

Пример построения системы прерывания

1 Исходные данные.

1.1 Требования к системе прерывания.

1.2 Структура ЭВМ с аппаратурой прерывания.

1.3 Структура аппаратуры прерывания.

2 Разработка системы прерывания.

2.1 Определение архитектуры и программирование.

2.2 Кодирование программы и распределение памяти программ и микропрограмм.

2.3 Разработка алгоритма работы и микропрограммная реализация ЭВМ.

2.4 Ввод и отладка микропрограммы командного цикла и программ решения задач.

Знать: понятие «прерывание программ», основные характеристики СП, основные фазы процесса прерывания программ.

Уметь: построить схему аппаратуру приоритетных прерываний с маскированием запросов.

Помнить: о взаимодействии аппаратных и программных средств в процессе прерывания программ

Литература:

- Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем. Учебник для вузов. – СПб.: Питер, 2004. – 668 с.

1 Исходные данные

1.1 Требования к системе прерывания

Пример разработки ЭВМ с системой прерывания программ включает в себя решение следующих задач.

1. Определение архитектуры ЭВМ.
2. Разработку микропрограммы командного цикла ЭВМ на базе структуры микропрограммируемой ЭВМ, используемой в лабораторном практикуме.
3. Отладку микропрограммы командного цикла ЭВМ с помощью программы "Имитатор микропрограммируемой микроЭВМ".
4. Разработку и отладку программ для исходных данных, приведенных ниже.

Основная программа и запросы на прерывание

- Основная программа производит вычисление суммы S ($S < 32768$) элементов $T_i \geq 0$ массива размерности N :

$$S = \sum_{i=1}^N T_i.$$

- В ЭВМ одновременно может поступать до восьми запросов на прерывание программ: $Z7, Z6, \dots, Z0$. При этом запрос с меньшим номером имеет более высокий приоритет.

Запросы на прерывание могут быть замаскированы с помощью 8-разрядной маски M , записываемой в регистр маски RM .

Запрос маскируется единичным значением одноименного разряда маски.

Виды и маски программ

- Каждая программа имеет маску, с помощью которой она может быть защищена от прерываний по тем или иным запросам.

Основная программа (P) имеет маску, разрешающую все прерывания.

Служебная программа (PS), выполняющая необходимые начальные установки, характеризуется маской запрещающей все прерывания.

- Программа P0 обработки прерывания нулевого уровня, вызываемая по запросу Z0, вычисляет частное от деления X на Y нацело ($X, Y > 0$, целые числа).

Остальные программы обработки прерываний (P1-P2) являются "пустыми" и после установки маски возвращают управление.

Порядок выполнения программ

- При запуске ЭВМ в начале выполняется служебная программа, которая затем передает управление основной программе.

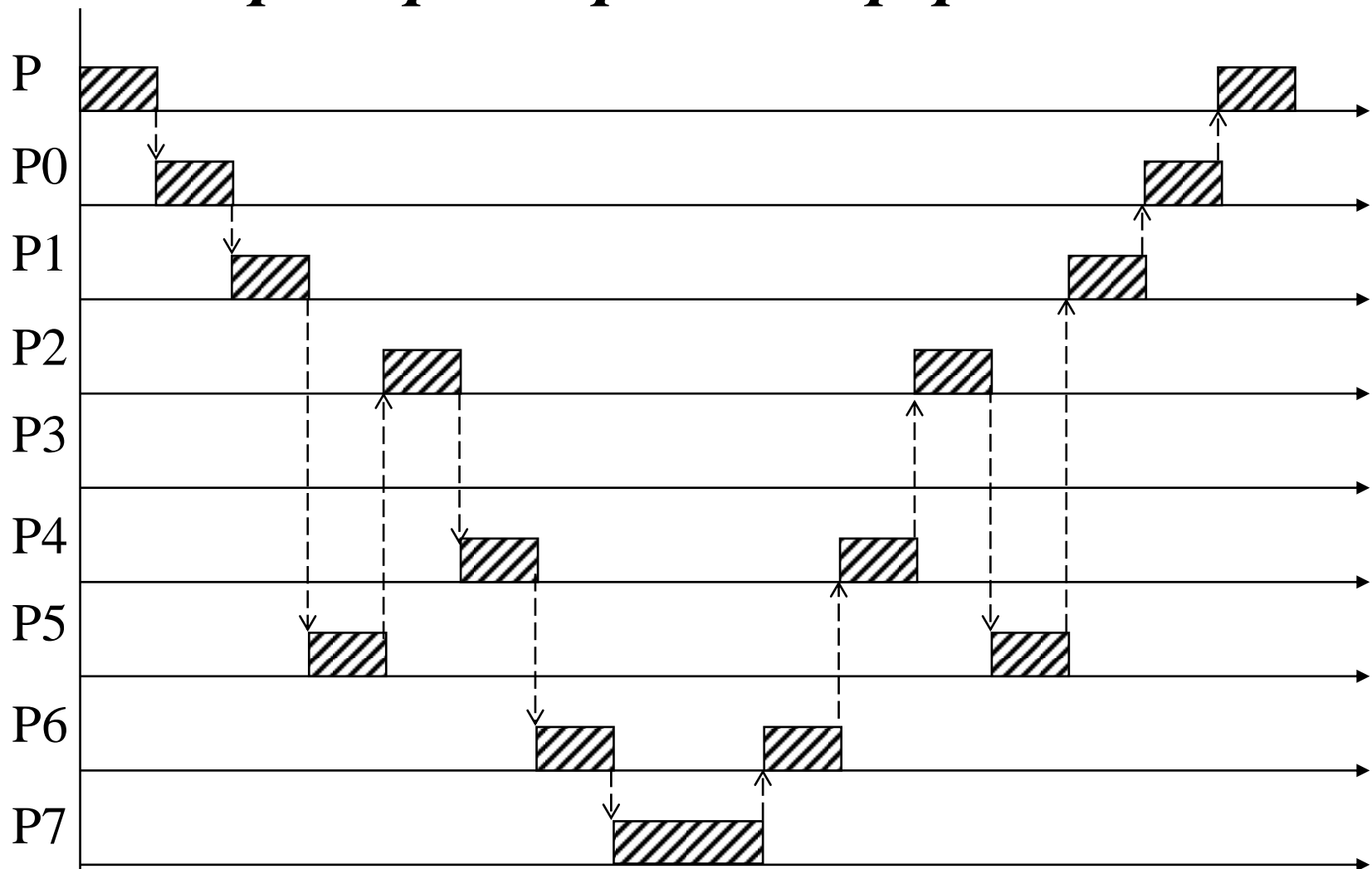
В процессе экспериментальных исследований разработанной ЭВМ необходимо, используя моделирующую программу "Имитатор микропрограммируемой микроЭВМ", построить диаграмму прерываний, исходя из анализа работы ЭВМ для заданного начального состояния системы прерывания.

Начальное состояние системы прерывания характеризуется набором масок программ и множеством запросов, поступивших в систему (таблица приведена далее).

Начальное состояние системы прерывания

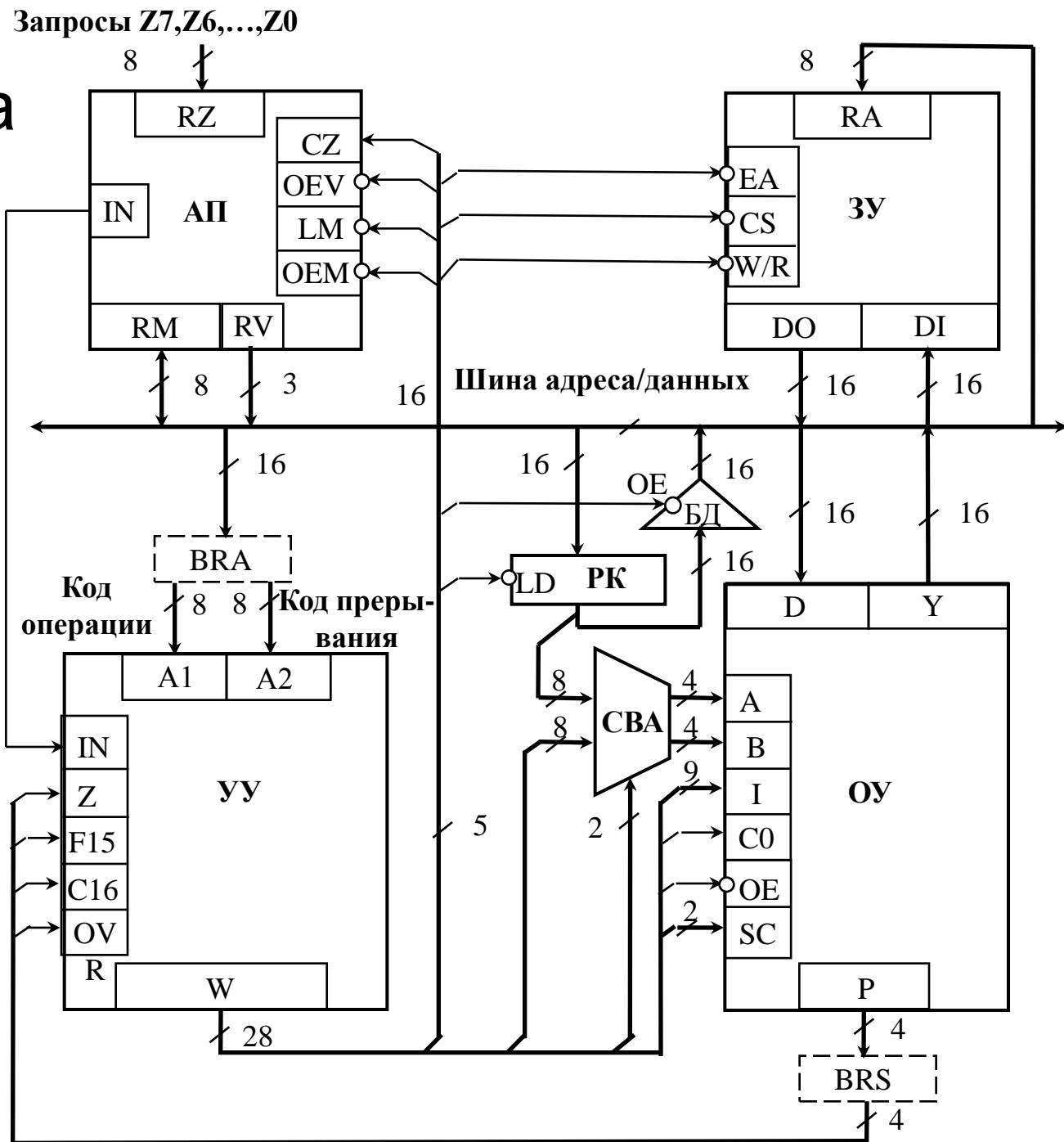
Объект	Условное обозначение	Значение
Маска программы P	M	00000000
Маска программы P0	M0	00000001
Маска программы P1	M1	01010111
Маска программы P2	M2	01001101
Маска программы P3	M3	10111100
Маска программы P4	M4	00110110
Маска программы P5	M5	01110011
Маска программы P6	M6	01110010
Маска программы P7	M7	11100110
Маска программы PS	MS	11111111
Множество запросов	Z	11110111

Пример диаграммы прерываний

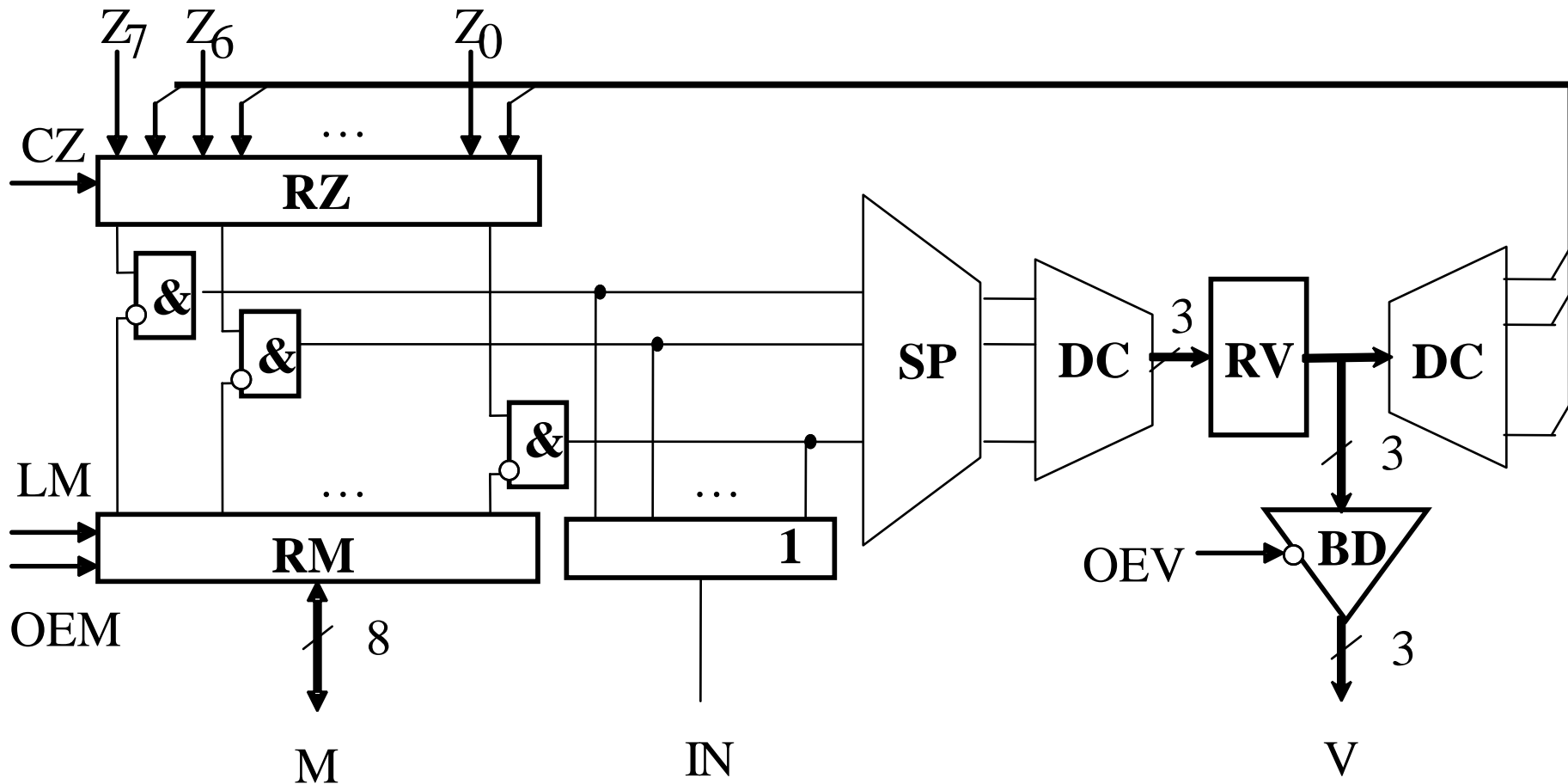


P – основная программа, P0-P7 – прерывающие программы

1.2 Структура ЭВМ с аппаратурой прерывания



1.3 Структура аппаратуры прерывания



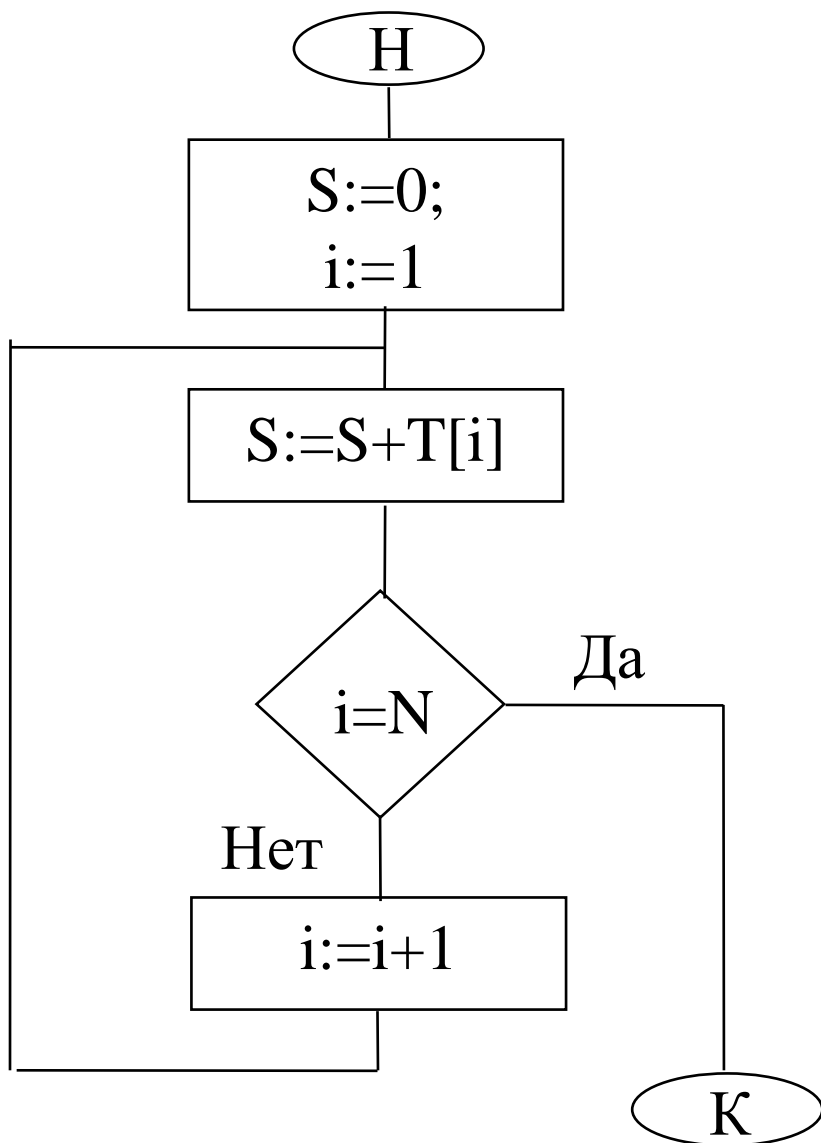
CZ – очистка в регистре запросов (RZ) разряда, соответствующего принятому к обработке запросу на прерывание; LM – загрузка маски с шины данных в регистр маски (RM); OEM – выдача маски из регистра маски на шину данных; OEV – выдача адреса (вектора) прерываний из регистра прерываний на шину адреса.

2 Разработка системы прерывания

2.1 Определение архитектуры и программирование

- Архитектура ЭВМ может быть получена путем развития архитектуры, разработанной при выполнении предыдущей лабораторной работы. Разработка архитектуры включает определение форматов данных, выбор состава программно-доступных регистров и определение системы команд.
- Дополнительные команды определяются в процессе разработки алгоритмов основной и прерывающих программ и анализа функций системы прерывания ЭВМ с учетом требований, приведенных в задании.
- Граф-схемы алгоритмов основной программы и программы обработки прерывания P0 приведены далее.

Граф-схема алгоритма основной программы



Основная программа
производит вычисление
суммы S элементов $T[i]$
массива размерности N :

$$S = \sum_{i=1}^N T_i .$$

Граф-схема алгоритма прерывающей программы P0

Программа P0 обработки прерывания нулевого уровня вычисляет частное Z от деления X на Y нацело ($X, Y > 0$, целые числа).



SAVE – подпрограмма сохранения,
RESTORE – подпрограмма восстановления содержимого регистров ЭВМ.

Определение форматов данных и программно-доступных регистров

- *Форматы данных.* Данные в примере являются целыми числами, изменяющимися в пределах от 0 до 32767. В этом случае любое число можно представить 16-разрядным двоичным кодом, старший разряд которого определяет знак числа.
- *Программно-доступные регистры.* ЭВМ имеет 10 16-разрядных программно-доступных регистров: шесть регистров общего назначения (r0-r5), программный счетчик — PC (r6), регистр признаков — RP (r7), содержащий разряды двух признаков: нуля (PZ) и знака (PS), регистр указателя стека — rSP (r8), регистр адреса таблицы прерываний — rATI, а также 8-разрядный регистр маски RM.

Программно-доступные регистры

Регистры ЭВМ

r0:	X			Делимое X
r1:	Y (T)			Делитель Y (суммируемые числа T)
r2:	Z			Частное Z
r3:	S			Сумма S
r4:	N			Число повторений цикла N
r5:	AM			Адрес массива AM
r6:	PC			Программный счетчик
r7:	PS	RP	PZ	Регистр признаков
r8:	rSP			Регистр указателя стека
r9:	rATI			Регистр адреса таблицы прерываний
RM:	M			Регистр маски программы

Разработка системы команд

Определяется набор операций, способы адресаций, модификации и форматы команд. В примере в систему команд ЭВМ, дополнительно включаются следующие две команды.

- *LM A* – *загрузка маски*. Данная команда обеспечивает загрузку в регистр маски младшего байта ячейки памяти, адрес A которой указан в команде.
- *IRET* – *возврат из прерывающей программы*. При выполнении команды из стека восстанавливаются значения регистра маски и программного счетчика, соответствующие последней прерванной программе, что обеспечивает возврат на команду, перед которой произошло прерывание. Предполагается, что сохранение содержимого программного счетчика и регистра маски в стеке осуществляется с помощью специальной подмикропрограммы командного цикла ЭВМ при переключении процессора на прерывающую программу.

Система команд

Наименование	Мнемоника	Описание	Признаки	
			PZ	PS
СУММИРОВАНИЕ	ADD r r*	$r := r + r^*, PC := PC + 1$	+	+
ВЫЧИТАНИЕ	SUB r r*	$r := r - r^*, PC := PC + 1$	+	+
ДОБАВЛЕНИЕ C	AD r C	$r := r + C, PC := PC + 1$	+	+
ВЫЧИТАНИЕ C	SB r C	$r := r - C, PC := PC + 1$	+	+
ЧТЕНИЕ В РЕГИСТР	LD r A	$r := M[A], PC := PC + 1$	—	—
ЗАПИСЬ РЕГИСТРА	MV r A	$M[A] := r, PC := PC + 1$	—	—
ЧТЕНИЕ В РЕГИСТР с индексацией	LDI r (r*)+	$r := M[r^*], r^* := r^* + 1;$ $PC := PC + 1$	+	+
ЗАПИСЬ В СТЕК	PUSH r (rSP)-	$M[rSP] := r;$ $rSP := rSP - 1, PC := PC + 1$	—	—
ЧТЕНИЕ ИЗ СТЕКА	POP r (rSP)+	$rSP := rSP + 1;$ $r := M[rSP], PC := PC + 1$	—	—

Система команд (продолжение)

Наименование	Мнемоника	Описание	Признаки	
			PZ	PS
ПЕРЕХОД	BR A	PC:=A	—	—
ПЕРЕХОД, ЕСЛИ НУЛЬ	BEQ A	Если PZ=1, то PC:=A, иначе PC:=PC+1	—	—
ПЕРЕХОД, ЕСЛИ МИНУС	BMI A	Если PS=1, то PC:=A, иначе PC:=PC+1	—	—
ОБРАЩЕНИЕ К ПОДПРОГРАММЕ	CALL (rSP)- A	M[rSP]:=PC; rSP:=rSP-1, PC:=A	—	—
ОСТАНОВ	HLT A	PC:=A, останов	—	—
ЗАГРУЗКА МАСКИ	LM A	RM:= M[A], PC:=PC+1	—	—
ВОЗВРАТ ИЗ ПРЕРЫВАЮЩЕЙ ПРОГРАММЫ	IRET	SP:=SP+1, RM:=M[rSP]; SP:=SP+1, PC:=M[rSP]	—	—

Обозначения в таблице системы команд

В таблице использованы следующие обозначения:

$r, r^* \{r_0, r_1, \dots, r_8\}$ – программно-доступные регистры:
регистр r^* является источником данных, а регистр r – приемником результата, но может также служить источником второго операнда;

RM – регистр маски, $M[A]$ – ячейка памяти с адресом A ;

знак "+" в описании признаков означает, что устанавливается новое значение признака по результату выполнения команды, а знак "-" свидетельствует о сохранении старого значения признака.

Программирование: служебная программа

- Программирование включает разработку основной и прерывающих программ и распределение программ и данных с учетом выбранного способа переключения ЭВМ на прерывающие программы.
- Для выполнения начальных установок вводится специальная служебная программа (PS), которая загружает маску, запрещающую все прерывания, инициализирует регистр указателя стека и регистр адреса таблицы прерываний, а затем передает управление основной программе.

LM AMS

Загрузка маски служебной программы

LD rSP ASP

Загрузка указателя стека SP

LD rATI ATI

Загрузка адреса таблицы прерываний ATI

LD PC AP

Загрузка начального адреса программы P

Программирование: основная программа

LM AMP	Загрузка маски программы P
LD r5 AAM	Загрузка адреса массива AM в регистр r5
LD r4 AN	Загрузка числа повторений цикла N в регистр r4
SUB r3 r3	Очистка регистра r3 для суммы S
m2 LDI r1 (r5)+	Чтение числа T[i] в регистр r1
ADD r3 r1	Суммирование
SB r4 "1"	Вычитание единицы из числа повторений цикла
BEQ m1	Если PZ=1 (r4=0), то переход на метку m1
BR m2	Переход на метку m2
m1 MV r3 AS	Запись в память суммы S по адресу AS
HLT SA	Загрузка PC и останов

Программирование: прерывающая программ P0

LM AM0	Загрузка маски программы P0
PUSH RP	Сохранение в стеке содержимого регистра RP
PUSH r0	Сохранение в стеке содержимого регистра r0
PUSH r1	Сохранение в стеке содержимого регистра r1
PUSH r2	Сохранение в стеке содержимого регистра r2
SUB r2 r2	Очистка регистра для частного Z
LD r0 AX	Загрузка X в регистр r0
LD r1 AY	Загрузка Y в регистр r1
m2 SUB r0 r1	Вычитание из делимого X делителя Y
BMI m1	Если PS=1, то переход на метку m1
AD r2 "1"	Увеличение на единицу частного Z
BR m2	Переход на метку m2
m1 MV r2 AZ	Запись частного Z
POP r2	Чтение из стека содержимого регистра r2
POP r1	Чтение из стека содержимого регистра r1
POP r0	Чтение из стека содержимого регистра r0
POP RP	Чтение из стека содержимого регистра RP
IRET	Возврат из программы P0

Программирование: прерывающие программы P_1, P_2, \dots, P_7

Программы обработки прерываний P_1, P_2, \dots, P_7 являются по заданию "пустыми" и состоят из двух команд: загрузки маски и возврата из прерывания. Пример такой программы приведен ниже.

Прерывающая программа $P[K]$ ($K=1,2,3,\dots,7$)

LM AM[K]	Загрузка маски программы $P[K]$
IRET	Возврат из программы $P[K]$

2.2 Кодирование программы и распределение памяти программ и данных

- Наименования, мнемонические обозначения и коды операций приведены ниже в таблице.

Наименование	Мнемоника	Код операции
СУММИРОВАНИЕ	ADD	01
ВЫЧИТАНИЕ	SUB	02
ДОБАВЛЕНИЕ КОНСТАНТЫ	AD	9
ВЫЧИТАНИЕ КОНСТАНТЫ	SB	A
ЧТЕНИЕ В РЕГИСТР	LD	B
ЗАПИСЬ РЕГИСТРА	MV	C
ЧТЕНИЕ В РЕГИСТР с индексацией	LDI	10

Коды операций (продолжение)

Наименование	Мнемоника	Код операции
ЗАПИСЬ В СТЕК	PUSH	03
ЧТЕНИЕ ИЗ СТЕКА	POP	04
ПЕРЕХОД	BR	05
ПЕРЕХОД, ЕСЛИ НУЛЬ	BEQ	06
ПЕРЕХОД, ЕСЛИ МИНУС	BMI	07
ОБРАЩЕНИЕ К ПОДПРОГРАММЕ	CALL	8
ОСТАНОВ	HLT	00
ЗАГРУЗКА МАСКИ	LM	11
ВОЗВРАТ ИЗ ПРЕРЫВАЮЩЕЙ ПРОГРАММЫ	IRET	12

Форматы команд

Φ					
15	14...12	11...8	7...4	3...0	
0	K1		r	r*	ADD, SUB, LDI, PUSH, POP
0	K2		A		BR,BEQ,BMI, IRET, LM A, HLT
1	K3	r	C		AD, SB
1	K4	r	A		LD, MV, CALL

Распределение памяти

Адрес	Назначение
00-08	Служебная программа
0A-11	Таблица прерываний
12-1A	Маски программ
20-2D	Основная программа
40-54	Прерывающая программа P0
60-6D	Прерывающие программы P1-P7
...-FF	Область памяти стека

Распределение памяти: служебная программа

Адрес	Код	Мнемоника	Комментарий
Служебная программа			
00	0005	SA	Начальный адрес служебной программы
01	00FF	MS	Маска служебной программы
02	FFFF	ASP	Начальный адрес области памяти стека
03	000A	ATI	Начальный адрес таблицы прерываний
04	0013	AP	Начальный адрес основной программы P
05	1101	LM AMS	Загрузка маски служебной программы
06	B802	LD rSP ASP	Загрузка указателя стека SP
07	B903	LD rATI ATI	Загрузка адреса таблицы прерываний
08	B604	LD PC AP	Загрузка начального адреса программы P
09			Свободная ячейка памяти

Распределение памяти: таблица прерываний

Адрес	Код	Мнемоника	Комментарий
0A	0043	AP0	Адрес прерывающей программы P0
0B	0060	AP1	Адрес прерывающей программы P1
0C	0062	AP2	Адрес прерывающей программы P2
0D	0064	AP3	Адрес прерывающей программы P3
0E	0066	AP4	Адрес прерывающей программы P4
0F	0068	AP5	Адрес прерывающей программы P5
10	006A	AP6	Адрес прерывающей программы P6
11	006C	AP7	Адрес прерывающей программы P7

Распределение памяти: маски программ

Адрес	Код	Мнемоника	Комментарий
12	0000	M	Маска основной программы P
13	00FF	M0	Маска прерывающей программы P0
14	0000	M1	Маска прерывающей программы P1
15	0000	M2	Маска прерывающей программы P2
16	0000	M3	Маска прерывающей программы P3
17	0000	M4	Маска прерывающей программы P4
18	0000	M5	Маска прерывающей программы P5
19	0000	M6	Маска прерывающей программы P6
1A	0000	M7	Маска прерывающей программы P7

Распределение памяти: основная программа

Адрес	Код	Мнемоника	Комментарий
20	002E	AM	Начальный адрес массива
21	0003	N	Количество чисел в массиве
22		S	Результат – сумма чисел
23	1112	LM AMP	Загрузка маски программы P
24	B520	LD r5 AAM	Загрузка адреса массива AM в регистр r5
25	B421	LD r4 AN	Загрузка числа повторений цикла N
26	0233	SUB r3 r3	Очистка регистра r3 для суммы S
27	1015	LDI r1 (r5)+	Чтение числа T[i] в регистр r1
28	0131	ADD r3 r1	Суммирование
29	A401	SB r4 "1"	Вычитание единицы из числа N
2A	062C	BEQ m1	Если PZ=1 (N=0), то переход на метку m1
2B	0527	BR m2	Переход на метку m2
2C	C322	MV r3 AS	Запись суммы S адресу AS
2D	0005	HLT SA	Загрузка PC и останов

***Распределение памяти:
прерывающая программа P0***

Адрес	Код	Мнемоника	Комментарий
40		X	Делимое X
41		Y	Делитель Y
42		Z	Частное Z
43	1113	LM AM0	Загрузка маски программы P0
44	0370	PUSH RP	Сохранение содержимого регистра RP
45	0300	PUSH r0	Сохранение содержимого регистра r0
46	0310	PUSH r1	Сохранение содержимого регистра r1
47	0320	PUSH r2	Сохранение содержимого регистра r2
48	0222	SUB r2 r2	Очистка регистра для частного Z
49	B040	LD r0 AX	Загрузка X в регистр r0
4A	B141	LD r1 AY	Загрузка Y в регистр r1
4B	0201	SUB r0 r1	Вычитание из делимого X делителя Y

***Распределение памяти:
прерывающая программа P0 (продолжение)***

Адрес	Код	Мнемоника	Комментарий
4C	074F	BMI m1	Если PS=1, то переход на метку m1
4E	054B	BR m2	Переход на метку m2
4F	C242	MV r2 AZ	Запись частного Z
50	0420	POP r2	Чтение из стека содержимого регистра r2
51	0410	POP r1	Чтение из стека содержимого регистра r1
52	0400	POP r0	Чтение из стека содержимого регистра r0
53	0470	POP RP	Чтение из стека содержимого регистра RP
54	1200	IRET	Возврат из программы P0

***Распределение памяти:
прерывающие программы P1-P5***

Адрес	Код	Мнемоника	Комментарий
60	1114	LM AM1	Загрузка маски программы P1
61	1200	IERT	Возврат из программы P1
62	1115	LM AM2	Загрузка маски программы P2
63	1200	IERT	Возврат из программы P2
64	1116	LM AM3	Загрузка маски программы P3
65	1200	IERT	Возврат из программы P3
66	1117	LM AM4	Загрузка маски программы P4
67	1200	IERT	Возврат из программы P4
68	1118	LM AM5	Загрузка маски программы P5
69	1200	IERT	Возврат из программы P5

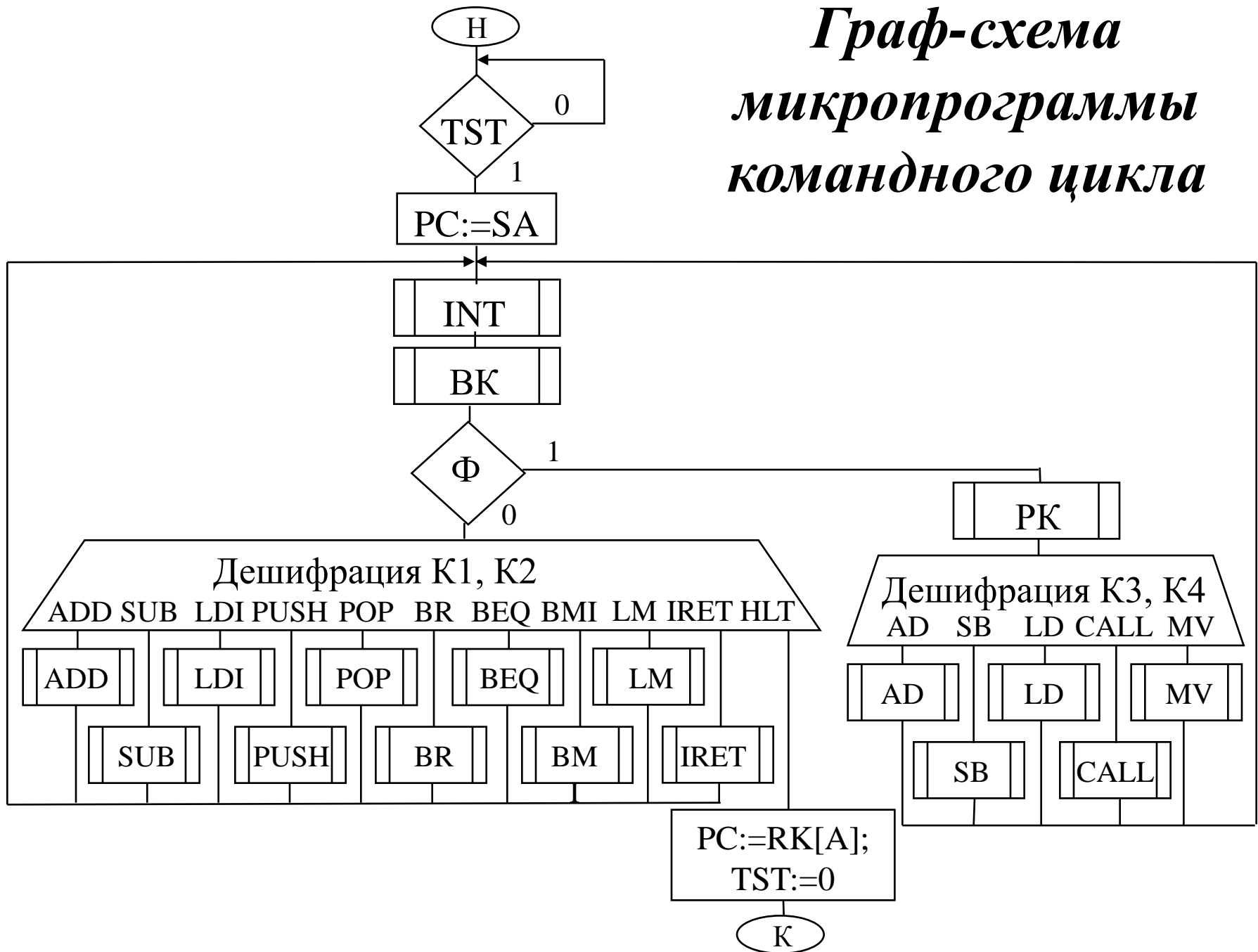
***Распределение памяти:
прерывающие программы P6-P7, стек***

Адрес	Код	Мнемоника	Комментарий
6A	1119	LM AM6	Загрузка маски программы P6
6B	1200	IERT	Возврат из программы P6
6C	111A	LM AM7	Загрузка маски программы P7
6D	1200	IERT	Возврат из программы P7
Область памяти стека			
...			
FC			
FD			
FE			
FF			

2.3 Разработка алгоритма работы и микропрограммная реализация ЭВМ

- Предполагается, что прерывания в ЭВМ допускаются перед считыванием очередной команды.
- Граф-схема микропрограммы командного цикла приведена на рисунке. Подмикропрограмма прерывания INT опрашивает выход IN аппаратуры прерывания и при наличии незамаскированных запросов на прерывания осуществляет переключение ЭВМ на программу обработки прерывания, номер которого сформирован аппаратурой прерывания.
- Микропрограммная реализация ЭВМ включает: распределение внутренних регистров микропроцессора, разработку и кодирование микропрограммы командного цикла, а также распределение памяти микропрограмм.

Граф-схема микропрограммы командного цикла

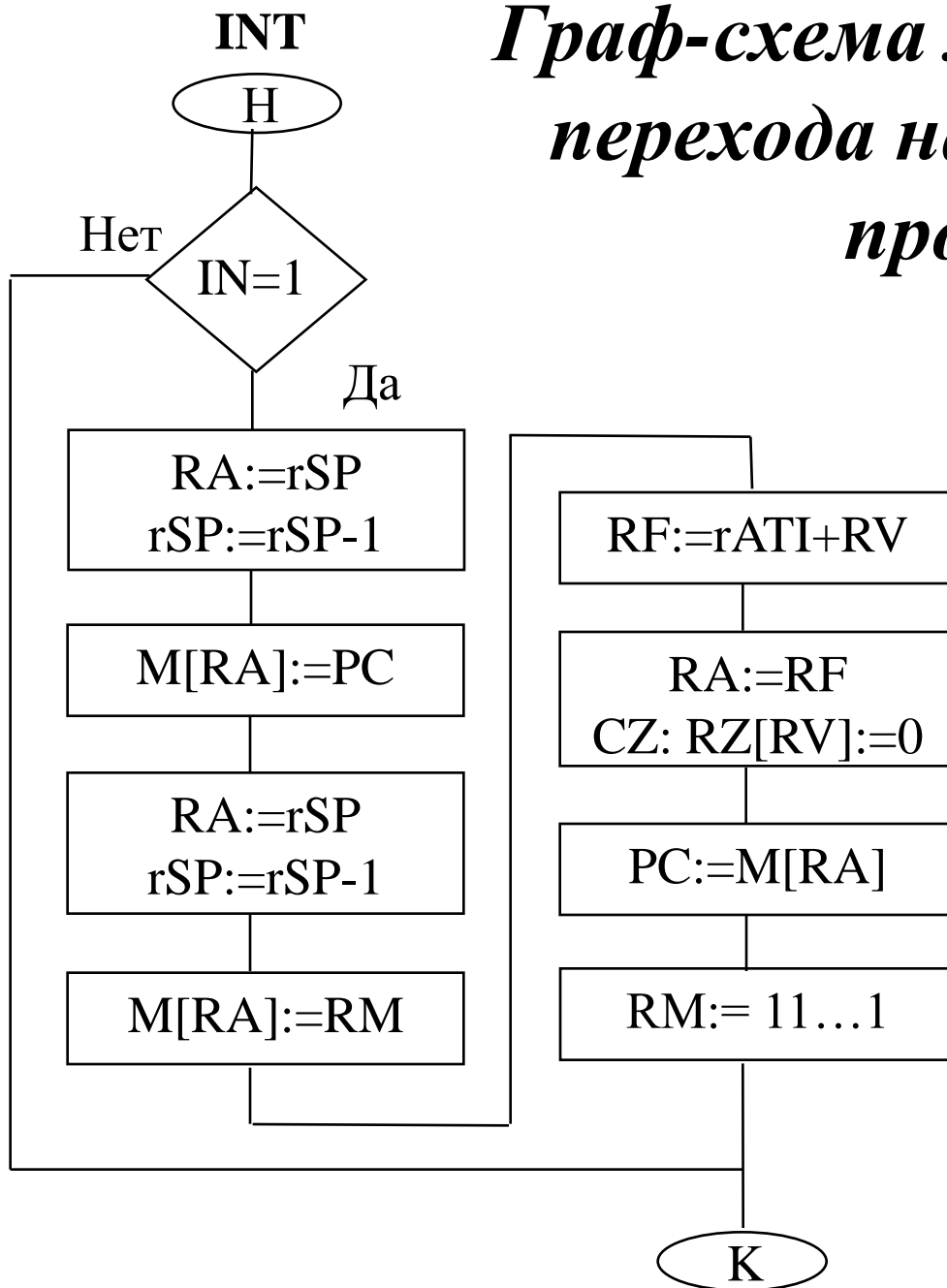


Функции микропрограммы перехода на прерывающую программу

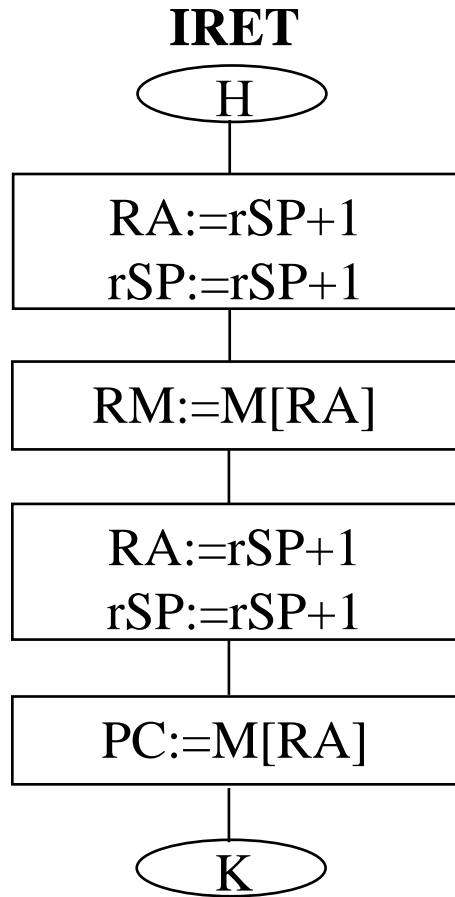
Для переключения ЭВМ на прерывающую программу микропрограмма INT выполняет следующие действия:

- записывает в стек адрес следующей команды прерываемой программы (из РС) и содержимое регистра маски RM (считывая его из АП);
- используя адрес таблицы прерываний и номер запроса на прерывание, загружает в программный счетчик адрес первой команды прерывающей программы;
- выдает сигнал CZ, управляющий сбросом соответствующего обрабатываемому прерыванию разряда регистра запросов RZ;
- заносит в регистр маски код 11111111, что означает запрещение прерываний.

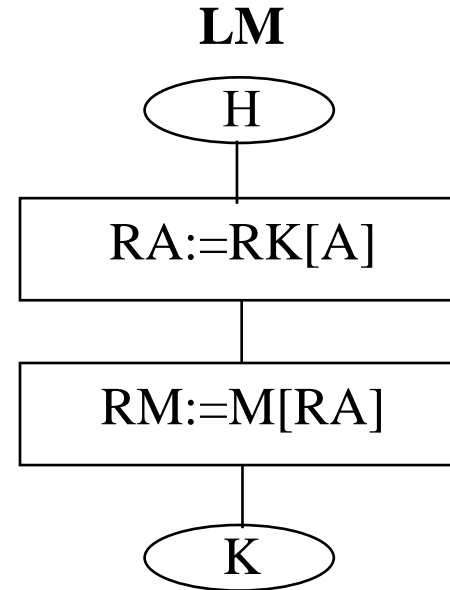
Граф-схема микропрограммы перехода на прерывающую программу



Граф-схемы микропрограмм операций



Возврат из
прерывающей
программы



Загрузка маски
программы

Регистры ЭВМ (программно доступные)

- ЭВМ имеет 10 16-разрядных программно-доступных регистров:
 - шесть регистров общего назначения (r0-r5),
 - программный счетчик – PC (r6),
 - регистр признаков – RP (r7), содержащий разряды двух признаков: нуля (PZ) и знака (PS),
 - регистр указателя стека – rSP (r8),
 - регистр адреса таблицы прерываний – rATI,
 - 8-разрядный регистр маски RM.
- Все 16-разрядные программно-доступные регистры отображаются на регистры R0-R9 операционного устройства.
- Регистр маски RM входит в состав аппаратуры прерывания.

Регистры ЭВМ (программно недоступные)

- Кроме одиннадцати программно-доступных регистров в состав ЭВМ входят восемь программно-недоступных регистров. Четыре таких регистра (два для команды, и по одному для константы, счетчика адреса ЗУ, операнда Y) отображаются на регистры R13-R15 и RQ операционного устройства.
- Кроме того, программно-недоступными регистрами являются:
 - регистр адреса ЗУ (RA) и регистр команд (RK),
 - регистр запросов на прерывание (RZ) и регистр номера (вектора) запроса на прерывание (RV).

Регистры ЭВМ

РЗУ (R0-R7)

0:	r0			X
1:	r1			Y (T)
2:	r2			Z
3:	r3			S
4:	r4			N
5:	r5			AM
6:	r6 (PC)			
7:	PS	r7 (RP)	PZ	
RM:	Маска			
RA:	Адрес ЗУ			
RZ:	Запросы			
RV:	Вектор			

РЗУ (R8-R15)

8:	r8 (rSP)		
9:	r9 (rATI)		
10:			
11:			
12:			
13:	Буферный РК		
14:	Регистр константы		
15:	Сч. адреса ЗУ (RK[A])		
RQ:	Регистр операнда		
RK:	Регистр команды		

Дешифрация кода операции

Код операции преобразуется в начальный адрес подмикропрограммы (адрес первой МК) соответствующей операции с помощью преобразователя начального адреса (ПНА), входящего в состав устройства управления ЭВМ.

Мнемо-ника	Код операции	Адрес первой МК
ADD	01	1F
SUB	02	1E
AD	9	20
SB	A	21
LD	B	1A
MV	C	1C
LDI	10	22
PUSH	03	27

Мнемо-ника	Код операции	Адрес первой МК
POP	04	29
BR	05	2C
BEQ	06	2B
BMI	07	2D
CALL	8	30
<i>LM</i>	<i>11</i>	<i>37</i>
<i>IRET</i>	<i>12</i>	<i>33</i>
HLT	00	35

Кодирование микропрограммы (начало)

№	МИ	РЗУ		Упр. АЛУ			Упр. ОЗУ			Шина	МИ	Упр. усл.			Упр. УУ			Упр. ПК		
МК	I8-0	A	B	C0	OE	SC	CS	W	EA	D11-0	I3-0	A	U	CCE	C0	R	OE	M	L	OE
1A	133	0	F	0	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1	1
RA:=RF[A] (LD r A)																				
1B	337	0	0	0	1	00	0	1	1	00A	3	000	1	1	1	1	0	01	1	1
r:=M[A]																				
1C	133	0	F	0	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1	1
RA:=RF[A] (MV r A)																				
1D	133	0	0	0	0	00	0	0	1	00A	3	000	0	1	1	1	0	01	1	1
M[A]:=r																				
1E	311	0	0	1	0	00	1	1	1	00A	3	000	0	1	1	1	0	11	1	1
r:=r-r* (SUB r r*)																				
1F	301	0	0	0	0	00	1	1	1	00A	3	000	0	1	1	1	0	11	1	1
r:=r-r* (ADD r r*)																				
20	301	F	0	0	0	00	1	1	1	00A	3	000	0	1	1	1	0	01	1	1
r:=r+C (AD r C)																				
21	311	F	0	1	0	00	1	1	1	00A	3	000	0	1	1	1	0	01	1	1
r:=r-C (SB r C)																				
22	104	0	0	0	0	00	1	1	0	000	E	000	0	0	1	1	0	11	1	1
RA:=r*; (LDI r (r*)+)																				
23	337	0	0	0	1	00	0	1	1	004	C	000	0	1	1	1	0	11	1	1
r:=M[r*] PA/ЦИ:=4																				
24	733	C	C	0	0	00	1	1	1	024	9	000	0	0	1	1	0	00	0	1
RC - сдвиг влево PA/ЦИ:=PA/ЦИ-1																				
25	303	0	0	1	1	00	1	1	1	000	E	000	0	0	1	1	0	11	1	1
r*=r*+1																				
26	133	0	F	0	0	00	1	1	1	00A	3	000	0	1	1	1	0	00	0	1
RK:=RF (для формирования признаков)																				

Кодирование микропрограммы (окончание)

№	МИ	РЗУ		Упр. АЛУ			Упр. ОЗУ			Шина	МИ	Упр. усл.			Упр. УУ			Упр. РК		
МК	I8-0	A	B	C0	\overline{OE}	SC	\overline{CS}	\overline{W}	\overline{EA}	D11-0	I3-0	A	U	\overline{CCE}	C0	\overline{R}	\overline{OE}	M	L	\overline{OE}
27	213	8	8	0	0	00	1	1	0	000	E	000	0	1	1	1	0	00	1	1
RA:=SP; SP:=SP-1 (PUSH r)																				
28	133	0	0	0	0	00	0	0	1	004	3	000	0	1	1	1	0	11	1	1
M[SP]:=r																				
29	303	0	8	1	0	00	1	1	0	000	E	000	0	1	1	1	0	00	1	1
SP:=SP+1; RA:=SP (POP r)																				
2A	337	0	0	0	1	00	0	1	1	004	3	000	0	1	1	1	0	11	1	1
r:=M[SP]																				
2B	113	0	7	0	1	00	1	1	1	004	3	000	0	0	1	1	0	00	1	1
Переход, если R7[0]=0 (BEQ A)																				
2C	334	F	6	0	1	00	1	1	1	004	3	000	0	1	1	1	0	00	1	1
R6:=RF (PC:=RK[A]) (BR A)																				
2D	113	0	7	0	1	00	1	1	1	004	3	001	1	0	1	1	0	00	1	1
Переход, если R7[15]=0 (BMI A)																				
2E	334	F	6	0	1	00	1	1	1	004	3	000	0	1	1	1	0	00	1	1
R6:=RF (PC:=RK[A])																				
2F	334	F	6	0	1	00	1	1	1	004	3	000	0	1	1	1	1	00	1	1
R6:=RF (PC:=RK[A]) (HLT A)																				
30	213	8	8	0	0	00	1	1	0	000	E	000	0	1	1	1	0	00	1	1
RA:=SP, SP:=SP-1 (CALL A)																				
31	133	0	6	0	0	00	0	0	1	000	E	000	0	1	1	1	0	00	1	1
M[SP]:=r																				
32	334	F	6	0	1	00	1	1	1	004	3	000	0	1	1	1	0	00	1	1
R6:=RF (PC:=RK[A])																				
33

2.4 Ввод и отладка микропрограммы командного цикла и программ решения задач

- Для ввода и отладки используется программа "Имитатор микропрограммируемой микроЭВМ".
- В программе создаются файлы с расширением ".mvm", содержащие двоичные коды микропрограмм, программ и данных, записанных в ЗУ, ПНА и ПА, а также настройки параметров имитатора.
- Интерфейс пользователя этой программы имеет много общего с интерфейсом программы "Имитатор микропрограммируемого микропроцессора".
- После загрузки и запуска программы она переводится в режим ввода микропрограмм, в котором производится ввод разработанной микропрограммы командного цикла. Отладка микропрограммы осуществляется в режиме выполнения микрокоманд (см. экранную форму).

Окно выполнения микрокоманд

Имитатор микропрограммируемой микроЭВМ - C:\Организация ЭВМ\Суммирование6.mvm

Файл Редактирование Режим Выполнить Скорость ?

ОУ и УУ ОЗУ, ПНА и ПА Настройки

2/8/16cc

№	МИ	РЗУ		Упр. АЛУ		Упр. ОЗУ		Шина		МИ	Упр. усл.		Упр. УУ		Упр. РГК					
МК	I8-0	A	B	CO	^OE	SC	^CS	^W	^EA	D11-0	I3-0	A	U	^CCE	CO	^RLD	^OE	MS12	^LD	^OE
1E	B11	0	0	1	0	00	1	1	1	00A	3	000	0	1	1	1	0	11	1	

r1:= r1-r2 (SUB r1 r2)

Операция: S-R-1+CO R=A S=B Хранение МИ: CJP Условие: всегда истинно

Сдвиги: РЗУ: нет PQ: x Загрузка: РЗУ: F->B PQ: нет Выход АЛУ (Y): F

До выполнения микрокоманды 06:

РЗУ (R0-R7):

0:	00000000000000111
1:	00000000000000101
2:	00000000000000000
3:	00000000000000000
4:	00000000000000010
5:	00000000000011001
6:	00000000000100010
7:	00000000000000000

РЗУ (R8-R15):

8:	0000000011111101
9:	00000000000000000
10:	00000000000000000
11:	00000000000000000
12:	00000001000100010
13:	00000000000000000
14:	00000000111111111
15:	00000000000100000

Y: ZZZZZZZZZZZZZZZZZZ PQ: 00000000000000000

Z F15 OVR C16 PR0 PR15 PQ0 PQ15

После выполнения микрокоманды 06:

РЗУ (R0-R7):

0:	00000000000000111
1:	00000000000000101
2:	00000000000000000
3:	00000000000000000
4:	00000000000000010
5:	00000000000011001
6:	00000000000100010
7:	00000000000000000

РЗУ (R8-R15):

8:	0000000011111101
9:	00000000000000000
10:	00000000000000000
11:	00000000000000000
12:	00000001000100010
13:	00000000000000000
14:	00000000111111111
15:	00000000000100010

Y: ZZZZZZZZZZZZZZZZZZ PQ: 00000000000000000

Z F15 OVR C16 PR0 PR15 PQ0 PQ15

Шина Y: 000000000110 =>0: Стек

PCMK: 000000000111

PA/CC: 000000000000

^PE 0 ^ME 1 ^VE 1 ^FL 1

РГК: 0000001000100010

Шина Y: 000000011110 =>0: Стек

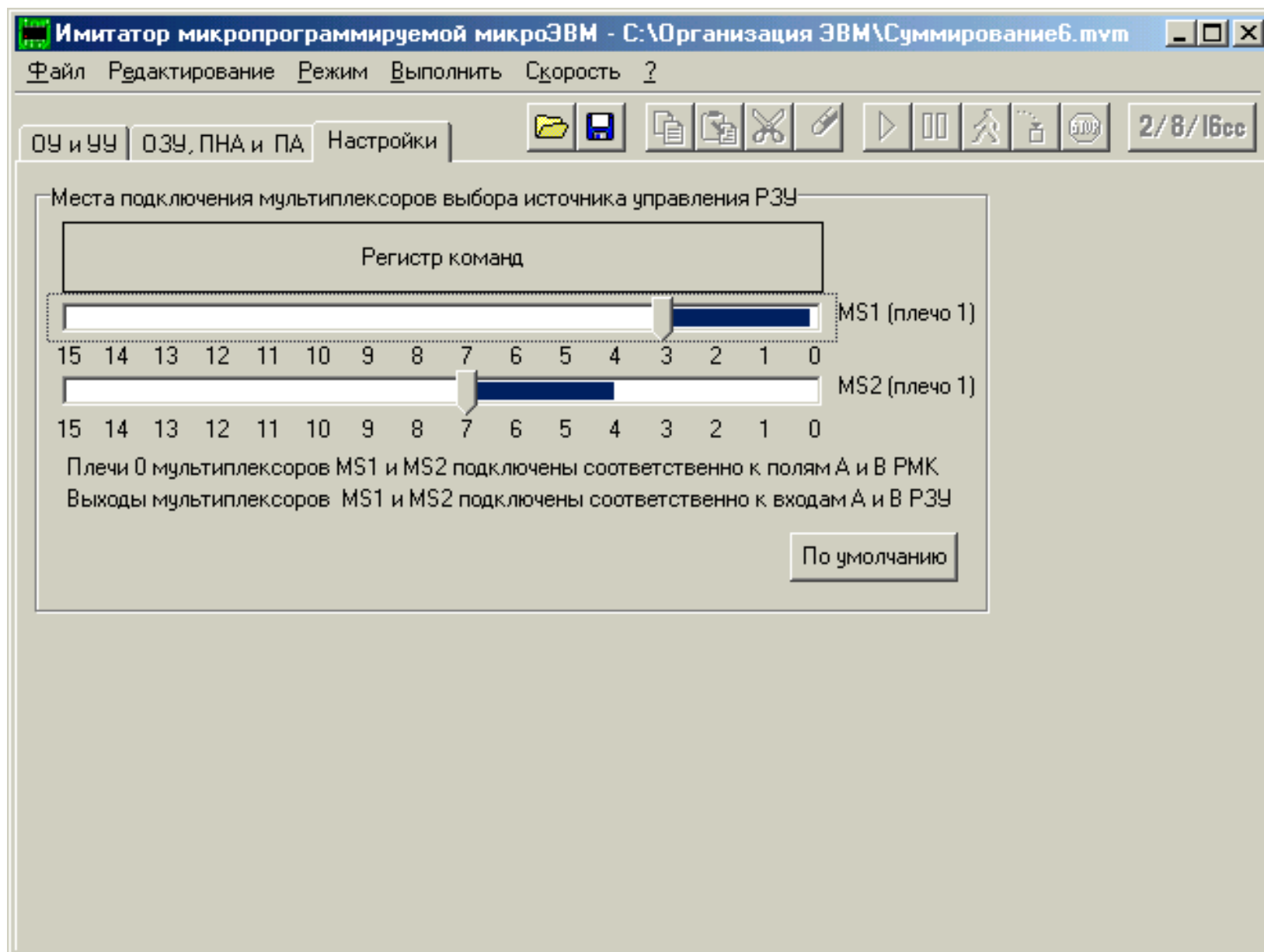
PCMK: 000000011111

PA/CC: 000000000000

^PE 1 ^ME 0 ^VE 1 ^FL 1

РГК: 0000001000100010

Настройка схемы выбора адреса



Экранная форма отладки программы

Имитатор микропрограммируемой микроЭВМ - C:\Организация ЭВМ\Суммирование6.mym

Файл Редактирование Режим Выполнить Скорость ?

ОУ и ЧУ ОЗУ, ПНА и ПА Настройки

2/8/16сс

ОЗУ	Комментарии	ПНА	ПА
00: 1005	LDI RX (RAM)+...	00: 2F	00: 00000000
0E: 8920	CALL AD.....	01: 1F	01: 00000000
0F: 0131	ADD RS RZ.....	02: 1E	02: 00000000
10: A401	SB RN "1".....	03: 27	03: 00000000
11: 0614	BEQ M2.....	04: 29	04: 00000000
12: 050B	BR M3.....	05: 2C	
13: A201	SB RQ "1".....	06: 2B	
14: C304	MV RS AS.....	07: 2D	
15: C205	MV RQ AQ.....	08: 30	
16: 0007	HLT SA.....	09: 20	
17: 0005	Y1.....	0A: 21	
18: 0007	X1.....	0B: 1A	
19: 0008	Y2.....	0C: 1C	
1A: 0009	X2.....	0D: 20	
1B: 0002	Y3.....	0E: 00	
1C: 0007	X3.....	0F: 00	
1D: 0003	Y4.....	10: 22	
1E: 0001	X4.....	11: 37	
1F: 0000	12: 33	
20: 0320	PUSH RQ.....	13: 00	
21: 0222	SUB RZ RZ.....	14: 00	

РЗУ

0:	00000000000000111
1:	00000000000000101
2:	00000000000000000
3:	00000000000000000
4:	00000000000000010
5:	00000000000011001
6:	0000000000100010
7:	00000000000000000
8:	0000000011111101
9:	00000000000000000
10:	00000000000000000
11:	00000000000000000
12:	0000001000100010
13:	00000000000000000
14:	00000000111111111
15:	0000000000100010

PQ: 00000000000000000

Окно аппаратуры прерывания

АП

7 0

Z: 11101111

RZ: 11101111

RM: 11111111

CD: 00000000

RV: 000

DC: 00000001

INT: 0 Y: 222

Прерывания
запрещены.

АП

7 0

Z: 11101111

RZ: 11101111

RM: 00000000

CD: 11101111

RV: 000

DC: 00000001

INT: 1 Y: 222

Определение
запроса с
наивысшим
приоритетом.

АП

7 0

Z: 11101111

RZ: 11101110

RM: 00000000

CD: 11101110

RV: 000

DC: 00000001

INT: 1 Y: 222

Сброс
запроса,
принятого к
обработке.

АП

7 0

Z: 11101111

RZ: 11101110

RM: 11111111

CD: 00000000

RV: 000

DC: 00000001

INT: 0 Y: 222

Прерывания
снова
запрещены.