Форматы

1. Форматы данных
2. Система команд. Форматы команд
3. Режимы адресации памяти. Стек
4. Прерывания

1. Форматы данных

- Машинная команда может работать с данными вполне определенного формата.
- При выполнении команды микропроцессор получает из оперативной памяти последовательность 0 и 1 и интерпретирует эту последовательность как символ, как адрес, как число в зависимости от команды.

- **Формат данных** — это способ интерпретации микропроцессором данных в процессе их обработки.
- МП может работать, например, с данными следующих форматов:
- 8-битовые целые числа без знаков. 1 байт воспринимается как целое неотрицательное число. Диапазон от 0000 0000 до 1111 1111в двоичной системе, 0 -256 в десятичной.

- 8-битовые целые числа со знаком.
- В байте для записи цифр числа используют 7 младших разрядов.
- Старший разряд байта используется как знаковый: 0 в этом разряде соответствует положительному числу, 1 — отрицательному.
- Отрицательные числа здесь представляются в дополнительном коде. Диапазон чисел -128 до +127.

- 16-битовые целые числа без знака (аналогично)
- 16-битовые числа со знаком
- 8-битовые двоично-десятичные числа. Используют для обработки десятичных чисел без знака. Для кодирования десятичных цифр здесь применяют 4-разрядные двоичные коды, например, 1 -0001, 2-0010. Используют упакованный и не упакованный форматы. В упакованном формате используются оба полубайта, диапазон от 0 до 99. В неупакованном формате используют только младший полубайт, диапазон от 0 до 9.

- 8-битовые символы. Для кодирования букв латинского и национального алфавитов, знаков препинания, цифр, математических символов, псевдографики, управляющих символов. Код ASCII.
- Битовые поля. Непрерывная последовательность битов, каждый разряд рассматривается независимо от остальных, например, содержимое регистра слова состояния процессора.

- Строки переменной длины. Последовательность байтов или слов. Строки могу иметь любую длину в пределах до :; Кбайт. Команды, манипулирующие с этим форматом, ведут обработку побайтно или по словам.
- адреса, указатели, векторы.

2. Система команд

Команды, которое может выполнять МП, составляют его систему команд

Система команд любой ЭВМ содержит следующие группы команд:

1. Передача данных, копирование данных из одного места в другое.
2. Арифметические операции. Основные — сложение и вычитание. Чаще всего вычитание сводится к сложению. Умножение и деление обычно выполняются по специальным программам.
3. Логические операции. Сравнение, И, ИЛИ, НЕ. Сюда добавляют анализ отдельных битов кода, их сброс и установку.
4. Сдвиги двоичного кода влево и вправо.
5. Команды ввода и вывода информации. В некоторых ЭВМ внешние устройства имеют специальные адреса, и эти операции сводятся к передаче данных.
6. Команды управления. Организуют нелинейные алгоритмы: условный и безусловный переход, обращение к подпрограмме. Останов, нет операции.

- Существую два направления в системах команд современных компьютеров.
- Компьютер с полным набором команд — CISC (Complex Instruction Set Computer),
- с ограниченным набором – RISC (Reduced Instruction Set Computer).
- Некоторые команды компьютер выполняет часто, другие — лишь эпизодически. RISC-машина имеет ограниченный набор простых и коротких операций, остальные действия реализованы программно.

Формат машинной команды

- **Формат машинной команды** — это сведения о длине, составе, назначении и взаимном расположении частей команды. Части команды называются полями. Например, каждая команда содержит поле кода операции, многие команды имеют поля операндов.
- **Описание формата команды —** это описание способа кодирования в команде всех сведений, необходимых МП для ее выполнения.

- Команда называется **безадресной**, если для ее выполнения не нужны операнды.
- Команда называется **адресной,** если ей требуются операнды, которые обычно расположены в основной памяти. Различают одноадресные, двухадресные и трехадресные команды

- Основные форматы:
- безадресная команда
- Все поле команды занимает код операции (КОП). Примеры, стоп, возврат из подпрограммы.

- Одноадресная команда
- Одно поле занимает КОП, второе поле хранит информацию о нахождении операнда.
- Пример, увеличить содержимое ячейки на 1.

КОП	Операнд

- Двухадресная команда
- Предусмотрены два поля для адресации двух операндов. Результат выполнения операции запоминается на месте одного из операндов.
- Пример, добавить содержимое одной ячейки к содержимому другой.

КОП	операнд 1	операнд 2

- Трехадресная команда
- Поле операнда 3 будет содержать результат операции.

КОП	Операнд 1	Операнд 2	Операнд 3

3. Режимы адресации памяти. Стек

- Режим адресации памяти — это схема преобразования адресной информации об операнде в команде в его исполнительный адрес.
- Операнд может хранится в регистре процессора или в оперативной памяти. Информация об этом и хранится в поле адресации: информация о методе адресации и номер регистра общего назначения, который участвует в команде.
-

•Пример, одноадресная команда:

КОП	Режим адресации	Номер регистра
	операнд	

- Рассмотрим режимы адресации, общие для многих МП.
- **Регистровый режим.** Операнд находится в регистре общего назначения. Обозначается Rn: R1, R2, цифры — номера регистров.
- **Регистровый косвенный режим.** Операнд находится в основной памяти, а его адрес — в регистре. Обозначается (Rn)

- **Автоинкрементный режим**. Подобен косвенному регистровому, но после адресации операнда и выполнения над ним операции содержимое регистра увеличивается на 2 (в командах выполнения операций над словами) и на 1 (в командах операции над байтами). Обозначение (Rn)+.

- **Автодекрементный режим**. В отличие от инкрементного режима здесь предварительно, до доступа к операнду содержимое регистра уменьшается на 2 или 1и, только после этого, регистр используется как адрес операнда. Обозначается -(Rn).

- **Операции с константой**. Данные могут непосредственно включаться в состав команды, т.е. размещаются после кода операции.
- **Сегментный способ.** Адрес ОЗУ вычисляется как сумма двух чисел: сегмента и смещения. Одно из них сдвинуто влево на 4 двоичных разряда (умножено на 16). Пусть сегмент A000h, смещение1000h. Запись адреса A000:1000. Итоговый адрес
- A000(0)
- + 1000
- A1000
- При этом один адрес может быть представлен несколькими комбинациями.

- **Стек.** Это структура данных, в которую можно данные добавлять и извлекать, но для работы доступен только последний элемент (первый вошел — последним вышел). Необходимый элемент — указатель стека, им может быть регистр общего назначения. Указатель хранит адрес верхнего занятого элемента

⇒	6			6			6			6	
	4		⇒	4	n1		4	n1		4	n1
	2			2			2	n2	⇒	2	n2
	0			0		⇒	0	n3		0	n3
а)			б)			в)			г)		

- Рассмотрим пример. Пусть требуется временно сохранить значения целочисленных переменных n1, n2, n3. Перед началом указатель стека содержит адрес 6 (рис.а). Даем команду: записать в стек n1.

- Процессор:
- уменьшает указатель стека на 2 (целое число занимает в памяти 2 байта), получит адрес 4;
- запишет n1 поэтому адресу (рис. б).

- Аналогично при выполнении команд "записать в стек n2" и "записать в стек n3" значения этих переменных попадут в ячейки 2 и 0 (рис.в).
- Теперь будем восстанавливать информацию из стека. Указатель содержит адрес 0, указывает на переменную n3. Подадим команду"прочитать из стека n3".

При этом процессор:

- считает значение n3 из стека;
- увеличит указатель стека на 2 (рис.г).
- Стек используется для вызова подпрограмм, для сохранения состояния прерванной программы.

4. Прерывания

- Микропроцессорные прерывания составляют важную часть архитектуры МП. Они обращают внимание МП на особые условия, возникающие в самом процессоре или вне его.
- Прерывание — это прекращение выполнения текущей последовательности команд для обработки события, которое произошло в микропроцессоре или ином оборудовании компьютера. Запрос на прерывание может возникнуть в самом МП или поступить от других устройств.

- Различают следующие классы прерываний:
- **Внутримикропроцессорные прер**ывания, вызываются событиями внутри МП (деление на нуль, переполнение регистров).
- **Внутрисистемные прерывания.** Возникают вследствие событий во внешних устройствах компьютера (от таймера, от принтера, когда он готов принять информацию).
- **Программные прерывания.** Намеренно заложены в программу.
- **Межмашинные прерывания.** Возникают при посылке коммуникационных сигналов от одного компьютера другому.

- Внутрисистемные прерывания бывают двух типов: маскируемые и немаскируемые.
- Программист или системная программа могут запретить или разрешить (замаскировать) маскируемые прерывания. Если в регистре флагов выставить 1 у бита признака разрешения прерываний, то прерывание обрабатываться не будет.
- Маскируемые прерывания возникают от устройств ввода-вывода.
- Немаскируемые прерывания наступают при катастрофических событиях (сбой по питанию), ошибка памяти, ошибка четности по магистрали. Эти прерывания имеют наивысший приоритет и процессор обрабатывает их в первую очередь.

- При обработке прерывания процессор
- запоминает состояние прерванной программы (в стэке);
- распознает источник прерывания (анализируется регистр флагов и состояние внешних устройств);
- маскирует повторные прерывания от этого источника;
- вызывает и выполняет специальную системную программу обработки прерываний (выводит сообщение о причине прерывания и рекомендацию по возможным действиям);
- восстанавливает состояние прерванной программы, продолжает ее исполнение.

Вопросы

- Что такое формат данных?
- Перечислите и опишите известные вам форматы данных
- Какие группы команд образуют систему команд любой ЭВМ?
- Опишите два направления в системах команд современных компьютеров

- Что такое формат машинной команды?
- Что означает адресная и безадресная команда?
- Опишите основные форматы команд
- Что такое режим адресации памяти?
- Где могут хранится операнды?
- Перечислите и опишите основные режимы адресации

- Опишите работу стека
- Что такое прерывание?
- Назовите классы прерываний
- Что означает маскируемое прерывание? Какие события к ним относятся?
- Какие события вызывают немаскируемые прерывания?
- Опишите последовательность действий МП при обработке прерывания.