

# Архитектура VM

**Архитектура** – это функциональная организация VM без физической реализации.

1. Представление данных и их связь с назначением, организация хранения и способы кодирования.
2. **Адресация** – способ определения адреса операнда по информации в адресной части команды.
3. Структура команд: части команд и взаимодействие частей.
4. Система команд – перечень команд.
5. Организация вычислительного процесса.
6. Организация ввода-вывода.
7. Система прерываний.

# Архитектура УМ

**Представление данных и их связь с назначением, организация хранения и способы кодирования.**

**Слово в УМ равно 3 байтам.**

**УМ – двоичная машина, так как для кодирования информации в ней принят двоичный алфавит.**

**Данные в УМ представлены в виде целых чисел в дополнительном коде в формате слово.**

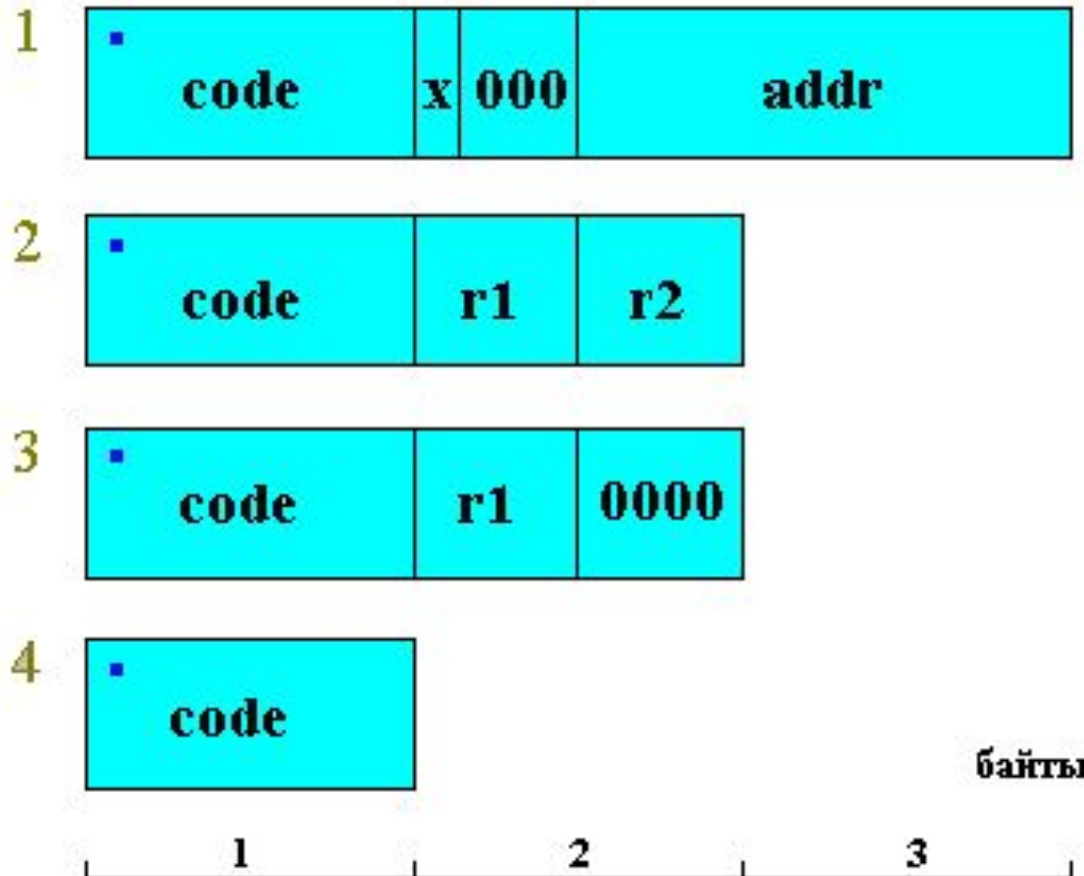
**Для хранения данных предусмотрено:**

- 3 регистра: - А – аккумулятор;**
- X – индексный регистр;**
- L – регистр связи;**

**ЗУ (запоминающее устройство) объемом до 4096 байт.**

# Архитектура УМ

**Структура команд: части команд и взаимодействие частей.**

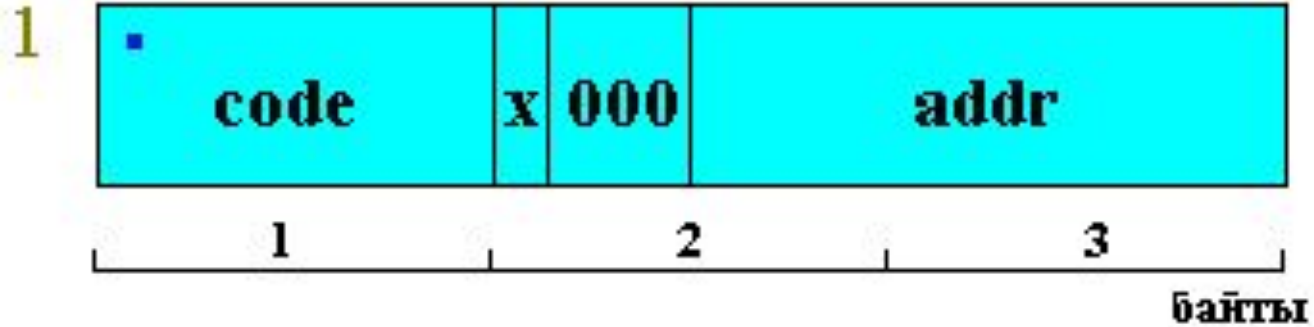


**code** – код операции;

**r1, r2** – номера регистров  
над которыми  
производится  
операция: **A – 0; X – 1,**  
**L – 2.**

# Архитектура УМ

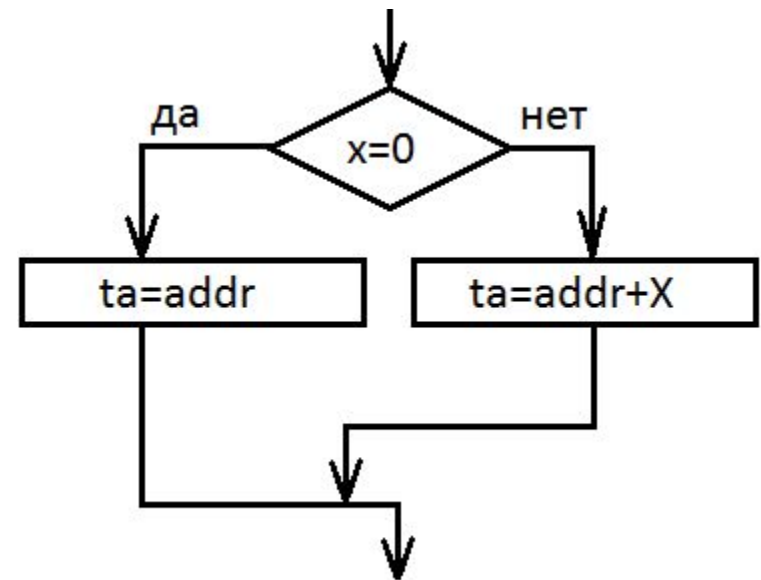
## Адресация



Виды адресации в УМ:

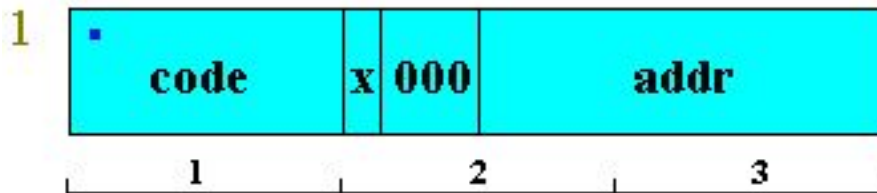
-прямая:  $ta = addr$ ;

-индексная:  $ta = addr + X$ .



# Архитектура УМ

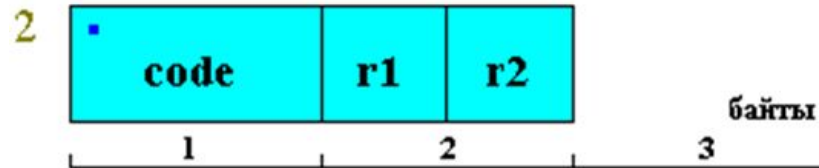
## Система команд – перечень команд



№	Название	Мнемоника	Код	Алгоритм	Изм. СС
1	Сложение	add	18	$A=A+W[ta]$	да
2	Вычитание	sub	1C	$A=A-W[ta]$	да
3	Сравнение	comp	28	$A-W[ta]==0$	да
4	Умножение	mul	20	$A=A*W[ta]$	да
5	Деление	div	24	$A=A \text{ div } W[ta]$	да
6	Загрузка аккумулятора	lda	00	$A=W[ta]$	нет
7	Сохранение аккумулятора	sta	0C	$W[ta]=A$	нет
8	Безусловный переход	j	3C	$PC=ta$	нет
9	Переход, если "меньше"	jlt	38	if $CC==1$ then $PC=ta$	нет
10	Переход, если "равно"	jeq	30	if $CC==0$ then $PC=ta$	нет
11	Переход, если "больше"	jgt	34	if $CC==2$ then $PC=ta$	нет
14	Загрузка регистра X	ldx	04	$X = W[ta]$	нет
15	Сохранение регистра X	stx	10	$W[ta] = X$	нет
18	Переход к подпрограмме	jsub	48	$L = PC; PC = ta$	нет
20	Загрузка регистра L	ldl	08	$L = W[ta]$	нет
21	Сохранение регистра L	stl	14	$W[ta] = L$	нет

# Архитектура УМ

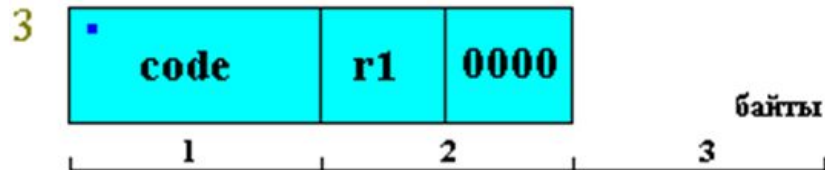
## Система команд – перечень команд



№	Название	Мнемоника	Код	Алгоритм	Изм. СС
16	Пересылка регистровая	rto	АС	$R[r2] = R[r1]$	нет

# Архитектура УМ

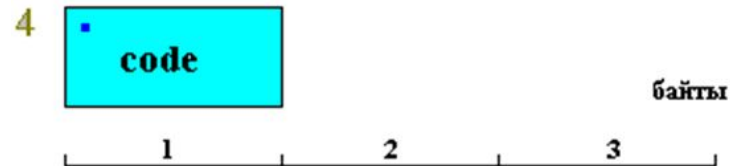
## Система команд – перечень команд



№	Название	Мнемоника	Код	Алгоритм	Изм. СС
17	Очистка регистра	clear	B4	$R[r1] = 0$	нет

# Архитектура УМ

## Система команд – перечень команд



№	Название	Мнемоника	Код	Алгоритм	Изм. СС
12	Останов	hlt	FF	Останов	нет
13	Нет операции	nop	FE	Нет операции	нет
19	Возврат из подпрограммы	rsub	08	PC=L	нет



# Архитектура УМ

## Организация вычислительного процесса

Программа в УМ состоит из последовательности команд, размещаемых в ЗУ.

Порядок выборки команд из ЗУ в процессор для исполнения устанавливается с помощью программного счетчика (**РС**).

Адрес первой исполняемой команды программы (*пусковой адрес*) устанавливается в **РС** с пульта управления перед пуском машины.

*Команды управления*-специальные команды, позволяющие изменить ход вычислительного процесса.

Изменение естественного порядка следования команд принято называть *переходом* в программе, а адрес команды, к которой выполняется переход, *адресом перехода*.

Арифметические команды формируют специальный двухбитовый *признак результата* (**СС**)

**СС = 0**, если  $A == 0$ ;

**СС = 1**, если  $A < 0$ ;

**СС = 2**, если  $A > 0$ ;

**СС = 3**, если зафиксировано переполнение.

# Архитектура УМ

## Организация ввода-вывода

В УМ не предусмотрены команды ввода/вывода. Предполагается, что ввод и вывод организуются средствами пульта управления.

Функции пульта управления УМ:

- ввод программ и данных в ЗУ;
- вывод данных из ЗУ;
- установка пускового адреса в РС. Пусковой адрес также называют точкой входа в программу;
- пуск машины для выполнения программы;
- останов машины;
- индикация состояния машины (STOP /RUN);
- установка пошагового режима выполнения программы с целью ее отладки.

## Система прерываний

Система прерываний в УМ не предусмотрена.

