

ГЛАВА 3

Функциональная организация фон-неймановской ВМ

Данная глава посвящена рассмотрению базовых принципов построения и функционирования фон-неймановских вычислительных машин.

Функциональная схема фон-неймановской вычислительной машины

Чтобы получить более детальное представление о структуре и функциях устройств ВМ, представим фон-неймановскую ВМ в виде гипотетической машины с аккумуляторной архитектурой (рис. 3.1).

Примем, что гипотетическая ВМ имеет следующие особенности:

- *Одноадресные команды.* Адресная часть команды (АЧ) содержит только один адрес. При выполнении операций с двумя операндами предполагается, что операнд, адрес которого в команде не указан, находится в специальном регистре АЛУ — аккумуляторе, а также, что результат остается в аккумуляторе.
- *Единство форматов.* Длина команд и данных совпадает с разрядностью ячеек памяти, то есть любая команда или операнд занимают только одну ячейку памяти. В этом случае адрес очередной команды в памяти может быть получен путем прибавления единицы к адресу текущей команды, а для извлечения из памяти любой команды или любого операнда достаточно одного обращения к памяти.

Список команд, выполняемых гипотетической ВМ, приведен в табл. 3.1.

На функциональной схеме (рис. 3.1) показаны типовые узлы гипотетической ВМ, а также сигналы, инициирующие выполнение отдельных операций по пересылке информации и ее обработке.

Устройство управления

Назначение устройства управления (УУ) было определено ранее при рассмотрении структурной схемы ВМ, где отмечалось, что эта часть ВМ организует

автоматическое выполнение программ и функционирование ВМ как единой системы. Теперь остановимся на описании узлов, входящих в состав УУ.

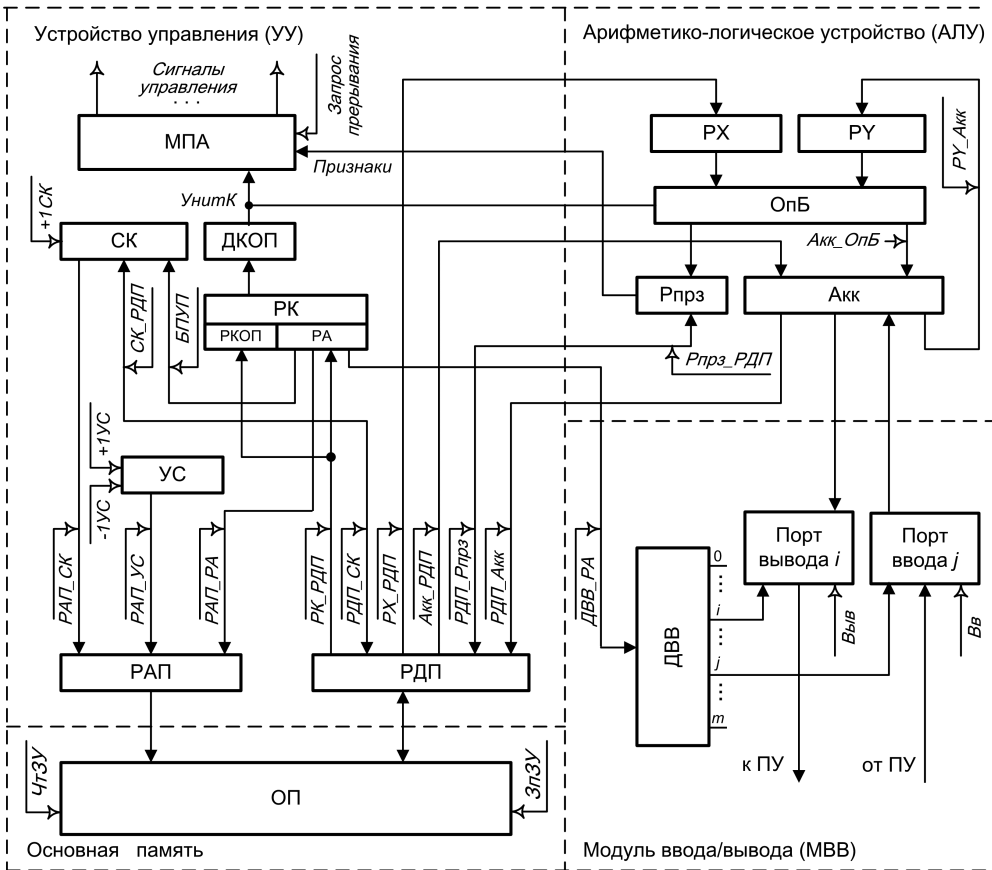


Рис. 3.1. Функциональная схема гипотетической фон-неймановской ВМ

Таблица 3.1. Команды гипотетической вычислительной машины

Мнемоническое обозначение	КОП ¹	АЧ	Описание
LDA	1	ADR	Загрузка в аккумулятор содержимого ячейки основной памяти (ОП) с адресом ADR
STA	2	ADR	Запись содержимого аккумулятора в ячейку ОП с адресом ADR
ADD	3	ADR	Сложение содержимого аккумулятора и ячейки ОП, имеющей адрес ADR. Результат остается в аккумуляторе

продолжение ⇨

Таблица 3.1 (продолжение)

Мнемоническое обозначение	КОП ¹	АЧ	Описание
SUB	4	ADR	Вычитание из содержимого аккумулятора числа, хранящегося в ячейке ОП, имеющей адрес ADR. Результат остается в аккумуляторе
INP	5	IPRT	Ввод в аккумулятор информации с периферийного устройства, подключенного к порту ввода с номером IPRT
OUT	6	OPRT	Вывод содержимого аккумулятора на периферийное устройство, подключенное к порту вывода с номером OPRT
JMP	7	ADR	Безусловный переход к команде, хранящейся по адресу ADR
BRZ	8	ADR	Переход к команде, хранящейся по адресу ADR, при условии, что результат предыдущей арифметической операции равен 0, иначе естественный порядок вычислений не нарушается
	9-E		Прочие возможные команды
HLT	F		Останов вычислений

¹ Шестнадцатеричный код операции.

Счетчик команд

Счетчик команд (СК) — неотъемлемый элемент устройства управления любой фон-неймановской ВМ. В соответствии с принципом программного управления команды программы хранятся в смежных ячейках памяти ВМ, то есть в ячейках со смежными адресами. Выполняются команды в естественной последовательности, в порядке их следования в программе (если изменение этой последовательности не предусмотрено программой). Из этого вывод: адрес очередной команды может быть вычислен путем добавления единицы к адресу текущей команды (в нашем примере любая команда занимает одну ячейку). Такое вычисление обеспечивает счетчик команд. Перед началом вычислений в СК заносится адрес ячейки основной памяти, где хранится первая команда. К моменту завершения очередной команды, не изменяющей естественную последовательность вычислений, содержащее СК увеличивается на единицу (по сигналу управления +1СК). Таким образом, адрес следующей команды программы всегда берется из счетчика команд. Для изменения естественного порядка вычислений (перехода в иную точку программы) вместо прибавления единицы нужно занести в СК адрес точки перехода.

Хотя термин «счетчик команд» считается общепринятым, его нельзя признать удачным из-за того, что он может создать неверное впечатление о задачах данного узла. По этой причине разработчики ВМ используют и иные названия, в частности *программный счетчик* (PC, Program Counter) или *указатель команды* (IP, Instruction Pointer). Последнее определение представляется наиболее удачным, поскольку точнее отражает назначение рассматриваемого узла УУ.

В заключение добавим, что в ряде ВМ роль СК выполняет не двоичный счетчик, а обычный регистр, увеличение содержимого которого обеспечивается специальной внешней схемой (схемой инкремента/декремента).

Регистр команды

Счетчик команд определяет лишь местоположение команды в памяти, но не хранит информации о ее содержании. Чтобы приступить к выполнению команды, ее необходимо извлечь из памяти и разместить в *регистре команды* (РК). Этот этап носит название *выборки команды*. Только с момента загрузки команды в РК она становится «видимой» для процессора. В РК команда хранится в течение всего времени ее выполнения. Как уже отмечалось ранее, любая команда содержит два поля: поле кода операции и поле адресной части. Учитывая это обстоятельство, регистр команды иногда рассматривают как совокупность двух регистров — *регистра кода операции* (РКОП) и *регистра адреса* (РА), в которых хранятся соответствующие составляющие команды.

Если команда занимает несколько смежных ячеек, то код операции всегда находится в том слове команды, которое извлекается из памяти первым. Это позволяет по коду операции определить, требуется ли считывание из памяти и загрузка в РК остальных слов команды. Собственно выполнение команды начинается только после занесения в РК ее полного кода.

Указатель стека

Указатель стека (УС) — это регистр, где хранится адрес вершины стека. В реальных вычислительных машинах стек реализуется в виде участка основной памяти, при этом вершина стека — это ячейка, куда при работе со стеком была произведена последняя по времени запись. Для хранения адреса такой ячейки и предназначен УС. При занесении информации в стек (операция *push*) содержимое УС с помощью сигнала -1УС сначала уменьшается на единицу, после чего используется в качестве адреса, по которому производится запись. Соответствующая ячейка становится новой вершиной стека. Считывание из стека (операция *pop*) происходит из ячейки, на которую указывает текущий адрес в УС, после чего содержимое указателя стека сигналом $+1\text{УС}$ увеличивается на единицу, то есть вершина стека опускается, а считанное слово считается удаленным из стека. Хотя физически считанное слово и осталось в ячейке памяти, при следующей записи в стек оно будет заменено новой информацией.

Регистр адреса памяти

Регистр адреса памяти (РАП) предназначен для хранения исполнительного адреса ячейки основной памяти, вплоть до завершения операции (считывания или записи) с этой ячейкой. Наличие РАП позволяет компенсировать различия в быстродействии ОП и прочих устройств машины.

Регистр данных памяти

Регистр данных памяти (РДП) призван компенсировать разницу в быстродействии запоминающих устройств и устройств, выступающих в роли источников и потребителей информации. При чтении из памяти в РДП заносится считанное содержимое ячейки ОП. В случае записи информация, подлежащая сохранению в ячейке ОП, должна быть предварительно помещена в этот регистр. Собственно момент считывания и записи в ячейку определяется сигналами ЧтЗУ и ЗпЗУ соответственно.

Дешифратор кода операции

Дешифратор кода операции (ДКОП) преобразует код операции в форму, требуемую для работы микропрограммного автомата (МПА). С этих позиций ДКОП правильнее было бы назвать не дешифратором, а преобразователем кодов. В рассматриваемой гипотетической ВМ код операции преобразуется в унитарный код УнитК, в котором каждой команде (каждому КОП) соответствует отдельный бит.

Микропрограммный автомат

Микропрограммный автомат (МПА) правомочно считать центральным узлом устройства управления. Именно МПА формирует последовательность сигналов управления, в соответствии с которыми производятся все действия, необходимые для выборки команд из памяти и их выполнения. Исходной информацией для МПА служат: преобразованный в ДКОП код операции, состояние признаков (флагов), характеризующих результат предшествующих вычислений, а также внешние запросы на прерывание текущей программы.

Арифметико-логическое устройство

Это устройство, как следует из его названия, предназначено для арифметической и логической обработки данных. В машине, изображенной на рис. 3.1, оно содержит следующие узлы.

Операционный блок

Операционный блок (ОПБ) представляет собой ту часть АЛУ, которая, собственно, и выполняет арифметические и логические операции над поданными на вход операндами. Выбор конкретной операции (из возможного списка операций для данного ОПБ) определяется кодом операции команды. В нашей ВМ код операции поступает непосредственно из регистра команды. В реальных машинах КОП зачастую преобразуется в МПА в иную форму и уже из микропрограммного автомата поступает в АЛУ. Операционные блоки современных АЛУ строятся как комбинационные схемы, то есть они не обладают внутренней памятью, и до момента сохранения результата операнды должны присутствовать на входе блока.

Регистры операндов

Регистры РХ и РУ обеспечивают сохранение операндов на входе операционного блока вплоть до получения результата операции и его записи (в нашем случае в аккумулятор).

Регистр признаков

Регистр признаков (Рпрз) предназначен для фиксации и хранения признаков, характеризующих результат последней выполненной арифметической или логической операции. Такие признаки могут информировать о равенстве результата нулю, о знаке результата, о возникновении переноса из старшего разряда, переполнении разрядной сетки и т. д. Содержимое Рпрз обычно используется устройством управления при выполнении команд условных переходов. Под каждый из возможных признаков отводится один разряд Рпрз.

Формирование признаков осуществляется специальным узлом АЛУ, который может быть либо частью ОПБ, либо внешней схемой, располагаемой между операционным блоком и Рпрз.

Аккумулятор

Аккумулятор (Акк) — это регистр, на который возлагаются самые разнообразные функции. Так, в него предварительно загружается один из операндов, участвующих в арифметической или логической операции. В аккумуляторе может храниться результат предыдущей команды и в него же заносится результат очередной операции. Через аккумулятор зачастую производятся операции ввода и вывода.

Строго говоря, аккумулятор в равной мере можно отнести как к АЛУ, так и к УУ, а в ВМ с регистровой архитектурой его можно рассматривать как один из регистров общего назначения.

Основная память

Вне зависимости от типа используемых микросхем, основная память (ОП) представляет собой массив запоминающих элементов (ЗЭ), организованных в виде ячеек, способных хранить некую единицу данных, обычно один байт. Каждая ячейка имеет уникальный адрес. Ячейки ОП образуют матрицу, а выбор ячейки осуществляется путем подачи разрешающих сигналов на соответствующие строку и столбец этой матрицы. Выбор обеспечивается дешифратором адреса памяти, преобразующим поступивший из РАП адрес ячейки в разрешающие сигналы, подаваемые в горизонтальную и вертикальную линии, на пересечении которых расположена адресуемая ячейка. Поскольку для современных ВМ характерна значительная емкость ОП, приходится использовать несколько микросхем запоминающих устройств (ЗУ). В этих условиях процесс обращения к ячейке состоит из выбора нужной микросхемы (на основании старших разрядов адреса) и выбора ячейки внутри микросхемы (определяется младшими разрядами адреса). Первая часть процедуры производится внешними схемами, а вторая — внутри микросхем ЗУ.

Модуль ввода/вывода

Структура приведенного на рис. 3.1 *модуля ввода/вывода* (МВВ) обеспечивает только пояснение логики работы системы ввода/вывода ВМ. В реальных ВМ реализация модуля ввода/вывода может существенно отличаться от рассматриваемой. Задачей МВВ является обеспечение возможности подключения к вычислительной машине различных периферийных устройств (ПУ) и обмена информацией с ними. В рассматриваемом варианте МВВ состоит из дешифратора номера порта ввода/вывода, множества портов ввода и множества портов вывода.

Порты ввода и порты вывода

Портом называют схему, ответственную за передачу информации из периферийного устройства ввода в аккумулятор АЛУ (порт ввода) или из аккумулятора на периферийное устройство вывода (порт вывода). Схема обеспечивает электрическое и логическое сопряжение ВМ с подключенным к нему периферийным устройством.

Дешифратор номера порта ввода/вывода

В модуле ввода-вывода рассматриваемой ВМ предполагается, что каждое ПУ подключается к своему порту. Каждый порт имеет уникальный номер, который указывается в адресной части команд ввода/вывода. *Дешифратор номера порта ввода/вывода* (ДВВ) обеспечивает преобразование номера порта в сигнал, разрешающий операцию ввода или вывода в соответствующем порту. Непосредственно ввод (вывод) происходит при поступлении из МПА сигнала Вв или Выв.

Микрооперации и микропрограммы

Для пояснения логики функционирования ВМ ее целесообразно представить в виде совокупности узлов, связанных между собой коммуникационной сетью (рис. 3.2).

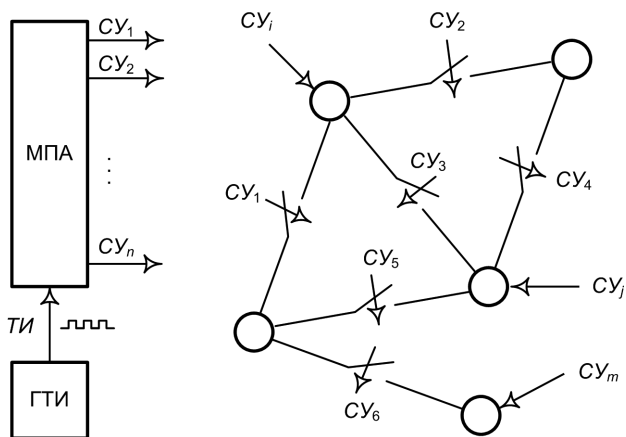


Рис. 3.2. Вычислительная машина с позицией микроопераций и сигналов управления

Функционирование вычислительной машины можно описать как последовательность пересылок информации между ее узлами и элементарных действий, выполняемых в узлах. Понятие узла здесь трактуется весьма широко: от регистра до АЛУ или основной памяти. Также широко следует понимать и термин «элементарное действие». Это может быть установка регистра в некоторое состояние или выполнение операции в АЛУ. Любое элементарное действие производится при поступлении из микропрограммного автомата УУ соответствующего сигнала управления (СУ). Возможная частота формирования сигналов на выходе автомата определяется синхронизирующими импульсами (ТИ), поступающими от генератора тактовых импульсов (ГТИ). Элементарные пересылки или преобразования информации, выполняемые в течение одного такта сигналов синхронизации, называются *микрооперациями*. В течение одного такта могут одновременно выполняться несколько микроопераций. Совокупность сигналов управления, порождающих микрооперации, выполняемые в одном такте, называют *микрокомандой*. Относительно сложные действия, осуществляемые вычислительной машиной в процессе ее работы,

реализуются как последовательность микроопераций и могут быть заданы последовательностью микрокоманд, называемой *микропрограммой*. Реализует микропрограмму (вырабатывает управляющие сигналы, задаваемые ее микрокомандами) *микропрограммный автомат* (МПА).

Способы записи микропрограмм

Для записи микропрограмм в компактной форме используются граф-схемы алгоритмов и языки микропрограммирования.

Граф-схемы алгоритмов

Граф-схема алгоритма (ГСА) имеет вид ориентированного графа. При построении графа оперируют пятью типами вершин (рис. 3.3).

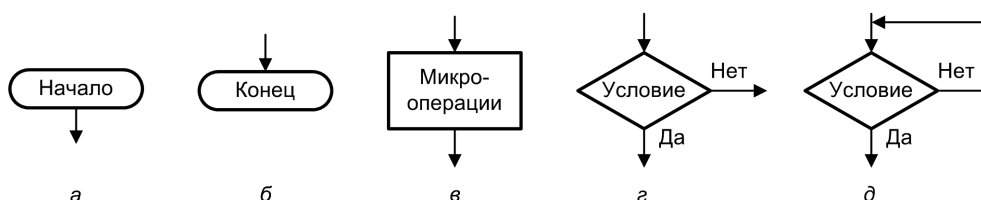


Рис. 3.3. Разновидности вершин граф-схемы алгоритма: а — начальная; б — конечная; в — операторная; г — условная; д — ждущая

Начальная вершина (см. рис. 3.3, а) определяет начало микропрограммы и не имеет входов. Конечная вершина (см. рис. 3.3, б) указывает конец микропрограммы, поэтому имеет только вход. В операторную вершину (см. рис. 3.3, в) вписывают микрооперации, выполняемые в течение одного машинного такта. С вершиной связаны один вход и один выход. Условная вершина (см. рис. 3.3, г) используется для ветвления вычислительного процесса. Она имеет один вход и два выхода, соответствующие положительному («Да») и отрицательному («Нет») исходам проверки условия, записанного в вершине. С помощью ждущей вершины (см. рис. 3.3, д) можно описывать ожидание в работе устройств. В этом случае выход «Да» соответствует снятию причины, вызвавшей ожидание.

Граф-схемы алгоритмов составляются в соответствии со следующими правилами.

1. ГСА должна содержать одну начальную, одну конечную и конечное множество операторных и условных вершин.
2. Каждый выход вершины ГСА соединяется только с одним входом.
3. Входы и выходы различных вершин соединяются дугами, направленными от выхода к входу.
4. Для любой вершины ГСА существует, по крайней мере, один путь из этой вершины к конечной вершине, проходящий через операторные и условные вершины в направлении соединяющих их дуг.
5. В каждой операторной вершине записываются микрооперации y , соответствующие одной микрокоманде Y . Сами микрокоманды указываются возле операторных вершин.

6. В каждой условной вершине записывается один из элементов множества логических условий x .
7. Начальной вершине ставится в соответствие фиктивный оператор y_0 , а конечной — фиктивный оператор y_k .

На рис. 3.4 показан пример микропрограммы, записанной на языке ГСА.

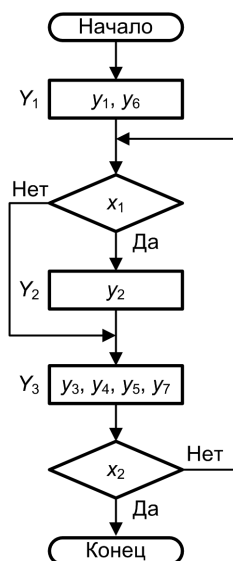


Рис. 3.4. Пример граф-схемы микропрограммы

В примере микрокоманда Y_1 инициирует микрооперации y_1 и y_6 , микрокоманда Y_2 — микрооперацию y_2 , а Y_3 — микрооперации y_3, y_4, y_5 и y_7 .

Языки микропрограммирования

Языки микропрограммирования (ЯМП) обеспечивают описание функционирования ВМ в терминах микроопераций.

Если средства языка ориентированы на запись микропрограммы без привязки к конкретным средствам для их реализации, то такой ЯМП называют *языком функционального микропрограммирования*, а соответствующие микропрограммы — *функциональными микропрограммами* [15]. Подобные микропрограммы служат исходной формой для описания функционирования ВМ.

В случае, когда средства языка нацелены на описание микропрограмм, привязанных к конкретной реализующей их структуре, ЯМП называют *языком структурно-функционального микропрограммирования*.

В последующих разделах для описания функционирования ВМ будет использоваться язык микропрограммирования, предложенный в [20]. Ниже рассматриваются основные средства языка.

Описание слов, шин, регистров

Основным элементом данных, с которым оперирует микропрограмма, является *слово*. Описание слова, шины (совокупности цепей, используемых для передачи слов), регистра состоит из названия (идентификатора) и разрядного указателя. Идентификатором может служить произвольная последовательность букв и цифр, начинающаяся с буквы. Разрядный указатель состоит из номеров старшего и младшего разрядов описываемого объекта, разделенных горизонтальной чертой (дефис) и заключенных в круглые скобки. Разрядный указатель может опускаться, если это не вызывает недоразумений (например, если объект уже был описан раньше). Ниже приводятся примеры описания слова, шины и регистра.

- Аисп(23–0) — описание слова, представляющего 24-разрядный исполнительный адрес $A_{\text{исп}} = a_{23}, a_{22}, \dots, a_0$;
- ША(31–0) — описание 32-разрядной шины адреса;
- РК(31–0) — описание 32-разрядного регистра команды.

Если объект содержит несколько полей, то каждое из них может быть описано отдельно с сохранением общего идентификатора объекта. Так, пусть команда имеет длину 32 бита и состоит из 8-разрядного поля кода операции (КОП), 4-разрядного поля способа адресации (СА) и 20-разрядного поля адреса (А). Тогда описание отдельных полей регистра команды выглядит следующим образом: РК(31–24), РК(23–20), РК(19–0). Вместо номеров разрядов в разрядном указателе можно записывать наименование поля слова, и два первых поля регистра команды могут быть представлены так: РК(КОП), РК(СА). С другой стороны, описание объекта может быть представлено через описание его полей. Например, описание 32-разрядного регистра РПЗ для хранения чисел с плавающей запятой, где число состоит из трех полей: S (поле знака мантиссы, бит 31), P (поле порядка, биты 30–23) и M (поле мантиссы, биты 22–0), может быть задано в виде РПЗ(31 • 30–23 • 22–0) или РПЗ(S • P • M). Здесь точка обозначает операцию конкатенации (составления целого слова из его частей).

Описание памяти, слова памяти

Описание модуля памяти или отдельной его ячейки напоминает описание рассмотренных ранее объектов и отличается тем, что между идентификатором и разрядным указателем размещается адресный указатель, заключаемый в квадратные скобки. В случае описания модуля памяти он содержит два адреса, разделенных двоеточием (слева от двоеточия адрес первого, а справа — адрес последнего слова памяти). При описании ячейки адрес этой ячейки заключается в квадратные скобки. Ниже даны примеры описания модулей памяти и ячеек:

- ОЗУ[0000:1023](7–0) — описание модуля оперативного запоминающего устройства (ОЗУ) емкостью 1 Кбайт;
- ПЗУ[0000₁₆:0FFF₁₆](0–31) — описание модуля постоянного запоминающего устройства (ПЗУ) емкостью 8192 32-разрядных слова (адреса слов указаны в шестнадцатеричном коде, в каждом слове старший разряд имеет номер 0, а младший — 31);