

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ПСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ**

КАФЕДРА ЭЛЕКТРОПРИВОДА И СИСТЕМ АВТОМАТИЗАЦИИ

УЧЕБНОЕ ПОСОБИЕ ПО КУРСУ

ОСНОВЫ ЧИСЛОВОГО ПРОГРАММНОГО УПРАВЛЕНИЯ

**ДЛЯ СТУДЕНТОВ СПЕЦИАЛЬНОСТИ 140604
"ЭЛЕКТРОПРИВОД И АВТОМАТИКА ПРОМЫШЛЕННЫХ
УСТАНОВОК"**

ПСКОВ

2006

**РЕКОМЕНДОВАНО К ИЗДАНИЮ НАУЧНО-МЕТОДИЧЕСКИМ
СОВЕТОМ ПСКОВСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА**

**АВТОР: ДОЦЕНТ КАФЕДРЫ ЭЛЕКТРОПРИВОДА И СИСТЕМ
АВТОМАТИЗАЦИИ, К.Т.Н ХИТРОВ АЛЕКСАНДР ИВАНОВИЧ**

ПРЕДИСЛОВИЕ

Применение числового программного управления (ЧПУ) промышленными установками и технологическими комплексами позволяет существенно сократить время подготовки нового производства и переналадки оборудования, открывает возможности создания комплексов автоматических линий и гибких автоматизированных производств (ГАП), а также автоматизированных систем контроля (АСК) параметров технологического производства и оборудования.

Курс "Основы ЧПУ " направлен на привитие студентам знаний и навыков по следующим направлениям:

- принципы построения систем числового программного управления;
- особенности аппаратного и программного обеспечения систем ЧПУ;
- электроприводы систем ЧПУ и промышленных роботов и особенности программного управления их движением.

Изучение этого курса важно и потому, что по мнению ряда ведущих специалистов, в частности профессора Боузе (B.Bose)-специалиста в области силовой электроники “ **на сверхавтоматизированном индустриальном фронте следующего столетия две технологии будут иметь наибольшее значение - компьютеры и силовая электроника с автоматизированным электроприводом.**“ Темпы развития вычислительной техники и их эволюция таковы, что не учитывать в современном автоматизированном производстве их особенности не представляется возможным для специалистов в области электропривода.

СОДЕРЖАНИЕ

| | |
|--|-----------|
| Эволюция вычислительной техники и технологии | стр. 6 |
| 1. Общие сведения о системах ЧПУ | 11 |
| 1.1. Определения и терминология..... | 11 |
| 1.2. Цифровые коды в системах ЧПУ..... | 13 |
| 1.3. Код ISO-7 bit | 16 |
| 1.4. Задачи, решаемые устройствами ЧПУ. | 19 |
| 1.5. Классификация систем ЧПУ | 20 |
| 1.6. Обобщенная структурная схема системы CNC-типа и ее особенности | 23 |
| 2. Создание вычислительных сетей на производстве | 27 |
| 3. Выбор информационной системы АСУТП | 34 |
| 3.1.Программируемые логические контроллеры | 37 |
| 3.2.Однокристальные микроЭВМ | 44 |
| 3.3.Однокристальный микроконтроллер INTEL 8XC196MC..... | 46 |
| 3.3.1.Программное обеспечение микроконтроллера MCS-96..... | 52 |
| 3.4.Цифровые сигнальные процессоры | 55 |
| 4.Системные интерфейсы управляющих МП-систем | 59 |
| 5.Элементы теории дискретных САУ | 66 |
| 5.1.Математическое описание процесса квантования | 69 |
| 5.2.Нахождение Z- преобразований | 73 |
| 5.3.Передаточные функции импульсных систем | 74 |
| 5.4.Синтез цифровых систем | 75 |
| 5.5.Анализ цифровых систем | 79 |
| 5.6.Устойчивость цифровых САУ | 81 |
| 6.Структура и программирование прикладных интерфейсов..... | 83 |
| 6.1.Интерфейс КАМАК..... | 86 |
| 6.2.Интерфейс с шиной VME | 92 |

| | |
|--|-----|
| 7. Программное обеспечение систем ЧПУ | 99 |
| 7.1. Системное программное обеспечение | 100 |
| 7.2. Операционные системы реального времени | 104 |
| 7.3. ОС РВ QNX. | 107 |
| 7.4. Прикладные пакеты SCADA для АСУТП | 110 |
| 8. Промышленные роботы | 112 |
| 8.1. Алгоритмы позиционного управления | 116 |
| 8.2. Алгоритмы контурного управления | 118 |
| 9. Электроприводы промышленных роботов и систем ЧПУ... .. | 132 |
| 9.1. Шаговый электропривод..... | 133 |
| 9.2. Электропривод с асинхронными и вентильно-индукторными двигателями..... | 144 |
| 10. Перспективы развития робототехники..... | 149 |
| Заключение..... | 154 |
| Литература..... | 154 |

ЭВОЛЮЦИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И ТЕХНОЛОГИИ.

Рассмотрим историю развития средств вычислительной техники и технологии, которая была представлена в [1], с некоторыми дополнениями. При этом выделим те принципы, которые легли в основу создания автоматизированных производственных систем и комплексов.

4000 - 1300 до н.э. - Представители первой известной шумерской цивилизации записывают информацию на покрытых глиной дощечках. В Вавилоне изобретены счеты.

1642 - 1643 г.г. - Блез Паскаль создал первый механический сумматор на основе зубчатых колес.

1674 г. - Годфрид Лейбниц построил “шаговый счетчик”- калькулятор, использующий ступенчатую передачу, и изобрел двоичный код.

В 18 веке появился прообраз современного программноносителя - кулачковый вал, включавший и выключавший исполнительные механизмы.

1712 - 1725 г.г. - в России А.К. Нартовым создаются станки с принципами копировального управления.

1801 г. - в ткацком станке Джозефа-Мари Жаккарда в Лионе для управления процессом изготовления ткани используется связанная последовательность перфокарт. Уже через 10 лет после изобретения 12000 ткацких станков было внедрено во французской промышленности.

1822 г. - Чарльз Бэббидж приступил к созданию “разностной машины”. Он обосновал работу компьютера для автоматического расчета таблиц в страховании и астрономии. По замыслу Бэббиджа один набор карт в его машине должен был реализовать умножение, а другой решать уравнения. Предполагалось наличие “памяти” (склада), арифметического устройства(мельницы) и блока управления. Машина должна была выполнять сложение за 1 сек., умножение и деление за 1 мин. Ч.Бэббидж считается “отцом “ вычислительной техники и не без основания.

1843 - 1842 - г.г. - Августа Ада, графиня Лавлейс написала первую программу и стала первым программистом в истории человечества,(язык программирования АДА носит ее имя.)

1854 г. - Джордж Буль опубликовал “Исследование законов мышления”, где привел описание системы логических рассуждений (булева алгебра), которая стала фундаментом для компьютерного проектирования.

1901 г. - появился и мало изменился за последующие полстолетия клавишный перфоратор.

1904 - 1906 г.г. - Джон Флеминг и Ли де Форест создают диодную и триодную лампы.

1920 - 1921 г.г. - Карел Чапек в своей фантастической пьесе впервые

употребляет слово “робот” (производное от чешского термина, обозначающего принудительный труд).

1937 г. - В статье Алана Тьюринга “О вычислимых числах” введено понятие машины Тьюринга, в которой был предложен принцип цифровой обработки информации. Его модель цифрового компьютера состояла из бесконечной бумажной ленты и головки для считывания и обработки записанных на ней символов. В этом же году Говард Эйкен предложил фирме IBM, созданной в 1924 г., разработать цифровую вычислительную машину, способную выполнять 4 фундаментальных арифметических операции в определенном порядке.

1943 г. - в Англии построена первая ЭВМ COLOSSUS. Она была создана в британской разведке, где А. Тьюринг работал дешифровальщиком. В Германии придумывались в это время изоцифренные системы шифровки, но они в тот же день расшифровывались в Лондоне. Тьюринг разработал сложный алгоритм, основанный на корреляции перехваченного сообщения с неким эталонным текстом, который и поныне является секретом британской разведки. Пока система шифровки была относительно простой, удавалось выполнять дешифровку за один проход текстов, но с усложнением кодирования требовалось множество таких проходов и перфоленты с эталонным и зашифрованным текстом такой нагрузки не выдерживали. Было создано релейной устройство для хранения и многократного считывания текстов. Это и был первый компьютер.

1944 г. - в Гарвардском университете открыта построенная Эйкеном машина Howard Mark 1, состоящая из 150000 деталей и 35 км. проводов с быстродействием, не превышающим одной операции за 10 сек.

1945 г. - Д. Мокли и Дж. Эккерт создают компьютер ENIAC на базе 18000 вакуумных приборов массой в 30 т., площадью 15000 м² и стоимостью 2,5 млн. долларов. Джон фон Нейман вводит понятие запоминаемой программы.

1947 г. - Джон Бардин, Уолтер Бреттейн, Уильям Шокли разработали первый транзистор.

1948 г. - Г. А. Спыну в СССР создал систему управления револьверным станком на базе использования в качестве программоносителя магнитной ленты и шагового электропривода. Клод Шеннон опубликовал “Математическую теорию связи”, заложив основу современного понимания коммуникационных процессов.

1949 г. - в СССР под руководством С.А.Лебедева построена малая электронно-счетная машина (МЭСМ).

1953г. - выпускается мэйнфрейм (большая ЭВМ) IBM 650, получивший признание как первый коммерческий компьютер. В СССР выпущены

серийные ЭВМ - Стрела и М2 с быстродействием до 20000 операций в сек.

1954 г. - Фирма Texas Instrument (TI) объявила о выпуске кремниевого транзистора. Дж. Девол подал заявку на конструкцию, которая считается первым промышленным роботом.

1957 г. - Джон Бэкус и его сотрудники установили в IBM первый компилятор языка FORTRAN. В России запущен первый спутник и началась космическая гонка, подстегнувшая дальнейшее развитие средств вычислительной техники.

1959 г. - Организованная в 1956 г. Д. Энгельбергером и Дж. Деволом робототехническая компания "CONDEC" (прототип UNIMATION) изготовила для завода GENERAL MOTORS первый робот UNIMATE. Робот извлекал из литейной формы раскаленные заготовки и складывал их в стеллажи. Позже фирма AMF выпустила робот VERSATRAN. В СССР в экспериментальном НИИ металлорежущих станков создается цифровая система управления, удостоенная первой премии на Международной выставке в Брюсселе.

1959 г. - Роберт Нойс и Гордон Мур от имени Fairchild Semiconductor подали заявку на патентование технологии интегральных микросхем.

1960 г. - компания Digital Equipment Corp.(DEC), основанная в 1958 г., объявила о выпуске PDP-1, первого коммерческого компьютера с монитором и клавиатурным вводом.

1963 г. - Американский институт национальных стандартов (ANSI) сертифицировал код ASCII 7.

1964 г. - IBM объявляет о производстве серии компьютеров "третьего поколения" - SYSTEM-360. Дан Энгелбарт изобретает компьютерную мышь.

1965г. - DEC выпустила PDP-8 первый миникомпьютер на базе транзисторных схем.

1968 г. - Р.Нойс, Э. Гроув и Г. Мур основали корпорацию INTEL. Эдсгар Дейкстра обосновал вредность оператора GOTO, что явилось началом внедрения элементов структурного программирования.

1969г. - BELL LABS приступила к созданию операционной системы UNIX. Введен в действие стандарт RS-232, облегчающий обмен данными между компьютерами и периферийными устройствами.

1970 г. - Появились первые гибкие (floppy) диски. Деннис Ричи и Кеннет Томсон разрабатывают UNIX.

1971 г. - Тед Хофф, С. Мэзор, Ф.Фэджин - разработали микропроцессор INTEL - 4004 - первый "компьютер на чипе". Никлаус Вирт предложил язык PASCAL.

1972г. - появился INTEL -8008, первый восьмиразрядный микропроцессор.

Создан язык СИ (его предшественник фирмы XEROX имел имя ВИ). Появились на рынке компьютеры PDP-11-45 фирмы DEC.

1973 г. - в фирме XEROX разработан экспериментальный ПК, названный ALTO, в котором применялась мышь, ETHERNET и графический интерфейс пользователя. В Стэнфордском Университете под руководством Винтона Серфа началась работа над TCP (Transmission control protocol) для распределенных систем передачи и обработки данных.

1975 г. - Первый ПК - ALTAIR 8800, доступный для потребителя в виде комплекса компонентов появился на обложке журнала "Popular Electronics". В IBM Д. Кокке начал работу над проектом создания процессора с сокращенным набором команд- RISC (Reduced Instruction Set Computer) в отличие от процессоров со сложным набором команд - CISC (Complex).

1976 г. - Гэри Килдэлл разработал операционную систему CP-M для восьмиразрядных ПК. Стив Джобс и Стив Возняк спроектировали и построили на основе монтажных плат компьютер APPLE 1 и учредили компанию APPLE Computer.

1977 г. - Билл Гейтс и Пол Аллен основывают фирму MICROSOFT.

1978 г. - DEC выпустила VAX 11-780 - 32 разрядный компьютер, ставший популярным в научных и технических приложениях.

1979 г. - MOTOROLA выпустила микропроцессор M68000, который позднее поддержал компьютеры MACINTOSH.

1980 г. - IBM выбрала PC DOS от MICROSOFT в качестве операционной системы для своих ПК. Создан язык АДА, предназначенный для встроенных приложений и управления процессами в реальном масштабе времени. Уэйн Рэтлифф разработал dBase - первую СУБД для ПК. Создается первый "портативный" (переносной) компьютер Osborne 1 весом 24 фунта и размером с чемодан.

1981 г. - Началось массовое производство IBM PC с открытой архитектурой, что стало сигналом о начале рыночной экспансии персональных компьютеров.

1982 г. - Журнал "TIME" назвал компьютер "человеком года". В Японии приступили к проекту "компьютерные системы 5 поколения", ориентируясь на принципы искусственного интеллекта.

1983 г. - Появление и переход на протокол TCP-IP ознаменовал начало эпохи создания глобальной "сети-сетей".

1984 г. - Устройство CD-ROM, введенные компаниями SONY и PHILIPS обеспечили значительное увеличение объема памяти для цифровых данных. MOTOROLA выпустила MC 68020 - чип на базе 250000 транзисторов. INTEL приступила к производству 16-разрядных

процессоров I80286.

1985 г. - Фирма INMOS (Англия) представила транспьютеры на основе специальной архитектуры параллельной обработки информации. Транспьютер - это новый микроэлектронный прибор, объединяющий на одном кристалле мощный микропроцессор, быструю память, интерфейс внешней памяти и каналы ввода-вывода, предназначенные для подключения периферийных устройств. Слово “транспьютер” образовалось от слов транзистор и компьютер, чем фирма INMOS указывает на то, что этот элемент является основой для построения нового поколения компьютеров, точно также как транзистор послужил в свое время базой для создания первых полупроводниковых машин.

1985 г. - с разработкой WINDOWS 1.0 компания MICROSOFT привнесла в DOS-совместимые компьютеры особенности, до того присущие только компьютерам MACINTOSH. INTEL начала выпускать чип 80386 с 32-битовой обработкой и встроенным управлением памятью.

1988 г. - Выпущенная фирмой MOTOROLA серия 88000 32-разрядных RISC микропроцессоров позволила довести скорость обработки до 17 млн. операций в сек. Студент выпускного курса Роберт Моррис, запустив разработанную им вирусную программу в INTERNET, обратил всеобщее внимание на проблему обеспечения безопасности сетей.

1989 г. - INTEL представила чип 80486 на 1,2 млн. транзисторов.

1990 г. - MICROSOFT представлена WINDOWS 3.0, что дало новый импульс юридическому спору с APPLE относительно сходства реализованных принципов “look and feel” с решениями в операционной системе компьютеров MACINTOSH - OS SYSTEM 7. HEWLETT-PACKARD и IBM приступили к производству компьютеров с RISC-архитектурой.

1991 г. - Создан альянс IBM, MOTOROLA, APPLE по разработке микропроцессора POWER PC на основе RISC-архитектуры.

Японское министерство труда и промышленности принимает решение о прекращении программы работы по компьютерам пятого поколения, и приступает к разработке компьютеров шестого поколения на основе нейронных сетей.

Исследования по нейроинформатике начались в Красноярске с открытия В.А.Охониным в 1986 г. принципа двойственности для обучения нейронных сетей, на основе которого фактически решается задача оптимизации в пространстве с очень большой размерностью с возможностью самообучения. Нейронные сети обычно решают задачи за несколько тактов, поэтому решение даже сложных задач обученной сети составляет наносекунды, а это уже новые горизонты технологии.

1993 г. - INTEL представила микропроцессор PENTIUM.

1994 г. - В университете Южной Калифорнии Леонард Адлеман продемонстрировал, что ДНК может быть использовано как вычислительное средство.

1996 г. - INTEL объявила о выпуске процессора PENTIUM PRO.

Это далеко не полный и естественно не окончательный перечень тех эволюционных шагов, которые прошло человечество для решения задач информатизации и создания современных средств автоматизации производства. Все современные типы автоматизированных производств включают в той или иной степени средства программного управления, в построении которых исторически можно выделить следующие принципы автоматизации производственных систем:

- принцип механического счета (Б.Паскаль, Г.Лейбниц, Ч.Бэббидж);
- принцип цифровой обработки информации (А. Тьюринг);
- принцип хранимой программы (Дж. Фон Нейман);
- принцип копировального управления (А.К. Нартов);
- принцип управления с обратной связью (Д.Максвелл);
- принцип программного управления (Д. Жаккард).

1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМАХ ЧПУ.

1.1. ОПРЕДЕЛЕНИЯ И ТЕРМИНОЛОГИЯ.

Системы ЧПУ строятся на базе аппаратных и программных средств, ориентированных на применение микропроцессоров и современных микроЭВМ с периферийным оборудованием, на применении быстродействующего автоматизированного электропривода, обеспечивающего воспроизведение программных траекторий движения, а также системах автоматизированного проектирования, подготовки и отладки системного и прикладного программного обеспечения.

Под **ЧИСЛОВЫМ ПРОГРАММНЫМ УПРАВЛЕНИЕМ** понимается управление с помощью аппаратных и программных средств, обеспечивающее быстрый переход на различные программы работы оборудования путем набора ее или записи условным кодом на программноносителе.

В качестве программоносителей используются перфоленты, магнитные ленты и диски, программируемые постоянные запоминающие устройства, входящие в состав микропроцессорных контроллеров.

Ниже приведены некоторые термины и понятия в области систем числового программного управления, установленные ГОСТ 20523-80.

Позиционное ЧПУ - ЧПУ, при котором перемещение происходит в заданные точки, причем траектории перемещения не задаются.

Контурное ЧПУ - ЧПУ, при котором перемещение происходит по заданной траектории с заданной скоростью для получения необходимого контура обработки изделия.

Групповое ЧПУ - ЧПУ группой объектов от ЭВМ, имеющей общую память для хранения управляющих программ, распределяемых по запросам от объектов.

Программоноситель - носитель данных, на котором записана программа с использованием цифрового кода.

Программное обеспечение системы ЧПУ - совокупность программ и документации для реализации целей и задач систем ЧПУ.

Устройство ЧПУ (УЧПУ) - устройство, выдающее управляющие воздействия на исполнительНнъФ1пфЦ□□ъdq□□ъб□т□u5□
1sЮ□Г□□цЕГЦъeudЪБ□□Се□□□йqdШq□□флЕ□ФГй□□1pdЄпБтЮЛй
□□qГ□фе□□юей□Цл5□□Лб□це?_

Б□□□**БбцШлбdcAǳфЪйёц□п1Тъв1l4**устройство числового управления, алгоритмы работы которого реализуются схемным путем и не могут быть изменены после изготовления устройства.

Программное устройство ЧПУ - устройство ЧПУ, алгоритмы работы которого реализуются с помощью программ, вводимых в память, и могут быть изменены после изготовления устройства.

Система ЧПУ - совокупность функционально взаимосвязанных и взаимодействующих технических и программных средств, обеспечивающих числовое программное управление объектом.

Кадр управляющей программы - составная часть управляющей программы, вводимая и отрабатываемая как единое целое и содержащая не менее одной команды.

Дискретность перемещения - минимальное перемещение, которое задается и контролируется в процессе управления.

Интерполяция - получение координат промежуточных точек траектории движения в плоскости или в пространстве.

Код - ряд правил, посредством которых выполняется преобразование данных из одного вида в другой. Применение кода (кодирование) сводится к записи информации в виде комбинации символов.

Интерфейс - совокупность аппаратных и программных средств, обеспечивающих совместимость функциональных блоков, образующих измерительную, вычислительную или управляющую систему в соответствии с требуемыми условиями, например, видом кода, моментом выдачи (приема) информационных и управляющих сигналов, формой представления информации (аналоговая или цифровая).

1.2.ЦИФРОВЫЕ КОДЫ В СИСТЕМАХ ЧПУ.

При передаче, хранении и преобразовании информации применяют ее кодирование с использованием определенного набора символов. Множество символов составляет алфавит кода. Количество символов в алфавите называют основанием кода - **m**. Так десятичная система счисления является кодом с основанием, равным 10. Код Морзе имеет основание 2, а его алфавит состоит из двух символов: точка и тире. Двоичная система счисления имеет также основание 2, а алфавит состоит из чисел 1 и 0. Восьмеричная система счисления имеет основание 7, и использует для кодирования числа 0,1,2,3,4,5,6,7.

Любую упорядоченную выборку символов из алфавита называют кодовой комбинацией. Число символов в кодовой комбинации называют длиной кодовой комбинации - **n** (разрядностью кода).

КОДОМ называют любое упорядоченное множество кодовых комбинаций.

Число кодовых комбинаций в коде называют мощностью или объемом кода. Максимальная мощность (емкость) кода равна:

$$N = m^n$$

Максимальное число кода $N_{\max} = 2^n - 1$, минимальное $N_{\min} = 1$, динамический диапазон разрядной сетки:

$$D = \frac{N_{\max}}{N_{\min}} = 2^n - 1.$$

Эти же числовые параметры справедливы и для представления чисел со знаком в естественной форме (с фиксированной запятой/ точкой).

Эта форма представления чисел предполагает, что положение запятой, отделяющей целую часть числа от дробной, фиксировано в разрядной сетке (запятая подразумевается, но аппаратно не реализуется).

Для представления знака выделяется специальный знаковый разряд сетки и используется два основных способа расположения фиксированной запятой: перед старшим разрядом для правильных дробей, после младшего разряда для целых дробей. Наибольшее распространение получило представление целых чисел со знаком как целых.

Выбор числовых кодов для представления целых чисел со знаком (биполярных) осуществляется с учетом следующих факторов:

- общее количество положительных и отрицательных чисел не может превышать емкость разрядной сетки;
- целесообразно, чтобы общее количество отрицательных и положительных чисел было одинаково;
- для выполнения основных арифметических и логических операций желательно использовать одинаковую аппаратуру.

Выбор числовых кодов для представления целых чисел со знаком (биполярных) осуществляется с учетом следующих факторов:

- общее количество положительных и отрицательных чисел не может превышать емкость разрядной сетки;
- целесообразно, чтобы общее количество отрицательных и положительных чисел было одинаково;
- для выполнения основных арифметических и логических операций желательно использовать одинаковую аппаратуру.

Среди биполярных кодов различают:

- прямой код со знаком;
- обратный код;

- дополнительный код;
- смещенный код.

Особенности биполярных кодов рассматриваются в курсе “**Элементы систем автоматики**”.

Кодовым расстоянием **d** между двумя комбинациями называют количество несовпадений их разрядов.

Количество единичных символов в комбинации двоичного кода называют ее **весом** и обозначают **l**.

Имеется ряд признаков классификации кодов:

1. По основанию: коды с основанием 2 - двухпозиционные, с основанием **m** больше 2 - многопозиционные.
2. По длине кодовых комбинаций: равномерные при **n = const**, и неравномерные, если **n ≠ const**.
3. По весу комбинаций: коды равновесные, если **l = const** и неравновесные в противном случае.
4. По четности или нечетности веса комбинаций - коды четные и нечетные.
5. По арифметическим свойствам кода: коды арифметические и комбинаторные.

К комбинаторным кодам относят равновесные, четные, нечетные коды, код Грея.

Коды, используемые для представления чисел посредством цифр, определяются как **системы счисления**. Наиболее часто применяются позиционные системы счисления: унитарная, двоичная, восьмеричная, шестнадцатеричная. У последней вместо требуемых цифр 10,11,12,13,14,15 используются буквы латинского алфавита A,B,C,D,E,F. Для позиционных систем характерны 3 важные характеристики:

- количество используемых цифр равно основанию системы счисления;
- наибольшая цифра на единицу меньше основания;
- каждая цифра в числе умножается на основании в степени, значение которой определяется позицией цифры в числе.

Запись произвольного числа **A** базируется на представлении этого числа в виде полинома:

$$A = a_{n-1} \times m^{n-1} + a_{n-2} \times m^{n-2} + \dots + a_1 \times m + a_0; \quad (1.1)$$

где - **a(i)** - весовые коэффициенты (цифры),
m- основание системы (2,8,16),
n- номер разряда.

В унитарной системе счисления имеется только один цифровой

символ - 1. Система используется для записи заданного количества импульсов на магнитных лентах, но является достаточно громоздкой, особенно при записи больших чисел.

Двоичная система счисления является базовой для решения задач автоматики и вычислительной техники, так как на основе аппарата булевой алгебры описывает работу устройств, имеющих два устойчивых состояния 0 и 1.

Приведем некоторые правила перевода чисел из одной системы счисления в другую.

Пример 1. Задано целое число в двоичной форме
1110010011010

Представить его в восьмеричной, десятичной и шестнадцатеричной системе счисления.

Для перевода из двоичной системы счисления в восьмеричную, число нужно разделить на триады влево и вправо от запятой (при записи чисел с запятой). Если самая левая или правая триада окажется неполной, то ее дополняют нулями. Для перевода в шестнадцатеричную систему число разбивают аналогично на группы по четыре цифры.

Для заданного примера получим:

$$1110010011010_2 = 001\ 110\ 010\ 011\ 010 = 16232_8$$

$$1110010011010_2 = 0001\ 1100\ 1001\ 1010 = 1C9A_{16}$$

Для получения заданного числа в десятичной системе счисления используем соотношение (1.1) и получим сумму следующих слагаемых:

$$1 \times 2^{12} + 1 \times 2^{11} + 1 \times 2^{10} + 1 \times 2^7 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 = 7322_{10}$$

Следует обратить внимание, что младший разряд - это число в степени 0, и соответственно в сумму не входят те слагаемые, где весовые коэффициенты равны 0. Проверим правильность вычислений, осуществив перевод из шестнадцатеричной системы в десятичную:

$$1 \times 16^3 + 12 \times 16^2 + 9 \times 16^1 + 10 \times 16^0 = 4096 + 3072 + 144 + 10 = 7322_{10}.$$

Целые числа, записанные в одной системе счисления, в новую систему переводятся последовательным делением числа и получаемых частных на основании той системы, в которую оно переводится, до тех пор, пока не получится частное меньше основания системы счисления. Результатом будут остатки от деления, прочитанные в порядке обратном их получению, и последнее частное как старший разряд числа.

1.3. КОД ISO-7 BIT.

До сих пор одним из носителей цифровой информации для станков с ЧПУ остается перфолента, на которой кодирование информации осуществляется с использованием кода ISO - 7 bit. Вид стандартной перфоленты, используемой для записи программ, шириной 25,4мм представлен на рис.1., в таблице 1 представлено подмножество кода [2].

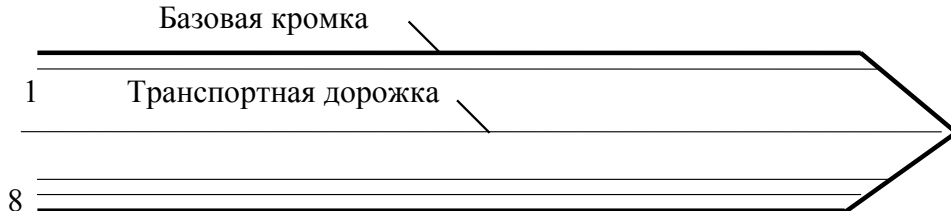


Рис.1.

Кодовые дорожки нумеруются последовательно с первой по восьмую, начиная с базовой кромки - края перфоленты. Между транспортной дорожкой и базовой кромкой расположены три кодовые дорожки. Каждая строка на перфоленте должна содержать лишь одну кодовую комбинацию с соответствующим знаком четности.

Дорожкам 1-4 приписаны веса двоично-десятичного кода 8-4-2-1.

Дорожки 5-6-7 - дорожки признаков. Перфорирование отверстий на дорожках 5 и 6 является признаком десятичных цифр 0-9. Буквы латинского алфавита от А до О, выражаемые комбинациями десятичных цифр от 0 до 15, определяются перфорированием дорожки 7. Признак букв Р-Z определяется перфорированием отверстий на дорожках 5 и 7.

Часть адресов языка ISO обязательна при применении в следующем синтаксическом контексте:

N - номер кадра, G - подготовительная функция, X, Y, Z - размерные перемещения, I, J, K - параметры интерполяции, F - скорость подачи, S - скорость главного движения, T - функция инструмента, M - вспомогательная функция.

Управляющая программа должна начинаться с символа % и заканчиваться кадром, содержащим M2. Каждый кадр начинается с символа N (номер кадра) или: (главный кадр). За номером следуют информационные слова. Рекомендуются следующая последовательность информационных слов: "подготовительная функция"- G; "размерные перемещения "- X,Y,Z,U,V,W,

P,Q,R,I,J,K,A,B,C,D,E; " функция подачи "- F, "скорость главного движения"- S, "номер инструмента"- T, "вспомогательная функция"- M.

Кадр должен заканчиваться символом LF. В пределах одного кадра не должны использоваться слова с одинаковыми адресами (за исключением G и M).

Подготовительная функция G служит для настройки системы управления на определенный режим работы. Значения G задают систему координат (абсолютную или относительную), смещение нуля системы координат, вид интерполяции и т.д. Вспомогательная функция M задает режим работы оборудования и используется для управления электроавтоматикой станка.

Более подробное изучение особенностей программирования с использованием кода ISO-7 bit можно найти в [2].

Таблица 1.

| N | Сим вол | 8 | 7 | 6 | 5 | 4 | T | 3 | 2 | 1 | Код Кои -7 |
|----|------------|---|---|---|---|---|---|---|---|---|------------------|
| 1 | LF | | | | | * | | | * | | 012 |
| 2 | . | | | * | | * | . | * | * | | 056 |
| 3 | % | * | | * | | | . | * | | * | 045 |
| 4 | : | | | * | * | * | | | * | | 072 |
| 5 | 0 | | | * | * | | . | | | | 060 |
| 6 | 1 | * | | * | * | | . | | | * | 061 |
| 7 | 2 | * | | * | * | | . | | * | | 062 |
| 8 | 3 | | | * | * | | . | | * | * | 063 |
| 9 | 4 | * | | * | * | | . | * | | | 064 |
| 10 | 5 | | | * | * | | . | * | | * | 065 |
| 11 | 6 | | | * | * | | . | * | * | | 066 |
| 12 | 7 | * | | * | * | | . | * | * | * | 067 |
| 13 | 8 | * | | * | * | * | . | | | | 070 |
| 14 | 9 | | | * | * | * | . | | | * | 071 |
| 15 | A | | * | | | | . | | | * | 101 |
| 16 | B | | * | | | | . | | * | | 102 |
| 17 | C | * | * | | | | . | | * | * | 103 |
| 18 | D | | * | | | | . | * | | | 104 |
| 19 | E | * | * | | | | . | * | | * | 105 |
| 20 | F | * | * | | | | . | * | * | | 106 |
| 21 | G | | * | | | | . | * | * | * | 107 |

| | | | | | | | | | | | |
|----|-----|---|---|---|---|---|---|---|---|---|-----|
| 22 | H | | * | | | * | . | | | | 110 |
| 23 | I | * | * | | | * | . | | | * | 111 |
| 24 | J | * | * | | | * | . | | * | | 112 |
| 25 | K | | * | | | * | . | | * | * | 113 |
| 26 | L | * | * | | | * | . | * | | | 114 |
| 27 | M | | * | | | * | . | * | | * | 115 |
| 28 | N | | * | | | * | . | * | * | | 116 |
| 29 | O | * | * | | | * | . | * | * | * | 117 |
| 30 | P | | * | | * | | . | | | | 120 |
| 31 | Q | * | * | | * | | . | | | * | 121 |
| 32 | R | * | * | | * | | . | | * | | 122 |
| 33 | S | | * | | * | | . | | * | * | 123 |
| 34 | T | * | * | | * | | . | * | | | 124 |
| 35 | U | | * | | * | | . | * | | * | 125 |
| 36 | V | | * | | * | | . | * | * | | 126 |
| 37 | W | * | * | | * | | . | * | * | * | 127 |
| 38 | X | * | * | | * | * | . | | | | 130 |
| 39 | Y | | * | | * | * | . | | | * | 131 |
| 40 | Z | | * | | * | * | . | | * | | 132 |
| 41 | + | | | * | | * | . | | * | * | 053 |
| 42 | - | | | * | | * | . | * | | * | 055 |
| 43 | (| | | * | | * | . | | | | 050 |
| 44 |) | * | | * | | * | . | | | * | 051 |
| 45 | / | * | | * | | * | . | * | * | * | 057 |
| 46 | nul | | | | | | . | | | | 000 |

* - соответствует пробитому отверстию.

А- поворот вокруг оси X, В- поворот вокруг оси Y,
 С- поворот вокруг оси Z, D- вторая функция инструмента
 Е- вторая функция подачи F- первая функция подачи
 G- подготовительная функция Н- не определена
 I- параметр интерполяции X J- параметр интерполяции Y
 К- параметр интерполяции Z L- обращение к подпрограмме
 М- вспомогательная функция N- номер кадра
 О- не определено Р-третичная длина перемещения по X
 Q- третичная длина перемещения параллельно Y
 R- третичная длина перемещения параллельно Z
 S- функция главного движения Т- первая функция инструмента
 U-вторичная длина перемещения параллельно X

V- вторичная длина перемещения параллельно Y
W- вторичная длина перемещения параллельно Z
X- первичная длина перемещения параллельно X
Y- первичная длина перемещения параллельно Y
Z- первичная длина перемещения параллельно Z
nul - пропуск кадра, + , - направление перемещения, ()- начало и конец комментария, / - пропуск кадра, LF- конец кадра, . - десятичная точка, %- начало программы, : - главный кадр.

1.4. ЗАДАЧИ, РЕШАЕМЫЕ УСТРОЙСТВАМИ ЧПУ.

Задачи, решаемые устройствами ЧПУ можно разделить на 4 класса:

- геометрические,
- логические,
- технологические,
- терминальные.

Геометрическая задача - получение изделия, соответствующего чертежу, управляя приводами подачи станка и используя алгоритмы интерполяции программных траекторий движения.

Логическая задача - управления электроавтоматикой объекта управления, связанная с автоматизацией на станке вспомогательных или циклических операций (зажима - разжима, подвода - отвода, пуска - останова, автоматической смены инструмента и т.д.).

Технологическая задача - присутствует в тех случаях, когда основной рабочий процесс сам становится объектом управления (либо с целью его поддержания, либо с целью оптимизации), например, в гибком автоматизированном производстве.

Терминальная задача - поддерживается устройством ЧПУ при применении компьютера. Диалог с оператором предполагает управление объектом в различных режимах, работы по созданию и отладке программного обеспечения, информационный обмен с ЭВМ верхнего уровня.

Решения отдельных задач и их совокупности может быть осуществлено с применением систем ЧПУ различного типа, сведения о некоторых из которых помещены в последующие разделы пособия.

1.5. КЛАССИФИКАЦИЯ СИСТЕМ ЧПУ.

Системы ЧПУ по степени совершенства, функциональным воз-

возможностям и в соответствии с международной классификацией можно разделить на следующие основные классы:

- 1-системы числового управления Numerical Control (**NC**-типа),
- 2-системы Stored Numerical Control (**SNC**-типа),
- 3-системы Computer Numerical Control (**CNC**- типа),
- 4-системы Direct Numerical Control (**DNC**-типа),
- 5-системы Handled Numerical Control (**HNC**-типа),
- 6-системы Voice Numerical Control (**VNC**-типа).

Системы первого и второго типов в основном работают по программам, которые записываются на перфоленты в коде ISO-7 bit. Внешним признаком является способ считывания и отработки программы. В системах NC - типа принято по кадровое чтение перфоленты, а системы SNC отличаются от первых увеличенным объемом памяти, что позволяет прочитать все кадры программы и поместить информацию в запоминающем устройстве. Перфолента читается только один раз перед обработкой деталей и поэтому мало изнашивается. К системам ЧПУ этих классов относятся устройства типов "Контур 2ПТ ", H22 - для токарных станков; "Контур 3П", H33, H55 - для фрезерных, "Размер 2М", П32 - для координатно-расточных; П111М - для шлифовальных.

Системы DNC - типа предназначены для управления группой станков и другим оборудованием технологического участка, транспортной системой, промышленными роботами от управляющей ЭВМ верхнего уровня. Такие системы строятся с использованием принципов централизованного и децентрализованного управления, а также с использованием локальных сетей.

Централизованная структура характеризуется параллельной передачей информации от одной центральной ЭВМ к отдельным элементам устройства управления. Такие системы обладают большой скоростью передачи информации, но в связи с тем, что сигналы управления подвержены затуханию и влиянию помех централизованные системы имеют ограниченную длину линий связи, требуют использовать дорогостоящие линии связи и обладают низкой живучестью, т.к. выход из строя центральной ЭВМ приводит к остановке всех элементов объекта управления.

Более перспективно использование децентрализованных систем управления иерархического типа. В такой структуре отдельные элементы объекта управления имеют в своем составе системы CNC-типа и функции управления рассредоточены между ЦВМ верхнего уровня и ЦВМ нижнего уровня. В последних решается задача непосредственного управления объектом, а на верхнем уровне производится подготовка и хранение управляющих программ, оптимизация работы локальных подсистем. В такой структуре легче организовать взаимодействие локальных подсистем, связанных через верхний уровень, может быть обеспечена более высокая живучесть за счет возможности автономной работы отдельных подсистем объекта управления.

Системы HNC - типа позволяют осуществлять ручной ввод

программ в память микроЭВМ с пульта устройства ЧПУ. Современные системы этого класса строятся на базе систем CNC - типа с повышенным объемом памяти встроенной микроЭВМ. Подобные устройства позволяют вести программирование с пульта УЧПУ в режиме диалога и при использовании архива стандартных подпрограмм. Эти подпрограммы по команде с пульта вызываются на экран дисплея, на экране высвечивается как графика (схема обработки), так и текст (перечень необходимых данных). В общем случае такие системы позволяют вести подготовку управляющих программ непосредственно у станка по чертежу детали.

Дальнейшим развитием систем HNC являются системы - VNC, у которых управляющая информация может вводиться непосредственно с голоса. Принятая информация затем отображается на экране дисплея, что обеспечивает визуальный контроль за правильностью ввода.

Однако большинство систем ЧПУ построено по типу CNC с применением в канале управления управляющей микроЭВМ. К отечественным системам класса CNC относятся УЧПУ 2У22,2Р22,2С42,2М42 2С85,2Р32,ИЦО-П, "Электроника НЦ80-31".

Эти УЧПУ реализуются в основном на базе микроЭВМ "Электроника - 60" с использованием системной магистрали МПИ и стандартных интерфейсов типа ИРПР и ИРПС. Обозначение УЧПУ может состоять из нескольких элементов: буква означает тип системы П - позиционная, Н - непрерывная (контурная), У - универсальная или контурно-позиционная. Цифра впереди букв означает модификацию системы. Первая цифра за буквой - общее число управляемых по программе координат, вторая цифра за буквой - число одновременно управляемых координат, третья цифра за буквой - тип привода: 1 - шаговый, 2 - следящий.

Например, УЧПУ типа Н 221. Система непрерывная, с двумя управляемыми координатами, одновременно управляемых координат - две, предназначена для управления приводами с шаговыми двигателями.

Устройства 2У22,2Р22 имеют в своем составе микроЭВМ с быстродействием 200 тыс. операций в секунду, интерполяторы, преобразователи 10-2 и 2-10, ОЗУ емкостью 4-8 Кслов, ППЗУ емкостью 16-20 Кслов. Устройства используют в токарных станках с шаговым или следящим приводом.

Устройство 2С42 позволяет использовать до трех каналов управления перемещениями при шаговом электроприводе и до

шести - при следящем. Применяется для управления работой фрезерных и сверлильно-расточных станков.

Устройства 2М42,2С85,2Р32 - обеспечивают управление работой оборудования при контурно-позиционном формообразовании с числом одновременно управляемых движений от двух до пяти. Устройства 2С85 и 2Р32 позволяют вводить программы с перфоленты и вручную, обеспечивают основные функции, характерные для систем CNC-типа и используются для управления фрезерными, токарными и многоцелевыми станками.

Устройство ИЦО-П позволяет вести управление одновременно по десяти координатам, может быть подключено к каналу связи с микроЭВМ верхнего уровня, обеспечивает автоматический контроль ряда параметров технологического процесса и их оптимизацию, обладает возможностью управлять модулем гибкой автоматической линии, состоящим из станка с ЧПУ и промышленного робота, управлять работой станков различных типов и транспортной системой.

К УЧПУ ИЦО-П могут быть подключены периферийные внешние устройства: дисплеи, перфораторы, графопостроители и др.

Модель "Электроника НЦ80-31"("Электроника МС2101") выполнена по блочно-модульному принципу. В зависимости от типа станка и задач управления в состав УЧПУ входят два или три блока. Первый блок - дисплейный, имеет одинаковую аппаратную часть и связан со вторым блоком телеграфным каналом, обеспечивающим скорость передачи информации 19,2 Кбод(1бод=1бит/с). Второй блок УЧПУ предназначен для управления приводами и электроавтоматикой станка, а также для хранения технологического программного обеспечения. В таблице 2 приведены некоторые характеристики различных исполнений УЧПУ "Электроника МС2101."

Таблица 2.

| ТИП УЧПУ | ГРУППА СТАНКОВ | ХАРАКТЕРИСТИКА ВТОРОГО БЛОКА |
|------------|----------------|---|
| МС 2101.1 | ТОКАРНЫЕ | 5 каналов связи с импульсными датчиками, 3 канала управления приводом, 128 дискретных входов, 64 выхода. |
| МС 2101.04 | ШЛИФОВАЛЬНЫЕ | 5 каналов связи с импульсными датчиками, 4 канала управления приводом, 64 дискретных входа и 32 выхода, 4 канала АЦП на 9 разрядов. |

1.6. ОБОБЩЕННАЯ СТРУКТУРА СИСТЕМЫ CNC-ТИПА И ЕЕ ОСОБЕННОСТИ.

Особенностью систем CNC-типа является наличие управляющей микроЭВМ или микропроцессорного контроллера для управления технологическим оборудованием в канале управления. Связь с датчиками управляемых координат, приводами технологической установки осуществляется через стандартные или специализированные устройства сопряжения с объектом (УСО), выпускаемые промышленностью или изготавливаемыми разработчиками самостоятельно.

Обобщенная структура типовой автоматизированной системы управления промышленной установкой CNC - типа представлена на рис.2.

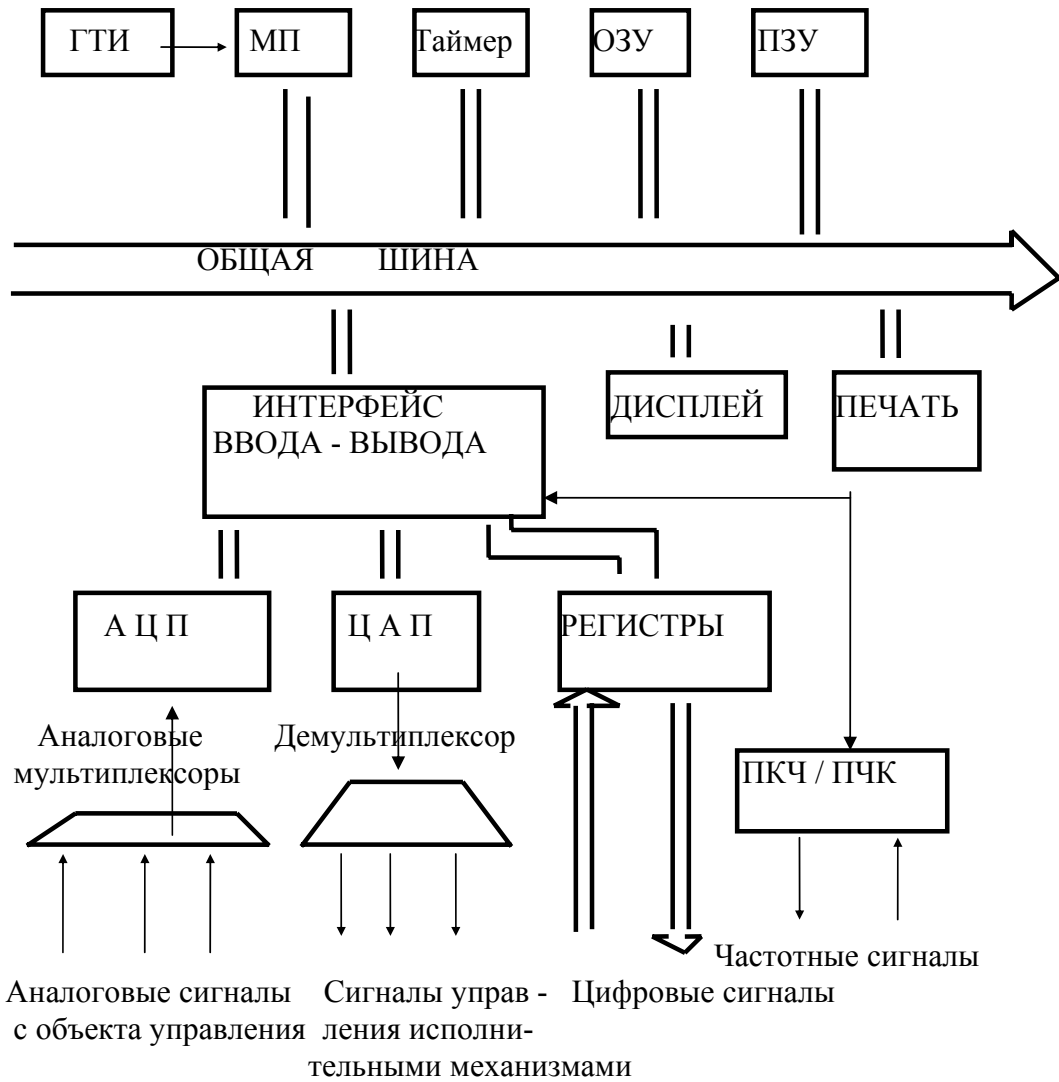


Рис. 2.

В общем случае в состав такой системы входят 4 подсистемы:

- подсистема преобразования и ввода информации с датчиков координат электропривода и состояний объекта управления;
- подсистема преобразования и вывода информации в исполнительные органы технологической установки;
- подсистема обработки информации и внешней памяти;
- подсистема оперативной связи с оператором-технологом.

В состав подсистемы преобразования и ввода информации могут входить: аналоговые мультиплексоры, аналого-цифровые преобразователи, мультиплексоры цифровых сигналов, наборы входных регистров, преобразователи частота-код и др.

В состав подсистемы преобразования и вывода информации входят обычно: цифроаналоговые преобразователи, наборы выходных регистров, демультиплексоры аналоговых сигналов, преобразователи код-частота и др.

В основе обобщенной структуры организации системы ЧПУ CNC-типа лежит **МАГИСТРАЛЬНО-МОДУЛЬНЫЙ** принцип построения, при котором общая шина является той магистралью, через которую осуществляется обмен информацией между всеми устройствами, подключенными к общей шине.

Принцип модульности означает возможность развития системы путем сопряжения требуемых дополнительных устройств в форме отдельных модулей, подключаемых к общей шине посредством контроллеров.

Несмотря на разнообразие задач, которые решает управляющая ЭВМ все процессы, происходящие на системной магистрали, определяются операциями записи, чтения, прерывания и прямого доступа к памяти (рис.3.).

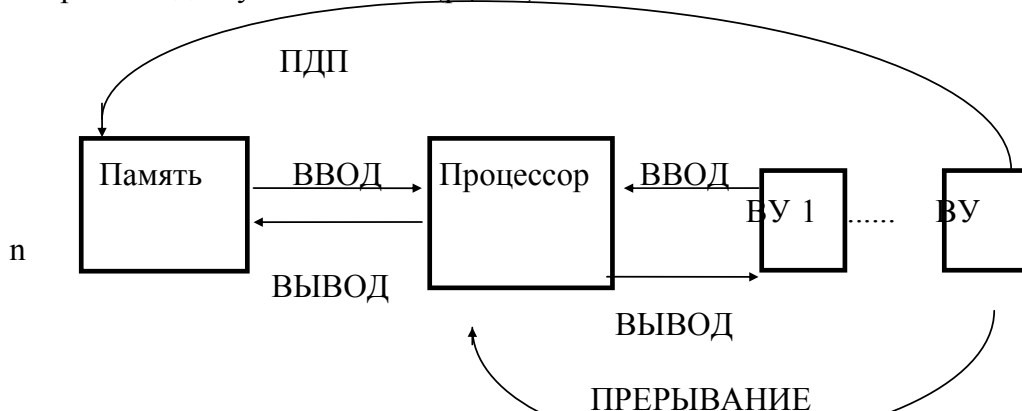


Рис.3.

В процессе взаимодействия любых двух устройств одно из них выполняет активную - управляющую роль, второе является управляемым исполнителем. При обмене информации между устройствами работает принцип **КВИТИРОВАНИЯ** (запроса-ответа), когда каждый управляющий сигнал, посланный задатчиком, подтверждается сигналом исполнителя. Активным устройством в большинстве случаев выступает центральный процессор.

В системах CNC-типа используется три основных способа обмена информацией:

- **программируемый ввод-вывод;**
- **ввод-вывод по прерыванию;**
- **обмен информацией с прямым доступом к памяти (ПДП).**

Использование того или иного способа зависит от быстродействия самого процессора и обслуживаемого им внешнего устройства. В том случае, когда длительность приема и обработки вводимой информации не приводит к ее потере, приемлем программируемый ввод-вывод путем последовательного опроса (поллинга) каналов ввода-вывода по заложенной программе под контролем и по инициативе процессора.

По сигналу синхронизации от процессора на адресные линии поступает информация об адресе конкретного внешнего устройства (порта), который дешифрируется внешним устройством, а затем процессор инициализирует процедуру "Чтение- запись". Запрошенный по указанному адресу порт активизируется и передает информацию на общую шину или принимает ее.

Роль процессора и внешнего устройства меняется при использовании режима ввода-вывода по прерыванию. В этом случае требование обмена информацией выставляется со стороны внешнего устройства в виде сигнала требования прерывания (ТПР), а готовность провести обмен данными подтверждается процессором сигналом предоставления прерывания (ППР). Удовлетворяя запрос на прерывание, процессор завершает выполнение текущей команды, и переходит к выполнению подпрограммы обслуживания прерываний в соответствии с заданным вектором прерывания для данного внешнего устройства.

При этом работает принцип **LIFO** (Last in, first out - последний пришел, первый вышел) или **FIFO** (first in, first out -первый пришел, первый вышел) взаимодействия головной программы и подпрограмм обработки прерываний от внешних устройств через стековую память процессора. Обмен по прерыванию позволяет эффективнее использовать ресурсы процессора, но требует более

развитых аппаратных и программных средств (контроллеры прерываний).

При необходимости обработки больших массивов информации с высоким быстродействием используется принцип обмена с прямым доступом в память, который предполагает исключение процессора из руководства обменом информацией и предоставление этого руководства внешнему устройству, которое захватывает управление общей шиной на себя. Аппаратное исполнение устройств, работающих по этому принципу, самое сложное и предполагает наличие контроллеров ПДП.

2..СОЗДАНИЕ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ НА ПРОИЗВОДСТВЕ.

Объединение всех производственных сфер с целью оптимального и эффективного их использования стало возможным благодаря стремительному развитию систем коммуникации и их стандартизации.

Вычислительные сети подразделяются по своему назначению на сети:

глобальные - **WAN** - Wide Area Network (например, INTERNET);
региональные - **MAN** - Metropolitan Area Network;
локальные - **LAN** - Local Area Netware.

Основными характеристиками сетей являются:

Скорость передачи канала связи.

Единицей измерения является скорость передачи данных в бит/сек. Часто используется - бод - число изменений состояния среды передачи в секунду. Так как каждое изменение состояния может соответствовать нескольким битам данных, то реальная скорость в бит/с, может превышать скорость в бодах.

Пропускная способность сети.

Определяет количество знаков, передаваемых по каналу связи за секунду.

При этом в состав сообщения включаются и все служебные символы.

Теоретически пропускная способность канала определяется

скоростью передачи данных. Реальная пропускная способность зависит от ряда факторов, среди которых и способ передачи, и качество канала связи и структура сообщений. Единица измерения пропускной способности - знак в секунду.

Достоверность передачи.

Характеризует отношение количества ошибочно переданных знаков к общему числу переданных. Единица измерения достоверности - количество ошибок на знак. Для вычислительных сетей этот показатель находится обычно в пределе - 10^{-6} – 10^{-7} .

Надежность сети.

Определяет долю времени исправного состояния, в общем, времени работы. Единица измерения надежности - среднее время безотказной работы в часах.

При создании локальной вычислительной сети (ЛВС) выделяют следующие ее важные характеристики:

- топологию, т.е. способ распределения всех технических средств: процессоров и периферии;
- протокол обмена, т.е. соглашение или правила передачи данных;
- физическую среду передачи, т.е. различные виды кабелей: типа “витая пара”, коаксиальные, многожильные, волоконно-оптические.

Международная организация по стандартизации ISO (International Standards Organization) разработала эталонную модель открытой системы взаимодействий OSI (Open System Interconnection), которая имеет семь уровней обмена информацией [3]:

- физический - **Physical Layer** - управление оборудованием;
- канальный - **DataLink** - для управления линией передачи данных, подразделенный на два подуровня: управления логическим звеном LLC (Logical Link Control) и управления доступом к среде MAC (Media Access Control);
- сетевой - **Network Layer** - межсетевые соединения;
- транспортный - **Transport Layer** - сетевая адресация;
- сеансовый - **Session Layer** - отправка и прием сообщений;
- представление данных, шифрование, сжатие, кодовое преобразование при разном внутреннем представлении - **Presentation Layer**;
- прикладной - **Application Layer**- обслуживание конечного пользователя.

При обмене информацией в сети каждый уровень модели открытой системы реагирует на свой заголовок, т.е. происходит взаимодействие двух одноименных уровней модели взаимодействия в различных абонентских ЭВМ. Правила и последовательность выполнения действий при обмене информацией, определенные протоколом, должны быть реализованы в программе. Обычно функции протоколов различных уровней реализуются в драйверах для различных вычислительных сетей. Так, например, протоколы канального уровня делятся на две основные группы:

- байт-ориентированные (BSC - Binary Synchronous Communication),
- бит-ориентированные (HDLC - High-level Data Link Control).

Работа протокола BSC осуществляется в 3 фазы: установления соединения, поддержание сеанса передачи сообщений, разрыв соединения. Протокол требует на каждый переданный кадр посылки квитанции о результате его приема. Кадры, переданные с ошибкой, передаются повторно. Протокол определяет максимальное число повторных передач. Квитанция представляет собой управляющий кадр, в котором содержится подтверждение приема сообщения (положительная квитанция) или отказ от приема из-за ошибки (отрицательная квитанция). Передача следующего кадра возможна только тогда, когда получена положительная квитанция на прием предыдущего. Это существенно ограничивает быстродействие протокола и предъявляет высокие требования к качеству канала связи.

Бит-ориентированный протокол предусматривает передачу информации в виде потока битов, не разделяемых на байты. Поэтому для разделения кадров используются флаги (старт и стоп-биты.).

HDLC - (высший уровень управления каналом) управляет информационным каналом с помощью специальных управляющих кадров, в которых передаются команды. Он позволяет без получения положительной квитанции передавать в канал до 3-5 кадров.

На каждом из уровней иерархии единицы информации называют по-разному. На физическом - **бит**, на канальном – **пакет** (фрейм), на сетевом-**дейтаграмма**, на транспортном - **сегмент**, на прикладном - **сообщение**.

Важно понимать, что эталонная модель не является чем-то

реальным, таким, что обеспечивает связь. Сама по себе она не заставляет коммуникации функционировать, а служит лишь для классификации. Она классифицирует то, что непосредственно работает, а именно - протоколы.

Канальный уровень формирует из битов, полученных с физического уровня, последовательность пакетов. Здесь осуществляется управление доступом к разделяемой всеми сетевыми устройствами передающей среде и обнаруживается и корректируется часть ошибок. Как и большинство других уровней, канальный добавляет заголовок к передаваемой информации. В заголовке обычно содержится физический адрес приемника, адрес источника и другая информация.

Сетевой уровень заведует движением информации по сетям, состоящим из нескольких сегментов. Для успешного решения этой задачи в протоколах данного уровня вносится информация о логическом адресе источника и адресате пакета. При прохождении пакетов через узлы, соединяющие различные сети, эта информация анализируется, и пакет пересылается к следующему узлу, принадлежащему уже другому сегменту. Таким образом, пакеты путешествуют по сети, переходя от узла к узлу.

Международный институт инженеров по электротехнике и электронике (Institute of Electrical and Electronics Engineers - IEEE) разработал стандарт протоколов передачи данных в ЛВС - IEEE 802.

- ***IEEE802.1** - верхние уровни и административное управление.
- ***IEEE802.2** - управление логическим звеном данных (LLC-Logical Link Control).
- ***IEEE802.3** - случайный метод доступа к среде (CSMA/CD - Carrier Sense Multiple Access with Collission Detection) - множественный доступ с контролем передачи и обнаружением столкновений.
- ***IEEE802.4** - общая (маркерная) шина.
- ***IEEE802.5** - маркерное кольцо.
- ***IEEE802.6** - городские сети.

Метод доступа **ETHERNET** разработан в 1975 г. фирмой XEROX, учредившей корпорацию ETHERNET, вместе с INTEL и DEC. Этот метод используется в топологии “общая шина”, когда к общему кабелю подключены все узлы сети. Он реализует множественный доступ с опросом состояния канала и разрешением конфликтов - CSMA-CD. Стандарт протокола обмена - IEEE 802.3.

Метод доступа **ARCNET** разработан фирмой DATAPOINT CORP. и использует топологию сети - “звезда”, когда каждый компьютер через специальный сетевой адаптер подключается к одному

объединительному устройству. Стандарт протокола обмена - IEEE 802.4.

Метод доступа **TOKEN-RING** (маркерное кольцо) был разработан фирмой IBM и рассчитан на кольцевую топологию сети, когда данные передаются по кольцу. Стандарт протокола обмена - IEEE 802.5.

Доминирующим протоколом обмена данными в ЛВС является протокол **TCP-IP** (Transmission Control Protocol - Internet Protocol), обеспечивающий высокую надежность.

Протокол **MAP**(Manufacturing Automation Protocol) определяет взаимосвязь производственного оборудования.

Для сетей со световодами применяется протокол **FDDI**.

Оптоволокно все шире используется при создании ЛВС(Ethernet, FDDI, Token Ring). В состав схемы волоконно-оптической линии связи (ВОЛС) входят: сетевой адаптер (например, плата Ethernet), устанавливаемый в рабочую станцию или сервер, приемопередатчик (оптический Ethernet 3Com), оптический соединитель (например, SMA), оптический кабель, например, ОКГ-50-2) .

Чрезвычайно высокая частота несущей (10^{14} Гц) обуславливает широкополосность оптических сигналов. Это означает, что по оптической линии связи можно передавать информацию со скоростью 10^{12} бод. Другими словами, по одному волноводу можно обеспечить передачу сразу 10 млн. телефонных разговоров или 10 тыс. телевизионных программ. Физический предел плотности передаваемой информации по оптоволокну сегодня не достигнут. Оптоволокно отличается от других сред передачи очень малым затуханием светового сигнала, что позволяет строить линии связи длиной до 100 км без регенерации сигналов. Волокно изготавливается из кварца, основу которого составляет двуокись кремния SiO₂. Т.к. стеклянные волокна не проводят электричества, то автоматически обеспечивается гальваническая развязка сегментов. Линии связи, использующие оптоволокно устойчивы к электромагнитным помехам, передаваемые по световодам данные защищены от несанкционированного доступа. FDDI - самая динамично развивающаяся в настоящее время технология ЛВС.

Транспортный уровень находится в самом центре эталонной модели. Он отвечает за гарантированную доставку данных, компенсацию ошибок, которые могут возникнуть при работе на нижележащем уровне. Надежная реализация протоколов транспортного уровня обеспечивает подтверждение успеха или неуспеха доставки, информируя вышележащие уровни.

Сеансовый уровень - отвечает за вызов удаленных процедур (RPC - Remote Procedure Calls). Это специально поддерживаемый соответствующими протоколами интерфейс, при котором вызов программной процедуры проводится на одном компьютере, а выполнение на другом, после чего результат возвращается вызвавшей программе, так, словно процедура была выполнена локально. Сеансовый уровень также контролирует установление, течение и завершение сеанса связи между взаимодействующими программами.

Представительный уровень занимается преобразованием формата, упаковкой, распаковкой, шифрованием и дешифрованием, т.е. представляет данные в том виде и формате, какой необходим для последующего и вышележащего уровней.

Прикладной уровень - отвечает за интерфейс с пользователем и взаимодействие прикладных программ, выполняемых на взаимодействующих компьютерах. Представляемые услуги - электронная почта, идентификация пользователей, передача файлов и т.п.

Представим терминологию, сложившуюся и характерную для ЛВС.

Одноранговая ЛВС.

В такой сети нет единого центра и нет единого устройства для хранения данных. Сетевая операционная система (ОС) распределена по всем рабочим станциям. Каждая станция сети может выполнять функции, как клиента, так и сервера. Пользователю сети доступны все устройства, подключенные к другим станциям.

Достоинства одноранговой сети низкая стоимость и высокая надежность.

Недостатки: зависимость эффективной работы сети от количества станций, сложность управления сетью, сложность обеспечения защиты информации, трудность обновления и изменения программного обеспечения станций. Наибольшей популярностью пользуются одноранговые сети на базе сетевых ОС - LANtastic, NETwareLite, WINDOWS 95.

Сеть с выделенным сервером.

В сети с выделенным сервером один из компьютеров выполняет функции хранения данных, предназначенных для использования всеми рабочими станциями, управления взаимодействием между

станциями и ряд сервисных функций. На сервере сети устанавливается сетевая ОС, к нему подключаются внешние устройства - жесткие диски, принтеры, модемы.

Топологическая организация такой сети, как правило, звезда.

Достоинства: надежная система защиты информации, быстроедействие, отсутствие ограничений на число рабочих станций, простота управления по сравнению с одноранговыми сетями.

Недостатки: высокая стоимость из-за выделения одного компьютера на сервер, зависимость быстрогодействия и надежности сети от сервера.

Сети с выделенным сервером являются наиболее распространенными.

Сетевые ОС - LANServer (IBM), WINDOWS NT Server версий 3.51,4.0 и 5.0(Microsoft) и NetWare (Novell).

Метод доступа к передающей среде.

Это метод, обеспечивающий выполнение совокупности правил, по которым узлы сети получают доступ к ресурсу. Существует два основных класса методов доступа: детерминированные (метод опроса (“звезда”) и метод передачи права (“ кольцо “)) и недетерминированные (предусматривает конкуренцию всех узлов сети за право передачи - CSMA/CD). Контроль несущей частоты заключается в том, что узел, желающий передать сообщения, “прослушивает” передающую среду, ожидая ее освобождения. Если среда свободна, узел начинает передачу.

Узлом ЛВС может быть рабочая станция, интеллектуальный (с памятью и процессором) терминал или сервер, вычислительное устройство, обладающее большими ресурсами, которыми может пользоваться любой узел сети.

Стандартная семиуровневая архитектура эталонной модели OSI определяет также и другие сетевые средства.

Повторитель - Repeater - восстанавливает и усиливает сигналы на длинных кабельных соединениях и служит для пространственного расширения сетей, передает информацию на физическом уровне.

Концентратор - Hub - применяют для соединения в одной сети разнородных по своим физическим свойствам проводов,

например, волоконно-оптический с коаксиальным.

Мост - Bridge - служит для объединения двух сетей с различной физической средой или нескольких частей ЛВС на втором уровне модели OSI. Мосты прозрачны для протоколов более высоких уровней, т.е. принимают решение о передаче кадра из одного сегмента в другой только на основании информации из заголовка канального уровня, в частности физического адреса станции - получателя. Кадры, предназначенные для станции того же сегмента, не передаются через мост. В сегментах, соединяемых мостом, могут применяться как одинаковые, так и разные канальные протоколы.

Маршрутизатор - Router - служит для объединения на уровне 3. Он производит протокольную обработку и выбор маршрута доставки информации. Маршрутизаторы могут иметь собственные протоколы, к известным относится протокол RIP (Routing Information Protocol). Современные маршрутизаторы могут маршрутизировать пакеты многих протоколов (TCP-IP, Vines IP, Apple Talk, XNS). Маршрутизаторы выпускаются фирмами 3COM (адаптеры ETHERNET), Nowell, Cisco. Они могут обмениваться информацией друг с другом для сбора данных о топологии и состоянии сети, работоспособности каналов и доступности узлов.

Устройство, сочетающее функции моста и маршрутизатора получило название - Brouter. Такое комбинированное устройство поддерживает определенные типы протоколов и обеспечивает мостовое соединение однородных сетей. Исполняются эти устройства на базе RISC-процессоров.

Для объединения на уровне 4 используется **шлюз** - Gateway, соединяющий сети с разным методом доступа. Например, многоканальный шлюз X.25 может объединить сеть Ethernet с Token Ring.

С помощью шлюзов можно подключить ЛВС к главному компьютеру, а также подключиться к глобальной сети.

В последнее время при создании децентрализованных систем управления иерархического типа и локальных производственных сетей наряду с широко распространенным протоколом RS-232 получил распространение и RS-485. Это простая сетевая технология активного кольца со скоростью реакции 256 запросов - ответов за 1,5 сек при максимальной протяженности до 12 км. Используется широко в модулях фирмы **Advantech**.

Аналоговая техника модемов имеет скорость передачи, которая ограничена величиной 9,6 Кбод, поэтому перспективно использование чисто цифрового способа передачи информации со

скоростью 64 Кбод (система CDI в операционной среде UNIX). В этой среде передача графического файла (изображения) в 3 Мбайта составляет около 2-5 мин., тогда как через модем на это требуется 30-40 мин.

3. ВЫБОР ИНФОРМАЦИОННОЙ СИСТЕМЫ АСУТП.

Максимальная надежность АСУТП достигается с помощью разумного сочетания централизованного и децентрализованного управления при использовании алгоритмов дублирования передаваемых данных.

При этом неисправность линий связи с центром или машины в центре не должно повлиять на автономную работу объекта, неисправность сервера базы данных не должна влиять на автономную работу. В то же время будут предусмотрены возможности централизованного управления и наблюдения за системой и обнаружения нештатных ситуаций. К последним можно отнести: повреждение какого-либо участка сети, неисправность ЭВМ нижнего уровня, попытка несанкционированного доступа к системе, перегрузка сети, дисковой памяти и вычислительных ресурсов.

При построении систем CNC - типа возникает множество вариантов построения информационной системы, отличающихся:

- типом и количеством используемых микроЭВМ;
- способами их соединения (системный интерфейс);
- способами подключения устройств ввода-вывода (прикладной интерфейс, УСО);
- типом операционной системы (системное ПО);
- распределением функций (программ обработки) между отдельными узлами системы;
- составом и объемом устройств памяти;
- способом взаимодействия с оператором - технологом.

Выбор наилучшего варианта сложная системотехническая задача - структурного синтеза на основе методов системного анализа и принятия решений, включающая:

- анализ структуры с целью определения потребности системы в различных ресурсах микроЭВМ, необходимых в процессе ее функционирования (этап функционального анализа);
- анализ структуры с целью определения возможности технических средств, в том числе устройств ввода-вывода (этап технического анализа);
- синтез рациональной технической структуры, обеспечивающий

необходимые для ее функционирования ресурсы. Например, иерархическая структура систем управления автоматически действующих роботов включает несколько уровней управления [4]:

■ **исполнительный и тактический;**

■ **стратегический и высший.**

На **исполнительном** уровне решаются задачи управления приводами манипулятора. При этом ЭВМ должна определять величину рассогласования задающего сигнала и сигнала датчика обратной связи и передавать управляющие воздействия в соответствии с выбранным законом пропорционального, пропорционально - интегрального или другого закона регулирования.

На **тактическом** уровне ЭВМ решает прямую или обратную задачу кинематики или динамики. При этом по заданному линейному и угловому положению схвата манипулятора вычисляют соответствующие значения положений звеньев манипулятора или по известным положениям звеньев определяют положение схвата. На решение обратной задачи расходуется много машинного времени, поэтому она решается в нескольких узловых точках траектории. На тактический уровень возлагается, как правило, задача интерполяции по тому или иному закону между узловыми точками траектории. При решении задачи динамики вычисляются значения сил или моментов, развиваемых конкретными приводами на основе заданного вектора и момента сил в узловой точке захвата.

На **стратегическом** уровне на основе полученного задания (перенести деталь А, например, из определенной точки и вставить ее в деталь В, окрасить поверхность детали по заданной траектории краскопульта) ЭВМ определяет траекторию движения схвата (узловые точки траектории) и требуемые усилия захвата. Если технологическая операция сложная, то на стратегическом уровне разбивают эту операцию на элементарные действия и определяют траекторию движения и необходимые усилия для элементарных операций. Стратегический уровень управления обеспечивает также адаптацию робота к меняющейся обстановке и содержит в своем составе элементы сенсорного очувствления (системы технического зрения, локационные системы обнаружения препятствий и т.д.).

На **высшем** уровне решаются сложные задачи принятия решений и распознавания образов. На этом уровне ЭВМ должна анализировать достаточно сложную обстановку, решать задачу взаимодействия между отдельными локальными подсистемами, передавать задачи управления на стратегический уровень.

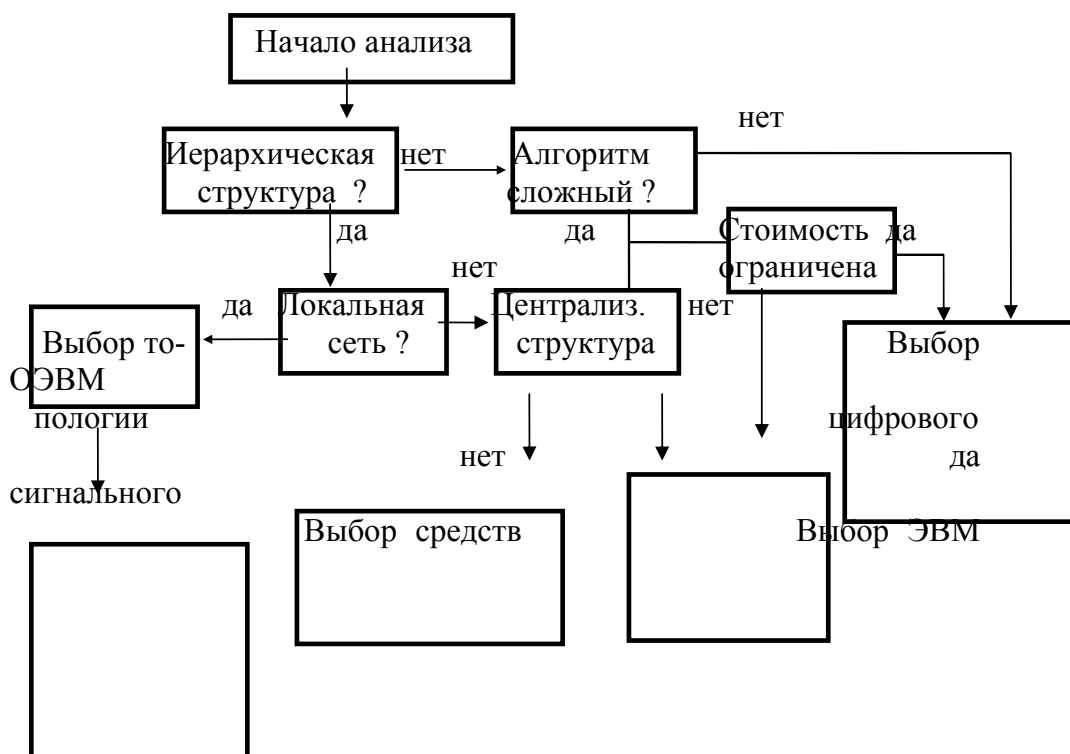
Из приведенного примера ясно, что выбор той или иной структуры

информационной системы зависит от многих факторов, в том числе и от стоимости. При проведении работ по модернизации информационной системы или на этапе ее создания рекомендуется ответить на ряд вопросов:

1. Можно ли в составе новой (модернизированной) системы использовать уже имеющиеся компьютеры?
2. К какому классу относятся имеющиеся компьютеры и выбранные дополнительные ЭВМ? Необходимо ли для их сопряжения и установки дополнительное специоборудование?
3. Стандартная ли используется операционная система, стандартны ли сетевые протоколы? Как будут взаимодействовать машины различных производителей? Можно ли увеличивать вычислительную мощность, объем дисковой памяти, полосу пропускания сети?
4. Как осуществляется поддержка программного обеспечения, какими предположительно будут стоимость и время разработки нового ПО?

Есть ли средства автоматизированной разработки и подготовки программ?

На рис.4. представлен предлагаемый автором пособия алгоритм выбора информационной системы.



процессора,
Выбор аппара-
контролл.
ных и програм-
ных средств
в соответствии
со стандартом
O S I

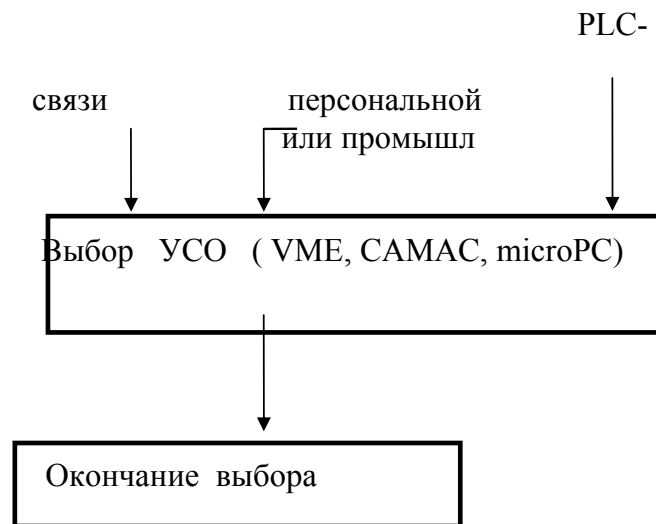


Рис.4.

В заключении заметим, что включение в канал регулирования ЭВМ при управлении технологическим оборудованием, приводит к необходимости учета особенностей процесса квантования сигналов в дискретных системах управления. Этим особенностям посвящена пятая глава учебного пособия.

3.1. ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ

Построение АСУТП на основе концепции открытых систем требует системной интеграции, подразумевающей, что аппаратно - программные средства различных фирм изготовителей совместимы снизу доверху и комплексное подключение и проверка всей АСУТП обеспечивает фирма-интегратор, которая по спецификации Заказчика подбирает все необходимое аппаратное и программное обеспечение.

Как правило, АСУТП представляет собой двухуровневую систему управления. На нижнем уровне программируемые логические контроллеры- PLC, имеющие локальные средства индикации и средства взаимодействия с оператором, на верхнем уровне - достаточно мощные компьютеры, выполняющие функции серверов баз данных, рабочих станций, обеспечивающие хранение и анализ информации, а также ее визуализацию с применением пакетов **SCADA - Supervisory Control and Data Acquisition**).

Смысл открытой модульной архитектуры контроллеров и их требований к контроллерам АСУТП можно сформулировать в терминах:

OPEN - открытая архитектура, обеспечивающая интеграцию широко распространенного на рынке аппаратного и программного обеспечения;

■ **MODULAR** - модульная архитектура, позволяющая использовать компоненты в режиме Plug & P15=j

D

QQFL_y` □бЕшт□е□□Г□Б□Лу□фЕйІФпїц□бyдІЮп□□□лһюJ□ћ
 d□бз□Ю1йдьБЕ□ЮЗй□Ъл1□□□□□юЗjd□лнрЬгЗр□ГйюдепЬ\$л
 □□□БбцШЛБd□кѓфРб□□рЗбШj_ **F-NOMICAL-** экономичная
 архитектура, обеспечивающая невысокую стоимость жизненного
 цикла контроллерного оборудования;

- **MAINTAINABLE** - легко обслуживаемая архитектура, выдерживающая напряженные условия работы в цехах и простая в ремонте и обслуживании.

На нижнем уровне используются PLC-контроллеры, а также все шире внедряемые PC- контроллеры. Специфика условий работы контроллеров предусматривает работу в реальном масштабе времени под управлением ОС реального времени (например, QNX), которая должна быть компактной и иметь возможность запуска из ПЗУ или флэш-памяти.

Промышленный контроллер- это контроллер, предназначенный для применения в условиях промышленного производства, для которого характерны значительные изменения параметров окружающей среды, наличие вредных для эксплуатации факторов: электромагнитные помехи, вибрации , запыленность.

Для связи с объектом управления эти контроллеры имеют несколько аналоговых и / или цифровых входов/выходов, зачастую в них встроены интерпретаторы специализированных языков (например, языка релейной автоматики), они могут быть снабжены возможностью подключения устройств для связи с оператором (Man Machine Interfase).

На нашем рынке производители PLC и промышленных контроллеров представлены такими фирмами как Siemens, GE-Fanuc, ABB, Advantech, Octagon Systems и рядом других.

Первоначально программируемые контроллеры применялись только для замены дискретных систем логического управления, в настоящее время эти функции значительно расширились. Первый программируемый контроллер был создан фирмой DEC в 1969 году, по требованиям фирмы General Motors, которая выдвинула следующие требования к контроллеру [13]:

- простота программирования и перепрограммирования;
- удобство обслуживания и ремонтпригодность;
- меньшие габариты, энергопотребление, более высокая надежность, чем при релейном управлении;

- возможность вывода информации на централизованную систему сбора данных;
- конкурентоспособность по стоимости и сложности со схемами на контактной и полупроводниковой базе;
- входы и выходы должны допускать непосредственное подключение стандартных датчиков сигналов и органов управления;
- управляющие системы должны иметь возможность модификации в рамках минимальной до максимальной конфигурации;
- объем памяти для программ пользователя должен составлять не более 4Кслов.

Эти требования во многом перекликаются с требованиями, сформулированными нами выше. Созданный DEC промышленный контроллер PDP-14 имел входной язык с 9 инструкциями и охватывал функции логических схем, программного управления, таймеров и счетчиков.

При проектировании АСУТП выбор решения между монтажным и программируемым решением неоднозначен и в общем случае сложен.

Помочь сделать правильный выбор могут следующие критерии:

1. Анализ объема решаемой задачи (информационный фактор).

$$I_v = X_d + Y_d + n * (X_c + Y_c) + T + C,$$

где X_d, Y_d - число двоичных входов и выходов,

n - размер слова цифровых данных в битах,

X_c, Y_c - число цифровых входов и выходов,

T - количество реле времени, C - количество счетчиков/таймеров.

В формуле не учтены аналоговые входы / выходы, удельный вес которых для PLC обычно не превышает 5-10 %, в тех случаях, когда применение PLC-целесообразно.

2.Сложность обработки информации.

Сложность операций обработки данных в зависимости от их характера обычно измеряют коэффициентом I_c [13], являющимся суммой относительных сложностей операций обработки:

- сложение, вычитание, сложение слов, выдержка времени, счет - 1;
- умножение и деление - 2,
- перекодирование, вычисление среднего - 3;
- извлечение квадратного корня - 4;
- регулирование с применением модели на основе разностного уравнения -5.

3. Объем требуемой памяти.

Для приблизительной оценки требуемого объема памяти PLC обычно умножают общее число входов и выходов на 10, что дает оценку числа требуемых инструкций в памяти.

В таблице 3 приведена сравнительная оценка при выборе решений по применению того или иного средства автоматизации.

Таблица 3

| Критерии | Монтажная логика | Микропроцессор | PLC | МИНИЭВМ |
|---------------------------------|---------------------|--------------------------------------|---|---|
| Размер системы I_v | $I_v < 100$ | Стоимость и сложность растут с I_v | $20 < I_v < 10000$ | I_v несколько сотен |
| Сложность I_c | $I_c = 1$ | $I_c > 1$ от серии и стоимости | $I_c > 1$ пригоден для несложных вычислений | $I_c > 8$ пригоден для сложных вычислений |
| Надежность в промышленной среде | Высокая | Средняя | Высокая | Достаточно высокая |
| Специализация персонала | Традиционная логика | Электроника и информатика | Традиционная логика и информатика | Информатика |

В таблице 4 представлен обзор некоторых отечественных PLC - контроллеров.

Таблица 4

| Тип PLC | Кол-во входов/выходов | Параметры входов | Параметры выходов | Программирование |
|-----------|-----------------------|---|---------------------|-------------------------------------|
| КПМ-11-03 | 16/16 | = 24, = 110 V | = 24 V | Ассемблер пульт программатора |
| МКП1 | 48 | =24V | =24 V | |
| МикроДАТ | 32, 64, 128 | =12,24 V ~110,220 V | =24 V, ~110 V | РКС, булева алгебра, программатор |
| Б9601 | 256/256 | =5,12,15,24, 110,220 V ~24,110,220V | = 24V ~24, 110 V | РКС, булева алгебра, программатор |

| | | | | |
|--|---------|-------|-------|--|
| | | | | р |
| Ремиконт Р-110,Р112, Р-120,Р-122 | 512/256 | =24 V | =24 V | Имеются аналоговые блоки,библио тека, сервисное оборудо- вание – Димиконт- 120 |

Наиболее развитую среди отечественных PLC номенклатуру имеет МикроДАТ, разработанный НПО САУ (г. Харьков).

Гамма PLC МикроДАТ включает в себя три класса: микроконтроллеры, малые и большие контроллеры. Микроконтроллеры представлены устройствами типа МП 59.01, к малым контроллерам относятся типы - МБ57.0 и МУ 58.0, а большим МУ57.0 и МУ 59.0.

Дополнительный блок связи обеспечивает сопряжение с ЭВМ по каналу ИРПС. Программирование в виде инструкций булевой алгебры. Рабочая программа представляет собой список инструкций, каждая из которых имеет свой номер от 000 до 999.

В таблице 5. представлены некоторые сведения по зарубежным PLC.

Таблица 5

| ТИП | Страна | Разрядность | Кол-во входов/выход ов | Время скани- рования 1Кслов,мс |
|-----------|---------|-------------|------------------------------|--------------------------------------|
| IPC 620 | США | 8 | 4096 | 3 |
| PLC 2/30 | США | 16 | 1792 | 2 |
| SMC 500 | Франция | 16 | 1024 | 4 |
| SIMATICS5 | ФРГ | 16 | 2048 | 5 |

Наибольшее распространение в отечественной промышленности в последнее время получают PLC SIMATIC S5 фирмы SIEMENS. SIMATIC - семейство гибко программируемых PLC, в которое входят контроллеры, программаторы, модули интеллектуальной периферии.

К стандартным системам относятся S5 - 95U, S5 - 115U, S5 - 155U.
В таблице 6 приведены технические характеристики стандартных систем.

Таблица 6

| Характеристика | S5-95 U | S5-115U | S5-155U |
|--------------------------|------------------|-----------------------------|------------------------------------|
| Число процессоров | Однопроцессорный | Однопроцессорный CPU941-945 | Многопроцессорный CPU 922,928,948. |
| Основная память | 16Кб | 256 Кб | 640 Кб |
| Время обработки 1 Кб | 2 mc | 0,8 ms | 0,18 mc |
| Таймеры/счетчики | 128/128 | 256/256 | 256/256 |
| Аналоговые входы /выходы | 32 | 256/256 | 256/256 |
| Цифровые входы/выходы | 448 | 4096/4096 | 8192/8192 |
| Язык программирования | STEP 5 | STEP5 | STEP 5 |

Организация обработки логической информации в PLC.

Для описания процессов в PLC обычно используют языки релейно-контакторных символов - РКС, функциональных блоков и др.

Накопленный многими фирмами опыт был обобщен в виде стандарта Международной электротехнической комиссии (МЭК) IEC 1131-3, в котором определены 5 языков программирования PLC - контроллеров:

1. **SFC** - последовательных функциональных схем.
2. **LD** - релейных диаграмм.
3. **FBD** - функциональных блокковых диаграмм.
4. **ST** - структурированного текста.
5. **IL** - инструкций.

Использование данного стандарта соответствует концепции открытых систем, т.е. делает программу для контроллера независимой от конкретного оборудования - ни от типа процессора, ни от операционной системы, ни от плат ввода/вывода.

В настоящее время программы ряда фирм поддерживают этот стандарт.

Наиболее известна реализация этого стандарта пакет **ISaGRAF** фирмы CJ International. Программа, созданная с помощью этого пакета, может быть загружена и исполнена на процессорах INTEL и MOTOROLA под управлением ОС: DOS, OS-9, QNX, VMEехес, Windows NT и др.

Основное достоинство пакета - простой графический интерфейс, встроенные средства отладки, моделирования, тестирования и документирования, поддержка промышленных сетей.

В ранних разработках программного обеспечения для PLC использовали табличный способ, при котором вектор состояния входных, выходных и внутренних переменных, использовался как адрес для поиска в таблице переходов вектора входных и внутренних переменных на следующем такте работы (**метод конечного автомата**).

Этот способ рационален при небольшом числе дискретных переменных.

Другим способом является обработка информации с использованием РКС и булевой алгебры.

Адресация в контроллерах достаточно разнообразна, но в той или иной степени содержит признаки:

- код признака, указывающего назначение вход или выход;
- номер каркаса (крейта), в котором расположен модуль;
- номер модуля, который определяется местонахождением модуля в крейте (адрес);
- номер входа/ выхода в модуле (субадрес).

К графическим языкам PLC относится язык STEP5 фирмы Siemens, оригинальные проблемно-ориентированные языки высокого уровня Графсет, Ярус-2, Фокон-2 и др.

3.2. ОДНОКРИСТАЛЬНЫЕ МИКРОЭВМ.

Однокристалльные микроЭВМ (ОМЭВМ) представляют собой приборы, конструктивно выполненные в виде одной БИС и включающие в себя все устройства, необходимые для реализации цифровой системы управления минимальной конфигурации: центральный процессор, запоминающие устройства данных и команд, внутренний генератор тактовых импульсов, порты ввода-

Вывода.

Архитектура ОМЭВМ - результат эволюции архитектуры микропроцессоров и микропроцессорных систем, обусловленный стремлением существенно снизить их аппаратные затраты и стоимость. Структурная организация, набор команд и аппаратно-программные средства ввода-вывода ОМЭВМ лучше приспособлены для решения задач управления и регулирования в приборах, устройствах и системах автоматики, но не для решения задач обработки больших массивов данных.

В настоящее время ОМЭВМ практически становятся “монополистами” при построении АСУТП нижнего тактического уровня. Фирмы производители ОМЭВМ: INTEL, SIEMENS, AMD, TI, MOTOROLA, MICROCHIP.

В 1976 г. фирма INTEL выпустила семейство 8-разрядных ОМЭВМ MCS-48, базовой моделью которой являлась микросхема 8048 (отечественные аналоги БИС КР1816 ВЕ 35(48)). ОМЭВМ имеет три 8-разрядных порта ввода - вывода (P0, P1,P2) , общий объем адресуемой памяти 4 К.

В 1981 г. появляется 8 - разрядный MCS -51 (отечественный аналог появился в 1986 году - КР 1816 ВЕ51). MCS-51 обеспечил увеличение объема памяти, содержит четыре 8 - разрядных программируемых канала ввода-вывода, два 16 - битовых счетчика-таймера, систему прерываний, тактовый генератор и последовательный интерфейс. В таблице 7 представлены некоторые сведения по ОМЭВМ типа MCS-51.

Таблица 7

| ФИРМА | SIEMENS | SIEMENS | INTEL | INTEL | РОССИЯ | РОССИЯ |
|-----------------------|---------|--------------|-------|---------|----------------|-----------------|
| ОМЭВМ | SAB8051 | SAB83C5 | 80C51 | 87C51 | КР1816 ВЕ51 | КР1816 ВЕ751 |
| Тактовая частота, МГц | 12,16 | 18 | 12,16 | 16 | 12 | 12 |
| Объем ПЗУ, Кб | 4 | EEPROM 32 | 4 | EPROM 8 | 4 | EPROM 4 |

| | | | | | | |
|--------------------------|------|------|------|------|------|------|
| Объем ОЗУ, байт | 128 | 2 К | 128 | 256 | 128 | 128 |
| Число 8-разрядных портов | 4 | 7 | 4 | 6 | 4 | 4 |
| АЦП | | 12 р | | 8 р | | |
| Таймер/счетчик | 2 | 4 | 2 | 3 | 2 | 2 |
| Число каналов ШИМ | | 21 | | 5 | | |
| Послед. Связь | Есть | Есть | Есть | Есть | Есть | Есть |

В 1985 г. INTEL выпустила MCS-96, а в 1990 г. свой 16-разрядный микроконтроллер представила фирма SIEMENS SAB80C166. В таблице 8 представлена сравнительная техническая характеристика этого прибора и прибора фирмы INTEL 80C196 MC. Все приборы SAB 80C166 работают на частоте 20 МГц. Встроенные блоки периферийного контроллера событий, устройства для измерения и выдачи временных последовательностей и многорежимные таймеры позволяют выполнять операции ввода-вывода, не загружая процессор. Периферийный контроллер событий позволяет любые 8 каналов ввода-вывода настроить на режим ПДП. Система команд контроллера SAB80C166 включает полный набор логических и арифметических операций над 8 и 16 разрядными данными. SIEMENS выпустила комплекс SUPPORT TOOLS для разработки аппаратного и программного обеспечения, который работает под управлением MS-DOS.

Таблица 8

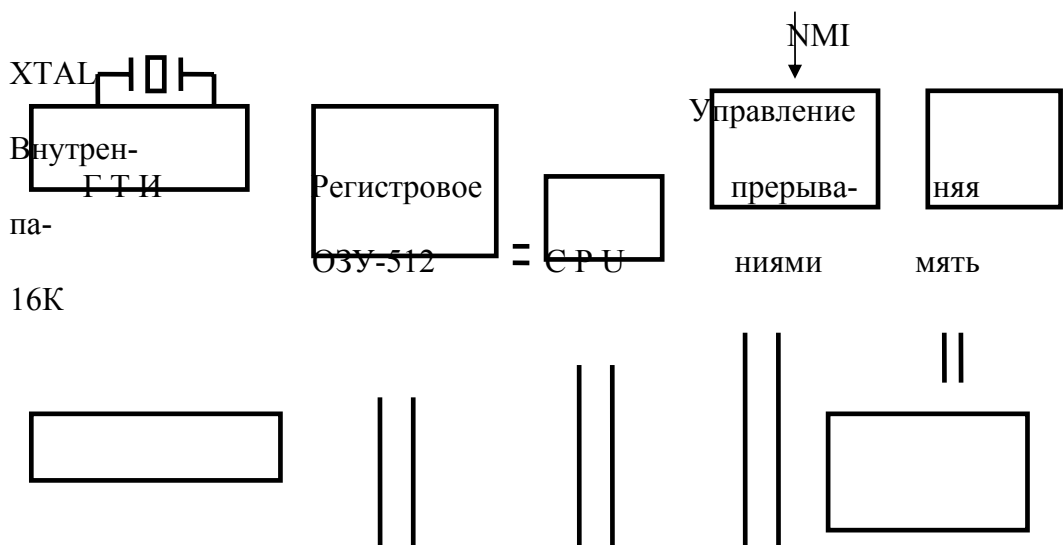
| ФИРМА | SIEMENS | INTEL |
|---------------------------------|-----------|----------|
| ОМЭВМ | SAB80C166 | 80C196MC |
| Тактовая частота, МГц | 20 | 16 |
| Время выполнения инструкций, нс | 100 | 500 |
| Размер адресуемого пространства | 256 К | 128 К |
| Объем ПЗУ | 8К | 16К |

| | | |
|--|---------------|-----------|
| Объем ОЗУ, байт | 1024 | 256 |
| Число таймеров/счетчиков | 5 | 2 |
| Число источников/уровней прерываний | 32/64 | 14/1 |
| Время реакции на прерывание, нс | 400 | 2000 |
| Число каналов ПДП | 8 | нет |
| Число каналов/разрядность/время преобразования АЦП | 10/10/9,75мкс | 8/10/7мкс |
| Сторожевой таймер | есть | есть |
| Число линий В/В | 76 | 53 |
| Число таймеров | 2 | 2 |
| Последовательная связь | есть | есть |

3.3. ОДНОКРИСТАЛЬНЫЙ МИКРОКОНТРОЛЛЕР INTEL 8XC196МС.

Контроллер предназначен для использования при управлении скоростными процессами контроля и управления и с успехом может применяться для управления электроприводами [5].

На рис.5. представлена блок схема контроллера.



8/7

WATCHDOG

Управление 8

шинами

8

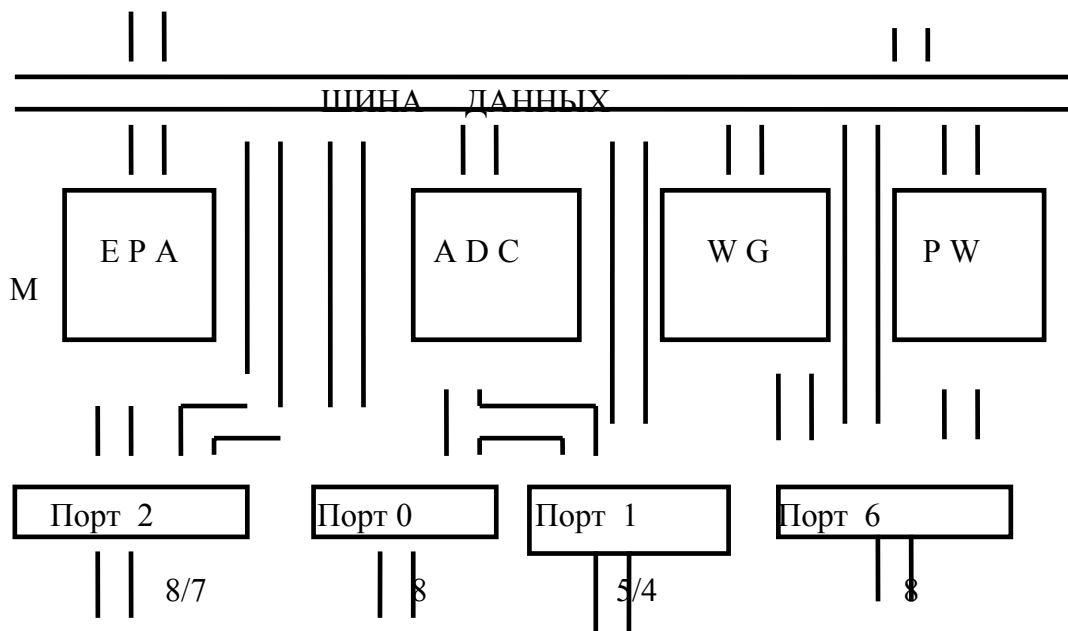


Рис.5.

Его основные компоненты: центральный процессор(CPU), несколько типов памяти, семь портов ввода-вывода, несколько внутрикристалльных периферийных устройств.

В эти устройства входят аналого-цифровой преобразователь, контроллер событий, трехфазный генератор периодических сигналов, широтно-импульсный модулятор, два таймера.

Контроллер событий (ЕРА - Event Processor Array) - реализует функции ввода-вывода, связанные с двумя таймерами. В режиме ввода ЕРА контролирует появление сигнала на клемме ввода и записывает значение времени, когда произошло событие (процедура “захват”). В режиме вывода ЕРА ждет, когда состояние таймера сравнивается с заданным значением времени и затем устанавливает, сбрасывает или переключает состояние выхода (процедура “привязки”). Оба эффекта инициализируют прерывания, которые могут поддерживаться обычными подпрограммами обслуживания прерываний или PTS(периферийный сервер транзакций).

Контроллер 8XC196 MC имеет 4 модуля захвата-привязки и 4 модуля, работающих только на привязку.

Два 16 - битовых таймера могут тактироваться внутренним генератором, а один из таймеров и от внешнего источника. Режим внешнего тактирования может быть эффективен для измерения скорости и положения с использованием инкрементальных датчиков.

Генератор периодических сигналов (WG - Waveform Generator) формирует три пары взаимосвязанных широтно-модулированных сигналов. Это устройство идеально подходит для управления трехфазными двигателями переменного тока. С его помощью можно также управлять бесконтактными двигателями постоянного тока (БДПТ) и силовыми электронными преобразователями. Генератор временных меток и фазоинвертирующая цепь обеспечивают не перекрывающиеся импульсные последовательности для каждой пары выводов, причем каждый сигнал программируется независимо. Программно можно задавать тип сигналов: симметричный относительно середины или начала периода или запускаемый фронтом. Все три фазы имеют одинаковый период и временную задержку. Схема защиты внутри кристалла отключает все 6 выходов в ответ на внешний сигнал EXTINT. Если не использовать его для схемы защиты, он может быть использован как вход внешнего прерывания.

Широтно-импульсный модулятор (PWM - Pulse Width Modulator) обслуживает два выхода. Блок является дополнением к WG. Рабочий цикл и период каждого выхода программируется посредством 8 - битового регистра периодов.

АЦП (ADC - Analog Digital Converter) имеет 13 каналов и может осуществлять 10-битовые или 8-битовые быстрые преобразования. Автоматическое аналого-цифровое преобразование и сохранение результата упрощается при применении PTS. Время выборки - хранения и время преобразования программируются. АЦП может использоваться также как программируемый компаратор и являться источником прерываний при переходе входным сигналом порогового уровня. АЦП содержит 13-канальный мультиплексор, регистр защелку и собственно 10-разрядный АЦП поразрядного уравнивания. Входы АЦП совпадают с входами порта 0 и входами 0-4 порта 1. АЦП поддерживает ряд тестовых режимов: сдвиг и компенсация нуля и др. Шаг приближения 5 mV при 10-

битном и 20-mV при 8-битном преобразовании. Величина опорного напряжения составляет 5,12 Вольт.

Портов ввода-вывода (P0 - P6 - Port) - семь. P1 - 5 - битовый, остальные 8- битовые. P0, P1, P2 управляются с помощью регистра специальных функций(SFR). P0 и P1 используются как входы 13 канального АЦП, а также могут быть прочитаны как логические входы. P2 может быть сконмутирован либо как стандартный порт ввода-вывода, либо для использования специальных функций. P6-выходной порт PWM и WG. Порты 3, 4, 5 адресуются только через 16-битовые адреса. P3 и P4 также используются в качестве 16-битовой внешней мультиплексированной шины адреса - данных. Линии порта 5 могут быть выбраны для обычного ввода-вывода или для использования в качестве выводов управления системы.

Система поддержки прерываний имеет две основные компоненты:

- программируемый контроллер прерываний;
 - периферийный сервер транзакций (блочных пересылок) - PTS.
- (Транзакция - логически законченный набор операций, выполняемый ЭВМ, который может быть обменом с диском, обращением к операционной системе и передачей данных из одной подсистемы в другую).

Контроллер прерываний имеет аппаратную схему приоритетов, которые можно модифицировать программой пользователя. Эти прерывания вызывают написанные пользователем подпрограммы обслуживания прерываний. Прерывания от внутрикристалльных устройств или внешних сигналов обрабатываются в соответствии с рис. 6.



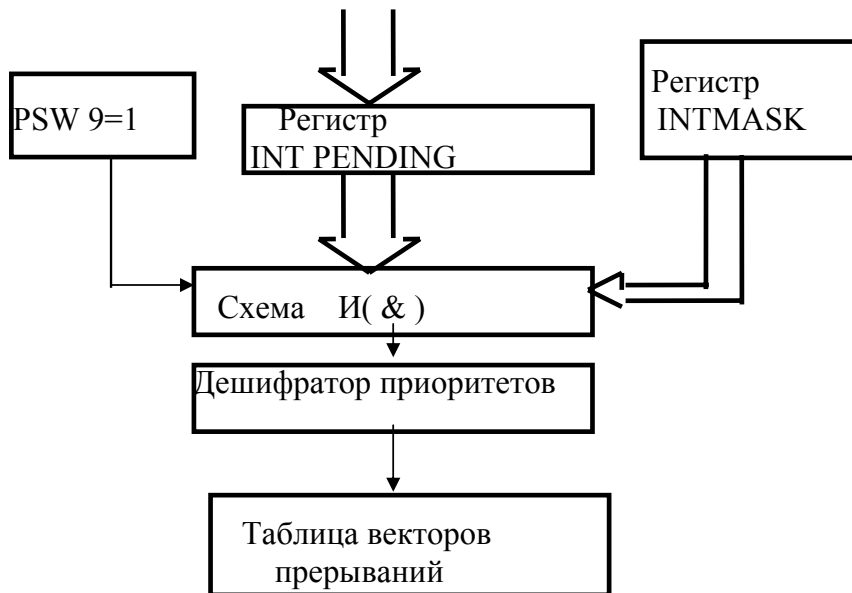


Рис.6.

Детектор перехода идентифицирует сигнал прерывания и устанавливает в регистре (INT PENDING) ожидания прерывания бит, соответствующий этому источнику прерывания. Обслуживание запроса возможно, если установлен бит разрешения в регистре слова состояния процессора (PSW9) и установлен бит в регистре маскирования. Приоритеты прерываний выбираются аппаратно. Немаскируемое прерывание NMI обслуживается иначе. Это внешнее аппаратное прерывание имеет наивысший приоритет и не может быть заблокировано регистром маскирования или PSW9. 8XC196 MC может обрабатывать 16 источников прерывания и имеет 15 векторов прерывания.

Пользователь может большинство прерываний обслуживать с помощью **периферийного сервера транзакций** (PTS-Peripheral Transaction Server), который имеет несколько подпрограмм обслуживания прерываний, поддержанных кодированием на аппаратном уровне, выполнение которых происходит в промежутке между нормальным выполнением команд. В результате реализуется высокоскоростная, прозрачная для пользователя поддержка прерывания. Обработка транзакций чередуется с выполнением обычных команд посредством операции захвата цикла. Основное преимущество PTS по сравнению с обычной обработкой прерываний состоит в меньших временных затратах, операции PTS не вызывают изменения стека и PSW и позволяют продолжать выполнение обычных команд. Другим преимуществом PTS является

его способность работать в одном из следующих режимов:

- пересылка блока байт/слов в памяти;
- управление многоканальным АЦП по выбранным каналам;
- синхронный и асинхронный обмен по последовательному каналу.

Прерывания обслуживаются PTS, если установлен глобальный бит разрешения PTS в PSW (10 бит) и выбран источник прерывания. Все PTS - запросы имеют более высокий приоритет, чем обычные запросы прерываний. Внешние сигналы, обрабатываемые PTS, это все сигналы модулей захвата-привязки и внешнее прерывание EXTINT.

Режим последовательного ввода/вывода имеет два подрежима: асинхронный (ASIO) и синхронный (SSIO) ввод/вывод. С кварцем 16 МГц PTS поддерживает полудуплексный, последовательный ввод-вывод со скоростью 9600 бод (бит/с), загружая процессор не более чем на 4 %. Посылка может быть длиной до 16 бит, включая бит паритета и стоп-бит. ЕРА обеспечивает синхронизацию и обслуживает выводы для реализации линии приема. Для передачи могут использоваться все доступные линии портов вывода.

Сторожевой таймер (Watchdog) - является таймером самоконтроля. Таймер периодически через 65535 тактов будет сбрасывать 8XC196 MC, если программно не установлен в регистре конфигурации соответствующий бит. Чтобы очистить Watchdog, необходимо записать 1EH по адресу 0AH и 0E1H в ячейку 0AH. После очистки схема управления сбросит 8XC196 MC через 65535 тактов. Для нормальной работы системы до этого момента нужно снова успеть очистить таймер. Ненормальная работа программы (зависание и т.п.) почти гарантированно не будет выполнять эту периодическую очистку таймера, что приведет к формированию сигнала системного сброса и восстановит работу системы.

8XC196MC содержит 16 Кс адресами 2000H - 5FFFH, содержащую перепрограммируемую EPROM с ультрафиолетовым стиранием и поддерживает три режима программирования -

- режим автопрограммирования без программатора;
- подчиненный режим, обеспечивающий стандартный режим от программатора;
- динамический режим, позволяющий отдельной области EPROM быть динамически запрограммированной под собственным программным управлением.

3.3.1. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МИКРОКОНТРОЛЛЕРА MCS-96.

Система команд микроконтроллера базируется на системе команд 8096ВН [5]. Она использует различные режимы адресации и включает полный набор арифметических и логических команд для 8-битовых и 16-битовых типов данных. 32-битовые типы данных поддерживаются для результата умножения $16 * 16$, деления $32/16$ и операнда в операции сдвига. Операции с плавающей точкой поддерживаются библиотекой FPAL-96, которая реализует подмножество операций однократной точности из стандарта IEEE.

Существует 6 базовых режимов адресации:

- прямой регистровый;
- косвенный и косвенный с автоматическим инкрементом;
- непосредственный;
- индексный короткий и индексный длинный.

1. Прямой регистровый используется для прямого обращения к 256 регистрам, расположенным на кристалле в младшем регистровом файле и к регистрам специальных функции (адреса 1F00H - 1FDFH).

ADD AX, BX, CX; AX := BX+CX

INCB CL ; CL := CL + 1.

2. Косвенный способ - LD BX, [AX] ; BX := memword (AX).

Если перед вычислением содержимое регистра AX = 2FC2H, а содержимое ячейки 2FC2H= 3F26H, то после выполнения команды содержимое BX = 3F26H.

3. Косвенная адресация с автоинкрементом-

LD BX, [AX] + ; BX= memword (AX); AX:=AX+2.

ADDB AL , BL , [CX] + ; AL:=BL + membyte (CX) , CX:=CX+1.

4. Непосредственная адресация-

ADD AX , # 340 ; AX:=AX+340;

PUSH # 1234H ; SP:= SP-2 , memword (SP):= 1234H.

5. Короткая индексная -

LD AX, 4 [BX]; AX:= memword (BX+4), перед вычислением содержимое BL=52H (младший байт), BH = A1H (старший байт), тогда адрес операнда A152H+04H= A156H.

6. Длинная индексная отличается от предыдущей тем, что из команды берется 16-битовое смещение (индекс) и прибавляется к базовому адресу.

AND AX, BX , TABLE [CX] ; AX:= BX and memword (TABLE +CX).

Существуют и другие методы адресации через нулевой регистр, через стековую адресацию с особенностями которых, а также другими особенностями программирования на языке ассемблера микроконтроллера MCS-96 можно ознакомиться в [5].

К программным средствам поддержки относятся ассемблеры,

компиляторы, моделирующие программы (Simulators) , программы отладки (Debuggers), а также учебный программный комплекс AP-BUILDER (таблица 9).

Таблица 9

| Средство поддержки | Наименование | Фирма изготовитель | Базовая ОС |
|------------------------------|--------------|----------------------|-----------------------|
| Ассемблер | ASM-96 | INTEL | DOS 3.3 и выше |
| Компилятор | iC-96 | INTEL | DOS 3.3 и выше |
| Компилятор | PL/ M-96 | INTEL | DOS 3.3 и выше |
| Симулятор-отладчик | SIM-96 | LEAR COM.COMPAN. | MS DOS 3.0 и выше |
| Учебный программный комплекс | AP BUILDER | DITEL | WINDOWS 3.0 и выше |
| Кросс -ассемблер | 96XASM | LEAR COM. COMPANY | MS-DOS 3.0 и выше |

Ассемблер ASM-96 предназначен для преобразования программы, написанной на входном языке ASM-96, в перемещаемый объектный модуль. Макросредства ассемблера позволяют многократно включать в программу фрагменты, составленные один раз.

В состав ассемблера входят вспомогательные программы (utilities): редактор связей (Linker/ Locator), выполняющий объединение объектных модулей в единую программу с абсолютными адресами; библиотечарь (Librarian), создающий и поддерживающий библиотеку объектных модулей;

библиотека стандартных подпрограмм для арифметических операций с числами в форме с плавающей точкой (Floating-Point Arithmetic Library-FPAL-96);

преобразователь объектных кодов в машинные коды (Object - to hex Converter), которые могут быть загружены в память с использованием обычного программатора.

Компилятор i C- 96 предназначен для преобразования программ, написанных на алгоритмическом языке Си в перемещаемый объектный модуль. Созданный компилятором объектный модуль включается в состав общей программы с использованием вспомогательных программ, входящих в состав ассемблера.

Компилятор PL/ M преобразует программу в перемещаемый объектный модуль с языка PL/ M. Кросс - ассемблер 96XASM преобразует программу, составленную на языке ASM-96 в абсолютный объектный модуль. Полученная программа может быть

исполнена с использованием моделирующей программы отладчика (Simulator-Debugger).

Учебный программный комплекс AP BUILDER позволяет изучать структуру микроконтроллера MCS-96, приобретать навыки формирования управляющих слов для периферийных устройств и вычисления времени выполнения команд.

Заканчивая рассмотрение особенностей микроконтроллера MCS-96, представим сравнительную характеристику некоторых исполнений этих контроллеров. Данные сведены в таблицу 10.

Таблица 10

| Кристалл | 8XC196 KB | 8XC196 KC | 8XC196 JR/JQ | 8XC196 MC | 8XC196 NT |
|------------------------------|--------------|--------------|-----------------|--------------|--------------|
| Каналов АЦП | 8 | 8 | 6 | 13 | 4 |
| Послед. Каналов | 1 | 1 | 1 | 1 | 2 |
| Каналов ШИМ | 1 | 3 | нет | 2+(WG) | нет |
| Наличие PTS | нет | Есть | нет | есть | нет |
| Модули Захвата | 4 | 4 | 6 | 4 | 10 |
| Модули привязки | 6 | 6 | 2 | 4 | 2 |
| Тактовая частота в МГц | 16 | 20 | 16 | 16 | 20 |
| Адресное пространст во | 64К | 64К | 64К | 64К | 1М |
| Цена в долларах США | 10 | 13 | 13 | 15 | 17 |

3.4. ЦИФРОВЫЕ СИГНАЛЬНЫЕ ПРОЦЕССОРЫ

Цифровой сигнальный процессор (DSP - Digital Signal Processor - это специализированный процессор с RISC - архитектурой, предназначенной для решения задач цифровой обработки сигналов (ЦОС)) [6].

ЦОС - это цифровая фильтрация, кодирование и

декодирование, распознавание звука и речи, обработка изображений, спектральный анализ, цифровая звукотехника, медицина, измерительная техника, управляющие системы. Отличительная особенность ЦОС - большой объем вычислений в реальном или близком реальному времени. Система команд процессоров общего назначения (в том числе INTEL) не всегда эффективна для ЦОС. В тоже время, включив в ПК DSP (в качестве параллельно работающего сопроцессора), можно значительно повысить производительность измерительной системы и освободить процессор от несвойственных ему функций. Чем отличается DSP от обычного микропроцессора?

В первую очередь архитектурой и системой команд. В основу построения DSP положены следующие принципы:

- использование гарвардской архитектуры;
- сокращение длительности командного цикла;
- применение конвейеризации и аппаратного умножителя;
- включение в систему команд специальных команд ЦОС.

Гарвардская архитектура - подразумевает хранение программ и данных в двух отдельных ЗУ. Соответственно на кристалле имеются отдельные шины адреса и данных (в некоторых процессорах их несколько). Это позволяет совмещать во времени выборку и исполнение команд.

Конвейерный режим используется для сокращения длительности командного цикла. Обычно применяются двух или трех каскадный конвейер, что позволяет одновременно обрабатывать 2 или 3 инструкции.

Аппаратный умножитель применяется для сокращения времени выполнения одной из основных операций ЦОС - умножения. В процессорах общего назначения эта операция реализуется за несколько тактов сдвига и сложения и занимает значительное время, а в DSP за один командный цикл.

Рынок DSP начал формироваться в начале 80-х годов, когда фирма TI начала выпуск микропроцессора TMS 32010, который оперирует с 16 разрядными словами и способен выполнять 5 млн. операций умножения и сложения в секунду.

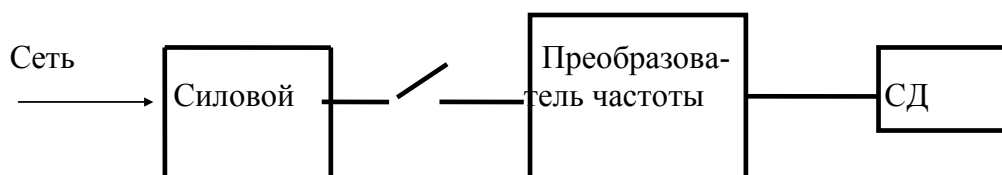
Эти процессоры имели командный цикл 160-280 нс, встроенное ОЗУ 256 слов и ПЗУ программ на 4 Кслов, применяется ППЗУ с ультрафиолетовым стиранием 4К слов, разрядность АЛУ и аккумулятора 32 бит, имеется встроенный умножитель 16*16 бит с 32 разрядным результатом, 8 параллельных 16 разрядных портов, 16- разрядная внешняя шина имеет пропускную способность - 50 Мбит/с.

В настоящее время созданы DSP третьего поколения, выпускаемые фирмами TI, Analog Devices, Hitachi, Philips, Motorola, среди них TMS320C30 (TI), DSP 96002 (Motorola).

TMS 320C30 имеет длительность цикла 60 нс, два блока ОЗУ по 1К 32-разрядных слов с возможностью одновременного доступа, кэш-память команд объемом 64 32-разрядных слова, встроенный 32-разрядный умножитель с плавающей запятой, 40-разрядное АЛУ, 8 регистров для операций с повышенной точностью, встроенный контроллер ПДП, используется 32-разрядная шина команд и данных и 24-разрядная шина адреса, имеется в составе аппаратный ШИМ - модулятор для формирования выходной трехфазной синусоидальной системы напряжений, осуществляется поддержка языков высокого уровня. Специально для этого процессора создан набор инструментальных и программных средств разработки и отладки программ на платформах IBM PC, MACINTOSH, SUN, VAX.

Цифровые сигнальные процессоры могут найти применения в различных системах управления электроприводов и интеллектуальных роботов. В качестве примера можно привести структуры, описанные в журнале “Современные системы автоматизации” - СТА 4 - 1997г. Первая структура предназначена для управления синхронным двигателем, а вторая для построения системы управления интеллектуальным мобильным роботом.

В настоящее время широко внедряются частотно-регулируемые асинхронные и синхронные электроприводы. Силовая часть таких электроприводов включает в себя: регулируемый или нерегулируемый выпрямитель, сглаживающий фильтр, инвертор напряжения или тока, обеспечивающий работу асинхронного или синхронного электродвигателя с переменной скоростью. В АО “НИИэлектроэнергетики” (ВНИИЭ) был разработан и внедрен частотно-регулируемый синхронный электропривод в составе тиристорного пускового устройства синхронного двигателя 20 МВт для Новолипецкого металлургического комбината с микропроцессорной системой управления векторного типа, структурная схема которой представлена на рис.7.[7].



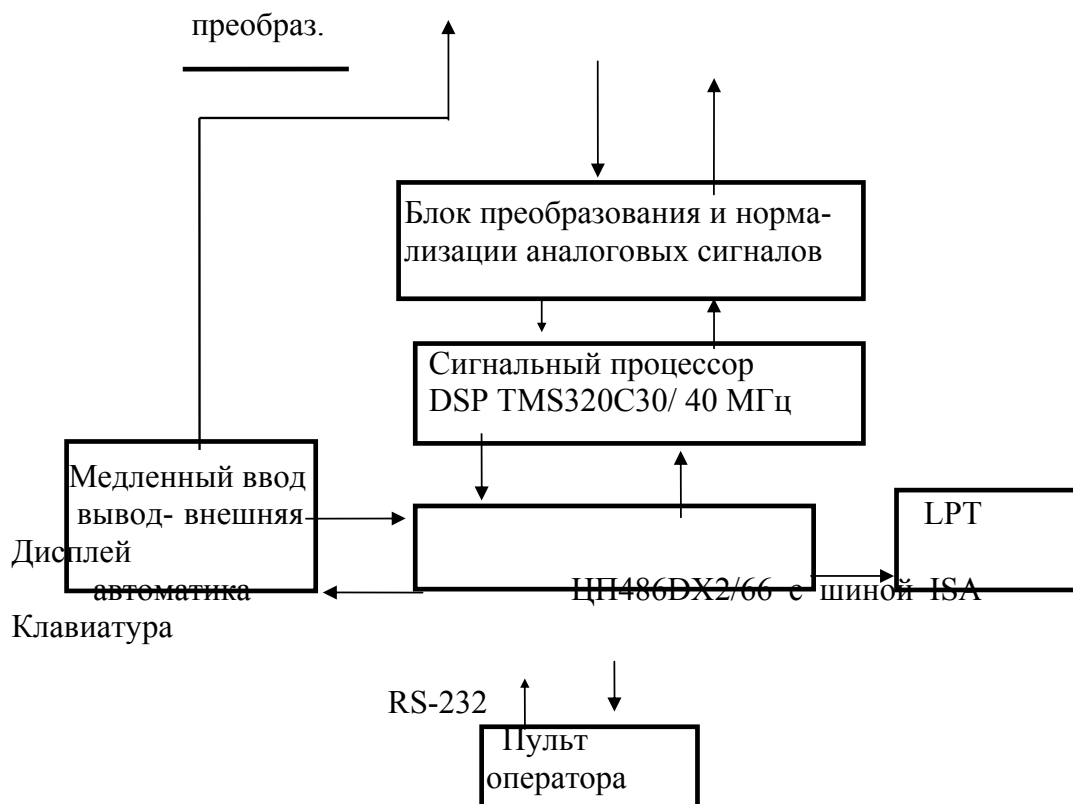


Рис. 7.

Силовая часть электропривода представляет собой зависимый инвертор тока и состоит из двух одинаковых трехфазных мостовых тиристорных выпрямителей, включенных своими шинами постоянного тока встречно - параллельно через сглаживающий реактор. В качестве сигналов обратной связи системы управления измеряются фазные напряжения и токи на входе выпрямителя и выходе инвертора, а также напряжения и токи на входе возбuditеля. Используемая плата "TORNADO" имеет в своем составе DSP TMS320C30 и вставляется в слот шины ISA промышленного компьютера фирмы ADVANTECH.

При построении интеллектуальных мобильных роботов (ИМР) должны решаться сложные алгоритмические задачи, такие как: обработка сенсорной информации (телевизионной, дальнометрической, тепловизионной) и формирование модели видимой зоны среды движения, накопление и корректировка базы знаний робота о среде на основе моделей, формируемых с помощью сенсорных устройств, планирование оптимального движения робота к цели, формирование управляющих воздействий на исполнительные приводы робота, обработка навигационной информации и определение текущих координат положения робота в

среде, а также текущих значений его курса, крена.

ИМП должна обладать быстродействием для решения задач реального времени и типичными требованиями, предъявляемыми к бортовым системам: компактность, надежность, малое энергопотребление. Структурная схема ИМП представлена на рис.8.



Рис. 8.

В состав системы входят 4 подсистемы: подсистема восприятия информации, подсистема планирования движения, навигационная подсистема и исполнительная подсистема. Каждая из этих подсистем представляет собой самостоятельное вычислительное устройство, координация работы которых осуществляется с помощью центрального процессора. В качестве базовых средств УСО используются модули фирмы ADVANTECH. В качестве процессора модуль на базе 486 DX2-66. Все устройства, входящие в

состав системы связаны по шине ISA, для чего используется пассивная объединительная плата фирмы ADVANTECH , рассчитанная на 12 мест .

4. СИСТЕМНЫЕ ИНТЕРФЕЙСЫ УПРАВЛЯЮЩИХ МП-СИСТЕМ.

Любая МП - система, включающая в свой состав микроЭВМ, имеет системный интерфейс, обеспечивающий взаимодействие между отдельными модулями системы.

По конструктивному признаку интерфейсы подразделяются на интерфейсы:

- внутрислатные, обеспечивающие взаимосвязь между БИС на печатной плате или в пределах небольшой группы близко расположенных модулей;
- межплатные (внутриблочные), обеспечивающие взаимодействие на уровне плат и рассчитанные, как правило, на многопроцессорный режим;
- межблочные, обеспечивающие взаимодействие компонентов на уровне прибора, автономного устройства или блока (шкафа) управления.

По способу передачи информации интерфейсы подразделяются на :

- параллельные;
- последовательные,
- параллельно-последовательные.

По принципу обмена информацией и режиму передачи различают интерфейсы:

- синхронные и асинхронные;
- с односторонней (симплекс), с двухсторонней поочередной (полудуплекс), с двухсторонней одновременной передачей данных (дуплекс).

По назначению можно выделить интерфейсы:

- вычислительных микропроцессорных систем (Q-bus, ISA, EISA , Nubus и др.);
- интерфейсы систем сбора и передачи данных (КАМАК, VMEbus);
- интерфейсы интеллектуальных приборов(HP-IB).

Основная функция интерфейсов в обеспечении информационной,

электрической и конструктивной совместимости между функциональными компонентами ПЭВМ, систем и сетей.

В таблице 11 представлена классификация базовых системных интерфейсов ПЭВМ:

Таблица 11

| Шины данных | Интерфейс (фирма) | Тип МП | Примечание |
|---------------|---|---------------------------------------|--------------------|
| 8-разрядная | PC/ XT (IBM) Microbus | I 8088, I8086 I8080, Z80 | Раздельные ША и ШД |
| 16-разрядная | PC/ AT (IBM), ISA | I 80286, I80386 | Раздельные ША и ШД |
| 16-разрядная | Q- bus (DEC) | microVAX, LSI-11, J11 | Совмещенные ША/ШД |
| 16- разрядная | VMEbus (Motorola) | M 68000- M68010 | |
| 32- разрядная | MCA (IBM) EISA (Compaq) VME-bus | I 80386, I80486 M68020, M68030 | |

Рассмотрим эволюцию развития и технические особенности системных интерфейсов фирмы IBM.

Первой системной шиной, разработанной для компьютеров PC/XT была шина PC/XT bus (Personal Computer eXtended Technology - ПК с расширенной технологией).

Она была с 8 разрядной шиной данных (ШД) и 20 разрядной шиной адреса (ША), а ее контроллер обеспечивал работу на частоте 4,77 МГц, имела 4 линии для аппаратных прерываний и 4 канала ПДП (DMA).

С появлением PC/AT была разработана шина PC/AT bus (PC Advanced Technology - с усовершенствованной технологией).

Она имеет 16-разрядную ШД и 24-разрядную ША, имеет частоту 8 МГц, но может использоваться и с МП частотой 16 МГц.

В связи с ростом тактовой частоты МП до 12-16 МГц, контроллер выполнял деление ее пополам для обеспечения приемлемой тактовой частоты работы шины. На базе PC/ XT и PC/AT bus был разработан стандарт ISA (Industry Standard Architecture).

Типовым значением тактовой частоты ISA - 8 МГц. Деление частоты остается функцией контроллеров системной шины, но поскольку

произошло дальнейшее увеличение тактовых частот МП до 25,33,50 МГц коэффициент деления увеличен. Кроме простого увеличения разрядности, совершенствование шин сопровождалось увеличением числа линий прерываний (IRQ) и числа каналов ПДП(DMA). В то же время сохранялась преимущество системных шин, в том числе на уровне контактов разъемов. Для слотов расширения на материнской плате компьютеров с шиной ISA-16 устанавливается стандартная пара разъемов (с числом контактов 36-62), а на шине ISA- 8 - 62 контактный разъем.

Шина имеет 15 линий прерываний, 17 каналов ПДП.

С появлением 32 разрядных МП I80386 (DX) и I80486 попытки совершенствования шины были продолжены разработкой новых стандартов. IBM создала систему машин PS/2 и для них шину MCA (Micro Channel Architecture), отказавшись от концепции совместимости с ISA. Фирмы Compaq, NEC создали компьютеры на базе иной, но тоже 32-разрядной системной шине EISA (Extended ISA - усовершенствованная ISA), достоинством которой является обеспечение совместимости с ISA.

Шина имеет 32 разрядную ШД и ША, адресное пространство составляет 4 Гбайта, пропускная способность 33 Мбайт/с, шина поддерживает многопроцессорную архитектуру.

Погнавшись за увеличением скорости работы системной шины (тактовая частота MCA - 10 МГц), IBM стала использовать для слотов два стандартных 98- контактных разъема, но изменила разводку цепей, введя множество “земляных” линий. Итогом этого стала полная невозможность использовать уже существующие контроллеры периферии, что затормозило развитие компьютеров PS/2.

Что касается стандарта EISA, то его преимущество с ISA, обеспечивается использованием “двухэтажного разъема”, первый этаж которого полностью повторяет разъем шины ISA.

Скорости передачи информации, разрядность данных в современных условиях продолжает возрастать. Преимущество шин в основном обеспечивается, но появляется все большее количество графических средств, и компьютеры стали подолгу задумываться. И хотя к 1991 г. появились специальные ISA - контроллеры с графическими WINDOWS - акселераторами, пробил час так называемых локальных шин.

Локальные шины (Local bus) - шины, электрически выходящие непосредственно на контакты микропроцессора. Она обычно

объединяет процессор, память, схемы буферизации для системной шины и ее контроллер. Локальная шина работает на частоте, равной внешней частоте микропроцессора, и обеспечивает связь с некоторыми скоростными внешними по отношению к микропроцессору устройствами: основной и внешней памятью, видеосистемами и др. Сейчас существует 2 основных стандарта локальных универсальных шин:

- шина VLB;

- шина PCI.

VLB - (Vesa Local Bus - локальная шина VESA - video electronics standards associations) - по существу является расширением внутренней шины МП для связи с видеоадаптером, платами Multimedia, сетевыми адаптерами. Недостатки шины: рассчитана на работу с МП 80386,80486 и не адаптирована для Pentium, Pentium Pro, Power PC; малое количество подключаемых устройств(4), отсутствует арбитраж шины.

PCI - (Peripheral Component Interconnect - соединение внешних устройств) разработана фирмой Intel в 1993 году. Имеет свой адаптер, позволяющий настраиваться на любой МП, имеет арбитраж, средства управления передачей данных. Шина PCI хотя и является локальной, выполняет и многие функции шины расширения, в частности шины типа ISA, EISA, MCA (совместима с ними) при наличии шины PCI подключаются не непосредственно к МП (как это имеет место в VLB), а к самой шине PCI через интерфейс расширения.

К основной шине PCI через разработанные для ПК мостовые микросхемы "PCI-PCI" может подключаться добавочная шина PCI. Шина реализует технологию "Plug and Play", является процессорно-независимой и может работать в ПК не только с Pentium, Pentium Pro фирмы Intel, но и с новейшими RISC - микропроцессорами Power PC фирм IBM и Motorola, ALPHA фирмы DEC и рядом других.

Шина PCI - шина для тех, кто на мощных серверах использует большие дисковые массивы, строит многозадачные комплексы на основе NT и OS/2 или собирает высокопроизводительные рабочие станции для обработки больших объемов графики, видео и данных. Для остальных эта шина дает большие возможности, чем требуется.

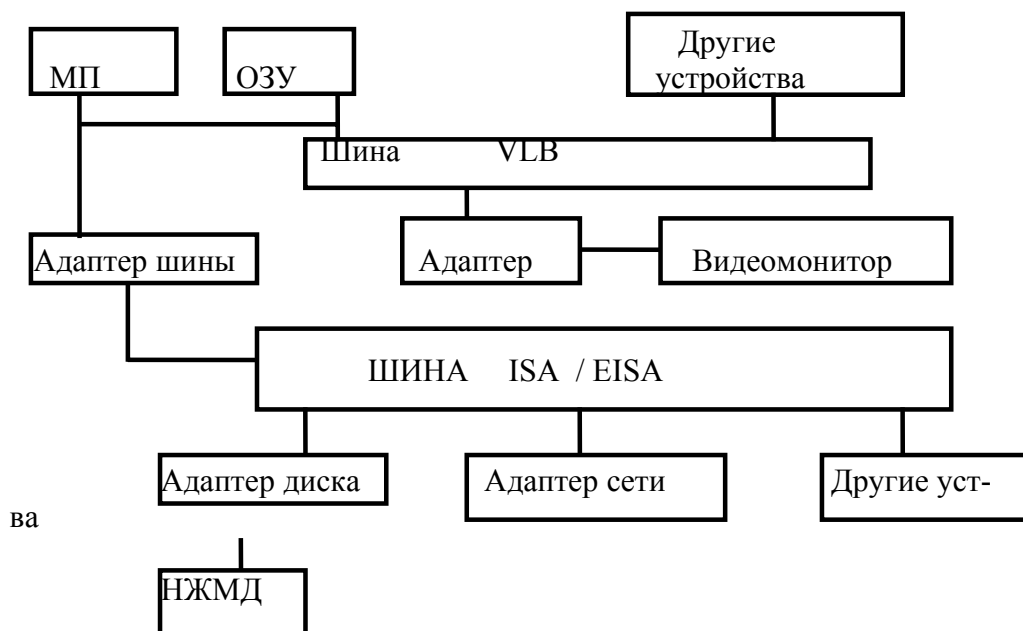
Использование в ПК шин VLB и PCI возможно только при наличии соответствующей VLB и PCI - материнской платы. Выпускаются

материнские платы с мультишинной структурой, позволяющей использовать ISA/EISA, VLB, PCI, так называемые материнские платы с шиной VIP (по начальным буквам VLB, ISA, PCI).

Конфигурация системы с шинами VLB и PCI представлены на рис.9, а в таблице 12 представлены некоторые характеристики рассмотренных шин.

Таблица 12

| Параметр | ISA | EISA | MCA | VLB | PCI |
|--------------------------------|-------|-------|----------|----------|--------------------|
| ШД / ША | 16/24 | 32/32 | 32,64/32 | 32,64/32 | 32,64/32 |
| f, МГц | 8 | 8-33 | 10-20 | до 33 | до 33 |
| V, Мб/с теоретическая/практ | 4/2 | 33/8 | 76/20 | 132/80 | 132,264/50, 100 |
| Числ. Под- кл. уст-в. | 6 | 15 | 15 | 4 | 10 |



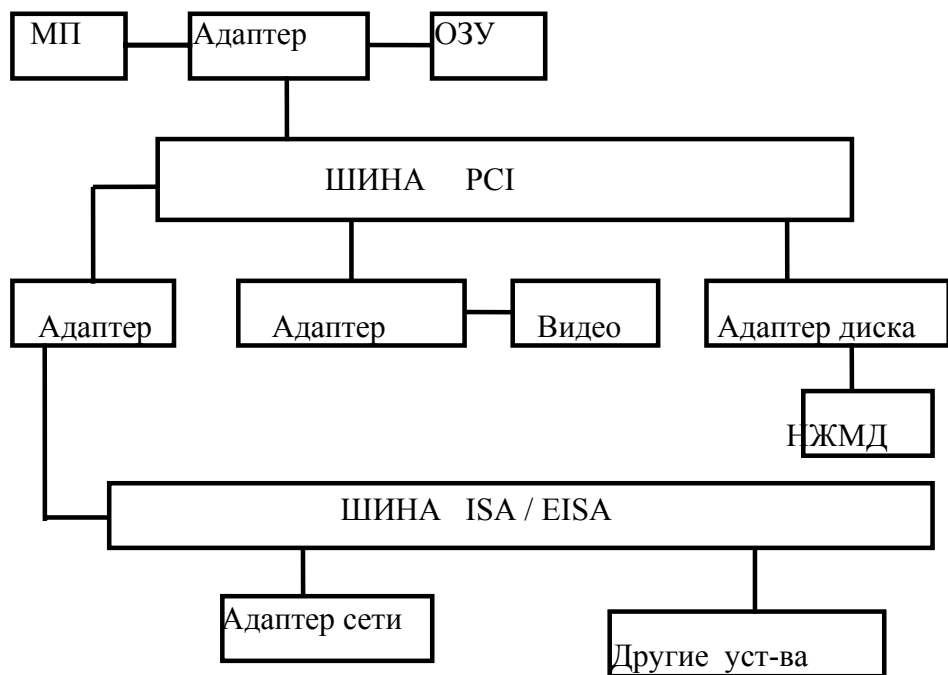


Рис.9.

В настоящее время развивается направление создания нового поколения функционально-модульных управляющих компьютеров (ФМУК), основанных на использовании наборов стандартных, электронных микропроцессорных и микросхемных модулей, подключаемых через соединители к общей стандартной шине.

Все модули имеют идентичную одноплатную конфигурацию и вставляются в общий крейт (кассету) - слот. Число слотов от 4 до 20.

Объединительная (пассивная) печатная плата с соединителями, по которой проходит общая стандартная системная шина, крепится в

задней

части. Стандарт на шину PCI поддерживается консорциумом PCISIG(PCI Special Interest Group), который рассматривает несколько вариантов использования шины PCI в промышленных компьютерах, из которых наиболее радикальными является создание в 1994 году ФМУК Compact PCI фирмы Ziatech Corporation.

Для функциональных модулей Compact PCI выбраны наиболее распространенные типы плат евромеханики с форм-фактором 3U (100*160 mm) , 6U (233,35*160 mm).

Основной системный процессорный модуль устанавливается в системный слот, обеспечивая выполнение централизованного послотового арбитража с использованием одного сигнала на слот, выдачу синхроимпульсов, выполнения сброса в исходное состояние для всех адаптеров периферийных модулей, выходящих на шину Compact PCI. Системная шина дает возможность независимо адресовать в функциональном периферийном модуле до 8 функциональных частей.

Используя архитектуру ПК, Compact PCI могут работать со многими ОС: Windows 95, Windows NT, OS/2, UNIX, QNX, OS-9 и др. Стоимость вычислительных систем, созданных на основе Compact PCI, несколько выше, чем стоимость аналогичных систем на базе ПК. Однако, они дешевле, чем такие же вычислительные системы, созданные на основе традиционных функционально-модульных средств с шиной VME.

В 1996 году многие американские фирмы в том числе DEC, Texas Microsystem, Ziatech приступили к выпуску Compact PCI.

Фирма Ziatech выпустила системный процессорный модуль ZT6500(3U), использующий Pentium с частотой 133 МГц и поддерживающий до 48 Мбайт оперативной памяти и до 4 Мбайт флэш-памяти. Модуль также имеет 24 цифровых канала ввода-вывода, 2 последовательных порта, параллельный порт, контроллер клавиатуры, часы реального времени и сторожевой таймер. Фирма Ziatech выпускает также несколько периферийных модулей Compact PCI, в том числе модуль для сбора видеоинформации (ZT6610).

Фирма Pro-Log выпустила модуль 8831- контроллер ЛВС Ethernet на скорость 100 Мбит/с., модули контроллеров SVGA - дисплея и флоппи-диска емкостью 1,44 Мбайт и накопителя НЖМД 1Гбайт и скоростью передачи данных 16 Мбайт/с.

Фирма "Gespac " выпустила модуль PCIMPU-56, который использует

МП Pentium с внутренней тактовой частотой 90 МГц, оперативную память объемом до 32 Мбайт, контроллер SVGA, контроллер жесткого диска с интерфейсом EIDE, интерфейс флоппи-диска, интерфейс клавиатуры и инфракрасный канал данных. Для локального расширения возможностей системного процессорного модуля на нем расположен соединитель 16-битной шины PC/104, который позволяет подсоединять функциональные модули ввода-вывода с шиной PC/104 непосредственно к модулю PCIMPU56.

Шина PC/104 является адаптацией 16-битной шины ISA, работающей на частоте 8 МГц, и используется в качестве объединительной системной шины в разработанных в конце 80-х годов ФМУК PC/104.

ФМУК PC/104 обладает гораздо меньшей производительностью, чем Compact PCI, и предназначены, в первую очередь, для применений, встраиваемых в оборудование. На рынке имеется довольно большая номенклатура модулей PC/104, которые выпускаются более 50 фирмами, ведущая из которых "Ampro".

Уже первые результаты создания нового поколения ФМУК Compact PCI позволяет сделать вывод, что они окажут существенное влияние на верхний уровень иерархии управляющих вычислительных систем, используемых в различных системах автоматизации процессов и объектов [11].

В последнее время появилась новая PCI платформа для систем автоматики и измерений, получившая название PXI (фирма NI - National Instruments) [12]. С точки зрения конструктива это - промышленный стандарт CompactPCI, который базируется на спецификации Евромеханики (типоразмеры 3 U, 6 U (100* 160 mm, 233,35*160 mm)). Крейт оснащен соединительной платой на 8 PCI модулей с коммуникационной способностью до 132 Мб /с для 32-разрядной 33 МГц PCI - шины и вдвое больше для 64-разрядной. Стандарт полностью совместим с Windows NT/95.

где K - коэффициент передачи импульсного элемента (например, ЦАП),

T_k - период квантования,

p - оператор Лапласа.

ЭВМ можно представить в канале регулирования в виде эквивалентной схемы, содержащей линейный дискретный фильтр, изменяющий закон модуляции входных импульсов в соответствии с необходимым алгоритмом (программой) и последовательно включенный элемент запаздывания, задерживающий результирующие импульсы на временной интервал, необходимый для выполнения алгоритма.

Дискретный характер управления при использовании экстраполятора нулевого порядка в области частот, ограниченных частотой среза САУ, несущественно влияет на вид передаточной функции САУ, и поэтому во многих инженерных приложениях такие дискретные системы можно рассчитывать приемами и методами, предназначенными для непрерывных систем. Многие инженеры просто переносят алгоритмы аналогового управления на цифровой объект, используя малые периоды дискретизации для обеспечения устойчивости системы. Выбор периода квантования T_k в инженерной практике осуществляют на базе теоремы Котельникова:

“ Если непрерывная функция $X(t)$ удовлетворяет условиям Дирихле и ее спектр ограничен частотой f_m , то существует шаг дискретизации T_k с помощью которого можно без погрешности восстановить функцию $X(t)$ по ее дискретным значениям. Этот шаг дискретизации выбирается как

$$T_k < \pi / \omega_m = 1 / f_m.$$

Круговая частота ω_m определяет полосу пропускания дискретной системы. Эта теорема лежит в основе всей импульсной связи и показывает, при каких условиях передача непрерывной функции может быть сведена к передаче ее отдельных значений. В настоящее время рекомендуется рассматривать теорему Котельникова как приближенное утверждение, относящееся к функциям с неограниченным спектром.

В системах ЧПУ решающим фактором при выборе T_k

является требуемая точность воспроизведения запрограммированной траектории и требуемая производительность, т.е. скорость движения по воспроизводимому контуру (геометрическая задача ЧПУ).

Максимальное отклонение от заданной траектории логично ограничить разрешающей способностью датчика положения - s , тогда

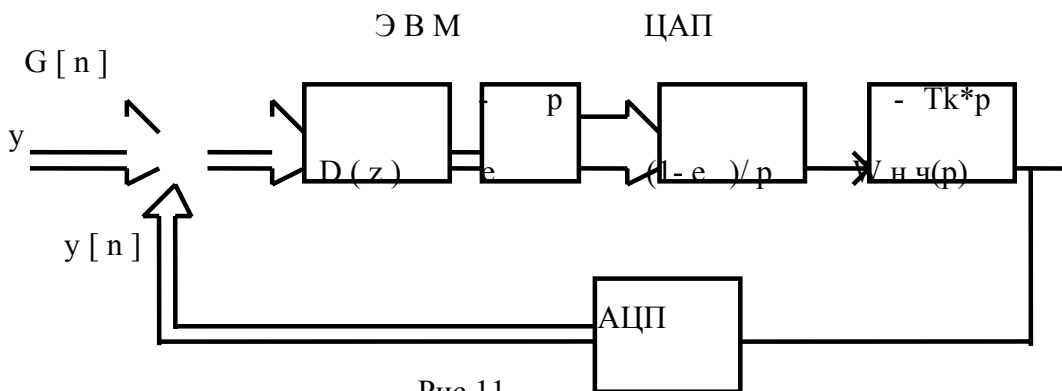
$$T_k < \frac{2}{\pi} (s / (V_k * w_c)),$$

где V_k - контурная скорость обработки, w_c - частота среза.

Современные быстродействующие приводы с транзисторными широтно-импульсными преобразователями (ШИП) и частотой модуляции, превышающей 1, 5 КГц, могут обеспечить воспроизведение линейных траекторий с ошибкой, не превышающей s , если $T_k < 2,5 \text{ мс}$ [8].

Если инженерный подход выбора периода квантования оказывается неэффективным для задачи управления объектом от ЭВМ, следует использовать математический аппарат теории импульсных систем управления, базовые положения которого приводятся ниже.

На рис.11. представлена структурная схема цифровой САУ с ЭВМ в канале регулирования.



5.1. Математическое описание процесса квантования.

Цифровая САУ реагирует на значения воздействия, приложенного к ее входу, и отвечает реакцией только в равноотстоящие друг от друга

дискретные моменты времени. Поэтому воздействие и реакция могут быть заменены **решетчатой функцией**, числовой последовательностью значений некоторой величины, определенных в дискретные моменты времени. Между этими моментами решетчатая функция равна нулю.

Ординаты решетчатой функции называются дискретами, интервал между соседними дискретами - **период квантования T_k** .

Можно считать, что с выхода импульсного элемента поступает последовательность d - функций. Каждая величина этой последовательности имеет вес, равный значению квантуемого сигнала в момент квантования $t = n \cdot T_k$, где n - натуральный ряд чисел (1, 2, ..., n).

$$X(t) \rightarrow d(t) \times X(t) \rightarrow X^*(t) \rightarrow X[nT_k].$$

ЭВМ преобразует дискретную последовательность входа в дискретную последовательность выхода, указанное преобразование осуществляется в соответствии с определенной программой работы ЭВМ и сводится к решению разностного уравнения.

Соотношения, связывающие решетчатую функцию, и ее разности различных порядков, называются уравнениями в конечных разностях или **разностными уравнениями**.

Решетчатую функцию, соответствующую $f(t)$ в дальнейшем будем обозначать $f[n]$. Решетчатая функция физически нереализуема, но служит удобной формой представления последовательности реальных импульсов. Введение идеального квантователя, формирующего d -импульсы, позволяет упростить математический аппарат, используемый для описания процессов прохождения дискретных сигналов через линейные динамические звенья [9].

Таким образом, для математического описания импульсных систем во временной области используется аппарат теории разностных уравнений, которые имеют для импульсных систем то же значение, какое дифференциальные уравнения имеют для непрерывных систем.

Скорость изменения решетчатой функции (производная) характеризуется разностью ее последних значений. Приведем соотношения для прямых и обратных разностей:

$$\Delta f[n] = f[n+1] - f[n] - \text{прямая разность 1 порядка,}$$

$$\nabla f[n] = f[n-1] - f[n] - \text{обратная разность 1 порядка,}$$

$$\Delta^2 f[n] = \Delta f[n+1] - \Delta f[n] = f[n+2] - f[n+1] - f[n+1] + f[n] = f[n+2] - 2 * f[n+1] + f[n] - \text{прямая разность 2 порядка,}$$

$$\nabla^2 f[n] = \nabla f[n-1] - \nabla f[n] = f[n-2] - f[n-1] - f[n-1] + f[n] = f[n-2] - 2 * f[n-1] + f[n] - \text{обратная разность 2 порядка.}$$

Разности более высоких порядков получаются аналогично.

Пример. Записать разности 1 и 2 порядка для функции



Решетчатая функция линейной непрерывной функции:

$$f[n] = a \times n \times Tk = a1 \times n,$$

где $a1 = a \times T k$.

Разности первого порядка:

$$\Delta f[n] = f[n+1] - f[n] = a \times (n+1) - a \times n = a$$

$$\nabla f[n] = f[n-1] - f[n] = a \times (n-1) - a \times n = -a$$

Разности второго порядка:

$$\Delta^2 f[n] = a \times (n+2) - 2 \times a \times (n+1) + a \times n = 0$$

$$\nabla^2 f[n] = a \times (n-2) - 2 \times a \times (n-1) + a \times n = 0$$

Сумма решетчатых функций играет роль аналогичную роли интеграла в непрерывном анализе.

Линейное дифференциальное уравнение, как известно, имеет вид:

$$f(t) = b_m \times d^m y(t) / dt + \dots + b_1 \times dy(t) / dt + b_0 \times y(t)$$

где b_i - коэффициенты дифференциального уравнения.

Линейное разностное уравнение с постоянными коэффициентами можно представить в следующем виде:

$$f[n] = b_m \times \Delta^m y[n] + b_{m-1} \times \Delta^{m-1} y[n] + \dots + b_1 \times \Delta y[n] + b_0 \times y[n]$$

Порядок высшей разности определяет порядок разностного уравнения.

Заменяя разности решетчатых функций их выражениями в соответствии с вышеприведенными правилами можно получить более удобную форму записи:

$$f[n] = a_m \times y[n+m] + a_{m-1} \times y[n+m-1] + \dots + a_0 \times y[n],$$

которая позволяет по заданным значениям $y[0], y[1], \dots, y[n+m-1]$

вычислять

у $[n+m]$ последовательно при $n=0,1,2,\dots$

Численные методы решения разностных уравнений рассматриваются в [9].

Одним из математических методов, разработанных для анализа и синтеза дискретных систем, является метод **Z-преобразования**.

Аппарат Z-преобразования играет для цифровых систем ту же роль, что преобразование Лапласа для непрерывных систем.

Пусть выходной сигнал идеального квантователя $f^*(t)$ определен соотношением:

$$f^*(t) = \sum_{n=0}^{\infty} f[nTk] \times d(t - nTk) = f^*(t) \times d(t),$$

где d - дельта - функция.

Изображение по Лапласу будет иметь вид :

$$L\{f^*(t)\} = F^*(s) = \sum_{n=0}^{\infty} f(nTk) \times \exp(-nTk).$$

Преобразуя иррациональную функцию $F^*(s)$ в рациональную $F(z)$, посредством замены комплексной переменной s на другую комплексную переменную $z = \exp(Tk \times s)$ получаем Z-преобразование функции в виде:

$$F^*[s = (1/Tk) \times \ln z] = F(z) = \sum_{n=0}^{\infty} f[nTk] \times z^{-n}$$

Поскольку Z-преобразование $f(t)$ получается из преобразования Лапласа для функции $f^*(t)$ заменой $z = \exp(Tk \times s)$, то в общем случае для любой функции, имеющей преобразование Лапласа, существует также и Z-преобразование.

Приведем некоторые полезные свойства Z-преобразования

1. Линейность.

$$Z\{a \times x_1[nTk] + b \times x_2[nTk]\} = a \times Z\{x_1[nTk]\} + b \times Z\{x_2[nTk]\}.$$

2. Сдвиг по времени вправо.

$$Z\{x[nTk] - d \times Tk\} = z^{-d} \times x(z) \\ d > 0$$

3. Сдвиг по времени влево .

$$Z\{x[nTk] + d \times Tk\} = z^d \times (x(z) - \sum_{q=0}^{d-1} x[q \times Tk] \times z^{-q})$$

4. Изменение масштаба по переменной Z.

$$Z\{x(n) \times \exp(-i \times nTk)\} = x(z \times \exp(i \times Tk)).$$

5. Начальное значение.

$$X(+0) = \lim_{z \rightarrow \infty} X(z)$$

6. Конечное значение.

$$\lim_{n \rightarrow \infty} x[nTk] = \lim_{z \rightarrow 1} ((z-1)/z) \times x(z) = \lim_{z \rightarrow 1} (z-1) \times x(z)$$

7. В отличие от преобразования Лапласа, для которого прямой и обратный переходы $f(t) \leftrightarrow f(s)$ выполняются однозначно, Z-преобразование этим свойством не обладает. Объясняется это тем, что оно не учитывает поведение функции $f(t)$ в промежутках между моментами срабатывания квантователя. На практике обратное Z-преобразование вычисляют, записывая функции $x(z)$ как сумму элементарных членов, содержащихся в таблицах Z-преобразований или просто поделив числитель $X(z)$ на его знаменатель.

Полученный ряд:

$$X(z) = C_0 + C_1 \times z^{-1} + C_2 \times z^{-2} + \dots,$$

где $x(0) = C_0$, $x(1) = C_1$, $x(2) = C_2$ и т.д. и будет обратным Z-преобразованием.

Оператор Z рассматривается не только как комплексная переменная Z-преобразования, но и как оператор временного сдвига по оси дискретного времени,

$$\text{т.е.: } Z^k \times Y[n] = Y[n+k].$$

Имея передаточную функцию дискретной САУ в виде Z-преобразования, можно перейти к разностному уравнению.

Пример. Задана передаточная функция инерционного звена

$$W(z) = Y(z) / x(z) = K \times z^{-1} \times (1-a) / (1-z^{-1} \times a),$$

где $a = -Tk/T$, T - постоянная времени инерционного звена.

Переход к разностному уравнению осуществляется следующим образом:

$$Y(z) \times (1-z^{-1} \times a) = K \times z^{-1} \times (1-a) \times x(z)$$

$$Y[n] - Y[n-1] \times a = K \times (1-a) \times x[n-1],$$

$$Y[n] = Y[n-1] \times a + x[n-1] \times K \times (1-a).$$

Последнее выражение и будет разностным уравнением для инерционного звена. Такое решение будет математически точным, т.е. значения, полученные с помощью уравнения, совпадают с решением уравнения в дискретные моменты времени, кратные периоду квантования.

5.2.Нахождение Z-преобразований.

Процедура нахождения Z -преобразований непрерывной функции включает следующие 3 этапа :

- определение решетчатой функции $f^*(t)$ как выходного сигнала идеального квантователя для входной функции $f(t)$,
- определение дискретного преобразования Лапласа $f^*(t)$,
- замена $\exp(Tk \times s)$ на z и получение выражения в виде бесконечного ряда.

Однако неудобство полученного выражения очевидно.

Альтернативным выражением для нахождения Z - преобразования функции является выражение полученное при использовании теоремы вычетов.

Если известно преобразование Лапласа $F(S)$, заданное в виде $F(s) = N(s) / D(s)$ и $F(S)$ имеет конечное число простых полюсов k , то Z- преобразование функции находится в виде:

$$F(z) = \sum_{n=1}^k \frac{N(s_k)}{D'(s_k)} \times \frac{1}{1 - \exp(s_k \times T) \times z^{-1}} \quad (*)$$

где s_k - простой полюс, $D'(s_k) = \frac{d D(s)}{d s} \bigg|_{s=s_k}$

Если $F(S)$ имеет кратные полюсы s_1, s_2, \dots, s_k с кратностью $m_1 \dots m_k$, то Z- преобразование находится с использованием соотношения:

$$F(z) = \sum_{n=1}^k \sum_{i=1}^{m_n} \frac{(-1)^{m_n-i}}{(m_n-i)!} \times K_{ni} \times \left(\frac{d^{m_n-i}}{ds^{m_n-i}} \times \frac{1}{1 - \exp(-T \times s)} \right)$$

при подстановке $z = \exp(T \times S)$, где K_{ni} определяется по формуле:

$$K_{ni} = 1 / (i-1)! \{ d^{i-1} / ds^{i-1} [(s - s_n)^{m_n} \times F(s)] \}$$

при подстановке $S = s_n$ после проведения операции

дифференцирования.

Пример1. Найдем Z-преобразование функции

$$F(t) = \exp(-x \times t).$$

Изображение по Лапласу экспоненциальной функции:

$$F(s) = 1 / (s + x),$$

где $s_1 = -x$ (простой полюс).

Тогда Z-преобразование находится с использованием выражения (*)

$$F(z) = \sum_{k=1}^n \frac{N(s)}{D^l(s)} \times \frac{1}{1 - \exp(s_1 \times Tk) \times z^{-1}} =$$

$$= 1 / (1 - \exp(-x \times Tk) \times z^{-1}) = z / (z - \exp(-x \times Tk)),$$

т.к. $D^l(s) = 1$.

5.3. ПЕРЕДАТОЧНЫЕ ФУНКЦИИ ИМПУЛЬСНЫХ СИСТЕМ.

Импульсные и линеаризованные цифровые системы в простейшем виде могут быть представлены в виде последовательного соединения импульсного элемента (ИЭ), экстраполятора или формирующей цепи с передаточной функцией $H(p)$ и непрерывной части с передаточной функцией $W(p)$.

Импульсным фильтром называется устройство, которое получает входные сигналы и одновременно выдает выходные сигналы в определенные дискретные моменты времени $t = n \times Tk$.

Z-преобразование входной и выходной величины импульсного фильтра связаны между собой его передаточной функцией:

$$Y(z) = W(z) \times X(z)$$

Реакция импульсного фильтра (непрерывной части совместно с экстраполятором) на входное воздействие в виде единичного импульса, Z-преобразование которого равно 1, называется приведенной весовой функцией фильтра, а передаточная функция является Z-преобразованием от приведенной весовой функции импульсного фильтра:

$$W(z) = \sum_{n=0}^{\infty} w(n) \times z^{-n} = (b_0 + b_1 \times z^{-1} + \dots + b_m \times z^{-m}) / (a_0 + a_1 \times z^{-1} + \dots + a_m \times z^{-m})$$

В теории импульсных систем различают, по аналогии с непрерывными, передаточные функции разомкнутой и замкнутой системы, передаточные функции по ошибке [9].

Последнее выражение можно представить в виде разностного уравнения, связывающего две решетчатые последовательности $X[n]$ и $Y[n]$.

Если объект управления является статическим, коэффициент усиления импульсной системы можно определить, воспользовавшись теоремой о конечном значении:

$$K = \lim_{z \rightarrow 1} W(z) = (b_0 + b_1 + \dots + b_m) / (1 + a_1 + a_2 + \dots + a_m)$$

Астатическая система первого порядка имеет полюс $Z = 1$ и имеет передаточную функцию:

$$W(z) = \frac{z}{z-1} \times \frac{b_0 + b_1 \times z^{-1} + \dots + b_m \times z^{-m}}{1 + a_1 \times z^{-1} + \dots + a_m \times z^{-(m-1)}}$$

Системы с запаздыванием согласно теореме о сдвиге влево имеют передаточную функцию:

$$F(z) = W(z) \times z^{-d}, \quad d = 1, 2, 3, \dots$$

5.4. СИНТЕЗ ЦИФРОВЫХ СИСТЕМ.

Под синтезом цифровых систем (электроприводов) понимается инженерная задача, заключающаяся в выборе типа и расчете параметров **цифрового корректирующего устройства (ЦКУ)**, обеспечивающего требуемые динамические свойства проектируемого электропривода.

В процессе синтеза определяются передаточные функции отдельных звеньев системы, вычисляются общие передаточные функции разомкнутой и замкнутой системы и осуществляется переход к Z -преобразованиям и разностным уравнениям для дальнейшего составления программного обеспечения системы электропривода.

Таким образом, синтез импульсной системы включает последовательное решение задач синтеза структуры системы и нахождения ЦКУ.

На рис.12. представлена структурная схема простейшей импульсной

системы с ЦКУ, имеющим передаточную функцию $D(z)$.

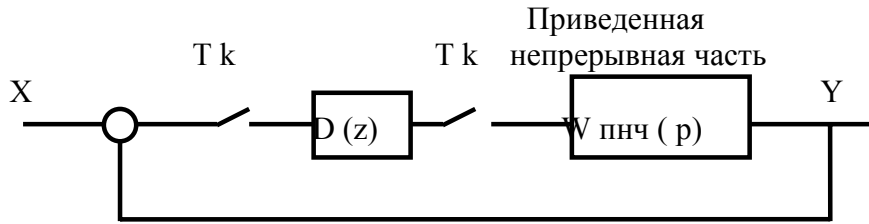


Рис.12.

При синтезе передаточная функция замкнутой желаемой системы имеет вид:

$$F(z) = D(z) \times W(z) / (1 + D(z) \times W(z)),$$

где $D(z)$, $W(z)$ - передаточные функции регулятора и приведенной непрерывной части (экстраполятор + непрерывная часть).

Передаточная функция регулятора определяется как:

$$D(z) = 1 / W(z) \times F(z) / (1 - F(z)).$$

При выборе передаточной функции руководствуются требованиями физической реализуемости и грубости системы, принимая во внимание, что степень числителя передаточной функции меньше или равна степени ее знаменателя, нули и полюса $D(z)$ должны быть по модулю меньше 1.

В качестве непрерывной части рассмотрим двигатель постоянного тока с безинерционным управляемым преобразователем и экстраполятором первого порядка. Тогда передаточная функция приведенной непрерывной части будет иметь вид :

$$W_{\text{пнч}}(z) = Z \left\{ \frac{1 - e^{p \times T_k}}{p} \times \frac{K}{T_e \times T_m \times p^2 + T_m \times p + 1} \right\} = \frac{a1 \times z + a2}{z^2 + b1 \times z + b2}.$$

В последней формуле для вычисления коэффициентов используются следующие формулы в зависимости от типа полюсов передаточной функции.

1. Если выполняется условие $Tm \leq 4 \times Te$, то непрерывная часть является колебательным звеном и

$$T = \sqrt{Tm \times Te}, \quad v = \sqrt{(Tm / Te) / 4},$$

а коэффициенты могут быть найдены следующим образом:

$$a1 = K \times (1 - \exp(u \times Tk) \times \cos(w \times Tk) + u / w \times \exp(u \times Tk) \times \sin(w \times Tk));$$

$$a2 = K \times (\exp(2 \times u \times Tk) - u / w \times \exp(u \times Tk) \times \sin(w \times Tk) - \exp(u \times Tk) \times \cos(w \times Tk));$$

$$w = 1 / T(\sqrt{1 - v^2}); \quad b1 = -2 \times \exp(u \times Tk) \times \cos(w \times Tk); \quad b2 = \exp(2 \times u \times Tk);$$

$$u = -v \times T;$$

2. Если выполняется условие $Tm > 4 \times Te$, то непрерывная часть является аperiодическим звеном второго порядка, для которого

$$T1, T2 = 1 / 2(Tm \pm \sqrt{(Tm \times (Tm - 4Te))})$$

$$a1 = K \times (1 - \exp(-Tk / T2) \times (T2 / (T2 - T1)) + \exp(-Tk / T1) \times (T1 / (T2 - T1)))$$

$$a2 = K \times (\exp(-Tk / T1) \times \exp(-Tk / T2) + \exp(-Tk / T2) \times (T1 / (T2 - T1)) - \exp(-Tk / T1) \times (T2 / (T2 - T1)));$$

$$b1 = -(\exp(-Tk / T1) + \exp(-Tk / T2)); \quad b2 = \exp(-Tk / T1) \times \exp(-Tk / T2).$$

Для желаемого аperiодического переходного процесса с конечной длительностью желаемая передаточная функция замкнутой системы имеет

$$\text{вид: } F(z) = (1 - a) / (z - a),$$

где параметр a определяет быстродействие системы.

Тогда передаточная функция ЦКУ может быть найдена в виде:

$$D(z) = 1 / W(z) \times F(z) / (1 - F(z)).$$

$$D(z) = \frac{(1 - a) \times (1 + b1 \times z^{-1} + b2 \times z^{-2})}{(a1 + a2 \times z^{-1}) \times (1 - z^{-1})}.$$

Полученная передаточная функция является желаемой передаточной функцией ЦКУ.

Обычно функции, реализуемые ЦКУ, задаются разностными уравнениями, либо передаточными функциями вида $D(z)$.

Рассмотрим ряд типовых линейных ЦКУ.

1. Интегрирующие звенья первого порядка.

$$D(z) = T_k \times Z\{1/T \times p\} = T_k / T \times (z / (z - 1)) = T_k / T \times (1 - 1/z)^{-1}$$

Соответствующее разностное уравнение для реализации И-регулятора:

$$U_{и} [n T_k] = U_{и} [(n-1) T_k] + T_k/T_{и} \times e [n T_k],$$

где e - сигнал ошибки между заданным и реальным значением на входе И-регулятора, $U_{и}$ - сигнал на его выходе.

Следует отметить, что интегрирующим свойством обладает любое звено с передаточной функцией:

$$\frac{T_k \times (1 + a \times (1/z))}{(1 + a) \times (1 - (1/z))} = D(z)$$

Ниже приведен фрагмент программы реализации интегрального регулятора для DEC - архитектурной машины (например, типа “Электроника-60” [10]).

Пусть сигнал задания $U_{зс}$ поступает в ячейку 2300, сигнал обратной связи $U_{ос}$ в регистр R5, коэффициент интегратора $K_{и} = T_k / T_{и}$ хранится в ячейке 2302, значение напряжения интегрирования на предыдущем такте

$U_{и} [(n-1) \times T_k]$ в ячейке 2304.

```
IREG :  MOV  @# 2300, R0; пересылка  $U_{зс}$  в R0
        SUB   R5 , R0 ; получение сигнала ошибки
        MOV  @# 2302, R1; коэффициент интегратора в R1
        JSR   R7 , MULT ; вызов подпрограммы умножения
        ADD  @# 2304, R2; получение в R2  $U_{и} [n T_k]$ 
        BVS   M1; переход при отсутствии арифметического
        TST   R2 ; проверка знака при переполнении
        BPL   M2;
        MOV  # 100000, @# 2304 ; запись максимального
```

отрицательного

; по модулю числа для 16-разрядных машин

BR M1

M2: MOV # 77777, @#2304; ограничение наибольшим по модулю

; положительным числом для 16-разрядной машины

M1: RTS R7 ; выход из подпрограммы И- регулятора.

2. Дифференциальный регулятор.

Выражение для цифровой реализации Д-регулятора может быть получено из передаточной функции для И - регулятора, если учесть, что

$$\frac{1}{p} = \frac{z}{z-1}, \quad p = (1 - z^{-1}).$$

Разностное уравнение для Д - регулятора записывается в виде:

$$Y[nT_k] = K_d \times [X[nT_k] - X[(n-1)T_k]].$$

3. Пропорционально-интегральный регулятор (ПИ).

Передаточная функция такого регулятора соответствует параллельному включению пропорционального и интегрального регуляторов:

$$U_{\text{пи}}(z) = U_p(z) + U_i(z) = K_p \times e(z) + U_i[nT_k] \times z^{-1} + K_i \times e(z).$$

Разностное уравнение ПИ- регулятора :

$$U_{\text{пи}}[nT_k] = K_p \times e[nT_k] + U_i[(n-1)T_k] + K_i \times e[nT_k].$$

5.5. АНАЛИЗ ЦИФРОВЫХ САУ.

1. Расчет динамических процессов в цифровых САУ.

Аналитический расчет переходной функции включает выполнение следующих действий:

а) нахождение Z-преобразования выходной координаты

$$Y(z) = W_{z,c}(z) \times X(z),$$

где $X(z)$ - изображение входного сигнала,

$W_{z,c}(z)$ - передаточная функция замкнутой системы;

б) нахождение обратного Z - преобразования для получения выражения выходной величины во временной области:

$$Z^{-1}[Y(z)] = y[nT_k]$$

Данный алгоритм применяется только в относительно простых случаях, поскольку вычисления прямого и обратного Z-преобразования требует больших аналитических затрат. На практике предпочтительнее оказываются методы, основанные на аппроксимации дифференциальных операторов разностными.

Пусть структурная схема анализируемой системы состоит из звеньев, представленных на рис.13.

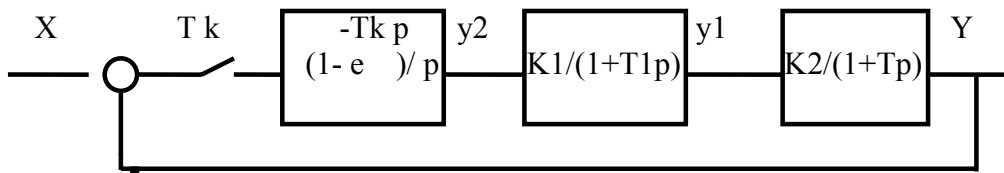


Рис. 13.

Запишем дифференциальные уравнения для звеньев:

$$W1(p) = K1 / (T1 \times p + 1) = Y1 / Y2$$

$$W2(p) = K2 / (T2 \times p + 1) = Y / Y1$$

$$Y1 \times (T1 \times p + 1) = K1 \times Y2;$$

$$T2 \times p \times Y + Y = K2 \times Y1;$$

$$p \times Y1 = -Y1 / T1 + K1 \times Y2 / T1;$$

$$p \times Y = -Y / T2 + K2 \times Y1 / T2.$$

Дифференциальное уравнение для переменной Y2 имеет простой вид:
 $p Y2 = 0$.

Последнее уравнение справедливо и для моментов времени между моментами квантования, поскольку в эти моменты времени сигнал на выходе фиксатора нулевого порядка постоянен, а изменение его происходит скачкообразно только при срабатывании импульсного элемента.

Полученные дифференциальные уравнения описывают динамику импульсной системы при ее движении между моментами квантования. В моменты квантования следует дополнить эти уравнения уравнением:

$$Y2[nTk] = X[(n-1) \times Tk] - Y[(n-1) \times Tk],$$

которое описывает изменение переменных при срабатывании

импульсного элемента.
Преобразование дифференциального уравнения в разностное осуществляется подстановкой:

$$p = 2 / T_k \times (z - 1) / (z + 1)$$

$$p = 1 / T_k \times (z - 1) / z$$

Первая подстановка основана на применении метода трапеций, а вторая на методе прямоугольника [9].

Полная разностная модель динамики линейной системы для первого случая будет иметь вид:

$$Y2[n] = X[n-1] - Y[n-1],$$

$$Y1 = Y1[n-1] + T_k / (2 \times T1) \times (-Y1[n] - Y1[n-1] + K1 \times Y2[n] + K1 \times Y2[n-1]),$$

$$Y2 = Y[n-1] + T_k / (2 \times T2) \times (-Y[n] - Y[n-1] + K2 \times Y1[n] + K2 \times Y1[n-1])$$

Последняя система уравнений решается с применением итерационных методов [9,10].

В указанной литературе можно ознакомиться и с другими методами построения переходных процессов и оценками качества регулирования в импульсных системах.

5.6. УСТОЙЧИВОСТЬ ЦИФРОВЫХ САУ.

Устойчивость замкнутой цифровой системы определяется видом корней характеристического уравнения. На p -плоскости корни устойчивой системы должны лежать в левой полуплоскости, что является необходимым и достаточным условием устойчивости непрерывной САУ.

Переход к комплексной переменной $Z = \exp(p \times T_k)$ отображает левую полуплоскость во внутреннюю часть круга единичного радиуса с центром в начале координат Z - плоскости. Поэтому в устойчивой системе корни характеристического уравнения, найденные из уравнения:

$$1 + W(z) = 0$$

должны лежать внутри круга единичного радиуса, т.е. по модулю должны быть меньше или равны 1.

Использование известных из теории непрерывных САУ критериев устойчивости в данном случае затруднительно.

Целесообразно от Z -преобразования перейти к **W- преобразованию**, используя соотношения:

$$w = (z - 1) / (z + 1)$$

$$z = (1 + w) / (1 - w).$$

Эта операция отображает внутреннюю часть круга единичного радиуса на левую половину **W** - плоскости, что позволяет использовать известные алгебраические критерии устойчивости для характеристического уравнения:

$$1 + W\{(1 + w) / (1 - w)\} = 0.$$

При переходе к частотной передаточной функции $W(\exp(j \times w \times t))$ или $W(j \times wf)$ подстановкой $w = j \times wf \times T / 2$ могут использоваться частотные критерии устойчивости (wf - псевдочастота).

Применение критерия Гурвица для анализа устойчивости.

Характеристическое уравнение импульсной системы имеет вид:

$$a(0) \times z^n + a(1) \times z^{n-1} + \dots + a(n-1) \times z + a(n) = 0.$$

С помощью замены переменной $z = (w + 1) / (w - 1)$ получаем новое уравнение:

$$A(0) \times w^n + A(1) \times w^{n-1} + \dots + A(n) = 0.$$

Для полученного характеристического уравнения можно применять критерий Гурвица, который устанавливает устойчивость системы по анализу неотрицательности коэффициентов характеристического уравнения и всех диагональных миноров матрицы Гурвица.

Пример 1. Передаточная функция разомкнутой импульсной системы имеет вид:

$$W(z) = K \times (1 - a) / (z - a), a = \exp(-Tk / T).$$

Определить при каких K замкнутая система будет устойчивой?

Характеристическое уравнение имеет вид:

$$1 + W(z) = z - a + K \times (1 - a) = 0,$$

после подстановки $z = (w + 1) / (w - 1)$ получим уравнение вида:

$$w \times (1 + a - K + K \times a) + (1 - a + K - K \times a) = 0.$$

Из условия неотрицательности коэффициентов уравнения легко найти,

что

$$0 \leq K \leq (a+1)/(a-1)$$

Пример2.

Характеристическое уравнение замкнутой системы после подстановки $z = (w+1)/(w-1)$ имеет вид:

$$B(w) = A \times w^3 + (14-A) \times w^2 + (16-A) \times w + 10 + A = 0.$$

Определить при каких A система устойчива?

Необходимое условие выполняется при $0 < A < 14$, а достаточным условием является выполнение условия:

$$\frac{(14-A) \times (16-A)}{10+A} \geq (10+A) \times \dots$$

Следовательно, система устойчива в диапазоне $A (0, 5, 6)$.

Использование псевдочастоты в частотных критериях устойчивости.



При подстановке частотные характеристики оказываются периодическими функциями частоты w с периодом $2 \times \pi / T$.

Более удобным использовать понятие псевдочастоты wf на основе W -преобразования:

$$W = (\exp(j \times w \times T) - 1) / (\exp(j \times w \times T) + 1) = j \times \operatorname{tg}(w \times T / 2) = j \times wf \times T / 2$$

где $wf = 2 \times \operatorname{tg}(w \times T / 2) / T$ - абсолютная псевдочастота.

При использовании псевдочастоты построение частотных характеристик более удобно, т.к. при изменении псевдочастоты в пределах от 0 до π / T псевдочастота изменяется в пределах от 0 до бесконечности.

Это делает частотные характеристики дискретных систем подобным частотным характеристикам непрерывных систем. Заметим, что для малых частот $w < 2/T$, когда выполняются условия теоремы Котельникова, псевдочастота практически совпадает с обычной частотой, что оказывается удобным при рассмотрении низкочастотной части характеристик, которые в основном определяют точностные показатели системы.

Пример. Передаточная функция разомкнутой импульсной системы

имеет вид: $W(z) = K \times T / (z - 1)$.

Частотная передаточная функция при подстановке $z = \exp(j \times \omega \times T)$, будет иметь вид:

$$W = K \times T / (\exp(j \times \omega \times T) - 1) = -(K \times T / 2) - j \times (K \times T / 2 \times \operatorname{ctg}(\omega \times T / 2)).$$

Построение частотных характеристик даже в этом простом случае оказывается затруднительным.

Используя понятие псевдочастоты после подстановки

$$z = (1 + \omega) / (1 - \omega) = (1 + j \times \omega f \times T / 2) / (1 - j \times \omega f \times T / 2),$$

получим:

$$W(j \times \omega f) = K \times (1 - j \times (\omega f \times T / 2)) / j \times \omega f.$$

При этом построение частотных характеристик и анализ устойчивости упрощается.

6. СТРУКТУРА И ПРОГРАММИРОВАНИЕ ПРИКЛАДНЫХ ИНТЕРФЕЙСОВ.

Программируемые логические контроллеры положили начало развитию средств сопряжения с объектами АСУТП, но они не являются единственными представителями устройств сопряжения с объектом управления и регулирования (УСО).

Определено два основных уровня иерархии интеллектуальных АСУТП:

- встраиваемые непосредственно в ЭВМ (слот) УСО нулевого уровня;
- выносные УСО первого уровня (крейты специализированных УСО, крейты КАМАК(CAMAC), модули интерфейса VME, контроллеры и модули microPC фирмы Octagon Systems, Compact PCI и др.).

На УСО возлагаются обычно следующие функции:

- нормализация аналогового сигнала (0-5 В, -5 - + 5 В, 0-10 В);
- предварительная низкочастотная фильтрация аналогового сигнала;
- обеспечение гальванической изоляции между источниками аналогового или дискретного сигнала и измерительными и / или статусными каналами системы;
- организация работы с ЦАП и АЦП;
- обслуживание дискретных входов и выходов;
- организация одного из интерфейсов последовательной передачи;
- другие функции сбора и обработки данных.

Среди производителей современных средств УСО можно выделить фирмы ADVANTECH, ANALOG DEVICES, GRAYHILL, OCTAGON SYSTEMS.

Фирма ADVANTECH предлагает полный набор решений задач АСУТП на базе IBM PC совместимых персональных компьютеров. В него входят промышленные PC и рабочие станции, одноплатные компьютеры

и платы ввода-вывода и сбора данных, а также необходимое программное обеспечение. Модули фирмы имеют аббревиатуру ADAM (Advantech Data Acquisition Modules), среди которых модули ADAM-3000/4000/5000:

- ADAM-3011 - модуль гальванического изолированного ввода сигнала с термопар;
- ADAM-3012 - модуль гальванического изолированного аналогового ввода, работающий в режиме измерения напряжения и тока;
- ADAM-3021 - модуль гальванически изолированного аналогового

вывода;

- ADAM - 4000 - устройства удаленного сбора данных и управления на базе канала RS-485 (габаритные размеры 112*60*25 мм) и другие.

Устройства ADAM-4000 обеспечивают прием и дешифрацию команд по каналу RS-485 с линией передачи симметричная экранированная витая пара и скоростью передачи - 1200,2400,4800,9600,19200 бод. Длина сегмента сети - 1200 м., количество модулей в пределах сегмента- 32, максимальное количество модулей в сети - 255. Протокол канального уровня - ASCII с непосредственной адресацией абонентов. Режим обмена данными - асинхронный, полудуплексный, 1 старт-бит, 1 стоп-бит, 8 бит данных.

В [14] приведено описание автоматизированной системы контроля (АСК) технологических параметров тепличного комбината на основе модулей ADAM-4000. В основе АСК - технологическая сеть на основе интерфейса RS-485 с подключенным к ней IBM PC, измерительными модулями ADAM-4000 и расходомером - тепловым счетчиком MT-2000.

Для подключения большого количества датчиков к одноканальным модулям ADAM-4012 используется 4 релейных коммутатора на 16 дифференциальных каналов типа PCLD-788. С помощью коммутаторов опрашивается 23 датчика температуры (по 4 датчика в 5 теплицах, датчик температуры поливочной воды, подпочвенной воды, наружного воздуха), 15 датчиков положения исполнительных механизмов (5 клапанов и 10 фрамуг) и датчики давления пара и воды в различных трубопроводах. Расходомер - тепловой счетчик контролирует давление, температуру и расход воды в прямом и обратном трубопроводах.

Программное обеспечение реализовано на языке Borland Pascal 7.0 под управлением MS DOS 3.3.

Фирма ANALOG DEVICES предлагает преобразователи и нормализаторы аналоговых сигналов различного типа, выполненных в виде серий: 2B, 3 B, 5B, 6B, 7B. Каждая серия имеет широкую номенклатуру модулей, которые работают с различными источниками входных сигналов (токи, напряжения, сигналы от термопар, терморезисторов и др.).

Фирма GRAYHILL выпускает твердотельные реле для промышленных применений, модули ввода-вывода с оптической развязкой, одноплатные контроллеры. Дискретные модули UCO G5 - обеспечивают опрос датчиков с релейным выходом, концевых выключателей, контроль наличия напряжения, тока, а выходные

оптомодули формируют сигналы для управления исполнительными устройствами, пускателями и т.д. Напряжение изоляции у модулей до 4 КВ.

Фирма Octagon Systems выпускает семейство IBM PC совместимых промышленных компьютеров и контроллеров MicroPC. Их отличительные особенности: малые размеры плат, значительная вычислительная мощность, высокая вибро - и ударопрочность, малое энергопотребление, высокая надежность и работоспособность в диапазоне температур от - 40 С до + 85 С. Все это обеспечивает возможность широкого применения данных устройств в системах автоматизации в промышленности, на транспорте и других отраслях [15].

В настоящее время фирмой производятся платы с процессорами от 8088/ 12 МГц до 5x86/133 МГц, содержащие встроенные средства аналогового и дискретного ввода/ вывода.

Практика решения задач автоматизации показывает, что большая часть задач автоматизации может быть решена с помощью встраиваемых одноплатных компьютеров, оснащенных стандартными внешними интерфейсами и средствами ввода/вывода электрических сигналов.

УСО нулевого уровня выполняются, как правило, в одноплатном варианте и содержат в своем составе интерфейсную часть, соответствующую системной шине той ЭВМ, в слот которой осуществляется подключение платы, и функциональную часть, которая может иметь регистры, счетчики, ЦАП, АЦП и др.

УСО первого уровня содержат в своем составе модуль контроллера для связи с ЭВМ и набор интерфейсных модулей для подключения всего многообразия датчиков и исполнительных устройств объекта управления. При этом существенно расширяются возможности по модернизации системы управления объектом, появляется независимость от типа управляющей ЭВМ, т.к. при смене управляющей ЭВМ меняется только контроллер крейта без изменения структуры измерительной и регулирующей части АСУТП.

Первой системой такого типа была модульная программно-управляемая система CAMAC(Computer Applications to Measurement and Control), разработанная под эгидой комитета ESONE (European Standards of

Nuclear Electronics). Промышленностью выпускалась номенклатура модулей КАМАК, превышающая 100 наименований и охватывающая решение всех задач по приему, преобразованию цифровых и аналоговых сигналов, управлению цифровыми измерительными приборами в системах АСНИ - (автоматизированных системах научных исследований). Аппаратные и программные решения, заложенные в реализации данного интерфейса, с успехом были адаптированы на более современные уровни (интерфейс VME в частности).

6.1. ИНТЕРФЕЙС КАМАК.

Рассмотрим логическую организацию и основы программирования данного интерфейса, который можно по праву считать “отцом” средств автоматизации сбора данных и управления оборудованием [16].

Основным элементом интерфейса КАМАК является крейт, объединяющий группу функциональных модулей, содержащий источник питания для всех подключенных модулей и систему вентиляции. Крейт имеет 25 позиций (станций), причем 24 и 25 станции отводились для подключения контроллера крейта (модуля K-109 для DEC- архитектурных машин с шиной МПИ (Q-bus) и C-111 для IBM архитектурных с шиной ISA), а остальные станции предназначены для установки номенклатуры требуемых модулей.

Существует стандарт на логику обмена информацией (протокол) между функциональными модулями и контроллером крейта, который осуществляется через унифицированную магистраль КАМАК.

Большинство линий магистрали сквозные, объединяющие все одноименные контакты первых 24 станций. На магистрали выделены линии сигналов команд КАМАК (NAF): 24 линии номера (N), 4 линии субадреса (A), 5 линий функций для обеспечения адресации модулей, находящихся в крейте. Каждый функциональный модуль соединен с контроллером крейта индивидуальной линией LAM-запроса (“look at me”) для работы в режиме прерывания. Магистраль имеет сквозные 24-разрядные шины данных и адреса и линии сигналов состояния и управления: Q - ответ модуля, X - команда принята, В - занято, Z - начальная установка крейта, С - сброс, I - запрет, S1 и S2 - стробирующие сигналы управления циклом передачи по магистрали КАМАК.

В каждой станции установлен 86-контактный разъем, обеспечивающий доступ к магистрали. Во время командных операций контроллер генерирует команду, состоящую из сигналов: на индивидуальной шине номера, на сквозных шинах субадреса, и на функциональных 5 сквозных линиях F для определения выполняемой операции.

Дешифрация функции осуществляется в модуле. Регламентирующими документами по системе КАМАК были определены 18 из 32 возможных операций (остальные резервные), основные из которых представлены ниже:

1. **F0**- чтение состояния регистра,
2. **F2** -чтение и сброс регистра,
3. **F8** - проверка сигнала LAM- запроса , установленного модулем,
4. **F10** - сброс сигнала LAM- запроса,
5. **F16** - запись информации в регистр модуля,
6. **F24** - запрещение работы модуля,
7. **F25**- запуск модуля (исполнение).

Например, NAF - инструкция N(9) A(4) F(2) означает, что из модуля установленного на станции 9, по субадресу внутри модуля 4, произойдет считывание информации и затем сброс всех триггеров регистра модуля.

Общий принцип программирования аппаратуры КАМАК заключается в создании требуемой последовательности NAF - инструкций, техника программирования зависит от типа используемого контроллера крейта. Рассмотрим некоторые особенности программирования для контроллера K-109.

Для обеспечения программного обращения к модулям в контроллере предусматривается выделение для каждого модуля 16 адресов (по числу возможных субадресов), 16 адресов выделено для самого контроллера (хотя используется лишь 3). Таким образом, для каждого крейта КАМАК в адресном поле ЭВМ отводится $16 \cdot 24 = 384$ адреса, при этом в установку может входить до 4 крейтов КАМАК. Базовый адрес крейта устанавливается в контроллере 2 переключками и имеет адреса:

160000 , 162000 , 164000 , 166000 .

Характерной особенностью контроллера является то, что для каждого модуля выделен адрес в адресном пространстве от 160000 до 177776, определяемый по соотношению для восьмеричных чисел:

$$ADRMOD = \langle \text{адрес контроллера} \rangle + \langle N * 40 \rangle + \langle A * 2 \rangle .$$

Контроллер крейта имеет три программно адресуемых регистра: регистр контроля и состояния (CSR), регистр маски и запросов (DMR) и регистр старшего байта (DHR). CSR имеет адрес, равный базовому адресу контроллера, DMR на 2 больше, а DHR- на 4 соответственно.

Формат CSR следующий:

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|----|----|----|----|----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Q | X | | I | | X | Z | C | D | d | | F4 | F3 | F2 | F1 | F0 |

15- разряд- индикация сигнала Q- ответа модуля,
 14- разряд- индикация сигнала команда принята,
 12- разряд- индикация состояния линии запрета по магистрали,
 10- разряд- разрешение прерывания по X -ответу модуля,
 9 и 8 разряды- генерация сигналов Z и C,
 7- разряд- запрос прерывания, 6 разряд- разрешение прерываний,
 разряды 0- 4- задание функции КАМАК.

Фрагменты программирования с использованием CSR.

Пример 1. Пусть на первой станции крейта КАМАК с базовым адресом 164000 установлен счетчик с предустановкой. Требуется привести аппаратуру в начальное состояние и установить в счетчике число 10000 по субадресу 0.

NAF - инструкция в этом случае соответствует процедуре записи числа-

N (1)A(0)F(16). Процедура выглядит следующим образом:

```
MOV # 1000,      @# 164000; генерация сигнала очистки Z;
MOV # 16.,       @# 164000; код функции записан в CSR;
MOV #10000.,     @# 164040; запись числа в модуль.
```

Пример 2. Требуется программно проверить наличие и работоспособность модулей, установленных в этом же крейте на станциях 4 и 5, если известно, что в обоих модулях выполняется команда F (9) по субадресу A (0).

```

MOV # 1000,    @# 164000
MOV # 9.,      @# 164000
TST  @# 164200 ; запуск модуля на 4 станции;
BIT  # 40000,  @# 164000 ; проверка X - ответа модуля;
BEQ  ERR1 ; при отсутствии ответа вызов подпрограммы ;
TST  @# 164240 ; запуск модуля на станции 5 ;
BIT  # 40000,  @# 164000
BEQ  ERR2 ; переход на метку ERR2 , если X= 0.

```

Обработка прерываний в модулях КАМАК.

Введение запроса на обслуживание L от модуля играет аналогичную роль требования прерывания от ЭВМ. Модуль КАМАК может содержать несколько функциональных узлов, каждый из которых может требовать обслуживания. Независимо от числа одновременно возникших LAM-запросов модуль посылает в контроллер только один L - запрос по индивидуальной линии магистрали КАМАК.

В интерфейсной части модуля КАМАК располагается триггер LAM - статуса, который устанавливается в 1 сигналом от LAM - источника и служит для фиксирования и отображения LAM - состояния или LAM - запроса. Кроме того, интерфейсная часть модуля содержит триггер маски, сигнал с которого управляет прохождением L - запроса при установке LAM - маски в 1. Сброс триггера маски (команда F24) запрещает дальнейшую обработку LAM - запроса. Каждый источник LAM - запросов в модуле имеет свой статусный триггер и триггер маски с соответствующими субадресами. Сигнал L - запроса от модуля образуется путем логического сложения LAM-требований: $L = M(1) * LAM(1) + M(2) * LAM(2) + + M(i) * LAM(i)$.

Таким образом, модуль выставляет L - запрос на обслуживание при возникновении хотя бы одного LAM - требования, т.е. не замаскированного LAM - запроса. Идентификация источника L - запроса производится путем проверки состояния LAM - требования командой F(8), в которой каждому LAM - требованию ставится в соответствие определенный субадрес. Действие команды сводится к отображению LAM - требования сигналом Q- ответа, который модуль выставляет на магистраль КАМАК. Сброс в нуль триггера LAM - статуса после идентификации источника запроса осуществляется

командой F(10).

В контроллере за обработку LAM - запросов отвечает регистр маски и запросов - DMR - Demand and Mask Register, формат которого представлен ниже:

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| L8 | L7 | L6 | L5 | L4 | L3 | L2 | L1 | M | M | M | M | M | M | M | M |
| | | | | | | | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

DMR предназначен для хранения маски (младший байт) и отображения упорядоченной информации о L - запросах функциональных модулей (L1 -L23 и внутреннем запросе контроллера крейта L (X)(старший байт).

Упорядочение L- запросов производится с помощью коммутатора запросов. Все запросы от модулей и внутренний запрос контроллера разбиваются на группы L1 - L8, устанавливается приоритет обслуживания групп. Обычно контроллер содержит схему приоритетных прерываний, на вход которой поступают незамаскированные запросы. Если одновременно поступают запросы от нескольких групп, то прерывание предоставляется группе с более высоким приоритетом (чем выше номер группы, тем выше приоритет). Если прерывания разрешены ($d=1$ в CSR), и аппаратно установлен разряд D в CSR, то генерируется сигнал требования прерываний ТПР. Адрес вектора прерываний формируется схемой формирования адреса прерываний, на вход которой поступает номер i - группы, которой предоставлено прерывание.

$$ABП = \langle 300 \rangle + \langle 40 \rangle * (k-1) + \langle 4 \rangle * (i-1),$$

где АВП - адрес вектора прерывания, k - номер крейта, i - номер группового запроса. В простых и средних по сложности схемах число источников запросов невелико, и за каждой группой можно закрепить только один источник из L1 - L8.

Приведем примеры обслуживания L - запросов.

Пример 1. На станции 5 установлен модуль АЦП, который по окончании преобразования, выставляет L - запрос и блокирует свой вход. Результат может быть считан по команде N(5)A(0)F(2), при этом происходит сброс L- запроса и разблокировка входа АЦП. Необходимо снять 200 значений.

CSR= 160000

```

MAS: . BLKW 200. ; резервирование памяти с адреса MAS;
      MOV # MAS, R0 ; адрес массива в R0;
LOOP: MOVB # 8., CSR ; подготовка команды F (8);
WAIT: TST @# 164240; запуск и проверка Q-ответа
      MOVB #2, CSR
      MOV @# 164240, ( R0 ) +; передача кода АЦП в массив;
      CMP # MAS+400., R0; массив заполнен ?
      BNE LOOP

```

BPL WA

Этот пример показывает как можно обрабатывать L - запросы при использовании средств в модуле и CSR.

Другим способом может служить работа с DMR. Пусть запрос заведен на коммутационное поле 1(L1). В этом случае процедура имеет вид:

```

CSR= 160000 ; адрес контроллера крейта;
MAS: . BLKW 200. ; резервирование памяти с адреса MAS;
      MOV # MAS, R0 ; адрес массива в R0;
      MOVB # 2, CSR ; подготовка команды F (2);
WAIT: BIT #400, CSR+2 ; проверка запроса L1 в DMR;
      MOV @# 164240, ( R0 ) +; передача кода АЦП в массив;
      CMP # MAS+400., R0; массив заполнен ?
      BNE WAIT

```

BEQ WA

В заключении рассмотрения интерфейса КАМАК отметим, что регистр старшего байта предназначен для работы с 24 - разрядными словами данных. Аппаратно он представляет собой два регистра, один из которых доступен только для чтения с связан с линиями магистрали КАМАК R17-R24, а другой для записи, связанный с линиями W17-W24.

Поскольку оба регистра имеют один и тот же адрес, то для программиста они выступают как один регистр с двунаправленной передачей данных.

Для облегчения программирования аппаратуры КАМАК существуют пакеты прикладных программ на языках высокого уровня (в частности ФОРТРАН и ПАСКАЛЬ). Пакет включает подпрограммы декларации адресов регистров модулей КАМАК, выполнения одиночных команд КАМАК, управления крейтом и обработки L - запросов.

VME (Versa Module Europe) - шина, являющаяся международным стандартом для 32 разрядных микропроцессорных систем (IEC 821 BUS)[17].

В ноябре 1984 г. образовалась Международная ассоциация по поддержке и распространению архитектуры VME - VITA(VME-bus International Trade Association), спонсорами выступили MOTOROLA и SUN Microsystems.

В мире производится более 1000 изделий на основе шины VME более 150 фирмами. Технология шины VME зародилась в 1979 году, как спецификация фирмы MOTOROLA, а в 1987-1988 гг. была признана международным стандартом. Несмотря на почтенный возраст, шина VME остается лидером для систем промышленной автоматизации за рубежом.

Так по исследованиям корпорации VDC распределение продаж на рынке компьютерных плат в 1996 году выглядит следующим образом:

VME - bus - 1262 млн. долларов;

PC - архитектуры (включая ISA,EISA, Multibus I и II, PC/ 104) - 720 млн.;

интерфейс PCI - 238 млн. долларов;

встроенные материнские платы - 206 млн. долларов.

Членами VITA являются около 100 европейских, азиатских, американских производителей совместимой продукции VMEbus: DEC, HP, IBM и др.

Технология VME позволяет создавать вычислительные системы в очень широком диапазоне производительности от настольных компьютеров до многопроцессорных супер-ЭВМ, от простых и дешевых промышленных контроллеров до мощнейших микропроцессорных систем управления десятками тысяч аналоговых и цифровых каналов ввода/вывода.

Не претендуя на достижение рекордных показателей, VME-bus обеспечивает наилучшее соотношение цена/ производительность и представляет хорошие возможности для наращивания ресурсов. Ввод в строй нескольких расширений и новой версии стандарта для 64 - разрядной передачи данных VME 64 показал, что в настоящее время потенциал шины далеко не исчерпан. Новейшие реализации VMEbus

обеспечивают пропускную способность 320 Мбайт/с.

Архитектура VME выросла вокруг семейства MOTOROLA 68 xxx, но сейчас имеются VME - реализации для RISC - процессоров, рабочих станций SUN, DEC, HP, Intel, Power PC.

Шина VME является развитием (дополнением) шины КАМАК. Несмотря на отличие электронных частей шин, практически все новые магистрали используют единый механический стандарт - **Евромеханику (МЭК 297-1,297-3)**. Конечная система компонуется из функциональных модулей VME, устанавливаемых в крейты, число которых не ограничено. Крейт представляет собой каркас с объединительной магистралью VME, источником питания и вентиляцией.

В каждый крейт можно поместить до 21 модуля VME. Модули соединяются через объединительную плату с нормированным волновым сопротивлением. В качестве соединителей используются надежные 96-штырьковые разъемы DIN 602-3, причем 8 и 16 - разрядные модули имеют один разъем, 32/64 - разрядные два. Высота крейта 16 - разрядной шины (разъем J2) может быть 100 мм, а в 32 - разрядном варианте - 233 мм. Модули в глубину имеют размер 160 мм.

Сегодня технология VME включает и несколько расширений. Технология оперативной замены LIVE INSERTION представляет собой минимальное аппаратное дополнение, позволяющее беспрепятственно вставлять/ вынимать модули из работающей системы. Для реализации горячей замены предложен специальный механизм изоляции модуля от шины.

Шина VME используется как самостоятельно, так и в сочетании с шинами VMX и VMS.

VME- 32 разрядный асинхронный параллельный интерфейс с 7 уровнями прерывания, позволяющий сосредоточить в одном крейте память до 4Гбайт. Захват шины модулем, который должен в данный момент ею управлять, осуществляется специальным модулем - арбитром шины.

Шина VME рассчитана на широкую область применения от систем реального времени до мультипроцессорных систем обработки данных. В число наиболее распространенных типов модулей входят

процессорные платы с различными микропроцессорами (INTEL 80386, MC 68020 и др.), модули каналов аналоговых и цифровых входов/выходов, интерфейсы локальных сетей и сетей ЭВМ, интерфейсы шин Q-bus, Unibus, ISA, контроллеры графических дисплеев, интерфейс приборной шины HP-IB (МЭК 625), интерфейсы персональных компьютеров. Конкретные существующие VME - приложения применяются: в АСУТП, САПР, станках с ЧПУ, робототехнике, ГПС, системах машинной графики, локальных сетях, АСНИ.

Семейство VME включает три магистрали: собственно VME, последовательную линию передачи сообщений - VMS и дополнительную локальную шину, объединяющую до 6 модулей.

VMS ориентирована на связь модулей внутри одного крейта, а не организацию межкрейтного обмена. Необходимость дополнительной магистрали возникает при организации многопроцессорных систем, процессоры которых имеют локальные ресурсы (дополнительную память, специализированные устройства ввода/вывода и др.), общение с которыми перегружает основную магистраль VME. С помощью дополнительной магистрали возможно осуществление межкрейтной связи. Предложено несколько вариантов организации локальной шины: VMX, MVMX32, VMC, VSB.

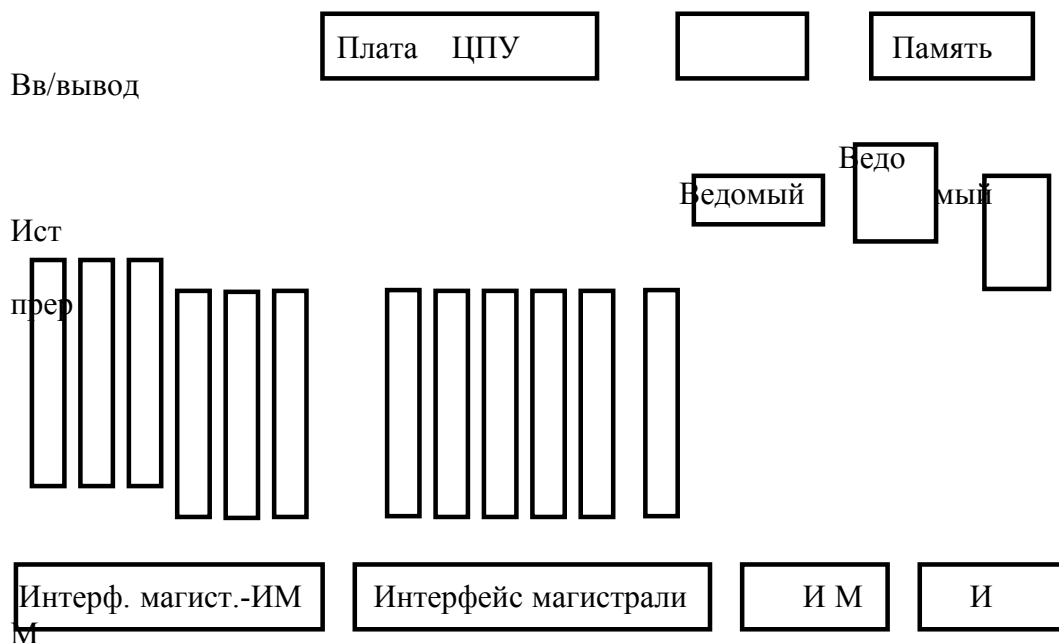
VMX - имеет 12 линий адреса и 32 линии данных. Линии адреса мультиплексированы, что обеспечивает адресное пространство до 16 Мбайт. Скорость передачи по VMX несколько выше, чем по VME, за счет простого протокола и меньшей длины самой магистрали.

MVMX32 - ориентирована на МП Motorola 68020, имеет 32 разрядную мультиплексированную шину адреса и данных.

VMC - имеет 32 линии адреса и 16 линий данных для МП 68000.

Система VME состоит из интерфейсных логических схем, набора функциональных модулей, 4 групп сигнальных линий: передачи данных, арбитража, прерываний и служебной шины.

На рис.14 представлена организация шины VME -bus.



| |
|-----------------|
| ПЕРЕДАЧА ДАННЫХ |
| ПРЕРЫВАНИЯ |
| АРБИТРАЖ |
| СЛУЖЕБНАЯ |

Рис .14.
Магистраль передачи данных.

Магистраль VME имеет отдельные стробы адреса и данных, что позволяет ведущему выставить адрес для следующего цикла в момент передачи данных предыдущего. Этот способ выставления адреса называется адресным конвейером. Асинхронный протокол шины позволяет каждому устройству работать с требуемой скоростью. Минимальное время цикла составляет 100 нс, что соответствует скорости передачи 40 Мбайт /с.

Арбитраж магистрали.

Процесс, определяющий какой из ведущих сможет использовать магистраль при одновременном поступлении требований, называется арбитражем магистрали.

Магистраль арбитража решает две основные задачи:

- предотвращение одновременного использования шины двумя ведущими;
- оптимальное распределение магистрали данных по требованиям от различных ведущих.

Она использует: линии требований(BR0-BR3),4 последовательных цепочки предоставления магистрали(BG0-BG3), линии занято (BBSY) и “освободить” (BCLR).

Магистраль арбитража использует два типа функциональных модулей: арбитр и источник требований магистрали. Арбитр принимает запросы от источников требований и предоставляет контроль над ней только одному из них при одновременном поступлении нескольких требований.

Источник требований магистрали размещается на одной плате с ведущим или устройством обработки прерываний, выставляет запрос BR n на передачу по магистрали и ожидает разрешения, устанавливает сигнал BBSY (удерживает его до окончания обмена) по получению контроля над магистралью.

Протокол VME описывает **магистраль прерываний**, включающую в себя линии, необходимые для выработки, приема и обслуживания прерываний. Спецификация VME не накладывает ограничений на подпрограммы обслуживания прерываний, каждая из подпрограмм - рассматривается как отдельная задача, запускаемая соответствующим прерыванием.

Процесс прерывания можно разделить на 3 фазы:

- выработка запроса прерывания - с момента установления низкого уровня источника прерывания по шине IRQ n до получения магистрали данных устройством обработки прерываний в цикле арбитража;
- подтверждение прерывания - устройство обработки прерывания считывает по магистрали данных статус источника прерывания и передает управление соответствующей подпрограмме;
- обслуживание прерывания - работает подпрограмма обслуживания прерывания.

Магистраль прерываний использует 7 линий запроса прерываний (IRQ1-IRQ7, 7 - наивысший приоритет), линию подтверждения прерываний- IACK, которая соединяет все станции крейта с контактом IACKIN первой станции. Низкий уровень этой линии инициирует последовательную цепочку подтверждения прерываний через ее драйвер. Драйвер последовательной цепочки располагается в первой станции крейта. В его функции входит инициация последовательной цепочки подтверждения прерывания, гарантирующей участие только одного источника в цикле подтверждения прерывания.

Системы, использующие прерывания, могут быть разделены на 2 группы:

- с централизованной обработкой запросов, имеющих только один модуль, обрабатывающий все запросы прерываний;

- с распределенной обработкой запросов, включающих несколько модулей до 7, каждый из которых обрабатывает свою группу запросов.

В системах первой группы все программы обслуживания прерываний выполняются одним процессором (супервизором). В системах второй группы имеется от 2 до 7 центров обработки прерываний, каждый из которых обрабатывает свою группу (от 1 до 6). Такая структура удобна для многопроцессорных систем с распределенной обработкой данных, включающих несколько равноправных процессоров. Процессору А, чтобы связаться с В, достаточно запросить прерывание по одной из соответствующих процессору В линий IRQ n.

Служебная магистраль - определяет сигналы, предназначенные для организации работы модулей VME, содержит функциональные модули

системного генератора, генератора тактовых импульсов для последовательной магистрали, контроллера питания.

Системный генератор формирует импульсы с частотой 16 МГц, задает стандартную частоту для определения временных интервалов, располагается на плате системного контроллера в первой станции (его работа никак не синхронизирована с сигналами на шинах VME).

Генератор тактовых импульсов для последовательной магистрали выдает сигналы специальной формы, используемые интерфейсами шины VMS, расположенными на платах VME.

Контроллер питания обеспечивает необходимую информацию о состоянии источника питания крейта.

Достоинства интерфейса VME:

- сравнительно высокие технические характеристики,
- относительно высокий уровень проработки идеологии системного параллельного интерфейса,
- достаточно гибкий арбитраж и относительно простой протокол обмена,
- возможность построения на его основе мультимикропроцессорных систем,
- объединение в одной системе модулей, работающих с 8, 16, 32 - разрядами данных, 16, 24, 32 - разрядами адреса.

Недостатки интерфейса VME:

- отсутствие контроля передач,
- использование последовательного арбитража по цепочечным линиям,
- невозможность осуществления “широковещательных передач типа” один - всем “ и “ все - одному “ (broadcast и broadcast),
- большое число функциональных линий, которое вызывает необходимость применения второго интерфейсного разъема для обеспечения полного набора свойств.

Такие недостатки могут ограничивать использование интерфейса VME в системах, где достоверность обработки информации, надежность функционирования, а также живучесть (отказоустойчивость), являются определяющими характеристиками. Однако, можно рассчитывать, что после того, как новые расширения VME 64, будут освоены на рынке высокопроизводительной аппаратуры, высокая рентабельность VME - систем будет восстановлена, ведь “Versa” - значит - изменяемый,

подвижный.

Сейчас VME - bus ощущает сильное давление дешевых систем на базе PC- систем.

7.1. СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Системное программное обеспечение состоит из операционной системы и системы программирования.

Операционная система.

Операционная система (ОС) выполняет следующие функции:

- организует передачу информации между внутренними устройствами ПК;
- поддерживает работу периферийных устройств;
- распределяет ресурсы между задачами и поддерживает взаимодействие задач при параллельной обработке;
- организует диалог с пользователем и обеспечивает выполнение системных и пользовательских (прикладных) программ.

В зависимости от количества одновременно обрабатываемых задач (процессов) и числа пользователей, которые могут обслуживаться операционной системой, можно выделить:

- однопользовательские однозадачные ОС (CP/M фирмы Digital Research для 8-битовых ПК, PC DOS, MS DOS, RT11SJ);
- однопользовательские многозадачные ОС (CP/M86, RT11FB, OS/2 и др.);
- многопользовательские многозадачные ОС (UNIX, Windows NT, QNX).

Наиболее популярной ОС является MS DOS, которая появилась в 1981 году, а начиная с 1996 года распространяется в виде WINDOWS 95, а затем в виде WINDOWS NT. В последней ОС реализован новый индустриальный мощный код, плюс встроенная поддержка работы в сети, что превращает ОС и аппаратную среду в единое и неразрывное целое.

Это и есть та самая “ новая технология “ (New Technologie), отразившаяся в названии ОС как NT.

ОС WINDOWS 3.1(3.11) и WINDOWS 95 обеспечивают мультизадачный режим работы, оптимальное управление ресурсами компьютера, графический пользовательский интерфейс, возможность работы в сетевой среде, интерфейс мультимедиа.

Различают 2 типа многозадачности:

- кооперативную (непреemptивную или неpreemptивную);
- preemptивную (преemptивную с разделением времени).

В первом случае (в Windows 3.1.) ОС не выполняет

полноценный контроль за тем, как отдельные задачи распределяют ресурсы. Момент передачи управления системе здесь зависит от хода выполнения задачи.

Во втором случае (OS/2 и Windows NT) каждая задача получает фиксированный квант времени процессора. По истечении этого кванта система получает управление, чтобы выбрать другую задачу для активизации. Впрочем, если возникает системное событие (завершение ввода-вывода) или задача обращается к системе до истечения ее кванта времени, то это также служит причиной передачи управления системе и последующего переключения задач. Для переключения между программами (приложениями) используются функции специальной программы - Менеджер задач.

Для работы ОС Windows NT необходимо ПК на базе не ниже 386 SX и требует 8 Мбайт памяти. Windows 3.1. требует не менее 2 Мбайт. Параллельно могут выполняться не только Windows, но и DOS - приложения. Для этого в большой по размеру рабочей памяти эмулируется (создается по аналогии) несколько виртуальных процессоров (машин), подчиненных своим приложениям.

Виртуальная машина - это некоторая часть разделяемых ресурсов компьютеров, представленных в одной задаче.

Виртуальная память - это расширенное адресное пространство задачи, полученное отображением части адресного пространства на внешнюю память. Если задаче требуется обращение к адресам, которые отсутствуют в оперативной памяти, то ненужный фрагмент адресного пространства вытесняется во внешнюю память, а необходимый фрагмент адресного пространства загружается (подкачивается) с диска.

Используемый для этой цели объем внешней памяти, называемый файлом подкачки (свопинга), примерно в 3 раза больше объема оперативной памяти, причем участок диска, необходимый для отображения части адресов, должен быть непрерывным. По сути, это дисковое пространство представляет собой буфер, необходимый для обмена фрагментами единого адресного пространства - на место ненужного фрагмента адресного пространства в ОЗУ загружается требуемый фрагмент из файла подкачки.

Графика Windows (окна и их содержимое), полученная стандартными библиотечными процедурами Windows, является универсальной, т.к. пользователю и программисту предоставлены средства, снимающие проблему обеспечения совместимости с конкретным типом дисплея или принтера (за счет драйверов). Почти все драйверы Windows

фактически выполняют функции BIOS, работая с устройствами напрямую.

Графический пользовательский интерфейс реализует “концепцию рабочего стола” (система окон аналогична листу бумаги) и принцип WYSIWYG (What You See What You Get) - “что видишь, то и получаешь”).

Windows 95 - высокопроизводительная, многозадачная и многопоточная 32 разрядная ОС, поддерживающая 16 - разрядные приложения. Она является первой ОС, не требующей MS DOS.

Основные преимущества, реализованные в Windows -95 :

- интегрированная ОС;
- вытесняющая многозадачность;
- многопоточность.

Интегрированная ОС - ОС, ядро, которой загружаемое в момент включения компьютера, активизирует графический интерфейс пользователя и обеспечивает полную совместимость с MS DOS.

Вытесняющая многозадачность - свойства ОС самостоятельно в зависимости от внутренней ситуации передавать или забирать управление у того или иного приложения.

Многопоточность - свойства ОС выполнять операции одновременно над потоками нескольких 32 - битовых приложений, называемых процессами.

Процесс состоит из потоков. Поток - это некоторая часть процесса, которой может быть выделено процессорное время для одновременного выполнения наряду с другими потоками того или иного процесса.

В Windows NT появился новый класс прикладных программ многонитиевого типа (multithreading), который имеет способность одной прикладной программой запускать одновременно несколько прикладных задач.

Windows NT - ОС, обеспечивающая возможность работы в системах реального времени.

OS/2 - поддерживает мультипроцессорную обработку - до 16 процессоров типа Intel, PowerPC. Версия OS/2 Warp имеет встроенный доступ к сети Internet, систему распознавания речи Voice Type. В OS/2 могут выполняться прикладные программы Windows 3.1., но не могут Windows 95 и Windows NT.

UNIX - многопользовательская, многозадачная ОС реализует

принципы открытых систем и широкие возможности по комплексированию в составе одной вычислительной системы разнородных технических и программных средств.

UNIX обладает наиболее важными качествами:

- переносимость прикладных программ с одного компьютера на другой;
- поддержка распределенной обработки данных в сети ЭВМ;
- совместимость с RISC - процессорами.

Согласно прогнозам объем UNIX систем будет возрастать, особенно при переходе к сетевым технологиям.

Сетевые ОС - комплекс программ, обеспечивающий обработку, передачу и хранение данных в сети, используют архитектуру “клиент-сервер” или одноранговую архитектуру(LANServer, Netware, VINES, Windows NT).

Система программирования.

Система программирования обеспечивает автоматизацию процесса разработки и отладки программного обеспечения. Она включает в себя:

- языки программирования;
- трансляторы и инструментальные средства программирования;
- текстовые редакторы;
- системы машинной графики;
- системы создания и управления базами данных.

Языки программирования подразделяются на:

- машинные языки и коды;
- машино - ориентированные (ассемблеры);
- алгоритмические (универсальные - Паскаль, Фортран, Бейсик и др.);
- проблемно - ориентированные (Кобол, Лого, Форт, Смолток и др.);
- функциональные (ЛИСП, Снобол, Пролог и др.).

На рынке программных продуктов все большее значение приобретают интегрированные системы программирования, позволяющие вести подготовку, отладку ПО в одной программной среде. К таким средствам можно отнести Borland Pascal, современные средства Delphi 2.0, Clarion for Windows, Java.

Разработчику ПО в система Delphi 2.0 предоставлены:

- объектно-ориентированный язык программирования;
- высокопроизводительный компилятор - скорость 120000 строк /мин;
- объектно-ориентированная модель многократно используемых компонентов;

- средства визуального создания программ - набор средств для создания системы меню, экранных форм, отчетных форм, использование библиотеки визуальных компонентов и визуальных объектов;
- масштабируемая технология работы с базами данных - использование языка SQL, встроенная поддержка баз данных под управлением СУБД Oracle, Sybase, Interbase;
- принцип “открытой системы” - возможность добавления новых средств и перенос на другие платформы.

Для объектно-ориентированного проектирования ПО характерно:

- разработка диаграммы аппаратных средств системы обработки данных, показывающей процессоры, внешние устройства, вычислительные сети и их соединения;
- разработка структуры классов, описывающей связь между классами и объектами (класс-совокупность объектов, характеризующихся общностью применяемых методов обработки или свойств);
- разработка диаграмм объектов, показывающих взаимосвязь с другими объектами;
- разработка внутренней структуры программного продукта.

Все создаваемые программы в этой системе программирования разрабатываются как экранные формы. По отзывам экспертов, Delphi наиболее перспективная среда разработчика Windows - ориентированных приложений, функционирующих в архитектуре клиент - сервер.

7.2.ОПЕРАЦИОННЫЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ.

В настоящее время не существует одной-единственной ОС, устраивающей всех пользователей без исключения.

При выполнении задач АСУТП пользователь рано или поздно столкнется с серьезными ограничениями “офисных” ОС типа DOS, Windows, UNIX, Netware, предназначенных не для работы в реальном времени.

В этом случае придется выбирать и использовать ОС РВ (реального времени).

Большое разнообразие ОС и ядер реального времени убедительно показывает, что не родился еще человек или фирма, способные предложить оптимальное решение для всех случаев жизни. Из всех ОС РВ наиболее глубокие корни в России имеет ОС - QNX.

К объективным причинам этого можно отнести засилие у нас архитектуры микропроцессоров X86 фирмы Intel, к субъективным - наличие нескольких компаний, занимающихся дистрибуцией QNX.

Система реального времени должна гарантировать оперативное выполнение всех запросов в течении заданных интервалов времени.

Скорость вычислительных процессов в ЧПУ согласуется ОС со скоростью процессов в технологическом оборудовании благодаря организации приоритетного обслуживания различных объектов.

Система называется системой реального времени, если правильность ее функционирования зависит не только от логической корректности вычислений, но и от времени, за которое эти вычисления производятся. Говорят, что система работает в реальном времени, если ее быстродействие адекватно скорости протекания физических процессов на объектах контроля и управления. Таким образом, большинство систем автоматизации являются системами реального времени.

Принято различать системы “жесткого” и “мягкого” реального времени. Системой “жесткого” реального времени называется система, где неспособность обеспечить реакцию на какие-либо события в заданное время является отказом и ведет к невозможности решения поставленной задачи. Время реакции в таких системах должно быть минимальным.

Точного определения для “мягкого” реального времени нет, поэтому будем считать, что сюда относятся все системы, не попадающие в категорию “жестких”.

В таблице 13 представлены некоторые ОС РВ.

Таблица 13

| Компания | Название | Компьютеры микропроцессоры |
|------------------------|--------------|---|
| DEC | Digital UNIX | MIPS R4000, SPARC, VAX family, M68060, M68040, M68030, M68020 |
| APPLE | ARTA | M68040, M68030, M68020 M68000, Power PC-601, MIPS R3000 |
| Hewlett Packard (HP) | HP-RT | |

| | | |
|----------------------|----------|--|
| IBM | OS/OPEN | Power PC-604,603,601 |
| MOTOROLA | VME exec | M68060,M68040,M68030 M68020,M68010,M68000 |
| SIEMENS | RMOS | Siemens 167,166 |
| QNX Software Systems | QNX | I80386,I80486,Pentium PRO, AM386 EM |

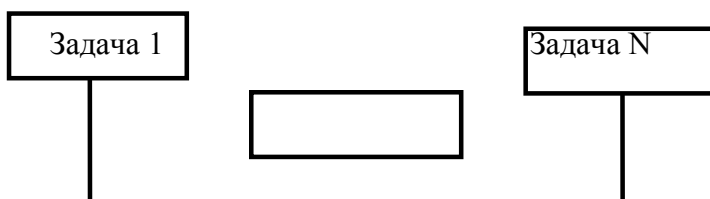
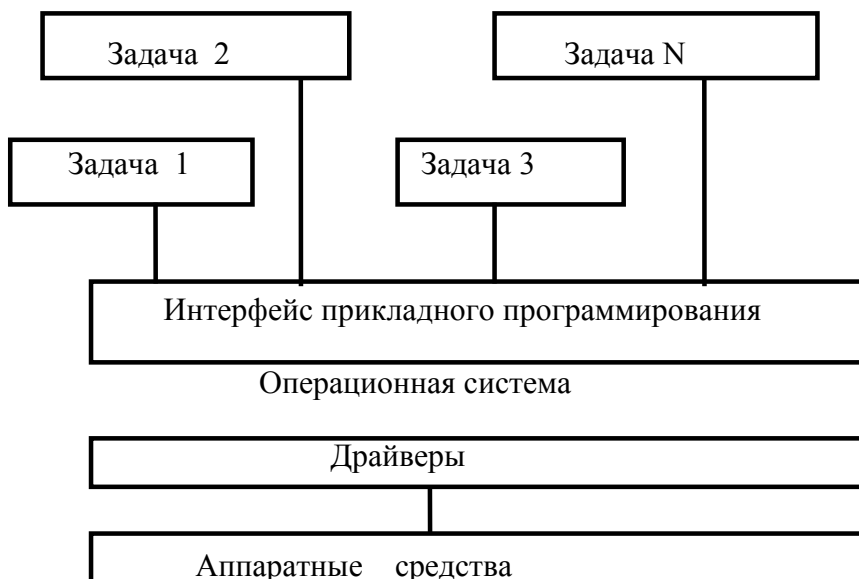
Все ОС РВ как правило являются многозадачными ОС.

По своей внутренней архитектуре ОС РВ можно условно разделить:

- на монолитные ОС;
- ОС на основе микроядра;
- объектно-ориентированные ОС.

Ядра представляют пользователю такие базовые функции, как планирование и синхронизация задач, управление памятью и т.п. ОС РВ имеют файловую систему, сетевую поддержку, интерфейс с оператором и другие средства высокого уровня.

На рис. 15 представлены функциональные схемы трех разновидностей ОС РВ.



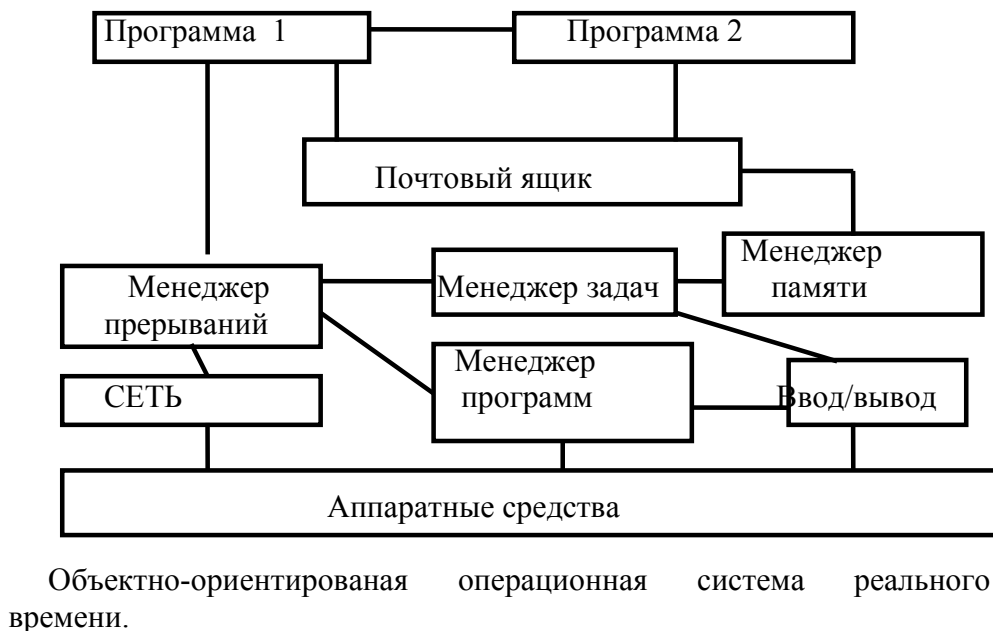
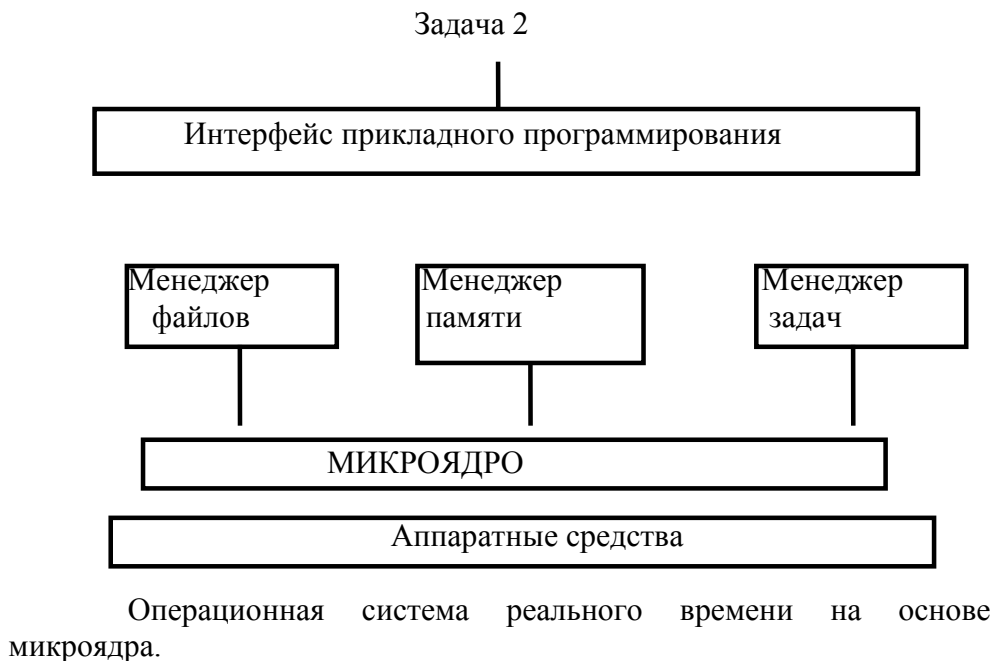


Рис.15.

7.3.ОС РВ QNX.

QNX - ОС РВ - сетевая, многозадачная, многопользовательская операционная система реального времени для машин IBM/PC [18].

Этапы развития QNX .

1970 г. - Гордон Белл и Дэн Додж начали разработку высокопроизводительного ПО для МП фирмы Intel 8088/8086 в университете Ватерлоо (Онтарио, Канада).

1980 г. - основание фирмы Quantum Software Systems.

1982 г. - создание ОС РВ QNX - первой в мире многопользовательской, многозадачной ОС РВ.

1984 г. - разработка версии QNX для PC/AT. Включение в QNX поддержки в рамках локальной сети. QNX становится первой коммерческой ОС, выполняющей DOS, как одну из своих задач.

1987 г. - создание версий QNX 2.0 для PS/2.

1989 - 1991гг. - разработка QNX 4.0, удовлетворяющей стандарту открытых систем POSIX.

1993 г. - разработана версия QNX 4.2 для 32-разрядной архитектуры.

Ее можно назвать 16/32 разрядным гибридом. В каждом узле локальной сети QNX обеспечивается поддержка до 100 пользователей и выполнение до 255 параллельных процессов.

Процесс представляет собой отдельно загружаемый программный модуль (файл), который, как правило, во время исполнения имеет в памяти свои независимые области для кода и данных.

В отличие от этого **поток** может использоваться общими участками кода и данных в рамках единого программного модуля (многопоточная программа редактор WORD) .

Принято различать две разновидности задач: **процессы и потоки**.

Для локальной сети ОСРВ QNX доступны все аппаратные и программные ресурсы, имеющиеся на каждой ЭВМ.

QNX находит применение в системах добычи и транспортировки газа и нефти, управлении технологическими процессами в металлургии и машиностроении, в водоснабжении, энергетике, включая атомные станции, в управлении роботехническими комплексами.

С помощью эмулятора RunDOS программы из DOS могут работать в среде QNX. В QNX реализованы средства интеграции с другими ОС по протоколам Netbios и TCP/IP с использованием адаптеров Arcnet, Ethernet, FDDI и последовательного канала. QNX может распознавать пакеты других сетей, что позволяет к одному физическому кабелю

присоединять ЭВМ, принадлежащие различным локальным сетям и при этом различные сети не мешают друг другу.

Среда QNX похожа на другие ОС. Она включает в себя текстовые редакторы, файл - менеджеры, большое количество утилит, табличные процессоры, языки программирования (Си, Ассемблер, Паскаль, Фортран), базы данных.

ОС QNX выполнена в виде очень малого ядра (10 Кбайт) памяти, которое реализует всего 4 функции:

- взаимодействие процессов;
- сетевые операции нижнего уровня;
- планирование процессов;
- диспетчеризация прерываний.

Выполнение остальных функций обеспечивается задачами - администраторами, которые связываются с ядром системными вызовами, основанными на прерываниях и специальных механизмах связи.

Администраторы не являются обязательными компонентами системы и могут включаться по мере необходимости.

QNX содержит:

- администратор процессов (Process Manager), который отвечает за распределение памяти, запуск и окончание задач в системе;
- администратор периферийных устройств (Device Manager), управляющий работой периферии;
- администратор файловой системы (File System Manager);
- администратор сети (Network Manager), обеспечивающий коммуникации в сети.

Технология микроядра позволяет конструировать необходимую среду верхнего уровня, из которой можно легко получить доступ ко всем функциональным возможностям уровня аппаратного обеспечения.

Преимущество концепции микроядра заключается в том, что каждый компонент системы представляет собой самостоятельный процесс, запуск или остановка которого не отражается на работоспособности других процессов.

Архитектура QNX позволяет применять в составе АСУТП компьютеры различной конфигурации и производительности. В QNX применяется единый универсальный механизм для взаимодействия между всеми процессами в сети (системными и прикладными), основанный на

передаче сообщений, благодаря чему легко реализуется распределенная сетевая обработка.

При работе QNX в локальной сети ядро, получив сообщение, предназначенное для передачи с одного узла на другой, передает его администратору сети, который непосредственно связан с ядром и обеспечивает передачу сообщений через локальную сеть.

Параметры QNX:

- файловая система поддерживает до 32767 файлов, размер каждого файла до 1 Тбайта (1 099 511 627 776 байт);
- поддержка сетей до 254 узлов;
- многопользовательский режим: до 57 пользователей в каждом узле;
- многозадачный режим: до 150 задач;
- управление 32 прерываниями, 16-доступны пользователю;
- выполнение практически всех прикладных программ среды MS DOS.

QNX может загружаться как из ПЗУ, флэш-памяти, так и с помощью загрузки по сети, поддерживает все шины, используемые в PC-контроллерах, включая ISA, PCI, Compact PCI, PC/104, VME.

Более 100 фирм выпускают продукцию, ориентированную на QNX.

Для IBM - архитектурных машин в настоящее время QNX занимает около 80 % рынка ОС РВ.

7.4. Прикладные пакеты SCADA для АСУТП.

SCADA(Supervisory Control and Data Acquisitin)-системы диспетчерского контроля и сбора данных.

Пакеты SCADA позволяют в сжатые сроки вести разработку ПО для основного уровня управления в иерархической структуре АСУТП, на котором осуществляется сбор и обработка информации, поступающей от датчиков и контроллеров технологического оборудования.

Системы SCADA обычно содержат следующие средства:

- базы данных (БД) параметров и диалоговый редактор;
- средства сбора и отображения данных по любому параметру БД с учетом предыстории;
- графический редактор статических изображений;
- графический редактор динамических изображений;
- графический редактор символов, позволяющий создавать библиотеки типовых пиктограмм;
- генератор отчетов, использующий данные реального времени и предыстории;
- средства отображения и обработки событий реального времени

(сигнализация на мнемосхемах, сводки событий и тревог, звуковая и речевая сигнализация);

- параллельная защита и средства разграничения доступа по уровням прав операторов.

Существует готовый пакет SCADA для QNX - RealFlex фирмы BJ Software Systems (США), обладающий возможностями графического отображения данных, генерации отчетов, парольной защиты, многотерминальной поддержки, поддерживает работу до 16 драйверов устройств ввода-вывода, хранит данные предистории, выполняет вычисления в оперативном режиме.

Пакет может работать с внешними устройствами и платами ввода-вывода фирм Allen - Bradley, Buft-Bown, с отечественным ПЛК - Ремиконт-110. Помимо специальных контроллеров в качестве средств ввода-вывода аналоговой и дискретной информации пакет RealFlex позволяет использовать промышленные компьютеры MicroPC фирмы Octagon Systems с записанной в ПЗУ или флэш-память операционной системой QNX 2.21 или QNX 4.22.

Пакет рекомендуется для применений в АСУТП с большим числом входных параметров от 5000 до нескольких десятков тысяч и для распределенных систем, в которых должны работать одновременно несколько операторов.

Существуют и другие пакеты, работающие с QNX, сведения по которым представлены в таблице 14.

Таблица 14

| Название | Версия QNX | IBM/PC | ЛВС | Объем ОЗУ |
|--------------|------------|-------------|--------------------------------|-----------|
| RealFlex 1.3 | 2 | 286-Pentium | Arcnet | 8Мбайт |
| RealFlex 4 | 4 | 386-Pentum | Arcnet,Ethern et,Token Ring | 12 Мбайт |
| Sitex | 4 | 386 | то же | 8 Мбайт |
| Qdiagem | 4 | 286-Pentium | тоже | 8 Мбайт |
| RESY-USS | 4 | 386-Pentium | Ethernet, TC/IP | 16 Мбайт |

Фирма Ad Astra представила на рынок графическую SCADA - систему для разработки АСУТП верхнего уровня и программирования контроллеров MicroPC - TRACE MODE.

Разработка программ в ней основана на объектной технологии. Программа имеет набор векторных редакторов, средства рисования, включает объемные векторные изображения. Представляет возможности построения крупных распределенных АСУ ТП,

включающие до 200 операторных станций на базе ведущих ОС с единым сетевым временем, имеет встроенную поддержку плат ввода-вывода фирм Advantech, Octagon Systems.

TRACE MODE реализует функции и имеет характеристики:

- опроса 4096 аналоговых и 98000 дискретных входов;
- минимальный период обмена с УСО - 0,055 с.;
- встроенная поддержка наиболее распространенных российских и зарубежных контроллеров через 16 последовательных портов;
- алгоритмы дискретного и аналогового управления, в том числе П, ПИ, ПИД законы, ШИМ аналоговых переменных;
- объемное изображение технологических объектов;
- связь с СУБД и электронными таблицами;
- работает под MS DOS, Windows 3.1., Windows 95, Windows NT.

Фирма ICONICS является одним из лидеров в создании ПО для верхнего уровня на базе IBM PC совместимых компьютеров и предлагает программные SCADA продукты: Genesis for Windows (GFW), AlarmWorX+, TrenWorX+, работающих в среде Windows.

Основные возможности GFW:

- модульная структура, основанная на принципе “клиент-сервер“, позволяющая создавать из отдельных полностью независимых составных частей пакета тот интерфейс оператора, который необходим для решения задачи;
- встроенный язык программирования, совместимый с VBA (Visual Basic for Applications) фирмы Microsoft;
- для непрограммистов предусмотрен специальный инструмент разработки программ - Script Wizard(Мастер Скриптов), позволяющий программировать задачи простым выбором инструкций;
- работа в реальном масштабе времени под управлением сердца пакета GFW сервера реального времени (Real Time Server-RTS), который выполняет функции многозадачного ядра реального времени и отвечает за отсутствие потерь в потоке данных от аппаратуры нижнего уровня АСУТП.

GFW имеет драйверы к более 250 из производимых в мире контроллеров и систем ввода-вывода различных фирм.

В России дистрибьютером выступает фирма PROSOFT.

8. ПРОМЫШЛЕННЫЕ РОБОТЫ

Эксперимент, в основу которого был положен дельфи-метод,

по опросу более 150 ученых-экспертов, представляющих различные теоретические и прикладные науки, показал, что по мнению ученых, отвечавших на вопрос “что такое робот, чем он отличается от автоматических машин и линий ?” можно сформировать следующую концепцию:

“ Робот представляет собой подвижную компактную систему, отличительными признаками которой являются: чувствительные элементы, манипуляторы и некоторая степень искусственного интеллекта, отождествленная со способностью к обучению и изменению поведения.

Идея создания человекоподобных механизмов пришла из глубокой древности. Вся предистория развития роботов может быть условно разбита на 3 этапа: мечты и легенды, первые андройды и кибернетические игрушки и роботы нашего времени.

Аристотель упоминает о статуе Венеры, которая могла передвигаться благодаря какому-то устройству, помещенному внутри нее и приводимому в действие с помощью ртути. Одним из первых андройдов создал согласно легенде алхимик Альберт Великий (1193-1281 гг.). Его андройд представлял куклу в рост человека, которая вставала с места, когда стучали в дверь, шла открывать их и даже заговаривала с входящим. Описание устройства не найдено, т.к. согласно легенде оно было разбито на куски одним из учеников Альберта философом Фомой Аквинским.

Создание технических устройств в то время было затруднено из-за отсутствия соответствующих технических средств.

В средние века создавались в большом количестве человекоподобные механизмы - андройды. Первые андройды изготовили швейцарский часовщик Пьер-Жак Дро и его сын Анри Дро в 1774 г. Они показали сконструированного механического писца, который макал перо в чернильницу и писал, двигая при этом головой и оглядывая написанное. Закончив писать, он посыпал бумагу песком и встряхивал ее. В центре андройда был установлен барабан с набором кулачков, в качестве источника энергии использовались - пружины. Считают, что слово “андройд” создано из первых букв имени и фамилии этих людей. Все андройды были произведениями механиков-умельцев. Конец 19 и 20 века характеризуется, как известно, выдающимися открытиями в области науки и техники, что привело к созданию в этом веке большого количества антропоморфных (човекоподобных) устройств - в том числе и промышленных роботов.

Промышленные роботы условно подразделяют на:

- ручные манипуляторы (экзоскелетоны);
- промышленные роботы, работающие по жесткой программе технологического цикла;
- промышленные роботы с перепрограммируемым циклом операций;
- промышленные роботы, программируемые методом обучения;
- промышленные роботы с числовым программным управлением;
- адаптивные промышленные роботы;
- промышленные роботы с элементами искусственного интеллекта.

Функциональная схема промышленного робота представлена на рис.16.

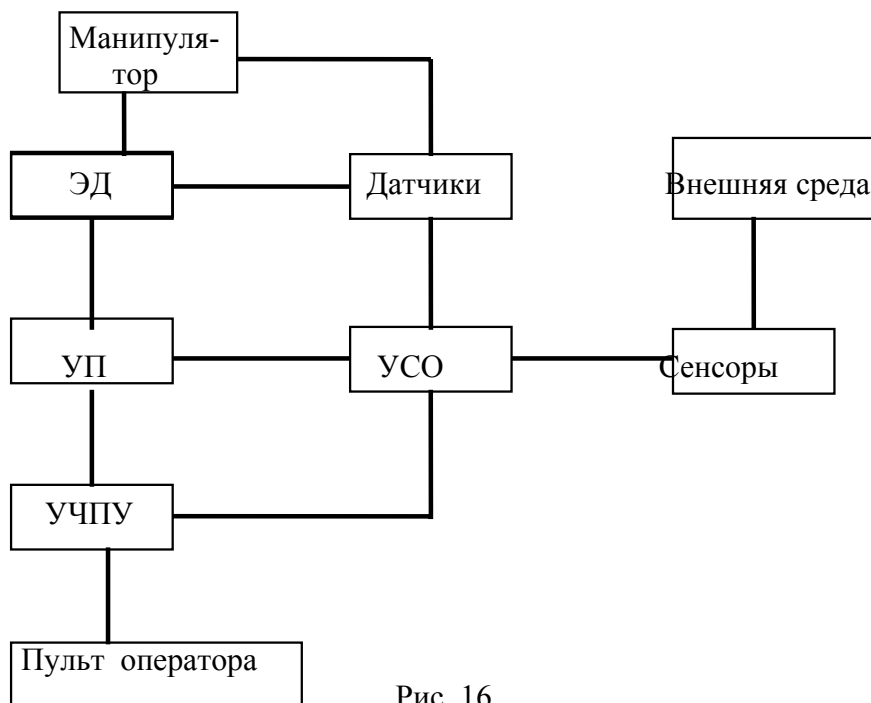


Рис. 16.

В настоящее время к промышленным роботам относят автоматические устройства, которые способны выполнять разнообразные производственные операции, оснащенные манипулятором и управляемые от специализированного устройства управления или ЭВМ. Объектами роботизации, как правило, являются - высокоточные, быстрые, тяжелые, монотонные операции, непрерывные технологические процессы.

Промышленный робот - автоматическая машина, представляющая совокупность манипулятора и перепрограммируемого устройства

управления для выполнения в производственном процессе двигательных и управляющих функций человека при перемещении предметов производства и (или) технологической оснастки.

Адаптивный робот - устройство, способное программно организовать свою деятельность в соответствии с внешними обстоятельствами, которые анализируются с помощью технических средств, соответствующих по функциям органам чувств человека.

Интеллектуальный робот - устройство, способное при перемещении в пространстве, справляться с задачами анализа сцен и распознавания образов, умеющее анализировать обстановку с помощью датчиков обратной связи и сенсорных устройств, а также прогнозировать ситуации, опираясь на собственный опыт и доступную информацию.

Манипулятор - дистанционно управляемое устройство, представляющее собой разомкнутую кинематическую цепь, оснащенное рабочим органом и приводами для выполнения двигательных функций, заменяющих аналогичные действия человека при перемещении предметов. Движение звеньев в сочленениях осуществляется при помощи специальных исполнительных электроприводов, являющихся аналогами мышц руки человека. Рука человека обладает большой подвижностью и может совершать более 20 различных независимых движений. Для манипулятора 6 движений оказывается достаточным для осуществления переноса предмета в любую точку рабочего (сервисного) пространства и ориентации его относительно трех осей декартовой системы координат. Для приведения в действие захвата требуется еще одна степень подвижности.

Таким образом, универсальным является манипулятор, рабочий орган которого имеет не менее 7 степеней подвижности: три - транспортирующие; три - ориентирующие и одна, направленная на удержание предмета.

Кинематические схемы большинства манипуляторов промышленных роботов представляют собой разомкнутую кинематическую цепь, состоящую из последовательного соединения кинематических пар, как правило, пятого класса, работающих на изгиб, вращение, поступательное перемещение. Кинематическая пара 5 класса имеет одну степень подвижности, 4 класса - две и т.д. Каждое звено пространственного механизма (абсолютно твердое тело) имеет 6 степеней подвижности. Отдельные звенья объединяются в кинематические пары. Минимальное число обобщенных координат

звена, полностью определяющее положение и возможные направления его движения, называют числом степеней подвижности звена W :

$$W = 6 - L,$$

где L - число условий связи или ограничений.

Для неподвижного соединения $L=6$, для кинематической пары 5 класса - 5.

Кинематические звенья манипулятора образуют манипуляционную систему. Число степеней подвижности манипуляционной системы производственного оборудования определяет пространственное положение всех звеньев и определяется по формуле **Сомова-Малышева**:

$$W = 6 \times n - 5 \times P_5 - 4 \times P_4 - 3 \times P_3 - 2 \times P_2 - P_1,$$

где n -число звеньев манипуляционной системы, $P_5 - P_1$ - кинематические пары соответствующих классов.

УЧПУ - реализует функции:

- выполнения алгоритмов циклового, позиционного и контурного управления движением промышленного робота,
- хранения информации о требуемой траектории движения,
- синхронизации работы всей манипуляционной системы с внешними обслуживающими робот устройствами, используя информацию от датчиков исполнительных механизмов и сенсорных датчиков среды.

Структура и состав УЧПУ разнообразен и определяется составом технологического оборудования, но, как правило, использует структуры CNC-типа.

Цикловые управляющие устройства промышленных роботов реализуют позиционный по упорам способ управления, работают по принципу конечного автомата. К этим устройствам можно отнести устройство ЭЦПУ - 6030 промышленного робота МП-9С, устройство циклового манипулирования УЦМ-663, УЦМ-100 и др.

Позиционные управляющие устройства промышленных роботов обеспечивают выполнение алгоритмов позиционного управления РТР (Point to Point- от точки к точке). При реализации этих алгоритмов задается требуемое количество узловых точек траектории, через которые обязательно должна пройти рука манипулятора. К числу таких устройств можно отнести: промышленный робот НЦТМ, устройства УПМ-331 с шаговыми электроприводами по каждой из трех степеней подвижности, УПМ-552, УПМ- 772 со следящими электроприводами и 5 и 7 степенями подвижности соответственно.

Контурные управляющие устройства - обеспечивают выполнения алгоритмов контурного управления СР (Continuous Path-непрерывный путь). К числу таких устройств можно отнести устройства УКМ-552, УКМ-772 [19].

8.1. Алгоритмы позиционного управления.

Простейший алгоритм позиционного управления предусматривает сведение к минимально возможному отклонения

$$X_{i,l} - X_{i,lp} \leq \alpha \quad \text{по каждой степени подвижности,}$$

где $X_{i,lp}$ – программное задание на перемещение по координате.

При этом в момент достижения программного значения по какой-либо степени подвижности соответствующая скорость равна нулю, а точность позиционирования - (ошибка позиционирования) зависит от реализованной в УЧПУ минимальной дискретности перемещения h_x, h_y, h_z и разрешающей способности используемого датчика перемещения.

В простейшем случае движение манипулятора по каждой степени подвижности происходит независимо или с одновременным стартом, что определяется аппаратными и программными средствами реализации алгоритма. Более сложные алгоритмы позиционного управления сводятся к формированию требуемой тахограммы движения по каждой степени подвижности. Ниже приведены соотношения, которые могут быть использованы для расчета типового процесса движения для вращательной степени подвижности с трапециидальной тахограммой движения.

Типовой процесс (как показано на рис.17.) состоит из трех фаз движения

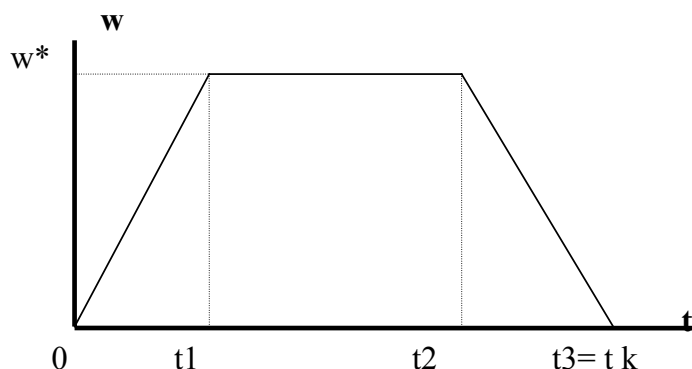


Рис. 17.

- разгона с ускорением e_1 ; $w(t) = e_1 \times t, 0 \leq t \leq t_1$,
 - движения с постоянной скоростью w^* ; $w(t) = w^*$, $t_1 < t < t_2$
 - торможения с ускорением e_2 , $w(t) = w - e_2 \times (t - t_2), t_2 \leq t \leq t_k$,
- где t_k - время отработки кадра.

Полное приращение угла за время отработки кадра:

$$\Delta\varphi = \int_0^{t1} e1 \times t \times dt + \int_{t1}^{t2} w \times dt + \int_{t2}^{tk} [w - e2 \times (t - t2)] dt =$$

$$= e1 \times t1^2 / 2 + w \times (tk - t1) - e2 \times (tk - t2)^2 / 2.$$

С другой стороны, из графика скорости можно получить:

$$\Delta\varphi = w^2 / (2 \times e1) + w \times (t2 - t1) + w^2 / (2 \times e2).$$

Отсюда можно определить требуемое время движения с постоянной скоростью:

$$t2 - t1 = \Delta\varphi / w - w \times (1 / e1 + 1 / e2) / 2. \quad (**)$$

В процессе программирования необходимо знать интервалы времени $t1, t2, tk$.

Если при заданных $\Delta\varphi, w, e1, e2$ из (**) получается, что $(t2 - t1) > 0$, то

$t1 = w / e1$ - время движения с постоянным ускорением,
 $t2 = t1 + \Delta\varphi / w - w / 2 \times (1 / e1 + 1 / e2) = \Delta\varphi / w + w / 2 \times (1 / e1 - 1 / e2)$ - время движения с постоянным ускорением и постоянной скоростью,

$$t3 = tk = t2 + w / e2 = \Delta\varphi / w + w / 2 \times (1 / e1 + 1 / e2).$$

Если при заданных $\Delta\varphi, w, e1, e2$ из (**) получается, что $(t2 - t1) < 0$, то

$$t1 = t2 = \sqrt{(2 \times \Delta\varphi \times e2) / ((e1 \times (e1 + e2)))}$$

$$tk = \sqrt{(2 \times \Delta\varphi \times e1) / ((e2 / (e1 + e2)))} + \sqrt{(2 \times \Delta\varphi \times e2) / ((e1 \times (e1 + e2)))}.$$

В этом случае фаза движения с постоянной скоростью отсутствует, т.к. максимальная скорость, которая достигается в процессе движения меньше w^* :

$$w^* = \sqrt{2 \times \Delta\varphi \times e1 \times e2 / (e1 + e2)}.$$

8.2. Алгоритмы контурного управления.

При управлении манипулятором по непрерывной траектории необходимо обеспечивать непрерывную и синхронную обработку отдельных координат.

Существует два основных способа построения контурных управляющих устройств, отличающихся различным соотношением

между запоминающей и вычислительной частью устройства управления.

Первый способ основан на записи информации о требуемом положении по каждой координате в память, что требует запоминающих устройств с большим объемом памяти.

Второй способ заключается в записи информации о положении в виде конечного числа узловых точек траектории и расчете непрерывной траектории между этими точками путем интерполяции по определенному алгоритму. В этом случае объем запоминающего устройства может быть невелик, но необходимо иметь аппаратный или программный блок для расчета промежуточных участков траектории - интерполятор.

Интерполятор - устройство, на вход которого кадр за кадром подается информация в виде цифровых кодов, а на выходе выдается информация для каждой координаты в виде унитарного кода, т.е. последовательности импульсов.

Решение задачи контурного управления разбивается обычно на этапы:

- подготовка исходной информации о требуемой траектории, которая включает аппроксимацию траектории заданным набором функций;
- ввод информации в систему программного управления;
- расчет заданных значений координат, расположенных на траектории движения, с использованием выбранного метода интерполяции;
- расчет числа импульсов по каждой из координат и выдача управляющих воздействий на исполнительные приводы с требуемой частотой, которая определяет контурную скорость движения по каждой из координат.

Интерполяторы по способу реализации подразделяются на:

- аппаратные;
- программные.

По виду интерполируемой траектории движения интерполяторы делятся на:

- линейные;
- нелинейные (второго порядка - круговые, параболические, n-порядка).

В основном в системах ЧПУ применяются линейные и круговые интерполяторы, т.к. до 90 % траекторий могут быть с достаточной степенью точности представлены совокупностью отрезков прямых и дуг окружности.

Существуют различные алгоритмы интерполяции реального

времени, которые условно можно разделить на две группы:

- алгоритмы единичных приращений (метод оценочной функции, метод цифро-дифференциальных анализаторов);
- алгоритмы равных времен (метод цифрового интегрирования, прогноза и коррекции, итерационно-табличные методы).

В первых, определяются моменты времени, необходимые для выдачи единичных приращений по одной или нескольким координатам.

Во вторых рассчитываются координаты точек траектории, через определенные и равные промежутки времени, по истечении которых выдается требуемое количество импульсов на привода исполнительного механизма.

Практически интерполяцию организуют следующим образом.

В результате очередного вычислительного цикла, выполняемого с максимально высокой скоростью в машинном масштабе времени, определяют в какие приводы подачи должны быть выданы дискреты на текущем этапе оперативного управления. Результат сохраняют в буфере, который опрашивают с частотой, соответствующей скорости подачи для ведущей координаты. Таким образом, расчеты машинного масштаба привязывают к реальному времени.

Принципы построения интерполяторов.

Пусть необходимо воспроизвести произвольную кривую второго порядка на плоскости XU . Каноническое уравнение кривой второго порядка имеет вид [20]:

$$A \times x^2 + 2 \times B \times x \times y + C \times y^2 + 2 \times D \times x + 2 \times E \times y + F = 0,$$

где A, B, C, D, E, F - вещественные числа.

Продифференцируем уравнение по x и разрешим относительно dy/dx

$$dy/dx = -\frac{A \times x + B \times y + D}{C \times y + B \times x + E}.$$

Для перехода к координатам времени и скорости отработки траектории запишем уравнение в параметрической форме:

$$dx/dt = w \times (B \times x + C \times y + E)$$

$$dy/dt = -w \times (A \times x + B \times y + D),$$

где w - число, знак которого определяет направление, а значение - скорость движения по траектории.

На рис.18 приведена структурная схема цифрового

дифференциального анализатора (ЦДА), предназначенного для воспроизведения любой кривой второго порядка со скоростью, пропорциональной w .

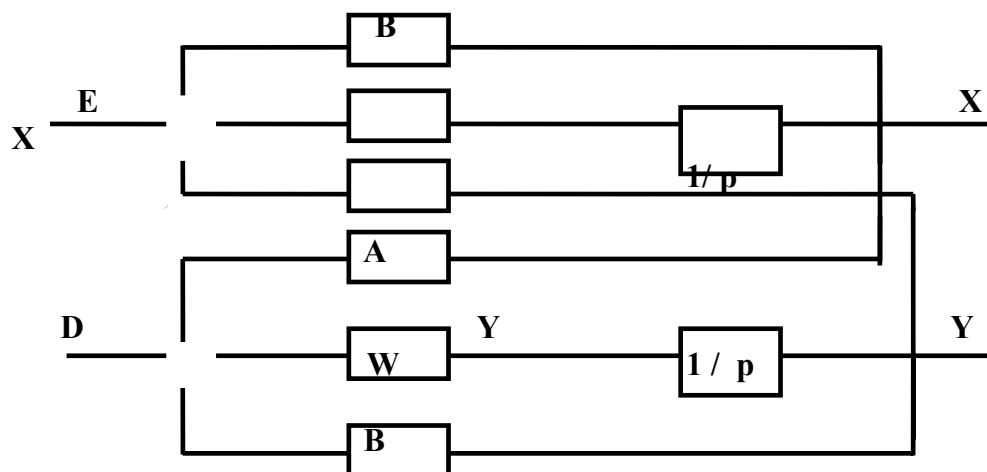


Рис.18

Известно, что последние уравнения вырождаются в уравнения эллипса

при $A \times C > B^2$, параболы при $A \times C = B^2$, гиперболы при $A \times C < B^2$,

окружности при $B=0, A=C=1, D = -x \text{ с}, E = -y \text{ с}$.

$F = Xc^2 + Yc^2 - R^2$, где $x \text{ с}, y \text{ с}$ - координаты центра, R - радиус окружности.

Воспроизведение прямой по полученным уравнениям состоит в реализации интерполятора на базе двух самостоятельных интеграторов по координатам X, Y соответственно ($A=B=C=0, E=x_k - x_0, D=y_k - y_0$),

где x_k, x_0, y_k, y_0 - начальные и конечные координаты перемещений в кадре.

Если V - постоянная скорость движения по прямой, а L - длина отрезка прямой, то при линейной интерполяции

$$w = V / L = 1 / Tk,$$

где Tk - время отработки кадра, т.е. время отработки соответствующего перемещения на участке интерполяции.

Можно также показать, что для уравнения окружности при круговой интерполяции w - угловая скорость движения точки окружности радиуса $R \text{ с}$

контурной скоростью $V = \sqrt{V_x^2 + V_y^2}$,

где $V_x = V / R \times (y - y_c)$
 $V_y = -V / R \times (x - x_c).$

Из рис.18 видно, что при круговой интерполяции структурная схема состоит из двух интеграторов, охваченных перекрестными связями.

ИНТЕРПОЛЯЦИЯ ПО МЕТОДУ ЦДА.

Все виды интерполяторов, независимо от алгоритмов их работы и способа реализации, функционально моделируют процесс движения изображающей точки по заданной траектории. Технические средства, при помощи которых решаются дифференциальные уравнения в аналоговой форме - называют АВМ (аналоговыми вычислительными машинами), а в дискретной - цифровыми дифференциальными анализаторами (ЦДА). Основой АВМ являются операционные усилители, а ЦДА - интеграторы. Цифровой интегратор может быть построен по схеме параллельного или последовательного переноса [4].

На рис.19 представлена упрощенная схема линейного интерполятора.

Из блока задания программы (БП) информация о приращениях по координатам в заданном кадре поступает в соответствующие регистры, а о скорости перемещения на управляемый делитель частоты (УДЧ).

Регистры соединены со счетчиками X, Y через вентили И. Количество разрядов n в регистрах и счетчиках определяется максимальным приращением координат и может быть найдено из формулы:

$$n > \lg(Nm + 1) / \lg 2,$$

где N m - максимальное число, которое записывается в регистр.

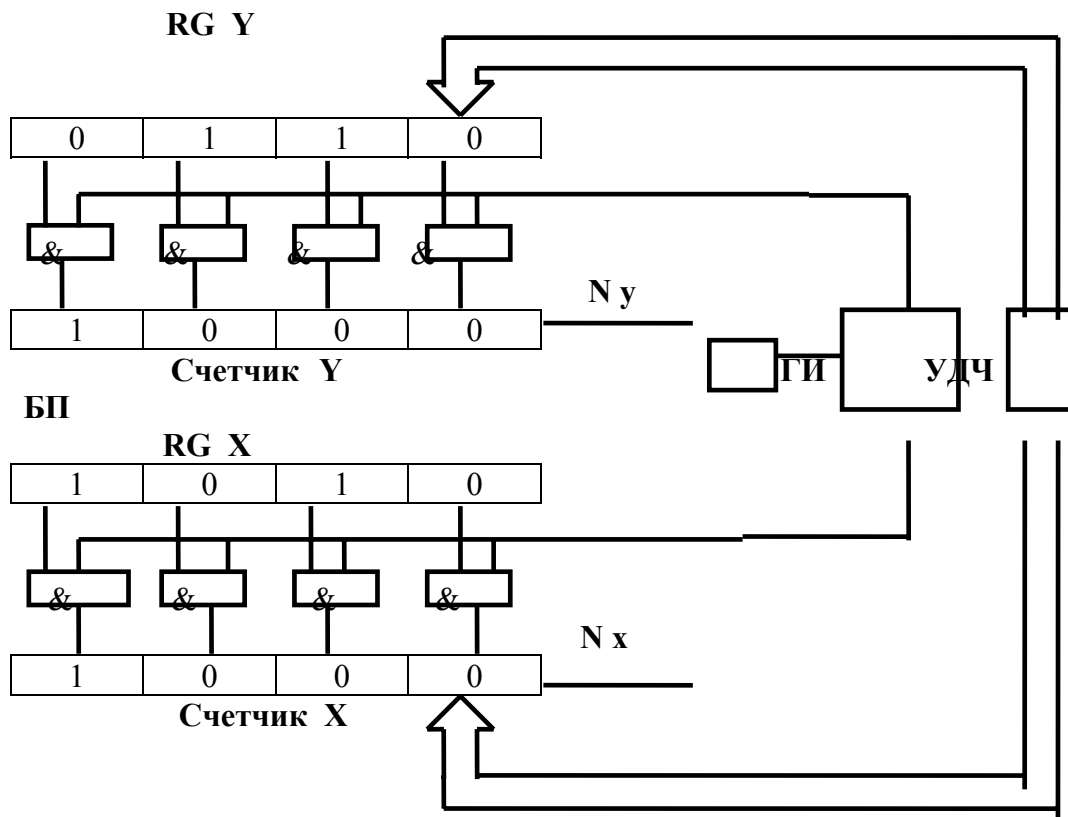


Рис.19.

Пусть приращения координат в кадре составляют по $X=1$ мм, $Y=0,6$ мм,
 при шаге на один импульс $h_x=h_y=0,1$ мм.
 Величина перемещения по координатам в элементарных шагах
 (например, шагового электропривода) $X=1010=RG_x$,

13

рїетЃй□Б`1]`□5□\$ФпћдЃБ□ЮСе□□ю1□фЃѓ□фф65фІ6Г□drз□Фе
йю\$ллфц□ћ□□П□dцйЕ□Ф=5р□гйю□55 \$qq□□Л1□ЃгЛь□и□ю
бћфтЕлР□иБцжия счета.

Импульсы от УДЧ проходят в счетчик тогда, когда на обоих входах схем совпадения 1, т.е. при поступлении каждого тактового импульса от УДЧ к числу, имеющемуся в счетчике, добавляется число, записанное в регистре, т.е. в счетчик X добавляется двоичное число 1010, а в Y - 0110.

При переполнении счетчиков появляются импульсы, подаваемые с выхода интерполятора на исполнительные приводы X и Y.

В таблице 15 представлены состояния интерполятора, а на рис.20 представлен вид получаемой линейной траектории.

Таблица 15

| Номер тактового импульса | X состояние счетчика +1010 | X сигнал на привод | Y состояние счетчика +0110 | Y сигнал на привод |
|--------------------------|----------------------------|--------------------|----------------------------|--------------------|
| 0 | 1000 | 0 | 1000 | 0 |
| 1 | 0010 | 1 | 1110 | 0 |
| 2 | 1100 | 0 | 0100 | 1 |
| 3 | 0110 | 1 | 1010 | 0 |
| 4 | 0000 | 1 | 0000 | 1 |
| 5 | 1010 | 0 | 0110 | 0 |
| 6 | 0100 | 1 | 1100 | 0 |
| 7 | 1110 | 0 | 0010 | 1 |
| 8 | 1000 | 1 | 1000 | 0 |
| 9 | 0010 | 1 | 1110 | 0 |
| 10 | 1100 | 0 | 0100 | 1 |
| 11 | 0110 | 1 | 1010 | 0 |
| 12 | 0000 | 1 | 0000 | 1 |
| 13 | 1010 | 0 | 0110 | 0 |
| 14 | 0100 | 1 | 1100 | 0 |
| 15 | 1110 | 0 | 0010 | 1 |
| 16 | 1000 | 1 | 1000 | 0 |

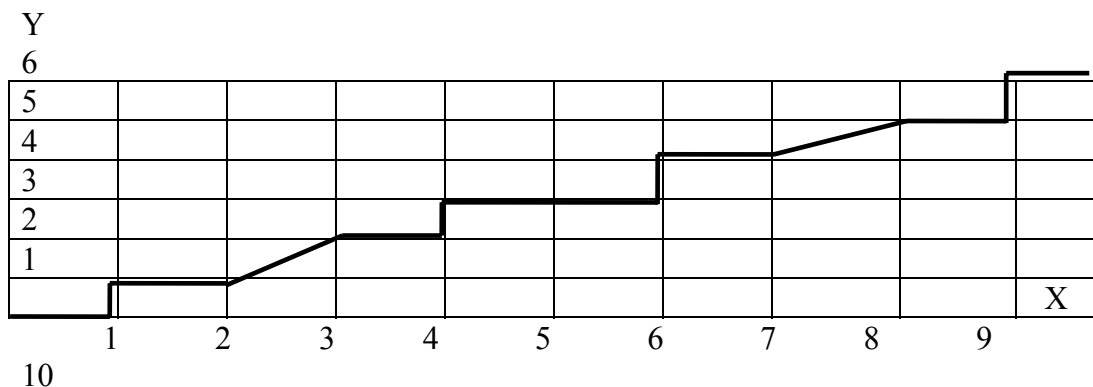


Рис.20.

Таким образом, интерполяторы ЦДА строятся с дискретными интеграторами, импульсы переполнения регистров N_x , N_y в виде унитарного кода используются для управления электроприводами координат, ошибка в обработке траектории не превышает цены шага по координате h_x , h_y .

При осуществлении круговой интерполяции можно использовать описанный выше алгоритм, предварительно разбив дугу окружности на отрезки прямых. Чем больше таких отрезков, тем меньше погрешность обработки траектории. Однако, большое число отрезков означает более трудоемкое программирование, возрастает объем программы. Ограничивающим фактором при аппроксимации служит допустимая погрешность обработки траектории - q . При аппроксимации окружности N - равномерными участками прямых центральный угол i - участка можно найти из соотношения:

$$\alpha_i = \varphi / 2 + (i - 1) \times \varphi = \varphi \times (i - 1 / 2),$$

где $\varphi = \pi / (2 \times N)$ - приращение угла на участке аппроксимации.

Длина отрезка прямой L и погрешность q (при $q \ll R$) связаны соотношением:

$L = \sqrt{8 \times R \times q}$, с другой стороны $L = 2 \times R \times \sin(\varphi / 2)$, что позволяет определить приращения координат на каждом i - участке аппроксимации:

$$x_i = L \times \sin(\varphi \times (i - 1 / 2))$$

$$y_i = L \times \cos(\varphi \times (i - 1 / 2)).$$

Скорость движения на траектории V_i определяется требованиями технологического цикла. Время обработки кадра:

$$t_i = t_{i+1} - t_i = \sqrt{(x_i^2 + y_i^2)} / V_i.$$

Скорости движения по координатам:

$$V_{xi} = \frac{Xi}{ti} = Xi \times Vi / \sqrt{Xi^2 + Yi^2};$$

$$V_{yi} = Yi / ti = Yi \times Vi / \sqrt{Xi^2 + Yi^2}$$

За время i - цикла по координате X должно быть выдано $Nxi = \frac{Xi}{hx}$ импульсов с частотой $fxi = \frac{Nxi}{\Delta ti}$, а по координате Y - $Nyi = \frac{Yi}{hy}$ с частотой $f_{yi} = \frac{Nyi}{\Delta ti}$.

Если приращения координат подаются на вход интерполятора в двоичном коде, то можно записать:

$$\frac{fx,i}{fy,i} = \frac{An,x \times 2^n + An-1,x \times 2^{n-1} + \dots + A1,x \times 2^1 + A0,x \times 2^0}{An,y \times 2^n + An-1,y \times 2^{n-1} + \dots + A1,y \times 2^1 + A0,y \times 2^0} = \frac{Kx}{Ky}$$

K_x, K_y - кодовые комбинации приращений по координатам в i - такте.

В рассматриваемом случае, для воспроизведения отрезка прямой достаточно, чтобы отношение частот следования импульсов унитарных кодов было равно отношению соответствующих приращений координат.

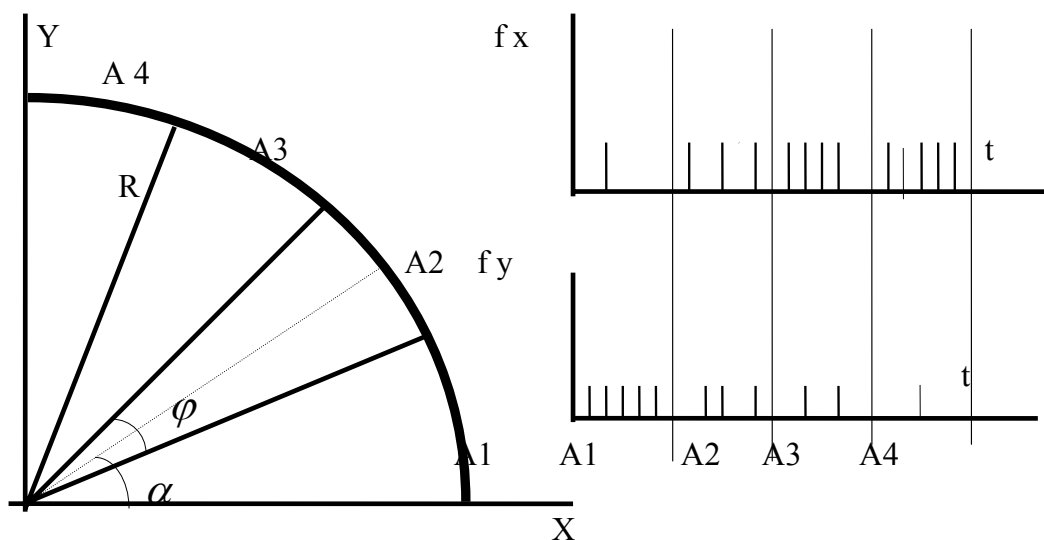


Рис .21.

ИНТЕРПОЛЯЦИЯ ПО МЕТОДУ ОЦЕНОЧНОЙ ФУНКЦИИ

Согласно методу оценочной функции при интерполяции моделируется алгебраическое уравнение воспроизводимой кривой.

Сущность метода в том, что по результатам шага вдоль какой-либо управляемой координаты, вычисляется оценочная функция F , знак которой определяет направление следующего шага, причем перемещение, возникающее в результате этого шага, приближает обрабатываемую траекторию к идеальной кривой [4, 8].

При линейной интерполяции по методу оценочной функции сама функция вводится в виде:

$$Fi = yi \times X + xi \times Y,$$

где xi , yi - координаты текущей точки интерполяционной кривой,

X , Y - приращения по соответствующим координатам в обрабатываемом кадре интерполяции.

Применяют либо обычный, либо усовершенствованный алгоритм интерполяции.

Обычный алгоритм. Если сделан шаг по оси X , то оценочная функция вычисляется с использованием соотношения:

$$Fi+1 = yi \times X - (xi+1) \times Y = yi \times X - xi \times Y - Y = Fi - Y.$$

Если сделан шаг по оси Y , аналогично получим:

$$Fi+1 = (yi+1) \times X - xi \times Y = Fi + Y.$$

Вычисления нового значения оценочной функции опираются на сохраняемое предыдущее значение.

Усовершенствованный алгоритм позволяет увеличить быстродействие интерполяции, за счет одновременного управления координатами.

Если сделан шаг по оси X и Y новое значение оценочной функции определяется как:

$$Fi+1 = (yi+1) \times X + (xi+1) \times Y = Fi + X - Y.$$

Использование фиксированного цикла с расчетом одного шага интерполяции в пределах каждого цикла предъявляет высокие требования к быстродействию вычислителя или накладывает ограничения на скоростные возможности объекта управления.

Рассмотрим пример реализации линейной интерполяции с кадровыми перемещениями $X=10$, $Y=6$ и использованием усовершенствованного метода.

Примем следующее условие: если точка интерполяционной кривой находится ниже идеальной прямой, то на следующем шаге осуществляется перемещение по оси X и Y ($Fi < 0$), если выше или на ней, то по оси X

$(F_i > 0)$.

Данные расчетов представлены в таблице 16, на рис.22 представлена интерполяционная кривая.

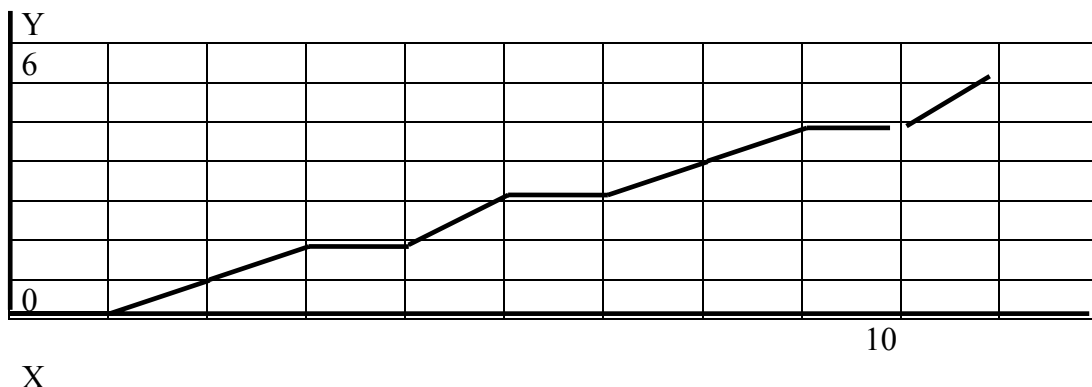


Рис . 22 .

При круговой интерполяции по методу оценочной функции вводится оценочная функция вида:

$$Fi = xi^2 + yi^2 - R^2.$$

Для интерполирования дуги окружности задается номер квадранта и координаты начальной и конечной точек дуги. Для воспроизведения дуги окружности в положительном направлении (против часовой стрелки) в первом квадранте осуществляется следующий алгоритм управления.

1. Если оценочная функция $Fi > 0$, то очередной шаг делается вдоль отрицательного направления оси X, т.е. $xi+1 = xi - 1$.
2. Если $Fi < 0$, то шаг делается в положительном направлении Y, т.е.
 $yi+1 = yi + 1$.

3. После очередного шага по оси X новое значение оценочной функции:

$$Fi+1 = (xi - 1)^2 + yi^2 - R^2 = Fi - 2 \times xi + 1.$$

4. После шага по оси Y новое значение оценочной функции:

$$Fi+1 = xi^2 + (yi + 1)^2 - R^2 = Fi + 2 \times yi + 1.$$

Для повышения контурной скорости применяют усовершенствованный алгоритм, при котором, как и при линейной интерполяции, по ведущей координате приращения выдаются на каждом шаге.

Ведущей координатой является та, которая требует большего числа управляющих импульсов на соответствующем такте расчета.

Если импульсы выдаются одновременно по обеим координатам, то оценочная функция определится следующим образом:

$$Fi+1 = (xi - 1)^2 + (yi + 1)^2 - R^2 = Fi - 2 \times xi + 2 \times yi + 2.$$

Отметим следующее обстоятельство: при движении по дуге окружности в пределах от нуля до $\pi/4$ ведущей является координата - Y, а далее до $\pi/2$ координата X.

В таблице 17 представлен расчет траектории для воспроизведения дуги окружности в первом квадранте ($R=5$, $x_0=5$, $y_0=0$, $x_k=0$, $y_k=5$), а на рис.23 представлен вид интерполяционной кривой при простом алгоритме.