

Учебная машина УМ-3

КОП А1 А2 А3

- Память учебной машины будет состоять из 512 ячеек, имеющих адреса от 0 до 511. Каждая ячейка состоит из 32 двоичных разрядов.
- Машинное слово, трактуемое как команда, представляться в следующей форме:

КОП А1 А2 А3

КОП А1 А2 А3

- Первое поле с именем КОП содержит число от 0 до 31.
- Поля с именами А1, А2 и А3 задают адреса операндов (это целые числа от 0 до 511).
- А1- адрес результата.

Регистры

- RA – регистр, называемый *счётчиком адреса*, он имеет 9 разрядов и после выполнения текущей команды будет содержать адрес команды, которая должна будет выполняться следующей;
- RK – регистр команд имеет 32 разряда и содержит текущую выполняемую команду, которая, содержит код операции КОП и адреса операндов A1, A2 и A3;
- w – регистр длиной два бита, его название произносится как "омега" (регистр-признак результата). В этот регистр после выполнения некоторых команд (у нас это будут арифметические команды сложения, вычитания, умножения и деления) записывается число от 0 до 2 по следующему правилу (S – числовой результат арифметической операции, полученной на регистре-сумматоре АЛУ):

$$w := \begin{cases} 0, & S = 0 \\ 1, & S < 0 \\ 2, & S > 0 \end{cases}$$

- Err – регистр ошибки, содержащий нуль в случае успешного выполнения очередной команды и единицу в противном случае.

Язык машины

КОП	Смысл операции и её мнемоническое обозначение
01	СЛВ – сложение вещественных чисел
11	СЛЦ – сложение целых чисел
02	ВЧВ – вычитание вещественных чисел
12	ВЧЦ – вычитание целых чисел
03	УМВ – умножение вещественных чисел
13	УМЦ – умножение целых чисел
04	ДЕВ – деление вещественных чисел
14	ДЕЦ – деление целых чисел (то же, что и div в Паскале)
15	МОД – остаток от деления (то же, что и mod в Паскале)
00	ПЕР – пересылка: $\langle A1 \rangle := \langle A3 \rangle$
10	ЦЕЛ – вещественное в целое: $\langle A1 \rangle := \text{Round}(\langle A3 \rangle)$
30	ВЕЩ – целое в вещественное: $\langle A1 \rangle := \text{Real}(\langle A3 \rangle)$
09	БЕЗ – безусловный переход: goto A2, т.е. RA:=A2
19	УСЛ – условный переход: Case w of 0: goto A1; 1: goto A2; 2: goto A3 end

Язык машины

20	ПР – переход по $S = 0$: if $w=0$ then goto A2
21	ПНР – переход по $S \diamond 0$: if $w \langle \rangle 0$ then goto A2
22	ПБ – переход по $S > 0$: if $w=2$ then goto A2
23	ПМ – переход по $S < 0$: if $w=1$ then goto A2
24	ПБР – переход по $S \geq 0$: if w in $[0, 2]$ then goto A2
25	ПМР – переход по $S \leq 0$: if $w < 2$ then goto A2
31	СТОП – остановка выполнения программы
06	ВВВ – ввод A2 вещественных чисел в память, начиная с адреса A1 for $i:=1$ to A2 do Readln($\langle A1+i-1 \rangle$)
07	ВВВ – вывод A2 вещественных чисел, начиная с адреса A1 for $i:=1$ to A2 do Writeln($\langle A1+i-1 \rangle$)
16	ВВЦ – ввод массива целых чисел, аналогично ВВВ
17	ВВЦ – вывод массива целых чисел, аналогично ВВВ

Схема выполнения команд

- **repeat**
- 1. $RK := \langle RA \rangle$; чтение очередной команды из ячейки памяти с адресом RA на регистр команд RK в устройстве управления.
- 2. $RA := RA + 1$; увеличение счётчика адреса на единицу, чтобы следующая команда (если текущая команда не является командой перехода) выполнялась из следующей ячейки памяти.
- 3. Выполнение операции, заданной в коде операции (КОП) в АЛУ. При несуществующем коде операции или других ошибках при выполнении команды (например, делении на ноль) выполнение команды прекращается и производится присваивание $Err := 1$.
- **until** ($Err=1$) **or** (КОП=СТОП)

Таким образом, как и в машине Фон
Неймана, АЛУ, подчиняясь
управляющим сигналам со стороны УУ,
выполняет бинарную операцию по
схеме

$$R1 := \langle A2 \rangle; R2 := \langle A3 \rangle; S := R1 \square R2;$$
$$\langle A1 \rangle := S;$$

Формирование регистра w для
арифметических операций

- В УМ-3 первичной загрузкой программы в память и формированием начальных значений регистров в устройстве управления занимается *устройство ввода*. Для этого на устройстве ввода имеется специальная кнопка ПУСК. При нажатии этой кнопки **устройство ввода** самостоятельно (без сигналов со стороны устройства управления, которое ещё не функционирует) выполняет следующую последовательность действий:
 1. Производит ввод расположенного на устройстве ввода массива машинных слов в память, начиная с *первой ячейки*; этот массив машинных слов заканчивается специальным признаком *конца ввода*.
 2. $RA := 1$; первой будет выполняться команда из ячейки a номером один.
 3. $w := 0$; начальное значение признака результата нулевое.
 4. $Err := 0$; признак ошибки сбрасывается (устанавливается равным **false**).



- «Стрелá» — [советская ЭВМ первого поколения](#).
- Разработана в СКБ-245, г. [Москва](#)Разработана в СКБ-245, г. Москва (с 1958 года это НИИ электронных математических машин — НИЭМ, с 1968 года — НИЦЭВТ [\[1\]](#)). Главный конструктор — [Юрий Яковлевич Базилевский](#). В числе помощников был [Башир Рамеев](#). В числе помощников был Башир Рамеев, впоследствии ставший главным конструктором ЭВМ серии [Урал](#). В числе помощников был Башир Рамеев, впоследствии ставший главным конструктором ЭВМ серии Урал. Разработка завершена в [1953 году](#).
- Выпускалась серийно на [Московском заводе счётно-аналитических машин \(САМ\)](#), с [1953](#) по [1956](#) гг. Всего было выпущено семь машин, которые были установлены в вычислительном центре Академии наук СССР, в [МГУ](#), в вычислительных центрах нескольких министерств ([ВЦ-1 МО СССР](#) и др.).
- Быстродействие машины — 2000 оп/с. Элементная база — 6200 электровакуумных ламп, 60 000 полупроводниковых диодов. [Оперативная память на электронно-лучевых трубках](#), 2048 слов. Длина слова — 43 двоичных разряда (из них — 35 бит на мантиссу и 6 на экспоненту). Постоянное ЗУ на полупроводниковых диодах. Внешнее ЗУ — два накопителя на магнитной ленте. Ввод данных — с перфокарт и с магнитной ленты. Вывод данных — на магнитную ленту, на перфокарты и на широкоформатный принтер.
- Последний вариант «Стрелы» использовал память на магнитном барабане (4096 слов), вращающемся со скоростью 6000 об/мин.

Пример 1. Оператор присваивания

Составим программу, которая вводит целое число x и реализует арифметический оператор присваивания языка Паскаль:

$$y := (x+1)^2 \bmod (2*x)$$

На языке Паскаль эта программа может выглядеть, например, так:

- **Program** First(input,output);
- **Var** x,y: integer;
- **Begin** Read(x); $y:=\text{sqr}(x+1) \bmod (2*x)$;
Writeln(y) **end.**

Текст программы с комментариями

№	Команда				Комментарий
001	16	101	001	000	Read(x)
2	11	103	101	008	r1 := (x+1)
3	13	103	103	103	r1 := (x+1) ²
4	11	104	101	101	r2 := (x+x)=2*x
5	15	102	103	104	y := r1 mod r2
6	17	102	001	000	Write(y)
7	31	000	000	000	Стоп
8	00	000	000	001	Целая константа 1

X -> 101
x+1 -> 103
(x+1)² -> 103
(x+x) -> 104
Y->102
1 -> в 8-ой ячейке

Пример 2. Условный оператор

$$y := \begin{cases} x+2, & \text{при } x < 2, \\ 2, & \text{при } x = 2, \\ 2^*(x+2), & \text{при } x > 2; \end{cases}$$

Текст программы:

$$y := \begin{cases} x+2, & \text{при } x < 2, \\ 2, & \text{при } x = 2, \\ 2*(x+2), & \text{при } x > 2; \end{cases}$$

№	Команда				Комментарий
001	ВВЦ	100	001	000	Read(x)
2	СЛЦ	101	100	<u>011</u>	y := x+2
3	ВЧЦ	000	100	<u>011</u>	<000> := x-2; формирование w
4	УСЛ	005	<u>007</u>	<u>009</u>	Case w of 0: goto 005; 1: goto 007; 2: goto 009 end
5	ВЫЦ	<u>011</u>	001	000	Write(2)
6	СТОП	000	000	000	Конец работы
7	ВЫЦ	101	001	000	Write(y)
8	СТОП	000	000	000	Конец работы
9	УМЦ	101	<u>011</u>	101	y := 2*y
010	БЕЗ	000	007	000	Goto 007
1	00	000	000	002	Целая константа 2

X -> 100
x+2 -> 101
X-2 -> 0

Пример 3. Реализация цикла

$$y = \sum_{i=1}^n 1/i$$

Программа для решения этой задачи

$$y = \sum_{i=1}^n 1/i$$

№	Команда				Комментарий
001	ВВЦ	100	001	000	Read(n)
2	ВЧВ	101	101	101	y := 0.0
3	ПЕР	102	000	<u>013</u>	i := 1
4	ВЧЦ	000	102	100	<000> := i-n; формирование w
5	ПВ	000	<u>011</u>	000	if i>n then goto 011
6	ВЕЩ	000	000	102	<000> := Real(i)
7	ДЕВ	000	014	000	<000> := 1.0/Real(i)
8	СЛВ	101	101	000	y := y+<000>
9	СЛЦ	102	102	<u>013</u>	i := i+1
010	БЕЗ	000	004	000	Следующая итерация цикла
1	ВЫВ	101	001	000	Write(y)
2	СТОП	000	000	000	Стоп
3	00	000	000	001	Целая константа 1
4		<1.0>			Вещественная константа 1.0

n -> 100
y -> 101
i -> 102

Пример 4. Работа с массивами

$$S = \sum_{i=1}^{100} x[i]$$

Решение задачи

$$S = \sum_{i=1}^{100} x[i]$$

№	Команда				Комментарий
001	ВВВ	200	100	000	Read(x); массив x в ячейках 200÷299
2	СЛВ	<u>008</u>	200	<u>008</u>	S := S+x[1]
3	СЛЦ	002	002	<u>011</u>	Модификация команды в ячейке 2
4	ВЧЦ	<u>010</u>	<u>010</u>	<u>009</u>	n := n-1
5	ПВ	000	002	000	if n>0 then goto 002
6	ВЫВ	<u>008</u>	001	000	Write(S)
7	СТОП	000	000	000	Стоп
8		<0.0>			Вещественная переменная S = 0.0
9	00	000	000	001	Целая константа 1
010	00	000	000	100	Переменная n с начальным значением 100
1	00	000	001	000	Константа переадресации

Рассматриваемая программа выделяется новым приёмом программирования, она является *самомодифицирующейся программой*

- В нашей учебной машине
самомодифицирующаяся программа –
это *единственный* способ обработки
достаточно больших массивов.

Формальное описание учебной машины

- Что, например, будет происходить после выполнения команды из ячейки с адресом 511?
- Какое значение после нажатия кнопки ПУСК имеют ячейки, расположенные вне введённого массива машинных слов?
- Как представляются целые и вещественные числа? Как будет, например, выполняться такая машинная команда ввода массива: ВВВ 500 100 000 ?

Вопросы и упражнения

1. Что такое код операции?
2. Для чего необходим регистр признака результата?
3. Объясните, как в нашей учебной машине должна выполняться такая команда ввода массива вещественных чисел `BVV 100 500 000`. Напишите соответствующую ветвь в формальном описании УМ-3 для реализации команды ввода вещественных чисел.
4. Реализуйте в модели на Паскале выполнение команд ввода/вывода учебной машины.
5. Почему при программировании на языке машины не существуют константы, как, например, в языке Паскаль?
6. Что такое переменная с начальным значением и как такую переменную разместить в памяти учебной ЭВМ?
7. Что такое псевдокод?
8. Объясните, почему для машины УМ-3 при решении некоторой задачи нельзя сделать такое распределение памяти: "Пусть массив X располагается в ячейках с адресами от 100 до 199, а константа $n=100$ – в ячейке с адресом 200" ?
9. Что такое самомодифицирующаяся программа?
10. Почему в учебной машине УМ-3 обработка больших массивов возможна только при помощи самомодифицирующейся программы?