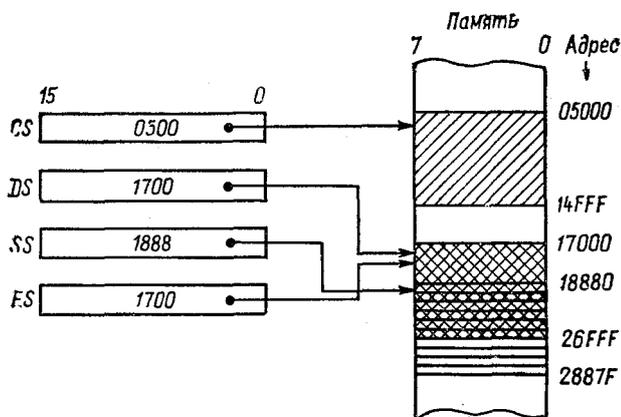


Рис. 3-15. Адресация сегментов в МП 8086

Четыре 16-битных сегментных регистра CS, DS, SS и ES используются для задания текущих сегментов по 64К байт, адресуемых при выполнении программы (рис. 3-15). Каждый регистр адресует конкретный текущий сегмент, что отражено в

их названиях: код, данные, стек, дополнительные данные. Сегментные регистры автоматически действуют в качестве базовых регистров при всех обращениях к памяти, т. е. исполнительный адрес суммируется с содержимым определенного сегментного регистра. Выборка всех команд осуществляется из текущего сегмента кода, а смещением служит содержимое указателя команд IP. Все исполнительные адреса данных (кроме адресов, формируемых через BP или SP, а в операциях с цепочками—через DI) суммируются с содержимым DS. Обращения к стеку, включая неявные действия при прерываниях, вызовах подпрограмм и возвратах, а также явные стековые включения и извлечения, осуществляются через SS. В операциях с цепочками и использованием DI физический адрес формируется через ES. Отметим, что любой команде с вычислением исполнительного адреса может предшествовать определяемый пользователем однобайтный префикс, в котором двухбитное поле явно определяет используемый сегмент.

Основное назначение сегментных регистров — динамическое перемещение программ в памяти, которое необходимо в мультипрограммной среде. По существу, для такого перемещения достаточно модифицировать содержимое сегментного регистра CS при условии, что программа сама не изменяет его содержимого. Кроме того, модифицируя содержимое DS, прикладная программа может манипулировать данными такого объема, который превышает емкость сегмента.

16-битный регистр состояния (регистр флажков) FLAGS разделен на две половины, младшая из которых FLAGSL полностью соответствует регистру флажков микропроцессора 8080. В старшей половине введены 4 новых флажка: OF — арифметического переполнения; DF — направления (определяет направление сканирования массива в операциях с цепочками); IF — прерывания (выполняет функцию маскирования внешних прерываний);

TF — прослеживания (обеспечивает выполнение программы по командам).

Система команд микропроцессора 8086 не является подмножеством системы команд микропроцессора 8080. Хотя сохранена преемственность большинства команд, но некоторые редко используемые команды (например, условные вызовы и возвраты) исключены и введено много новых команд.

Команды манипулируют одним или двумя операндами, а результат замещает один (часто любой) из операндов. Многие команды обрабатывают 8- и 16-битные операнды. Первым операндом в двухоперандной команде обычно является регистр или память, а во втором — регистр или константа (непосредственный операнд). В общем, нумерация операндов чисто условная и не отражает никакой направленности в передаче данных.

Общий формат двухоперандной команды, когда вторым операндом является регистр, показан на рис. 3-16. Первый байт содержит код операции и два однобитных поля: *d* (направление) и *w* (слово). Поле *d* определяет направление передачи, относящееся ко второму операнду — регистру, идентифицируемому 3-битным полем *reg* второго байта команды. Если *d* = 1, то направление передачи—в регистр, а если *d*=0, то направление—из регистра. Поле *w* определяет тип операнда: слово (*w* == 1) или байт (*w* = 0).

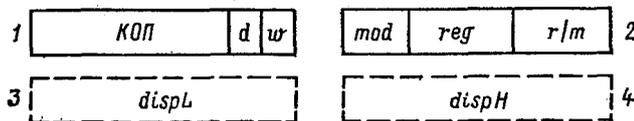


Рис. 3-16. Формат двухоперандной команды

Второй байт команды, называемый постбайтом, имеет три поля. Поле *reg* определяет второй операнд, обязательно находящийся в регистре. В этом поле кодируются следующие 8- или 16-битные регистры:

<i>reg</i> (<i>r/m</i>) . . .	000	001	010	011	100	101	110	111
8 бит	AL	CL	DL	BL	AH	CH	DH	BH
16 бит	AX	CX	DX	BX	SP	BP	SI	DI

Поле режима *mod* определяет используемый режим адресации. В частности, оно влияет на интерпретацию содержимого поля *r/m* (регистр/память) при нахождении первого операнда. Поле *mod* имеет следующий общий смысл

$$mod = \begin{cases} 00, 01, 10 & \text{— операнд содержится в памяти;} \\ 11 & \text{— операнд содержится в регистре.} \end{cases}$$

Когда *mod* = 11, поле *r/m* определяет 8- или 16-битный регистр в соответствии с приведенным выше кодированием.

Если *mod* = 00(01,10), это поле определяет, каким образом используется необязательное смещение *disp*:

$$mod = \begin{cases} 00, disp=0 & \text{(байты } disp\ L \text{ и } disp\ H \text{ отсутствуют);} \\ 01, disp=disp\ L & \text{(с расширением знака до 16 бит);} \\ 10, disp=disp\ H, disp\ L & \text{(два байта смещения).} \end{cases}$$

В этом же случае поле *r/m* определяет, как формируется исполнительный адрес *EA* команды:

<i>r/m</i>	<i>EA</i>
000	(BX) + (SI) + <i>disp</i>
001	(BX) + (DI) + <i>disp</i>
010	(BP) + (SI) + <i>disp</i>
011	(BP) + (DI) + <i>disp</i>
100	(SI) + <i>disp</i>
101	(DI) + <i>disp</i>
110	(BP) + <i>disp</i>
111	(BX) + <i>disp</i>

Приведенные правила имеют только одно исключение:

если *mod* = 00 и *r/m* = 110, то *EA* = *disp H*, *disp L*. Такой способ формирования *EA* называется прямой адресацией, т. е. адрес содержится в команде.

Следовательно, операнд в памяти может адресоваться прямо (16-битное смещение) или косвенно (8- или 16-битное смещение). Во втором случае допускается адресация посредством содержимого базового регистра (BP или BX), индексного регистра (SI или DI), а также суммы содержимого базового и индексных регистров. Всего получается 24 режима адресации: 3 способа интерпретации поля *mod* и 8 способов интерпретации поля *r/m*.

Режимы адресации спроектированы с учетом эффективной реализации языков высокого уровня. Например, к простой переменной можно обратиться в режиме прямой адресации, а к элементу массива — в режиме косвенной адресации посредством BX, SI и смещения. Режим адресации через BP предназначен для доступа к данным из сегмента стека, а не из сегмента данных. При реализации рекурсивных процедур и компиляторов языков высокого уровня с блоковой структурой данные часто запоминаются в сегменте стека.

Система команд насчитывает 113 базовых команд, объединяемых в

следующие группы:

Команды передачи данных. Команды данной ГРУППЫ подразделяются на четыре разновидности:

а) команды передачи данных между регистрами и памятью, включения в стек и извлечения из стека (адресуемого SP), а также команда обмена содержимым источника и приемника;

б) команды ввода, вывода, табличного преобразования;

в) команды загрузки исполнительного адреса в регистры общего назначения, а также загрузки 4-байтного адресного объекта в регистры-указатели (начальный адрес сегмента и смещение в сегменте); эти команды предоставляют пользователю возможность управлять механизмом адресации микропроцессора;

г) команды передачи содержимого регистра флажков (запоминание в памяти и загрузка из памяти регистра FLAGSL, включение в стек и извлечение из стека всего регистра FLAGS).

Арифметические команды. Микропроцессор имеет команды сложения, вычитания, умножения и деления двоичных знаковых и беззнаковых чисел. Произведение и делимое представляются числами двойной длины. Предусмотрены команды коррекции сложения и вычитания упакованных двоично-десятичных чисел, а также команды коррекции сложения, вычитания, умножения и деления неупакованных двоично-десятичных чисел. Интересно отметить, что для деления коррекция делимого осуществляется до выполнения операции.

Логические команды и команды сдвига. В эту группу входят поразрядные операции инверсии, конъюнкции, дизъюнкции и исключающего ИЛИ, а также команда TEST проверки операнда. Последняя выполняет поразрядную конъюнкцию, устанавливает по результату флажки, но никуда не загружает результат. Наряду с командами однобитных (статических) сдвигов имеются команды многобитных (динамических) сдвигов, в которых константа сдвига находится в регистре CL.

Команды передачи управления (переходы, вызовы, возвраты) имеют две разновидности: внутрисегментные («близкие»), целевой адрес которых, загружаемый в указатель команд IP, находится в текущем сегменте кода, и междусегментные («далекие»), целевой адрес которых определяет новые значения регистров CS и IP. Передачи управления могут быть прямыми (целевой адрес находится в команде) и косвенными (целевой адрес вычисляется с использованием стандартных режимов адресации). В 16 командах условных переходов проверяются отношения знаковых и беззнаковых чисел.

Имеются 4 команды управления циклами, которые рассчитаны на передачу числа повторений цикла в регистре CX (счетчик).

Команды обработки цепочек данных. Команды этой группы манипулируют цепочками данных, т. е. последовательностями байт или слов в памяти. Время выполнения таких операций, как передача цепочки, проверка и сканирование, значительно сокращается по сравнению с их программной реализацией.

Микропроцессор имеет двухуровневую векторную приоритетную систему прерываний с двумя линиями запросов прерываний. Вход NMI немаскируемого прерывания имеет больший приоритет, чем вход INTR маскируемых прерываний. Прерывания воспринимаются между командами, причем запросы INTR воспринимаются при условии, что IF=1. Реагируя на прерывание, микропроцессор

автоматически включает в стек содержимое регистров FLAGS, CS и IP. После этого сбрасываются биты IF и TF, а в регистры CS и IP загружаются новые значения векторов прерываний, что инициирует выполнение соответствующей подпрограммы обслуживания прерывания. Таблица векторов прерываний занимает первые 1024 байта физической памяти. Каждый вектор представлен 4 байтами и содержит новые значения, загружаемые в регистры CS и IP.

В мультипроцессорных системах с разделенными ресурсами необходим механизм управляемого доступа к ресурсам. Обычно такой механизм реализуется операционной системой, но требуются и некоторые аппаратные средства. Одно из простых средств, реализованных в микропроцессоре 8086, заключается в заблокированном обмене. Специальный однобайтный префикс LOCK, который может предшествовать любой команде, вызывает формирование сигнала LOCK, блокирующего доступ к шине на время выполнения команды от других компонентов системы.