МАШИНКИ

УМ-3

Простая и логичная архитектура

1 команда – 3 обращения к ОП

Избыточность кода

УМ-2

Быстродействие, как у УМ-3 (тоже 3 обращения к ОП)

Объём программы меньше

Удобство программирования

УМ-П

Быстродействие хуже, чем у УМ-2 (медленнее такт ЦП)

Объём программы < УМ-2 (нет избыточной информации)

Удобство программирования ≈ УМ-2 (удобно)

УМ-1

Быстродействие >> УМ-2 (1 обращение к ОП)

Объём программы < (команды короче)

Удобство программирования хуже (следить за содержимым S)

УМ-Р

ЦП сложнее, чем в УМ-2 (регистры + сложнее система команд)

Быстродействие >> УМ-2 (исключение пересылок ЦП <=> ОП)

Объём программы < УМ-2 (команды короче)

УМ-М

Надёжность

Быстродействие

Сложнее ЦП

Медленнее выполняются команды работы с ОП

УМ-С

Быстродействие: немного хуже УМ-2 (Арифм. операции - 3 обращения к ОП + работа с SP)

Объём программы меньше (короткие команды)

Удобство программирования хуже

Определения

Архитектура- то, как устроен компьютер с точки зрения программиста.

Машинная операция- элементарное действие, реализованное аппаратно.

Машинная команда- приказ ЦП выполнить одну машинную операцию с указанными операндами.

Формат команд- правила записи машинной команды.

Система команд процессора-набор машинных операций + формат команд.

Машинная программа- последовательность машинных команд.

Такт работы ЦП- выполнение одной машинной команды.

Быстродействие- среднее количество машинных операций, выполняемое в единицу времени.

Ячейка- минимальная адресуемая единица памяти.

Разрядность ячейки- кол-во разрядов в ячейке.

Шина- пучок проводов + электронная схема.

Вентиль- электронная схема, которая выполняет простейшую логическую операцию (or, not, and)

Флаг- ячейка в ЦП, состоит из 1 разряда.

Нормализованное число- старшая цифра мантиссы не равна нулю.

Консоль- текстовой окно для ввода и вывода текста, запуска программы из командной строки.

Стек- структура LIFO (last in, first out).

Мнемокод- мнемоническое(запоминающееся) название кода машинной операции.

Автокод(ассемблер)- язык, основанный на машинном языке, в котором используются символические обозначения.

Транслятор(ассемблер)- программа, которая переводит символическую запись программы в машинную программу.

Операционная система(ОС)- специальная программа, которая управляет работой компьютера и организует выполнение пользовательских программ.

Непосредственный операнд- число, явно указанное в машинной команде.

Символ- байт, число без знака.

Строка- последовательность символов-байтов.

Лексема- минимальная осмысленная единица текста.

Идентификатор- последовательность букв, цифр и специальных символов, начинающиеся не с цифр.

Директива- приказ ассемблеру выполнить какие-либо действия.

Примитив- операция с одним элементом строки.

Фрейм стека- область работы процедур с стеком.

Модуль- часть программы, которая может транслироваться отдельно, независимо от остальных частей программы.

Загрузчик- часть ОС. Он загружает модуль в ОП, выполняет необходимые настройки и запускает программу на счёт.

Прерывания- возможность ЦП переключаться с выполнения одной программы на выполнение другой.

Защита памяти- обеспечить безопасность адресного пространства одной программы от вмешательства других программ.

Таймер- помогает определить зацикливание программы и прервать её выполнение.

Мультипрограммный режим- такой режим работы ЭВМ, при котором в ОП одновременно находятся несколько готовых к счёту пользовательских программ, и процессор выполняет их, переключаясь между программами.

Статическое связывание- связи между программой и процедурой устанавливаются на этапе компоновки; программа и процедура загружаются в ОП одновременно.

Динамическое связывание- связи между программой и процедурой устанавливаются на этапе счёта; процедуры загружаются в ОП в момент обращения к ней.

Промах КЭШ- если данных нет в КЭШ, то сначала из ОП идёт запись в КЭШ, а затем из КЭШ в ЦП.

Адрес ячейки - порядковый номер ячейки в оперативной памяти.

Машинное слово - содержимое ячейки памяти (может быть командой или числом).

Код операции - целое число, он указывается в машинной команде и задает выполняемую машинную операцию.

Регистр - ячейка, которая находится не в оперативной памяти, а в другом устройстве (ЦП, устройстве ввода/вывода и т.д.)

Счётчик адреса – регистр в центральном процессоре, в нём хранится адрес следующей команды, т.е. команды, которая будет выполняться на следующем такте работы процессора (EIP).

Регистр команд – регистр центрального процессора, содержит текущую выполняемую команду.

Самомодифицирующаяся программа (самомодифицирующийся код) – программа, которая изменяет свой код в процессе работы.

Как в ЭВМ определяется, что в данный момент находится в ячейке памяти – данное (число) или команда?

По внешнему виду содержимого ячейки в памяти нельзя различить команды и данные.

Если содержимое ячейки поступает в регистр команд (РК), оно трактуется как команда, а когда содержимое ячейки поступает на регистры операндов (в АЛУ), оно трактуется как данное.

Как в ЭВМ определяется, какое число – со знаком или без знака – находится в ячейке?

По внешнему виду содержимого ячейки невозможно отличить число со знаком от числа без знака.

Как трактовать содержимое ячейки – как знаковое или беззнаковое число – определяется кодом операции команды.

.**Как в Ассемблере объявить имя внешним; зачем нужны такие имена?**

Внешние имена объявляются в директиве extrn.

Внешние имена позволяют осуществлять доступ к именам, описанным в других модулях программы.

Как в Ассемблере объявить имя общедоступным (общим); зачем нужны такие имена?

Общие имена переменных, меток, имён процедур и констант объявляются в директиве public. Общие имена доступны другим модулям программы. При трансляции модуля ассемблер сохраняет информацию о значениях общих имён в объектном модуле; компоновщик использует эту информацию при объединении объектных модулей в единую программу

В программе на Ассемблере есть предложения extrn X:abs и mov AX,X. Какая служебная программа подставляет конкретное значение на место операнда X в команде mov AX,X?

Текст программы на Ассемблере первоначально обрабатывает транслятор, он переводит предложения программы на машинный язык; вместо каждой ассемблерной команды транслятор строит машинную команду – выбирает нужный КОП, заполняет операнды. Поскольку имя X – внешнее, транслятор не знает его значение, поэтому оставляет соответствующее поле машинной команды незаполненным. После транслятора программу обрабатывает компоновщик (редактор внешних связей), который объединяет несколько модулей в единую программу, заполняя значения внешних имён.

Практически все программы работают со стеком, хотя бы для того, чтобы вызвать процедуры, реализующие ввод данных и печать результатов. Однако ни одна программа не загружает начальное значение в регистр ESP – указатель на вершину стека. Как же ESP получает значение? Загрузчик после размещения программы в ОП, и перед запуском её на счёт, отводит место в ОП для стека и записывает нужное значение в ESP.

Одна из функций компоновщика (редактора внешних связей) – редактирование (разрешение) межмодульных связей. Объяснить: а) в чем смысл этой функции; б) почему ее не может выполнить транслятор Ассемблера

а) Редактирование межмодульных связей – это замена внешних имён (имён из других модулей) их значениями (адресами для имён переменных, меток и числовыми значениями для имён констант). б) Поскольку Ассемблер транслирует каждый модуль программы по отдельности, то он не знает значения имён из других модулей (внешних имён) и потому не может сделать такую замену.

Такт работы процессора – выполнение одной машинной команды. Такт состоит из следующих этапов 1) Выборка команды из оперативной памяти по адресу, находящемуся в регистре счѐтчик адреса и запись команды в регистр команд, схематично RK:=Пам[RA] или RK:=. 2) Увеличение значения регистра счѐтчик адреса на единицу RA:=RA+1. 3) Выполнение команды, находящейся в регистре команд. 2. Выпишем, какие значения получат регистры в результате выполнения команд