Системы реального времени. Занятие 2

Лектор: ст. преподаватель кафедры ИСУ
Елсукова Е.А.

Тема 2: Аппаратурная среда

- Типы компьютеров, применяемых в СРВ;
- Особенности оборудования, на котором работают СРВ;
- Устройство связи с объектом управления ;
- Кросс-системы.

Типы компьютеров, применяемых в СРВ

- “Обычные” компьютеры;
- Промышленные компьютеры;
- Встраиваемые системы.

“Обычные” компьютеры

По логическому устройству совпадают с настольными системами.

Различия в аппаратном устройстве:

Монитор – ЖСК, (светочувствительное покрытие);

Процессор, память... размещены на съемной плате;

Корпус компьютера обеспечивает защиту от пыли и механических повреждений;

Процессоры – Intel 80×86;

“Обычные” компьютеры

Назначение:

- не управляют оборудованием;
- терминалы для взаимодействия с объектом,
- визуализация состояния оборудования и технологического контроля.

“Обычные” компьютеры

ОС – “обычные” ОС с дополнительными программными комплексами, адаптирующими их к требованиям реального времени.

Промышленные компьютеры. Аппаратное устройство

1 плата: Процессор, Память, Контроллеры,
Программируемые таймеры;
шина VME (Compact PCI);

Корпус: платы ЦАП (DAC), АЦП (ADC),
платы управления оборудованием;

Специальный корпус компьютера (крейт) обеспечивает требуемый температурный режим, защиту от пыли и механических повреждений.

Промышленные компьютеры. Аппаратное устройство

дисковые накопители,
мышь,
стандартная клавиатура,
вентиляторы –
! отсутствуют !

Промышленные компьютеры. Процессор

Доминируют: PowerPC (Motorola-IBM), Sparc (Sun), Motorola 68××(Motorola), Intel80×86, Intel8096×, ARM(ARM);

Определяющие факторы при выборе процессора:

- Получение требуемой производительности при наименьшей тактовой частоте;
- Наименьшее время переключения контекста;
- Наименьшее время реакции на прерывание.

Промышленные компьютеры. Процессор

Актуальная характеристика работы процессора: наименьшая рассеиваемая мощность

Обеспечивает высокую отказоустойчивость системы!

Процессор

"Закон Мура" подтверждается с 1965 года в виде удвоения числа транзисторов на единицу площади каждые два года.

Сегодня самым главным препятствием на пути повышения производительности стал неконтролируемый *рост* плотности *рассеиваемой процессором мощности*.

Если эта тенденция сохранится, процессоры Intel скоро будут выделять больше тепла на квадратный сантиметр, чем поверхность Солнца.

Многоядерные компьютерные системы: от игровых приставок до серверов и суперкомпьютеров
Золотарев С., Рыбаков А,, ЗАО "РТСофт", Электронные компоненты 3/2006

Промышленные компьютеры. Процессор

характеристики работы связанные с подсистемой прерываний:

- Наличие большого количества уровней прерываний (IRQ levels);
- Наименьшее время реакции на прерывание.

Промышленные компьютеры. Программируемые таймеры

Программируемые таймеры – аппаратные устройства, выдающие прерывания через заданные промежутки времени, которые могут работать в периодическом или ждущем режимах.

RTS сама генерирует периодические процессы, которыми управляет (луч радара, движение пресса).

Промышленные компьютеры. Память

1. ПЗУ (ROM) – ОСРВ (500Кб);
2. ОЗУ (RAM) – код и данные ОСРВ (16Мб);
3. Статическое ОЗУ (static RAM, батарея) – критически важные данные (2 Мб); типичное время хранения данных 5 лет;
4. Флеш-память (flash RAM, элективески програмируемое ПЗУ) – роль диска (4Мб);

Промышленные компьютеры. Контроллеры

- Контроллеры памяти,
- периферийных устройств – SCSI, Ethernet, COM-портов, параллельного порта.

Промышленные компьютеры

Назначение - управление оборудованием, объектом.

Для взаимодействия с ними используют “обычные” компьютеры, соединенные через COM-порт, Ethernet.

Промышленные компьютеры

ОС – специализированные ОС реального времени, учитывающие особенности оборудования, компьютера.

Промышленные компьютеры - компьютеры в рабочих спецовках

Встраиваемые системы

Устанавливаются внутри оборудования, которым они управляют. Зависят от размеров этого оборудования.

Встраиваемые системы - Embedded systems

Встраиваемые системы

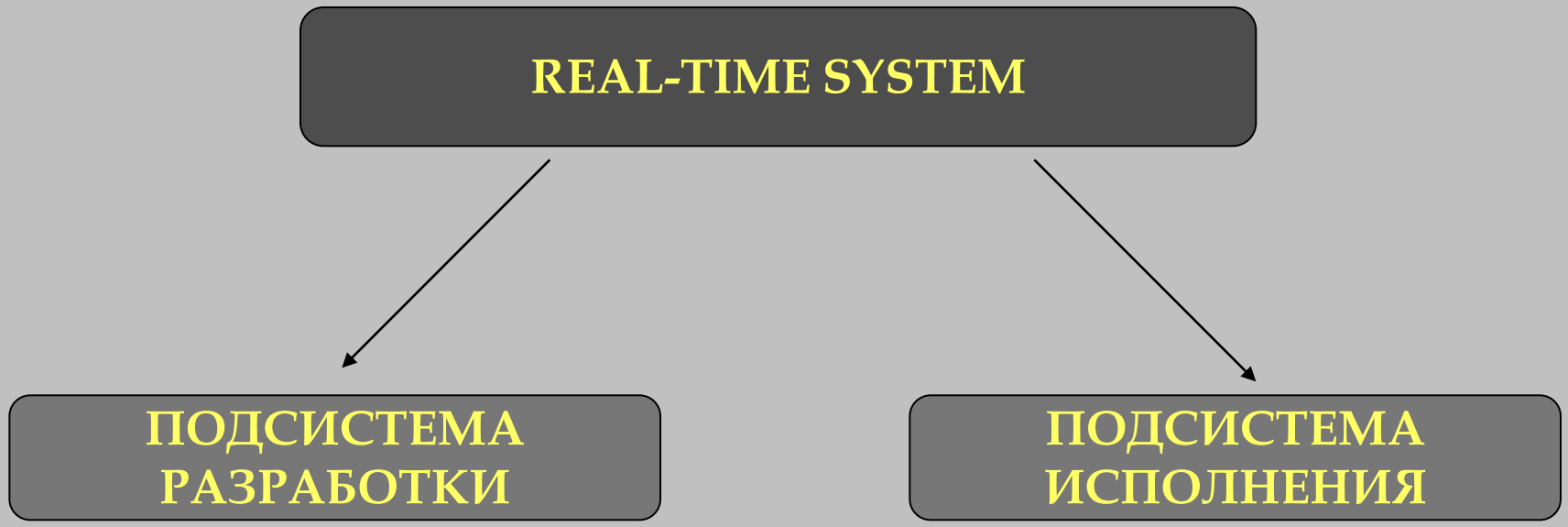
- ✓ Крупное оборудование (локомотив, самолет): по исполнению – промышленные компьютеры.

- ✓ Небольшое оборудование (принтер, цифровой осциллограф) – процессор с сопутствующими элементами, размещенный на плате с другими компонентами оборудования.

- ✓ Миниатюрное оборудование (мобильный телефон, фотокамеры): процессор с сопутствующими элементами – часть интегральной схемы.

Встраиваемые системы

- программное и аппаратное обеспечение, составляющее компоненты другой, *большей* системы и работающее без вмешательства человека.


REAL-TIME SYSTEM
ПОДСИСТЕМА РАЗРАБОТКИ
ПОДСИСТЕМА ИСПОЛНЕНИЯ

Кросс-системы

Программное средство, позволяющее разрабатывать программы для специализированных (промышленных) компьютеров, на которых нет возможности создавать программы и в которые готовая программа "прошивается".

англ. to cross - переходить

Кросс-системы

Исходная (Host) вычислительная система (ВС) - ВС, на которой программа готовится к выполнению.

Целевая (Target) ВС - ВС, на которой программа выполняется.

К*росс-системы* - системы подготовки программ, в которых исходная ВС отличается от целевой.

Кросс-системы – системы, полностью ориентированные на работу с host-машиной

Кросс-системы. Применение

- разработка программного обеспечения встроенных вычислительных систем.

  ресурсов целевой ВС недостаточно для выполнения на ней системного программного обеспечения подготовки программ, тем более - для выполнения интерактивных систем программирования с развитым интерфейсом пользователя.
- При разработке новых ВС создание программного обеспечения для них ведется параллельно с разработкой аппаратной части. Подготовка и отладка программ для проектируемой ВС должна вестись, когда целевой ВС еще не существует физически.

Кросс-системы. Состав

ПОДСИСТЕМА РАЗРАБОТКИ

набор компиляторов и ассемблеров, работающих на host-системе (реже - загружаемых с host-машины в целевую систему),

библиотеки, выполняющие большую часть функций ОС при работе программы (но не загрузку этой программы!),

средства отладки.

ПОДСИСТЕМА ИСПОЛНЕНИЯ

ОСРВ, обеспечивающая работу приложений РВ (ядро, драйверы, исполняемые модули)

Средства отладки

Проблемы комплексной отладки ПО СРВ:

- невозможность обеспечения в процессе отладки внешних воздействий, *адекватных* всем возможным условиям эксплуатации;
- необходимость сохранения технологических средств отладки на весь период эксплуатации ПО СРВ.

Отладка в RTS

Особенности отладки ПО:

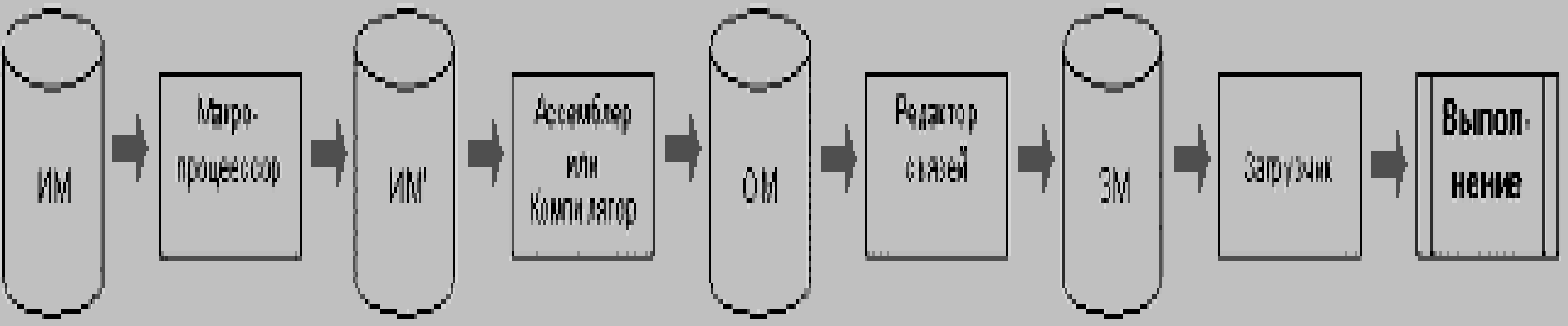
- Этапность отладки:
  - отладка процесса во взаимодействии с ОС,
  - совместная отладка вычислительных процессов внутри ЭВМ,
  - отладка процессов ввода-вывода с отдельными абонентами,
  - комплексная отладка.

Отладка в RTS

Технологическое обеспечение этапов отладки:

- встроенные программные имитаторы (Интерпретаторы),
- использование технологической ЭВМ для имитации внешних воздействий,
- динамический отладчик.

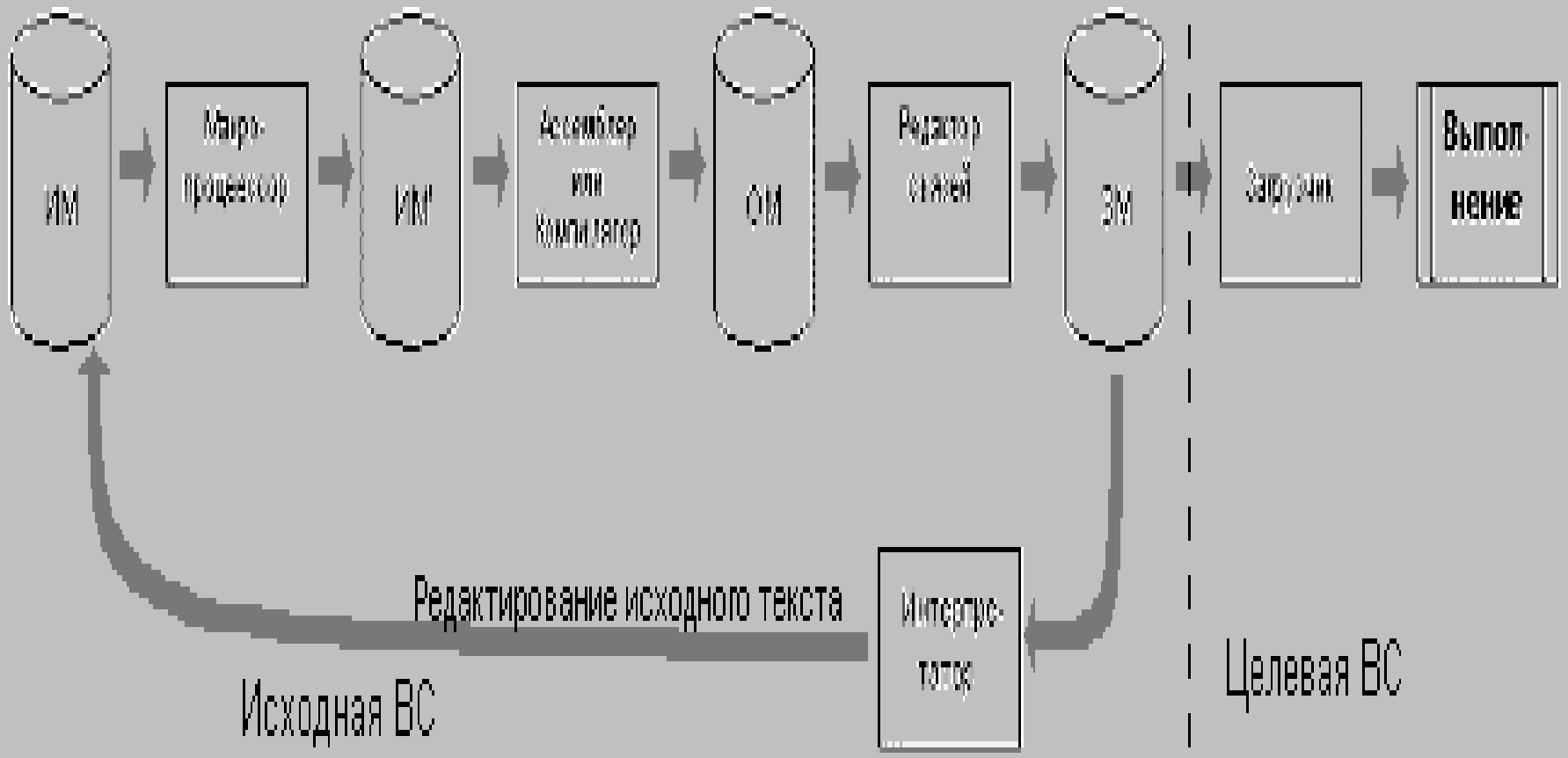
Процесс подготовки программы на кросс-системе аналогичен процессу их подготовки в однородной системе.



Загрузка и выполнение должна обязательно происходить на целевой системе, а предшествующие этапы (все или некоторые) могут быть перенесены на исходную систему. Любой из файлов-результатов выполнения очередного этапа подготовки может быть перенесен из исходной системы в целевую.

Однако, такая схема не в полной мере использует возможности кросс-системы.

Важным этапом подготовки программ является их отладка. Отладка также может проводиться на кросс-системе (частично или полностью) с использованием Интерпретатора.



программа-Интерпретатор

Модель целевой вычислительной системы состоит из компонентов, моделирующих программно-доступные компоненты целевой ВС (т.е. такие, с которыми работают команды отлаживаемой программы) и включает в себя следующие составляющие:

- модель регистров
- модель оперативной памяти
- модель процессора
- модель системы прерывания
- модель системы ввода-вывода.

модель регистров

Модель регистров включает в себя, как минимум:

- регистры общего назначения (РОНы)
- регистр-счетчик адреса (PC)
- регистр состояния (PSW).

Регистры моделируются переменными Интерпретатора.

Модель оперативной памяти

Возможные варианты задания конфигурации:

- Потребовать, чтобы любая ячейка памяти, к которой обращается программа, была описана в программе.
- Описать конфигурацию памяти в отдельном файле, являющимся входным для Интерпретатора.

второй подход более универсальный, так как:

- обращение в программе по неописанному в ней адресу памяти возможно (особенно это касается программ для встроенных ВС с абсолютными программами и жестки распределением памяти);
- определение памяти в программе также является объектом проверки/отладки и может содержать ошибки;
- в Ассемблере нет средств описания ОЗУ/ПЗУ.

Модель оперативной памяти

Внешнее описание памяти считывается Интерпретатором в начале работы и превращается в таблицу фрагментов вида:

Начальный адрес фрагмента	Размер фрагмента (Байт)	Признак разрешения/запрета записи

Модель оперативной памяти

Каждый байт целевой памяти представляется двумя байтами исходной памяти.

В первом байте представления хранятся собственно данные, а во втором - ряд признаков, характеризующих ячейку целевой памяти.

Модель процессора

Работа процессора моделируется алгоритмом работы Интерпретатора. Основной алгоритм работы модели состоит из цикла, в каждой итерации которого моделируется выполнение основных фаз исполнения одной команды целевой программы (выборка команды, вычисление адресов операндов, выборка операндов и т.д.).

Итерация этого цикла начинается с выборки байта, записанному в модели памяти по адресу, содержащемуся в модели регистра-счетчика адреса. В подавляющем большинстве ВС первый байт команды содержит код операции, позволяющий однозначно идентифицировать команду. Интерпретатор выполняет поиск по коду операции в таблице команд.

Модель процессора

При реализации алгоритмов выполнения отдельных команд возможны два подхода, которые называют RISC и CISC-моделями, по аналогии с архитектурами процессоров (однако выбор программной RISC или CISC-модели необязательно должен совпадать с реальной архитектурой процессора).

Модель процессора

Смысл RISC-модели состоит в том, что разветвление алгоритма выполняется сразу же после распознавания команды и выполнение каждой команды полностью реализуется кодами соответствующей ветви.

Время

Время выполнения программы на Интерпретаторе ни в коей мере не соответствует времени ее выполнения на реальной ВС.

Более того, временные соотношения между выполнением различных частей программы на модели также не соответствуют соотношениям выполнения частой программы на реальном оборудовании.

Поэтому время также является *моделируемым* компонентом.

Время

Моделью времени является целая *переменная* большой разрядности.

модель системы прерываний

Является самым сложным для моделирования компонентом. Трудность состоит в том, что прерывания поступают асинхронно, без привязки к выполнению программы.

модель системы прерываний

При выполнении Интерпретатора в пошаговом режиме прерывания могут задаваться командами, вводимыми человеком-оператором.

Более универсальным является прием, предполагающий создание в отдельном файле "программы поступления прерываний". Каждый "оператор" этой "программы" содержит идентификатор типа прерывания и время (модельное) поступления прерывания. Эти "операторы" должны быть упорядочены по возрастанию времен поступления.

Модель системы ввода-вывода

Операции ввода-вывода целевой ВС моделируются *файловым* вводом-выводом исходной ВС.

Данные, которые целевая ВС вводит с внешнее устройство, читаются моделью из файла. Данные, которые целевая ВС выводит на внешнее устройство, записываются моделью в файл.

Для каждого внешнего устройства удобно назначать свой файл. В частном случае это может быть файл клавиатуры или файл экрана. Данные в файл, имитирующий устройство ввода, должны быть занесены заранее.

На вход Интерпретатора должна подаваться таблица соответствия файлов устройствам.

Модель системы ввода-вывода. Синхронный ввод/вывод

При синхронном вводе-выводе (например, через порты) операция ввода-вывода завершается вместе с завершением команды ввода-вывода.

Модель системы ввода-вывода. Асинхронный ввод-вывод

При асинхронном вводе-выводе (КПДП, каналы ввода-вывода) команда ввода-вывода только запускает операцию ввода-вывода и заканчивается. Выполнение операции ввода-вывода далее происходит параллельно с выполнением команд программы, а об окончании ввода-вывода устройство сигнализирует прерыванием.

Взаимодействие с человеком-оператором

Интерпретатор может выполняться в автоматическом или пошаговом режиме.

В автоматическом режиме Интерпретатор моделирует выполнение команд программы без остановок до команды типа HALT или до точки останова. В точке останова оператор может вводить команды, управляющие действиями Интерпретатора и выбрать режим продолжения выполнения.

В пошаговом режиме Интерпретатор после выполнения каждой команды программы останавливается и предоставляет оператору возможность вводить команды управления.

Взаимодействие с человеком-оператором

Командами управления работой Интерпретатора :

- команды на отображение/изменение состояния/содержимого компонентов модели;
- команды задания точек останова;
- команды моделирования прерываний;
- команды установки режима выполнения;
- команда окончания работы.

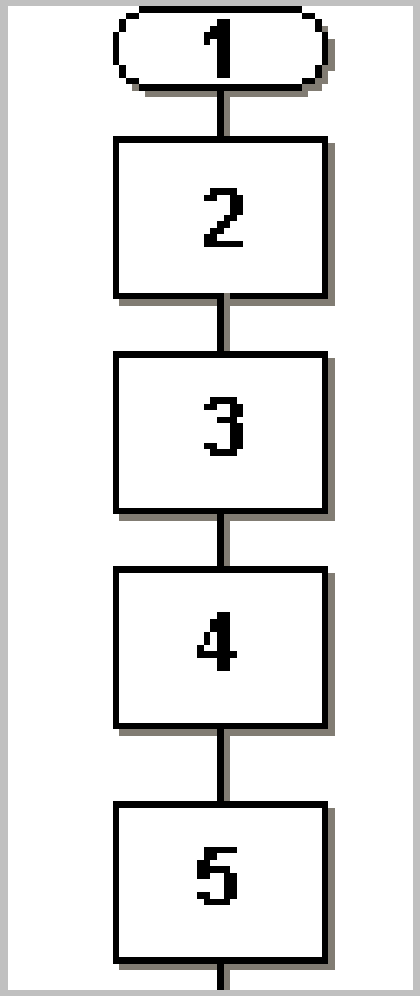
Внесение изменений в ходе отладки

Схема решения представлена на рисунке

Итоговая схема алгоритма Интерпретатора

1. Запуск Интерпретатора.
2. Открытие исходных файлов - результатов работы Кросс-Ассемблера и файлов с управляющей информацией (описание файлов - внешних устройств, программа поступления прерываний, описание фрагментов памяти и отдельных ячеек и т.п.).
3. Считывание управляющей информации.
4. Установка начальных значений для компонентов модели (содержимое памяти, регистры, счетчик модельного времени).
5. Интерактивное задание/корректировка управляющей информации (режим выполнения, точки останова и т.п.).

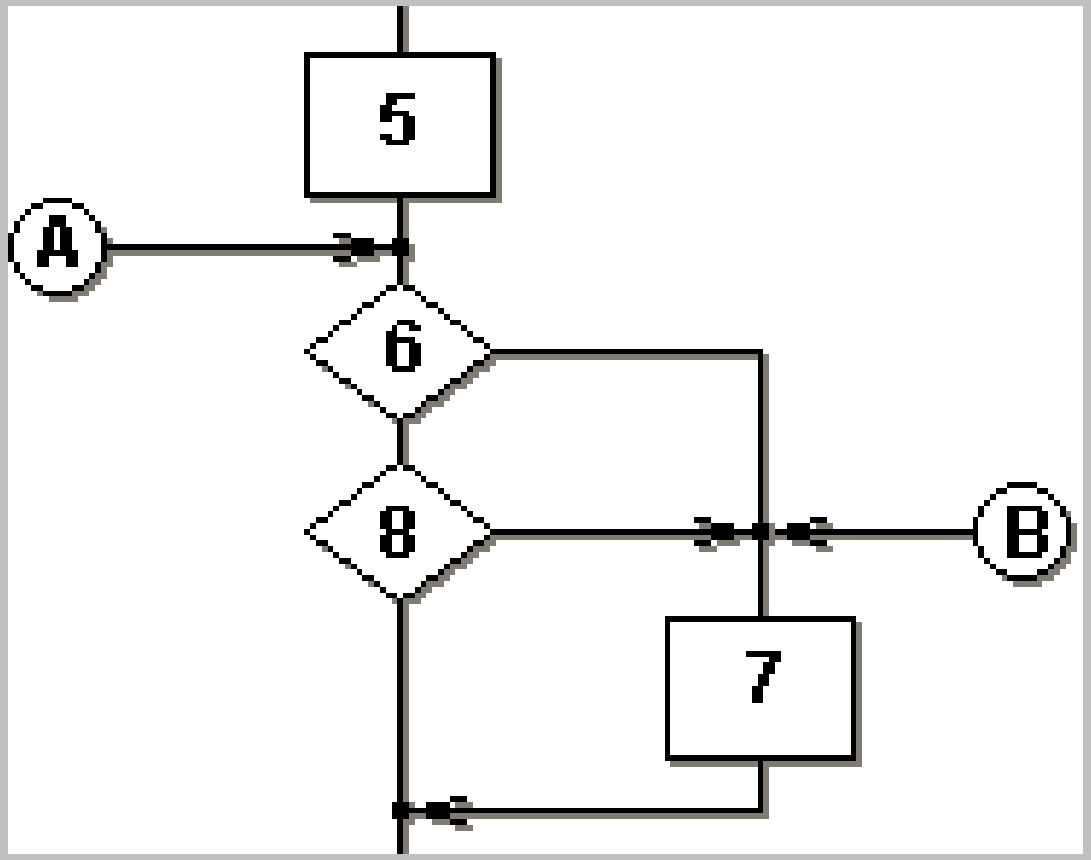
Итоговая схема алгоритма Интерпретатора. Блоки 1 ÷ 5



Итоговая схема алгоритма Интерпретатора

6. Автоматический режим?
7. Если установлен пошаговый (не автоматический) режим выполнения, выполняется ввод и обработка команд оператора в интерактивном режиме. Эта обработка может заканчиваться либо продолжением выполнения интерпретатора в пошаговом или автоматическом режиме, либо завершением его работы по команде оператора.
8. Если установлен автоматический режим выполнения, но текущее значение регистра - счетчика адреса совпадает с одной из заданных точек останова, также выполняется ввод и обработка команд оператора в интерактивном режиме.

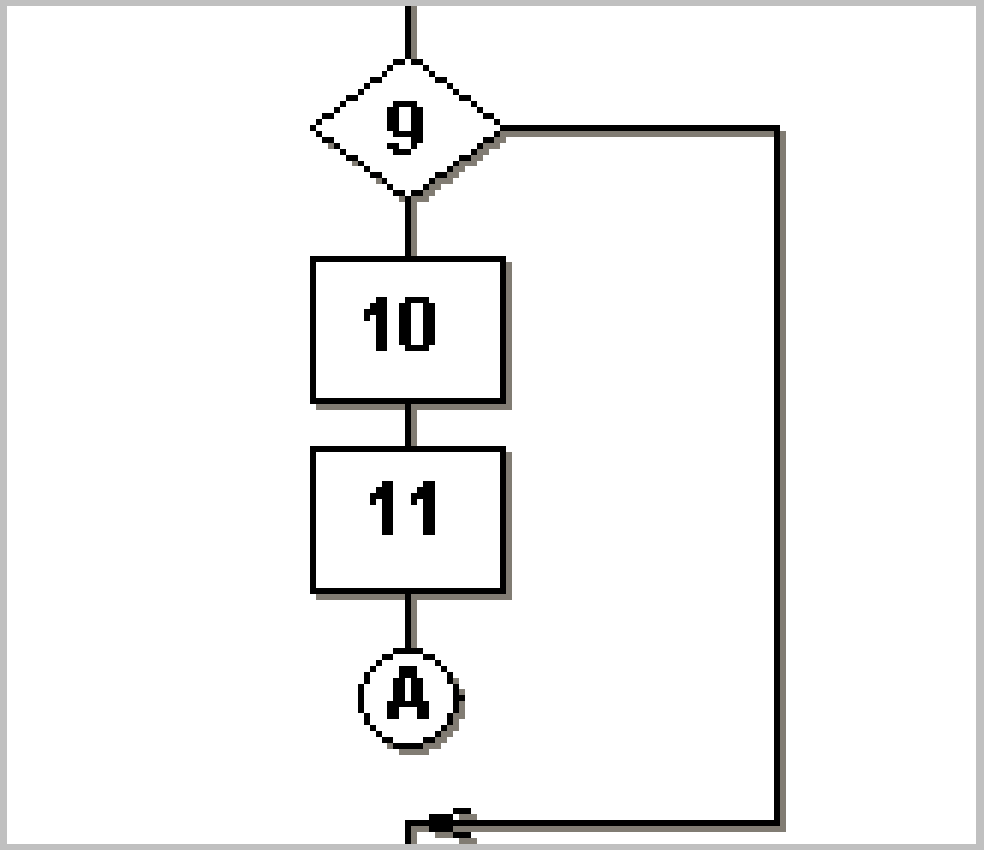
Итоговая схема алгоритма Интерпретатора. Блоки 6÷8



Итоговая схема алгоритма Интерпретатора

9. Проверяется счетчик модельного времени сравнивается с временем поступления первого прерывания в списке прерываний.
10. Если счетчик модельного времени больше или равен времени поступления первого прерывания в списке, выполняется сохранение текущего состояния и занесение в регистр-счетчик адреса секции обработки прерывания данного типа.
11. Первый элемент удаляется их списка прерываний и происходит возврат на начало итерации обработки команды.

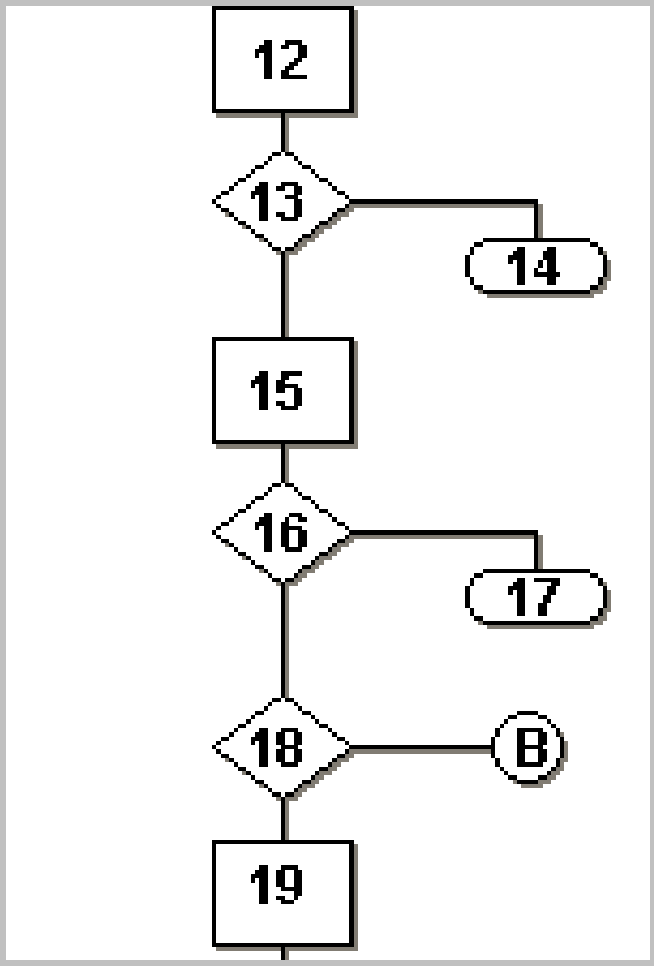
Итоговая схема алгоритма Интерпретатора. Блоки 9÷11



Итоговая схема алгоритма Интерпретатора

12. Если прерывание не поступило, выбирается первый байт команды (при отладке по объектному модулю) или ее мнемоника (при отладке по исходному тексту).
13. Код операции или мнемоника команды ищется в таблице команд.
14. При неуспешном поиске Интерпретатор заканчивается с сообщением об ошибке.
15. Выбор операндов из кода команды или из текста оператора.
16. Проверка правильности кодирования операндов, проверка корректности обращения к памяти.
17. При ошибках в операндах или в обращении к памяти Интерпретатор заканчивается с сообщением об ошибке.
18. Задан ли для адреса операнда останов при обращении? Если да - возврат на выполнение команд в интерактивном режиме.
19. Интерпретация команды и запись результата

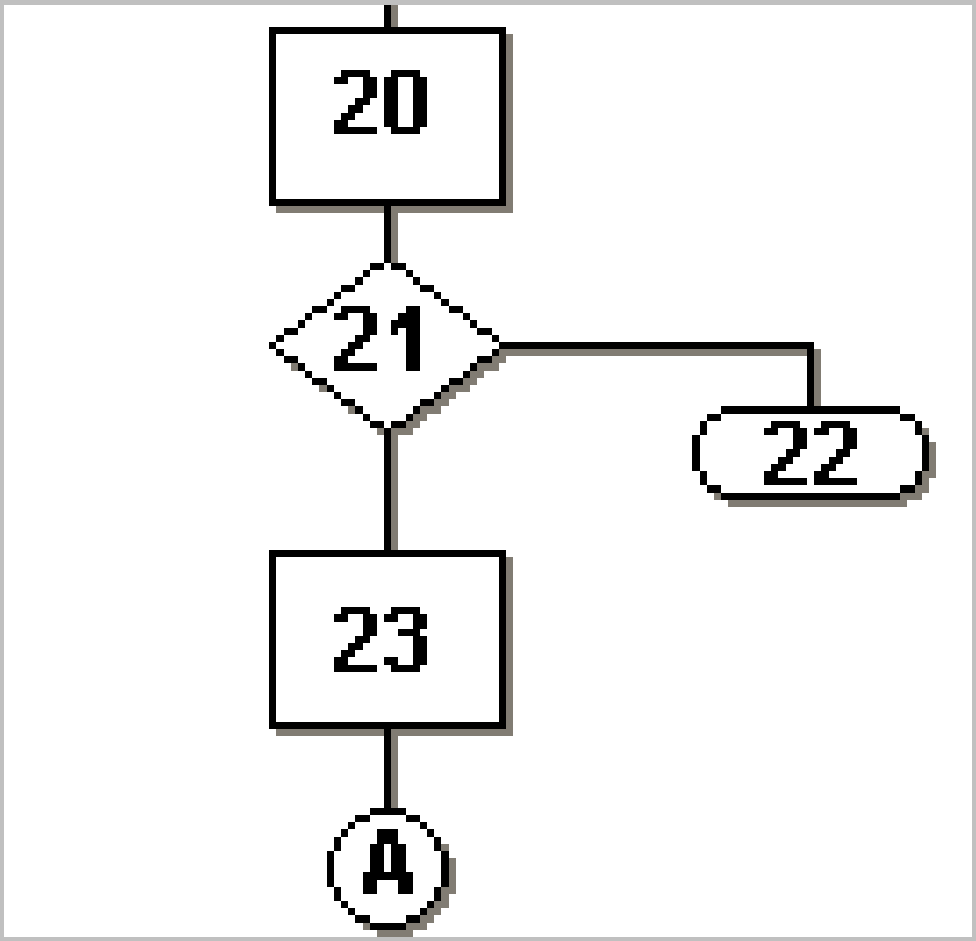
Итоговая схема алгоритма Интерпретатора. Блоки 12-19. Основные фазы исполнения команды



Итоговая схема алгоритма Интерпретатора

20. Вычисление и занесение в регистр-счетчик адреса следующей команды.
21. Проверка, является ли адрес в регистре-счетчике адреса адресом 1-го байта команды
22. Если это не так, Интерпретатор заканчивается с сообщением об ошибке.
23. Модификация счетчика модельного времени и переход на выполнение следующей команды.

Итоговая схема алгоритма Интерпретатора. Блоки 20÷23



Взаимодействие с человеком-оператором

Окончание работы Интерпретатора может происходить:

- при обнаружении ошибки в программе;
- при вводе оператором интерактивной команды завершения работы;
- при обработке команды останова (HALT) в программе.

Были рассмотрены:
- технические средства RTS,
- особенности компьютеров,
- реализации средств связи с объектом управления,
-кросс-системы,
- примеры реализации на основе Интерпретатора.