

Пример построения микроЭВМ с развитой системой адресации. Лабораторная работ №2

1 Пример задания

2 Разработка архитектуры ЭВМ

2.1 Форматы данных и программно-доступные регистры

2.2 Программистская структура

2.3 Система команд

2.4 Форматы команд

3 Рабочий цикл

4 Микропрограммная реализация ЭВМ

1 Пример задания

- Определить архитектуру, разработать и отладить микропрограмму командного цикла ЭВМ, составить и выполнить программу вычисления суммы частных S :

$$S = \sum_{i=1}^N Z_i, \quad Z_i = \left\lfloor \frac{X_i}{Y_i} \right\rfloor,$$

- где Z_i – частное от деления целых чисел X_i и Y_i нацело, $X_i, Y_i, Z_i \in [0 \dots 32767]$. Кроме результата S , необходимо формировать и записывать в ОЗУ значения признака переполнения Q , которые могут возникнуть при делении чисел.
- Для деления чисел использовать подпрограмму на основе программы, составленной при выполнении предыдущей лабораторной работы.
- Обмен данными между программой и подпрограммой должен производиться через стек.

2 Разработка архитектуры ЭВМ

2.1 Форматы данных и программно-доступные регистры

- 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).

2.2 Программистская структура



2.3 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)).

Коды операций

| Мнемоника | Код операции |
|-----------|--------------|
| ADD | 01 |
| SUB | 02 |
| AD | 9 |
| SB | A |
| LD | B |
| MV | C |
| LDI | 10 |
| PUSH | 03 |
| POP | 04 |
| BR | 05 |
| BEQ | 06 |
| BMI | 07 |
| CALL | 8 |
| HLT | 00 |

Основная программа

| | | | |
|----|------|--------------|--|
| 06 | B801 | LD r8 AASP | Загрузка регистра указателя стека SP |
| 07 | B502 | LD r5 AAM | Загрузка адреса массива AM в регистр r5 |
| 08 | B403 | LD r4 AN | Загрузка числа повторений цикла N |
| 09 | 0233 | SUB r3 r3 | Очистка регистра для суммы S |
| 0A | 0222 | SUB r2 r2 | Очистка регистра для признака Q |
| 0B | 1015 | LDI r1 (r5)+ | Чтение делителя Y в регистр r1 |
| 0C | 0713 | BEQ m1 | Если PZ=1 (Y=0), то переход на метку m1 (13) |
| 0D | 1005 | LDI r0 (r5)+ | Чтение делимого X в регистр r0 |
| 0E | 8920 | CALL AD | Обращение к подпрограмме по адресу AD |
| 0F | 0131 | ADD r3 r1 | Суммирование |
| 10 | A401 | SB r4 "1" | Вычитание единицы из числа N |
| 11 | 0614 | BEQ m2 | Если N=0, то переход на метку m2 (14) |
| 12 | 050B | BR m3 | Переход на метку m3 (0D) |
| 13 | A201 | SB r2 "1" | Запись единиц в регистр признака Q |
| 14 | C304 | MV r3 AS | Запись суммы S адресу AS |
| 15 | C205 | MV r2 AQ | Запись признака Q адресу AQ |
| 16 | 0007 | HLT SA | Загрузка PC и останов |

Программа деления чисел нацело

| Подпрограмма деления чисел нацело | | | |
|-----------------------------------|------|-----------|--|
| 20 | 0320 | PUSH r2 | Сохранение содержимого регистра Q |
| 21 | 0222 | SUB r2 r2 | Очистка регистра для частного Z |
| 22 | 0201 | SUB r0 r1 | Вычитание из делимого X делителя Y |
| 23 | 0726 | BMI m1 | Если PS=1, то переход на метку m1 (26) |
| 24 | 9201 | AD r2 "1" | Увеличение на единицу частного Z |
| 25 | 0522 | BR m2 | Переход на метку m2 (22) |
| 26 | 0211 | SUB r1 r1 | Очистка регистра rY |
| 27 | 0112 | ADD r1 r2 | Запись частного Z в регистр r1 |
| 28 | 0420 | POP r2 | Восстановление содержимого регистра rQ |
| 29 | 0460 | POP r6 | Возврат из подпрограммы |

2.4 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 |
|---|----|---|---|--------------|

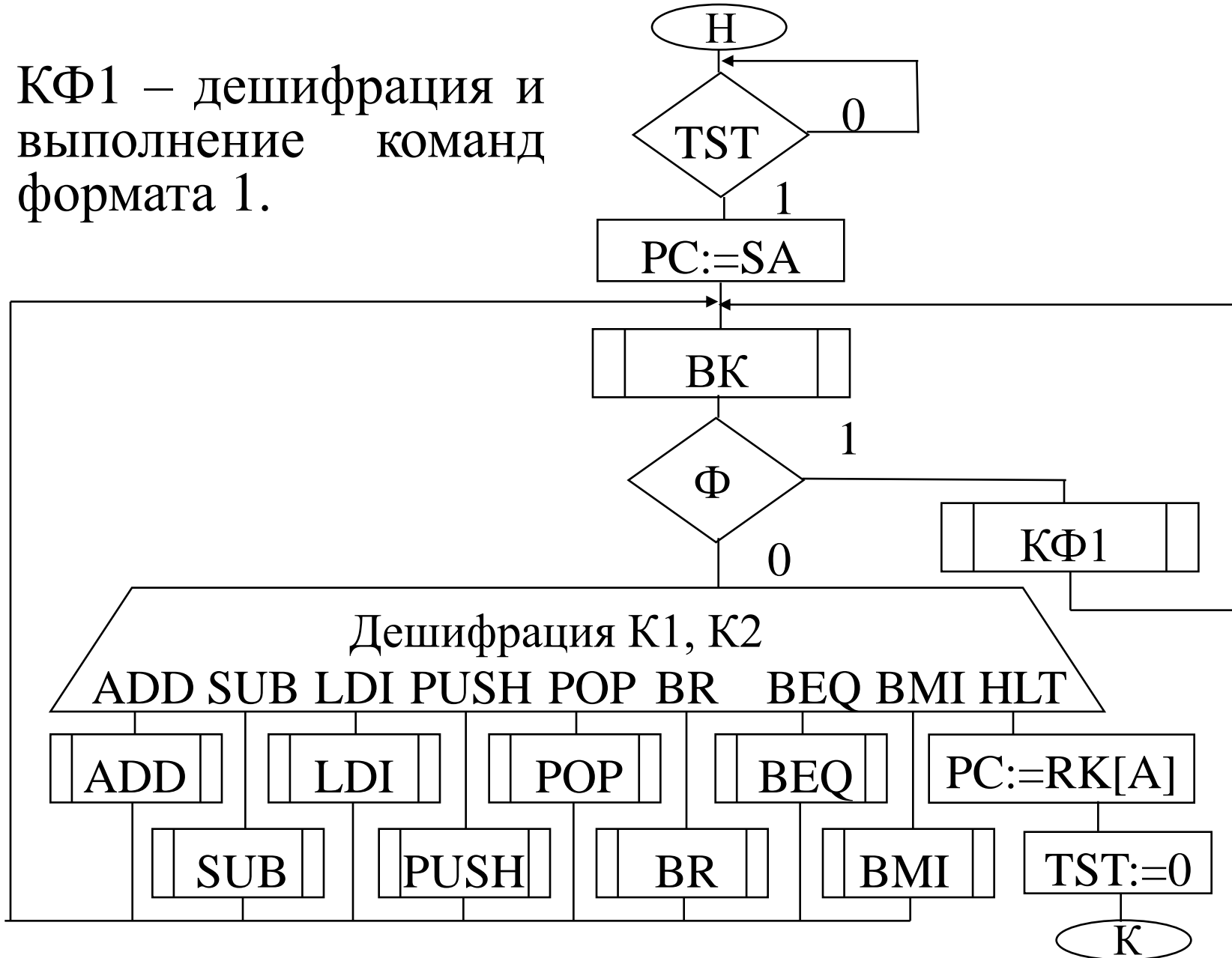
Ф – поле признака формата команды; K1, K2, K3, K4 – поля кода операции; r, r* – номера регистров; A – адрес; C – непосредственный операнд.

3 Рабочий цикл

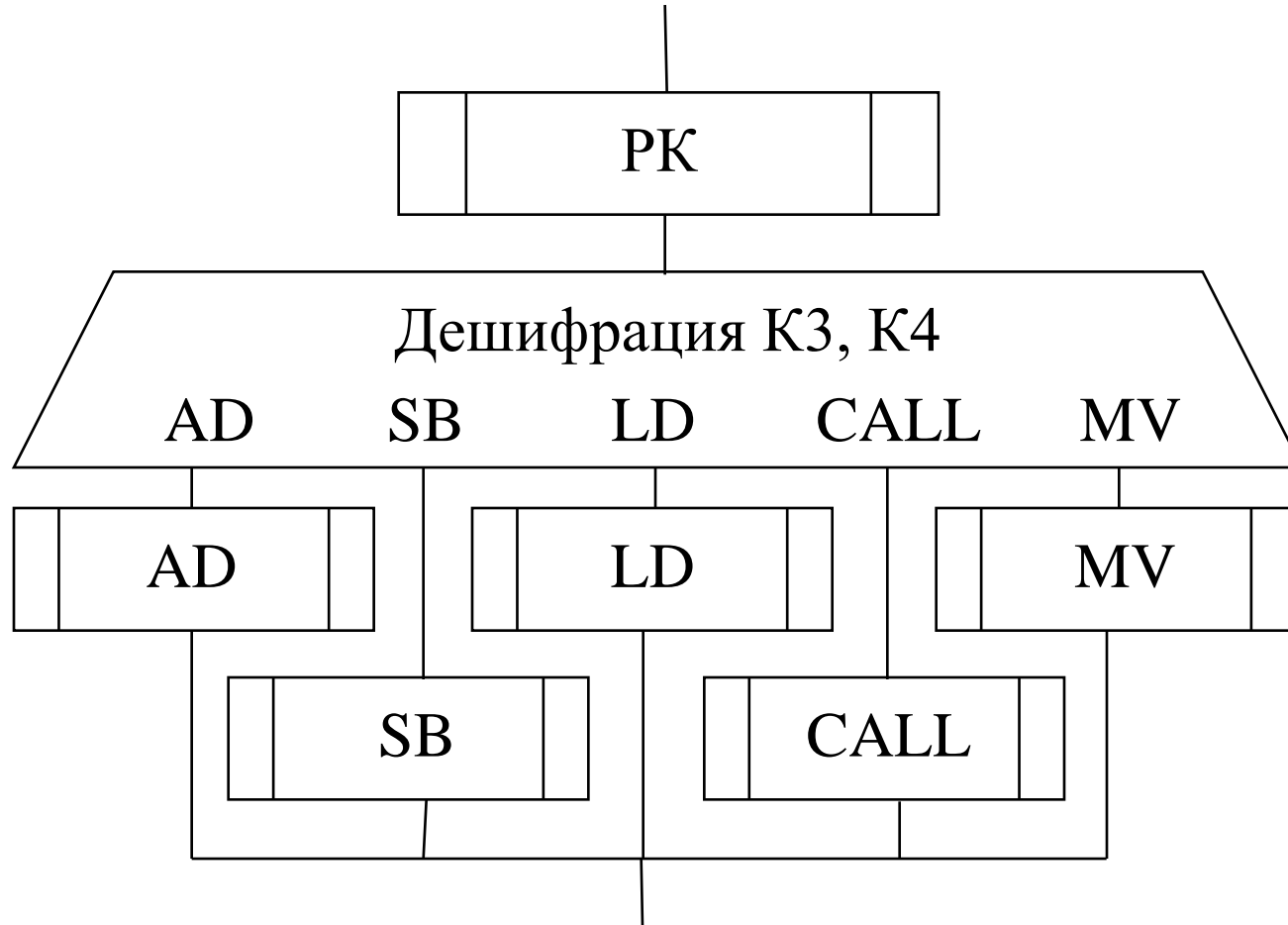
- Алгоритм работы ЭВМ представлен на рисунке ниже в виде укрупненной граф-схемы микропрограммы командного цикла, содержащие подмикропрограммы: выборки команды (ВК), анализа признака формата (Ф), дешифрации кода операции (для команд первого формата ($\Phi=0$), заданных операций; дешифрации и выполнения 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

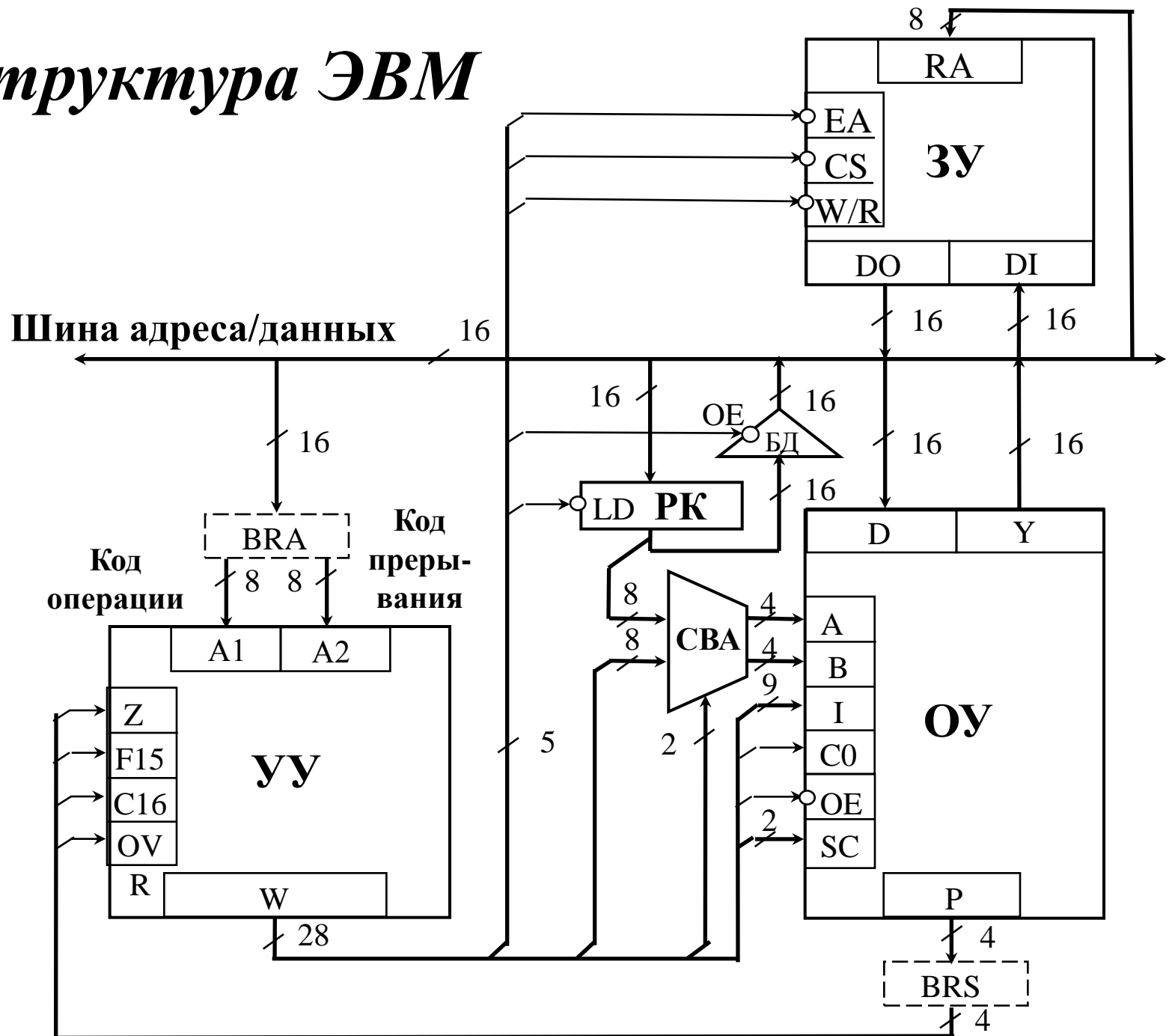


ПК – распаковка команд.

4 Микропрограммная реализация 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)

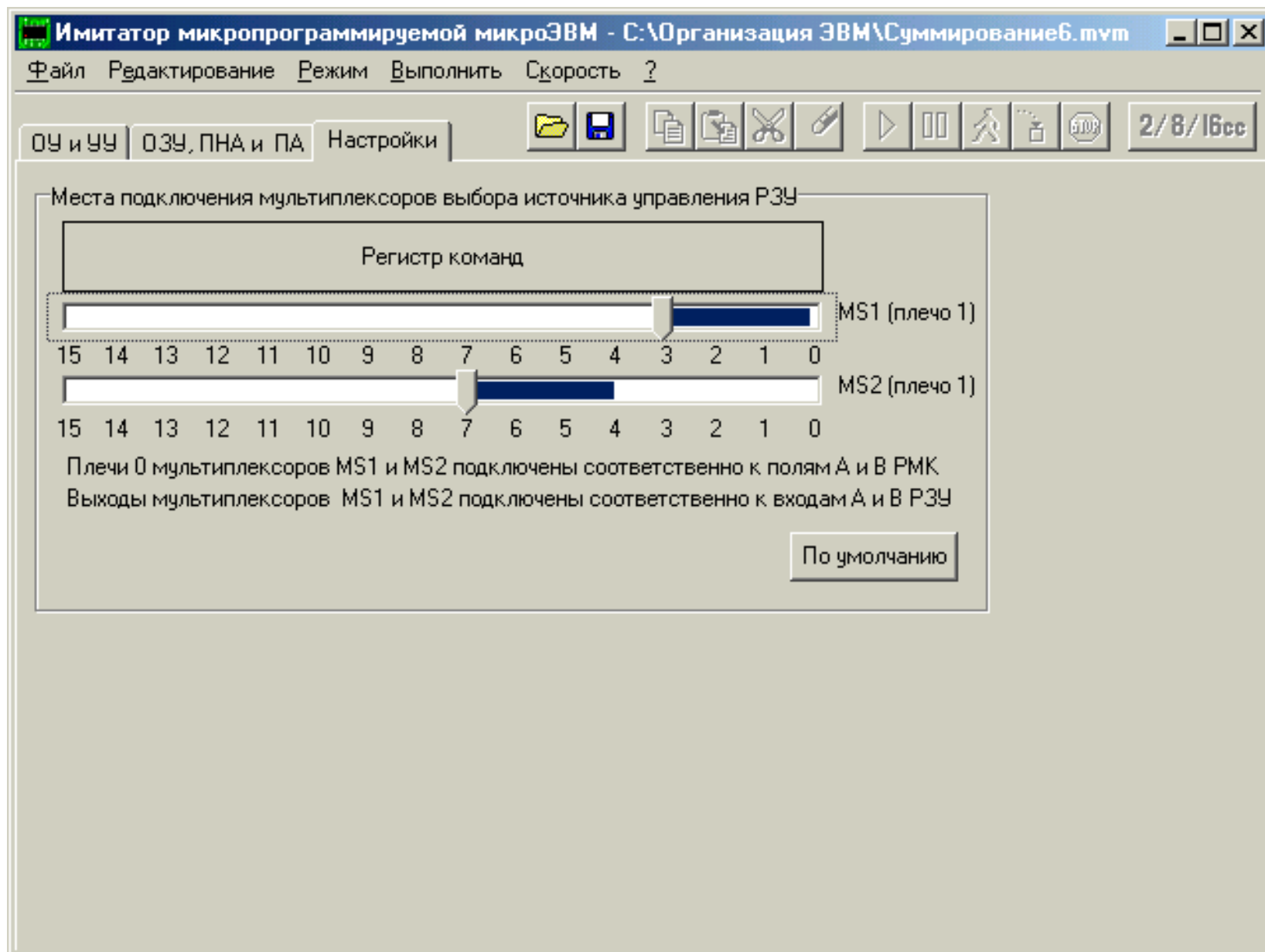
| | | | |
|-----|-------------------------|--|--|
| 8: | r8 (SP) | | |
| 9: | | | |
| 10: | | | |
| 11: | | | |
| 12: | | | |
| 13: | Буферный регистр команд | | |
| 14: | Регистр константы | | |
| 15: | Счетчик адреса ЗУ | | |

| | |
|-----|-----------|
| RQ: | Регистр Y |
|-----|-----------|

Принятая настройка схемы выбора адреса (СВА)

| MS1 (управляю щий вход СВА) | A (адресный вход РЗУ) | | MS2 (управляю щий вход СВА) | B (адресный вход РЗУ) |
|---|------------------------------------|--|---|------------------------------------|
| 0 | РМК[A] | | 0 | РМК[B] |
| 1 | РК[3:0] | | 1 | РК[7:4] |

Вкладка "Настройки"



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 |

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 | | | | | | | | | | | | | | | | | | | | |