ТЕХНИЧЕСКОЕ ЗАДАНИЕ.

1. Выбрать логическую структуру процессора (состав устройств,

их функции и принципы взаимодействия), исходя из принципов работы

ЭВМ Единой системы (ЭВМ общего назначения).

2. Обосновать выбор технических параметров процессора с целью

обеспечения достижения заданного значения производительности

- производительность П = млн.оп/с):

- машинный такт Т = 10-100 нс;

- цикл локальной памяти Тл = 10-100 нс;

- время выборки из основной памяти Тп = 100-1000 нс.

3. Разработать временные диаграммы и определить длительность

выполнения основных команд процессора.

4. Оптимизировать технико-экономические параметры процессора для

достижения максимальной его эффективности: максимальной

производительности при минимальной стоимости устройства, исходя из

объема оборудования устройства (количество используемых элементов)

или объема памяти, статистики использования памяти и следующего

соотношения: стоимость элементов устройства обратно пропорциональна

техническим параметрам.

В В Е Д Е Н И Е

Центральный процессор выполняет основную работу по

преобразованию данных в вычислительной системе и, кроме того,

осуществляет в ней функции автоматизированного управления в

соответствии с алгоритмами управляющей программы операционной

системы. В частности, центральный процессор взаимодействует с

каналами ввода-вывода, запуская операции ввода-вывода и получая

информацию о результатах их выполнения, а также о состоянии системы

ввода-вывода.

Центральный процессор - это устройство, обеспечивающее обработку

данных по заданнной программе. Центральный процессор производит

следующие основные виды операций: выполнение команд, прерывание,

сброс, регистрацию состояния (запись информации о состоянии

вычислительной системы в целом или ее отдельных компонентов в

определенные области основной памяти). Программу и обрабатываемые по

ней данные процессор выбирает из основной (оперативной) памяти.

Процессор включает в себя, в большенстве случаев, одно или

несколько операционных (или арифметическо-логических устройств),

устройство управления, локальную память, средства контроля и

диагностики.

Арифметически-логическое устройство (АЛУ) выполняет операции

преобразования данных. Оно включает в себя один или несколько

сумматоров и регистры для хранения промежуточных данных и результатов

преобразований.

Арифметическо-логическое устройство может быть расширино

специализированными операционными устройствами: сдвигателем, быстрым

умножителем, десятичным сумматором, конвертером и др.

Устройство управления (УУ) - автомат управляющий процессами

передачи и обработки информации в процессоре. Это устройство

принимает команды и формирует последовательность управляющих

сигналов, проверяет и т.п. Оно входит в работы функциональных узлов

путем выдачи синхронизирующих и управляющего сигналов.

В составе процессора может находится локальная память

различного функционального назначения: рабочие регистры, РОН,

регистры указатели, управляющие регистры, регистры служебных слов и

т.п. Служебная память может использоваться для буферизации данных и

команд, хранения таблиц преобразования адреса, ключей защиты и др.

Процессор может включать в себя набор специальных

системных средств: службу времени (суточные часы, таймер и т.п.),

средства межпроцессорной связи, пульт управления системой и др.

Средства контоля и диагностики позволяют обнаруживать и

устранять неисправности без потери производитетьности процессора.

ЛОГИЧЕСКАЯ СТРУКТУРА ПРОЦЕССОРА.

Процессор выполняет программу, выбираемую из основной

(оперативной) памяти, в нескольно этапов: выборка команды, распаковка

команды и выборка операндов, выполнение операции и запись результатов

в основную память, а, при необходимости, и обработку прерывания,

изменение состояния процессора или системы в целом.

Логическую структуру ЦП (рис.1) включает ряд функциональных

средств: средства обработки, средства управления системой и

программой, локальная память, средства управления каналами и основной

памятью, системные средства.

Средства обработки обеспечивают выполнение операций с

фиксированной и плавающей запятой, операций с десятичными данными и

полями переменной длины. Локальная память состоит из регистров общего

назначения и регистров с плавающей точкой, а также управляющих

регистров.

Средства управления каналами обеспечивают хранение данных

(буфер данных канала), подготовленных к передаче или принятых из

канала, а также управление приоритетными доступам обрабатывающей

подсистемы через канал ввода/выводда к перефирийному оборудованию.

Центральный процессор ЕС ЭВМ включает в себя следующие базовые

средства принципов работы ЕС ЭВМ:

- локальная память: регистры общего назначения (16х32 р),

регистры плавающей запятой (4х64 р), управляющие регистры (16х32 р);

- полный набор команд: команды арифметики с фиксированной и

плавающей запятой, команды десятичная арифметика;

- системные средства: прямое управление, интервальный таймер,

часы, компаратор, защита памяти, средство условного обмена и т.д.

Для хранения текущей информации обрабатываемой программы в

центральном процессоре имеется локальная память: 16 регистров общего

назначения, 4 регистра (для операндов) с плавающей точкой (по 64

двоичных разряда). Процессор также использует область основной

памяти, которая является постоянно распределенной областью

процессора для хранения управляющей информации, информации прерывания

и контроля.

Центральный процессор может обращаться за информацией к 16

общим регистрам, имеющие нумерацию от 0 до 15. Они могут быть

использованы для хранения индекса в операциях над адресами, как

накапливающие регистры в арифметических операциях с фиксированной

точкой и в логических операциях. В общий регистр можно поместить

одно слово (32 байта).

Регистры общего назначения идентифицируются числами от 0 до 15 и

задаются в команде с помощью четырех битового поля R1. Два

смежных общих регистра (четный и следующей нечетный) могут

использоваться для хранения одного операнда размером 8 байт.

В операционной арифметике с плавающей точкой используются 4

специализированных регистра емкостью 8 байт каждый. Они

идентифицируются номерами 0, 2, 4 и 6. Каждый из них может содержать

короткое (32-битовое) или длинное (64-битовое) число в формате с

плавающей точкой. В операциях арифметики с плавающей точкой

расширенной точности для 128-битовых чисел используются пары

регистров с плавающей точкой 0-2 и 4-6.

К средствам управления памятью относятся буферная память,

память ключей, защита и средства управления доступом к основной

памяти.

К системным средствам относятся средства службы времени:

часы астрономического времени, таймер и компаратор.

Интервальный таймер используется для отчета времени суток и

интервалов времени. Обращение к интервальному таймеру для записи или

чтения его значение может осуществляться любой командой, в которой

предусмотрено обращение к основной памяти.

Компаратор используется для программной привязки процессов к

определенному моменту времени.

Для приближенной оценки производительности процессора будем

использовать ограниченный набор форматов команд и операндов.

Форматы команд:

┌────────┬────┬────┐

│ КОП │ R1 │ R2 │ Формат регистр-регистр (RR)

└────────┴────┴────┘

0 15

┌────────┬────┬────┬────┬────────────┐

│ КОП │ R1 │ Х2 │ В2 │ D2 │ Формат регистр-память (RХ)

└────────┴────┴────┴────┴────────────┘

0 31

КОП - код операции;

R1 - регистр первого операнда;

R2 - регистр первого операнда;

Х2 - регистр индекса;

В2 - регистр базы;

D2 - смещение адреса.

Форматы операндов:

┌──┬───────────────┐

│Зн│ │ Короткий операнд с фиксированной запятой

└──┴───────────────┘

0 15

┌──┬─────────────────────────────────┐

│Зн│ │ Нормальный операнд

└──┴─────────────────────────────────┘ с фиксированной запятой

0 31

┌──┬───────┬─────────────────────────┐

│Зн│Порядок│ Мантисса (24 разряда) │ Короткий операнд с

└──┴───────┴─────────────────────────┘ плавающей запятой

0 7 8 31

┌──┬───────┬────────────────────────── ─ ─ ─ ─────┐

│Зн│Порядок│ Мантисса (56 разрядов) │ Длинный операнд с

└──┴───────┴────────────────────────── ─ ─ ─ ─────┘ плавающей запятой

0 7 8 63

Зн - разряд знака числа (мантиссы).

┌───────────────┐ ┌─────────────┐ ┌────────────┐ ┌─────────────┐

│ Часы суточные │ │ Управляющие │ │ Регистры │ │ Регистры │

├───────────────┤ │ регистры │ │ общего │ │ с плавающей │

│ Компаратор │ │ 16х32 │ │ назначения │ │ точкой │

├───────────────┤ │ разряда │ │ 16х32 │ │ 4х64 │

│ Таймер │ ├─────────────┤ │ разряда │ │ разряда │

├───────────────┤ │ РССП │ └────────────┘ └─────────────┘

│ Таймер │ └─────────────┘ ║ ║

│ интервальный │ ║ ║ ║

└───────────────┘ ╔══════╝ ║ ║

│ ╔═══════════╣ ╔═══════════════════╣ ║

│ ║ ║ ║ ║ ║

┌───────────────────────┐ ┌───────────────────────────────────┐

│ Средства управления │ │ Средства обработки │

├───────────┬───────────┤ ├────────────┬──────────┬───────────┤

│ Управление│Управление │ │ Операции с │Операции с│Операции │

│ системой │программой │─┬─│ фиксирован-│плавающей │десятичной │

└───────────┴───────────┘ │ │ ной точкой │точкой │арифметики │

│ │ │ ║ │ │ │ │и полей пе-│

│ │ │ ║ │ │ │ │ременной │

│ │ │ ║ ┌─┘ │ │ │длины │

│ │ └───────────╫───┼─┐ └────────────┴──────────┴───────────┘

│ └─────────┐ ╠═══╪═╪════════════════════╣ ╔═════════════

┌────────────┐ │ ║ │ └────────────────────╫──╫─────────────

│ Пульт │ │ ║ └────────┐ ║ ║

│ управления │ │ ║ ┌─────┴────────┬──────────┬──────────┐

└────────────┘ │ ║ │ Средства │ Средства │ Средства │

│ ║ │ динамического│ коррекции│ защиты │

│ ║ │ преобразова- │ ошибок │ памяти │

┌────────┐ │ ния адреса │ │ │

│ Каналы │ ├──────────────┴──────────┴──────────┤

│ ввода- │ │ Основная память │

│ вывода │ │ │

└────────┘ └────────────────────────────────────┘

Рис 1. Логическая схема процессора.

ВЫБОР СТРУКТУРЫ ПРОЦЕССОРА.

Выбор структуры процессора определяется двумя факторами:

достижение заданной производительности при минимальных затратах

оборудования.

Известны следующие способы повышения производительности

процессра:

1) совмешение отдельных этапов выполнения последовательно

выбираемых команд и этапов выполнения операций;

2) предвыборка и буферизация команд и операндов;

3) введение нескольких операционных устройств;

4) выбор алгоритмов ускоренного выполнения операций;

5) специализация операционных устройств;

6) введение в структуру буферной памяти (кэш-память);

7) повышение уровня системы команд (спецоперации).

Каждый способ влечет за собой увеличение обьема оборудования, а

значит и увеличение стоимости процессора и, следовательно, снижение

его эффективности, поэтому необходимо оценивать целесообразность

использования каждого способа с учетом затрат оборудования.

На рис.2 показаны временные диаграммы выполнения команды с

разбиением на этапы выполнения: ВК - выборка команды; РК - распаковка

команды; АО - вычисление адреса операнда; ВО - выборка операнда;

ОП - выполнение операции; ЗР - запись результата. При этом отдельные

этапы (РК, АО) могут быть выполнены за один такт, все другие этапы

могут потребовать для своего выполнения нескольких тактов. Совмещение

может выполняться только для этапов одинаковой длины; для случая,

когда длительность совмещаемых этапов различна, длительность

выполнения этапа выбирается по самому большому времени выполнения.

Совмещение не может быть выполнено для взаимозависимых команд, так

как выполнение следующей команды зависит от результата предыдущей

операции: операция перехода или использование результата как операнда

следующей операции.

Время выполнения выполнения команды процессором, при наличии

совмещения, определяется выражением:

Ткт = (N-Nc+1),

где Ткт - время выполнения команды (тактов);

N - количество тактов выполнения команды;

Nc - количество совмещенных тактов.

Таким образом производительность процессора определяется как

количеством тактов выполнения каждой команды и временем перехода

команд, так и количеством уровней совмещения обработки команд.

Метод совмещения выполнения команд обеспечивает увеличение

пропускной способности устройств и при этом не изменяет время

обработки отдельной команды. Поэтому, если среди команд встречаются

зависимые, то пропускная способность процессора снижается на величину

определенную характером зависимости совмещаемых команд. Зависимость

команд как бы уменьшает число уровней совмещения, а следовательно, и

пропускную способность процессора.

Следует учитывать, что совмещение выполнения команд увеличивает

объем оборудования и усложняет схемы управления тем сильнее, чем

больше число уровней совмещения.

При выборе структуры процессора с совмещением выполнения команд

должно быть определено:

- количество независимых исполнительных блоков;

- структура и алгоритмы работы исполнительных блоков;

- организация выполнения команд передачи управления;

- организация внутренней памяти процессора;

- степень совместного использования оборудования процессора в

разных режимах обработки и управления.

На рис.3 представлена структурная схема процессора с полным

совмещением выполнения команд. Блок выборки команд (БВК) содержит

собственный сумматор для вычисления адреса операнда. Буферы команд

предназначены для хранения последовательности исполняемых команд, в

том числе по двум альтернативным направлениям для быстрого перехода.

Блок выборки операндов имеет раздельные регистры адреса и опреранда,

что позволяет совмещать запрос на выборку и прием предыдущего

операнда. Буферная память операндов хранит последовательность

операндов и коды операций последовательно исполняемых команд.

Арифметико-логическое устройство (АЛУ) имеет регистры на входе и

выходе основного сумматора, что позволяет подавать новые операнды

одновременно с записью результата предыдущей операции.

Сверхоперативная буферная память позволяет сократить время

обращения к основной памяти до 2-х машинных тактов.

Критерием эффективности структурного построения процессора для

всех моделей ЕС ЭВМ является отношение производительности к

стоимости, т.е. достижение заданной производительности при наименьших

затратах оборудования. Технико-экономическая эффективность может быть

определена следующим образом:

Р = Пр/Ср (ком/с\*руб),

где Пр - реальная производительность пароцессора (млн.ком/с);

Ср - стоимость процессора (млн.руб).

В расчетах Ср учитываются только основные устройства, указанные

в структуре. Проработка устройства управления не производилась.

а) выполнение одиночной команды: Тк = Твк+Тво+Топ+Тзр+2\*Тм

ВК РК АО ВО ОП ЗР

│----├────┼────┤----│════│----│

N такта: 1 2 3 4 5 6

б) выполнения команд с частичным совмещением:

Команды: Тк = Твк+Тво+Топ+Тзр

ВК РК АО ВО ОП ЗР

N │----├────┼────┤----│════│----│

│ │ │ │ │ ВК РК АО ВО ОП ЗР

N+1 │ │ │ │ │----├────┼────┤----│════│----│

│ │ │ │ │ │ │ │ │ ВК РК АО ВО

N+2 │ │ │ │ │ │ │ │ │----├────┼────┤----│

│ │ │ │ │ │ │ │ │ │ │ │ │

N такта: 1 2 3 4 5 6 7 8 9 10 11 12

в) выполнение команд в режиме полного совмещения:

Команды: Тк = Топ

ВК РК АО ВО ОП ЗР

N │----├────┼────┤----│════│----│

│ │ ВК │ РК │ АО │ ВО │ ОП │ ЗР

N+1 │ │----├────┼────┤----│════│----│

│ │ │ ВК │ РК │ АО │ ВО │ ОП │ ЗР

N+2 │ │ │----├────┼────┤----│════│----│

│ │ │ │ ВК │ РК │ АО │ ВО │ ОП │ ЗР

N+3 │ │ │ │----├────┼────┤----│════│----│

│ │ │ │ │ ВК │ РК │ АО │ ВО │ ОП │ ЗР

N+4 │ │ │ │ │----├────┼────┤----│════│----│

│ │ │ │ │ │ ВК │ РК │ АО │ ВО │ ОП │ ЗР

N+5 │ │ │ │ │ │----├────┼────┤----│════│----│

│ │ │ │ │ │ │ │ │ │

N такта: 1 2 3 4 5 6 7 8 9 10 11

Рис.2. Принципы совмещения выполнения команд.

Команды от ОП

Буферы ┌───────────┬──────────────┐ ┌────────────

команд ┌──┴──┐ ┌──┴──┐ ┌───┴───────┴────┐

├─────┤ ├─────┤ │Сверхоперативная├───────

├─────┤ ├─────┤ │буферная память │ к ОП

├─────┤ ├─────┤ │ (кэш-память) ├──┬────

└──┬──┘ └──┬──┘ └───┬────────────┘ │

┌ ─ ─ ─ ─ ─│─ ─ ─ ─ ─ ─│─ ─ ─ ─ ─ ┐ │ Операнды │

┌───────┴───────────┴───────┐ │ ┌─────────┐ │

┌──┼──┤ Регистр команд ├──┼───┼──┤Локальная│ │

│ └───────┬───────────────────┘ │ │ память │ │

│ │ │ │ │ └──┬────┬─┘ │

│ БВК │ ┌──────────────┼─────┤ │ │

│ │ ──────┴──── ────┴────── │ │ │ │ │

│ Адресный сумматор │ │ │ │

│ │ ──────────┬────────── │ │ │ │ │

│ └ ─ ─ ─ ─ ─ ─ ─ ─│─ ─ ─ ─ ─ ─ ─ ─ ┘ │ │ │ │

│ ├────────────────────┼─────┘ │ │

│ ┌ ─ ─ ─ ─ ─ ─ ─ ─│─ ─ ─ ─ ─ ─ ─ ─ ┐ │ │ │

│ ┌─────────────┴─────────────┐ │ │ │

│ │ │ Регистр адреса │ │ │ │ │

│ └─────────────┬─────────────┘ │ │ │

│ │ └────────────────┼───┼──────────┼────┘

│ БВО ┌────────────────────┤ ┌────┤

│ │ │ │ ┌─┴─────┴─┐ │

│ ┌─────────────┴─────────────┐ │Буферная │ │

│ │ │ Регистр операнда │ │ │ память │ │

│ └─────────────┬─────────────┘ │операндов│ │

│ └ ─ ─ ─ ─ ─ ─ ─ ─│─ ─ ─ ─ ─ ─ ─ ─ ┘ └─┬───────┘ │

│ ├────────────────────┘ │

│ ┌ ─ ─ ─ ─ ─ ─ ─ ─│─ ─ ─ ─ ─ ─ ─ ─ ┐ │

│ ┌─────┴─────┐ ┌─────────────────────┤

│ │ ┌───────┴───┐ ┌────┴───┴──┐ │ │

│ │ Регистр 1 │ │ Регистр 2 │ │

│ │ └───────┬───┘ └────┬──────┘ │ │

│ │ │ │

│ │ ───────┴──── ────┴─────── │ │

└──────── Сумматор │

│ ──────────┬─────────── │ │

┌─────────┴──────────┐ │

│ АЛУ │ Регистр результата ├─────┼──────────────┘

└────────────────────┘

└ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘

Рис.3. Структурная схема процессора с совмещением

выполнения команд.

ВЫБОР АЛГОРИТМА И ВРЕМЕННЫХ ДИАГРАМ ВЫПОЛНЕНИЯ ОСНОВНЫХ ОПЕРАЦИЙ.

Оценим длительность основных операций, имеющих наибольшую

встречаемость (вес) и потому определяющих производительность

процессора (табл.2). Будем делать оценку времени выполнения (Т)

команды в количестве машинных тактов (Тм).

1) Загрузка (регистр-регистр):

┌────────┬────┬────┐

│ КОП │ R1 │ R2 │ L R1,R2 (R2) = (R1)

└────────┴────┴────┘

0 15

По команде LR производится выборка из регистра второго операнда

и запоминание полученного значения в регистре первого операнда, с

одновременным уничтожением его исходного содержимого.

RK BO ЗР Т = 3\*Тм

└────┴────┴────┘ Т =

2) Загрузка (память-регистр):

┌────────┬────┬────┬────┬────────────┐

│ КОП │ R1 │ Х2 │ В2 │ D2 │ L R1,D2(X2,B2)

└────────┴────┴────┴────┴────────────┘ {D2+(X2)+(B2)}=(R1)

0 31

Команда L производит пересылку полного слова по адресу

D2+(X2)+(B2) из основной памяти в регистр общего назначения R1.

РK АО BO ЗР Т = 4\*Тм + Тп

└────┴────┴----┴────┘ Т =

3) Запоминание (регистр-память):

┌────────┬────┬────┬────┬────────────┐

│ КОП │ R1 │ Х2 │ В2 │ D2 │ ST R1,D2(X2,B2)

└────────┴────┴────┴────┴────────────┘ {D2+(X2)+(B2)}=(R1)

0 31

По команде ST содержимое регистра общего регистра R1 помещается

в основную память по адресу, которой определяется как D2+(X2)+(B2).

РК ВО ЗР Т = 3\*Тм + Тп

└────┴────┴----┘ Т =

4) Сложение/вычитание, фиксированная запятая (регистр-регистр):

┌────────┬────┬────┐

│ КОП │ R1 │ R2 │ AR R1,R2 (R1)+(R2) = (R1)

└────────┴────┴────┘

0 15

При выполнение команды AR содержимое регистра R1 складывается с

содержимым регистра R2 и результат помещается в R1.

При выполнение SR от содержимого регистра R1 вычитается содер-

жимое регистра R2 и результат заносится в R1.

РК ВО ОП ЗР

└────┴────┴────┴────┘ Т = 4\*Тм ; Т =

5) Сложение/вычитание, фиксированная запятая (регистр-память):

┌────────┬────┬────┬────┬────────────┐

│ КОП │ R1 │ Х2 │ В2 │ D2 │ A R1,D2(X2,B2)

└────────┴────┴────┴────┴────────────┘ (R1)+{D2+(X2)+(B2)}=(R1)

0 31

По команде A содержимое первого операнда, считываемого из R1,

складывается с содержимым второго операнда, находящегося в памяти по

адресу D2+(B2)+(X2). Результат помещается в R1.

РК АО ВО ОП ЗР Т = 5\*Тм + Тп

└────┴────┴----┴────┴────┘ Т =

6) Сложение/вычитание, плавающая запятая (регистр-регистр):

┌────────┬────┬────┐

│ КОП │FPR1│FPR2│ AER R1,R2 (FPR1)+(FPR2) = (FPR1)

└────────┴────┴────┘

0 15

При выполнении команды AER содержимое регистра плавающей запятой

FPR1 складывается с содержимым регистра FPR2 в следующей последовате-

льности: сравниваются (СП) и выравниваются порядки (ВП),отрицательный

операнд преобразуется в дополнительный код (ДК), результат операции

(ОП) нормализуется (НР) и записывается в FPR1. При выполнении SER из

содержимого регистра FPR1 в той же последовательности вычитается

содержимое регистра FPR2, результат нормализуется и заносится в FPR1.

РК ВО СП ВП ДК ОП НР ЗР Т = 8\*Тм

└────┴────┴────┴────┴────┴────┴────┴────┘ Т =

7) Сложение/вычитание: плавающая запятая, регистр-память

┌────────┬────┬────┬────┬────────────┐

│ КОП │ R1 │ Х2 │ В2 │ D2 │ AE R1,D2(X2,B2)

└────────┴────┴────┴────┴────────────┘(R1)+{D2+(X2)+(B2)}=(R1)

0 31

По команде AE содержимое первого операнда, считываемого из

регистра с плавающей запятой PPR1, суммируется, в указанном выше

порядке, с содержимым второго операнда, находящегося в памяти по

адресу D2+(B2)+(X2). Результат заносится в FPR1.

РК АО ВО СП ВП ДК ОП НР ЗР Т = 9\*Тм + Тп

└────┴────┴----┴────┴────┴────┴────┴────┴────┘

[\* конец страницы \*] Т =

8) Умножение, фиксированная запятая (регистр-регистр):

┌────────┬────┬────┐

│ КОП │ R1 │ R2 │ МR R1,R2 (R1+1)\*(R2) = (R1)

└────────┴────┴────┘

0 15

Сомножители (слово) находятся в регистрах R2 и R1+1. По команде

MR сомножители перемножаются и результат (двойное слово) заносится в

R1 и следующий за ним регистр, обозначаемый как R1+1. Используется

алгоритм умножения одновременно на 2 разряда множимого. Принимается

соотношение коротких и длинных операндов: 80% и 20%.

РК ВО ОП ЗР Т = 10\*Тм\*0.8+19\*Тм\*0.2

└────┴────┴-----------┴────┘ Т =

9) Умножение, плавающая запятая (регистр-регистр):

┌────────┬────┬────┐

│ КОП │FPR1│FPR2│ MER R1,R2 (FPR1)\*(FPR2) = (FPR1)

└────────┴────┴────┘

0 15

Содержимое регистра FPR1 (двойное слово) умножается на

содержимое регистра FPR2, результат (двойное слово) нормализуется

(НР) и помещается в FPR1. Операция суммирования порядков совмещается

с операцией перемножения мантис. Используется алгоритм умножения

одновременно на 2 разряда множимого. Принимается соотношение коротких

и длинных операндов: 80% и 20%.

РК ВО ОП НР ЗР Т = 16\*Тм\*0.8 + 30\*Тм\*0.2

└────┴────┴-----------┴────┴────┘ Т =

10) Деление, фиксированная запятая (регистр-регистр):

┌────────┬────┬────┐

│ КОП │ R1 │ R2 │ DR R1,R2 (R1)/(R2) = (R1,R1+1)

└────────┴────┴────┘

0 15

По команде DR двойное слово в регистре R1 и R1+1 делиться на

содержимое регистра R2. По окончании выполнения деления частное

помещается в R1+1, а остаток - в R1. Применяется алгоритм без

восстановления остатка. Принимается соотношение коротких и длинных

операндов: 80% и 20%.

РК ВО ОП ЗР Т = 19\*Тм\*0.8 + 35\*Тм\*0.2

└────┴────┴-----------┴────┘ Т =

11) Деление: плавающая запятая, регистр-регистр

┌────────┬────┬────┐

│ КОП │FPR1│FPR2│ DER R1,R2 (FPR1)/(FPR2) = (FPR1)

└────────┴────┴────┘

0 15

Делимое считывается из FPR1, делитель из FPR2. По окончании

выполнения операции частное нормализуется и помещается в FPR1.

Принимается соотношение коротких и длинных операндов: 80% и 20%.

Преобразование порядка совмещается с операцией деления.

РК ВО ОП НР ЗР Т = 28\*Тм\*0.8 + 56\*Тм\*0.2

└────┴────┴-----------┴────┴────┘

Т =

12,13) Условный переход

┌────────┬────┬────┬────┬────────────┐

│ КОП │ M1 │ Х2 │ В2 │ D2 │ BC M1,D2(X2,B2)

└────────┴────┴────┴────┴────────────┘ {D2+(X2)+(B2)}=(RPSW)

0 31

При выполнении условия (ПУ), содержащегося в поле M1, происходит

выбор следующей команды по адресу D2+(B2)+(X2), в противном случае

выполняется следующая по порядку команда. При наличии альтернативного

буфера команд считаем вероятность нахождения команды в буфере р= 0.8.

а) Успешный переход:

РК ПУ ВА ВК Т = 4\*Тм\*0.8 + Тп\*0.2

└────┴────┴────┴----------┘ Т =

б) Неуспешный переход:

РК ПУ Т = 2\*Тм ; Т =

└────┴────┘

14) Операция алгебраического сравнения (регистр-регистр):

┌────────┬────┬────┐

│ КОП │ R1 │ R2 │ CR R1,R2 (R1)/(R2) = (R1,R1+1)

└────────┴────┴────┘

0 15

Команда CR производит сравнение содержимого (R2) и (R1) и

устанавливает признак результата в соответствии с этой разностью.

Содержимое регистров остается неизменным.

РК ВО ОП ЗР Т = 4\*Тм

└────┴────┴────┴────┘ Т =

15) Операция сдвига влево/вправо (регистр-память):

┌────────┬────┬────┬────┬────────────┐

│ КОП │ R1 │ │ В2 │ D2 │ SLL R1,D2(B2)

└────────┴────┴────┴────┴────────────┘

0 31

Команда SLL производит сдвиг содержимого регистра R1 влево на

количество разрядов, указанное как адрес второго операнда.

Принимается средне вероятное значение сдвига на 3 разряда.

РК ВА ВК ЗР Т = 6\*Тм

└────┴────┴---------┴────┘ Т =

16) Определим время Тп, исходя из таблицы значений зависимости

вероятности обращения к буферной памяти от ее емкости:

Таблица 1.

┌──────────────────────────────┬────┬────┬────┬────┬────┬────┬────┐

│ Емкость буферной памяти (Кб) │ 2 │ 4 │ 8 │ 16 │ 32 │ 64 │128 │

├──────────────────────────────┼────┼────┼────┼────┼────┼────┼────┤

│ Вероятность обращения (р) │ 90 │ 92 │ 94 │ 96 │ 97 │ 98 │ 99 │

└──────────────────────────────┴────┴────┴────┴────┴────┴────┴────┘

Время обращения к памяти будем определять условно в количестве

машинных тактов и исходя из следующего выражения:

Тп = р\*Тб + (1-р)\*Тц; Тп =

где р - вероятность нахождения информации в кэш-памяти;

Тб - цикл обращения буферной памяти;

Тц - цикл обращения основной памяти.

РАСЧЕТ ПРОИЗВОДИТЕЛЬНОСТИ ПРОЦЕССОРА:

Таблица 2.

┌───┬────────────────────────┬───────┬─────────────┬──────────────┐

│ NN│ Наименование │ Вес │Длитель.(такт) Среднее знач.│

│ пп│ и тип операции │команды├──────┬──────┼──────┬───────┤

│ │ │ % │б/совм│совмещ│такт\*%│ мкс\*% │

├───┼────────────────────────┼───────┼──────┼──────┼──────┼───────┤

│ 1.│Загрузка:память-регистр │ 12.4 │ │ │ │ │

│ 2.│ регистр-память │ 6.2 │ │ │ │ │

│ 3.│ регистр-регистр│ 21.6 │ │ │ │ │

│ │Сложение/вычитание: │ │ │ │ │ │

│ 4.│ ф.зпт, регистр-регистр│ 14.4 │ │ │ │ │

│ 5.│ ф.зпт, регистр-память │ 8.5 │ │ │ │ │

│ 6.│ пл.зпт, регистр-регистр│ 6.4 │ │ │ │ │

│ 7.│ пл.зпт, регистр-память │ 3.5 │ │ │ │ │

│ │Умножение: │ │ │ │ │ │

│ 8.│ ф.зпт, регистр-регистр│ 1.8 │ │ │ │ │

│ 9.│ пл.зпт, регистр-регистр│ 2.6 │ │ │ │ │

│ │Деление: │ │ │ │ │ │

│10.│ ф.зпт, регистр-регистр│ 0.6 │ │ │ │ │

│11.│ пл.зпт, регистр-регистр│ 1.2 │ │ │ │ │

│ │Условный переход: │ │ │ │ │ │

│12.│ успешный │ 10.4 │ │ │ │ │

│13.│ неуспешный │ 2.2 │ │ │ │ │

│ │Логические операции: │ │ │ │ │ │

│14.│ сравнение (ф.зпт) │ 4.8 │ │ │ │ │

│15.│ сдвиг (на 3 разряда) │ 3.4 │ │ │ │ │

├───┼────────────────────────┼───────┼──────┼──────┼──────┼───────┤

│ │ Суммарное значение │ │ -- │ -- │ │ │

└───┴────────────────────────┴───────┴──────┴──────┴──────┴───────┘

Среднее время выполнения команды (такт): Тст = Ткт/100

Тст =

Производительность процессора (ком/такт): Пт = 1/Тст

Пт =

Требуемый машинный такт (мкс): Тм є Пт/Пз

Тм =

Среднее время выполнения команды (мкс): Тсм = Ткм/100

Тсм =

Реальная производительность процессора (млн.ком/с): Пр = 1/Тсм

Пр =

Технико-экономическая эффективность (ком/с\*руб): Р = Пр/Ср

З А К Л Ю Ч Е Н И Е

Результаты разработки процессора ЭВМ Единой системы заданной

производительности показывает, что для данного уровня производитель-

ности технически и экономически целесообразным является использование

элементной базы, позволяющей иметь машинный такт не более 60 нс,

совмещенная обработка, при которой можно выполнять одновременно до 5

команд, и использование сверхоперативной буферной памяти емкостью не

менее 64 Кбайт. Применение других методов ускорения операций:

совмещение на этапе выполнения операции - получение промежуточных

результатов (полусумм и поразрядных переносов); реализация операций

умножения/деления на специальном устройстве с конвейерной обработкой

и др., не целесообразно.

\_