

Типы данных (скалярные)

- Целочисленный
- Двоично-десятичный
- Бинарный вещественный
- Десятичные вещественный

Целочисленный тип данных

- Разрядность : 8, 16 , 32, 64 бита.
- Знаковые и без знаковые
 - Знаковые данные представляются в дополнительном коде.

Пример знаковых чисел:

Число	Разрядность	Код
-2	8	1111 1110
-2	16	1111 1111 1111 1110

Двоично-десятичный тип данных

- Разрядность : 8, 16 , 32, 64 бита.
- Каждая четверка бит представляет десятичную цифру.

Пример:

Число	Разрядность	Код
5	8	0000 0101
1234	16	0001 0010 0011 0100

Бинарный вещественный тип данных

- Формат



- Разрядность

Тип	Разрядность		
	Всего	Мантисса	Экспонента
Half float	16	10	5
Float	32	23	8
Double	64	52	11
Long double	80	63	15
Quad float	128	112	15

Десятичные вещественный тип данных

- Формула бинарного вещественного типа

$$(-1)^{sign} * mantissa * 2^{exponenta}$$

- Формула десятичного вещественного типа

$$(-1)^{sign} * mantissa * 10^{exponenta}$$

- Причина: Повышение точности бизнес вычислений.

Типы данных (векторные)

- Вектор - это набор однотипных элементов в едином блоке.



- «Размер вектора» = «размер элемента» \times
«количество элементов»

Типы данных (векторные)

- Размеры вектора: 64, 128 бит.
- Тип элемента вектора: целочисленный и вещественный
- Размер элемента вектора: 8, 16, 32, 64 бита

Где могут находиться операнды инструкции ?

- В коде инструкции
- В регистрах
- В памяти

Режимы адресации

- 1. Непосредственная (Immediate). Операнд в теле инструкции.
- 2. Регистровый (Register). В инструкции храниться адрес регистра, который содержит операнд.

Пример:

Тип адресации	Инструкция	Действие
Регистровая	<i>add r1, r2, r3</i>	$Reg[r1] = Reg[r2] + Reg[r3]$
Явная	<i>add r1, r2, #3</i>	$Reg[r1] = Reg[r2] + 3$

Режимы адресации операндов в памяти.

Названия режима адресации	Формула адреса	Пример инструкции	Выполняемое действие
Индексная (Displacement)	$\text{Reg}[\text{Base}] + \text{Displ.}$	add r1, r2, 100(r3)	$\text{Reg}[\text{r1}] = \text{Reg}[\text{r2}] + \text{Mem}[100 + \text{Reg}[\text{r3}]]$
Базовая или Косвенная регистровая (Register Indirect)	$\text{Reg}[\text{Base}]$	add r1, r2, (r3)	$\text{Reg}[\text{r1}] = \text{Reg}[\text{r2}] + \text{Mem}[\text{Reg}[\text{r3}]]$
Базово индексная (Indexed)	$\text{Reg}[\text{Base}] + \text{Reg}[\text{Index}]$	add r1, r2, (r3+r4)	$\text{Reg}[\text{r1}] = \text{Reg}[\text{r2}] + \text{Mem}[\text{Reg}[\text{r3}] + \text{Reg}[\text{r4}]]$

Режимы адресации операндов в памяти (продолжение).

Названия режима адресации	Формула адреса	Пример инструкции	Выполняемое действие
Базово индексная со смещением	$\text{Reg}[\text{Base}] + \text{Reg}[\text{Index}] + \text{Disp.}$	add r1,r2, 100(r3)[r4]	$\text{Reg}[r1] = \text{Reg}[r2] + \text{Mem}[\text{Reg}[R3] + \text{Reg}[r4] + 100]$
Явная или абсолютная (direct or absolute)		add r1,r2, (100)	$\text{Reg}[r1] = \text{Reg}[r2] + \text{Mem}[100]$
Косвенная (Memory Indirect)	$\text{Mem}[\text{Mem}[\text{Reg}[r]]]$	add r1,r2,@(r3)	$\text{Reg}[r1] = \text{Reg}[r2] + \text{Mem}[\text{Mem}[\text{Reg}[r3]]]$

Режимы адресации операндов в памяти (продолжение).

- Адресация с масштабированием .
 - Характеризуется наличием масштабирующего коэффициента. Может сочетаться со следующими типами адресации: индексной, базово-индексной, базово-индексной со смещением, абсолютной.
 - Коэффициент равен размерам базовых типов в байтах {2,4,8}. Храниться в коде инструкции.

Пример:

Тип адресации	Инструкция	Действие
Масштабируемая Базово-индексная	$add\ r1,r2,[r3+4*r4]$	$Reg[r1] = Reg[r2] + Mem[Reg[r3]+4*Reg[r4]]$

Режимы адресации операндов в памяти (продолжение).

- Адресация с автотомодификацией
 - Изменяет значение базового регистра на заданное число.
 - Число задается непосредственно или хранится в регистре.
 - Бывает постфиксная и префиксная.
 - Применяется совместно с базовой, индексной или базово-индексной адресациями.

Пример:

Тип адресации	Инструкция	Действие
Базовая с постмодификацией	$add\ r1, r2, (r3)+$	$Reg[r1] = Reg[r2] + Mem[Reg[r3]]$ $Reg[r3] = Reg[r3] + Disp.$

Режимы адресации операндов в памяти (окончание).

- Относительный режим.
 - Индексный режим, где в качестве базового регистра используется счетчик команд.

Пример:

Тип адресации	Адрес
Относительный индексный	$Ea = PC + Disp.$
Относительный базово - индексный	$Ea = PC + Reg[r]$

Выравнивание адреса операнда в памяти

- Адрес операнда в памяти выровнен, если выполняется следующее условие:

$a \bmod s = 0$, где a – адрес операнда, s размер операнда.

Пример:

Тип	Размер	Адрес
int	4 байта	...XX00
double	8 байт	...X000

Порядок байт в слове

- От младшего к старшему (*little-endian*)
- От старшего к младшему (*big-endian*)

Пример расположения int-а в памяти.

Байт 0 младший, байт 3 старший.

	Big-endian	Little-endian
Адреса+3	Байт 0	Байт 3
Адреса+2	Байт 1	Байт 2
Адреса+1	Байт 2	Байт 1
Адреса+0	Байт 3	Байт 0

Преимущества *little-endian*

В памяти расположено int число 0x00000003 по адресу A.

Адрес	Значение памяти	16 битное число из A	8 битное число из A
A+3	00		
A+2	00		
A+1	00	00	
A	03	03	03

Задачи решаемые при проектировании набора инструкций

- Функциональная полнота.
- Простота декодирования
- Уменьшение объема программного кода
- Достижение высокого быстродействия

Плотность кода

- Определяет количество элементарных операций размещаемых в единице объема кода.

Тип кодирования инструкций

- Постоянной длины
- Переменной длины
- Комбинированный

Типы инструкций

- Инструкции пересылки
- Арифметико-логические и сдвиговые инструкции.
- Инструкции управления

Типы инструкций пересылки

- Инструкции загрузки и сохранения данных из/в память.
- Инструкции загрузки констант в регистры.
- Безусловные инструкции пересылки между регистрами процессора
- Условные инструкции пересылки

Характеристики инструкций загрузки и сохранения

- 1. Разрядность данных: 8,16,32,64,128 бит.
- 2. Тип загружаемых данных: целочисленный беззнаковый, целочисленный знаковый, вещественный, векторный.
- 3. Тип используемой адресации.
- 4. Порядок байт в слове.

Отличие знаковой и беззнаковой загрузки целочисленных данных

- Беззнаковая инструкция загрузки дополняет старшие биты числа нулями.
- Знаковая инструкция загрузки дополняет старшие биты числа знаковым битом.

Пример:

Тип загрузки	Число в памяти (8 бит)	Число в регистре (32 бита)
Беззнаковая	1111 1110 (0xFE)	0x00 00 00 FE
Знаковая	1111 1110 (0xFE)	0xFF FF FF FE

Дополнительные характеристики инструкций пересылки.

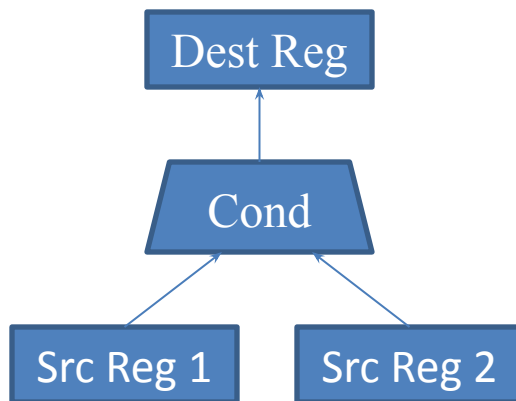
- Характеристики инструкций загрузки констант
 - 1. Разрядность константы.
 - 2. Способ расширения константы до требуемого размера.
- Характеристики инструкций безусловной пересылки между регистрами процессора.
 - Типы регистров между которыми возможна прямая пересылка данных.

Характеристики инструкций условной пересылки

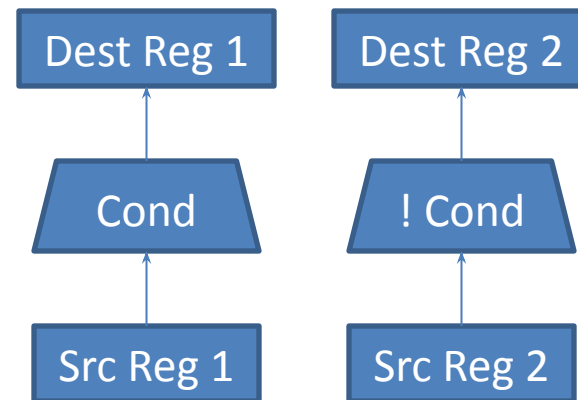
- Тип: трехадресный, четырехадресный.
- Способ задания условия.

Пример:

`move dest reg, src reg1, src reg2`



Трехадресная



четыреадресная

Использование инструкций условной пересылки

Код на C

Код на
ассемблере

Код на ассемблере с
инструкцией
условной пересылки

If (condition)

 a = b;

Else

 a=c;

cmp condition

je m1

mov a,c

jmp m2

m1: mov a,b

m2:

cmp condition

move a,b,c