

Машинно-зависимые ЯЗЫКИ программирования

Курс лекций

Процессор, МП и МПС

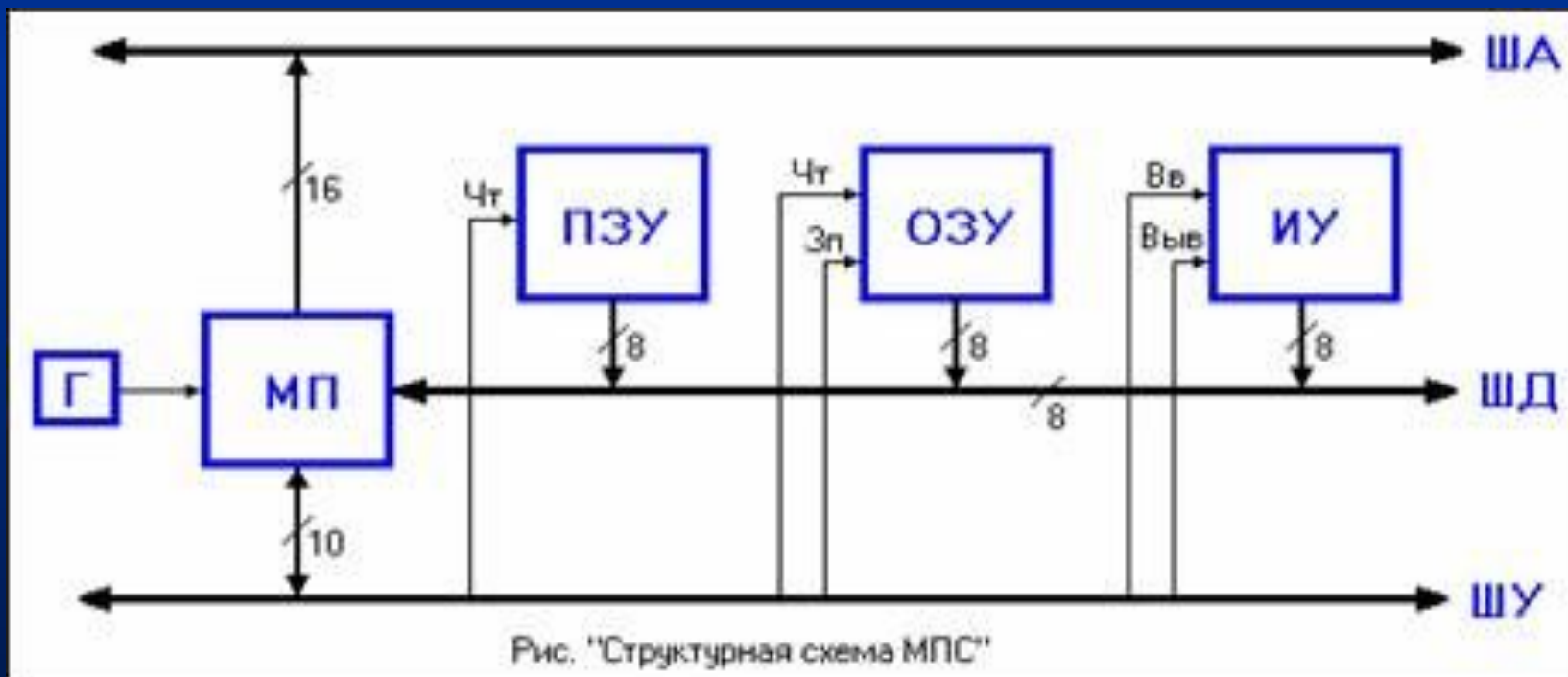
- Процессором называют программно-управляемое устройство, осуществляющее процесс обработки информации и управление
- Микропроцессором (МП) называют построенное на одной или нескольких БИС/СБИС программно-управляемое устройство, осуществляющее процесс обработки информации и управление им.
- МП система (МПС) – совокупность МП, памяти и устройства ввода/вывода (внешние устройства).

Структура МПС

Структура МПС является магистрально-модульной.

Типична 3-шинная структура МПС с шинами адресов ША (АВ -Address Bus), данных ШД(Data Bus) и управления ШУ(Control Bus).

Структура МПС с простым МП от Intel.



МП система

- Микропроцессор МП. Выполняя программу, МП обрабатывает команду за командой. Команда задает выполняемую операцию и содержит сведения об участвующих в ней операндах. После приема команды происходит ее расшифровка и выполнение, в ходе которого МП получает необходимые данные из памяти или внешних устройств. Ячейки памяти и внешние устройства (порты) имеют номера, называемые адресами, которыми они обозначаются в программе.

МП система

- Генератор Г задает МП тактовые импульсы. По каждому МП выполняет команду.
- Однонаправленная адресная шина ША. По ней МП посылает адреса, определяя объект, с которым будет обмен.
- Двунаправленная шина данных ШД. По ней МП обменивается данными с модулями (блоками) системы.
- Шина управления ШУ. По ней идет обмен управляющей информацией.

Архитектура МПС

МПС разделяют:

- По особенностям набора регистров, формата команд и данных:

CISC (Complex instruction set computing) — компьютер с комплексным набором команд;

RISC (Restricted (reduced) instruction set computer) — компьютер с сокращённым набором команд;

VLIW (very long instruction word — «очень длинная машинная команда») — архитектура процессоров с несколькими вычислительными устройствами.

Аппаратная платформа включает:

- АСК - Архитектура системы команд (ISA - instruction set architecture). — это приблизительно то же самое, что и модель программирования.
- Микропрограмма (firmware - микрокод). Это системное программное обеспечение, встроенное («зашитое») в аппаратное устройство.
- Микроархитектура — это способ, которым данная архитектура набора команд реализована в процессоре..

Аппаратная платформа включает:

- Центральный процессор ЦП (или центральное процессорное устройство — ЦП; central processing unit - CPU) - микросхема, исполнитель машинных инструкций (кода программ), главная часть аппаратного обеспечения компьютера.

Архитектура фон Неймана

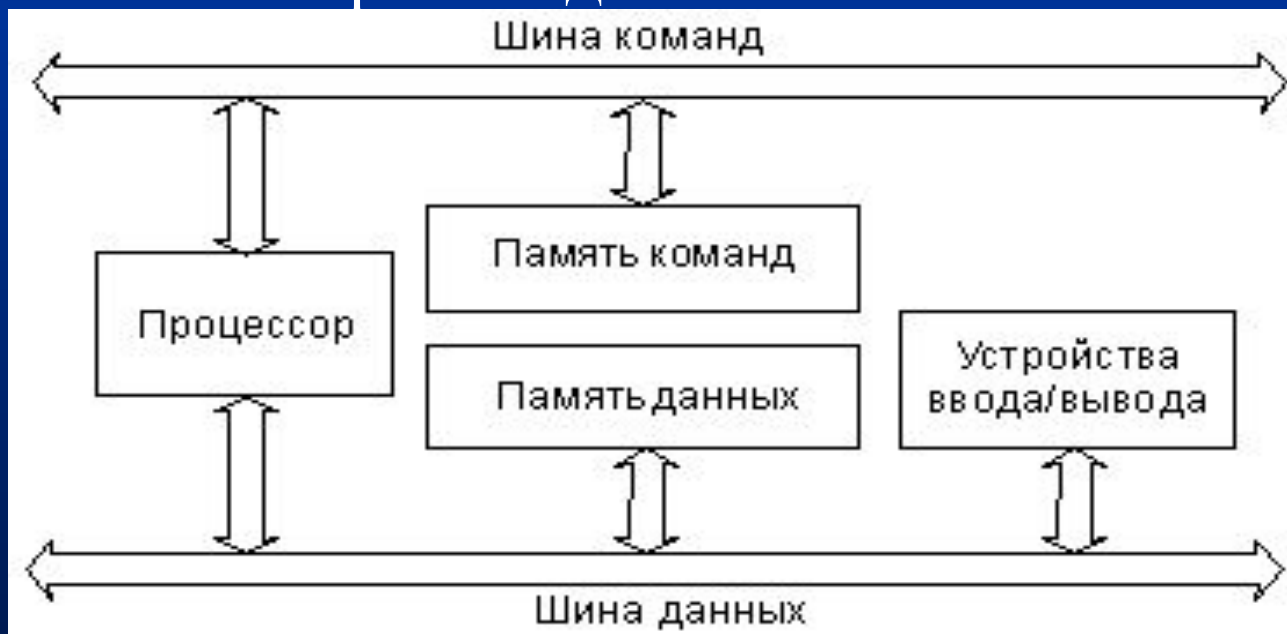


Особенности:

- память данных и команд находятся в процессоре
- память данных и команд общая

Гарвардская архитектура

- это архитектура с отдельными шинами данных и команд (двухшинная, или гарвардская). Эта архитектура предполагает наличие в системе отдельной памяти для данных и отдельной памяти для команд. Обмен процессора с каждым из двух типов памяти происходит по своей шине.



Параллельная архитектура

Параллельные процессоры используются в суперкомпьютерах. Возможными вариантами параллельной архитектуры могут служить (по классификации Флинна):

- SISD — один поток команд, один поток данных;
- SIMD — один поток команд, много потоков данных;
- MISD — много потоков команд, один поток данных;
- MIMD — много потоков команд, много потоков данных.

Конвейерная архитектура

Конвейер МП с архитектурой IA-32:

- Команда разбивается на этапы
- Однотипные этапы разных команд выполняются своим устройством
- Устройство последовательно во времени выполняет однотипные этапы разных команд

Суперскалярная архитектура

В МП с такой архитектурой применяется:

- Параллельное исполнение команд
- Решение о параллельном исполнении команд принимается аппаратурой процессора

Память

Память программ

- Хранит коды программы

Память данных

- Хранит данные
- Хранит численные данные

Кэш-память

- Быстродействующее ЗУ
- Доступ быстрее чем для внешней оперативной памяти

Интерфейс определяет правила взаимодействия компонент и модулей системы.

Типы интерфейсов:

- **Последовательный.** Биты данных передаются последовательно во времени по одному каналу.
- **Параллельный.** Данные передаются группами битов, для каждого бита свой канал.
- **Инфракрасный.** Данные передаются последовательно с использованием канала с инфракрасным лучом.
- **Bluetooth.** Данные передаются последовательно с использованием радиоканала.

Интерфейс

Типы интерфейсов:

- **USB.** Представляет собой шину, по которой к периферийному устройству подводится питание и осуществляется двунаправленный побитовый обмен данными.
- **Интерфейс JTAG.** Средство тестирования. Вывод на периферию внутрипроцессорных данных по выбору
- **Flash память.** Электрически перепрограммируемое запоминающее устройство. Запись и чтение только блоками

Программа Ассемблер

Ассемблер (assembler — сборщик) — компьютерная программа, компилятор исходного текста программы, написанной на языке ассемблера, в программу на машинном языке.

Ассемблеры, как правило, специфичны конкретной архитектуре, операционной системе и варианту синтаксиса языка.

Версии ассемблеров

- **Microsoft Macro Assembler (MASM)**
- **Flat assembler (FASM)**
- **NASM (Netwide Assembler)**
- **Turbo Assembler (TASM)**
- **GoAsm**

Ассемблеры

FASM - это свободно распространяемый ассемблер, способен создавать файлы в формате Unicode

MASM — ассемблер для процессоров семейства x86. Произведён компанией Microsoft для написания программ в операционной системе MS-DOS.

TASM — программный пакет компании Borland, предназначенный для разработки программ на языке ассемблера для архитектуры x86.

GoAsm — ассемблер для процессоров семейства x86, созданный для написания программ для операционных систем семейства Windows, способен создавать 32- и 64-битных версий, а также программы с поддержкой Unicode.

Среды разработки

✓ RadASM

✓ WinAsm Studio

✓ Fresh

✓ Easy Code

RadASM

RadASM — бесплатная интегрированная среда разработки программного обеспечения для ОС Windows и не только, изначально предназначенная для написания программ на языке ассемблера.

Имеет гибкую систему файлов настроек, благодаря чему может быть использована как среда разработки программного обеспечения на высокоуровневых языках, а также документов, основанных на языках разметки.

Создана программистом Ketil Olsen (KetilO).

Поддерживаемые ассемблеры:

MASM

FASM

NASM

TASM

GoAsm

WinAsm Studio

WinAsm Studio — бесплатная ИСР для Windows и DOS, изначально предназначенная для написания программ на языке ассемблера. Создана программистом Антонисом Киприану.

Возможности:

- Подсветка синтаксиса.
- Автодополнение кода.
- Менеджер проектов.
- Полная настройка ИСР.
- Есть окно вывода.
- Поддержка плагинов.
- Редактор ресурсов.
- Есть примеры использования.

Форматы представления двоичных чисел

С точки зрения длины представления чисел различают:

- Полубайт (Нибл). Содержит 4 бит. Отображает содержимое половинки байта.
- Байт. Содержит 8 бит. Отображает содержимое одной из 8-разрядных ячеек памяти или одного из 8-разрядных регистров. Это минимальный размер адресуемой в МП ячейки памяти.

- Слово. Содержит 2 байта, 16 бит. Отображает содержимое одной из 16-разрядных ячеек памяти или одного из 16-разрядных регистров.
- Двойное слово. Содержит 2 слова, 4 байта, 32 бит. Отображает содержимое 32-разрядных ячеек памяти или регистров, поэтому характеризует представление чисел с удвоенной точностью.
- Учетверенное слово. Содержит 2 двойных слова, 4 слова, 8 байт, 64 бит.

Различают две формы представления чисел:

с фиксированной точкой (ФТ), например, 12.34
– обычное представление вещественного
числа..

с плавающей точкой (ПТ), например, 1.234 E 2.

Это представление числа 12.34 в
показательной форме: $1.234 * 10^2$. 1.234 –
значащая часть (или мантисса), E –
разделитель, 2 – порядок.

Применяемые термины:

MSB (Most Significant Bit) – наиболее значащий бит.

LSB (Least Significant Bit) – наименее значащий бит.

В формате с фиксированной точкой в представлении данных в поле числа присутствует логическая позиция точки (бита точки нет, он логически подразумевается), задающая начало или конец значащей части.

Применяемые термины:

MSB (Most Significant Bit) – наиболее значащий бит.

LSB (Least Significant Bit) – наименее значащий бит.

В формате с фиксированной точкой в представлении данных в поле числа присутствует логическая позиция точки (бита точки нет, он логически подразумевается), задающая начало или конец значащей части.

- Число целое со знаком. Бит знака S размещается в MSB. Значащие биты выравниваются по правому краю формата. Логическая точка справа от LSB. Например, для 8-и разрядного процессора двоичное целое число 1101. Его десятичный эквивалент $8+4+1 = 13$.

Биты	7	6	5	4	3	2	1	0
Значения	0	0	0	0	1	1	0	1
	MSB=S							LSB

- Число целое без знака. Бит знака $S=0$ по умолчанию. Значащие биты начинаются с MSB (Most Significant Bit). При одинаковом N число битов значащей части в 2 раза больше. Значащие биты выравниваются по правому краю формата. Логическая точка справа от LSB. Например, для 8-и разрядного процессора двоичное целое число 101. Его десятичный эквивалент $4+1 = 5$.

Биты	7	6	5	4	3	2	1	0
Значения	0	0	0	0	1	1	0	1
	MSB							LSB

- Число дробное. Значащие биты выравниваются по левому краю формата. Логическая точка справа от бита знака S. Например, для 8-и разрядного процессора двоичное дробное число 0.101. Его десятичный эквивалент $0.5 + 0.125 = 0.625$.

Биты	7	6	5	4	3	2	1	0
Значения	0	1	0	1	0	0	0	0
	MSB=S							LSB

- Число дробное без знака. Значащие биты начинаются с MSB. При одинаковом N число битов значащей части в 2 раза больше. Значащие биты выравниваются по левому краю формата Логическая точка слева от бита знака S. Например, для 8-и разрядного процессора двоичное дробное число 0.0101. Его десятичный эквивалент $0.25+0.0625 = 0.3125$.

Биты	7	6	5	4	3	2	1	0
Значения	0	1	0	1	0	0	0	0
	MSB							LSB

Формат с плавающей точкой предназначен для компактного отображения вещественных чисел в очень широком диапазоне. Число представляется в алгебраическом формате:

$$(S)(F)*2^P,$$

где S (Sign) – знак числа. Для положительного $S = 0$, для отрицательного $S = 1$.

F (Fraction) - мантисса (значащая часть).

P (Power) – порядок.

Согласно стандарту **IEEE 754** слово данных разбивается на три поля:

Однобитовое поле S (sign - знак) используется для указания знака числа. Для положительного числа $S = 0$, для отрицательного $S = 1$.

Поле F (fraction). В нем записывается дробная часть мантиссы (fraction). Мантисса наряду с дробной частью содержит целую часть (1 или 0). Бит целой части мантиссы в памяти не хранится для уменьшения объема запоминаемых данных.

Поле экспоненты (E – exponent), содержит смещённый порядок $E = P + Bias$. Bias – смещение, выбирается так, чтобы смещённый порядок был положительным или равным нулю.

- Если целая часть мантиссы равна единице, то число считается нормализованным, а если она равна нулю, то ненормализованным. Целая часть мантиссы считается равной нулю, только в том случае, когда смещённый порядок числа также равен нулю. Во всех остальных случаях целая часть мантиссы равна единице.

В зависимости от точности представления форма с плавающей точкой имеет 3 стандарта:

- С одинарной точностью SP (Single Precision floating-point format).
- С двойной точностью DP (Double Precision floating-point format).
- С расширенной одинарной точностью формат SEP (Single Extended Precision floating-point format). Это формат для представления результатов промежуточных и конечных вычислений с расширенной одинарной точностью. Применяется для данных, которые не могут быть представлены в формах SP или DP.

Типы адресаций операндов

Адресацией называется обращение к операнду (число, участвующее в операции), указание на который содержится в команде. Операнды, в зависимости от места своего хранения, могут указываться разными способами, которым соответствуют разные типы адресации, или, коротко, разные адресации.

При описании различных адресаций операндов используют понятия адресного кода и исполнительного адреса. Адресный код АК – это информация об адресе операнда, содержащаяся в команде. Исполнительный адрес АИ – это номер физической ячейки памяти, к которой производится обращение.

Первая группа адресаций устанавливает АИ по значению АК. Сюда входят:

- Непосредственная адресация. Операнд указывается в команде константой. Эта адресация используется только для указания исходных данных.
- Прямая адресация. АИ совпадает с АК.
- Регистровая адресация. В команде указывается имя регистра процессора, в котором хранится операнд.
- Косвенная адресация. Используется в целях сокращения длины команды. В этом случае АК указывает имя регистра процессора, в котором находится АИ. Такой регистр называют регистром адреса..

Наборы регистров

Регистрами называются области

высокоскоростной памяти, расположенные внутри процессора в непосредственной близости от его исполнительного ядра. Доступ к ним осуществляется несравнимо быстрее, чем к ячейкам оперативной памяти.

Соответственно, машинные команды с операндами в регистрах выполняются максимально быстро, поэтому в программах на языке ассемблера регистры используются очень интенсивно.

Пользовательские регистры:

- Регистры общего назначения (РОН) - EAX/AX/AN/AL, EBX/BX/BN/BL, EDX/DX/DH/DL, ECX/CX/CH/CL, EBP/BP, ESI/SI, EDI/DI, ESP/SP предназначены для хранения данных и адресов, программист может их использовать (с определенными ограничениями) для реализации своих алгоритмов.
- Сегментные регистры - CS, DS, SS, ES, FS, GS используются для хранения адресов сегментов в памяти.
- Регистры сопроцессора - ST(0), ST(1), ST(2), ST(3), ST(4), ST(5), ST(6), ST(7) предназначены для написания программ, использующих тип данных с плавающей точкой.

Пользовательские регистры:

- Целочисленные регистры MMX-расширения - MMX0, MMX1, MMX2, MMX3, MMX4, MMX5, MMX6, MMX7;
- Регистры XMM-расширения с плавающей точкой - XMM0, XMM1, XMM2, XMM3, XMM4, XMM5, XMM6, XMM7;
- Регистры состояния и управления (регистр флагов E FLAGS/FLAGS и регистр указатель команды EIP/IP) содержат информацию о состоянии процессора исполняемой программы и позволяют изменить это состояние.

Системные регистры

- Управляющие регистры – CR0.. .CR4. Они определяют режим работы процессора и характеристики текущей исполняемой задачи.
- Регистры управления памятью - GDTR, IDTR, LDTR и TR используются в защищенном режиме работы процессора для локализации управляющих структур этого режима.
- Отладочные регистры DR0.. .DR7 предназначены для мониторинга и управления различными аспектами отладки;

Системные регистры

- Регистры типов областей памяти MTRR используются для аппаратного управления кэшированием в целях назначения соответствующих свойств областям памяти.
- Машинно-зависимые регистры MSR используются для управления процессором, контроля за его производительностью, получения информации обошибках.