

Раздел 1.
Введение в архитектуру
ЭВМ.

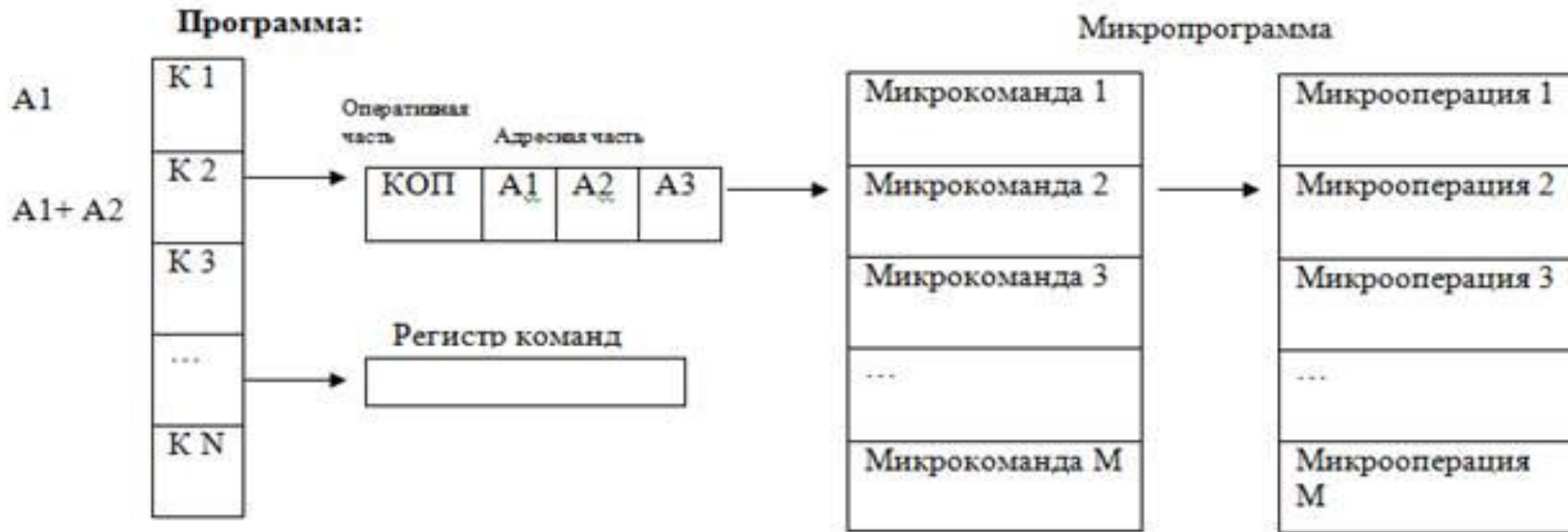
Форматы команд и способы
адресации ЭВМ .

Структура команды в общем случае имеет вид:

Команда – это инструкция машине на выполнение элементарной операции. Набор операций, которые может выполнять компьютер, и правил их записи образуют *машинный язык*.

КОП Код операции	A1 Адрес 1-го операнда	A2 Адрес 2-го операнда	A3 Адрес помещения результата	A4 Адрес следующей команды
-------------------------------	-------------------------------------	-------------------------------------	---	--

Общая структура программы



К – Команда - инструкция машине на выполнение элементарной операции.

КОП – Код операции – код, находящийся в оперативной части; определяет какая именно операция выполняется. Занимает 8 бит

Микрооперация – элементарное действие внутри ВМ.

А – Адресная часть – часть, где хранятся адреса.

Команды бывают:

- одноадресные,
- двухадресные,
- трехадресные,
- безадресные.

Формат команды определяет ее структуру:

1. количество двоичных разрядов, отводимых под команду;
2. количество и расположение отдельных полей команды.

Регистр команд – регистр, после помещения в который, тело команды начнет выполняться.

Общая длина команды описывается
следующим соотношением:

$$L_k = \sum_{i=1}^n R_i + R_{\text{коп}} + R_{\text{са}}, \text{ где:}$$

n – количество адресов в команде;

R_i – количество разрядов для записи i -го операнда;

$R_{\text{коп}}$ – разрядность поля для КОП;

$R_{\text{са}}$ – разрядность способа адресации.

$R_{\text{коп}} = \text{int}(\log_2 N_{\text{коп}})$, где:

$N_{\text{коп}}$ – количество команд в системе команд;

$\text{int}()$ – округление в большую сторону до целого.

$R_i = \text{int}(\log_2 N_i)$, где:

N_i – количество ячеек ОП, к которому можно обратиться с помощью i -го адреса;

$\text{int}()$ – округление в большую сторону до целого.

$R_{\text{СА}} = \text{int}(\log_2 N_{\text{СА}})$, где:

$N_{\text{СА}}$ – количество способов адресации;

$\text{int}()$ – округление в большую сторону до целого

Форматы команд

КОП	<i>нульадресная команда</i>			
КОП	операнд	<i>одноадресная команда</i>		
КОП	регистр	операнд	<i>полуторадресная команда</i>	
КОП	регистр	регистр	операнд	<i>полуторадресная команда</i>
КОП	операнд	операнд	<i>двухадресная команда</i>	
КОП	операнд	операнд	результат	<i>трехадресная команда</i>
КОП	операнд	операнд	результат	адрес следующей команды

Обобщенный формат команд ЭВМ архитектуры x86:

0-4 байта	1-2 байта	0-1 байт	0-1 байт	0-4 байта	0-4 байта
<i>Подполе префиксов: 1) префикс команды (повторения или блокировки доступа к шине данных); 2) префикс переопределения сегмента; 3) префикс переопределения размера операнда; 4) префикс переполнения размера адреса.</i>	<i>Подполе КОП</i>	<i>Подполе режимов адресации</i>	<i>Подполе индексного и базового регистров и масштаба индексирования (SIB)</i>	<i>Подполе смещения (DISP)</i>	<i>Подполе непосредственно операндов (DATA)</i>

Классификация команд.

Система команд – это совокупность различных типов команд вместе с правилами их кодирования.

Команды ЭВМ можно классифицировать :

1. состав и сложность команд;
2. место хранения операндов;
3. способ обработки данных и тип операндов.

Классификация по составу и сложности команд.

Существуют следующие системы команд:

1. полный набор команд (архитектура CISC);
2. сокращенный набор команд (архитектура RISC);
3. набор команд сверхбольшой длины (VLIW).

CISC-процессоры характеризуются:

1. небольшое число регистров общего назначения;
2. большое количество машинных команд;
3. большое количество методов адресации;
4. большое количество форматов команд различной разрядности;
5. наличие команд обработки типа «регистр-память», «память-память».

RISC-архитектура характеризуется:

1. отделение команд обработки от команд работы с памятью;
2. реализация конвейерной обработки данных;
3. реализация аппаратного и микропрограммного принципов управления;
4. наличие большого числа регистров;
5. использование преимущественно 3-х адресных команд;
6. простота архитектуры.

VLIW-архитектура

характеризуется:

1. на выполнение идет фиксированное количество команд, которые сформатированы либо как одна большая команда, либо как пакет команд фиксированной длины;
2. использование **трассировочного планирования** для выдачи команд (из последовательности исходной программы генерируются длинные команды путем просмотра программы за пределами базовых блоков – линейных участков программы без ветвлений).

Путь, имеющий наибольшую вероятность выполнения называется **трассой**.