

ПРОГРАММНАЯ МОДЕЛЬ МИКРОПРОЦЕССОРА INTEL 8080. РЕГИСТРЫ: КВАНТОВАННЫЙ УЧЕБНЫЙ ТЕКСТ С ЗАДАНИЯМИ В ТЕСТОВОЙ ФОРМЕ

**Александр Рыбанов,
Лидия Макушкина,**

Волжский политехнический институт (филиал),
Волгоградский государственный технический университет
vit@volpi.ru

1

Аванесов В.С.
Применение заданий в
тестовой форме и
квантованных учебных
текстов в новых
образовательных
технологиях //
Педагогические
измерения. 2012. № 2.
С. 75–91.

2

Рыбанов А.А.
Анализ качества кванто-
вания учебного текста //
Педагогические измере-
ния. 2014. №1. С. 3–16.

3

Макушкина, Л.А.
Автоматизированная си-
стема профессионально-
го отбора и повышения
квалификации персона-
ла сети магазинов Доб-
роСтрой / Макушкина
Л.А., Володькина П.Н.//
Вестник магистратуры.
2013. № 5. С. 53–55.

В данной статье приводится квантованный учебный текст и тестовые задания по дисциплине «Машинно-зависимые языки» для студентов направления 230100 «Информатика и вычислительная техника». Приведены количественные метрики для оценки качества квантования представленного учебного текста. В учебном тексте рассмотрена программная модель микропроцессора Intel 8080 и регистры. Представленный квантованный учебный текст и тестовые задания включены в авторскую автоматизированную обучающую систему по дисциплине «Машинно-зависимые языки».

Ключевые слова: машинно-зависимые языки, микропроцессор, регистры, квантованный текст, тестовые задания, качество квантования.

Введение

Структуризация лекционного материала обеспечивает высокую эффективность его восприятия студентами. Квантование текста¹ является одним из видов подобной структуризации. В настоящее время разрабатываются количественные методы оценки качества квантования учебного текста². Выделение ключевых элементов темы и формирование набора тестовых заданий для контроля знаний по данным элементам позволяют более точно обнаруживать сложные для усвоения студентами элементы курса, которые в дальнейшем могут быть модифицированы: упрощение стиля изложения элементов, расширение материала практическими примерами³ и т.п.

Учебный текст, представленный ниже, включён в теоретическую часть контента авторской автоматизированной обучающей системы по дисциплине «Машинно-зависимые языки»^{4,5}.

Проведённый анализ качества квантованного учебного текста⁶, представленный в табл. 1, показал улучшение его метрик по сравнению с исходным вариантом учебного текста.

Квантованный учебный текст «Программная модель микропроцессора INTEL 8080»

Программная модель микропроцессора

Программные модели более ранних микропроцессоров (МП) (I486, Pentium) отличаются меньшим размером адресуемого пространства оперативной памяти ($2^{32} - 1$ байт), т.к. разрядность их шины со-

ставляет 32 бита и отсутствием некоторых регистров.

Регистры — это память микропроцессоров. Программную модель микропроцессора Intel (рис. 1) составляют:

- 1) пространство адресуемой памяти (для Pentium III: $2^{36} - 1$);
- 2) набор регистров для хранения данных общего назначения;
- 3) набор сегментных регистров;
- 4) набор регистров состояния и управления;
- 5) набор регистров вычислений с плавающей точкой;
- 6) набор регистров целочисленного MMX-расширения, отображённых на регистры сопроцессоров;
- 7) набор регистров MMX-расширения с плавающей точкой;
- 8) программный стек.

Регистры общего назначения

Регистры общего назначения состоят из четырёх регистров

Таблица 1

Анализ качества квантования учебного текста

Метрика качества квантования учебного текста	Исходный текст	Квантованный текст
Сложность формальной удобочитаемости	4,379	3,853
Коэффициент лексического разнообразия	0,281	0,330
Коэффициент синтаксического разнообразия	0,956	0,948

4
Свид. о гос. регистрации программы для ЭВМ № 2013611600 от 28 января 2013 г. РФ, МПК. Программный модуль построения онтологической модели структуры учебного курса / Рыбанов А.А., Макушкина Л.А., Макушкин И.А.; ВолгГТУ. 2013.

5
Свид. о гос. регистрации программы для ЭВМ № 2014611622 от 06 февраля 2014 г. РФ, МПК. Онтологически-ориентированный программный модуль обучения и контроля знаний / Рыбанов А.А., Макушкина Л.А.; ВолгГТУ. 2014.

6
Рыбанов А.А. Количественные метрики для оценки качества квантования учебной информации // Педагогические измерения. 2013. № 4. С. 3–12.

данных (используются для хранения промежуточных данных) и четырёх регистров индексов и указателей.

Регистры данных:

1) *eax* (*ax/ah/al*) – аккумулятор, который иногда используется неявно, а в некоторых командах его использование обязательно;

2) *ebx* (*bx/bh/bl*) – базовый регистр, который используется для хранения базового адреса некоторого объекта в памяти;

3) *ecx* (*cx/ch/cl*) – регистр-счётчик, который применяется в командах, производящих некоторые повторяющиеся действия (например, цикл);

4) *edx* (*dx/dh/dl*) – регистр данных, который используется для хранения данных.

Регистры индексов и указателей:

1) *esi* (*si*) – индекс источника, используется в цепочных операциях и содержит текущий адрес элемента в цепочке-источнике;

2) *edi* (*di*) – индекс приёмника (получателя), используется в цепочных операциях и содержит текущий адрес в цепочке-приёмнике;

3) *esp* (*sp*) – регистр указателя стека, содержит указатель на вершину стека в текущем сегменте стека;

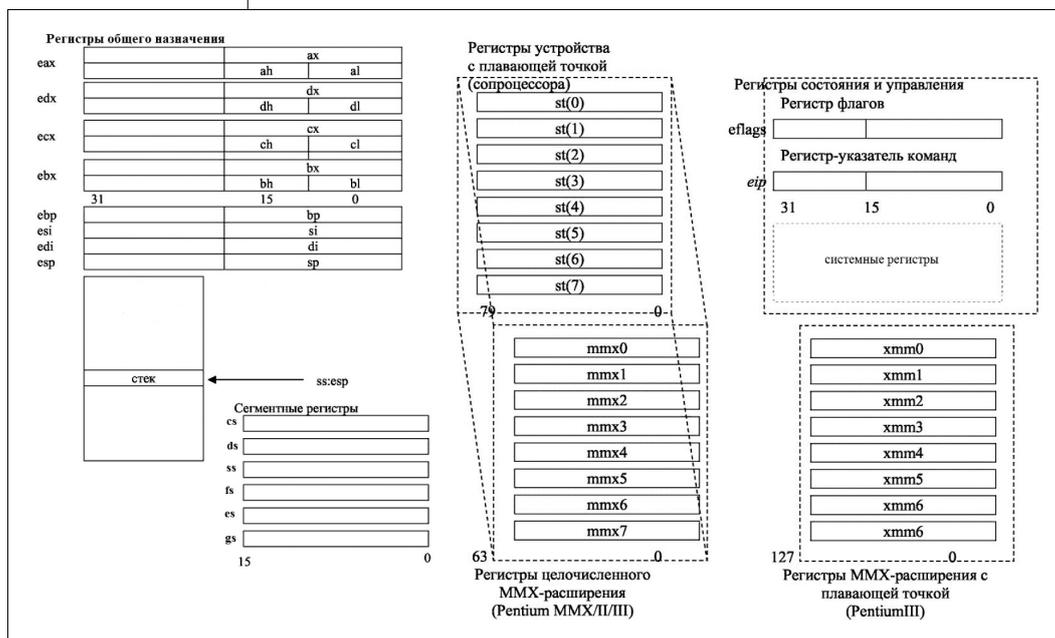


Рис. 1. Программная модель Intel (PentiumIII)

4) ebp (bp) — регистр указателя базы кадра стека, предназначен для организации произвольного доступа к данным внутри стека.

Модели использования оперативной памяти

МП аппаратно поддерживает две модели использования оперативной памяти, где оперативная память (оперативное запоминающее устройство) — это физическая память, к которой МП имеет доступ по шине адреса:

1. Сегментированная модель: программе выделяются непрерывные области памяти (сегменты), а сама программа может обращаться только к данным, которые находятся в этих сегментах.

2. Страничная модель: оперативная память рассматривается как совокупность блоков фиксированного размера (4 Кбайта). Основное применение этой модели связано с организацией виртуальной памяти, что позволяет операционной системе использовать для программ пространство памяти большее, чем объём физической памяти.

Сегментация

Сегментация — это механизм адресации, обеспечивающий

существование нескольких независимых адресных пространств, как в пределах одной задачи, так и в системе в целом для защиты задач от взаимного влияния. В основе механизма сегментации лежит понятие «сегмент», который представляет собой независимый, поддерживаемый на аппаратном уровне блок памяти. Каждая программа может состоять из любого количества сегментов (в зависимости от организации модели памяти), но реально имеется доступ к трем основным сегментам памяти: сегмент кода, сегмент данных, сегмент стека.

Модели сегментированной организации памяти

Внутри сегмента программа обращается к адресам относительно начала сегмента линейно, т.е. начиная с 0 и заканчивая адресом, равным размеру сегмента. Этот относительный адрес, который МП использует для доступа к данным внутри сегмента, называется эффективным. Различают три основных модели сегментированной организации памяти: сегментированная модель памяти реального режима, сегментированная модель памяти защищённого режима, сплошная модель памяти защищённого режима.

Формирование физического адреса в реальном режиме

Физический адрес — это адрес памяти, выдаваемый на шину адреса микропроцессора. Другое название физического адреса — линейный адрес. В реальном режиме механизм адресации физической памяти имеет следующие характеристики:

1) диапазон изменения физического адреса от 0 до 1 Мбайт.

2) максимальный размер сегмента 64 Кбайта.

Для обращения к конкретному физическому адресу оперативной памяти необходимо определить адрес начала сегмента (сегментную составляющую) и смещение внутри сегмента. Физический адрес получается следующим образом:
 $\langle \text{физический адрес} \rangle = \langle \text{смещение} \rangle + 2^4 * \langle \text{регистр сегмента} \rangle$.

Сегментные регистры

Сегментные регистры используются для доступа к соответствующему сегменту. Существуют четыре сегментных регистра:

1) Сегмент кода — содержит команды программы. Для доступа к этому сегменту служит регистр cs (codesegmentregister) — сегментный регистр кода, который содержит адрес сегмента с машинными

командами, к которому имеет доступ МП.

2) Сегмент данных — содержит обрабатываемые программой данные. Для доступа к этому сегменту служит регистр ds (datasegmentregister) — сегментный регистр данных, который хранит адрес сегмента данных текущей программы.

3) Сегмент стека представляет собой область памяти, называемую стеком. Для доступа к этому сегменту служит регистр ss (stacksegmentregister).

4) Дополнительный сегмент данных: если программе недостаточно одного сегмента данных, то она имеет возможность использовать ещё три дополнительных сегмента данных, адресуемых регистрами es, gs, fs.

Регистры состояния и управления

В МП включено несколько регистров, которые содержат информацию о состоянии как самого МП, так и программы, команды которой в данный момент загружены на конвейер: eflags/flags — регистр флагов (Размер регистра flags — 16 бит, eflags — 32 бита.); eip/ip — регистр-указатель команд.

Регистр флагов

Флаги регистра можно разделить на три группы: 8 флагов

состояния, 1 флаг управления (df — отражает направление в строковых командах), 5 системных флагов.

Флаги состояния могут изменяться после выполнения машинных команд, таких как арифметические операции (сложение, вычитание и т.д.), операции сравнения и т.д. Их можно использовать для анализа результата и изменения хода выполнения программы в зависимости от полученного результата.

Флаг управления используется цепочными командами (работа со строками).

Системные флаги управляются вводом/выводом, маскируемыми прерываниями, отлад-

кой, переключением между задачами и виртуальным режимом i8086.

Регистр-указатель команд

Указатель команд — это регистр `еір/ір`, в котором находится значение, представляющее собой смещение в памяти следующей команды, подлежащей выполнению, относительно начала текущего сегмента кода, на начало которого указывает регистр `сs`. Поэтому абсолютный адрес этой команды определяется парой `сs` и `ір`. Менять содержимое данного регистра можно только командами перехода.

ТЕСТОВЫЕ ЗАДАНИЯ

В табл. 2 представлены данные по покрытию фрагментов квантованного текста тестовыми заданиями.

Таблица 2

Покрытие фрагментов квантованного текста тестовыми заданиями

Фрагмент квантованного текста	Номера заданий	Количество заданий
Программная модель микропроцессора	1, 2, 3, 4, 5	5
Регистры общего назначения	6, 7, 8, 9, 10, 11	6
Модели использования оперативной памяти	12, 13	2
Сегментация	14, 15	2
Модели сегментированной организации памяти	16, 17	2
Формирования физического адреса в реальном режиме	18, 19	2
Сегментные регистры	20, 21, 22, 23	4
Регистры состояния и управления	24, 25	2
Регистр флагов	26, 27	2
Регистр-указатель команд	28, 29, 30	3

Вашему вниманию предлагаются задания, в которых могут быть один, два три и большее число правильных ответов. Нажимайте на клавиши с номерами всех правильных ответов:

1. В СОСТАВ ПРОГРАММНОЙ МОДЕЛИ ПРОЦЕССОРА ВХОДЯТ

- 1) стек
- 2) регистры
- 3) математический сопроцессор
- 4) виртуальное адресное пространство

2. КОЛИЧЕСТВО РЕГИСТРОВ В БАЗОВОЙ МОДЕЛИ ПРОГРАММНОЙ АРХИТЕКТУРЫ МИКРОПРОЦЕССОРА

- 1) 8
- 2) 12
- 3) 14
- 4) 16

3. РЕГИСТРАМИ MMX-РАСШИРЕНИЯ ЯВЛЯЮТСЯ

- 1) для целых чисел процессора
- 2) для целых чисел математического сопроцессора
- 3) для вещественных чисел процессора
- 4) для вещественных чисел сопроцессора

4. В ПРОГРАММНОЙ МОДЕЛИ ПРОЦЕССОРА СУЩЕСТВУЮТ РЕГИСТРЫ

- 1) сегментные
- 2) регистры
- 3) указательные
- 4) для работы со строками
- 5) управления и состояния
- 6) отвечающие за точность вычислений

5. ВЕРНЫЕ УТВЕРЖДЕНИЯ:

- 1) регистры — это память микропроцессоров
- 2) разрядность регистра зависит от объёма оперативной памяти
- 3) разрядность регистра зависит от шины адреса
- 4) количество и перечень регистров одинаковы для всех типов микропроцессора

6. РЕГИСТР EAX ИМЕЕТ РАЗРЯДНОСТЬ

- 1) 8 бит
- 2) 16 бит
- 3) 32 бита

7. В СОСТАВ РЕГИСТРОВ ОБЩЕГО НАЗНАЧЕНИЯ ВХОДЯТ

- 1) ax
- 2) ecx
- 3) cs
- 4) ds
- 5) sp
- 6) bp

8. В АРИФМЕТИЧЕСКИХ ОПЕРАЦИЯХ НЕЯВНО УЧАСТВУЕТ РЕГИСТР

- 1) ax
- 2) bx
- 3) cx
- 4) dx

9. КОЛИЧЕСТВО ПОВТОРОВ ДЛЯ ЦИКЛИЧЕСКИХ КОМАНД ХРАНИТСЯ В РЕГИСТРЕ

- 1) ax
- 2) bx
- 3) cx
- 4) dx

10. БАЗОВЫЙ АДРЕС ЭЛЕМЕНТА ПРИ АДРЕСАЦИИ ПО БАЗЕ И ПО БАЗЕ С ИНДЕКСИРОВАНИЕМ ХРАНИТСЯ В РЕГИСТРЕ

- 1) ax
- 2) bx
- 3) cx
- 4) dx

11. АДРЕС ДАННЫХ ХРАНИТСЯ В РЕГИСТРЕ

- 1) ax
- 2) bx
- 3) cx
- 4) dx

12. МОДЕЛИ ИСПОЛЬЗОВАНИЯ ОПЕРАТИВНОЙ ПАМЯТИ

- 1) сегментированная
- 2) страничная
- 3) странично-сегментированная
- 4) энергонезависимая
- 5) распределённая

13. СЕГМЕНТИРОВАННАЯ МОДЕЛЬ ИСПОЛЬЗОВАНИЯ ОП – ЭТО МОДЕЛЬ ПАМЯТИ, В КОТОРОЙ ВЫДЕЛЯЮТСЯ

- 1) непрерывные области памяти
- 2) области памяти определённого размера
- 3) области памяти определённого размера с организацией непрерывного размещения сегментов

14. ПРАВИЛЬНЫЕ УТВЕРЖДЕНИЯ

- 1) сегментация обеспечивает существование нескольких независимых адресных пространств
- 2) при сегментации данные одного блока записываются в оперативную память последовательно
- 3) при сегментации данные одного блока записываются в оперативную память произвольно в соответствии с распределением подсистемы управления памятью

15. СЕГМЕНТАЦИЯ ПОЗВОЛЯЕТ

- 1) изолировать адресные пространства разных задач друг от друга
- 2) объединять адресные пространства разных задач

16. МОДЕЛЯМИ СЕГМЕНТИРОВАННОЙ ОРГАНИЗАЦИИ ПАМЯТИ ЯВЛЯЮТСЯ

- 1) сегментированная модель памяти реального режима
- 2) сегментированная модель памяти защищённого режима
- 3) сплошная модель памяти реального режима
- 4) сплошная модель памяти защищённого режима

17. ВНУТРИ СЕГМЕНТА ПРОГРАММА ОБРАЩАЕТСЯ К АДРЕСАМ

- 1) линейно относительно начала сегмента
- 2) с использованием методов косвенной адресации
- 3) в соответствии с таблицей адресов виртуального адресного пространства процесса

18. ФИЗИЧЕСКИЙ АДРЕС – ЭТО АДРЕС ПАМЯТИ

- 1) используемый регистром команд ip
- 2) выдаваемый на шину адреса микропроцессора
- 3) хранящийся в сегментном регистре

19. ФИЗИЧЕСКИЙ АДРЕС РАССЧИТЫВАЕТСЯ ПО ФОРМУЛЕ

- 1) $\langle \text{физический адрес} \rangle = \langle \text{смещение} \rangle + 24 * \langle \text{регистр сегмента} \rangle$
- 2) $\langle \text{физический адрес} \rangle = 24 * \langle \text{смещение} \rangle + \langle \text{регистр сегмента} \rangle$
- 3) $\langle \text{физический адрес} \rangle = \langle \text{смещение} \rangle + 16 * \langle \text{регистр сегмента} \rangle$
- 4) $\langle \text{физический адрес} \rangle = 16 * \langle \text{смещение} \rangle + \langle \text{регистр сегмента} \rangle$

20. ОСНОВНЫЕ СЕГМЕНТЫ ПАМЯТИ

- | | |
|-----------|-----------|
| 1) данных | 4) памяти |
| 2) стека | 5) кода |
| 3) адреса | |

21. В СОСТАВ СЕГМЕНТНЫХ РЕГИСТРОВ ВХОДЯТ

- | | |
|--------|-------|
| 1) ax | 5) sp |
| 2) ecx | 6) bp |
| 3) cs | 7) ss |
| 4) ds | |

22. АДРЕС НАЧАЛА СЕГМЕНТА КОДА ХРАНИТСЯ В РЕГИСТРЕ

- 1) cs
- 2) ds
- 3) ss
- 4) es
- 5) gs

23. АДРЕС НАЧАЛА СЕГМЕНТА ДАННЫХ ХРАНИТСЯ В РЕГИСТРЕ

- 1) cs
- 2) ds
- 3) ss
- 4) es
- 5) gs

24. В СОСТАВ РЕГИСТРОВ СОСТОЯНИЯ И УПРАВЛЕНИЯ ВХОДЯТ

- 1) регистр флагов
- 2) регистр состояния процессора
- 3) регистры-указатели
- 4) индексные регистры
- 5) регистр состояния сопроцессора

25. РЕГИСТРА EFLAGS ИМЕЕТ РАЗМЕР

- 1) 8 битов
- 2) 16 битов
- 3) 32 бита
- 4) 64 бита

26. В РЕГИСТРЕ FLAGS НАХОДИТСЯ ЧИСЛО ФЛАГОВ

- 1) 5
- 2) 7
- 3) 9
- 4) 16

27. НАПРАВЛЕНИЕ В КОМАНДАХ РАБОТЫ СО СТРОКАМИ ОТРАЖАЕТ ФЛАГ

- 1) zf
- 2) cf
- 3) sf
- 4) df
- 5) af

28. В РЕГИСТРЕ КОМАНД СОДЕРЖИТСЯ

- 1) абсолютный адрес текущей выполняемой команды
- 2) относительный адрес текущей выполняемой команды
- 3) абсолютный адрес команды, которая будет выполнена следующей
- 4) относительный адрес команды, которая будет выполнена следующей

29. ДЛЯ ПОЛУЧЕНИЯ АДРЕСА ВЫПОЛНЯЮЩЕЙСЯ КОМАНДЫ НЕОБХОДИМО СЛОЖИТЬ ЗНАЧЕНИЯ РЕГИСТРОВ

- 1) cs и ip
- 2) ds и ip
- 3) cs и ds

30. ВЕРНЫЕ УТВЕРЖДЕНИЯ

- 1) в регистре ip всегда находится адрес команды, которая должна быть выполнена следующей
- 2) начальное значение, загружаемое в регистр ip, равно 0
- 3) начальное значение, загружаемое в регистр ip, равно 1.