

M1821BM85A

KM1821BM85

Восьмиразрядный процессор

Функциональный аналог
80C85A/80C85A-2 ф. Intel

Корпус 2123.40-6H

Условное графическое изображение

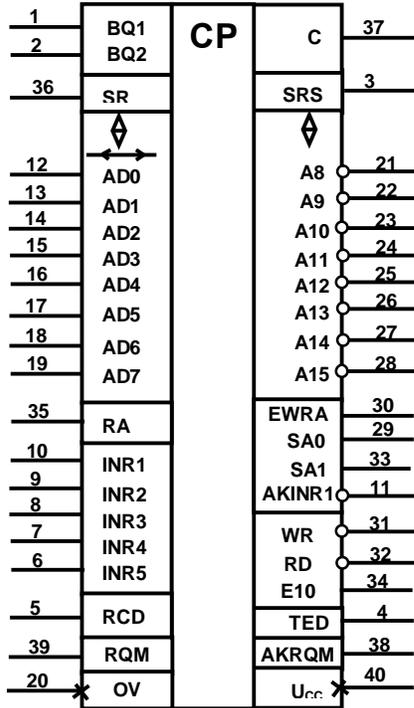


Таблица назначения выводов

Выводы	Обозначение	Наименование
1	BQ1	Вывод для подключения кварца RC-цепи, LC-цепи или внешнего генератора
2	BQ2	Вывод для подключения кварца RC-цепи
3	SRS	Выход «Начальная установка системы»
4	TFD	Выход «Передача последовательных данных»
5	RCD	Вход «Прием последовательных данных»
6	INR5	Вход «Прерывание 5»
7	INR4	Вход «Прерывание 4»
8	INR3	Вход «Прерывание 3»
9	INR2	Вход «Прерывание 2»
10	INR1	Вход «Прерывание 1»
11	AKINR1	Выход «Подтверждение прерывания»
12	AD0	Вход/выход «Адрес-данные»
13	AD1	Вход/выход «Адрес-данные»
14	AD2	Вход/выход «Адрес-данные»
15	AD3	Вход/выход «Адрес-данные»
16	AD4	Вход/выход «Адрес-данные»
17	AD5	Вход/выход «Адрес-данные»
18	AD6	Вход/выход «Адрес-данные»
19	AD7	Вход/выход «Адрес-данные»
20	OV	Общий вывод
21	A8	Выход «Адрес»
22	A9	Выход «Адрес»
23	A10	Выход «Адрес»
24	A11	Выход «Адрес»
25	A12	Выход «Адрес»
26	A13	Выход «Адрес»
27	A14	Выход «Адрес»
28	A15	Выход «Адрес»
29	SA0	Выход «Состояние»
30	EWRA	Выход «Разрешение записи адреса»
31	WR	Выход «Запись»
32	RD	Выход «Считывание»
33	SA1	Выход «Состояние»
34	E10	Выход «Разрешение обращения к устройствам ввода/вывода»
35	RA	Вход «Готовность»
36	SR	Вход «Установка процессора в исходное состояние»
37	C	Выход «Тактовая частота»
38	AKRQM	Выход «Подтверждение запроса прямого доступа к памяти»
39	RQM	Вход прямого доступа к памяти
40	Ucc	Питание

Основные электрические параметры при температуре 25°C

Параметр, режим измерения	Буквенное обозначение	Норма	
		не менее	не более
Ток потребления, мкА при $U_{CC}=5,5$ В	I_{CC}	–	100
Входной ток низкого и высокого уровня, мкА при $U_{CC}=5,5$ В, $U_{IL}=0$ В, $U_{IH}=U_{CC}$	I_{IL} I_{IH}	–	0,1
Выходной ток низкого и высокого уровня в состоянии «Выключено», мкА при $U_{CC}=U_{OH}=5,5$ В, $U_{IL}=0,8$ В, $U_{IH}=3,0$ В, $U_{OL}=0$	I_{OZL} I_{OZH}	–	0,5
Выходное напряжение низкого уровня, В при $U_{CC}=4,5$ В, $U_{IL}=0,8$ В, $U_{IH}=3,0$ В, $I_{OL}=2,0$ мА	U_{OL}	–	0,4
Выходное напряжение высокого уровня, В при $U_{CC}=4,5$ В, $U_{IL}=0,8$ В, $U_{IH}=3,0$ В, $I_{OH}=1,2$ мА	U_{OH}	3,0	–
Время задержки сигнала EWRA относительно сигналов A8...A15, нс, при $U_{CC}=4,5$ В, $U_{IL}=0...0,5$ В, $U_{IH}=4...4,5$ В, $C_L=150$ пФ, $f_{CO}=5,0$ МГц, $f_{CO}=3,0^*$ МГц	t_D (A-EWRA)	–	50 115*
Время установления выходных данных D0...D7 относительно сигнала WR, нс при $U_{CC}=4,5$ В, $U_{IL}=0...0,5$ В, $U_{IH}=4...4,5$ В, $C_L=150$ пФ, $f_{CO}=5,0$ МГц, $f_{CO}=3,0^*$ МГц	t_{SU} (WR-D)	230 420*	–
Время удержания выходных данных D0...D7 относительно сигнала WR, нс при $U_{CC}=4,5$ В, $U_{IL}=0...0,5$ В, $U_{IH}=4...4,5$ В, $C_L=150$ пФ, $f_{CO}=5,0$ МГц, $f_{CO}=3,0^*$ МГц	t_H (WR-D)	60 100*	–
Частота следования импульсов тактовых сигналов на выходе, МГц, при $U_{CC}=4,5$ В, $U_{IL}=0...0,5$ В, $U_{IH}=4...4,5$ В, $C_L=150$ пФ	f_C	–	5,0 3,0*

Технические характеристики

Напряжение питания 5,0 В ± 10 %
 Диапазон рабочих температур от – 45°C до + 85°C (– 60°C до + 85°C)*
 Разрядность данных – 8
 Разрядность адреса – 16
 Количество каналов обмена – 256 адресуемых 8-разрядных внешних устройств
 Количество команд – 123
 Объем адресуемой памяти – 64 кбайт
 Количество уровней прерывания – 5
 Частота следования импульсов тактовых сигналов на входе не более 10 МГц (7,2* МГц)
 Частота следования импульсов тактовых сигналов на выходе не более 5,0 МГц (3,6* МГц)
 Динамическая потребляемая мощность не более 121 мВт (110 * мВт)

Примечание – знаком «*» отмечены динамические параметры для микросхемы M1821BM85A.

Техническое описание работы схемы

Введение

Настоящее техническое описание микросхем M1821BM85A, B1821BM85A-4, KM1821BM85, KM1821BM85A, KP1821BM85, KR1821BM85A, KB1821BM85-4, KB1821BM85A-4 является приложением к техническим условиям БКО.347.489-01ТУ, БКО.349.017-01ТУ соответственно.

Микросхема является однокристалльным статическим 8-разрядным параллельным центральным процессорным устройством (микропроцессором), изготавливаемым по КМОП технологии, и предназначена для построения микро-ЭВМ, используемых в системах передачи и обработки информации.

Микросхема имеет десять программно доступных регистров, два из которых являются 16-разрядными, а остальные 8-разрядными; причём шесть из них могут объединяться в три 16-разрядные регистровые пары.

Микросхема имеет 16-разрядную шину адреса, позволяющую адресоваться к 64 килобайтам памяти, к такому же объёму внешнего стека и 256 внешним устройствам. Младший байт адреса и данные передаются по одной 8-разрядной двунаправленной шине «адреса/данные».

Микросхема выполняет 123 команды и имеет 5 уровней прерываний.

Система команд M1821BM85A, B1821BM85A-4 совместима с системой команд микросхемы 580BM80.

Схема и назначение выводов

Условно графическое обозначение приведено на рисунке 1.

Функциональное назначение выводов приведено в таблице 1.

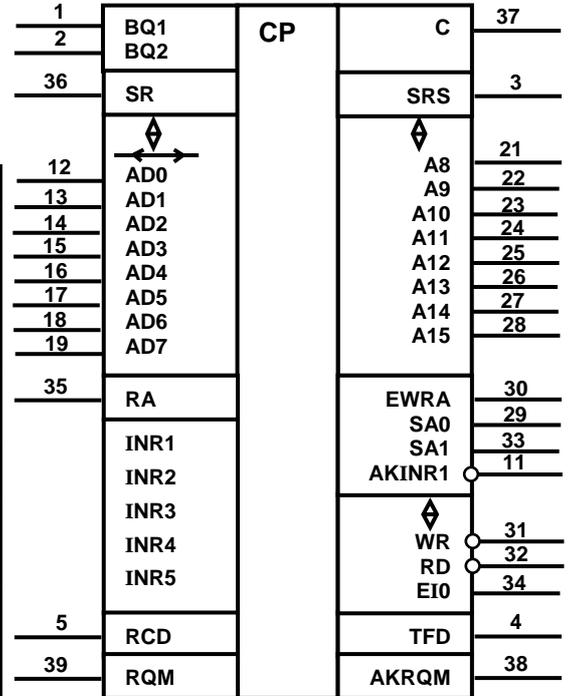


Рисунок 1

Таблица 1 – Назначение выводов

Номер вывода	Обозначение	Функциональное назначение				
1	BQ1	Выводы для подключения кварца, RC-цепи, LC-цепи или внешнего генератора.				
2	BQ2	Для обеспечения внутренней синхронизации входная частота делится на 2.				
3	SRS	Выход «Начальная установка системы» Высокий уровень на выходе SRS является ответным на низкий уровень входа SR, синхронен с тактовым сигналом и длится целое число тактовых периодов микропроцессора. Свидетельствует о том, что микропроцессор удерживается в исходном состоянии.				
4	TFD	Выход «Передача последовательных данных» Является выходом внутреннего триггера, в который при выполнении команды SIM передаются данные из старшего разряда аккумулятора.				
5	RCD	Вход «Приём последовательных данных» Опрашивается во время выполнения команды RIM и одноразрядные данные передаются в старший разряд аккумулятора.				
6	INR5	Вход «Прерывание 5» Вход немаскируемого прерывания с наивысшим приоритетом и фиксированным адресом подпрограммы прерывания (таблица 2). На него не действуют маски и разрешение прерывания. Воспринимается в то же время, что и INR1. Имеет защиту от дрейфа.				
7	INR4	Входы прерываний, которые могут быть отдельно маскированы при выполнении команды SIM. Уровни приоритетов, фиксированные адреса подпрограмм прерываний и вид управляющих сигналов соответствуют представленным в таблице 2. Воспринимается в то же время, что и INR1. Разрешаются и запрещаются с помощью команд EI, DI. Запрещаются сигналом SR и предшествующим прерыванием.	Вход «Прерывание 4»			
8	INR3		Вход «Прерывание 3»			
9	INR2		Вход «Прерывание 2»			
10	INR1	Вход «Прерывание 1» Вход немаскируемого прерывания с наименьшим приоритетом. Сигнал высокого уровня на входе соответствует подаче на микропроцессор сигнала прерывания общего вида. Микропроцессор внутренне фиксирует состояние этого входа во время, предшествующее последнему тактовому периоду команды, а также во время действия команды HLT. После приёма прерывания запрещается приращение программного счётчика. При отсутствии прерываний более высокого приоритета и при разрешённой системе прерываний выдаётся сигнал AKINR1 и управление передаётся программе обработки прерывания путём выполнения команд RST или CALL, содержащих начальный адрес подпрограммы. Действие сигнала INR1 разрешается и запрещается с помощью команд EI, DI, запрещается сигналом SR и предшествующим прерыванием.				
11	AKINR1	Выход «Подтверждение прерывания» Низкий уровень на выходе AKINR1 является ответным на сигнал INR1 и используется, после принятия сигнала INR1, вместо сигнала RD.				
12	AD0	36	SR	Вход-выход «Адреса/данные» Двунаправленная шина адресов/данных (младший байт адреса) с тремя состояниями. Адреса выставляются на шине в течение первого тактового периода каждого машинного цикла. В течение второго и третьего тактового периода является шиной данных.		
13	AD1					
14	AD2					
15	AD3					
16	AD4					
17	AD5					
18	AD6					
19	AD7					
20	0V	37	C	Общий		
21	A8			Выход «Адреса»		
22	A9			Шина адресов (старший байт) с тремя состояниями.		
23	A10			38	AKRQM	Выход «Подтверждение запроса прямого доступа к памяти» Сигнал высокого уровня на выход AKRQM (в ответ на сигнал RQM) показывает, что шины AD0...AD7, A8...A15 и выходы RD, WR, EIO находятся в третьем состоянии.
24	A11					
25	A12					
26	A13					
27	A14					
28	A15					
29	SA0					40
30	EWRA	Выход «Состояние» SA0 активен (низкий уровень) от начала до конца цикла чтения. Совместно с выходами SA1, EIO отражает внутреннее состояние микропроцессора (таблица 3)				
31	WR	31	RD	Выход «Разрешение записи адреса» Высокий уровень появляется во время первого тактового периода каждого машинного цикла. Спад сигнала EWRA используется для фиксации младшего байта адреса во внешнем регистре.		
32	RD			Выход «Запись» с тремя состояниями Низкий уровень на выходе указывает на готовность данных на шине «адреса/данные» для записи в выбранные УВВ или память.		
33	SA1	32	EIO	Выход «Считывание» с тремя состояниями Низкий уровень на выходе указывает на готовность шины «адреса/данные» для передачи данных выбранных из УВВ или памяти.		
34	EIO			Выход «Состояние» SA1 активен (низкий уровень) от начала до конца цикла записи.		
35	RA	39	RQM	Выход «Разрешение обращения к устройствам ввода-вывода» с тремя состояниями Сигнал EIO активен (высокий уровень) от начала до конца машинного цикла чтения или записи в УВВ.		
				Вход «Готовность» Опрашивается во втором тактовом периоде каждого машинного цикла. При наличии на входе RA сигнала низкого уровня микропроцессор формирует целое число тактовых периодов ожидания готовности данных на шине.		

Таблица 2

Наименование прерывания	Уровень приоритета	Адрес подпрограммы	Вид управляющего сигнала
INR5	1	24*	Фронт сигнала и высокий уровень пока не будет воспринят
INR4	2	3C*	Фронт сигнала
INR3	3	34*	Высокий уровень пока не будет воспринят
INR2	4	2C*	Высокий уровень пока не будет воспринят
INR1	5	**	Высокий уровень пока не будет воспринят

Примечания: * – шестнадцатеричный адрес; ** – адрес зависит от поступившего прерывания

Таблица 3

EIO	SA1	SA0	Состояние микропроцессора	Примечание
0	0	1	Запись в память	КОП – код операции команды
0	1	0	Чтение из памяти	УВВ – устройство ввода-вывода
1	0	1	Запись в УВВ	Z – третье состояние
1	1	0	Чтение из УВВ	X – состояние неопределено
0	1	1	Чтение КОП из памяти	
1	1	1	Обработка прерывания	
Z	0	0	Останов (действие команды HLT)	
Z	X	X	Запрос прямого доступа к памяти (действие сигнала RQM)	
Z	X	X	Установка процессора в исходное состояние (действие сигнала SR)	

Функциональные узлы

Микросхема микропроцессора имеет следующие функциональные узлы (рисунок 2) :

- арифметико-логическое устройство;
- аккумулятор;
- регистр признаков;
- регистр команд;
- дешифратор команд и шифратор циклов;
- блок регистров, состоящий из 6-ти регистров общего назначения (B, C, D, E, H, L), регистров W, Z, указателя стека SP, программного счётчика PC и регистра адреса со схемой инкремента/декремента;
- буфер адреса (A8...A15);
- буфер адреса/данных (AD0...AD15);
- блок синхронизации и управления;
- блок управления прерываниями;
- блок последовательного ввода и вывода.

Арифметико-логическое устройство (8-разрядное) выполняет все арифметические и логические операции, операции сдвига и управления, предусмотренные системой команд.

Аккумулятор представляет собой 8-разрядный программно доступный регистр данных, взаимодействующий с блоком регистров общего назначения и другими функциональными узлами микропроцессора, и предназначен для хранения результатов операций арифметико-логического устройства или данных при вводе/выводе и обмене с другими функциональными узлами микропроцессора.

Регистры:

– регистр временного хранения – представляет собой 8-разрядный вспомогательный регистр и используется при выполнении некоторых команд только в течение времени исполнения этих команд, будучи недоступным для использования извне, помимо этих команд;

– программно доступный регистр признаков – предназначен для внутренней фиксации дополнительных характеристик результатов операций и состояний арифметико-логического устройства. В регистр входят семь триггеров признаков :

- * триггер знака (S);
- * триггер переноса (CY);
- * триггер вспомогательного переноса (AC);
- * триггер нуля (Z);
- * триггер чётности (P);
- * триггер переполнения (V);
- * триггер вспомогательного знака (AS).

– 8-разрядный регистр команд – используется для хранения выбранной команды для дешифратора команд и шифратора машинных циклов.

Дешифратор команд и шифратор машинных циклов осуществляет дешифрацию кодов команд, поступающих из регистра команд, и производит установку счётчиков шифратора машинных циклов в соответствии с этими кодами.

Блок регистров (программно доступных, кроме W и Z) предназначен для хранения и выдачи различной информации, участвующей в процессе выполнения команд :

- 16-разрядный регистр указателя стека (SP) – предназначен для хранения адреса ячейки стека, к которой было произведено последнее обращение;
- 16-разрядный регистр программного счётчика (PC) – предназначен для хранения адреса следующей выполняемой команды;
- 16-разрядный регистр адреса со схемой инкремента/декремента – позволяет увеличивать или уменьшать на один содержимое регистра адреса;
- шесть 8-разрядных регистров общего назначения (B, C, D, E, H, L) – каждый из них можно использовать для хранения данных независимо от остальных, регистры могут объединяться в регистровые пары (B – C; D – E; H – L) и образовывать 16-разрядные регистры данных или адреса;
- два 8-разрядных регистра (W, Z) – используются для запоминания промежуточного адреса, при выполнении команд перехода к подпрограммам, команд возвратов и команд ветвления.

Буфера :

– буфер старших разрядов адреса – представляет собой 8-разрядный выходной формирователь с тремя состояниями, переключаемый в третье высокоимпеданное состояние («Выключено») во время действия сигналов SR, RQM или команды «Останов»;

– буфер адреса/данных – представляет собой 8-разрядный входной-выходной формирователь с тремя состояниями, предназначенный для выдачи либо младших разрядов, либо приёма-выдачи данных переключаемый в третье высокоимпеданное состояние («Выключено») во время действия сигналов SR, RQM или команды «Останов». В первом тактовом периоде машинного цикла буфером адреса/данных выводятся восемь младших разрядов адреса, во втором и третьем периодах производится ввод или вывод информации.

Блок синхронизации и управления обеспечивает внутреннюю синхронизацию микропроцессора и выдачу внешних сигналов, необходимых для работы с микропроцессором других устройств. Внутренняя синхронизация микропроцессора осуществляется от встроенного на кристалл тактового генератора, возбуждение которого задаётся внешним кристаллическим кварцевым резонатором, RC-цепью или LC-цепью, подключённым к выводам BQ1, BQ2 (предусмотрена возможность подключения внешнего источника тактовых сигналов к выводу BQ1). Возбуждаемая частота внутренними схемами делится на 2 и используется для синхронизации узлов как самого микропроцессора, так и внешних устройств системы с использованием вывода С.

Блок управления прерываниями переключает микропроцессор с выполнения одной программы на выполнение другой с помощью внешних сигналов прерывания и позволяет вводить пять уровней прерываний (таблица 2).

Блок последовательного ввода и вывода управляется командой RIM при вводе последовательных данных и командой SIM при выводе последовательных данных и осуществляет ввод одноразрядных данных от входа RCD в старший разряд аккумулятора или вывод одноразрядных данных от старшего разряда аккумулятора на выход TFD.

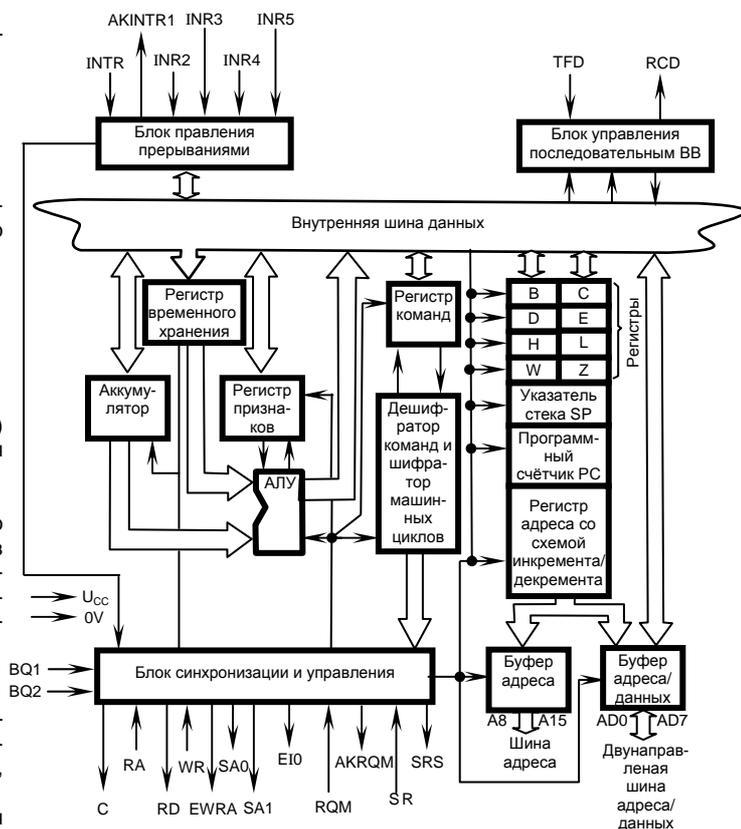


Рисунок 2

Работа микропроцессора при выполнении команд

Выполнение любой программы микропроцессором производится посредством операций записи или считывания. Каждая операция записи или считывания выполняется в течение машинного цикла М. Выполнение команды состоит в том, что микропроцессор обрабатывает последовательность машинных циклов М, количество которых может быть от одного до пяти. В свою очередь каждый машинный цикл содержит от трёх до шести тактовых периодов Т, причём каждый тактовый период соответствует определённому состоянию микропроцессора.

Сущность и последовательность машинных циклов определяется кодом операции команды, полученным в первом машинном цикле М1. Из таблицы 6 видно, что ни одна команда не содержит более 5-ти машинных циклов, при этом любой машинный цикл будет одним из представленных в таблице 4. Любой из указанных в таблице 4 машинных циклов однозначно определён состоянием выводов E10, SA0, SA1, RD, WR, AKINR1. Каждый машинный цикл обычно состоит из трёх тактовых периодов, за исключением машинных циклов чтения кода операции, которые могут быть длиной от 4-х до 6-ти тактовых периодов. Реальное число тактовых периодов при выполнении какой-либо команды определяется выполняемой командой, количеством тактовых периодов в цикле чтения кода операции и числом тактовых периодов ожидания, которые формируются, если на входе «Готовность» низкий уровень. Состояние микропроцессора и соответствующие состояния выводов приведены в таблице 5. На рисунке 3 представлена упрощённая диаграмма работы микропроцессора в ходе машинного цикла с учётом действия сигналов «Готовность» (RA), «Запрос прямого доступа к памяти» (RQM), «Установка процессора в исходное состояние» (SR) и прерываний.

Машинный цикл	E10	SA1	SA0	RD	WR	AKINR1
Чтение кода операции команды	0	1	1	0	1	1
Чтение из памяти	0	1	0	0	1	1
Запись в память	0	0	1	1	0	1
Чтение из порта	1	1	0	0	1	1
Запись в порт	1	0	1	1	0	1
Обработка прерывания INR1	1	1	1	1	1	0
Обработка прерывания INR5, INR4, INR3, INR2	1	1	1	1	1	1
*HLT	Z	0	0	Z	Z	1
*DAD, RDEL, DSUB, ARHL, LDSI, LDHI	0	1	0	1	1	1

* – машинные циклы, при которых не используется шина данных (для LDHI – только M3)

HLT – действие команды HLT
DAD – действие команды DAD и др.
«0» – низкий уровень сигнала
«1» – высокий уровень сигнала
Z – третье состояние («Выключено»)

Тактовые периоды	SA0, SA1	E10	A8...A15	AD0...AD7	RD, WR	AKINR1	EWRA
T1	X	X	X	X	1	1	1*
T2	X	X	X	X	X	X	0
T ожидания	X	X	X	X	X	X	0
T3	X	X	X	X	X	X	0
T4	1	0**	X	Z	1	1	0
T5	1	0**	X	Z	1	1	0
T6	1	0**	X	Z	1	1	0
T _{SR}	X	Z	Z	Z	Z	1	0
T _{HLT}	0	Z	Z	Z	Z	1	0
T _{EQM}	X	Z	Z	Z	Z	1	0

* – за исключением : M2, M3 команд DAD, RDEL, DSUB; M2 команды ARHL; M3 команд LDSI, LDHI
** – E10 = 1 в T4...T6 для машинного цикла обработки прерываний
X – неопределённое состояние (либо «0», либо «1» в зависимости от вида машинного цикла)
Z – состояние «Выключено» (третье состояние)
T_{SR} – установка в исходное состояние (действие сигнала SR)
T_{HLT} – действие команды «Останов»
T_{EQM} – действие сигнала RQM

Машинный цикл чтения кода операции

На рисунке 4 представлена типовая временная диаграмма машинного цикла М1 – чтения кода операции.

Состояние выводов E10 = 0, SA1 = 1, SA0 = 1 свидетельствует о том, что этот машинный цикл является циклом чтения кода операции из памяти. Адрес ячейки памяти выставляется на адресных шинах и удерживается в течение T1 на шинах AD0...AD7 и в течение T1, T2, T3 – на шинах A8...A15 (для правильного обращения к внешнему устройству памяти целесообразно фиксировать состояние шин AD0...AD7 во внешней защёлке, используя в качестве управляющего сигнал EWRA).

Низкий уровень сигнала RD в начале второго тактового периода T2, разрешает подключение выбранной ячейки памяти к шине данных AD0...AD7. Шина данных AD0...AD7, начиная от середины тактового периода T2 и до второй половины тактового периода T3, является входной и подключена к регистру команд микропроцессора, поэтому содержимое ячейки памяти копируется в регистр команд. В середине тактового периода T3 сигнал RD принимает высокий уровень и отключает устройство памяти. В T4 микропроцессор декодирует код операции команды и либо формирует тактовые периоды T5, T6, если M1 декодированной команды состоит из 6-ти тактовых периодов (рисунок 4), либо переходит к следующему машинному циклу, если M1 декодированной команды состоит их 4-х тактовых периодов (рисунок 5). На рисунке 5 также показано действие сигнала «Ожидание» (RA). Из рисунков 3 и 5 видно, что когда микропроцессор находится в T2, то он опрашивает состояние входа RA. Если на входе RA высокий уровень, микропроцессор переходит к T3. Если на входе RA низкий уровень, то микропроцессор формирует состояние «Ожидание», которое будет сохраняться целое число тактовых периодов, пока на входе RA не установится высокий уровень. Необходимо отметить, что «растягивание» временной диаграммы путём формирования тактов «Ожидание» полностью сохраняет состояние микропроцессора.

Машинный цикл чтения из памяти

Рисунок 6 демонстрирует временную диаграмму машинных циклов чтения из памяти : на рисунке слева – без состояния «Ожидание», справа – с одним состоянием «Ожидание». Машинный цикл чтения из памяти всегда заканчивается тактовым периодом T3, после которого следует тактовый период T1 следующего машинного цикла. Синхронизация сигналов машинного цикла чтения из памяти аналогична машинному циклу чтения кода операции команды за исключением состояния вывода SA0 = 0. Считанные данные помещаются в любой внутренний регистр, определяемый командой.

Машинный цикл записи в память

На рисунке 7 показана временная диаграмма машинных циклов записи в память : слева – без состояния «Ожидание», справа – с состоянием «Ожидание». Так же как и машинный цикл чтения из памяти, машинный цикл записи в память всегда заканчивается тактовым периодом T3, после которого следует тактовый период T1 следующего машинного цикла. Состояние выводов SA1 = 0, SA0 = 1, E10 = 0 характеризуют циклы, как циклы записи в память. Вывод WR изменяет своё состояние аналогично выводу RD. Записываемые данные помещаются на шину адреса/данные в начале T2 и удерживаются на ней до конца тактового периода T3. Сигнал WR обеспечивает запись данных в выбранную ячейку памяти.

Машинные циклы чтения из порта и записи в порт

На рисунках 8 и 9 показаны временные диаграммы выполнения команды OUT – вывод данных и команды IN – ввод данных, которые используют машинные циклы записи в порт и чтения из порта. Из сопоставления машинных циклов M2 и M3 команды IN видно, что единственное отличие машинного цикла чтения из памяти от машинного цикла чтения из порта состоит в состоянии вывода E10, а именно : E10 = 0 – для памяти и E10 = 1 – для порта. Подобное обстоятельство имеет место и для машинного цикла записи в порт. Ещё одно обстоятельство отличает машинные циклы записи и чтения порта : 8-разрядный адрес, используемый для адресации порта, задаётся во 2-м байте команды и выставляется как на AD0...AD7, так и на A8...A15.

Сигнал «Готовность» действует в машинных циклах чтения из порта и записи в порт аналогично ранее описанному.

Машинные циклы обработки прерываний

Процедура аппаратной обработки прерывания заключается в приёме сигнала прерывания, установления его достоверности, запрещения последующих прерываний путём сброса флага разрешения прерываний, помещения в область стековой памяти состояния программного счётика и формирования адреса подпрограммы.

Прерывания действуют по определённом приоритету, который определяет очерёдность их обработки, если поступило несколько прерываний (таблица 2). Однако имеется возможность путём использования в подпрограмме прерывания команды EI (разрешение прерываний) приступить к обработке прерывания с низким приоритетом ещё до завершения выполнения подпрограммы прерывания с более высоким приоритетом.

Существуют 3 различных типа входов прерываний :

- входы INR1, INR2, INR3 чувствительны к высокому уровню сигнала;
- вход INR4 чувствителен к фронту сигнала, устанавливающему в активное состояние внутренний триггер, который остаётся в активном состоянии до тех пор, пока прерывание не будет обработано, а затем сбрасывается автоматически (а также командой SIM или сигналом SR), при этом маска прерывания INR4 не действует на триггер, т. е. триггер может быть установлен даже тогда, когда прерывание INR4 замаскировано;
- вход INR5 имеет защиту от дребезга, чувствителен к фронту и высокому уровню, т. е. сигнал на входе должен переходить от низкого к высокому уровню и оставаться высоким до тех пор, пока не будет воспринят, а для повторного инициирования входа INR5 необходимо перевести сигнал в низкое состояние и затем опять в высокое.

При прерывании INR1 адрес подпрограммы задаётся внешним устройством, которое управляется сигналом AKINR1 и формирует команды CALL или RST, содержащие заранее заданный произвольный адрес подпрограммы. Диаграмма обработки этого вида прерывания изображена на рисунке 10.

При прерываниях INR2, INR3, INR4, INR5 микропроцессор внутренне формирует фиксированные адреса подпрограмм (рисунок 11, таблица 2).

В первых машинных циклах обработки прерываний INR2, INR3, INR4, INR5 действие сигнала RA игнорируется.

Каждый из 3-х входов прерываний INR2, INR3, INR4 может быть по отдельности замаскирован. Программирование масок осуществляется командой SIM, а установка – сигналом SR.

Для разрешения прерываний INR1, INR2, INR3, INR4 необходимо установить флаг разрешения прерываний, который устанавливается командой EI, а сбрасывается командой DI, любым выполненным прерыванием или сигналом SR.

На прерывание INR5 не действуют ни маски, ни флаг разрешения прерывания. Этот вид прерывания имеет наивысший приоритет, используется в катастрофических ситуациях, таких как отключение питания, столкновения на шинах и т. д. Особенность прерывания INR5 состоит в том, что он, в отличие от иных прерываний, позволяет восстановить состояние флага разрешения прерывания, в котором флаг разрешения прерываний находился до обработки прерывания. Если использовать после обработки прерывания INR5 команду RIM, то в 3-ем разряде аккумулятора будет отражено состояние флага разрешения прерывания, имевшее место до обработки прерывания. Для всех остальных прерываний в этом разряде аккумулятора после выполненного прерывания и последующей команды RIM будет зафиксировано сброшенное состояние флага разрешения прерывания.

На рисунке 12 изображено действие сигнала прерывания после того, как микропроцессор выполнил команду HLT («Останов»).

Входы прерываний внутренне опрашиваются каждый тактовый период. При наличии хотя бы на одном из входов прерываний действующего запроса прерывания микропроцессор формирует ещё два тактовых периода T_{HLT} и затем приступает к формированию машинного цикла M1, сутью которого определяется видом обрабатываемого прерывания.

Вход RQM, так же как и входы прерывания, опрашивается в каждом тактовом периоде T_{HLT} .

На рисунке 13 изображён случай, когда сигналы прерывания и сигнал RQM одновременно активны в одном тактовом полупериоде. В этом случае микропроцессор воспринимает сигналы обоих типов, но первоначально переходит к состоянию T_{RQM} , которое может длиться любое число тактовых периодов, в каждом из которых производится опрос входа и лишь после того, как вход RQM перейдёт к низкому уровню, приступит к формированию машинного цикла M1 обработки прерывания.

Диаграмма машинного цикла M1 (в данном случае обработки одного из прерываний INR2...INR5) демонстрирует, что вход RQM в дальнейшем опрашивается во 2-ом и последующих тактовых периодах, в случае подтверждения запроса прямого доступа к памяти (высокий уровень на выводе AKRQM).

Использование выводов TFD (передача последовательных данных) и RCD (приём последовательных данных)

На рисунке 14 изображена диаграмма выполнения команды RIM (чтение масок прерываний) и команды SIM (установка масок прерываний).

Как следует из диаграммы, вывод RCD внутренне опрашивается микропроцессором в 3-ий тактовый период при выполнении команды RIM. Состояние вывода временно запоминается и в M1 T3 следующей команды заносится в старший разряд аккумулятора, а в 0...6 разряды аккумулятора заносится содержимое масок прерываний, флага разрешения прерываний и входов прерываний INR2, INR3, INR4, опрос которых производится микропроцессором в предыдущем такте (M1 T2 следующей после RIM команды).

Передача последовательных данных на вывод TFD из старшего разряда аккумулятора осуществляется в M1 T2 следующей после SIM команды, при условии, если в 6-ом разряде аккумулятора содержится единица. В то же время осуществляется и программирование масок, если в 3-ем разряде аккумулятора единица.

Включение питания и установка процессора в исходное состояние

Микросхема сконструирована таким образом, что после включения напряжения питания необходимо некоторое время, чтобы она стала работоспособной. Это время определяется частотой сигнала на выводах BQ1, BQ2 и должно составить величину :

не менее чем 3τ , где $\tau=2/f$, f – частота сигнала на выводах BQ1, BQ2 (рисунок 15).

В течение этого времени уровень сигнала на выводе SR должен оставаться низким, что легко может быть достигнуто с помощью простой RC-цепочки, подключённой ко входу SR.

Под воздействием сигнала низкого уровня на входе SR микропроцессор устанавливается в исходное состояние, которое характеризуется следующим состоянием основных внутренних узлов :

- | | | | |
|-------------------------------------|---------------|---|------------|
| * маски прерываний INR2, INR3, INR4 | – установлены | * триггеры машинных циклов M1...M5 | – сброшены |
| * программный счётчик PC | – сброшен | * триггеры входов INR4, INR5 | – сброшены |
| * регистр команд | – сброшен | * внутренне фиксируемые триггеры входов RQM, INR1, RA | – сброшены |
| * флаг разрешения IE | – сброшен | | |
| * триггеры состояний T1...T6 | – сброшены | | |

Вход SR внутренне опрашивается в каждом тактовом периоде и после того, как сигнал на нём достигнет высокого уровня, микропроцессор, сформировав ещё один тактовый период T_{SR} , приступит к формированию M1 T1, т. е. начнёт исполнение программы с нулевого адреса (рисунок 15).

При необходимости возможна установка микропроцессора в исходное состояние в процессе исполнения программы. Это достигается путём подачи на вход SR сигнала низкого уровня в течение времени, величина которого должна быть не менее 3-х тактовых периодов сигнала C (рисунок 15).

Возбуждение внутреннего тактового генератора

Внутренний тактовый генератор может быть запущен подключением к выводам BQ1, BQ2 кварца, RC-цепочки, LC-цепочки, или внешнего генератора. Параметры внешних запускающих элементов должны выбираться с учётом следующего обстоятельства : частота сигнала на выводе C получается путём деления пополам частоты на выводах BQ1, BQ2.

В тех случаях, когда требуется высокая стабильность по частоте, рекомендуется использовать кварц, например :

РК 242МА–14ЕЯ–6000К–Г–В (стабильность в диапазоне температур и напряжений питания не хуже 5×10^{-4} , частота возбуждения 6,0 МГц).

С целью улучшения условий запуска внутреннего генератора при использовании кварца рекомендуется подключать конденсатор ёмкостью 20 пФ между выводом BQ2 и шиной «Общий» (рисунок 16).

При отсутствии жёстких требований к стабильности частоты генерации может использоваться LC-цепочка или RC-цепочка (рисунки 16в,с).

Параметры LC-цепочки определяются из выражения :

$$f = \frac{1}{2\pi \sqrt{L_{EX}(C_{EX} + C_{IN})}}, \text{ где } \begin{cases} C_{EX} - \text{внешняя ёмкость.} \\ C_{IN} - \text{ёмкость между выводами BQ1, BQ2.} \end{cases}$$

Для минимизации нестабильности частоты целесообразно выполнение условия $C_{EX} > 2C_{IN}$

Параметры RC-цепочки могут быть выбраны произвольным образом, исходя из условий обеспечения требуемой частоты генерации, с учётом ограничения : величина резистора RC-цепочки должна быть не менее 5,0 кОм.

Внешний генератор может подключаться непосредственно к выводу BQ1, а вывод BQ2 может при этом не использоваться. Однако, с целью повышения стабильности частоты при её максимально допустимых значениях целесообразно подключать вывод BQ2 к внешнему генератору через инвертирующий элемент (рисунок 16DE).

Система команд

Система команд микропроцессора приведена в таблице 6 и содержит следующие группы команд :

- | | |
|---|---|
| * команды пересылки и загрузки | * команды арифметических операций и команды инкрементирования-декрементирования |
| * команды пересылки в стек и загрузки из стека | * команды логических операций и циклических сдвигов |
| * команды ветвления и команды переходов к подпрограммам | * специальные команды и команды управления |
| * команды возвратов и команды повторного запуска | * дополнительные команды |
| * команды ввода-вывода | |

Формат данных и команд

Память микропроцессора имеет байтовую организацию, поэтому наличие 16-разрядной адресной шины позволяет адресоваться к 65536 байтам памяти.

Формат слова данных :

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Длина команды может быть в 1, 2 или 3 байта. Многобайтовая команда должна храниться в последовательных ячейках памяти. Адрес первого байта должен использоваться как адрес команды.

Формат команды зависит от выполняемой операции.

Адресация

Микросхема использует следующие типы адресации данных, находящихся в памяти или регистрах :

– прямая – 2-ой и 3-ий байты команды содержат адрес памяти, где хранятся данные (2-ой байт – младший, 3-ий байт – старший байт адреса);

– регистровая – командой определяется регистр или пара регистров, в которых размещены данные;

– косвенная регистровая – командой определяется пара регистров, содержащих адрес ячейки памяти, в которой записаны данные (старший байт в 1-ом регистре, младший байт во 2-ом регистре);

– непосредственная – команда содержит данные (однобайтовая или двухбайтовая величина).

Если при выполнении программы (т. е. последовательности команд) не встречаются команды ветвления или прерывания, то выполнение команд происходит путём последовательного увеличения адресов ячеек памяти.

При наличии команд ветвления или прерывания этот порядок нарушается и команда ветвления определяет адрес следующей команды одним из двух способов адресации :

– прямой – команда ветвления содержит адрес следующей команды, которая будет выполняться;

– косвенный регистровый – команда ветвления указывает пару регистров, указывающих адрес следующей команды, которая будет выполняться.

Флаги условий

При выполнении команд в микросхеме используется 7 условных флагов: флаг нуля, флаг знака, флаг чётности, флаг переноса, флаг вспомогательного переноса, флаг переполнения, флаг вспомогательного знака.

Каждый из флагов представлен в микросхеме 1-разрядным регистром. Флаг считается установленным, если в регистре записана единица, и сброшенным, если в регистре записан нуль.

Действие команд на флаги условий происходит следующим образом :

- флаг нуля Z (zero) – если результат команды имеет величину «0», то этот флаг установлен, иначе – сброшен;
- флаг чётности P (parity) – если сумма по модулю два результата выполнения команды равна «0» (т. е. результат является чётным), то флаг установлен, иначе – сброшен;
- флаг переноса C (carry) – если в результате выполнения команды происходит перенос при сложении или заём при вычитании, то флаг считается установленным, иначе – сброшен;
- флаг вспомогательного переноса AC (auxiliary carry) – если команда вызывает перенос из 3-го бита в 4-ый бит результата, то флаг устанавливается, если переноса не было – сброшен;
- флаг знака S (sign) – отражает значение старшего разряда результата; при работе с числами в дополнительном двоичном коде он свидетельствует о знаке результата, если нет переполнения, или численно равен значению старшего разряда результата при наличии переполнения (в этом случае истинный знак результата отражается флагом AS);
- флаг переполнения V – отражает значение логической функции переполнения, являющейся функцией «исключающее ИЛИ» значения переноса в старший разряд АЛУ и значения переноса из старшего разряда АЛУ, и свидетельствует о переполнении при операциях с числами в дополнительном двоичном коде;
- флаг вспомогательного знака AS (auxiliary sign) – отражает значение логической функции "исключающее ИЛ" значения старшего разряда АЛУ и значения логической функции переполнения и свидетельствует об истинном знаке результата при работе с числами в дополнительном двоичном коде. При отсутствии переполнения флаг AS равен флагу S. При наличии переполнения флаг AS принимает значение инверсного флагу S. Для команд INX и DCX поведение флага AS идентично флагу CY, т. е. флаг устанавливается в случае переноса или заёма.

Формат слова состояний флагов :

7	6	5	4	3	2	1	0
S	Z	AS	AC	0	P	V	C

Условные обозначения и сокращения к таблице 6

Байт 2	– второй байт команды	PC	– 16-разрядный регистр программного счётчика
Байт 3	– третий байт команды	PCH	– старший байт регистра программного счётчика
КОП	– код операции	PCL	– младший байт регистра программного счётчика
Данные	– 8-разрядные данные	SP	– 16-разрядный регистр указателя стека
Адрес	– 16-разрядный адрес	SPH	– старший байт регистра указателя стека
Порт	– 8-разрядный адрес устройства ввода/вывода	SPL	– младший байт регистра указателя стека
R, R1, R2	– один из регистров A, B, C, D, E, H, L	HBA	– старший байт адреса
DDD, SSS	– обозначение программно доступных регистров общего назначения :	LBA	– младший байт адреса
DDD	– код регистра приёмника	M _n	– машинный цикл (П = 1...5)
SSS	– код регистра источника	()	– содержание регистра
Коды регистров :		[() ()]	– содержание ячейки памяти
111	– регистр A (аккумулятор)	*	– выходные данные
000	– регистр B	←	– передаётся
001	– регистр C	∧	– логическое И
010	– регистр D	∨	– логическое ИЛИ
011	– регистр E	-∨-	– исключающее ИЛИ
100	– регистр H	+	– сложение
101	– регистр L	–	– вычитание
110	– регистр M (память)	CY, A	– дополнение
		↔	– обменивается
		ЯП	– ячейка памяти
		M	– память
		###	D7 D6 D5 D4 D3 D2 D1 D0
		X	Незаполненная графа

Таблица 6

Условное обозначение команды	Наименование команды	Содержание команды	Код команды ### Вид адресации	Кол-во циклов	Кол-во тактов	Кол-во байтов	Состояние выводов AD0...AD7 по циклам										Изменяемые флаги условий	Примечания
							M1		M2		M3		M4		M5			
							Адрес	Данные	Адрес	Данные	Адрес	Данные	Адрес	Данные	Адрес	Данные		
Команды пересылки и загрузки																		
MOV R1, R2 (R1)←(R2)	Пересылка данных из регистра 2 в регистр 1	Копия содержимого регистра 2 пересылается в регистр 1	01DDSSS регистровая	1	4	1	PC	КОП	X	X	X	X	X	X	X	X	X	
MOV M, R [(H)(L)]←(R)	Пересылка данных из регистра в память	Копия содержимого регистра R пересылается в ЯП с адресом в регистрах H, L	01110SSS косвенная регистровая	2	7	1	PC	КОП	HBA=(H) LBA=(L)	*Из регистра SSS	X	X	X	X	X	X	X	
MOV R, M (R)←[(H)(L)]	Пересылка данных из памяти в регистр	Копия содержимого ЯП с адресом в регистрах H, L загружается в регистр R	01DDD110 косвенная регистровая	2	7	1	PC	КОП	HBA=(H) LBA=(L)	В регистр DDD	X	X	X	X	X	X	X	
MVI R (R)←(байт2)	Пересылка непосредственных данных в регистр	Байт 2 команды загружается в регистр R	00DDD110 непосредственная	2	7	2	PC	КОП	PC+1	В регистр DDD	X	X	X	X	X	X	X	
MVI M [(H)(L)]←(байт2)	Пересылка непосредственных данных в память	Байт 2 команды пересылается в ЯП с адресом в регистрах H, L	00110110 непосредственная	3	10	2	PC	КОП	PC+1	В регистр BX	HBA=(H) LBA=(L)	*Из регистра BX	X	X	X	X	X	BX – временное хранение
LXI B (B)←(байт3) (C)←(байт2)	Загрузка непосредственная пары регистров B, C	Байт 3 команды загружается в регистр B. Байт 2 команды загружается в регистр C.	00000001 непосредственная	3	10	3	PC	КОП	PC+1	В регистр C	PC+2	В регистр B	X	X	X	X	X	X
LXI D (D)←(байт3) (E)←(байт2)	Загрузка непосредственная пары регистров D, E	Байт 3 команды загружается в регистр D. Байт 2 команды загружается в регистр E.	00010001 непосредственная	3	10	3	PC	КОП	PC+1	В регистр E	PC+2	В регистр D	X	X	X	X	X	X
LXI H (H)←(байт3) (L)←(байт2)	Загрузка непосредственная пары регистров H, L	Байт 3 команды загружается в регистр H. Байт 2 команды загружается в регистр L.	00100001 непосредственная	3	10	3	PC	КОП	PC+1	В регистр L	PC+2	В регистр H	X	X	X	X	X	X
LXI SP (SPH)←(байт3) (SPL)←(байт2)	Загрузка непосредственная указателя стека	Байт 3 команды загружается в регистр SPH. Байт 2 команды загружается в регистр SPL.	00110001 непосредственная	3	10	3	PC	КОП	PC+1	Для SPL	PC+2	Для SPH	X	X	X	X	X	X
STAX B [(B)(C)]←(A)	Косвенная запись в память содержимого аккумулятора по регистрам B, C	Копия содержимого аккумулятора пересылается в ЯП с адресом в регистрах B, C.	00000010 косвенная регистровая	2	7	1	PC	КОП	HBA=(B) LBA=(C)	*Из аккумулятора	X	X	X	X	X	X	X	X
LDAX B (A)←[(B)(C)]	Косвенная загрузка аккумулятора по регистрам B, C	Копия содержимого ЯП с адресом в регистрах B, C загружается в аккумулятор	00001010 косвенная регистровая	2	7	1	PC	КОП	HBA=(B) LBA=(C)	В аккумулятор	X	X	X	X	X	X	X	X
LDAX D (A)←[(D)(E)]	Косвенная загрузка аккумулятора по регистрам D, E	Копия содержимого ЯП с адресом в регистрах D, E загружается в аккумулятор	00011010 косвенная регистровая	2	7	1	PC	КОП	HBA=(D) LBA=(E)	В аккумулятор	X	X	X	X	X	X	X	X
STA [(байт3) (байт2)]←(A)	Непосредственная запись в память содержимого аккумулятора	Копия содержимого аккумулятора пересылается в ЯП с адресом, заданным в байтах 2, 3 команды	00110010 прямая	4	13	3	PC	КОП	PC+1	В регистр Z	PC+2	В регистр W	HBA=(W) LBA=(Z)	*Из аккумулятора	X	X	X	X
LDA (A)←[(байт3) (байт2)]	Непосредственная загрузка аккумулятора	Копия содержимого ЯП с адресом, заданным в байтах 2, 3 команды, загружается в аккумулятор	00111010 прямая	4	13	3	PC	КОП	PC+1	В регистр Z	PC+2	В регистр W	HBA=(W) LBA=(Z)	В аккумулятор	X	X	X	X
SHLD [(байт3)(байт2)]←(L) [(байт3)(байт2)+1]←(H)	Непосредственная запись в память содержимого регистров H, L	Копия содержимого регистра L пересылается в ЯП с адресом, заданным в байтах 2, 3 команды. Копия содержимого регистра H пересылается в последующую ЯП.	00100010 прямая	5	16	3	PC	КОП	PC+1	В регистр Z	PC+2	В регистр W	HBA=(W) LBA=(Z)	*Из регистра L	[(W)(Z)+1]	*Из регистра H	X	X
LHLD (L)←[(байт3)(байт2)] (H)←[(байт3)(байт2)+1]	Непосредственная загрузка регистров H, L	Копия содержимого регистра ЯП с адресом, заданным в байтах 2, 3 команды, загружается в регистр L. Копия содержимого последующей ЯП загружается в регистр H.	00101010 прямая	5	16	3	PC	КОП	PC+1	В регистр Z	PC+2	В регистр W	HBA=(W) LBA=(Z)	В регистр L	[(W)(Z)+1]	В регистр H	X	X
XCHG (H)↔(D) (L)↔(E)	Обмен данными между регистрами D, E и H, L	Содержимое регистров H, L меняется с содержимым регистров D, E.	11101011 регистровая	1	4	1	PC	КОП	X	X	X	X	X	X	X	X	X	X
Команды пересылки в стек и загрузки из стека																		
PUSH B [(SP)-1]←(B) [(SP)-2]←(C) (SP)←(SP)-2	Пересылка в стек содержимого регистровой пары B, C	Копия содержимого регистра B пересылается в ЯП с адресом на 1 меньше содержимого регистра SP. Копия содержимого регистра C пересылается в ЯП с адресом на 2 меньше содержимого регистра SP.	11000101 косвенная регистровая	3	12	1	PC	КОП	SP-1	*Из регистра B	SP-2	*Из регистра C	X	X	X	X	X	В M1 указатель стека предварительно уменьшается на 1
PUSH D [(SP)-1]←(D) [(SP)-2]←(E) (SP)←(SP)-2	Пересылка в стек содержимого регистровой пары D, E	Копия содержимого регистра D пересылается в ЯП с адресом на 1 меньше содержимого регистра SP. Копия содержимого регистра E пересылается в ЯП с адресом на 2 меньше содержимого регистра SP.	11010101 косвенная регистровая	3	12	1	PC	КОП	SP-1	*Из регистра D	SP-2	*Из регистра E	X	X	X	X	X	
PUSH H [(SP)-1]←(H) [(SP)-2]←(L) (SP)←(SP)-2	Пересылка в стек содержимого регистровой пары H, L	Копия содержимого регистра H пересылается в ЯП с адресом на 1 меньше содержимого регистра SP. Копия содержимого регистра L пересылается в ЯП с адресом на 2 меньше содержимого регистра SP.	11100101 косвенная регистровая	3	12	1	PC	КОП	SP-1	*Из регистра H	SP-2	*Из регистра L	X	X	X	X	X	
PUSH PSW [(SP)-1]←A [(SP)-2] ₆ ←(CY) [(SP)-2] ₂ ←(P) [(SP)-2] ₄ ←(AC) [(SP)-2] ₆ ←(Z) (SP)←(SP)-2 [(SP)-2] ₁ ←(V) [(SP)-2] ₃ ←(O) [(SP)-2] ₅ ←(AS) [(SP)-2] ₇ ←(S)	Пересылка в стек содержимого аккумулятора и регистра признаков	Копия содержимого регистра A пересылается в ЯП с адресом на 1 меньше содержимого регистра SP. Копия содержимого регистра признаков (слово состояний флагов) пересылается в ЯП с адресом на 2 меньше содержимого регистра SP. Формат слова состояний D7 D6 D5 D4 D3 D2 D1 D0 S Z AS AC O P V CY	11110101 косвенная регистровая	3	12	1	PC	КОП	SP-1	*Из регистра A	SP-2	*Из регистра признаков	X	X	X	X	X	

Условное обозначение команды	Наименование команды	Содержание команды	Код команды ## Вид адресации	Копичество циклов	Копичество тактов	Копичество байтов	Состояние выводов AD0...AD7 по циклам										Изменяемые флаги условий	Примечания		
							M1		M2		M3		M4		M5					
							Адрес	Данные	Адрес	Данные	Адрес	Данные	Адрес	Данные	Адрес	Данные				
POP B (C)←(SP) (B)←[(SP)+1] (SP)←(SP)+2	Загрузка из стека пары регистров B, C	Копия содержимого ЯП с адресом, содержащимся в регистре SP, загружается в регистр C. Копия содержимого ЯП с адресом на 1 больше содержимого регистра SP загружается в регистр B.	11000001 косвенная регистровая	3	10	1	PC	КОП	SP	В регистр C	SP+1	В регистр B	X	X	X	X	X	В M3 происходит последующее увеличение указателя стека на 1		
POP D (E)←(SP) (D)←[(SP)+1] (SP)←(SP)+2	Загрузка из стека пары регистров D, E	Копия содержимого ЯП с адресом, содержащимся в регистре SP, загружается в регистр E. Копия содержимого ЯП с адресом на 1 больше содержимого регистра SP загружается в регистр D.	11010001 косвенная регистровая	3	10	1	PC	КОП	SP	В регистр E	SP+1	В регистр D	X	X	X	X	X			
POP H (L)←(SP) (H)←[(SP)+1] (SP)←[(SP)+2]	Загрузка из стека пары регистров H, L	Копия содержимого ЯП с адресом, содержащимся в регистре SP, загружается в регистр L. Копия содержимого ЯП с адресом на 1 больше содержимого регистра SP загружается в регистр H.	11100001 косвенная регистровая	3	10	1	PC	КОП	SP	В регистр L	SP+1	В регистр H	X	X	X	X	X			
POP PSW (CY)←(SP) ₀ (V)←(SP) ₁ (P)←(SP) ₂ (O)←(SP) ₃ (AC)←(SP) ₄ (AS)←(SP) ₅ (Z)←(SP) ₆ S←(SP) ₇ (A)←[(SP)+1] (SP)←[(SP)+2]	Загрузка из стека аккумулятора и регистра признаков	Копия содержимого ЯП с адресом, содержащимся в регистре SP, загружается в регистр признаков. Копия содержимого ЯП с адресом на 1 больше содержимого регистра SP загружается в регистр A.	11110001 косвенная регистровая	3	10	1	PC	КОП	SP	В регистр признаков	SP+1	В регистр A	X	X	X	X	X	Z, S, P, CY, AC, AS, V		
XTHL (L)←(SP) (H)←[(SP)+1]	Обмен между регистром верхнего уровня стека и регистрами H, L	Содержимое регистра L обменивается с содержимым ЯП, адрес которой определён содержимым регистра SP. Содержимое регистра H обменивается с содержимым ЯП, адрес которой на 1 больше содержимого регистра SP.	11100011 регистровая	5	16	1	PC	КОП	SP	В регистр L	SP+1	В регистр H	SP+1	*Из регистра H	SP	*Из регистра L	X			
SPHL (SP)←(H)(L)	Пересылка содержимого регистров H, L в указатель стека	Копия содержимого регистров H, L пересылается в регистр SP	11111011 регистровая	1	6	1	PC	КОП												
Команды ветвления																				
(PC)←(байт3)(байт2)	Переход :		непосредственная																	
JMP	Безусловный	Управление передаётся команде, адрес которой задаётся байтом 2 и байтом 3 команды (адрес перехода)	11000011	3	10	3	PC	КОП	PC+1	В регистр Z (адрес перехода)	SP+2	В регистр W (адрес перехода)	X	X	X	X	X	X	Если переход, то адресом следующей команды является адрес перехода, записанный в течение M2, M3 в регистры W,Z. Если перехода нет, то цикл M3 отсутствует, при этом после выполнения цикла M2 программный счётчик увеличивается на 2.	
JC	Если: перенос (CY=1)	Если проверяемое условие выполнено, то управление передаётся команде, адрес которой задаётся байтом 2 и байтом 3 команды (адрес перехода).	11011010	2/3	7/10															
JNC	не перенос (CY=0)	Если проверяемое условие не выполнено, то управление передаётся команде, адрес которой задаётся байтом 2 и байтом 3 команды (адрес перехода).	11010010																	
JZ	нуль (Z=1)	Если проверяемое условие выполнено, то управление передаётся команде, адрес которой задаётся байтом 2 и байтом 3 команды (адрес перехода).	11001010																	
JNZ	не нуль (Z=0)	Если проверяемое условие не выполнено, то управление передаётся команде, адрес которой задаётся байтом 2 и байтом 3 команды (адрес перехода).	11000010																	
JP	плюс (S=0)	Если проверяемое условие не выполнено, то выполняется следующая команда.	11110010																	
JM	минус (S=1)	Если проверяемое условие не выполнено, то выполняется следующая команда.	11111010																	
JPE	чётно (P=1)	Если проверяемое условие не выполнено, то выполняется следующая команда.	11101010																	
JPO	нечётно (P=0)	Если проверяемое условие не выполнено, то выполняется следующая команда.	11100010																	
PCHL (PCH)←(H) (PCL)←(L)	Пересылка содержимого регистров H, L в счётчик команд	Копия содержимого регистра H пересылается в старшие 8 разрядов регистра PC. Копия содержимого регистра L пересылается в младшие 8 разрядов регистра PC.	11101001 регистровая	1	6	1	PC	КОП	X	X	X	X	X	X	X	X	X	X		
Команды переходов к подпрограммам																				
[(SP)-1]←(PCH) [(SP)-2]←(PCL) (SP)←(SP)-2 (PC)←(байт3)(байт2)	Переход к подпрограмме :		непосредственная и косвенная регистровая																	
CALL	Безусловный	8 старших разрядов адреса следующей команды (адрес возврата) пересылаются в ЯП, адрес которой на 1 меньше содержимого регистра SP.	11001101	5	18	3	PC	КОП	PC+1	В регистр Z (адрес перехода)	PC+2	В регистр W (адрес перехода)	SP-1	*Адрес возврата (PCH)	SP-2	*Адрес возврата (PCL)	X			
CC	Если: перенос (CY=1)	Если проверяемое условие выполнено, то 8 старших разрядов адреса следующей команды (адрес возврата) пересылаются в ЯП, адрес которой на 1 меньше содержимого регистра SP.	11011100																	
CNC	не перенос (CY=0)	Если проверяемое условие не выполнено, то 8 старших разрядов адреса следующей команды (адрес возврата) пересылаются в ЯП, адрес которой на 1 меньше содержимого регистра SP.	11010100																	
CZ	нуль (Z=1)	Если проверяемое условие выполнено, то 8 младших разрядов адреса следующей команды (адрес возврата) пересылаются в ЯП, адрес которой на 2 меньше содержимого регистра SP.	11001100																	
CNZ	не нуль (Z=0)	Если проверяемое условие не выполнено, то 8 младших разрядов адреса следующей команды (адрес возврата) пересылаются в ЯП, адрес которой на 2 меньше содержимого регистра SP.	11000100																	
CP	плюс (S=0)	Управление передаётся команде, адрес которой задаётся байтом 2 и байтом 3 команды (адрес перехода).	11110100																	
CM	минус (S=1)	Управление передаётся команде, адрес которой задаётся байтом 2 и байтом 3 команды (адрес перехода).	11111100																	
CPE	чётно (P=1)	Управление передаётся команде, адрес которой задаётся байтом 2 и байтом 3 команды (адрес перехода).	11101100																	
CPO	нечётно (P=0)	Управление передаётся команде, адрес которой задаётся байтом 2 и байтом 3 команды (адрес перехода).	11100100																	
Команды возвратов																				
(PCL)←[(SP)] (PCH)←[(SP)+1] (SP)←(SP)+2	Возврат:		косвенная регистровая																	
RET	Безусловный	Копия содержимого ЯП, адрес которой задан содержимым регистра SP, загружается в младшие 8 разрядов PC. Копия содержимого ЯП, адрес которой на 1 больше содержи-	11001001	3	10	1	PC	КОП	SP	Для PCL (адреса возврата)	SP+1	Для PCH (адреса возврата)	X	X	X	X	X	X	Если возврат, то адресом следующей команды является адрес возврата, записанный в	

