

# **Структура и рабочий цикл процессора**

## **1 Базовые структуры процессора (ПР).**

1.1 Структура с несколькими устройствами управления.

1.2 Каноническая структура.

## **2 Процессор учебной ЭВМ.**

2.1 Архитектура.

2.2 Структура.

2.3 Рабочий цикл.

## **3 Процессор учебной ЭВМ с развитой системой адресации.**

3.1 Архитектура.

3.2 Рабочий цикл.

3.3 Микропрограммная реализация.

- **Знать:** две базовых структуры процессора: с несколькими устройствами управления и каноническую, их особенности достоинства и недостатки. Архитектуры, структуры и рабочие циклы простейших процессоров, выполняющих основные виды операций и использующих наиболее распространенные способы адресации операндов. Два направления реализации процессора: структурную и микропрограммную и их особенности.
- **Уметь:** для заданной архитектуры разработать структурную схему и микропрограмму командного цикла процессора.
- **Помнить:** о том, что для архитектуры существует максимально соответствующая ей структура.
- **Литература:** [1,14].

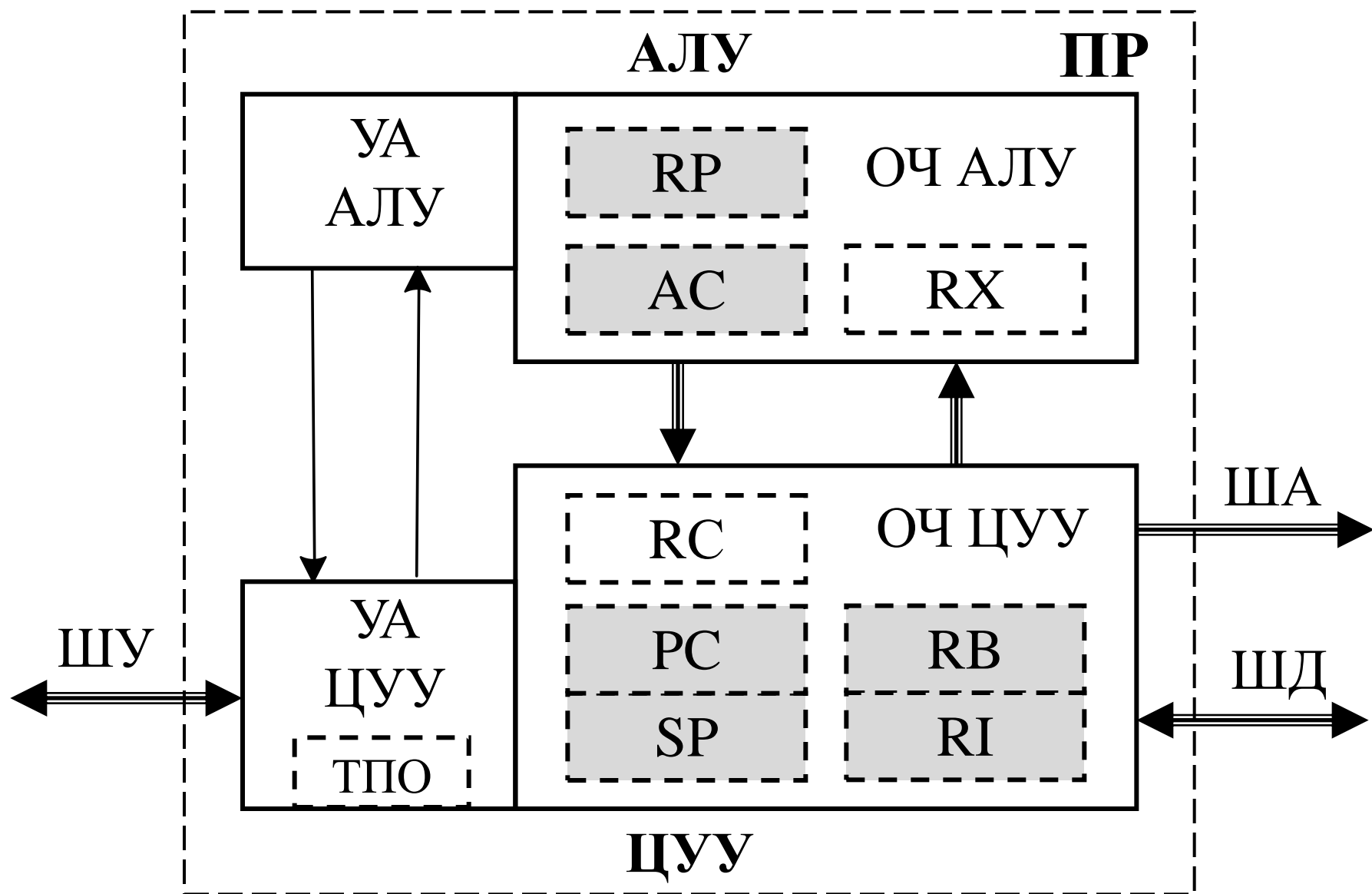
# **1 Базовые структуры процессора**

## **1.1 Структура с несколькими устройствами управления**

- Процессор (ПР) может состоять из нескольких устройств. В простейшем случае ПР состоит из двух устройств: центрального устройства управления (ЦУУ) и арифметико-логического устройства (АЛУ).
- ЦУУ осуществляет выборку всех команд. Кроме того, ЦУУ исполняет команды, если это команды не арифметико-логических операций. Если считанная команда содержит код арифметико-логической операции (АЛО), то ЦУУ обеспечивает выполнение этой операции в АЛУ.
- АЛУ выполняет АЛО над операндами предварительно помещенными в его внутренние регистры и формирует значения признаков в соответствии с результатами выполняемых операций.

- В соответствии с моделью дискретного преобразователя каждое из устройств (ЦУУ и АЛУ) может быть разделено на операционную часть (ОЧ ЦУУ и ОЧ АЛУ) и управляющую часть (управляющий автомат - УА) (УА ЦУУ и УА АЛУ).
- В ОЧ АЛУ могут быть выделены регистры: второго операнда (RX), признаков (RP), аккумулятора (AC),
- В ОЧ ЦУУ могут быть выделены регистры: команд (RC), указателя стека (SP), базы (RB), индекса (RI), а также программный счетчик (PC).
- Процессор имеет следующие внешние шины: адреса (ША), данных (ШД) и управления (ШУ).
- В УА ЦУУ можно выделить специальный триггер пуска-останова (ТПО), установка которого означает запуск, а сброс - останов УА ЦУУ.

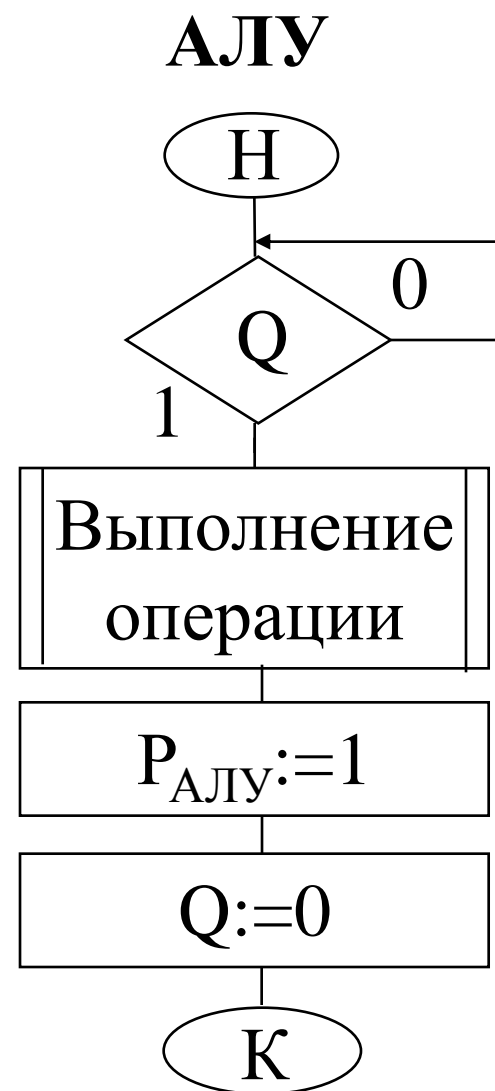
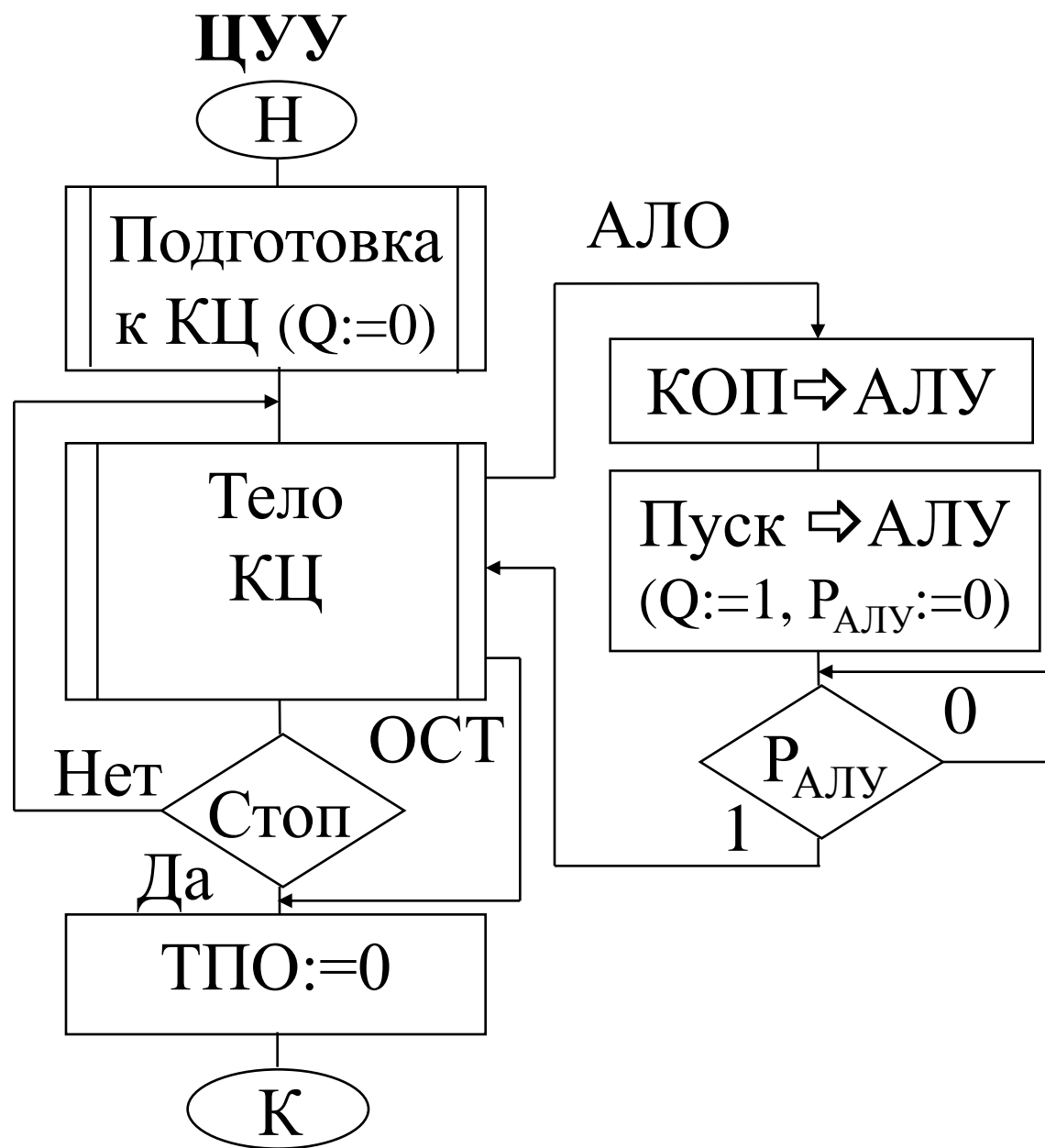
# *Структура, ориентированная на использование нескольких устройств*



# *Рабочий цикл ЦУУ*

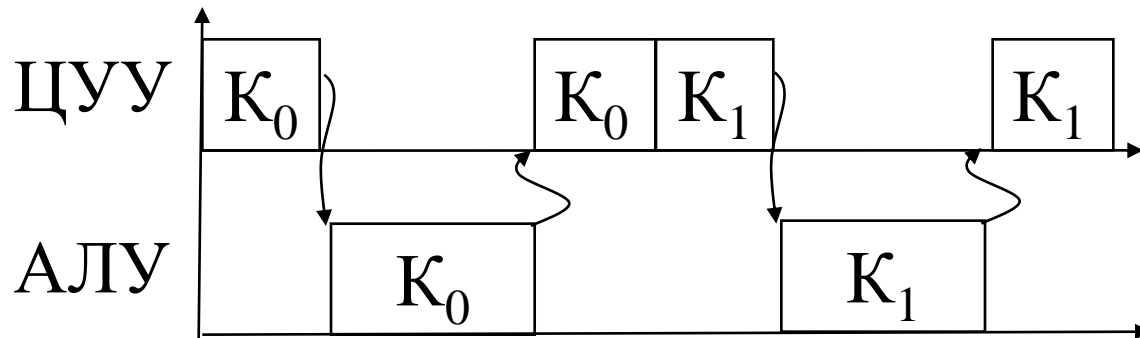
- Рабочему циклу ЦУУ предшествует подготовка (начальные установки установка PS, SP и т.п.).
- Рабочий цикл ЦУУ включает в себя следующие действия.
  - Выборку команды из памяти, формирование адреса следующей команды, дешифрацию КОП.
  - Выполнение операции, заданной в команде, если это не арифметико-логическая операция (АЛО).
  - Выборку операндов АЛО и загрузку их в регистры АЛУ (если это необходимо).
  - Передачу в АЛУ кода АЛО и запуск его на выполнение операции.
  - Запись в память результата АЛО (если это необходимо).
- Микропрограмму командного цикла (КЦ) в процессоре выполняет УА ЦУУ, находящийся на более высоком уровне управления, чем подчиненный ему УА АЛУ.

# Микропрограммы работы ЦУУ и АЛУ

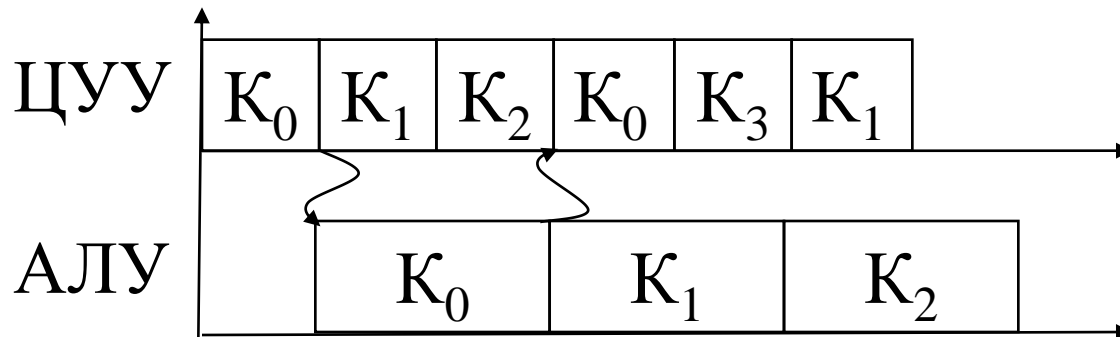


# *Взаимодействие ЦУУ и АЛУ в процессе выполнения команд*

## Выполнение команд АЛО



## Совмещение во времени работы ЦУУ и АЛУ



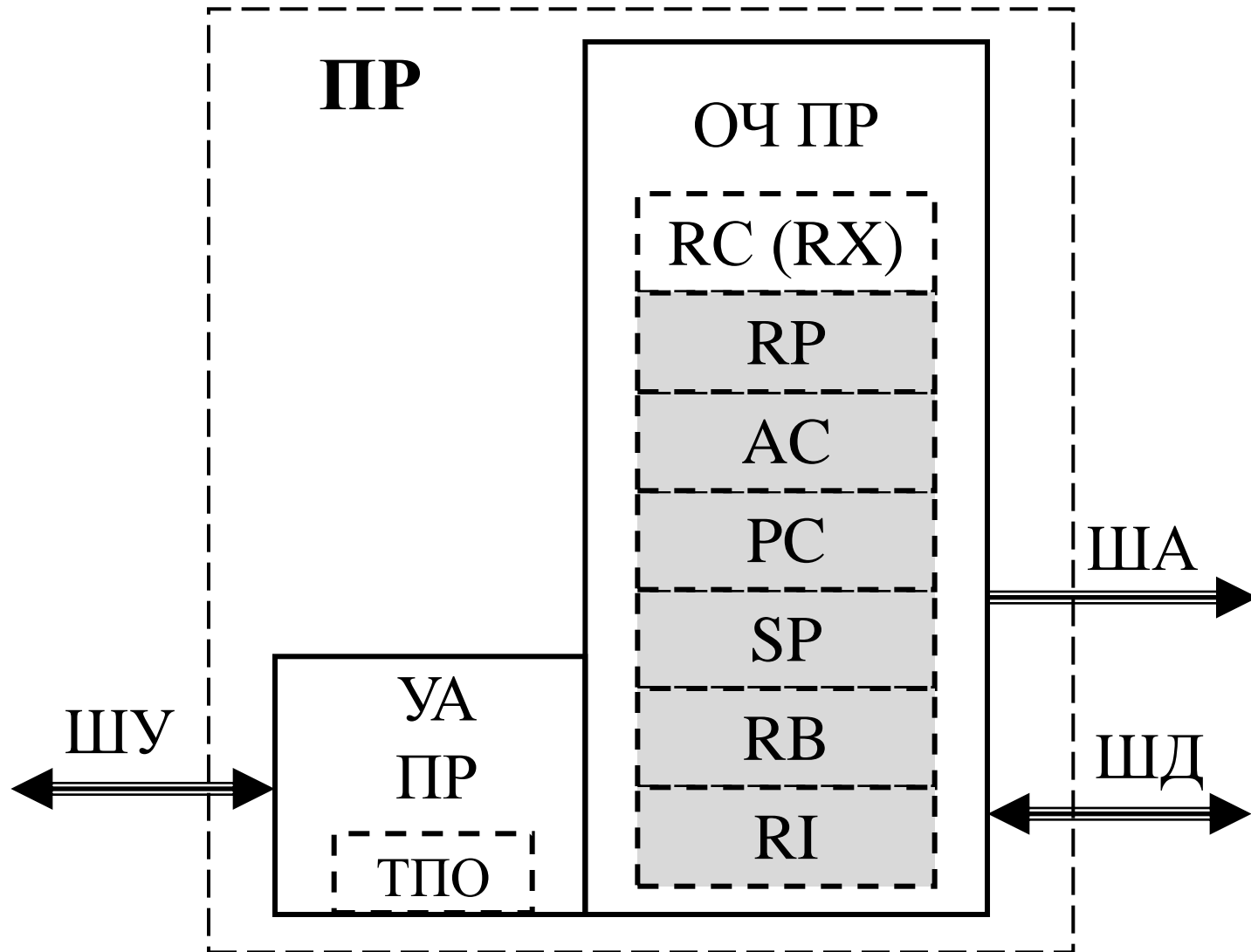
Наличие в ПР двух устройств с собственными управляющими автоматами дает возможность организовать их параллельную работу и повысить быстродействие процессора.



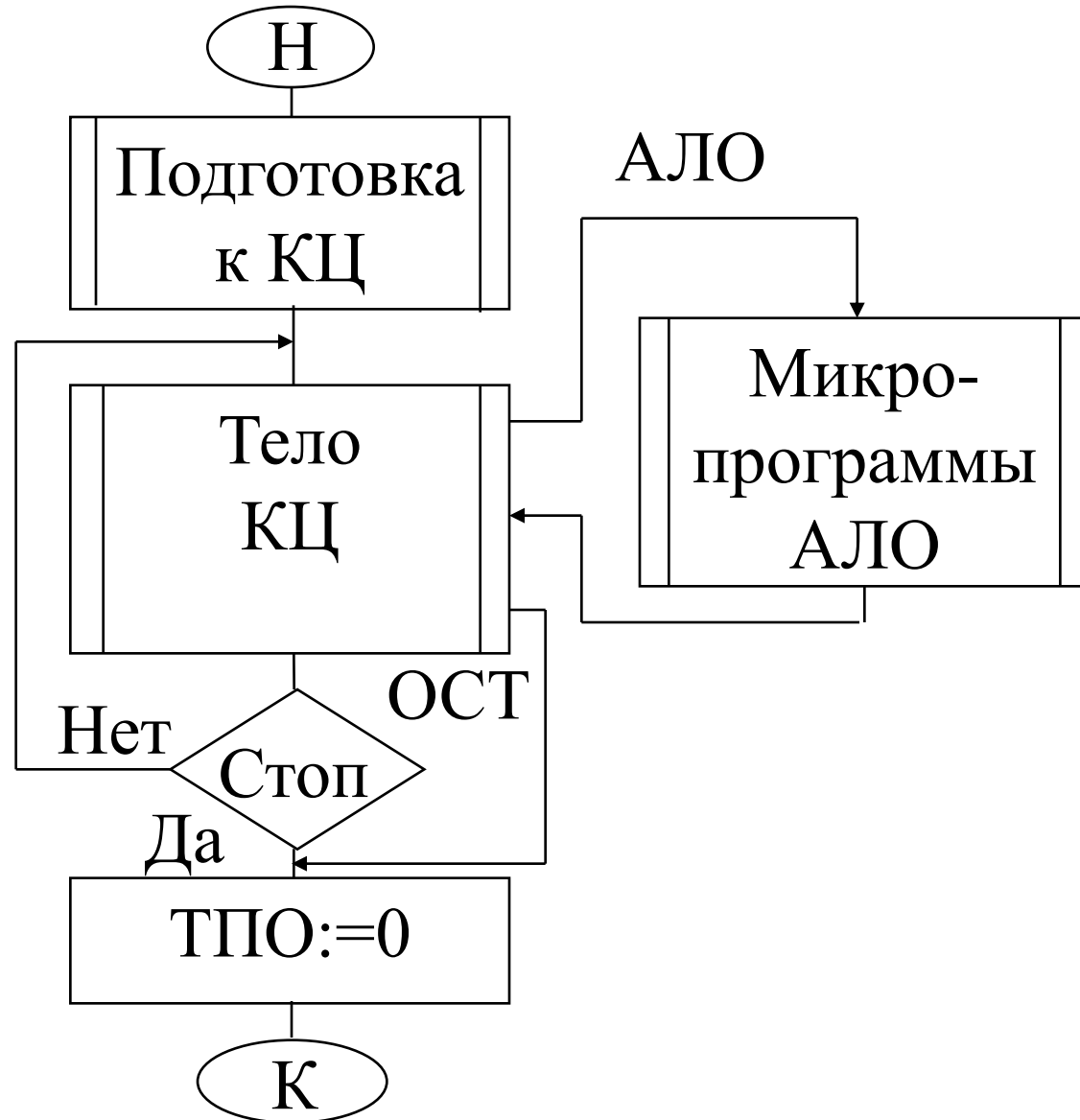
## 1.2 Каноническая структура

- Каноническая структура процессора может быть получена из структуры, содержащей несколько устройств с собственными УА путем функциональной интеграции (ФИ).
- При ФИ происходит совмещение не используемых одновременно регистров в операционных частях всех устройств. Примерами совмещаемых регистров могут служить регистр второго операнда в АЛУ и регистр команд в ЦУУ.
- В результате ФИ управляющих автоматов получается единый управляющий автомат процессора обеспечивающий выполнение всех команд в едином операционном устройстве.

# *Структура с одним управляющим автоматом*



# *Микропрограмма работы УА ПР*



## *Достоинства канонической структуры*

- Функциональная интеграция позволяет сократить объем аппаратуры ПР.
- При достаточном числе унифицированных регистров в ОУ и достаточном объёме памяти микропрограмм в УУ с программируемой логикой в такой структуре микропрограммно может быть реализована любая архитектура процессора.
- Использование данной структуры позволяет создавать процессоры с динамичной архитектурой, которые в процессе работы могут изменять свою архитектуру путём перезагрузки ОЗУ микропрограмм.

# 2 Процессор учебной ЭВМ

## 2.1 Архитектура

- Форматы данных. Данные представляются 16-разрядным двоичным кодом, старший разряд которого определяет знак числа.
- Программистская структура.

AC		
PS	RP	PZ
PC		

Аккумулятор

Регистр признаков

Программный счетчик

- Система команд. Команда состоит содержит два поля: код операции К и адреса А (см. таблицу).

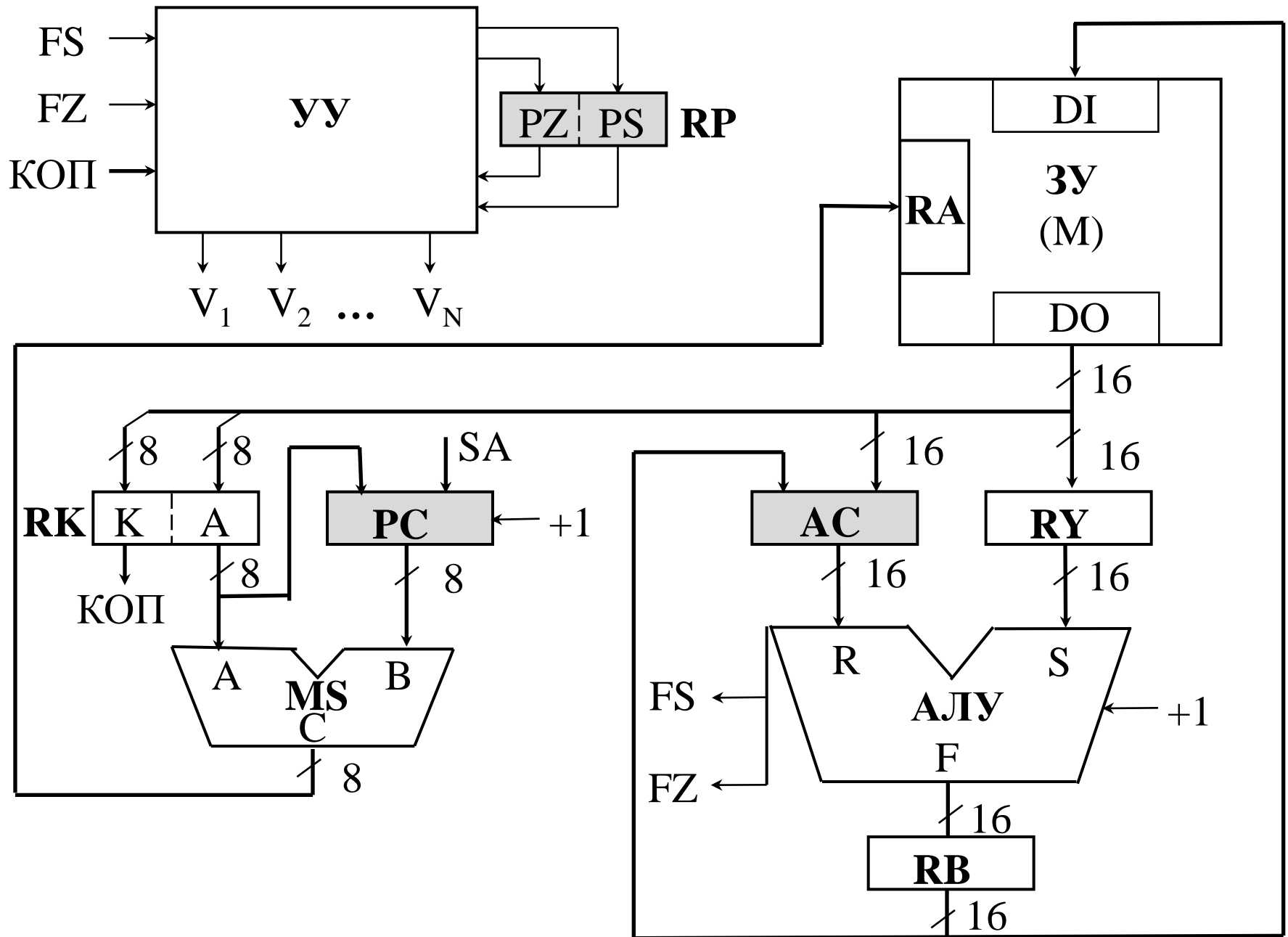
Наименование	Мнемоника	Описание	Признаки	
			PZ	PS
ВЫЧИТАНИЕ	SUB A	AC:=AC-M[A], PC:=PC+1	+	+
ОЧИСТКА	CLM A	M[A]:=00...0, PC:=PC+1	—	—
ЗАПИСЬ ЕДИНИЦ	MVC A	M[A]:=11...1, PC:=PC+1	—	—
ЗАПИСЬ AC	MOV A	M[A]:=AC, PC:=PC+1	—	—
ИНКРЕМЕНТ	INC A	M[A]:= M[A]+1, PC:=PC+1	—	—
ЗАГРУЗКА AC	LDA A	AC:= M[A], PC:=PC+1	+	+
НЕТ ОПЕРАЦИИ	NOP	PC:=PC+1	—	—
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	—	—
ОСТАНОВ	HLT A	PC:=A, останов	—	—

+ - установка, — - сохранение признака

## 2.2 Структура

- ПР, кроме программно-доступных регистров АС, РС, РР, содержит программно недоступные регистры. Это, например, регистр команд (RK) и регистры операндов (RY, RB), используемые для временного хранения команд и операндов в процессе работы ПР.
- Связи между регистрами ЭВМ и схемы преобразования данных определяются требуемыми пересылками и преобразованиями данных и команд.
- Структурная схема ЭВМ приведена ниже, где FS, FZ – флаги соответственно "знака" и "нуля"; V1, V2, ..., VN – управляющие сигналы, вырабатываемые УУ при выполнении команды.
- Разряды PZ, PS регистра РР, устанавливаются после выполнения определенных команд, а состояния флагов FZ, FS изменяются после выполнения каждой МК в зависимости от результата операции в АЛУ.

# Структурная схема процессора

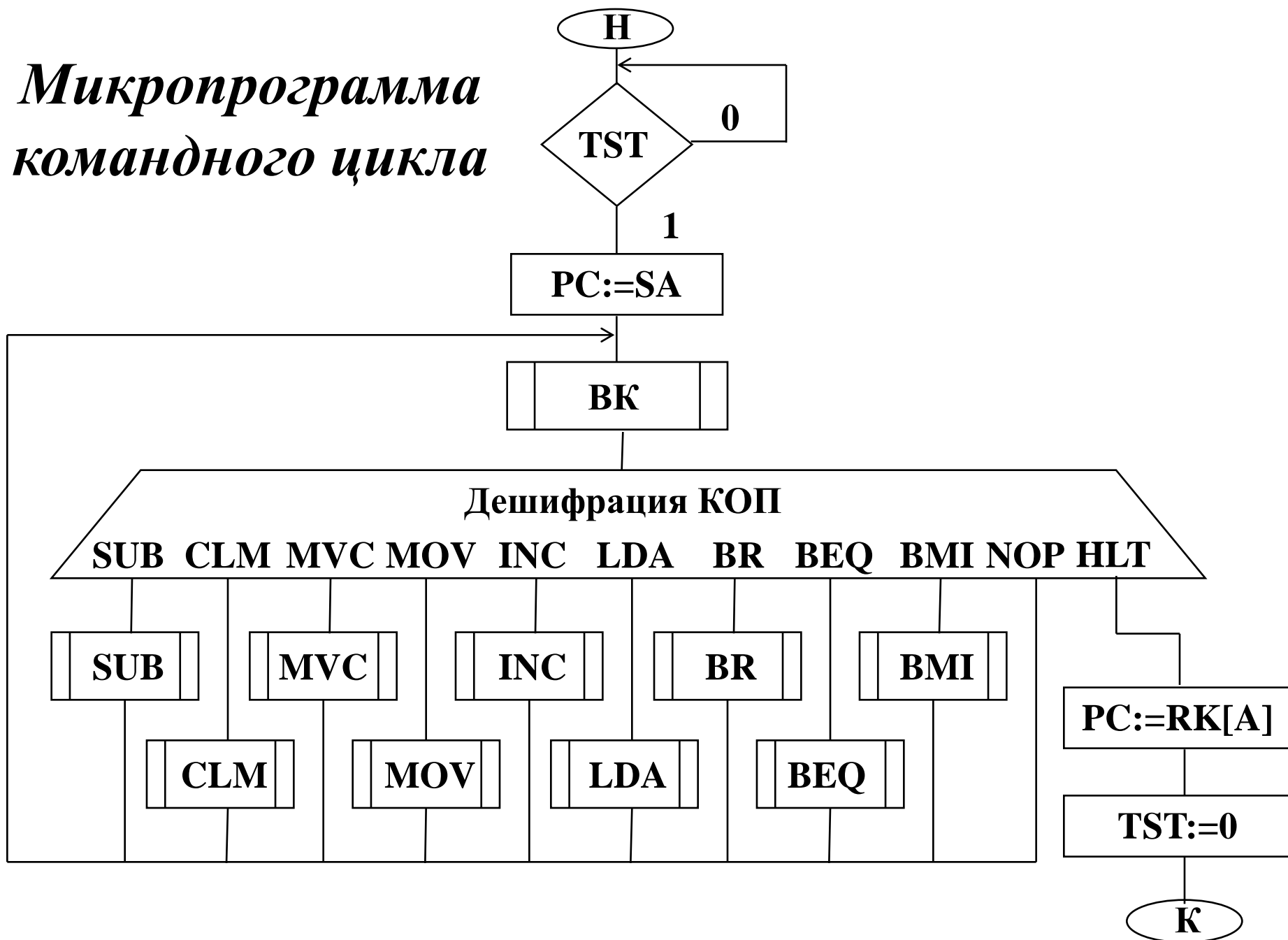




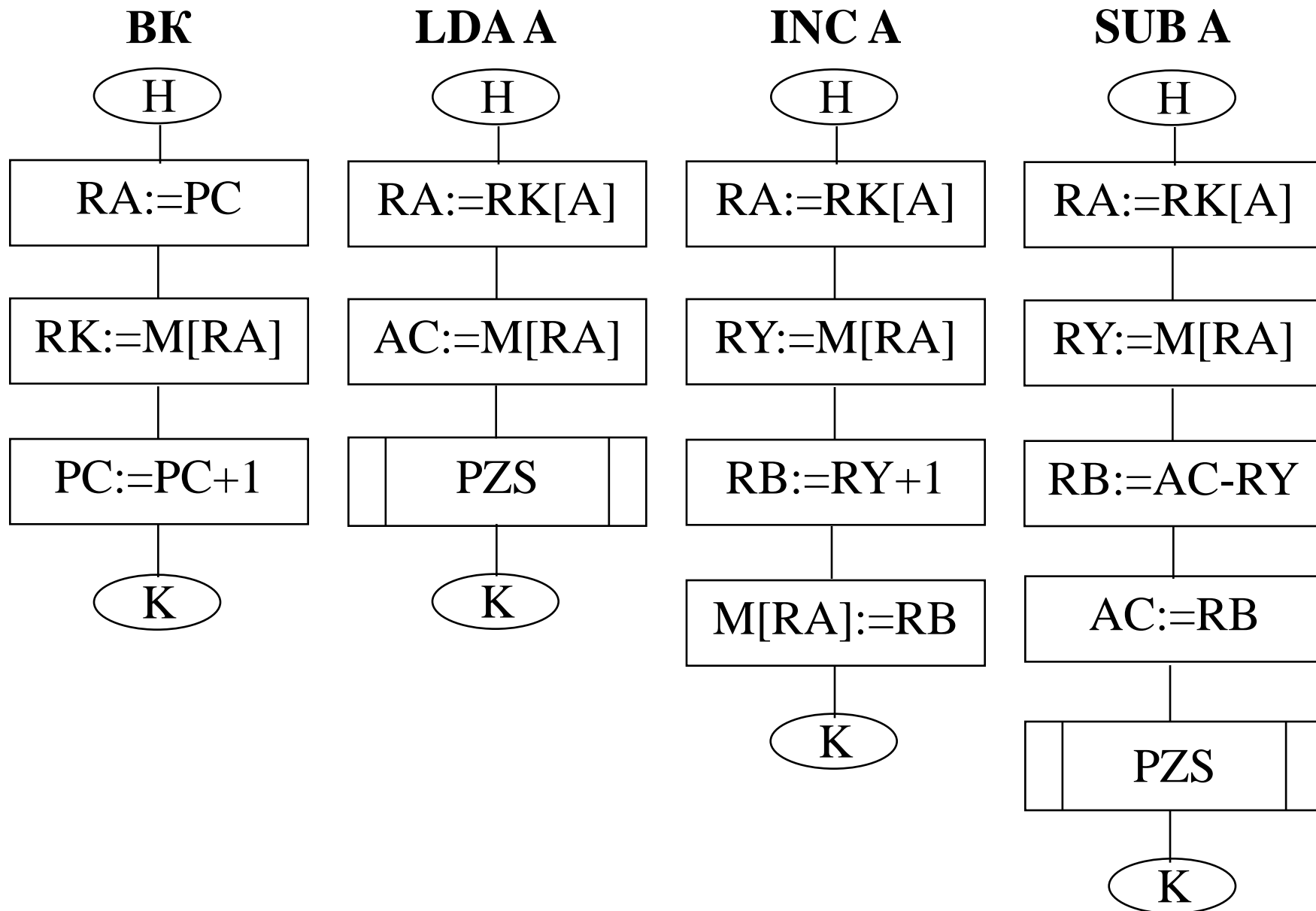
## 2.3 Рабочий цикл

- Алгоритм работы ЭВМ представлен на рисунке ниже в виде укрупненной граф-схемы микропрограммы командного цикла, содержащей подмикропрограмму выборки команды (ВК) и подмикропрограммы заданных операций.
- Подготовка к циклу включает состояние ожидания сигнала пуска (установки специального флага:  $TST=1$ ) и загрузку начального адреса программы (SA) в программный счетчик. В рассматриваемом примере командный цикл включает три этапа: выборку команды, дешифрацию кода операции и выполнение заданной операции.
- Выход из командного цикла производится при выполнении команды HLT.
- Граф-схемы подмикропрограмм, выделенных в микропрограмме командного цикла, приведены на рисунках далее, где PZS - подмикропрограмма установки признаков PZ и PS в регистре признаков RP.

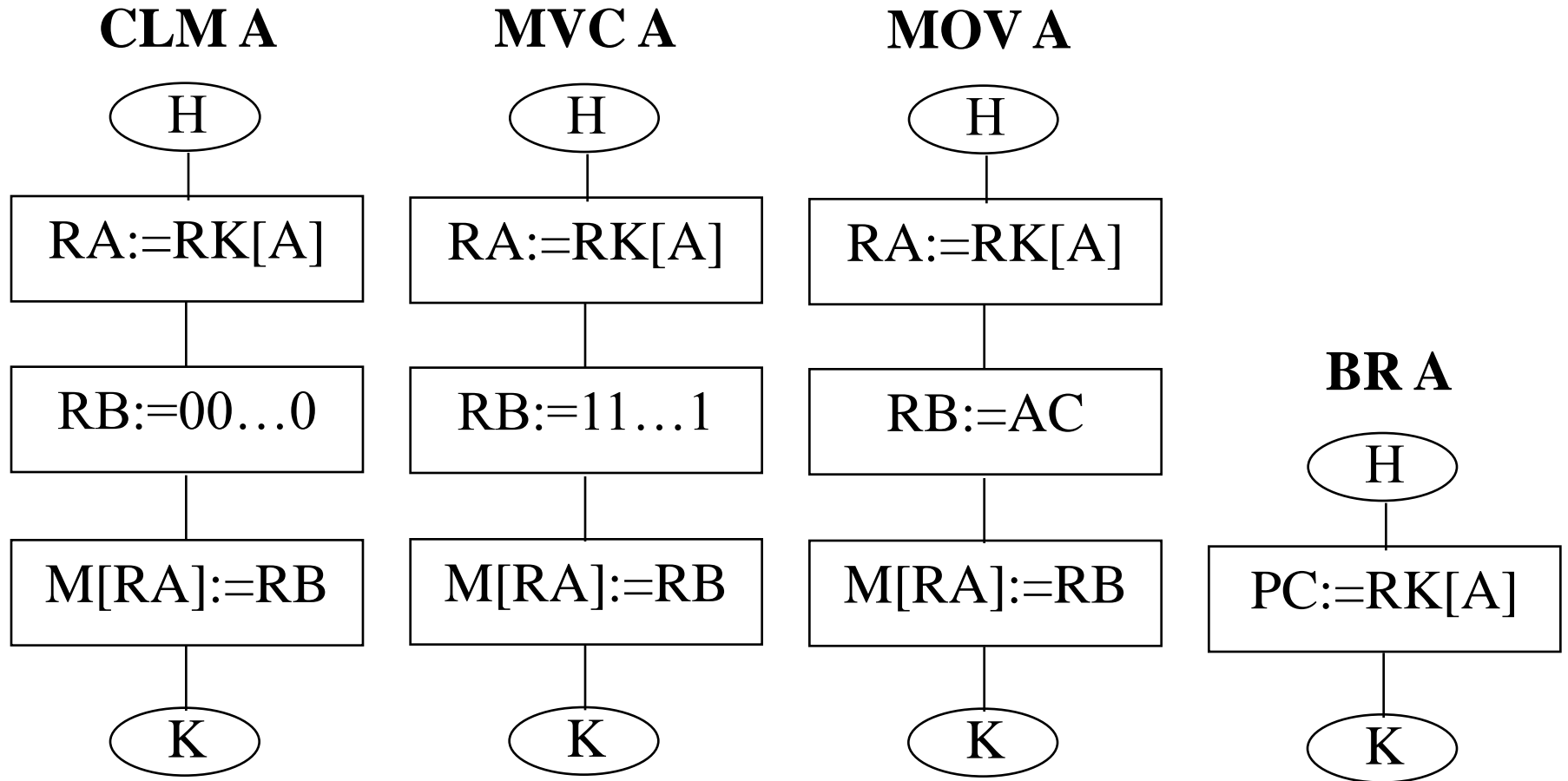
# *Микропрограмма командного цикла*



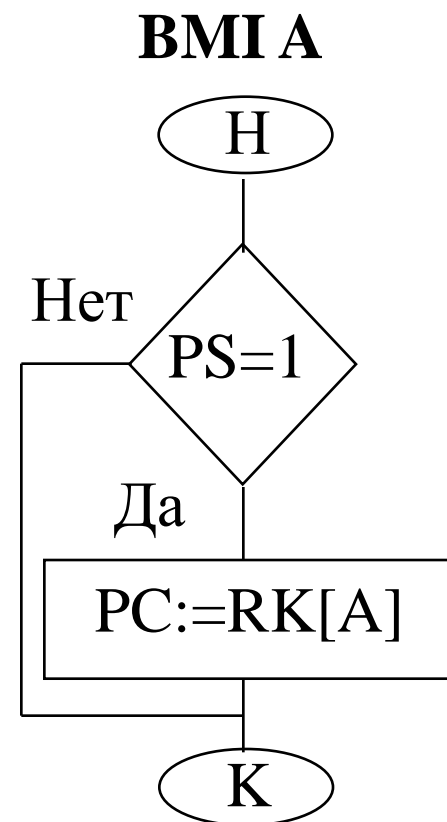
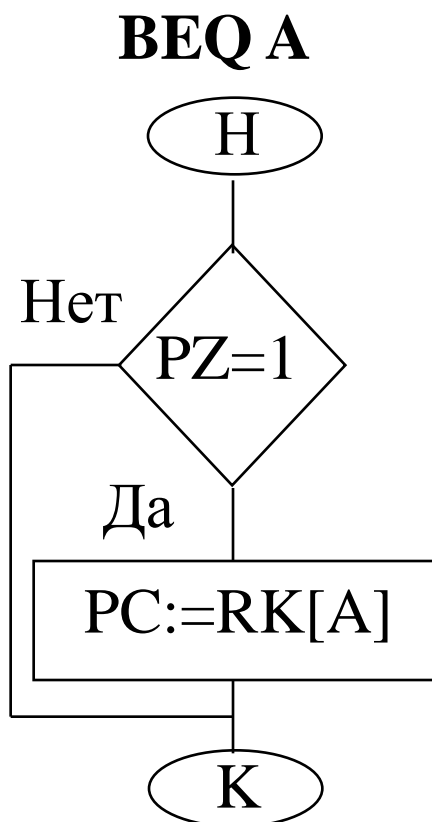
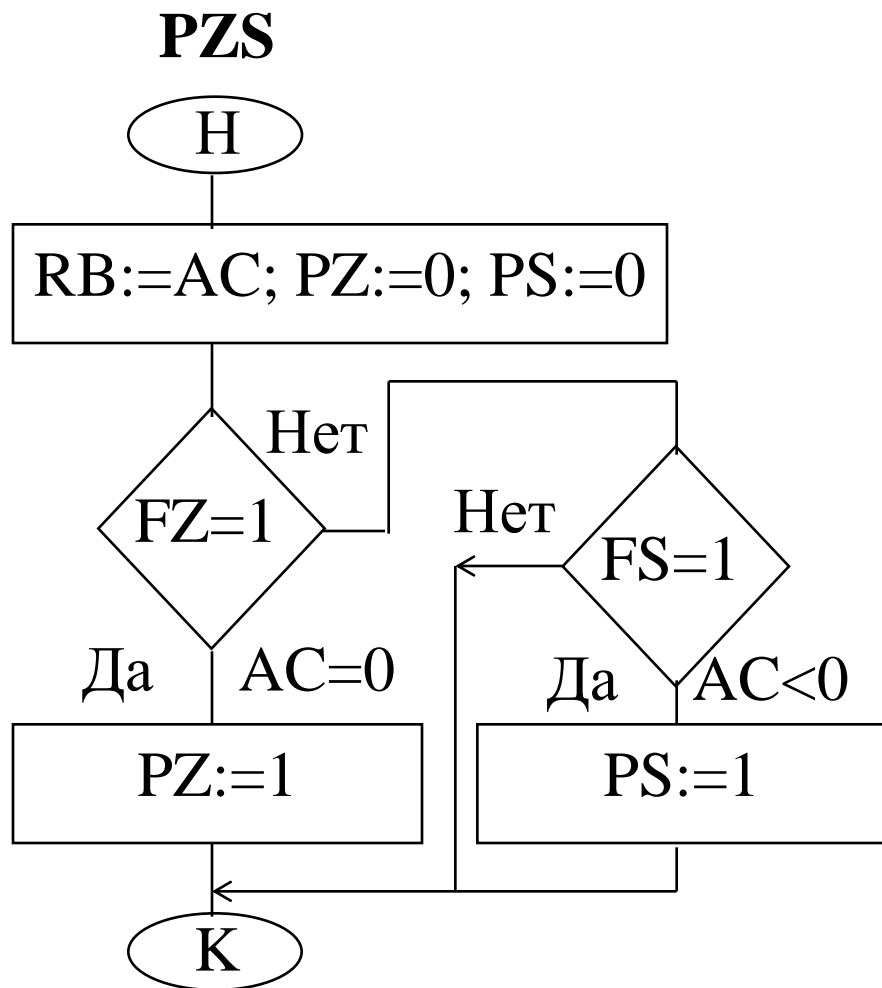
# *МП выборки команды и МП операций*



# *Микропрограммы операций*



# *МП установки признаков и МП операций условных переходов*



# **3 Процессор учебной ЭВМ с развитой системой адресации**

## **3.1 Архитектура**

- Форматы данных. Данные представляются 16-разрядным двоичным кодом, старший разряд которого определяет знак числа.
- Программно-доступные регистры. ЭВМ имеет девять 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), а также регистр указателя стека — rSP (r8).

# *Программистская структура*



- Система команд. Разработка системы команд предполагает определение набора операций, способов адресаций, модификаций и форматов команд.
- Для рассматриваемого примера система команд ЭВМ приведена ниже в таблице. В таблице использованы следующие обозначения:
  - $r, r^* \in \{r_0, r_1, \dots, r_8\}$  – программно-доступные регистры:
  - регистр  $r^*$  является источником данных, а регистр  $r$  – приемником результата, но может также служить источником второго операнда;
  - $M[A]$  – 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.



# *Арифметические команды и команды пересылки*

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	+	+

# *Команды управления и работы со стеком*

Наименование	Мнемоника	Описание	PZ	PS
ЗАПИСЬ В СТЕК	PUSH r (rSP)-	$M[rSP] := r;$ $rSP := rSP - 1, PC := PC + 1$	—	—
ЧТЕНИЕ ИЗ СТЕКА	POP r (rSP)+	$rSP := rSP + 1;$ $r := M[rSP], PC := PC + 1$	—	—
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	—	—

# *Особенности системы команд*

- Чтение данных в регистр возможно из ячейки памяти, адрес которой содержится в регистре ( $\text{LDI } r \ (r^*)+$ ), при этом номера регистров источника и приемника задаются в команде, а адрес после выполнения операции увеличивается на единицу.
- В командах чтения из стека ( $\text{POP } r \ (r\text{SP})+$ ) и записи в стек ( $\text{PUSH } r \ (r\text{SP})-$ ) содержимого регистра  $r$  в качестве накопителя стека используются область памяти, а указатель стека размещается в специальном регистре –  $r\text{SP}$ .
- В систему команд включена команда обращения к подпрограмме ( $\text{CALL } (r\text{SP})- \ A$ ). Эта команда сохраняет содержимое программного счетчика в стеке и обеспечивает переход на начальный адрес подпрограммы ( $A$ ). Функция команды возврата из подпрограммы может быть выполнена командой чтения из стека в программный счетчик.

# *Способы адресации*

- – *Прямая (абсолютная)* – в адресной части команды указан адрес ячейки памяти, к которой происходит обращение при выполнении команды. Применяется в командах: LD r A, MV r A (по второму адресу).
- – *Непосредственная* – в адресной части команды содержится операнд, используемый при выполнении команды. Применяется в командах AD r C, SB r C (по второму адресу), а также в командах BR A, BEQ A, BMI A, HLT A, CALL (rSP)- A (по второму адресу), где в качестве операнда выступает адрес перехода A, над которым выполняется операция пересылки в регистр PC.
- – *Регистровая неявная* – номер регистра в команде не указывается, а определяется кодом операции: BEQ A, BMI A, CALL (rSP)-, HLT A – регистр PC.

# *Способы адресации (продолжение)*

- *Регистровая прямая* — в адресной части команды содержатся номера регистров: ADD r r\*, SUB r r\* (первый и второй адрес), AD r C, SB r C, LD r A, LDI r (r\*)+, PUSH r (rSP)-, POP r (rSP)+, MV r A (первый адрес).
- — *Регистровая косвенная автоинкрементная* — в адресной части команды указан номер регистра, содержащего адрес ячейки памяти, после обращения к которой в процессе выполнения команды содержимое регистра увеличивается на единицу (LDI r (r\*)+, POP r (rSP)+ (второй адрес)).
- — *Регистровая косвенная автодекрементная* — в адресной части команды указан номер регистра, содержащего адрес ячейки памяти, до обращения к которой в процессе выполнения команды содержимое регистра уменьшается на единицу (PUSH r (rSP)- (второй адрес), CALL (rSP)- A (первый адрес)).

# Форматы команд

Ф

15 14...12 11...8 7...4 3...0

0	K1	r	r*	ADD, SUB, LDI, PUSH, POP
---	----	---	----	--------------------------

0	K2	A	BR, BEQ, BMI, HLT
---	----	---	-------------------

1	K3	r	C	AD, SB
---	----	---	---	--------

1	K4	r	A	LD, MV, CALL
---	----	---	---	--------------

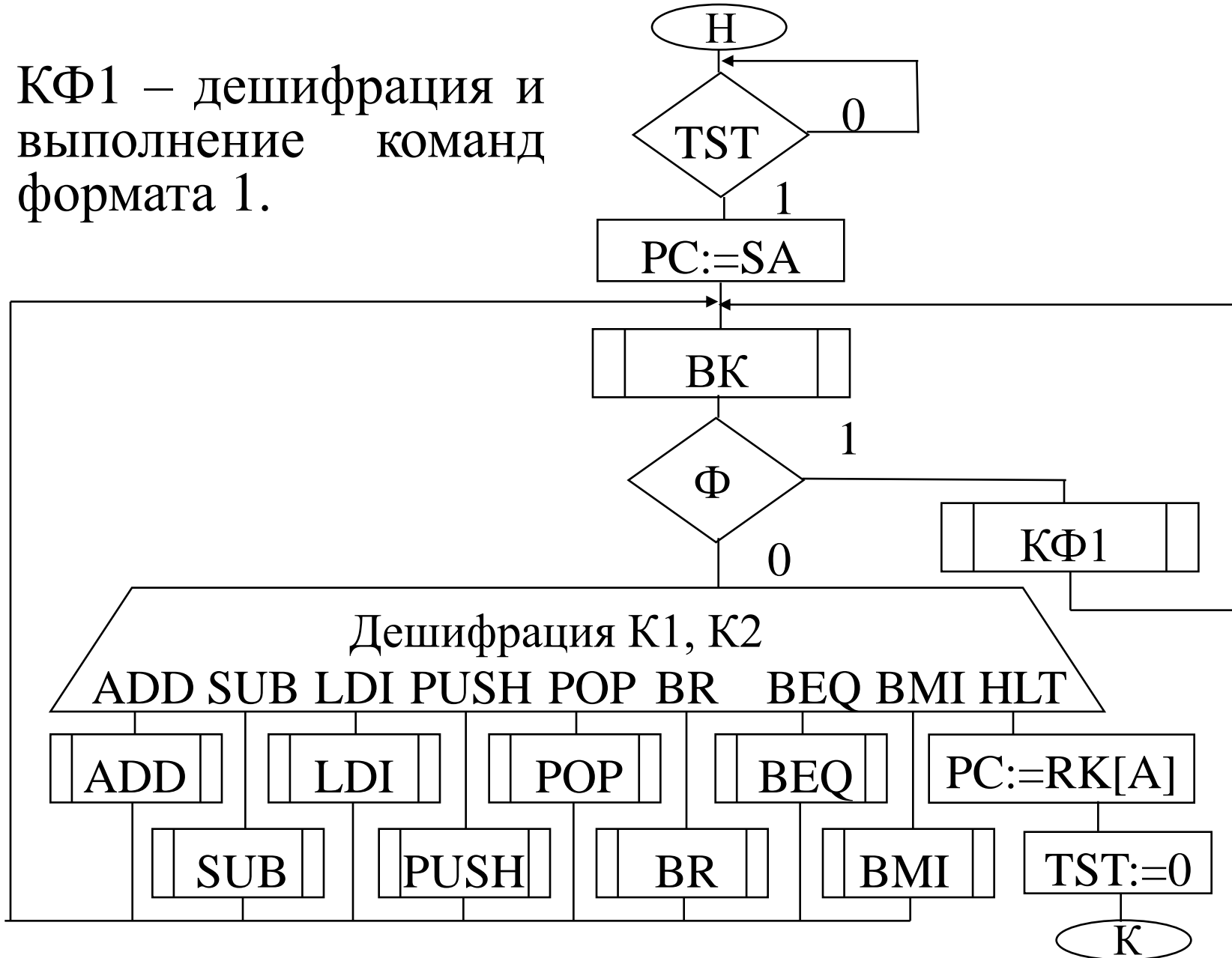
Ф – поле признака формата команды; K1, K2, K3, K4 – поля кода операции; r, r\* – номера регистров; A – адрес; C – непосредственный операнд.

## 3.2 Рабочий цикл

- Алгоритм работы ЭВМ представлен на рисунке ниже в виде укрупненной граф-схемы микропрограммы командного цикла, содержащие подмикропрограммы: выборки команды (ВК), анализа признака формата (Ф), дешифрации кода операции (для команд первого формата ( $\Phi=0$ ), заданных операций; дешифрации и выполнения команд (КФ1) второго формата ( $\Phi=1$ ),
- Подготовка к циклу включает состояние ожидания сигнала пуска (установки специального флага –  $TST=1$ ) и загрузку начального адреса программы (SA) в программный счетчик.
- Командный цикл для команд первого формата ( $\Phi=0$ ) включает три этапа: выборку команды, дешифрацию кода операции и выполнение заданной операции.
- Команды второго формата выполняются за четыре этапа, так как добавляется еще этап распаковки команд (подмикропрограмма РК).
- Выход из командного цикла производится при выполнении команды HLT.

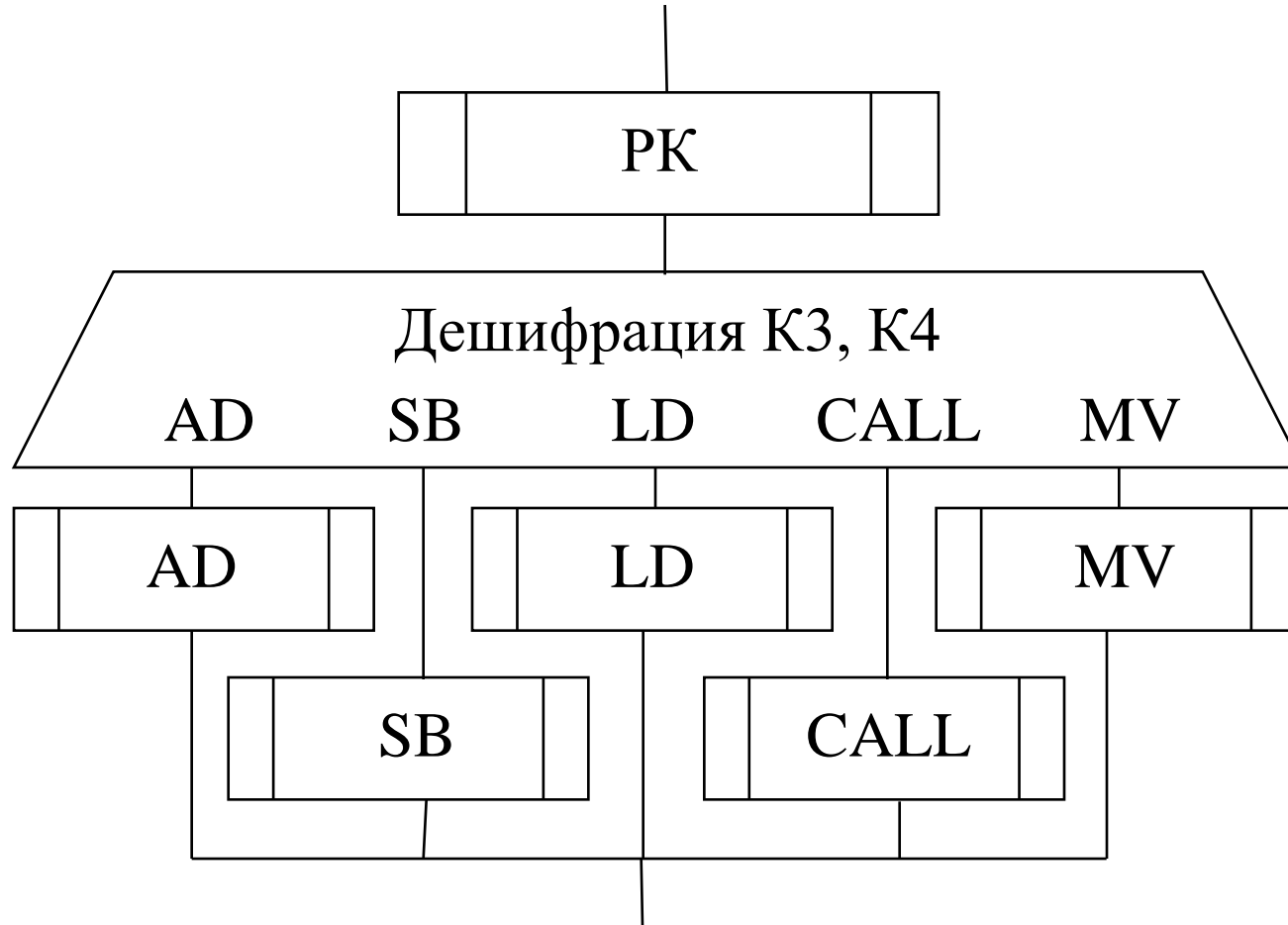
# Микропрограмма командного цикла

КФ1 – дешифрация и выполнение команд формата 1.





# *Микропрограмма дешифрации и выполнения команд формата 1*



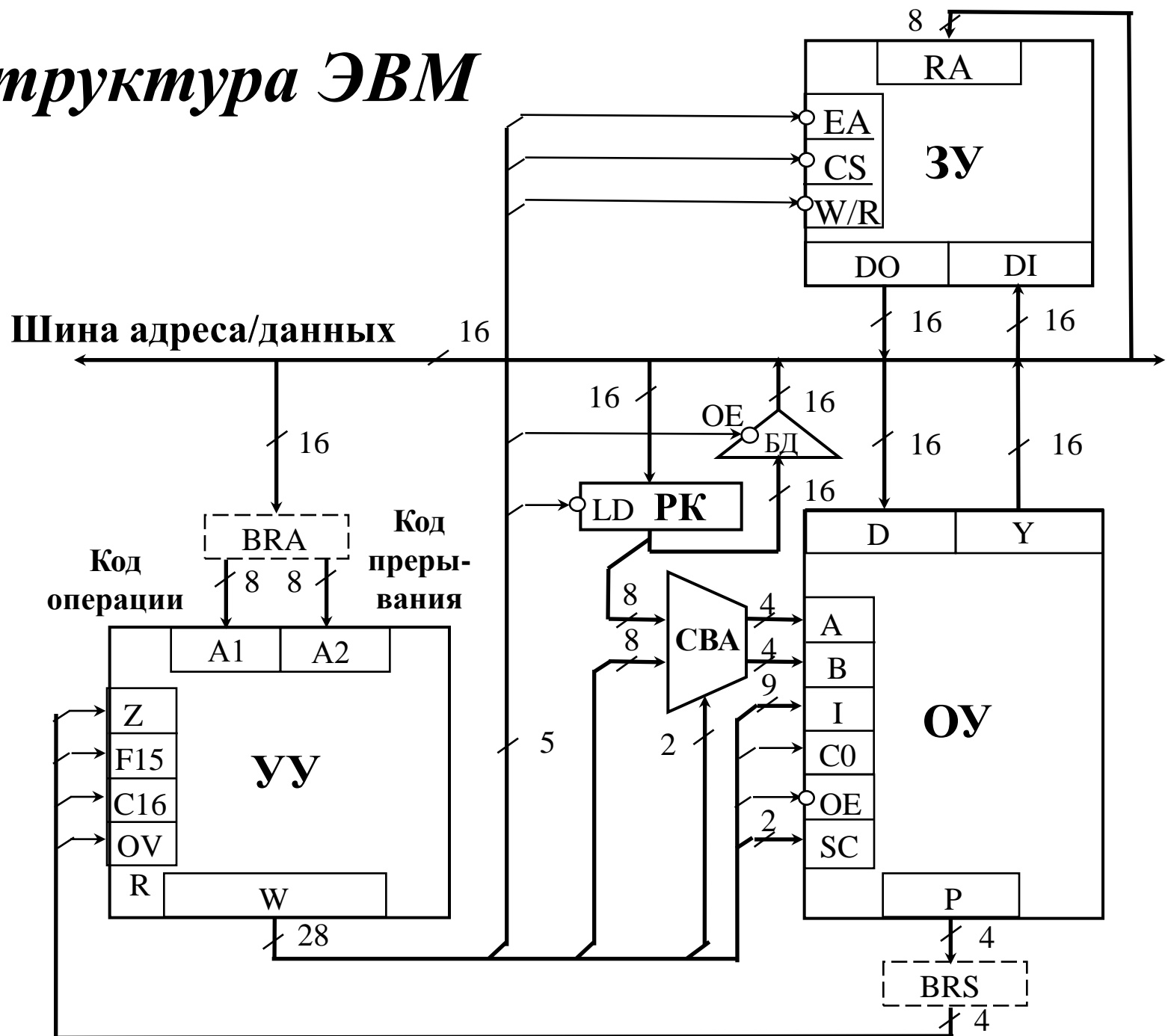
ПК – распаковка команд.

## 3.3 Микропрограммная реализация ЭВМ

### *Структура ЭВМ*

- ЭВМ может быть реализована микропрограммно с использованием аппаратных средств микропрограммируемого микропроцессора. Структура необходимых аппаратных средств показана ниже на рисунке.
- Микропрограммная реализация ЭВМ включает:
  - распределение внутренних регистров микропроцессора,
  - разработку и кодирование подмикропрограмм командного цикла,
  - распределение памяти микропрограмм.

# Структура ЭВМ



# *Регистры ЭВМ*

- Распределение внутренних регистров микропроцессора показано далее на рисунке.
- Кроме девяти программно-доступных регистров r0-r8 в состав ЭВМ входят пять программно-недоступных регистров (два для команды, и по одному для константы, счетчика адреса ЗУ, операнда Y).
- Программно-доступные регистры отображаются на регистры (R0-R8), а программно-недоступным регистрам соответствуют регистры R13-R15 и RQ операционного устройства микропроцессора.
- Кроме того, дополнительными программно-недоступными регистрами являются регистр адреса ЗУ (RA) и регистр команд (RK).

# *Распределение внутренних регистров*

РЗУ (R0-R7)

0:	r0		
1:	r1		
2:	r2		
3:	r3		
4:	r4		
5:	r5		
6:	r6 (PC)		
7:	PS	r7 (RP)	PZ

RA:	Адрес ЗУ
RK:	Регистр команд

РЗУ (R8-R15)

8:	r8 (SP)		
9:			
10:			
11:			
12:			
13:	Буферный регистр команд		
14:	Регистр константы		
15:	Счетчик адреса ЗУ		

RQ:	Регистр Y
-----	-----------

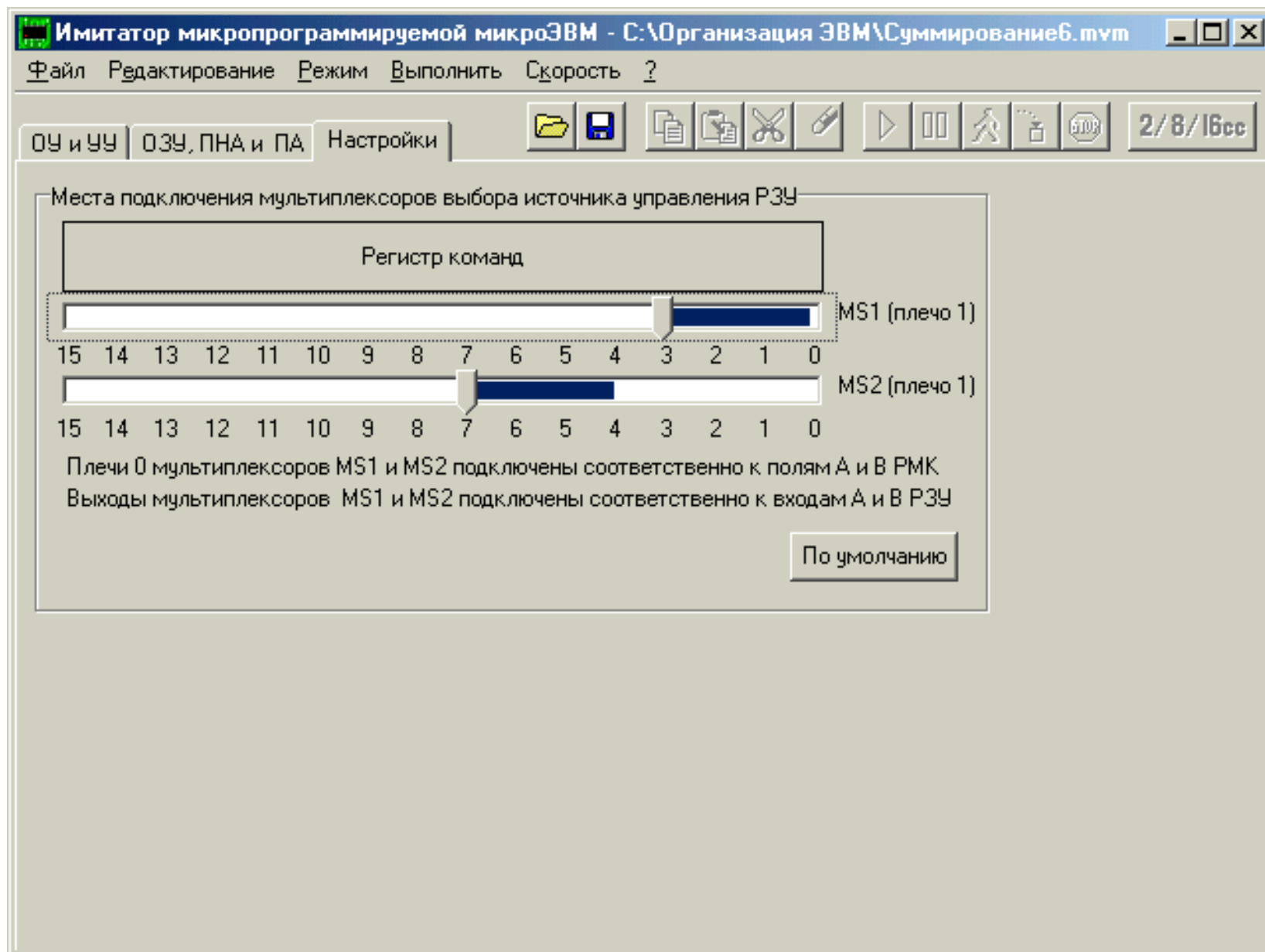
# *Дешифрация кода операции*

Мнемоника	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	26
POP	04	2B
BR	05	32
BEQ	06	31
BMI	07	33
CALL	8	36
HLT	00	35

# *Принятая настройка схемы выбора адреса (СВА)*

<b>MS1</b> (управляю щий вход СВА)	<b>A</b> (адресный вход РЗУ)		<b>MS2</b> (управляю щий вход СВА)	<b>B</b> (адресный вход РЗУ)
0	PMK[A]		0	PMK[B]
1	PK[3:0]		1	PK[7:4]

# Вкладка "Настройки"





## *Фрагмент микропрограммы КЦ*

№	МИ	РЗУ		Упр. АЛУ			Упр. ОЗУ			Шина	МИ	Упр. усл.			Упр. УУ			Упр. РК		
М К	I8-0	A	B	C0		S C				D11-0	I3-0	A	U	$\overline{OE}$	C0		$\overline{OE}$	M	L	$\overline{OE}$
00	571	E	E	0	0	00	1	1	1	006	C	000	0	0	1	1	0	00	1	1
RE:=0111111111111111; PA/CI:=6																				
01	533	0	E	0	0	00	1	1	1	001	9	000	0	0	1	1	0	00	1	1
RE – сдвиг вправо; PA/CI:=PA/CI-1																				
02	143	0	6	0	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1	1
RA:=0																				
03	337	0	6	0	1	00	0	1	0	000	E	000	0	0	1	1	0	00	1	1
R6:=SA (PC:=SA)																				
04	203	6	6	1	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1	1
RA:=R6; R6:=R6+1 (RA:=PC; PC:=PC+1)																				
05	337	0	C	0	1	00	0	1	1	007	3	001	1	0	1	1	0	00	0	1
RK:=K; RC:=K																				
06	345	E	F	0	1	00	1	1	1	000	2	000	0	0	1	1	0	00	1	0
RF:=K[A]      Переход по КОП																				