

Тема 15

Команды вызова подпрограмм

Теория

Безусловный вызов подпрограмм

Осуществляется с помощью команды CALL A16. Команды вызова подпрограммам показаны в таблице 1.

В результате выполнения команды CALL A16 осуществляются следующие действия:

- 1) при считывании трёх байт самой команды формируется адрес следующей команды – это **адрес возврата** в основную программу из подпрограммы;
- 2) адрес возврата записывается в стековую память – он будет считан оттуда при выполнении команды безусловного или условного возврата из подпрограммы (тема 16);
- 3) в программный счетчик будет записан код **адреса начала подпрограммы** – число A16, предусмотренное в команде.

Примечание: МЦ1, МЦ4 и МЦ5 команды CALL A16 аналогичны МЦ1, МЦ2 и МЦ3 команды PUSH RP (тема 10).

Пример 1

Команда CALL 0900H, записанная в программной памяти с адреса 0830H. До выполнения команды в указателе стека (SP)=0BВ0H – это вершина стека в учебном стенде УМПК-80.

Команда 3-байтная (п.4.1 в таблице 1), записана в программной памяти следующим образом:

Адрес	Данные	Характер данных
0830	CD	КОП
0831	00	МБ адреса начала подпрограммы
0832	09	СБ адреса начала подпрограммы

Команды вызова подпрограмм

	Мнемокод	Операция	байт	МЦ	МТ	Действие
4.1	CALL A16	(PCH)→[(SP)-1], (PCL)→[(SP)-2], A16→PC	3	5	17	Вызов подпрограммы с начальным адресом A16
4.2	CNZ A16	Если Z=0 , то (PCH)→[(SP)-1], (PCL)→[(SP)-2], A16→PC; если Z=1 , то выполняется следующая команда	3	5/3	17/11	Вызов подпрограммы с начальным адресом A16 при наличии в аккумуляторе <i>ненулевого</i> результата
	CZ A16	Если Z=1 , то (PCH)→[(SP)-1], (PCL)→[(SP)-2], A16→PC; если Z=0 , то выполн. следующая команда	3	5/3	17/11	Вызов подпрограммы с начальным адресом A16 при наличии в аккумуляторе <i>нулевого</i> результата
	CNC A16	Если C=0 , то (PCH)→[(SP)-1], (PCL)→[(SP)-2], A16→PC; если C=1 , то выполняется следующая команда	3	5/3	17/11	Вызов подпрограммы с начальным адресом A16 при отсутствии переноса из старшего разряда аккумулятора или получении отрицательной разности
	CC A16	Если C=1 , то (PCH)→[(SP)-1], (PCL)→[(SP)-2], A16→PC; если C=0 , то выполняется следующая команда	3	5/3	17/11	Вызов подпрограммы с начальным адресом A16 при наличии переноса из старшего разряда аккумулятора или получении положительной разности
	CPO A16	Если P=0 , то (PCH)→[(SP)-1], (PCL)→[(SP)-2], A16→PC; если P=1 , то выполняется следующая команда	3	5/3	17/11	Вызов подпрограммы с начальным адресом A16 при наличии в аккумуляторе числа с нечетным количеством единиц
	CPE A16	Если P=1 , то (PCH)→[(SP)-1], (PCL)→[(SP)-2], A16→PC; если P=0 , то выполняется следующая команда	3	5/3	17/11	Вызов подпрограммы с начальным адресом A16 при наличии в аккумуляторе числа с четным количеством единиц
	CP A16	Если S=0 , то (PCH)→[(SP)-1], (PCL)→[(SP)-2], A16→PC; если S=1 , то выполняется следующая команда	3	5/3	17/11	Вызов подпрограммы с начальным адресом A16 при наличии в аккумуляторе числа с нулевым старшим разрядом
	CM A16	Если S=1 , то (PCH)→[(SP)-1], (PCL)→[(SP)-2], A16→PC; если S=0 , то выполняется следующая команда	3	5/3	17/11	Вызов подпрограммы с начальным адресом A16 при наличии в аккумуляторе числа с единичным старшим разрядом

В результате выполнения команды адрес возврата 0833H будет сохранен в стековой памяти побайтно:

- ✓ СБ – число 08H - будет записан в ячейку с адресом 0BAFH;
- ✓ МБ – число 33H - будет записан в ячейку с адресом 0BAEH.

В последнем МТ выполнения команды в программный счетчик будет загружен адрес перехода – число 0833H.

На рисунке 1 показана временная диаграмма, а в таблице 2 – выполнение команды.

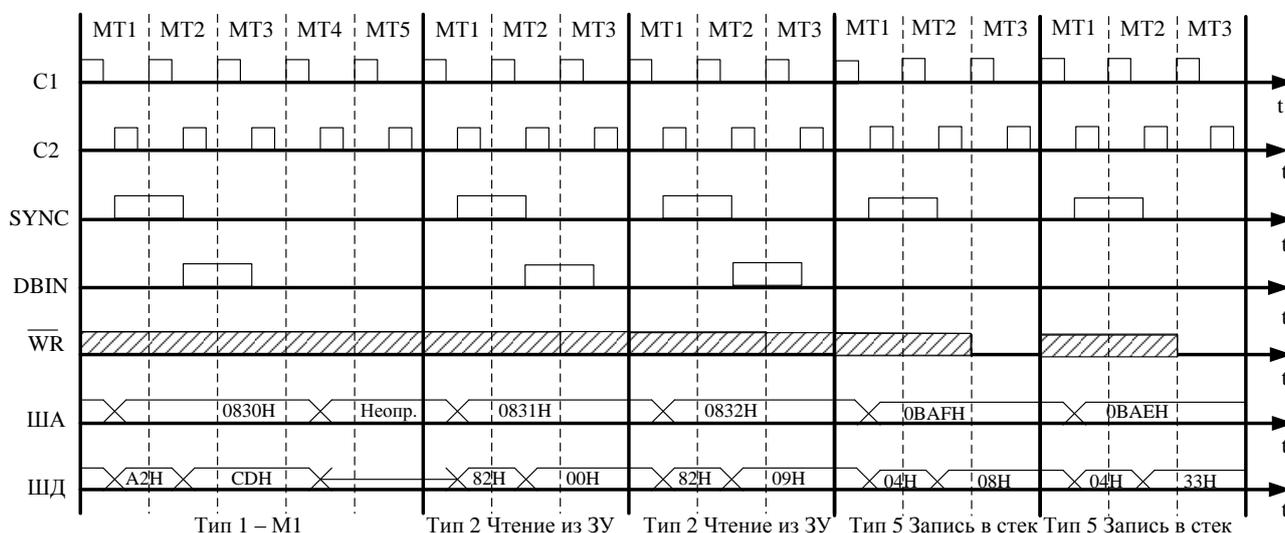


Рис. 1. Временная диаграмма выполнения команды **CALL 0900H**, записанной в программной памяти с адреса 0830H

Таблица 2

Выполнение команды **CALL 0900H**, записанной в программной памяти с адреса 0830H

МЦ	МТ	Действие	Пояснение
1	1	(PC) → PA → ША; 0830H → ША	МП выставляет адрес на ША из программного счетчика
	2	(PC) + 1 → PC 0831H → PC	Содержимое программного счетчика увеличивается на 1 – это подготовка для считывания из программной памяти второго байта команды
	2,3	КОП → РК; CDH → РК	Считывание кода операции команды в регистр команд

	4	Дешифрация команды	<i>Передача команды из регистра в дешифратор команд и запуск соответствующей логической схемы</i>
	5	$(SP) - 1 \rightarrow SP; 0BAFH \rightarrow SP$	<i>Подготовка первого адреса стека для записи</i>
2	1	$(PC) \rightarrow PA \rightarrow ША$ $0831H \rightarrow ША$	<i>МП выставляет адрес на ША из программного счетчика</i>
	2	$(PC) + 1 \rightarrow PC$ $0832H \rightarrow PC$	<i>Содержимое программного счетчика увеличивается на 1 – это подготовка для считывания из программной памяти третьего байта команды</i>
	2,3	МБ адреса начала подпрограммы \rightarrow в Z; $00H \rightarrow Z$	<i>Считывание второго байта команды в программно-недоступный регистр Z</i>
3	1	$(PC) \rightarrow PA \rightarrow ША$ $0832H \rightarrow ША$	<i>МП выставляет адрес на ША из программного счетчика</i>
	2	$(PC) + 1 \rightarrow PC$ $0833H \rightarrow PC$	<i>Содержимое программного счетчика увеличивается на 1 – это подготовка для считывания из программной памяти первого байта следующей команды программы</i>
	2,3	СБ адреса начала подпрограммы \rightarrow в W; $09H \rightarrow W$	<i>Считывание второго байта команды в программно-недоступный регистр W</i>
4	1	$(SP) \rightarrow PA \rightarrow ША$ $0BAFH \rightarrow ША$	<i>МП выставляет адрес на ША из указателя стека</i>
	2	$(SP) - 1 \rightarrow SP; 0BAEH \rightarrow SP$	<i>Подготовка в указателе стека адреса для второй записи, подготовка данных для записи в стек</i>
	3	$(PCH) \rightarrow$ стек; $08H \rightarrow [0BAFH]$	<i>Копирование содержимого старшего регистра пары в стек</i>
5	1	$(SP) \rightarrow PA \rightarrow ША$ $0BAEH \rightarrow ША$	<i>МП выставляет адрес на ША из указателя стека</i>
	2	<i>Подготовка данных для записи в стек</i>	
	3	$(PCL) \rightarrow$ стек; $08H \rightarrow [0BAEH]$ $(WZ) \rightarrow PC; 0900H \rightarrow PC$	<i>Копирование содержимого младшего регистра пары в стек. Загрузка адреса начала подпрограммы в программный счетчик</i>

Вызов подпрограммы по условию

Формат этих команд такой же, как у команды CALL A16, различается выполнение (таблица 1). Оно может осуществляться в двух вариантах:

1) если условие, предусмотренное в команде, **выполняется**, то временная диаграмма команды аналогична выполнению команды CALL A16 – выполнение происходит за 5 машинных циклов. Результатом выполнения команды является переход на подпрограмму по адресу A16 и сохранение в стековой памяти адреса возврата в основную программу.

2) Если указанное в команде условие *не выполняется*, то работы со стеком не происходит (нет МЦ4 и МЦ5), а выполнение команды сводится к увеличению программного счетчика в течение трех машинных циклов, когда считывается сама команда – все три байта. В данном случае перехода на подпрограмму нет, и основная программа выполняется дальше.

В качестве условий (как и в командах условного перехода) выступают значения признаков результатов операций:

- ✓ ***NZ (no Zero – нет нуля) – признак нуля $Z=0$*** , т.е. результат в аккумуляторе отличен от числа 00H;
- ✓ ***Z (Zero – есть нуль) – признак нуля $Z=1$*** , т.е. результат в аккумуляторе равен 00H;
- ✓ ***NC (no Carry – нет переноса) – признак переноса $C=0$*** , т.е. есть переноса из старшего разряда аккумулятора нет или получено положительное число при вычитании;
- ✓ ***C (no Carry – есть перенос) – признак переноса $C=1$*** , т.е. есть перенос из старшего разряда аккумулятора или получено отрицательное число при вычитании;
- ✓ ***PO (parity – равенство, odd number – нечетное число)*** – признак четности $P=0$; в аккумуляторе число, содержащее нечетное количество единиц;
- ✓ ***PE (parity – равенство, even number – четное число)*** – признак четности $P=1$; в аккумуляторе число, содержащее четное количество единиц;
- ✓ ***P (Positive – положительное число) – признак знака $S=0$*** . Результат в аккумуляторе имеет в старшем разряде 0 – число считается положительным;
- ✓ ***M (Minus – минус) – признак знака $S=1$*** . Результат в аккумуляторе имеет в старшем разряде 1 – число считается отрицательным.

Пример 2

Команда CZ 095FH, записанная в программной памяти с адреса 0870H. Перед выполнением команды признак $Z = 0$.

Формат данной команды

Байт	Данные		Адрес программной памяти
	функционал	16-ричный код	
1	Код операции	CC	0870
2	МБ адреса перехода на подпрограмму	5F	0871
3	СБ адреса перехода на подпрограмму	09	0872

Поскольку указанное в программе условие ($Z=1$) не выполняется, команда будет выполняться за 3 машинных цикла и перехода на подпрограмму не произойдет. Временная диаграмма представлена на рисунке 2, пояснения – в таблице 3.

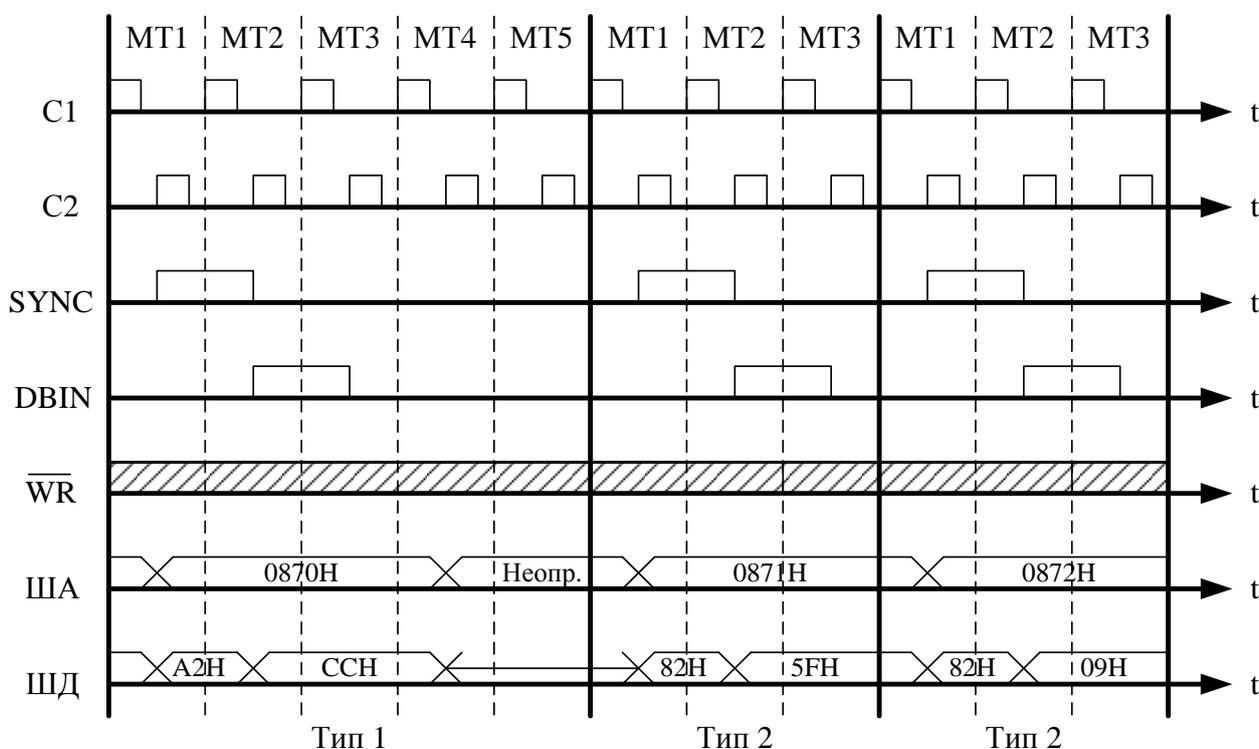


Рис. 2. Временная диаграмма команды CZ 095F при $Z=0$

За счет того, что принудительной загрузки программного счетчика не происходит, после команды CZ 095FH будет выполняться следующая команда программы, записанная в программной памяти, начиная с адреса 0873H.

Команда выполняется за 3 МЦ, 10 МТ.

Таблица 2

Выполнение команды CZ 095FH, записанной в программной памяти с адреса 0870H. Перед выполнением команды признак Z = 0.

МЦ	МТ	Действие	Пояснение
1	1	(PC) → PA → ША; 0870H → ША	МП выставляет адрес на ША из программного счетчика
	2	(PC) + 1 → PC 0871H → PC	Содержимое программного счетчика увеличивается на 1 – это подготовка для считывания из программной памяти второго байта команды
	2,3	КОП → РК; ССН → РК	Считывание кода операции команды в регистр команд
	4	Дешифрация команды	Передача команды из регистра в дешифратор команд и запуск соответствующей логической схемы
	5	(SP) – 1 → SP; 0BAFH → SP	Подготовка первого адреса стека для записи
2	1	(PC) → PA → ША 0871H → ША	МП выставляет адрес на ША из программного счетчика
	2	(PC) + 1 → PC 0872H → PC	Содержимое программного счетчика увеличивается на 1 – это подготовка для считывания из программной памяти третьего байта команды
	2,3	МБ адреса начала подпрограммы → в Z; 5FH → Z	Считывание второго байта команды в программно-недоступный регистр Z
3	1	(PC) → PA → ША 0872H → ША	МП выставляет адрес на ША из программного счетчика
	2	(PC) + 1 → PC 0873H → PC	Содержимое программного счетчика увеличивается на 1 – это подготовка для считывания из программной памяти первого байта следующей команды программы
	2,3	СБ адреса начала подпрограммы → в W; 09H → W	Считывание второго байта команды в программно-недоступный регистр W

Пример 3

Команда CM 0958H, записанная в программной памяти с адреса 0894H. До выполнения команды в указателе стека (SP)=0BB0H, (F)=82H.

Команда 3-байтная (п.4.2 в таблице 1), записана в программной памяти следующим образом:

Адрес	Данные	Характер данных
0894	FC	КОП
0895	58	МБ адреса начала подпрограммы
0896	09	СБ адреса начала подпрограммы

Поскольку (F)=82H=10000010₂, признак знака S=1, число считается отрицательным, следовательно, условие, поставленное в команде, выполняется. Поэтому команда будет выполняться за 5 МЦ, 17 МТ, аналогично команде CALL A16 (см. выше пример 1).

В результате выполнения команды адрес возврата 0897H будет сохранен в стековой памяти побайтно:

- ✓ СБ – число 08H - будет записан в ячейку с адресом 0BAFH;
- ✓ МБ – число 97H - будет записан в ячейку с адресом 0BAEH.

В последнем МТ выполнения команды в программный счетчик будет загружен адрес начала подпрограммы – число 0958H.

На рисунке 3 показана временная диаграмма, а в таблице 4 – выполнение команды.

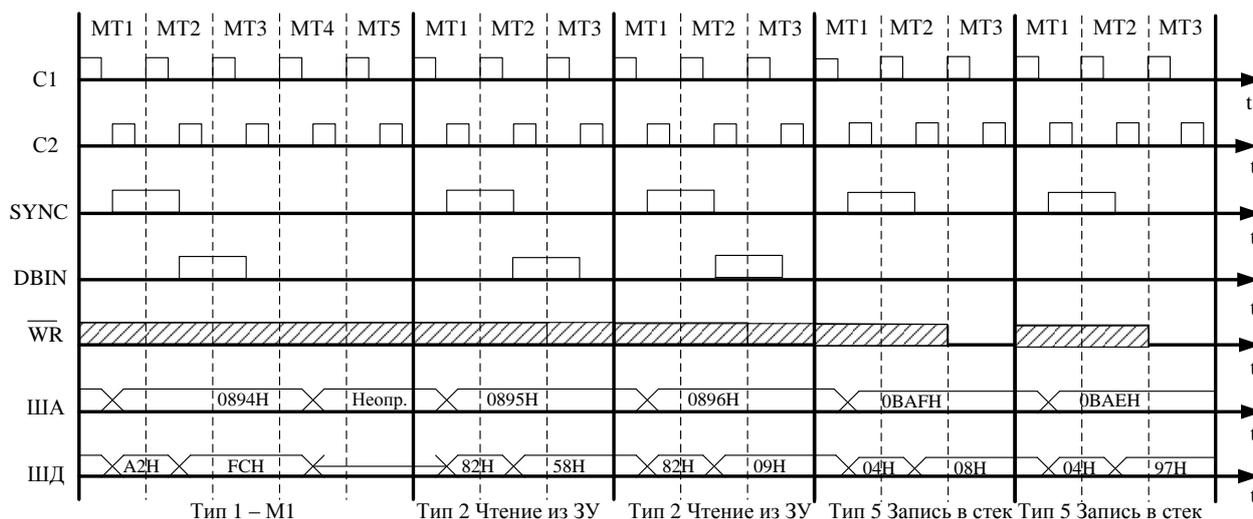


Рис. 3. Временная диаграмма выполнения команды CM 0958H, записанной в программной памяти с адреса 0894H. До выполнения команды в указателе стека (SP)=0BB0H, (F)=82H

Таблица 4

Выполнение команды **CM 0958H**,
записанной в программной памяти с адреса 0894H (S=1)

МЦ	МТ	Действие	Пояснение
1	1	(PC) → PA → ША; 0894H → ША	МП выставляет адрес на ША из программного счетчика
	2	(PC) + 1 → PC 0895H → PC	Содержимое программного счетчика увеличивается на 1 – это подготовка для считывания из программной памяти второго байта команды
	2,3	КОП → РК; FCH → РК	Считывание кода операции команды в регистр команд
	4	Дешифрация команды	Передача команды из регистра в дешифратор команд и запуск соответствующей логической схемы
	5	(SP) – 1 → SP; 0BAFH → SP	Подготовка первого адреса стека для записи
2	1	(PC) → PA → ША 0895H → ША	МП выставляет адрес на ША из программного счетчика
	2	(PC) + 1 → PC 0896H → PC	Содержимое программного счетчика увеличивается на 1 – это подготовка для считывания из программной памяти третьего байта команды
	2,3	МБ адреса начала подпрограммы → В Z; 58H → Z	Считывание второго байта команды в программно-недоступный регистр Z
3	1	(PC) → PA → ША 0896H → ША	МП выставляет адрес на ША из программного счетчика
	2	(PC) + 1 → PC 0897H → PC	Содержимое программного счетчика увеличивается на 1 – это подготовка для

			<i>считывания из программной памяти первого байта следующей команды программы</i>
	2,3	СБ адреса начала подпрограммы → в W; 09H → W	<i>Считывание второго байта команды в программно-недоступный регистр W</i>
4	1	(SP) → PA → ША 0BAFH → ША	<i>МП выставляет адрес на ША из указателя стека</i>
	2	(SP) – 1 → SP; 0BAEH → SP	<i>Подготовка в указателе стека адреса для второй записи, подготовка данных для записи в стек</i>
	3	(PCH) → стек; 08H → [0BAFH]	<i>Копирование содержимого старшего регистра пары в стек</i>
5	1	(SP) → PA → ША 0BAEH → ША	<i>МП выставляет адрес на ША из указателя стека</i>
	2	<i>Подготовка данных для записи в стек</i>	
	3	(PCL) → стек; 97H → [0BAEH] (WZ) → PC; 0900H → PC	<i>Копирование содержимого младшего регистра пары в стек. Загрузка адреса начала подпрограммы в программный счетчик</i>

Вызов подпрограмм с фиксированным адресом

Осуществляется командой RST N, где N=0, 1, ..., 7. Данные команды приведены в таблице 5.

Таблица 5

Команды безусловного вызова подпрограммы с фиксированным адресом

	Мнемокод	Операция	Кол-во байт	Кол-во МЦ	Кол-во МТ	Изменяемые регистры	Изменяемые признаки	Содержание операции
4.3	RST N	(PCH) → [(SP)-1], (PCL) → [(SP)-2], 0000 0000 00NN N000 ₂ → PC	1	3	11	-	-	Переход на подпрограмму обслуживания прерывания

После выполнения команды RST N происходит вызов подпрограммы с начальным адресом 0000 0000 00NN N000₂, где NNN – двоичный код, соответствующий номеру N.

Пример 4

Команда RST 1, записанная в программной памяти по адресу 0842H. До выполнения команды в указателе стека (SP) = 0B0H. Временная диаграмма команды такая же, как у команды PUSH RP (тема 10), с тем лишь отличием, что

в стеке сохраняется не регистровая пара, а программный счетчик. Временная диаграмма представлена на рис. 4, выполнение – в таблице 6.

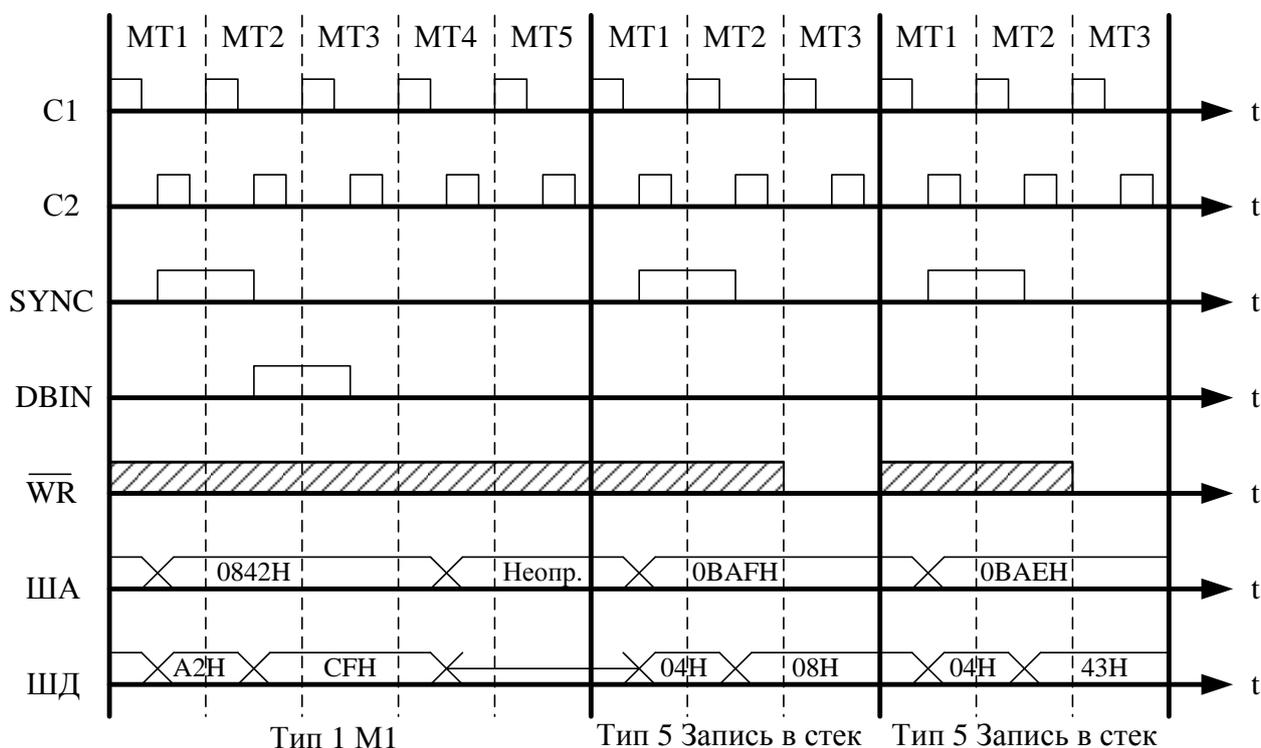


Рис. 4. Временная диаграмма команды RST1

Таблица 6

Выполнение команды CZ 095FH, записанной в программной памяти с адреса 0870H. Перед выполнением команды признак Z = 0.

МЦ	МТ	Действие	Пояснение
1	1	(PC) → PA → ША; 0842H → ША	МП выставляет адрес на ША из программного счетчика
	2	(PC) + 1 → PC 0843H → PC	Содержимое программного счетчика увеличивается на 1 – это формирование адреса возврата из подпрограммы в основную программу
	2,3	КОП → РК; CFH → РК	Считывание кода операции команды в регистр команд
	4	Дешифрация команды	Передача команды из регистра в дешифратор команд и запуск соответствующей логической схемы
	5	(SP) – 1 → SP; 0BAFH → SP	Подготовка первого адреса стека для записи
2	1	(SP) → PA → ША 0BAFH → ША	МП выставляет адрес на ША из указателя стека
	2	(SP) – 1 → SP; 0BAEH → SP	Подготовка в указателе стека адреса для

			<i>второй записи, подготовка данных для записи в стек</i>
	3	(PCH)→стек; 08H→[0BAFH]	<i>Копирование содержимого старшего регистра пары в стек</i>
3	1	(SP) → PA → ША 0BAEH →ША	<i>МП выставляет адрес на ША из указателя стека</i>
	2	<i>Подготовка данных для записи в стек</i>	
	3	(PCL)→стек; 43H→[0BAEH] 0000 0000 00NN N000 ₂ →PC; 0000 0000 0000 1000 ₂ →PC	<i>Копирование содержимого младшего регистра пары в стек. Загрузка адреса начала подпрограммы в программный счетчик</i>

После выполнения этой команды начинается выполнение подпрограммы с начальным адресом 0008H.