Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

Сибирский государственный университет телекоммуникаций и информатики

(ФГОБУ ВПО СибГУТИ)

Кафедра вычислительных систем (ВС)

Расчётно-графическое задание

по курсу

Архитектура вычислительных систем

вариант 2

Выполнила: Варежникова М.А.

группа ИП-10

Проверил: доцент кафедры ВС

к.т.н. Ефимов А.В.

г. Новосибирск, 2013 г.

Содержание

Постановка задачи3

Анализ архитектуры ЭВМ Дж. фон Неймана 4

Пример функциональной структуры ЭВМ I поколения8

Блок-схема -алгоритма умножения матриц11

Список литературы14

**Постановка задачи**

1. Произвести анализ архитектуры ЭВМ Дж. фон Неймана. Привести пример функциональной структуры ЭВМ I поколения.
2. Построить блок-схему -алгоритма умножения матриц:

А[1:K;1:M]

Е[1:N;1:K]

обеспечивающего распределение элементов результирующей матрицы по горизонтальным полосам в элементарных машинах ВС.

Отыскать максимум коэффициента  накладных расходов при реализации -алгоритма на вычислительной системе, имеющей следующие параметры:

– разрядность 

– полосу пропускания канала между машинами  Мегабод;

– время выполнения операции сложения  мкс;

– время выполнения операции умножения мкс.

1. **Анализ архитектуры ЭВМ Дж. фон Неймана. Привести пример функциональной структуры ЭВМ I поколения.**

**Архитектура фон Неймана**

В каждой области науки и техники существуют некоторые фундаментальные идеи или принципы, которые определяют её содержание и развитие. В компьютерной науке роль таких фундаментальных идей сыграли принципы, сформулированные независимо друг от друга – американским математиком и физиком Джоном фон Нейманом (1903-1957) и советским учёным Сергеем Лебедевым (1902-1974). Эти принципы определяют основные организации компьютера.

В 1946 году Нейман на основе критического анализа конструкции ENIAC предложил ряд новых идей организации ЭВМ, в том числе концепцию хранимой программы, он предложил записывать и хранить в памяти алгоритм вычислений вместе с данными. Принципы Дж. фон Неймана показались вначале простыми и очевидными и лишь в дальнейшем они приобрели статус фундаментальных положений, надолго определивших направление развития вычислительной техники. В результате реализации идей фон Неймана была создана архитектура ЭВМ, во многих чертах сохранившаяся до настоящего времени.

В отчете "Предварительное обсуждение логического конструирования электронного вычислительного устройства" Дж. фон Нейман опубликовал основные принципы, которые заключались в следующем:

1. Компьютеры на электронных элементах должны работать не в десятичной, а в двоичной системе счисления.

2. Компьютер управляется программой, составленной из отдельных шагов - команд. Программа должна размещаться в одном из блоков компьютера - в запоминающем устройстве, обладающем достаточной емкостью и скоростью выборки команд.

3. Команды, так же как и числа, с которыми оперирует компьютер, записываются в двоичном коде. Это обстоятельство приводит к следующим важным последствиям:

а) промежуточные результаты вычислений, константы и другие числа могут размещаться в том же запоминающем устройстве, что и программа;

б) числовая форма записи программы позволяет производить операции над величинами, которыми закодированы команды программы;

в) появляется возможность перехода в процессе вычислений на тот или иной участок программы в зависимости от результатов вычислений, условных переходов.

4. Трудности физической реализации запоминающего устройства, быстродействие которого соответствует скорости работы логических схем требует иерархической организации памяти.

5. Арифметическое устройство конструируется на основе схем, выполняющих операцию сложения - создание специальных устройств для выполнения других операций нецелесообразно.

6. Необходимо использовать параллельный принцип организации вычислительного процесса (операции над словами производятся одновременно во всех разрядах слова).

**Принцип использования двоичной системы счисления** расширил набор физических приборов и явлений, которые можно использовать для представления информации в операционных и запоминающих устройствах компьютера. Две цифры для отображения "1" и "0" могут отображаться состоянием любой двухстабильной системы. Например, открытое и закрытое состояние электронного ключа (ламповой схемы), два состояния триггера, намагниченным или не намагниченным состоянием ферромагнитной поверхности. Ну, а в настоящее время набор электронных приборов и физических явлений, позволяющих получить два состояния для записи и обработки информации, стал намного шире. В двоичной системе счисления возможно построение логических схем и реализация функций алгебры логики или Булевой алгебры.

**Принцип хранимой в памяти программы,** представленной в двоичном коде, позволяет производить не только вычисления, направляя команду в устройство управления, а данные в арифметическое устройство, но и преобразовывать сами команды, например в зависимости от результатов вычислений, используя для преобразования коды команд и оперируя с ними, как с данными.

    **Принцип реализации условных переходов** позволяет осуществлять программы с циклическими вычислениями с автоматическим выходом из цикла. Благодаря принципу условного перехода сокращается число команд, в программе, так как не требуется повторять одинаковые участки программы.

**Принцип иерархической организации памяти** был сформулирован в связи с тем, что с самого первого компьютера с сохраняемой программой существовало несоответствие между быстродействием арифметического устройства и оперативной памяти. Противоречия бы не существовало, если выполнить память на тех же элементах, что и арифметическое устройство, но такая память получалась слишком дорогой, кроме того, непомерно увеличивалось количество радиоламп, что заметно снижало надежность компьютера. Иерархическое построение оперативного запоминающего устройства позволяет иметь быстродействующую память небольшого объема только для данных и команд, подготовленных к выполнению. Все остальное хранится в запоминающем устройстве более низкого уровня, для этого стали использоваться появившиеся вскоре магнитные носители информации.

    **Параллельный принцип организации вычислений** позволяет значительно увеличить скорость вычислений, хотя это и приводит к более значительным затратам оборудования.

**Недостатки и современные перспективы архитектуры фон Неймана**

Архитектура фон Неймана неоднократно подвергалась критике (и, учитывая это, вызывает искреннее удивление ее живучесть, тогда как подавляющее большинство поздних альтернатив является теперь или музейными экспонатами, или прототипами, и вообще никак не могут соревноваться с ней в популярности).

Можно выделить два основных вектора такой критики:

**«Семантический разрыв»**

Хотя это не касается непосредственно принципов фон-Неймана, но часто апеллируют именно к «классической архитектуры фон-Неймана» в критике ее достаточно примитивного и низкоуровневого набора команд, который, по мнению критиков, абсолютно не соответствует современному состоянию дел в индустрии разработки программного обеспечения, в частности в наличии языков высокого уровня, которые намного повышают производительность труда программиста за счет предложения ему более высокоуровневых абстракций, и нужно обычно до нескольких сот машинных команд вместо одной команды языка высокого уровня. Этот дисбаланс в принципе успешно решается на программном уровне с помощью компиляторов, но в 60-70 годы XX века было довольно много попыток реализовать машинные языки высокого уровня аппаратно. Среди отечественных разработок в этом направлении следует выделить ЭВМ серии «МИР», а среди серьезных критиков системы фон-Неймана, в том числе и за низкий семантический уровень команд, академика В. М. Глушкова. Определенной степени, попыткой «повысить семантический уровень» можно считать и CISC -архитектуры системы команд, хотя как доказало время, перспективным оказался прямо обратное направление максимальной «примитивизации» набора команд, реализованный в RISC -архитектурах.

**Разделение операционного устройства и памяти**

Разделение их хранения и памяти в классической архитектуре фон-Неймана считается ее существенным недостатком. Проблема решается за счет построения более сложной иерархии памяти , в частности введением кэш-памяти, более быстрой (но и более дорогой, чем основная), где хранятся данные, которые часто используются в вычислениях, чтобы не обращаться за ними к медленной основной памяти. Существуют также и радикальные предложения, которые в последнее время начали воплощаться в жизнь, и заключаются в создании так называемой «умной памяти», которая бы интегрировала запоминая ячейки со схемами обработки данных.

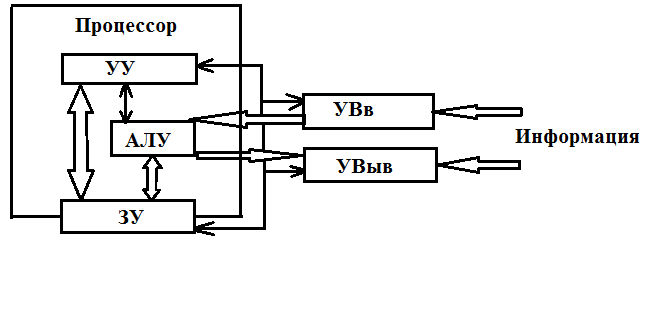
Другим примером частичного решения этой проблемы является гарвардская архитектура, в которой память команд и данных разделена, что позволяет интенсифицировать обмен между запоминающим устройством и центральным процессором.

**Последовательный принцип выполнения**

Архитектура фон Неймана является принципиально последовательной. И это является существенным ограничивающим фактором в повышении быстродействия машин с такой организацией, исключает введение явного параллелизма в систему. Прежде всего это вопрос не технический, а концептуальное и связано с самой парадигмой программирования для фон-неймановской машин. Именно поэтому параллельные вычислительные машины, хотя и успешно выполняют свои задачи, еще долго, видимо, не смогут вытеснить эту классическую архитектуру.

Таким образом, был произведён анализ архитектуры ЭВМ Дж. фон Неймана. Были указаны основные принципы, которые Дж. фон Нейман опубликовал в отчёте "Предварительное обсуждение логического конструирования электронного вычислительного устройства". Также приведены недостатки и современные перспективы архитектуры фон Неймана.

**Функциональная структура ЭВМ Дж. Фон Неймана**

****

**Рис. 1. Функциональная структура ЭВМ Дж. Фон Неймана**

АЛУ – арифметико-логическое устройство;

ЗУ – запоминающее устройство;

УВв – устройства ввода;

УВыв – устройство вывода;

УУ – устройство управления;

- операнды и команды;

- управляющие сигналы.

Каноническую функциональную структуру ЭВМ, представленную на рис. 1, связывают с именем Дж. Фон Неймана.

Функциональное назначение устройств ЭВМ:

АЛУ служит для выполнения арифметических и логических операций над данными (операндами: числами или словами, в частности, буквенными последовательностями), а также операций условного и безусловного переходов;

ЗУ используется для хранения программ и данных;

УВв – для ввода программ и данных;

УВыв – для вывода из ЭВМ любой информации (в частности, результатов);

УУ координирует работу всех остальных устройств при выполнении программ.

Композицию из АЛУ, УУ и части ЗУ называют процессором. Если процессор имеет интегральное исполнение (т.е. представлен одной или несколькими БИС), то его называют микропроцессором.

**Функциональная структура и принцип работы ЭВМ 1-го поколения**

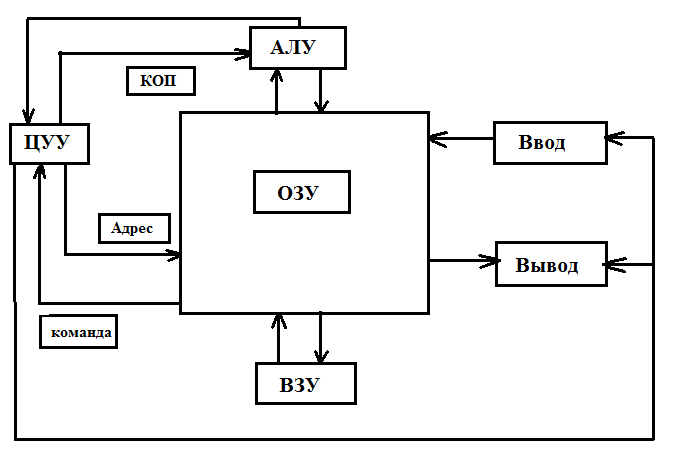
**В качестве элементной базы** машин первого поколения (50-е годы) использовались электронные лампы. Машины были громоздкими, потребляли много электроэнергии.

**Режим работы** – монопрограммный однопользовательский. Программист присутствовал при отладке и счёте своей задачи, контролировал её ход по световой индикации на пульте управления ЭВМ и мог по ходу вычислений переключать работу в однотактный ручной режим для контроля или внесения в программу изменений. Этот режим был очень удобен для программиста, так как машина всегда оставалась в его распоряжении, однако имел чрезвычайно низкий коэффициент использования оборудования.

**Способ обработки информации** – последовательный.

**Алгоритм управления вычислительным процессом –** неизменный (закладывается в аппаратуру при конструировании машины и неизменный впоследствии)

Функциональная структура ЭВМ 1-го поколения (Бэсм-2)



**Рис. 2. Функциональная структура ЭВМ 1-го поколения**

ОЗУ – оперативное запоминающее устройство, разделенное на ячейки, называемые адресом и пронумерованные от 0 до N. В каждой ячейке содержится одно слово. Машинное слово равным образом представляет и команду, и число. Команда – инструкция для выполнения одной операции в ЭВМ, записанная в числовой форме. Из команд состоит программа работы ЭВМ, Количество разрядов в ОЗУ в порядке номеров ячеек. Если слово «берётся» из ОЗУ, то оно как бы считывается, а оригинал остаётся в ячейке. Когда слово «посылается» в ОЗУ, то оно записывается поверх старого значения (старое значение затирается).

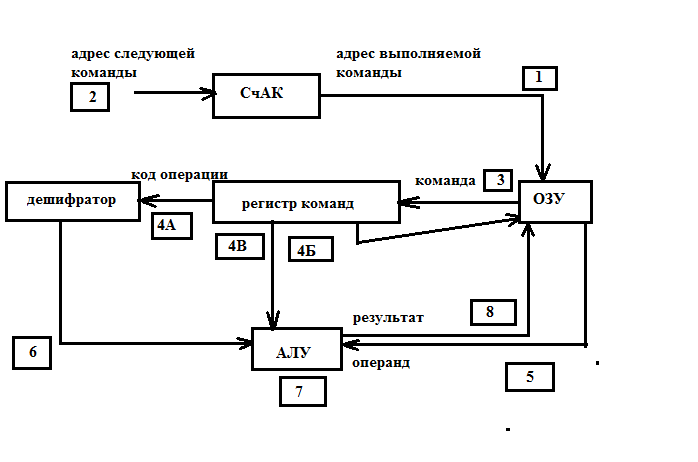
ЦУУ – центральное устройство управления. В него поступают команды. ЦУУ обрабатывает код очередной команды на специальном блоке, называемом дешифратором, и превращает КОП в серию управляющих импульсов, предназначенных для тех устройств, которые участвуют в данной операции. Например, в случае арифметической команды, управляющие импульсы будут направлены в АЛУ. Затем расшифровывается адресная часть команды, и управляющие импульсы посылаются в ОЗУ для выборки слов по соответствующим адресам.

В основе работы ЦУУ лежит «машинный такт». Каждая машинная команда делится на ряд элементарных унифицированных составных частей, называемых тактами. В зависимости от сложности команды она может быть реализована за разное число тактов. Собственно запоминающее устройство называют оперативным потому, что скорость выборки из него данных или посылка в него результата происходят в темпе работы АЛУ, т.е. в соответствии с режимом выполнения машиной одной операции за один рабочий такт.

При выполнении команды ЭВМ проделывает определённые стандартные действия:

* согласно содержимому счетчика адреса команд, считывается очередная команда программы (её код обычно заносится на хранение в специальный регистр УУ, который носит название регистра команд);
* счётчик команд автоматически изменяется так, чтоб в нём содержался адрес следующей команды (в простейшем случае для этой цели достаточно к текущему значению счётчика прибавить некоторую константу, определяющуюся длиной команды);
* считанная в регистр команд операция расшифровывается, из оперативной памяти извлекаются необходимые данные и над ними выполняются требуемые действия;
* полученный результат сохраняется в оперативной памяти.

Полный такт работы ЭВМ представлен на следующем рисунке:

****

**Рис. 3. Полный такт работы ЭВМ**

где:

* СчАК – счётчик адреса команд – регистр, содержащий адрес выполняемой в данный момент программы; во время каждого полного такта из ОЗУ выбирается слово с адресом, равным содержимому СчАК, а в СчАК добавляется 1.
* Регистр команд – регистр, предназначенный для хранения команды, выполняемой в данный момент и выбранный из ОЗУ;
* АЛУ – арифметико-логического устройство;
* ОЗУ – оперативно-запоминающее устройство;
* Дешифратор – устройство (принадлежит ЦУУ), выполняющее расшифровку команды. КОП из регистра команд попадает в дешифратор, превращается там в управляющие импульсы и посылается на те устройства, которые должны выполнять расшифрованную операцию. По адресной части команды вырабатываются сигналы управления для ОЗУ, обеспечивающие чтение операндов по адресам А1 и А2, посылку их в АЛУ и запись результатов по адресу А3.

**N**

-номер последовательности выполнения действия.

АЛУ – арифметико-логическое устройство, выполняющее, как минимум арифметические и логические операции. Для выполнения машинной операции АЛУ должно быть настроено на её выполнение. Операнды в АЛУ посылаются из ОЗУ, туда же записывается и результат операции.

ВЗУ – внешнее запоминающее устройство.

**Область применения первых ЭВМ:** научные расчёты.

Технология программирования: в машинных кодах, автокод (или ассемблер – упрощённое кодирование команд, предполагающее мнемонический КОП и нумерацию ячеек не по физическим адресам, а с опорой на буквенный базис: а+1, а+2 и т.д.).

**Машинами первого поколения были:** Стрела, Бэсм-2, М-2, М-3, Урал-1, Урал - 2.

Таким образом, была приведена функциональная структура ЭВМ первого поколения, описание работы ЭВМ сопровождается схемами и их пояснениями. На схеме также показан полный такт работы ЭВМ.

1. **Построить блок-схему -алгоритма умножения матриц:**

**A[1:K, 1:M], E[1:N, 1:K]**

обеспечивающего распределение элементов результирующей матрицы по горизонтальным полосам в элементарных машинах ВС.

Отыскать максимум коэффициента  накладных расходов при реализации -алгоритма на вычислительной системе, имеющей следующие параметры:

– разрядность 

– полосу пропускания канала между машинами  Мегабод;

– время выполнения операции сложения  мкс;

– время выполнения операции умножения мкс.

А[1:K;1:M] x Е[1:N;1:K] = C[1:N;1:M].

Для построения р-алгоритма прежде всего требуется осуществить распределение исходного массива данных.

Осуществим следующие распределения:

Матрицу А разобьём на n равных вертикальных полос, а матрицу Е на n равных горизонтальных полос.

Вычислитель 1

Вычислитель 2

Вычислитель l

Вычислитель n

n-ый вычислитель

n-ый вычислитель

1

Вычислитель 2

Вычислитель l

Вычислитель n

n-ый вычислитель

Вычислитель 1

Вычислитель 2

Вычислитель l

Вычислитель n

n-ый вычислитель

n-ый вычислитель

Вычислитель 1

1

**А Е С**

**Распределение данных по вычислителям ВС**

**Блок-схема**

**НАЧАЛО**

а :=0

j:=1

а :=а +1

l =a

&

**j>а ]N/n[**

**передача**

**||а1j,…аbj,..аkj||**

j:=j+1

**Вычисление**

**Сij= еibаbj**

**Прием**

**|| а1j,…аbj,..аkj ||**

а =n

**КОНЕЦ**

а – номер передающего вычислителя

{1, 2, .., а-1, а +1,.., n} – номера принимающих вычислителей

Эффективность параллельного алгоритма умножения матриц можно характеризовать показателями:



Максимум коэффициента накладных расходов достигается при ρ=1

Таким образом, максимум коэффициента ε накладных расходов определяется формулой:

ε = tn / (ty + tc)

tn – время пересылки

tу – время умножения

tс  - время сложения

Подставим, известные численные значения в формулу:

tn = l / ν = 32 / 1 \* 106 = 0.0032 мкс, тогда

ε = 0.000032/ (10 + 1) = 2.909 мкс

**Ответ:** Максимум коэффициента накладных расходов равенε=2,909 мкс

**Список литературы**

1. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации. Учебник для вузов. 2-е изд. – СПб.: Питер,2004
2. Жмакин А.П. Архитектура ЭВМ. – СПб.: БХВ — Петербург, 2006
3. Крейгое. Х. Архитектура компьютера и её реализация. Учебное пособие. – С-Пб., Мир, 2004.
4. Ланина Э.П. История развития вычислительной техники. ИрГТУ, Иркутск – 2001 г.
5. Семененко В.А. Электронные вычислительные машины. Учеб. пособие для ПТУ – М.: Высшая школа, 1991
6. Таненбауэм. Э. Архитектура компьютера. Научная литература. – С-Пб., Питер, 2003.
7. Хорошевский В.Г., Архитектура вычислительных систем. – М.: СибГУТИ, 2002.